Tableau Desktop and Web Authoring Help

Version 2021.2; Last Updated 4/20/2023

Contents

What's New in Tableau Desktop and Web Authoring

Browse summaries of new features for the current release.

Tableau Desktop Release Notes | Known Issues

New in version 2021.2

Connect to and Prepare Data

Design Views and Analyze Data

Tableau Desktop Features Now in Tableau Online and Tableau Server

- **Create any action in Tableau Online and Tableau Server, including URL actions** on page 10
- **Condition and Top/Bottom filters now available on the web** on page 10
- **Create parameter actions on the web** on page 9
- **Create set actions on the web** on page 9

Unique Features in Tableau Online and Tableau Server

- **Organize items in a collection** on page 11
- **Open and edit workbooks in Tableau Desktop from the web** on page 11
- **Include a target value for your metric** on page 11

Add-ons

- **Tableau Catalog - part of the Data Management Add-on** on page 12

Connect to and Prepare Data

Connect to Datorama and Dremio data with new partner-built connectors

Tableau Desktop now includes two new partner-built connectors:

- Datorama by Salesforce lets you connect Tableau to your Datorama data.
- Dremio by Dremio lets you connect to your Dremio data.

Scan entire file to build schema for large JSON files

If you have a very large JSON file (more than 10,000 rows), you can now use the new “Scan Entire Document” option to pre-scan all data and build your schema. The speedier option of selecting schema levels without pre-scanning is also available.

Multiple analytics extensions in a site

Beginning with version 2021.2, you can now have multiple analytics extensions connections for each site. This means that organizations with different language or connection requirements can use analytics extensions on the same Tableau Server site, with different workbooks using different connections. For details on using analytics extensions, see **Pass Expressions with Analytics Extensions** on page 2690. For more information on configuring analytics extensions for a site, see Configure Connection with Analytics Extensions (Tableau Online Help) or Configure Connection with Analytics Extensions (Tableau Server Help).

Note: As of 2021.2, Einstein Discovery analytics extensions are still limited to one per site.

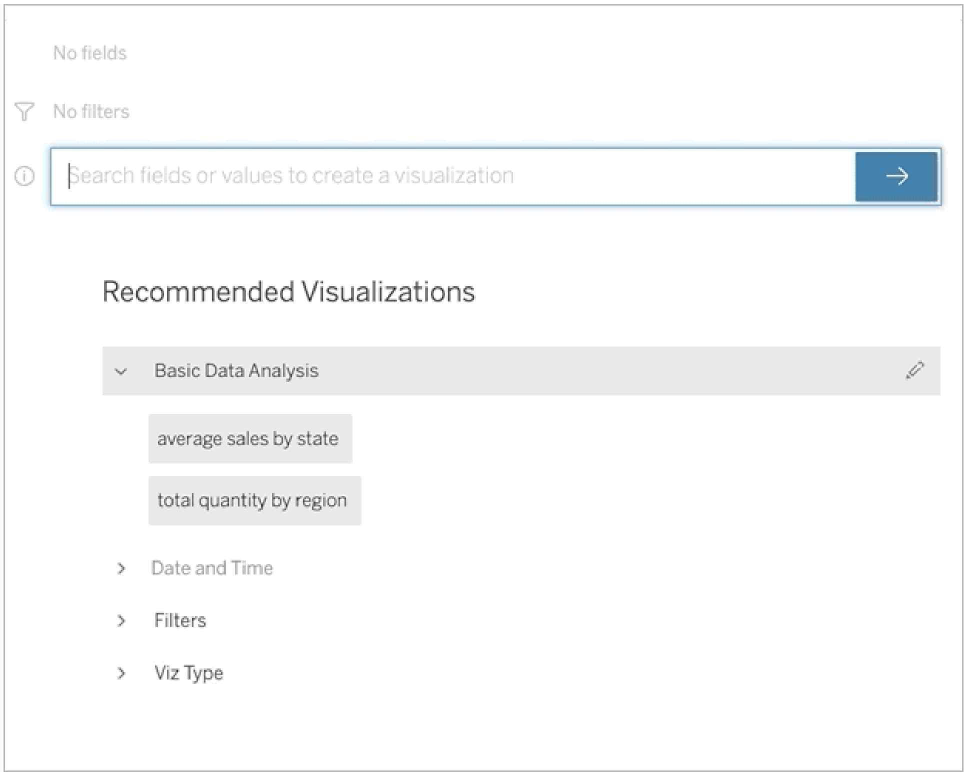
Design Views and Analyze Data

Ask Data now available to Viewers

Users with the Viewer license make up the largest number of people analyzing data with Tableau, and now they can ask questions of data in an intuitive, search-based style just like Creators and Explorers. For more information, see **Automatically Build Views with Ask Data** on page 1109.

Guided search in Ask Data

A revamped Ask Data search experience shows relevant fields, values, and synonyms as users type entries. The process quickly guides users who have little understanding of the data structure, helping them to easily create effective visualizations.



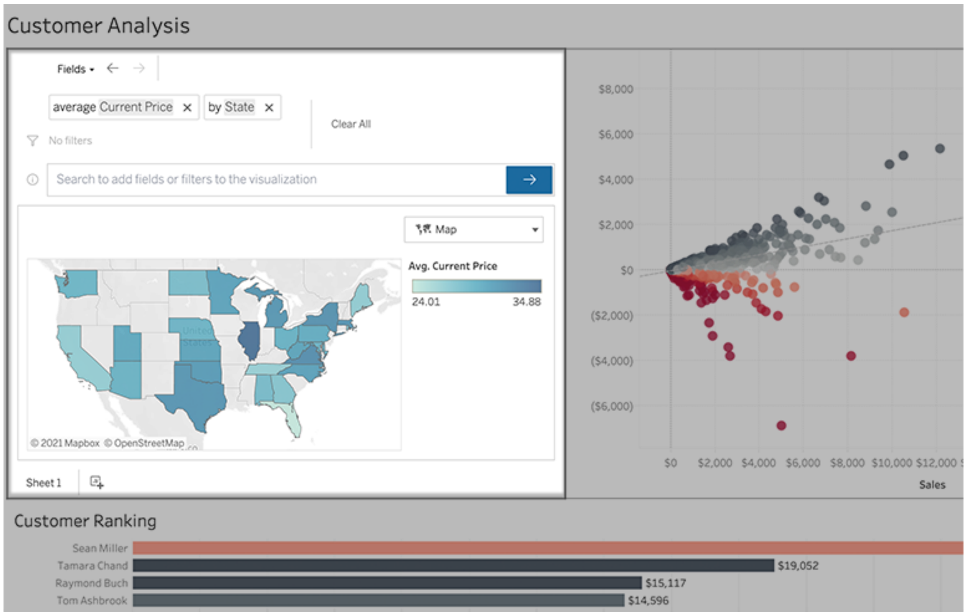
Use lenses that customize Ask Data for specific audiences

Most people don’t need information from an entire data source but instead want data visualizations relevant to their job function, like sales, marketing, or support. To optimize Ask Data for different audiences like these, Tableau Creators and Explorers now author separate Ask Data *lenses*, which query a selected subset of fields. (Previously, only data source owners were able to curate data for Ask Data.)

For the selected fields, authors can specify synonyms for field names and values, reflecting terms the lens audience uses in common language (for example, "SF" for "San Francisco"). Lens authors then customize the recommended visualizations that appear below the Ask Data query box, which provide answers to users with a single click.

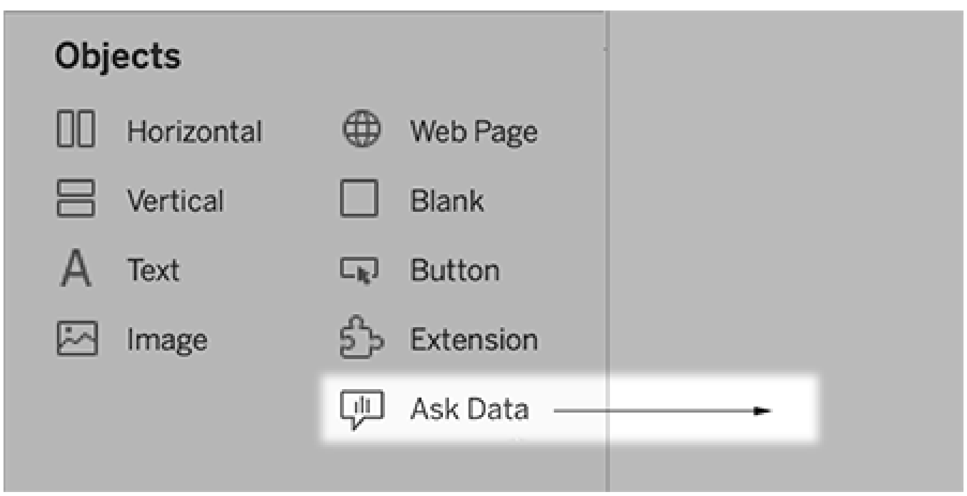
Lenses can be accessed from multiple places: the All Lenses page at the top level of a Tableau site, individual data source pages, and Ask Data objects on dashboards (shown below).

For more information, see **Create Lenses that Focus Ask Data for Specific Audiences** on page 1126.



Incorporate Ask Data into dashboards

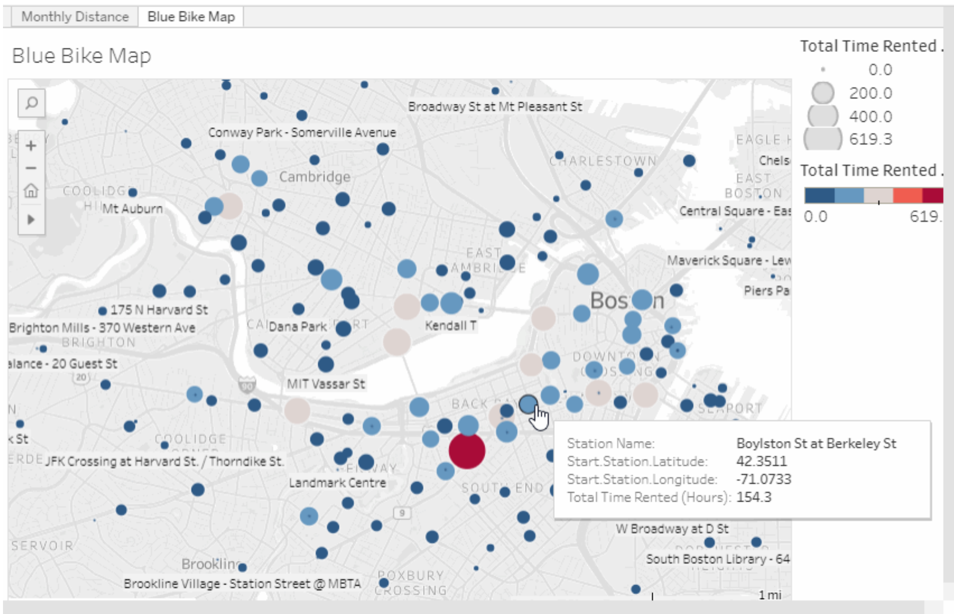
To quickly get up and running with lenses, Tableau dashboard authors can add the new Ask Data object. Anyone with access to the workbook and connected data sources will then be able to interact with data and create visualizations in an easy, search-based style.



Share explanations from Explain Data in published dashboards

Explain Data has a reimagined user interface optimized for a broader audience of business users. Authors (Creators or Explorers with editing permissions) can now enable Explain Data for viewers of dashboards and sheets in published workbooks. Viewers can select a mark of interest in the view and run Explain Data to explore their data more deeply than before.

Access to Explain Data in viewing mode is not enabled by default, so authors will need to enable this option. For information on how to enable Explain Data in viewing mode, see **Enable Explain Data in viewing mode in published workbooks and dashboards** on page 2075.

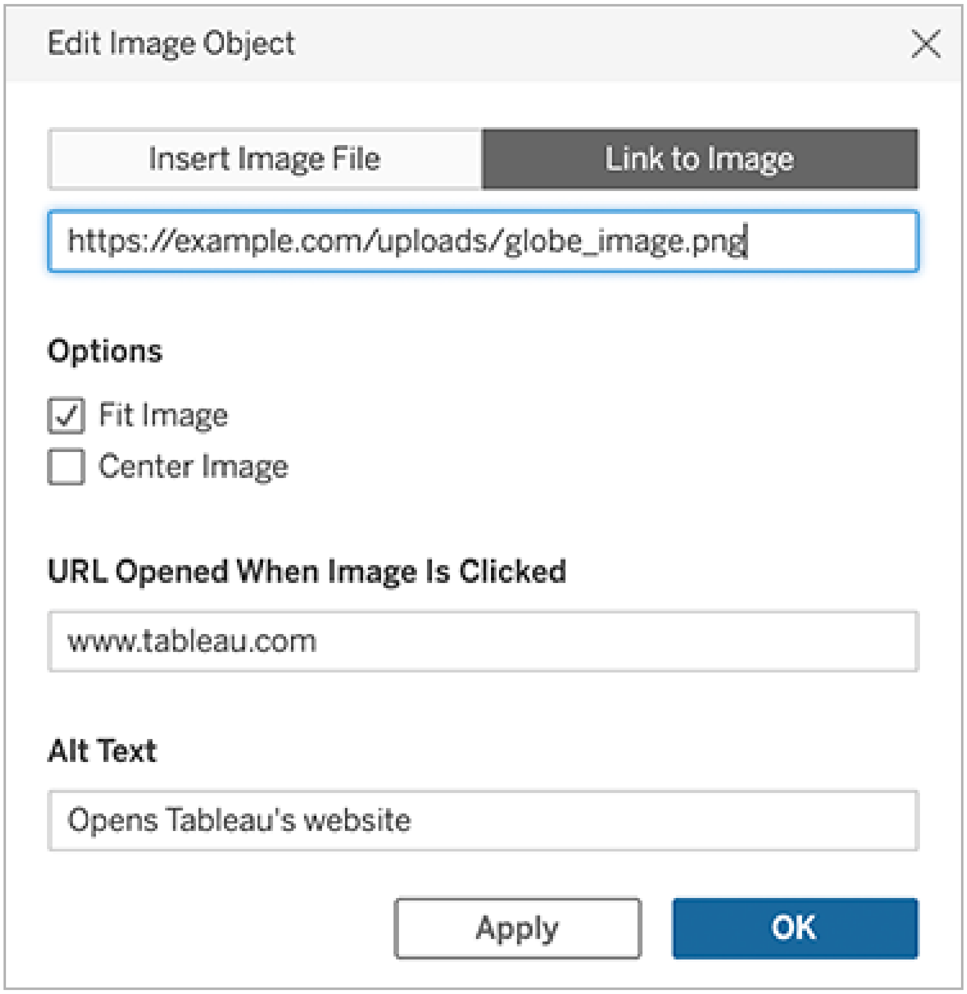


For more information, see **Discover Insights Faster with Explain Data** on page 2035, **Get Started with Explain Data** on page 2037, and **Requirements and Considerations for Using Explain Data** on page 2070.

Display web-based images in dashboards

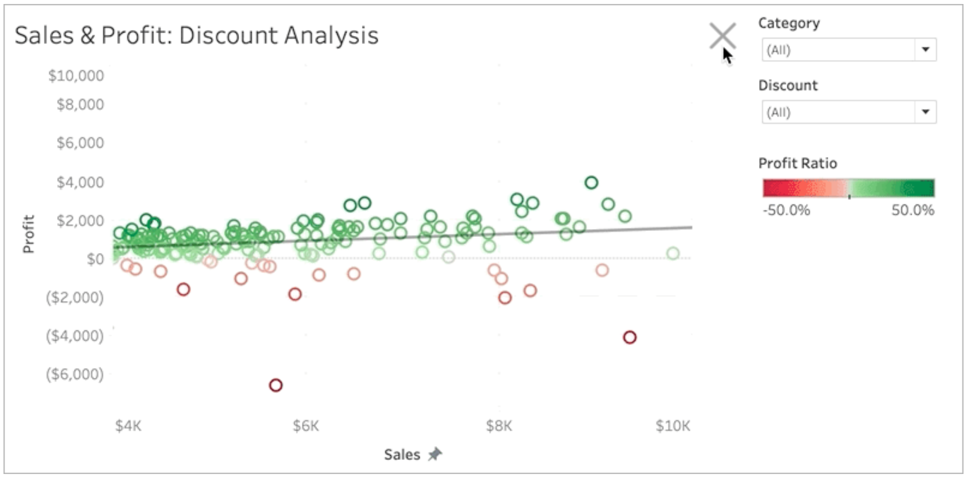
Image objects can now link to web-based images, as an alternative to images inserted in a workbook. This is particularly advantageous for large images, which load much more quickly when hosted on the web. (By contrast, inserted images must be downloaded every time a sheet is opened, slowing performance.) Linked web images also allow you to use animated GIFs, which Tableau doesn't natively support.

For more information, see **Detailed options for Image objects** on page 2745.



Create Show/Hide buttons for any dashboard object

Previously, Show/Hide buttons were limited to floating layout containers, but now you can create them for any dashboard object, floating or tiled. For more information, see **Show and hide objects by clicking a button** on page 2747.



New protocols supported for URL actions

You can now target a wider variety of web content that uses these URL prefixes: news, gopher, tsc, tsl, sms, tel, and file. For more information, see **URL Actions** on page 1509.

Use animations by default

In 2020.1, we introduced viz animations to help you see changes and recognize patterns in your data. When you create workbooks after 2021.2, Tableau enables animations for your vizzes by default. If you prefer, you can disable animations for a sheet or workbook. For more information about animations, see **Format Animations** on page 2887.

New spatial area calculation function

In 2020.2, we're introducing the Spatial Area function. Use the Spatial Area function to find the area of any polygon in a geographic role. For more information, see **Spatial Functions** on page 2245.

Layer control panel

Give users more control of geographic data with the Layer control panel. New to 2021.2, Layer Control is a panel that expands in the view when selected, displaying all the geographic data layers, and giving you quick access to each marks layer. For more information, see **Create Geographic Layers for Maps** on page 1869.

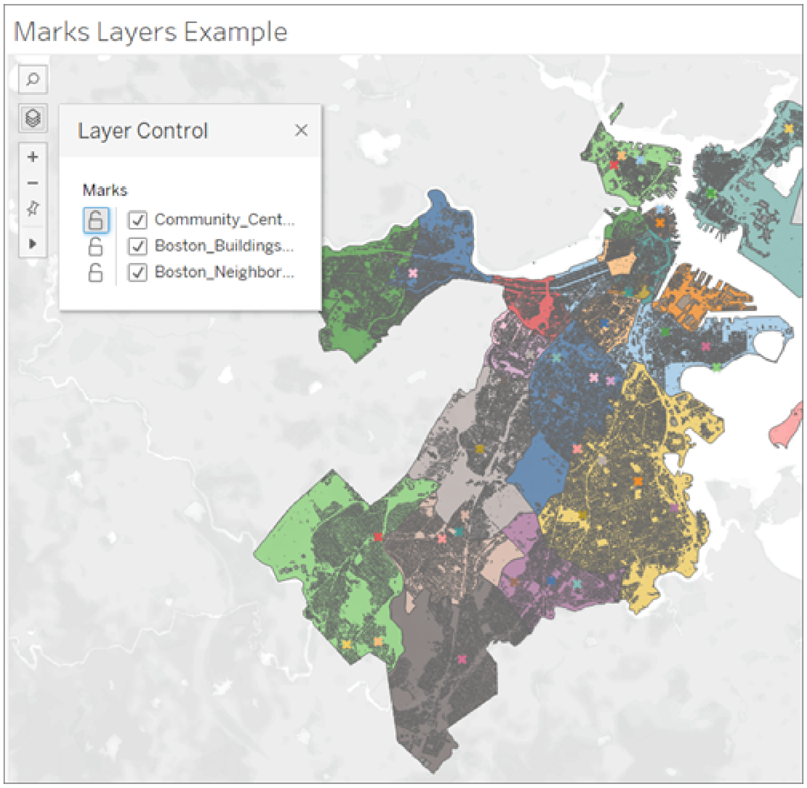


Tableau Desktop Features Now in Tableau Online and Tableau

Create parameter actions on the web

You can now create parameter actions in both Tableau Online and Tableau Server. When you create a parameter action, you customize how data is displayed in your viz. And users can change a parameter value by interacting directly with your viz. For more information, see **Parameter Actions** on page 1516.

Create set actions on the web

You can now create set actions in both Tableau Online and Tableau Server. Use sets to define a subset of data, and then use set actions to control how members in a set compare to other data in the view. For more information, see **Set Actions** on page 1540.

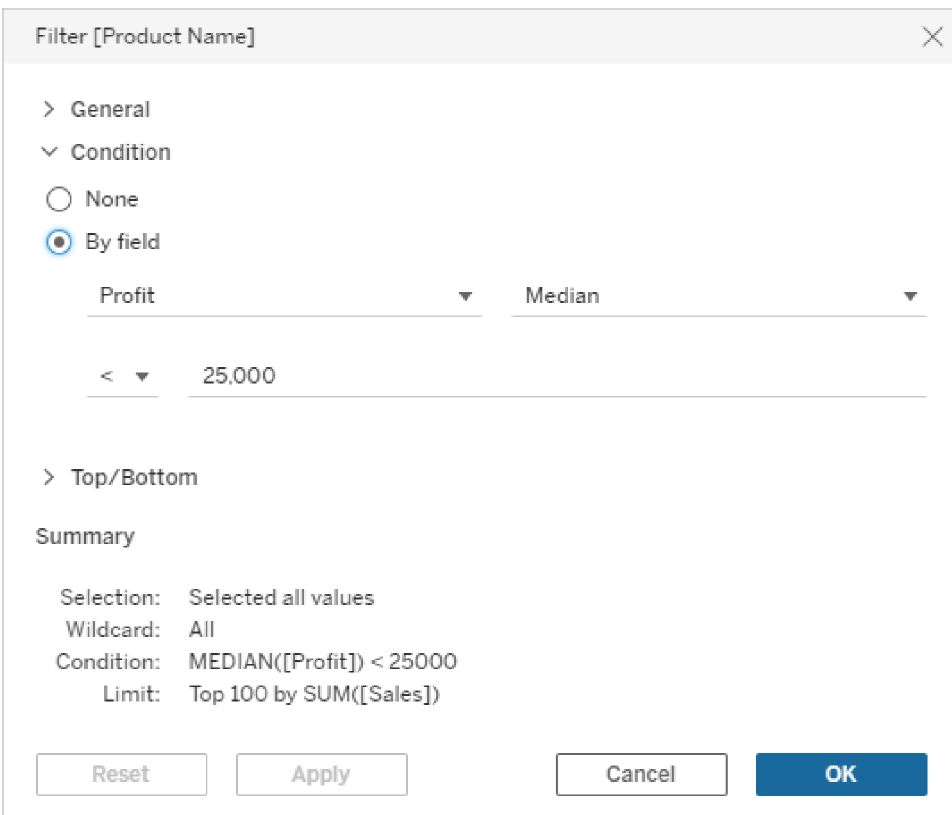
Create any action in Tableau Online and Tableau Server, including URL actions

Web-authoring now supports URL actions, with a revised, more intuitive layout of URL Target and Data Value options. For more information, see **URL Actions** on page 1509.

With this change, you can now author all of the same actions in Tableau Online, Tableau Server, and Tableau Desktop!

Condition and Top/Bottom filters now available on the web

Until now, when authoring in Tableau Online or Tableau Server, you could only filter categorical data by selecting a discrete value or by using a custom value list. But now, as of 21.2, you have more advanced filtering options: Condition and Top/Bottom. The Condition option allows you to define filtering rules with cutoffs, ranges, and more. The Top/Bottom option allows you to see only the top (or bottom) N results of a dimension, for instance, the top 100 products sold by a company.

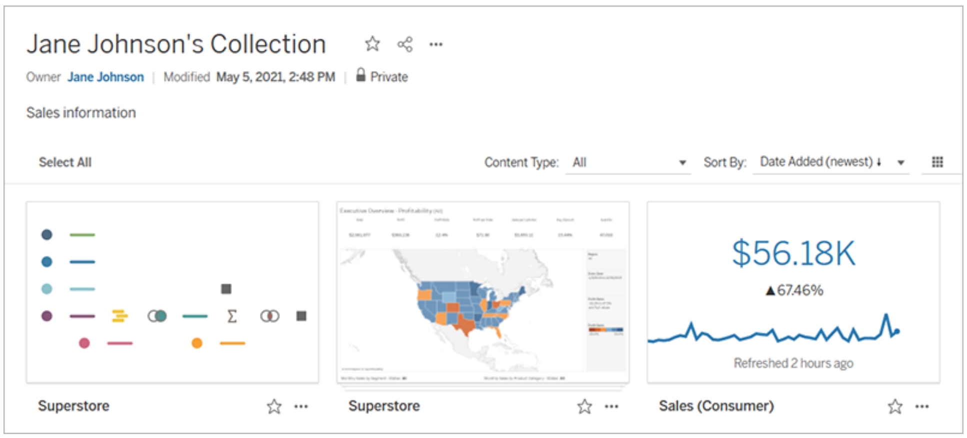


For more information, see **Filter Data from Your Views** on page 1411.

Unique Features in Tableau Online and Tableau Server

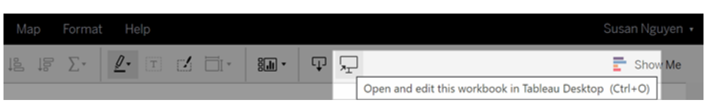
Organize items in a collection

Collections are like playlists that let you organize your Tableau content in a way that is meaningful for you or the people you share the collection with. Create a private collection for easy access to the workbooks, views, and metrics you need for a particular project. Or create a collection with onboarding resources and share it with new members of your team. For more information, see **Organize Items in a Collection** on page 3199.



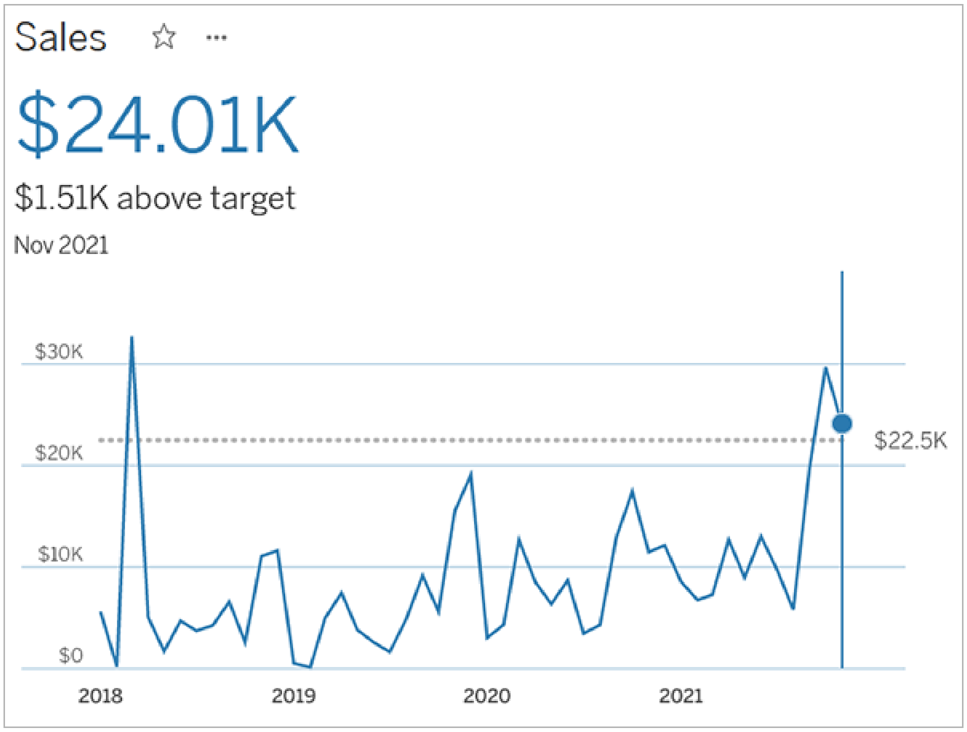
Open and edit workbooks in Tableau Desktop from the web

Use the new Edit in Tableau Desktop action to open a Tableau Server or Tableau Online workbook in Tableau Desktop, then publish changes when you're finished. This button lets you use whichever product works best for your task, and eliminates the need to download a workbook to open and edit in Tableau Desktop. You can publish your workbook back to Tableau Server or Tableau Online like usual. For more information, see **Edit Tableau Views on the Web** on page 3149.



Include a target value for your metric

Now, when you create a metric, you can specify a target value to compare your metric against. You might include a target that represents a threshold to stay above or a goal you're working toward. This target value will be used as the basis for the comparison shown below the metric value, both on the metric card and on the metric details. For more information, see **Create and Troubleshoot Metrics** on page 3152.



Add-ons

Tableau Catalog - part of the Data Management Add-on

Add Field Descriptions - You can add descriptions for fields in a published data source that then display in the Description column on the data source page Lineage tab. You can also see the description on the Data tab when you build a visualization. For more information, see **Manage Web Content** on page 3205.

Data Quality Warning Updates

- **Formatted text support for warning messages** - When you create or edit a data quality warning message, you can format the text with bold, underline, and italics, and include a link or an image. For more information, see "How to set a quality warning" in the Tableau Server or Tableau Online Help.

- **Message required for warnings** - When you create a data quality warning, you're now required to enter a message to display to users. (This doesn't apply to warnings generated when you set Tableau to monitor for extract refresh or flow failures.) For more information, see "How to set a quality warning" in the Tableau Server or Tableau Online Help.

Inherited descriptions appear in Tableau Desktop - When a description for a field exists upstream from the field, it now appears in Tableau Desktop with information about where the description is inherited from. For more information, see **Manage Web Content** on page 3205.

Full table name appears - The table's full name has been added to the table page.

Filter by tags - You can now filter databases, files, and tables by tags in the Connect To dialog box.

Filter by embedded or non-embedded - On the External Assets page, you can filter databases, files, and tables by asset category: embedded or non-embedded.

Features Introduced in Previous Versions of Tableau Desktop and Web Authoring

Browse summaries of features for previously released versions.

All Tableau Desktop Release Notes | All Known Issues

Added in previous versions

Version 2021.1

Added in version 2020.1

Install and Deploy Tableau Desktop

- **Zero downtime licensing** on the facing page
- **Amazon Redshift driver no longer installed automatically** on the facing page.

Connect to and Prepare Data

- Get web authoring support from more Tableau connectors
- Use Tableau connectors with Azure Active Directory authentication
- Snowflake connector supports virtual data warehouses
- New JDBC driver for SAP HANA connector in Windows has arrived
- Connect to Anaplan data using certificates

Design Views and Analyze Data

- **Integrate predictions from Einstein Discovery into Tableau** on page 16
- **Add dashboard extensions directly in Tableau** on page 18
- **Regularization and augmentation supported in predictive modeling functions** on page 19
- **Create Level of Detail expressions more easily with Quick LODs** on page 19

Prepare, Author, and View Data on the Web

- **Quickly access assistance for Ask Data** on page 20
- **Exclude values of specific fields from query interpretations in Ask Data** on page 21
- **Get notified when animations are paused for a viz** on page 22
- **Create, rearrange, and preview device-specific layouts on the web** on page 22
- **Drag and drop headers on the web** on page 23
- **See connected metrics on a view** on page 23

Collaborate and Explore on the Web

- **Keep track of your data with notifications on the web** on page 24

Add-ons

- **Tableau Catalog updates - part of the Data Management Add-on** on page 24

Install and Deploy Tableau Desktop

Zero downtime licensing

Starting in Tableau version 2021.1, internet-connected Tableau Desktop and Tableau Prep Builder users no longer have to manually refresh product keys. Licenses are automatically refreshed without requiring any action as their subscription approaches expiration. For more information, see **Maintain Licenses for Tableau Desktop and Tableau Prep** on page 3339.

Amazon Redshift driver no longer installed automatically

Starting in Tableau version 2021.1, Tableau no longer automatically installs the Amazon Redshift driver as part of the product install process. If you need this driver, download it from the Driver Download page.

Connect to and Prepare Data

Get web authoring support from more Tableau connectors

Tableau web authoring support has expanded to include connectivity using these connectors:

- Amazon Athena
- Apache Drill (available for Tableau Server, but not Tableau Online)
- OData
- SharePoint Lists
- Teradata

For a full list of connectors you can use for web authoring in Tableau Online and Tableau Server, see Creators: Connect to data on the web.

Use Tableau connectors with Azure Active Directory authentication

Tableau has enhanced these connectors to include support for authentication via Azure Active Directory:

- Azure Synapse Analytics (See Azure SQL Synapse Analytics.)
- Azure Data Lake Storage Gen2 (New connector! See Azure Data Lake Storage Gen2.)
- Azure SQL Database (See Azure SQL Database.)
- Databricks (See Databricks.)

Snowflake connector supports virtual data warehouses

Beginning with Tableau 2021.1, Tableau automatically uses your default Snowflake virtual warehouse if none is specified. When you create a Snowflake data source, Tableau uses your virtual warehouse. Later, if you publish that data source or a workbook, Tableau uses the default Snowflake virtual warehouse of the person who accesses it. (For more details, see Snowflake.)

New JDBC driver for SAP HANA connector in Windows has arrived

The SAP HANA connector now uses JDBC drivers for Mac, Linux, and Windows. The JDBC driver for Windows is new in Tableau version 2021.1. JDBC drivers for Mac and Linux were introduced with Tableau version 2020.2.

Connect to Anaplan data using certificates

Beginning in February 2021, you can choose to connect to your Anaplan data with a certificate and private key. (For more details, see Anaplan.)

Design Views and Analyze Data

Integrate predictions from Einstein Discovery into Tableau

Previously, you could access Einstein Discovery only in Salesforce. But now you can integrate predictions directly into Tableau views and dashboards using models deployed by Einstein Discovery in Salesforce to combine powerful predictive and visual analytics. For related information in Tableau help, see **Integrate Einstein Discovery Predictions in Tableau** on page 2671.

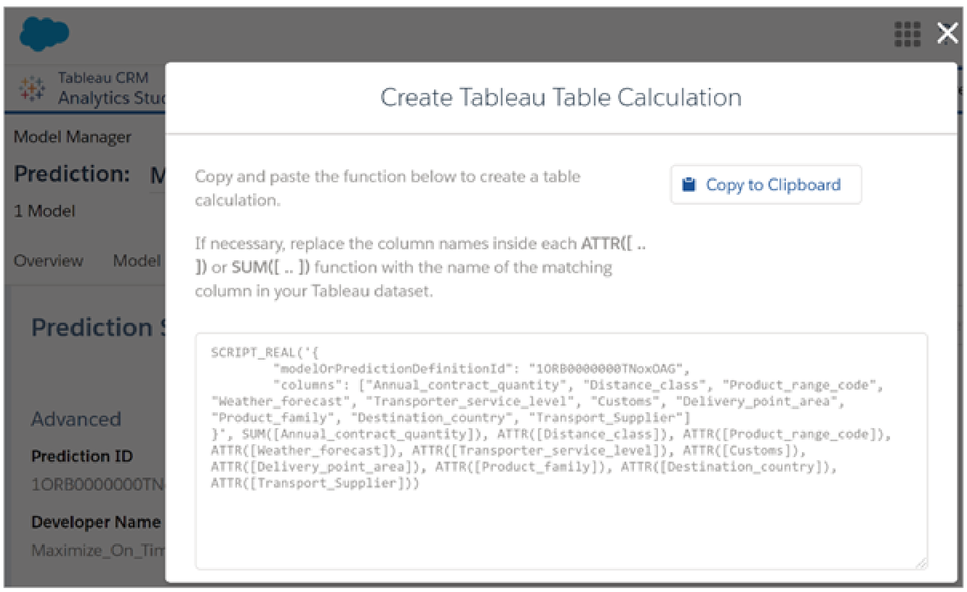
For information on required licenses, access, and permissions in Salesforce and Tableau, see **Requirements for access** on page 2672.

For an introduction to Einstein Discovery in Tableau, see the Einstein Discovery in Tableau page and demo on tableau.com.

Connect Tableau table calculations to Einstein Discovery predictions

Supported in Tableau Desktop, Tableau Online, and Tableau Server

With Tableau connected to the Einstein Discovery analytics extension, you can embed Einstein Discovery predictions directly in calculated fields. In Salesforce, use Model Manager to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook.



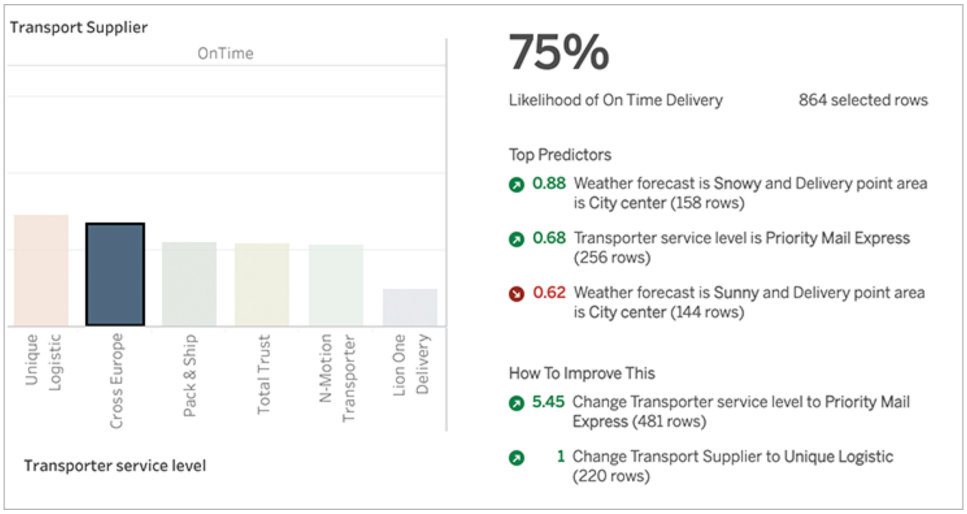
For more information, see **Einstein Discovery** on page 2693 in **Pass Expressions with Analytics Extensions** on page 2690.

> **Note**: To let users see predictions in published table calculations, Tableau Server administrators need to enable analytics extensions for the site. For more information, see Configure Connection with Analytics Extensions in Tableau Server help.

Add Einstein Discovery predictions to Tableau dashboards

Supported in Tableau Desktop, Tableau Online, and Tableau Server

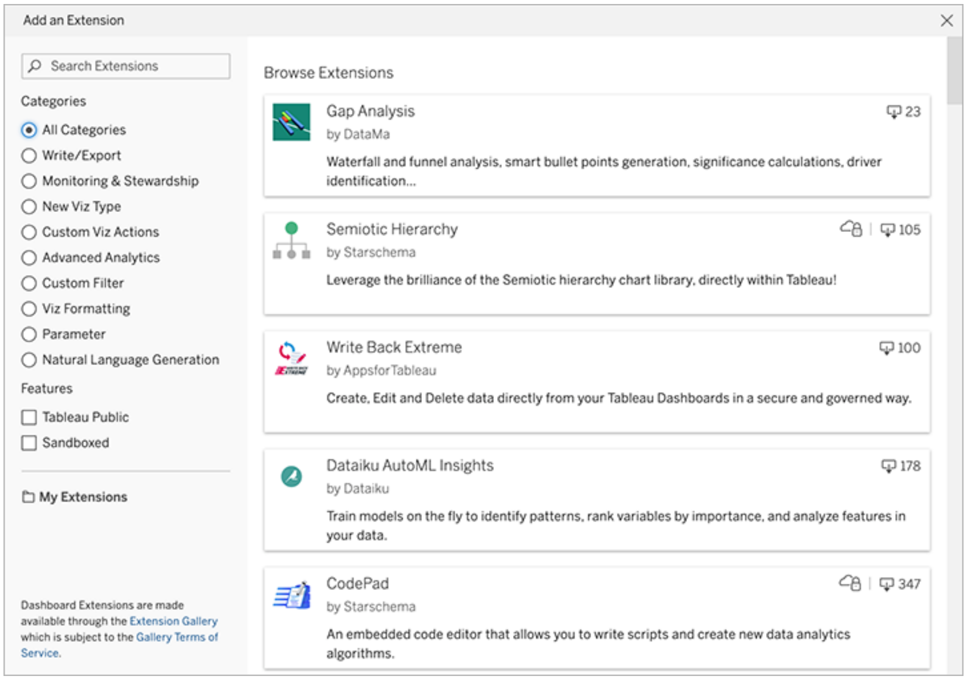
With the new Einstein Discovery dashboard extension, as users select marks in a view, they see dynamic updates to predictions and suggestions to improve predicted outcomes. Authors simply drag the Extension object onto the dashboard canvas, select Einstein Discovery from the extensions gallery, and then configure the extension in Tableau. For more information, see **Explore Predictions in Tableau with the Einstein Discovery Dashboard Extension** on page 2676.



Add dashboard extensions directly in Tableau

In the past, adding a dashboard extension required you to open a browser and then download and navigate to a file. But now you can quickly add extensions directly in Tableau. Simply drag the Extension object into your dashboard, and then search for and select the desired extension.

For more information, see **Use Dashboard Extensions** on page 2839.



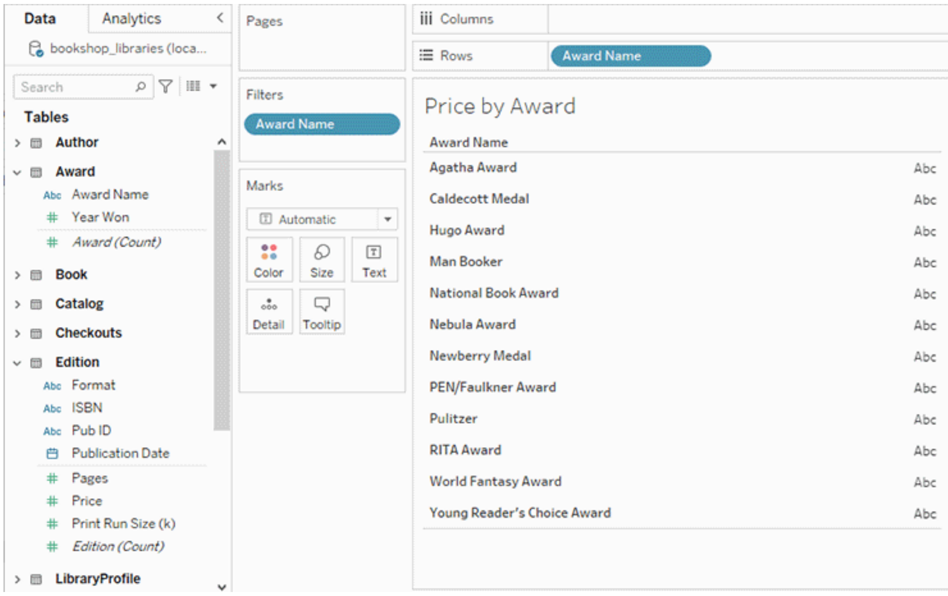
Regularization and augmentation supported in predictive modeling functions

In 2020.3, we introduced the MODEL_QUANTILE and MODEL_PERCENTILE table calculation functions, which allow you to leverage the power of a statistical engine to build predictive models. Now in 2021.1, if you're an advanced user, you'll be able to fine-tune your predictions with two new arguments in your prediction calculations: **lambda** (or the "regularization parameter") and **augmentation**. Lambda is a scalar you can use to adjust the regularization effect on your model. Augmentation allows you to increase the ability of your models to pick up non-linear patterns.

For more information, see **How Predictive Modeling Functions Work in Tableau** on page 2606 and **Regularization and Augmentation in Predictive Modeling** on page 2659.

Create Level of Detail expressions more easily with Quick LODs

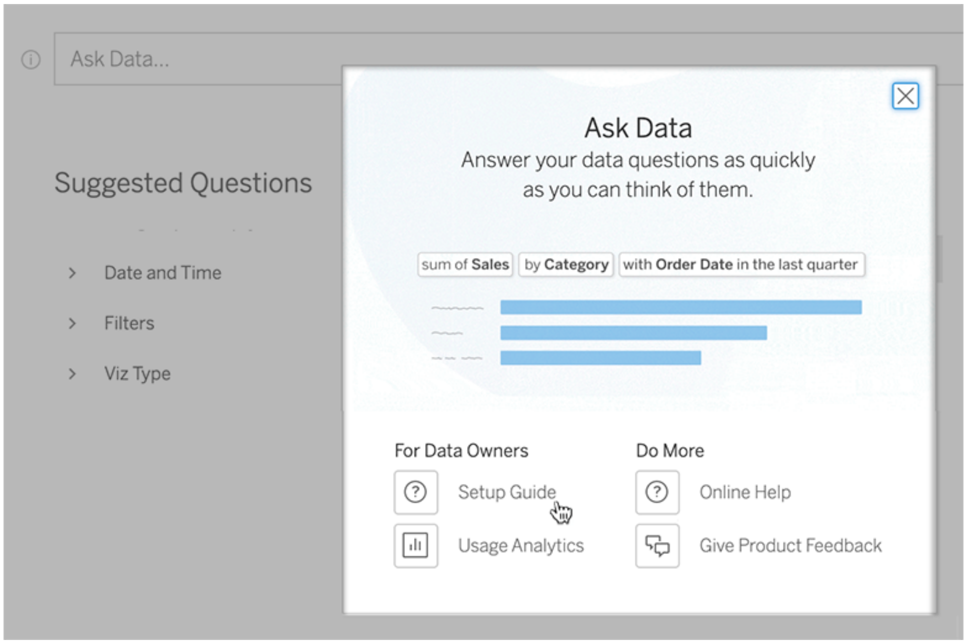
Create Level of Detail expressions directly in the Data pane using drag-and-drop or context menus. Quick LODs make it easy to bring a calculation from one table to a different level of detail in another table. Press Ctrl (Windows) or Command (Mac), and then click and drag a measure onto a dimension to automatically create an LOD expression. For more information, see **Use a Quick LOD expression** on page 2452.



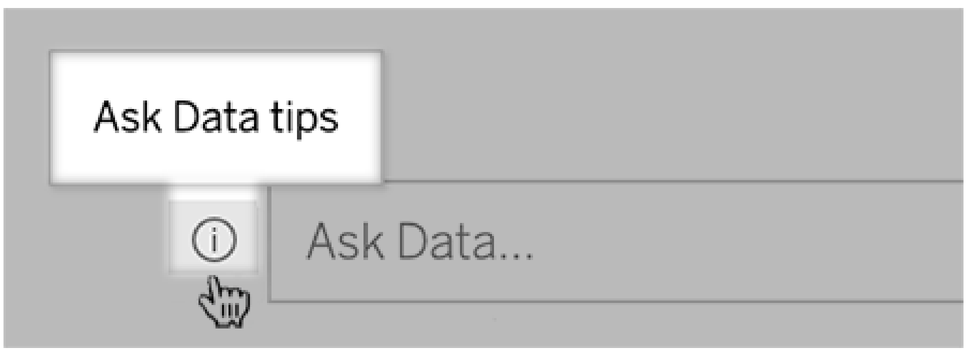
Prepare, Author, and View Data on the Web

Quickly access assistance for Ask Data

When you first navigate to Ask Data, a dialog of helpful tips appears. Data owners see links to all resources pictured below, including unique ones for setting up data and analyzing usage. Data consumers see only the links under "Do More" for Help and the product feedback form, which sends feature requests directly to our development team at Tableau.



To later access the tips again, simply click the "i" to the left of the Ask Data text box.

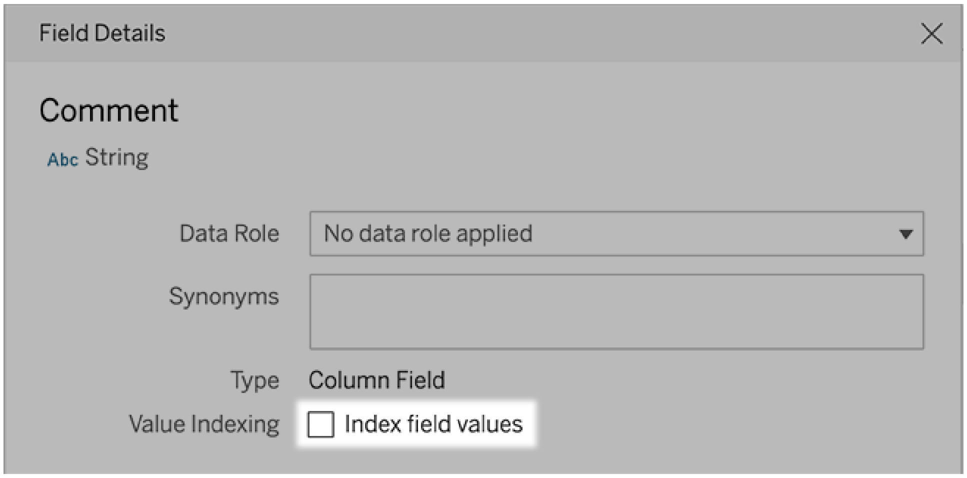


For more information, see **Automatically Build Views with Ask Data** on page 1109.

Exclude values of specific fields from query interpretations in Ask Data

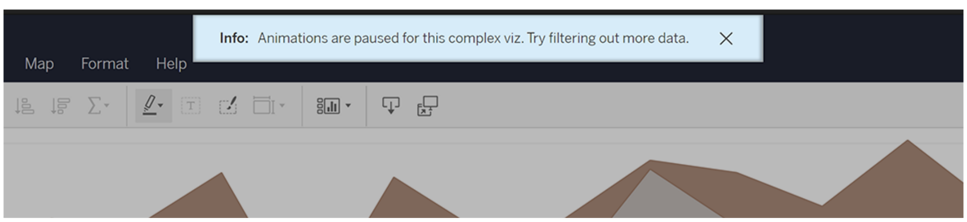
To improve the usability of query interpretations in Ask Data, data source owners can exclude the values of specific fields from indexing. Though Ask Data doesn't add non-indexed values to query interpretations, the values still appear in visualization results when relevant. For example, if you don't index values from a "Product" field because they add unnecessary detail to query interpretations, Ask Data can still display values such as "iPhone 12" in resulting data visualizations.

For more information, see **Exclude values of specific fields from search results** on page 1134.



Get notified when animations are paused for a viz

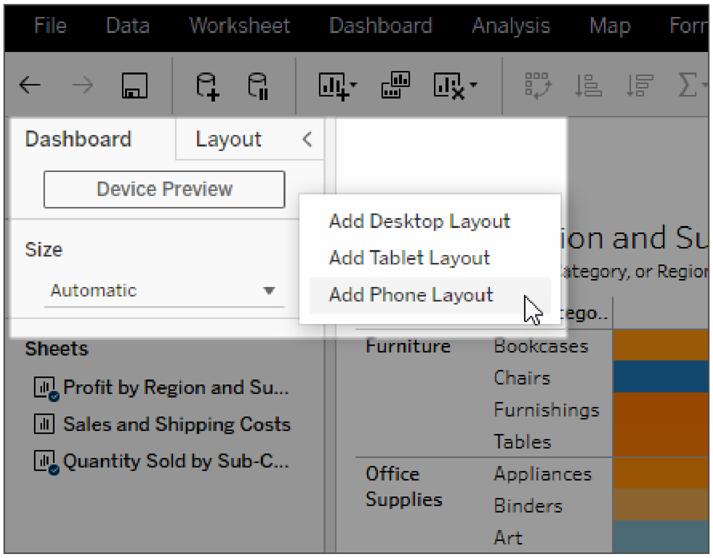
If a viz is particularly complex or uses density marks, it's rendered on the server rather than your device, disabling animations in the browser. Tableau now notifies you about this and suggests that you filter out data, which sometimes simplifies the viz enough for animations to play.



For more information, see **Why animations won't play** on page 2891.

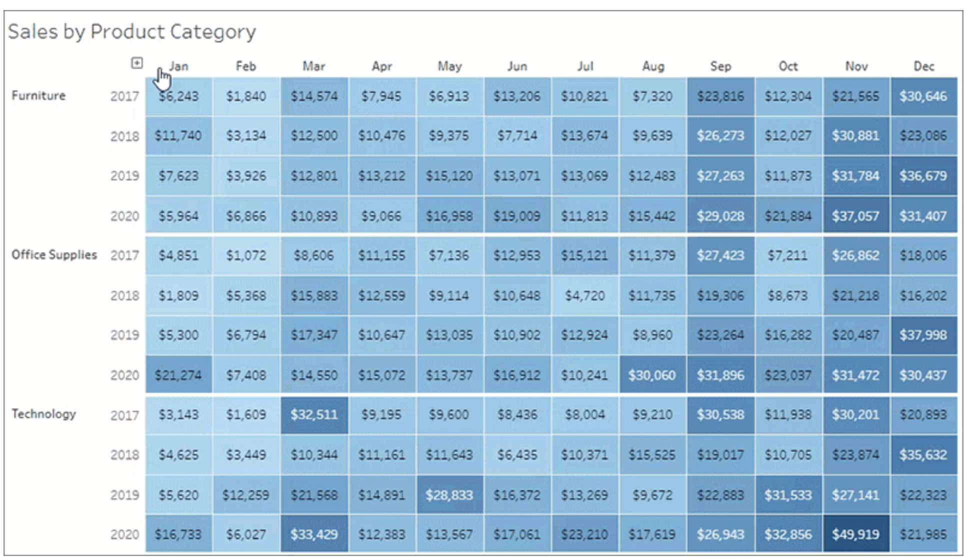
Create, rearrange, and preview device-specific layouts on the web

You can now preview and customize what your dashboard will look like across different device types when authoring on the web, including changing the dashboard layout to show differently i. Previously, customers could see and view device-specific views, but couldn't edit or create new layouts. Create desktop, tablet, or phone layout by right-clicking the device preview area in the Dashboard left pane. For more information, see **Create Dashboard Layouts for Different Device Types** on page 2801.



Drag and drop headers on the web

Create a custom sort order by dragging-and-dropping headers to actively reorder columns/rows directly within a view on the web, just like in Tableau Desktop. With a sortable horizontal or vertical header, click to drag and drop the header to a new location within that header set. For more information, see **Sort Data in a Visualization** on page 1447.

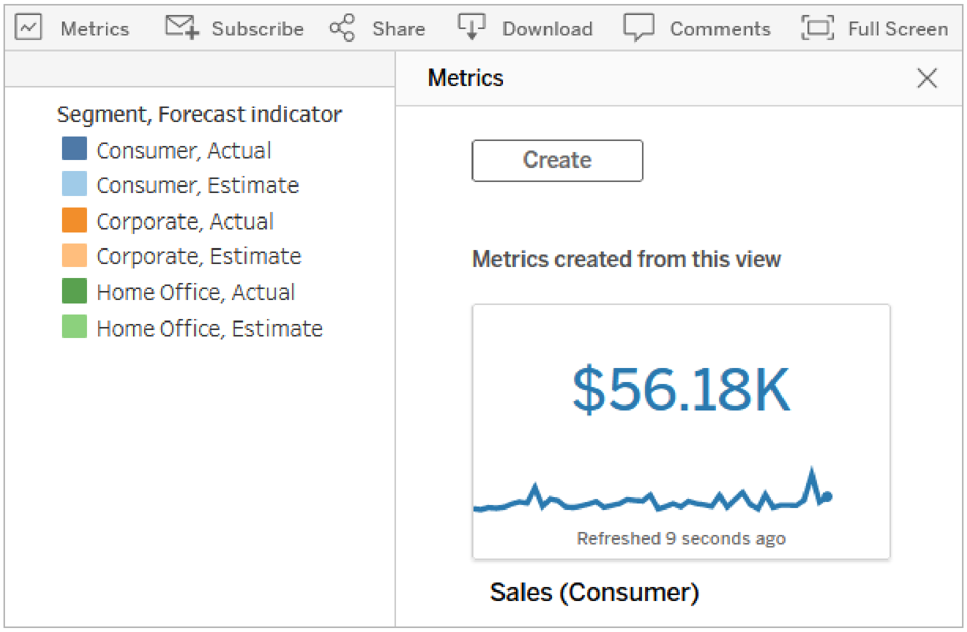


Sales by Product Category

		Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Furniture	2017	$6,243	$1,840	$14,574	$7,945	$6,913	$13,206	$10,821	$7,320	$23,816	$12,304	$21,565	$30,646
	2018	$11,740	$3,134	$12,500	$10,476	$9,375	$7,714	$13,674	$9,639	$26,273	$12,027	$30,881	$23,086
	2019	$7,623	$3,926	$12,801	$13,212	$15,120	$13,071	$13,069	$12,483	$27,263	$11,873	$31,784	$36,679
	2020	$5,964	$6,866	$10,893	$9,066	$16,958	$19,009	$11,813	$15,442	$29,028	$21,884	$37,057	$31,407
Office Supplies	2017	$4,851	$1,072	$8,606	$11,155	$7,136	$12,953	$15,121	$11,379	$27,423	$7,211	$26,862	$18,006
	2018	$1,809	$5,368	$15,883	$12,559	$9,114	$10,648	$4,720	$11,735	$19,306	$8,673	$21,218	$16,202
	2019	$5,300	$6,794	$17,347	$10,647	$13,035	$10,902	$12,924	$8,960	$23,264	$16,282	$20,487	$37,998
	2020	$21,274	$7,408	$14,550	$15,072	$13,737	$16,912	$10,241	$30,060	$31,896	$23,037	$31,472	$30,437
Technology	2017	$3,143	$1,609	$32,511	$9,195	$9,600	$8,436	$8,004	$9,210	$30,538	$11,938	$30,201	$20,893
	2018	$4,625	$3,449	$10,344	$11,161	$11,643	$6,435	$10,371	$15,525	$19,017	$10,705	$23,874	$35,632
	2019	$5,620	$12,259	$21,568	$14,891	$28,833	$16,372	$13,269	$9,672	$22,883	$31,533	$27,141	$22,323
	2020	$16,733	$6,027	$33,429	$12,383	$13,567	$17,061	$23,210	$17,619	$26,943	$32,856	$49,919	$21,985

See connected metrics on a view

When you open the metrics pane on a view, you will now see metrics that have been created from that view. Users with any site role can see connected metrics. Previously, the metrics pane was available only for users with a site role of Creator or Explorer (can publish) to create metrics. For more information, see **Create and Troubleshoot Metrics** on page 3152.



Collaborate and Explore with Tableau Online and Tableau Server

Keep track of your data with notifications on the web

Find out when someone comments or shares a view or workbook with you, right in your Tableau site. Select the bell icon in the top right corner to see all your notifications (including updates if you own flows or extracts) and take action. Control where you receive these notifications in your Account Settings--directly in Tableau, by email, or both. For more information, see **Tour Your Tableau Site** on page 3104.

Add-ons

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

The following features are included in this release:

Automated data quality warning - You can set Tableau to monitor for extract refresh failures and for flow run failures. When a failure occurs, Tableau generates a data quality warning that displays to users. For more information, see "Set a Data Quality Warning" in the Tableau Server or Tableau Online Help.

Quick search results expanded - Quick search results now include tags on external assets.

Version 2020.4

Added in version 2020.4

Connect to and Prepare Data

Design Views and Analyze Data

Prepare, Author, and View Data on Tableau Online and Tableau Server

Connect to and Prepare Data

Find and use partner-built connectors

Tableau has great connectivity that allows you to visualize data from virtually anywhere. In addition to dozens of built-in connectors for Tableau, you'll find more connectors built by Tableau partners in the Tableau Extension Gallery. For more information about using partner-built connectors, see **Use Partner-Built Connector from the Extension Gallery** on page 689.

Connect to PostgreSQL 12 using a new JDBC connector

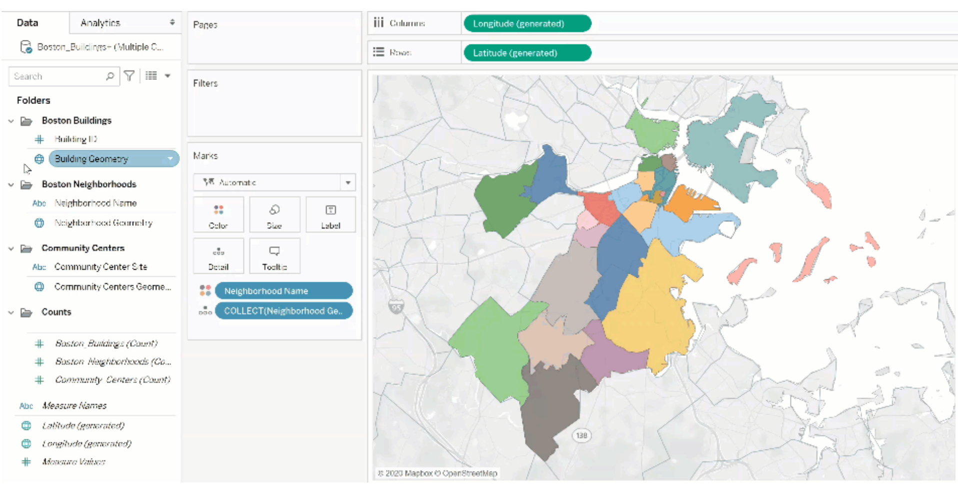
Now you can connect to PostgreSQL 12 databases from Tableau. A new JDBC driver for PostgreSQL supports these connections and is included with all Tableau products, except Tableau Desktop for Mac and Tableau Builder for Mac. (Online support for JDBC and PostgreSQL 12 connectivity will be rolled out more gradually.)

Design Views and Analyze Data

Build maps with multiple data layers: marks layers for maps

With 2020.4, we are introducing a new way to work with geographic data. Instead of being limited to two layers of data through a dual axis map, now you can add as many layers of geographic data as you want. Simply drag your geographic data to the view and drop it on the "Add Map Layer" control that appears in the view. Each layer can be customized independently using the marks card, giving you new levels of flexibility in map creation.

For more information, see **Create Geographic Layers for Maps** on page 1869.



Enhancements to dynamic parameters: refreshable parameter ranges

In 2020.1, we introduced dynamic parameters, which makes it possible to set a parameter’s current value or list of values to the result of a view-independent calculation every time the workbook opens. Now, you can also create dynamic parameters for “range” parameters. This means that min and max values for a parameter’s range will automatically update every time the workbook is opened. Tableau will query all values in a field's range and users can pick any value within the range, even if that value is not in the field's discrete set of values.

For more information, see this blog post on dynamic parameters. Also, see **Create Parameters** on page 1218.

More models supported for predictive modeling functions

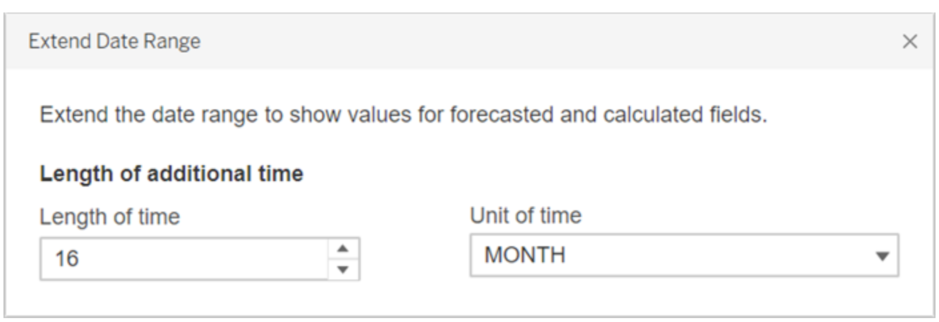
In 2020.3, we introduced the MODEL_QUANTILE and MODEL_PERCENTILE table calculation functions. By default, these functions use linear regression to generate predictions and explore relationships within your data. Now in 2020.4, you'll be able to leverage two more models: Gaussian process regression and regularized linear regression. With more models supported, you have greater flexibility and can choose the model that best fits your use case.

For more information, see **Choosing a Predictive Model** on page 2615 and **How Predictive Modeling Functions Work in Tableau** on page 2606.

Extend date ranges into the future

You probably already know that when you're working with dates, Tableau only shows the values that are represented in your data. Until now, users have been able to show missing values to indicate incomplete data in the underlying data source, but this only applies to dates within the boundaries of the existing data source. In addition, in 2020.3 we introduced the ability to perform calculations on missing values when you select **Analysis > Infer Properties from Missing Values**.

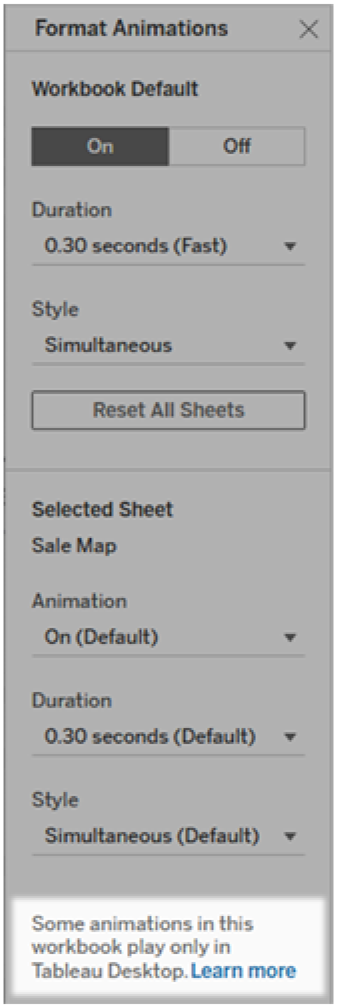
In 2020.4, we've added a simple menu option that lets you easily generate and display marks beyond the existing date range. To do this, simply open the context menu of a continuous (green) date pill, and select **Extend Date Range**.



To see an example of this, check out **Predictive Modeling Functions in Time Series Visualizations** on page 2649.

Get notified when features won't animate in a browser

If a workbook contains features that won't animate in a web browser, such as maps, polygons, and density marks, you now get notified at the bottom of the Format Animations pane. For more information, see **Unsupported browsers and features** on page 2892.



Prepare, Author, and View Data on Tableau Online and Tableau Server

Use Salesforce data sources for web authoring

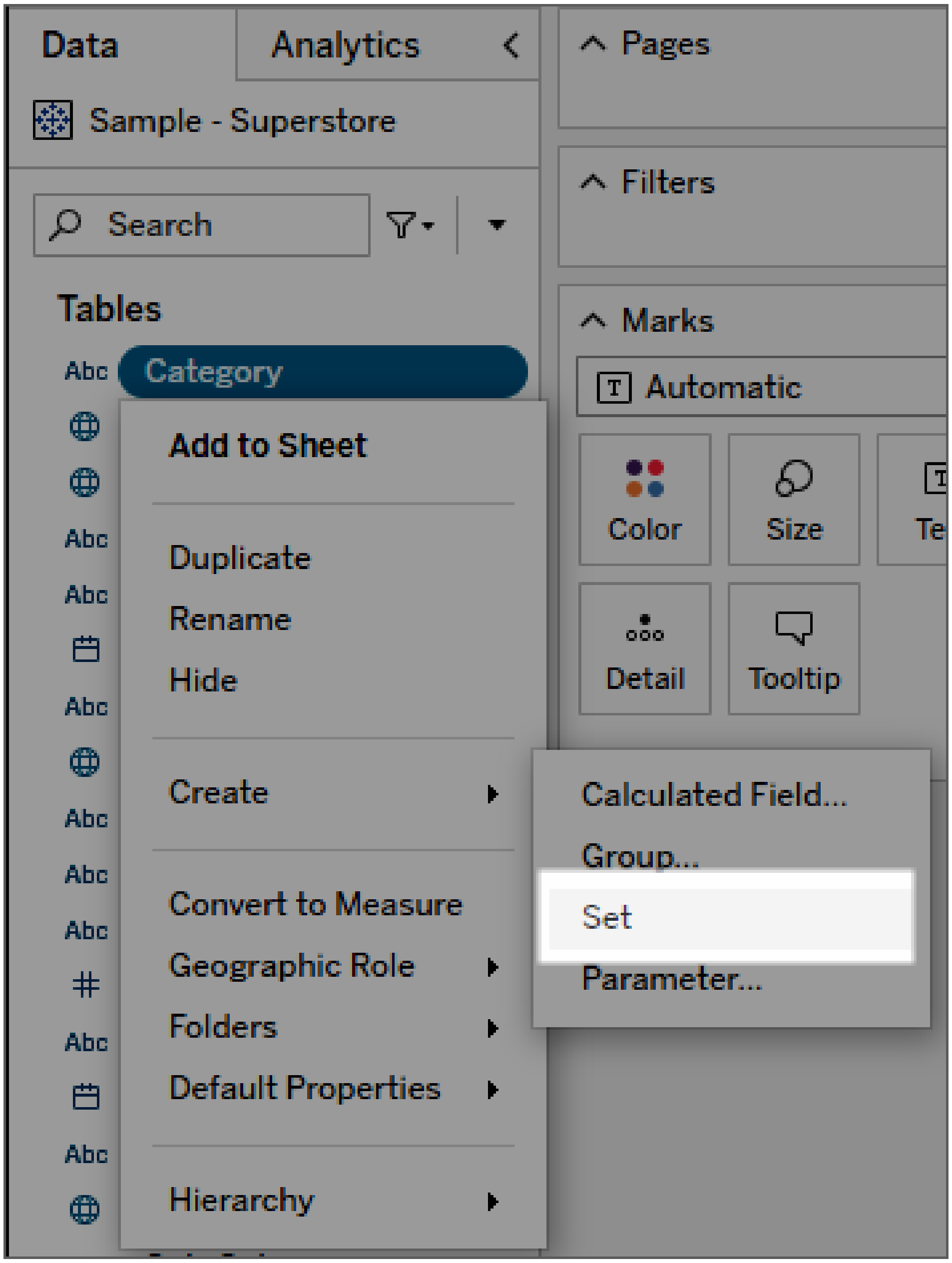
Creators can author and edit workbooks connected to Salesforce data sources in Tableau Server and Tableau Online.

Create highlight actions in Tableau Online and Tableau Server

Now you can create advanced highlight actions while authoring in Tableau Online and Tableau Server, from both dashboards and individual sheets. For more information, see **Create Advanced Highlight Actions** on page 1507.

Create fixed sets in Tableau Online or Tableau Server

In 2020.3, we introduced the ability to create dynamic sets on the web. Now in 2020.4, you can create *fixed* sets in Tableau Online and Tableau Server, too—right in the web authoring environment—with no conditions. To do this, simply open the dimension's context menu and select **Create > Set**.



For more information on sets in Tableau, see **Create Sets** on page 1186.

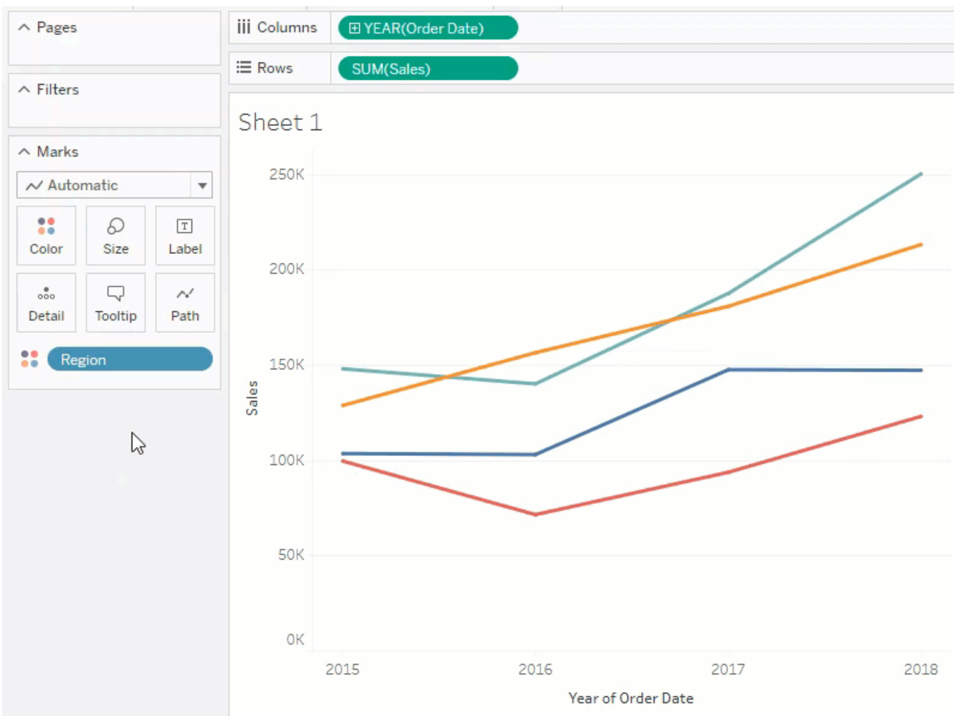
Apply filters to worksheets in Tableau Online or Tableau Server

Until now, when authoring on the web, applying a single filter to multiple sheets has been cumbersome. As of 2020.4, Tableau Online and Tableau Server customers can now apply a filter to selected worksheets by using a brand new dialog. In addition, the entire experience is keyboard accessible. To try it out, go to a filter’s context menu and select **Apply to Worksheets > Selected Worksheets**.

Add labels to your marks in Tableau Online or Tableau Server

You can now choose which marks to label in a visualization while authoring in Tableau Online or Tableau Server. The same options that are already available in Tableau Desktop are now available in web authoring. This means you can label all marks, only the minimum and maximum values, just the line ends, the most recent values, or only the selected or highlighted marks in the view.

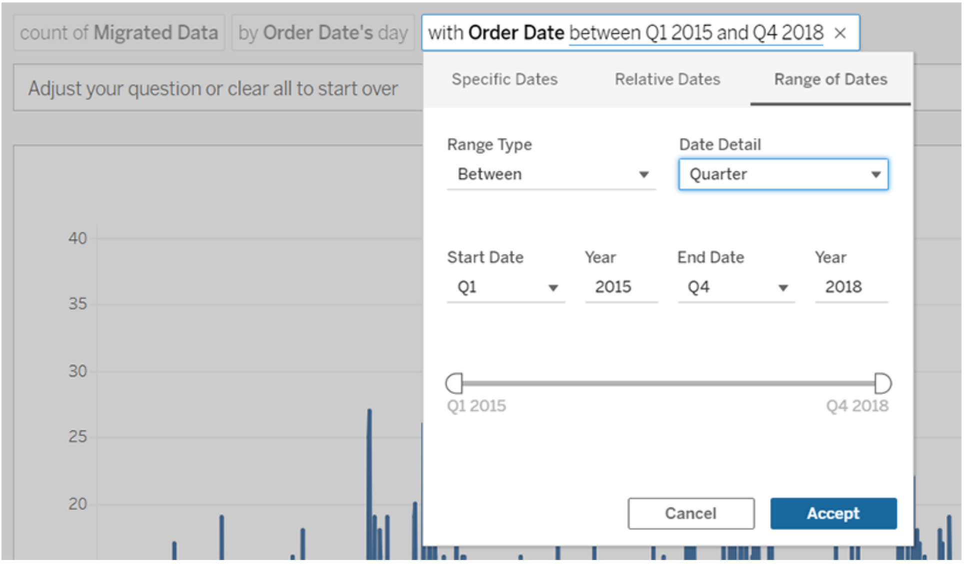
For more information, see **Show, Hide, and Format Mark Labels** on page 1378.



Refine date ranges with Ask Data

Now you can precisely adjust date ranges in the user interface. Simply click "between", "starting", or "ending" in your questions, and change range type, date detail, and start and end points.

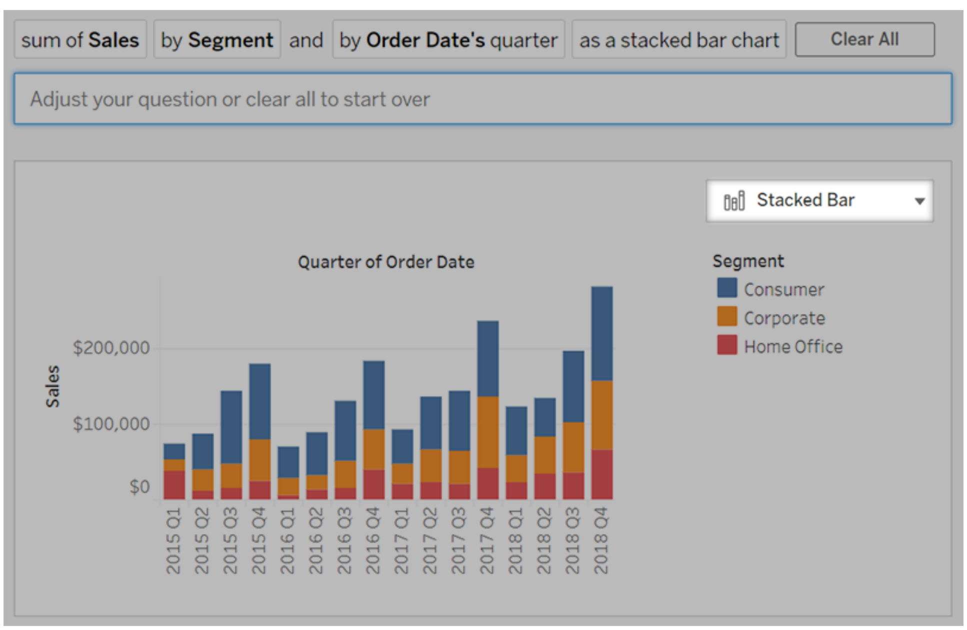
For more information, see **Adjust date filters** on page 1118.



See vizzes as stacked bar charts with Ask Data

Stacked bar charts provide a great way to see how different categories contribute to your overall data. Ask Data automatically displays stacked bar charts with data best visualized that way. But you can display any query this way by adding "as a stacked bar chart" to your question or choosing "Stacked Bar" from the viz-type menu.

For more information, see **Change the viz type** on page 1114.



Edit synonyms for field values with Ask Data

Synonyms ensure that Ask Data returns relevant vizzes, even if users' terminology differs from field names and values in a data source. Using data roles that apply across multiple data sources, you could previously edit synonyms only for field names, but now you can edit them for field values as well.

For more information, see **Edit field name or value synonyms for a data role** on page 1144.

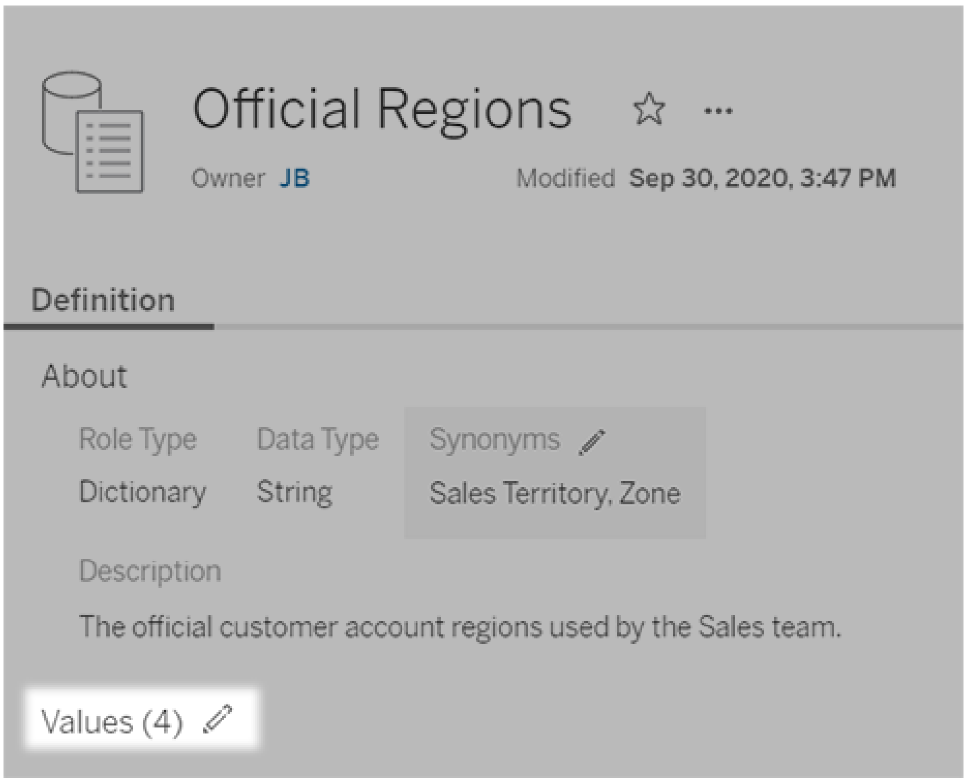


Image download improvements

When you download an image in Tableau Server and Tableau Online, selections made in the Page Setup dialog in Tableau Desktop (such as showing/hiding captions or legends) no longer apply. For more information, see **Download Views and Workbooks** on page 3328.

Collaborate and Explore with Tableau Online and Tableau Server

New ways to see and manage custom views

Quickly access and manage your custom views on Tableau Online or Tableau Server in one place. When you look at a workbook in Tableau Online or Tableau Server, you can use the new Custom Views tab to see your saved custom views as well as custom views others have made visible to others. As the workbook owner or administrator, you can use the Custom Views tab to see all custom views on the workbook as well as view count and when the custom view was last accessed. For more information about custom views, see **Use Custom Views** on page 3224.

Organize items in a collection (limited preview)

Collections are like playlists that let you organize your Tableau content. Create private collections for easy access to the workbooks, views, and metrics you need for your different work projects. Or make a public collection with getting started resources for new employees on your team. Collections are available as a limited preview that your administrator can sign up for on Tableau Online only. For more information, see **Organize Items in a Collection** on page 3199.

Add-ons

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

The following features are included in this release:

Data Quality Warning History admin view - Site administrators can see how data quality warnings are being used on the site using the pre-built admin view, Data Quality Warning History. For more information, see "Data Quality Warning History" in Tableau Server or Tableau Online Help.

See quality warning details in Tableau Desktop - Tableau Desktop users can hover over the warning icon on the Data tab to see details about the data quality warnings affecting the data in their workbook. For more information, see "Set a Data Quality Warning" in Tableau Server or Tableau Online Help.

Metrics appear in lineage and show data quality warnings - Data quality warnings now appear on metrics affected by a warning. Metrics also appear in the Lineage tool, enabling you to see how a metric might be affected by data changes when performing impact analysis. For more information, see "Embedded asset appears in External Assets" in Tableau Server or Tableau Online Help.

Install and deploy Tableau

Turn off Web Page objects in Tableau Desktop

If, for security or other reasons, you don't want Web Page objects in dashboards to display target URLs, you can turn these objects off in several ways:

- During install from the command line, set the `WEBVIEWTOGGLEWEBZONESENABLED` property to `0`.
- After installation, use the Windows registry or macOS Terminal app.
- In the Tableau Desktop interface, choose **Help** > **Settings and Performance** > **Set Dashboard Web View Security**, and deselect **Enable Web Page Objects**.

CPU requirement update

Beginning with version 2020.4, Tableau requires you to have CPUs that support SSE4.2 and POPCNT instruction sets. These CPUs have been common for more than 10 years and are installed on most newer computers. Tableau products are also not currently compatible with Apple Silicon processors.

If you attempt to install or upgrade Tableau on a computer that does not support these instruction sets or that uses the Apple Silicon processors, the installer will not allow you to continue.

To see a complete list of supported technologies, see tableau.com/techspecs.

Version 2020.3

Added in version 2020.3

Install and Deploy Tableau

- **Recent workbooks and settings are maintained on upgrade** on page 38

Connect to and Prepare Data

Design Views and Analyze Data

Prepare, Author, and View Data on the Web

Collaborate and Explore on Tableau Online and Tableau

Server

- **Find items shared with you** on page 46
- **Get notified when animations are paused for a viz** on page 22
- **Pick new options for PDF subscriptions** on page 47
- **Accessibility improvements for organizing your favorites** on page 47
- **Use Mapbox or WMS service connections in Tableau Online or Tableau Server** on page 44

Add-ons

- **Tableau Catalog updates - part of the Data Management Add-on** on page 48

Install and Deploy Tableau

Recent workbooks and settings are maintained on upgrade

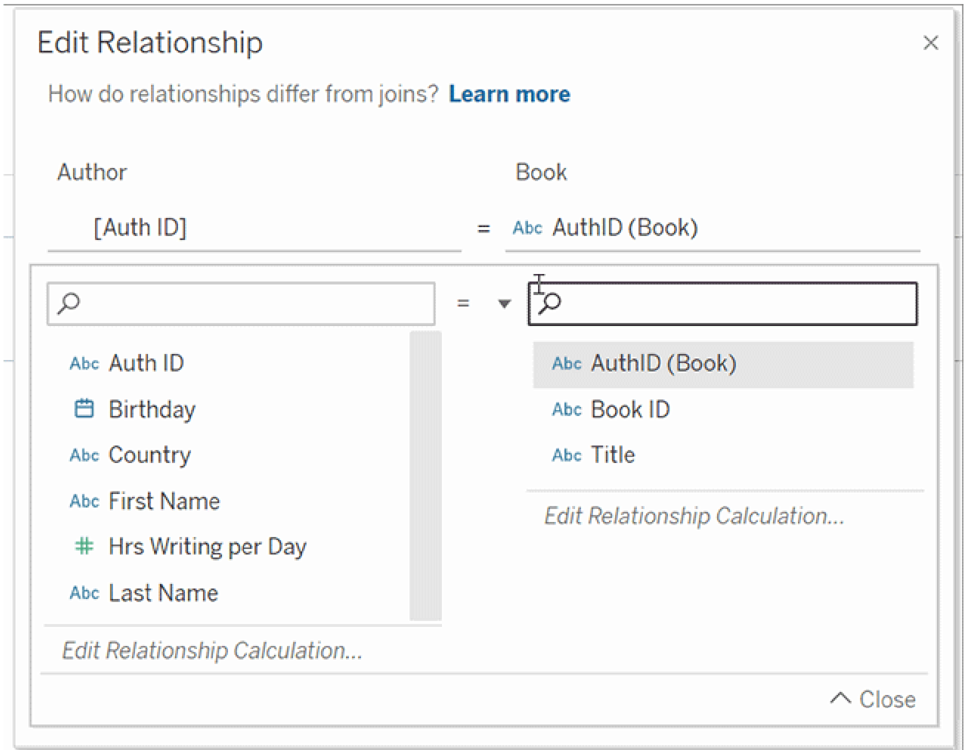
Now when you upgrade Tableau Desktop to the next major release, your recent workbooks are automatically imported and restored to your Start page along with your recent connections list and any organization-specific settings on the Discover pane. In previous releases, while you could still find your workbooks in your My Tableau Repository, your Start page would be empty.

Connect to and Prepare Data

Enhancements to relationships, data modeling, and the Data pane

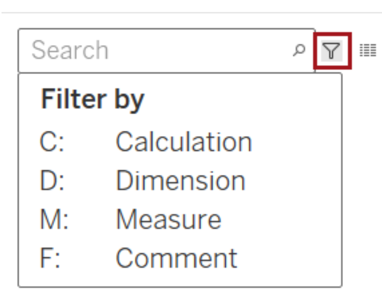
In 2020.2, relationships were introduced to Tableau. Relationships are a new, flexible way to combine data for multi-table analysis in Tableau. Data sources now support multi-table data models in a new logical layer. Tableau recognizes the separate tables in your data source and performs aggregations at the native level of detail for each table. To learn more about relationships and data modeling, see **Use relationships to combine data for multi-table analysis** on page 50 in this What's New topic.

In 2020.3, you can create relationships based on calculated fields, and compare fields used for relationships using operators in the relationship definition.



On the Data Source page, you can now move and replace tables in a data model using drag and drop.

In the Data pane, a filter has been added to search to make it easier for you to find fields of interest.



Also, you can now use Explain Data with data sources that contain multiple, related tables. Cardinality and Referential Integrity settings for relationships must be set up correctly for Explain Data to analyze multi-table, related data.

Override feature restrictions for cross-database joins

You can improve performance when joining data from a single file connection and a single database connection by allowing Tableau to perform the join using the database that you are connected to instead of Hyper. But certain conditions must be met to use this option.

To enable you to experiment and determine the optimal performance configuration when using this option, as an administrator, you can now override the file size, join type, and single file connection limitations and force Tableau to use the live database connection to perform the join.

For more information, see **Join Your Data** on page 883.

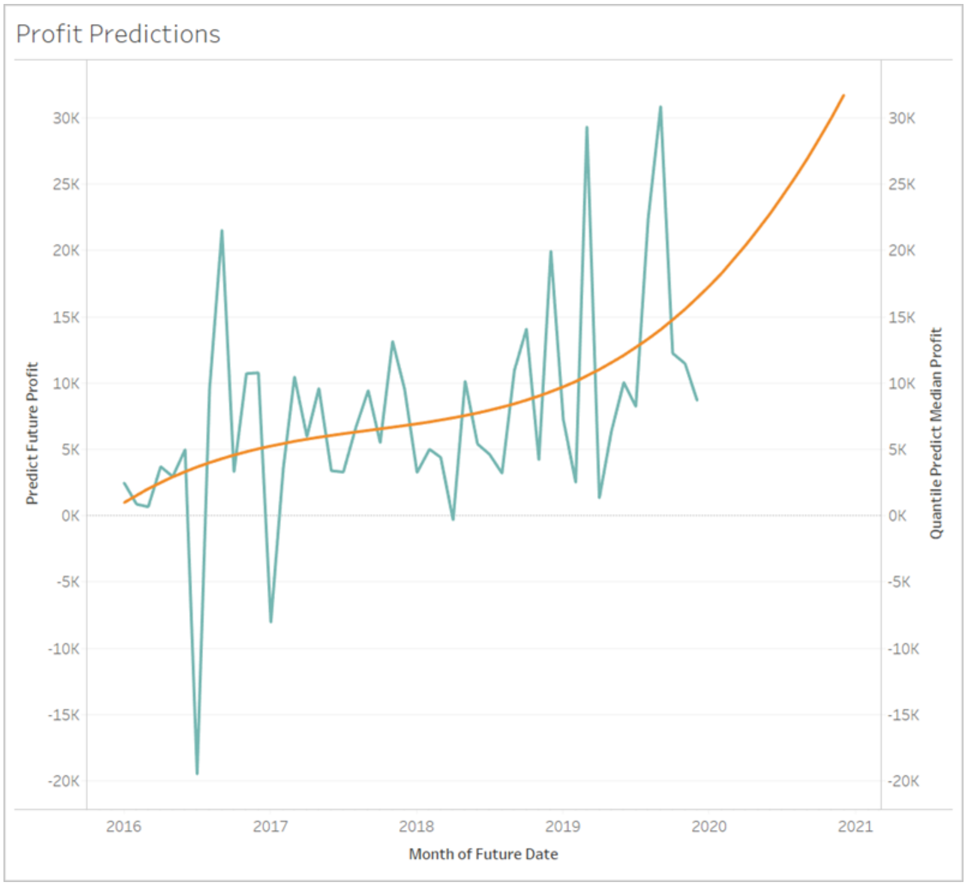
Combine spatial files with a Union join

With Tableau 2020.3, exploring geographic data is more flexible than ever. You can now combine spatial files with a Union join, as long as they are in the same data source and single-table format. For more information, see **Union Your Data** on page 942.

Design Views and Analyze Data

Create predictive models and forecasts with Predictive Modeling Functions

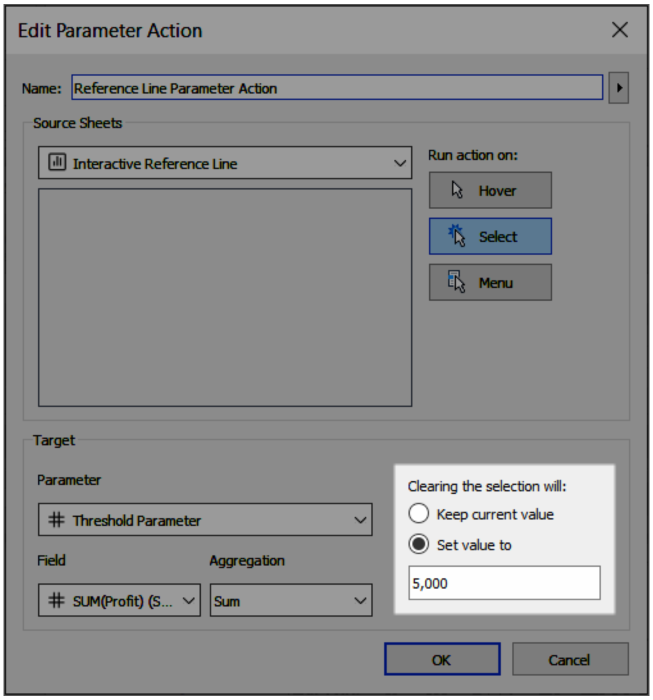
Using two new table calculation functions, MODEL_QUANTILE and MODEL_PERCENTILE, you can leverage a powerful statistical engine to make predictions and generate forecasts from your data, as well as identify unexpected values—all using the familiar table calculation interface.



Data can be filtered, aggregated, and transformed at all levels of detail, with inputs and predictions automatically recalculated to match the data on screen. For more information, see How Predictive Modeling Functions Work in Tableau. Or, check out this blog post, Generate predictions in Tableau with predictive modeling functions.

Specify value for when Parameter Action selection is cleared

Now, when using parameter actions, you can specify a parameter value when the selection in the viz is cleared. Previously, authors had no direct way to indicate a value for the parameter that the viz would return to during interaction. For more information, see **Parameter Actions** on page 1516.



Easily refresh your data source for dynamic parameters

Previously, dynamic parameter queries were only evaluated when the workbook was opened and you connected to the field's data source for the first time. Now, queries can be evaluated right from the workbook with the click of a button. In Tableau Desktop, simply press F5 or select Refresh from the data source's context menu. In Tableau Online or Tableau Server, click the Data Source Refresh button in the toolbar. For more information on dynamic parameters and how to use them, see this blog post.

Enhancements to Set Control

Recall that with a set control, your audience can add or remove set members by interacting directly with the card in the view. Now, your set control will respect any context filters you have in your view. This is especially important if you're an enterprise working with large data sources who uses context filters to improve performance, and to reduce the scope of the set's domain to include only the data of interest. Set controls are also now available in web authoring environments. For more information on set controls, see **Create Sets** on page 1186.

Download crosstab files using a dashboard object

With the Download object (previously named the "Export" object), dashboard viewers can now download selected sheets as crosstab files. Though you can add the Download object while authoring in Tableau Desktop, Tableau Online, or Tableau Server, crosstab download is possible only from a web browser.

For more information, see **Add dashboard objects and set their options** on page 2743.

Compare Fields with the IN operator

Easily compare lists of values with the new IN logical function. A Boolean calculation, the IN operator will take two or more fields and return a "true" value if any of the values in either field are the same. The IN operator combines the features of set intersections with the flexibility and speed of logical functions. For more information, see **Logical Functions** on page 2186.

Prepare, Author, and View Data on the Web

Filter data on your views and data sources in Tableau Online or Tableau Server

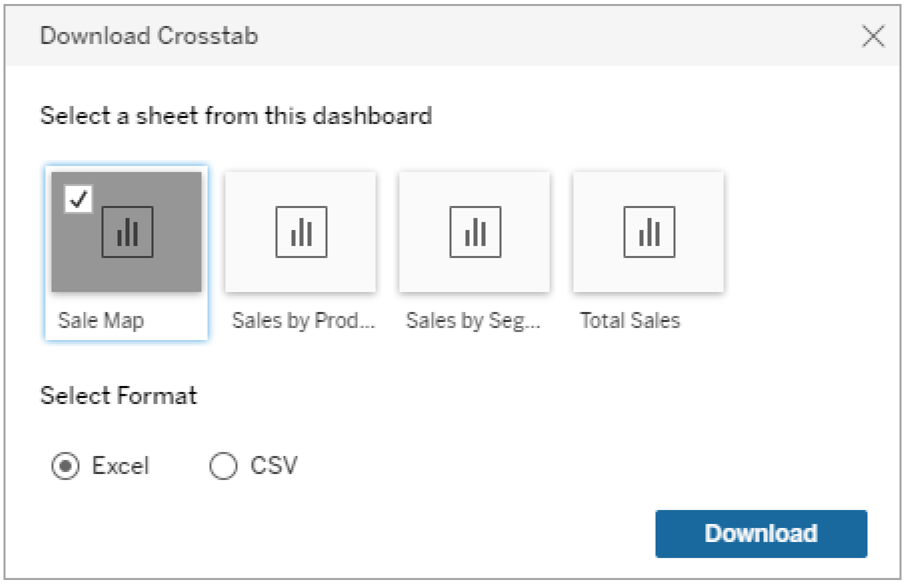
You can now create, edit, and remove filters for both workbooks and data sources while authoring in Tableau Online or Tableau Server. These improvements include categorical, discrete, and relative date filters. To set up a filter on a workbook, drag a field to the filters shelf.

If you have permissions to edit a data source, you can create a data source filter by selecting the Data Source tab. Under Filters, select **Add**.

For more information, see **Filter Data from Your Views** on page 1411 and **Filter Data from Data Sources** on page 1006

Download crosstab to Excel in Tableau Online or Tableau Server

You can now download views to .xlsx file format when using Tableau Online and Tableau Server. This download option retains numerical formatting, including international separator characters. For more information, see **Download Views and Workbooks** on page 3328.



Use Mapbox or WMS service connections in Tableau Online or Tableau Server

As of Tableau 2020.3, you can now publish a workbook with a Mapbox or WMS server connection. For more information, see **Use Mapbox Maps** on page 1993 or **Use Web Map Service (WMS) Servers** on page 1996.

Accessibility improvements to relative date filters in Tableau Online or Tableau Server

Now, relative date filters are both keyboard accessible and screen reader compatible when in web authoring environments. You can find more information on accessibility in Tableau here.

Create filter and Go to Sheet actions in Tableau Online or Tableau Server

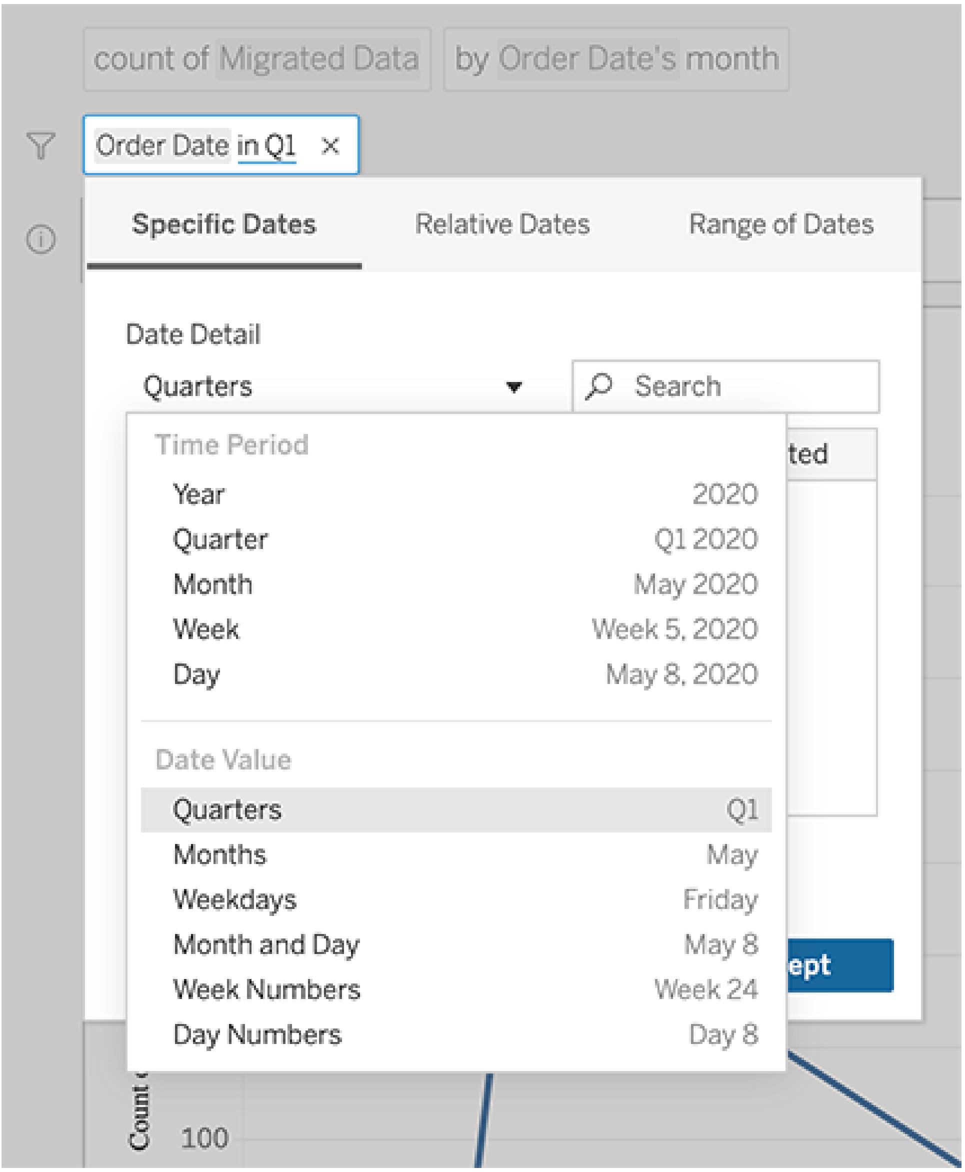
For the first time, you can create actions while authoring in Tableau Online and Tableau Server. Filter and Go to Sheet actions are now available from both dashboards and individual sheets. In web-authoring mode, filtering on selected fields has been greatly streamlined, letting you specify source and target fields without opening a separate dialog box.

For more information, see **Filter Actions** on page 1491 and **Navigate from one view to another view, dashboard, or story** on page 1571.

Precisely refine dates with Ask Data

You can now refine existing queries to show specific dates. Time Period options show one continuous date range; Date Value options show ranges that can repeat in multiple time periods. (For example, to see combined sales performance for Q1 across multiple years, under Date Value, choose Quarters.)

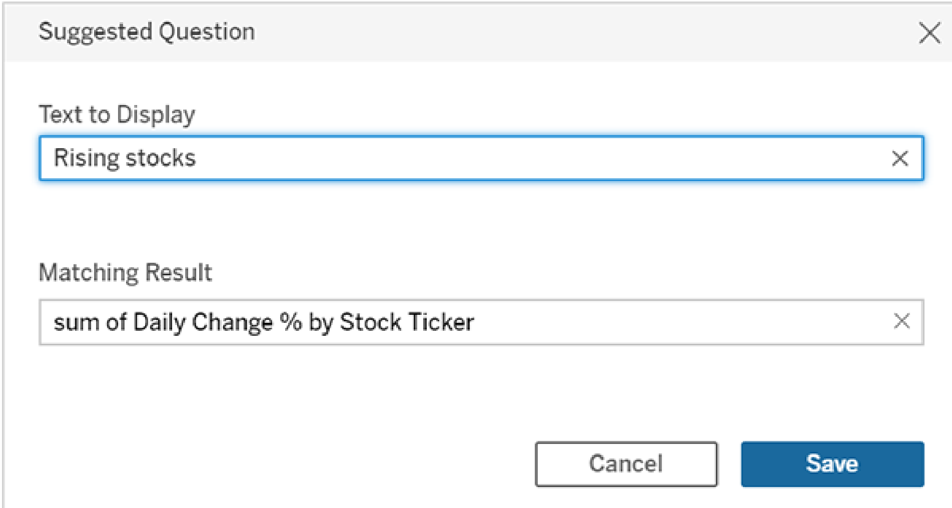
For more information, see **Adjust date filters** on page 1118.



Display suggested questions in natural language with Ask Data

If you're a data source owner or Tableau administrator, you can now display suggested questions in language familiar to your users, rather than as formal queries. Users can either click the displayed text or enter it in the “Ask Data...” box.

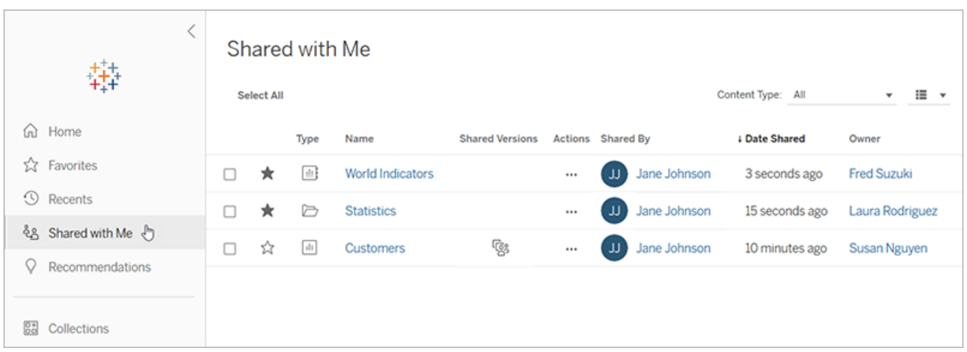
For more information, see **Optimize Data for Ask Data** on page 1132.



Collaborate and Explore on Tableau Online and Tableau Server

Find items shared with you

When an item is shared with you, it’s added to your Shared with Me page. This makes it easy to find shared items later, without needing to search your inbox or request another link. For more information, see **Share Web Content** on page 3243.

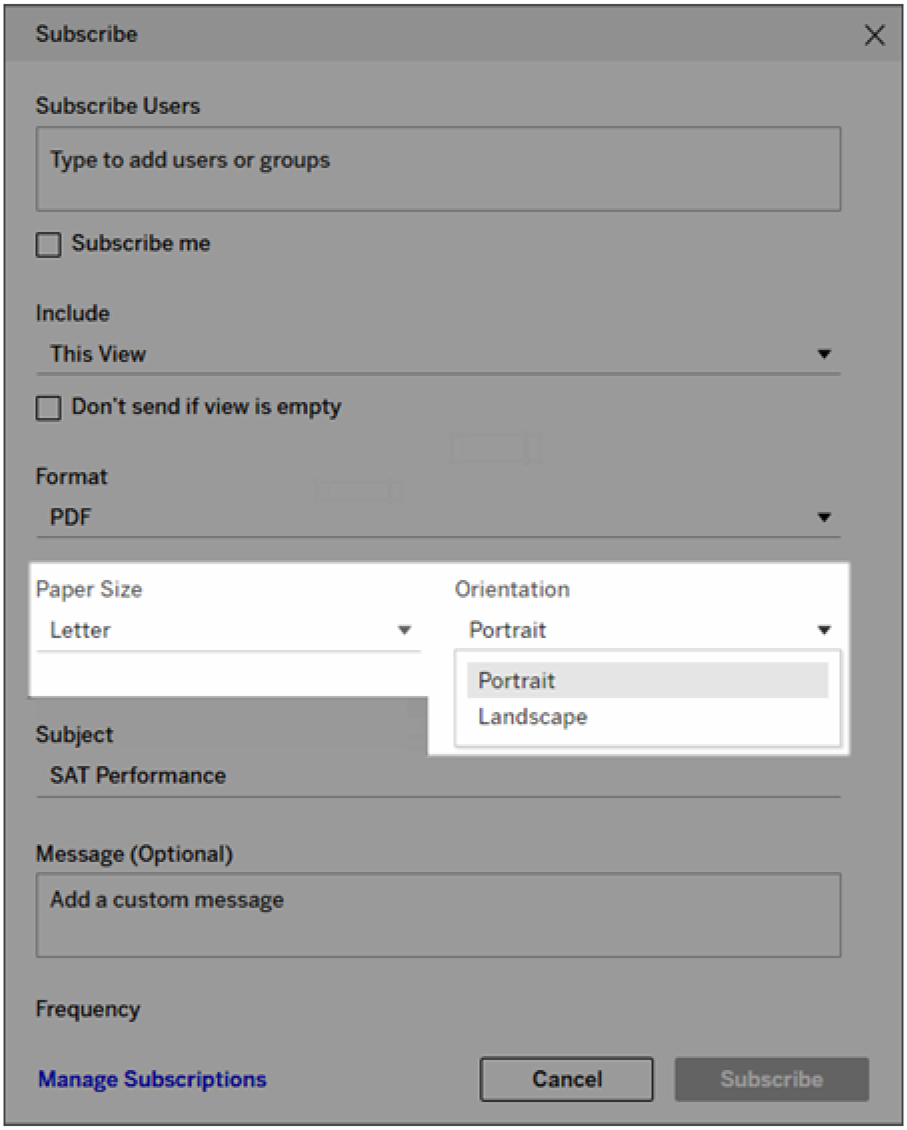


Get subscriptions when data refreshes

You can now set the frequency of email subscriptions to only send when data refreshes. Subscriptions let you receive email snapshots of a view or workbook at a frequency and schedule you choose. Setting the frequency to When Data Refreshes means you’ll know that the snapshot in your email contains the latest data. For more information, see **Create a Subscription to a View or Workbook** on page 3249..

Pick new options for PDF subscriptions

You can now choose the page size and orientation for PDF attachments when you create or modify a subscription to a view or workbook in Tableau Online or Tableau Server. For more information, see **Create a Subscription to a View or Workbook** on page 3249.



Accessibility improvements for organizing your favorites

You can now organize your favorites using a keyboard to perform drag and drop interactions. For more information, see **Keyboard Accessibility for Tableau on the Web** on page 3164.

Add-ons

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

The following features are included in this release:

See data quality warnings in Tableau Desktop - When you open a workbook in Tableau Desktop that has a data quality warning, the warning icon (either a blue circle or a yellow triangle with an exclamation point) appears on the Data tab on the sheet page of the workbook. For more information, see "Set a Data Quality Warning" in Tableau Server or Tableau Online Help.

'Sensitive data' warning type added - You can use this new warning type to set a data quality warning for sensitive data in your organization. For more information, see "Set a Data Quality Warning" in Tableau Server or Tableau Online Help.

Embedded assets are listed in External Assets - You can now see embedded assets listed in the External Assets page. For more information, see "Embedded asset appears in External Assets" in Tableau Server or Tableau Online Help.

Tag external assets - Add tags to databases, tables, and columns, which enables you to, among other things, tag tables and columns with personally identifiable information (PII). For more information, see **Use Tags** on page 3180.

Version 2020.2

Added in version 2020.2

Connect to and Prepare Data

- **Use relationships to combine data for multi-table analysis** on page 50
- **Configure proxy for connection to Snowflake** on page 51

Design Views and Analyze Data

Prepare, Author, and View Data on the Web

Collaborate with Tableau Online and Tableau Server

Add-ons

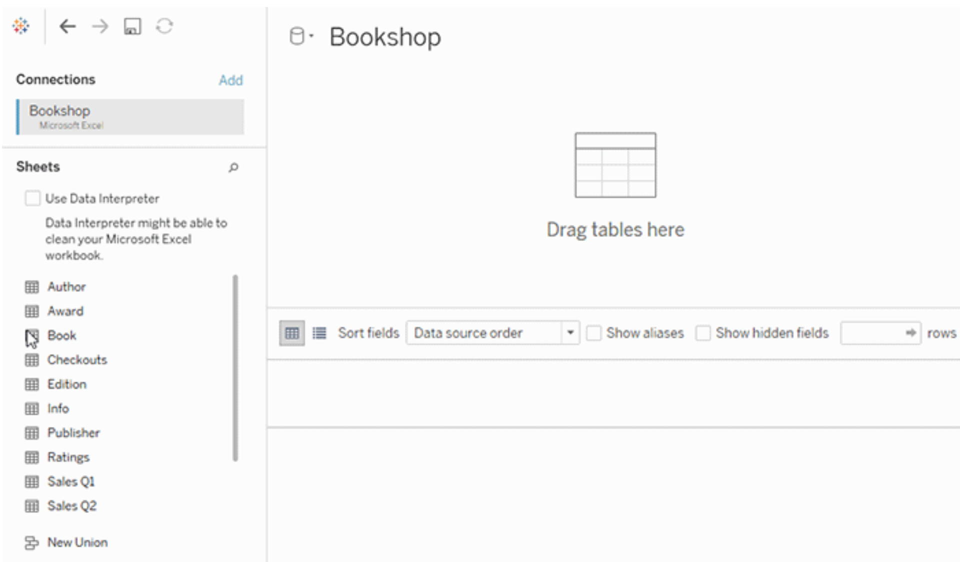
Connect to and Prepare Data

Use relationships to combine data for multi-table analysis

Relationships are a new, flexible way to combine data for multi-table analysis in Tableau. Data sources now support multi-table data models in a new logical layer. Tableau recognizes the separate tables in your data source and performs aggregations at the native level of detail for each table.

Creators can combine tables using relationships. Using relationships between logical tables makes it easier to explore and analyze multi-table data at the right level of detail, without having to write specialized calculations to control aggregations like averages and totals.

You do not need to specify join types for relationships; during analysis Tableau automatically selects the appropriate join types based on the fields and context of analysis in the worksheet.



Example of a data source that uses relationships between tables. Tables in the logical layer act like containers for tables in the physical layer of the data model.

In support of multi-table analysis, several parts of the Tableau interface have changed. The Data Source page (canvas, data grid), View Data window, and the Data pane in the worksheet have all been updated to support a multi-table analysis experience. For more information, see Changes to different parts of the interface.

Note: View Data column order has changed in Tableau 2020.2 and later. This change is required to support relationships and logical tables. For details, see View Data column order changes in Tableau 2020.2 and later.

For more information about relationships and changes to data sources, see these topics:

- What's Changed with Data Sources and Analysis in 2020.2
- Questions about Relationships, the Data Model, and Data Sources in 2020.2
- The Tableau Data Model
- Relate Your Data and Optimize Relationships Using Performance Options
- How Analysis Works for Multi-table Data Sources that Use Relationships
- Troubleshooting Multi-table Analysis
- Walkthrough: Don't Be Scared of Relationships

Watch a video: For an overview of data source enhancements and an introduction to using relationships in Tableau, see this 5-minute video.

Learn more about how relationship queries work in this Tableau blog post: Relationships, part 1: Introducing new data modeling in Tableau.

Configure proxy for connection to Snowflake

You can use the custom driver parameters field to configure a connection through a proxy for Snowflake. See **Snowflake** on page 641.

Connect to Azure Synapse Analytics (new connector name)

Microsoft renamed their product, and the connector for Tableau did too. The Azure SQL Data Warehouse connector is now called Azure Synapse Analytics. See **Azure SQL Synapse Analytics** on page 480.

Connect to Esri ArcGSI Server

Now you can connect Tableau to an Esri ArcGIS Server. See **Esri ArcGIS Server** on page 498.

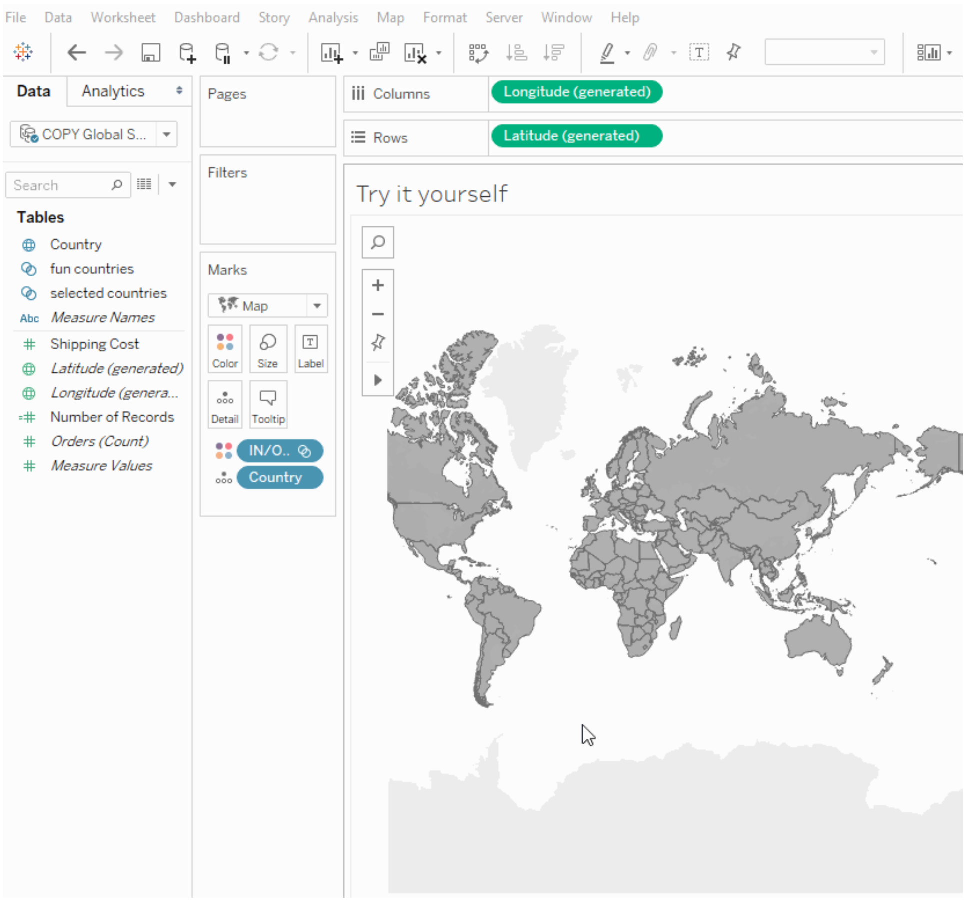
Connect to spatial fields in Oracle

You can now connect to spatial data fields in an Oracle data source. For more information, see **Connect to Spatial Data in a Database** on page 1799.

Design Views and Analyze Data

Add a Set Control to let users quickly modify the members of a set

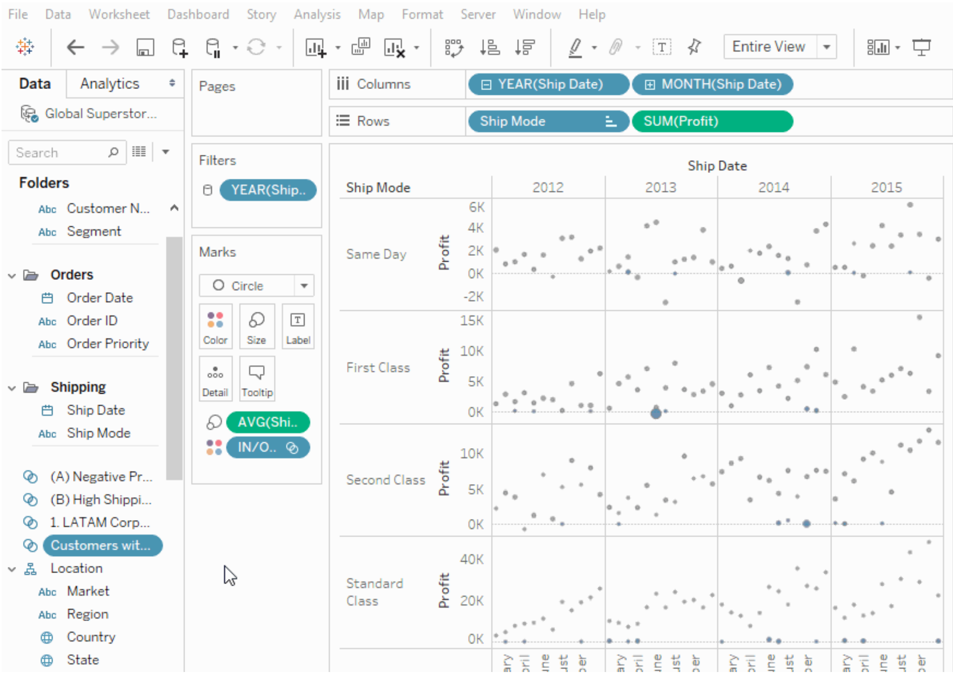
With a Set Control, your audience can select the members of a set without necessitating edit mode. Users can simply add or remove set members by interacting directly with the card in the view. Coupled with Set Actions, you have two powerful, complementary features that make it easy to visualize the set you're interested in.



For more information, see Create Sets, or check out this blog post, Powerful analytics in a user-friendly package with the set control.

Add or remove values from a set by interacting directly with the viz

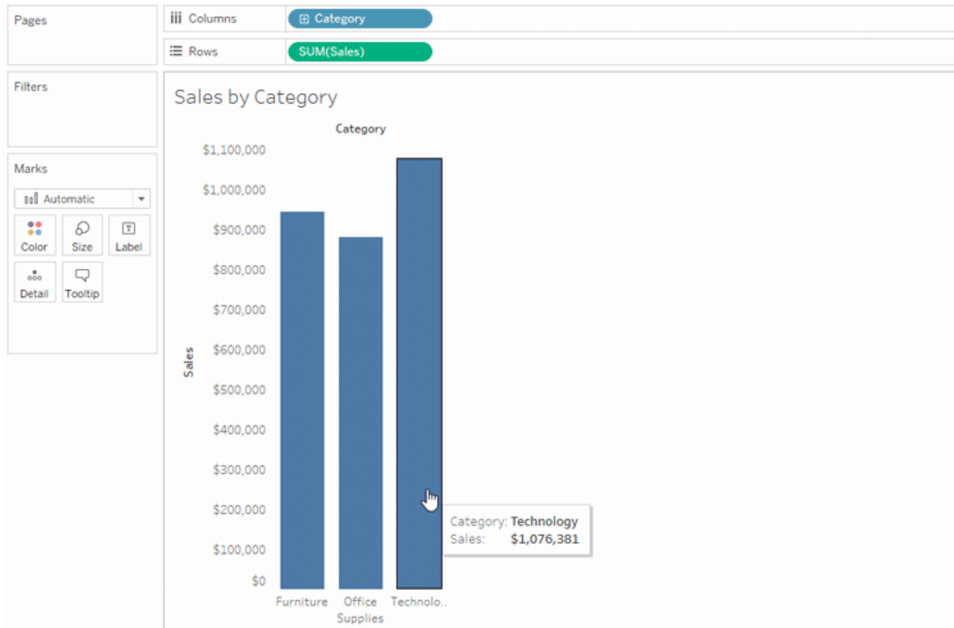
Now, using Set Actions, you and your audience can use the viz itself to incrementally add values to or remove values from a set. Simply interact with the viz and choose which marks to include or exclude. Coupled with the new Set Control feature, you have two powerful ways to gain deeper insights into your data.



For more information, see Set Actions.

Control the fields used for analysis by Explain Data

You can now specify which fields are modeled during analysis. With the ability to select which fields are included and excluded from explanations, dashboard authors can hide redundant results and add fields that are excluded by default.

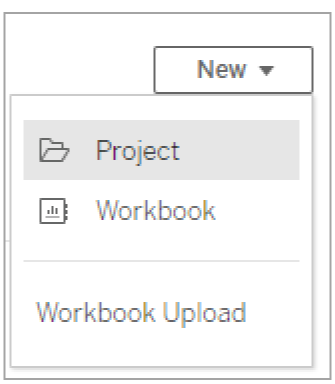


For more information, see **Discover Insights Faster with Explain Data** on page 2035 in **Discover Insights Faster with Explain Data** on page 2035.

Prepare, Author, and View Data on the Web

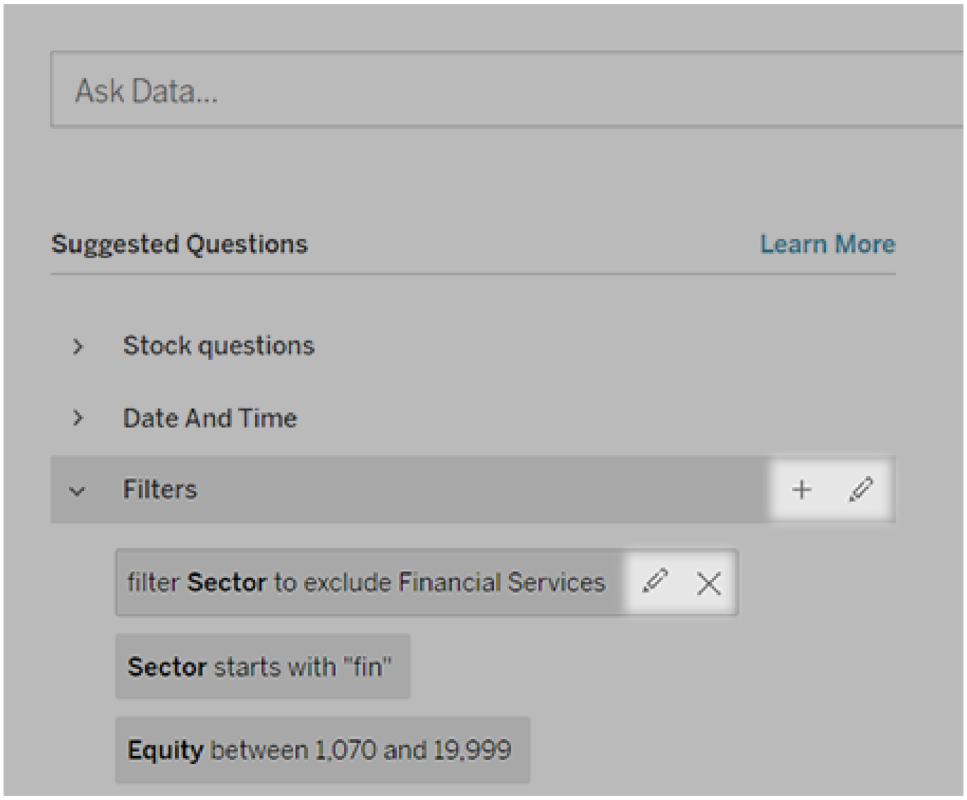
Upload and open workbooks in a browser

Users can now upload a Tableau workbook to your Tableau site without publishing through Tableau Desktop, the REST API, or TabCmd. In a Tableau site, users with a Creator or Explorer (can publish) site role can select **New** > **Upload Workbook** to upload a .twb or .twbx file (up to 50 MB) to a project. For more information, see **Upload Workbooks to a Tableau Site** on page 3146.



Customize suggested questions for Ask Data

Data source owners and Tableau administrators can customize suggested questions for a data source, tailoring them to the needs of users in your organization. For more information, see **Optimize Data for Ask Data** on page 1132.

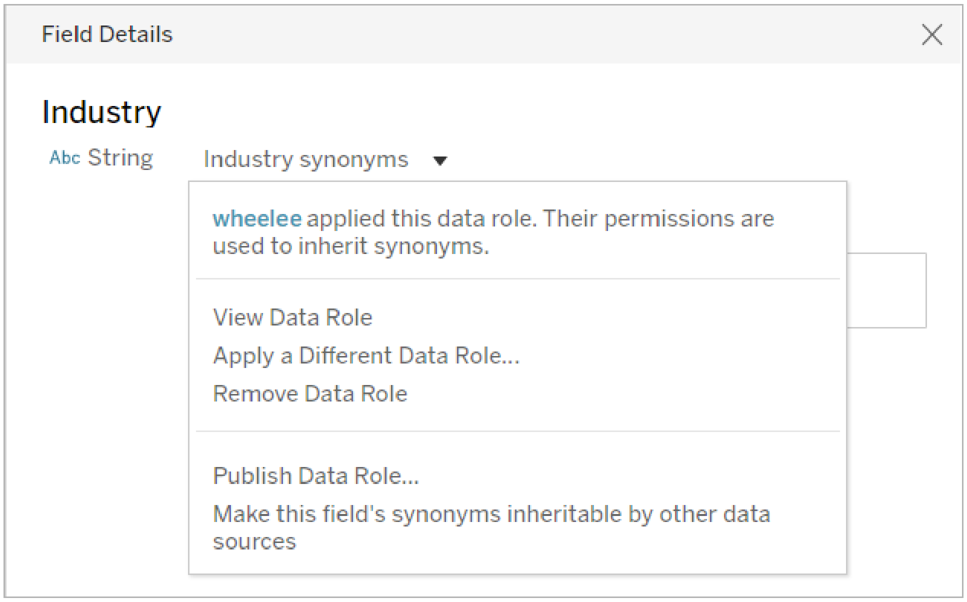


Add synonyms for multiple data sources with Ask Data

When a field or value has similar usage throughout your organization, you can use data roles to apply the same synonyms across multiple data sources. For example, data roles are a great choice if every subsidiary uses the same names for business departments, or if product lines have a consistent set of sub-products.

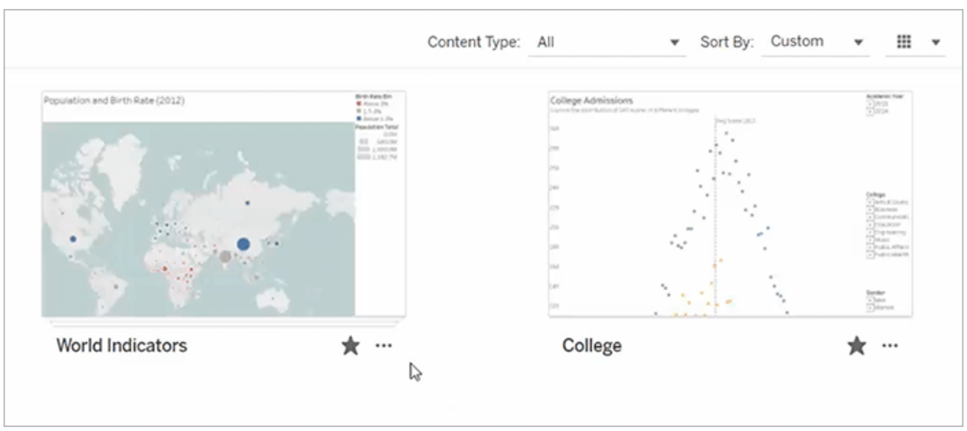
Using data roles for synonyms saves you a lot of time, because you can manage them centrally in Tableau Online or Tableau Server. From Ask Data, you can publish new data roles and apply existing ones created in Ask Data or Tableau Prep.

For more information, see **Add synonyms for multiple data sources** on page 1142.



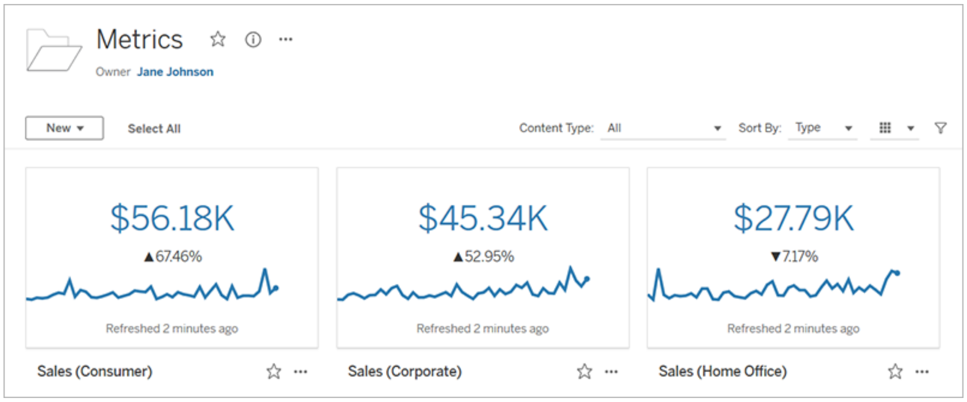
Organize your favorite content

Drag and drop cards to create a custom sorting order for your favorites. This option is available when interacting with your favorites in grid view on Tableau Online or Tableau Server. For more information, see **Mark Favorites** on page 3195.



Create metrics to track your data

Metrics help you track changes to your data. You can create metrics from a view on Tableau Online or Tableau Server. Use metrics to monitor key numbers, like daily sales. For more information, see **Create and Troubleshoot Metrics** on page 3152.



Collaborate with Tableau Online and Tableau Server

Dashboard download to crosstab improvements

You can now download to crosstab for dashboards, without selecting a sheet first. Before, the button was disabled until you selected a sheet to export to a .csv file. Now, you can select Download -> Crosstab to select the appropriate sheet from a list of options. For more information on exporting to crosstab, see **Export Data from Tableau Desktop** on page 3034.

Enhancements to editing Legends and Tooltips

Now with Tableau 2020.2, resizing and changing the placement of your Tooltips will persist even when you log out. For more information, see **Format Titles, Captions, Tooltips, and Legends** on page 2937.

Add-ons

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

The following features are included in this release:

Raise the visibility of important warnings - You can set a data quality warning to be highly visible, so that users of a visualization that's affected by the warning see a notification, and the warning displays in a more prominent yellow color elsewhere in Tableau. For more information, see "Set a Data Quality Warning" in Tableau Server or Tableau Online Help, and "Use Data Details to see visualization information" in **Explore and Inspect Data in a View** on page 2707.

Dashboards appear in the Lineage pane - When you select the Dashboards item in the lineage, you can see a list of downstream dashboards. For more information, see "Use Lineage for Impact Analysis" in Tableau Server or Tableau Online Help.

Filter external assets on Certified and data quality warnings - In addition to using search to narrow down your choices on External Assets, you can now filter databases, files, and tables by Certified and by data quality warnings.

More filters when connecting - When you connect to data, you can filter databases and table on connection type.

Version 2020.1

Added in version 2020.1

Install and Deploy Tableau

Connect to and Prepare Data

Design Views and Analyze Data

Prepare, Author, and View Data on the Web

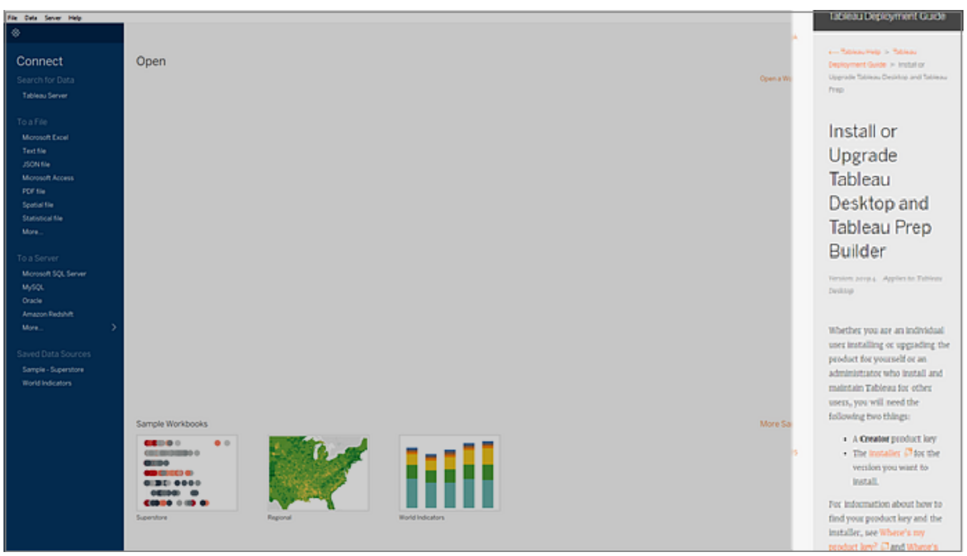
Add-ons

Install and Deploy Tableau

Customize the Discover pane

Now you can customize the Discover pane that shows on the Start page in Tableau Desktop to show your custom content instead of the links, blog posts and Tableau news that shows by default. Simply create your own web page hosted on a webserver that you provide and then use the new `DISCOVERPANEURL` install property to point to it. When you launch Tableau Desktop your own custom content will show instead. To help you get started quickly, Tableau Desktop provides a template that models the same layout as the Tableau provided Discover pane. Simply download the template and replace the links with your custom content.

Use this feature to provide links and information to content that is specific to your organization to make it easier for your users to get started using Tableau quickly.



To configure Tableau Desktop to show your custom discover pane content, during install from the command line (Windows), include the property `DISCOVERPANEURL="https://yourwebpageURL"`. If you want to update the discover pane content after installation is complete, add the `DiscoverPane` key with the `DiscoverPaneURL` set to your custom URL to the registry (Windows) or in the `.plist` file (MacOS) for each computer where Tableau Desktop is installed.

> **Note**: This feature is currently only available for Tableau Desktop.

For more information, see Installer properties or Change Installation Settings after Installation in the Tableau Desktop and Tableau Prep Deployment Guide.

Support for login-based license management

Login-based license management, helps you manage licensing for users with Creator roles on Tableau Server and Tableau Online. Users with Explorer or Viewer roles cannot use this feature. If you're using Role Based Subscriptions with Tableau Server or Tableau Online, you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Online, you don't need to manage any product keys at all.

Connect to and Prepare Data

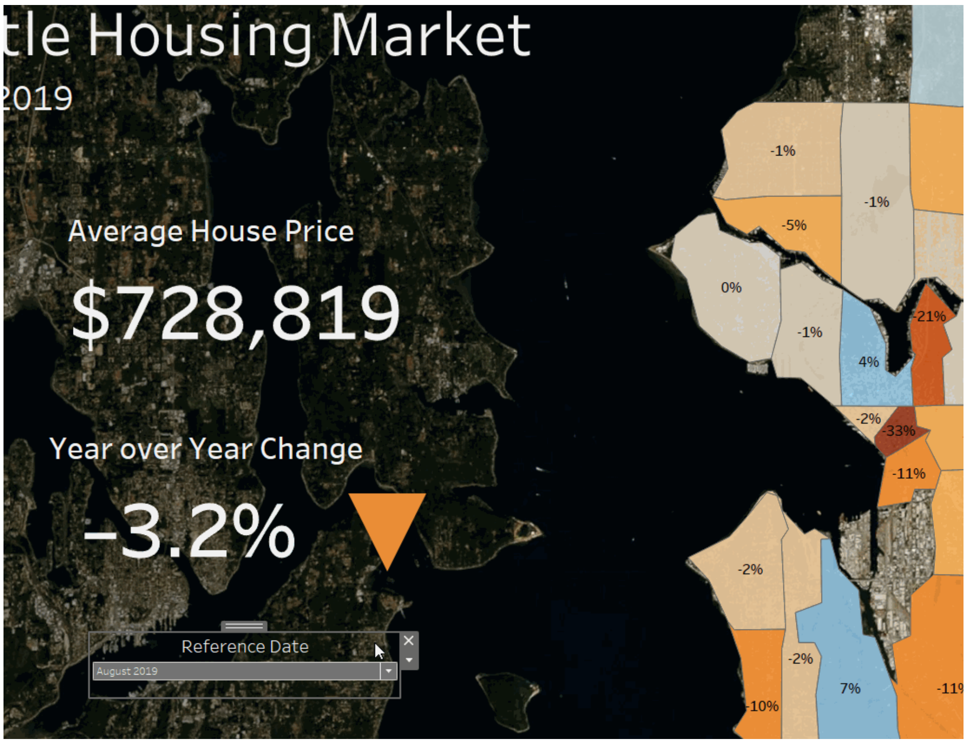
Salesforce connector has updates

We've refreshed our Salesforce connector to provide faster performance, greater flexibility, and even more data to play with. First, we changed the way we connect to Salesforce data; the connector now dynamically switches APIs depending on how large your dataset is, greatly increasing performance. Also, the Salesforce connector now supports using the SOQL language in the same way that Custom SQL can be used today, adding flexibility to your analysis. Finally, you will now be able to connect to additional data objects like Salesforce Activity History and Campaign Influence, giving you even more data to see and understand in Tableau.

Design Views and Analyze Data

Refresh parameters automatically with dynamic parameters

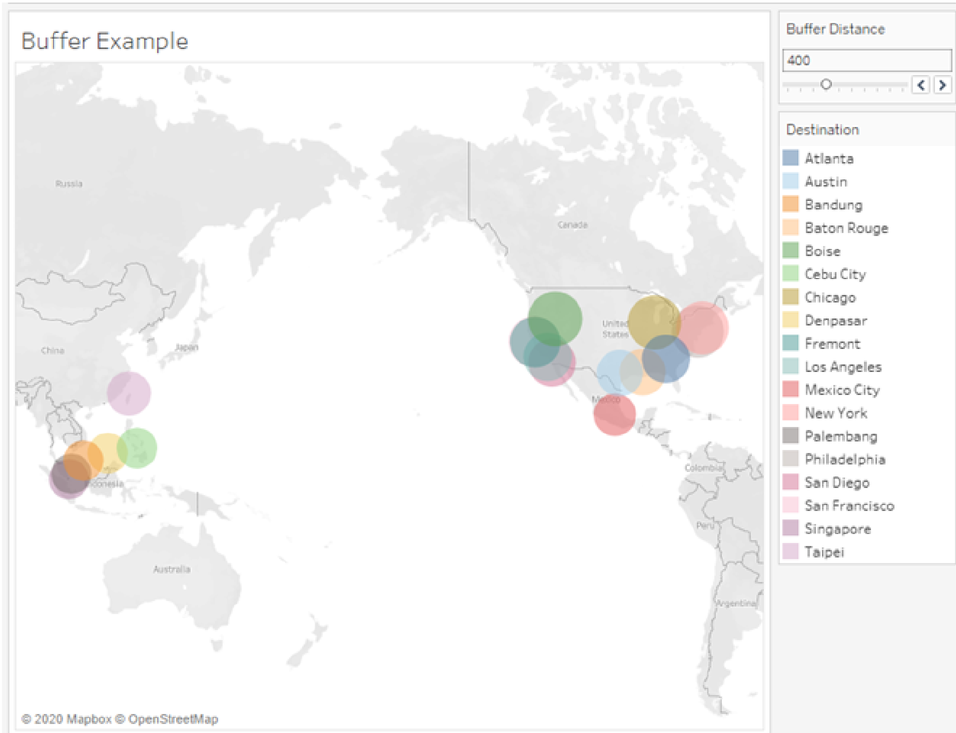
Now you can set a parameter's current value to be the result of a single-value, view-independent calculation. In addition, you can refresh a parameter's list of values (or domain) based on a data source column. This means that each time the workbook is opened and Tableau connects to the data source referenced by the parameter, every place in the workbook that references the parameter will use the latest value or domain.



For more information, see this blog post on dynamic parameters. Also, see **Create Parameters** on page 1218.

Visualize area on a map with the Buffer function

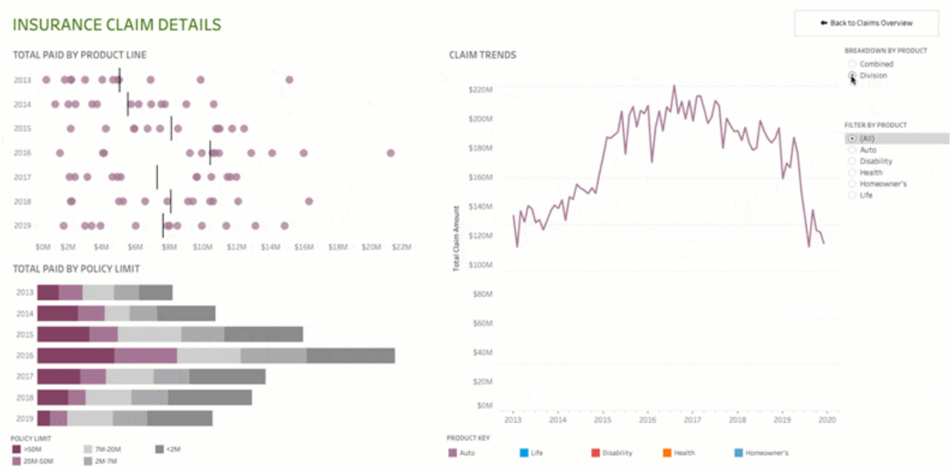
Finding the area around a point once required complicated calculations that were not guaranteed to be accurate. With the new Buffer function, we can define the exact distance from a point we want to visualize. This area will accurately reflect the actual area of a map, even when viewing points distorted by the map projection. For more information, see **Spatial Functions** on page 2245.



Animate mark transitions

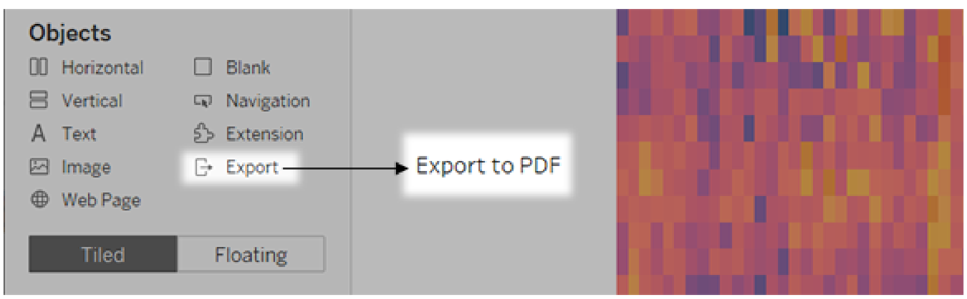
Animate visualizations to better highlight changing patterns in your data, reveal spikes and outliers, and see how data points cluster and separate.

Animations visually transition between filter, sort, and zoom settings, different pages, and changes to filter, parameter, and set actions. As visualizations animate in response to these changes, viewers can more clearly see how data differs, helping them make better informed decisions. For more information, see **Format Animations** on page 2887.



Export dashboards to PDF, PowerPoint, or images

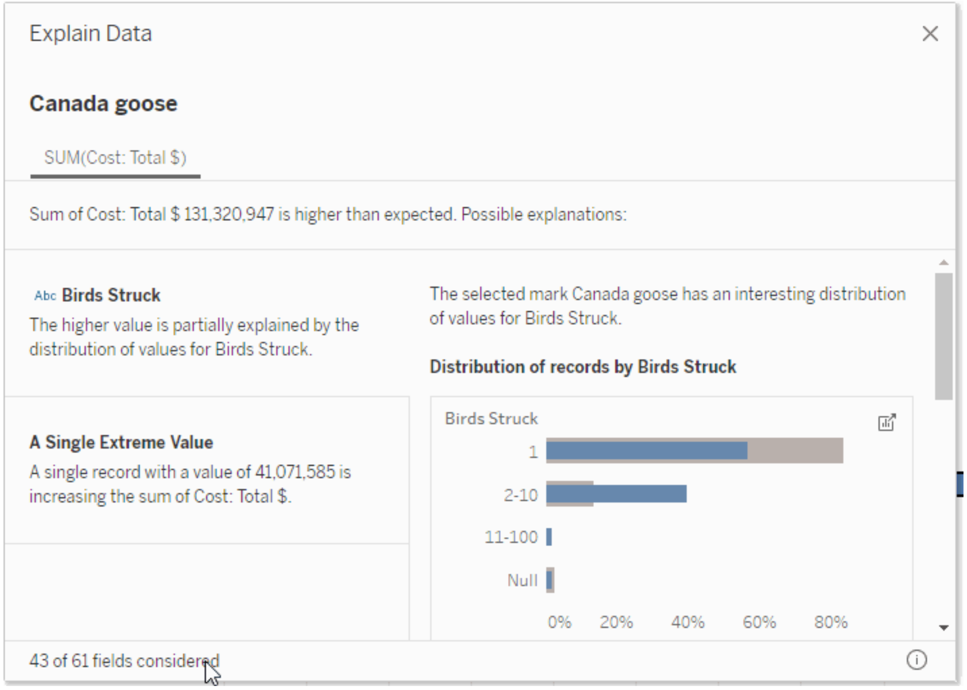
The new Export object lets your audience quickly create a PDF file, PowerPoint slide, or PNG image of a dashboard. Style and formatting options are similar to Navigation objects (previously known as “Button” objects). You can choose to display text or an image, specify custom border and background colors, and provide informational tooltips.



For more information, see **Add dashboard objects and set their options** on page 2743.

View status in Explain Data

The Explain Data window now includes a status bar that provides explanation alerts, messages, and a summary of investigated fields. You should also experience improved performance when using Explain Data on wide data sets (> 500 columns). Also, the Explain Data window can now be resized.



For more information, see **Discover Insights Faster with Explain Data** on page 2035 and **Analyzed Fields in Explain Data** on page 2088.

Prepare, Author, and View Data on the Web

Automatically add phone layouts to new dashboards

The web authoring environment now includes the command, **Dashboard** > **Add Phone Layouts to New Dashboards**, which is selected by default. In most cases, the automatically generated phone layout will be ideal for smaller screens, but you can always edit it in Tableau Desktop. For more information, see **Create Dashboard Layouts for Different Device Types** on page 2801.

Improved Tooltip formatting on the Web

Formatting your work on the web just got easier with the improvements to Tooltip formatting. Now you can include Viz in Tooltip and custom indents when you format your workbooks in the web authoring environment. For more information, see **Create Views in Tooltips (Viz in Tooltip)** on page 1475.

Add-ons

Tableau Catalog updates - part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

The following features are included in this release:

Added connectors - Connect to more data from external assets. We've added support for the following connectors: Google Big Query, Google Drive, Google Sheets, Box, Dropbox, and OneDrive.

Create workbooks - Create a workbook from the External Assets page or from a database or table page for Catalog-supported connections. For more information, see "Tableau Catalog users create workbooks from external assets" in **Creators: Get Started with Web Authoring** on page 3119.

Filter on Certified and data quality warnings - Filter data sources, databases, files, and tables by Certified, filter data sources by connections, and filter databases and tables by data quality warnings. For more information, see "Open the Connect to Data page" in **Creators: Connect to data on the web** on page 3135, or "Connect from Tableau Desktop" in **Connect to Data by Using Tableau Server or Tableau Online** on page 445.

Enhanced Data Details pane -

- See a sparkline with the number of times a sheet or dashboard was viewed over the last 12 weeks.
- Easily see that there are Warnings on a view.
- Expand a field to see its metadata, such as a description or calculation.
- When a field in a workbook inherits its description, see where the description comes from.

  For more information, see "Use Data Details to see visualization information" in **Explore and Inspect Data in a View** on page 2707.

Version 2019.4

Added in version 2019.4

Install and Deploy Tableau

Connect to and Prepare Data

Design Views and Analyze Data

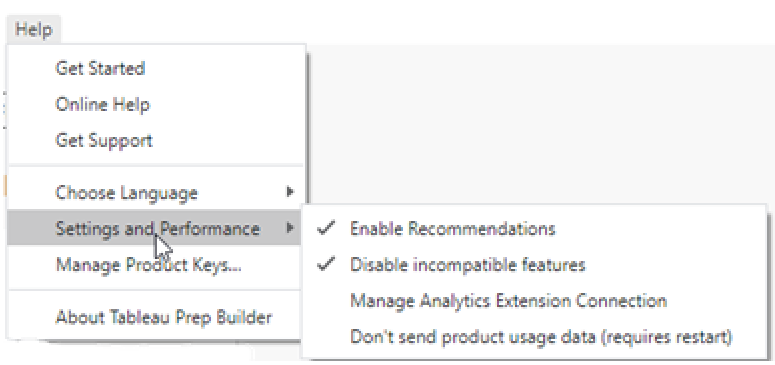
Prepare, Author, and View Data on the Web

Install and Deploy Tableau

Share usage data with Tableau

Tableau collects data that helps us learn how our products are being used so we can improve existing features and develop new ones. All usage data is collected and handled according to the Tableau Privacy Policy. But you can opt out if you don't want to participate.

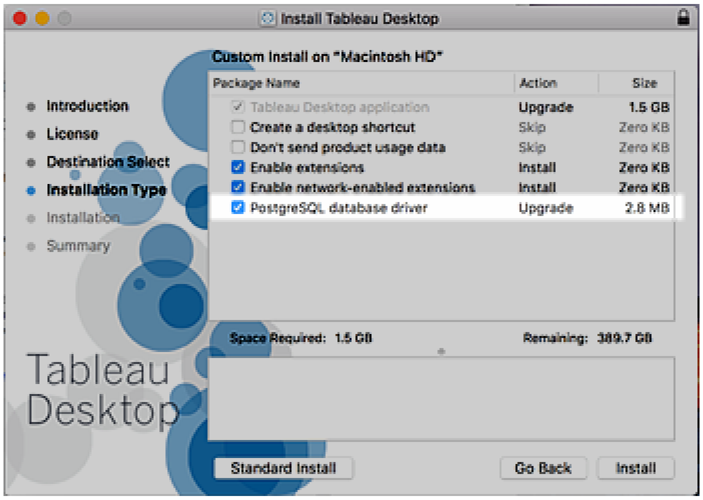
The opt out option has changed. Usage data collection is enabled by default. If you don't want to participate, select the **Don't send product usage data** check box from the installer or Customize installer (MacOS) dialog or from the Help menu after installation.



For more information, see Turn off usage reporting in the Tableau Desktop and Tableau Prep Deployment Guide.

Default database drivers no longer automatically installed for MacOS

When installing Tableau Desktop on Mac, we only install the PostgreSQL database driver automatically. If you need other drivers to connect to your data, you can install them from the Driver Download page.



Connect to and Prepare Data

Support added for six data connectors

Use the following connectors to connect Tableau to your data:

- **Alibaba AnalyticDB for MySQL** on page 453
- **Alibaba Data Lake Analytics** on page 455
- **Alibaba MaxCompute** on page 457
- **Kyvos** on page 547
- **LinkedIn Sales Navigator** on page 549
- **Qubole Presto** on page 601

Support added for connector plugins

You can connect to data using connectors created with the Tableau Connector SDK. For more information, see **Connectors Built with the Tableau Connector SDK** on page 690 For information about the Tableau Connector SDK, see the Tableau Connector SDK on GitHub.

Convert an SAP HANA level hierarchy in Tableau

When you connect Tableau to an SAP HANA calculation view with a level hierarchy, the structure of the SAP HANA level hierarchy appears in the **Data** pane. If you want to modify the

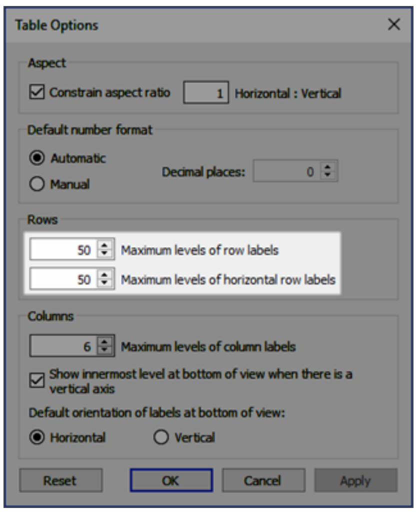
SAP HANA level hierarchy in Tableau, you need to convert it to a user hierarchy. For more information, see **Convert an SAP HANA level hierarchy to a user hierarchy** on page 612.

Design Views and Analyze Data

Author tables with up to 50 columns and scroll horizontally

Now you can author wide tables with up to 50 columns. In addition, wide tables with discrete fields can now be scrolled horizontally.

To change the maximum number of row and horizontal row labels, authors must use the Table Options dialog box in Tableau Desktop.



For more details, see **Define Table Structure** on page 2966. Also, see this post on 50-column tables from Tableau Public Ambassador Marc Reid on the datavis.blog.

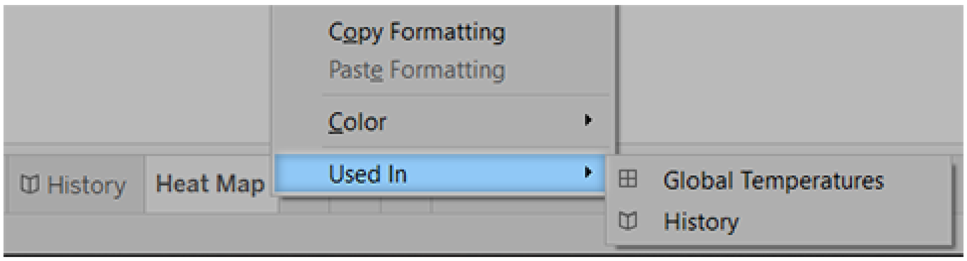
More data considered by Explain Data

Explain Data now analyzes unvisualized measures in your data source that might impact the value of a mark. Explanations show the relationship between the target measure and the unvisualized measure for all marks in the source visualization.

Calculated fields with aggregation type AGG (such as ratio calculations) can now be analyzed. For all unvisualized dimension and measure explanations, unvisualized calculated fields are considered as candidates for explanations.

Go from a sheet to related dashboards and stories

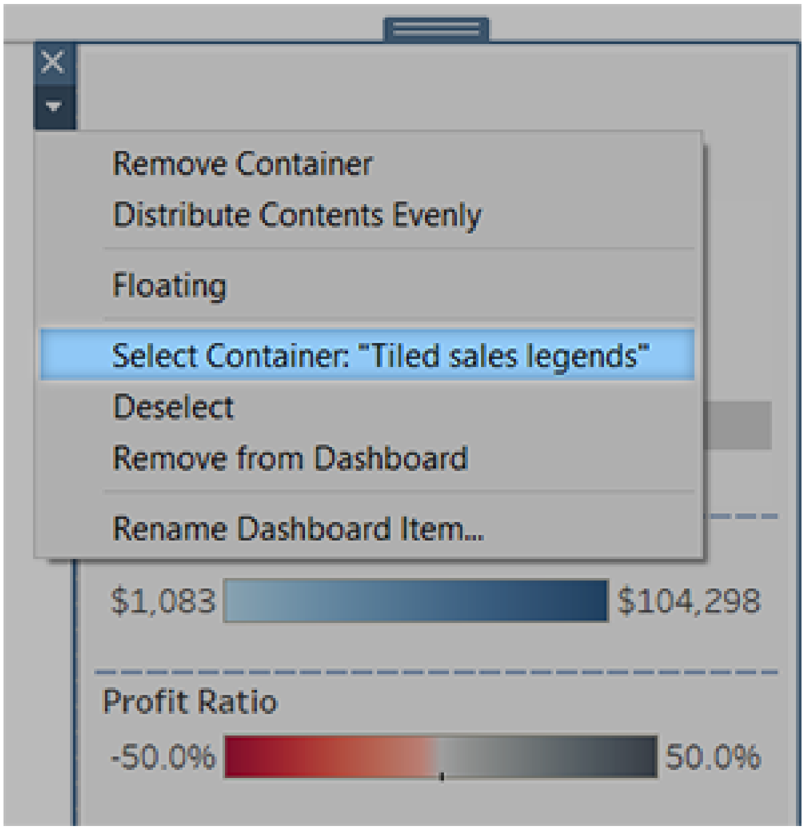
To quickly go to dashboards and stories a sheet appears in, right-click the sheet tab and open the new **Used in** menu. Think of this as a companion to the **Go to Sheet** command available from dashboards and stories. Using the two features together, you can quickly optimize sheet designs and data to better fit dashboard layouts and story points.



For more information, see **Navigate between sheets, dashboards, and stories** on page 2837.

Easily identify and select parent layout containers

From the menu for each dashboard object, customized names for parent layout containers now appear, letting you easily identify and select them.



For more information, see **Group items using layout containers** on page 2757.

Automatically add phone layouts to existing dashboards

To automatically create phone layouts whenever you open old dashboards that lack them, choose **Dashboard** > **Add Phone Layouts to Existing Dashboards**.

For more information, see **Automatically add phone layouts** on page 2803.

Ensure data security with Sandboxed dashboard extensions

New Sandboxed dashboard extensions run in a protected environment without access to any other resource or service on the web. As an author, you can quickly incorporate Sandboxed extensions into dashboards because they don't require approval from your Tableau administrator.

To see all the available Sandboxed extensions, go to extensiongallery.tableau.com.

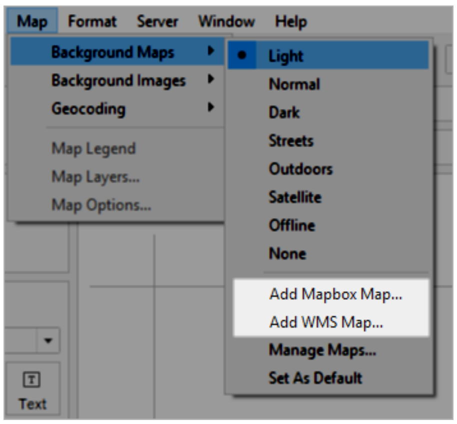
For more information, see **Use Dashboard Extensions** on page 2839.

Enhancements to Nested Sorting

Perform a Nested Sort on discrete measures and dimensions with the same ease you've come to expect from Tableau Desktop's Nested Sort feature.

Improved work flow when using WMS or Mapbox background maps

Add Mapbox or Web Map Service background maps to your visualization with the new, streamlined maps dialog. Each background map is now accessible directly from the Background Maps menu.

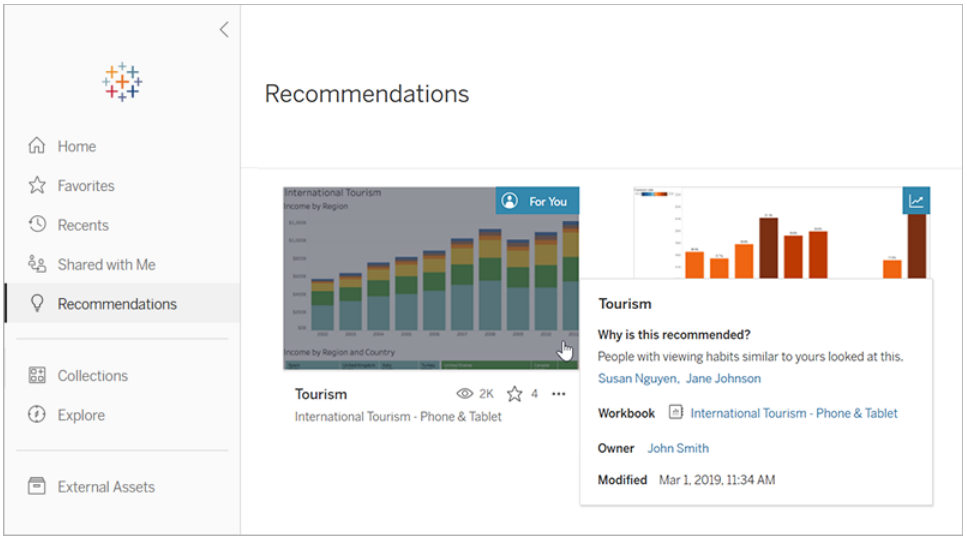


For more information, see **Use Web Map Service (WMS) Servers** on page 1996 or **Use Mapbox Maps** on page 1993.

Prepare, Author, and View Data on the Web

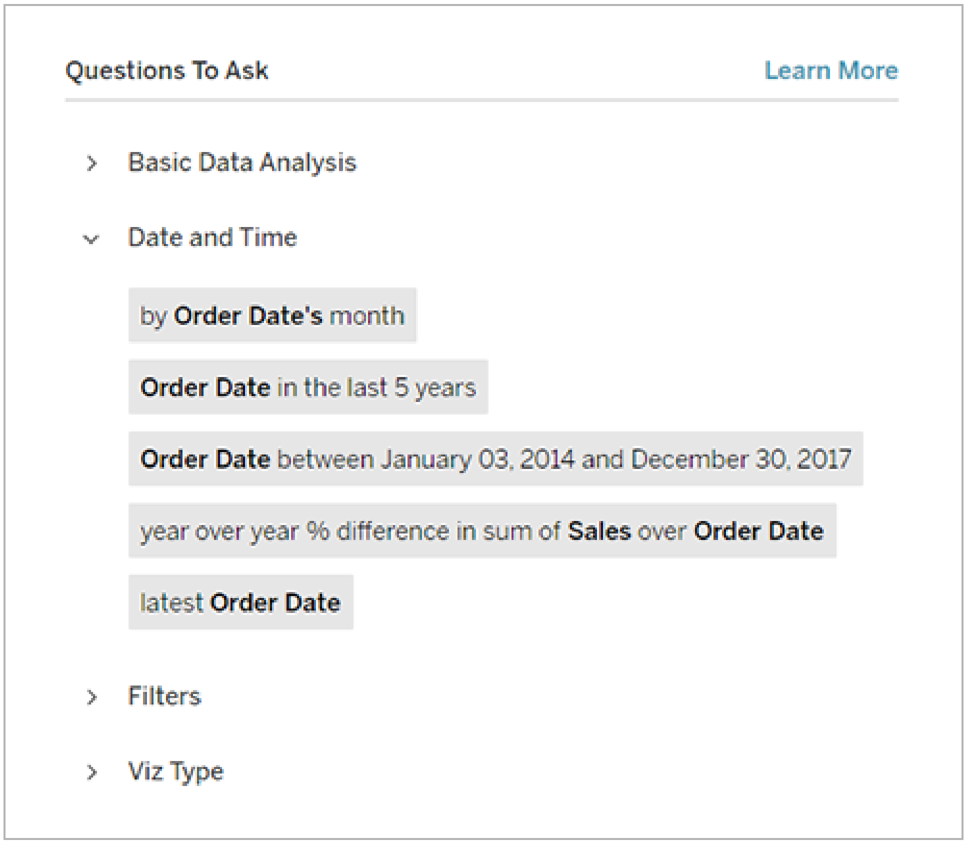
Discover recommended views

The new Recommendations page on Tableau Server and Tableau Online recommends views for you to explore. Recommendations help you find useful content on your site that you might not otherwise encounter. For more information, see **Discover Recommended Views on Your Tableau Site** on page 3231.



Choose from categories of suggestions for Ask Data

To help you quickly answer questions with Ask Data, we've grouped suggestions into four categories: Basic Data Analysis, Date and Time, Filters, and Viz Types. These suggestions make a great starting point for your analysis, which you can easily later revise as needed.



For more information, see **Automatically Build Views with Ask Data** on page 1109.

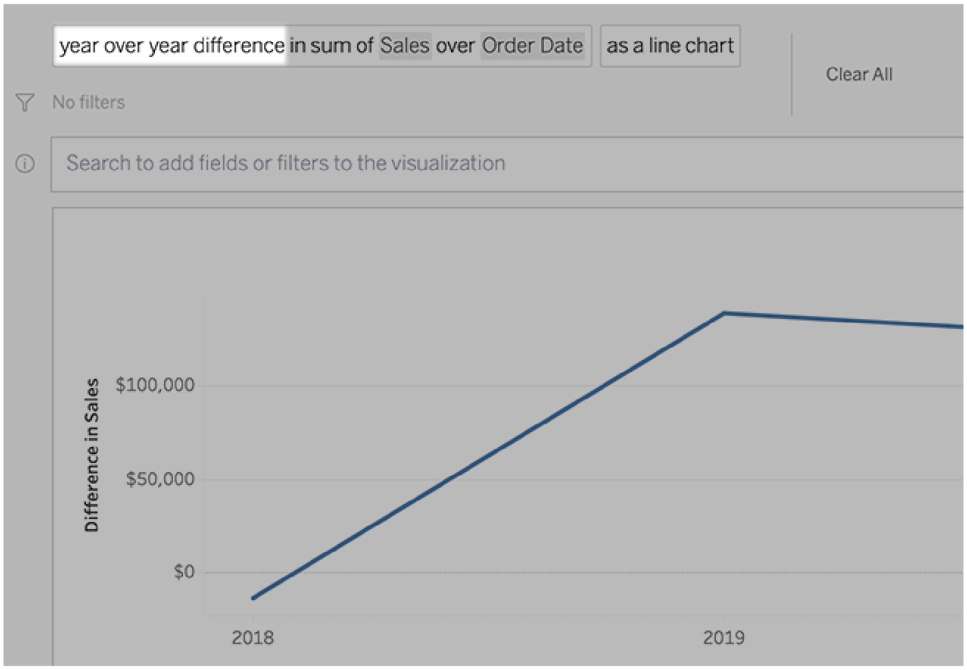
Share links to Ask Data vizzes

You can quickly share Ask Data vizzes without even saving a sheet. Simply click the link icon above the query box, and copy a link viewable by anyone with access to the data source.

Compare differences over time with Ask Data

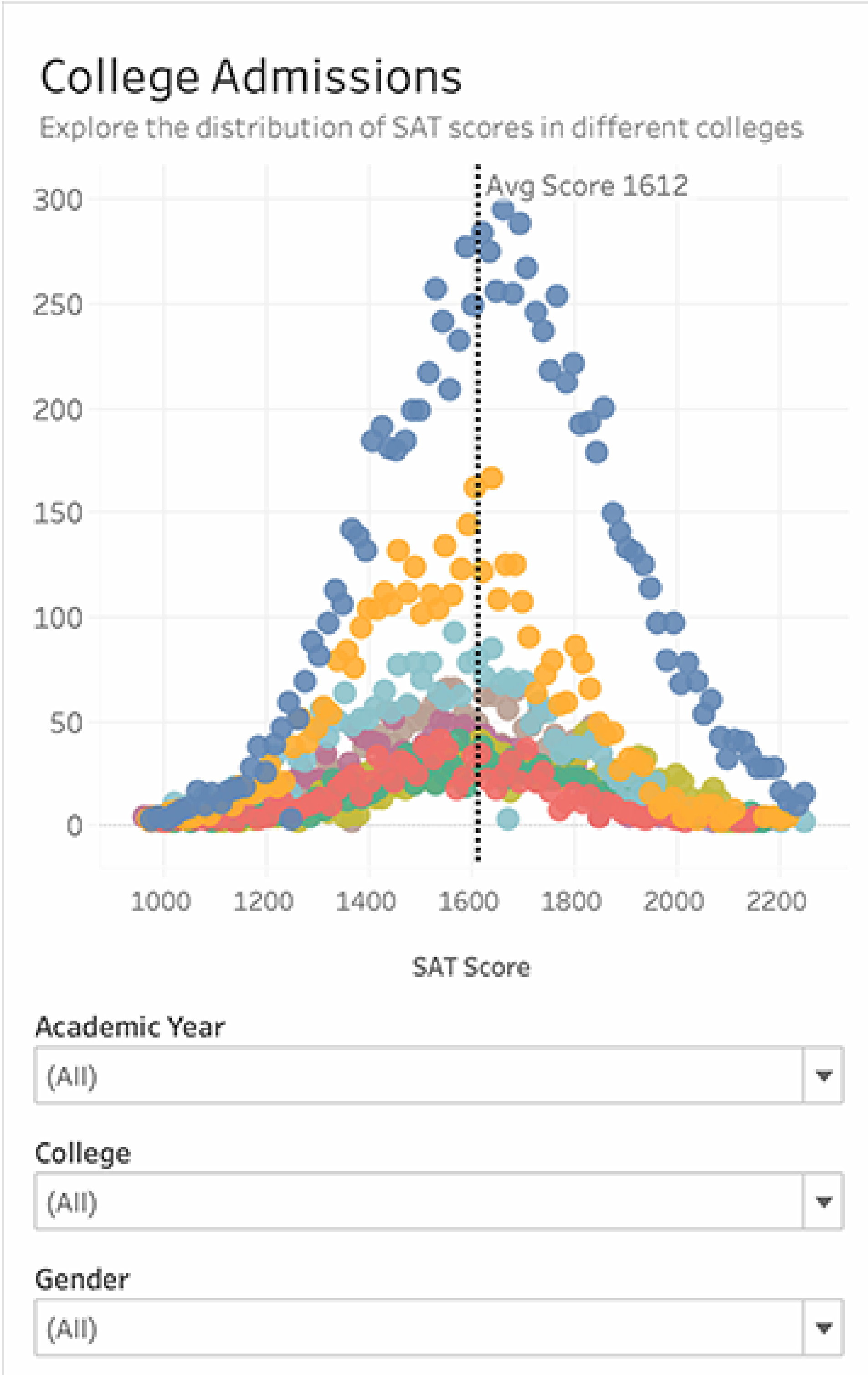
Ask Data now lets you compare time periods with phrases like “year over year”, “quarter over quarter”, or similar comparisons by month, week, or day. Differences can be expressed in either absolute figures or as a percentage.

For more information, see **Compare differences over time** on page 1119.



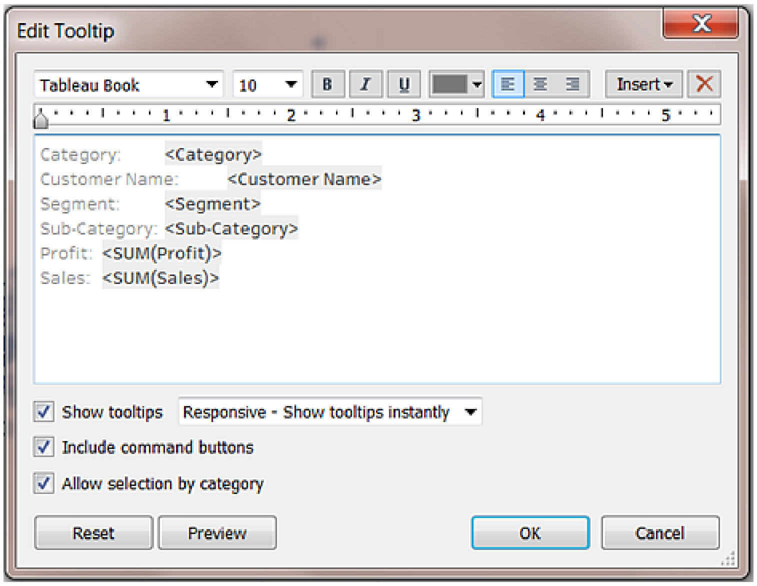
See optimized sheets on phones

When you view individual sheets on a mobile phone, Tableau automatically displays a vertical arrangement of filters, legends, and other cards.


College Admissions
Explore the distribution of SAT scores in different colleges
Avg Score 1612
300
250
200
150
100
50
0
1000
1200
1400
1600
1800
2000
2200
SAT Score
Academic Year
(All)
College
(All)
Gender
(All)

Edit Tooltips on the web

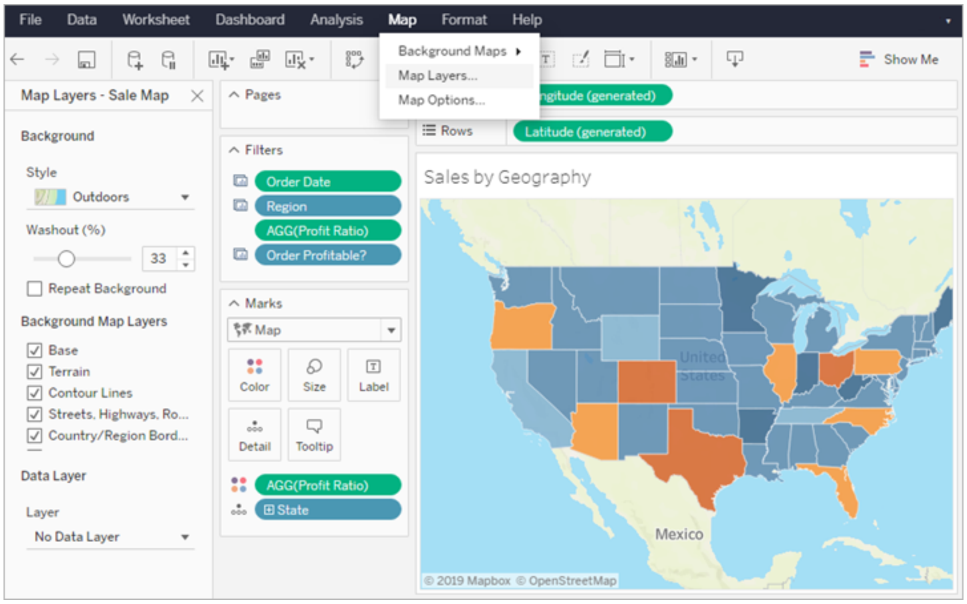
You can now format, edit, and use tooltips when editing your workbooks on the web using the same process you've learned in Tableau Desktop.



For more information, see **Format Titles, Captions, Tooltips, and Legends** on page 2937.

Customize maps on the web with Map Layers

Map layers allow you to customize the appearance of your map using Tableau's web editing features.

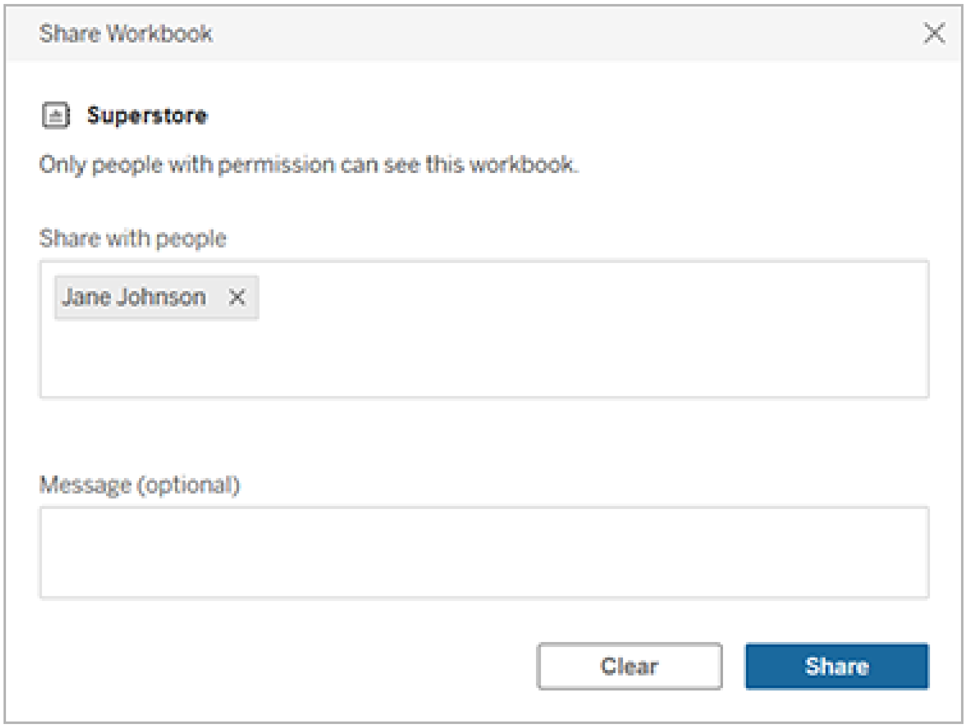


For more information, see **Customize How Your Map Looks** on page 1965.

Collaborate with Tableau Online and Tableau Server

Share content quickly via your Tableau site

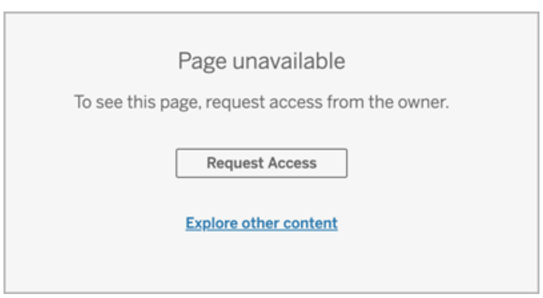
New sharing functionality lets you share content with others via your Tableau site—no need to compose a separate email. Additionally, if you make modifications to a view, by selecting filters or marks, those changes will now be captured when you share. For more information, see **Share Web Content** on page 3243.



Request access to content

You can now request access to content that you don't have permission to see a project, workbook, or view. When someone requests access, Tableau routes an email to the person who controls permissions for that content. For example, if a requested workbook is in a project where permissions are controlled at the project level (locked projects), the project owner receives an email with information about the request, a link to the requested content, and instructions about how to grant access or manage permissions.

For more information on access options, see **Manage Web Content** on page 3205



Accessibility improvements for Viewers in Tableau Online and Tableau Server

When you select a link to a view or workbook in Tableau Online or Tableau Server, you can use a screen reader or keyboard to navigate toolbar menus for collaboration, interact with an accessible view, or use the "Skip to Content" option to skip past the navigation panel and toolbar and jump straight to the view. This update also adds a focus indicator for keyboard navigation. Previously, views needed to be embedded for the experience to be accessible at this level. For more information, see **Keyboard Accessibility for Tableau on the Web** on page 3164 or see FAQ: Accessibility for Tableau's Accessibility Conformance Reports (VPATs).

Version 2019.3

Added in version 2019.3

Customize Language

Help Redesign

Connect to and Prepare Data

Design Views and Analyze Data

Prepare, Author, and View Data on the Web

Save and Publish Data Sources and Workbooks

Collaborate with Tableau Online and Tableau Server

- **Share more content types** on page 97

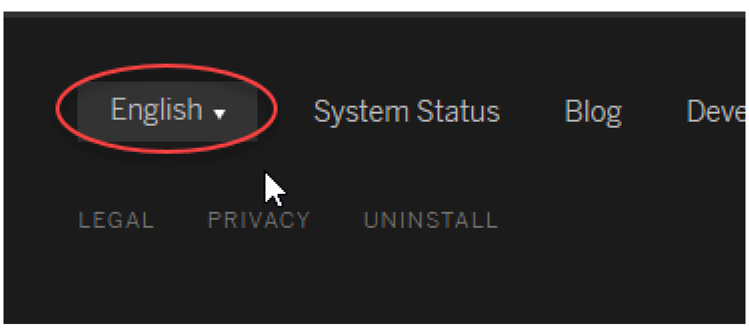
Support added for Italian

Tableau has added support for Italian. You can now set Tableau to display the user interface in one of eleven languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United Kingdom)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Spanish

For more information, see Set your display language for Tableau Desktop or Change language and locale for Tableau Online and Tableau Server.

In addition to supporting Italian in all Tableau products, the help is also available in these languages. To read the help in your preferred language, navigate to the bottom of the main Tableau Help page, and select the language from the footer.



Help Design

Provide feedback more easily

A blue **Send feedback** icon is displayed in the bottom right corner of every Help page to make it easier for you to send feedback about the content in a topic. Click the icon to let us know if a topic is helpful to you, and also to comment on what needs improvement.

Connect to and Prepare Data

Use database connections to perform cross-database joins

In past releases when performing cross-database joins between a file and a database, Tableau would pull all the data from the data sources into Hyper and perform the join there. Depending on the size of your data sets, this operation could take some time.

Now, to improve performance you can choose to allow Tableau to perform the join using the database that you are connected to instead. If you choose this option, Tableau decides which option is faster (Hyper or the connected database). If Tableau uses the connected database to perform the join, data from the file data source is copied into temporary tables in the connected database. Before selecting this option, make sure your database is from a trusted source.

To see this new option, in the **Data Source** tab, add one connection from a supported file type and one connection from a supported data source. The following file types and data sources and are supported for this release:

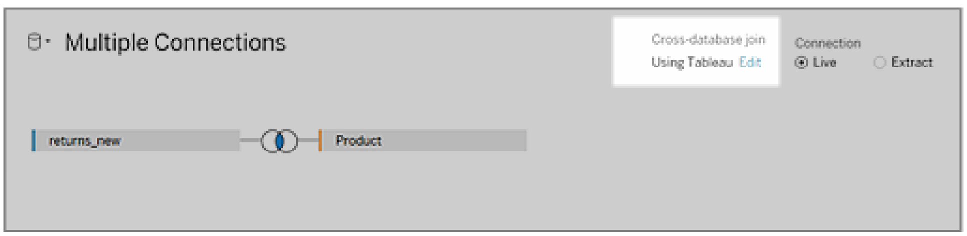
File types:

- Microsoft Excel
- PDF
- Text files (.csv, .txt, .tsv or .tab)

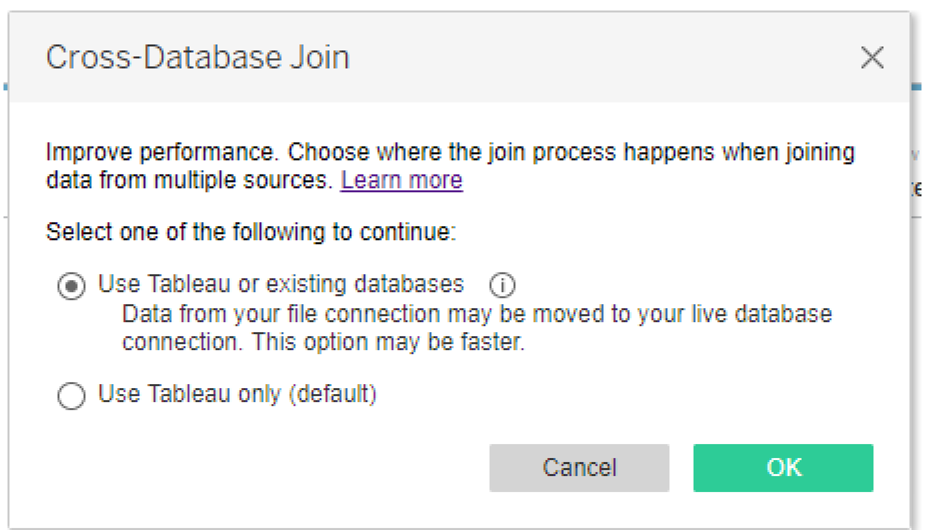
Data sources:

- Microsoft SQL Server
- Oracle
- PostgreSQL
- Vertica

The **Cross-database join** option displays in the top right corner of the canvas. To change how cross-database joins are performed, click **Edit**.



In the **Cross-Database Join** dialog, select the option you want to use.



The selected option displays in the canvas. You can change your selection at any time. This option is available in Tableau Desktop and for web editing in Tableau Server.

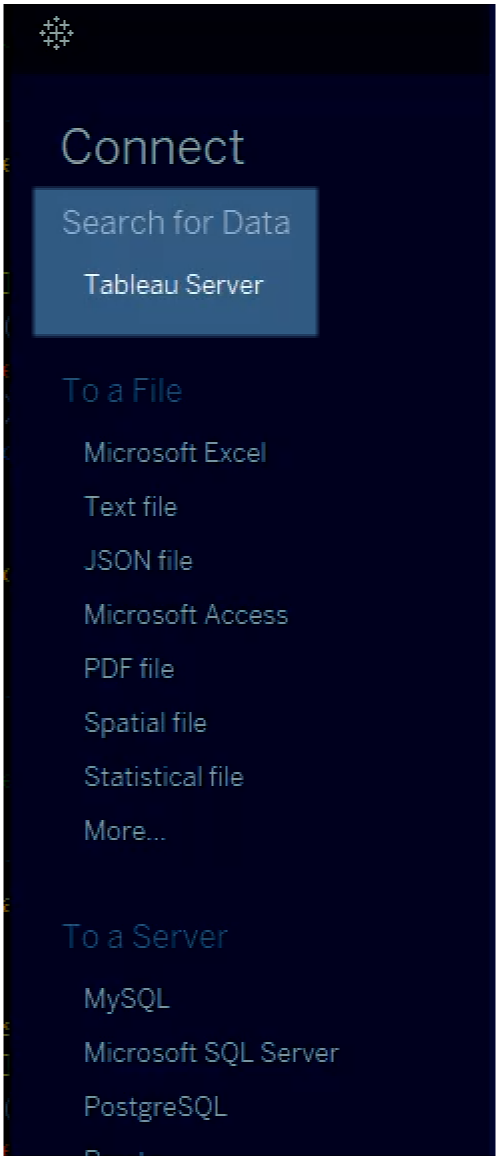
For more information about configuring cross-database join processing options, see **Join Your Data** on page 883.

Connect to databases and tables with Tableau Catalog (Data Management Add-on required)

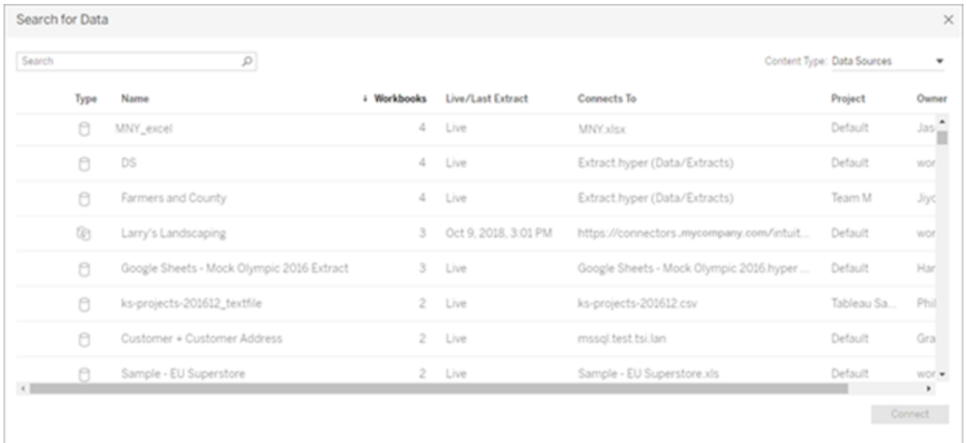
Tableau Catalog is available as part of the Data Management Add-on for Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to connecting to published data sources on Tableau Server and Tableau Online, you can connect to databases and tables from **Search for Data** on Tableau Desktop. For more information, see **Connect to Data by Using Tableau Server or Tableau Online** on page 445. For information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Tableau Server moves up on the Connect pane with a new Search for Data page

In the **Connect** pane on Tableau Desktop, **Tableau Server** moves to the top of the pane, under **Search for Data**. Select **Tableau Server** to connect to Tableau Online or Tableau Server.



When you connect to Tableau Server 2019.3 or later, on the **Search for Data** page you can search for and select a published data source to connect to. When you have Tableau Catalog enabled in your environment (Data Management Add-on required), you can also search for and connect to databases and tables.



For more information, see **Connect to Data by Using Tableau Server or Tableau Online** on page 445. For information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Connect to spatial files on the web and other spatial connector improvements

Connect to spatial files when authoring on the web in Tableau Server or Tableau Online. As a Creator, you can upload spatial file formats that only require one file (.kml, .geojson, .topojson, and Esri shapefiles packaged in a .zip and Esri File Geodatabases with the extension .gdb.zip) in the Files tab when you create a new workbook and connect to data. In Tableau Desktop, you can now connect to shapefiles in a .zip folder and Esri File Geodatabase folders. For more information, see **Creators: Connect to data on the web** on page 3135 on the Web and **Create Tableau Maps from Spatial Files** on page 1855.

Connect to Databricks

Use the Databricks connector to connect Tableau to your Databricks database. For more information, see **Databricks** on page 488.

Improved connections to TabPy

Customers can enable basic authentication for TabPy, requiring usernames and passwords for connections. Logging improvements allow for more granular tracking of which users executed what code. For more information, see **Pass Expressions with Analytics Extensions** on page 2690.

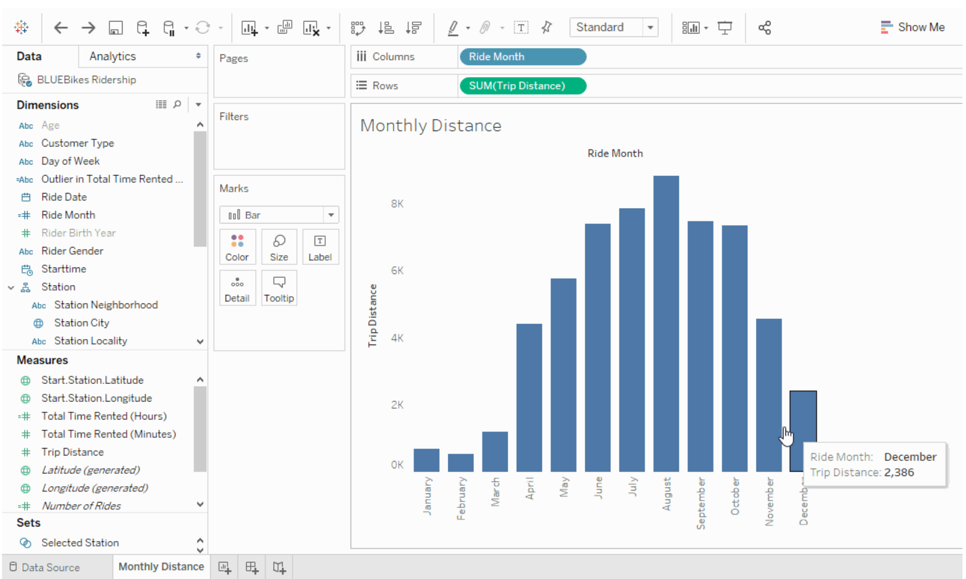
Design Views and Analyze Data

Explore and inspect views with Explain Data

Inspect, uncover, and dig deeper into marks in a viz using Explain Data. Explain Data gives you a new window into your data by proposing statistical explanations for a selected mark, including potentially related data from the data source that isn't used in the current view.

Creators and Explorers with editing permissions can use Explain Data when editing a view in Desktop, or editing a view on the web in Tableau Online or Tableau Server.

Explanations give you information and visualizations on the data points that might be affecting the mark value. You can then open visualizations for further exploration. For more information, see **Discover Insights Faster with Explain Data** on page 2035.

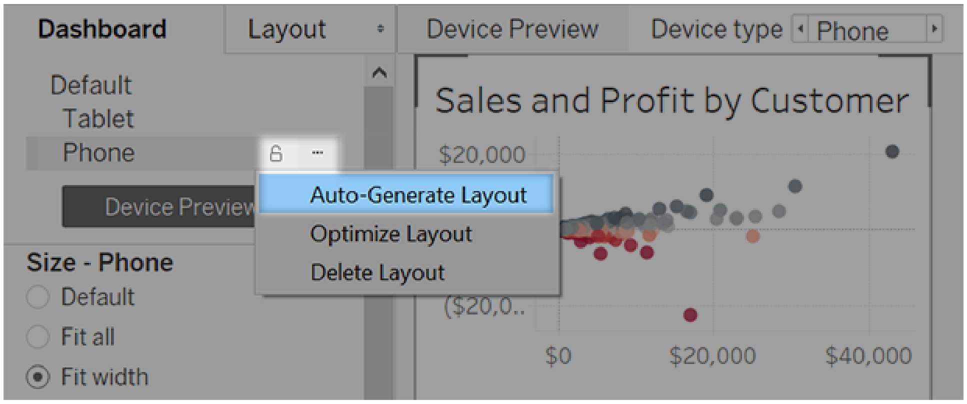


Create phone layouts with a streamlined workflow

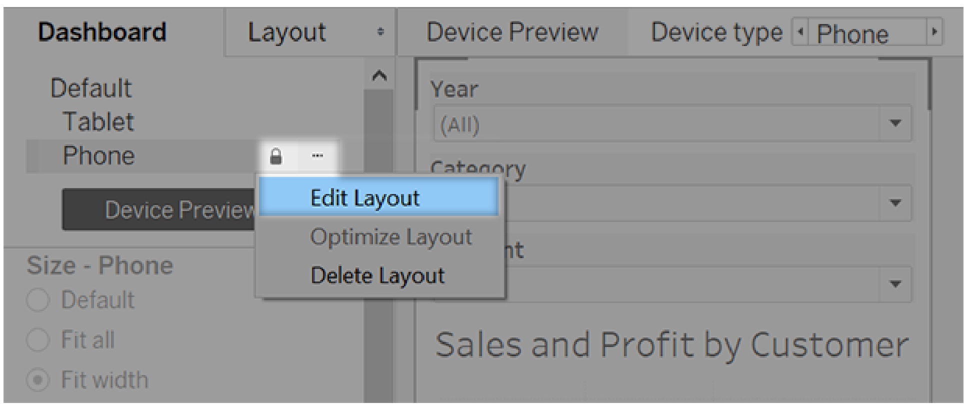
To always create phone layouts for a workbook, select **Dashboard** > **Add Phone Layouts to New Dashboards**. (This command was previously labeled "Auto-Generate Phone Layouts".)

Click the new lock icon to quickly toggle between automatic and manual Phone layouts, which you can also select via the pop-up menu:

- **Auto-Generate Layout** automatically synchronizes any changes to the Default dashboard, saving you time.



- **Edit Layout** creates a layout that's fully independent from the Default dashboard, letting you manually add and arrange items.

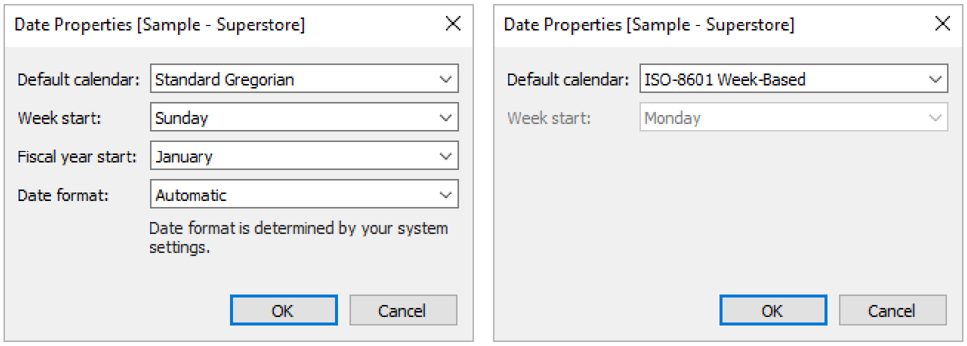


If you've chosen to edit a phone layout, select **Optimize Layout** to quickly optimize the placement of filters, remove white space, and more. (This command was previously labeled "Rearrange Layout".)

For more information, see **Create Dashboard Layouts for Different Device Types** on page 2801.

Quickly switch between the ISO-8601 calendar and the standard Gregorian calendar

With Tableau 2019.3, you can now change between using the week-based ISO-8601 calendar and the standard Gregorian calendar from the **Date Properties** dialog.

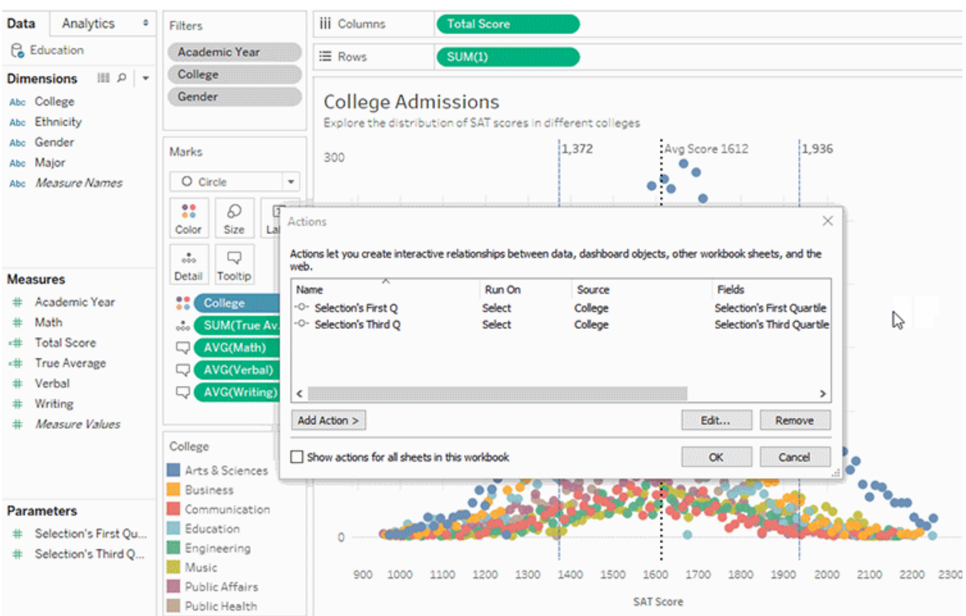


The ISO-8601 week-based calendar can be used to create 4-4-5 Quarters, a way of splitting each quarter into segments of 4, 4, and 5 weeks, a popular system in financial and retail reporting.

For more information, see **ISO-8601 Week-Based Calendar** on page 1273.

More aggregations available for parameter actions

Parameter actions now support concatenation, first quartile, and third quartile aggregations. Also, you can now create parameters while creating and editing a parameter action.

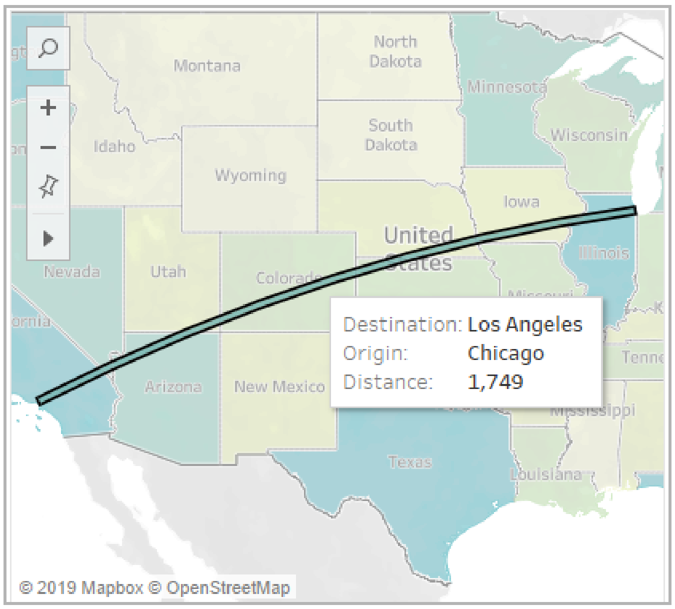


Use MAKEPOINT to convert projected coordinates into latitude/longitude

Convert projected coordinates into latitude and longitude values for use in spatial analysis and spatial joins. When created on live data sources, this function allows you to join your data with other spatial files using a calculated field, making it easy to create a data source for spatial analysis. For more information, see **Spatial Functions** on page 2245 and **Join Spatial Files in Tableau** on page 1815.

Use DISTANCE to measure between points on a map

Calculate the distance between two related points on your map using the spatial function Distance. You can specify kilometers, miles, meters, or feet as the unit of measurement. This function can only be created with a live data source. For more information, see **Spatial Functions** on page 2245.



Prepare, Author, and View Data on the Web

Connect to databases and tables with Tableau Catalog when web authoring (Data Management Add-on required)

Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When Tableau Catalog is enabled in your environment, in addition to navigating and connecting to published data sources from the **On this site** tab, you can connect to more kinds of data, like databases and tables. For more information, see **Creators: Connect to data on the web** on page 3135. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Embed Ask Data into web pages

After you ask a question with Ask Data, an **Embed Code** icon appears in the toolbar. Click it to copy code you can incorporate into any web page.

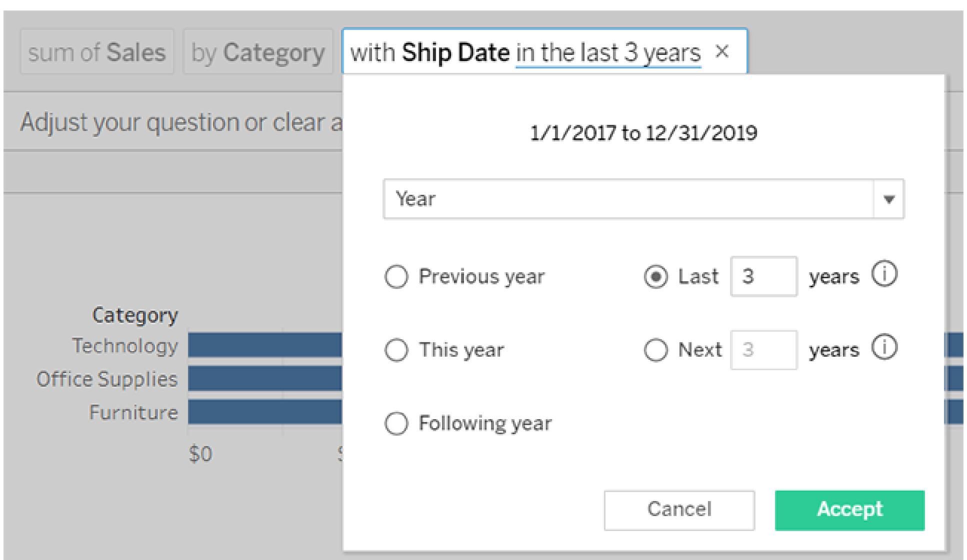
When web pages with Ask Data embedded in them load, Ask Data opens the data source without a viz, waiting for users to ask a question.

To see Ask Data embedded in a web page, users must have a Tableau license and permission to connect to the data source.

For more information, see **Embed an Ask Data lens in a web page** on page 1122.

Adjust relative date filters with Ask Data

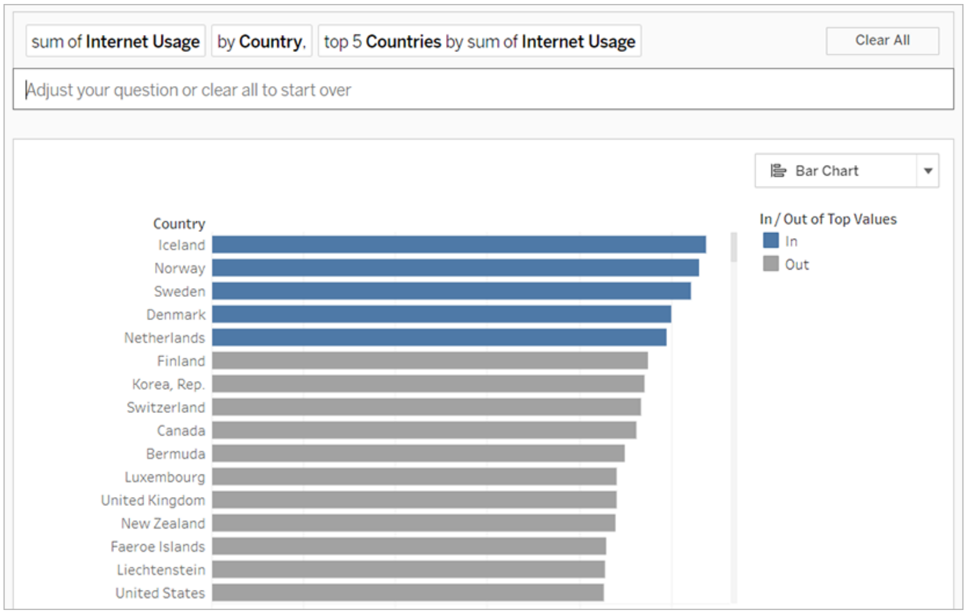
To adjust a relative date filter with Ask Data, click words like “last”, “previous”, “next”, or “following”. For more information, see **Change fields, filters, and displayed data** on page 1115.



Show top and bottom outliers in context with Ask Data

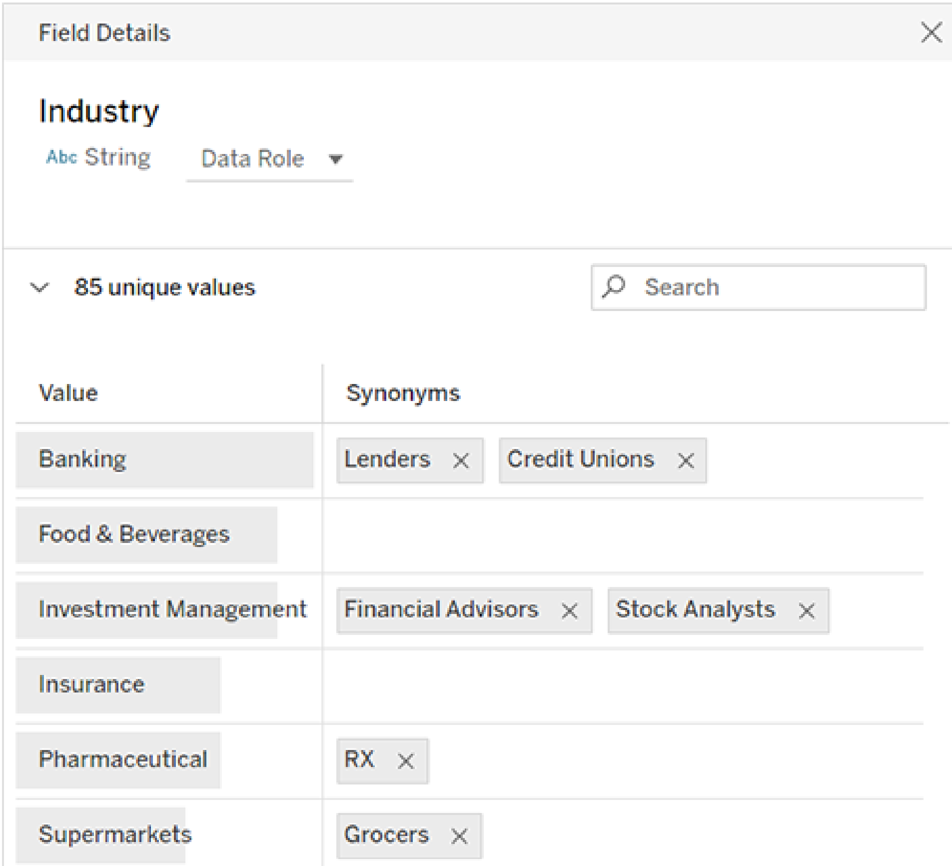
When you specify numerical limits like “top 5 countries”, Ask Data now shows those outliers in the context of the complete list for a dimension.

For more information, see **Numerical limits (top and bottom values)** on page 1125.



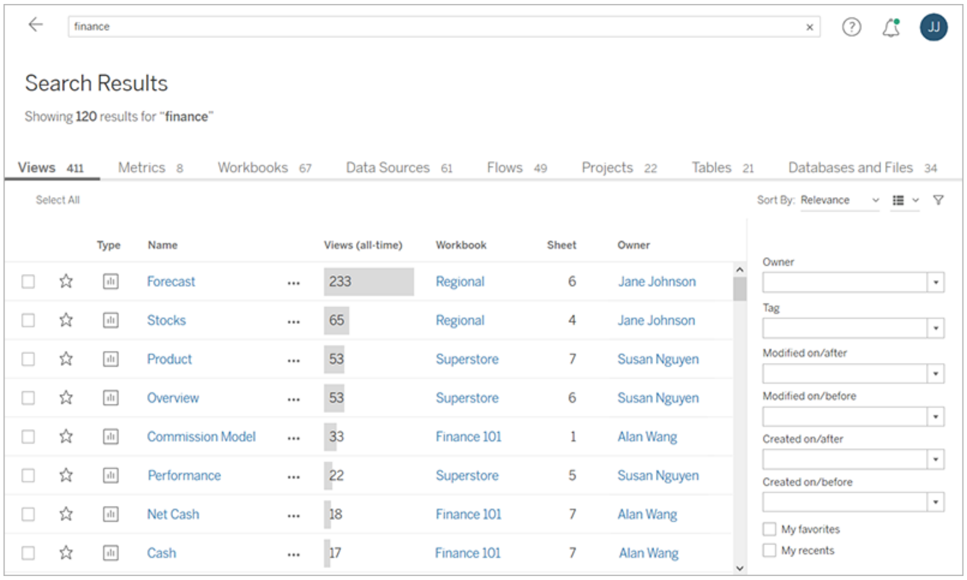
Add synonyms for field values with Ask Data

If you're a data source owner or Tableau administrator, you previously were limited to adding synonyms for field names, but now you can add them for field values as well. For more information, see **Add Synonyms for Ask Data** on page 1140.



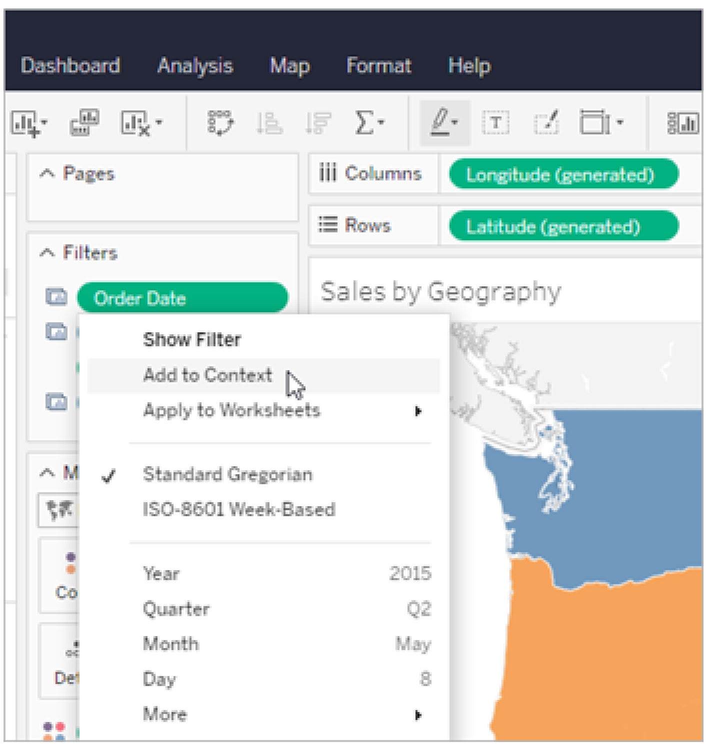
Search your Tableau site with improved results

When you search on Tableau Online and Tableau Server, you can now view a full page of search results. These results are sectioned by type, and you can filter and sort the results for increased precision. For more information, see **Search Your Tableau Site** on page 3189.



Set context filters

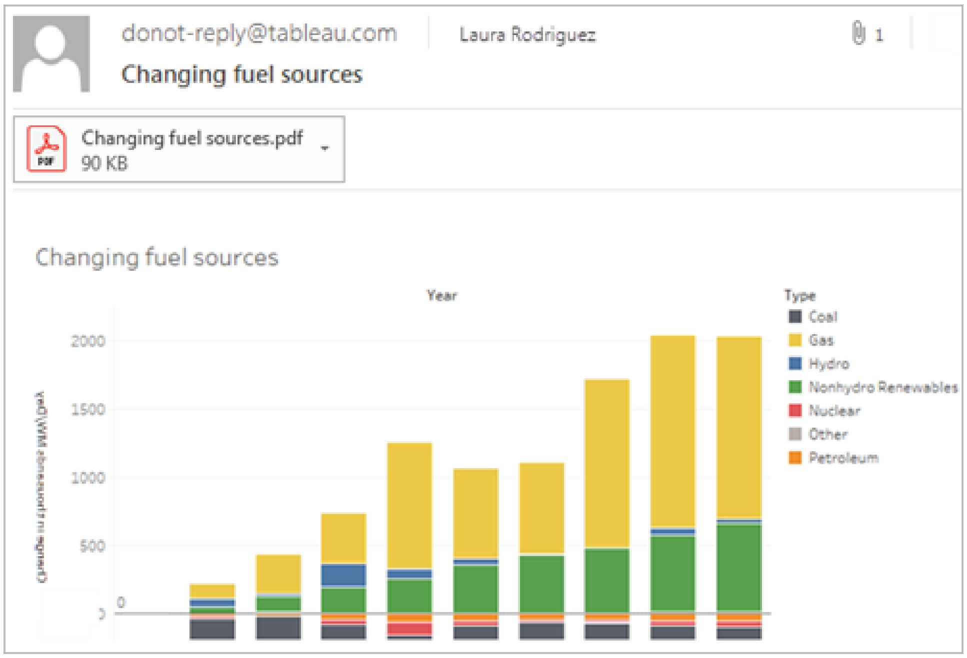
You can now create context filters (**Add to Context** option in the Filters shelf) and dependent filters (**Only Relevant Values** option in a filter control shown in a view).



Save and Publish Data Sources and Workbooks

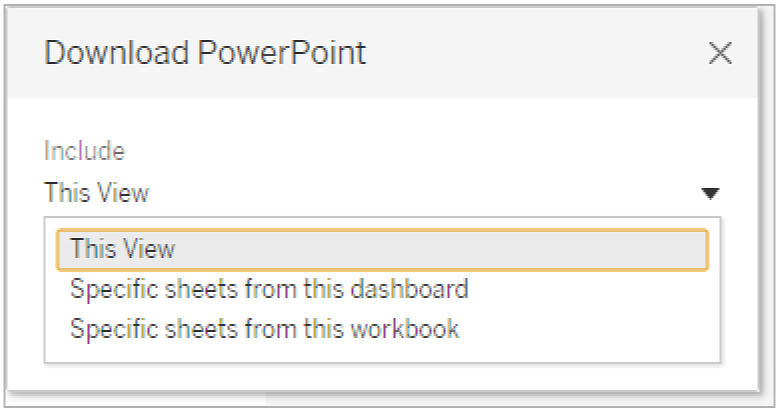
Receive subscriptions as PDF attachments

Subscriptions email you a snapshot of your view or workbook at an interval you set, and now you can receive these snapshots as PDF attachments. To set up a subscription as a PDF, select **Format** > **PDF** or **Image and PDF** when creating a subscription in Tableau Online or Tableau Server. For more information see **Create a Subscription to a View or Workbook** on page 3249



Create a PowerPoint file from selected workbook sheets

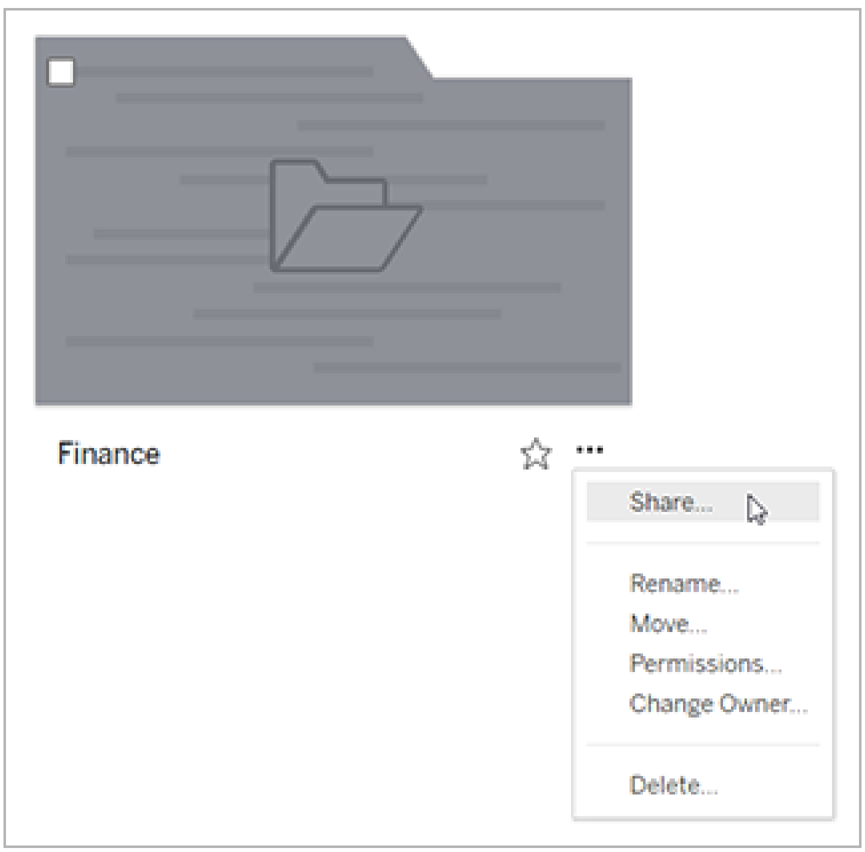
When you want to create a PowerPoint file from your Tableau workbook or dashboard, you can select which sheets and stories to include. Each selected sheet (or story point if exporting a story) becomes its own slide in the generated PowerPoint file. For more information, see **Export Views from Tableau Desktop to Another Application** on page 3032.



Collaborate with Tableau Online and Tableau Server

Share more content types

Get an easy-to-share link for projects, workbooks, data sources, and flows, in addition to views, which you could previously share. For more information, see **Share Web Content** on page 3243.



Version 2019.2

Added in version 2019.2

Customize Language

- **Select from ten different languages for Tableau** on the facing page

Help Redesign

- **Provide feedback more easily** on page 101

Connect to and Prepare Data

- **Connect to Anaplan data using certificates** on page 16
- **Use features added to the Marketo connector** on page 101
- **Connect to TabPy external services using SSL** on page 102

Design Views and Analyze Data

- **Use parameter actions to set parameter values interactively** on page 102
- **Pan, zoom, and explore with vector tile background maps** on page 103
- **Customize how your map looks with new map styles and layers** on page 103
- **Replace sheets in dashboards** on page 105
- **Use Tableau connectors with Azure Active Directory authentication** on page 16
- **Show and hide floating containers in dashboards** on page 106
- **Customize reference line tooltips** on page 106
- **Disable the sort icon** on page 106

Prepare, Author, and View Data on the Web

Save and Publish Data Sources and Workbooks

Collaborate with Tableau Online and Tableau Server

Customize Language

Select from ten different languages for Tableau

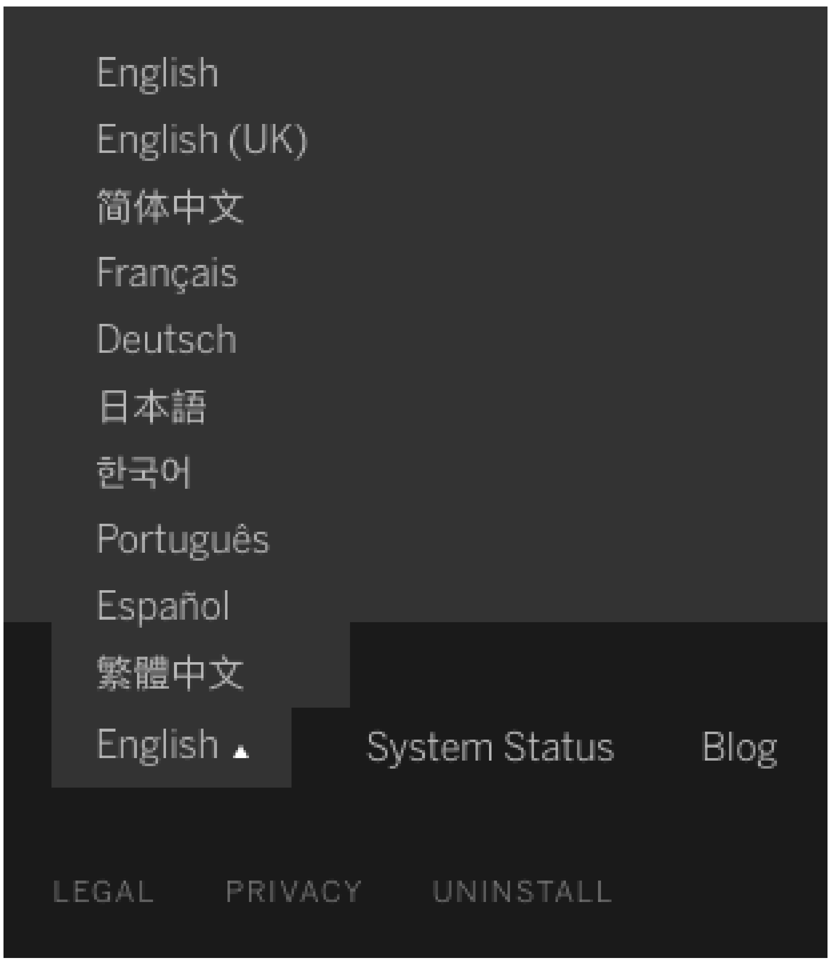
Tableau has added support for two new languages, Chinese (Traditional) and English (United Kingdom). You can now set Tableau to display the user interface in one of ten languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United Kingdom)
- English (United States)

- French
- German
- Japanese
- Korean
- Portuguese
- Spanish

For more information, see Set your display language for Tableau Desktop or Change language and locale for Tableau Online and Tableau Server.

In addition to supporting two new languages in all Tableau products, the help is also available in these languages. To read the help in your preferred language, navigate to the bottom of the main Tableau Help page, and select the language from the footer.



Help Design

Provide feedback more easily

A blue **Send feedback** icon is displayed in the bottom right corner of every Help page to make it easier for you to send feedback about the content in a topic. Click the icon to let us know if a topic is helpful to you, and also to comment on what needs improvement.

Connect to and Prepare Data

Connect to PostgreSQL + PostGIS and Pivotal Greenplum + PostGIS spatial data

Use Tableau to connect to spatial data in Pivotal Greenplum or PostgreSQL with the PostGIS extension. With these new spatial connectors, you can use CustomSQL to filter and validate spatial data, or use RAWSQL_SPATIAL functions to return spatial objects for advanced spatial analysis. For more information, see **Connect to Spatial Data in a Database** on page 1799.

Use features added to the Marketo connector

The Marketo connector includes these new features:

- Use of the bulk API to retrieve data for the All Activities, Leads, and Leads by List - <list name> tables, thereby reducing the number of API calls.
- Support for retrieving the number of API calls made using the Daily API Usage and Weekly API Usage tables so that you can monitor your API usage and not exceed your quota.

- Access to all Marketo tables, enabling you to access data that was previously not available.

For information about the Marketo connector, see **Marketo** on page 555.

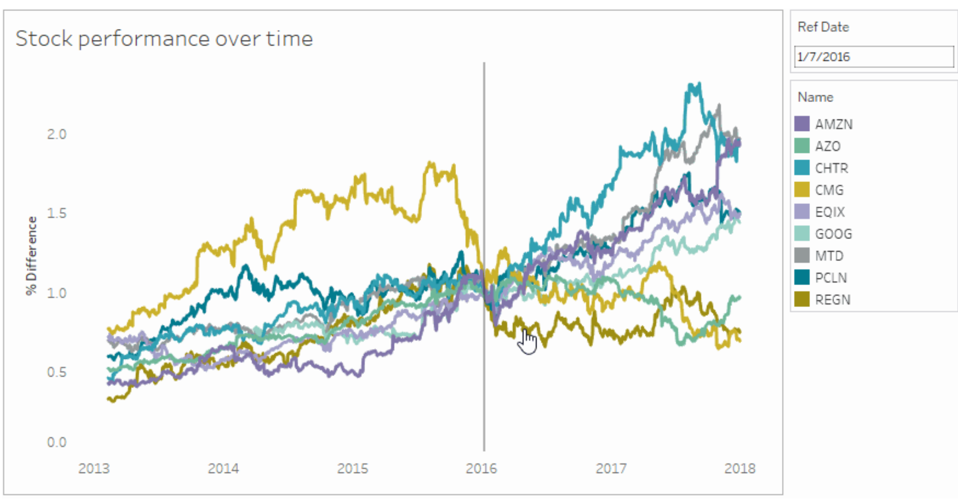
Connect to TabPy external services using SSL

In previous versions, SSL connections to TabPy were not supported. In Tableau Server 2019.2, you can configure SSL connections. Build visualizations using the Python language for advanced analytics and machine learning in a secure environment. TabPy can be hosted remotely from Tableau Server, and you can configure SSL connections to protect data in transit. For more information, see **Pass Expressions with Analytics Extensions** on page 2690.

Design Views and Analyze Data

Use parameter actions to set parameter values interactively

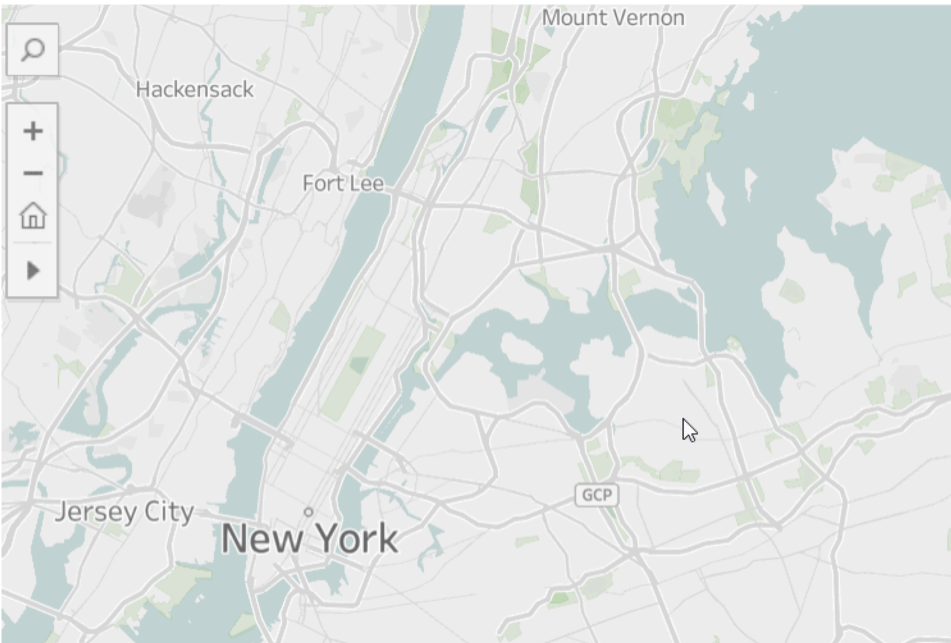
Use parameter actions to let your audience change a parameter value through direct interaction with a viz, such as clicking or selecting a mark. You can use parameter actions to drive reference lines, calculations, filters, and SQL queries. For more information, see **Parameter Actions** on page 1516.



Parameter actions open up new possibilities for creating summary values and statistics without using calculations. You can set parameter actions to let users select multiple marks that are automatically aggregated into a single parameter value. For example, a parameter action could show the AVG(Sales) or COUNTD(Orders) for currently selected marks.

Pan, zoom, and explore with vector tile background maps

Maps in Tableau now use vector tiles for quicker geographic data exploring. Maps now appear sharper and smoother when you pan and zoom, and labels and icons dynamically resize or reshape to fit your view. The new vector tile maps replace the underlying technology that powers Tableau background maps for a smoother and crisper experience. The background cartography for Tableau maps has also been updated.

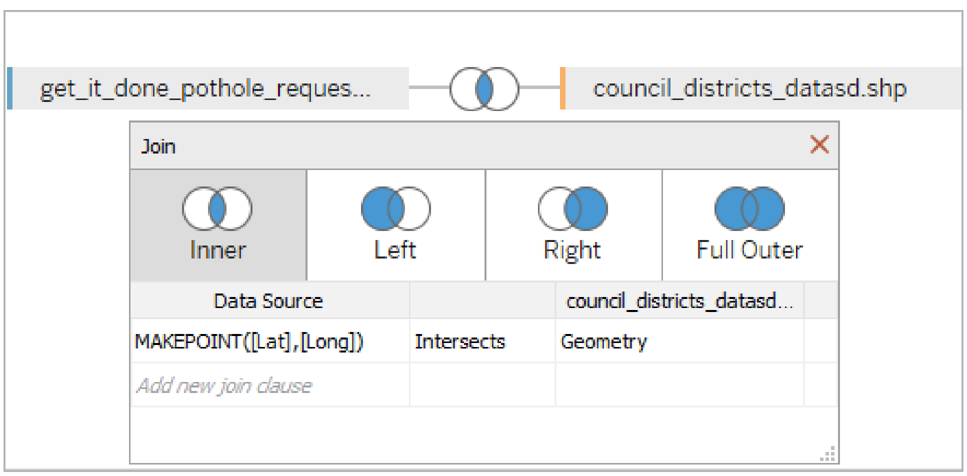


Customize how your map looks with new map styles and layers

You can customize how your Tableau map looks with three new built-in map styles: Streets, Satellite, and Outdoors. Each of these styles includes new map layer options, such as Points of Interest and Building Footprints, so that you can customize what information your analysis displays. For more information, see **Customize How Your Map Looks** on page 1965.

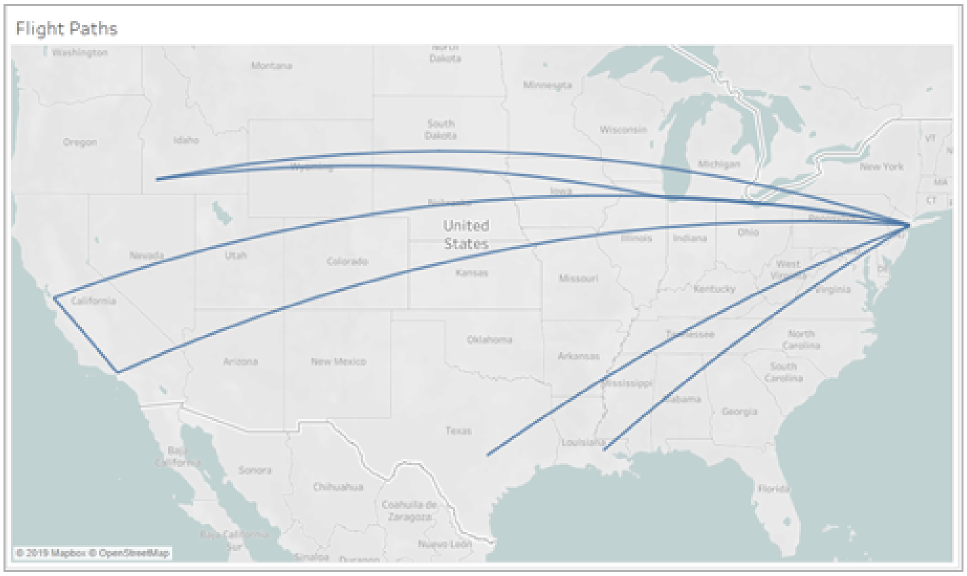
Create spatial fields and use spatial joins with MakePoint

Turn latitude/longitude values from text files, excel, or other data sources into spatial fields with the function MakePoint. This function allows you to spatially join your lat/lon data by using a calculated field; making it easy to aggregate your data. For more information, see **Spatial Functions** on page 2245 and **Join Spatial Files in Tableau** on page 1815.



Create origin-destination maps using MakeLine

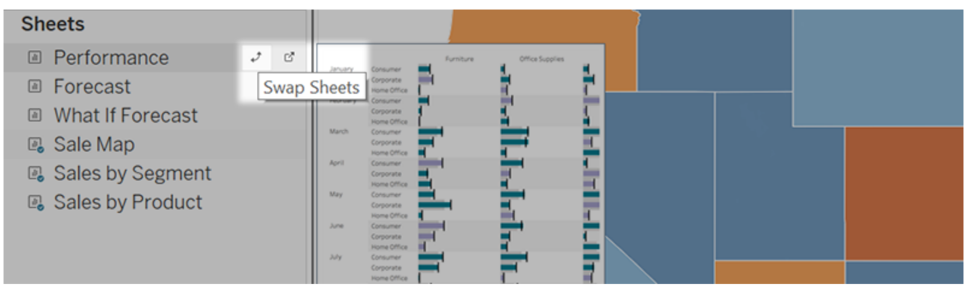
Turn related points on your map into origin-destination maps using the spatial function MakeLine. For more information, see **Spatial Functions** on page 2245.



Replace sheets in dashboards

You can quickly select a dashboard sheet and replace it with another, while retaining any padding, border, or background color, and your overall layout.

For more information, see **Create a dashboard, and add or replace sheets** on page 2741.



Use text for navigation buttons in dashboards

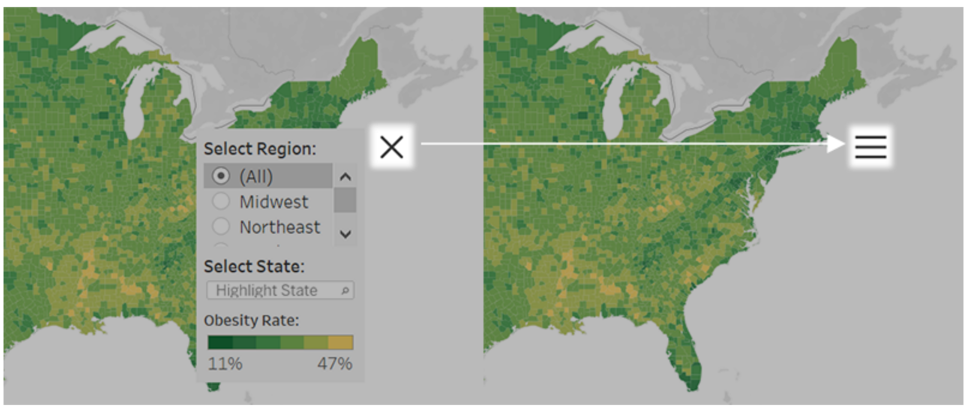
Button objects let your audience navigate from one dashboard to another, or to other sheets or stories. Previously, these buttons could only display images, but now you can instead display text to indicate a button's destination.

For more information, see **Detailed options for Navigation and Download objects** on page 2746.

Show and hide floating containers in dashboards

Show/Hide buttons let dashboard viewers toggle the visibility of floating layout containers, revealing them only when necessary. These buttons are particularly helpful when you want to temporarily hide groups of filters to reveal more of a visualization.

For more information, see **Size and Lay Out Your Dashboard** on page 2755.



Customize reference line tooltips

Add custom text and fields to reference line tooltips. For more information, see **Reference Lines, Bands, Distributions, and Boxes** on page 2502.

Disable the sort icon

You can now disable the sort icon in a view, which allows you to control sorting of the view before publishing. For more information, see **Sort Data in a Visualization** on page 1447.

Prepare, Author, and View Data on the Web

Ask questions in a conversational style with Ask Data

As new questions occur to you in Ask Data, you can quickly revise existing ones in a natural, conversational style. In the text box reading "**Adjust your question or clear all to start over**", do any of the following:

- Add functions with phrasing like "by country" or "in December".
- Replace fields by typing "replace [*existing field name*] with [*new field name*]"
- Remove fields by typing "remove [*field name*]"
- Clear the viz by typing "reset" or "clear".

For more information, see **Automatically Build Views with Ask Data** on page 1109.

Apply simple calculations with Ask Data

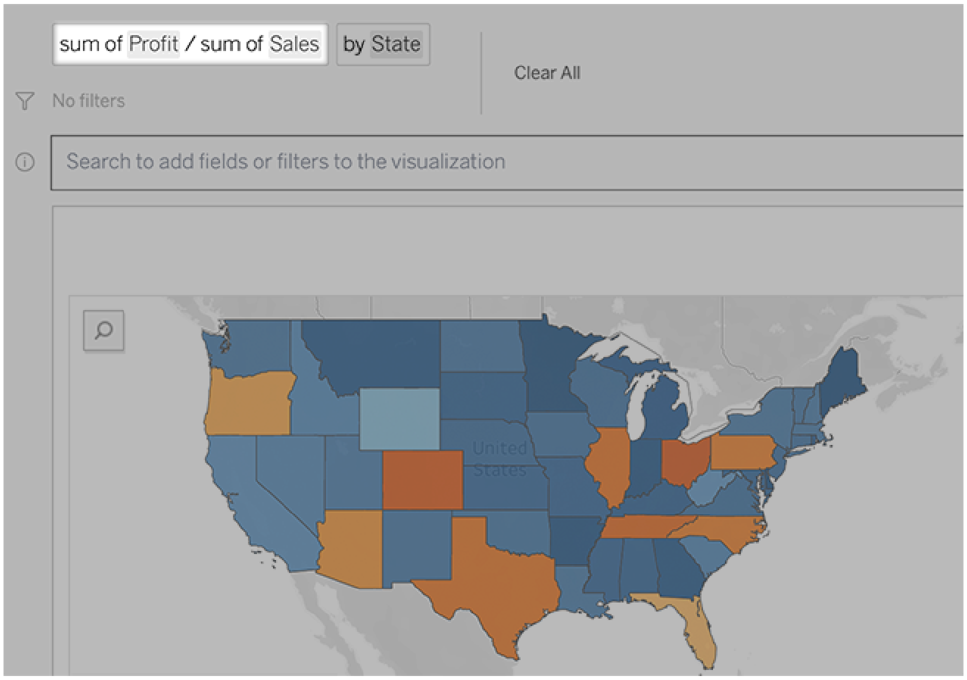
To enhance data analysis, Ask Data supports simple calculations between two measures, which you can apply using the symbols below:

+ sums the measures

- produces the difference between them

***** multiplies

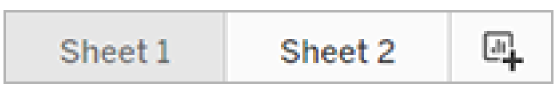
/ divides



For more information, see **Apply simple calculations** on page 1120.

Create multiple sheets from Ask Data

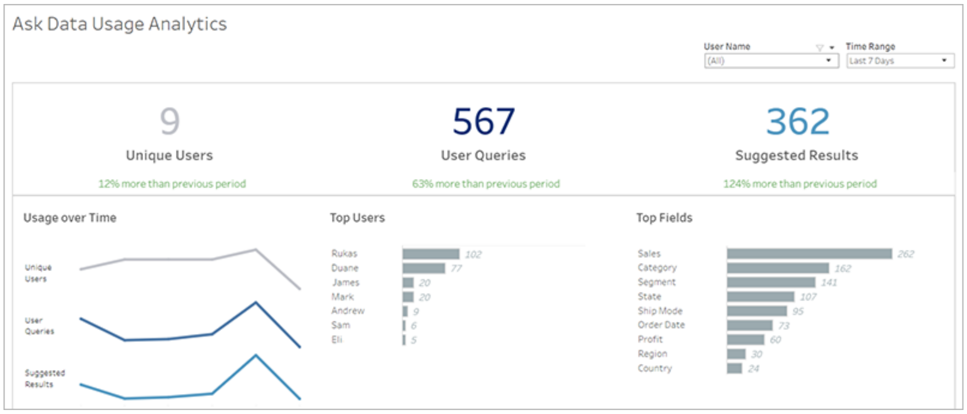
You can quickly create multiple different vizzes from a data source and then save them in a new workbook. Below any viz in Ask Data, simply click the **Add Sheet** icon to the right of named sheets.



For more information, see **Embed and Save Vizzes from Ask Data** on page 1122.

See how people are using Ask Data

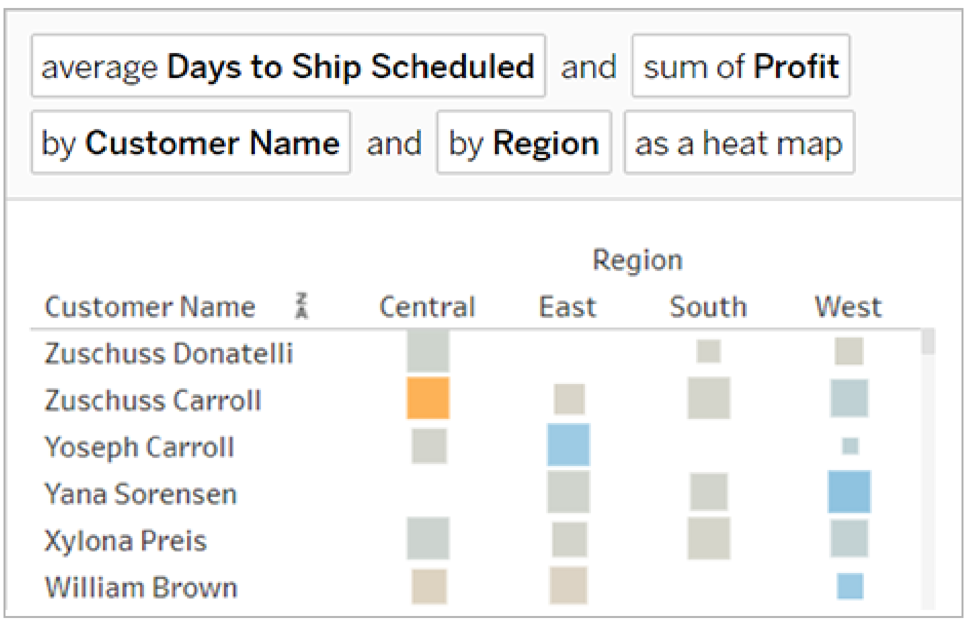
If you're a data source owner or Tableau administrator, Ask Data provides a dashboard that reveals the most popular queries and fields, the number of visualization results that users clicked, and other helpful information. Filters let you narrow data down to specific users and time ranges. These stats help you optimize data sources to increase the success of your users.



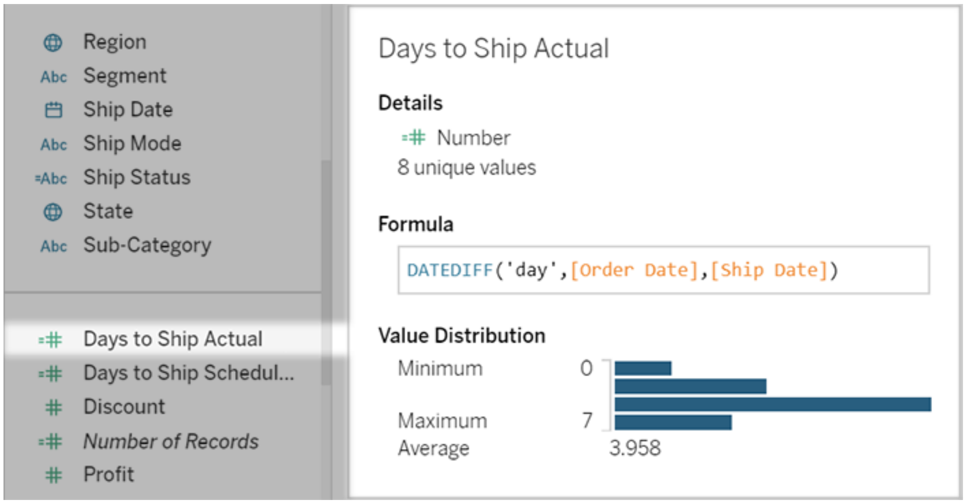
For more information, see **Optimize Data for Ask Data** on page 1132.

Additional enhancements to Ask Data

- Ask Data now supports Heat Map vizzes, which are great for revealing outliers from many dimensions and measures.



- When you hover over continuous measures at left, histograms appear in tooltips, quickly showing the range of available values.



- Terms like "Date", "Year", and "Month" are now reserved for use in analytical functions so should be avoided in field names. For the complete list of reserved terms, see Field names that Ask Data uses for analytical functions.
- Synonyms for field names can now be changed only by data source owners and Tableau administrators, providing a more consistent experience for users. For more information, see **Add Synonyms for Ask Data** on page 1140.

Create, edit, and remove parameters in web authoring

Now in web authoring, you can create parameters to replace constant values in calculations, filters, and reference lines. For more information, see **Create Parameters** on page 1218.

Save custom views with the Viewer role

Previously, creating custom views required a Creator or Explorer license, but now Viewers can create them too. Anyone with the URL for a custom view can see the filters and other changes that are applied. However, only public custom views created by Creators and Explorers are visible to others in the Custom Views dialog box.

For more information, see **Use Custom Views** on page 3224.

Use parameters with stored procedures in Tableau Online and Tableau Server

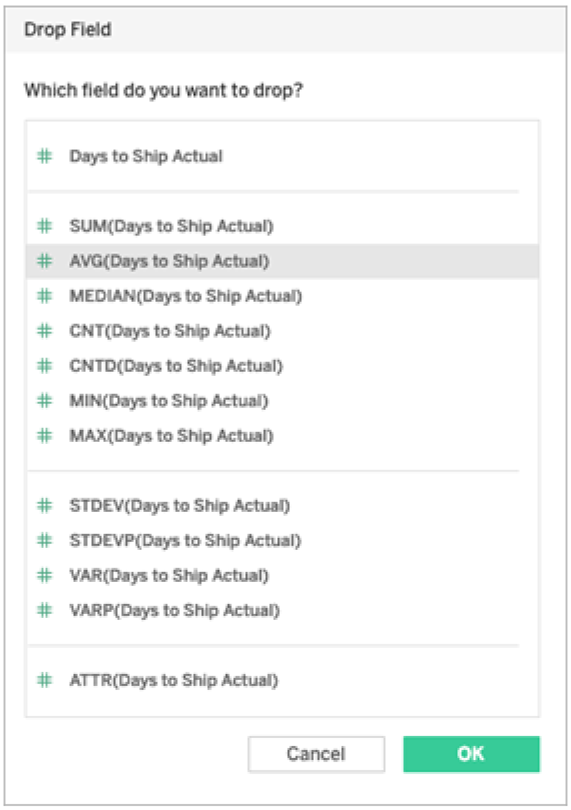
When creating a data source in Tableau Online or Tableau Server, you can now add a stored procedure with parameters to a join. When a stored procedure includes parameters, you'll see a dialog that displays parameters and their types as defined in the database, and edit those parameter values. For more information, see **Use a Stored Procedure** on page 972.

See your view's total marks and more in the status bar

The status bar in the lower left corner of your browser shows authors information about their view, including selected marks compares to total marks, rows and columns, and aggregations. For more information on the status bar, see Status Bar Information.

Drop field menu in web authoring

When creating or editing a view on the web, you can change the aggregation of a field using the drop field menu. Just right-click+drag (on macOS, left-click+Option+drag) a field on any card or shelf like rows, columns, or Filters, and select the aggregation type you need. This drops the field onto the card or shelf with the aggregation you selected.



Save and Publish Data Sources and Workbooks

Export stories into PowerPoint

Now when you use Export to PowerPoint to export your workbook, each story points becomes as separate slide in your Tableau-generated PowerPoint file. For more information on exporting your workbook into PowerPoint, see **Export Views from Tableau Desktop to Another Application** on page 3032.

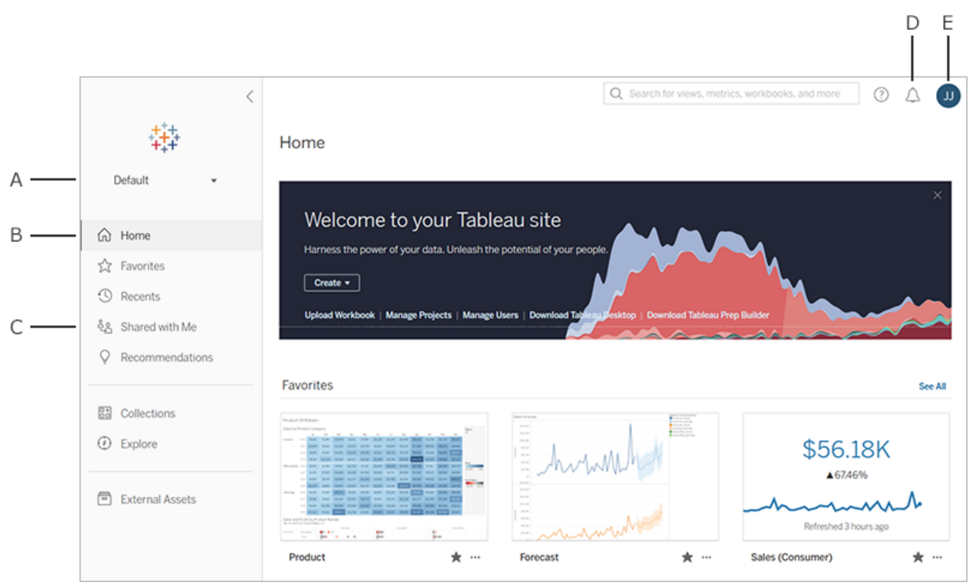
Collaborate with Tableau Online and Tableau Server

Discover the new face of Tableau Server and Tableau Online

Your Tableau site now makes it easier to get started and locate relevant content with several new pages and navigation changes.

Quickly access your most important content

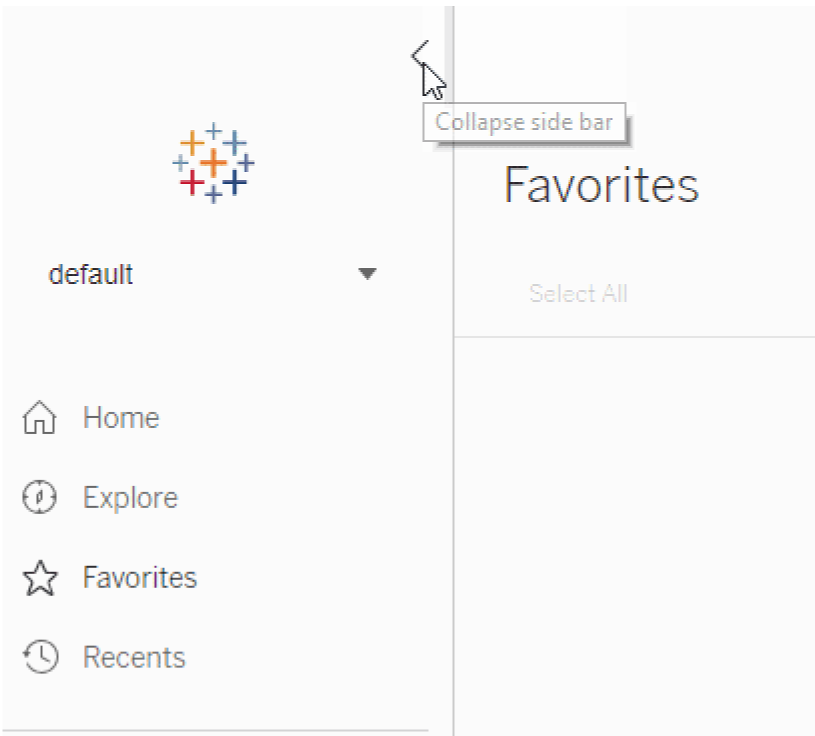
When you sign in to Tableau Online or Tableau Server, you'll be greeted by a new **Home** page. On Home, you can see visualizations you recently viewed, find your newest favorite content, and discover what's popular on your site, all in one place. The welcome banner and actions at the top of the page are tailored to your site role to help you get started.



Dedicated **Favorites** and **Recents** pages give you access to all of your most-important content. Use the sort and filter options on the right side of each page to find exactly what you are looking for. Add or remove favorite content at any time by clicking that content's star icon. Along with the Home page, you can access the Favorites and Recents pages from the left navigation panel.

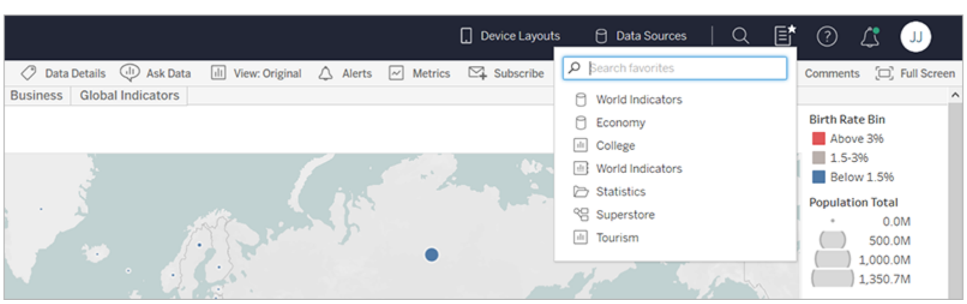
Navigate your site with ease

The new left navigation panel lets you quickly jump between important pages on your site, including the **Explore** page, where you can see all the content on your site in one place. The navigation panel is responsive to screen sizes and can be collapsed to give you optimal space for browsing.



View your data with minimal interruption

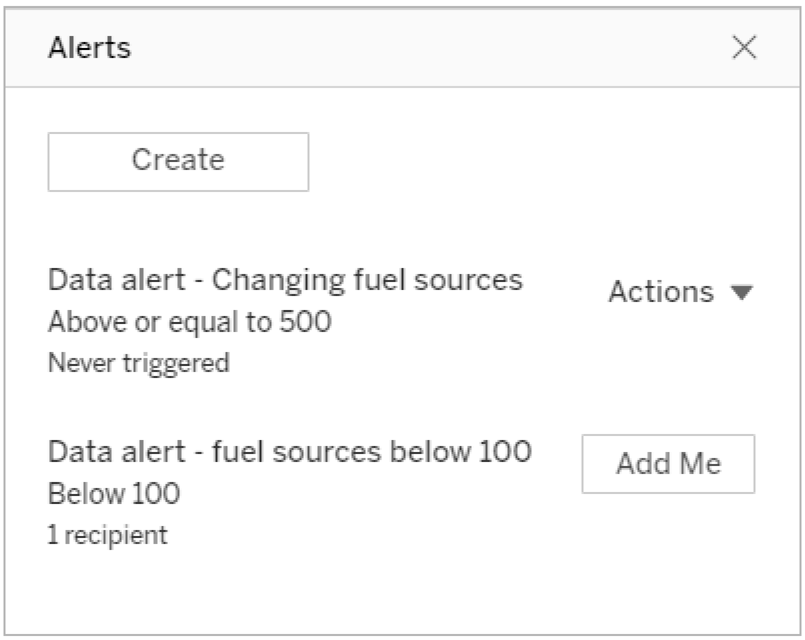
When you explore or author a visualization in Tableau Online or Tableau Server, the left navigation panel disappears to give you more space for analysis. The compact header allows you to navigate through project hierarchy, search the site, and access your favorites as needed.



To learn more about navigating your Tableau site, see **Tour Your Tableau Site** on page 3104.

Subscribe yourself to existing data-driven alerts

When you select **Alerts** in any view, a new panel shows alerts that already exist for the viz you are viewing. Just select "Add Me" to add yourself to an existing alert. To create an alert that is visible for others, select **Make visible to others** when creating the alert. Alerts created prior to 2019.2 are not visible to others, but can be updated to be. To learn more, see **Send Data-Driven Alerts from Tableau Online or Tableau Server** on page 3254.



Version 2019.1

Added in version 2019.1

Help Redesign

- **Right-navigation menu** on page 116

Connect to and Prepare Data

Design Views and Analyze Data

Prepare, Author, and View Data on the Web

Help Design

Right-navigation menu

Updates have been made to help you navigate our content, including a new right-hand menu to quickly scan and select different topics on a page.

Connect to and Prepare Data

Connect to Azure SQL Data Warehouse

Use the Azure SQL Data Warehouse connector to connect Tableau to your Azure SQL Data Warehouse database. For more information, see **Azure SQL Synapse Analytics** on page 480.

Connect to Google Ads

You can now connect Tableau to your Google Ads account. For more information, see **Google Ads** on page 505.

Connect to Google Drive

Directly access flat files on your Google Drive using the Google Drive connector. For more information, see **Google Drive** on page 520.

Connect to MariaDB

Use the MariaDB connector to connect Tableau to your MariaDB data. For more information, see **MariaDB** on page 554.

Use SSL with TIBCO Data Virtualization connector

You can now connect Tableau to TIBCO Data Virtualization using SSL. For more information, see **TIBCO Data Virtualization** on page 660.

Use new Snowflake connector features

The Snowflake connector now supports using OAuth as an authentication method. Support for Booleans and for Median and Percentile functions has also been added. For information about connecting Tableau to a Snowflake data warehouse, see **Snowflake** on page 641.

Use Integrated Authentication with the Denodo connector

You can use Integrated Authentication as an authentication method when connecting Tableau to Denodo. For more information, see **Denodo** on page 493.

Maintain case sensitivity when working with Microsoft Excel data

If you need to maintain case sensitivity for your data when performing joins, for example Japanese data that is Kana-sensitive, you can enable the **Maintain Character Case (Excel)** option from the Data menu. When this option is enabled, Tableau uniquely identifies values with different casing instead of combining them, resulting in a different number of rows.

For more information, see **Join Your Data** on page 883

Design Views and Analyze Data

Sort on two axes and other nested sorting improvements

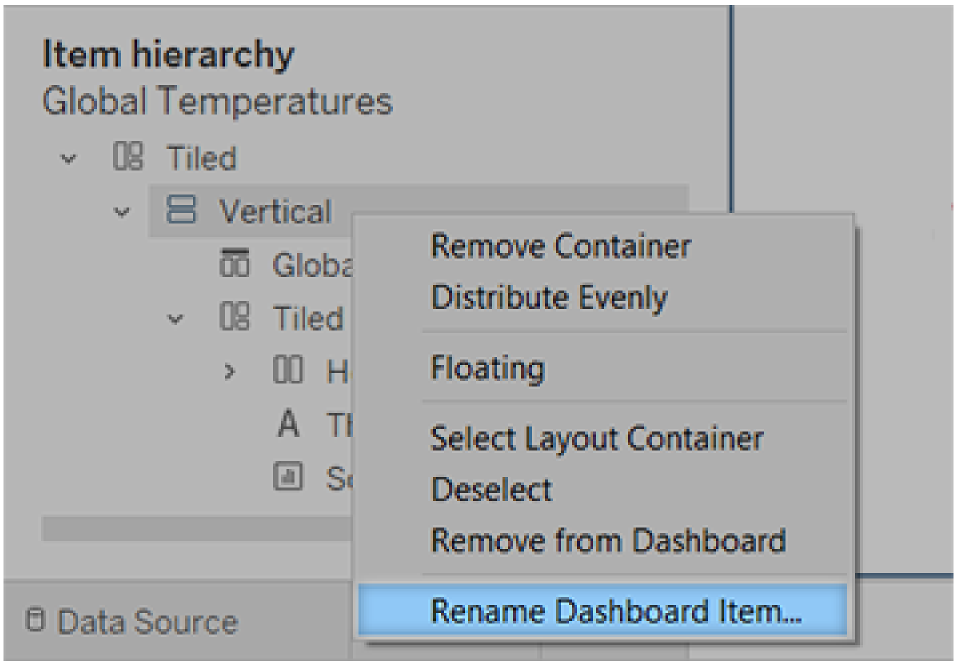
With Tableau Desktop 2019.1, the nested sort feature has received a significant upgrade. New capabilities of nested sort include sorting on two axes, accessing nested sort through the sort dialog, and selecting a single field to perform a nested sort on. For more information, see **Sort Data in a Visualization** on page 1447.

New geocoding data for maps

The geocoding database behind your Tableau map has been updated. While most maps will appear the same as earlier versions of Tableau, in some cases, the shape of the data may look slightly different, or contain different geocoding data. For information about specific location data, see Supported Map Data.

Give dashboard items unique names for easier identification

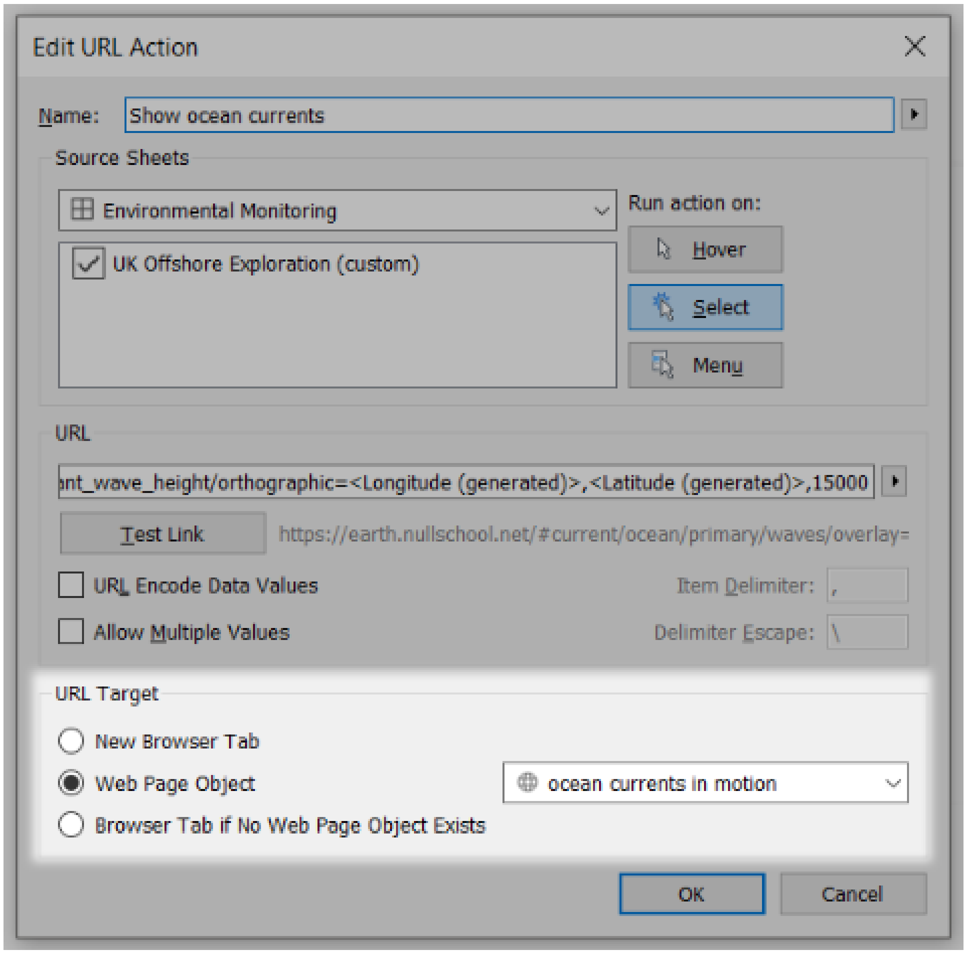
To better distinguish between dashboard items, you can now rename them using the **Item hierarchy** area of the Layout pane or the drop-down menu for each object on the canvas.



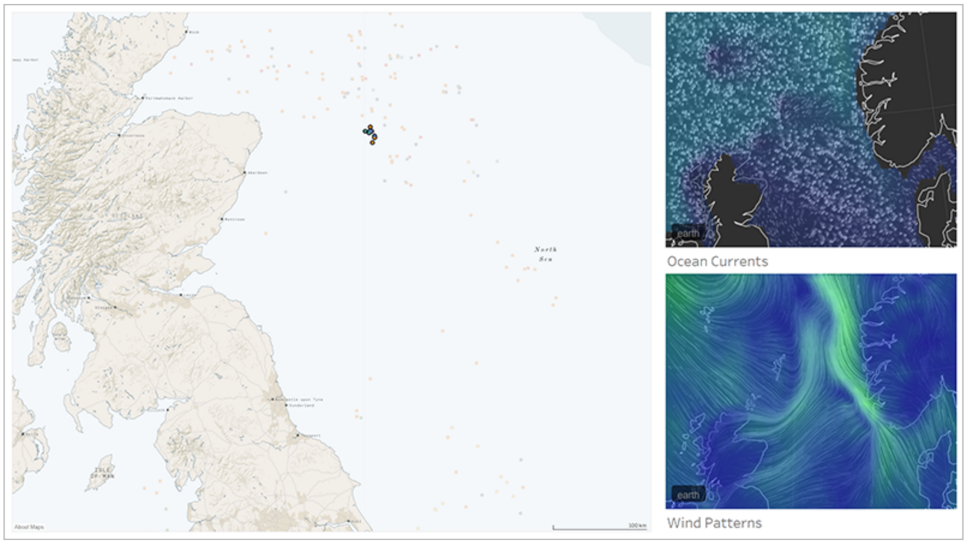
For more information, see **Size, position, reorder, and rename individual dashboard items** on page 2765.

Flexibly target web content from dashboards

For the first time, you can specify the precise target for a URL action, choosing between a new browser tab or a Web Page object. If a dashboard has multiple Web Page objects, you can target each of them with a unique URL action, creating visually dynamic designs. And managing multiple Web Page objects is far easier now that you can rename them.



In this example, selecting marks in the view passes latitude and longitude parameters to two different Web Page objects, each targeted by a unique URL action. (Amazing website images courtesy of earth.nullschool.net.)

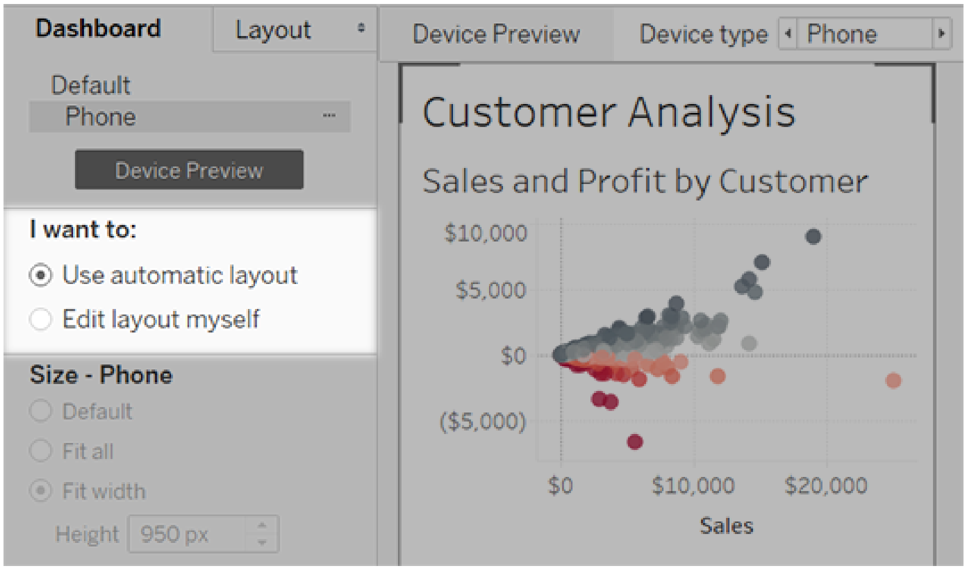


For more information, see **Open a web page with a URL action** on page 1510 and Interactively display a web page in a dashboard.

Create phone layouts automatically

By default, optimized phone layouts are now automatically generated whenever you create a new dashboard. To create them only for specific dashboards, from the **Dashboard** menu, deselect **Auto-Generate Phone Layouts**.

The default **Use automatic layout** option automatically synchronizes any changes to the Default dashboard, saving you time. If you choose **Edit layout myself**, the Phone layout becomes fully independent, so you'll need to manually add and arrange items to reflect changes to the Default dashboard.

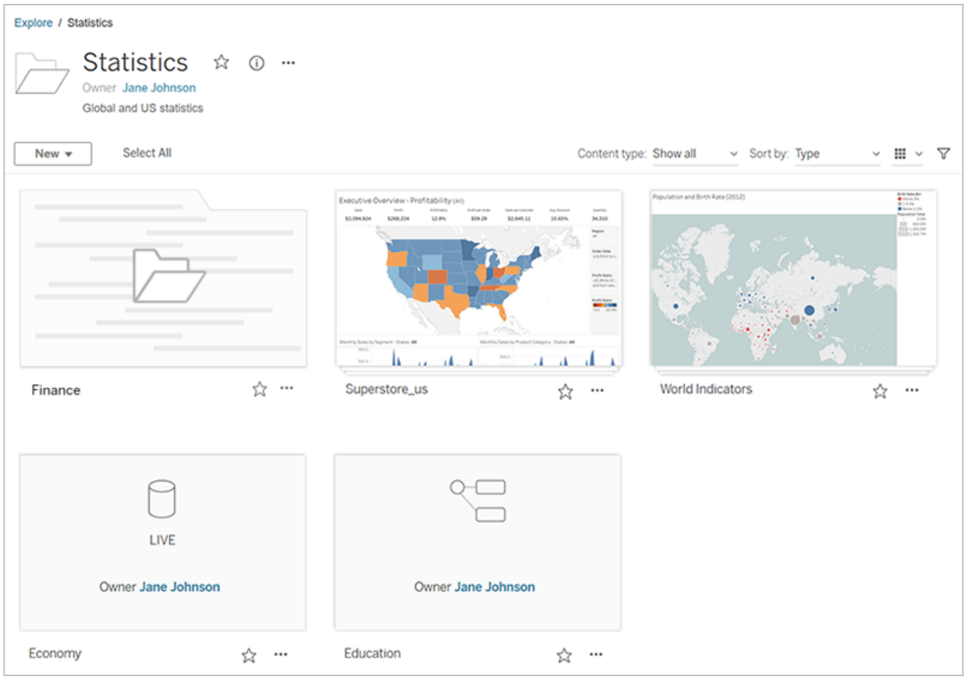


For more information, see **Create Dashboard Layouts for Different Device Types** on page 2801.

Prepare, Author, and View Data on the Web

Enjoy improved content browsing

With Tableau Online and Tableau Server 2019.1, you have a new way to browse shared content. When you open a project, you will be able to see all the content that project contains—workbooks, views, data sources, flows, and nested projects—in a single grid or list. Locating content is easier, because you no longer need to navigate separate pages for different content types. For more information, see **Tour Your Tableau Site** on page 3104.



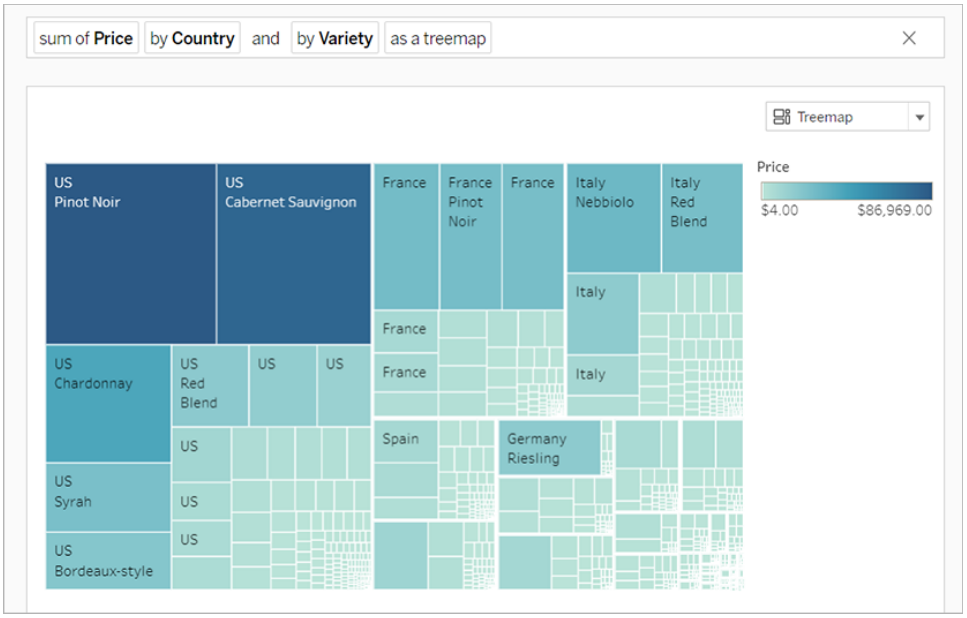
Ask Data to automatically create a viz

Ask Data introduces an entirely new way to interact with your data, letting you type a question and instantly get a response right in Tableau. Answers come in the form of automatic data visualizations, with no need to manually drag-and-drop fields or understand the nuances of your data's structure.

Ask Data lets you ask sophisticated questions naturally, with support for key analytical concepts like time series and spatial analysis, and an understanding of conversational phrases like "last year", "earliest", and "most popular".

Ask Data is available for all user roles with direct access to data sources: Creators, Explorers, and Interactors.

For more information, see **Automatically Build Views with Ask Data** on page 1109.



Use new cloud connectors for web authoring

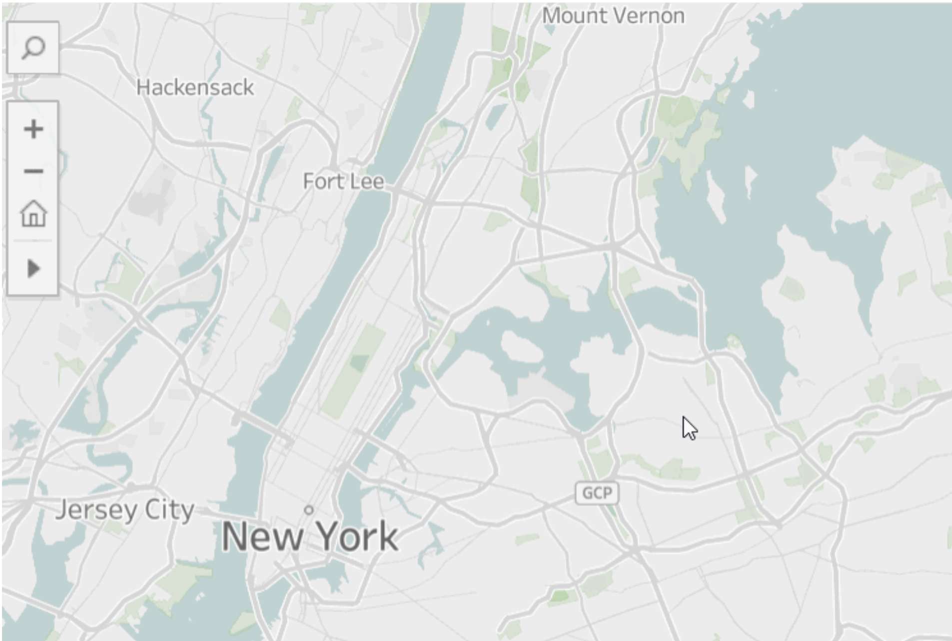
Use the Box, Dropbox, Google Drive, Google BigQuery, and OneDrive connectors to connect Tableau Online or Tableau Server to your data. For more information, see **Creators: Connect to data on the web** on page 3135.

Prepare data for web authoring with these improvements

In 2019.1, authors connecting to a new data source in Tableau Online or Tableau Server can now **Run Initial SQL** on page 736, and can **Connect to a Custom SQL Query** on page 959, and add a Join Calculation to resolve mismatches between fields in a join.

Pan, zoom, and explore with vector tile background maps

Maps created on or published to Tableau Online or Tableau Public now use vector tiles for quicker geographic data exploring. Maps now appear sharper and smoother when you pan and zoom, and labels and icons dynamically resize or reshape to fit your view. The new vector tile maps replace the underlying technology that powers Tableau background maps for a smoother and crisper experience. The background cartography for Tableau maps has also been updated.



See new demographic data in Tableau Online and Tableau Public

Tableau Online and Tableau Public have a new, different source of demographic data than Tableau Desktop. If your map uses a demographic data layer in Tableau Desktop, you may notice some slight differences when that map is published to Tableau Online or Tableau Public, including the time period and value breaks in a legend. In addition, maps that use the “Block Group” Data Layer option in Tableau Desktop will be automatically updated to “Census Tract” when published to Tableau Online or Tableau Public. Six demographic data layers that appear in Tableau Desktop will be absent when maps are published to Tableau Online or Tableau Public:

- Population growth projections
- Household growth projections
- Housing units growth projections
- Effective Buying Income (median)
- Year structure built (median)
- Length of Residence (average in years)

Publishing maps from Tableau Desktop to Tableau Server will not result in any discrepancies.

Toggle between desktop and mobile layouts

If Tableau Online or Tableau Server users find a Phone or Tablet layout too limiting, they can click **See Desktop Layout** in the toolbar. This toggle button lets users switch back to the mobile device layout at any time.

For more information, see **Confirm which layout a device will display** on page 2814.

Filter improvements

Use Ctrl-click (control-click on a Mac) to select multiple values in a multi-value filter in the view. The selected values can be non-contiguous and are indicated with a gray highlight. When the values are selected, you can then click any one of their check boxes to clear or choose the values at the same time.

Save and Publish Data Sources and Workbooks

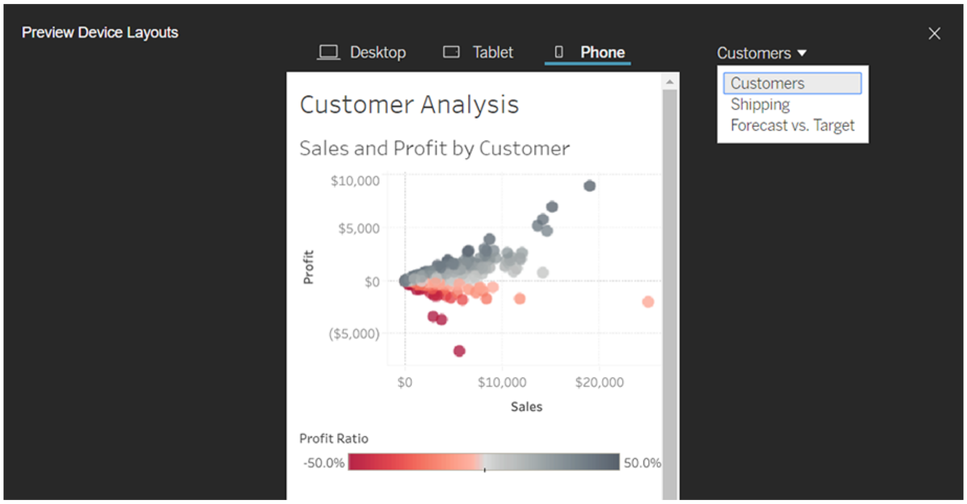
Export and download workbooks to PowerPoint

Export all the views and dashboards in your workbook into PowerPoint with one click. This option creates a new PowerPoint file with static images of each sheet in your workbook. When the file is generated from Tableau Online or Tableau Server, the new PowerPoint file provides a link back to the workbook where you can see the most up-to-date data and interact with it. To optimize your images for PowerPoint, select Size > PowerPoint (1600x900) in the dashboard layout. For more information, see **Export Views from Tableau Desktop to Another Application** on page 3032.

Preview device layouts on the web

To see how your designs appear on mobile devices, click **Preview device layouts** in the browser when publishing is complete. Then choose a device type above the viz, and a specific dashboard from the menu at right.

For more information, see **Comprehensive Steps to Publish a Workbook** on page 3058.



Collaborate with Tableau Online and Tableau Server

Get notified when subscriptions are suspended

When you subscribe to a view or a workbook, you receive an email snapshot of that content at regular intervals. Now, you'll receive an email notification if something caused your subscription to fail, with links to resume or delete the failing alert. For more information, see **Create a Subscription to a View or Workbook** on page 3249.

Version 2018.3

Added in version 2018.3

Help Redesign

- **Search, navigation, and layout improvements** on the next page

Connect to and Prepare Data

- **Store extract data in multiple tables** on the next page
- **New Other Databases (JDBC) connector** on page 129

Design Views and Analyze Data

Prepare, Author, and View Data on the Web

Help Redesign

Search, navigation, and layout improvements

Search results within the product help have been expanded to include results for all Tableau products and help articles. Help layout and navigation, such as table of contents placement and breadcrumbs, have been updated to create a more consistent cross-product help experience.

Connect to and Prepare Data

Store extract data in multiple tables

You can configure you extract to have its data stored in multiple tables. Storing your extract data in multiple tables can potentially improve performance and help reduce the file size. For more information, see **Decide how the extract data should be stored** on page 1022.

New Other Databases (JDBC) connector

Use the **Other Databases (JDBC)** connector to connect to files or database types not listed under **Connect** and that use a database driver that implements the JDBC (Java Database Connectivity) standard. For more information, see **Other Databases (JDBC)** on page 670.

New supported spatial files

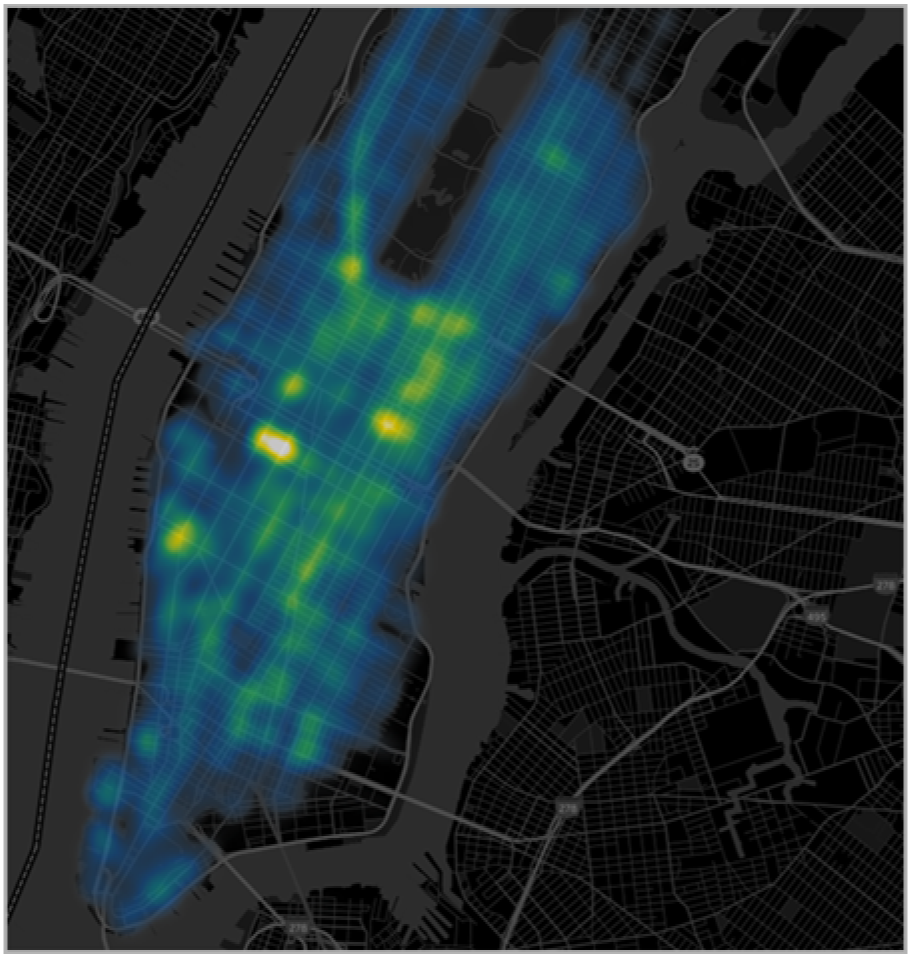
Use the Spatial Files connector to connect to TopoJSON files, Esri File Geodatabase, and multi-layer KML (Keyhole Markup Language) files using the spatial file connector. For more information, see **Create Tableau Maps from Spatial Files** on page 1855.

Design Views and Analyze Data

Build density maps and charts

Create and interact with in Tableau Desktop, Tableau Online, or Tableau Server

Use the density mark to create maps and charts that can help you identify "hot spots" in your data. The density mark lets users see insights into densely populated data and visualize it as a smooth surface which can reveal patterns or relative concentrations that might otherwise be hidden due to overlapping marks. For more information on building density maps and charts, see **Create Heatmaps that Show Trends or Density in Tableau** on page 1910, **Build with Density Marks (Heatmap)** on page 1600, and Density Marks.

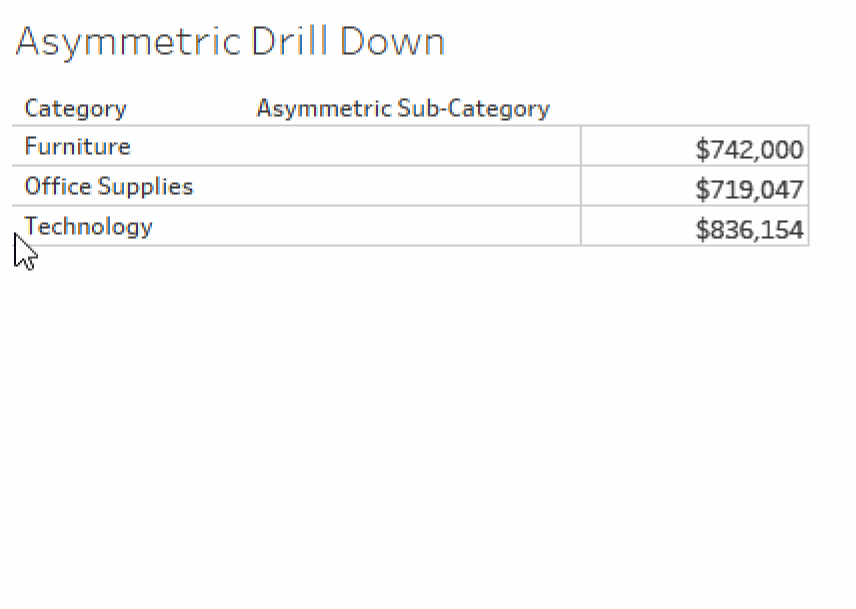


Use set actions to let your audience change set values

Create in Tableau Desktop for interaction in Tableau Desktop, Tableau Server, or Tableau Online

Let your audience select what values are in a set by directly interacting with marks on a viz. You can use set actions to visually drive calculations and create visualizations with multi-faceted comparative analysis like proportional brushing.

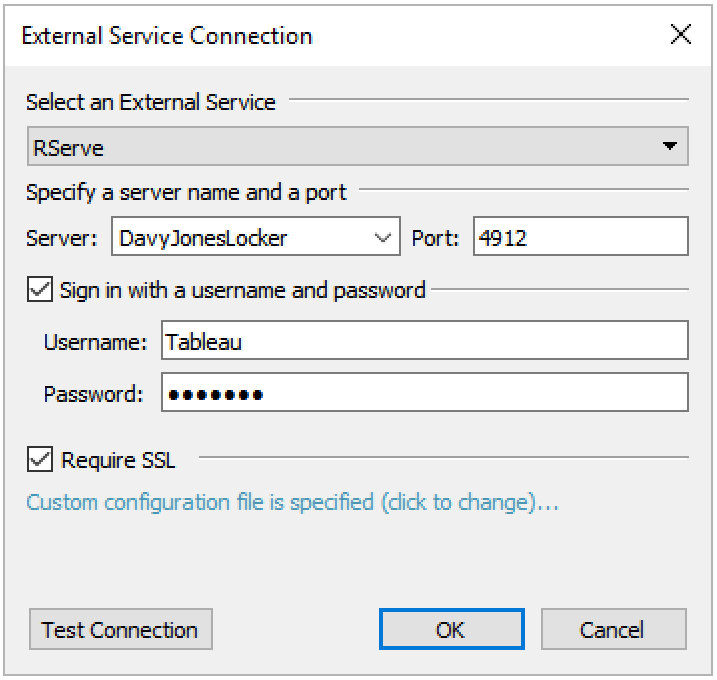
Set actions can be used across multiple sets for more complex interactions. In this example, sets and calculated fields are working together with set actions for Category and Sub-Categories to update the viz based on what marks are selected. When a user clicks a category, its subcategories are displayed. When a user clicks a subcategory, its manufacturers are displayed. For more information, see **Set Actions** on page 1540.



Set actions must be created in Tableau Desktop to be available for use in Tableau Online and Tableau Server.

Encrypted RServe connections

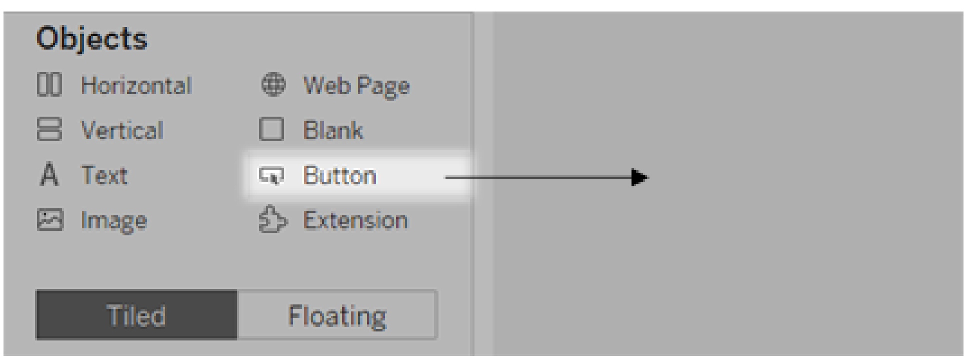
Connect to RServe as an external service over an encrypted channel from Tableau Desktop. Transfer data with an SSL/TLS secured connection for RServe integration. For more information, see **Pass Expressions with Analytics Extensions** on page 2690.



Navigate from dashboards more easily

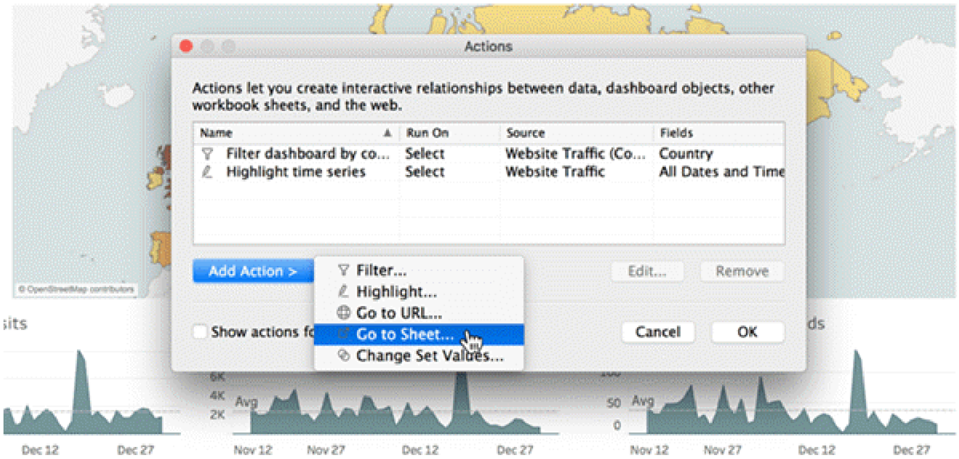
Two new dashboard features make navigating to other sheets much easier. Both maintain selected filters and marks and let you navigate to sheets published as separate views (without visible tabs).

- The Button object provides a quick way to navigate from one dashboard to another, or to other sheets or stories. To reveal a button's destination, add a custom image and informative tooltip text that appears when users hover over the object.



For more information, see **Add dashboard objects and set their options** on page 2743.

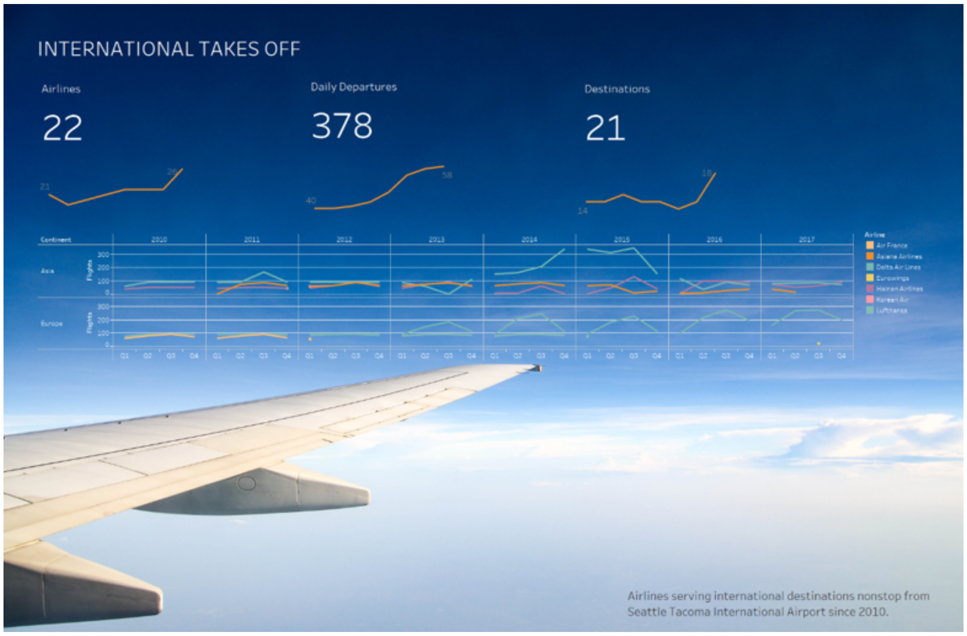
- The Go to Sheet action, which you can create only in Tableau Desktop, lets users navigate to other dashboards, sheets, or stories when they click on a mark or a tooltip menu item.



For more information, see **Actions and Dashboards** on page 1566.

Transparent worksheet backgrounds in dashboards

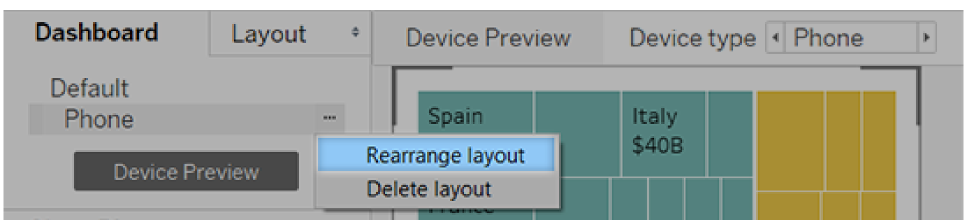
To make a worksheet transparent, set its background color to **None**. Visually integrate elements across your entire dashboard by combining transparent worksheets with transparent filters, highlighters, and parameters. For more information, see **Visually integrate elements with transparency** on page 2771.



Automatically rearrange phone layouts

As you adjust dashboard items on a phone layout, click the pop-up menu to the right of **Phone**, and select **Rearrange layout** to optimize the placement of filters, remove white space, and more.

In the example below, filters are automatically moved higher so phone users don't need to scroll down to see them.



For more information, see **Create Dashboard Layouts for Different Device Types** on page 2801.

Prepare, Author, and View Data on the Web

Improved content browsing

With Tableau Online and Tableau Server 2018.3, you can get a sneak preview of our reorganized web authoring environment (UI). The new UI enables you to see multiple content types—views, workbooks, data sources, flows, projects—in one grid or list. For example, you can now see all content inside a particular project and its sub-projects.

This new experience intends to make content discovery more intuitive, and it's the first in a set of planned changes that are aimed at improving self-service and governance of your Tableau content.

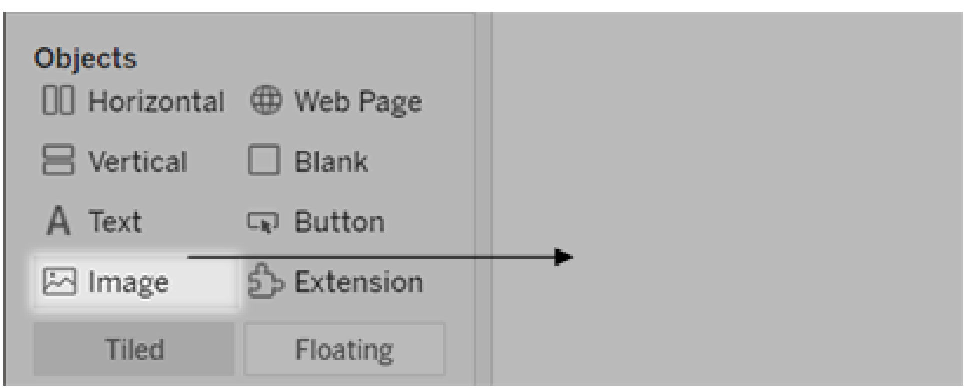
To try the new UI, ask your administrator to enable the site-level setting we've provided for it. For more information, see Preview the Updated Content Browsing Experience in the Tableau Server 2018.3 Help (this article applies to Tableau Online, too).

After you get access to the new UI, sign in to our Early Feedback site. There you can share your thoughts with us and get our regular development updates.

Add image objects to your dashboards

You can now add images to dashboards when authoring on the web. The new Image object lets you adjust image fit, add a target URL that opens when the image is clicked, and add alt text, which improves the accessibility of your dashboard for those who use screen readers.

While viewing a dashboard on the web, click **Edit**. From the Objects section at left, drag **Image** to the dashboard on the right.



For more information, see **Add dashboard objects and set their options** on page 2743.

Duplicate as Crosstab

To create a new cross-tab sheet based on the data in the current sheet in Tableau Online or Tableau Server, right-click the sheet tab (control-click on Mac) and select Duplicate as Crosstab. This inserts a new worksheet into your workbook and populates the sheet with a cross-tab view of the data from the original worksheet.

Create, move, and resize annotations

Create and delete annotations by right-clicking the view (in a worksheet or dashboard), and then selecting the type of annotation: point, mark, or area. Click the annotation to select it, then drag to move it, or click a selection handle to resize it. Right-click the annotation again to edit or remove the annotation.

Version 2018.2

Added in version 2018.2

Install and Deploy Tableau

- **Install Tableau Bridge separately from Tableau Desktop** on page 138

Connect to and Prepare Data

- **Join spatial files** on page 138
- **Enhancements for Intuit QuickBooks Online connector** on page 138

Design Views and Analyze Data

- **Perform a nested sort with one click** on page 138
- **Extend functionality with dashboard extensions** on page 139
- **Visually align dashboard items with a grid** on page 140

Prepare, Author, and View Data on the Web

Collaborate with Tableau Online and Tableau Server

Save and Publish Data Sources and Workbooks

Install and Deploy Tableau

Install Tableau Bridge separately from Tableau Desktop

Tableau Bridge is now a stand-alone client that you install separately from Tableau Desktop. For more information about Tableau Bridge requirements, compatibility with previous versions of Tableau Desktop, and the installation process, see Install Tableau Bridge in the Tableau Online Help.

Note: If you install Tableau Bridge on the same computer as Tableau Desktop, you can start Tableau Bridge from the Server menu like you did in previous versions.

Connect to and Prepare Data

Join spatial files

You can now join two spatial files, spatial database tables (Microsoft SQL Server only), or spatial fields across two data sources. For more information, see **Join Spatial Files in Tableau** on page 1815.

Enhancements for Intuit QuickBooks Online connector

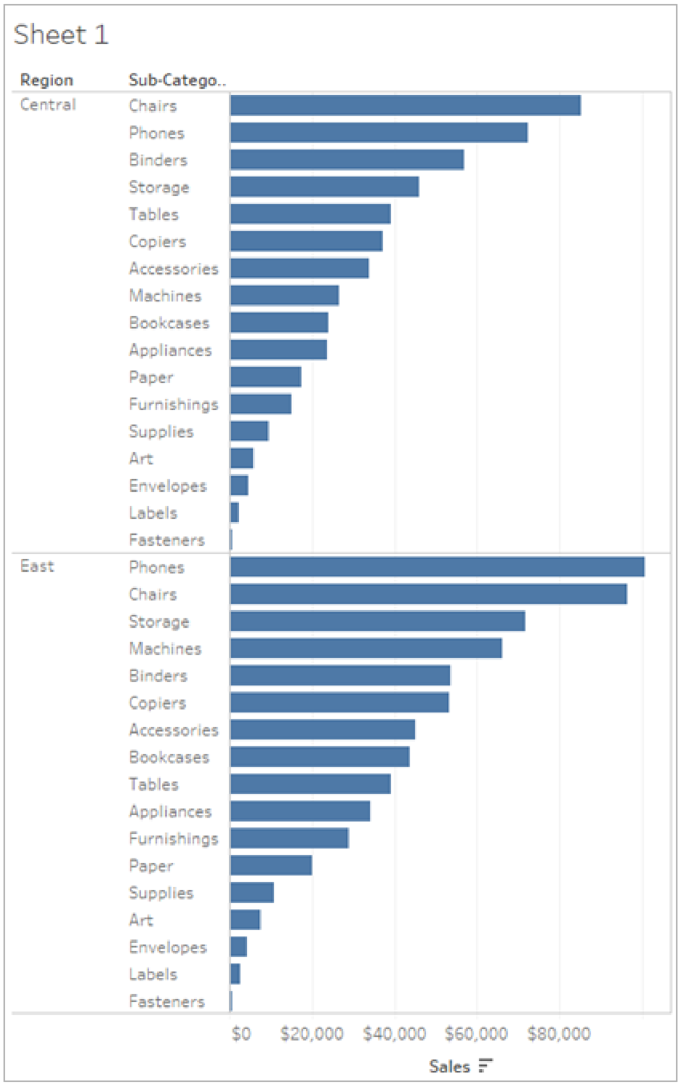
There are several changes and enhancements to the Intuit QuickBooks Online connector that provide you with more flexibility. New transaction types are included in the Sales and Expenses Line Items table: Vendor Credit, Credit Memo, Journal Entry, Deposit, Estimate, Bill Payment, Payment, Purchase Order and Refund Receipt. Additionally, account information has been added to Credit Memo and Vendor Credit.

Individual transaction tables now have a new field called "Transaction Type" so you can join other tables to the Sales and Expenses Line Items table.

Design Views and Analyze Data

Perform a nested sort with one click

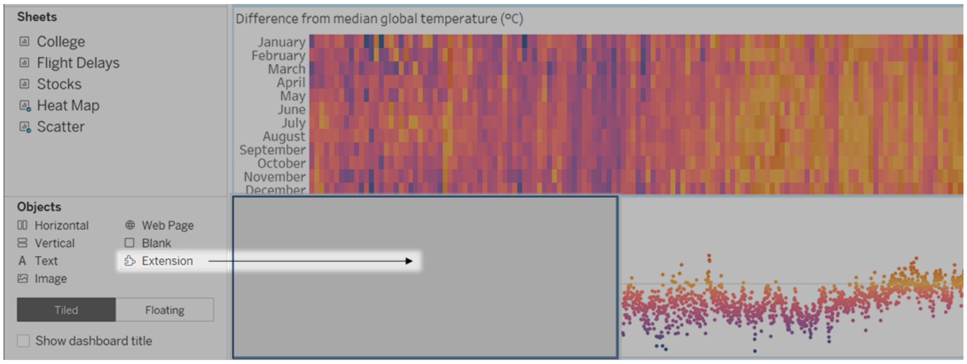
You can now use a nested sort operation without having to create calculated fields or combine dimensions. For more information, see **Nested sorts** on page 1452



Extend functionality with dashboard extensions

Extensions increase dashboard functionality with custom objects created by third-party developers. With extensions, you can add unique features to dashboards or directly integrate them with applications outside Tableau.

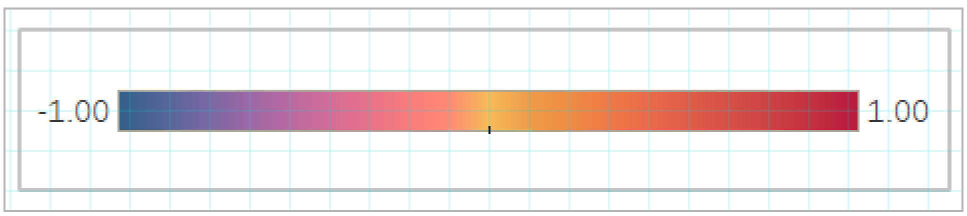
Adding an extension is easy. In Tableau Desktop or the web-authoring mode of Tableau Online or Tableau Server, simply drag the Extension object to a dashboard. Then click either **Extension Gallery** to choose from all available extensions, or **My Extensions** to select an extension you previously downloaded. (To directly browse the Extension Gallery, go to extensiongallery.tableau.com.)



For more information, see **Use Dashboard Extensions** on page 2839.

Visually align dashboard items with a grid

To present a visually consistent design, arrange and size dashboard items over a grid. Choose **Dashboard** > **Show Grid**, or quickly toggle the grid on and off by pressing the G key.



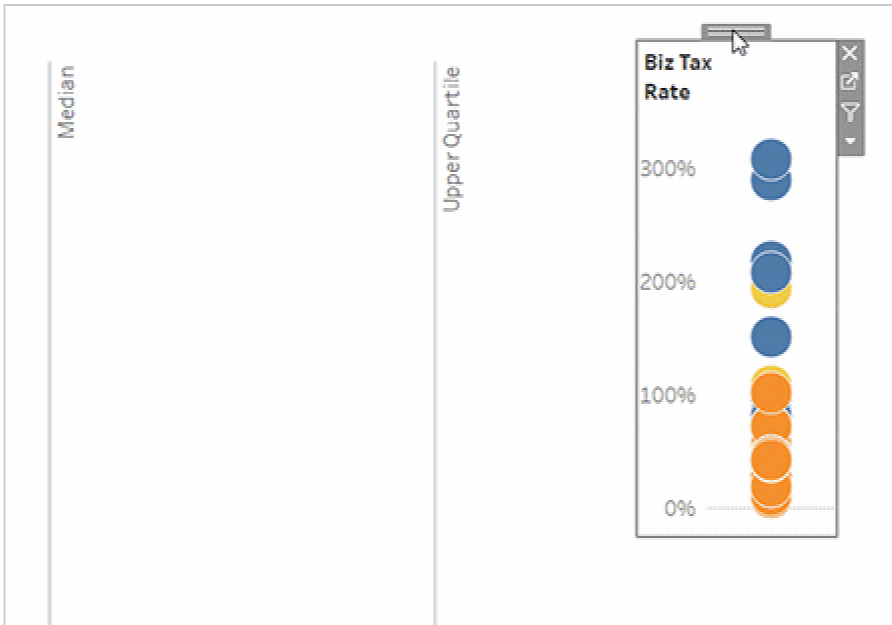
For more information, see **Size, position, reorder, and rename individual dashboard items** on page 2765.

Precisely position and size floating items with arrow keys

To precisely position a floating item, press arrow keys to move 1 pixel, or Shift+arrow keys to move 10 pixels. To resize items, add Alt (Windows) or Option (macOS) to the shortcuts above.

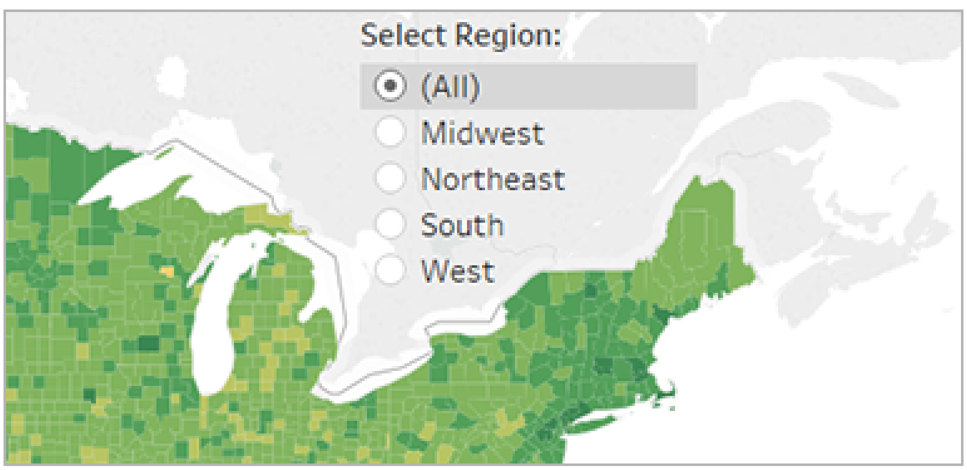
See a preview while dragging floating dashboard items

To give you a clear picture of how moved items will look in a new location, a preview appears while you drag them.



Float transparent filters, parameters, and highlighters over dashboards

To visually connect filters, parameters, and highlighters to related data, float these items, which are now transparent by default. Text always remains fully opaque, maintaining legibility.



For more information, see **Tile or float dashboard items** on page 2763.

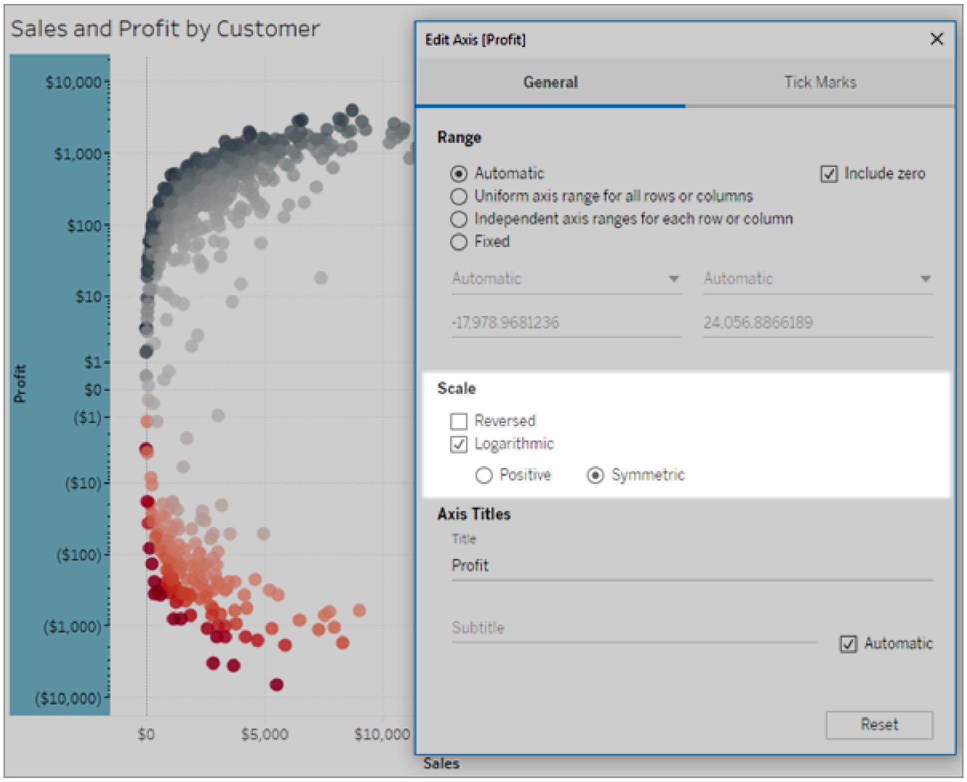
Automatically optimize dashboard layouts for mobile devices

When you create a device layout, Tableau Desktop now automatically optimizes for the device type, adjusting the arrangement of dashboard items to best fit a phone or tablet. Often, these automatic layouts will address all the needs of your mobile users, but you can always customize a layout further.

For more information, see **Create Dashboard Layouts for Different Device Types** on page 2801.

Show negative values on a log axis

In Tableau Desktop and web authoring, when you select **Logarithmic** scale for an axis, you now have the option to specify **Symmetric** to display data that contains 0 or negative values on a log scale axis. For related details, see **Change the axis scale to reversed or logarithmic** on page 2908.



Use ISO 8601 date standard in calculated fields

When creating a date calculation, you now have the option to use the ISO 8601 international standard to resolve the calculation. For more information, see **Date Functions** on page 2170.

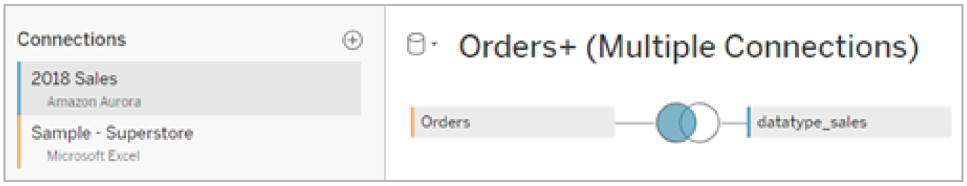
Prepare, Author, and View Data on the Web

Keyboard shortcuts for web authoring

Use keyboard shortcuts to quickly create and edit views on the web. For more information, see Shortcuts for web authoring.

Create multi-connection data sources and cross-database joins

Add a second connection to your data source and combine tables from different databases using a cross-database join. Your multi-connection data source can be saved for others in your organization to use. To learn more about cross-database joins, see **Join Your Data** on page 883.



Union your data on the web

Can create new unions and add to unions from the same connection in Tableau Online and Tableau Server. To learn more, see **Union Your Data** on page 942.

Pivot Data on the web

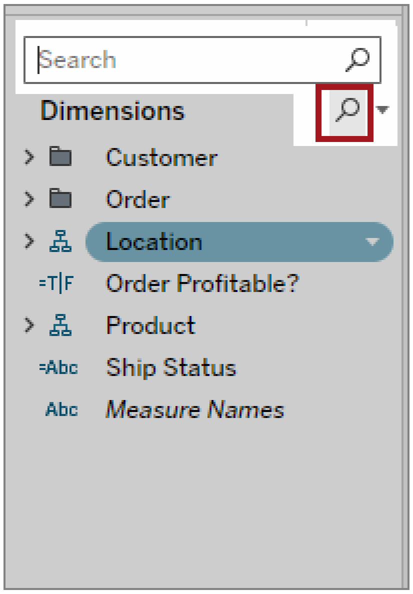
When working with file-based data sources on the web, you can pivot your data from crosstab format into columnar format. For more information, see **Pivot Data from Columns to Rows** on page 988.

Copy data on the web

Copy values in your data grid by selecting the values and then pressing Ctrl+C (Windows) or Command+C (macOS). Alternatively, to copy values in the metadata grid, select the values, right-click, and select Copy.

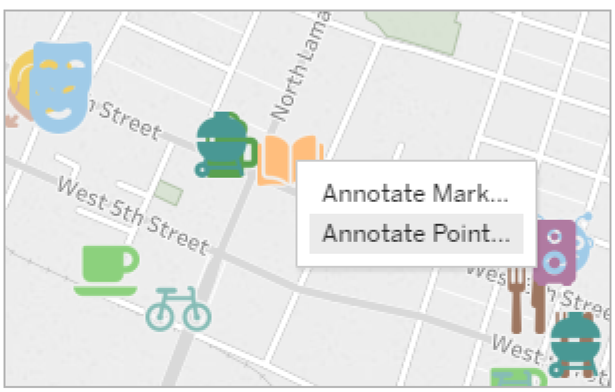
Schema search

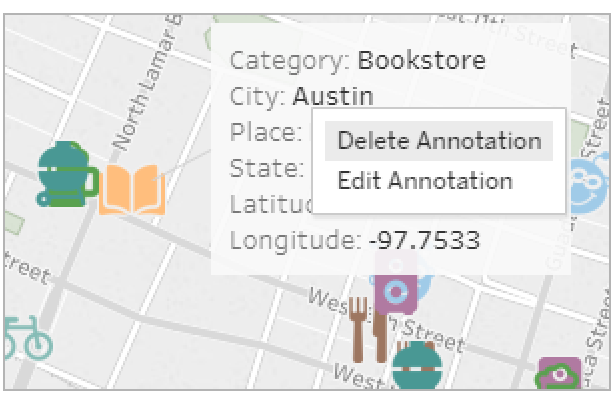
Quickly locate data fields in the Data pane using schema search.



Annotation improvements

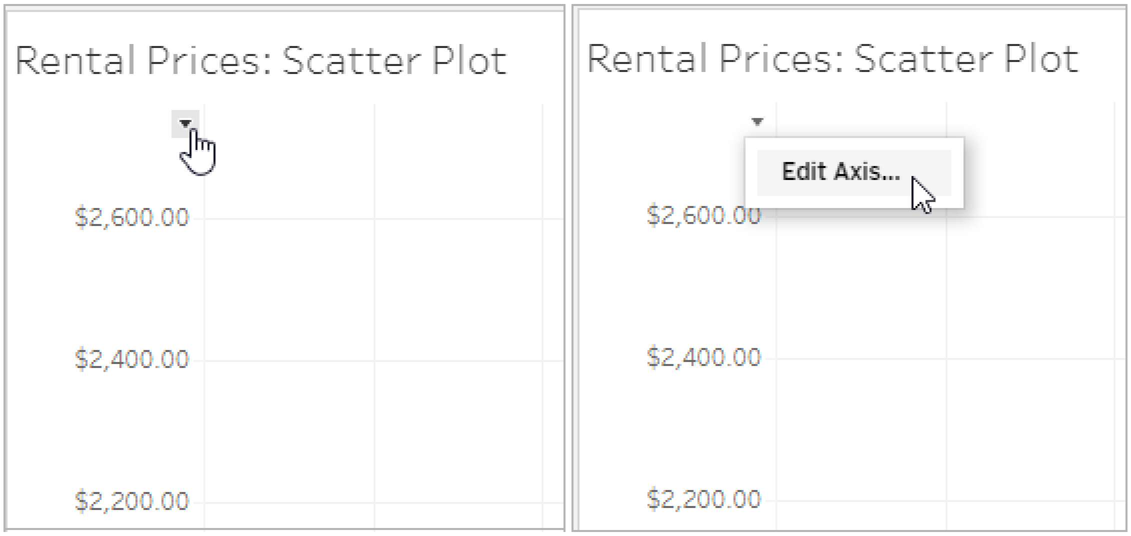
Create and delete annotations by right-clicking the view (in a worksheet or dashboard), and then selecting point or mark annotation. Right-click the annotation again to edit or remove the annotation.





Edit Axis button on axes

You now have two ways to open the **Edit Axis** dialog box. Hover near the top of the axis (vertical axis) or to the right (horizontal axis) and click the drop-down arrow that appears. Or, double-click the axis.



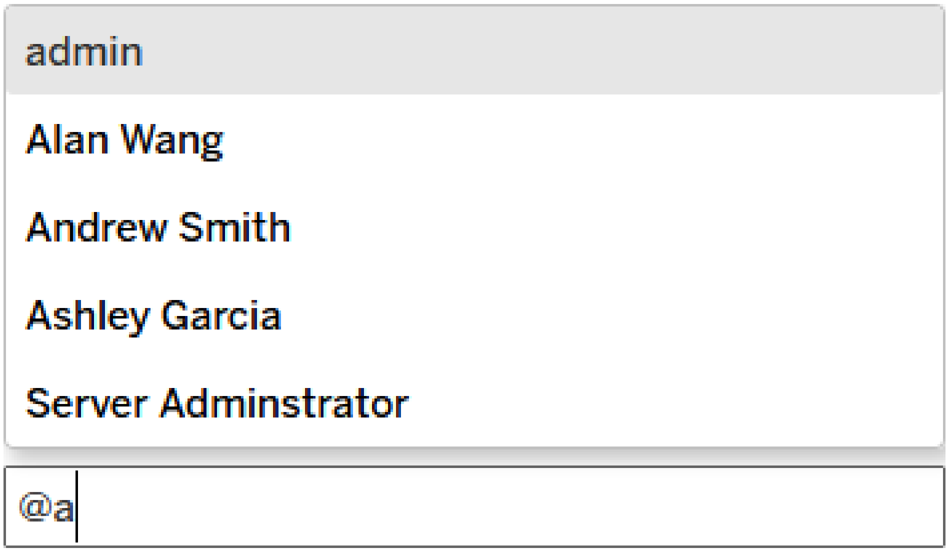
Better control over sorting

You can now access the **Sort** dialog box by right-clicking a dimension field on the Rows or Columns shelves in a view.

Collaborate with Tableau Online and Tableau Server

Include colleagues on data conversations with @mentions

To engage people in a conversation about compelling data you find in a view, @mention them in comments. As you begin to type a name, matching people with Tableau accounts at your organization appear in a list you can choose from. For more information, see **Comment on Views** on page 3259.



Quickly create data alerts on customized views

If you've changed a view (for example, by adjusting filters), you can now create alerts without first saving a custom view. For more information, see **Send Data-Driven Alerts from Tableau Online or Tableau Server** on page 3254.

Save and Publish Data Sources and Workbooks

Check how your workbook will look in an older version

It's important to know how your workbook will be affected before you export it to an older version of Tableau. The **File** > **Export As** command now gives you a clear picture, telling you if your workbook will change and which features are unsupported.

Version 2018.1

Added in version 2018.1

Tableau Licensing

Install and Deploy Tableau

Connect to and Prepare Data

Design Views and Analyze Data

Author and View Data on the Web

Tableau Licensing

New user-based term licenses are now available

Tableau now offers different types of user-based term licenses that grant a range of web authoring and other capabilities at various price points. This gives organizations more flexibility to fit licenses to the data analysis and visualization needs of different users. To learn more, see User-based licenses, or to see the capabilities of each license, see Tableau Pricing.

Install and Deploy Tableau

Register Tableau Desktop from the command line

When you install Tableau Desktop from the command line (for Windows) you can now register the product as part of the install without having to complete the registration for new users through the user interface or as a separate command line operation.

Add a single command line that includes the `REGISTER="1"` installer option. During the installation process, the installer will run the `-register` process and add the registration information. This process uses the registration information from the following location in the registry: `HKEY_CURRENT_USER\Software\Tableau\Registration\Data`.

For information about adding registration information to the registry and the property settings for the `-register` property, see the "Register Tableau Desktop" and the "Installer properties" sections in Deploy Tableau Desktop.

Mac OS system requirements

Tableau Desktop version 2018.1 for the Mac requires Mac OS X 10.11 El Capitan or later. All Tableau Desktop versions 10.0 through 10.5 are compatible with Mac OS X 10.10 Yosemite.

Connect to and Prepare Data

Support for remote config files for SAP NetWeaver Business Warehouse

Tableau now supports using landscape XML files located on a remote HTTP server.

Connect to spatial columns in Microsoft SQL Server

You can now use spatial columns when you connect to Microsoft SQL Server in Tableau. You can also use Custom SQL and RAWSQL to perform advanced spatial analysis in Tableau. For more information, see **Connect to Spatial Data in a Database** on page 1799.

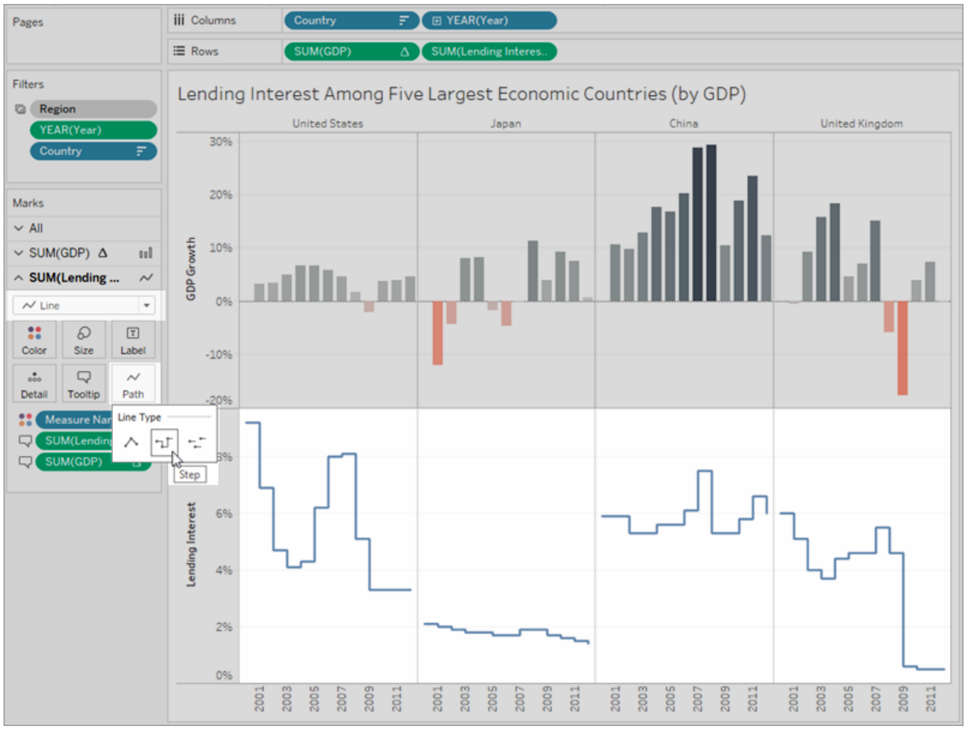
New connector name

Starting in Tableau 2018.1.4, the Cisco Information Server connector is renamed to **TIBCO Data Virtualization**.

Design Views and Analyze Data

Apply step lines and jump lines

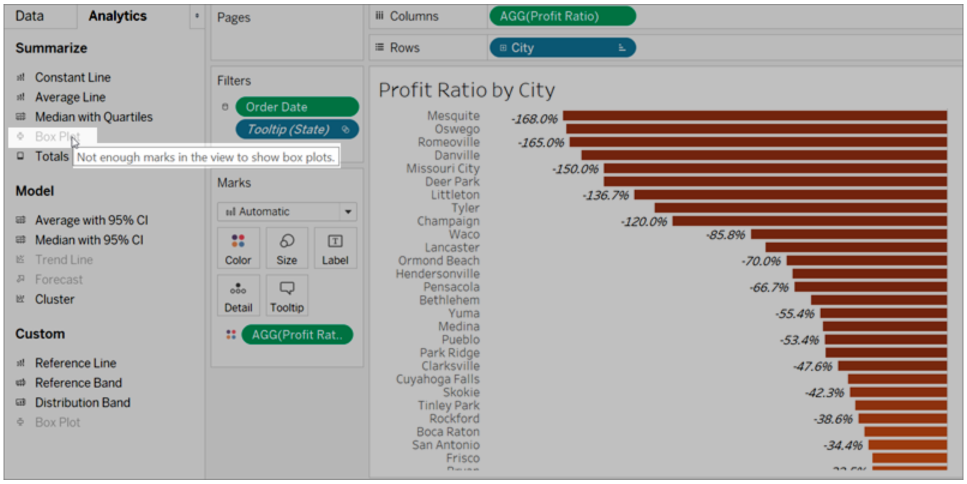
Change the Line mark type to a step line or a jump line by clicking the Path property in the Marks card.



Step lines and jump lines can help you see and compare incremental patterns of change over time. Use these line types for numeric data that remains constant for periods of time, with noticeable changes or deltas – such as account balances, inventory levels, or interest rates. For details, see **Draw paths between marks** on page 1376.

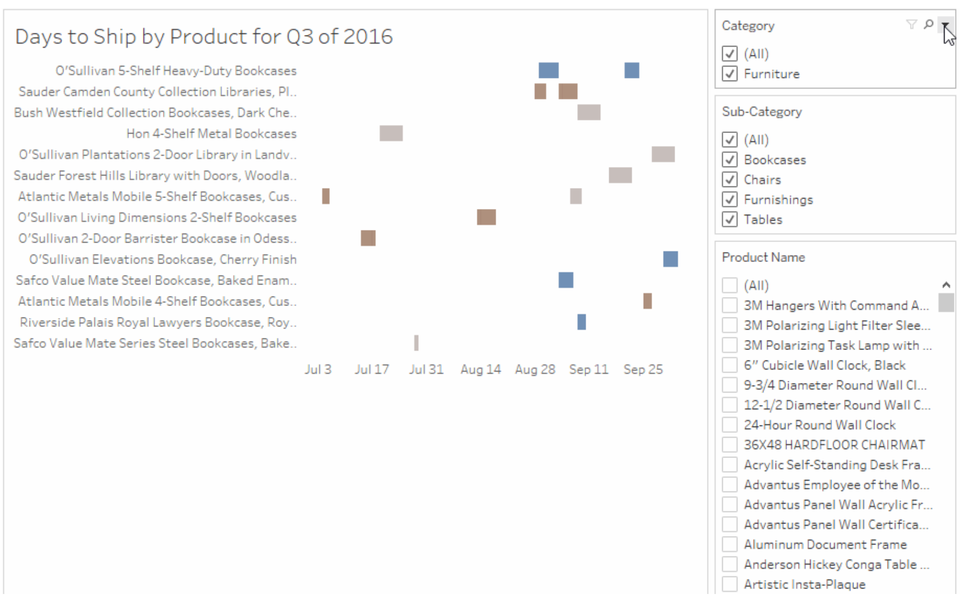
Tips for analytics objects

In the Analytics pane, available analytics objects vary based on the current state of the data in a visualization. For example, you can't apply trend lines to the view when it contains stacked marks. Tooltips in the Analytics pane explain why an Analytics object isn't available, or provide a tip on what is required to use that object in the view.



Filter hierarchical data

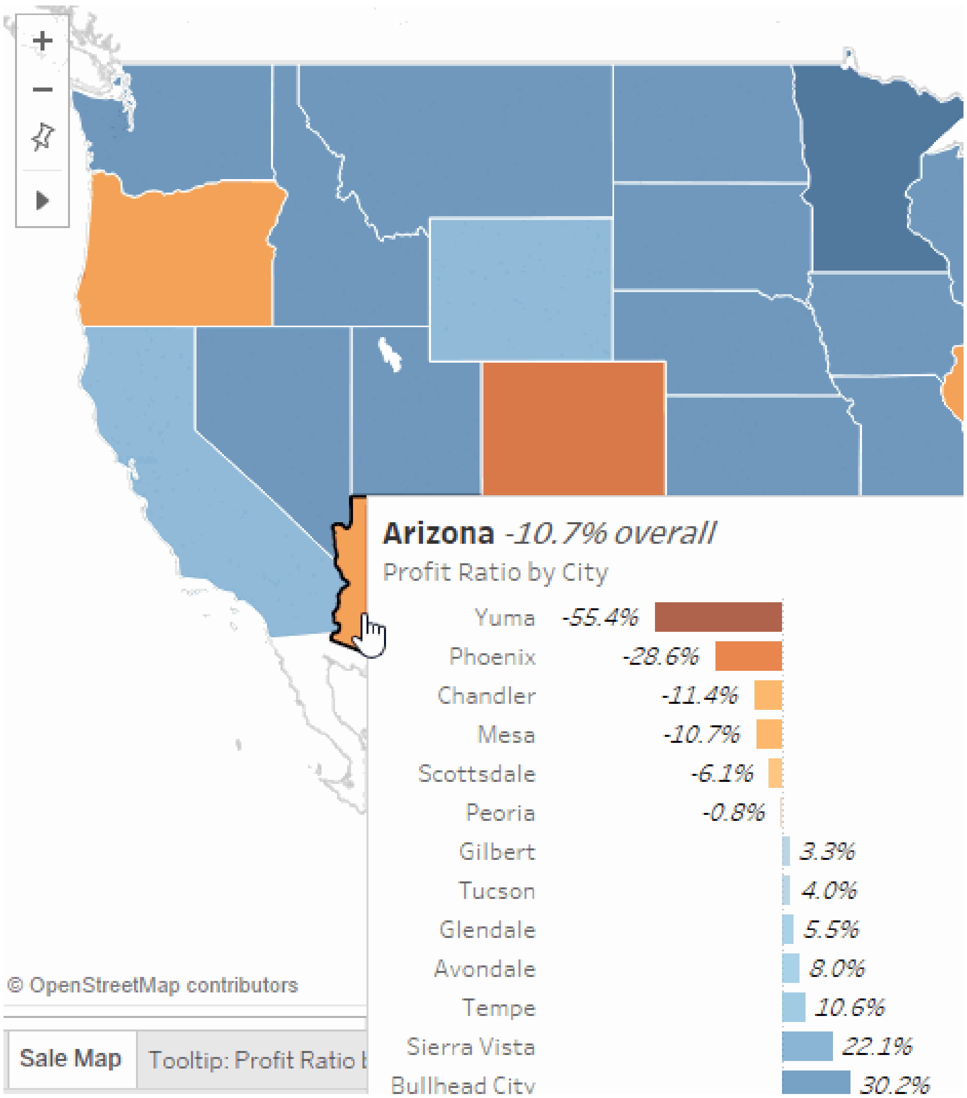
When the data in your view contains hierarchies, filters for those fields can use the **All Values in Hierarchy** option. Your filtering experience now matches the parent/child relationships in the hierarchy.



For details on filtering options, see **Set options for filter card interaction and appearance** on page 1423.

Hide a Viz in Tooltip worksheet

You can now hide worksheets used as Viz in Tooltips, the same way you would hide worksheets used in stories or dashboards. In the target worksheet that is the Viz in Tooltip, click **Hide**.



To show the Viz in Tooltip worksheet again, in the source worksheet, click **Unhide All Sheets**. For more details about Viz in Tooltip, see **Create Views in Tooltips (Viz in Tooltip)** on page 1475.

Refit saved clusters

Clusters are saved as groups so you can use them in other worksheets and workbooks, however, they don't automatically refresh. When the underlying data changes, you can now use the **Refit** option to refresh and recompute the data for your saved clusters. For related details, see **Refit saved clusters** on page 2562 in **Find Clusters in Data** on page 2554.

Synchronize dual axes for different numeric data types

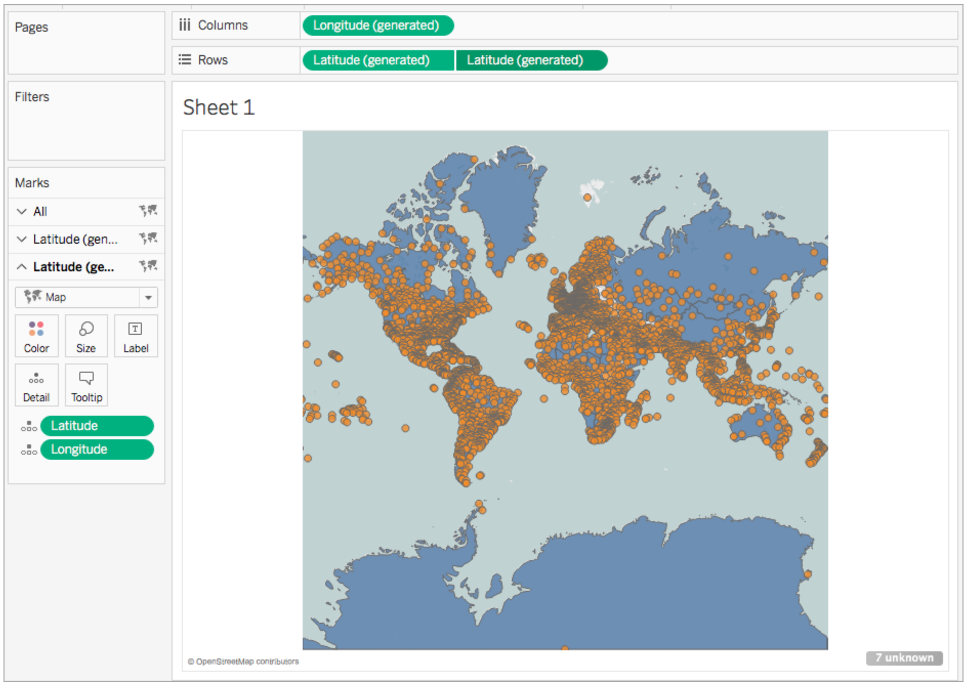
In a chart with dual axes, you can now synchronize dual axes for numeric data types that don't match. For example, you can synchronize an axis that uses an integer data type and an axis that uses decimal data type.

Navigate stories with arrows alone

If you like a clean, simple navigator style, select the new arrows-only option. For more information, see **Create a Story** on page 2852.

Create dual-axis (layered) maps from generated and custom latitude and longitude fields

You can now create dual-axis maps from a combination of Tableau generated latitude and longitude fields, and custom latitude and longitude fields. For more information, see **Create a dual-axis map from a combination of generated and custom latitude and longitude fields** on page 1948.



Author and View Data on the Web

Connect to data on the web

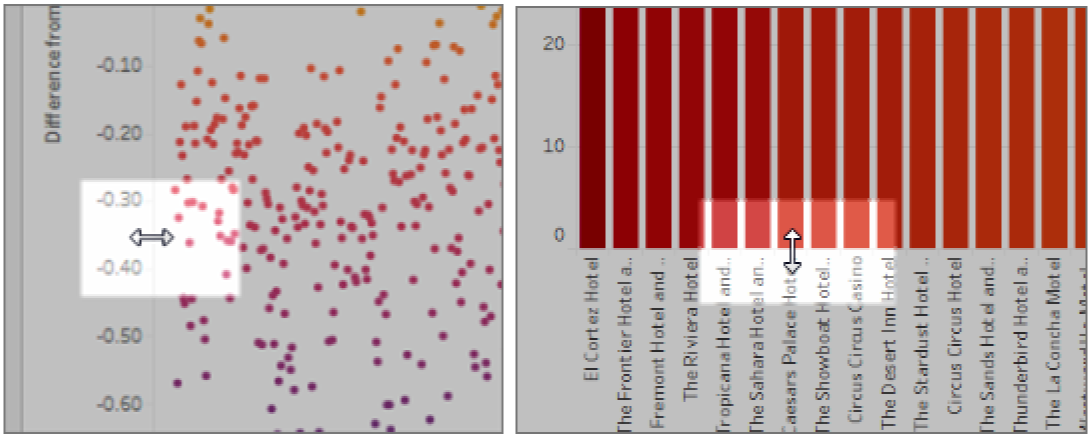
With a Creator user license, you can connect to data directly in your browser. Upload Excel or text-based data sources, connect to data housed in a cloud database or on a server in your enterprise, or connect to published data saved to your site. For more information, see Creators: Connect to data on the web.

Prepare your data on the web

With a Creator user license, use the Data Sources tab to prepare your data for analysis. Create joins, add new data sources, clean your data using the Data Interpreter, and more. For more information, see Creators: Prepare Data on the Web.

Resize headers on the web

Resize the width of row headers and the height of column headers.



Add an accessible toolbar to an embedded view

Authors can enable an accessible toolbar so that all users can add comments, download the view, or share it with others. For more information, see Publish and embed the view in Author Views for Accessibility.

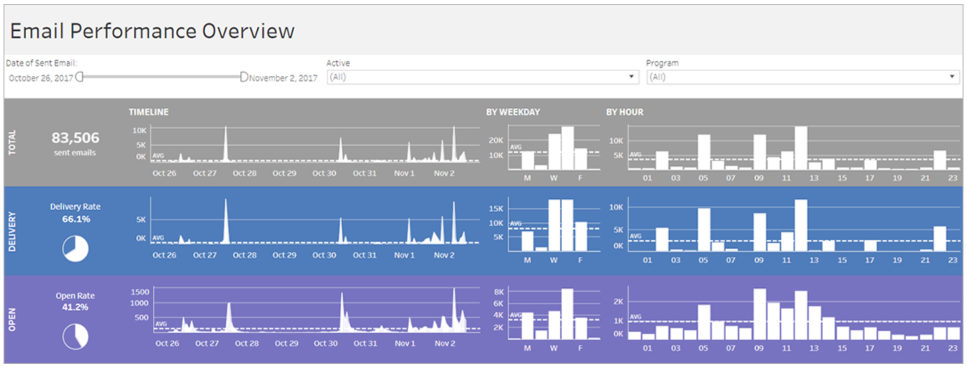
Other web authoring enhancements

- Show Me is now available for dashboards.
- Double-click an annotation to edit its text.
- Hide or show a Viz in Tooltip worksheet. In the target worksheet that is the Viz in Tooltip, click **Hide**. To show the Viz in Tooltip worksheet again, in the source worksheet, click **Unhide All Sheets**.
- Filter hierarchical data in views by using the **All Values in Hierarchy** filter option.
- Change the Line mark type to a step line or a jump line by clicking the Path property in the Marks card.
- Double-click an axis to edit axis settings.

Collaborate with Tableau Online and Tableau Server

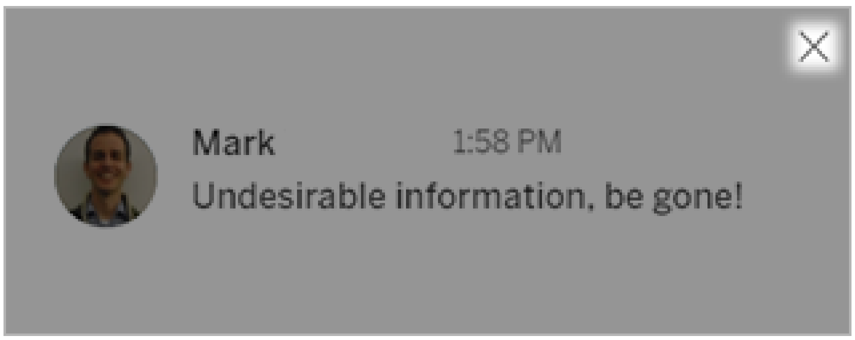
Quickly analyze cloud-based data with Dashboard Starters

If you use Tableau Online, Dashboard Starters help you quickly author and analyze data from cloud-based systems like Salesforce, ServiceNow, Marketo, and Eloqua. Simply create a new workbook and choose from several beautiful, informative designs that are tailor-made for key business metrics. For more information, see **Dashboard Starters for Cloud-based Data** on page 2750.



Streamline discussion threads by deleting comments

If a comment is unnecessary or inaccurate, you can quickly delete it. Just click the **X** in the upper-right corner. For more information, see **Comment on Views** on page 3259.



Version 10.5

Added in version 10.5

Install and Deploy Tableau

- **Activate Tableau Desktop from the command line** on page 159
- **Turn off driver downloads from the command line (Windows)** on page 159

Connect to and Prepare Data

Design Views and Analyze Data

Publish Data Sources and Workbooks

New in Web Authoring

Collaborate with Tableau Online and Tableau Server

- **Subscribe users in groups, and add custom messages** on page 166
- **Rename workbooks on Tableau Online or Tableau Server** on page 166

Install and Deploy Tableau

Activate Tableau Desktop from the command line

When you install Tableau Desktop from the command line you can now activate the product without having to complete the activation process through the user interface. Instead add a single command line that includes the installer option `ACTIVATE_KEY "<key>"`. During the installation process, the installer will run `-activate` and apply the product key you provide to activate the product automatically.

For more information about installing Tableau from the command line, see Deploy Tableau Desktop.

Turn off driver downloads from the command line (Windows)

When you install Tableau Desktop the most common drivers are automatically installed for you. To prevent all or some of these drivers from installing automatically, you can turn off that option during install.

- To prevent all drivers from being automatically installed, do one of the following:
  - During install, from the Installer screen, click **Customize**. Then clear the **Install database drivers: Microsoft SQL Server, PostreSQL and Amazon Redshift** check box.
  - From the command line, set the `DATABASEDRIVERS` = "0"
- To prevent one or more of these drivers from being automatically installed, do the following:
  - Leave the **Install database drivers: Microsoft SQL Server, PostreSQL and Amazon Redshift** check box selected, or from the command line, set the `DATABASEDRIVERS` = "1"

- Add a command line and set `DRIVER_MSSQL`, `DRIVER_POSTGRES` or `DRIVER_REDSHIFT` to "0" to disable an individual driver.

For more information about installing Tableau from the command line, see Deploy Tableau Desktop.

Desktop operating system requirements (64-bit replaces 32-bit)

Starting with version 10.5, Tableau Desktop, Tableau Reader, and Tableau Public (desktop) run only on 64-bit operating systems. Version 10.4 is the last version of Tableau Desktop, Tableau Reader, and Tableau Public to support 32-bit Windows operating systems.

Connect to and Prepare Data

Extracts use the new .hyper format

Beginning with version 10.5, when you create an extract, it uses the new .hyper format. Extracts in this new format take advantage of the improved data engine, which supports the same analytical and query performance as the data engine before it, but for even larger extracts. Similarly, when you perform an extract task on a .tde extract using version 10.5, the extract is upgraded to a .hyper extract. For more information, see **Extract Upgrade to .hyper Format** on page 1031.

Changes to the way values are computed

To improve data source efficiency and scalability, and produce results that are consistent with commercial databases, values in your data source can be computed differently in version 10.5. In some cases, these changes can cause differences with the values and marks in your view between version 10.4 (and earlier) and version 10.5 (and later). This applies to extract data sources, and can apply to multi-connection data sources, data sources that use live connections to file-based data, data sources that connect to Google Sheets data, cloud-based data sources, extract-only data sources, and WDC data sources. For more information, see **Changes to values and marks in the view** on page 1015.

Google authentication method update

Starting in October 2017, Google's updated security requirements changed the connection workflow for some Tableau connectors. Now Tableau opens at tab in your external default browser when connecting to data that uses OAuth authentication. The connectors affected are

Google Sheets on page 523, **Google BigQuery** on page 512, **Google Analytics** on page 508, **Dropbox** on page 497, **OneDrive** on page 573, and **Box** on page 483.

New Box connector

Use the Box connector to connect to Box data. For more information, see **Box** on page 483.

Support for SSL authentication to IBM DB2 and BigInsights databases

You can now connect to IBM DB2 and IBM BigInsights databases using SSL authentication. For more information, see **IBM DB2** on page 533 and **IBM BigInsights** on page 531.

Use the Extract API

You can use the Extract API 2.0 to create .hyper extracts. For tasks that you previously performed using the Tableau SDK, such as publishing extracts, you can use the Tableau Server REST API or the Tableau Server Client (Python) library. For refresh tasks, you can also use the Tableau Server REST API. For more information, see **Tableau Hyper API** on page 1059.

SAP HANA support for failover

Tableau supports connecting to a server that’s configured for failover. For more information, see **SAP HANA** on page 609.

SAP Sybase ASE support for SSO

You can now connect to SAP Sybase ASE data using single sign-on (SSO). For more information, see **SAP Sybase ASE** on page 629.

Support for OData V4

Tableau now supports OData version 4.0. For more information, see **OData** on page 571.

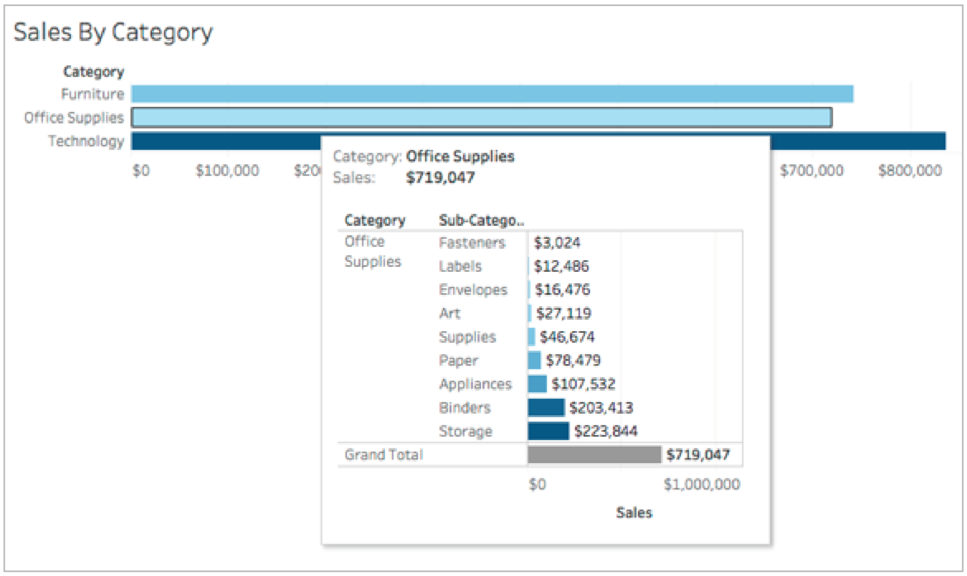
Oracle Eloqua connector update

The Oracle Eloqua connector now supports incremental refreshes, validation for tables with more than 250 fields, and support for additional tables. For more information, see **Oracle Eloqua** on page 583.

Design Views and Analyze Data

Show visualizations in tooltips

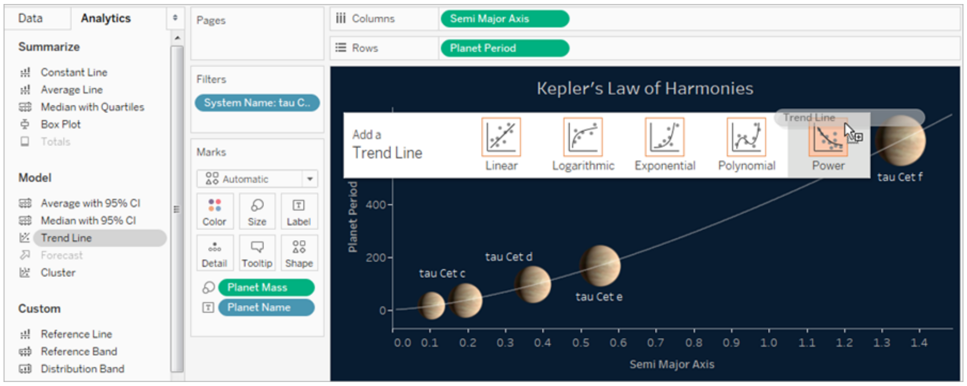
As you craft views and look for ways to reveal more details about data to your audience, you can embed visualizations within tooltips—aka "Viz in Tooltip."



When you show related views in tooltips, you can help your audience engage with the data at a deeper level, while maximizing the space available for the current view. For details, see **Create Views in Tooltips (Viz in Tooltip)** on page 1475.

Power trend lines

When you add trend lines to a view, a Power option is now available. For related details, see **Add Trend Lines to a Visualization** on page 2528 and the "Power" model type in **Trend Line Model Types** on page 2535.



Map data updates and additions

In Tableau 10.5, additional cities are available for South East Asia, Central America, and South America.

Demographic data updates including 2018 to 2023 statistical projections are now available.

U.S. Core Based Statistical Areas (CBSA) and Metropolitan Statistical Areas

(MSA) have been updated to 2015 definitions.

Updates are now available for U.S. postcodes.

Additionally, the following updates are now available for China:

- The areas of Beijing, Chongqing, Shanghai and Tianjin no longer contain city-level districts in the Administrative level 2/County geographic role.
- Administrative level 2 features in China (prefecture-level cities, prefectures, leagues, autonomous prefectures) are considered cities in Tableau. The center points of these areas are included in the City geographic role as well.
- Additional cities such as Tiemenguan are now included in the City geographic role.
- Names for Administrative level 1 and 2 features are now consistent with names published by the People's Republic of China's National Bureau of Statistics.

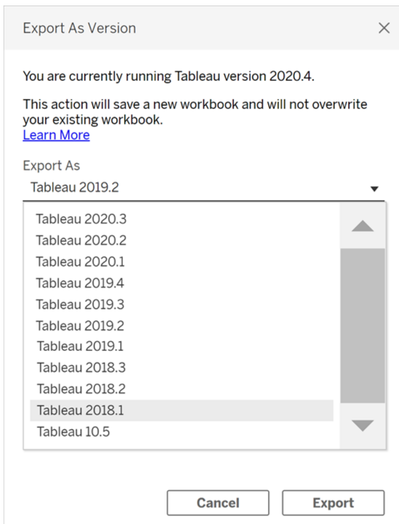
Extend dashboard dimensions to 10,000 pixels

To optimize designs for very large screens or vertical mobile layouts, you can now extend dashboard width and height to 10,000 pixels.

Publish Data Sources and Workbooks

Downgrade workbooks in Tableau Desktop

If you upgrade a workbook in Tableau Desktop to a newer version, you can now downgrade it to a previous version so that users who are using Tableau Desktop 10.2 and later can still open the workbook. To downgrade the workbook, in Tableau Desktop select **File** > **Export As** and select the version that you want to downgrade to. Then save it to your My Tableau Repository or a selected location.



The downgraded workbook can be opened in the version selected and later. Any features not available in that previous version will be removed when the workbook is downgraded.

> **Note**: This feature is not available for workbooks that include a Tableau Data Extract that has been refreshed in 10.5. The extract will have been converted to use the new .hyper format and this format can't be downgraded back to a .tde format.

New in Web Authoring

Edit axes

You can now edit axes on the web. To open the Edit Axis dialog box on the web, double-click an axis in the view. Options available from the Edit Axis dialog box include **Synchronize dual**

axes, clearing the axis range (**Reset**), and editing tick marks. You can also enable or disable **Dual axis** in a field context menu (right-click a measure field on Rows or Columns shelf).

Additional enhancements

- Add text objects to dashboards and edit them.
- Edit worksheet and dashboard titles. Double-click the title to open the **Edit Title** dialog box.
- Edit trend lines and view a description of the trend model. To view the trend model, hover over any part of the trend line. To edit a trend line on the web, click a trend line and hold the cursor in place, and then click the **Edit** menu.
- Trend lines now include a Power option.
- Create hierarchies by dragging one dimension onto another in the Data pane.
- Set the default color property for a field.
- For quantitative color legends, advanced settings are available (dates not included).

Collaborate with Tableau Online and Tableau Server

Upload custom images for user profiles

By default, users and groups now have profile images that reflect their first two initials. To help you visually distinguish between profile types, user images are circular, while group images are square.

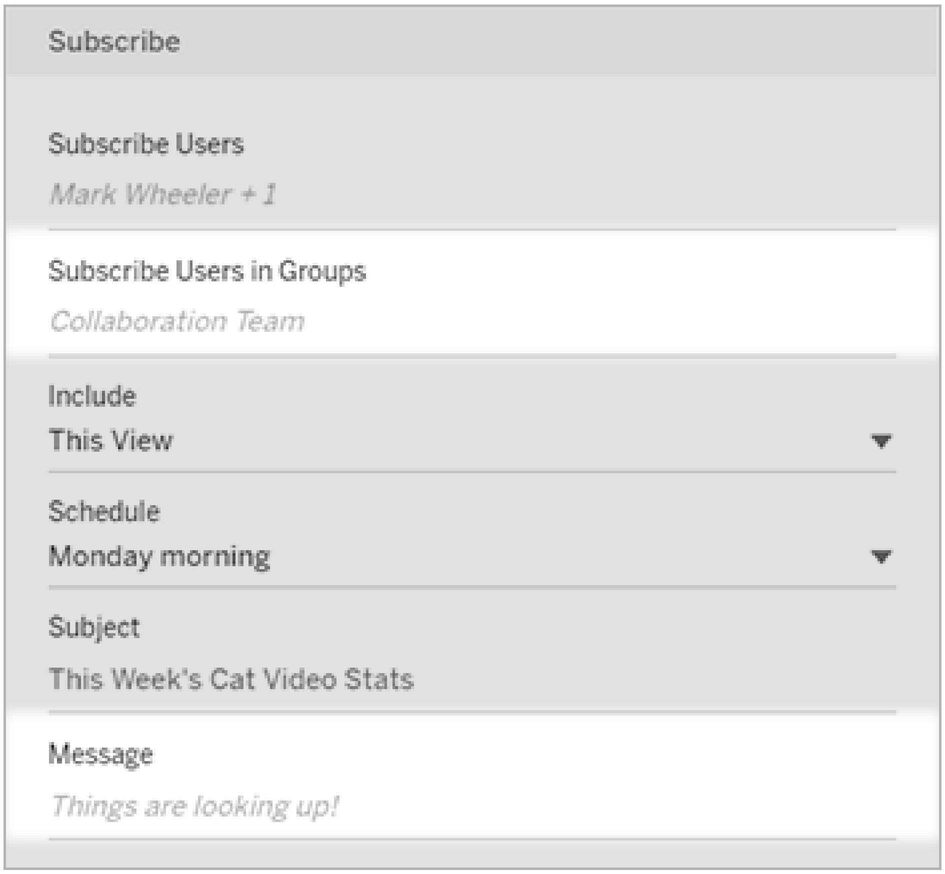
User profiles support custom images. To upload one, go to your account settings page, and click the current image or initials next to your name.

If your organization uses Active Directory, default profile images come from that system. But two-letter images may briefly appear until Active Directory next synchronizes with your Tableau server.

Subscribe users in groups, and add custom messages

To quickly subscribe several users, you can select entire groups created by your Tableau administrator. Each user in a group is subscribed independently, so the list of recipients remains fully customizable.

You can also add custom messages to subscription emails, clarifying their contents.

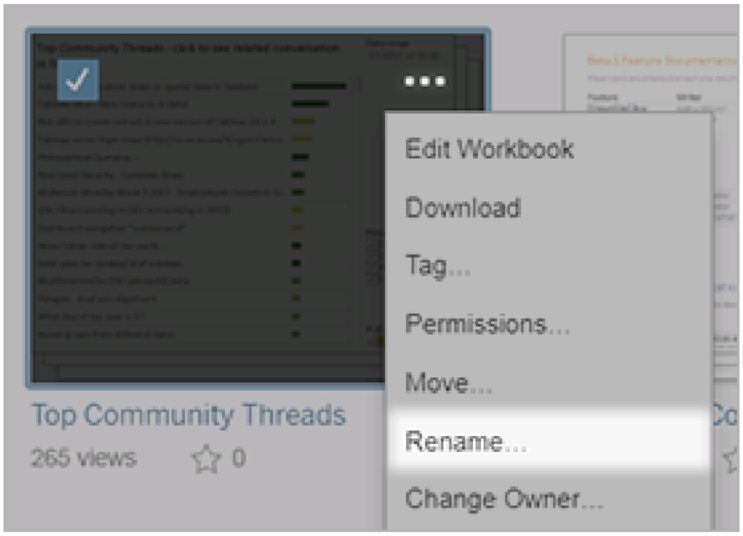


For more information, see **Create a Subscription to a View or Workbook** on page 3249.

Rename workbooks on Tableau Online or Tableau Server

There's no need to republish workbooks just to rename them. Now you can rename workbooks directly on Tableau Online or Tableau Server. When you do, workbook URLs remain the same, so any links you've shared keep working.

Simply navigate to the main **Workbooks** area, click the pop-up menu in the upper-right corner of a workbook, and select **Rename**.



To rename a workbook, users need the Save permission. You'll have that by default for workbooks you own, but you might need to add it for other users. For details, see Set Permissions and Ownership in Help for Tableau site administrators.

Get Started

This section gives you the basics on getting started with building views in Tableau, the Tableau workspace, and Tableau concepts.

For related topics and sites, also see:

- Short walkthrough of Tableau Desktop: **Build a Basic View to Explore Your Data** below
- In-depth Tableau Desktop tutorial: Get Started with Tableau Desktop
- On-demand webinars: Secrets of Visual Analytics
- Build basic chart types: **Build Common Chart Types in Data Views** on page 1577 and **Choose the Right Chart Type for Your Data** on page 371
- Build maps: **Maps and Geographic Data Analysis in Tableau** on page 1755
- Get started with using Tableau Online and Tableau Server: **Use Tableau on the Web** on page 3101.

Watch a video: To see related concepts demonstrated in Tableau, watch these free video tutorials: Getting Started (25 minutes), Getting Started with Data (6 minutes), and Getting Started with Visual Analytics (6 minutes). Use your tableau.com account to sign in. To view other introductory videos, go to Free Training Videos on the Tableau website. Also see different chart types under the How To category on the Free Training Videos page.

Build a Basic View to Explore Your Data

This topic uses the Sample - Superstore data source to walk through how to create basic views and explore your data. It shows how your view of data in Tableau evolves through your process of exploration.

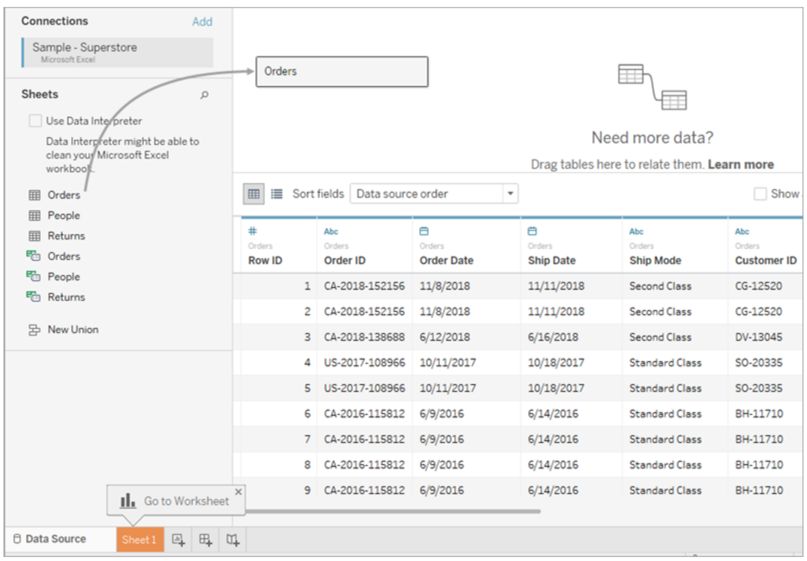
If you are using Tableau Online and Tableau Server to explore data and edit views, see **Use Tableau on the Web** on page 3101.

Connect to your data

The first step is to connect to the data you want to explore. This example shows how to connect to Sample - Superstore data in Tableau Desktop.

1. Open Tableau. On the start page, under **Connect**, click Microsoft Excel. In the Open dialog box, navigate to the `Sample - Superstore` Excel file on your computer. Go to `/Documents/My Tableau Repository/Datasources/`*`version number`*`/[`*`language`*`]`. Select Sample - Superstore, and then click **Open**.

2. After you connect to the Excel data, the *data source page* shows the sheets or tables in your data. Drag the "Orders" table to the *canvas* to start exploring that data.

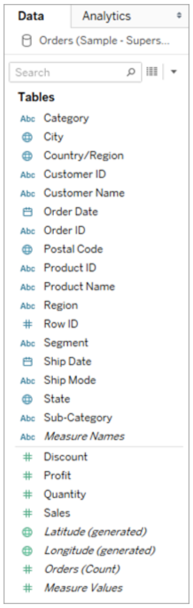


> Depending on how your data is structured, you might need to do more data preparation and integration before you start exploring it in Tableau. For more details on connecting to your data, see **Connect to and Prepare Data** on page 391 and **Tips for Working with Your Data** on page 398.

3. Click the sheet tab to go to the new worksheet and begin your analysis.

About the Data pane

In the worksheet, the columns from your data source are shown as fields on the left side in the **Data** pane. The **Data** pane contains a variety of fields organized by table. For each table or folder in a data source, dimension fields appear above the gray line and measure fields appear below the gray line. Dimension fields typically hold categorical data such as product types and dates, while measure fields hold numeric data such as sales and profit. In some cases, a table or folder might contain only dimensions, or only measures to start with. For more information, see **Dimensions and Measures, Blue and Green** on page 314.



For more information about parts of the workspace, see **The Tableau Workspace** on page 192.

If you have more than one data source in a workbook, click the data source connection name in the Data pane to select it for use. For more details, see **Navigating Data Sources in the Data Pane** on page 211.

> For details on the many ways you can customize the fields in the data pane, see **Organize and Customize Fields in the Data Pane** on page 1146, **Edit Default Settings for Fields** on page 1157, and **Work with Data Fields in the Data Pane** on page 203.

If you have related dimension fields, sometimes you might want to group them in a folder, or as a hierarchy. For example, in this data source, Country, State, City, and Postal Code are grouped into a hierarchy named Location. You can drill down into a hierarchy by clicking the + sign in a field, or drill back up by clicking the - sign in a field.

Build the view

A **view** is a visualization or viz that you create in Tableau. A viz might be a chart, a graph, a map, a plot, or even a text table.

Every view that you build in Tableau should start with a question. What do you want to know?

Every time you drag a field into the view or onto a shelf, you are asking a question about the data. The question will vary depending on where you drag various fields, the types of field, and the order in which you drag fields into the view.

For every question you ask, the view changes to represent the answer visually - with marks (shapes, text, hierarchies, table structures, axes, color).

Different ways to start building a view

When you build a view, you add fields from the **Data** pane. You can do this in different ways.

For example:

- Drag fields from the **Data** pane and drop them onto the cards and shelves that are part of every Tableau worksheet.
- Double-click one or more fields in the **Data** pane.
- Select one or more fields in the **Data** pane and then choose a chart type from **Show Me**, which identifies the chart types that are appropriate for the fields you selected. For details, see **Use Show Me to Start a View** on page 1323.
- Drop a field on the **Drop field here** grid, to start creating a view from a tabular perspective.

For more details on building visualizations with fields, see **Start Building a Visualization by Dragging Fields to the View** on page 1285.

As you start exploring data in Tableau, you will find there are many ways to build a view. Tableau is extremely flexible, and also very forgiving. As you build a view, if you ever take a path that isn't answering your question, you can always undo to a previous point in your exploration.

- To undo or redo, click undo ← or redo → on the toolbar.

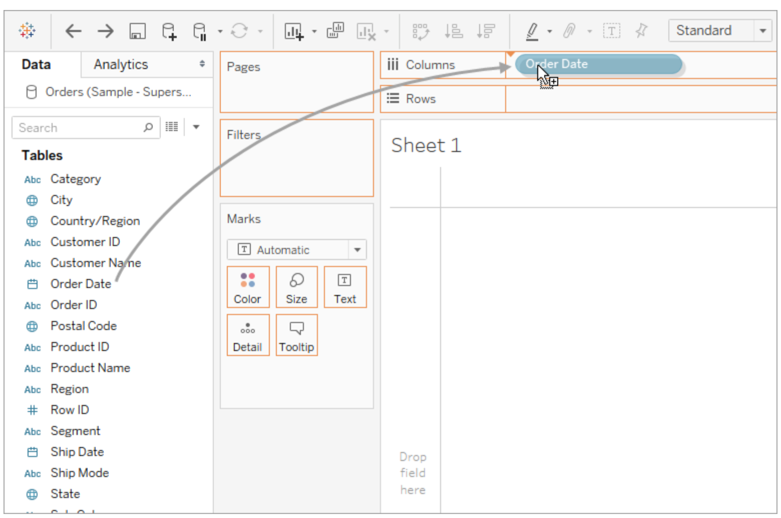
You can undo all the way back to the last time you opened the workbook. You can undo or redo an unlimited number of times.

Build a view from scratch

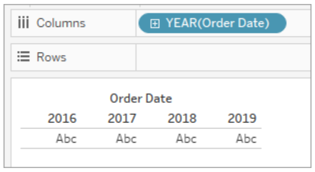
These steps show how to build a basic view that shows year-by-year profit.

1. From the **Dimensions** area in the **Data** pane, drag the **Order Date** field to the **Columns** shelf. You might need to expand the Order hierarchy to see Order Date.

   When you drag a field over a shelf, a plus sign indicates that the shelf can accept the field.

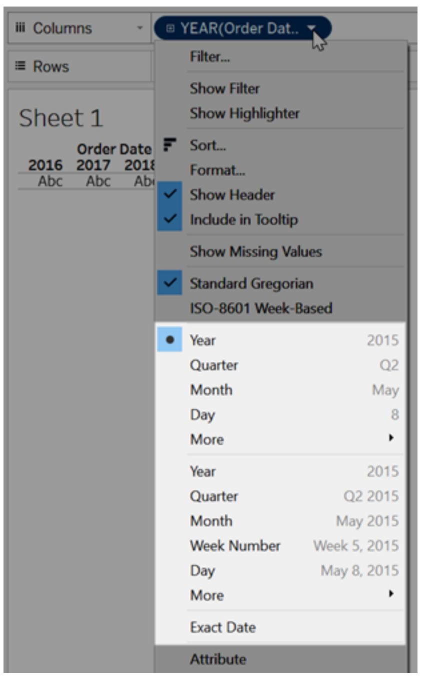
   

   The resulting table has four columns and one row. Each column header represents a member of the Order Date field (the default date level is YEAR). Each cell contains an "Abc" label, which indicates that the current mark type for this view is text.



Notice that the field is colored blue, which indicates that it is *discrete*. For more information, see **Dimensions and Measures, Blue and Green** on page 314.

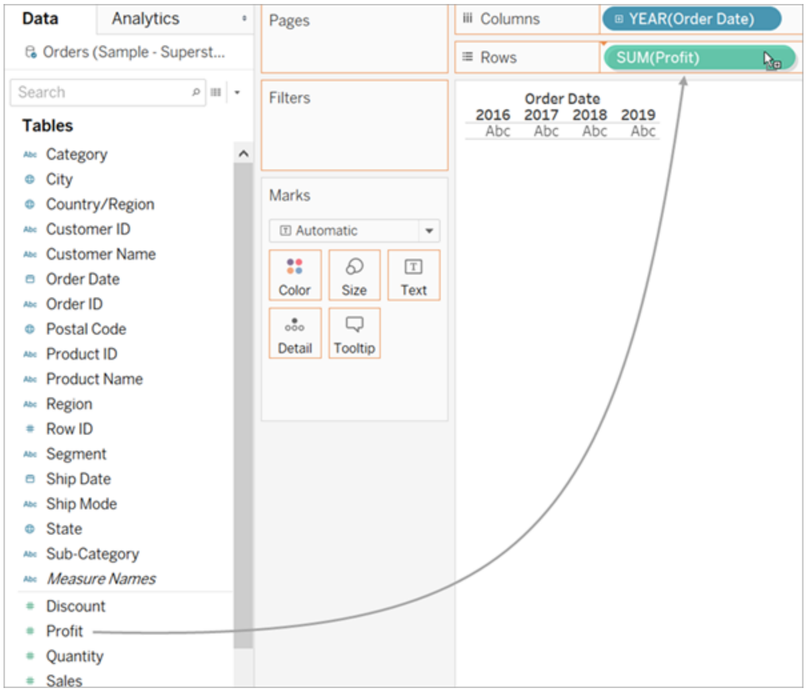
The default date level is determined by the highest level that contains more than one distinct value (for example, multiple years and multiple months). That means that if **[Order Date]** contained data for only one year but had multiple months, the default level would be month. You can change the date level using the field menu.



If you're wondering why there are two sets of date levels (from Year down to Day), the first set of options uses date parts and the second set of options uses date values. For more information, see **Change Date Levels** on page 1248.

Tip: To hide the title on the sheet, select the drop-down to the right of the title (Sheet 1), and select **Hide Title**.

2. From the **Data** pane, drag the **Profit** field to the **Rows** shelf.

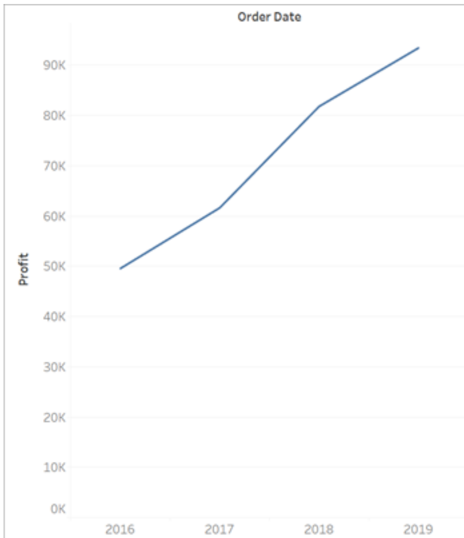


The **Profit** field is colored green on the **Rows** shelf, indicating that it is continuous. Also, the field name changes to **SUM(Profit)**. That's because measures are automatically aggregated when you add them to the view, and the default aggregation for this measure is SUM. For more information on what happens when you add a measure to a view—and why—see **Dimensions and Measures, Blue and Green** on page 314.

Tableau transforms the table into a line chart and creates a vertical axis (along the left side) for the measure.

A line chart is a great way to compare data over time and identify trends effectively.

This line chart shows profit over time. Each point along the line shows the sum of profit for the corresponding year.



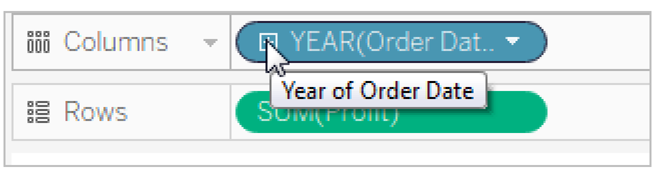
The next step is to drill a little deeper into the time dimension.

Drill into the data

This step shows how you can modify the view to show quarters in addition to years. As you drill down into the hierarchy, the view changes to become a nested table.

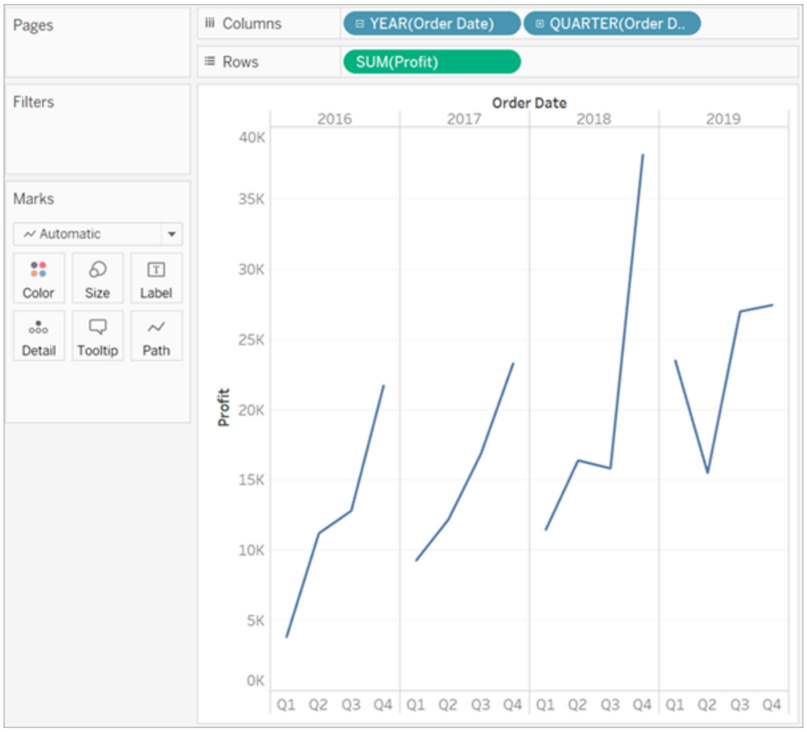
You can show **[Order Date]** by quarters using either of the following methods:

- Click the plus button on the left side of the field **YEAR(Order Date)** field on Columns.



- Drag the **Order Date** field (again) from the Data pane and drop it on the **Columns** shelf to the right of **Year(Order Date)**.

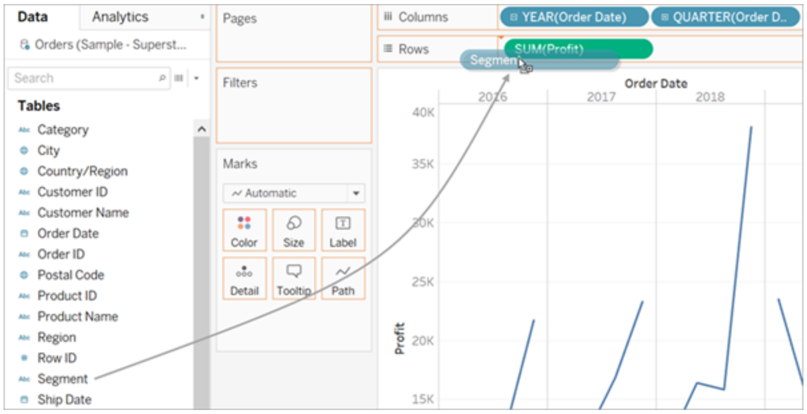
The new dimension divides the view into separate panes for each year. Each pane, in turn, has columns for quarters. This view is called a nested table because it displays multiple headers, with quarters nested within years. The word "headers" might be a bit misleading because while the year headers remain at the "head" of the view (that is, the top), the quarter headers are at the foot (that is, the bottom). So it might be more accurate to call them *footers*.



Increase the level of detail with small multiples

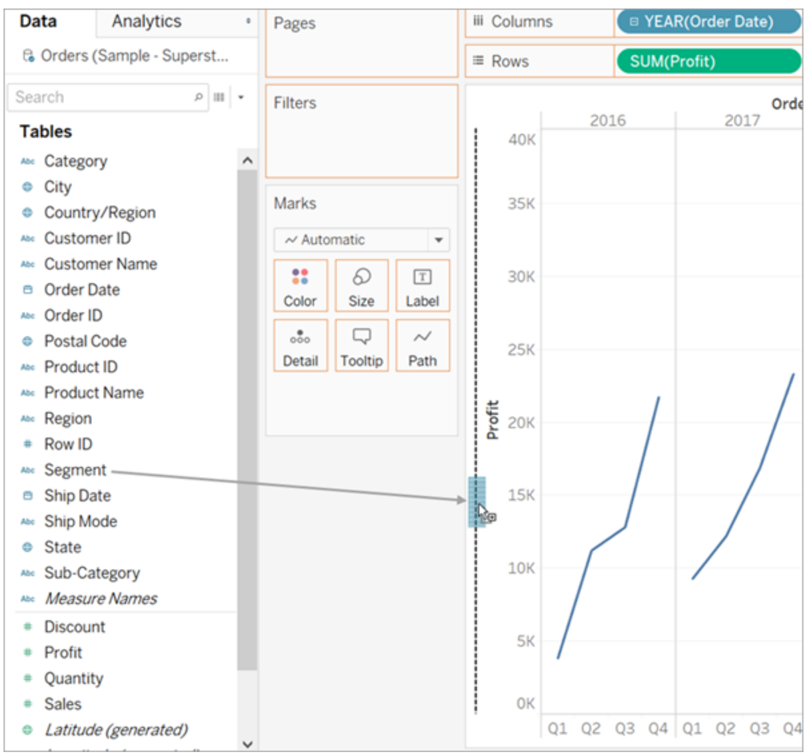
This step shows how you can modify the nested table view to add customer segment. This will create a small multiples view.

Drag the **Segment** dimension from the **Data** pane. Drop it just to the left of **SUM(Profit)** on Rows.



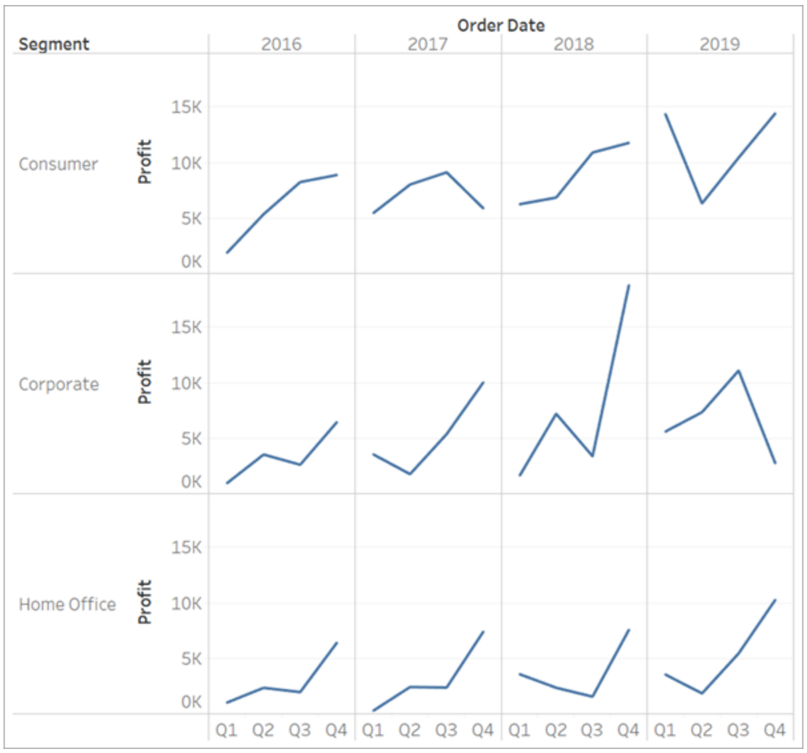
The field is added to the Rows shelf and row headers are created. Each header represents a member of the Segment field.

You could achieve the same result by dropping **Segment** just to the left of the Profit axis in the view (show below). Tableau often supports multiple ways to add fields to the view.



Note: Tableau does not allow you to place a dimension to the right of a measure on either the Rows or Columns shelves because that visual structure would not make sense in the view.

The new dimension divides the view into 12 panes, one for each combination of year and segment. This view is a more complex example of a nested table. Any view that contains this sort of grid of individual charts is referred to as a small multiples view.

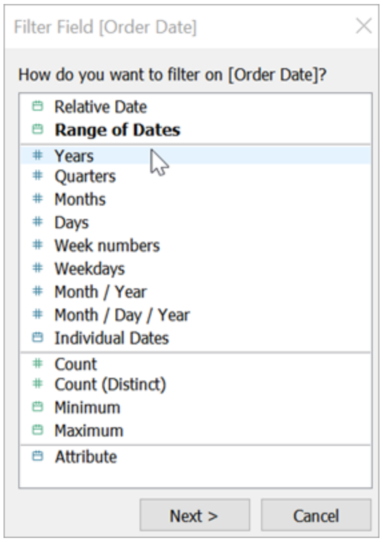


At this point you probably don't want to make the view any more granular; in fact, you may want to simplify the view by removing (that is, filtering) some data.

Filter the view to focus your exploration

This section shows how you can focus your exploration by only showing a subset of the data, such as modifying the view to show only data for orders in 2012 and 2013.

1. Drag the **Order Date** measure from the **Data** pane and drop it on the **Filters** shelf.



2. In the Filter Field dialog box, choose the date level you want to filter on—**Years**. Then click **Next**.
3. In the next pane, clear any two years that you do not want to include in the view.
4. When you are finished, click **OK**.

The view updates to only show data rows where Order Date is 2018 or 2019. Tableau can now allocate more space to the data that interests you.

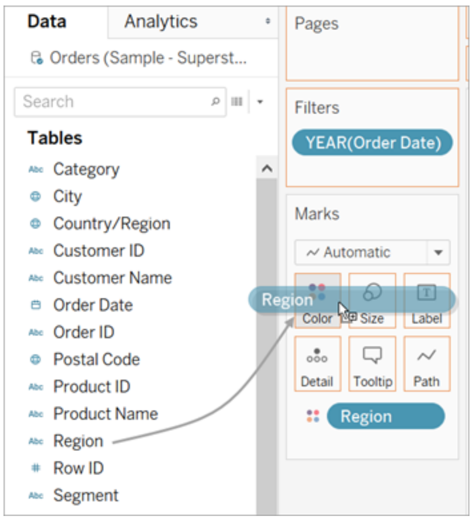
Next, you will increase the detail in your data exploration by dragging a field to Color on the Marks card.

Use the Marks card to add depth to your analysis

This step shows how you can modify the view to color the marks based on the region.

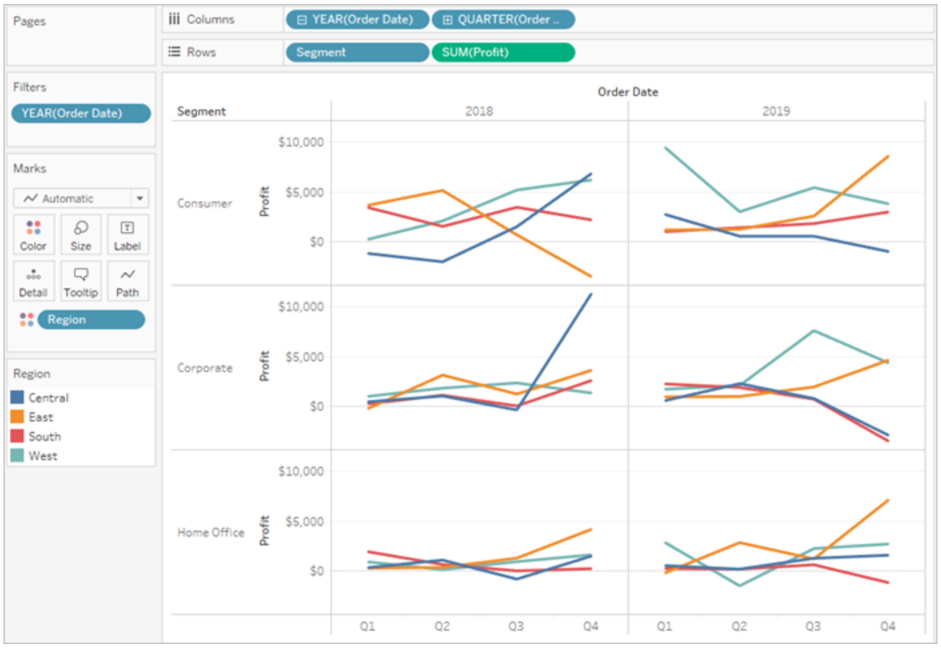
1. Drag the **Region** dimension from the **Data** pane and drop it on **Color**.

   Placing a dimension on **Color** separates the marks according to the members in the dimension, and assigns a unique color to each member. The color legend displays each member name and its associated color.

   Each pane now has four lines, one for each region. The viz is now showing profit data summarized the Region level of detail.

   This view now shows profit for each customer segment and region for 2012 and 2013.



For more information on the Marks card and level of detail, see **Shelves and Cards Reference** on page 222, **Marks** on page 249, and **How dimensions affect the level of detail in the view** on page 319.

More resources

For related topics and sites, also see:

- In-depth Tableau Desktop tutorial: Get Started with Tableau Desktop
- On-demand webinars: Secrets of Visual Analytics
- Build basic chart types: **Build Common Chart Types in Data Views** on page 1577 and **Choose the Right Chart Type for Your Data** on page 371
- Build maps: **Maps and Geographic Data Analysis in Tableau** on page 1755
- Use Tableau Online and Tableau Server to explore data and edit views: **Use Tableau on the Web** on page 3101

Watch a video: To see related concepts demonstrated in Tableau, watch these free video tutorials: Getting Started (25 minutes), Getting Started with Data (6 minutes), and Getting Started with Visual Analytics (6 minutes). Use your tableau.com account to sign in. To view other introductory videos, go to Free Training Videos on the Tableau website. Also see different chart types under the How To category on the Free Training Videos page.

Tour the Tableau Environment

This section provides information on Tableau pages and its workspace, including panes, shelves, icons, and other elements in Tableau Desktop.

- **The Tableau Workspace** on page 192
- **Start Page** below
- **Data Source Page** on page 188

If you are using Tableau on the web, see **Creators: Get Started with Web Authoring** on page 3119 and **Tour Your Tableau Site** on page 3104.

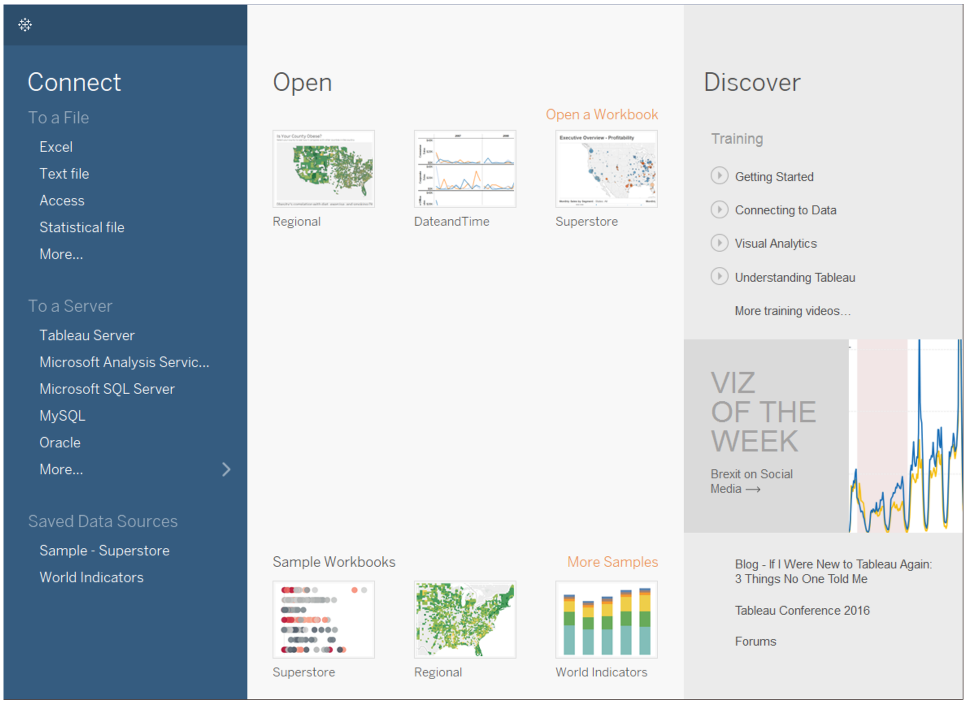
Watch a video: To see related concepts demonstrated in Tableau, watch Getting Started, a 25-minute free training video. Use your tableau.com account to sign in.

Start Page

The start page in Tableau Desktop is a central location from which you can do the following:

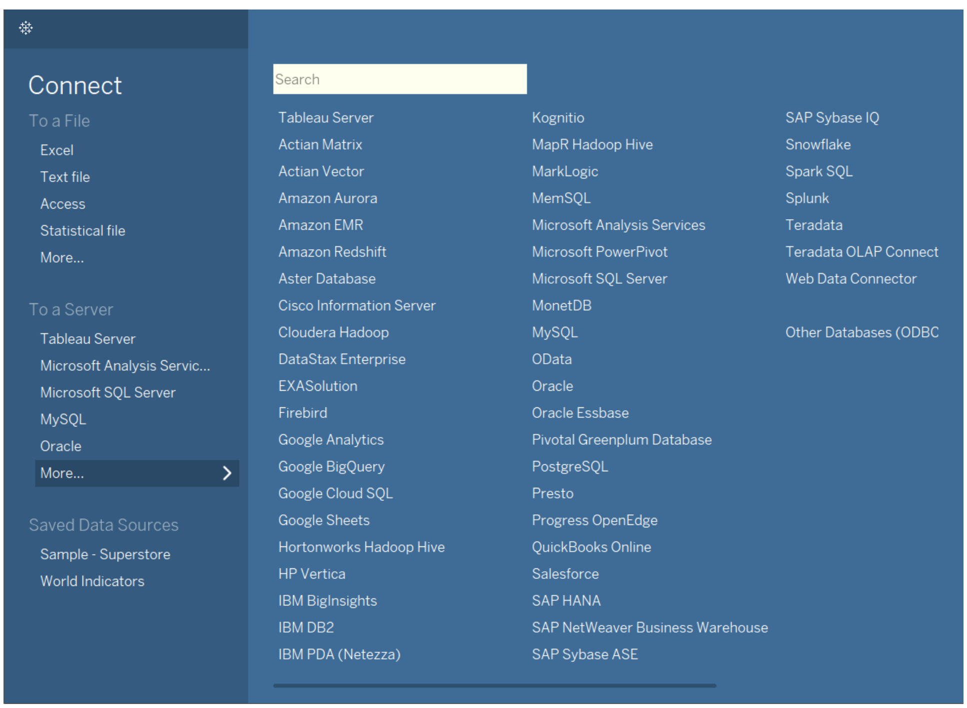
- Connect to your data
- Open your most recently used workbooks, and
- Discover and explore content produced by the Tableau community.

The start page consists of three panes: **Connect**, **Open**, and **Discover**.



Connect

Connect to data and open saved data sources.

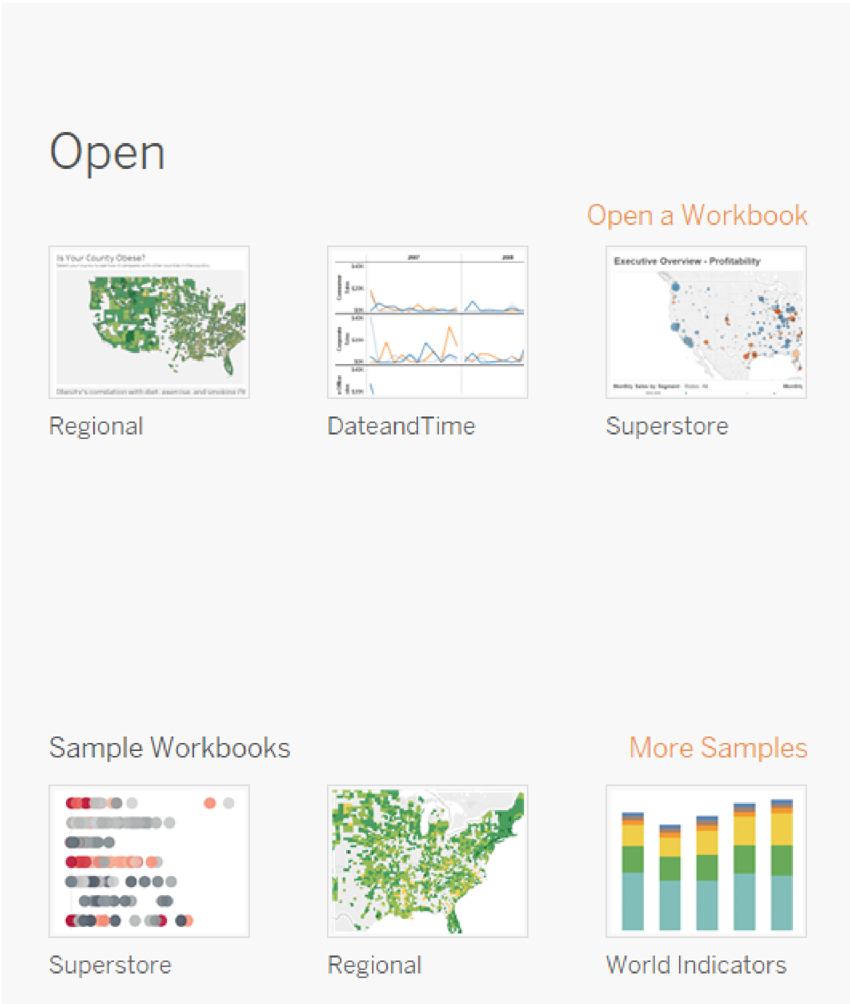


On the **Connect** pane, you can do the following:

- **Connect to data:** Under **To a File**, connect to data stored in Microsoft Excel files, text files, Access files, Tableau extract files, and statistical files, such as SAS, SPSS, and R. Under **To a Server**, connect to data stored in databases like Microsoft SQL Server or Oracle. The server names listed in this section change based on which servers you connect to and how often.
- **Open saved data sources:** Quickly open data sources that you have previously saved to your My Tableau Repository directory. Also, Tableau provides sample saved data sources that you can use to explore Tableau Desktop functionality. To follow along with examples in the Tableau Desktop documentation, you'll usually use the **Sample – Superstore** data source.

Open

Open recent workbooks, pin workbooks to the start page, and explore sample workbooks.



On the **Open** pane, you can do the following:

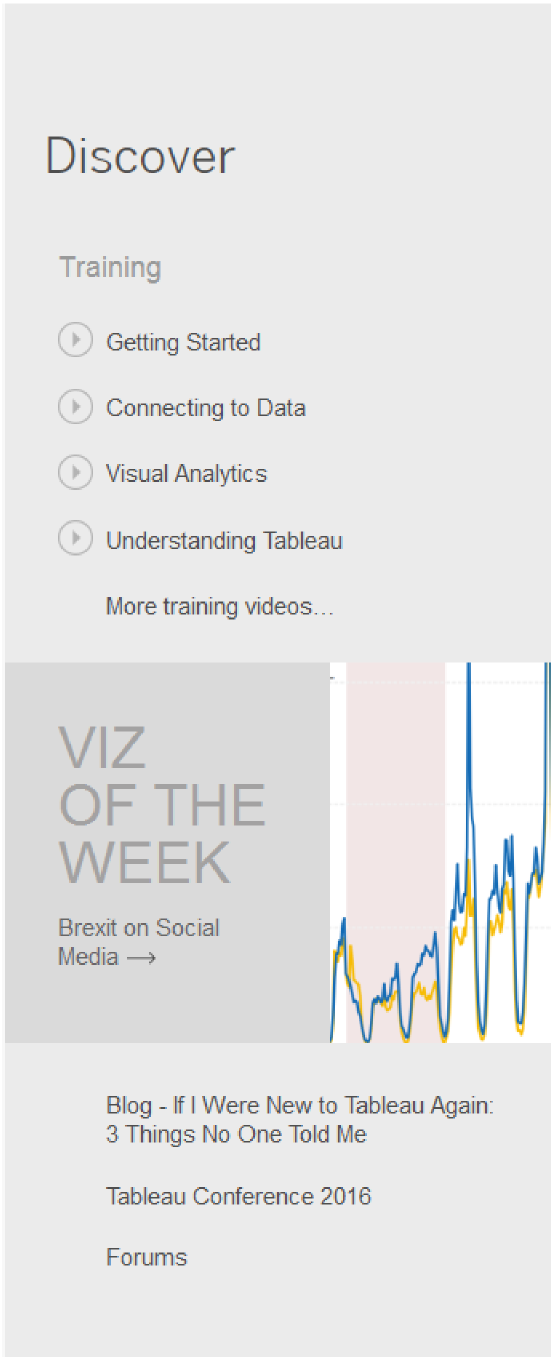
- **Open recently opened workbooks:** When you open Tableau Desktop for the first time, this pane is empty. As you create and save new workbooks, the most recently opened workbooks appear here. Click the workbook thumbnail to open a workbook, or if you don't see a workbook thumbnail, click the **Open a Workbook** link to find other workbooks that are saved to your computer.
- **Pin workbooks:** You can pin workbooks to the start page by clicking the pin icon that appears in the top-left corner of the workbook thumbnail. Pinned workbooks always appear on the start page, even if they weren't opened recently. To remove a recently opened or pinned workbook, hover over the workbook thumbnail, and then click the "x" that appears. The workbook thumbnail is removed immediately but will show again with

your most recently used workbooks the next time you open Tableau Desktop.

- **Explore sample workbooks:** Open and explore sample workbooks.

Discover

See popular views in Tableau Public, read blog posts and news about Tableau, and find training videos and tutorials to help you get started.

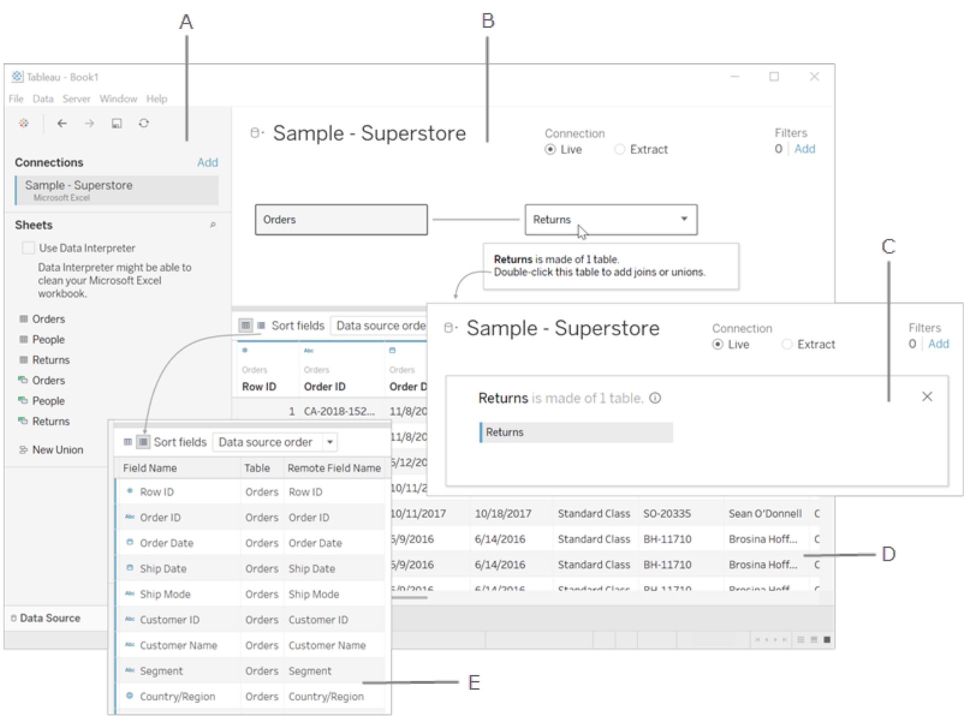


Data Source Page

Before or during your analysis, you may want to make changes to the Tableau data source. You can do that on the data source page. Tableau takes you to the data source page after you establish the initial connection to your data. You can also access the data source page by clicking the **Data Source** tab from any location in the workbook.

> **Watch a video**: To see related concepts on connecting to data and using the Data pane, watch Getting Started with Data, a 6-minute free training video. Use your tableau.com account to sign in.

Although the look of the page and the options available vary depending on the type of data that you are connected to, the data source page generally consists of four main areas: left pane, canvas, data grid, and metadata grid.



A.Left pane on the next page - Displays the connected data source and other details about your data.

B.Canvas on the next page: logical layer - The canvas opens with the logical layer, where you can create relationships between logical tables.

C.Canvas on the next page: physical layer - Double-click a table in the logical layer to go to the physical layer of the canvas, where you can add joins and unions between tables.

D.Data grid on the facing page - Displays first 1,000 rows of the data contained in the Tableau data source.

E.Metadata grid on page 192 - Displays the fields in your data source as rows.

Back to top

Left pane

The left pane of the data source page displays details about your data—the data that Tableau Desktop is connected to.

- For file-based data, the left pane might display the file name and the worksheets in the file.
- For relational-based data, the left pane might display the server, the database or schema, and the tables in the database.

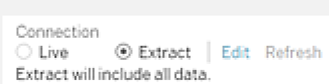
You can also use the left pane to add more connections to the data source to create cross-database joins.

The left pane does not display for cube (multidimensional) data.

Canvas

When connected to most relational and file-based data, you can drag one or more tables to the canvas area to set up your Tableau data source. The canvas now has two layers: a logical layer and a physical layer. The default view shows the logical layer, which uses relationships between logical tables. To go to the physical layer, double-click a logical table in the logical view and create joins and unions between tables in the physical layer. For more information, see Tableau Data Model.

When connected to cube data, the top of the data source page displays the available catalog or queries and cubes to select from to set up your Tableau data source.

> **Note**: Depending on the type of data you are connected to, you can refresh your data when that data has changed using a number of options. For example if connected to a data extract, you can refresh a selected extract using the **Refresh** button
>
> 
>
> that appears next to the Connection area in the left side of

the pane. For more information about refresh options see **Refresh Data Sources** on page 1071. For more information about refreshing data extracts, see **Refresh Extracts** on page 1041.

Refreshing your data source

Depending on the type of data you are connected to you can refresh your data when that data has changed using a number of options. For example if connected to a data extract,

Data grid

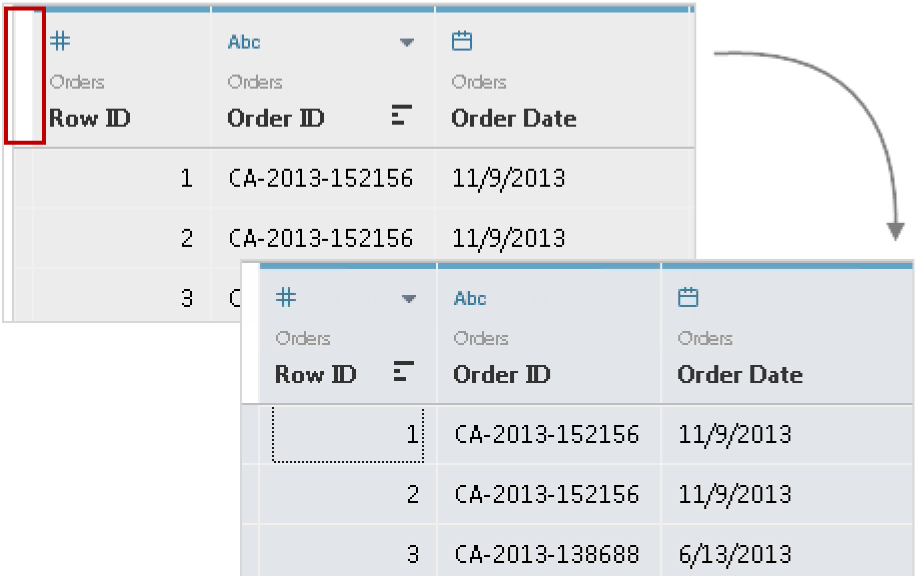
Use the data grid to review the fields and the first 1,000 rows of the data contained in the Tableau data source. You can also use the data grid to make general modifications to the Tableau data source like sorting or hiding fields; renaming fields or resetting field names; creating calculations; changing the column or row sort; or adding aliases. For the Web Data Connector, file- and relational-based data sources in Extract mode, you can see extract data in the grid, including extract only calculations.

When you're in the logical layer, you see the data for the selected table.

In the physical layer, you see the merged data based on the joins and unions.

To select multiple fields in the grid, click a column, and then drag the mouse to select more columns.

To select all fields, click the area in the upper-left corner of the grid like in the example, below.



The grid is not displayed for cube (multidimensional) data.

Metadata grid

Depending on the type of data that you are connected to, click the metadata grid button to navigate to the metadata grid. The metadata grid displays the fields in your data source as rows so that you can quickly examine the structure of your Tableau data source and perform routine management tasks, such as renaming fields or hiding multiple fields at once.

When connected to cube or some extract only data, the metadata grid displays by default.

The Tableau Workspace

The Tableau workspace consists of menus, a toolbar, the Data pane, cards and shelves, and one or more sheets. Sheets can be worksheets, dashboards, or stories. For details on dashboard or story workspaces, see **Create a Dashboard** on page 2741 or **The Story Workspace** on page 2845.

> **Watch a video**: To see related concepts demonstrated in Tableau, watch The Tableau interface in Getting Started, a 4-minute free training video. Use your tableau.com account to sign in.

If you are using Tableau on the web, see **Creators: Get Started with Web Authoring** on page 3119 and **Tour Your Tableau Site** on page 3104.

Workspace area

A. Workbook name. A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story. For more information, see **Workbooks and Sheets** on page 261.

B.Cards and shelves - Drag fields to the cards and shelves in the workspace to add data to your view.

C.Toolbar - Use the toolbar to access commands and analysis and navigation tools.

D.View - This is the canvas in the workspace where you create a visualization (also referred to as a "viz").

E. Click this icon to go to the Start page, where you can connect to data. For more information, see **Start Page** on page 183.

F.Side Bar - In a worksheet, the side bar area contains the Data pane and the Analytics pane.

G. Click this tab to go to the Data Source page and view your data. For more information, see **Data Source Page** on page 188.

H.Status bar - Displays information about the current view.

I. Sheet tabs - Tabs represent each sheet in your workbook. This can include worksheets, dashboards, and stories. For more information, see **Workbooks and Sheets** on page 261.

Back to top

Tableau Toolbar Button Reference

When you are creating or editing a view, you can use the toolbar at the top of the view to perform common actions.

In Tableau Desktop, you can hide or display the Tableau toolbar by selecting **Window** > **Show Toolbar**.

The table below explains the functions of each toolbar button. Some buttons are not available in all Tableau products. Also see **Visual Cues and Icons in Tableau Desktop** on page 277.

Toolbar Button	Description
	**Tableau icon**: Navigates to the start page. For more information, see **Start Page** on page 183. **Note**: Tableau Desktop only.
	**Undo**: Reverses the most recent action in the workbook. You can undo an unlimited number of times, back to the last time you opened the workbook, even after you have saved. For more information, see **Undo and Redo** on page 2725.
	**Redo**: Repeats the last action you reversed with the **Undo** button. You can redo an unlimited number of times.
	**Save**: In Tableau Desktop, saves the changes made to the workbook. For more information, see **Save Your Work** on page 3017. In Tableau Server or Tableau Online, click **File** > **Save** or **File** > **Save As** to save your changes.
	**New Data Source**: In Tableau Desktop, opens the **Connect** pane where you can create a new connection or open a saved connection. For

Toolbar Button	Description
	more information, see **Connect to Your Data** on page 391. In Tableau Server or Tableau Online, opens the **Connect to a Data Source** page, where you can connect to a published data source. For more information, see **Connect to published data sources while web authoring** on page 449.
	**Pause Auto Updates**: Controls whether Tableau updates the view when changes are made. Use the drop-down menu to automatically update the entire sheet or just use filters. For more information, see **Refresh Data or Pause Automatic Updates** on page 3223.
	**Run Update**: Runs a manual query of the data to update the view with changes when automatic updates are turned off. Use the drop-down menu to update the entire worksheet or just use filters. **Note**: Tableau Desktop only.
	**New Worksheet**: Creates a new blank worksheet, use the drop-down menu to create a new worksheet, dashboard, or story. For more information, see **Create new worksheets, dashboards, or stories** on page 262.
	**Duplicate**: Creates a new worksheet containing the same view as the current sheet. For more information, see **Duplicate a sheet** on page 264.
	**Clear**: Clears the current worksheet. Use the drop-down menu to clear specific parts of the view such as filters, formatting, sizing, and axis ranges.
	**Swap**: Moves the fields on the **Rows** shelf to the **Columns** shelf and vice versa. The **Hide Empty Rows** and **Hide Empty Columns** settings are always swapped with this button.

Toolbar Button	Description
	**Sort Ascending**: Applies a sort in ascending order of a selected field based on the measures in the view. For more information, see **Sort Data in a Visualization** on page 1447.
	**Sort Descending**: Applies a sort in descending order of a selected field based on the measures in the view. For more information, see **Sort Data in a Visualization** on page 1447.
	**Totals**: You can compute grand totals and subtotals for the data in a view. Select from the following options: • **Show Column Grand Totals**: Adds a row showing totals for all columns in the view. • **Show Row Grand Totals**: Adds a column showing totals for all rows in the view. • **Row Totals to Left**: Moves rows showing totals to the left of a crosstab or view. • **Column Totals to Top**: Moves columns showing totals to the top of a crosstab or view. • **Add All Subtotals**: Inserts subtotal rows and columns in the view, if you have multiple dimensions in a column or row. • **Remove All Subtotals**: Removes subtotal rows or columns. **Note**: Tableau Server and Tableau Online only. In Tableau Desktop, click **Analysis** > **Totals**. For more information, see **Show Totals in a Visualization** on page 1456.
	**Highlight**: Turn on highlighting for the selected sheet. Use the options on the drop-down menu to define how values are highlighted. For more information, see **Highlight Toolbar Button** on page 1505.
	**Group Members**: Creates a group by combining selected values. When multiple dimensions are selected, use the drop-down menu to specify

Toolbar Button	Description
	whether to group on a specific dimension or across all dimensions. For more information, see **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1184. **Note**: Tableau Desktop only. In Tableau Server and Tableau Online, create groups using the **Group Members** button on the tooltip.
	**Show Mark Labels**: Switches between showing and hiding mark labels for the current sheet. For more information, see **Show, Hide, and Format Mark Labels** on page 1378.
	**Fix Axes**: switches between a locked axis that only shows a specific range and a dynamic axis that adjusts the range based on the minimum and maximum values in the view. For more information, see **Edit Axes** on page 2900. **Note**: Tableau Desktop only.
	**Format Workbook**: Open the **Format Workbook** pane to change how fonts and titles look in every view in a workbook by specifying format settings at the workbook level instead of at the worksheet level. **Note**: Tableau Server and Tableau Online only. In Tableau Desktop, click **Format** > **Workbook**. For more information, see **Format at the Workbook Level** on page 2882.
  	**Fit**: Specifies how the view should be sized within the window. Select Standard, Fit Width, Fit Height, or Entire View. **Note**: This menu is not available in geographic map views. The **Cell Size** commands have different effects depending on the type of visualization. To access the **Cell Size** menu in Tableau Desktop click **Format** > **Cell Size**.

Toolbar Button	Description
	**Show/Hide Cards**: Shows and hides specific cards in a worksheet. Select each card that you want to hide or show on the drop-down menu. In Tableau Server and Tableau Online, you can show and hide cards for the **Title**, **Caption**, **Filter** and **Highlighter** only.
	**Presentation Mode**: Switches between showing and hiding everything except the view (i.e., shelves, toolbar, **Data** pane). For more information, see **Reorganizing the Workspace** on page 273. **Note**: Tableau Desktop only.
	**Download**: Use the options under **Download** to capture parts of your view for use in other applications. • **Image**: Displays the view, dashboard, or story as an image in a new browser tab. • **Data**: Displays the data from the view in a new browser window with two tabs: **Summary**, showing aggregated data for the fields shown in the view, and **Underlying**, showing underlying data for the selected marks in the visualization. If the new window does not open, you may need to disable your browser's popup blocker. • **Crosstab**: Saves the underlying data for the selected marks in the visualization to a CSV (comma-separated values) file which can then be opened in Microsoft Excel. • **PDF**: Opens the current view as a PDF in a new browser window. From there you can save it to a file. If the new window does not open, you may need to disable your browser's pop-up blocker. **Note**: Tableau Server and Tableau Online only.
	**Share Workbook With Others**: Publish your workbook to Tableau Server or Tableau Online. For more information, see **Simple Steps to Publish a Workbook** on page 3056.

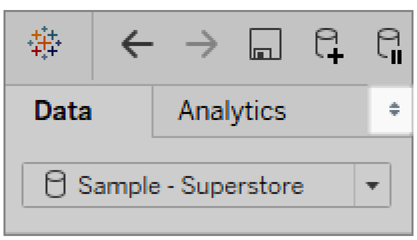
Toolbar Button	Description
	Note: Tableau Desktop only.
	**Show Me**: Helps you choose a view type by highlighting view types that work best with the field types in your data. An orange outline shows around the recommended chart type that is the best match for your data. For more information, see **Use Show Me to Start a View** on page 1323.

Back to top

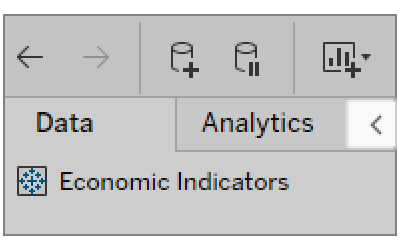
Show and Hide the Side Bar (Data pane)

The Side Bar contains the **Data** pane and the **Analytics** pane when you are editing a worksheet. Different panes are visible depending on what you are doing in the view (Data, Analytics, Story, Dashboard, Layout, Format). The most important thing to know about the Side Bar is that you can expand and collapse this area in the workspace.

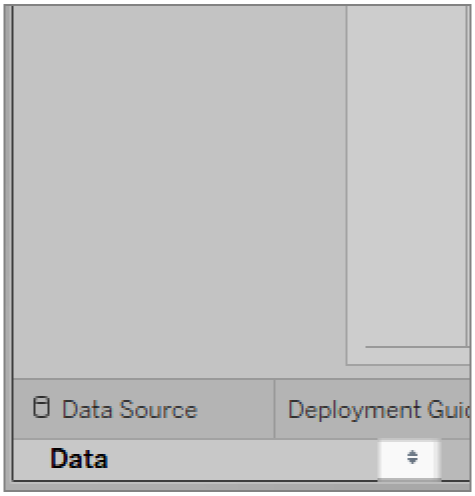
To hide the side bar in Tableau Desktop, click the collapse arrow in the side bar.



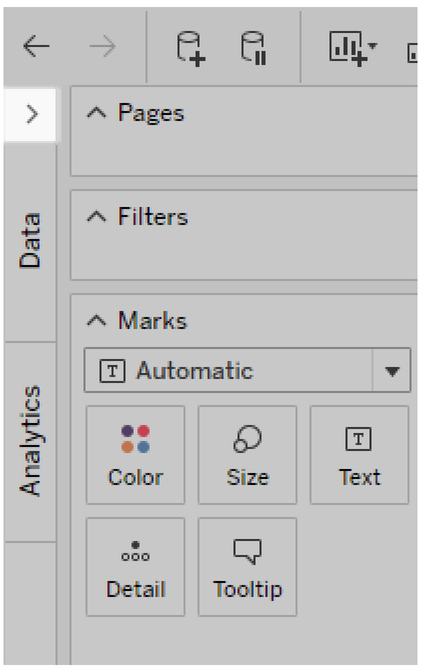
To hide the side bar on the web, click the collapse arrow in the side bar.



To show the side bar on Tableau Desktop, click the expand arrow in the bottom-left of the workspace (in the status bar.



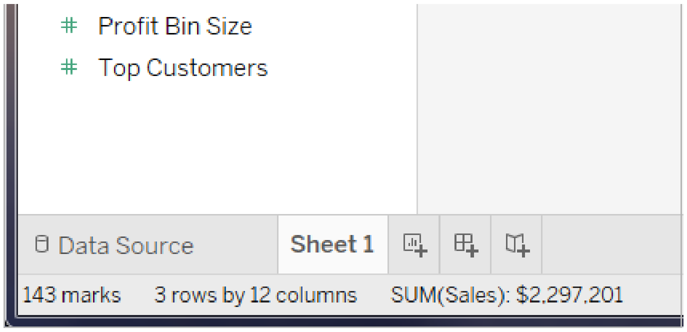
To show the side bar on the web, click the expand arrow in the side bar.



Back to top

Status Bar Information

The status bar is located at the bottom of the Tableau workspace. It displays descriptions of menu items as well as information about the current view. For example, the status bar below shows that the view has 143 marks shown in 3 rows and 12 columns. It also shows that the SUM (Sales) for all the marks in the view is $2,297,201.



You can hide the status bar by selecting **Window** > **Show Status Bar**.

Occasionally, Tableau will display warning icons in the bottom right corner of the status bar to indicate errors or warnings. Below are the possible warning icons and what they mean.

Warning Icon	Description
	**Cancel query indicator**: When you cancel multiple queries, an indicator appears to show you how many queries are still running on the database and using resources. When you cancel a query in Tableau, the database is told to stop processing the query. However, some databases do not support cancel (MS Excel, MS Access, Essbase, Microsoft Analysis Services 2000). If you cancel a query using one of these types of data sources, the query is abandoned by Tableau but is still running in the background and using resources. When you have abandoned queries, an indicator appears in the bottom right corner of the workbook showing the number of queries still running . As queries in the background complete, the number will go down. It is important to monitor the number of queries running and not let the number get too high, otherwise you will see performance degradation of both Tableau and the underlying database. **Note:** Text, Microsoft Excel, and Microsoft Access data sources may be temporarily unavailable after canceling a query because of a lock performed internally. You may have to wait until the

Warning Icon	Description
	abandoned query has completed before re-connecting.
	**Precision warning**: Some fields are more precise in the database than Tableau can model. When you add a field to a view that contains values with more precision than Tableau can model, a warning icon  is displayed in the bottom right corner of the status bar. For example, a value in the database may have 22 decimal places but Tableau only supports up to 15. When you add that field to the view, you get a precision warning. If you click on the warning, you can read more details including the number of decimal places that have been truncated in the view. Remember that the precision of the data displayed in Tableau will always first be dependent on the data in your database. If the values in your database exceed 15 decimal places, when you add them to the view, the value is truncated and a precision warning appears. **Note**: Extracts of relational databases always use the Hyper double precision type, and never use the Hyper numeric type. So if you create an extract of a relational database, you only get 15 digits of precision. You can write to the Hyper API and use the numeric type to get up to 18 digits. For related information, see Numeric Types in Hyper API documentation.
	**Special values indicator**: If your data contains null values, unknown geographic locations, or negative or zero values on a logarithmic axis; the values are shown with an indicator in the lower right corner of the view. Click the indicator for options for handling these values. See **Handling Null and Other Special Values** on page 1317 to learn more about this indicator and how to handle these values.

Back to top

Work with Data Fields in the Data Pane

Tableau displays data source connections and data fields for the workbook in the **Data** pane on the left side of the workspace.

> **Note**: For details on how to start creating visualizations, see **Get Started** on page 169. For details on the many ways you can customize the fields in the Data pane, see **Organize and Customize Fields in the Data Pane** on page 1146 and **Edit Default Settings for Fields** on page 1157.

Areas of the Data pane

After you connect to your data and set up the data source with Tableau, the data source connections and fields appear on the left side of the workbook in the Data pane. For details on connecting to data, see **Connect to and Prepare Data** on page 391.

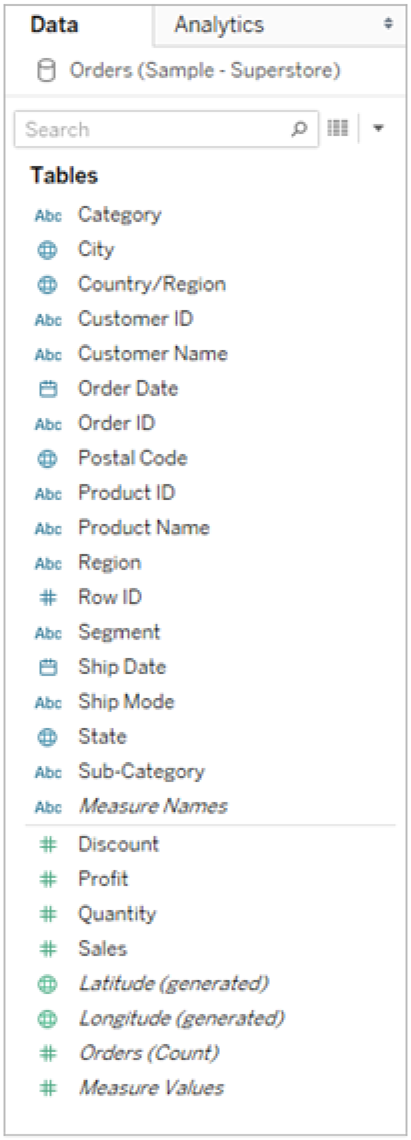
Current data source connections appear at the top of the **Data** pane. When you have more than one connection available, click a connection to select it and start working with that data.

You build visualizations by adding fields from the Data pane to the view. For details, see **Start Building a Visualization by Dragging Fields to the View** on page 1285.

Fields can be organized by table (**Group by Data Source Table**) or folder (**Group by Folder**). Dimensions are displayed above the gray line, and measures below the gray line for each table or folder. In some cases, a table or folder might contain only dimensions, or only measures to start with.

- Calculated fields are listed with their originating field, if all of their input fields come from the same table.
- Sets are listed with the table with their originating field.
- Parameters are global to the workbook and are displayed in the Parameters area.
- Fields that don't belong to a specific table are displayed in the general area below the tables. These include: aggregated calculations, calculations that use fields from multiple tables, Measure Names, and Measure Values.

Below the data source connections in the **Data** pane are the fields that are available in the currently selected data source. You can toggle between the **Data** and **Analytics** panes in a worksheet. For details on the Analytics pane, see **Apply Advanced Analysis to a View (Analytics Pane)** on page 213.



Fields from a single-table data source in the Data pane

The Data pane includes:

- **Dimension fields** – Fields that contain qualitative values (such as names, dates, or geographical data). You can use dimensions to categorize, segment, and reveal the details in your data. Dimensions affect the level of detail in the view. Examples of dimensions include dates, customer names, and customer segments.
- **Measure fields** – Fields that contain numeric, quantitative values can be measured. You can apply calculations to them and aggregate them. When you drag a measure into the view, Tableau applies an aggregation to that measure (by default). Examples of

measures: sales, profit, number of employees, temperature, frequency.

- **Calculated fields** – If your underlying data doesn't include all of the fields you need to answer your questions, you can create new fields in Tableau using calculations and then save them as part of your data source. These fields are called calculated fields.
- **Sets** – Subsets of data that you define. Sets are custom fields based on existing dimensions and criteria that you specify. For more information, see **Create Sets** on page 1186.

  Named sets from an MS Analysis Services server or from a Teradata OLAP connector also appear in Tableau in this area of the Data pane. You can interact with these named sets in the same way you interact with other custom sets in Tableau.
- **Parameters** – Values that can be used as placeholders in formulas, or replace constant values in calculated fields and filters. For more information, see **Create Parameters** on page 1218.

Note: For cube data sources, fields are explicitly defined as dimensions or measures when the database is created. For relational data sources, Tableau automatically organizes the fields. By default, fields containing text, date or boolean values are dimensions, while fields containing numerical values are measures.

By default the field names defined in the data source are displayed in the Data pane. You can rename fields and member names, create hierarchies, and organize the fields into groups and folders. For details, see **Edit Default Settings for Fields** on page 1157, **Organize and Customize Fields in the Data Pane** on page 1146, and **Create Hierarchies** on page 1170

Columns become measure and dimension fields in the view

Data sources contain fields. For relational data sources that you connect to, the fields are determined by the columns of a table or view. Each field contains a unique attribute of the data such as customer name, sales total, product type, and so on.

For cube (multidimensional) data sources, the fields are determined by the dimensions and measures of a cube. In Tableau, cube data sources are supported only in Windows.

Here's an example of fields from an Excel worksheet.

	A	B	C	D	E	F	G	H
1	Row ID	Order ID	Order Date	Ship Date	Ship Mode	Customer ID	Customer Na	Segment
2	1	CA-2013-152156	11/9/2014	11/12/2014	Second Class	CG-12520	Claire Gute	Consumer
3	2	CA-2013-152156	11/9/2014	11/12/2014	Second Class	CG-12520	Claire Gute	Consumer
4	3	CA-2013-138688	6/13/2014	6/17/2014	Second Class	DV-13045	Darrin Van H	Corporate
5	4	US-2012-108966	10/11/2013	10/18/2013	Standard Clas	SO-20335	Sean O'Donn	Consumer
6	5	US-2012-108966	10/11/2013	10/18/2013	Standard Clas	SO-20335	Sean O'Donn	Consumer
7	6	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
8	7	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
9	8	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
10	9	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
11	10	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
12	11	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
13	12	CA-2011-115812	6/9/2012	6/14/2012	Standard Clas	BH-11710	Brosina Hoffr	Consumer
14	13	CA-2014-114412	4/16/2015	4/21/2015	Standard Clas	AA-10480	Andrew Allen	Consumer
15	14	CA-2013-161389	12/6/2014	12/11/2014	Standard Clas	IM-15070	Irene Maddo	Consumer
16	15	US-2012-118983	11/22/2013	11/26/2013	Standard Clas	HP-14815	Harold Pawla	Home Office
17	16	US-2012-118983	11/22/2013	11/26/2013	Standard Clas	HP-14815	Harold Pawla	Home Office
18	17	CA-2011-105893	11/11/2012	11/18/2012	Standard Clas	PK-19075	Pete Kriz	Consumer
19	18	CA-2011-167164	5/13/2012	5/15/2012	Second Class	AG-10270	Alejandro Gr	Consumer
20	19	CA-2011-143336	8/27/2012	9/1/2012	Second Class	ZD-21925	Zuschuss Do	Consumer
21	20	CA-2011-143336	8/27/2012	9/1/2012	Second Class	ZD-21925	Zuschuss Do	Consumer
22	21	CA-2011-143336	8/27/2012	9/1/2012	Second Class	ZD-21925	Zuschuss Do	Consumer

When you start building a visualization in a worksheet, these columns are available as fields in the Data pane. For details, see **Areas of the Data pane** on page 203.

Each field has a data type (that you can change if needed), and a role: discrete dimension, continuous dimension, discrete measure, or continuous measure. For details, see **Data Types** on page 325 and **Dimensions and Measures, Blue and Green** on page 314.

Each field also includes some default settings, such as a default aggregation of SUM or AVG, depending on the structure of the current view. For details, see see **Edit Default Settings for Fields** on page 1157 and **Data Aggregation in Tableau** on page 341.

Fields that Tableau automatically creates

The Data pane can also contain a number of fields that do not come from your original data: Measure Names and Measure Values, Number of Records, Latitude and Longitude.

Measure Names and Measure Values

- The **Measure Values** field contains all the measures in your data, collected into a single field with continuous values. Drag individual measure fields out of the Measure Values card to remove them from the view.
- The **Measure Names** field contains the names of all measures in your data, collected into a single field with discrete values.

> For more details about how to use Measure Values and Measure Names in visualizations, see **Measure Values and Measure Names** on page 1310. To see measure names and measure values in action, watch the 5-minute Measure Names and Measure Values training video. Use your tableau.com account to sign in. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Count of Table

Starting with Tableau 2020.2, every table in a data source has a **Count** field, in the form of *NameofTable*(Count). The table count field is an automatically generated, calculated field.

COUNT of table = SUM of the number of records per table

To see the count for a table, drag its Count field into the view. To see the count for all tables, select the Count field for each table in the Data pane, and then click the Text Table in Show Me.

You can't build calculations on top of a table's Count field, and it is aggregate-only.

Number of Records (before version 2020.2)

You might see a Number of Records field if it is being used in a visualization from a previous version of Tableau.

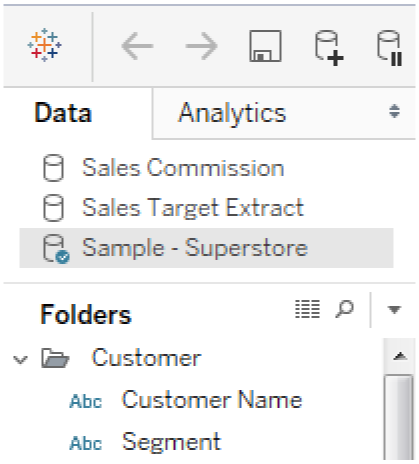
In versions of Tableau before 2020.2, the **Number of Records** field is an automatically generated, calculated field that is set to the number 1. That number gets associated with each row in the data source. If you add the Number of Records field to the view, you will see the summed count of all the rows in a data source (the number of records). You can use the Number of Records field to get quick counts of various dimensions' values. Viewing the Number of Records can help you to see if you data joins are working the way you expect them to.

Latitude and Longitude (generated)

When Tableau interprets fields to be geographic fields that can be used with maps, it automatically geocodes the data and includes **Latitude (generated)** and **Longitude (generated)** fields. You can use these fields to overlay your data on live maps. For more information on how to use these fields, and best practices for building maps in Tableau, see **Maps and Geographic Data Analysis in Tableau** on page 1755 **Assign Geographic Roles** on page 1834, **Location Data that Tableau Supports for Building Map Views** on page 1795,

Perform common tasks in the Data pane

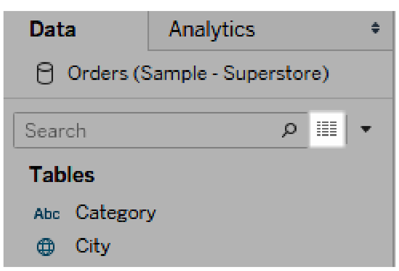
To select a data source connection for analysis, click the data source connection name in the Data pane. For more details, see **Navigating Data Sources in the Data Pane** on page 211.



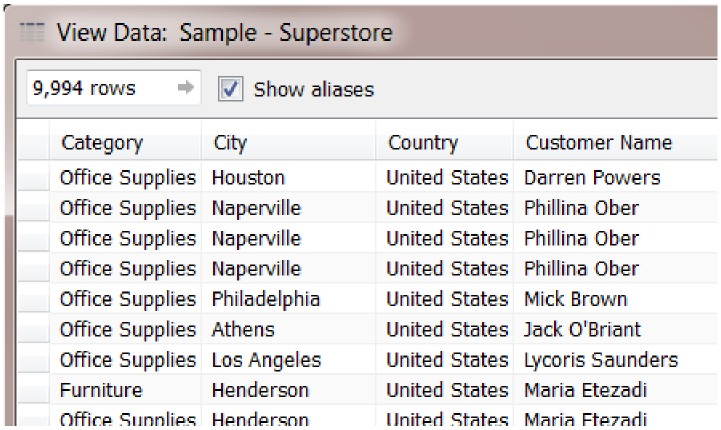
To view a context menu for the data source, click **Data** in the top menu and then click on the data source in the menu list. For more information, see **Edit Data Sources** on page 1060.

To search for fields in the Data pane, click the magnifying class icon and then type in the text box.

To see the underlying data, click the **View Data** icon at the top of the **Data** pane.



For more information , see **View Underlying Data** on page 2728



In cases where Tableau has misclassified a field as a dimension or a measure, possibly because of the data type, you can convert it and change its role.

To convert a measure to a dimension, drag the measure and drop it into the **Dimensions** area in the **Data** pane. For more details, see **Convert a Measure to a Dimension** on page 1165.

Change data fields to fit your needs

When you drag a field into the view, it will have certain default settings and characteristics. You can customize a field that is already in the view, just for that instance of the field. Or you can change its settings in the Data pane to make the field use those settings going forward.

You can control the definition of a field in the view, depending on how you want to work with that field data.

> **Note**: To change the default settings for a field before you drag it into the view, right-click it (Control-click on a Mac). You can then edit its settings and default properties from its context menu.

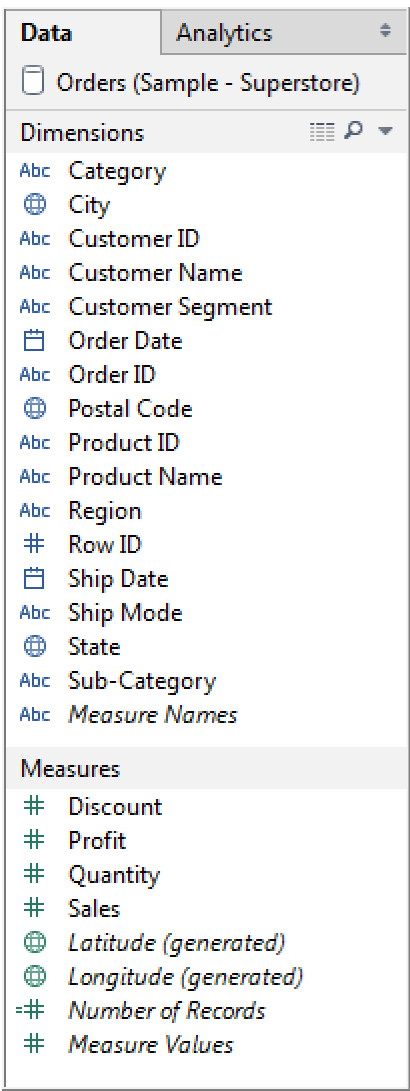
For details on the many ways you can customize the fields in the Data pane, see **Organize and Customize Fields in the Data Pane** on page 1146 and **Edit Default Settings for Fields** on page 1157.

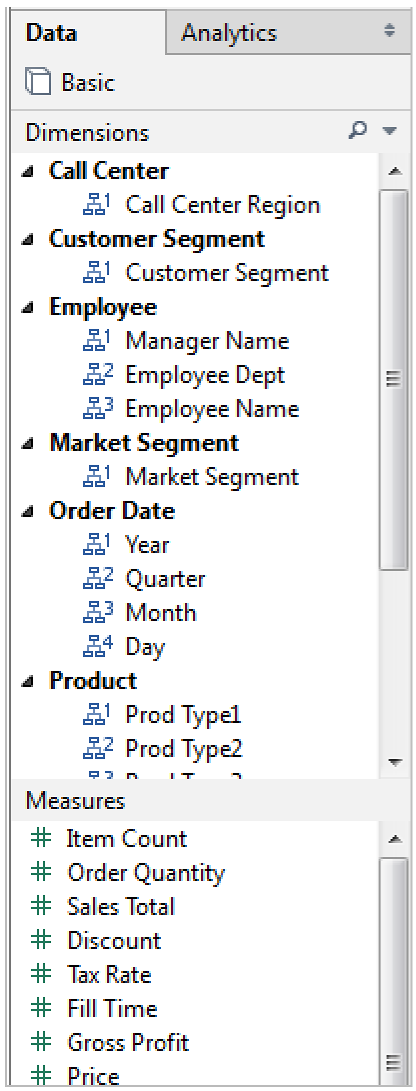
Relational versus cube data

The Data pane for a relational and cube data source are shown below. Note that the panes look essentially the same for both data sources in that the fields are organized into dimensions and measures. However, the cube data source contains hierarchies for dimensions. For example, notice that the Employee dimension in the cube Data pane contains hierarchical members such as Manager Name and Employee Dept.

Relational data sources don't have built-in hierarchies. However, relational data sources often have related dimensions that have an inherent hierarchy. For example, a data source may have fields for Country, State, and City. These fields could be grouped into a hierarchy called Location. You can assemble relational hierarchies by dragging and dropping in the Data pane. For details, see **Create Hierarchies** on page 1170.

> **Note:** In Tableau, cube (multidimensional) data sources are supported only in Windows.





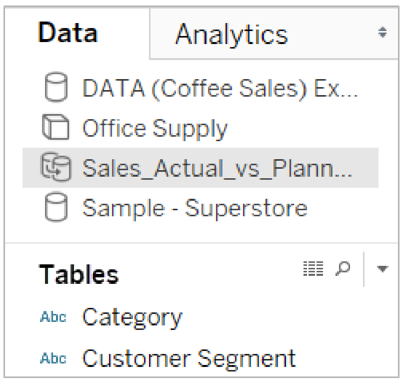
Data pane with relational data (left image) versus cube data (right image)

You can expand or collapse hierarchies in both relational and cube Data panes by clicking the arrow. You can hide the Data pane all together by clicking the minimize button in the upper-right corner of the Data pane.

For information on cube data sources, see **Cube Data Sources** on page 1085.

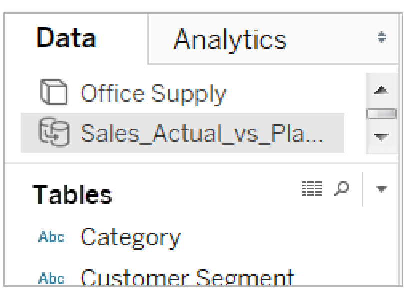
Navigating Data Sources in the Data Pane

The top of the Data pane lists all of the data sources in a given workbook. Click the data source you want to use to select it. The Data pane updates to show the corresponding fields in that data source.

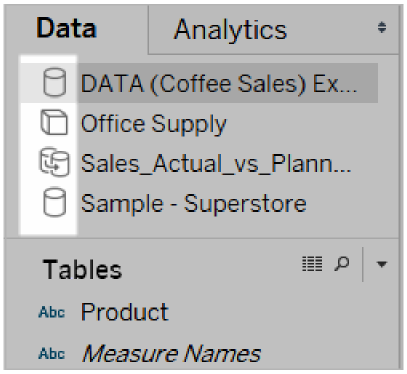


Note: images in this topic have not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

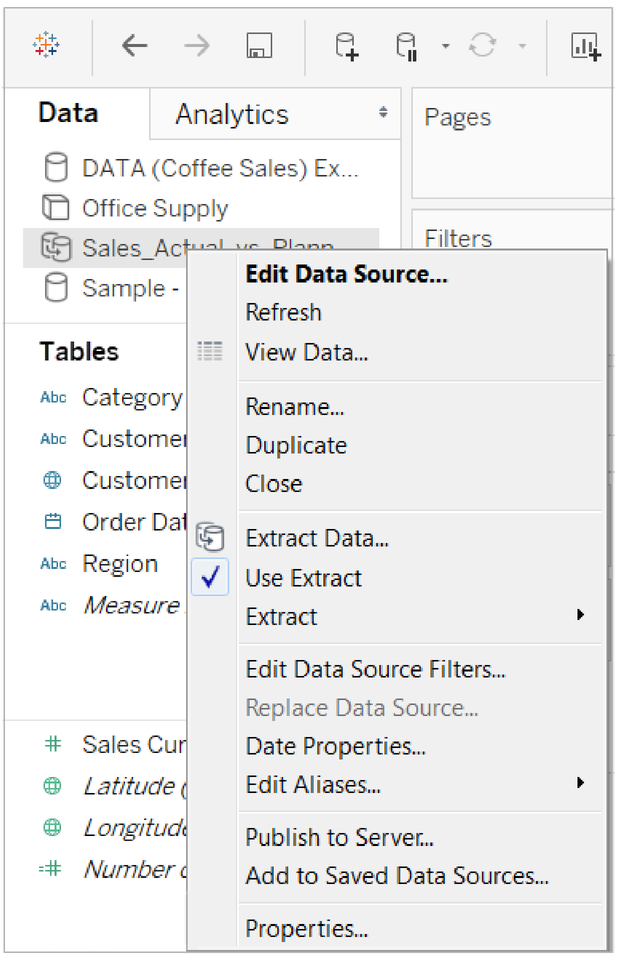
You can resize the data source list area in the Data pane to save space. When you resize to a limited vertical height, you can scroll down to a data source.



Each data source has an icon to indicate its type. For example, the icon can indicate whether the data source is relational, cube (multidimensional), or an extract. In Tableau, cube data sources are supported only in Windows.



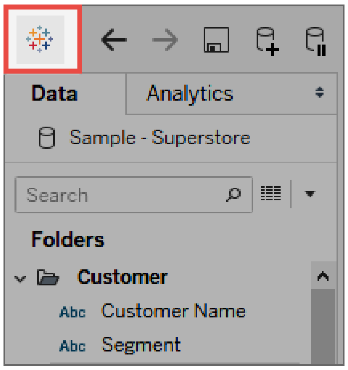
You can right-click (control-click on a Mac) the data source to access the commands that are on the Data menu. For example, you can right-click (control-click on a Mac) a data source and rename, export, or close it.



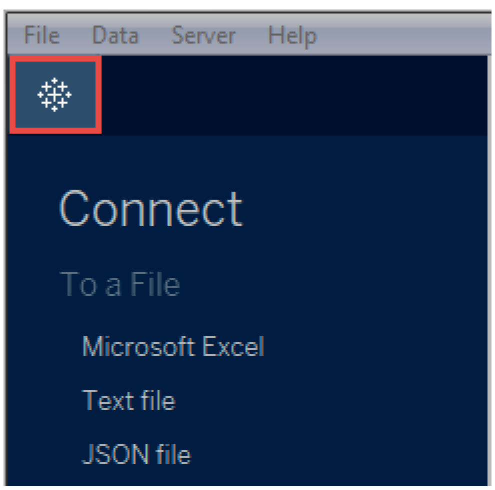
For details on the many ways you can customize and work with the fields in the Data pane, see **Organize and Customize Fields in the Data Pane** on page 1146, **Edit Default Settings for Fields** on page 1157, and **Work with Data Fields in the Data Pane** on page 203.

Navigate Between the Start Page and Workspace

When Tableau Desktop is already open, you can navigate to the start page by clicking the Tableau icon in the upper-left corner of the Tableau Desktop workspace.



To go back to the Tableau workspace, click the Tableau icon in the start page.



Apply Advanced Analysis to a View (Analytics Pane)

Drag reference lines, box plots, trend lines forecasts, and other items into your view from the **Analytics** pane, which appears on the left side of the workspace. Toggle between the **Data** pane and the **Analytics** pane by clicking the tabs at the top of the side bar.

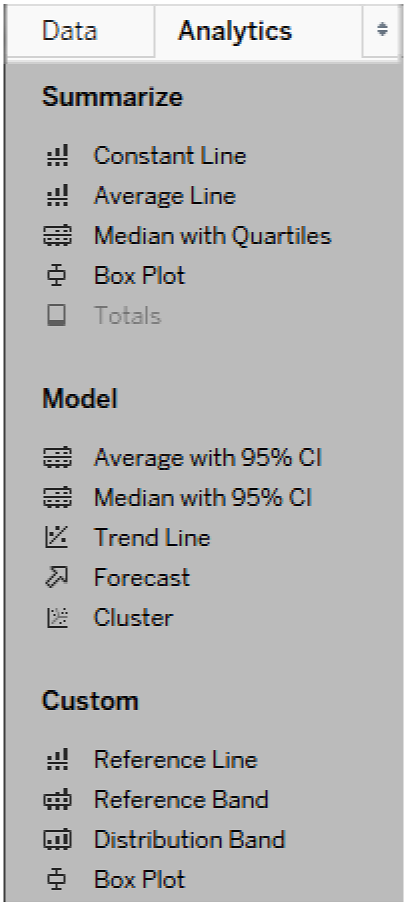


Tableau Desktop Analytics pane

In Tableau Desktop, options for adding Analytics objects to the view are available the **Analytics** pane or menu, or in context in the view. For example, reference lines and bands are available when you edit an axis, and trend lines and forecasts are available from the Analysis menu.

The **Analytics** pane provides drag-and-drop access for the various options.

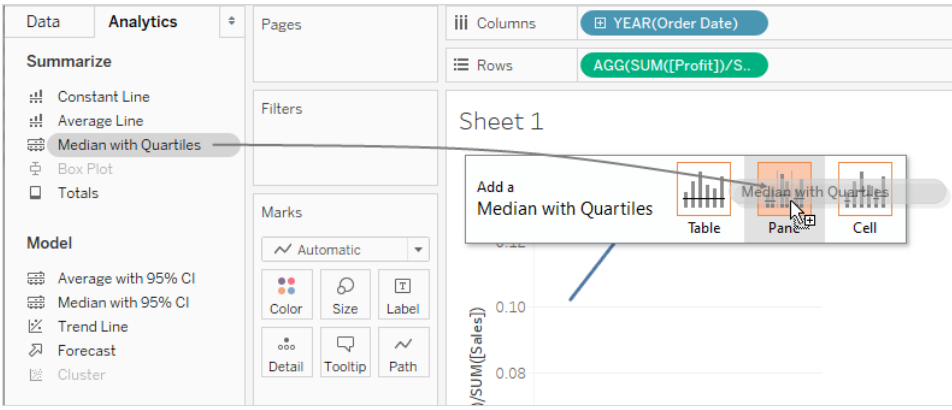
On the web, most Analytics objects are available from the Analytics pane.

Watch a video: To see related concepts on visual analytics in demonstrated in Tableau, watch the Getting Started with Visual Analytics (6-minutes) and Getting Started with Calculations (3-minutes), free training videos. Use your tableau.com account to sign in.

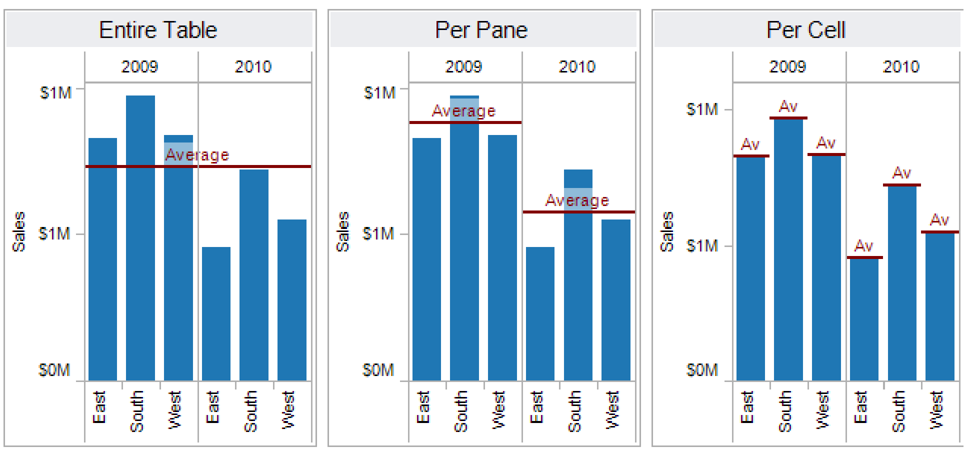
Add an analytics object to the view

To add an item from the **Analytics** pane, drag it into the view. When you drag an item from the **Analytics** pane, Tableau shows the possible destinations for that item. The range of choices varies depending on the type of item and the current view.

In a simple case, the drop target area would offer these three options:



The terms **Table**, **Pane** and **Cell** define the scope for the item:

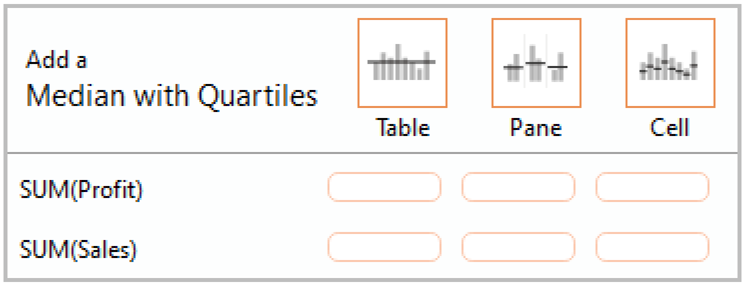


Adds a reference line to the entire table across all panes.

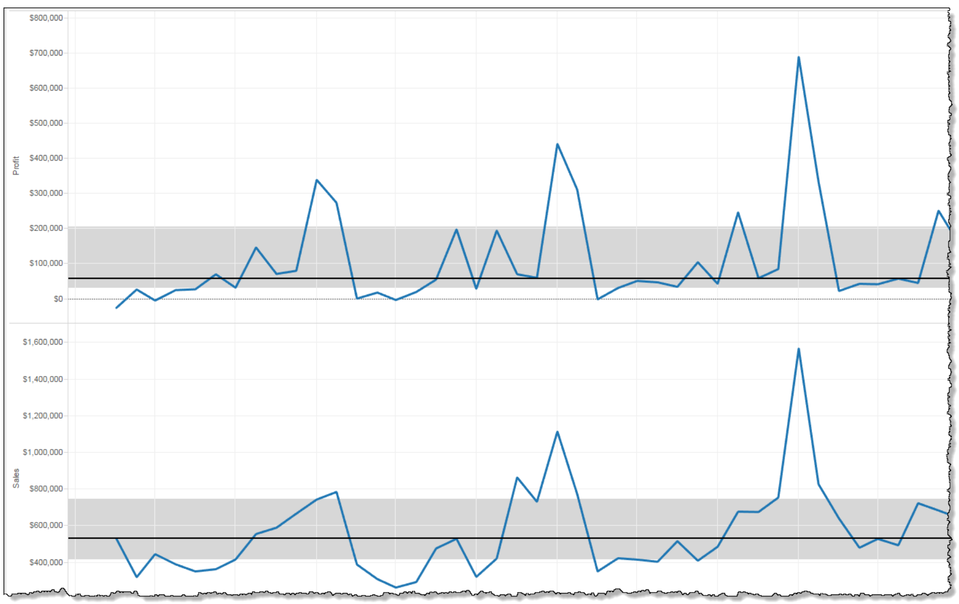
Adds a reference line on a per pane basis. Computed reference lines are recalculated for each pane in the view.

Adds a reference line within each cell. Computed reference lines are recalculated for each cell in the view.

For a more complicated view—for example, if the view contained a line chart with multiple or dual axes—Tableau would show you a drop target area that looked like this:



If you dropped the item in one of the three larger boxes in the header at the top of the drop target area--for example, the Table box--a separate median with quartiles would be added for each axis:



But if you drop the item in any of the six lower boxes aligned with a specific measure, the median with quartiles would only be added on the corresponding axis, with the specified scope.

Delete an analytics object from the view

You can delete an analytics object from the view by clicking Undo, or drag the object off the view to delete it.

You can also click on an item and choose **Remove** from the tooltip.

> **Note**: Some **Analytics** pane items (**Median with Quartiles** and **Average with 95% CI**) add both a reference line and a reference distribution. Unless you are using Undo, you would need to delete these items separately.

Edit an analytics object in the view

To edit an item you have added from the Analytics pane, click on the item and select **Edit** from the tooltip. For additional editing options, see the section for a particular item type under Analytics object definitions, below.

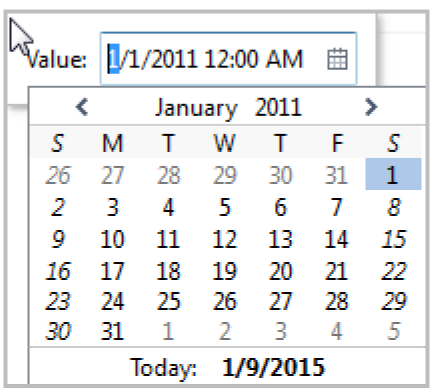
Analytics object definitions

The following items can be dragged from the **Analytics** pane and dropped in the view. If an analytics object cannot be applied to the current configuration of fields in the view, it isn't available.

Constant Line

Adds one or more constant lines to the view. You can add a constant line for a specific measure, for all measures, or for date dimensions. When you add a constant line, Tableau displays a Value prompt where you specify the value for the constant:

In Tableau Desktop, the Value prompt for a date value is a calendar control:



You can click on a resulting constant line and choose **Edit** or **Remove**. In Tableau Desktop there is a third option: **Format**. Choosing **Edit** opens the Edit Reference Line dialog box. For details, see **Edit Existing Reference Lines, Bands, and Distributions** on page 2527 in the **Reference Lines, Bands, Distributions, and Boxes** article. Another way to edit a line in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose **Edit Reference Line**.

Average Line

Adds one or more average lines to the view. You can add an average line for a specific measure or for all measures.

You can click on a resulting average line and choose a different aggregation, such as Total or Sum. You can also choose **Edit** or **Remove**. In Tableau Desktop there is a third option: **Format**. Choosing **Edit** opens the Edit Reference Line dialog box. For details, see **Edit Existing Reference Lines, Bands, and Distributions** on page 2527 in the **Reference Lines, Bands, Distributions, and Boxes** article. Another way to edit a line in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose **Edit Reference Line**.

Median with Quartiles

Adds one or more sets of median lines and distribution bands to the view. You can add a median with quartiles for a specific measure or for all measures.

The distribution bands are computed as quartiles; the middle two quartiles are shaded.

You can click on a resulting median line or distribution and choose **Edit** or **Remove**. In Tableau Desktop there is a third option: **Format**. Median lines and distributions must be edited, formatted, or removed separately. Choosing **Edit** opens the Edit Reference Line dialog box. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect. For details, see **Edit Existing Reference Lines, Bands, and Distributions** on page 2527 in the **Reference Lines, Bands, Distributions, and Boxes** article. Another way to edit a line or distribution in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose **Edit Reference Line**. A submenu will offer you two choices: **Quartiles** and **Median**.

For information on distribution types, including quartiles, see **Add Reference Distributions** on page 2518 in the **Reference Lines, Bands, Distributions, and Boxes** article.

Box Plot

Adds one or more box plots to the view. You can add box plots for a specific measure or for all measures. The scope for a box plot is always **Cell** (and never **Table** or **Pane**).

Click or hover over any of the horizontal lines in the box plot to see statistical information about the whiskers, quartiles, and median.

You can also choose **Edit** or **Remove** when you click on a line. In Tableau Desktop there is a third option: **Format**. Choosing **Edit** opens the Edit Reference Line dialog box. Another way to edit a box plot in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose **Edit Reference Line**.

> **Note**: In Tableau Desktop, there are two items named **Box Plot** in the Analytics pane. For the Box Plot option in the Summarize section, Tableau will automatically add a box plot for the specified target. For the Box Plot option in the Custom section, Tableau will open the Edit Reference Line, Band, or Box dialog box after you specify a target.

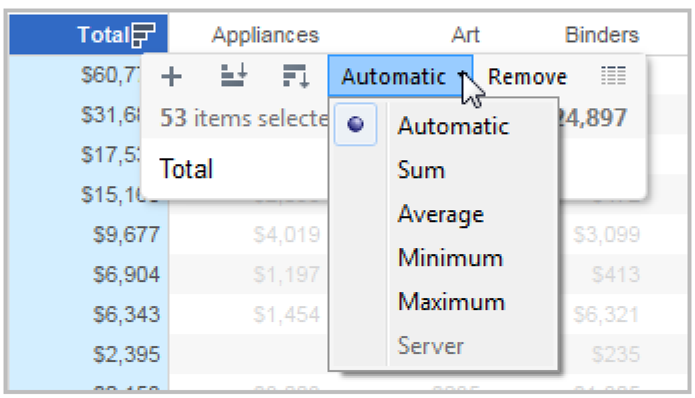
Totals

Adds totals to the view. When you add totals, the drop options are **Subtotals**, **Column Grand Totals**, and **Row Grand Totals**.

For details, see **Show Totals in a Visualization** on page 1456.

To remove totals, click the relevant column or row header and choose **Remove**.

In Tableau Desktop, you can also click on a totals column or row header after adding totals and set the aggregation for that row or column from the tooltip:



Average with 95% CI

Adds one or more sets of average lines with distribution bands; the distribution bands are configured at a 95% confidence interval. You can add these items for a specific measure or for all measures.

The confidence interval distribution bands shade the region in which the population average will fall 95% of the time.

You can click on a resulting average line or distribution and choose **Edit** or **Remove**. In Tableau Desktop there is a third option: **Format**. Choosing **Edit** opens the Edit Reference Line dialog box. The average lines and distributions must be edited, formatted, or removed separately. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect. Another way to edit a line or distribution in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose **Edit Reference Line**. A submenu will offer you two choices: **Average** and **95% Confidence Interval**.

You can also remove lines and bands by dragging them off the view.

Median with 95% CI

Adds one or more sets of median lines with distribution bands; the distribution bands are configured at a 95% confidence interval. You can add these items for a specific measure or for all measures.

The confidence interval distribution bands shade the region in which the population median will fall 95% of the time.

You can click on a resulting median line or distribution and choose **Edit**, **Format**, or **Remove**. In Tableau Desktop there is a third option: **Format**. Choosing **Edit** opens the Edit Reference Line dialog box. The median lines and distributions must be edited, formatted, or removed separately. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect. Another way to edit a line or distribution in Tableau Desktop is to right-click (control-click on Mac) the relevant axis and choose **Edit Reference Line**. A submenu will offer you two choices: **Median** and **95% Confidence Interval**.

You can also remove lines and bands by dragging them off the view.

Trend Line

Adds one or more trend lines to the view. When you add trend lines, the drop options identify the trend line model types available in Tableau: **Linear**, **Logarithmic**, **Exponential**, and **Polynomial**. For some views, only a subset of these options is available.

For details, see **Trend Line Model Types** on page 2535

Click on a trend line to remove or edit it, or to see a statistical definition. You can also remove a trend line by dragging it off the view.

Forecast

Adds a forecast to the view. This option is only available in Tableau Desktop—not when you edit a view on the web. Forecasting is only possible when there is at least one measure in the view.

Forecasting is not supported for views based on multidimensional data sources. In addition, the view cannot contain any of the following:

- Table calculations
- Disaggregated measures
- Percent calculations
- Grand Totals or Subtotals
- Date values with aggregation set to Exact Date

A time series containing null values also imposes constraints.

For details, see **Forecasting** on page 2580.

To remove, edit, or read a description of the current forecast, go to the Analysis menu and choose **Forecast**.

Custom Reference Line

You can add reference lines for a specific measure or for all measures in the view.

After you drag a reference line from the **Analytics** pane and drop it on a target, Tableau automatically opens an edit dialog box. See **Add a Reference Line** on page 2506 in the **Reference Lines, Bands, Distributions, and Boxes** article for information on the available options. To return to this dialog box later, click on the line and choose **Edit**.

Custom Reference Band

You can add reference bands for a specific measure or for all measures in the view.

After you drag a reference band from the **Analytics** pane and drop it on a target, Tableau automatically opens the Edit Reference Line, Band, or Box dialog box. See **Add Reference Bands** on page 2513 in the **Reference Lines, Bands, Distributions, and Boxes** article for information on the available options. To return to this dialog box later, click on the band and choose **Edit**. You must click on the outer edge of a reference band to see the options--clicking in the middle of the band has no effect.

Custom Distribution Band

You can add reference distributions for a specific measure or for all measures in the view.

After you drag a reference distribution from the **Analytics** pane and drop it on a target, Tableau automatically opens Edit Reference Line, Band, or Box dialog box. See **Add Reference Distributions** on page 2518 in the **Reference Lines, Bands, Distributions, and Boxes** article for information on the available options. To return to this dialog box later, click on the band and choose **Edit**. You must click on the outer edge of a distribution band to see the options--clicking in the middle of the band has no effect.

Custom Box Plot

In Tableau Desktop—but not when you edit a view on the web—you can drag a box plot from the Custom section of the **Analytics** pane and drop it on a target. (But keep in mind that on the web, you can add a box plot from the Summarize section on the Analytics pane.) When you drag Box Plot from the Custom section, Tableau automatically opens Edit Reference Line, Band, or Box dialog box. See **Add a Box Plot** on page 2523 in the **Reference Lines, Bands, Distributions, and Boxes** article for information on the available options. The scope for a box plot is always **Cell** (and never **Table** or **Pane**).

Click any of the horizontal lines in the box plot to see statistical information about the whiskers, quartiles, and median.

To return to this dialog box later, click on the band and choose **Edit**.

Shelves and Cards Reference

Every worksheet in Tableau contains shelves and cards, such as Columns, Rows, Marks, Filters, Pages, Legends, and more.

By placing fields on shelves or cards, you:

- Build the structure of your visualization.
- Increase the level of detail and control the number of marks in the view by including or excluding data.
- Add context to the visualization by encoding marks with color, size, shape, text, and detail.

Experiment with placing fields on different shelves and cards to find the optimal way to look at your data.

Options for starting a view

If you aren't sure where to place a field, you can get let Tableau help you determine the best way to display the data.

- You can drag fields from the **Data** pane and drop them onto the cards and shelves that are part of every Tableau worksheet.
- You can double-click one or more fields in the **Data** pane.
- You can select one or more fields in the **Data** pane and then choose a chart type from **Show Me**, which identifies the chart types that are appropriate for the fields you selected. For details, see **Use Show Me to Start a View** on page 1323.
- You can drop a field on the **Drop field here** grid, to start creating a view from a tabular perspective.

Columns and Rows shelves

Drag fields from the Data pane to create the structure for your visualizations.

The **Columns** shelf creates the columns of a table, while the **Rows** shelf creates the rows of a table. You can place any number of fields on these shelves.

When you place a dimension on the **Rows** or **Columns** shelves, headers for the members of that dimension are created. When you place a measure on the **Rows** or **Columns** shelf, quantitative axes for that measure are created. As you add more fields to the view, additional headers and axes are included in the table and you get an increasingly detailed picture of your data.

In the view shown below, the members of the **Segment** dimension are displayed as column headers, while the **Profit** measure is displayed as a vertical axis.

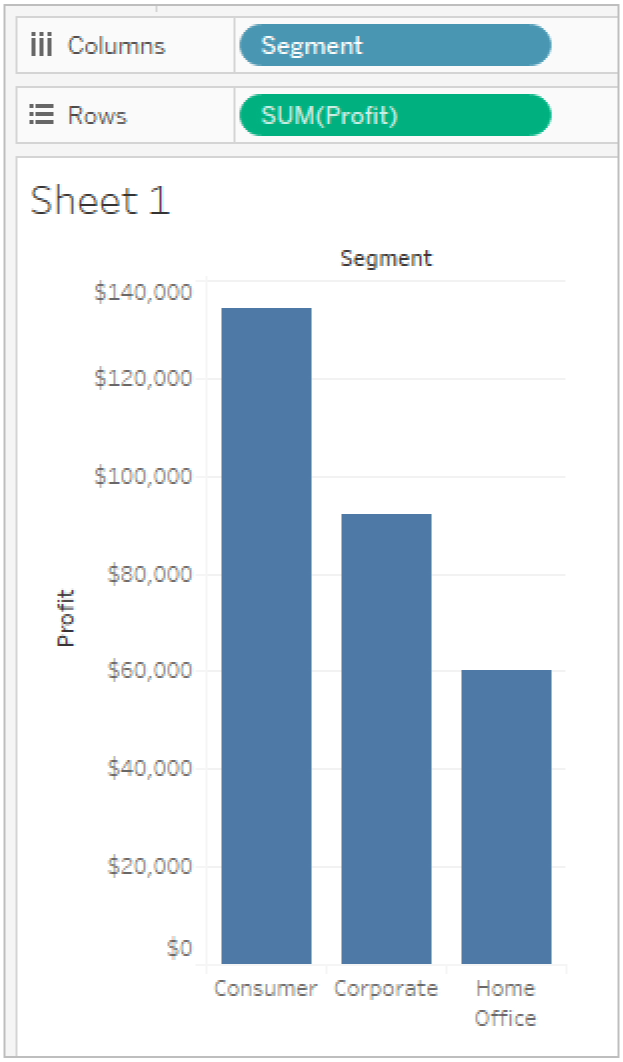
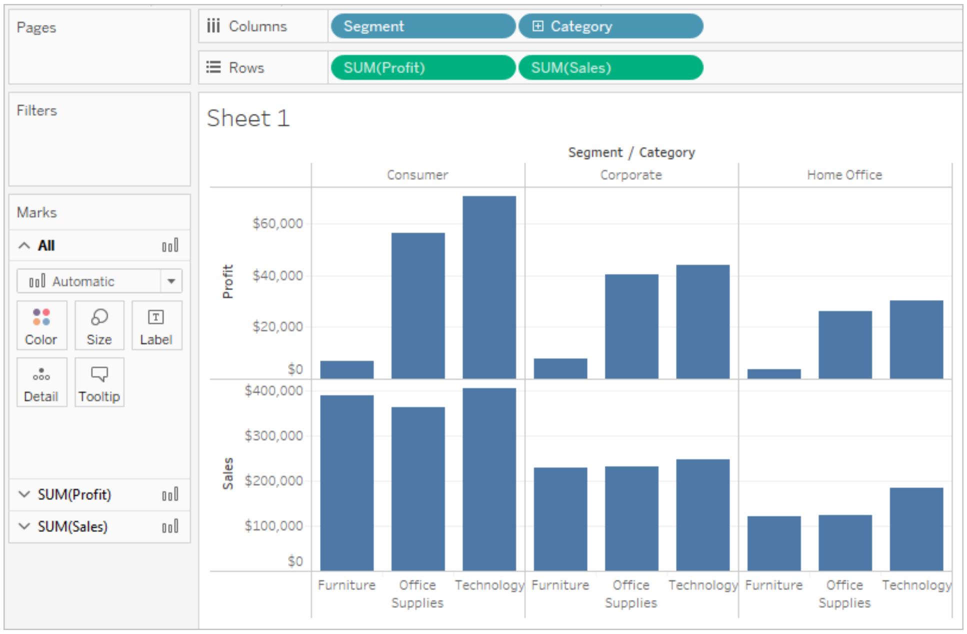


Tableau displays data using marks, where every mark corresponds to a row (or a group of rows) in your data source. The inner fields on the **Rows** and **Columns** shelves determine the default mark type. For example, if the inner fields are a measure and a dimension, the default mark type is a bar. You can manually select a different mark type using the Marks card drop-down menu. For more information, see **Change the Type of Mark in the View** on page 1332.

Adding more fields to the **Rows** and **Columns** shelves adds more rows, columns, and panes to the table.



Hide rows and columns

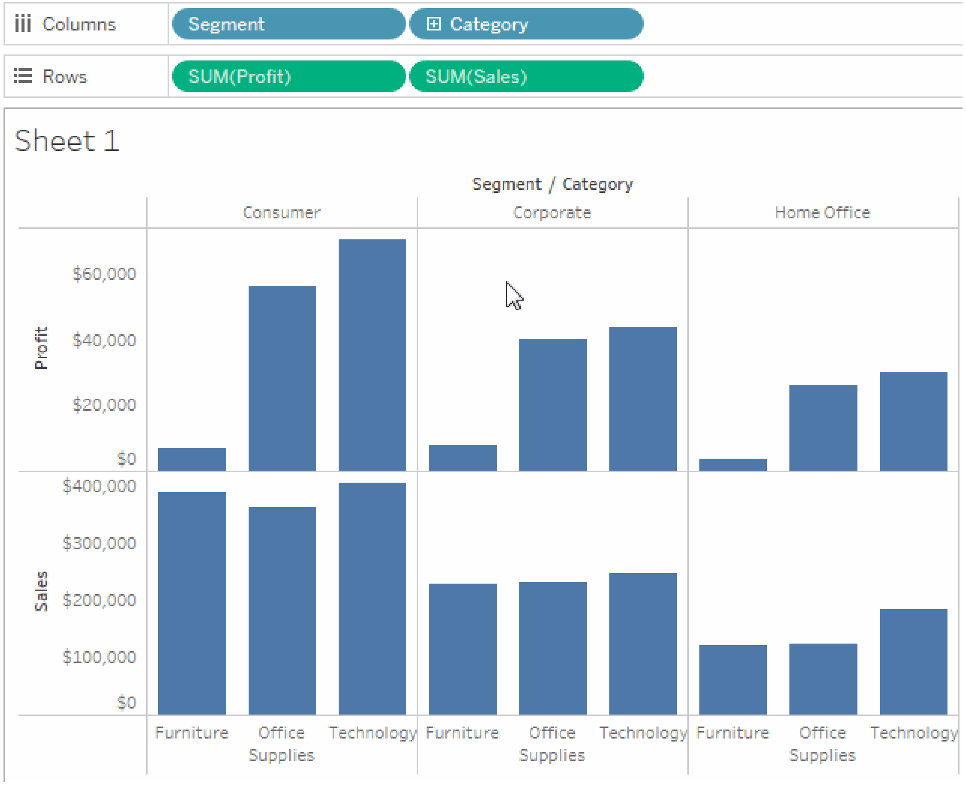
Generally you will add dimensions and measures to create the rows and columns of the table and you'll either include all data or add filters to only show a subset. However, when you filter data it is also excluded from calculations and other computations performed on the summarized data in the table. Instead of filtering the data, you can hide the row or column so it doesn't display in the view but it is still included in calculations.

Tip: Hiding columns is especially useful when using table calculations that compare to a previous or next date value. In these cases, there will be a row or column that doesn't show data because there is no data to be compared to. Hide the empty column to keep the table calculation intact. For more details on table calculations, see **Table Calculation Types** on page 2419.

For example, when calculating year-over-year growth, the first year wouldn't have a previous year to compare to, so the column is left blank. Filtering the first year would remove it from the view but it would also remove it from the calculation (the second year then doesn't have a previous year to compare to and is left blank). Instead of filtering, hiding the blank column keeps the calculation intact.

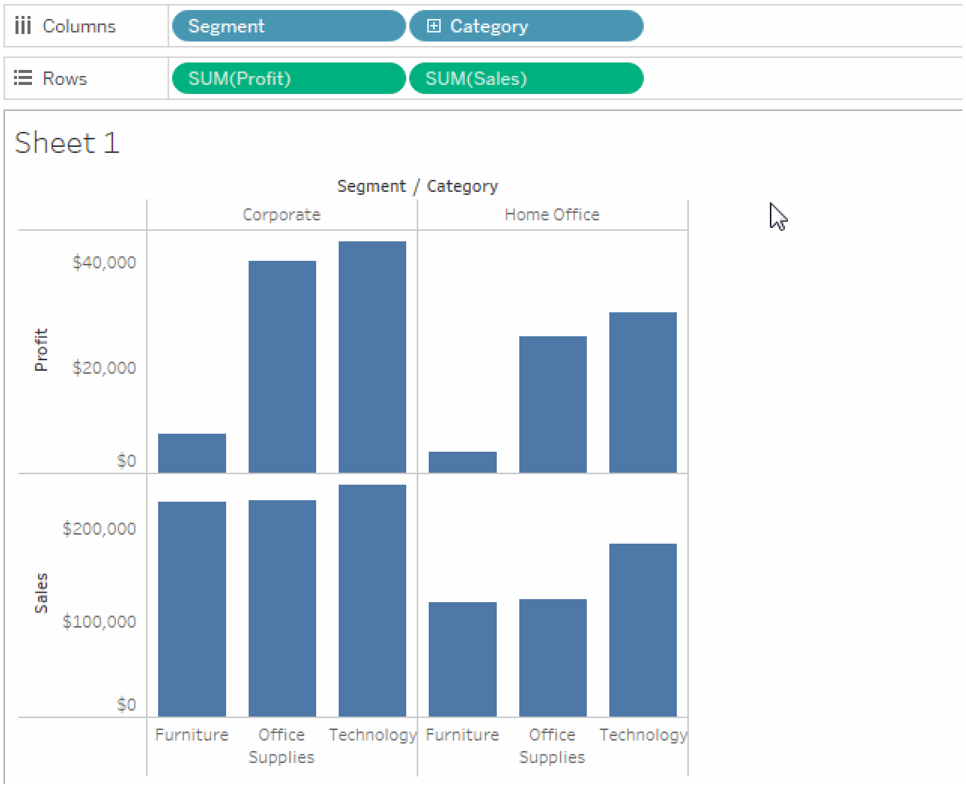
To hide a row or column:

Right-click (control-click on Mac) the row or column you want to hide, and then select **Hide**.



To show hidden data:

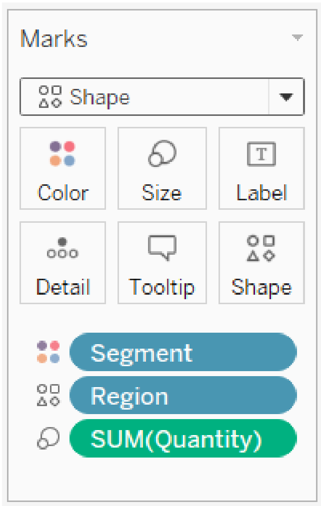
Open the field menu for a field that has hidden columns or rows and select **Show Hidden Data**.



Marks card

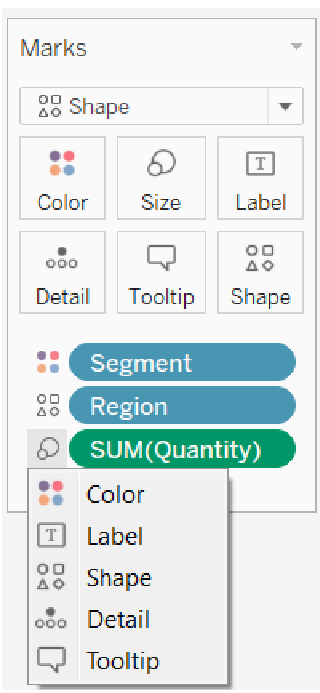
The Marks card is a key element for visual analysis in Tableau. As you drag fields to different properties in the Marks card, you add context and detail to the marks in the view.

You use the Marks card to set the mark type (see **Change the Type of Mark in the View** on page 1332), and to encode your data with color, size, shape, text, and detail. To change the mark settings, see **Control the Appearance of Marks in the View** on page 1356.



In this example, three different fields have been dragged to different properties in the Marks card. Segment is on Color, Region is on Shape, and Quantity is on Size.

After you add a field to the Marks card, you can click the icon next to the field to change the property it is using. You can also click the property buttons in the Marks card to change those settings.



Many properties can have multiple fields. For example, you can add multiple fields to Label, Detail, Tooltip, and Color. Size and Shape can only have one field at a time. For more details, see **Control the Appearance of Marks in the View** on page 1356.

> **Note:** By default, dragging a new field to **Color** replaces the existing fields. To add a new field to color without replacing the existing field, hold the SHIFT key on your keyboard while dragging a new field to **Color** on the Marks card.

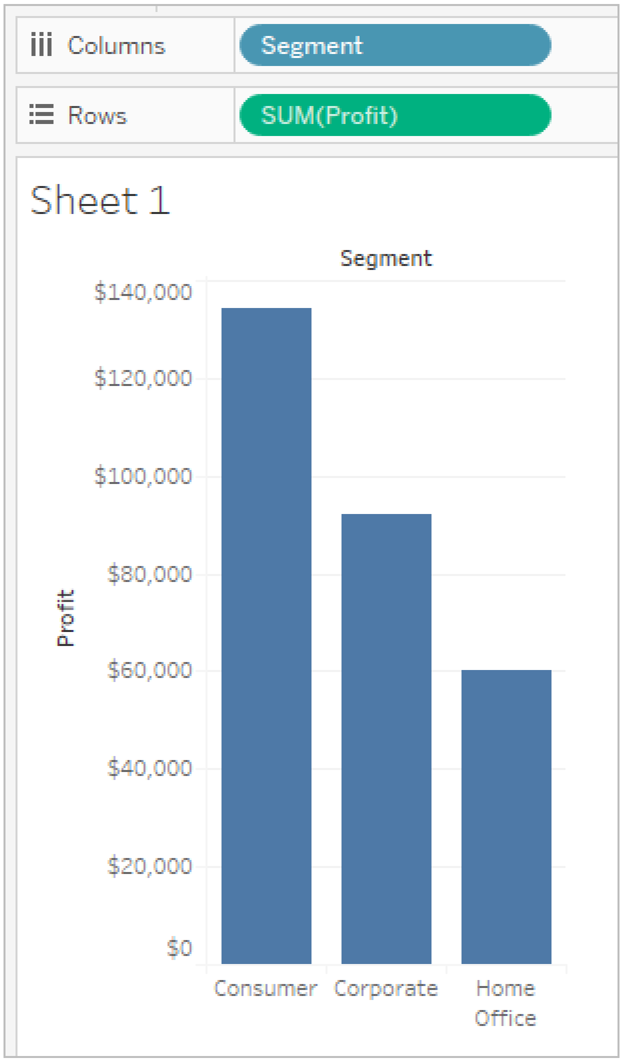
Filters shelf

The Filters shelf allows you to specify which data to include and exclude. For example, you might want to analyze the profit for each customer segment, but only for certain shipping containers and delivery times. By placing fields on the Filters shelf, you can create such a view.

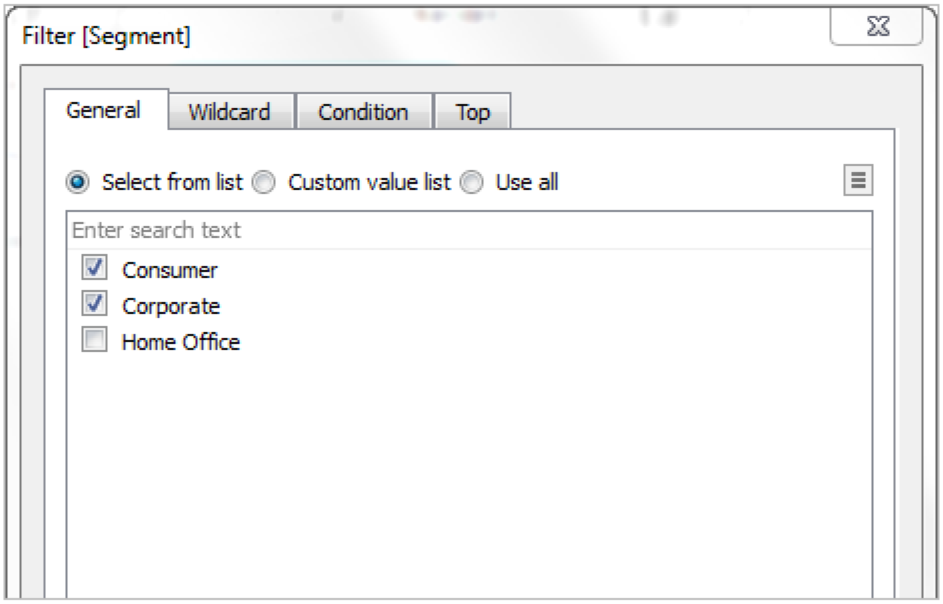
> **Note:** This section presents a brief overview of filtering. For more information on filtering, see **Filter Data from Your Views** on page 1411.

You can filter data using measures, dimensions, or both at the same time. Additionally, you can filter data based on the fields that make up the columns and rows of the table. This is called an internal filter. You can also filter data using fields that don’t contribute headers or axes to the table. This is called an external filter. All filtered fields display on the Filters shelf.

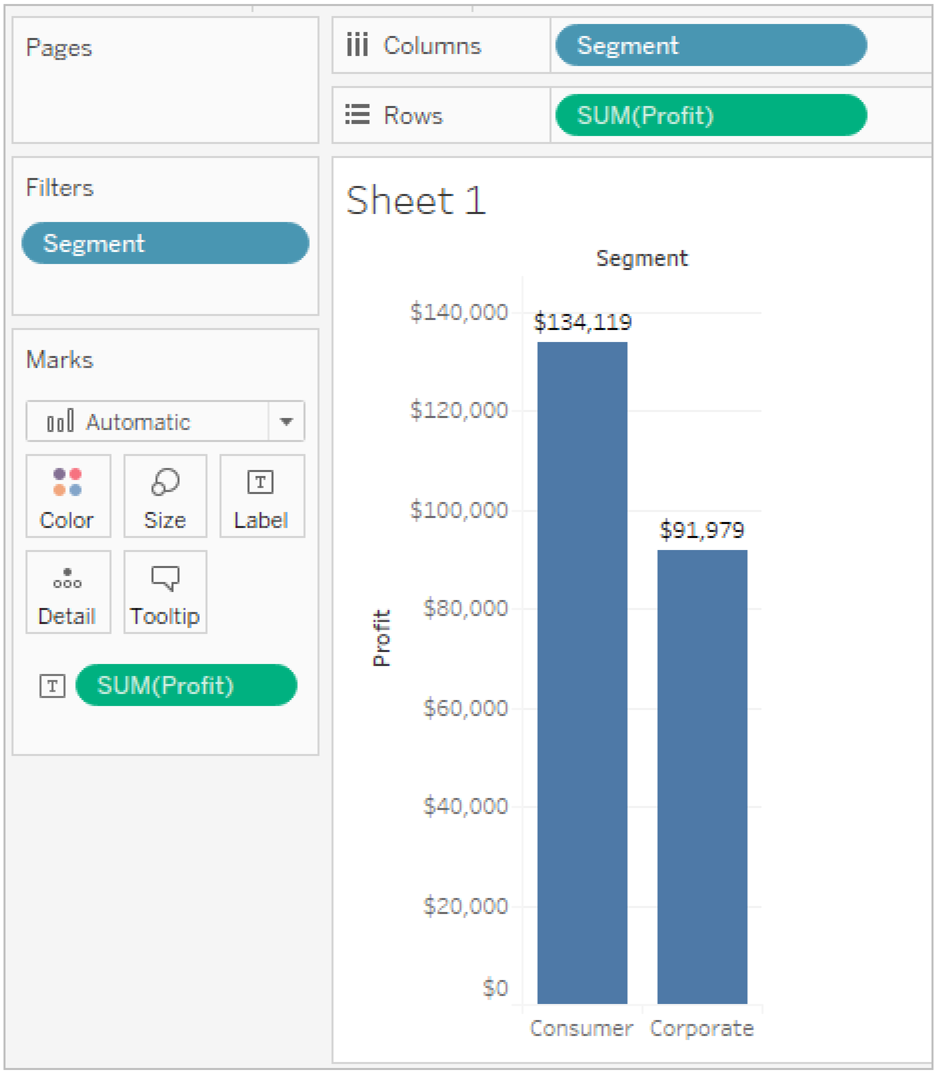
To illustrate the basic concepts of filtering, consider the following view.



Suppose you are not interested in the Home Office data. You can remove this column from the view by filtering the **Segment** dimension. To do so, select **Filter** on the field menu or drag the **Segment** dimension to the **Filters** shelf. The Filter dialog box opens. By default all members are selected. Clear the check box for **Home Office** to exclude it from the view. All selected members will be included.



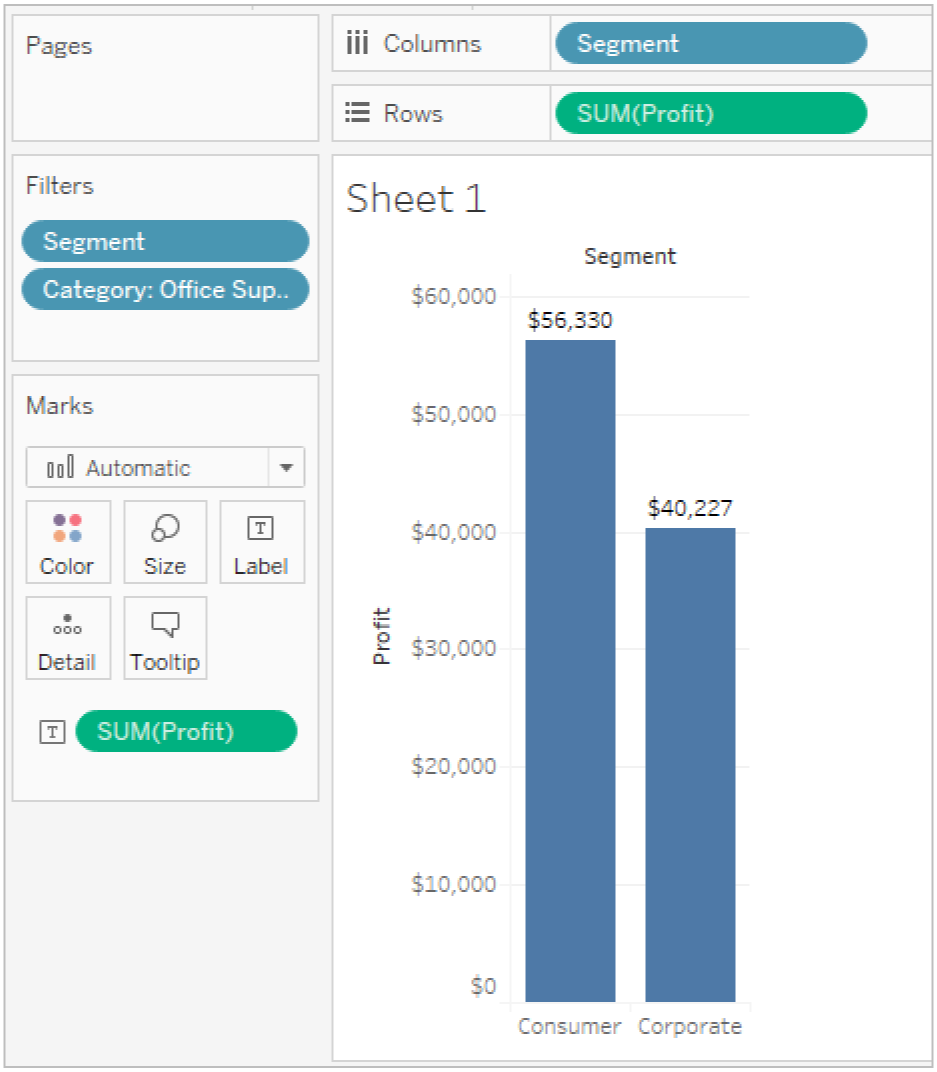
As shown below, the view updates and the Home Office column is removed. The filter is indicated by the **Segment** field on the **Filters** shelf.



Suppose you want to only view profit for a category of the products. Even though the **Category** field is not used on the **Rows** and **Columns** shelves or on the Marks card, you can still add a filter. Drag the **Category** dimension to the **Filters** shelf. This is an example of an external filter because Category is not part of the view.

The Filter dialog box automatically opens. By default, none of the members are selected. Select the members you want to keep as part of the view. All cleared members are excluded. In this example, Office Supplies is selected.

The modified data view is shown below. The mark label shows that the sum of the profit for the Consumer segment has decreased to $56,330. This number is derived by summing all the rows in the data source that are associated with the Corporate market and are part of the Office Supplies category.

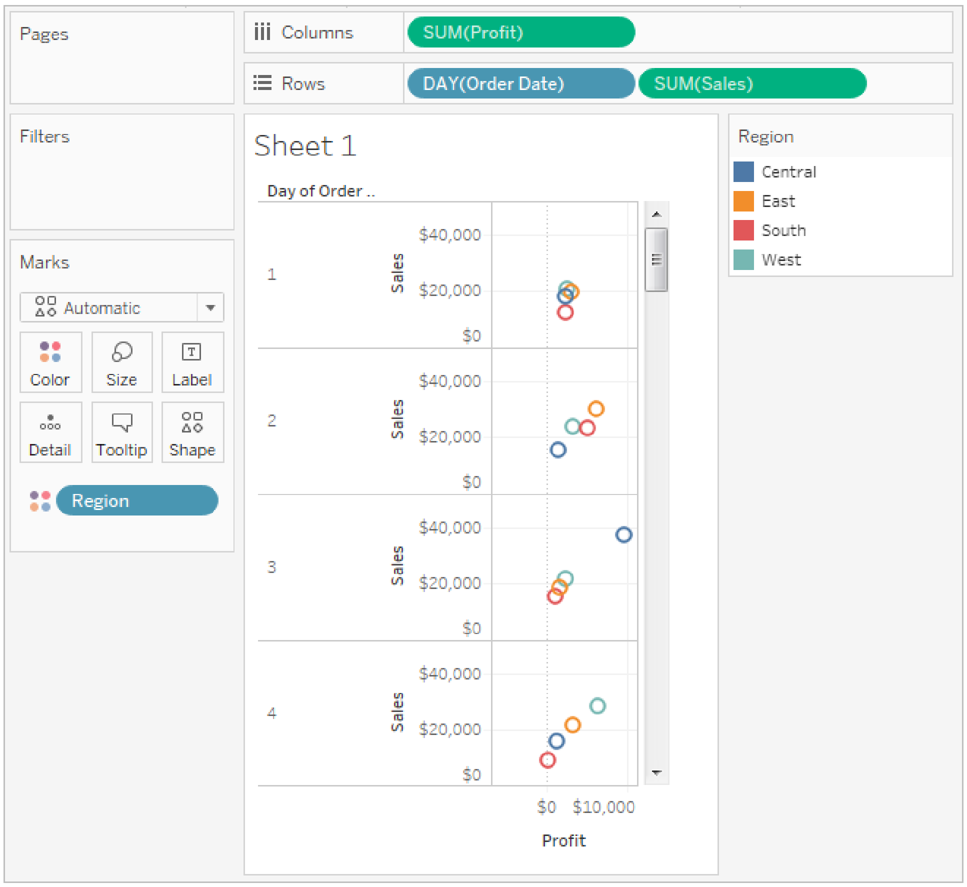


The order of fields placed on the Filters shelf does not affect the data view because the filters are independent. In other words, the result of filtering by customer segment, and then by container is the same as filtering by container and then by customer segment. For more information about the filtering order of operations, see **Filter Data from Your Views** on page 1411.

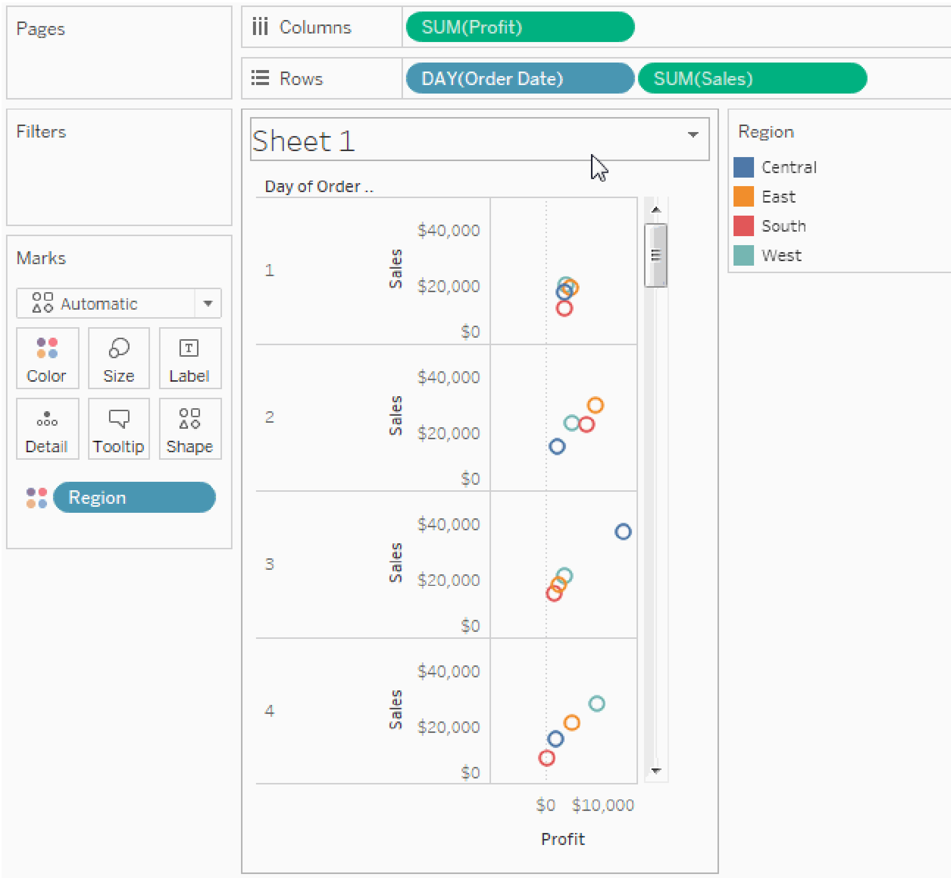
Pages shelf

The **Pages** shelf lets you break a view into a series of pages so you can better analyze how a specific field affects the rest of the data in a view. When you place a dimension on the **Pages** shelf you are adding a new row for each member in the dimension. When you place a measure on the **Pages** shelf, Tableau automatically converts the measure into a discrete measure.

The **Pages** shelf creates a set of pages, with a different view on each page. Each view is based on a member of the field you placed on the **Pages** shelf. You can easily flip through the views and compare them on a common axis, using the controls that get added to the view when you move a field to the **Pages** shelf. For example, the view below shows the **Profit** vs. **Sales** by **Region** for each day throughout the month. The image below shows days 1, 2, 3, and 4. You would have to scroll down to see other days in the month.



To make this view more user-friendly, move **DAY(Order Date)** to the **Pages** shelf and use the associated control to flip through the pages (one for each day). You can quickly discover hidden insights. In this example, it is interesting that the 19th is an especially big day in terms of sales and profit in the Western region.



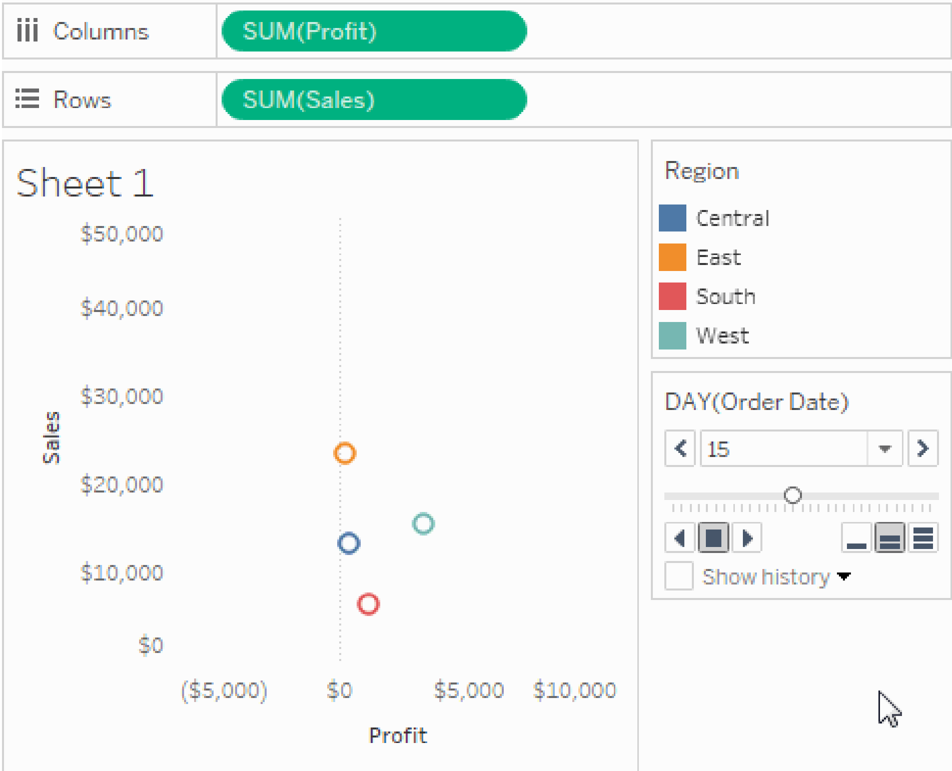
When you add a field to the **Pages** shelf, a page control is automatically added to the right of your view.

Use this control to navigate through the pages. There are three ways to navigate through the pages in a view:

- Jump to a specific page
- Manually advance through the pages
- Automatically advance through pages

Jump to a specific page

Select the member or value you want to view from the drop-down list to display a specific page.



Manually advance through the pages

You can manually advance through the sequence of pages by doing any of the following:

- Use the forward and back buttons on either side of the drop-down list to navigate through the pages one at a time.
- Use the Page Slider to quickly scroll forward and backward in the sequence of pages.
- Use any of the following keyboard shortcuts to scroll forward and backward in the sequence of pages.

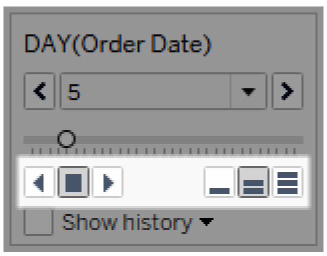
F4	Starts and stops forward playback
SHIFT + F4	Starts and stops backward playback
CTRL + .	Skip forward one page
CTRL + ,	Skip backward one page

On a Mac, these are the equivalent keyboard shortcuts.

F4	Starts and stops forward playback
Shift-F4	Starts and stops backward playback
Command-period	Skip forward one page
Command-comma	Skip backward one page

Automatically advance through pages

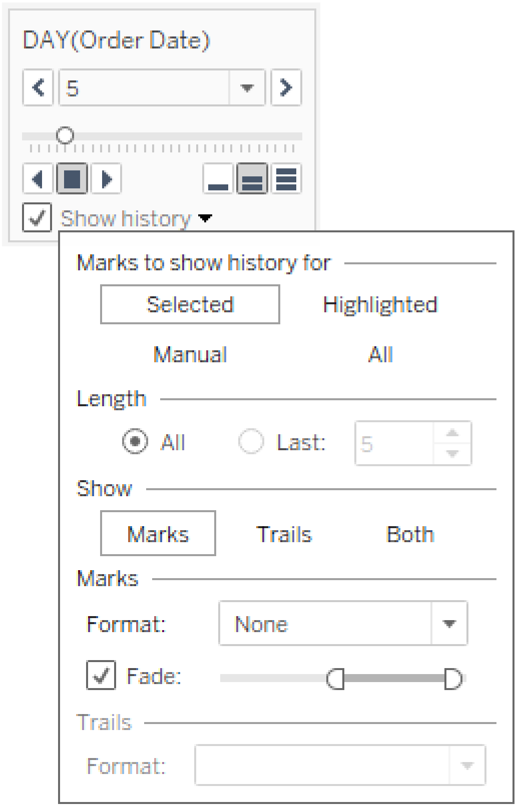
Use the playback controls to watch a slide show of the pages in the view. You can play forward or backward, and stop the playback at any time. You can control the speed of playback with the speed controls in the bottom right corner of the control. The smallest bar indicates the slowest playback speed.



Show the Page history

Show page history using the Show History check box. With page history, marks from previous pages are shown on the current page.

In Tableau Desktop only: Open the drop-down control for history to specify what marks to show and when to show them.



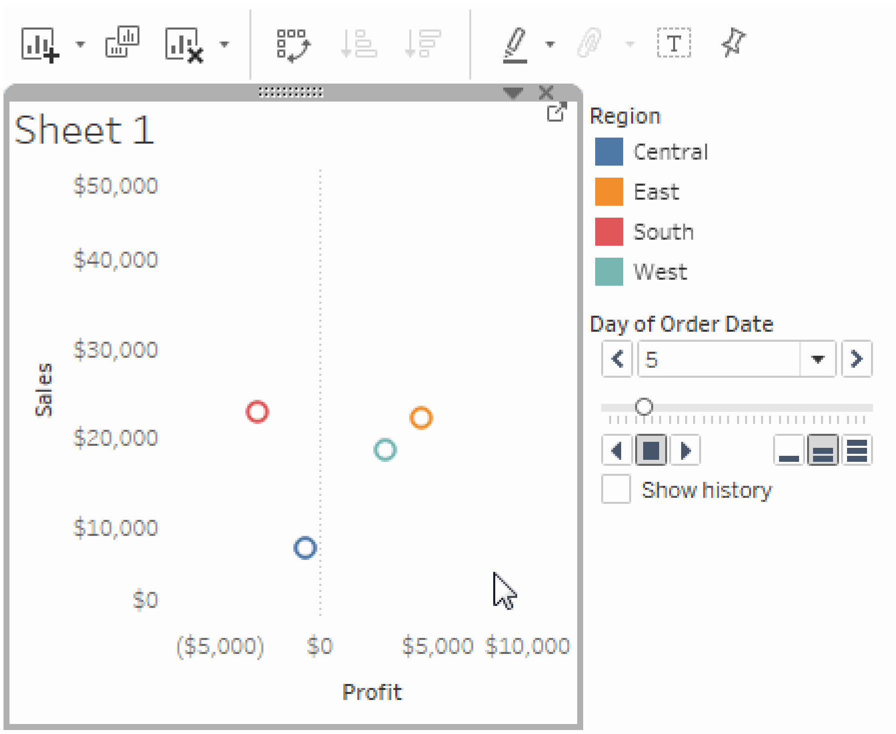
The history drop-down control has the following options:

- **Marks to show history for** – select whether you want to show history for just selected marks, highlighted marks, marks that you've manually selected to show history for, or all marks. To manually show history for marks, right-click (control-click on Mac) the mark in the view and select an option on the Page History menu.
- **Length** – select the number of pages to show in the history.
- **Show** – specify whether to show the historical marks, a line tracing through the previous values (trails), or both.
- **Marks** – format the historical marks including the color and how much to fade them if the color is set to automatic, the marks will either use the default mark color or the color encoding on the Color shelf.
- **Trails** – format the lines that are drawn through the historical marks. This option is only available if Trails is selected in the Show options.

Page trails may not display if there are multiple marks per color on a page. Make sure that the level of detail for the view is less than or equal to the level of detail on the **Pages** shelf and on the Color target. Also, trails are only supported for discrete mark types such as squares, circles, or shapes. They are not supported when the mark type is Automatic.

Pages on dashboards (Tableau Desktop only)

When a dashboard contains multiple views that use the same field on the **Pages** shelf, you can control all of the views with a single page control by selecting the **Synchronized** option. This option is only available on the page control shown on a dashboard.



Additional Shelves, Legends, Cards, and Controls

Some shelves, legends, cards, or controls are only displayed as a result of things that you do as you work with views. For example, the Color legend is only displayed when there is a field on Color.

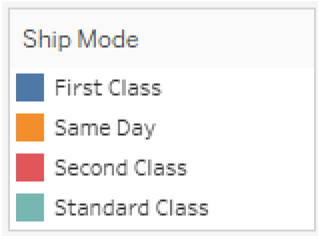


Tableau provides controls for moving or otherwise customizing these elements of the view.

The following list describes each such shelf, legend, card, or control.

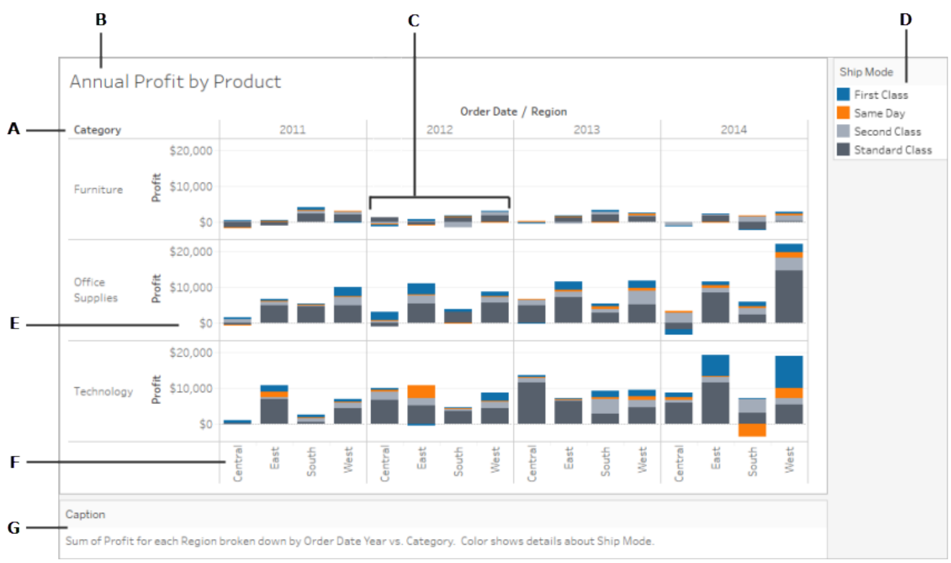
- **Measure Values Shelf**– Measure Values is a special field that always appears in the Data pane and contains all the measures of your data collected into one field. Tableau automatically adds Measure Values to the view when multiple measures are sharing the same axis. When Measure Values is in the view, Tableau displays a Measure Values shelf that shows which measures are being included. You can add measures to or remove measures from this card. For more information, see **Measure Values and Measure Names** on page 1310.
- **Color Legend** – Shows how colors are allocated when there is a field on Color.
- **Shape Legend** – Shows how shapes are allocated when there is a field on Shape.
- **Size Legend** – Shows how sizes are allocated when there is a field on Size.
- **Map Legend** – Shows the legend for the symbols and patterns on a map. The map legend is not available for all map providers.
- **Parameter Controls** – A separate parameter control is available for every parameter in the workbook. For more information, see **Create Parameters** on page 1218.
- **Title** – A title is displayed by default for every view. The default title is the sheet name. Double-click a title (Control-click on a Mac) to edit it.
- **Caption** – Choose **Show caption** from the Worksheet menu to display a caption for the view.
- **Summary Card** – Choose **Show summary** from the Worksheet menu to display a summary card for the view. For more information, see **Summary Card** on page 2727.
- **Page Control** – Provides options for navigating through pages when there is a field on the Pages shelf.

Parts of the View

This section describes the basic elements of views that you can create in Tableau. You can show or hide parts of the view as needed (described below). Every view has a table in some form, which may include rows, columns, headers, axes, panes, cells, and marks. Views can optionally include tooltips, titles, captions, field labels, and legends.

The View area

Data views are displayed in a table on every worksheet. A table is a collection of rows and columns, and consists of the following components: Headers, Axes, Panes, Cells, and Marks. In addition to these, you can choose to show or hide Titles, Captions, Field Labels, and Legends.



A. **Field Labels** on page 255 - The label of a discrete field added to the row or column shelf that describes the members of that field. For example, Category is a discrete field that contains three members; Furniture, Office Supplies and Technology.

B. **Titles** on page 252 - The name that you give your worksheet, dashboard, or story. Titles display automatically for worksheets and stories and you can turn them on to display them in your dashboards.

C. **Marks** on page 249 - The data that represents the intersection of the fields (dimensions and measures) included in your view. Marks can be represented using lines, bars, shapes, maps and so on.

D. **Legends** on page 257 - A key that describes how the data is encoded in your view. For example if you use shapes or colors in your view, the legend describes what each shape or color represents.

E. **Axes** on page 246 - Created when you add a measure (fields that contain quantitative, numerical information) to the view. By default, Tableau generates a continuous axis for this data.

F. **Headers** below - The member name of a field.

G. **Captions** on page 254 - Text that describes the data in the view. Captions can be automatically generated and can be toggled on and off.

Also see information on **Cells** on page 248 and **Panes** on page 248.

Headers

Headers are created when you place a dimension or discrete field on the **Rows** shelf or the **Columns** shelves. The headers show the member names of each field on the shelves. For example, in the view below the column headers show the members of the **Order Date** field and the row headers show the members of the **Sub-Category** field.

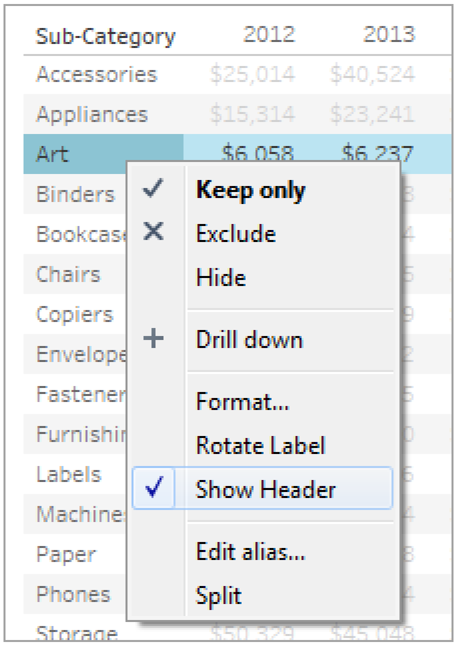
Sheet 1

	Order Date			
Sub-Category	2012	2013	2014	2015
Accessories	$25,014	$40,524	$41,896	$59,946
Appliances	$15,314	$23,241	$26,050	$42,927
Art	$6,058	$6,237	$5,910	$8,914
Binders	$43,488	$37,453	$49,485	$72,986
Bookcases	$20,037	$38,544	$26,275	$30,024
Chairs	$77,242	$71,735	$83,919	$95,554
Copiers	$10,850	$26,179	$49,599	$62,899
Envelopes	$3,856	$4,512	$4,730	$3,379
Fasteners	$661	$545	$960	$858
Furnishings	$13,826	$21,090	$27,874	$28,915
Labels	$2,841	$2,956	$2,827	$3,861
Machines	$62,023	$27,764	$55,907	$43,545
Paper	$14,835	$15,288	$20,638	$27,718
Phones	$77,391	$68,314	$78,660	$105,643
Storage	$50,329	$45,048	$58,632	$69,834
Supplies	$14,394	$1,952	$14,278	$16,049
Tables	$46,088	$39,150	$60,833	$60,894

You can show and hide row and column headers at anytime.

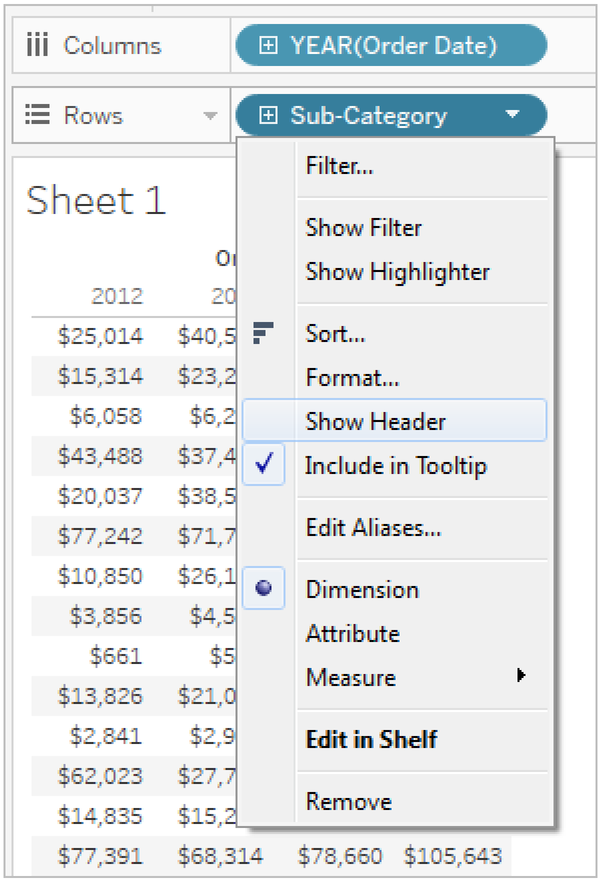
To hide headers:

- Right-click (control-click on Mac) the headers in the view and select **Show Header**.

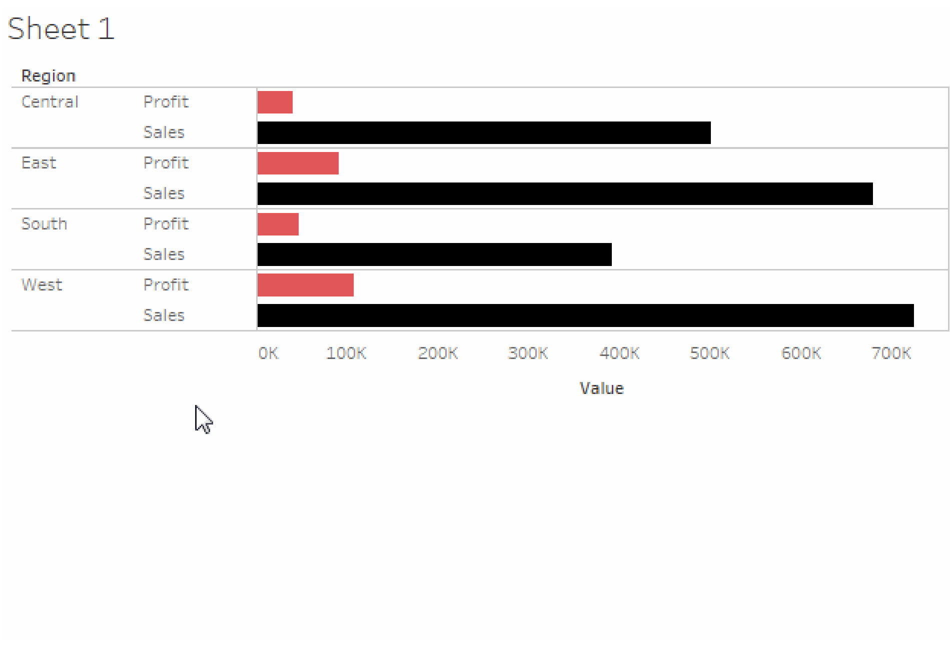


To show headers:

- Select the field in the view whose headers you want to show and select **Show Header** on the field menu.

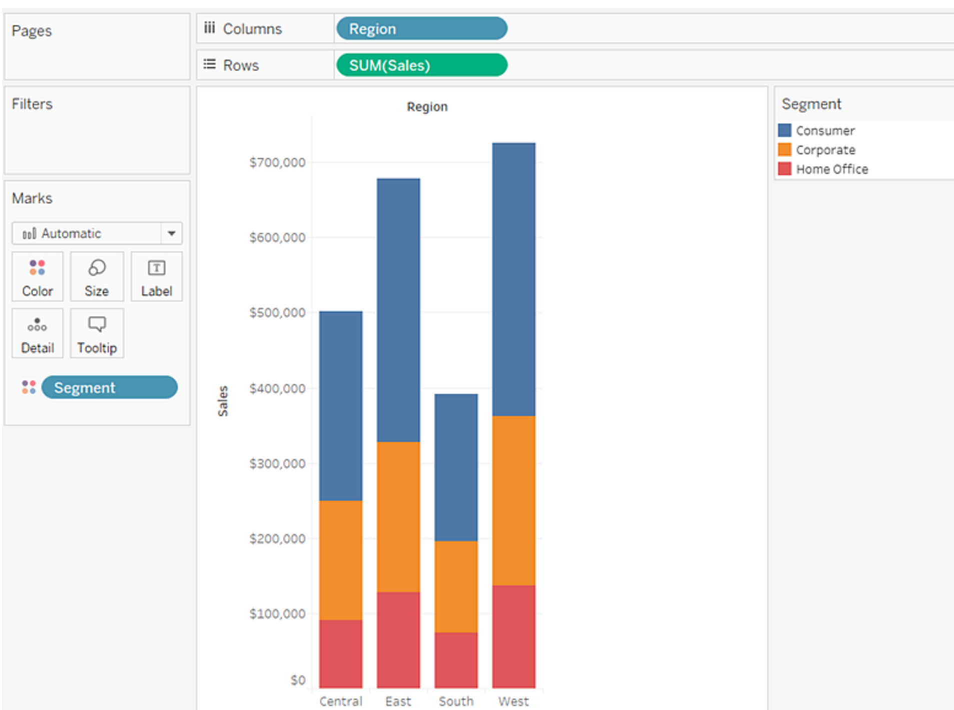


Hiding headers can be useful when you are working with multiple measures. For example, the view below shows both the sales and profit for each region along a single axis. You can see the view looks cluttered with the Measure Names headers showing. Because Measure Names is also indicated by the mark color, you can hide the excess headers to clean up the view.



Axes

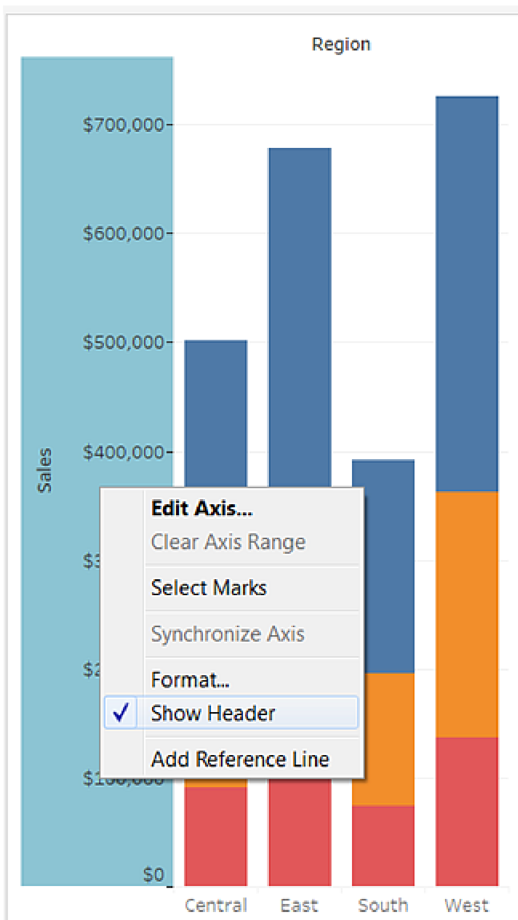
Axes are created when you place a measure or continuous field on the **Rows** or **Columns** shelves. By default, the values of the measure field are displayed along a continuous axis.



You can show and hide axes at any time.

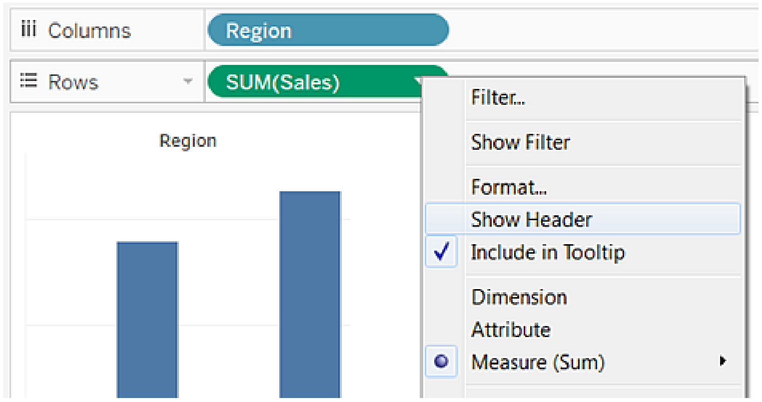
To hide axes:

- Right-click (control-click on Mac) the axis in the view and select **Show Header** to clear the check mark next to this option.



To show axes:

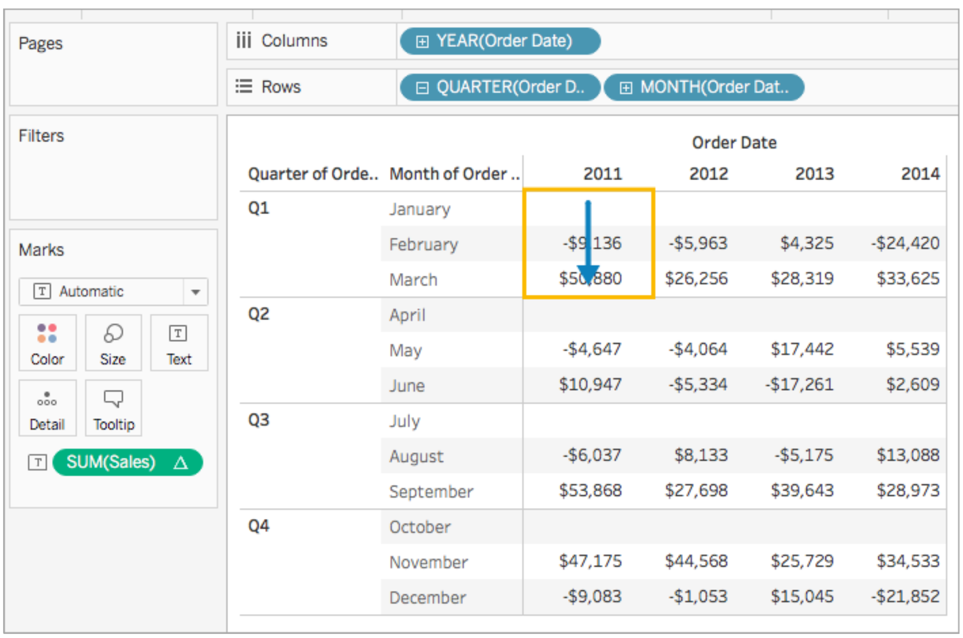
- Right-click (control-click on Mac) the measure in the view whose axis you want to show and select **Show Header** on the field menu.



Panes

A pane is defined by the intersection of fields on the rows and columns shelves.

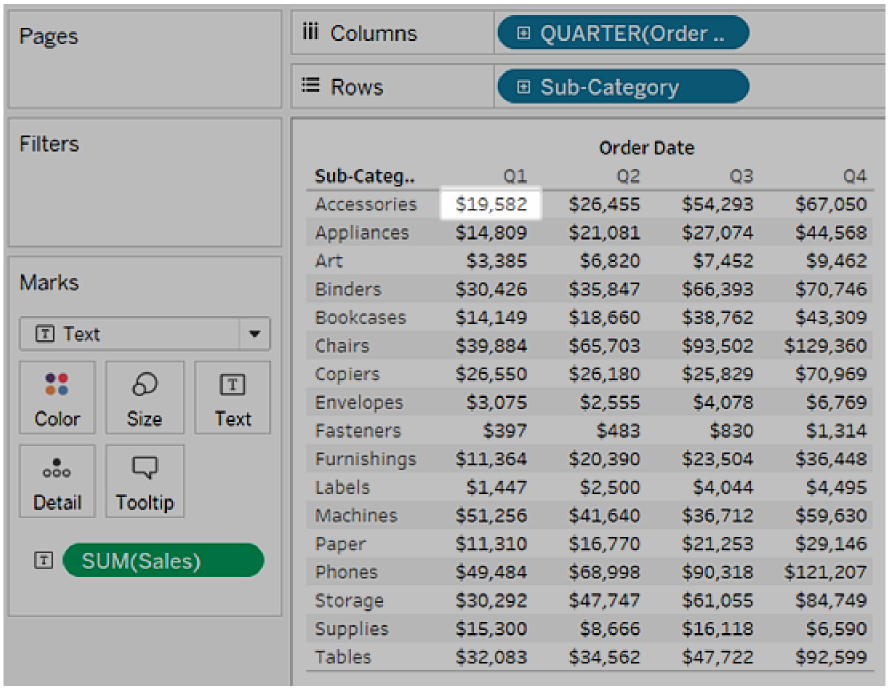
In a table calculation, this is seen as one or more cells that belong to the same field, which are computed down or across according to the calculation, as in the example below:



For more information, see **Transform Values with Table Calculations** on page 2406.

Cells

Cells are the basic components of any table you can create in Tableau, defined by the intersection of a row and a column. For example, in a text table, a cell is where the text is displayed, as shown in the view below:

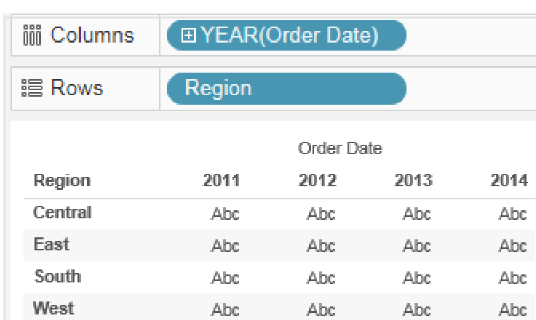


Sub-Categ..	Q1	Q2	Q3	Q4
Accessories	$19,582	$26,455	$54,293	$67,050
Appliances	$14,809	$21,081	$27,074	$44,568
Art	$3,385	$6,820	$7,452	$9,462
Binders	$30,426	$35,847	$66,393	$70,746
Bookcases	$14,149	$18,660	$38,762	$43,309
Chairs	$39,884	$65,703	$93,502	$129,360
Copiers	$26,550	$26,180	$25,829	$70,969
Envelopes	$3,075	$2,555	$4,078	$6,769
Fasteners	$397	$483	$830	$1,314
Furnishings	$11,364	$20,390	$23,504	$36,448
Labels	$1,447	$2,500	$4,044	$4,495
Machines	$51,256	$41,640	$36,712	$59,630
Paper	$11,310	$16,770	$21,253	$29,146
Phones	$49,484	$68,998	$90,318	$121,207
Storage	$30,292	$47,747	$61,055	$84,749
Supplies	$15,300	$8,666	$16,118	$6,590
Tables	$32,083	$34,562	$47,722	$92,599

Marks

When you drag fields to the view, the data are displayed using marks. Each mark represents the intersection of all of the dimensions in the view.

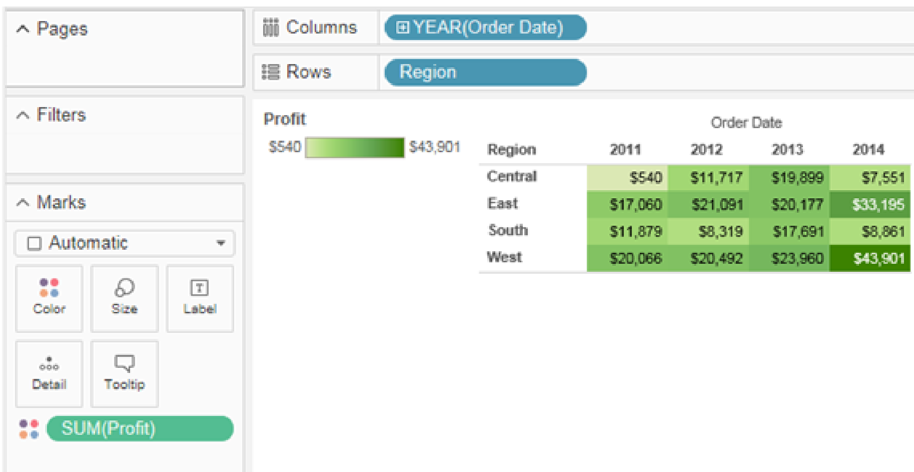
For example, in a view with **Region** and **Year** dimensions, there is a mark for every combination of those two dimensions (East 2011, East 2012, West 2011, West 2012, etc.). In this case, the mark type is set to Text, so the **Abc** represents the location where the value for the text mark will appear—once a measure such as **Sales** is added to the view.



Region	2011	2012	2013	2014
Central	Abc	Abc	Abc	Abc
East	Abc	Abc	Abc	Abc
South	Abc	Abc	Abc	Abc
West	Abc	Abc	Abc	Abc

For details on creating and customizing marks, see **Marks card** on page 227, **Control the Appearance of Marks in the View** on page 1356, **Change the Type of Mark in the View** on page 1332, **Stack Marks** on page 1392

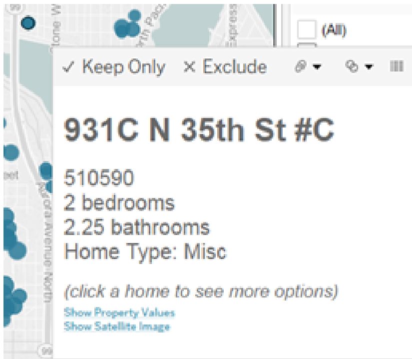
Marks can be displayed in many different ways including lines, shapes, bars, maps, and so on. You can show additional information about the data using mark properties such as color, size, shape, labels, etc. The type of mark you use and the mark properties are controlled by the Marks card. Drag fields to the Marks card to show more data. For example, the same view above is shown again below but this time with **Profit** on Color. With this additional information, it is clear that the West region had the highest profit in 2014.



Control the marks in the view using the Marks card. Use the drop-down menu to specify the type of mark to show. Drag fields to the Marks card and use the drop-down controls to add more information to the view and control the color, shape, size, labels, and number of marks in the view.

Tooltips

Tooltips are additional data details that display when you hover over one or more marks in the view. When you select one or more marks and hover, tooltips also include options to filter marks (exclude or keep only), display marks that have the same values, create groups, create sets, or display the underlying data. If you don't want users to be able to access tooltip commands, you can disable them.



For details on formatting tooltips and other tooltip settings, also see **Format Titles, Captions, Tooltips, and Legends** on page 2937, **Add tooltips to marks** on page 1366, and **Create Views in Tooltips (Viz in Tooltip)** on page 1475. For details on analysis options in tooltips, see **Explore and Inspect Data in a View** on page 2707

Tooltip command buttons for exploring data in a viz

The top of the tooltip lists commands for filtering data, creating a group, sorting the selection, and view the underlying data. For example, you can use the tooltip to quickly remove an outlier in a scatter plot. Each of the commands are described below.

To see tooltip commands, hover over a mark and then keep the cursor still. The

- **Keep Only** - creates a filter that removes all other data. See **Select to keep or exclude data points in your view** on page 1412 to learn more.
- **Exclude** - creates a filter that removes the selected data. See **Select to keep or exclude data points in your view** on page 1412 to learn more.
- **Group Members** - creates a group based on the selection. If the selection contains multiple dimensions, you can group on one dimension or all dimensions. See **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1184 to learn more.
- **Create Set** (Tableau Desktop only) - creates a new set containing the selected members. You can create a new set or add members to an existing set. See **Create Sets** on page 1186 to learn more.
- **View Data** - opens a window displaying the data. You can view the summarized data or the underlying data. See **View Underlying Data** on page 2728 to learn more.

These commands are visible by default. You can disable the commands in the **Edit Tooltip** dialog box by deselecting **Include command buttons**. Doing so will also hide the aggregation summary if multiple marks are selected.

Disable tooltip commands

If you don't want users to be able to access tooltip commands, you can disable them.

1. Click Tooltip on the Marks card or select **Worksheet** >**Tooltip**.
2. In the Edit Tooltip dialog box, clear the **Include command buttons** check box.

Tooltip settings apply to the active worksheet and can be different for each sheet in the workbook.

Body text and markup in tooltips

The body of a tooltip contains details about a specific mark or a selection of multiple marks. For example, in a bar chart showing sales by region, the tooltip body may include the actual sales amount and the region name. The default tooltip is based on the fields used in the view. You can customize what is shown in the tooltip by dragging fields to **Tooltip** on the Marks card.

To customize a tooltip and its formatting, click **Tooltip** on the Marks card.

Alternatively, you can select **Worksheet > Tooltip**.

Action links

If the sheet has any actions, the action links are listed below the body of the tooltip. An action adds context and interactivity to your data through filters, highlighting, and links to external resources. See **Actions** on page 1491 to learn more about adding actions to your workbook.

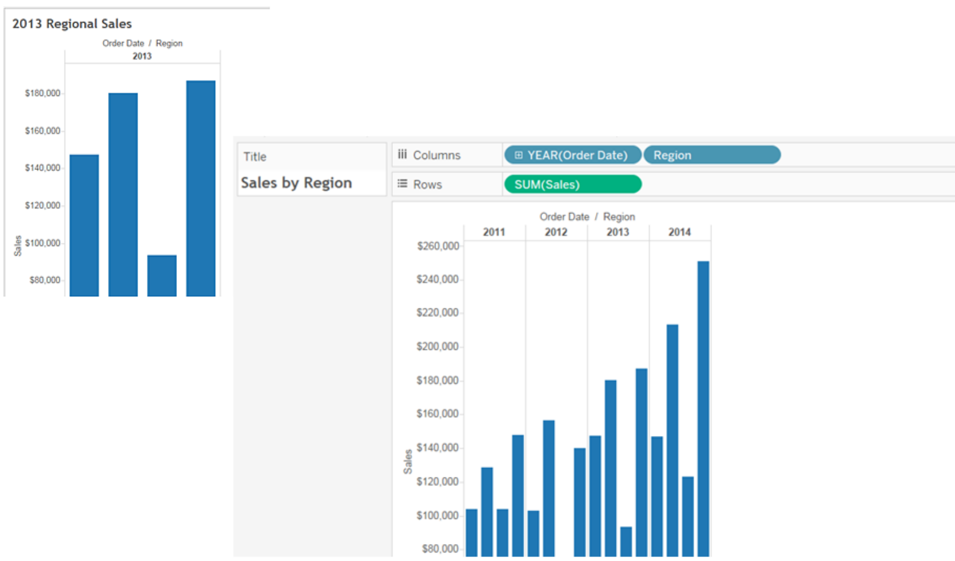
Titles

You can show titles on any worksheet, dashboard, or story. For worksheets and stories, a title is displayed by default, but you can remove it. For dashboards, you can add a title. By default, the title is the name of the sheet, but you can edit the title to change the text and include dynamic values such as page number and sheet name. For more information about how to format titles, see **Format Titles, Captions, Tooltips, and Legends** on page 2937.

Show and hide titles in worksheets

Titles are shown by default for worksheets and are included as part of the worksheet, shown at the top of the view. You can move the title to the sides or the bottom of the view. However, when you move the title from the top of the view, it becomes a **Title** card and displays like any other card in the view.

> **Note:** If you move a title from the top position and then hide it; when you show the title again, it appears back at the top of the worksheet in its default position.



To show or hide titles in a worksheet

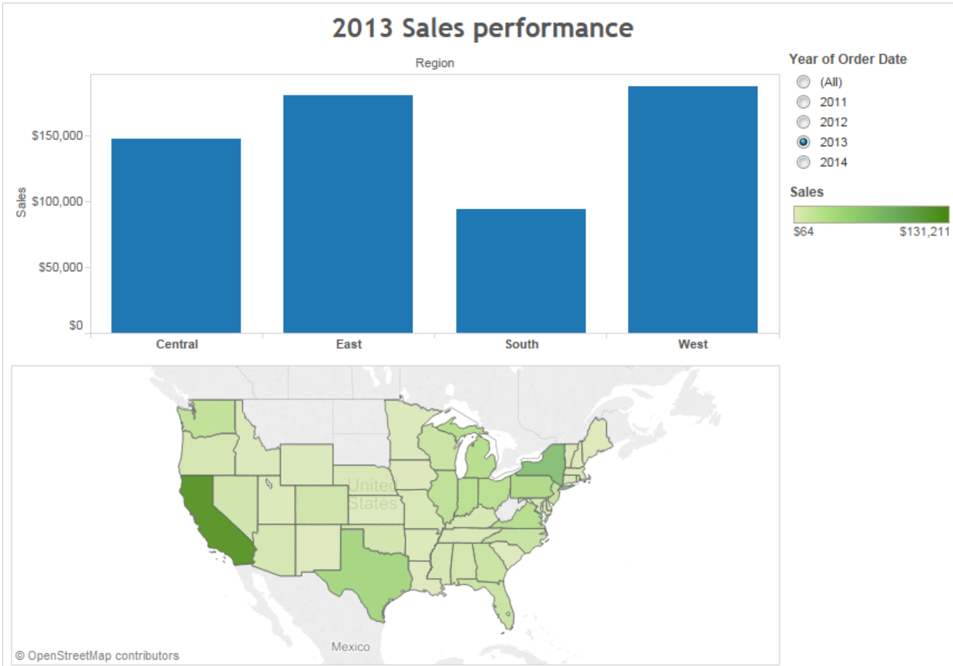
- From the toolbar menu, click **Worksheet** > **Show Title**.
- On the toolbar, click the drop-down arrow on the **Show/Hide Cards** button and select **Title** from the context menu.

  Toggle the check mark on or off to show or hide the title.

Show and hide titles in dashboards

You can turn on titles for dashboards. The title appears as part of the dashboard.

To show or hide titles on a dashboard, from the toolbar menu, select **Dashboard** > **Show Title**.



When you add worksheets to the dashboard, the title of the worksheet automatically shows, even if you turned off the title on the worksheet itself. To turn off the title for the worksheet on the dashboard, do the following steps:

1. In the dashboard, select the worksheet to highlight it.
2. In the top right corner of the highlighted worksheet, click the drop-down arrow and select **Title** in the context menu to clear the check mark.

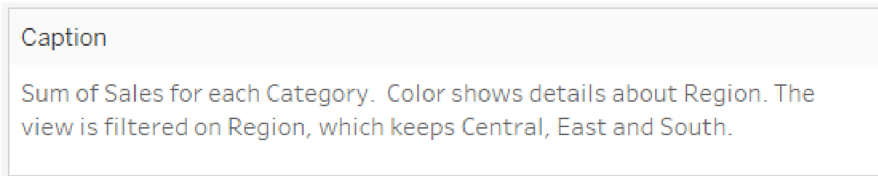
Show and hide titles in stories

Story titles are displayed by default. To toggle story titles on or off, in the top menu, select **Story** > **Show Title** to add or remove the check mark.

Captions

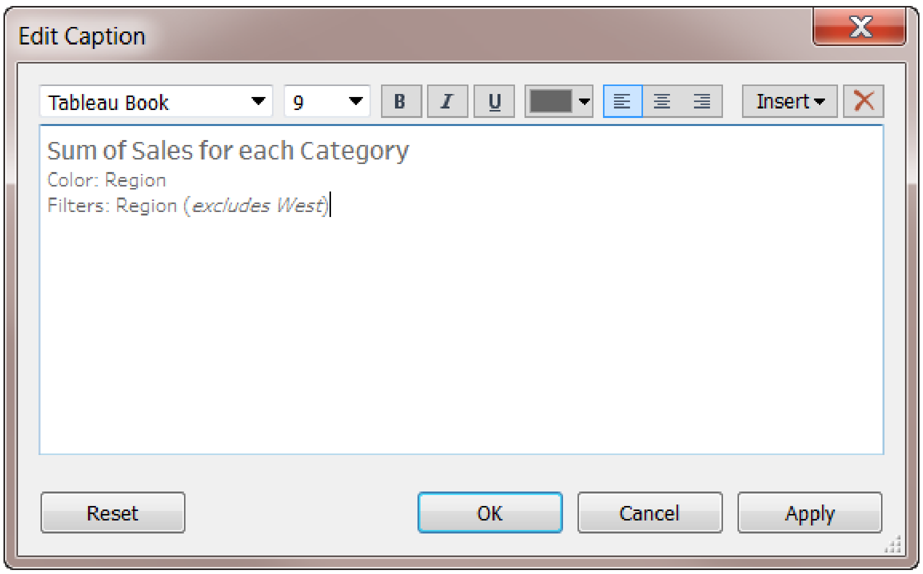
All views can have a caption that is either automatically generated or manually created. The caption is displayed on the Caption card.

To show a caption in a worksheet, select it on the **Show/Hide Cards** toolbar menu  or select **Worksheet** > **Show Caption**.



The caption is automatically generated by default.

To edit the caption, double-click the Caption area in the view. In the **Edit Caption** dialog box, you can use change the font, size, color, and alignment and style.



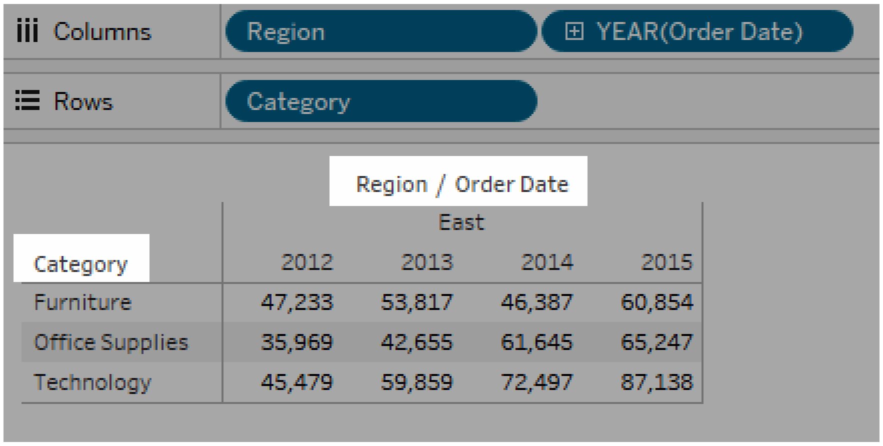
Click the **Insert** menu to add automatic text such as page number, sheet name, and field and parameter values.

The caption can optionally be included when printing, printing to PDF, and publishing to Tableau Server. When you export the view as an image to another application like Microsoft PowerPoint, you can optionally include the caption.

Field Labels

Placing discrete fields on the rows and column shelves creates headers in the view that display the members of the field. For example, if you place a field containing products on the rows shelf, each product name is shown as row headers.

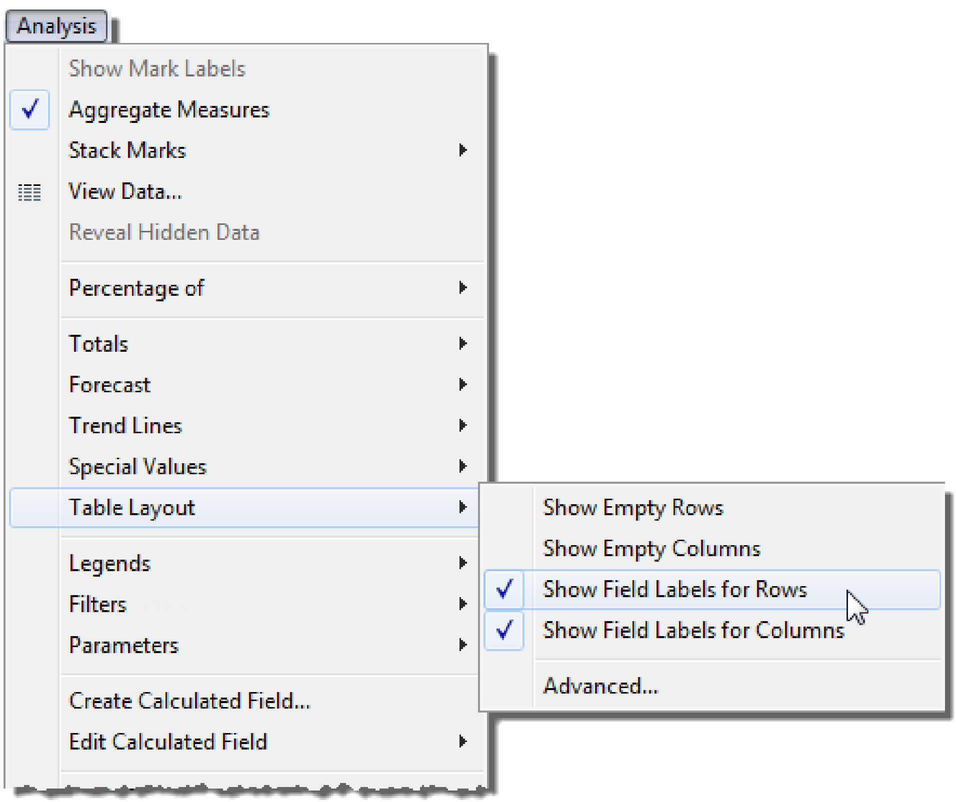
In addition to showing these headers, you can show field labels, which are labels for the headers. In this example, the rows are labeled **Category**, to indicate that the discrete category names are members of the **Category** field.



Field labels apply only to discrete fields. When you add continuous fields to the view, Tableau creates an axis. The axis is labeled with a header.

By default, field labels are shown.

To hide or show field labels, select **Analysis** > **Table Layout** > **Show Field Labels for Rows** or **Show Field Labels for Columns**.

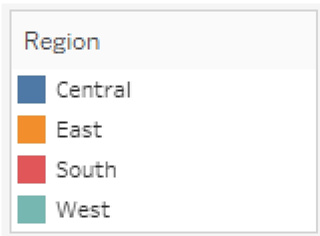


You can format the fonts, alignment, shading, and separators for field labels.

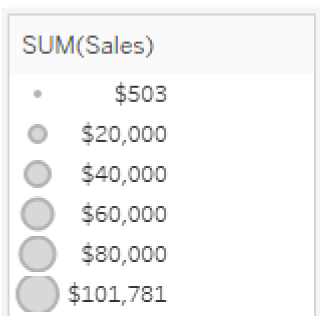
Legends

When you add fields to Color, Size, and Shape on the Marks card, a legend displays to indicate how the view is encoded with relation to your data.

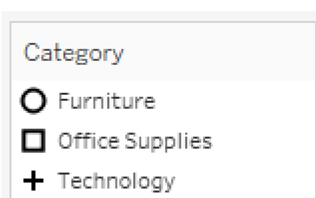
Color Legend



Size Legend



Shape Legend



Not only do legends help you understand encodings, you can also use legends to sort, filter, and highlight specific sets of data. For more information, see **Legend Highlighting** on page 1499.

Measure values and color legends

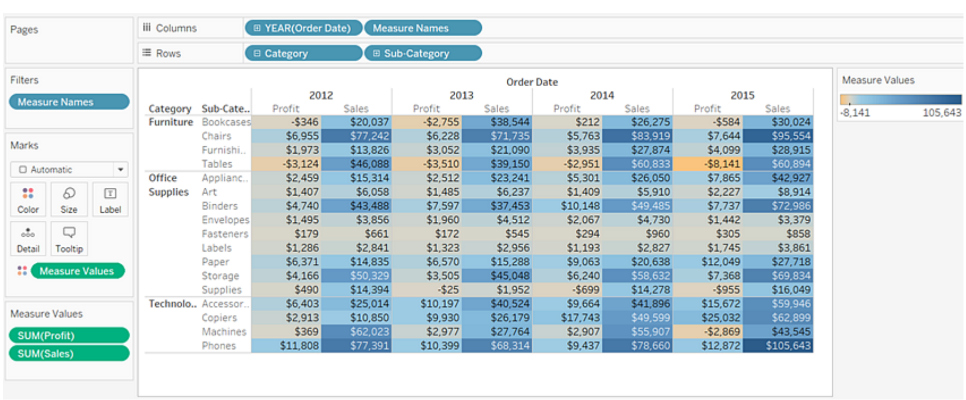
If you include the **Measure Values** and **Measure names** fields in your views, you can create either a single combined color legend or separate color legends for your measures. If you drag the **Measure Values** field to Color on the Marks card, by default Tableau creates a single color legend that applies one color palette to all marks in the view. If you want to differentiate certain measures in the view you can create separate color legends for the measures and assign a unique color palette to each legend.

For more information about Measure Values and Measure Names, see **Measure Values and Measure Names** on page 1310.

The following example shows how to create separate color legends. This example uses the Sample Superstore data set.

1. Connect to the Sample-Superstore data set.
2. From the Data pane:
    - Drag **Order Date** to the Columns shelf and **Category** and **Sub-Category** to the Rows shelf.
    - Drag **Measure Names** to the columns shelf and drop it to the right of **Order Date**.
    - Drag **Measure Values** to Color on the Marks card.
3. In the Measure Values card, drag measures off the card so that you keep **SUM(Sales)**, and **SUM(Profit)** only.
4. Click **Label** on the Marks card and select **Show mark labels** to show the measure values in the view.

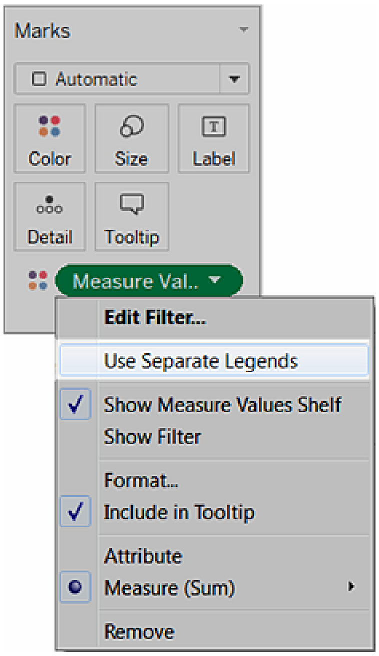
    When you drag **Measure Values** to Color on the Marks card, Tableau creates a single color legend and adds it to the view. Your view should look something like this.



5. To create separate legends for each measure on the view, click the drop-down arrow on

the **Measure Values** field in the Marks card and select **Use Separate Legends** from the context menu.

Tableau Desktop version



Web version

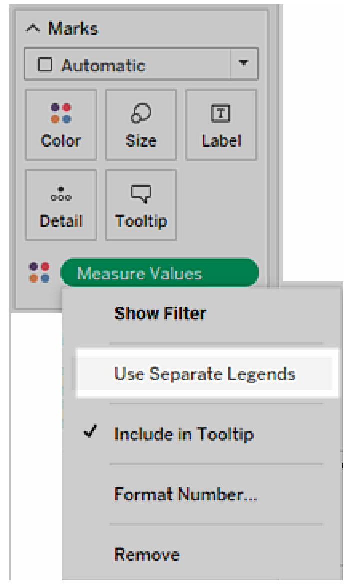
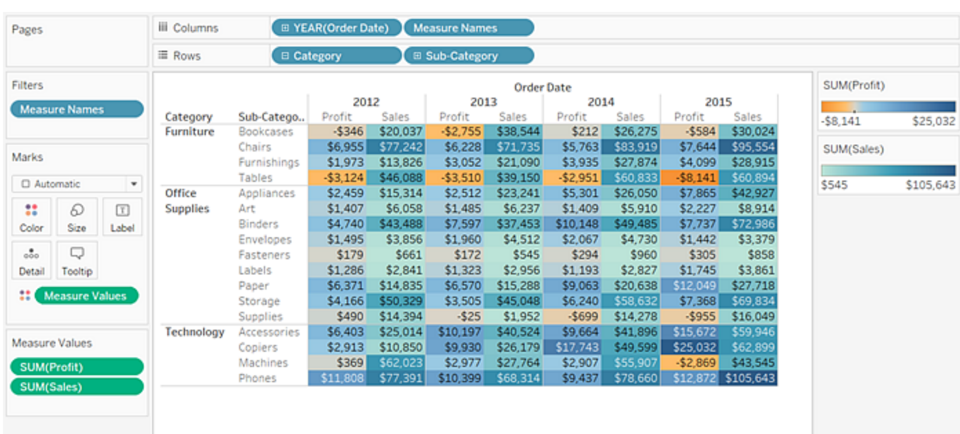


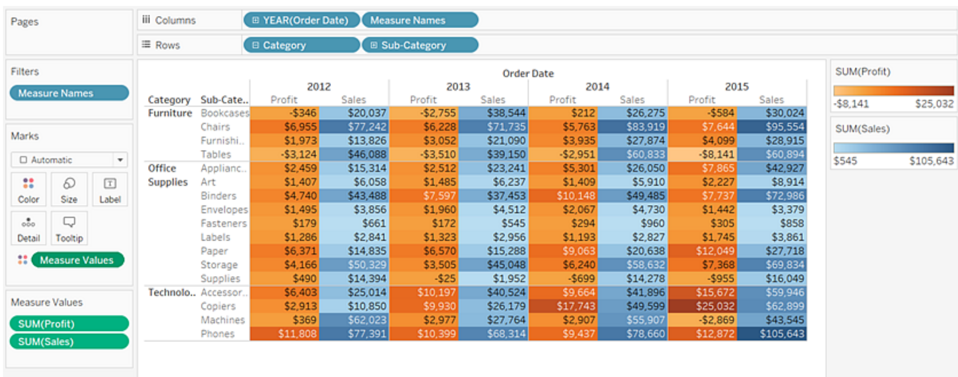
Tableau creates an individual color legend for each measure in the view using the default color palette.



To assign a different color palette to the color legend, do one of the following:

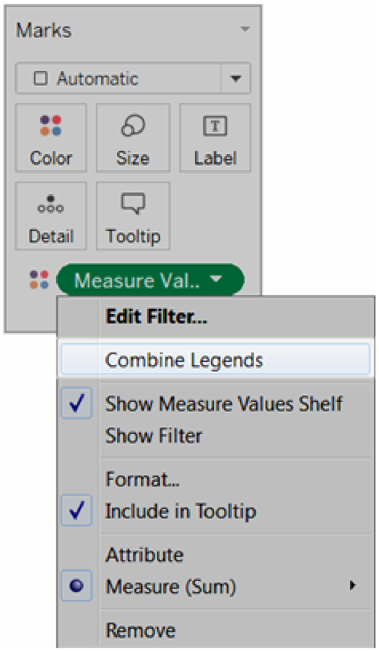
- In Tableau Desktop, click the drop-down arrow in the top right corner for each color legend and select **Edit Colors**. Then select a color from the Palette drop-down list.
- In Tableau Server or Tableau Online, click the drop-down arrow in the top right corner for each color legend. Then select a color from the Palette drop-down list.

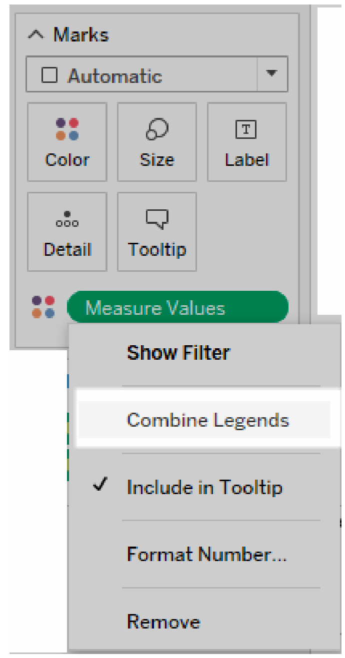
Your view might look something like the following example:



6. To combine the separate legends back to a single legend, click the drop-down arrow on the **Measure Values** field on the Marks card and select **Combine Legends** from the context menu.

Tableau Desktop version	**Web version**





Workbooks and Sheets

Tableau uses a workbook and sheet file structure, much like Microsoft Excel. A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story.

- A **worksheet** contains a single view along with shelves, cards, legends, and the Data and Analytics panes in its side bar. For details on the worksheet workspace, see **The Tableau Workspace** on page 192.
- A **dashboard** is a collection of views from multiple worksheets. The Dashboard and Layout panes are available in its side bar. For more details about creating dashboards, see **Dashboards** on page 2735.
- A **story** contains a sequence of worksheets or dashboards that work together to convey information. The Story and Layout panes are available in its side bar. For more details about creating stories, see **Stories** on page 2845.

For details on hiding or showing sheets in a workbook or Viz in Tooltip, see **Manage Sheets in Dashboards and Stories** on page 2834 and **Hide or show a Viz in Tooltip worksheet** on page 1483. For details on organizing sheets, see **Navigate and Organize Sheets** on page 269. For details on creating and opening workbooks, see **Create or Open Workbooks** on page 265.

About sheets

Each workbook can contain different types of sheets: views (also known as worksheets), dashboards, and stories.

- A worksheet is where you build views of your data by dragging and dropping fields onto shelves.
- A dashboard is a combination of several views that you can arrange for presentation or to monitor.
- A story is a sequence of views or dashboards that work together to convey information.

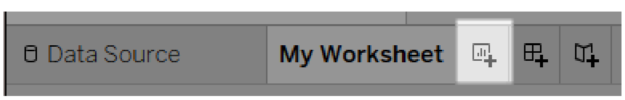
The sheets display along the bottom of the workbook as tabs. In this section you'll learn how to create, open, duplicate, hide, and delete sheets. You'll also learn how to organize sheets in a workbook.

Within a workbook, you can create new sheets, clear an entire worksheet, duplicate sheets, hide or show a worksheet, and delete a sheet. Tableau has several ways to view and organize the sheets in your workbook.

Create new worksheets, dashboards, or stories

There are several ways to create new sheets in a workbook, dashboard, or a story. You can create as many sheets in a workbook as you want.

To create a new worksheet, dashboard, or story, click the **New Worksheet**, **New Dashboard,** or **New Story** button at the bottom of the workbook.



The **New Worksheet** button.

To rename a new worksheet, dashboard, or story, right-click (Ctrl-click on a Mac) the tab and then select the **Rename** command.

More ways to create new worksheets

Create a new worksheet by doing one of the following:

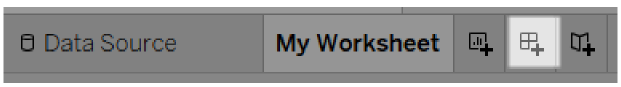
- Select **Worksheet** > **New Worksheet**.
- Right-click any open tab in the workbook, and select **New Worksheet** from the menu.

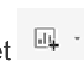
- On the toolbar, click the drop-down arrow on the **New Worksheet**  button and then select **New Worksheet**.
- Press Ctrl + M on your keyboard (Command-M on a Mac).

More ways to create new dashboards

Create a new dashboard by doing one of the following:

- Select **Dashboard** > **New Dashboard**.
- Click the **New Dashboard** button at the bottom of the workbook.

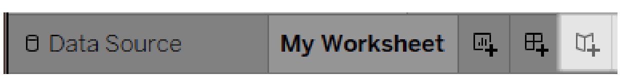


- Right-click on any open tab in the workbook, and select **New Dashboard** from the menu.
- On the toolbar, click the drop-down arrow on the New Worksheet  button and then select **New Dashboard**.

More ways to create new stories

Create a new story by doing one of the following:

- Select **Story** > **New Story**.
- Click the **New Story** button at the bottom of the workbook.



- Right-click on any open tab in the workbook, and select **New Story** from the menu.
- On the toolbar, click the drop-down arrow on the **New Worksheet**  button and then select **New Story**.

Undo, redo, or clear sheets

Every Tableau workbook contains a history of steps you have performed on the worksheets, dashboards, and stories in that workbook for the current work session.

To move backward through the history, click **Undo** ← on the toolbar or press Ctrl + Z on your keyboard (Command-Z on a Mac).

To move forward through the history, click **Redo** → on the toolbar or press Ctrl + Y (Command-Y on a Mac) on your keyboard.

To remove all fields, formatting, sizing, axis ranges, filters, sorts, and context filters in the sheet, click **Clear Sheet** on the toolbar.

To clear specific aspects of the view, use the Clear Sheet drop-down menu.

> **Note:** Using the clear commands on the toolbar does not clear the history. If you decide that you didn't want to clear the sheet, click the **Undo** button.

Duplicate a sheet

When you want to use an existing sheet as a jumping off point for more exploration, you can duplicate that sheet. The duplicated sheet contains all of the same fields and settings as a starting point for new analysis.

Duplicate creates a new version of a worksheet, dashboard, or story you can modify without effecting the original sheet.

To duplicate the active sheet, right-click the sheet tab (control-click on Mac) and select **Duplicate**.

> **Note**: When you duplicate a dashboard, a new version of the dashboard is created, but it still references the original worksheets that were used to create the dashboard.

Duplicate as Crosstab

A crosstab (sometimes referred to as a Pivot Table) is a table that summarizes data in rows and columns of text. It is a convenient way to display the numbers associated with the data view.

To create a new cross-tab sheet based on the data in the current sheet, right-click the sheet tab (control-click on Mac) and select **Duplicate as Crosstab**. Or select **Worksheet** > **Duplicate as Crosstab**.

This command inserts a new worksheet into your workbook and populates the sheet with a cross-tab view of the data from the original worksheet. Dashboards and stories cannot be duplicated as crosstabs.

Rename sheets

To rename the active sheet, right-click (control-click on Mac) the sheet in the sheet tab along the bottom of the workbook, and then click **Rename Sheet** .

Or, double-click the name of the sheet in a sheet tab, type a new name, and then click **Enter**.

See the underlying data in a sheet

To see the numbers behind the marks in your view

Hover over or click a mark to display the associated data in a tooltip. With the tooltip open, click the **View Data** command at the top of the tooltip to view underlying data.

Right-click the view (with no marks selected), and then click View Data.

You can copy and paste the data into Excel or another application. To export the underlying data, click **Export All**.

Delete sheets

Deleting a sheet removes it from the workbook. There must always be at least one worksheet in a workbook.

To delete the active sheet, right-click (control-click on Mac) the sheet in the sheet tab along the bottom of the workbook, and then click **Delete** .

Worksheets used in a dashboard or story cannot be deleted, but they can be hidden.

A worksheet used as Viz in Tooltip can be hidden or deleted.

Create or Open Workbooks

A workbook contains one or more sheets, each of which can be a worksheet, dashboard, or story. You can use workbooks to organize, save, share, and publish your results.

- **Create or open a workbook** on the next page
- **Open a workbook that has an unsupported data connection on the Mac** on the

next page

- **Open a published workbook from the server** on the facing page

Create or open a workbook

When you open Tableau, a new, blank workbook opens automatically.

To create a new workbook, select **File** > **New**

To open an existing workbook, click the thumbnail image of the workbook on the start page.

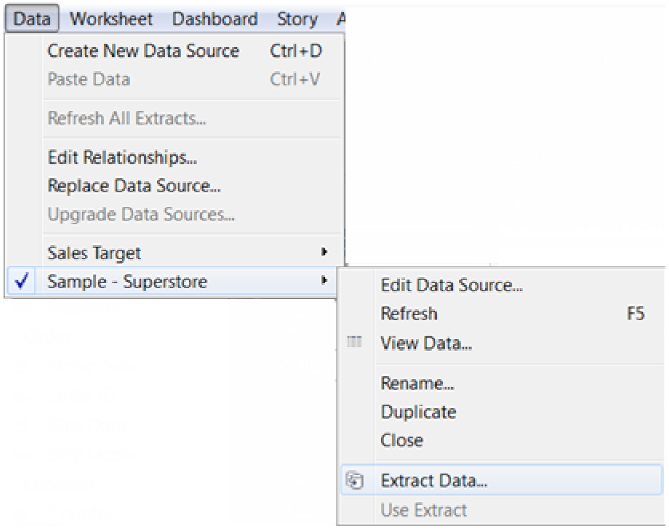
You can open multiple workbooks simultaneously. Each workbook appears in its own window.

> **Note:** When you open multiple workbooks in Tableau Desktop on the Mac, multiple instances of the application are created, each with its own icon in the Dock. This differs from typical Mac application behavior, where one instance of the application handles all open documents.

Open a workbook that has an unsupported data connection on the Mac

When working in Tableau Desktop on the Mac, you might need to open a Windows workbook that contains data sources that are not supported in Tableau Desktop on the Mac. To do so, follow the steps below:

1. In Tableau Desktop on Windows, save the workbook as an extract. To create and save an extract, on the **Data** menu, select a data source, and then select **Extract Data**.



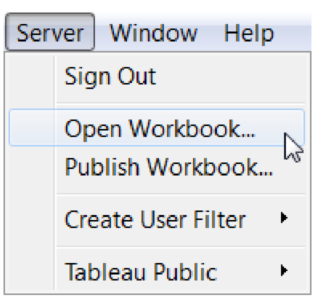
2. Open the extract file (.hyper) you saved in Tableau Desktop on your Mac.

Open a published workbook from the server

If you have been granted the **Download/Web Save As** permission for a published workbook, you can use Tableau Desktop to open the workbook from the server. When you open a workbook from the server and make changes, you can save it to your computer or, if you have been allowed the **Write/Web Save** permission, you can save the changes directly on the server.

To open a workbook from the server

1. Select **Server** > **Open Workbook**.

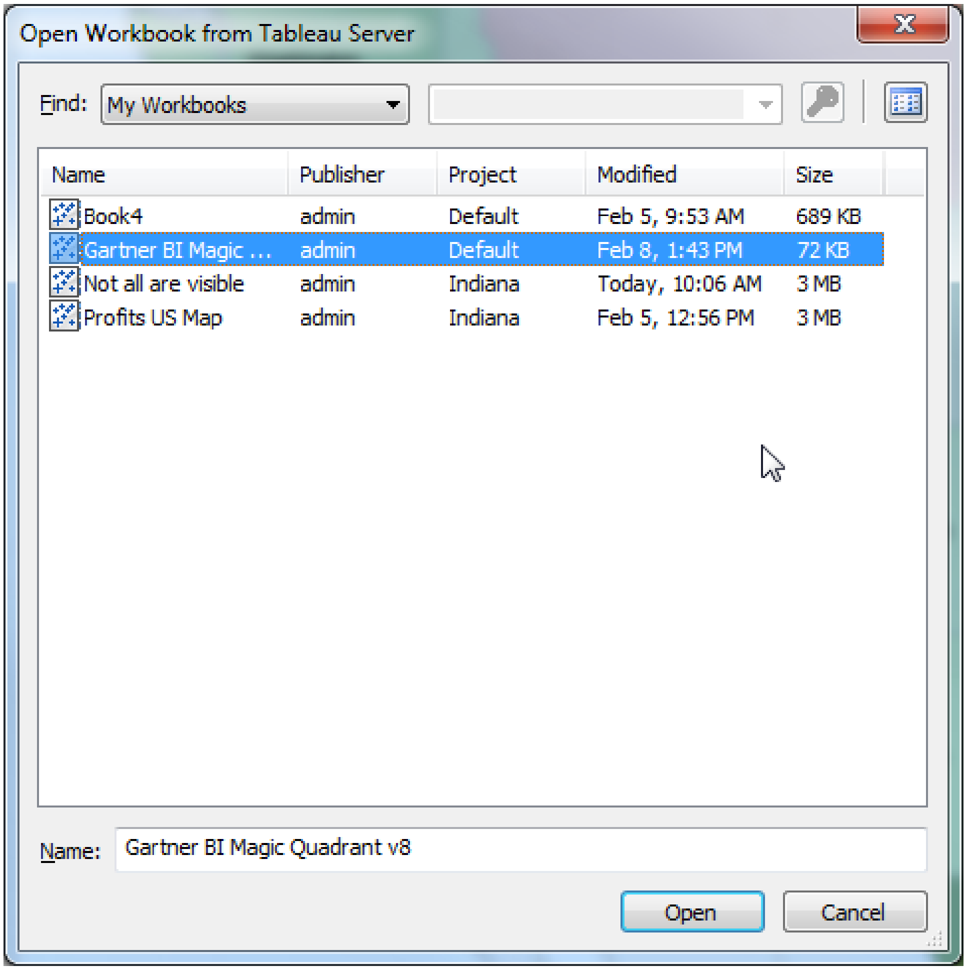


2. If you are not already signed in to Tableau Server or Tableau Online, do so at the prompt.

   If you're not sure how to sign in to the server you use, see **Sign in to Tableau Server or Tableau Online** on page 364.

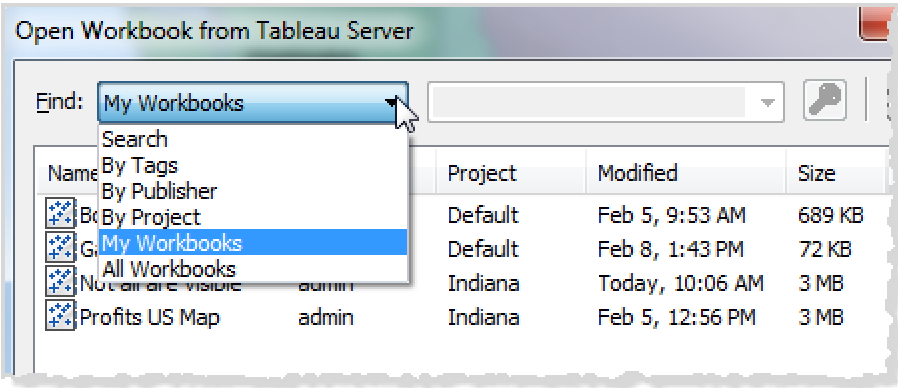
3. In the Open Workbook from Tableau Server dialog box, select the workbook you want to

open, and then click **Open**.



Tip: You can use the **Find** drop-down list or the Search box to browse or search for the workbook you want.

You can search all workbooks on the server or find by tags, publisher, project, or workbooks that you published.



Navigate and Organize Sheets

You can navigate and organize sheets in a workbook using the sheet tabs, the filmstrips, or the sheet sorter.

Options for navigating, viewing, and organizing sheets

There are three ways to navigate and view the sheets in a workbook:

- Tabs at the bottom of the workbook
- In the filmstrip view
- In the sheet sorter view

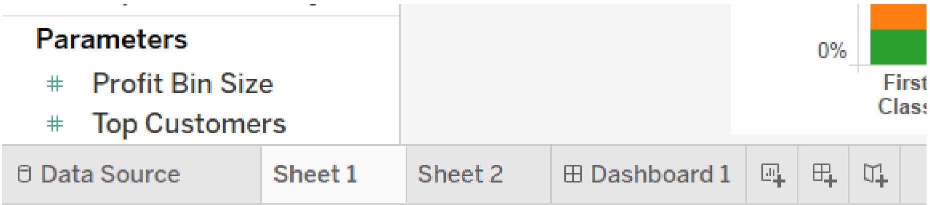
The tabs are useful when you want to quickly navigate between a small number of sheets. If your workbook has a large number of sheets, you can use the sheet sorter to easily navigate them all.

You can also drag and drop to do the following:

- Reorder the sheets
- Create new sheets
- Duplicate or delete existing sheets from any of the views

Navigate with sheet tabs

Each sheet is represented as a tab along the bottom of the workbook. Select any tab to open the corresponding worksheet.



In the bottom right corner of the application window, there are several controls that you can use to advance through each sheet or quickly jump to the first or last sheet in the workbook. These controls are only available when there are too many sheet tabs to show across the bottom of the application window.

You can also navigate between sheets using the window menu or move through the multiple worksheets by pressing the left or right arrow keys on your keyboard.

To navigate through multiple worksheets, select a worksheet tab at the bottom of the workbook first.

> **Note:** Navigating within a story is a different matter—there, you use the Navigator to move between story points. See **Create a Story** on page 2852.

Show thumbnails with the filmstrip

Similar to the sheet tabs, the filmstrip displays along the bottom of the workbook. However, instead of just sheet names, the filmstrip also shows a thumbnail image of each sheet. The filmstrip is useful when you are using Tableau to present your analysis and works well when you are working in Presentation mode.

Open the filmstrip by clicking the **Filmstrip** button on the status bar(bottom right corner) of the workbook. Just like with the tabs, select the thumbnail image for the sheet you want to open. You can right-click the images to specify commands that apply to each sheet.

> **Note:** When viewed from a Retina display, workbooks that are created on a standard resolution device will show only the first thumbnail in the filmstrip in high resolution. To

display all thumbnails in the filmstrip in high resolution, resave the workbook on a computer with Retina display.

Manage many sheets with sheet sorter

The sheet sorter shows all sheets in a workbook as thumbnail images on a single page. The sheet sorter is useful when you have a large number of sheets in a workbook. Open the sheet sorter by clicking the sheet sorter button on the status bar (bottom right corner of the workbook).

Right-click a sheet (Control-click on the Mac) to see these commands. You can also right-click (Control-click on the Mac) to refresh the thumbnail image of a particular sheet or **Refresh All Thumbnails** at once. If you have a large number of sheets, refreshing your view can take some time.

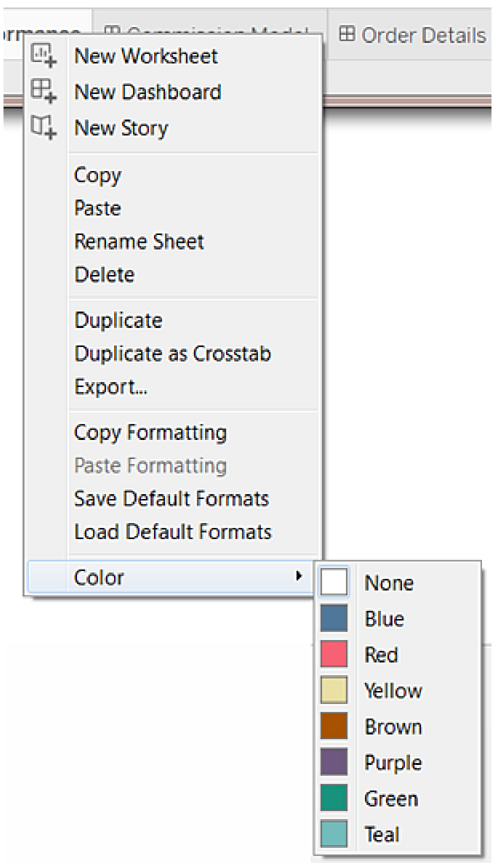
Sheet commands

Use sheet commands to manage and organize your worksheets. For example you can create new sheets, duplicate sheets, copy formatting, apply color, or delete the sheet entirely.

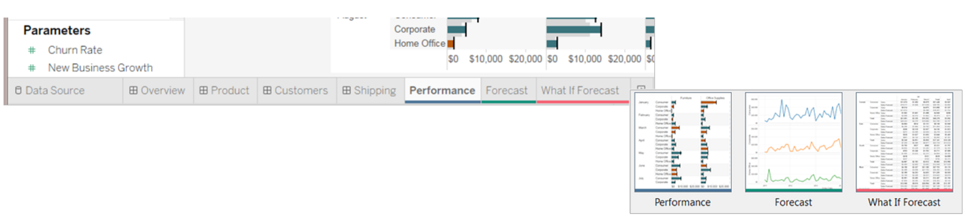
You can access sheet commands on the right-click menu (Control-click on a Mac) in the worksheet, sheet sorter, or the filmstrip view. To apply commands to multiple sheets at once, press the Ctrl key (Shift key on a Mac), and then select the sheets.

To make it easier to identify and group sheets, you can assign a color to sheets. You can select from seven different colors. Selecting **None** clears the color.

To assign a color to sheets, select one or more sheets, right-click the sheets (Control-click on the Mac), select **Color**, and then pick a color.



The color strip appears on the bottom of the tab or sheet.

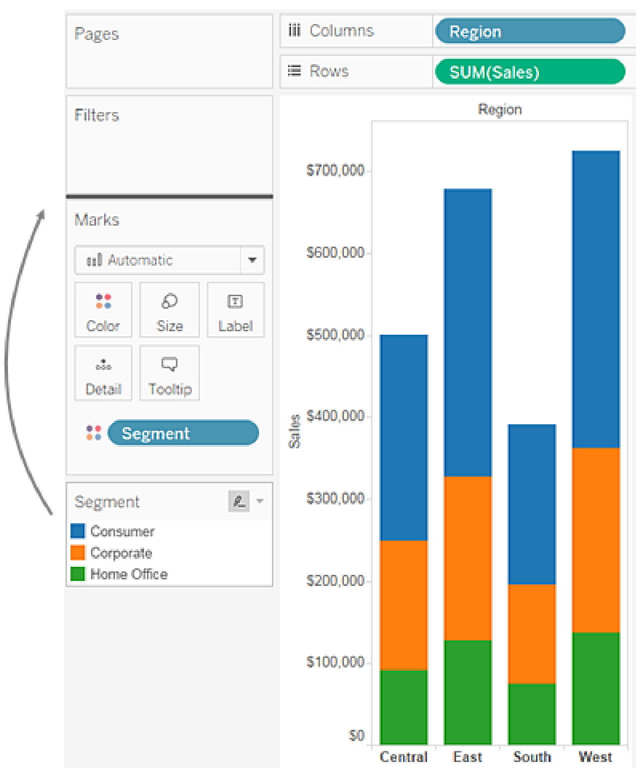


Reorganizing the Workspace

Every worksheet contains a variety of cards, shelves, legends, and so on. You can reorganize the workspace by rearranging cards, hiding and showing specific parts of the workspace, and hiding everything but the view using Presentation Mode.

Rearranging Cards

A worksheet contains several cards that contain shelves, legends, and other controls. Each card can be rearranged to create a custom workspace. To move a card, point the cursor at the title area of the card you want to move. When the cursor becomes the move symbol, click and drag the card to a new position. As you drag the card around the worksheet, the possible positions for it are highlighted with a black bar.



You can restore the worksheet windows to their default state by selecting **Reset Cards** on the Show/Hide Cards toolbar control.

Showing and Hiding Parts of the Workspace

Just about everything in the workspace can be turned on and off so you can avoid cluttering the worksheet with unnecessary cards, shelves, etc.

- To show and hide the toolbar or status bar, select **Window** and then select what you want to hide.
- To show and hide the window on the left side, which is either the Data pane (for worksheets), the Dashboard pane (for dashboards), or the Dashboards and Sheets pane (for stories); click the Minimize button in the upper right corner of the pane. The pane is minimized to the bottom left corner of the workbook. Click the same button again to restore the pane.
- To show or hide a card click **Show/Hide Cards** on the toolbar and then select the card you want to show or hide.

You can restore the worksheet windows to their default state by selecting **Reset Cards** on the **Show/Hide Cards** toolbar control.

Presentation Mode

Sometimes you may want to use Tableau to present your findings. Rather than hiding each card or shelf one at a time, you can switch to presentation mode. Presentation mode hides everything on the sheet except for the view and its associated legends, filter cards, parameter controls, and worksheet tabs.

- To switch in and out of presentation mode, click the Presentation Mode button on the toolbar or select **Window** > **Presentation Mode**.

Use the presentation mode controls in the bottom right corner to move between sheets and more. Each presentation mode control is described below.

	Show Filmstrip - shows the sheets as thumbnails at the bottom of the workspace.
	Show Tabs - shows the sheet tabs at the bottom of the workspace.

	Previous/Next Sheet - advances forward or backward through the sheets in a workbook.
	Enter/Exit Full Screen - switches between expanding the workbook to fill the entire screen and showing it in a window.
	Exit Presentation Mode - returns the workbook to showing the entire workspace including the menus, toolbar, and the Data pane.

Tableau File Types and Folders

You can save your work using several different Tableau specific file types: workbooks, bookmarks, packaged data files, data extracts, and data connection files. Each of these file types are described below. For related details, see **Save Your Work** on page 3017.

- **Workbooks (.twb)** – Tableau workbook files have the .twb file extension. Workbooks hold one or more worksheets, plus zero or more dashboards and stories.
- **Bookmarks (.tbm)** – Tableau bookmark files have the .tbm file extension. Bookmarks contain a single worksheet and are an easy way to quickly share your work. For more information, see **Save a bookmark** on page 3019.
- **Packaged Workbooks (.twbx)** – Tableau packaged workbooks have the .twbx file extension. A packaged workbook is a single zip file that contains a workbook along with any supporting local file data and background images. This format is the best way to package your work for sharing with others who don’t have access to the original data. For more information, see **Packaged Workbooks** on page 3022.
- **Extract (.hyper or .tde)** – Depending on the version the extract was created in, Tableau extract files can have either the .hyper or .tde file extension. Extract files are a local copy of a subset or entire data set that you can use to share data with others, when you need to work offline, and improve performance. For more information, see **Extract Your Data** on page 1014.
- **Data Source (.tds)** – Tableau data source files have the .tds file extension. Data source files are shortcuts for quickly connecting to the original data that you use often. Data source files do not contain the actual data but rather the information necessary to connect to the actual data as well as any modifications you've made on top of the actual data such as changing default properties, creating calculated fields, adding groups, and

so on. For more information, see **Save Data Sources** on page 1075.

- **Packaged Data Source (.tdsx)** – Tableau packaged data source files have the .tdsx file extension. A packaged data source is a zip file that contains the data source file (.tds) described above as well as any local file data such as extract files (.hyper or .tde), text files, Excel files, Access files, and local cube files. Use this format to create a single file that you can then share with others who may not have access to the original data stored locally on your computer. For more information, see **Save Data Sources** on page 1075.

These files can be saved in the associated folders in the My Tableau Repository directory, which is automatically created in your My Documents folder when you install Tableau. Your work files can also be saved in other locations, such as your desktop or a network directory.

Changing the Repository Location

You can specify a new location for the Tableau repository if you are not using the default location in your Documents folder. For instance, if you are required to have your data on a network server instead of on your local machine, you can point Tableau at the remote repository.

1. Select **File** > **Repository Location**.
2. Select a new folder that will act as the new repository location in the Select a Repository dialog box.
3. Restart Tableau so that it uses the new repository.

Changing the repository location does not move the files contained in the original repository. Instead, Tableau creates a new repository where you can store your files.

Language and Locale

Tableau Desktop is localized into several languages.

When you first run Tableau, it recognizes your computer locale and uses the appropriate language if it is supported. If you are using an unsupported language, the application defaults to English.

You can configure Tableau to display the user interface (menus, messages, etc.) by choosing **Help** > **Choose Language**. After you change this setting, you'll need to restart the application for the changes to take effect. You do not need to change this setting for every workbook.

To configure date and number formatting, choose **File** > **Workbook Locale**. By default, the locale is set to **Automatic**, which means the locale will match the locale when the workbook is opened. This can be useful if you are authoring a workbook that will be viewed in many different languages and you want the dates and numbers to update accordingly. When you select a specific locale, the workbook will not change regardless of who opens it.

Tableau checks the following, in order, to determine the workbook locale:

- Workbook Locale (explicit setting)
- Windows Locale or Mac language
- Tableau Language

If none of the above is set, then the workbook locale defaults to English.

Day of the Week Sorting

If you are working in a language for which Tableau does not provide a local version, set your workbook locale to assure that Tableau can sort the days of the week in the correct chronological order. Otherwise, Tableau will sort the names of the days alphabetically. If none of the supported locales is appropriate, you can sort the days of the week manually. See **Sort Data in a Visualization** on page 1447.

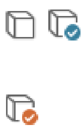
Visual Cues and Icons in Tableau Desktop

Tableau provides many visual cues to help you evaluate the type of data that's displayed in the Data pane and the state of a data view.

Data sources in the Data pane

The following table explains each of the icons used to describe the type of data sources in the Data pane. Each icon in the table can be modified by one of two indicators.

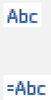
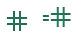
- A blue check mark indicates that the data source is the primary data source in the workbook.
- An orange check mark indicates that the data source is the secondary data source in the workbook.

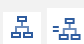
Visual Cue	Description
	The workbook is directly connected to a relational data source or file.
	The workbook is connected to a cube (multidimensional) data source. In Tableau, cube data sources are supported only in Windows.
	The workbook is connected to an extract that still references the underlying data.
	The workbook is connected to an Excel table, sub-table, or named range.
	The workbook is connected to a data source that has been published to Tableau Server.

Fields in the Data Pane

The following table explains each of the icons displayed in the Data pane. Each icon in the table can be modified by one of four indicators.

- Blue icons indicate that the field is discrete. Abc
- Green icons indicate that the field is continuous. #
- Icons preceded by the equal sign (=) indicate that the field is a user-defined calculation or a copy of another field. =Abc

Visual Cue	Description
	The field contains text values.
	The field contains numeric values.
	The field is a calculation defined on the server.
	The field contains only date values.
	The field contains both date and time values.
	The field contains geographical data and has been assigned a geographic role. Use these fields when building map views. See **Maps and Geographic Data Analysis in Tableau** on page 1755.
	The field contains geographical data from an active custom geocoding file. See **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map** on page 1818.
	The field contains boolean (true or false) values.
	The field is a calculation that is defined in the database by an administrator. These fields are marked with a cylinder icon and are not available for all data sources.
	The field is a user-defined set. See **Create Sets** on page 1186.

Visual Cue	Description
	The field is a server named set.
	The field is a set that was automatically created as a result of an action.
	The field is a user filter, used when publishing to the web. See **Restrict Access at the Data Row Level** on page 3088.
	The field is a numeric bin. See **Create Bins from a Continuous Measure** on page 1705.
	The field is a group. See **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1184.
	The field is a cluster group. See **Find Clusters in Data** on page 2554.
	The field is a relational hierarchy. See **Create Hierarchies** on page 1170.
	A folder that contains one or more fields. Folders are used to organize fields in the Data pane. See **Organize the Data Pane** on page 1146.
	The field is an attribute of a cube (multidimensional) data source. In Tableau, cube data sources are supported only in Windows.
	The field is a varying attribute of a cube (multidimensional) data source.
	The field is a level in a multidimensional hierarchy. Levels greater than five are shown without numbers.
	The field is blended with a field from another data source. See **Blend Your Data** on page 910

Visual Cue	Description
⊂/⊃	The field is not blended with a field from another data source. See **Blend Your Data** on page 910

Fields on Shelves

Fields placed on shelves use a combination of icons, colors, and text styles as visual cues.

Visual Cue	Description
Region	A blue field on a shelf indicates a discrete field. In most cases, adding a dimension to a shelf results in a blue field. Blue fields are discrete—they contain a finite number of values. Adding a blue field to a shelf creates headers. For details, see **Headers** on page 242.
SUM(Sales)	A green field on a shelf indicates a continuous field. In most cases, adding a measure to a shelf results in a green field. Green fields are continuous—they contain an infinite number of values. Adding a green field to a shelf creates an axis. For details, see **Axes** on page 246.
Company	The Sort icon indicates a field that has either a computed or manual sort order applied. See **Sort Data in a Visualization** on page 1447.
Σ Date.Fiscal	The sigma icon indicates a slicing filter in a multidimensional (cube) data source. See **Create Slicing Filters** on page 1091.
Core Product Gro..	The Venn diagram icon indicates a set. See **Create Sets** on page 1186.

Visual Cue	Description
Core Product ..	A field name shown in italics indicates a filtered set.
Segment: Corporate	A gray field on the Filters shelf indicates a context filter. See **Improve View Performance with Context Filters** on page 1439.
AVG(Close) △	The delta icon indicates that the field is a table calculation. See **Transform Values with Table Calculations** on page 2406.
⊟ Country ⊞ State	The plus and minus controls appear when the field is part of a hierarchy that you can traverse.
SUM(Sales)	The arrow icon indicates that a forecast is being displayed for the field. This icon is also used on the Forecast Indicator field, which is an automatic field used to distinguish between actual and forecast values. See **Forecasting** on page 2580.
SUM(Sales ..	The field is from a secondary data source. See **Blend Your Data** on page 910.
Last Sale Price	The field is assigned to a specific worksheet.
Segment	The field is assigned to all worksheets with the same data source.
Internet Sales Amount !	The field is incompatible with one or more other fields in the view.

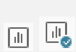
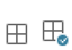
Fields on the Marks card

Fields placed on the Marks card use specific icons to describe how they appear in the view. For more details, see **Control the Appearance of Marks in the View** on page 1356.

Visual Cue	Description
Segment	The field is applied to Color on the Marks card.
SUM(Profit)	The field is applied to Size on the Marks card.
AVG(Discou..	The field is applied to Label on the Marks card.
Sub-Category	The field is applied to Shape on the Marks card.
SUM(True Average)	The field is applied to Detail on the Marks card.
AVG(Math)	The field is applied to Tooltip on the Marks card.
YEAR(Date)	The field is applied to Path on the Marks card. Path is only available when the Line or Polygon mark type is selected from the Marks drop-down menu.

Sheets in the Dashboards and Worksheets pane

The following table explains each of the icons used to describe the type of sheet that can be placed in a story. A blue check mark indicates that a sheet is being used in one or more story points.

Visual Cue	Description
	The sheet is a worksheet.
	The sheet is a dashboard.

Fields in the Calculation editor

Fields in the calculation editor are color coded by field type.

Visual Cue	Description
[ORANGE]	The field is a dimension, set, or measure.
[Parameters].[PURPLE]	The field is a parameter. Tableau appends [Parameters] to avoid ambiguity when a parameter has the same name as another field
[ORANGE].[ORANGE]	The field is from a secondary data source.
BLUE()	The field is a calculation function.

Tableau Concepts

What are dimensions and measures? Why are some fields dimensions and others measures?

Why is the background color blue for some fields, and green for others?

How does adding a filter affect my viz?

How does Tableau aggregate data in a view?

The topics in this section cover concepts that answer these questions to help you understand why Tableau does what it does.

If you're new to Tableau Desktop, also see other topics in **Get Started** on page 169.

If you're new to Tableau Public, also see Tableau Public Resources.

Watch a video: To see related concepts demonstrated in Tableau, watch the free training videos listed under Why is Tableau Doing That? Use your tableau.com account to sign in.

Structure Data for Analysis

There are certain concepts that are fundamental to understanding data prep and how to structure data for analysis. Data can be generated, captured, and stored in a dizzying variety of formats, but when it comes to analysis, not all data formats are created equal.

Data preparation is the process of getting well formatted data into a single table or multiple related tables so it can be analyzed in Tableau. This includes both the structure, i.e. rows and columns, as well as aspects of data cleanliness, such correct data types and correct data values.

Tip: It may help to go through the following topic with a data set of your own. If you do not already have a data set you can use, see our tips for finding good data sets.

How structure impacts analysis

The structure of your data may not be something you can control. The rest of this topic assumes you have access to the raw data and the tools needed to shape it, such as Tableau Prep Builder. However, there may be situations when you can't pivot or aggregate your data as desired. It is often still possible to perform the analysis but you may need to change your calculations or how you approach the data. For an example of how to perform the same analysis with different data structures, see Tableau Prep Day in the Life Scenarios: Analysis with the Second Date in Tableau Desktop. But if you can optimize the data structure it will likely make your analysis much easier.

Data Structure

Tableau Desktop works best with data that is in tables formatted like a spreadsheet. That is, data stored in rows and columns, with column headers in the first row. So what should be a row or column?

What is a row?

A row, or record, can be anything from information around a transaction at a retail store, to weather measurements at a specific location, or stats about a social media post.

It's important to know what a record (row) in the data represents. This is the *granularity* of the data.

Here, each record is a day

Here, each record is a month

Date	Max TemperatureF	Mean TemperatureF	Min TemperatureF
1/1/2015	42		
1/2/2015	42		
1/3/2015	41		
1/4/2015	51		
1/5/2015	54		
1/6/2015	54		
1/7/2015	46		
1/8/2015	46		
1/9/2015	50		
1/10/2015	46		

Date	Max TemperatureF	Mean TemperatureF	Min TemperatureF
January	63	45	26
February	62	49	33
March	69	51	31
April	77	52	37
May	82	59	43
June	92	68	49
July	95	71	54
August	92	69	54
September	81	61	45
October	74	58	45
November	60	44	25

Tip:A best practice is to have a unique identifier, a value that identifies each row as a unique piece of data. Think of it like the social security number or URL of each record. In Superstore, that would be Row ID. Note that not all data sets have a UID but it can't hurt to have one.

Try to make sure you can answer the question "What does a row in the data set represent?". This is the same as answering "What does the TableName(Count) field represent?". If you can't articulate that, the data might be structured poorly for analysis.

Aggregation and Granularity

A concept related to what makes up a row is the idea of aggregation and granularity, which are opposite ends of a spectrum.

Aggregation

- refers to how multiple data values are brought together into a single value, such as counting all the Google searches for Pumpkin Spice or taking the average of all the temperature readings around Seattle on a given day.
- By default, measures in Tableau are always aggregated. The default aggregation is SUM. You can change the aggregation to options like Average, Median, Count Distinct, Minimum, etc.

Granularity

- refers to how detailed the data is. What does a row or record in the data set represent? A person with malaria? A provinces' total cases of malaria for the month? That's the granularity.
- Knowing the granularity of the data is crucial to working with level of detail (LOD) expressions.

Understanding aggregation and granularity is a critical concept for many reasons; it impacts things like finding useful data sets, building the visualization you want, relating or joining data correctly, and using LOD expressions.

Tip: For more information, see the training video Aggregation and Granularity or Data Aggregation in Tableau in the Help

What is a field or column?

A *column* of data in a table comes into Tableau Desktop as a *field* in the data pane, but they are essentially interchangeable terms. (We save the term *column* in Tableau Desktop for use in the columns and rows shelf and to describe certain visualizations.) A field of data should contain items that can be grouped into a larger relationship. The items themselves are called *values* or *members* (only discrete dimensions contain members).

What values are allowed in a given field are determined by the *domain* of the field (see the note below). For example, a column for "grocery store departments" might contain the members "deli" "bakery", "produce", etc., but it wouldn't include "bread" or "salami" because those are items, not departments. Phrased another way, the domain of the department field is limited to just the possible grocery store departments.

Additionally, a well-structured data set would have a column for "Sales" and a column for "Profit", not a single column for "Money", because profit is a separate concept from sales.

- The domain of the Sales field would be values ≥ 0, since sales cannot be negative.
- The domain of the Profit field, however, would be all values, since profit can be negative.

Note: *Domain* can also mean the values present in the data. If the column "grocery store department" erroneously contained "salami", by this definition, that value would be in the domain of the column. The definitions are slightly contradictory. One is the values that could or should be there, the other is values that actually are there

Categorizing fields

Each column in the data table comes into Tableau Desktop as a field, which appears in the **Data** pane. Fields in Tableau Desktop must be either a dimension or measure (separated by a line within tables in the **Data** pane) and either discrete or continuous (color coded: blue fields are discrete and green fields are continuous).

- *Dimensions* are qualitative, meaning they can't be measured but are instead described. Dimensions are often things like city or country, eye color, category, team name, etc. Dimensions are usually discrete.
- *Measures* are quantitative, meaning they can be measured and recorded with numbers. Measures can be things like sales, height, clicks, etc. In Tableau Desktop, measures are automatically aggregated; the default aggregation is SUM. Measures are usually continuous.
- *Discrete* means individually separate or distinct. Toyota is distinct from Mazda. In Tableau Desktop, discrete values come into the view as a label and they create panes.
- *Continuous* means forming an unbroken, continuous whole. 7 is followed by 8 and then it's the same distance to 9, and 7.5 would fall midway between 7 and 8. In Tableau Desktop, continuous values come into the view as an axis.
- Dimensions are usually discrete, and measures are usually continuous. However, this is not always the case. Dates can be either discrete or continuous.
    - Dates are dimensions and automatically come into the view as discrete (aka date parts, such as "August", which considers the month of August without considering other information like the year). A trend line applied to a timeline with discrete dates will be broken into multiple trend lines, one per pane.
    - We can chose to use continuous dates if preferred (aka date truncations, such as "August 2017", which is different than "August 2018"). A trend line applied to a timeline with continuous dates will have a single trend line for the entire date axis.

Tip: For more information, see the training video Understanding Pill Types or Dimensions and Measures, Blue and Green in the Help.

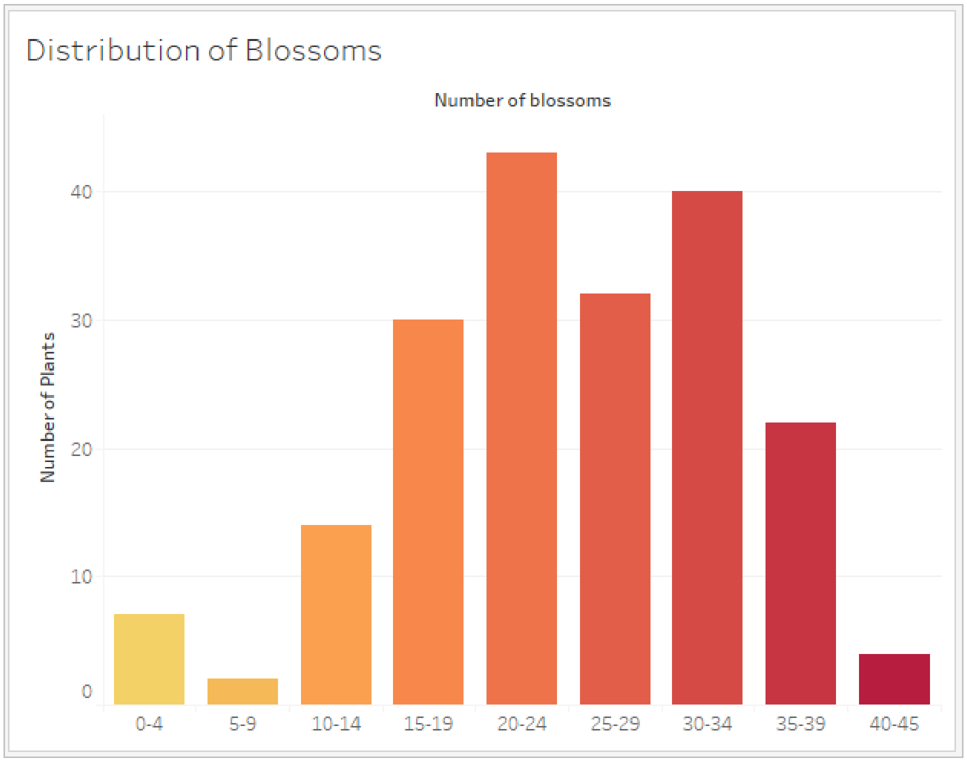
In Tableau Prep, no distinction is made for dimensions or measures. Understanding the concepts behind discrete or continuous are important, however, for things like understanding the detail versus summary presentation of data in the profile pane.

- Detail: the detail view shows every domain element as a discrete label and has a visual scrollbar to provide a visual overview of all the data.
- Summary: the summary view shows the values as binned on a continuous axis as a histogram.

Binning & Histograms

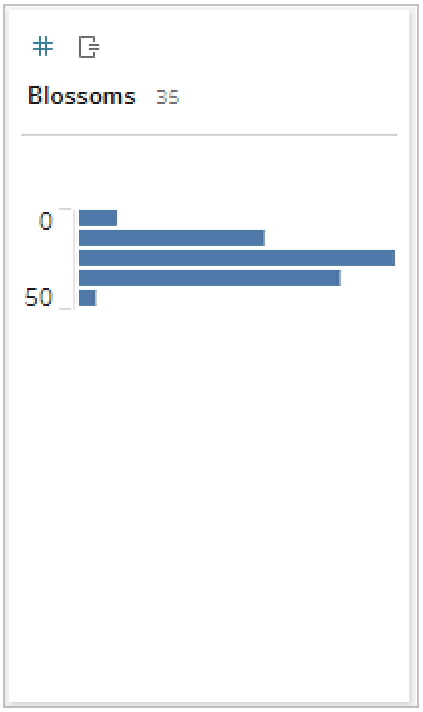
A field like age or salary is considered continuous. There is a relationship between the age 34 and 35, and 34 is as far from 35 as 35 is from 36. However, once we're past age 10 or so, we usually stop saying things like we're "9 and a half" or "7 and ¾". We’re already binning our age to neat year-sized increments. Someone who is 12,850 days old is older than someone who is 12,790 days old, but we draw a line and say they're both 35. Similarly, age groupings are often used in place of actual ages. Child prices for movie tickets might be for kids 12 and under, or a survey may ask you to select your age group, such as 20-24, 25-30, etc.

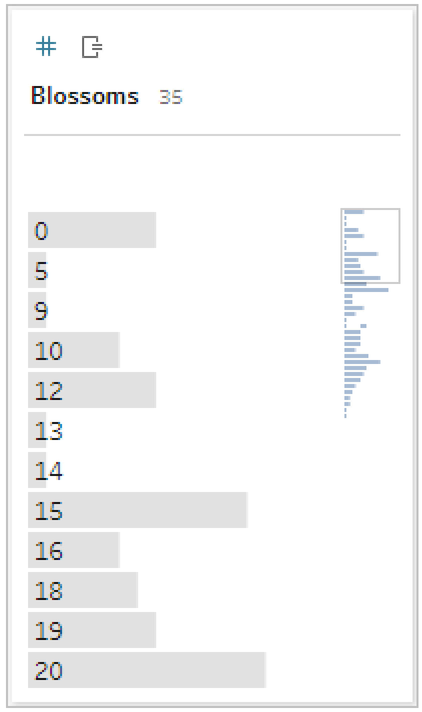
Histograms are used to visualize the distribution of numerical data using binning. A histogram is similar to a bar chart, but rather than being discrete categories per bar, the rectangles making up the histogram span a *bin* of a continuous axis, such as range of the number of blossoms (0-4, 5-9, 10-14, etc.). The height of the rectangles is determined by frequency or count of those values. Here, the y-axis is the count of plants that fall into each bin. Seven plants have 0-4 blossoms, two plants have 5-9 blossoms, and 43 plants have 20-24 blossoms.



In Tableau Prep, the summary view is a histogram of binned values. The detail view shows the frequency for every value and has a visual scrollbar off to the side that shows the overall distribution of the data.

Summary view	Detail view





Distributions and outliers

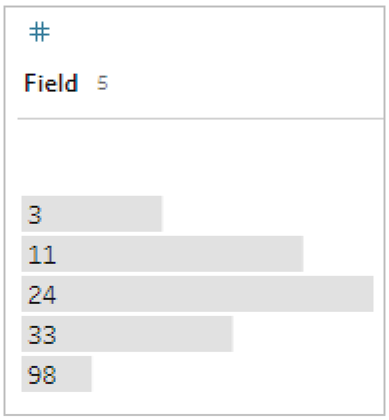
Seeing the distribution of a data set can help with outlier detection.

- *Distribution*: the shape of the data in a histogram, though this depends on the size of the bins. Being able to see all your data in a histogram view can help identify if the data seems correct and complete. The shape of the distribution will only be of use if you know the data and can interpret whether or not the distribution makes sense.
  - For example, if we were to look at a data set of the number of homes with broadband internet from 1940-2017, we'd expect to see a very skewed distribution. However, if we were to look at the number of homes with broadband internet from January 2017 to December 2017 we'd expect a fairly uniform distribution.
  - If we were to look at a data set of Google searches for "Pumpkin Spice Latte", we'd expect to see a fairly sharp peak in the fall, whereas searches for “"convert Celsius to Fahrenheit" would likely be fairly stable.
- *Outlier*: a value that is extreme compared to other values. Outliers may be correct values or they may be indicative of an error.

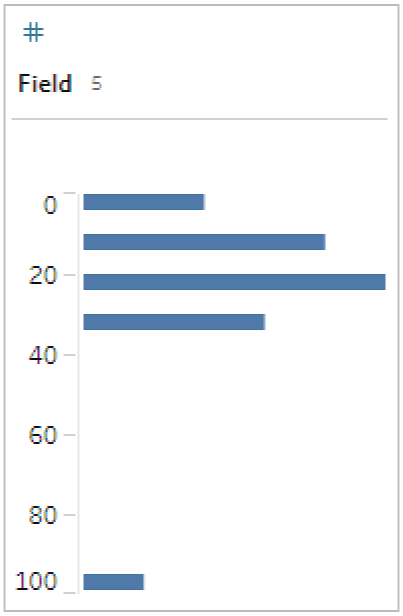
- Some outliers are correct and indicate actual anomalies; these should not be removed or modified.
- Some outliers indicate issues with data cleanliness, such as a salary of $50 instead of $50,000 because a period was typed instead of the comma.

Visual outlier detection with distributions

If you were to see a list like this:



at first glance it doesn't look odd. But if, instead of a list of labels, this was plotted on a continuous binned axis, it would look like this:



And it's much more obvious that the last observation is farther away from the first and may be an outlier due to error.

Data Types

Databases, unlike spreadsheets, usually enforce strict rules on data types. Data types classify the data in a given field and provide information about how the data should be formatted, interpreted, and what operations can be done to that data. For example, numerical fields can have mathematical operations applied to them and geographic fields can be mapped.

Tableau Desktop assigns whether a field is a dimension or measure, but fields have other characteristics that depend on their data type. These are indicated by the icon each field has (though some types share an icon). Tableau Prep uses the same data types. If data type is enforced on a column and an existing value doesn't match its assigned data type, it may be displayed as null (because "purple" doesn't mean anything as a number).

Some functions require specific data types. For example, you cannot use CONTAINS with a numerical field. Type functions are used to change the data type of a field. For example, DATEPARSE can take a text date in a specific format and make it a date, thus enabling things like automatic drill down in the view.

Icon	Data type
Abc	Text (string) values
	Date values
	Date & Time values
#	Numerical values
T|F	Boolean values (relational only)
	Geographic values (used with maps)

Tip: For more information, refer to the Help article on Data Types.

Pivot and Unpivot Data

People-friendly data is often captured and recorded in a wide format, with many columns. Machine-readable data, like Tableau prefers, is better in a tall format, with fewer columns and more rows.

> **Note**: Traditionally, pivoting data means going from tall to wide (columns to rows), and unpivoting means going from wide to tall (rows to columns). However, Tableau uses the word *pivot* to mean going from wide (people-friendly) to tall (machine-readable) by turning rows into columns. In this document, *pivot* will be refer to the Tableau sense of the word. For clarity, it can help to specify "pivot columns to rows" or "pivot rows to columns".
>
> For more information, refer to the Help articles Pivot Your Data and Tips for Working with your Data.

Wide data

In the WHO malaria data set, there is a column for country, then a column per year. Each cell represents the number of cases of malaria for that country and year. In this format we have 108 rows and 16 columns.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1	**Country**	**2014**	**2013**	**2012**	**2011**	**2010**	**2009**	**2008**	**2007**	**2006**	**2005**	**2004**	**2003**	**2002**	**2001**	**2000**
2	**Afghanistan**	61362	39263	54840	77549	69397	64880	81574	92202	86129	116444	242022	360940	415356		94475
3	**Algeria**	0	0	55	1	1	0	3	27	1	2	3	6	8	8	35
4	**Angola**	2298979	1999868	1496834	1632282	1682870	1573422	1377992	1533485	1082398	889572					
5	**Argentina**	0	0	0	0	14	86	130	387	212	252	115	122	125	215	440
6	**Armenia**					0	0	0	0	0	7	47	29	52	79	141
7	**Azerbaijan**	0	0	3	4	50	78	72	108	143	242	386	482	506	1058	1526
8	**Bahamas**				6	1		14	6	49	1	2	3	1	4	2
9	**Bangladesh**	10216	3864	9901	51773	55873	63873	84690	59866	32857	48121	58894	54654	62269	54216	55599
10	**Belize**	19	20	33	72	150	256	540	845	844	1549	1066	1084	1134	1162	1486
11	**Benin**	1044235	1078834	705839	422968		889597									
12	**Bhutan**	19	15	0	194	436	972	329	793	1868	1825	2670	3806	6511	5982	5935
13	**Bolivia (Plurinatic**	7401	7342	7415	7143	13769	9743	9748	14610	19725	21442	14910	20343	14276	15765	31469
14	**Botswana**	1346	456	193	432	1046	1024	927	390		670	198	591	1640	3720	10510
15	**Brazil**	143415	177767	242758	267146	334667	309316	315746	458652	549469	606067	465004	408886	348259	388303	613241
16	**Burkina Faso**	5428655	3769051	3858046	428113	804539	182527	36514	44246	44265	21335	18256				

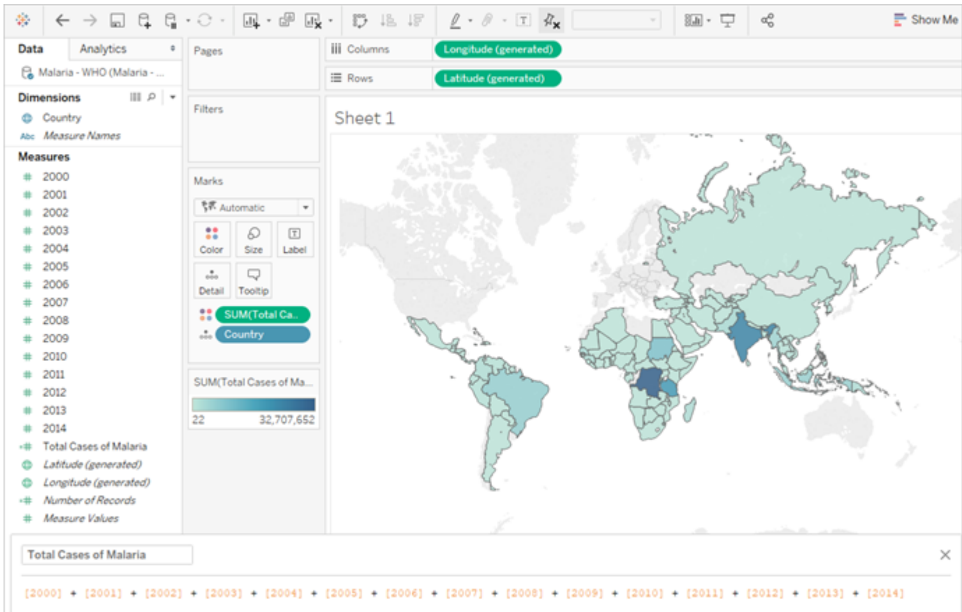
It's easy for a person to read and understand this format. However, if we were to bring this data into Tableau Desktop, we get a field per column. We have a field for 2000, a field for 2001, a field for 2002, etc.

To think of it another way, there are 15 fields that all represent the same basic thing—number of reported cases of malaria—and no single field for time. This makes it very hard to do analysis across time as the data is stored in separate fields.

Example: working with wide data

Q: How would we create a map that shows the total number of malaria cases per country from 2000 to 2014?

A: Create a calculated field to sum all the years.



Note: this image has not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

Another indication that this format isn't ideal for analysis can be seen in the fact that nowhere do we have information about what the actual values mean. For Algeria in 2012, we have the value 55. Fifty five what? It's not clear from the structure of the data.

	A	B	C	D	E
1	Country	2014	2013	2012	2011
2	Afghanistan	61362	39263	54840	77549
3	Algeria	0	0	55	1
4	Angola	2298979	1999868	1496834	1632282

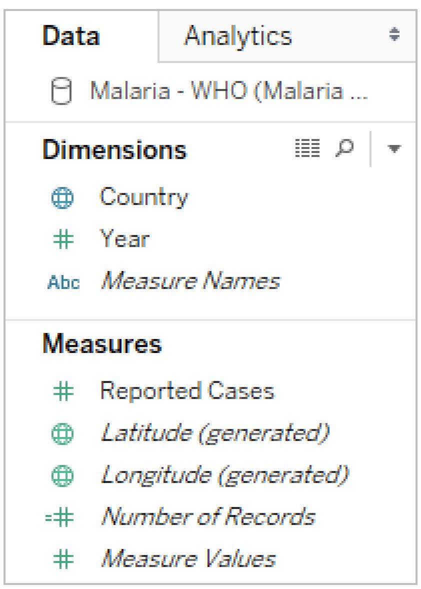
If the name of the column isn't describing what the values are but rather conveys additional information, this is a sign the data needs to be pivoted.

Tall data

If we pivot the data, we reshape the data from wide to tall. Now, rather than having a column for each year, we have a single column, Year, and a new column, Reported Cases. In this format we have 1,606 rows and 3 columns. This data format is taller rather than wider.

	A	B	C
1	Country	Year	Reported Cases
2	Afghanistan	2000	94,475
3	Afghanistan	2001	
4	Afghanistan	2002	415,356
5	Afghanistan	2003	360,940
6	Afghanistan	2004	242,022
7	Afghanistan	2005	116,444
8	Afghanistan	2006	86,129
9	Afghanistan	2007	92,202
10	Afghanistan	2008	81,574
11	Afghanistan	2009	64,880
12	Afghanistan	2010	69,397
13	Afghanistan	2011	77,549
14	Afghanistan	2012	54,840
15	Afghanistan	2013	39,263

Now in Tableau Desktop, we have a field for Year and a field for Reported Cases as well as the original Country field. It's much easier to do analysis because each field represents a unique quality about the data set—location, time, and value.

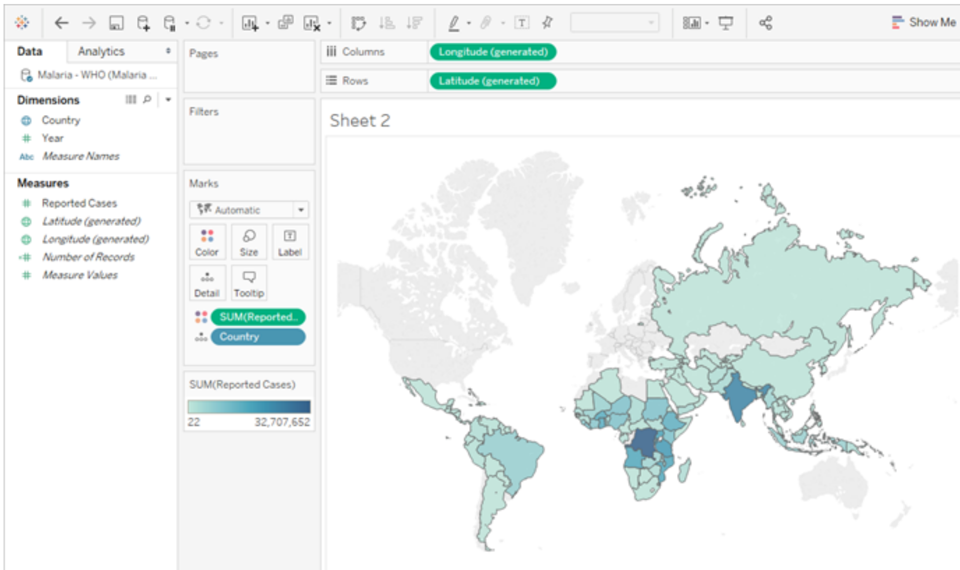


Note: this image has not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

Example: working with tall data

Q: How would we create a map that shows the total number of malaria cases per country from 2000 to 2014?

A: Use the Reported Cases field.



Note: this image has not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

Now it's easy to see that for Algeria in 2012, the 55 refers to the number of reported cases (because we could label this new column).

	A	B	C
1	Country	Year	Reported Cases
28	Algeria	2011	1
29	Algeria	2012	55
30	Algeria	2013	0

Note: In this example, the wide data consisted of a single record per country. With the tall data format, there are now 15 rows for each country (one for each of the 15 years in the data). It's important to keep in mind that there are now multiple rows per country.

If there was a column for Land Area, that value would be repeated for each of the 15 rows for each country in a tall data structure. If you created a bar chart by bringing out **Country** to **Rows** and **Land Area** to **Columns**, by default the view would sum the land area for all 15 rows per country.

For some fields it may be necessary to compensate for double counting values by aggregating with an average or minimum rather than sum or filtering.

Normalization

Relational databases are made up of multiple tables that can be related or linked together in some way. Each table contains a unique identifier, or key, per record. By relating or joining on the keys, records can be linked to provide more information than is contained in a single table. What information goes into each table is dependent on the data model used, but the general principle is around reducing duplication.

For example, consider event planning for an event like a wedding. We need to keep track of information at the level of groups (such as families or couples) as well as the level of individuals.

A table could be created that combines all the information together:

ID	Name	Group	Dietary	Seating	Attending	Address	Invitation	Gift	Bride
10	Cedar	Tree	Omnivore	1	yes	87 Forest Ave	sent	pebble	yes
11	Redwood	Tree	Omnivore	1	yes	87 Forest Ave	sent	pebble	yes
12	Fir	Tree	Vegan	1	yes	87 Forest Ave	sent	pebble	yes
13	Madrona	Tree	Omnivore		no	87 Forest Ave	sent	pebble	yes
14	Chanterelle	Mushroom	Omnivore	2	yes	3 Troll St	sent	leaf	no
15	Cremini	Mushroom	Kosher	2	yes	3 Troll St	sent	leaf	no
16	Portobello	Mushroom	Omnivore	2	yes	3 Troll St	sent	leaf	no
17	Hydrangea	Flower				652 Meadow Ln	undeliverable		yes
18	Dahlia	Flower				652 Meadow Ln	undeliverable		yes

However, if an address is incorrect and needs to be fixed, it must be fixed across multiple rows, potentially leading to errors or conflicts. A better structure is to create two tables, one for information that pertains to the group (such as address and if the invitation was sent) and one for information pertaining to the individuals (for things like seating assignments and dietary restrictions).

Group table

Individual table

ID	Group	Address	Invitation	Gift	Bride
34	Tree	87 Forest Ave	sent		
35	Mushroom	3 Troll St	sent		
36	Flower	652 Meadow Ln	undeliverable		

ID	Name	Group	Dietary	Seating	Attending
10	Cedar	Tree	Omnivore	1	yes
11	Redwood	Tree	Omnivore	1	yes
12	Fir	Tree	Vegan	1	yes
13	Madrona	Tree	Omnivore		no
14	Chanterelle	Mushroom	Omnivore	2	yes
15	Cremini	Mushroom	Kosher	2	yes
16	Portobello	Mushroom	Omnivore	2	yes
17	Hydrangea	Flower			
18	Dahlia	Flower			

It's much easier to track and analyze group-level information in the group table and individual-level information in the individual table. For example, the number of chairs needed can be obtained from the number of Attending = Yes records in the individual table, and the number of stamps needed for thank-yous can be obtained from the number of records in the group table where Gift isn't null.

The process of breaking up all the data into multiple tables—and figuring out which table contains which columns—is called *normalization*. Normalization helps reduce redundant data and simplifies the organization of the database.

However, there may be times when information is needed that spans multiple tables. For example, what if we wanted to balance seating arrangements (individuals) such that groups from the bride's side are intermingled with groups from the groom's side? (The bridge or groom affiliation is tracked at the group level.) To achieve this, we need to relate the tables back together so the individuals are associated with information about their group. Proper normalization isn't just breaking up tables, it also requires the presence of a shared, related field or unique identifier than can be used to combine the data back together again. Here, that related field is Group. That field is present in both tables, so we can join on this field and return to our original, single table format. This is a *denormalized* structure.

So why didn't we just keep the original denormalized table? It is harder to maintain and was storing redundant information. At scale, the level of data duplication can be come massive. Storing the same information over and over isn't efficient.

Normalized tables have a few key properties:

- Each row needs a unique identifier
- Each table needs a column or columns that can be used to connect it back to other

tables (key).

These shared (key) columns are used for relating or joining tables back together. For our data, the relationship or join clause would be on the Group field in each table.

Join Types

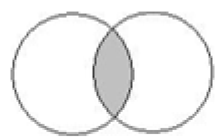
Although the default method for combing data in Tableau Desktop is relating, there are cases when you may want to join tables in Tableau Desktop or Tableau Prep Builder. For a basic overview of joins and join types, see Join Your Data.

Name	# of Siblings
Taylor	2
Alex	3
Shannon	0
Tracy	1

Name	Eye Color
Taylor	Blue
Alex	Brown
Morgan	Brown

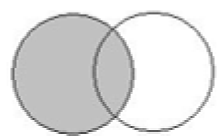
Inner Join

Name	# of Siblings	Eye Color
Taylor	2	Blue
Alex	3	Brown



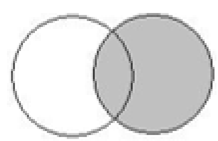
Left Join

Name	# of Siblings	Eye Color
Taylor	2	Blue
Alex	3	Brown
Shannon	0	*null*
Tracy	1	*null*



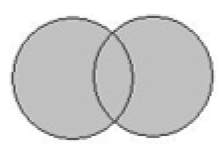
Right Join

Name	Eye Color	# of Siblings
Taylor	Blue	2
Alex	Brown	3
Morgan	Brown	*null*



Outer Join

Name	# of Siblings	Eye Color
Taylor	2	Blue
Alex	3	Brown
Shannon	0	*null*
Tracy	1	*null*
Morgan	*null*	Brown



"Tidy" Data

Hadley Wickham published an article in 2014 in the Journal of Statistical Software called "Tidy Data" (August 2014, Volume 59, Issue 10). This article does an excellent job of laying out a framework for data that is well-structured for analysis. The article can be found here (Hadley Wickham's Academic Portfolio) or here (hosted by r-project.org).

> **Note**: The article is hosted on external websites. Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the owners if you have questions regarding their content.

Find Good Data Sets

A good way to learn how to use Tableau Desktop (or build sample or proof-of-concept content) is to find a data set you find interesting. When you have genuine questions you want to answer with data, the steps of the analysis becomes easier and more meaningful.

The reality of data sets

There are two unavoidable facts about trying to find a data set that's not official, business-sanctioned data.

You won't find what you're looking for.

- Try to avoid an overly specific preconceived notion of what you want.
- Stay flexible and open minded about what you can use for a given project.
- Sometimes the data you want is behind a paywall—decide if it's worth it or not.

You will have to clean up the data.

- Be prepared for basic cleaning and shaping to make sure the data is well structured for analysis
- You may need to bring in additional data sets.
- Having a data dictionary or metadata can be vital.
- Calculations may be necessary.

What makes a good data set

A good data set is one that suits your purpose. As long as that need is met, it's a good data set. However, there are some considerations that can help you weed out data sets that are unlikely to suit your purpose. On the whole, look for data sets that meet the following conditions:

1. Contains the elements you need
2. Is disaggregated data
3. Has at least a couple dimensions and a couple measures

4. Has good metadata or a data dictionary
5. Is useable (not in a proprietary format, too messy, or too cumbersome)

What makes Superstore super?

Superstore is one of the sample data sources that come with Tableau Desktop. Why is it such a good data set?

- **Necessary elements**: Superstore has dates, geographic data, fields with a hierarchy relationship (Category, Sub-Category, Product), measures that are positive and negative (Profit), etc. There are very few chart types you can't make with Superstore alone, and very few features it can't be used to demonstrate.
- **Disaggregated**: The row-level data is each item in a transaction. Those items can be rolled up to the order level (via Order ID) or by any of the dimensions (such as date, customer, region, etc.)
- **Dimensions and measures**: Superstore has several dimensions allowing us to "slice and dice" by things like category or city. There are also multiple measures and dates, which opens the possibilities for chart types and calculations.
- **Metadata**: Superstore has well named fields and values. You don’t need to look up what any values mean.
- **Small and clean**: Superstore is only a few megabytes so it takes up very little room in the Tableau installer. It’s also very tidy data, with only the correct values in each field and a good data structure.

1. A good data set has the elements you need for your purposes

If you're looking for a data set to build a specific visualization or to showcase specific functionalities, make sure the data set has the types of fields you need. For example, maps are a great visual but require geographic data. Basic demos often involve drilling down into dates, so the data would need at least one date field (and it would need to be more granular than just year to show drill down). Not all data sets need all these elements—know what you need for your purpose and don't waste time with data sets that are missing key elements.

Common elements for analysis:

- Dates
- Geographic data
- Hierarchical data

- “Interesting” measures—either substantial variation in magnitude or positive and negative values

Some features or viz types may require specific characteristics of the data such as:

- Clusters
- Forecasting
- Trend lines
- User filters
- Spatial calculations
- Certain calculations
- Bullet charts
- Control charts

2. A good data set is disaggregated (raw) data

If the data is too aggregated, there isn't much you can do for analysis. For example, if you want to look at trends in people Googling "Pumpkin Spice" but have yearly data, you can only look at a very high level overview. Ideally you want to get your hands on daily data, so you could see the huge spike when Starbucks starts offering #PSL.

What counts as disaggregated can vary by analysis. Note that due to privacy or practicality, some data sets will never be more granular than a certain level. For example, you would be unlikely to find a data set with case-by-case reporting of malaria cases, so monthly totals by region might be granular enough.

Aggregation and granularity

Understanding aggregation and granularity is a critical concept for many reasons; it impacts things like finding useful data sets, building the visualization you want, combining data correctly, and using LOD expressions. Aggregation and granularity are opposite ends of a spectrum.

Aggregation refers to how the data is combined together, such as summing all the searches for Pumpkin Spice or taking the average of all the temperature readings around Seattle on a given day.

- By default, measures in Tableau are aggregated. The default aggregation is SUM. You can change the aggregation to things like Average, Median, Count Distinct, Minimum, etc.

Granularity refers to how detailed the data is. What does a row (aka record) in the data set represent? A person with malaria? A provinces' total cases of malaria for the month? That's the granularity. Knowing the granularity of the data is crucial to working with LOD (level of detail) expressions.

For more information, see the Free Training Video on aggregation and granularity or the Help topic **Data Aggregation in Tableau** on page 341.

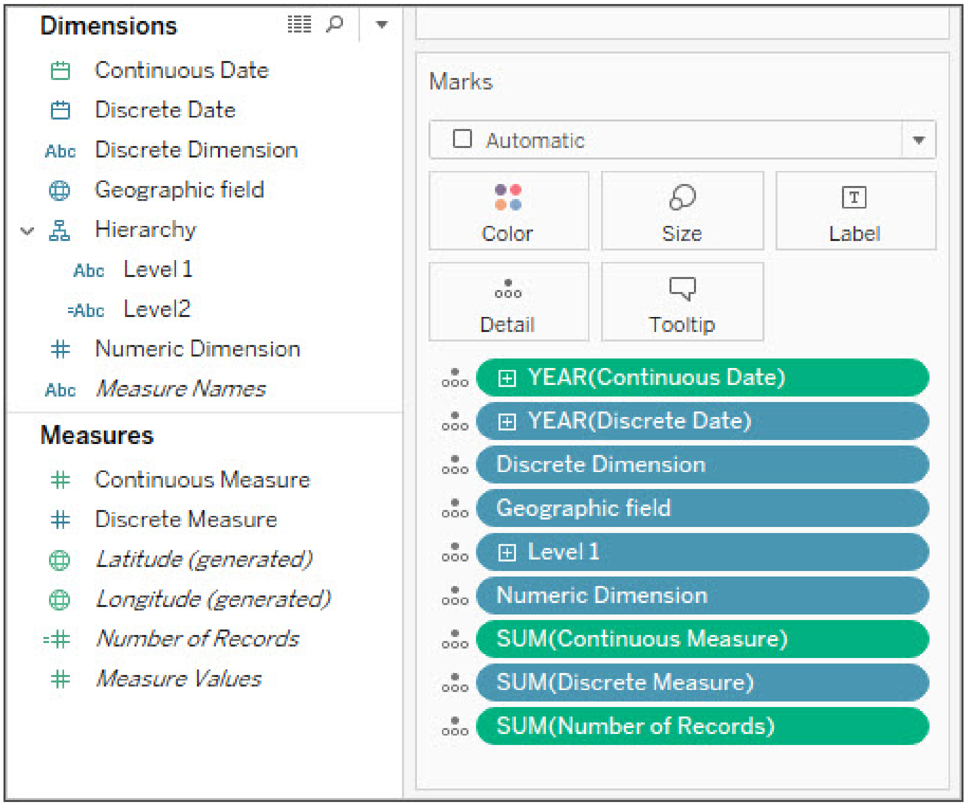
3. A good data set has dimensions and measures

Many visualization types require **dimensions** and **measures**

- If you only have dimensions, you're mostly limited to counting, calculating percentages, or using the Number of Records field.
- If you only have measures, you can't break out the values by anything. You can disaggregate the data entirely or work with the overall SUM or AVG, etc.

Which isn't to say a data set with only dimensions can't be useful. Demographic data is an example of dimension-heavy data, and much analysis around demographics is counting or percentage-based. But for a more analytically rich data set, you want at least a few dimensions and measures.

Dimensions and Measures, Discrete and Continuous



In the image above, note that the **Numeric Dimension** does not have an aggregation on the Marks card, unlike both the **Continuous Measure** and **Discrete Measure**.

Dimensions and Measures

Fields are broken out into dimensions and measures in the data pane. In Tableau, dimensions come out into the view as themselves, whereas measures are automatically aggregated; the default aggregation for a measure is SUM.

- **Dimensions** are qualitative, meaning they are described, not measured.
  - Dimensions are often things like city or country, eye color, category, team name, etc.
  - Dimensions are usually discrete.
- **Measures** are quantitative, meaning they can be measured and recorded (numeric).
  - Measures are often things like sales, height, number of clicks, etc.
  - Measures are usually continuous.

If you could do math with a field, it should be a measure. If you're ever not sure if a field should be a measure or a dimension, think about if you can do meaningful math with the values. Is there any meaning to AVG(RowID), the sum of two Social Security numbers, or dividing a postal code by 10? No. Those are dimensions that happen to be written as numbers. Think about how many countries have alphanumeric postal codes–they're just labels, even though in the US they're only numeric. Tableau can recognize many field names that indicate a numeric field is actually an ID or a postal code and tries to make those dimensions, but it's not perfect. Use the "could I do math with this?" test to decide if a numeric field should be a measure or dimension and rearrange the data pane as necessary.

> **Note**: Although you can do math with dates (such as the DATEDIFF calculation), the standard convention is to categorize dates as dimensions.

Discrete and Continuous

Discrete or Continuous fields are somewhat aligned with the concepts of dimension and measure, but they're not identical.

- **Discrete** fields contain distinct values. They make headers or labels in the view and the pills are blue
- **Continuous** fields "form an unbroken whole". They make an axis in the view and the pills are green

A good way of understanding discrete and continuous is to look at a date field. Dates can be either discrete OR continuous.

- Looking at average temperatures in August across a decade or century means "August" is being used as a discrete, qualitative date part.
- Looking at the overall trend in reported malaria cases since 1960 would take a single, unbroken axis, meaning the date is being used as a continuous, quantitative value.

For more information, see the Free Training Video on Understanding Pill Types, or the Help topic **Dimensions and Measures, Blue and Green** on page 314.

Fields Created by Tableau

Tableau will automatically create three fields, no matter what the data set is:

- *Measure Names* (a dimension)
- *Measure Values* (a measure)
- *Number of Records* (a measure)

And if there are geographic fields in the data set, Tableau will also create *Latitude (generated)* and *Longitude (generated)* fields.

Measure Names and **Measure Values** are two extremely useful fields. For more information, see the Free Training Video or the Help topic **Measure Values and Measure Names** on page 1310.

Number of Records is a field that basically assigns a “1” to every row in the data set. This enables you to have at least one measure in your data set and can help with some analysis. You must understand the granularity of your data (what a row represents) to be able to define what the number of records means.

Here, each row is a day, so the number of records would be the number of days:

Date	Max TemperatureF	Mean TemperatureF	Min TemperatureF
1/1/2015	42	34	26
1/2/2015	42	37	32
1/3/2015	41	38	35
1/4/2015	51	45	38
1/5/2015	54	52	49
1/6/2015	54	49	43
1/7/2015	46	44	42
1/8/2015	46	41	35
1/9/2015	50	44	38
1/10/2015	46	45	43

Here, each row is a month, so the number of records would be the number of months:

Date	Max TemperatureF	Mean TemperatureF	Min TemperatureF
January	63	45	26
February	62	49	33
March	69	51	31
April	77	52	37
May	82	59	43
June	92	68	49
July	95	71	54
August	92	69	54
September	81	61	45
October	74	58	45
November	60	44	25

4. A good data set has metadata or a data dictionary

A data set can only be useful if you know what the data *is*. There are few things more frustrating in the hunt for good data than opening a file that looks like this:

Diet	Source	Donor	Collection	Sex	OTU0	OTU1	OTU2	OTU3	OTU4
0	4	0	0	0	7.38E-14	9.05E-11	8.22E-11	3.86E-11	8.25E-11
1	12	1	0	0	1.54E-13	5.67E-11	3.12E-11	8.52E-11	1.73E-11
1	4	3	0	0	5.52E-13	3.46E-11	0.00077	4.76E-11	9.80E-11
0	4	0	0	1	5.69E-13	2.75E-11	8.08E-11	9.94E-11	5.90E-12
1	4	3	0	0	8.97E-13	8.33E-11	0.000274	8.49E-11	5.38E-11
0	4	0	0	0	9.43E-13	6.21E-11	1.41E-11	6.32E-11	9.17E-11
1	11	0	0	0	1.13E-12	1.85E-11	9.04E-11	4.09E-11	5.31E-11
1	4	0	0	0	1.49E-12	7.86E-12	8.87E-11	3.48E-11	8.16E-11
1	6	0	1	0	1.63E-12	4.42E-11	9.12E-11	8.53E-11	9.40E-12

What does a Source of 4 or 12 mean? And what information is in the fields OTU0-OTU4?

A good data set is one that has either well-labeled fields and members or a data dictionary so you can relabel the data yourself. Think of Superstore—it's immediately obvious what the fields and their values are, such as Category and its members Technology, Furniture, and Office Supplies. Or, for the microbiome data set in the image above, there is a data dictionary which explains each Source (4 is feces and 12 is stomach) and the taxonomy of each OTU (OTU3 is a bacterium of the genus *Parabacteroides*).

Data dictionaries can also be called metadata, indicators, variable definitions, glossaries, or any number of other things. At the end of the day, a data dictionary provides information about column names and members in a column. That information can be brought into the data source or viz in several ways, including:

- Rename the columns so they are easier to understand (this can be done in the data set itself or in Tableau).
- Re-alias the members of the field (this can be done in the data set itself or in Tableau).
- Create calculations to add the data dictionary information.
- Comment on the field in Tableau (comments do not appear on published vizzes, only in the authoring environment).
- Use the data dictionary as another data source and combine the two data sources.

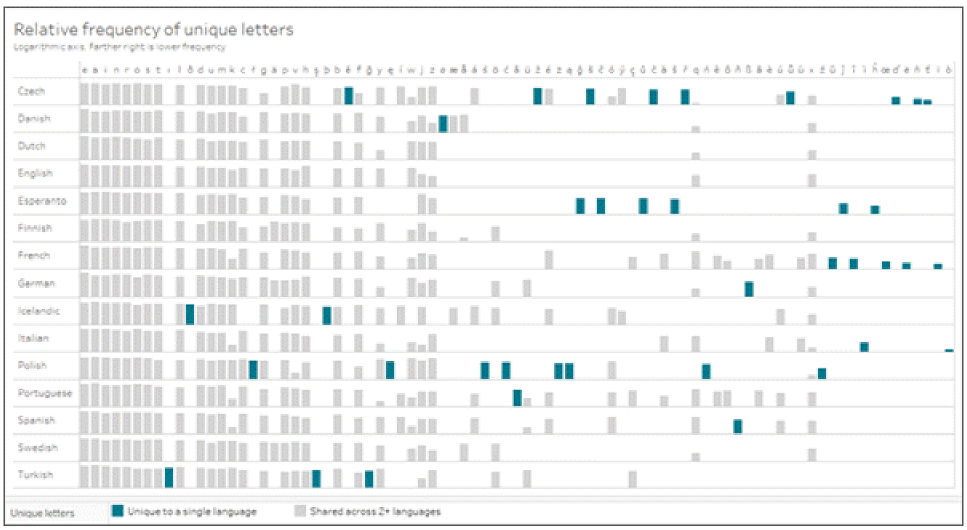
Losing a data dictionary can render a data set useless. If you're bookmarking a data set, bookmark the data dictionary, too. If you're downloading, download both and keep them in the same place.

5. A good data set is one you can use

As long as you can understand the data set and it has the information you need, even a small data set can pack a punch for analysis. Smaller data sets are also easy to store, share, and publish, and are likely to perform well.

Similarly, even if you find the “perfect” data set for your needs, if it requires an unrealistic amount of effort to clean up, it’s not perfect after all. Knowing when to walk away from a data set that is too messy is important.

For example, this data set is from a Wikipedia article on relative letter frequencies. It started as 84 rows and 16 columns (pivoted to be 1,245 rows and 3 columns). The Excel file is 16KB. But with some groups, sets, calculations, and other manipulations, it enables robust analysis and interesting visuals.



Click the image to download the workbook.

Relabel your data

Once you find a good data set, you’ll often need to relabel it. Relabeling data can be useful to either create fake data for samples or proof-of-concepts, or to make the data more readable.

Renaming a field changes how that field appears in Tableau, such as renaming “Sales” to “Pipeline Sales” or “State” to “Province”.

Re-aliasing changes how the members of a field are displayed, such as re-aliasing values in a Country field so that CHN becomes China and RUS becomes Russia.

- The values in a discrete dimension field are called **members**. Only members can be re-aliased. Consider a measure field for temperature. A value of 54°F can't be changed without changing data itself. But re-aliasing the member "CHN" as "China" in a Country field is the same information, just labeled in another way.

Renaming and re-aliasing mean almost the same thing. It's the convention in Tableau that fields are named and members are aliased. For more information, see **Organize and Customize Fields in the Data Pane** on page 1146 and **Create Aliases to Rename Members in the View** on page 1161.

> **Note**: Renaming or re-aliasing only changes the appearance in Tableau Desktop; no changes are written back to the underlying data.

Relabel to make fake data

Relabeling existing data sets is a great way to make samples or proof-of-concept content more compelling.

1. Use an easy data set (like Superstore) to build what you want (a specific chart type, showing off certain functionality, etc.)
2. Rename the relevant fields, change tooltips, and otherwise change the textual aspects to mask what the data actually represents.

Important: Only do this when it's clear the information is fake. Be careful that people don't think it's real data and try to use it for analysis. For example, use silly names or meaningless field names like colors or animals.

Re-alias to make the data easier to use

It's more efficient to store the data as numeric values rather than string values, though numeric encoding can make the data harder to understand. For small data sets it probably won't make a performance impact, so prioritize being able to understand the data easily.

A downside to re-aliasing is that you no longer have access to those numeric values (making it harder to do things like sort, assign color gradients, etc.). Consider duplicating the field and re-aliasing the copy. Alternatively, a calculation in Tableau can be a good way to preserve the original information while also making it more easily understandable.

Re-alias with the CASE function

Calculations can be very powerful for re-aliasing. For example, CASE functions allow you to say, essentially, "when this field has a value of A, give me X. When the value is B, give me Y".

Here, the CASE function looks at the F-scale in a tornado data set and provides the written description associated with each numeric value:

```
CASE [F-scale]
WHEN "0" THEN "Some damage to chimneys; branches broken off
trees; shallow-rooted trees pushed over; sign boards damaged."
WHEN "1" THEN "The lower limit is the beginning of hurricane wind
speed; peels surface off roofs; mobile homes pushed off
foundations or overturned; moving autos pushed off the roads..."
WHEN "2" THEN "Roofs torn off frame houses; mobile homes
demolished; boxcars overturned; large trees snapped or uprooted;
highrise windows broken and blown in; light-object missiles
generated."
WHEN "3" THEN "Roofs and some walls torn off well-constructed
houses; trains overturned; most trees in forest uprooted; heavy
cars lifted off the ground and thrown."
WHEN "4" THEN "Well-constructed houses leveled; structures with
weak foundations blown away some distance; cars thrown and large
missiles generated."
WHEN "5" THEN "Strong frame houses lifted off foundations and
carried considerable distances to disintegrate; ... trees
debarked; steel reinforced concrete structures badly damaged."
END
```

Now we can chose to use either the original "F-scale" field (0-5) or the "F-scale damage description" field in the viz.

Tips when looking for data sets

Note: Try to make sure you can answer the question "What does a row (aka record) in the data set represent?" If you can't articulate that, you might not understand the data well enough to be able to use it or it might be structured poorly for analysis.

- Keep track of where the data came from.
- Keep the data dictionary information with the data itself.
- Avoid stale data if you need the content to stay evergreen. Look for:
  - updatable data (stocks, weather, regularly published reports, etc.)
  - timeless data (the average mass of various animals isn’t going to change from year to year)
  - data you can future-proof by artificially changing to historical or future dates
- Try simply Googling what you’re looking for, you might be surprised.
- Don’t be afraid to give up on a data set if it’s too much work to prep.

Places to look for data

Where can you look for data? There are a potentially overwhelming number of places to find data sets. Here are some options to get you started. Note that the reality of data sets does apply to these sites—you probably won’t find what you’re thinking of right now, and you will most likely need to do some cleaning to get the data ready for analysis.

> **Disclaimer**: Although we make every effort to ensure these links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Listing a site here is not an endorsement of any content or organization. Contact the external site for answers to questions regarding its content.

Tableau Public: Tableau Public is an amazing resource for Tableau-friendly data sets. Search for workbooks that are on a topic you’re interested in, browse for inspiration, then download the workbook to access the data. Or check out the curated Sample Data.

Wikipedia tables: Get data out of Wikipedia tables by: copying and pasting into a spreadsheet, copying and pasting directly into Tableau, or using Google sheets and the IMPORTHTML function to create a Google spreadsheet of the data.

Google Dataset Search: "A search engine to unite the fragmented world of online datasets."

Data is Plural : Subscribe for a weekly newsletter with data sets, or browse the archive.

Makeover Monday: “Join us every Monday to work with a given data set and create better, more effective visualizations and help us make information more accessible.” You can see what other people have done with the same data set, kickstarting your analysis or giving inspiration. Use #makeovermonday on Twitter to participate.

Other sites

- Tableau Web Data Connectors
- Data.world and its WDC for Tableau
- Github Open Data
- Kaggle
- datahub.io
- r/datasets
- WHO
- Data.UN.org
- WorldBank
- data.gov, data.gov.au, data.gov.za, data.gov.uk, etc.
- Airbnb
- Yelp
- Zillow

Dimensions and Measures, Blue and Green

When you connect to a new data source, Tableau assigns each field in the data source as dimension or measure in the **Data** pane, depending on the type of data the field contains. You use these fields to build views of your data.

> **Watch a Video**: To see related concepts demonstrated in Tableau, watch Understanding Pill Types, a 5-minute free training video. Use your tableau.com account to sign in.

About data field roles and types

Data fields are made from the columns in your data source. Each field is automatically assigned a data type (such as integer, string, date), and a role: Discrete Dimension or Continuous Measure (more common), or Continuous Dimension or Discrete Measure (less common).

- *Dimensions* contain qualitative values (such as names, dates, or geographical data). You can use dimensions to categorize, segment, and reveal the details in your data. Dimensions affect the level of detail in the view.
- *Measures* contain numeric, quantitative values that you can measure. Measures can be aggregated. When you drag a measure into the view, Tableau applies an aggregation to that measure (by default).

Blue versus green fields

Tableau represents data differently in the view depending on whether the field is discrete (blue), or continuous (green). *Continuous* and *discrete* are mathematical terms. Continuous means "forming an unbroken whole, without interruption"; discrete means "individually separate and distinct."

- Green measures SUM(Profit) and dimensions YEAR(Order Date) are continuous. Continuous field values are treated as an infinite range. Generally, continuous fields add axes to the view.

- Blue measures SUM(Profit) and dimensions Product Name are discrete. Discrete values are treated as finite. Generally, discrete fields add headers to the view.

Possible combinations of fields in Tableau

This table shows examples of what the different fields look like in the view. People sometimes call these fields "pills", but we refer to them as "fields" in Tableau help documentation.

Discrete Dimensions	Product Name
Continuous Dimensions (dimensions with a data type of String or Boolean cannot be continuous)	YEAR(Order Date)
Discrete Measures	SUM(Profit)
Continuous Measures	SUM(Profit)

Note: With a cube (multidimensional) data source, the options for changing data roles are limited. In Tableau Desktop, cubes are supported only on Windows.) You can change some measures from continuous to discrete, but in general, you cannot change data roles for fields in cube data sources. For related details, see **Cube Data Sources** on page 1085.

A visual cue that helps you know when a field is a measure is that the field is aggregated with a function, which is indicated with an abbreviation for the aggregation in the field name, such as:

. To learn more about aggregation, see **List of Predefined Aggregations in Tableau** on page 345 and **Aggregate Functions in Tableau** on page 2192.

But there are exceptions:

- If the entire view is disaggregated, then by definition no field in the view is aggregated. For details, see **How to Disaggregate Data** on page 350.
- If you are using a multidimensional data source, fields are aggregated in the data source and measures fields in the view do not show that aggregation.

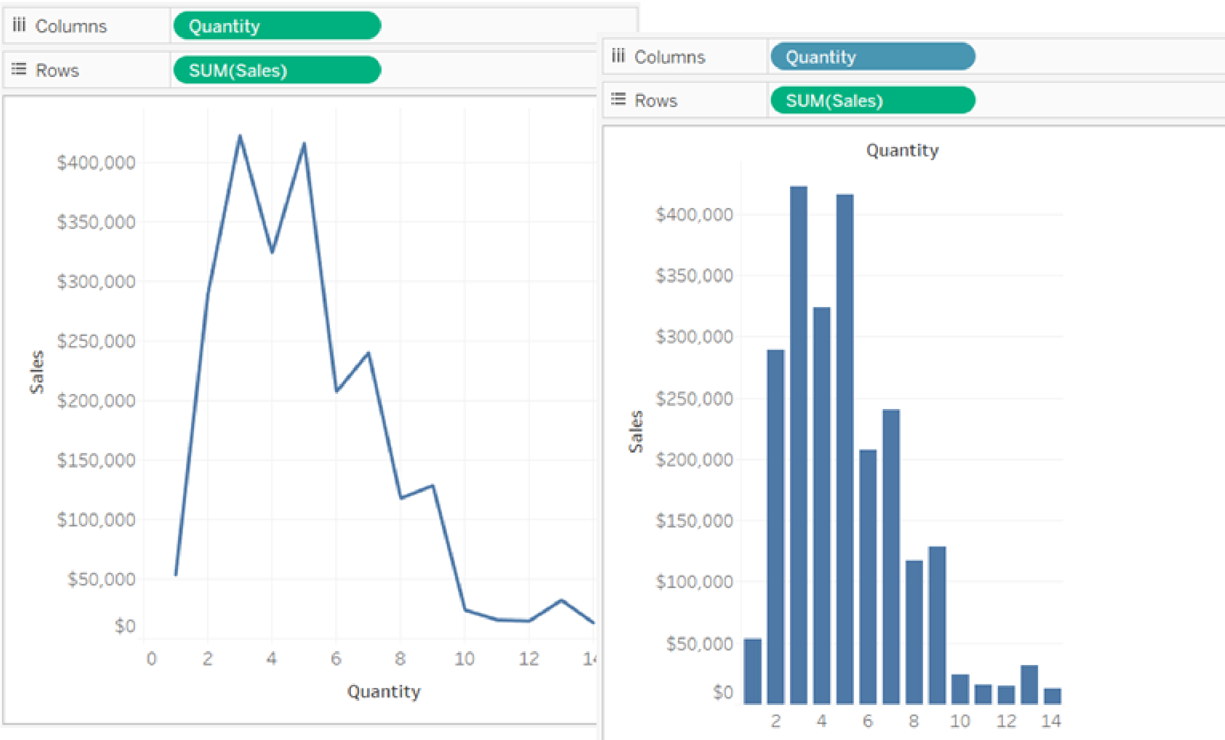
Note: You can set the default aggregation and other properties and settings for fields. For details on the many ways you can customize the fields in the Data pane, see **Organize and Customize Fields in the Data Pane** on page 1146, **Edit Default Settings for Fields** on page 1157, and **Work with Data Fields in the Data Pane** on page 203.

Examples of continuous and discrete fields used in a view

In the example on the left (below), because the **Quantity** field is set to **Continuous**, it creates a horizontal axis along the bottom of the view. The green background and the axis help you to see that it's a continuous field.

In the example on the right, the **Quantity** field has been set to **Discrete**. It creates horizontal headers instead of an axis. The blue background and the horizontal headers help you to see that it's discrete.

Continuous **Discrete**

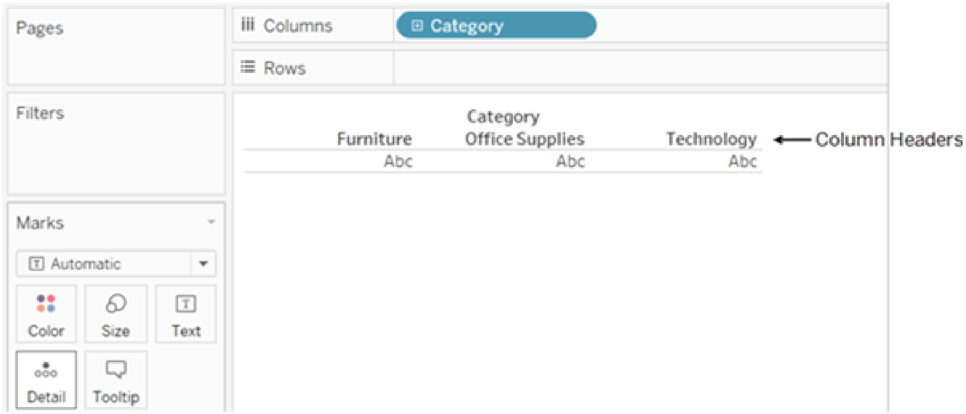


In both examples, the **Sales** field is set to **Continuous**. It creates a vertical axis because it continuous and it's been added to the Rows shelf. If it was on the Columns shelf, it would create a horizontal axis. The green background and aggregation function (in this case, SUM) help to indicate that it's a measure.

The absence of an aggregation function in the **Quantity** field name help to indicate that it's a dimension.

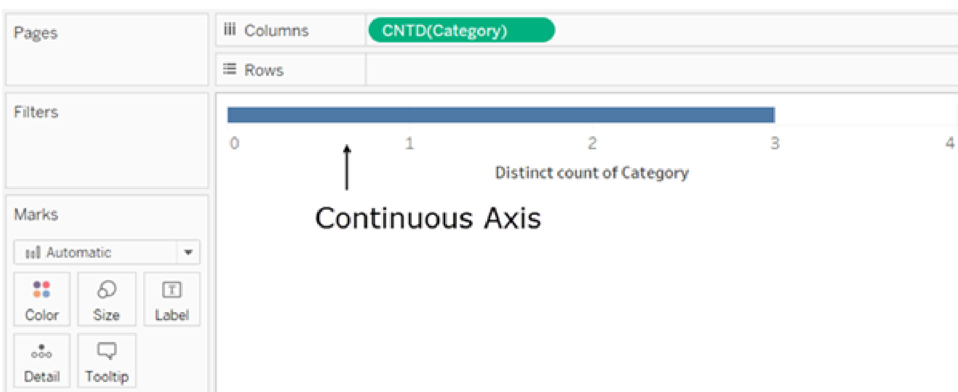
Dimension fields in the view

When you drag a discrete dimension field to **Rows** or **Columns**, Tableau creates column or row headers.



In many cases, fields from the **Dimension** area will initially be discrete when you add them to a view, with a blue background. Date dimensions and numeric dimensions can be discrete or continuous, and all measures can be discrete or continuous.

After you drag a dimension to **Rows** or **Columns**, you can change the field to a measure just by clicking the field and choosing **Measure**. Now the view will contain a continuous axis instead of column or row headers, and the field's background will become green:



Date dimensions can be discrete or continuous. Dimensions containing strings or Boolean values cannot be continuous.

Tableau does not aggregate dimensions. For a discussion of the different types of aggregation Tableau can perform, see **List of Predefined Aggregations in Tableau** on page 345.

In Tableau queries, dimensions in the view are expressed in SQL as "Group By" clauses.

For details on converting fields between continuous and discrete, see **Convert Fields between Discrete and Continuous** on page 1164.

How dimensions affect the level of detail in the view

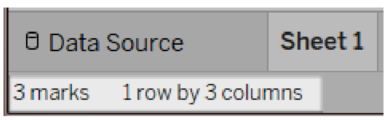
The level of detail in a view refers to how granular the data is given the dimension and measure data in the view.

As you add dimensions to **Rows** or **Columns**, the number of marks in the view increases.

To understand why adding dimensions increases the number of marks in the view, do the following:

1. Drag **Segment** to **Columns**.

   The status bar at the bottom of the Tableau window shows you that there are now three marks in the view:

   Those marks just contain placeholder text, `Abc`, because you are only building the view's structure at this point.

2. Drag **Region** to **Columns**.

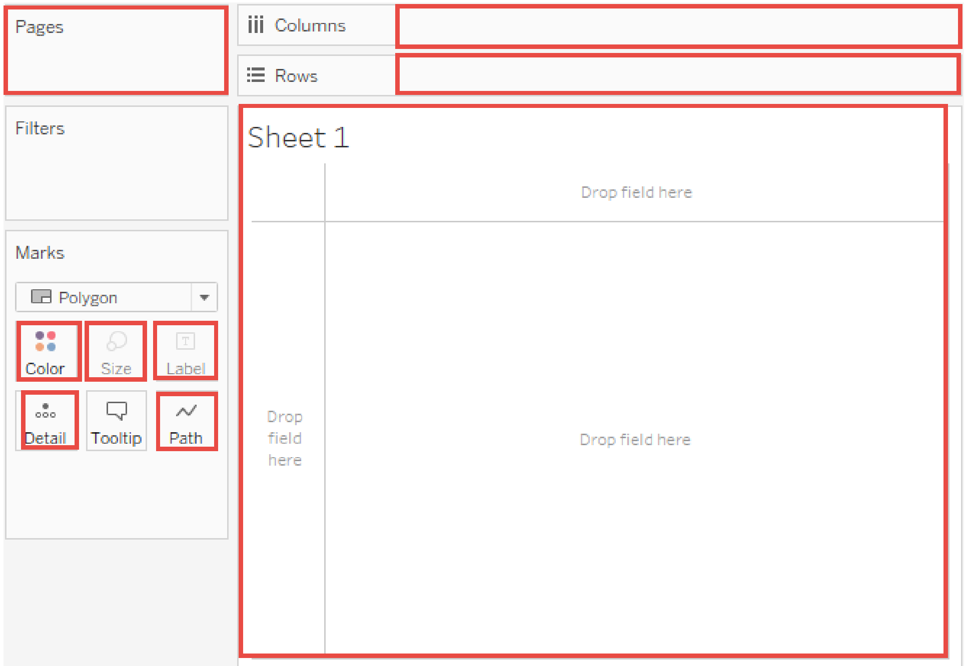
   Now there are 12 marks. Three values in **Segment** multiplied by four values in **Region** is 12.

3. Drag [**Ship Date**] to **Rows**.

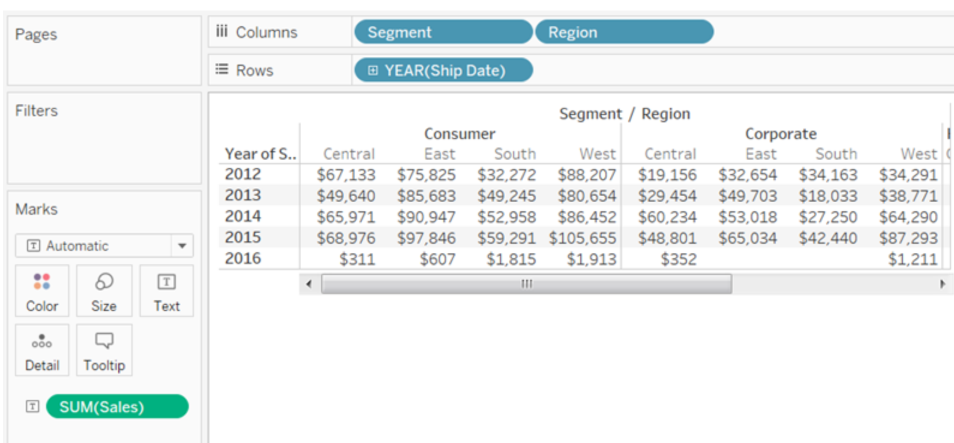
   The total is now 57 marks (three segments by four regions by five years is 60, but there are three combinations of the dimensions in the view for which there is no data in the data source).

   We could continue adding dimensions to **Rows** and **Columns** and observe as the number of total marks continues to increase. Dragging a dimension to a location on the Marks card such as Color or Size will also increase the number of marks, though it will not increase the number of headings in the view. The process of adding dimensions to the view to increase the number of marks is known as setting the *level of detail*.

   Adding a dimension to any of the following locations in Tableau affects the level of detail:



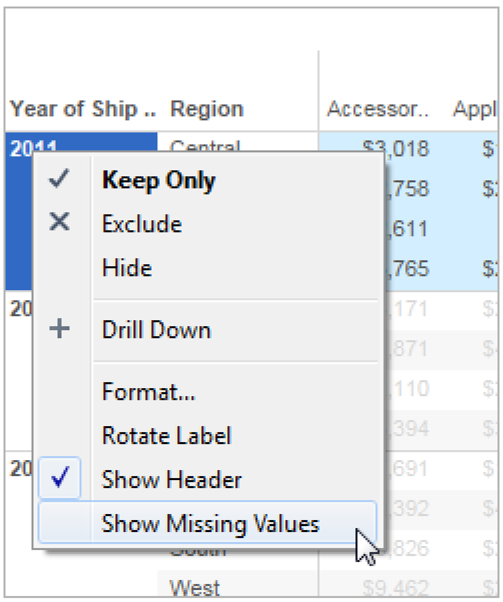
4. The view now contains 57 separate instances of `Abc`—the view is all structure and no content. Rectify this by dragging **Sales** to Text. The view can now be considered complete:



Notes

- In some cases, adding a measure to the view can increase the number of marks in the view. For example, if you dropped **Sales** on **Rows** in the view above, the number of marks would be 57. But if you then also dropped **Profit** on **Rows**, the number of marks would increase to 114. But this is not the same as changing the view's level of detail.

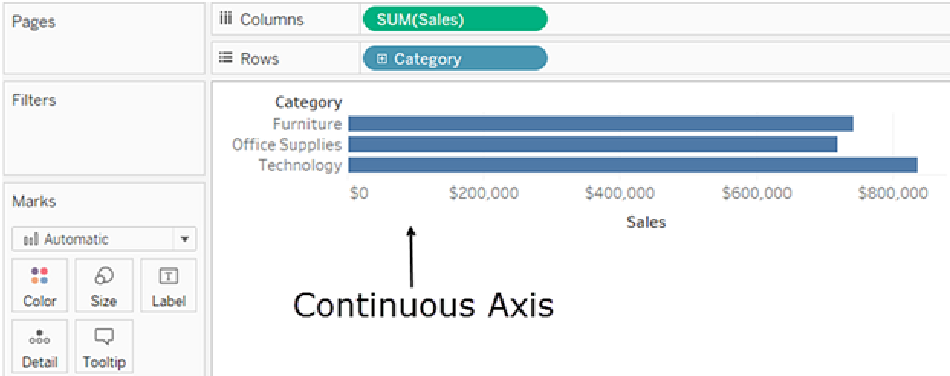
- The number of marks in the view is not guaranteed to correspond to the number you would get by multiplying the number of dimension values in each of the dimensions that make up the level of detail. There are multiple reasons why the number of marks could be lower. To increase the number of marks in this view from 57 to 60 in the view above, right-click (Control-click on a Mac) on one of the Date headers in the view and the date or bin headers and choose **Show Missing Values**. For more information about how to show missing values, see **Show or Hide Missing Values or Empty Rows and Columns** on page 1321.



Measure fields in the view

When you drag a measure to the view, it is aggregated by default. The type of aggregation will vary depending on the type of view. You should always check the aggregation and change it if necessary. For details, see "Change the default aggregation" in **Edit Default Settings for Fields** on page 1157. For more details about aggregation, see **Data Aggregation in Tableau** on page 341.

When you drag a continuous field from the **Data** pane to **Rows** or **Columns**, Tableau creates a continuous axis for that field.



If you click the field and change it to **Discrete**, the values become column headers.

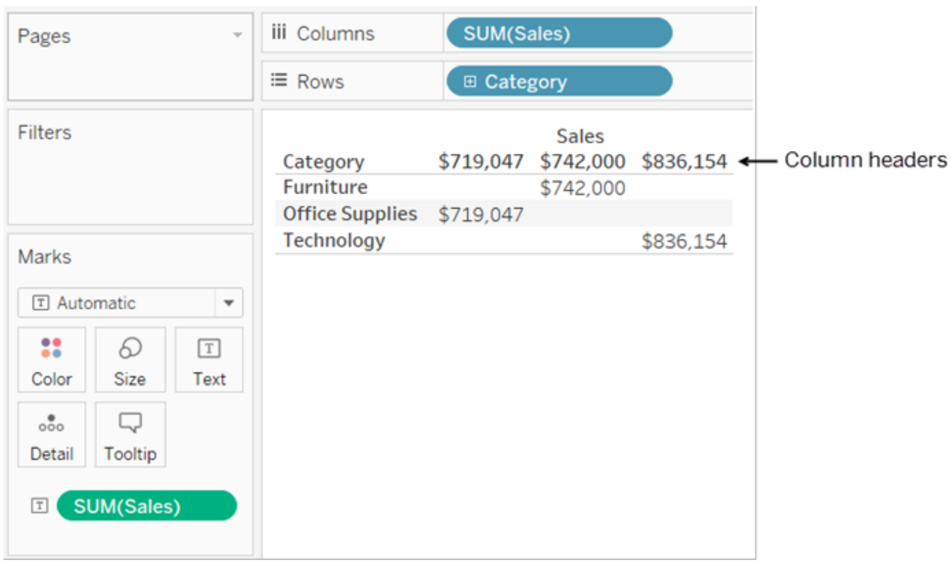


Tableau continues to aggregate values for the field, because even though the field is now discrete, it is still a measure, and Tableau aggregates measures by default.

In cases where Tableau has misclassified a field as a dimension or a measure, possibly because of the data type, you can convert it and change its role. If a measure contains numbers that don't need to be aggregated (such as a field that contains date values), you may want to convert it to be a dimension.

For related details, see **Convert a Measure to a Dimension** on page 1165.

For details on converting fields between continuous and discrete, see **Convert Fields between Discrete and Continuous** on page 1164.

How continuous and discrete fields change the view

Continuous and *discrete* are mathematical terms. Continuous means "forming an unbroken whole, without interruption"; discrete means "individually separate and distinct."

In Tableau, fields can be either continuous or discrete. When you drag a field from the **Data** pane to **Columns** or **Rows**, the values are continuous by default and Tableau creates an axis. When you drag a field from the **Data** pane to **Columns** or **Rows**, the values are discrete by default and Tableau creates column or row headers.

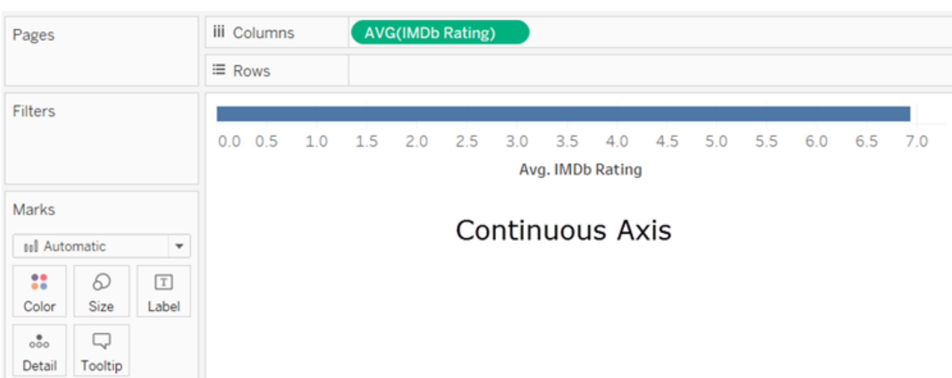
Continuous fields produce axes

If a field has values that are numbers that can be added, averaged, or otherwise aggregated, it is added as a measure field in the **Data** pane when you first connect to a data source. Tableau is assuming that the values are continuous.

Tableau displays an axis when you drag a continuous field to **Rows** or **Columns**. An axis is a measuring line that shows values between a minimum and a maximum. Rulers and analog thermometers are examples of physical objects that display axes.

Tableau must be able to show a range of actual and potential values, because in addition to the initial values in the data source, it is always possible that new values will emerge as you work with a continuous field in the view.

While there are value labels on a continuous axis (0, 0.5, ... 3.0 in the following image), actual marks don't have to align with these labels as they would with column headers. For example, in the following image, the blue bar actually extends to a value of 6.940 on the horizontal axis, not 7.0 exactly.



The number of potential values for continuous fields is impossible to anticipate. For example, if you have a field named **Ratings** and the initial values are 1, 3, 3.5, 3.6, and 4, that's five distinct values. But if you drop **Ratings** on **Rows**, Tableau automatically aggregates that value as SUM (which you would then immediately change to AVG, because it's more logical to average grades than to add them), and that would then create a sixth value (3.02) that didn't exist until you added the field to the view. And if you then applied a filter that eliminated two of the initial values, the average would change as well, so that would be yet another value. And then if you changed the aggregation, ... You get the idea. The number of potential values is, if not infinite, then certainly immense.

The fact that a field contains numbers does not automatically indicate that those values are continuous. Postal codes are the classic example: though they are often composed entirely of numbers, they are actually string values which shouldn't be added or averaged. If Tableau assigns such a field as a measure to the **Data** pane, you should drag it to a dimensions area to change it to a dimension field.

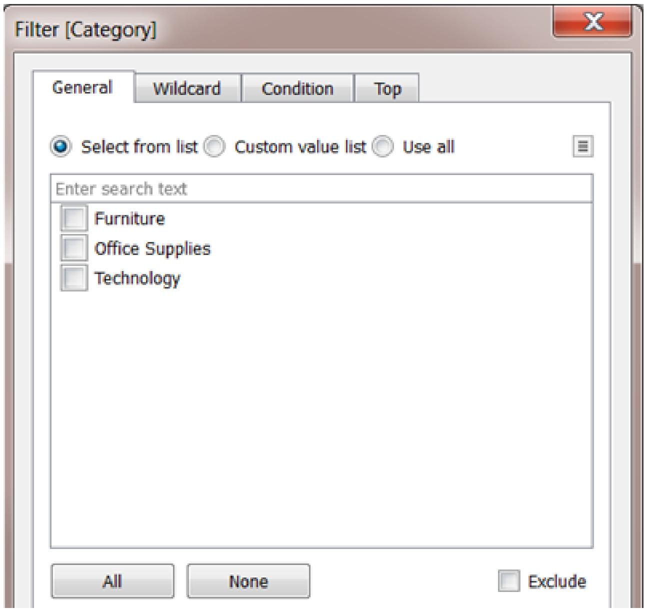
Discrete fields create headers

If a field contains values that are names, dates, or geographical locations—anything other than numbers—it is added as a dimension field in the **Data** pane when you first connect to a data source. Tableau treats the values as discrete.

Tableau creates headers when you drag a discrete field to **Columns** or **Rows**. The individual values for a discrete field become the row or column headings. Because these types of values are never aggregated, no new field values are created as you work with your view, so there is no need for an axis.

Discrete versus continuous fields on filters

- When you drop a discrete dimension field on the **Filters** shelf, Tableau prompts you to choose which "members" of the discrete field to include in the view.



- When you drop a Date field on Filters, the result can be a discrete filter or a continuous filter. For more information, see **Filter dates** on page 1417.
- When you drop a continuous measure on Filters, Tableau first prompts you to choose an aggregation for the filter, and then prompts you to specify how to filter the continuous range of values.
- When you drop a continuous dimension on Filters (other than a Date), Tableau prompts you to specify how to filter the continuous range of values.

For more on filtering various types of fields, see **Drag dimensions, measures, and date fields to the Filters shelf** on page 1413.

Discrete versus continuous fields on color

When you drop a discrete field on Color in the Marks card, Tableau displays a categorical palette and assigns a color to each value of the field.

When you drop a continuous field on Color, Tableau displays a quantitative legend with a continuous range of colors.

For more information about color palettes, see **Color Palettes and Effects** on page 1394.

Data Types

All fields in a data source have a data type. The data type reflects the kind of information stored in that field, for example integers (410), dates (1/23/2015) and strings (“Wisconsin”). The data type of a field is identified in the Data pane by one of the icons shown below.

Data type icons in Tableau

Icon	Data type
	Text (string) values
	Date values
	Date & Time values
	Numerical values
	Boolean values (relational only)
	Geographic values (used with maps)
	Cluster Group (used with **Find Clusters in Data** on page 2554)

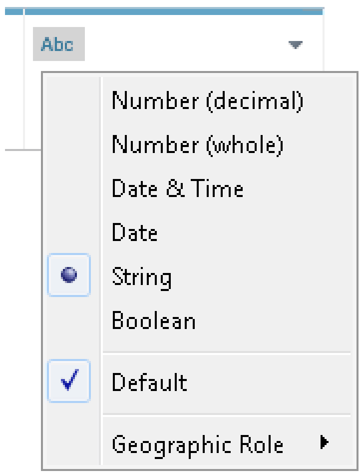
You can change the data type for a field either on the **Data Source** page or in the **Data** pane.

Change the data type for a field in the Data Source page

Sometimes Tableau incorrectly interprets the data type of a field. For example, Tableau might interpret a field that contains dates as an integer data type, rather than a date data type.

You can change the data type for a field that was part of the original data source (as opposed to a calculated field created in Tableau) on the **Data Source** page.

1. Click the data type icon for the field (as shown in the table above).
2. Choose a new data type from the drop-down list:

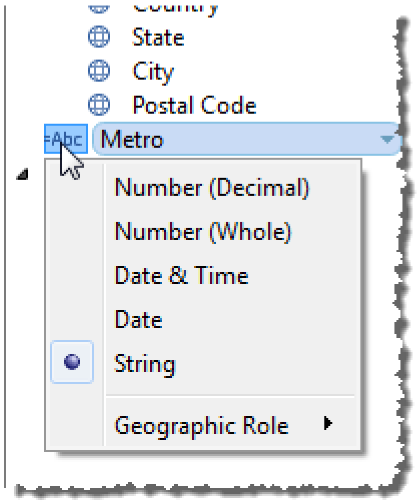


Tip: Be sure to change data types before you create an extract. Otherwise, your data may not be accurate. For example, if a floating-point field in the original data source is interpreted as an integer by Tableau, and you create your extract before you change the field's data type, the resulting floating-point field in Tableau will have some of its precision truncated.

For information on changing data types on the **Data Source** page, see **Data Source Page** on page 188.

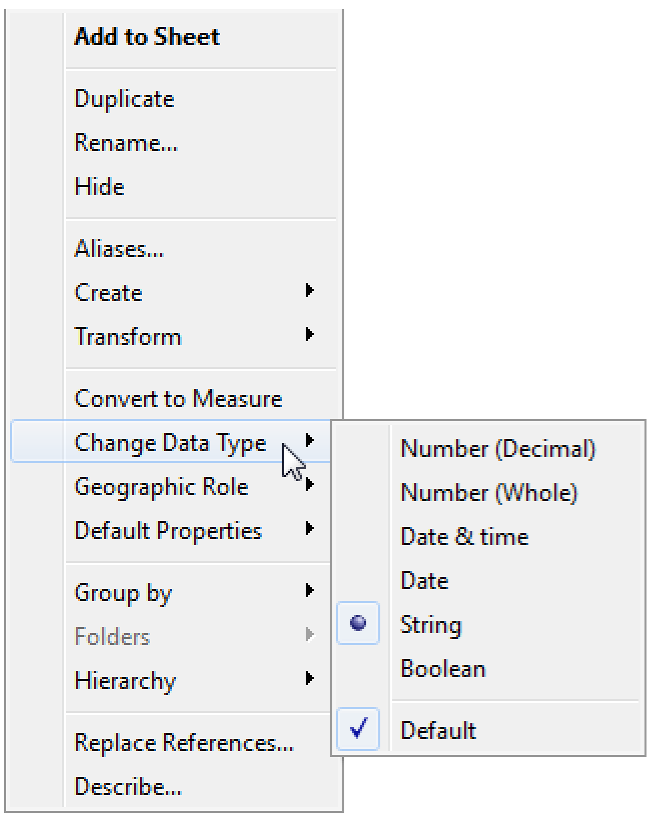
Change the data type for a field in the Data pane

To change the data type of a field in the **Data** pane, click the icon to the left of the field name, and then choose a new data type from the drop-down list.



Change the data type for a field in the view

To change a field's data type in a view, right-click (control-click on a Mac) the field in the **Data** pane, choose **Change Data Type**, and then select the appropriate data type from the drop-down list.



Note: Sometimes the data in your database is more precise then Tableau can model. When you add these values to the view, a precision warning appears in the right corner of the status bar. See **Status Bar Information** on page 200.

Mixed data types in data from files

Most columns in a Microsoft Excel, Microsoft Access, or CSV (comma-separated value) file contain values that are all of the same data type (Booleans, dates, numbers, or text). When you connect to the file, Tableau creates a field in the appropriate area of the **Data** pane for each column. Dates and text values are dimensions, and numbers are measures.

However, files that you connect to might include columns that have a mixture of data types, such as numbers and text, or numbers and dates. When you connect to the file, the mixed-value column is mapped to a field with a single data type in Tableau. Therefore, a column that contains numbers and dates might be mapped as a number data type (making it a measure) or it might be mapped as a date data type (in which case Tableau treats it as a dimension.)

Tableau determines how to map mixed-value columns to data types by the data types of the first 10,000 rows in an Excel data source, and the first 1,024 rows in a CSV data source. For example, if most of the first 10,000 rows are text values, the entire column is mapped to use the text data type.

Note: Empty cells also create mixed-value columns because their formatting is different from text, dates, or numbers.

When Tableau determines a data type for each field, if the values in a field don't match that data type, Tableau handles the field in one of several different ways, depending on the data type. For example, sometimes Tableau populates those fields with Null values, as shown in the following table:

Mapped data type	**Treatment of other data types in the field.**
Text	Dates and numbers are treated as text. Nulls are not created.
Dates	Text is treated as Null. A number is treated as the day in numeric order from 1/1/1900.
Numbers	Text is treated as Null. A date is treated as the number of days since

Mapped data type	Treatment of other data types in the field.
	1/1/1900.
Boolean	Text, dates, and numbers are treated as Null.

If using fields that are based on mixed-value columns introduces difficulties when analyzing your data, you can do one of the following:

- Format empty cells in your underlying data source so that they match the data type of the column.
- Create a new column that does not contain the mixed values.

Tableau's Order of Operations

The order of operations in Tableau, sometimes called the query pipeline, is the order in which Tableau performs various actions. Actions are also known as operations. Many operations apply filters, which means that as you build a view and add filters, those filters always execute in the order established by the order of operations.

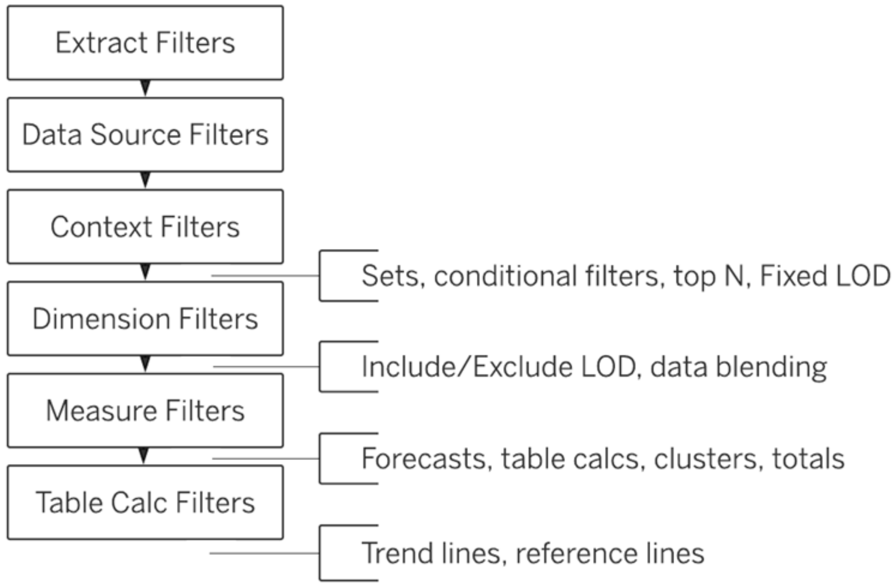
> **Watch a Video**: To see related concepts demonstrated in Tableau, watch Understanding Tableau's Order of Operations, a 1-hour, free video presentation.

This article includes two scenarios for updating a view to correct problems resulting from the order of operations: converting a dimension filter to a context filter, and converting a table calculation to a FIXED level of detail expression.

About the order of operations (aka query pipeline)

Sometimes, you might expect Tableau to execute filters in one order, but the order of operations dictates that they be executed in a different order, which gives you unexpected results. When this happens, you can sometimes change the order in which operations are executed in the pipeline.

The Tableau order of operations includes the following, from top to bottom.



> **Note**: In the order of operations, the latest date filter is global to the workbook, while context filters apply per worksheet. The latest date is determined just after the workbook opens for first use, after data source filters, but before context filters. At that point the date is set, and the latest date preset is used as a dimension filter.

Example 1: Convert a Dimension Filter to a Context Filter

This and the following example use the **Sample – Superstore** data source provided with Tableau Desktop.

In this example, the view addresses the following question: Who are the top 10 customers, by total sales, in New York City?

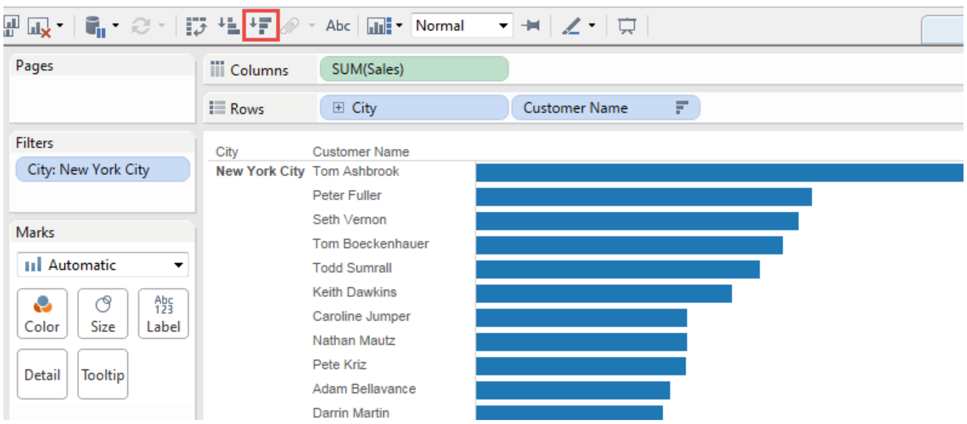
The view contains two dimension filters, one that you create on the **General** tab in the Filters dialog box, and the other on the **Top N** tab. The problem is that these filters are executing simultaneously, whereas you would like to general filter to be applied before the top n filter, so that the top n filter can act on the results as previously filtered by the general filter. The solution is to redefine one of the filters as a context filter so that a clear order of precedence is established.

Here are the steps for building this view.

1. Drag **Sales** to **Columns**.
2. Drag **City** and **[Customer Name]** to **Rows**.
3. Drag **City** from the **Data** pane again, this time to **Filters**. On the **General** tab in the Filter dialog box, set the filter to show just a single value: New York City. Do this by clicking **None** and then choosing **New York City**.

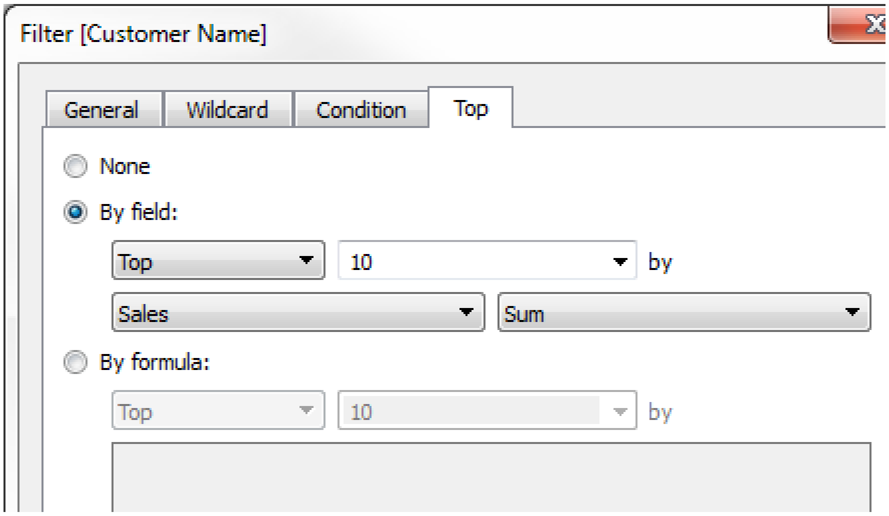
   This creates a general dimension filter.

4. Click the **Sort Descending** button () on the toolbar. Your view now looks like this:

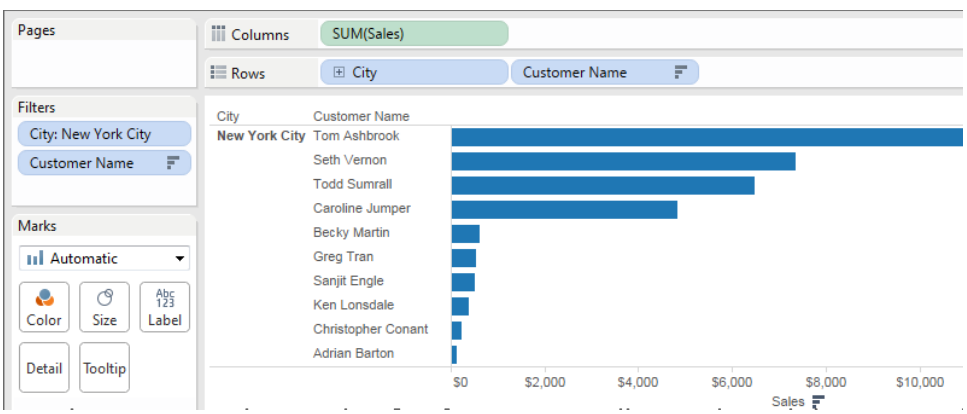


   Note the first few names in the list: Ashbrook, Fuller, Vernon, etc.

5. Now drag **[Customer Name]** from the **Data** pane to **Filters**, and create a Top 10 Filter, to show only the top 10 customers by total sales:

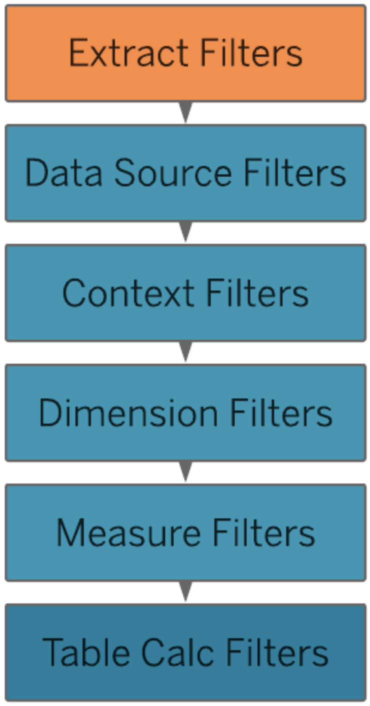


After you apply this second filter, the view looks right, but notice that the names shown are no longer the same as before:



What happened to Peter Fuller, formerly in second place? The goal was to show the top 10 customers in New York City, but now the view is actually showing the top 10 customers overall.

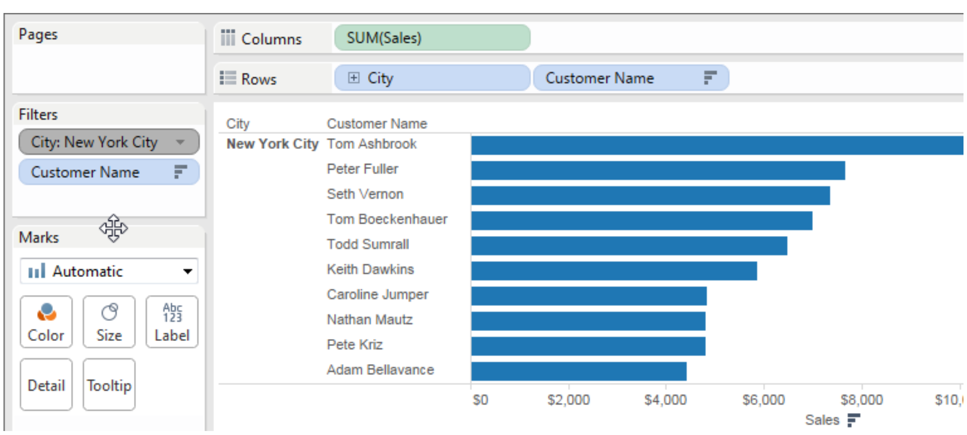
The problem is that top and general dimension filters are applied simultaneously—they are both dimension filters, and they appear in the same place in the Tableau order of operations:



The solution is to add the general dimension filter (on **City**) to context—that is, by turning it into a context filter, which is executed before any other filter that you create in a worksheet.

For details, see **Improve View Performance with Context Filters** on page 1439.

6. Right-click **City** on the **Filters** shelf (Control-click on a Mac) and choose **Add to Context**. As a context filter, this filter now takes precedence over the dimension filter, and so the view now shows what it's supposed to:



Example 2: Convert a Table Calculation to a FIXED Level of Detail Expression

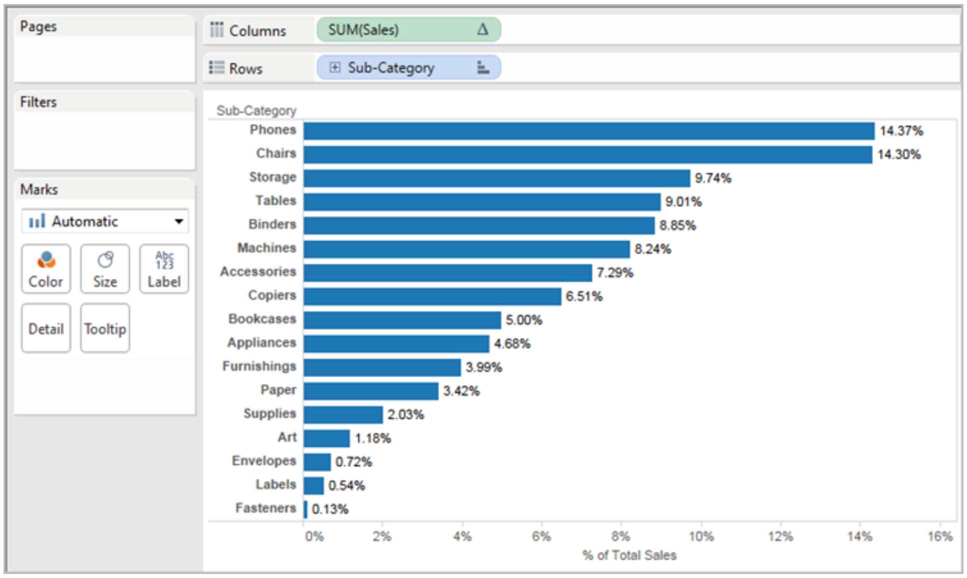
In this example, the view addresses the following question: What is the percent of total sales by product sub-category?

The view contains a dimension filter and a table calculation. Tableau applies the dimension filter before executing the table calculation. To invert the order of these operations, use a FIXED level of detail expression instead of a table calculation.

Here are the steps for building this view.

1. In a new worksheet, drag **Sales** to **Columns**.
2. Drag **Sub-Category** to **Rows**.
3. Right-click **SUM(Sales)** on **Columns** and select a quick table calculation – **Percent of Total**.
4. Click the **Sort Descending** button () on the toolbar to sort the categories from most to least.
5. Click the **Show Mark Labels** button () on the toolbar to display measure values in the view.

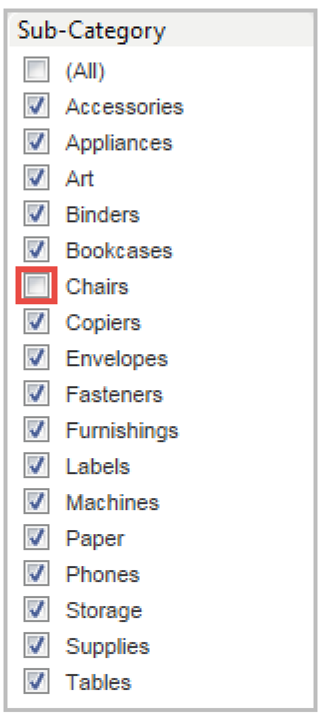
   Your view now looks like this:



Note the percentages for the first few items: 14.37%, 14.30%, etc.

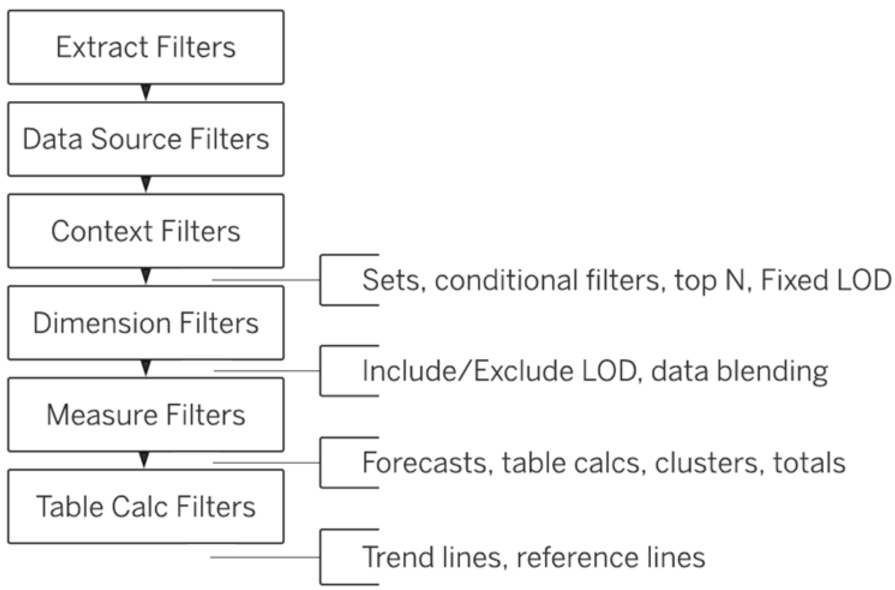
6. Right-click **Sub-Category** on **Rows** and choose **Show Filter**.

7. Clear the check mark for **Chairs** in the filter.



In the view, the percentages are now different—the highest percentage is now over 16%. In some cases, this may be just the result that you want (that is, for percentages to be recalculated as you work with the quick filter). But in other cases you may want the percentages to hold steady even as you filter items in or out. That's what we want in this case.

In the order of operations, a dimension filter is applied before a table calculation:



To have Tableau calculate the percentages before it acts on the quick filter, you create a FIXED level of detail expression, and then use that instead of the table calculation.

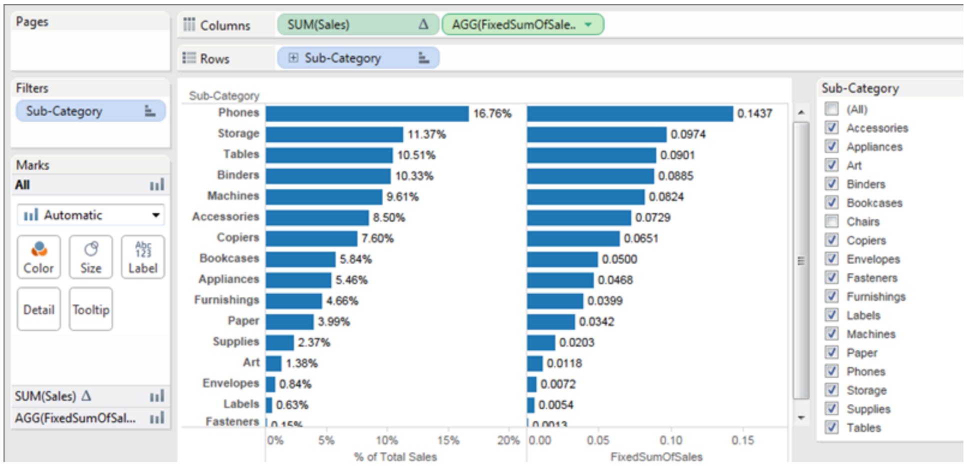
FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view. In this case you'll use it to establish percentages for the various sub-categories—percentages that won't be affected by your general dimension filter. Why? Because FIXED level of detail expressions are computed before dimension filters are applied.

For details, see **Create Level of Detail Expressions in Tableau** on page 2449.

8. The FIXED level of detail expression must divide the sum of **Sales** (for a particular measure value) by the total sum of **Sales** for the view. Because the numerator is aggregated, the denominator must be as well, so the expression you write is:

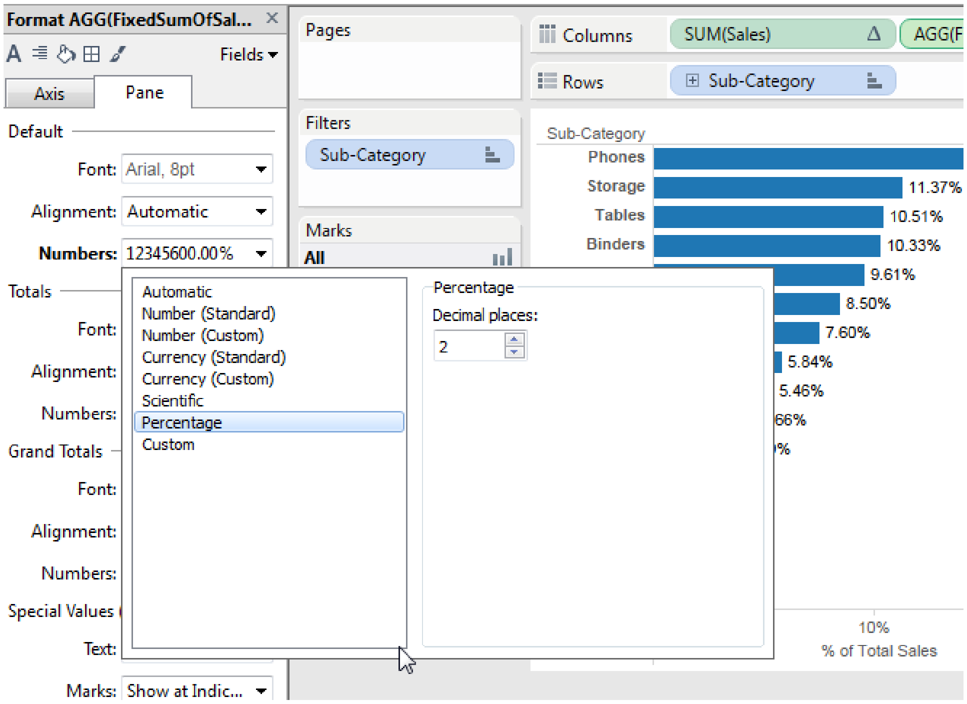
```
SUM([Sales])/SUM({FIXED : SUM([Sales])})
```

9. Save that expression as **FixedSumOfSales** and then drag it from the **Data pane** to **Columns**, dropping it to the right of the existing **SUM(Sales)** field that uses the table calculation. (Keep them both in the view for comparison.) Here is what your view now looks like:

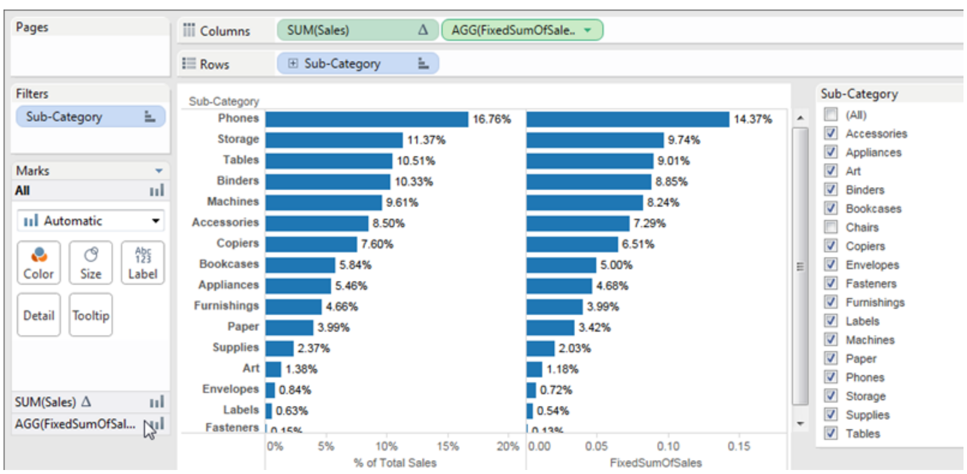


The percentage numbers in the chart on the right are now consistent, regardless of which fields you select or don't select with the quick filter. All that remains is to format the values for **FixedSumOfSales** so that they show as percentages.

10. Right-click **FixedSumOfSales** on **Columns** and choose **Format**. In the **Format** pane, choose **Numbers** and then **Percentage**:



This gives you the final view:



As you select or clear items in the **Sub-Category** quick filter, the percentages in the bar chart on the left change, but the percentages in the bar chart on the right do not.

Data Aggregation in Tableau

In Tableau, you can aggregate measures or dimensions, though it is more common to aggregate measures. Whenever you add a measure to your view, an aggregation is applied to that measure by default. The type of aggregation applied varies depending on the context of the view.

> **Watch a Video**: To see related concepts demonstrated in Tableau, watch Aggregation, Granularity, and Ratio Calculations, a 4-minute free training video. Use your tableau.com account to sign in.

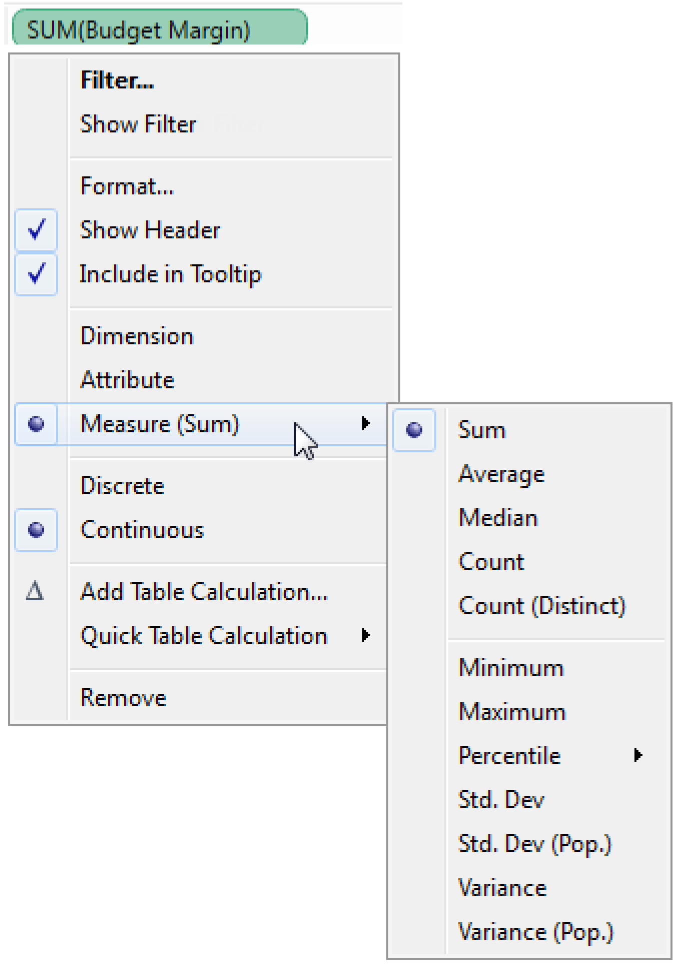
Change the Aggregation of a Measure in the View

When you add a measure to the view, Tableau automatically aggregates its values. Sum, average, and median are common aggregations; for a complete list, see **List of Predefined Aggregations in Tableau** on page 345.

The current aggregation appears as part of the measure's name in the view. For example, **Sales** becomes **SUM(Sales)**. Every measure has a default aggregation which is set by Tableau when you connect to a data source. You can view or change the default aggregation for a measure—see **Set the Default Aggregation for a Measure** on page 348.

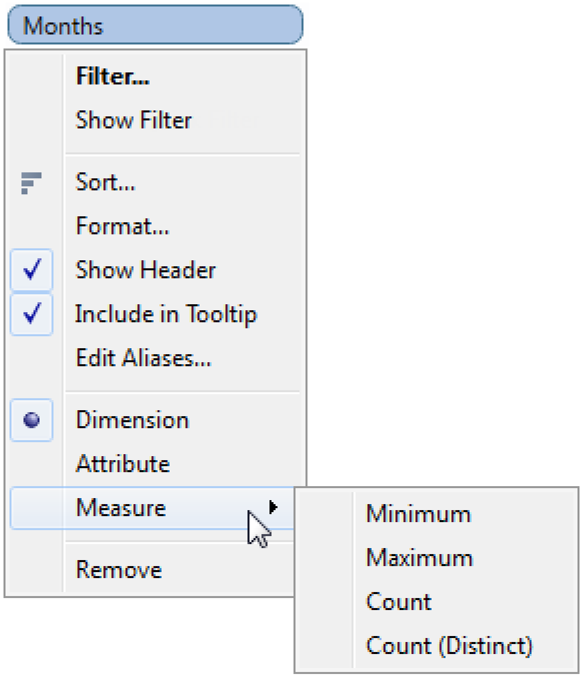
> You can aggregate measures using Tableau only for relational data sources. Multidimensional data sources contain data that is already aggregated. In Tableau, multidimensional data sources are supported only in Windows.

You can change the aggregation for a measure in the view from its context menu:



Aggregating Dimensions

You can aggregate a dimension in the view as **Minimum**, **Maximum**, **Count**, or **Count (Distinct)**. When you aggregate a dimension, you create a new temporary measure column, so the dimension actually takes on the characteristics of a measure.



> **Note**: The **Count (Distinct)** aggregation is not supported for Microsoft Access data sources, and for Microsoft Excel and Text File data sources using the legacy connection. If you are connected to one of these types of data sources, the **Count (Distinct)** aggregation is unavailable and shows the remark "Requires extract." If you save the data source as an extract, you will be able to use the **Count (Distinct)** aggregation.

Another way to view a dimension is to treat it as an Attribute. Do this by choosing **Attribute** from the context menu for the dimension. The **Attribute** aggregation has several uses:

- It can ensure a consistent level of detail when blending multiple data sources.
- It can provide a way to aggregate dimensions when computing table calculations, which require an aggregate expression.
- It can improve query performance because it is computed locally.

Tableau computes Attribute using the following formula:

```
IF MIN([dimension]) = MAX([dimension]) THEN MIN([dimension]) ELSE
"*" END
```

The formula is computed in Tableau after the data is retrieved from the initial query. The asterisk (*) is actually a visual indicator of a special type of Null value that occurs when there are multiple values. See **Troubleshoot Data Blending** on page 930 to learn more about the asterisk.

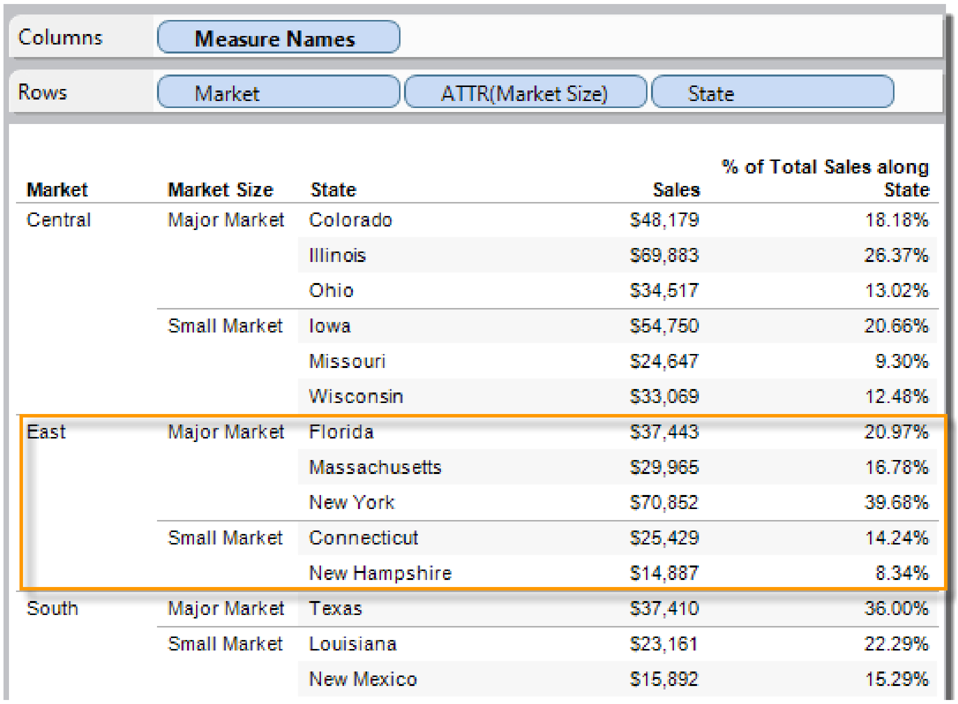
Below is an example of using Attribute in a table calculation. The table shows sales by market, market size, and state. Suppose you wanted to compute the percent of total sales each state contributed to the market. When you add a Percent of Total quick table calc (see **Quick Table Calculations** on page 2440) that computes along State, the calculation computes within the red area shown below. This is because the Market Size dimension is partitioning the data.

Columns: Measure Names

Rows: Market | Market Size | State

Market	Market Size	State	Sales	% of Total Sales along State
Central	Major Market	Colorado	$48,179	31.58%
		Illinois	$69,883	45.80%
		Ohio	$34,517	22.62%
	Small Market	Iowa	$54,750	48.68%
		Missouri	$24,647	21.92%
		Wisconsin	$33,069	29.40%
East	Major Market	Florida	$37,443	27.08%
		Massachusetts	$29,965	21.67%
		New York	$70,852	51.25%
	Small Market	Connecticut	$25,429	63.07%
		New Hampshire	$14,887	36.93%
South	Major Market	Texas	$37,410	100.00%
	Small Market	Louisiana	$23,161	34.82%
		New Mexico	$15,892	23.89%

When you aggregate Market Size as an Attribute, the calculation is computed within the Market (East, in the following image), and the Market Size information is used purely as a label in the display.



Market	Market Size	State	Sales	% of Total Sales along State
Central	Major Market	Colorado	$48,179	18.18%
		Illinois	$69,883	26.37%
		Ohio	$34,517	13.02%
	Small Market	Iowa	$54,750	20.66%
		Missouri	$24,647	9.30%
		Wisconsin	$33,069	12.48%
East	Major Market	Florida	$37,443	20.97%
		Massachusetts	$29,965	16.78%
		New York	$70,852	39.68%
	Small Market	Connecticut	$25,429	14.24%
		New Hampshire	$14,887	8.34%
South	Major Market	Texas	$37,410	36.00%
	Small Market	Louisiana	$23,161	22.29%
		New Mexico	$15,892	15.29%

List of Predefined Aggregations in Tableau

Sometimes it is useful to look at numerical data in an aggregated form such as a summation or an average. The mathematical functions that produce aggregated data are called aggregation functions. Aggregation functions perform a calculation on a set of values and return a single value. For example, a measure that contains the values 1, 2, 3, 3, 4 aggregated as a sum returns a single value: 13. Or if you have 3,000 sales transactions from 50 products in your data source, you might want to view the sum of sales for each product, so that you can decide which products have the highest revenue.

You can use Tableau to set an aggregation only for measures in relational data sources. Multidimensional data sources contain aggregated data only.

> **Note**: Using floating-point values in combination with aggregations can sometimes lead to unexpected results. For details, see **Understanding data types in calculations** on page 2146.

Tableau provides a set of predefined aggregations that are shown in the table below. You can set the default aggregation for any measure that is not a calculated field that itself contains an aggregation, such as `AVG([Discount])`. See **Set the Default Aggregation for a Measure** on page 348. You can also set the aggregation for a field already in the view. For details, see **Change the Aggregation of a Measure in the View** on page 341.

Aggregation	Description	Result for measure that contains 1, 2, 2, 3
Attribute	Returns the value of the given expression if it only has a single value for all rows in the group, otherwise it displays an asterisk (*) character. Null values are ignored. This aggregation is particularly useful when aggregating a dimension. To set a measure in the view to this aggregation, right-click (control-click on Mac) the measure and choose **Attribute**. The field then changes to show the text ATTR: 	N/A
Dimension	Returns all unique values in a measure or dimension.	3 values (1, 2, 3)
Sum	Returns the sum of the numbers in a measure. Null values are ignored.	1 value (8)
Average	Returns the arithmetic mean of the numbers in a measure. Null values are ignored.	1 value (4)
Count (Distinct)	Returns the number of unique values in a measure or dimension. When applied to a dimension, Tableau creates a new temporary column that is a measure because the result of a count is a number. You can count numbers, dates, booleans, and strings. Null values are ignored in all cases. This aggregation is not available for the following types of workbooks: • Workbooks created before Tableau Desktop 8.2 and that use Microsoft Excel or Text File data sources. • Workbooks that use legacy connections.	1 value (3)

Aggregation	Description	Result for measure that contains 1, 2, 2, 3
	• Workbooks that use Microsoft Access data sources. If you are connected to a workbook that uses of one of these types, Count (Distinct) is unavailable and Tableau shows the message "Requires extract." To use this aggregation, extract your data. See **Extract Your Data** on page 1014.	
Minimum	Returns the smallest number in a measure or continuous dimension. Null values are ignored.	1 value (1)
Maximum	Returns the largest number in a measure or in the given expression based on a sample population. Null values are ignored. Returns a Null if there are fewer than 2 members in the sample that are not Null. Use this function if your data represents a sample of the population.	1 value (3)
Std. Dev (Pop.)	Returns the standard deviation of all values in the given expression based on a biased population. Assumes that its arguments consist of the entire population. Use this function for large sample sizes.	1 value (0.7071)
Variance	Returns the variance of all values in the given expression based on a sample. Null values are ignored. Returns a Null if there are fewer than 2 members in the sample that are not Null. Use this function if your data represents a sample of the population.	1 value (0.6667)
Variance (Pop.)	Returns the variance of all values in the given expression based on a biased population. Assumes that its arguments consist of the entire population. Use this function for large sample sizes.	1 value (0.5000)

Aggregation	Description	Result for measure that contains 1, 2, 2, 3
Disaggregate	Returns all records in the underlying data source. To disaggregate all measures in the view, select **Aggregate Measures** from the Analysis menu (to clear the check mark). Tableau allows you to view data in disaggregated form (relational databases only). When data are disaggregated, you can view all of the individual rows of your data source. For example, after discovering that the sum of sales for rubber bands is $14,600, you might want to see the distribution of individual sales transactions. To answer this question, you need to create a view that shows individual rows of data. That is, you need to disaggregate the data (see **How to Disaggregate Data** on page 350). Another way to look at disaggregated data is to view the underlying data for all or part of a view. For more details, see **View Underlying Data** on page 2728.	4 values (1, 2, 2, 3)

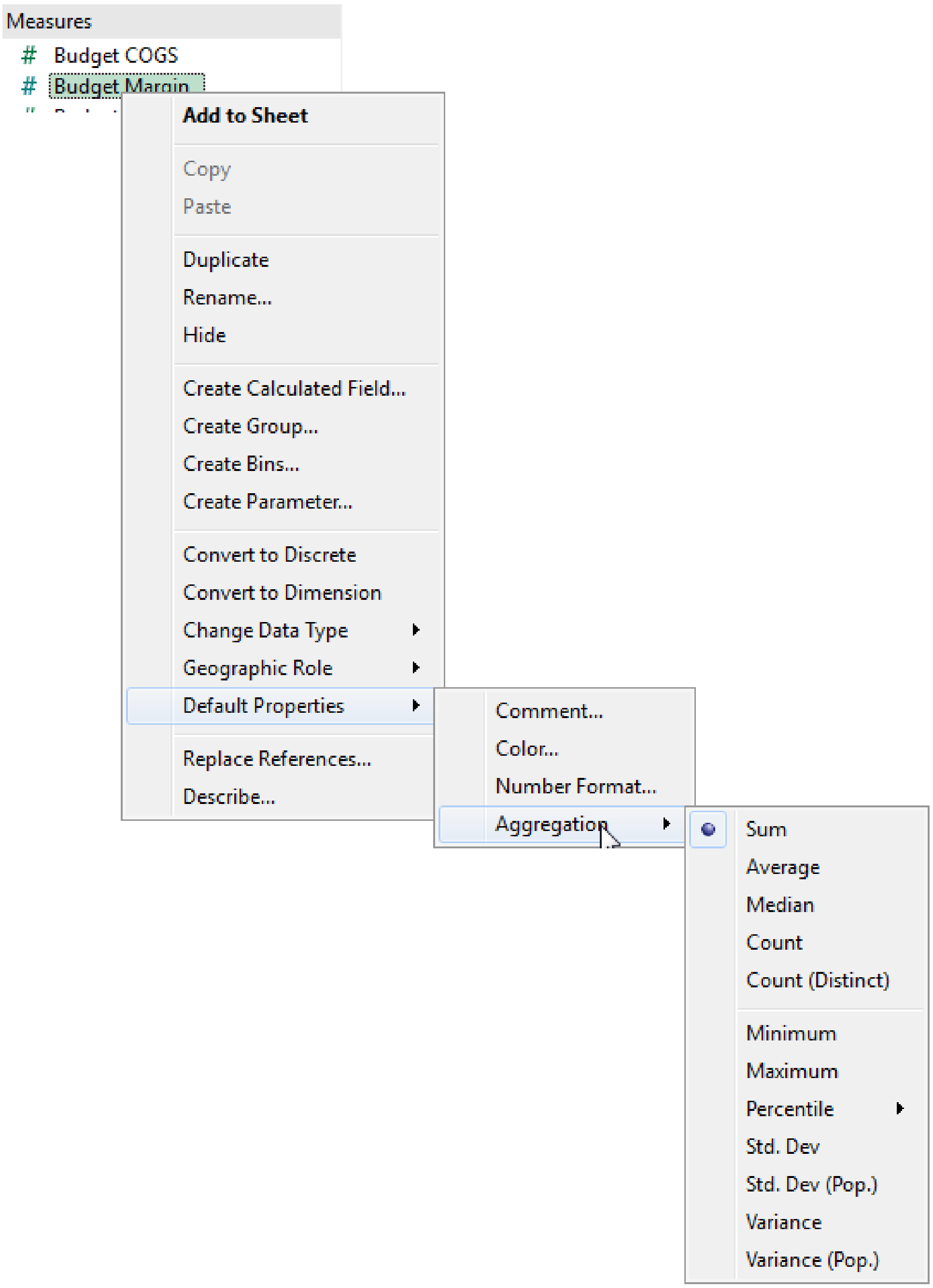
You can also define custom aggregations as described in **Aggregate Functions in Tableau** on page 2192. Depending on the type of data view you create, Tableau will apply these aggregations at the appropriate level of detail. For example, Tableau will apply the aggregation to individual dimension members (the average delivery time in the East region), all members in a given dimension (the average delivery time in the East, West, and Central regions), or groups of dimensions (the sum of sales for all regions and for all markets).

Set the Default Aggregation for a Measure

You can set the default aggregation for any measure that is not a calculated field that itself contains an aggregation, such as `AVG([Discount])`. A default aggregation is a preferred calculation for summarizing a continuous or discrete field. The default aggregation is automatically used when you drag a measure to a view.

To change the default aggregation:

Right-click (control-click on Mac) a measure in the Data pane and select **Default Properties** > **Aggregation**, and then select one of the aggregation options.



Note: You can use Tableau to aggregate measures only with relational data sources. Multidimensional data sources contain aggregated data only.

You cannot set default aggregations for published data sources. The default aggregation is set when the data source is initially published. Create a Local Copy of the published data source to adjust the default aggregation.

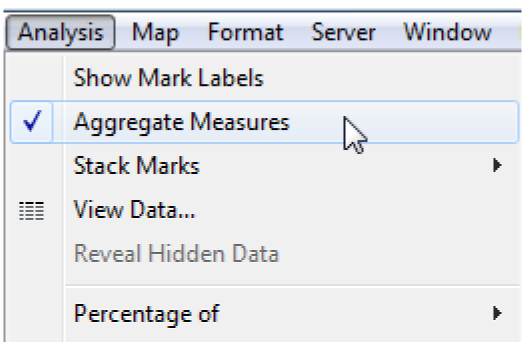
How to Disaggregate Data

Whenever you add a measure to your view, an aggregation is applied to that measure by default. This default is controlled by the **Aggregate Measures** setting in the **Analysis** menu.

If you decide you want to see all of the marks in the view at the most detailed level of granularity, you can disaggregate the view. Disaggregating your data means that Tableau will display a separate mark for every data value in every row of your data source.

To disaggregate all measures in the view:

- Clear the **Analysis >Aggregate Measures** option. If it is already selected, click **Aggregate Measures** once to deselect it.



When **Aggregate Measures** is selected, Tableau will attempt to aggregate measures in the view by default. This means that it collects individual row values from your data source into a single value (which becomes a single mark) adjusted to the level of detail in your view.

The different aggregations available for a measure determine how the individual values are collected: they can be added (SUM), averaged (AVG), or set to the maximum (MAX) or minimum (MIN) value from the individual row values.

For a complete list of the available aggregations, **List of Predefined Aggregations in Tableau** on page 345.

The level of detail is determined by the dimensions in your view—for information about the concept of level of detail, see **How dimensions affect the level of detail in the view** on page 319.

Disaggregating your data can be useful for analyzing measures that you may want to use both independently and dependently in the view. For example, you may be analyzing the results from a product satisfaction survey with the Age of participants along one axis. You can aggregate the **Age** field to determine the average age of participants or disaggregate the data to determine at what age participants were most satisfied with the product.

Disaggregating data can be useful when you are viewing data as a scatter plot. See **Example: Scatter Plots, Aggregation, and Granularity** below.

> **Note:** If your data source is very large, disaggregating the data can result in a significant performance degradation.

Example: Scatter Plots, Aggregation, and Granularity

If you place one measure on the **Rows** shelf and another measure on the **Columns** shelf, you are asking Tableau to compare two numerical values. Typically, Tableau chooses a scatter plot as the default visualization in such cases. The initial view will most likely be single mark, showing the sum for all values for the two measures. This is because you need to increase the level of detail in the view.

Start building the scatter plot

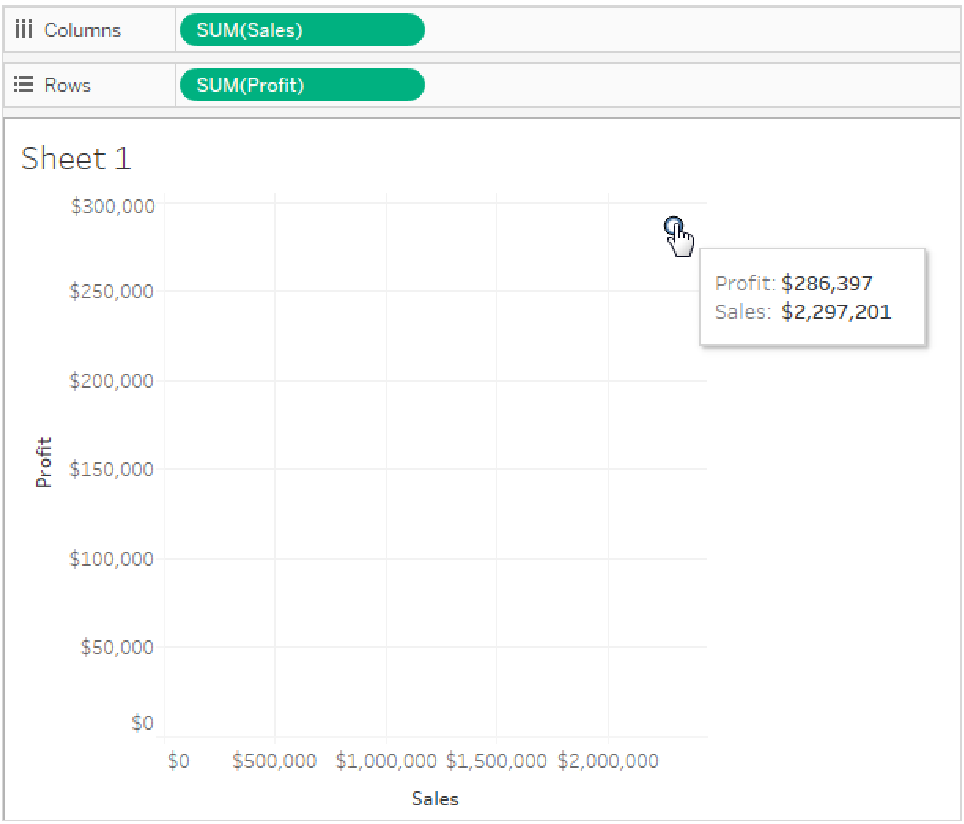
There are various ways to add detail to a basic scatter plot: **you can use dimensions to add detail**, you can add additional measures and/or dimensions to the Rows and Columns shelves to create multiple one-mark scatter plots in the view, or you can **disaggregate the data**. And, you can also use any combination of these options. This topic looks at these alternatives using the **Sample-Superstore** data source.

To create the initial view, follow these steps:

1. Place the **Sales** measure on the **Columns** shelf.

2. Place the **Profit** measure on the **Rows** shelf.

The measures are automatically aggregated as sums. The default aggregation (SUM) is indicated in the field names. The values shown in the tooltip show the sum of sales and profit values across every row in the data source.



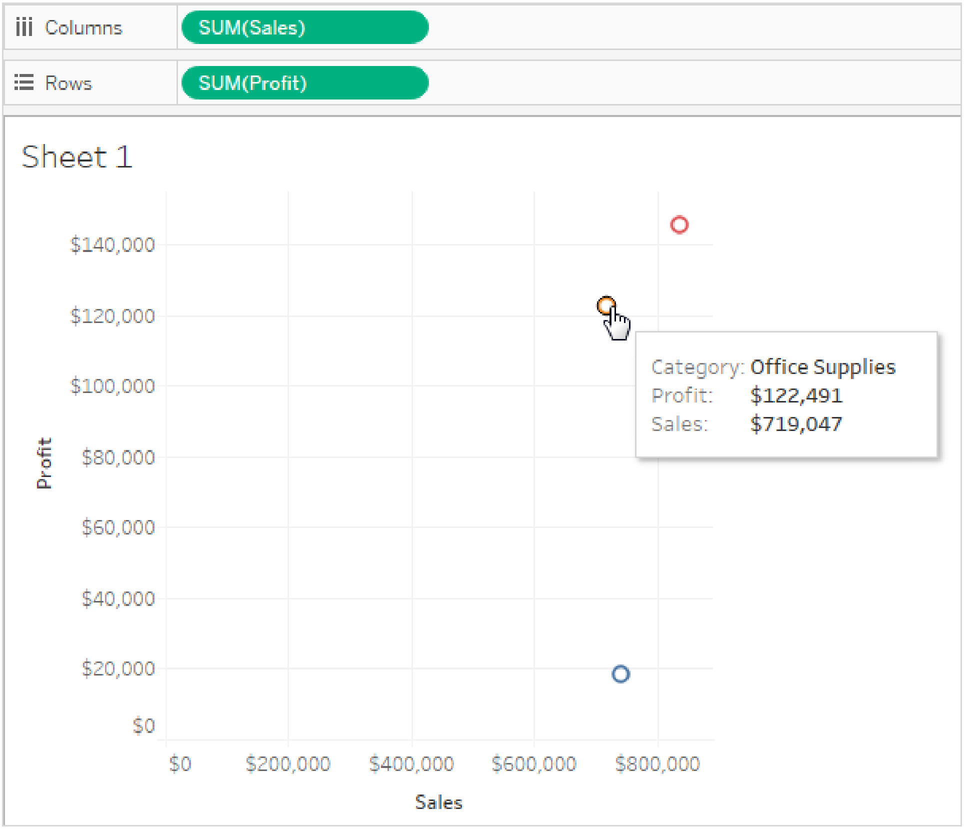
Follow the steps below to use dimensions to add detail to the view and to disaggregate data.

Use dimensions to add detail

Follow these steps to develop the scatter plot view you created above by adding dimensions to show additional levels of detail.

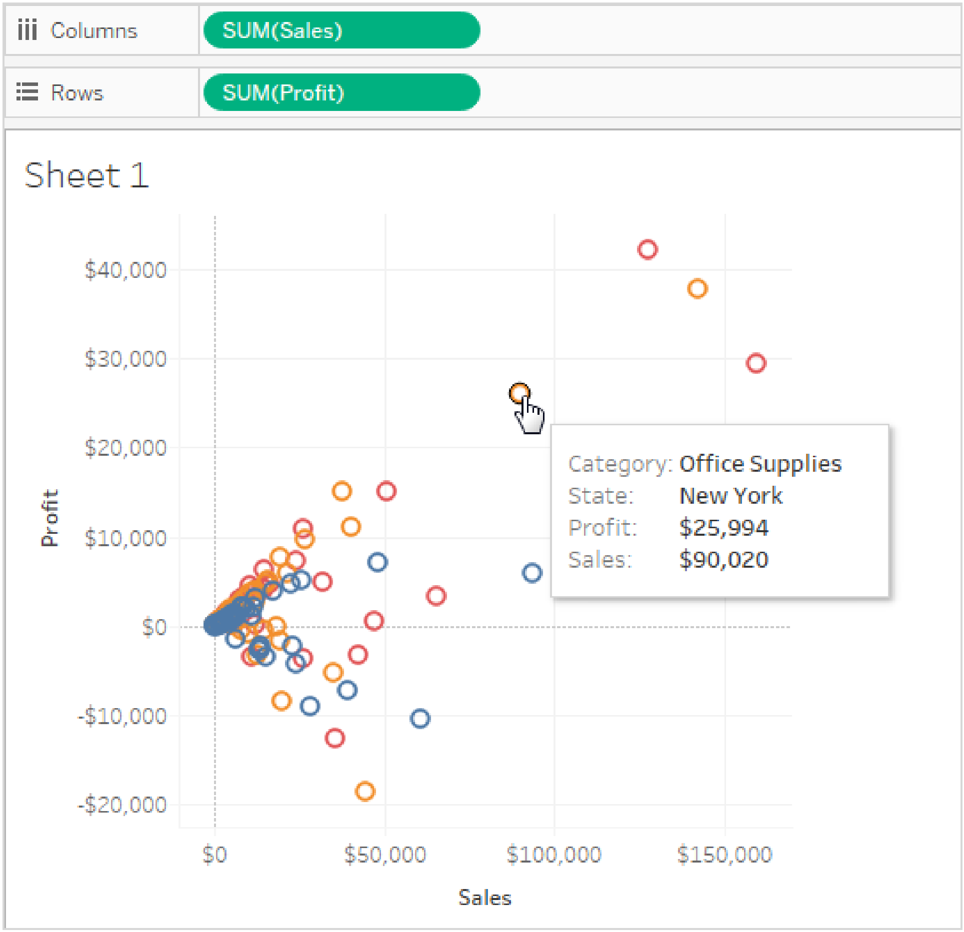
1. Drag the **Category** dimension to **Color** on the Marks card.

   This separates the data into three marks—one for each dimension member—and encodes the marks using color.



2. Drag the **State** dimension to **Detail** on the Marks card.

   Now there are many more marks in the view. The number of marks is equal to the number of distinct states in the data source multiplied by the number of categories.



Although more marks are now displayed, the measures are still aggregated. So regardless of whether there is one row in the data source where State = North Dakota and Category= Furniture, or 100 such rows, the result is always a single mark.

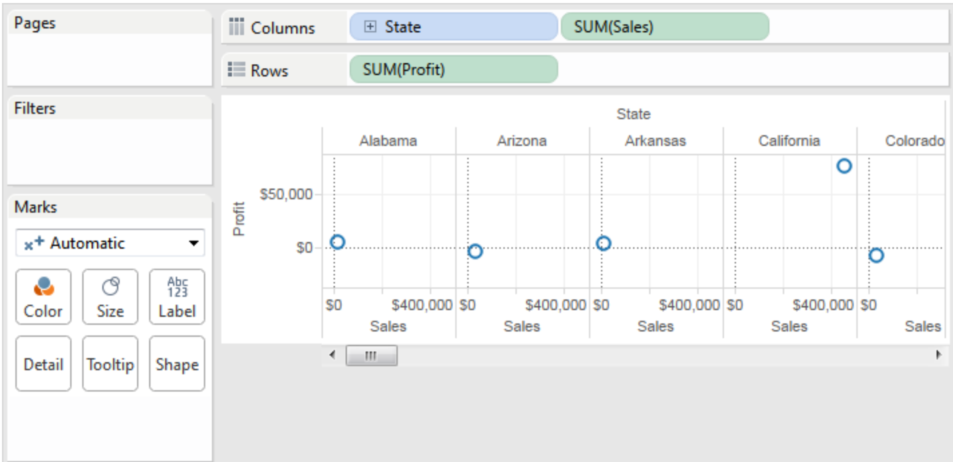
Maybe this process is developing the view in a direction you find useful, or maybe you prefer to go in a different direction—for example, by adding a time dimension to the view, or by introducing trend lines or forecasting. You decide what questions to ask.

Try adding more fields to the rows and columns shelves

Revert to the original one-mark view and follow these steps to develop the scatter plot view by adding fields to the **Rows** and **Columns** shelves.

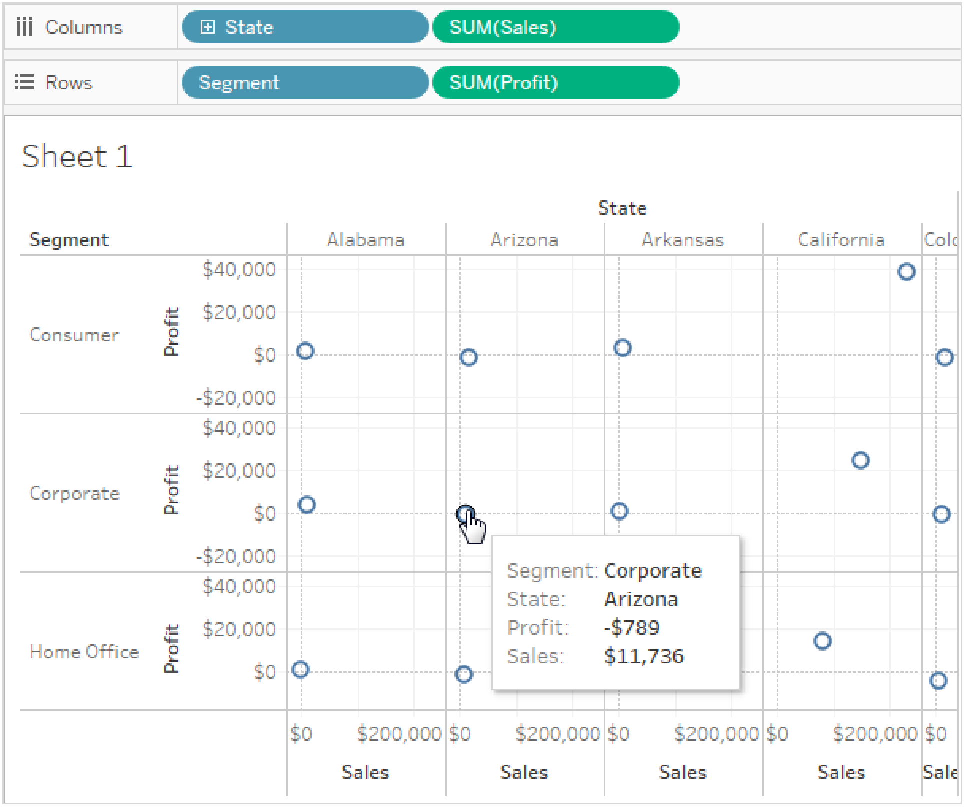
1. Drag the **State** dimension to the **Columns** shelf.

Even if you drop **Continent** to the right of **SUM(Sales)**, Tableau moves it to the left of **SUM(Sales)**. This is because you cannot insert a dimension within a continuous axis. Instead, your view shows a separate axis for each member of the dimension.



2. Drag the **Segment** dimension to the **Rows** shelf.

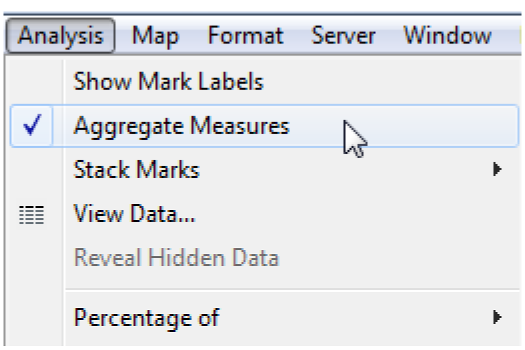
   You now have a view that provides an overview of Sales and Profit across states and customer segments. It can be interesting to hover over the marks in the view to see tooltip data for various segments:



Try disaggregating the data

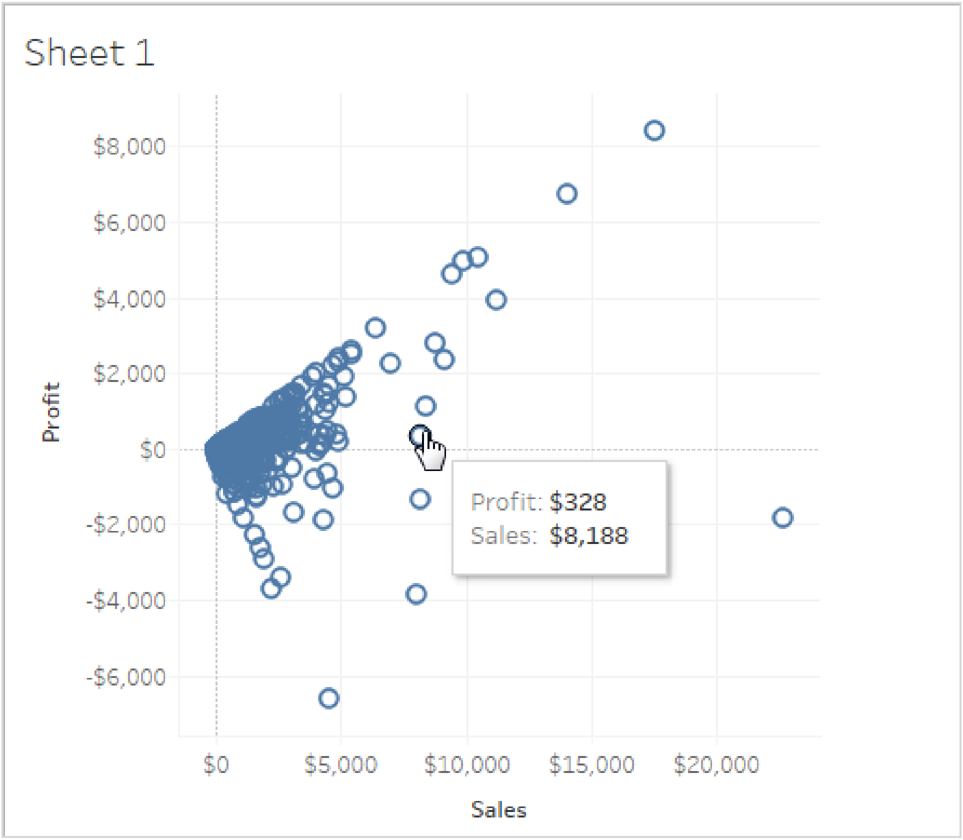
Another way to modify your original one-mark scatter plot to display more marks is by disaggregating the data.

Clear the **Analysis >Aggregate Measures** option. If it is already selected, click **Aggregate Measures** once to deselect it.



What you have actual done is to dis-aggregate the data, because this command is a toggle that was originally selected (check mark present). Tableau aggregates data in your view by default.

Now you see a lot of marks—one for each row in your original data source:



When you disaggregate measures, you no longer are looking at the average or sum for the values in the rows in the data source. Instead, the view shows a mark for every row in the data source. Disaggregating data is a way to look at the entire surface area of the data. It's a quick way to understand the shape of your data and to identify outliers. In this case, the disaggregated data shows that for many rows in the data, there is a consistent relationship between sales income and profit—this is indicated by the line of marks aligned at a forty-five degree angle.

The Bookshop data set

The Bookshop data set was created for Tableau Desktop 2020.2 to showcase relationships—the new data modeling capabilities for combining data.

Download the files

You can choose to download the raw data and start from scratch building your own data model, or download one of the pre-built data sources to jump right into analysis with relationships.

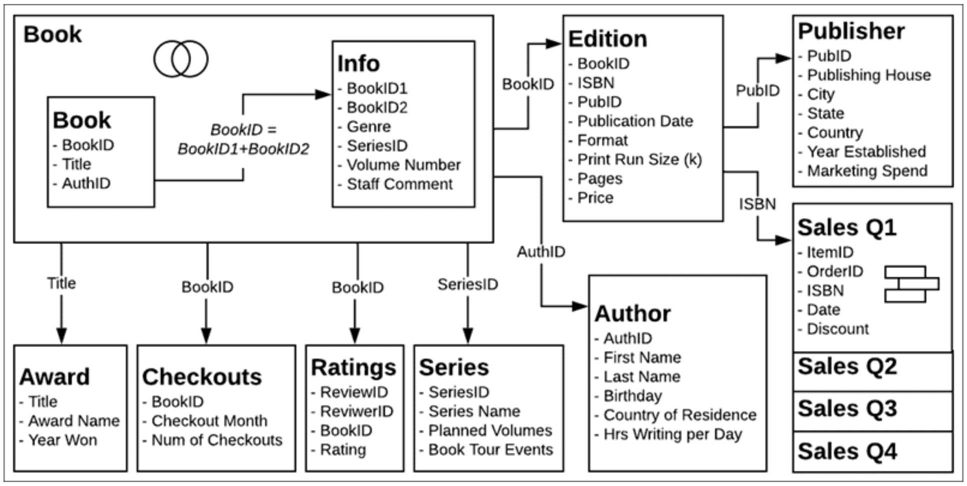
- Raw data (xlsx)
  - Bookshop.xlsx - the raw data as an .xlsx file
  - BookshopLibraries.xlsx - additional library tables that introduce many-to-many relationships (contains only the additional library tables)
- Packaged data source (tdsx) files
  - Bookshop.tdsx - a packaged .tdsx file with the related data source already built and some metadata customizations added
  - MinimalBookshop.tdsx - the same tables as Bookshop.tdsx but without any metadata or cleanup
  - Bookshop_libraries.tdsx - a packaged .tdsx file that adds the library tables to Bookshop.tdsx (includes all tables).

To connect to a downloaded .tdsx

1. Open Tableau Desktop.
2. In the Connect pane on the left, choose the **More...** option under the **To a File** heading.
3. Navigate to where you downloaded the .tdsx and double click it (or select it then click Open).

About the data set

Bookshop consists of 13 tables, combined in the following manner:



Note: A central concept for this data set is the idea of a *book* versus an *edition*. A book is a concept with attributes such as author, title, and genre. An edition is a physical version of the book, with attributes such as format (hardcover, paperback), publication date, and page count.

Data Dictionary

Some fields may need a little explanation.

- The **Rating** field in the Ratings table is on a scale of 1-5 with 5 being high.
- The **Format** field is a detailed breakdown of the format. Everything other than “Hardcover” could be considered “Paperback”.
- **ISBN** stands for International Standard Book Number and is a unique 13 digit identifier given to each edition of a book. The ISBN is represented in the barcode and is tied to the price.
- The **ItemID** and **OrderID** fields are hierarchical. An order may contain multiple items.
- The **Staff Comment** field contains summaries and reviews for some of the books.

If you build your own data source

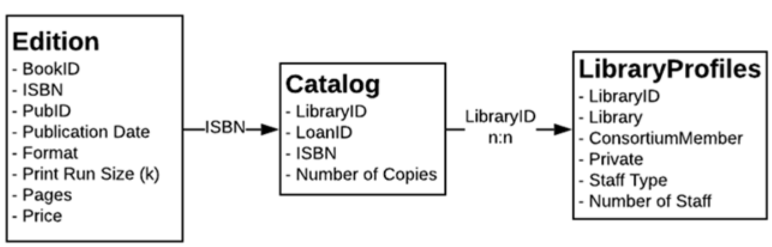
1. The **Publisher** and **Sales** tables must be related to the **Edition** table.
2. The **Book** and **Info** tables can be related or joined, but it must be on the calculation `BookID = [BookID1]+[BookID2]`.
   - An inner join is suggested.
   - The **Series** table can only be used once the Info table is part of the data source.

optional:

3. We recommend renaming the logical table of Book joined with Info “Book” or something similar.
4. The four Sales tables can be analyzed independently, but if they are unioned, we recommend renaming the union “Sales” or something similar.
5. Most of the fields used to form relationships aren’t needed for analysis and can be hidden once the data model has been built
   - Any field ending with ID can be hidden (the only “ID” fields that do are ReviewID and ReviewerID from Ratings and ItemID and OrderID from Sales).
   - ISBN should be kept, ideally from the Edition table, as the identifier of each edition. The ISBN field in the unioned Sales table can be hidden.
   - The Title field in the Award table can be hidden.
   - The sheet and table fields from the Union also don’t contain unique information and can be hidden.
6. To simplify the data model, the core tables are **Book**, **Author**, and **Edition**. The easiest tables to leave out with minimal impact would be **Checkouts**and **Ratings**, followed by **Award**, **Publisher**, **Sales**, or **Info** and **Series**.

Bookshop Libraries

The BookshopLibraries.xlsx file introduces new tables around libraries for the purpose of supporting many-to-many relationships. The Catalog table relates to the Editions table on ISBN. The Library Profiles table relates to the Catalog table as a many-to-many (n:n) relationship on LibraryID.



Data Dictionary

Library Profile

- The **Library** field is the name of the library.
- The **Library ID** field an ID also used in the Catalog table.

- The **Consortium Member** field is a yes/no field that indicates if the library is part of the larger network of libraries that offer inter-library loans and share other services
- The **Private** field is a yes/no field that indicates if this is a members-only library or a public institution
- **Staff Type** and **Number of Staff** together describe how many Librarians, Library Assistants, and Library Technicians work for each library.

Catalog

- **LoanID** is a unique identifier for the combination of ISBN and library that is used to track the **Number of Copies** a library has. For example, if the Idle Hour Library has two paperback copies and one hardcover copy of a title, there will be two Loan IDs.
- The **Library ID** field is an ID also used in the Library Profile table.

Structure of the Library Profile table

The Library Profile table was originally formatted as an unpivoted table with a column for each staff type, below.

Library	LibraryID	Consortium Member	Private	Librarians	Library Technicians	Library Assistants
Idle Hour Library	L-IHL	Yes	No	53	61	16
The Bibliophile’s Shelves	L-BS	Yes	Yes	4	3	0
Armchair Athanaeum	L-AA	No	Yes	6	0	0
Old Friend Library	L-OFL	Yes	No	3	5	17
Bide Awhile	L-BA	Yes	No	9	20	6
IndieUnBound	L-IUB	Yes	No	7	2	47
Page Station Book Exchange	L-PS	No	Yes	3	1	4

The table was pivoted across the Librarians, Library Technicians, and Library Assistants columns. The final format has a column for Staff Type and a column for Number of Staff. However, this means that there is more than one row for each library. The Library ID values are therefore not unique, and the pivoted version of the table must be related many-to-many to the Catalog table.

While this has the benefit of allowing a many-to-one relationship between Catalog and Profile, it isn't an ideal structure for data analysis.

Explore

This data set, while fictional, supports various analytical scenarios and plenty of exploration. Some suggestions include:

- What books are the most popular? The least popular? Is that based on sales, reviews, checkouts, or another metric?
- Who was the youngest debut author? Who was the oldest?
- Do some publishing houses seem to specialize in any way?
- What was the longest time between editions of the same book?
- Are there any seasonal trends for sales? What about checkouts? Do any titles or genres have seasonal fluctuations?
- Are there any correlations between checkouts, print run size, book review ratings, and sales volume?
- Do the authors who spend the most time writing have the most successful books? Do they have the highest page count?
- When are most books published? Are there any anomalies?
- Are there any trends for genre, format, and price?
- What sort of distributions do the ratings have? Do those distributions vary by book? By genre? Do they seem to align with awards?
- How would you calculate the sales price, given that there is sometimes—but not always—a discount given at the time of sale?
- Do sales approximate the Pareto principle?
- Are there any patterns in the discounts?
- Do any tables in particular appear to have dirty data?

Bookshop Data Set Easter Eggs

- *Etaoin Shrdlu* is a reference to linotype hot type machines. *Palimpsest* is a manuscript-making reference.
- Author names were derived from a list of famous American authors with their first and last names shuffled.
- Publication dates today are usually on Tuesdays. This was reflected in the data, which assumes this industry standard holds into the year 2178.
- One book has no reviews, checkouts, or sales.
- The Checkouts data is based actual library data, with BookIDs mapped to titles, so the checkout trends are real-world.
- The Ratings data is based on actual book rating data, with BookIDs mapped to titles, so the distribution of ratings for a given book is real.
- The proportion of sales that are for a single book versus multiple books is based on real-world data from an independent bookstore.
- The Sales data is completely generated, but is based on actual seasonal and weekday trends for a resort town with a tourism-based economy (proportionally by month and day of the week, and for spring break and the winter holidays).
  - Sales spike over long weekends and spring break.
  - Sales are much higher on weekends and at their lowest on Tuesdays and Thursdays.
  - Peak season is the summer between the 4th of July and Labor Day.
  - Sales spike immediately after Thanksgiving and in the weeks leading up to Christmas.
- The ISBNs in the data set are fictitious but constructed following ISBN-13 principles:
  - EAN—989 (not actually in use)
  - Group number—28
  - Publisher registrant element—varying length (2-4 digits)
  - Title element, varying length (3-5 digits, depending on the registrant element length)
  - Check digit, created with the ISBN-13 check digit modular arithmetic calculation
- One of the developers working on this feature suggested a book with no editions, and the title *The Deep Grey*, so its BookID ends with the last three digits of his work phone number.
- There is an (intentional) error in the Edition table where two editions of BookID PA169 were labeled as Palimpsest Printing, but the ISBN indicates they were actually printed by Etaoin Shrdlu Press.

The person who created this data set had way too much fun making it, and clearly has a deep obsession with books. She hopes you enjoy the data set and it helps you appreciate the power, elegance, and beauty of relationships in Tableau.

Sign in to Tableau Server or Tableau Online

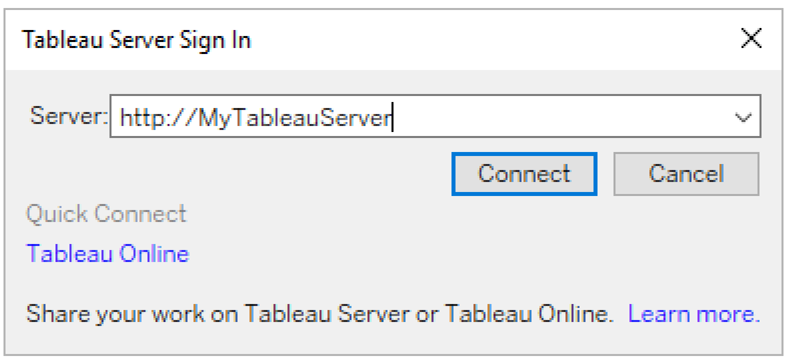
To access content on Tableau Server or Tableau Online, sign in via Tableau Desktop or a web browser.

Note: If you sign in via a web browser, it must be configured to allow first-party cookies.

Sign in from Tableau Desktop

Tableau Server

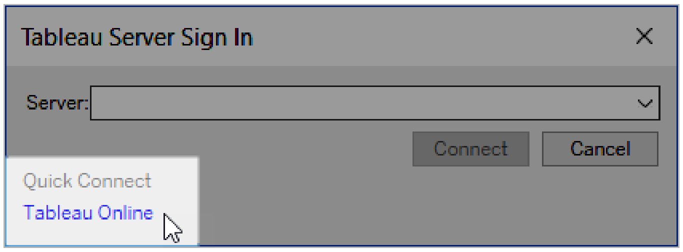
1. In Tableau Desktop, select **Server** > **Sign In**. Enter the name or address of your Tableau server, and then click **Connect**.



2. Enter your user name and password, and then click **Sign In**. If you have access to multiple sites, select the one you want to use. You can also enter the name of the site and search for it.

Tableau Online

1. In Tableau Desktop, select **Server** > **Sign In**. Click **Tableau Online** in the bottom left corner.



2. If your site is set up to use Tableau authentication, enter your user name (email address) and password you use for Tableau Online, and then click **Sign In**.

   If multi-factor authentication (MFA) is enabled with Tableau authentication, you are prompted to verify your identity using the verification you selected during the MFA registration process. After verifying your identity, you are redirected to your site.

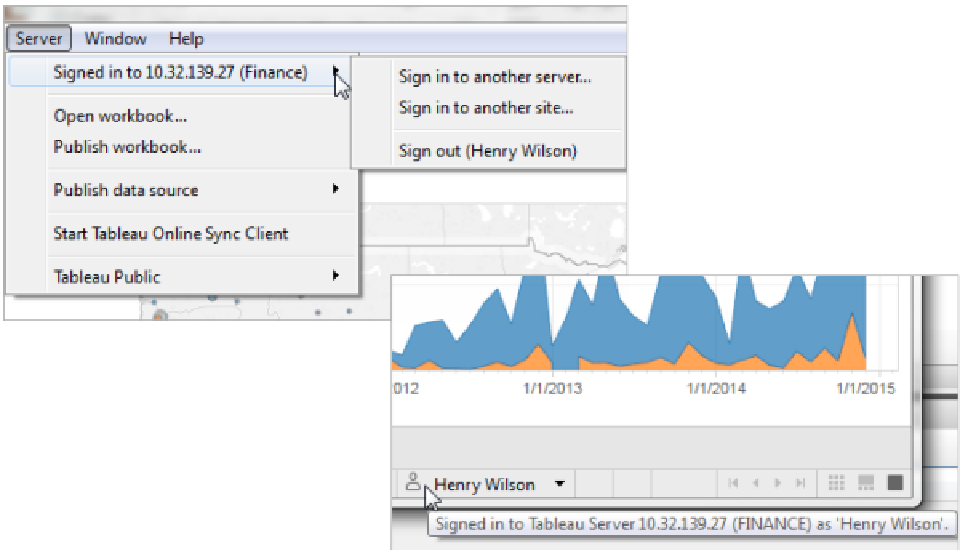
   Notes:
   - If you are signing in to Tableau Online or registering for MFA for the first time, see Register for multi-factor authentication.
   - You must be running Tableau Desktop 2021.1 and later to use MFA with Tableau authentication.
3. If you have access to multiple sites, select the one you want to use.

Automatically keep Tableau Desktop connected to Tableau Server or Online

When you connect to Tableau Server or Tableau Online, Tableau Desktop keeps you signed in from session to session, if you don't sign out. Next time you launch Tableau Desktop, you are automatically signed in to your most recent server connection. You can easily switch to other available server and site connections.

You can see the server and site you are signed into, as well as who you are signed in as, in two places:

- The **Server** menu shows the server and site that you are signed in to. **Sign out** shows you are signed in as.

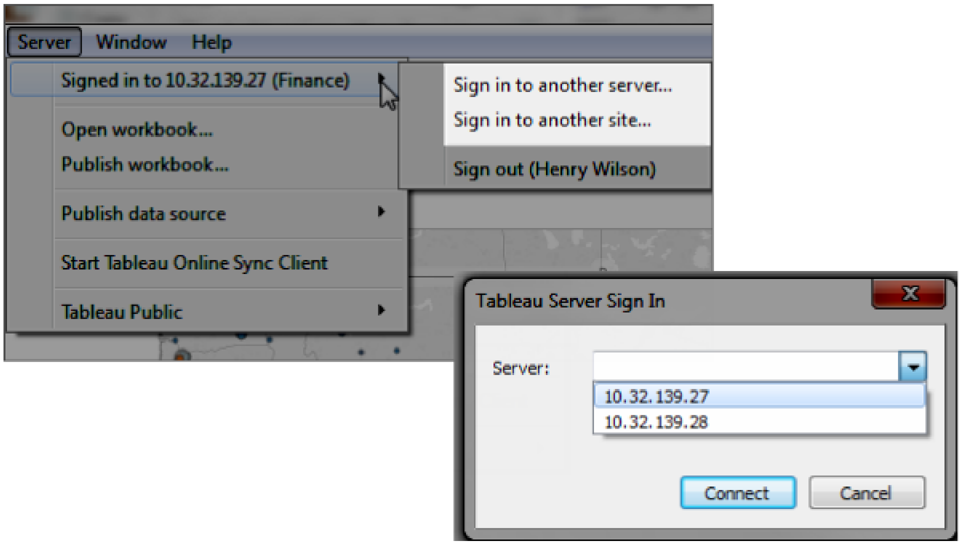


- The status bar also provides this information. Hover over the username to see the current server and site.

Switch servers, switch sites

To switch servers, on the **Server** menu, click **Sign in to another server**, and then enter the server name and your sign-in credentials.

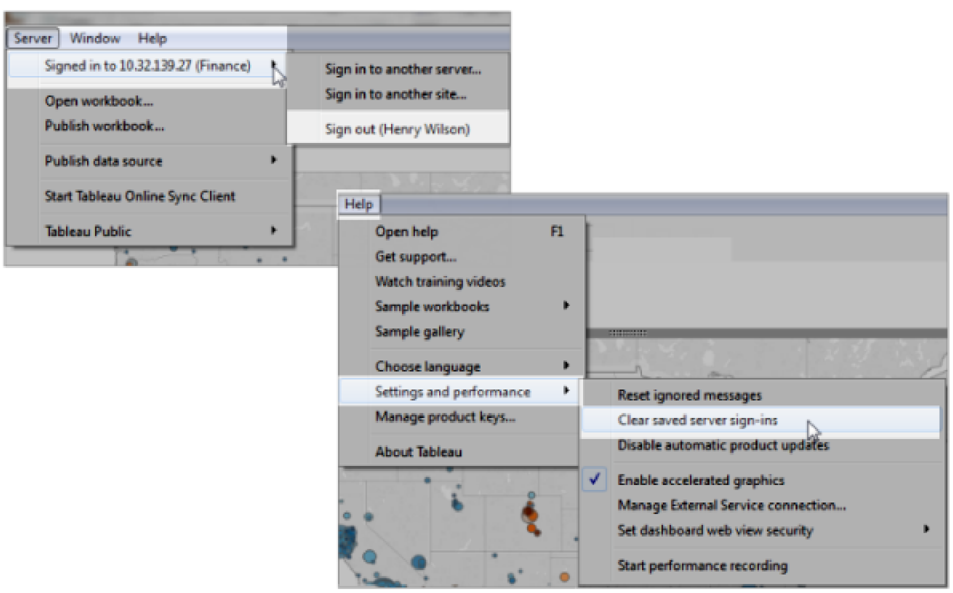
If you have access to multiple sites, on the **Server** menu, click **Sign in to another site**.



Sign out or clear all saved connections

When you sign in to a server and site, Tableau Desktop stores your credentials in a secure token that remembers your connection. After this token is in place, you can access the server directly, without having to sign in.

Click **Server > Sign out** to clear your sign-in credentials for the current connection. To remove all of your existing server connections, click **Help > Settings and performance > Clear saved server sign-ins**.

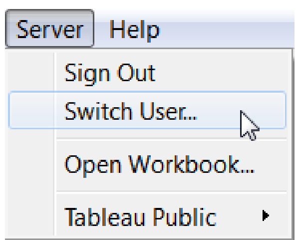


If you never want server sign-ins to be saved, a Tableau Server administrator can change the server settings to disallow connected clients. Be aware that this setting also affects other Tableau clients like Tableau Mobile. For more information, see Authentication for Connected Devices in the Tableau Server Help.

Switch user accounts from Tableau Desktop (Kerberos authentication only)

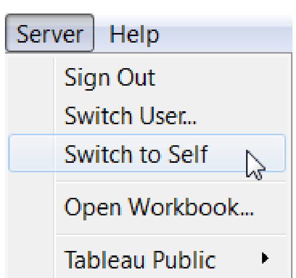
If your organization uses Tableau Desktop with Kerberos authentication and it doesn't succeed, you're prompted to provide a user name and password. To sign in as a different user for testing purposes, complete these steps:

1. On the Server menu, select **Switch User**.



2. In the Tableau Server Sign In dialog box, provide the new user name and password.

To later sign in using your normal credentials, select **Server** > **Switch to Self**.



Sign in with another user account in a browser

If you are signed in to the wrong account, you can switch to another one.

1. Click **Sign Out** on the user menu in the upper-right corner of the page.

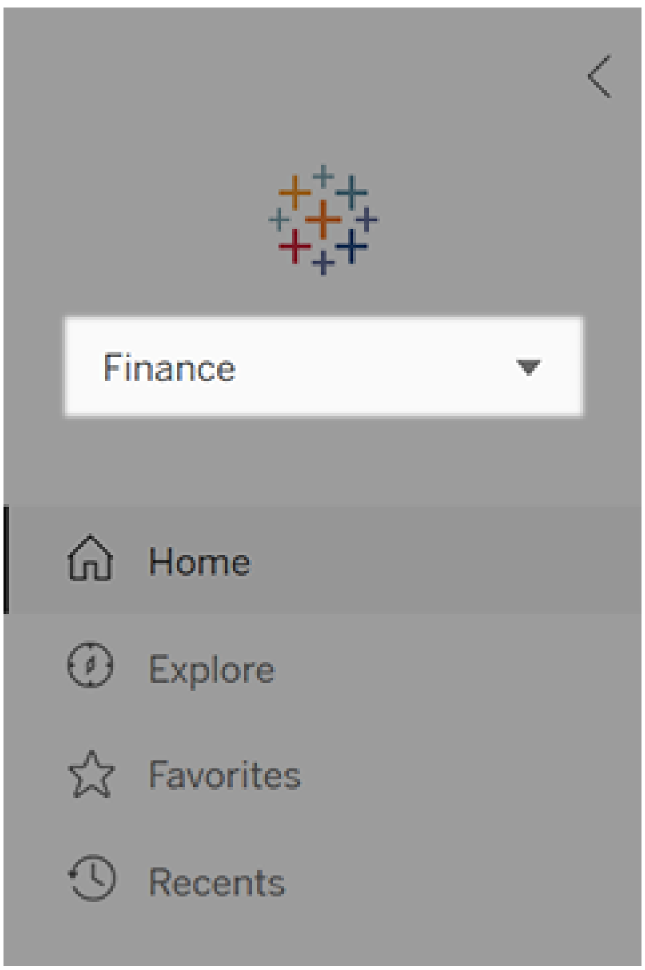
> **Note:** If you do not see a **Sign Out** command on the Tableau Server user menu, you must use the sign-out interface provided by your organization's external identity provider (IdP). For more information, contact your administrator.

2. Type your user name and password, and then click **Sign In**

Sign in to a specific site in a browser (Tableau Server only)

If you are a member of multiple sites, you're prompted to select one when you sign in to the server. Either click a visible site name or search for a name to select from a list of matching sites.

The name of the current site is displayed on the site picker.



Choose the Right Chart Type for Your Data

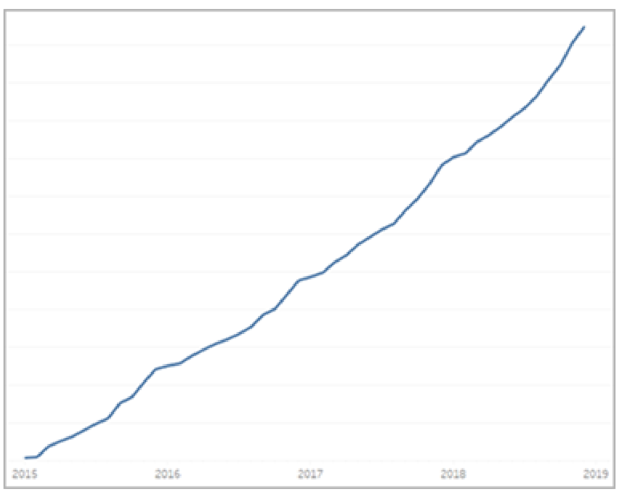
What chart or graph works best for your data? In Tableau, form follows function. The visualization (or viz) you create depends on:

- The questions you are trying to ask
- The properties of your data
- How you want to present and communicate your insights to others

For example, showing the growth in sales each year requires a different visualization than showing the connection between discounted items and their profitability. Knowing what you need to show will help determine how you want to show it.

This topic presents nine different types of information that you can display with a visualization. This isn't a comprehensive list, and there are bound to be exceptions to these categories. With experience you will be able to more quickly assess what chart type you want to create. Because Tableau is flexible, we encourage you to think outside the box. However, before you think outside the box, it's helpful to start with some common chart types.

Change over time



Showing a change over time for a measure is one of the fundamental categories of visualizations. There are many options for exploring change over time, including line charts, slope charts, and highlight tables.

To show change

Related topics

- **Building Line Charts** on page 1631
- **Dates and Times** on page 1245
- **Continuous Dates** on page 1283

over time, you need to know the value you expect to change, and how to work with Date fields in Tableau.

What kind of question does this chart answer?

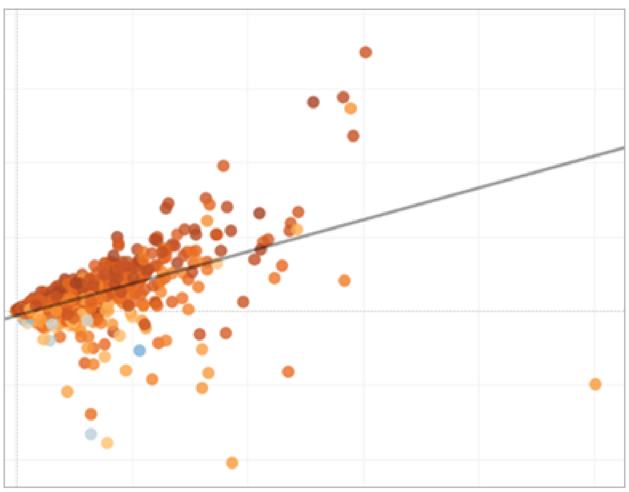
- How has this measure changed in the past year?
- When did this measure change?
- How quickly has this measure changed?

More inspiration

- Understand change over time with time-

series analysis
- New Ways to Visualize Time
- Visualizing Time: Beyond the Line Chart
- Examples of Change Over Time

Correlation



Sometimes you have two variables and are looking for the relationship between them. For example, you may be looking for the relationship between classroom size and school graduation rate, or how much lung capacity relates to endurance. (But remember, correlation does not always equal

Related topics

- **Build a Scatter Plot** on page 1651
- **Create Heatmaps that Show Trends or Density in Tableau** on page 1910
- **Add Trend Lines to a Visualization** on page 2528

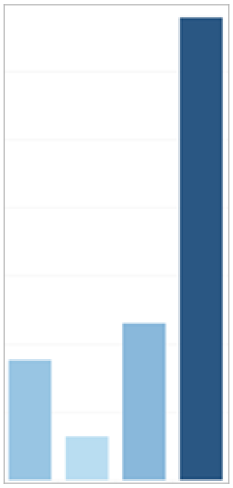
causation.)

Correlation can be shown with scatter plots or highlight tables, and you can use Tableau's analytics objects to show the strength of the correlation.

What types of question can this chart answer?

- Are these two measures related? How strongly?
- Are some measures more related than others?
- How strongly related are these measures?

Magnitude



Related topics

- **Build a Bar Chart** on page 1579
- **Build a Packed Bubble Chart** on page 1639
- **Building Line Charts** on page 1631

Magnitude shows the relative size or value of two or more discrete items. If you are comparing sales for different regions, you are looking at magnitude.

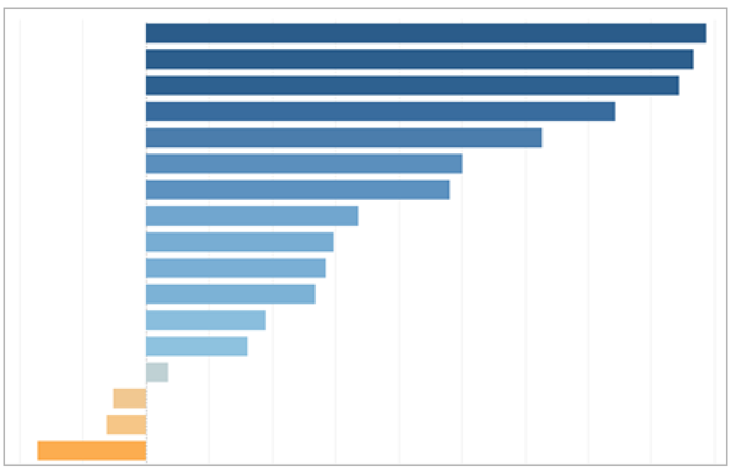
Magnitude charts include bar charts, packed bubble charts, and line charts.

What types of question can this chart answer?

- Which of these dimension members has the highest measure?
- Are there any exceptional dimensions?
- How large

of a gap is there between the lowest and highest measure between these dimensions?

Deviation



Deviation charts show how far a value varies from some baseline, such as the average or median. If you wanted to know which items had unusually high or low profit margins, you would use a deviation chart.

You can use bullet charts, bar charts, and combination charts to show deviation. You can also find the

Related topics

- **Build a Bar Chart** on page 1579
- **Build a Bullet Graph** on page 1597
- **Calculate Z-scores** on page 1682

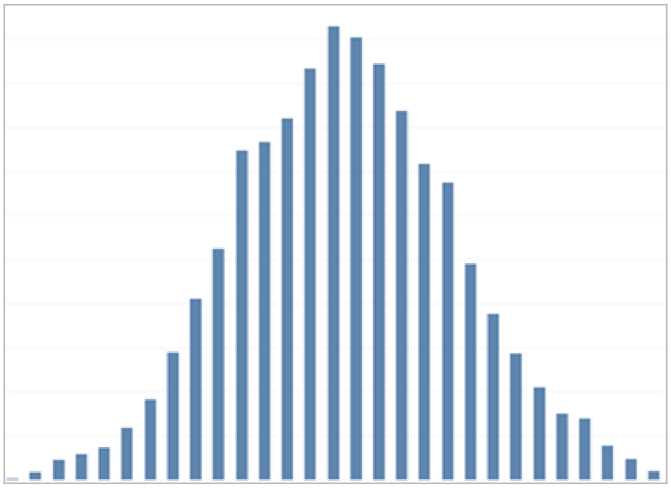
statistical significance of the deviation using a Z-score.

What types of question can this chart answer?

- How far from the norm does this measure stray?
- How important are the deviations in this measure?
- Is there a pattern to the deviations?

Distribution

When you are trying to find the frequency of events within a population, you are looking at the distribution. If you are showing the number of



respondents to a survey by age, or the frequency of incoming calls by day, a distribution chart might be the best choice.

Distribution charts include histograms, population pyramids, Pareto charts, and box plots.

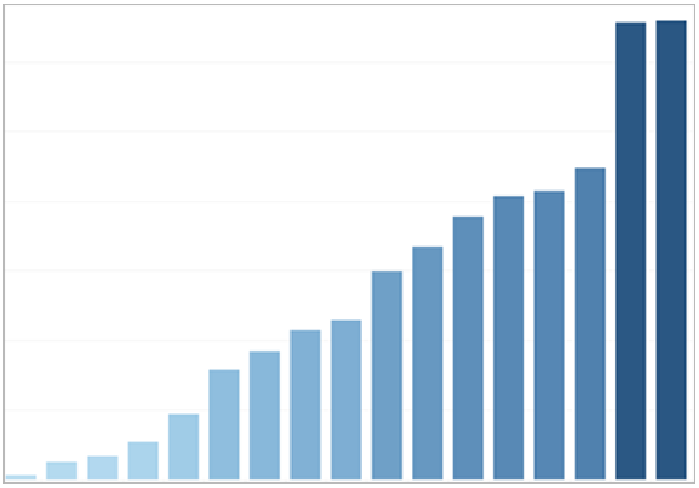
What types of question can this chart answer?

- Are events clustered around a certain probability?
- Which population group buys the most items?
- When are the busiest times in our work day?

Related topics

- **Build a Histogram** on page 1623
- **Build a Box Plot** on page 1590
- **Create a Pareto Chart** on page 1688
- **Create a Population Pyramid** on page 1694

Ranking



Sometimes you not only want to depict the magnitude of some value, but also the relative ranking of all the members of your dimension. Showing the top ten sales people or demonstrating the under-performing states use a ranking chart.

Ranking charts are usually bar charts that integrate rank calculations, top n sets, or key progress indicators.

What types of question can this chart answer?

- How many people are under-per-

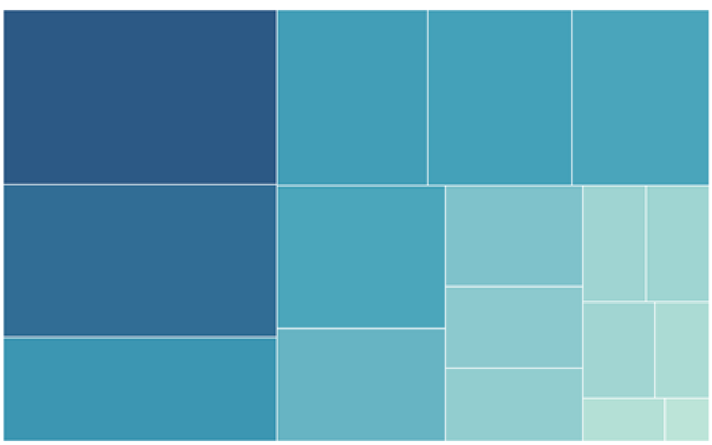
Related topics

- **Build a Bar Chart** on page 1579
- **Table Calculation Types** on page 2419
- **Sets for Top N and Others** on page 1209
- **Visualize Key Progress Indicators** on page 1684

forming in the company?

- How much revenue is generated by our top ten customers?
- What is the value of our ten lowest revenue properties?

Part-to-Whole



Part-to-Whole charts show how much of a whole an individual part takes up. For example, if you are showing how much each region contributes to overall sales, or how expensive each different shipping mode is for an individual product, you would use a part to whole chart.

Related topics

- **Build a Pie Chart** on page 1645
- **Build a Treemap** on page 1664
- **Create an Area Chart** on page 1577
- **Stack Marks** on page 1392

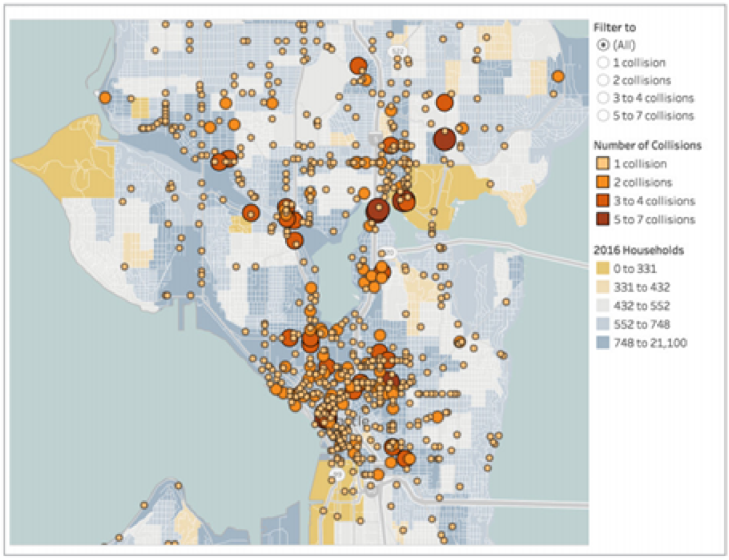
Part-to-Whole charts can be pie charts, area charts, stacked bar charts, or treemaps.

What types of question can this chart answer?

- How much does this value contribute to the total?
- How does the distribution of costs change each year?
- Do different items contribute different amounts to sales by region?

Spatial

Spatial charts can precise locations



Related topics

- **Build a Simple Map** on page 1848
- **Create Maps that Show Quantitative Values in Tableau** on page 1883
- **Create Maps that Highlight Visual Clusters of Data in Tableau** on page 1889

and geographical patterns in your data. Showing the airport terminals with the most foot traffic or a map of all sales across the country are examples of spatial maps.

Spatial maps include filled maps, point distribution maps, symbol maps, and density maps.

What types of question can this chart answer?

- Which city has the highest sales?
- How far from distribution centers are our customers?
- How many people arrive at

which gate?

Flow

Flow charts can be maps that convey movement over time, such as Sankey diagrams. Flow maps include path over time and path between origin and destination charts.

What types of question can this chart answer?

- What is the longest shipping route?
- How long are people lingering around gates?
- What are the bottlenecks to

Related topics

- **Create Maps that Show Paths Between Origins and Destinations in Tableau** on page 1917
- **Create Maps that Show a Path Over Time in Tableau** on page 1903

traffic in the city?

Related resources

For related content and examples of different types of visualizations you can create with Tableau, see:

- **Build Common Chart Types in Data Views** on page 1577 and **Build Advanced Chart Types** on page 1674 in Tableau help.
- Which Chart or Graph is Right for You? whitepaper on the Tableau website. Use your tableau.com account to sign in.
- Visual Vocabulary on Tableau Public by Tableau Zen Master Andy Kriebel (also see Andy's blog).

Also see these free training videos and presentations:

- Andy Cotgreave's Best of the Tableau Web and other blog posts on tableau.com.
- Getting Started with Visual Analytics (6 minutes) video tutorial. Use your tableau.com account to sign in. Also see different chart types under the How To category on the Free Training Videos page.
- Secrets of Visual Analytics This four-part, on-demand webinar series introduces how to use Tableau for visual analytics. Use your tableau.com account to sign in.
- Travel Tips for Your Visual Analysis Journey (50 minutes) video presentation from Tableau Conference.

Tableau Workbook Performance Checklist

Before you start building your visualizations and dashboards, you will want to be aware of what choices might impact performance in your workbooks on Tableau Desktop, Tableau Online, and Tableau Server.

This checklist is intended to make you aware of things you will want to consider related to workbook performance. Remember that all environments are different and there is no simple cure for workbook performance. The items below are areas to explore. Any given item may not apply to your situation or be the cause of poor performance. Always test what helps in your specific case.

Learn more: For more in-depth and related information beyond this topic, see: **Optimize Workbook Performance** on page 2993 topics in Tableau help

Designing Efficient Workbooks whitepaper on tableau.com
Designing Efficient Workbooks (1 hour) and Best Practices for Dashboard Performance (50 minutes) free, video presentations
The Tableau Performance Optimization Flowchart and The Tableau Performance Checklist in the InterWorks blog.

General tips

This general list of considerations is a start in learning how to approach optimizing workbooks. It comes from the TL;DR page in the Designing Efficient Workbooks whitepaper.

- The cleaner your data is and the better it matches the structure of your questions, the faster your workbooks will run and the happier your life will be. Only connect to the data that you need for analysis.
- Extracts are a quick and easy way to make most workbooks run faster. If you don't need real-time data and aren't working over billions of rows of data, you should try them.
- The majority of slow dashboards are caused by poor design—in particular, too many charts on a single dashboard, or trying to show too much data at once. Keep it simple. Allow your users to incrementally drill down to details, rather than trying to show everything then filter (guided analysis).
- Don't work with data you don't need—both in terms of the fields you reference as well as the granularity of the records you return. Use filters, hide unused fields and aggregate.
- While reducing the data, make sure you use filters efficiently.
- Strings and dates are slow, numbers and Booleans are fast.
- There is no silver bullet for inefficient workbooks. Start by looking at the performance recorder to understand where the time is going. Long-running queries? Lots of queries? Slow calculations? Complex rendering? Use insights to focus your efforts in the right direction.
- Upgrading to the newest version may boost performance without needing anything else.
- If it is slow in the data source, it will be slow in Tableau Desktop. If it is slow in Tableau Desktop, it will (almost always) be slow in Tableau Server.
- Performance tuning is highly individualized to the context of your environment, data, analysis, and workbook. What works in one situation may not work in another--test and iterate to see what works in your specific case.

Data source and structure

- Prepare your data to match the structure of your analysis before you connect to it in Tableau Desktop. Tableau Prep is a helpful tool for getting your data ready.
- Enable support for referential integrity.
- Index the tables in your relational database.
- Make sure database permissions support creating temp tables.
- Connect with native database drivers.
- Whenever possible, connect to an extract rather than a live data source.
- Limit the amount of data that you bring into Tableau to what is necessary for your analysis.
- Consider using aggregated extracts and extract filters.
- Optimize extracts and hide unused fields before creating an extract.
- Avoid using custom SQL.
- Identify when you should be joining, blending, or using a cross-database join.
- Keep joins as limited as possible. Consider using multiple data sources so each type of analysis is built off a well-designed join.
- Consider aggregating before blending.

Related topics and resources

- **Know Your Data at the Database Level** on page 2994
- **Test Your Data and Use Extracts** on page 2996
- **Create Efficient Combined Data Sources** on page 2998
- Designing Efficient Workbooks

Building views and dashboards

- Keep it simple. Break massive workbooks up into separate files, and consider using fewer sheets on a dashboard.
- Reduce the number of marks in the view—only add the fields you need, and use filters

judiciously.

- Turn off automatic updates when building a view. Set up multiple steps before asking Tableau to perform the actions.
- When possible, try to use the same level of detail on multiple sheets on a dashboard to enable query batching.
- Limit the number of filters used—use guided analytics and filter actions when possible.
- Use fixed-size dashboards. They don't have to be re-rendered for different window sizes.
- Custom geocoding can massively inflate the size of a workbook.

Related topics and resources

- **Design for Performance While You Build a View** on page 2999
- **Make Visualizations Faster** on page 3006
- **Turn off Automatic Updates to Boost Performance** on page 3014
- Designing Efficient Workbooks

Filtering

- Filtering is one of the most common causes of poorly performing workbooks and dashboards.
- Use Extract and Data Source filters to limit the amount of data being brought into Tableau.
- Filters are more efficient when the data source is indexed.
- Context filters are evaluated before other filters in the view and should be used to help force an order of operations, but they no longer help improve query performance.
- Be aware that using **Keep only** and **Exclude** to filter out discrete items can be performance heavy.
- Filtering on a categorical dimension value that summarizes many points—such as filtering a state instead of all the cities in that state—is more efficient.
- Filtering on a range of values is often faster than filtering on large itemized lists of discrete values.
- Cross-database filters can require multiple queries when values are updated. Enable the

Show Apply Button option on filters to let users apply the filter after making their selections.

- The **Only Relevant Values** filter option creates a query each time the other filters are changed. For dashboards that use guided analytics, try using **Filter Actions** on page 1491 instead.
- Applying filters to multiple worksheets creates multiple queries.
- Enable the **Show Apply Button** option on filters. This lets you and your users click **Apply** to run a single query for multi-select filters.

Related topics and resources

- **Filter Your Data Carefully** on page 3000
- Designing Efficient Workbooks

Calculations

- Booleans and integers are faster than strings and dates.
- MIN and MAX functions perform better than AVG and ATTR.
- When possible, use native Tableau features—groups, sets, bins, custom date fields, combined fields, and aliases—instead of calculations.
- Use optimized functions like CONTAINS and DATEPARSE or MAKEDATE when possible, rather than more complicated manual calculations, especially with dates.
- COUNTD is one of the slowest functions types, avoid when possible.
- If a table calculation is performing badly, see if it can be expressed via a LOD expression, and vice versa.

Related topics and resources

- **Create Efficient Calculations** on page 3003
- Designing Efficient Workbooks

Other Suggestions

- Publishing to Tableau Server will not improve performance. Tune your workbook in Tableau Desktop first.
- Use the performance recorder to identify where to focus tuning efforts.
- Upgrade to the latest version of your Tableau product.

Related topics and resources

- **Record and Analyze Workbook Performance** on page 3008
- Designing Efficient Workbooks

About Tableau Help

Addressing Implicit Bias in Technical Language

In an effort to align with one of our core company values, equality, we have changed terminology to be more inclusive where possible. Because changing terms in code can break current implementations, we maintain the current terminology in the following places:

- Tableau APIs: methods, parameters, and variables
- Tableau CLIs: commands and options
- Installers, installation directories, and terms in configuration files
- Tableau Resource Monitoring Tool (we plan to make changes to non-inclusive terminology in the web interface, error messages, and related documentation soon.)
- Third-party systems documentation

For more information about our ongoing effort to address implicit bias, see Salesforce Updates Technical Language in Ongoing Effort to Address Implicit Bias on the Salesforce website.

Connect to and Prepare Data

Before you begin your analysis, you must connect to your data and then set up the data source. There are many optional configurations that you can make on the Data Source page that Tableau uses to interpret and interact with your data.

The topics in this section describe how to connect to your data and use these configurations in order to optimize your data source for analysis.

Connect to Your Data

Before you can build a view and analyze your data, you must first connect Tableau to your data. Tableau supports connecting to a wide variety of data, stored in a variety of places. For example, your data might be stored on your computer in a spreadsheet or a text file, or in a big data, relational, or cube (multidimensional) database on a server in your enterprise. Or, you might connect to public domain data available on the web such as U.S. Census Bureau information, or to a cloud database source, such as Google Analytics, Amazon Redshift, or Salesforce.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to connecting to published data sources, you can connect to databases and tables from **Tableau Server** on the **Connect** pane in Tableau Desktop. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Tableau Desktop

When you launch Tableau Desktop, the data connectors that are available to you are listed on the **Connect** pane, which is the left pane on the **Start** page. Under **Search for Data**, select **Tableau Server** to find data using Tableau Server or Tableau Online. File types are listed next, then common server types, or servers that you've recently connected to. Click **More** to see the complete list of data connectors you can use.

For supported files and databases, Tableau provides native connectors that are built for and optimized for those types of data. If your file or database type is listed under **Connect**, use this native connector to connect to your data. If your file or database type is not listed, you might have the option of creating your own connection using **Other Databases (JDBC)**, **Other**

Databases (ODBC), a **Web Data Connector**, or a **Connector Plugin** built using the Tableau Connector SDK. Tableau provides limited support for connections that you create using these options.

The data connectors supported by your copy of Tableau Desktop are determined by the version you purchased. For more information, see the list of data connectors on the Tableau website. After you've connected to data, you can save the connections to have them show up under the **Saved data sources** section on the **Connect** pane.

You supply different information for each data connection that you want to make. For example, for most data connections, you'll need to supply a server name and your sign-in information. With some data connections, you can **Run Initial SQL** on page 736 statements, and SSL-enabled servers require that you select the **Require SSL** check box when you connect. The following sections discuss the specific information you need to provide for each type of data you want to connect to.

> **Tip:** You can quickly create a data source in Tableau by copying and pasting data using the clipboard. For more information, see **Create a Data Source or Add a New Connection with Clipboard Data** on page 957.

Tableau Server and Tableau Online web authoring

Sign in to your Tableau site and select **Create Workbook** on the Home page to open the **Connect to Data** page. The tabs that you see on the page depend on the product you have.

Tableau Server

On Tableau Server, select from the following tabs to connect to data:

- **On this site**. Browse to or search for published data sources.
- **Files**. Upload Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser.
- **Connectors**. Connect to data housed in a cloud database or on a server in your enterprise.

> **Note:** When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On This Site** to connect to databases and tables, as well as data sources.

For more information about connecting to data, see **Creators: Connect to data on the web** on page 3135.

Tableau Online

On Tableau Online, select from the following tabs to connect to data:

- **On this site**. Browse to or search for published data sources.
- **Files**. Upload Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser.
- **Connectors**. Connect to data housed in a cloud database or on a server in your enterprise.
- **Dashboard Starters**. Use to author and analyze data from Oracle Eloqua, Salesforce, ServiceNow ITSM, and QuickBooks Online.

Note: When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On This Site** to connect to databases and tables, as well as data sources.

For more information about connecting to data, see **Creators: Connect to data on the web** on page 3135.

The data connectors supported by your Tableau site are determined by your site's server and your license level. For more information, see **What Can I Do with a Tableau Site?** on page 3102

After you've connected to data, you can save the connections to have them show up in the **Data Sources** section of your site.

Request a new connector

If Tableau doesn't have a native (built-in) connector for your data, consider requesting one on Tableau Community. Use Ideas on Community to search for your connector to see if it's been requested, and if it has been, vote for it. If it's not listed, add it. Tableau regularly reviews Ideas on Community to help determine what features should be added to the product.

Supported Connectors

Follow the link below for information on how to connect to your specific data. Connectors are listed in the order that they appear on the **Connect** pane.

Excel

This article describes how to connect Tableau to Microsoft Excel file data and set up the data source. Tableau connects to .xls and .xlsx files.

To connect to a .csv file, use the Text file connector.

Make the connection and set up the data source

1. After you open Tableau, under **Connect**, click **Excel**.
2. Select the Excel workbook you want to connect to, and then click **Open**.

    NOTE: Beginning with Tableau 2020.2, legacy Excel and Text connections are no longer supported. See the Legacy Connection Alternatives document in Tableau Community for alternatives to using the legacy connection.

3. On the Data Source page, do the following:
    1. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    2. If your Excel file has one table, click the sheet tab to start your analysis.

        Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

        You can also connect to a named range or an Excel table (also known as an Excel list) in the same way you connect to a worksheet. Both the named range and Excel table function as a table in Tableau.

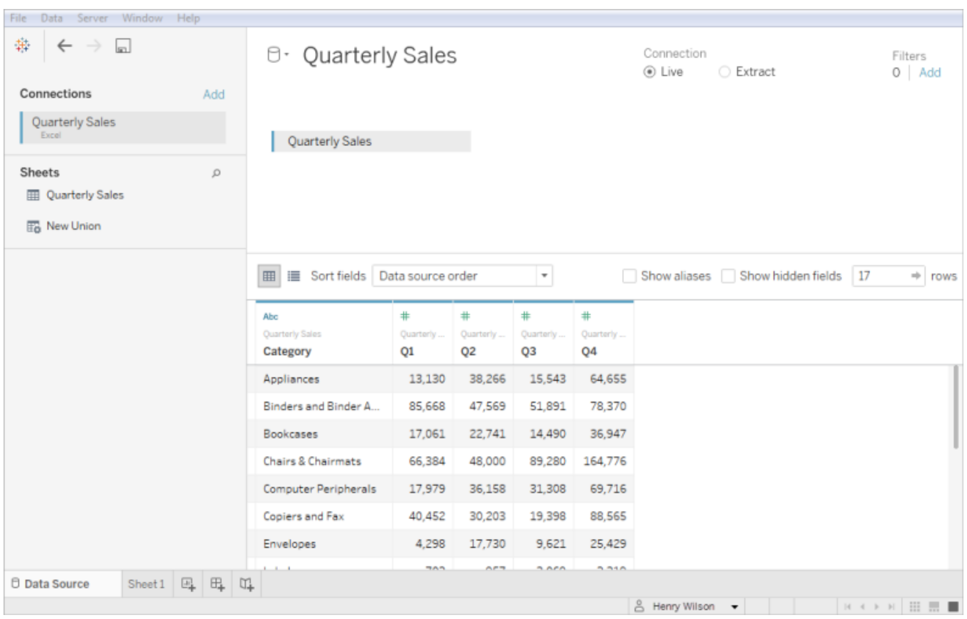
        You create named ranges in Excel by selecting a range of cells and then selecting **Define Name** on the **Formulas** tab. Similar to named ranges, you can create an Excel table in Excel by selecting a range of cells, and then selecting **Insert** > **Table**. When you connect to a named range or Excel table in Tableau, an icon appears next to the sheet in the Data Source page as shown below.

You can connect to multiple Excel workbooks at the same time as long as each connection in the data source has a unique name.

> **Note:** Tableau does not support pivot tables in Excel.

Microsoft Excel data source example

The following is an example of a Microsoft Excel data source.



> **Note:** If the Excel file contains columns that are more than 254 characters wide, Tableau Desktop can't use these fields for workbooks that were created before Tableau Desktop 8.2. Also, you cannot use the legacy connection to connect to this data. Either remove the columns, modify them to fit within 254 characters prior to connecting in Tableau Desktop,

or upgrade the Excel data source. For more information about upgrading data sources, see **Upgrade Data Sources** on page 1077.

Get more data

Get more data into your data source by adding more tables or connecting to data in a different database.

- **Add more data from the current file:** From the left pane, drag additional tables to the canvas to combine data using a join or union. For more information, see **Join Your Data** on page 883 or **Union Your Data** on page 942.
- **Add more data from different databases:** In the left pane, click **Add** next to Connections. For more information, see **Join Your Data** on page 883.

  If a connector you want is not listed in the left pane, select **Data** > **New Data Source** to add a new data source. For more information, see **Blend Your Data** on page 910.

Set Excel table options

Excel table options are scoped to the connection. To change the table options, on the canvas, click the table drop-down arrow and then specify whether the data includes field names in the first row. If so, these names will become the field names in Tableau. If field names are not included, Tableau generates them automatically. You can rename the fields later.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

Changes to the way values are computed

Starting from version 10.5, when you are working with extract data sources as well as data sources that use live connections to file-based data like Excel, the values in your data can be computed differently from previous versions of Tableau. This change means that you might see differences between the data and the marks in your view between version 10.4 (and earlier) and version 10.5 (and later). The purpose of this change is to improve the efficiency and scalability of your Excel data source. For more information, see **Changes to values and marks in the view** on page 1015.

In the case of an Excel data source, one example of this change is with case sensitivity. In version 10.4 (and earlier), for comparing and sorting purposes, string values are treated as case insensitive and therefore treated the same and stored as a single string value. In version 10.5 (and later), for sorting and comparing purposes, values remain case insensitive. However, values are case sensitive for storing purposes. This becomes evident when values are displayed on the data source page.

For example, suppose you have a column in your data that contains the values "House," "HOUSE," and "houSe." You see the following string values depending on the version of Tableau you are using:

- In version 10.4 and earlier, both on the data source page and view, you see: "House," "House," and "House."
- In version 10.5 and later, on the data source page you see: "House," "HOUSE," and "houSe." But in the view you see: "House," "House," and "House."

If you need to maintain case sensitivity for your data when performing a join, you can enable the **Maintain Character Case (Excel)** option from the Data menu. For more information about this option, see **Join Your Data** on page 883.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

- Data Prep with Text and Excel Files - Watch the Tableau video (registration or sign in required).

Tips for Working with Your Data

Data can be organized in various ways. To take advantage of Tableau Desktop, Tableau recommends that you connect to data that is formatted for analysis. Specifically, data that is:

- as granular as possible rather than aggregated (such as daily weather data rather than monthly averages)
- organized like a database table (rather than a column-oriented table such as a crosstab)
- stripped of extraneous information (anything that's not the data and its headers)

When data is structured for analysis, it's much easier to ask and answer questions. Tableau can aggregate raw data to the desired level, rather than being restricted by the aggregations already present in the data. Groups and hierarchies can be created as needed, and calculations can be performed in the flow of analysis.

Tableau Desktop has basic cleaning options and the Data Interpreter. Tableau Prep may be necessary for more complex formatting issues.

The following Tableau Desktop-specific sections highlight and provide suggestions for resolving some common formatting or issues that can make analyzing your data difficult.

Pivot data in crosstab format

When data is formatted as in crosstab format, the table is column oriented. In a column oriented table, the variables are stored as column headers. However, Tableau Desktop is optimized for row oriented data. In a row-oriented table, the variables are stored in the row values.

For example, suppose you have column-oriented table, which shows math, science, and history scores for grade school students.

ID	Gender	School	Math	Science	History
1	M	West	90	80	80
2	F	South	50	50	50
3	M	Central	50	80	80

Tableau Desktop is optimized to connect to row-oriented tables, where math, science, and history values are organized under a column called "Subject" and the scores for each student are organized under a column called "Score." You can pivot the columns in the rows, by manually editing your Excel data. Alternatively, connect to your Excel data from Desktop and then use the pivot option. For more information about the pivot option, see **Pivot Data from Columns to Rows** on page 988.

ID	Gender		School	Subject	Score
1	M		West	Math	90
1	M		West	Science	80
1	M		West	History	80
2	F		South	Math	50
2	F		South	Science	50
2	F		South	History	50
3	M		Central	Math	50
3	M		Central	Science	80
3	M		Central	History	80

Remove pre-aggregated data

Data can often come pre-aggregated. That is, data can contain sums, averages, medians, etc. A common of example of pre-aggregated data comes in the form of subtotals and grand totals. Subtotals and grand totals data are computed from the raw data, but is not part of raw data itself.

For example, suppose you have a table that contains a row of subtotals information.

ID	Gender	School	Math	Science	History
1	M	West	90	80	80
2	F	South	50	50	50
3	M	Central	50	80	80
4	M	Central	100	90	80
5	F	West	90	100	80
		Subtotals	**380**	**400**	**370**

In this case, pre-aggregated data needs to be removed. To use subtotals and grand totals in your analysis, manually remove this type of information from your table. Then, connect to you Excel data from Desktop and calculate subtotals and totals using the totals option. For more information, see **Show Totals in a Visualization** on page 1456. Alternatively, connect to your Excel data from Desktop, turn on Data Interpreter, and then use the totals option. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

ID	Gender	School	Math	Science	History
1	M	West	90	80	80
2	F	South	50	50	50
3	M	Central	50	80	80
4	M	Central	100	90	80
5	F	West	90	100	80
		~~Subtotals~~	~~380~~	~~400~~	~~370~~

Remove or exclude introductory text

Excel data that is delivered as a report can contain titles or blocks of introductory text. Because Desktop expects either column headers or row values in the first row of a table, this information can cause problems during your analysis.

For example, suppose you have a table that contains a report title and date.

Excel Report						
January, 2017						
	ID	**Gender**	**School**	**Math**	**Science**	**History**
	1	M	West	90	80	80
	2	F	South	50	50	50
	3	M	Central	50	80	80
	4	M	Central	100	90	80
	5	F	West	90	100	80

In this case, the title and date information needs to be removed. To use a title and date for a report, do one of the following:

- Manually remove this information from your Excel data. Then connect to your Excel data from Desktop and add a report title using the title option. For more information, see **Format Titles, Captions, Tooltips, and Legends** on page 2937.
- Connect to your Excel data from Desktop, turn on Data Interpreter, and then use the title option. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.
- If you cannot remove this information from your Excel data, create a named range and connect to the named range from Desktop. For more information, see **Excel** on page 394.

~~Excel Report~~						
~~January, 2017~~						
	ID	**Gender**	**School**	**Math**	**Science**	**History**
	1	M	West	90	80	80
	2	F	South	50	50	50
	3	M	Central	50	80	80
	4	M	Central	100	90	80
	5	F	West	90	100	80

Flatten hierarchical headers to a single row

In general, Tableau Desktop expects only the first row in your Excel data to contain column headers. Data that contains multiple layers of column headers can cause problems during your analysis.

For example, suppose you have a table that contains one major header and multiple sub-headers.

California					
ID	**Gender**	**School**	**Math**	**Science**	**History**
1	M	West	90	80	80
2	F	South	50	50	50
3	M	Central	50	80	80
4	M	Central	100	90	80
5	F	West	90	100	80

In this case, the hierarchy of headers must be flattened or removed. To do this, you can manually create a new column for each header in the hierarchy directly in your Excel data. Alternatively, connect to your Excel data from Tableau Desktop and then turn on Data Interpreter. Verify that your headers are flattened correctly. For more information about Data Interpreter, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

ID	**State**	**Gender**	**School**	**Math**	**Science**	**History**
1	California	M	West	90	80	80
2	California	F	South	50	50	50
3	California	M	Central	50	80	80
4	California	M	Central	100	90	80
5	California	F	West	90	100	80

Make sure there are no blank cells

If you create new columns for your hierarchical headers, make sure that each cell in the new columns contains values.

ID	State	Gender	School	Math	Science	History
1	California	M	West	90	80	80
2		F	South	50	50	50
3		M	Central	50	80	80
4		M	Central	100	90	80
5		F	West	90	100	80
6	Oregon	M	North	70	80	100
7		F	East	80	80	100
8		F	West	50	80	80
9		F	West	100	80	100
10		M	West	80	80	90

While you might repeat the same value for each row, it's important that each row contains the data that associates it with the data that was stored in the hierarchical header. You must manually remove blank cells from your Excel data.

ID	State	Gender	School	Math	Science	History
1	California	M	West	90	80	80
2	California	F	South	50	50	50
3	California	M	Central	50	80	80
4	California	M	Central	100	90	80
5	California	F	West	90	100	80
6	Oregon	M	North	70	80	100
7	Oregon	F	East	80	80	100
8	Oregon	F	West	50	80	80
9	Oregon	F	West	100	80	100
10	Oregon	M	West	80	80	90

Remove blank rows

Make sure that there are no blank rows in your data. To fix blank rows, you must remove the blank rows from your Excel data.

ID	Gender	School	Math	Science	History
1	M	West	90	80	80
2	F	South	50	50	50
3	M	Central	50	80	80
4	M	Central	100	90	80
5	F	West	90	100	80
22	M	North	70	80	100
23	F	East	80	80	100
24	F	West	50	80	80
25	F	West	100	80	100
26	M	West	80	80	90

Add missing headers

Make sure that there are no missing column headers. To fix missing headers, you must manually add the missing headers directly to your Excel data.

		Gender	School	Math	Science	History
6	Oregon	M	North	70	80	100
7	Oregon	F	East	80	80	100
8	Oregon	F	West	50	80	80
9	Oregon	F	West	100	80	100
10	Oregon	M	West	80	80	90

Text File

This article describes how to connect Tableau to text file data and set up the data source. Tableau connects to delimited text files (*.txt, *.csv, *.tab, *.tsv).

Make the connection and set up the data source

1. After you open Tableau, under **Connect**, click **Text File**.
2. Select the file you want to connect to, and then click **Open**.

> **NOTE:** Beginning with Tableau 2020.2, legacy Excel and Text connections are no longer supported. See the Legacy Connection Alternatives document in Tableau Community for alternatives to using the legacy connection.

3. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to. The default name is automatically generated based on the file name.

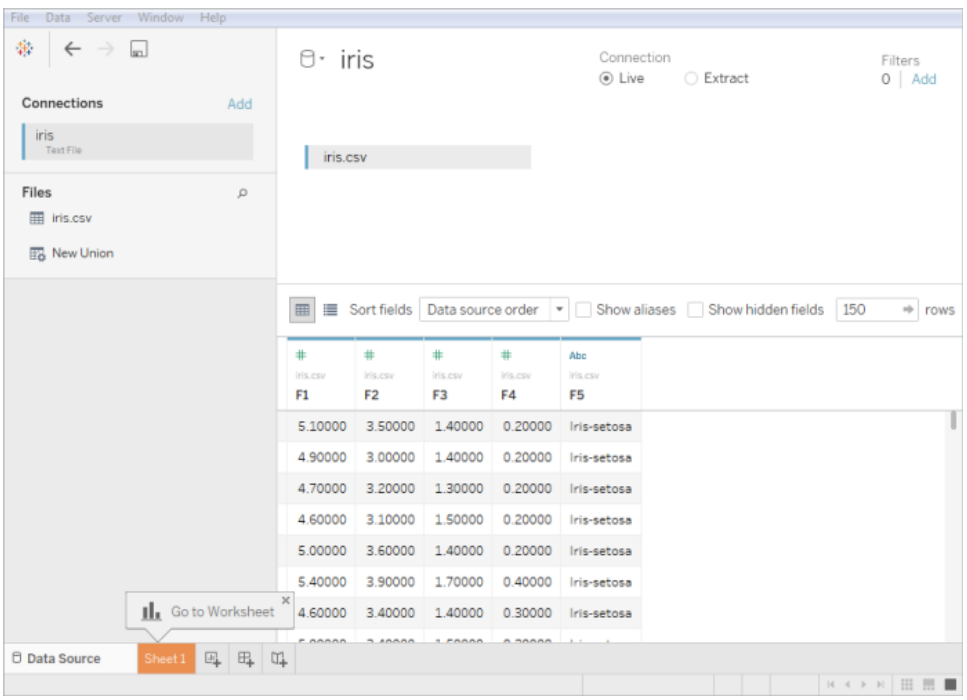
   b. Click the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

   Note: For text files, custom SQL is available only when using the legacy connection or in workbooks that were created before Tableau Desktop 8.2.

Text file data source example

Here is an example of a text file data source:



Optional settings

You can set the following options before building the view.

Get more data

Get more data into your data source by adding more tables or connecting to data in a different database.

- **Add more data from the current file:** From the left pane, drag additional tables to the canvas to combine data using a join or union. For more information, see **Join Your Data** on page 883 or **Union Your Data** on page 942.
- **Add more data from different databases:** In the left pane, click **Add** next to Connections. For more information, see **Join Your Data** on page 883.

  If a connector you want is not listed in the left pane, select **Data** > **New Data Source** to add a new data source. For more information, see **Blend Your Data** on page 910.

Clean your data with Data Interpreter

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

Set text file options

On the canvas, click the table drop-down arrow and then select whether the first row contains column names. This option is selected by default. Alternatively, you can have Tableau generate names when you connect. These names can be changed later. You can also select **Text file properties** to specify the following:

- Select the character that is used to separate the columns. Select from the list of characters or select **Other** to type in a custom character.
- Select the text qualifier that encloses values in the text file.
- Select a character set that describes the text file encoding. The available encodings are based on the operating system you are using. For example, on Windows, ANSI is listed as windows-1252 and OEM is listed as 437.

  > **Note:** In workbooks created prior to Tableau Desktop 8.2 or that use the legacy connection, you can select ANSI, OEM, UTF-8, UTF-16, or Other. When you select

Other, you must specify the character set in the provided text field. This value is verified when the connection is attempted.

- Select the locale by which the file should be parsed. This option tells Tableau which decimal and thousands separator to use.

Review the data, pivot, split, and create calculations

The first 1,000 rows of the data in the data source are automatically displayed below the canvas in the data grid. If you add tables, remove tables, or make changes to the join conditions, the data grid updates with your changes. You can also do the following in the data grid:

- Change the data type or geographical role of a column by clicking the data type icon.
- Hide a field by clicking the column drop-down arrow and selecting **Hide**.
- Rename a field by double-clicking the field name.
- Reset a field name by clicking the column drop-down arrow and selecting **Reset Name**.
- Sort fields in both the data grid and metadata grid by selecting a sort option from the **Sort fields** drop-down list.
- Sort rows in the data grid by clicking the sort button next to the column name.
- Pivot fields to transform data in a crosstab format into a columnar format. This is only available for non-legacy connection types. For more information, see **Pivot Data from Columns to Rows** on page 988.
- Split a string field into multiple fields. This is only available for non-legacy connection types. For more information, see **Split a Field into Multiple Fields** on page 1001.
- Create new calculations based on existing field in the Tableau data source. Click the column drop-down arrow and select **Create Calculated Field**.
- Copy values in the grid by selecting the values and then pressing Ctrl+C (Command-C on a Mac). Alternatively, to copy values in the metadata grid, select the values, right-click (Control-click on a Mac), and then select **Copy**.

Examine the data source structure and perform management tasks

Use the metadata area to quickly examine the general structure of the Tableau data source and its fields. If you are working with a particularly large data source, use the metadata area to perform routine data management tasks such as hide multiple fields at once or quickly rename or reset fields.

Connect live or use an extract

At the top of the Data Source page, select **Live** or **Extract** mode. If you choose to create an extract, the **Edit** link displays. Click **Edit** to set up filters that define a subset of the data that you want to include in the extract. For more information, see **Extract Your Data** on page 1014.

Add data source filters

At the top of the data source page, click **Add** to add data source filters to control what data is used in included in the data source and therefore restrict the visibility and use of the fields in the data source.

Best practices

Collect files in a single directory

Collect all relevant text files for a multi-table connection in a single directory, with nothing else in that directory, so that users cannot inadvertently select a file that is not appropriate for the connection.

Text file column width and size considerations

If the text file contains columns that are more than 254 characters wide, Tableau cannot use these fields for workbooks that were created before Tableau Desktop 8.2 or that use the legacy connection. Either remove the columns, modify them to fit within 254 characters prior to connecting in Tableau, or upgrade the text file data source.

Large text files often perform poorly as a data source, because the queries can take a long time.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Access

This article describes how to connect Tableau to a Microsoft Access file (*.mdb, *.accdb) and set up the data source. Tableau supports all Access data types except OLE Object and Hyperlink.

Before you begin

Before you begin, gather this connection information:

- Access file name.
- If the file is password protected, you need the database password.
- If the file has workgroup security, you need workgroup security credentials:
  - Workgroup file name
  - User
  - Password

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Access**, select the Access file that you want to connect to, and then select **Sign In**.
   - Password protected - If the Access file is password protected, select **Database Password**, and then enter the password.

- Workgroup security - If the Access file is protected by workgroup security, select **Workgroup Security**, and then enter the **Workgroup File** name, **User**, and **Password** into the corresponding text fields.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to. The default name is automatically generated based on the file name.

   b. Drag a table to the canvas. You can drag a table or query.

   c. Select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: If the Access file contains columns that are more than 254 characters wide, Tableau can't use these fields. Either remove the columns from the table or modify them to fit within 254 characters prior to connecting with Tableau.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

JSON File

This article describes how to connect Tableau to a local JSON file and set up the data source.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **JSON File**. Then do the following:

   a. Select the file you want to connect to, and then select **Open**.

   b. In the Select Schema Levels dialog box, select the schema levels you want to view and analyze in Tableau, and then select **OK**. For more information, see **Select schema levels** below.

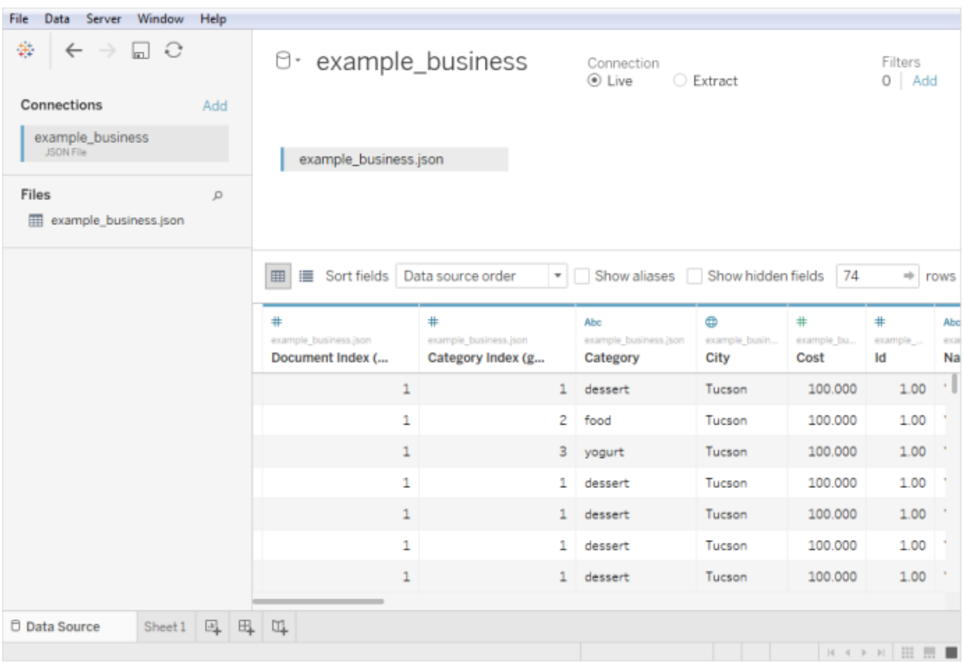
2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Select the sheet tab to start your analysis.

JSON file data source example

Here is an example of a JSON file as a data source using Tableau Desktop on a Windows computer:



Select schema levels

When you connect Tableau to a JSON file, Tableau scans the data in the first 10,000 rows of the JSON file and infers the schema from that process. Tableau flattens the data using this inferred schema. The JSON file schema levels are listed in the Select Schema Levels dialog box. In Tableau Desktop, if your JSON file has more than 10,000 rows, you can use the "Scan Entire Document" option to create a schema.

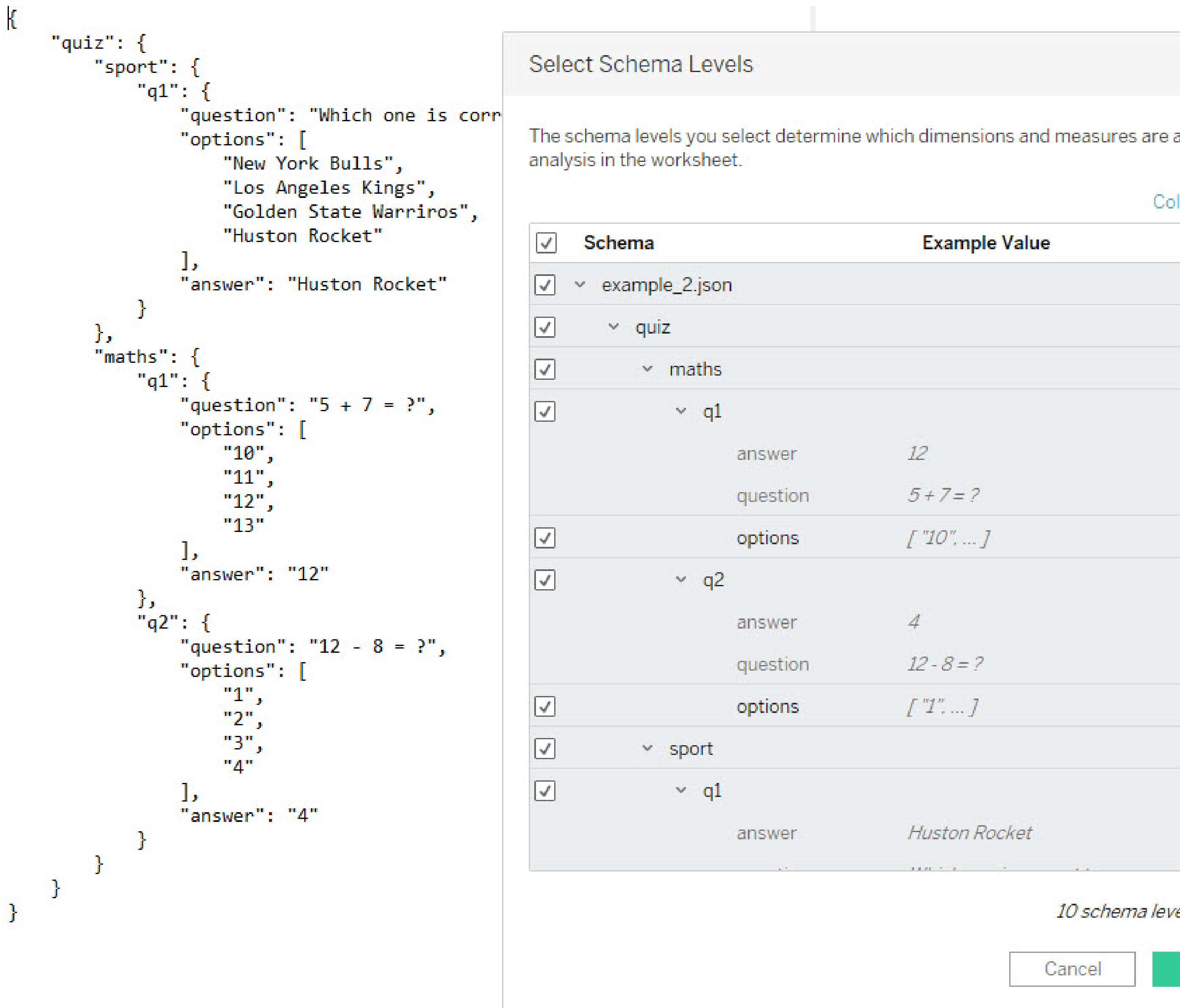
Note: The "Scan Entire Document" option displays only for JSON files with more than 10,000 rows. It is not available on the web.

The schema levels that you select in the dialog box determine which dimensions and measures are available for you to view and analyze in Tableau. They also determine which data is published.

Note: When you publish a workbook to the web, any schema updates will not be available to the web version of the viz and may cause errors if the viz is subsequently refreshed

If you select a child schema level, the parent level is also selected.

For example, here's a snippet of a JSON file:	The JSON file generates these schema levels:



Detect new fields

Sometimes, more fields exist in rows that were not scanned to create the inferred schema. If you notice that a field you need is missing under **Schema**, you can choose to do one of the following:

- Scan the entire JSON document. It may take a long time for the scan to complete.
- Select schema levels from the schema listed and then select **OK**. Tableau reads your

entire document and if more fields are found, they are listed in the Select Schema Levels dialog box.

Whenever Tableau detects that new fields are available, for example, during an extract refresh or when Tableau creates an extract after you've selected the schema levels, either an information icon near the file name or a notification on the Select Schema Levels dialog box will indicate that additional fields have been found.

Change schema levels

You can change the schema levels you selected by going to the data source page and selecting **Data > [JSON file name] > Select Schema Level**. Or, hover over the file name on the canvas and select the drop-down menu > **Select Schema Level**.

Union JSON files

You can union JSON data. To union a JSON file, it must have a .json, .txt, or .log extension. For more information about union, see **Union Your Data** on page 942.

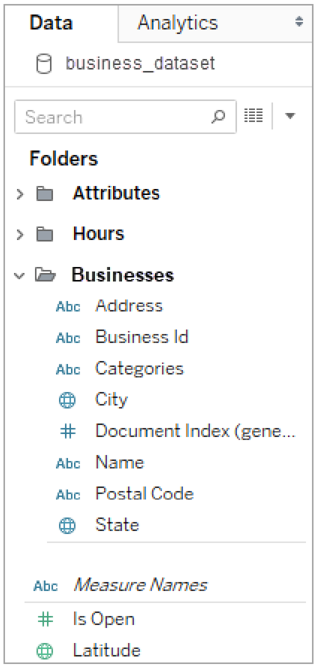
When you union JSON files, the schema is inferred from the first 10,000 rows of every file in the union.

You can change the schema levels after you union files. For more information, see **Change schema levels** above.

How dimension folders are organized for hierarchical JSON files

After you select the sheet tab, the selected schema levels of your JSON file show under **Dimensions** on the **Data** pane. Each folder corresponds with the schema level you selected, and the attributes associated with that schema level are listed as children of the folder.

For example, in the following image, **Address** is a dimension under the schema level **Businesses** folder. **Categories** is also a schema level, but because it is a list of values and not a hierarchy of data, it doesn't require its own folder, but is instead grouped under a parent folder. Note that schema levels in the Select Schema Levels dialog box do not map directly to the folder structure in the **Data** pane. Folders in the **Data** pane are grouped by object so that you can easily navigate to fields and still have context for where the fields come from.



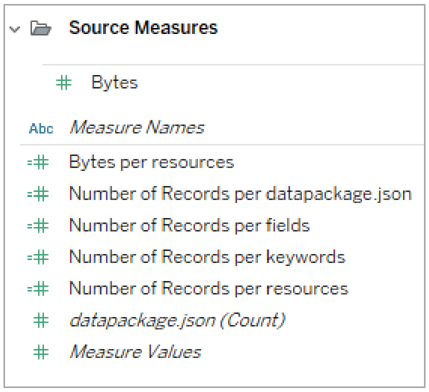
For each document, a unique index is generated and stored in the flattened representation of data. An index is also generated for each level in the schema.

For example, in the image above, in addition to the **Document Index (generated)** entry, the **Attribute** and **Hours** schema levels all have generated indexes.

Why measures are calculated in hierarchical JSON files

When a hierarchical JSON file is flattened, data might be duplicated. To keep measures consistent with their schema levels, Tableau creates level of detail (LOD) calculations to accurately represent the data at the schema level. The original measures are located in the **Source Measures** folder and you can use them, but we recommend that you use the calculated measures.

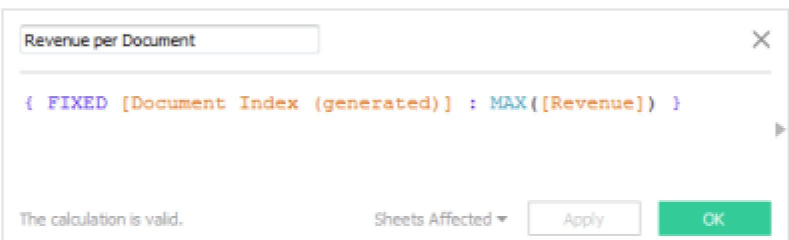
On the **Data** pane, calculated measures are labeled as **Number of <measure name> per <parent name>**.



To view the LOD calculation for a measure, follow these steps:

1. Select the measure.
2. Select the drop-down arrow, and then select **Edit**.

The following example shows the LOD calculation for **Revenue per Document**. The formula selects the maximum revenue for each document index value.



The use of LOD calculations means that you can select multiple schema levels and be confident that the measures are not over-counted.

Tips for working with JSON data

These tips can help you work with your JSON data in Tableau.

- Do not exceed the 10x10 limit for nested arrays.

  A high number of nested arrays creates a lot of rows. For example, 10x10 nested arrays result in 10 billion rows. When the number of rows Tableau can load into memory is exceeded, an error displays. In this case, use the Select Schema Levels dialog box to reduce the number of selected schema levels.

- A data source that contains more than 100 levels of JSON objects can take a long time to load.

A high number of levels creates a lot of columns, which can take a long time to process. As an example, 100 levels can take more than two minutes to load the data. As a best practice, reduce the number of schema levels to just the levels that you need for your analysis.

- A single JSON object cannot exceed 128 MB.

  When a single object top-level array exceeds 128 MB, you must convert it to a file where the JSON objects are defined one per line.

- The pivot option is not supported.

About TTDE and HHYPER files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

PDF File

This article describes how to connect Tableau to .pdf file data and set up the data source.

Note: Tableau doesn’t support right-to-left (RTL) languages. If your PDF includes RTL text, characters might display in reverse order in Tableau.

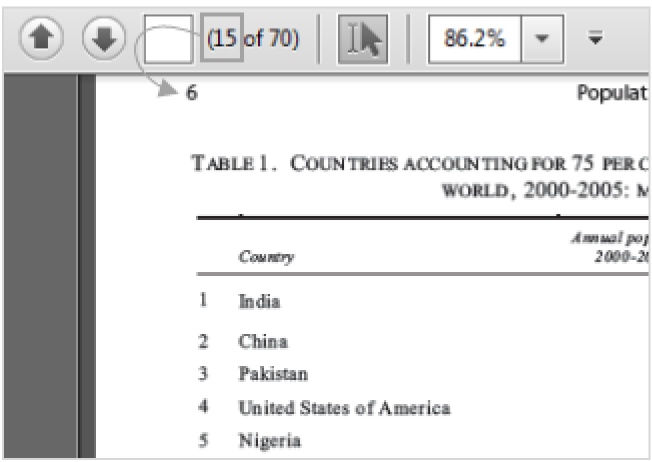
Make the connection and scan your document for tables

1. After you open Tableau, under **Connect**, click **PDF File**.
2. Select the file you want to connect to, and then click **Open**.

3. In the Scan PDF File dialog box, specify the pages in the file that you want Tableau to scan for tables. You can choose to scan for tables in all pages, just a single page, or a range of pages.

   Note: The scan counts the first page of the file as page 1, similar to most PDF readers. When you scan for tables, specify the page number that the PDF reader displays and not the page number that might be used in the document itself, which may or may not start from page 1.

   For example, suppose you want to use "Table 1" from the image below. The PDF reader displays a number, and the .pdf file displays a different number. To correctly scan for this table, specify the page number that the PDF reader displays. In this example, you specify page 15.



4. On the data source page, do the following:

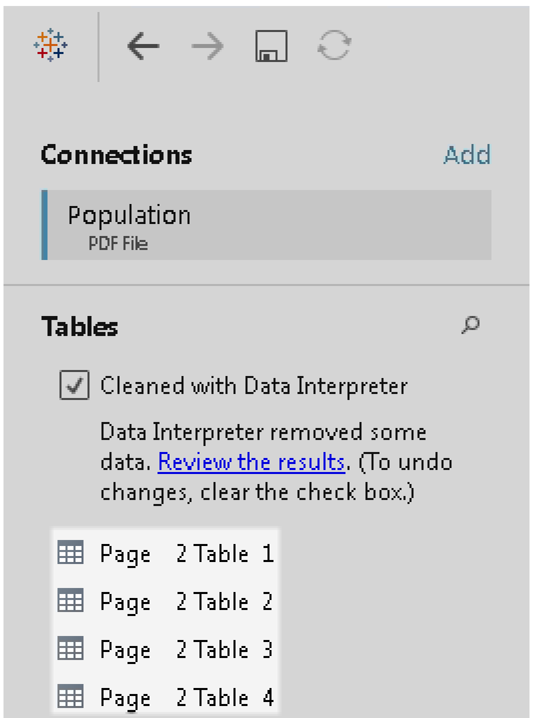
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to. The default name is automatically generated based on the file name.

   b. If your file contains one table, click the sheet tab to start your analysis. Otherwise, from the left pane drag a table onto the canvas and then click the sheet tab to start your analysis.

   About the tables in the left pane

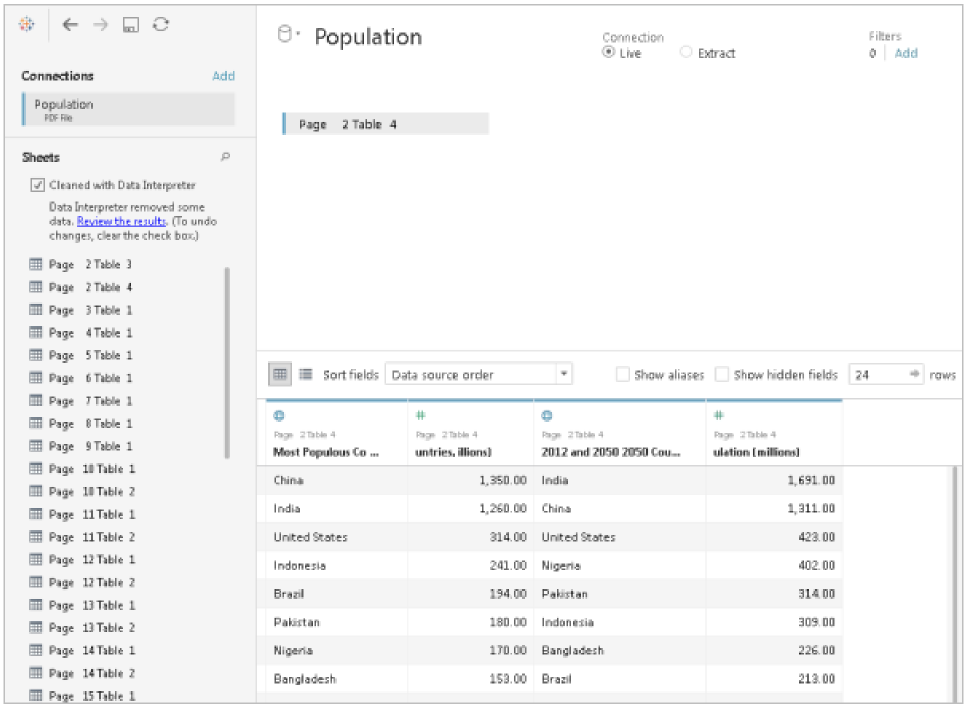
Tables that are identified in the .pdf file are given unique names and are displayed in the left pane after a scan. For example, you might see a table name like "Page 1, Table 1." The first part of the table name indicates the page in the .pdf file the table came from. The second part of the table name indicates the order the table was identified. If Tableau has identified more than one table on a page, the second part of the table name can indicate one of two things:

- Tableau has identified another unique table or sub-table on the page.
- Tableau has interpreted the table on the page in another way. Tableau might provide multiple interpretations of a table depending on how the table is presented in your .pdf file.



PDF file data source example

Here is an example of a PDF file data source:

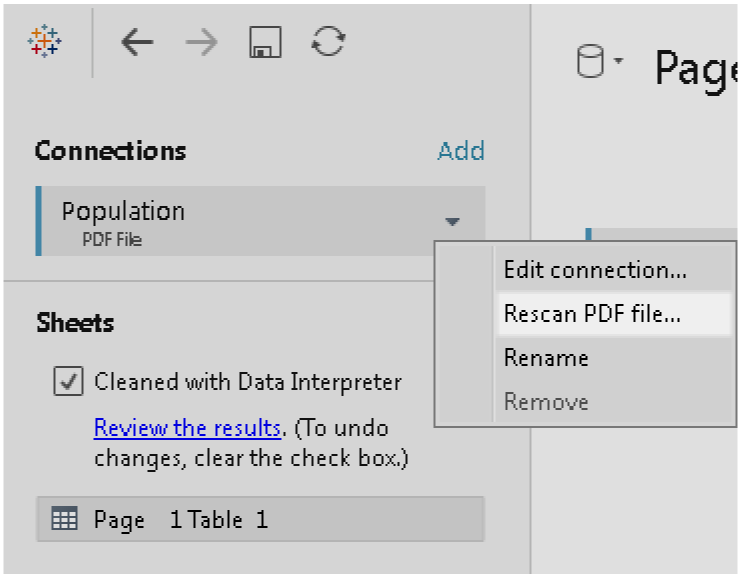


Get more data

Get more data into your data source by adding more tables or connecting to data in a different database.

- **Add more data from the current file:**
  - From the left pane, drag additional tables to the canvas to combine data using a join or union. For more information, see **Join Your Data** on page 883 or **Union Your Data** on page 942.
  - If the pages that were scanned in step 3 of the procedure listed above do not produce the tables that you need in the left pane, click the drop-down arrow next to the PDF File connection, and click **Rescan PDF file**. This option allows you to create a new scan so that you can specify different pages in the .pdf file to scan for

tables.



- **Add more data from a different database:** In the left pane, click **Add** next to Connections. For more information, see **Join Your Data** on page 883.

  If a connector you want is not listed in the left pane, select **Data** > **New Data Source** to add a new data source. For more information, see **Blend Your Data** on page 910.

Set table options

You can set table options. On the canvas, click the table drop-down arrow and then specify whether the data includes field names in the first row. If so, these names will become the field names in Tableau. If field names are not included, Tableau generates them automatically. You can rename the fields later.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

Union tables in your .pdf files

You can union tables in your file. For more information about union, see **Union Your Data** on page 942.

When you use wildcard search to union tables, the result is scoped to the pages that were scanned in the initial file you connected to. For example, suppose you have three files: A.pdf, B.pdf, and C.pdf. The first file you connect to is A and you limit the scan for tables to page 1. When you use wildcard search to union tables from files B and C, the additional tables included in the union can only come from page 1 of B and page 1 of C.

Tips for working with .pdf files

The following tips can help you work with your .pdf files in Tableau.

- **Use PDF File connector to identify just the tables in your .pdf file.**

  The primary goal of the PDF File connector is to find and identify *tables* in your .pdf file. Therefore, it ignores any other information in the file that does not appear to be part of a table, including titles, captions, and footnotes. If related data is stored in one of these areas, such as in the table title, you can use Tableau to first export the .pdf file data into a .csv file, manually add the data that was stored in the table title, and then connect to the .csv file instead. For more information, see **Export your data to .csv file** on page 3035.

- **Use standard tables.**

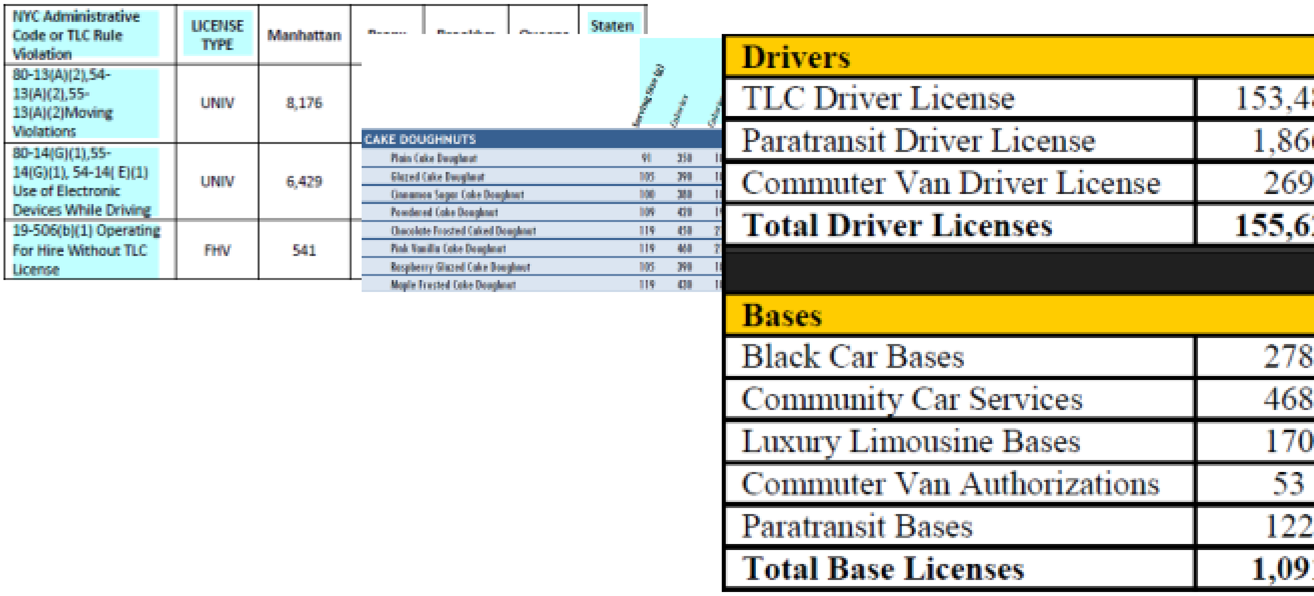
  In general, Tableau works best with standard tables that use a tabular format.

  Ideally, the tables in your .pdf file have column headers on a single line and have rows values on a single line as demonstrated in the example below.

CITY	RANK	TOTAL COMMUTE BY BIKE	TOTAL POPULATION
DAVIS, CA	1	23.2%	66,733
BERKELEY, CA	2	9.7%	118,851
BOULDER, CO	3	8.9%	105,101
SOMERVILLE, MA	4	7.4%	78,903
CAMBRIDGE, MA	4	7.4%	109,699
PALO ALTO, CA	5	7.3%	66,968
PORTLAND, OR	6	7.2%	619,445
EUGENE, OR	7	6.8%	160,552

  Colors and shading used in or around the tables can affect how the tables are identified.

  Tables that have unique formatting might require some cleanup or manual editing outside of Tableau. Unique formatting can include hierarchical headers, header names that span multiple lines, row values that span multiple lines, angle headers, and stacked tables as demonstrated in the examples shown below.



Note: Tableau does not support connections to .pdf files generated by scanning (optical character recognition) software.

- **Validate the data.**

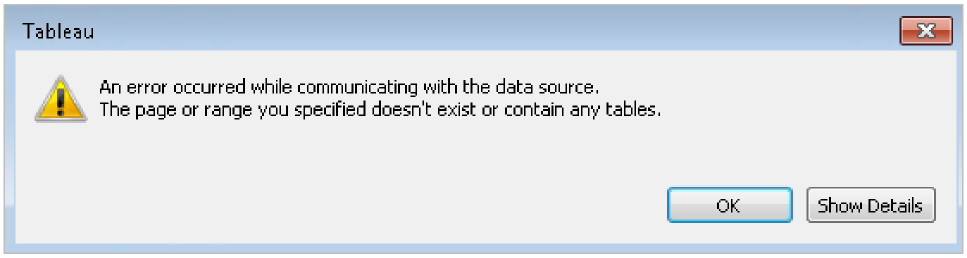
  Make sure that you validate the data in the tables that Tableau identifies in your .pdf file. You can validate the data by using either the data grid or if you used the Data Interpreter, the results workbook.

- **Avoid tables that span across pages.**

  If your .pdf file contains a table that spans across pages, Tableau interprets that table as multiple tables. To resolve this issue, use a union to combine the tables. For more information, see **Union Your Data** on page 942.

- **Rename .pdf files whose file names contain unicode characters.**

  After connecting to a .pdf file that contains unicode characters in its file name, you might see the following error.



To resolve this issue, rename the file using non-unicode characters, and connect to your .pdf file again.

- **Do not use password protected .pdf files.**

  After connecting to and scanning a .pdf file for tables, you might see the following error.

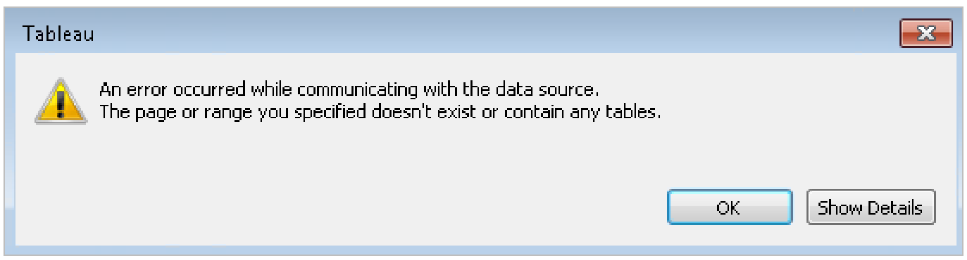
  

  Tableau shows this error when your .pdf file is password protected and unable to access its contents. Tableau is unable to support connections to password protected .pdf files.

- **Alias values that are interpreted differently or incorrectly.**

  In the data grid you might notice that some values are interpreted differently from the .pdf file. You can correct this interpretation by using aliases to rename specific values within a field.

  For example, suppose you see the following table after connecting to your .pdf file. Some state abbreviations are being interpreted in lowercase form, which are highlighted in blue.

City	Rank
College Station, tx	20
Tempe, az	20
Oakland, ca	20
Detroit, MI	1
Pittsburgh, PA	2
Cincinnati, OH	3
Portland, or	4

You can resolve this issue by using aliases to change the lowercase abbreviations to uppercase abbreviations. To do this, click the drop-down arrow next to the column name and select **Aliases**.

- **Resolve column headers that are interpreted as table values.**

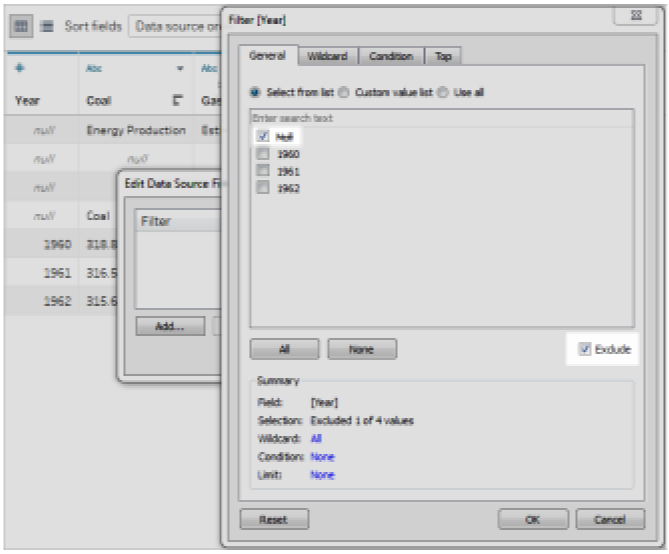
  In the data grid you might also notice that some column headers in your .pdf file are interpreted as table values instead. This can occur if your .pdf file contains tables with unique formatting or hierarchical headers. In this scenario, try the Data Interpreter first. If Data Interpreter doesn't resolve the issue, consider manually renaming the columns to their appropriate names and filtering header names that are being treated as values by using data source filters.

  For example, suppose you see the following table after connecting to your .pdf file. The table headers from the .pdf file are being interpreted as table values, which are highlighted in blue.

Abc F1	Abc F2	Abc F3	Abc F4
Table Pt 2.	Energy Production	Estimates	in Trillion BTu
null	*null*	*null*	*null*
Year	*null*	*null*	*null*
null	Coal	Gas	Oil
1960	318.8	.1	42.5
1961	316.5	.1	40.2
1962	315.6	.2	43.3

One way you can resolve a header issue like this is to follow steps similar to the following:

1. Double-click the column name, and then rename F1 to Year. Repeat this step for F2 through F4 for Coal, Gas, and Oil.
2. Click the data type icon for the Year column and change it to a number data type. This causes the non-numerical values in this column to convert to null values.
3. In the upper-right corner of the data source page, click **Add**, click the **Add** button, and then select the Year field.
4. In the Filter dialog box, select both the **Null** and **Exclude** check boxes.



The rows in the Year column that contain null values are removed from the data grid, which affect the rows from the other columns in the table.

# Year	Abc Coal	Abc Gas	Abc Oil
1960	318.8	.1	42.5
1961	316.5	.1	40.2
1962	315.6	.2	43.3

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Spatial File

This topic describes how to connect Tableau to Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, TopoJSON files, GeoJSON files, and Esri File Geodatabases.

Note: Connecting to spatial data is supported in Tableau Desktop version 10.2 and later.

Before you connect

Before you can connect to spatial files, make sure to include all of the following files in the same directory:

- **For Esri shapefiles**: The folder must contain `.shp`, `.shx`, `.dbf`, and `.prj` files as well as `.zip` files of the Esri shapefile.

  For Esri File Geodatabases: The folder must contain the File Geodatabase's `.gdb` or the `.zip` of the File Geodatabases’s `.gdb.`

- **For MapInfo tables** (Tableau Desktop only): The folder must contain `.TAB`, `.DAT`, `.MAP`, and `.ID` or `.MID` and `.MIF` files.
- **For KML files**: The folder must contain the `.kml` file. (No other files are required.)
- **For GeoJSON files**: The folder must contain the `.geojson` file.(No other files are required.)
- **For TopoJSON files**: The folder must contain the `.json` or `.topojson` file. (No other files are required.)

Note: Flat files will pull in the contents of the entire folder. For performance reasons, remove unneeded files and reduce the amount of data in the files.

You can connect only to point geometries, linear geometries, and polygons in current versions of Tableau. You cannot connect to mixed geometry types.

If your data does not display diacritics (accent marks on characters) properly, check to make sure the file is **UTF-8** encoded.

Make the connection and set up the data source

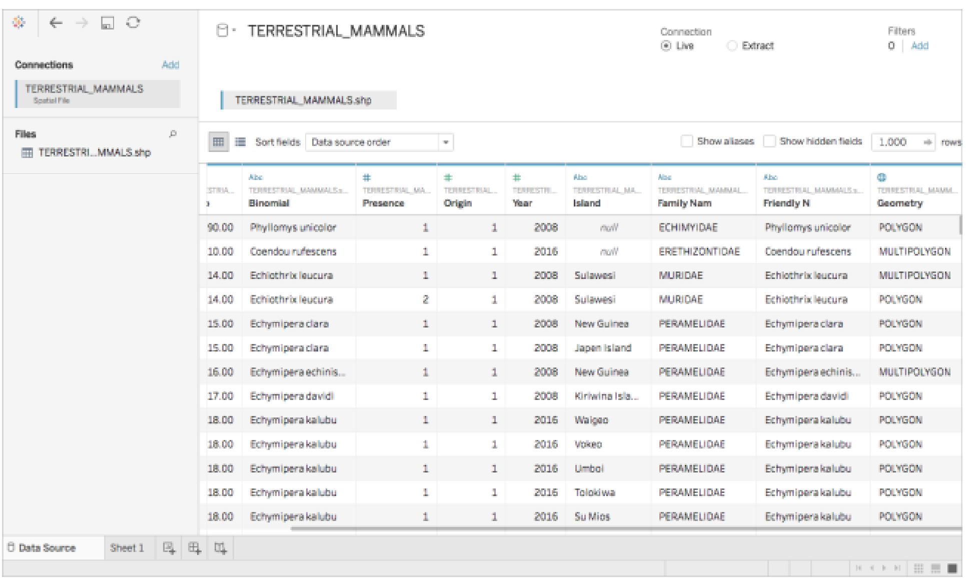
In Tableau Desktop: click the New Data Source icon and select **Spatial file**.
In Tableau Online or Tableau Server (Creator role): Select **Create** > **Workbook**. Select the **Files** tab.

Then do the following:

1. Navigate to the folder that contains your spatial data and select the spatial file you want to connect to.
2. Select **Open**.

Spatial file data source example

Here is an example of a spatial file data source using Tableau Desktop on a Mac computer:



Work with the Geometry column

There are many tasks that you can optionally perform on your data before you start your analysis, such as hiding or renaming fields. Note, however, the following restrictions apply when working with the Geometry column:

- You can't hide the Geometry column.
- You can't split the Geometry column.
- On the data source page, you can't create a calculated field using the Geometry column.

About .ttde and .hhyper files in Tableau Desktop

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

Create Tableau Maps from Spatial Files on page 1855

Statistical File

This article describes how to connect Tableau to a statistical file and set up the data source.

Tableau connects to SAS (*.sas7bdat), SPSS (*.sav), and R (*.rdata, *.rda) data files.

> **Note:** As of version 2020.1, Tableau no longer supports statistical files compressed with SASYZCR2. Use a different compression scheme to make the file readable for Tableau.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Statistical File**, select the file that you want to connect to, and then select **Open**.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out

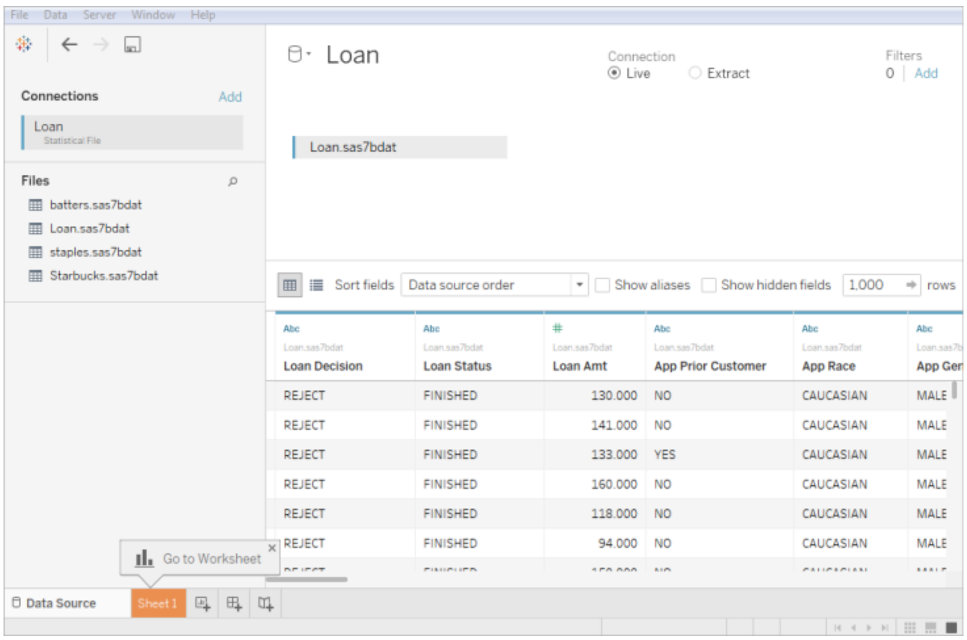
which data source to connect to.

b. Select the sheet tab to start your analysis.

For information about connecting to more than one table, see **Join Your Data** on page 883.

Statistical file data source example

Here is an example of a statistical file data source using Tableau Desktop on a Windows computer:



Note the following:

- The Statistical File connector does not support value labels.
- The Statistical File connector supports only one table per statistical file.

  If an error message appears, "An error occurred while communicating with the data source," make sure that your statistical file contains only one object, and that the object is a data frame or a matrix. R files may contain hidden objects in what appears to be a clean workspace. To check for hidden objects, run `ls(environment(), all.names=TRUE)` from RStudio.
- You can change the character encoding for a statistical file. For more information, see Change the character encoding for statistical files.

R data file object and format support

R data files should contain only one of the following types of objects:

- Two-dimensional matrices
- Vectors
- Factors
- Data frames

Note that if the R data file contains more than one object, Tableau will connect to the first one.

Tableau supports R data files that use binary format.

ASCII format files are not supported in Tableau Desktop. If you try to use an .rdata or .rda file with ASCII format, you'll see an error saying, "Unable to read from file the table '' ' [TableauTemp].[*filename*.rdata]' does not exist". In this case, you can use RStudio to save the file to a binary format, as follows:

1. Download RStudio if you don't already have it.
2. Open your R data file in RStudio. You should see your data object in the global environment window on the upper right.
3. Save the object to a new file with the "save()" function as follows:

   ```
   save(<objectName>,file="<filename>",ascii=FALSE)
   ```

   where: <objectName> is your data object, <filename> is the name of your new file, and "FALSE" is case-sensitive

   Tip: To quickly see documentation for this function, type "?save" at the command prompt.

Compressed files are not supported in Tableau Desktop. You must decompress files first with another program, such as Gzip or WinZip, before you connect with Tableau.

Change the character encoding for a statistical file

Tableau reads the data in a statistical file (for example, an SAS or R file) based on the file's character encoding or on information in an input file. R files typically use the character encoding of the operating system, compared to SAS and SPSS files, which include the character encoding information in the file. Sometimes, you might need to specify a different encoding. For example, if a colleague sends you a statistical file with Greek character encoding, then you must specify a Greek character set to use the file with Tableau. If you need to use a different character set when reading from a statistical file, you can create a Tableau data source customization (TDC) file and specify the encoding to use.

Create the TDC file

A .tdc file is an XML file that applies to a single data source and contains vendor and driver name information of the data source provider. For the statistical file connector, the vendor and driver name is `stat-direct`.

To create a TDC file:

1. Open a plain text editor, such as Notepad.
2. Copy the information from the sample provided below, paste it into your text file, and then specify the `source-charset` value. (For a list of encodings, see **User-defined Encodings Supported by the Statistical File Connector** on the facing page.)
3. Save the file with a .tdc extension—for example, `r-statsfile.tdc`—to the My Tableau Repository\Datasources folder.

Sample TDC file sets the source-charset value

The following TDC file example sets the `source-charset` value to `shift-jis` for a statistical file data source.

```
<connection-customization class='stat-direct' enabled='true' ver-
sion='10.0'>
<vendor name='stat-direct' />
<driver name='stat-direct' />
  <customizations>
    <customization name='source-charset' value='shift-jis' />
  </customizations>
</connection-customization>
```

Important: Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

User-defined Encodings Supported by the Statistical File Connector

This topic lists character encodings supported by Tableau Statistical File connector. The list includes single-byte, multi-byte, and Unicode user-defined encodings, as well as single-byte and multi-byte encodings that cannot currently be mapped to the corresponding SAS encodings.

Single-byte user-defined encodings

ASCII

CSASCII

US-ASCII

US

ISO_646.IRV:1991

ISO646-US

646

ISO-IR-6

IBM367

CP367

ANSI_X3.4-1986

ANSI_X3.4-1968

ISO-8859-1

CSISOLATIN1

LATIN1

L1

ISO_8859-1:1987

ISO8859-1

ISO-IR-100

ISO-8859-1

IBM819

CP819

ISO-8859-15

LATIN-9

ISO_8859-15:1998

ISO_8859-15

ISO8859-15

ISO-IR-203

IBM850

CSPC850MULTILINGUAL

CP850

850

WINDOWS-1252

MS-ANSI

CP1252

ISO-8859-7

CSISOLATINGREEKISO_8859-7:1987

ISO_8859-7

ISO-IR-126

ISO-8859-7

GREEK8

GREEK

ELOT_928

ECMA-118

WINDOWS-1253

MS-GREEK

CP1253

ISO-8859-10

CSISOLATIN6

LATIN6

L6

ISO_8859-10:1992

ISO_8859-10

ISO8859-10

ISO-IR-157

WINDOWS-1257

WINBALTRIM

CP1257

ISO-8859-2

CSISOLATIN2

LATIN2

L2

ISO_8859-2:1987

ISO_8859-2

ISO8859-2

ISO-IR-101

IBM852

CSPCP852

CP852

852

WINDOWS-1250

MS-EE

CP1250

ISO-8859-5

CSISOLATINCYRILLIC

ISO_8859-5:1988

ISO_8859-5

ISO8859-5

ISO-IR-144

CYRILLIC

WINDOWS-1251

MS-CYRL

CP1251

CP866

CSIBM866

IBM866

866

TIS-620

TIS620.2533-1

TIS620.2533-0

TIS620.2529-1

TIS620-0

TIS620

ISO-IR-166

ISO-8859-11

CP874

CSISOLATIN5

LATIN5

L5

ISO_8859-9:1989

ISO_8859-9

ISO8859-9

ISO-8859-9

ISO-IR-148

CSIBM857

IBM857

CP857

857

WINDOWS-1254

MS-TURK

CP1254

CP1129

VPS

WINDOWS-1258

CP1258

ISO-8959-6

CSISOLATINARABIC

ISO_8859-6:1987

ISO_8859-6

ISO8859-6

ISO-IR-127

ECMA-114

ASMO-708

ARABIC

WINDOWS-1256

MS-ARAB

CP1256

ISO-8859-8

CSISOLATINHEBREW

ISO_8859-8:1988

ISO_8859-8

ISO8859-8

ISO-IR-138

HEBREW

IBM864

CSIBM864

CP864

WINDOWS-1255

MS-HEBR

CP1255

IBM862

CSPC862LATINHEBREW0x2E

CP862

862

Multi-byte user-defined encodings

CP936

WINDOWS-936

MS936

GBK

GB2312

CSISO58GB231280

ISO-IR-58

GB_2312-80

CHINESE

ISO-2022-CN

CP950

windows-950

ms-950

ms950

CSBIG5

CN-BIG5

BIGFIVE BIG5

BIG-FIVE

BIG-5

BIG5HKSCS

BIG5-HKSCS

EUC-TW

CSEUCTW

EUCTW

EUC-JP

CSEUCPKDFMTJAPANESE

EXTENDED_UNIX_CODE_PACKED_FORMAT_FOR_JAPANESE

EUCJP

EUC-JP

ISO-2022-JP

CSISO2022JP

ISO-2022-JP

CSSHIFTJIS

SJIS

SHIFT_JIS

SHIFT-JIS

MS_KANJI

CP932

EUC-KR

CSEUCKR

EUCKR

EUC-KR

UHC

CP949

EUC-CN

CSGB2312

GB2312

EUCCN

CN-GB

Unicode user-defined encodings

TF-8

UCS-2

UCS-2BE

UCS-2LE

UCS-4

UCS-4BE

UTF-16

UTF-16BE

UTF-16LE

UTF-32

UTF-32LE

UTF-32BE

UTF-7

Single-byte encodings that can't be mapped to SAS encodings

MACROMAN

CSMACINTOSH

MACINTOSH

MAC

ISO-8859-14

LATIN8

L8

ISO_8859-14:1998

ISO_8859-14

ISO8859-14

ISO-IR-199

ISO-CELTIC

MACGREEK

MACICELAND

ISO-8859-3

CSISOLATIN3

LATIN3

L3

ISO_8859-3:1988

ISO_8859-3

ISO8859-3

ISO-IR-109

ISO-8959-4

CSISOLATIN4

LATIN4

L4

ISO_8859-4:1988

ISO_8859-4

ISO8859-4

ISO-IR-110

ISO-8959-13

LATIN7

L7

ISO_8859-13

ISO8859-13

ISO-IR-179

ISO-8859-13

MACCENTRALEUROPE

MACCROATIAN

IBM855

CSIBM855

CP855

855

KOI8-R

CSKOI8R

MACCYRILLIC

KOI8-U

CSKOI8R

MACUKRAINIAN

ISO-8859-16

LATIN10

L10

ISO_8859-16:2001

ISO_8859-16

ISO8859-16

ISO-IR-226

MACROMANIAN

ARMSCII-8

GEORGIAN-ACADEMY

MACTURKISH

TCVN

VISCII

CSVISCII

VISCII1.1-1

MACARABIC

MACHEBREW

WINDOWS-874

Multi-byte encodings that can't be mapped to SAS encodings

GB18030

HZ

HZ-GB-2312

CSISO2022JP

ISO-2022-JP

JOHAB

JOHAB

CP1361

ISO-2022-KR

CSISO2022KR

ISO-2022-KR

ISO-2022-JP

CSISO2022JP

ISO-2022-JP-1

ISO-2022-JP-2

CSISO2022JP2

ISO-2022-CN

CSISO2022CN

ISO-2022-CN-EXT

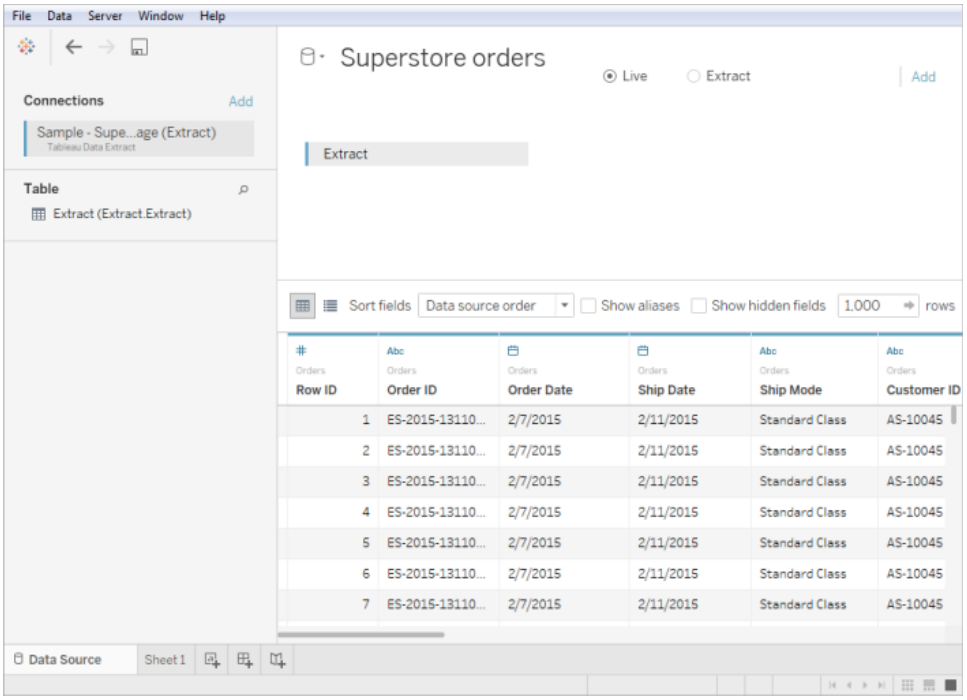
Other Files

This topic describes how to connect Tableau to supported file types, including Tableau Data Extract files and Tableau Workbooks.

1. Start Tableau and under **Connect**, select **Other Files**.
2. In the Open dialog box, navigate to and select a file.
3. Select **Open**.
4. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
5. Select the sheet tab to start your analysis.

Tableau Data Extract data source example

Here is an example of a Tableau Data Extract file data source using Tableau Desktop on a Windows computer:



See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Connect to Data by Using Tableau Server or Tableau Online

You can connect to data from either Tableau Desktop or a web browser by using Tableau Server or Tableau Online. The process differs slightly for each.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to connecting to published data sources from Tableau Server or Tableau Online, you can connect to databases and tables from the **Search for Data** results on Tableau Desktop. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

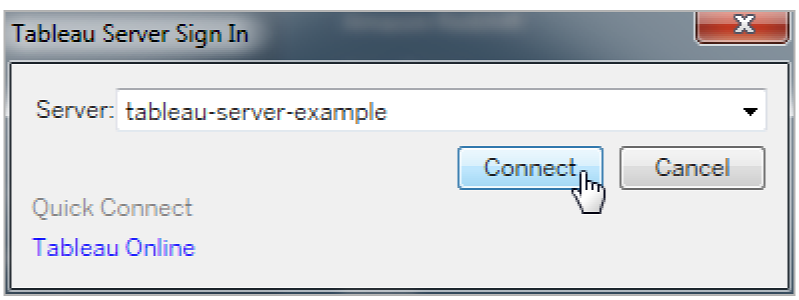
Tableau Prep Builder can connect to published data sources and you can use them as an input data source for your flow starting in version 2019.3.1. If you have the Data Management with Tableau Catalog enabled, starting in version 2020.2.2, you can also connect to databases and tables from the **Search for Data** results, just like Tableau Desktop. For more information, see Connecting to published data sources in the Tableau Prep Builder help.

Connect from Tableau Desktop

Note: Starting 2019.3, on the **Connect** pane, **Tableau Server** has moved to the top of the pane under **Search for Data**. Select this option to connect to Tableau Server or Tableau Online from Tableau Desktop.

1. Start Tableau Desktop and on the **Connect** pane, under **Search for Data**, select **Tableau Server**.
2. To connect to Tableau Server, enter the name of the server and then select **Connect**.

   To connect to Tableau Online, select **Tableau Online** under **Quick Connect**.

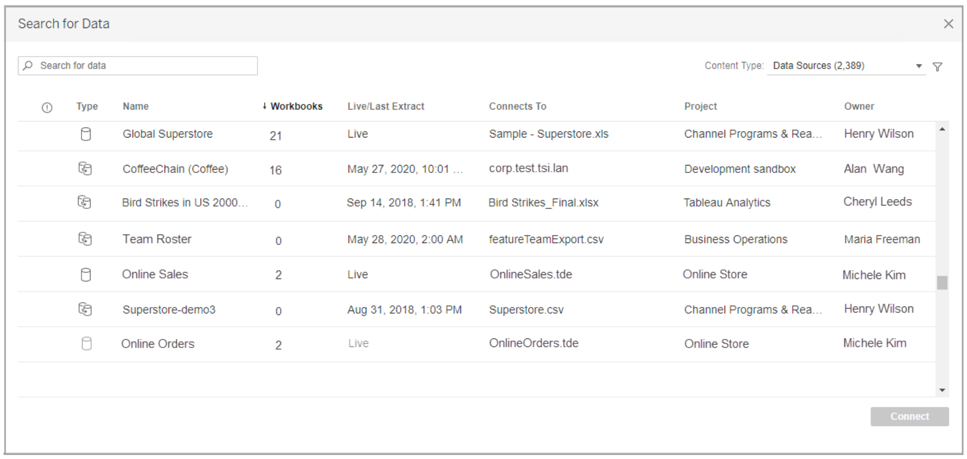
   

   Tip: If you don't sign out, Tableau Desktop saves your server connection, so you can skip step 3. You can also skip it if Kerberos is enabled on Tableau Server and your computer has valid credentials. For more information, see **Automatically keep Tableau Desktop connected to Tableau Server or Online** on page 365.

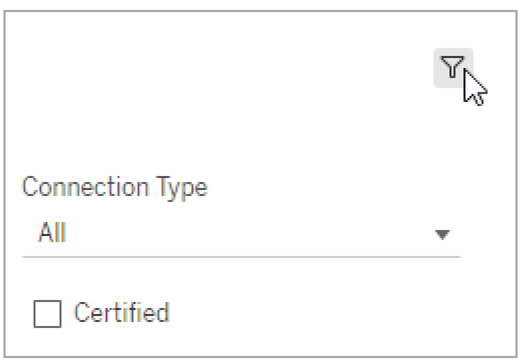
3. To sign in:
   - For Tableau Server, enter your user name and password.
   - For Tableau Online, enter your email address and password.

4. Select a data source from the list of published data sources. (Data that you don't have Connect permissions for is grayed out.)

   > **Note:** If you select a cube (multidimensional) data source, the Create Local Copy dialog box appears, and you must create a local copy of the data before you can start your analysis.

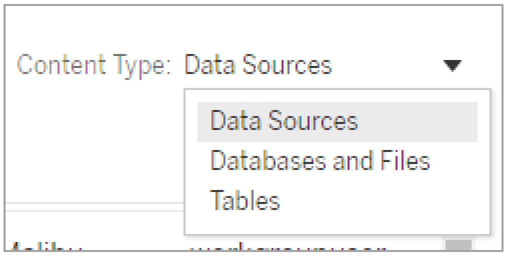


Search for data by using the Search box. Or you can sort the list by selecting the column headers. Column headers that support sorting are underlined when you hover over the header. You can also select the filter icon to filter data by Connection Type and by Certified.

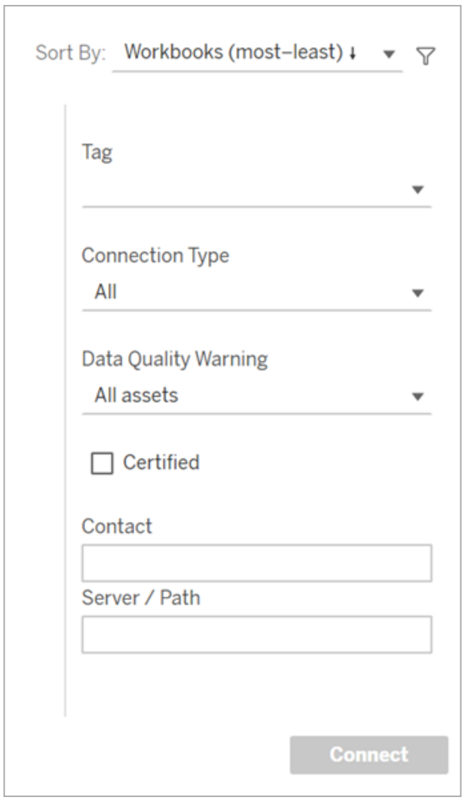


When Tableau Catalog is enabled,

- You can select from **Data Sources**, **Databases and Files**, or **Tables** to search for data.



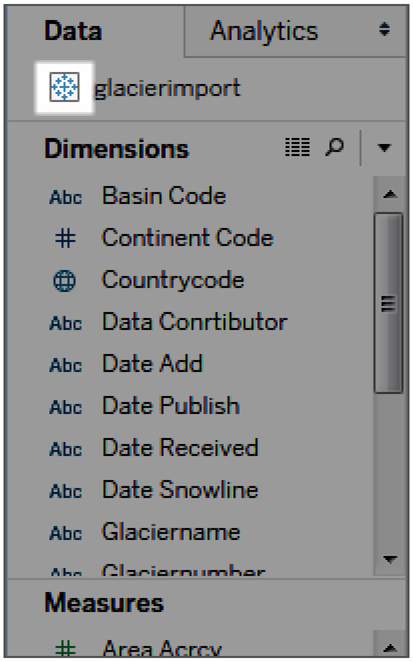
- You can filter assets by several options including Tag, Data Quality Warning, Certified, and more.



> **Note:** From the **Search for Data** results, you can make a connection with a subset of the data connectors that Tableau Desktop supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Server or Tableau Online. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

5. After you select the data you want to connect to, click **Connect**.
6. Select the sheet tab to start your analysis.

Data sources published to Tableau Server or Tableau Online are shown in the **Data** pane with a Tableau icon.



You might want to download a local copy of the data source so that you can, for example, work offline or make changes to a data source without modifying the original. To download a local copy, on the **Data** menu, select the data source, and then select **Create Local Copy**. A duplicate of the data source is added to the **Data** pane.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Connect to published data sources while web authoring

Tableau Desktop users who create and customize data connections for Tableau use can publish their data sources to Tableau Server or Tableau Online. Publishing enables sharing data among colleagues, including those who don't use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

If you have permission to edit workbooks in a browser, follow these steps to connect to published data sources when you're signed in to Tableau Server or Tableau Online. For more information see Edit Tableau Views on the Web and Using Tableau on the Web.

Note: To connect to data, you need permission to create and edit views.

Add more data to an existing view

1. On a Tableau Server or Tableau Online site, select a view to edit.
2. In editing mode, click the **New Data Source** icon .
3. In the **Connect to Data** page, the options you can see and select from depend on your permissions:
   - **On this site**. Select a published data source from the list, and then select **Add data source**.

     When Tableau Catalog is enabled in your environment, you can select **Data Sources**, **Databases and Files**, or **Tables** from the **Content Type** drop-down list. Then select the data you want to connect to and click **Connect**.
   - **Files**. Drag and drop a file or upload one from your computer.
   - **Connectors**. Connect to your data using one of the connectors listed.

Create a workbook with a published data source

You can also connect to a published data source when creating a new workbook:

1. On a Tableau Server or Tableau Online site, navigate to the **Explore** page, and select **All Data Sources** from the drop-down list.
2. In the list, select the check box next to the data you want to use.
3. At the top of the page, from the **Actions** drop-down menu, select **New Workbook**.

For information about connecting to data when you're authoring on the web, see Creators: Connect to Data on the Web.

Actian Matrix

This article describes how to connect Tableau to an Actian Matrix (formerly ParAccel) database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Actian Matrix**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.
   b. Enter the user name and password, and then select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Schema** drop-down list, select a schema.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Actian Vector

This article describes how to connect Tableau to an Actian Vector database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Virtual node name for the database you want to connect to
- Database name
- Authentication method: Authentication defined in the virtual node, or user name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Actian Vector**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Enter the name of the virtual node for the database and the name of the database you want to connect to.

    b. Specify whether to use authentication defined in the virtual node, or a specific user name and password.

c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

d. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.

c. Under **Table**, select a table or use the text box to search for a table by name.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Alibaba AnalyticDB for MySQL

This topic describes how to connect Tableau to an Alibaba AnalyticDB for MySQL database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name or IP address of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Alibaba AnalyticDB for MySQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the **Server** name or IP address.
   b. Enter the **Database** name.
   c. Enter the **Username** and **Password**.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   e. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table or use the text box to search for a table by name.
   c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Alibaba Data Lake Analytics

This topic describes how to connect Tableau to Alibaba Data Lake Analytics and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name or IP address for the server that hosts the database you want to connect to
- Port number
- Database name
- Username and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Alibaba Data Lake Analytics**. For a complete list of data connections, select **More** under **To a Server**. Then do the

following:

a. Enter the name or IP address for the **Server**.
b. Enter the **Port** number.
c. Enter the **Database** name.
d. Enter the **Username** and **Password**.
e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
f. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table or search for a table by name.
   c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Alibaba MaxCompute

This topic describes how to connect Tableau to Alibaba MaxCompute and set up the data source.

Before you begin

Before you begin, gather this connection information:

- URL of the MaxCompute service
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Alibaba MaxCompute**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the URL for the **Server**.
   b. Enter the **Username** and **Password**.
   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   d. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Under **Database**, select a database from the drop-down list or enter a database name.
c. Under **Table**, select a table or search for a table by name.
d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Amazon Athena

This article describes how to connect Tableau to Amazon Athena data and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to. The format should be as follows: athena.[region].amazonaws.com. For example: `athena.us-east-1.amazonaws.com`
- Name of the S3 staging directory. For example, `s3://aws-athena-query-results-123456785678-us-eastexample-2/`
- Amazon Web Services (AWS) access keys (access key ID and secret access key). For more information, see Access keys on the AWS website.
- (Optional) Initial SQL statement to run every time Tableau connects

You must have Java installed on the computer that runs Tableau Desktop. For more information, see Driver Download on the Tableau website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Amazon Athena**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server.
   b. Enter the name of the S3 staging directory.
   c. Enter your AWS access key ID in the **Username** field.
   d. Enter your AWS secret access key in the **Password** field.
   e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   f. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Under **Connections**, select the default data source.
   b. Select the catalog and database you want to work with.
   c. Drag one or more tables to your workspace.
   d. Select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Customize JDBC connections

Amazon Athena uses a JDBC connection, which you can customize using a properties file. For more information, see Customize JDBC Connections Using a Properties File in Tableau Community.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Connect to your S3 data with the Amazon Athena connector in Tableau 10.3 (and later) - Tableau blog post with links to Amazon Athena resources.
- Tableau Desktop Fails To Connect to Amazon Athena through Proxy With Error Code 37CE01A3 - Tableau Knowlege Base article with steps for resolving the issue.

Amazon Aurora for MySQL

This article describes how to connect Tableau to Amazon Aurora MySQL and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Amazon Aurora for MySQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the name of the server that hosts the database.

   b. Enter the user name and password.

   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

   d. Select **Sign In**.

   Select the **Require SSL** check box when connecting to an SSL server.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.

   c. Under **Table**, select a table or use the text box to search for a table by name.

   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Amazon EMR Hadoop Hive

This article describes how to connect Tableau to an Amazon EMR (Elastic MapReduce) Hadoop Hive database and set up the data source.

Note: Starting in version 2018.2, Tableau supports Amazon EMR Hadoop Hive only, not Impala. Amazon no longer provides Impala drivers.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Authentication method:
  - No Authentication
  - Kerberos
  - Username
  - Username and password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
  - Binary
  - SASL
  - HTTP
- Sign-in credentials depend on the authentication method you choose and can include

the following:

  - User name
  - Password
  - Realm
  - Host FQDN
  - Service Name
  - HTTP Path

- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** Make sure you use the latest available drivers. To get the latest drivers, see Amazon EMR Hadoop Hive on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Amazon EMR Hadoop Hive**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the name of the server that hosts the database and the port number to use.

   b. In the **Authentication** drop-down list, select the authentication method to use. The information you are prompted to provide depends on the authentication method you choose.

   c. If the **Transport** drop-down list is available, select the type of transport to use.

   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

e. Select **Sign In**.

Select the **Require SSL** option when connecting to an SSL server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see **Date Functions** on page 2170.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to **String** and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera's Impala and Hortonworks' Stringer project) are designed to address this limitation.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Amazon Redshift

This article describes how to connect Tableau to an Amazon Redshift database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

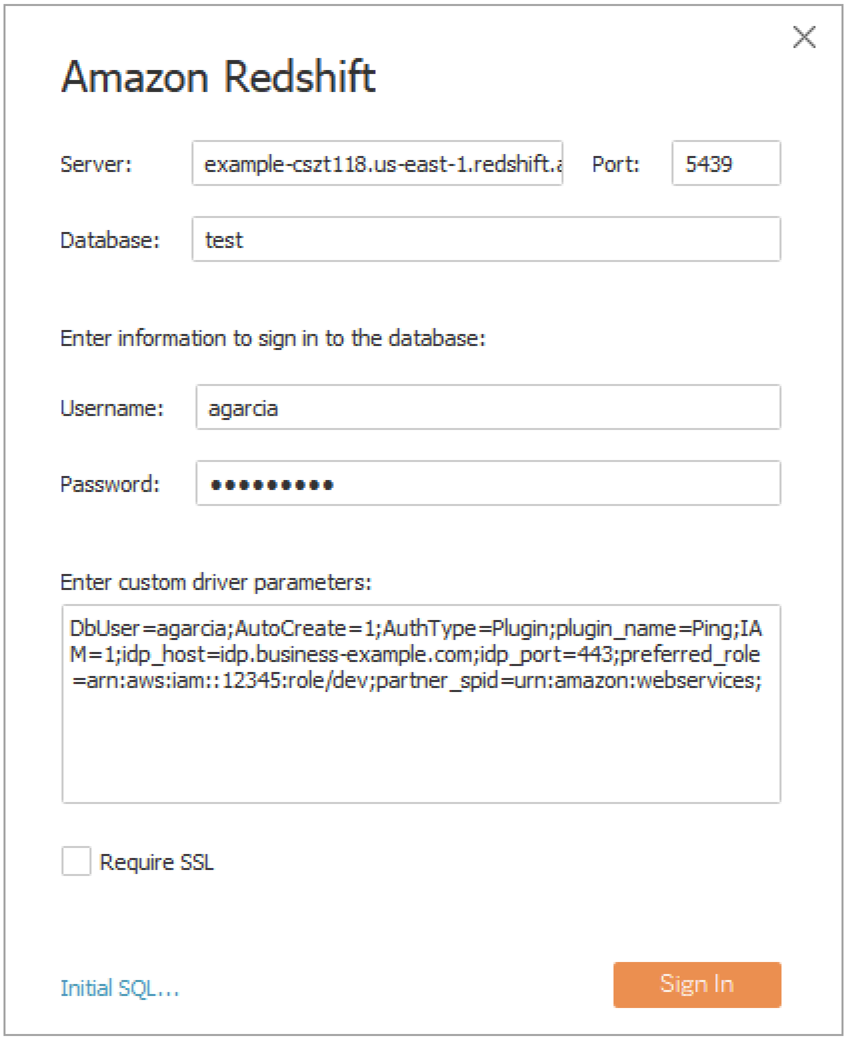
1. Start Tableau and under **Connect**, select **Amazon Redshift**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database you want to connect to.
   b. Enter the user name and password.
   c. (Optional) **Enter custom driver parameters** to modify the connection.
   d. Select the **Require SSL** check box when connecting to an SSL server.
   e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   f. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Customize the connection using driver parameters

There are times when you might want to modify the connection made with the Amazon Redshift connector. Starting in Tableau 2019.2.1, you can do this by appending driver parameters to the connection string in the **Enter custom driver parameters** text box:



Note: You can’t change the driver parameters that Tableau generates; you can only append parameters.

In this example, agarcia enters parameters that enable federated authentication using IDP:

```
DbUser=agarcia;AutoCreate=1;AuthType=Plugin;plugin_name-
e=Ping;IAM=1;idp_host=idp.business-example.com;idp_port-
t=443;preferred_role=arn:aws:iam::12345:role/dev;partner_
spid=urn:amazon:webservices;
```

In this example, you limit the number of rows fetched to 1000:

```
Fetch=1000;
```

For details about driver parameters, see ODBC Driver Configuration Options on the Amazon AWS website.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Optimizing Your Amazon Redshift and Tableau Software Deployment for Better Performance - Tableau whitepaper (registration or sign in required)
- Explore Big Data Analytics with Amazon Redshift - Tableau on-demand webinar (registration or sign in required)

Anaplan

This article describes how to connect Tableau to Anaplan data and set up the data source.

> **Note:** For the most up-do-date information on this connector, see the Anaplan help topic for the current version of Tableau.

Before you begin

Before you begin, gather this information for connecting:

- An email address and password for your Anaplan account.
- The Anaplan workspace, model, and any exports (in CSV format) that you want to use. An Anaplan export action must be created in Anaplan before you can use the export in Tableau. For more information about creating a module and export action in Anaplan, see Tableau Connector for Anaplan on the Anaplan Anapedia website.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Anaplan**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Do one of the following:

      - Select **Credentials**, then enter your email address and password.

        For single sign-on (SSO) support, ask your Anaplan Administrator to set up single sign-on. For more information, see Single Sign-on (SSO) on the Anaplan Anapedia website. Note that if SSO is enabled, you must be an exception user.
      - Select **Certificate**, then upload files for the **Certificate** and **Private Key**.

        Note: The uploaded certificate and private key files must be in PEM format.

        An SSL certificate file typically contains text similar to this:

        ```
        -----BEGIN CERTIFICATE-----
         A long base64 string
        -----END CERTIFICATE-----
        ```

        A private key file typically contains text similar to this:

        ```
        -----BEGIN PRIVATE KEY-----
         A long base64 string
        -----END PRIVATE KEY-----
        ```

   b. Select a workspace, model, and one or more exports.

> **Note:** The export data must be in CSV format.

c. Select **Connect**.

   If Tableau can't make the connection, verify that your credentials or certificate and private key files are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Anaplan administrator for more assistance.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under **Table**, select a table and drag it to the top of the canvas. If you have only one table, it appears on the canvas.

   c. Select the sheet tab to go to the worksheet.

   After you select the sheet tab, Tableau imports the data by creating an extract.

   Creating extracts may take some time depending on the amount of data that is included.

Date range selections can impact performance

It's tempting to gather as much data as possible when you do an analysis, however, retrieving records from Anaplan can be time-consuming. Tableau doesn't know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance.

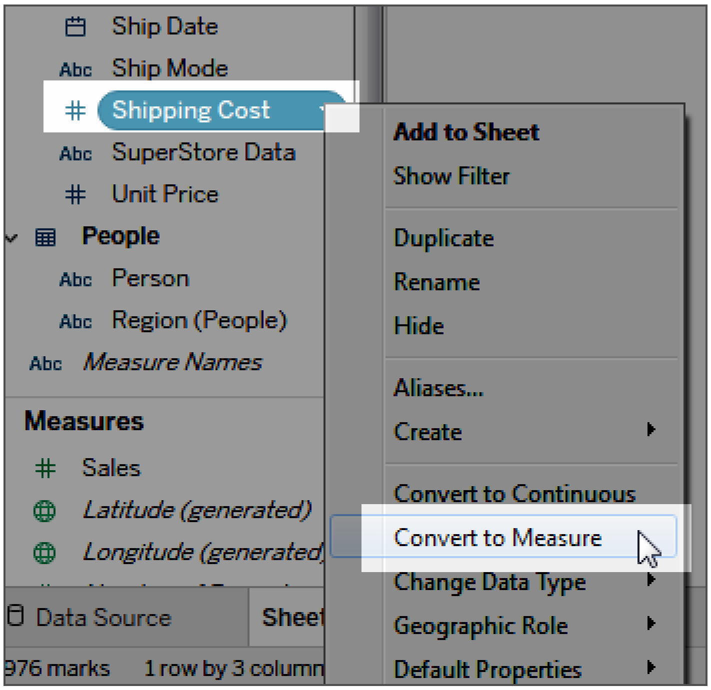
To give you a rough idea of how much time it might take to retrieve data from Anaplan, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

Number of records	Time to retrieve
100,000	1.7 minutes
250,000	4 minutes
500,000	8.5 minutes

Groom your data

All data from Anaplan is returned as type **String**. To groom your data, convert the fields to the proper data type, for example, **Number** or **Date**. You can also change **Geographic Role**, and you can convert a dimension to a measure. For information about changing the data type, see **Data Types** on page 325.

To convert a dimension to a measure, in the **Data** pane on the worksheet, select the drop-down arrow next to the field name and select **Convert to Measure**.



For information about dimensions and measures, see **Dimensions and Measures, Blue and Green** on page 314.

Refresh your data

Tableau supports only extract connections for Anaplan, not live connections. You can update the data by refreshing the extract; incremental refreshes are not supported. For more information, see **Refresh Extracts** on page 1041. For information about refresh schedules, see **Schedule Extract Refreshes as You Publish a Workbook** on page 3086.

Note: When there is a scheduled refresh of the Tableau extract on Tableau Server or Tableau Online, an export action is called in Anaplan. The export action creates a blocking operation that locks the model in Anaplan.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Apache Drill

This article describes how to connect Tableau to Apache Drill data and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Type of connection:
  - Direct: server name
  - Zookeeper: quorum and cluster ID
- Authentication:
  - No authentication
  - User name and password

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Apache Drill**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

a. Select the **Connection** method:

- **Direct** - Enter the **Server** name.
- **Zookeeper** - Enter the **Quorum** and **Cluster ID**.

b. Select the **Authentication** method:

- **No Authentication**
- **Username and password** - Enter the user name and password.

c. Select **Sign In**.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Under **Schema**, select a schema from the drop-down list.

c. Under **Table**, select a table and drag it to the canvas.

For information about connecting to more than one table, see **Join Your Data** on page 883.

d. Select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Aster Database

This article describes how to connect Tableau to Aster Database data and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Aster Database**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   d. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Use initial SQL to improve performance

In Aster, initial SQL can be used to generate an output table, which can improve performance of subsequent database access. Initial SQL in Aster also supports SQL-MapReduce, a framework created by Teradata Aster to enable developers to write powerful and highly expressive SQL-MapReduce functions.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Azure Data Lake Storage Gen2

This article describes how to connect Azure Data Lake Storage Gen2 and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Account credentials for Azure Active Directory via OAuth
- URL for your Azure storage endpoint

> **Note:** This connector uses only the default tenant for your Azure account. To use Azure Data Lake Storage Gen2 resources with Tableau, they must be associated with the default tenant.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Azure Data Lake Storage Gen2**. For a complete list of data connections, select **More** under **To a Server**.
2. In the tab Tableau opens in your default browser, do the following:
   a. Sign in to Azure Data Lake Storage Gen2 using your Azure account. Choose one from the list, or choose **Use another account**to enter your email address or phone number.
   b. Select **Accept** so that Tableau can access your Azure Data Lake Storage Gen2 data.
   c. Close the browser window when notified to do so.
3. On the data source page, do the following:
   a. On the **Browse** tab, enter the Azure storage endpoint for your Azure account.

      > **Note:** The storage account must have “Data Lake Storage Gen2” enabled.

   b. Navigate to the file you want to use and select it.

   Or

   - On the **File URL** tab, enter URL for the file.

About Azure storage accounts

When you use Tableau with Azure Data Lake Storage Gen2, the storage account you use must be set up with at least one of the following:

- Read permission explicitly granted for the files you want to access (Read / Execute for folders)
- Storage Blob Data Reader or Storage Blob Data Owner role assigned to the account

For more information about setting up a storage account, see Create a storage account to use with Azure Data Lake Storage Gen2 on the Microsoft site.

When the storage account is configured, it must enable these CORS options for the Blob service to allow proper access from Tableau:

- **Allowed origins** set to "*"
- **Allowed methods** set to "GET, OPTIONS"
- **Allowed headers** set to "*"

An Azure storage account can have multiple endpoints associated with various services. In Tableau, you'll connect to the storage endpoint that is enabled for "Data Lake Storage Gen2".

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Azure SQL Database

This article describes how to connect Tableau to Azure SQL Database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Fully qualified server name that hosts the database you want to connect to; for example, myserver-20171113.database.windows.net
- (Optional) Name of the database
- Authentication method:

- Username and password
- Active Directory password
- Azure Active Directory via OAuth (integrated)

> **Note:** To use this option, your system administrator must enable "user consent" in your organization's Azure Active Directory tenant. For more information, see Microsoft documentation on configuring user consent.

- Sign in credentials depend on the authentication method you choose and can include the following:
  - Username and password
  - Active Directory username and password
  - Azure Active Directory instance URL
- Are you connecting to an SSL server?
- Do you want to set the database isolation level to read uncommitted data?
- (Optional) Initial SQL statement to run every time Tableau connects

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Azure SQL Database**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the URL for the server that you want to connect to.
   b. (Optional) Enter a database name if you want to connect to a contained database.
   c. Specify how you want to sign in to the server — with a username and password, with an Active Directory username and password, or using an integrated Azure Active Directory sign-in via OAuth.
   d. Type your user name and password.
   Or, if you use Azure Active Directory via OAuth, type an Azure Active Directory instance URL. See documentation from Microsoft on endpoints for more details.

> **Note:** If your organization uses more than one Azure Active Directory tenant, the Azure Active Directory instance URL must match the tenant where the database is configured. You can find the tenant ID in your Azure portal, or from your Azure administrator.

e. Select the **Require SSL** check box when connecting to an SSL server.
f. Specify whether to **Read uncommitted data**. This option lets Tableau run at the read-uncommitted isolation level. Long queries from Tableau, including extract refreshes, can lock the database and delay transactions. Select this option to allow queries to read rows that have been modified by other transactions even when they have not been committed yet. When this option is cleared, Tableau uses the default isolation level specified by the database.
g. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
h. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select a database.

   c. Under **Table**, select a table or use the search field to search for a table by name.

      You can also specify a stored procedure in the database. For more information about stored procedures, including a list of constraints specific to SQL Server databases, see **Use a Stored Procedure** on page 972.

   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: Tableau Desktop does not support the Microsoft SQL Server TIME data type. Fields of this data type are not imported and do not appear in Tableau Desktop. If included in stored procedures, TIME data type fields will not appear in Tableau Desktop. For more information, see **Use a Stored Procedure** on page 972.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

> **Note:** Azure Active Directory via OAuth (integrated) is not supported on a Mac

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Azure SQL Synapse Analytics

This article describes how to connect Tableau to Azure Synapse Analytics (formerly Azure SQL Data Warehouse) and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Fully qualified server name that hosts the database you want to connect to; for example, myserver-20171113.database.windows.net
- (Optional) Name of the database
- Authentication method:
  - Username and password
  - Active Directory password
  - Azure Active Directory via OAuth (integrated)

    > **Note:** To use this option, your system administrator must enable “user consent” in your organization's Azure Active Directory tenant. For more information, see Microsoft documentation on configuring user consent.

- Sign in credentials depend on the authentication method you choose and can include the following:
  - Username and password
  - Active Directory username and password
  - Azure Active Directory instance URL
- Are you connecting to an SSL server?
- Do you want to set the database isolation level to read uncommitted data?
- (Optional) Initial SQL statement to run every time Tableau connects

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Azure Synapse Analytics**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the URL for the server that you want to connect to.
   b. (Optional) Enter a database name if you want to connect to a contained database.
   c. Specify how you want to sign in to the server — with a username and password, with an Active Directory username and password, or using an integrated Azure Active Directory sign-in via OAuth.
   d. Type your user name and password.
   Or, if you use Azure Active Directory via OAuth, type an Azure Active Directory instance URL. See documentation from Microsoft on endpoints for more details.

   > **Note:** If your organization uses more than one Azure Active Directory tenant, the Azure Active Directory instance URL must match the tenant where the database is configured. You can find the tenant ID in your Azure portal, or from your Azure administrator.

   e. Select the **Require SSL** check box when connecting to an SSL server.
   f. Specify whether to **Read uncommitted data**. This option lets Tableau run at the read-uncommitted isolation level. Long queries from Tableau, including extract refreshes, can lock the database and delay transactions. Select this option to allow queries to read rows that have been modified by other transactions even when they have not been committed yet. When this option is cleared, Tableau uses the default isolation level specified by the database.
   g. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in

to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

h. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select a database.

   c. Under **Table**, select a table or use the search field to search for a table by name.

      You can also specify a stored procedure in the database. For more information about stored procedures, including a list of constraints specific to SQL Server databases, see **Use a Stored Procedure** on page 972.

   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: Tableau Desktop does not support the Microsoft SQL Server TIME data type. Fields of this data type are not imported and do not appear in Tableau Desktop. If included in stored procedures, TIME data type fields will not appear in Tableau Desktop. For more information, see **Use a Stored Procedure** on page 972.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Note: Azure Active Directory via OAuth isn't supported on a Mac

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Box

This article describes how to connect Tableau to Box data and set up the data source.

Before you begin

Before you begin, get the email address and password for your Box account.

Make the connection and set up the data source

1. On the start page, under **Connect**, click **Box**. For a complete list of connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

    a. Enter your email and password and then click **Authorize**.

    b. Click **Grant access to Box**.

    c. Close the browser window when notified to do so.

    d. Search for or select the file to connect to, and then click **Connect**.

    Note: This connector might not support all file types. Unsupported file types are grayed out.

2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

    b. Click the sheet tab to start your analysis.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

Adding Tableau to your Box account

You can officially enable Tableau as an approved app for your Box account. For information on enabling this feature, please see Adding Apps to Your Box Account on the Box website.

Limitations and known issues

The Box connector limits connecting to a single Excel, JSON, or text-based file.

Known issues

- If you click **Grant Access to Box** repeatedly during the authentication phase an error results.
- Long file names cause the column to be pushed out of alignment.
- The file window does not resize.
- When using web authoring or publishing to the web, you cannot use multiple Box accounts in the same workbook. You can have multiple Box account connections in Desktop.
- In Internet Explorer 11 and Edge, you cannot access a server using an unsecured connection (http). Use a secure connection (https) or switch to another browser.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Cloudera Hadoop

This article describes how to connect Tableau to a Cloudera Hadoop database and set up the data source.

> **Note:** For new connections to Impala databases, use the Impala connector rather than this one. (You can continue using this connector for existing connections.)

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Type of database: Hive Server 2 or Impala
- Authentication method:
    - No Authentication
    - Kerberos

      > **Note:** Due to Kerberos Domain Controller (KDC) restrictions, connection with MIT Kerberos is not supported.

    - User Name
    - User Name and Password
    - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
    - Binary
    - SASL
    - HTTP
- Sign-in credentials depend on the authentication method you choose and can include the following:
    - User name
    - Password
    - Realm
    - Host FQDN

    - Service name
    - HTTP path
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** Make sure you use the latest available drivers. To get the latest drivers, see Cloudera Hadoop on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Cloudera Hadoop**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Enter the name of the server that hosts the database and the port number to use. If you are connecting using Cloudera Impala, you must use port 21050; this is the default port if you are using the 2.5.x driver (recommended).

    b. In the **Type** drop-down list, select the type of database to connect to. Depending on the version of Hadoop and the drivers you have installed, you can connect to one of the following:

        - **Hive Server 2**
        - **Impala**

    c. In the **Authentication** drop-down list, select the authentication method to use.

    d. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.

    e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

f. Select **Sign In**.

Select the **Require SSL** option when connecting to an SSL server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

   c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

   Note: This database type supports only equal (=) join operations.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use

DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see **Date Functions** on page 2170.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to **String** and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera's Impala and Hortonworks' Stringer project) are designed to address this limitation.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Databricks

This article describes how to connect Tableau to a Databricks database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- HTTP path to the data source

- Authentication method:
  - Azure Active Directory via OAuth

    Note: To use this option, your system administrator must enable “user consent” in your organization's Azure Active Directory tenant. For more information, see Microsoft documentation on configuring user consent.

  - Personal Access Token
  - Username / Password
- Sign in credentials that related to your chosen authentication method:
  - Azure AD endpoint
  - Personal Access Token password
  - Username and password
- In Databricks, find your cluster server hostname and HTTP path using the instructions in Construct the JDBC URL on the Databricks website.
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Note: If you are connecting with Azure Active Directory via OAuth, you will need version 2.6.15 or later for this driver.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Databricks**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the **Server Hostname**.

   b. Enter the **HTTP Path** to the data source.

   b. Select the **Authentication** method: **Azure Active Directory via OAuth**, **Personal Access Token**, or **Username / Password**.

> **Note**: You can use the **Username / Password** option only if single sign-on (SSO) is disabled on the server. Otherwise, use your personal access token.

c. Enter the sign in credentials for the authentication method you chose:
   - For Azure AD, type the URL for the **Azure AD Endpoint**.
   - For personal access token, type the corresponding **Password**. (See Personal Access Tokens on the Databricks website for information on access tokens.)
   - For username and password, type those in the fields provided.

d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

e. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under **Schema**, select a schema or use the text box to search for a schema by name.

   c. Under **Table**, select a table or use the text box to search for a table by name.

   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Databricks best practices and troubleshooting

For information about best practices and troubleshooting when using Tableau with Databricks clusters, see the Tableau topic on the Databricks website.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Datorama by Salesforce

Note: This is a partner-built connector, created and supported by Salesforce. See the Datorama web site for more information.

This article describes how to connect Tableau to a Datorama database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the environment and workplace ID that you want to connect to
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Datorama by Salesforce**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Choose the environment you want to work in.

   b. Enter the workplace ID.

   c. Enter the access token from Datorama. For more information, see **Get a token from Datorama** below below.

   d. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select a database.

   c. Under **Table**, select a table or use the search field to search for a table by name.

   d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Get a token from Datorama

You can get the access token you'll need in Tableau from Datorama. Access tokens are created by your Datorama administrator.

1. In Datorama, go to the “Tableau Data Connection” page.
2. Activate the connector, or copy the token for an activated connector. You’ll find this tile as the first tile on the connections list.

> **Note:** Only Administrators can create a token for the connector in Datorama. Power users can access the page and copy the token, but not make changes to it (refresh/revoke).

3. Select **Settings** and then select **Generate** to generate a token.
4. Choose the refresh schedule of the reports and save.
5. Select **Run Now**.

Copy this token to use with the Datorama by Salesforce connector in Tableau.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Denodo

This article describes how to connect Tableau to Denodo and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Name of the database
- Authentication method: Integrated Authentication, or Username and Password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Update 20170515 of Denodo 6.0 required

If you can’t connect Tableau to your Denodo data, you might need to ask your database administrator to apply update 20170515 to the Denodo 6.0 database. For more information about update 20170515 for Denodo 6.0, see the Denodo website. (Sign in required.)

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Denodo**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Enter the name of the server that hosts the database.

    b. Enter the name of the database.

    c. Select how you want to sign in to the server. Specify whether to use **Integrated Authentication** or **Username and Password**. If the server is password protected, and you are not in a Kerberos environment, you must enter the user name and password.

    > **Note:** If you're using a Mac, and it is not attached to the domain correctly, the Mac won't know that Kerberos is being used in the domain, and the **Authentication** drop-down list won't be available.

    Select the **Require SSL** check box when connecting to an SSL server.

    d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

    e. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table or use the text box to search for a table by name.
   c. Click the sheet tab to start your analysis.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Dremio Connector by Dremio

> **Note:** This is a partner-built connector, created and supported by Dremio. See the Dremio web site for more information.

This article describes how to connect Tableau to a Dremio data lake and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name or IP address for the server that hosts the database you want to connect to
- Port number
- Username and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Dremio** . For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name or IP address for the **Server**.
   b. Enter the **Port** number.
   c. Enter the **Username** and **Password**.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   e. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table or search for a table by name.
   c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Dropbox

This article describes how to connect Tableau to Dropbox data and set up the data source.

Note: "Team" folders are currently unavailable through the Dropbox connector.

Before you begin

Before you begin, get the email address and password for your Dropbox account.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Dropbox**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

   a. Enter your email and password and then select **Sign in**.

   b. Select **Allow** to let Tableau Desktop access your Dropbox files and folders.

   c. Close the browser window when notified to do so.

   d. Search for or select the file to connect to, and then select **Connect**.

   Note: This connector might not support all file types. Unsupported file types are grayed out.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Select the sheet tab to start your analysis.

Note: Tableau doesn't support pivot tables in Dropbox.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

Troubleshoot Dropbox issues

Connections to multiple accounts

When using web authoring or publishing to the web, you cannot use multiple Dropbox accounts in the same workbook. You can have multiple Dropbox account connections in Desktop.

Web authoring with Internet Explorer 11 and Edge

In Internet Explorer 11 and Edge, you cannot access a server using an unsecured connection (http). Use a secure connection (https) or switch to another browser.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Esri ArcGIS Server

This article describes how to connect Tableau to an Esri ArcGIS Server and set up the data source. The connector also supports connections via GeoService API.

This connector works with ArcGIS Server (AGS) v10.3.1 and higher.

> **Note:** This connector uses Tableau's Web Data Connector (WDC), which requires an internet connection and the ability to reach https://connectors.tableau.com/esri/arcgis-server/1/arcgis-server.html.

Before you begin

Before you begin, gather this connection information:

- URL for the Esri ArcGIS Server that you want to connect to
- URL for the GeoService API that points to the data you want to connect to

In Esri ArcGIS Server, make sure that the following conditions are met:

- Be sure to publish data to ArcGIS Online and share as "Public".
- Publish your data in a geo-database to ensure pagination support.
- Enable Query capability for published Esri services.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Esri ArcGIS Server**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the URL for the **Esri ArcGIS Server** that hosts the tables that you want to connect to.

      Tips:

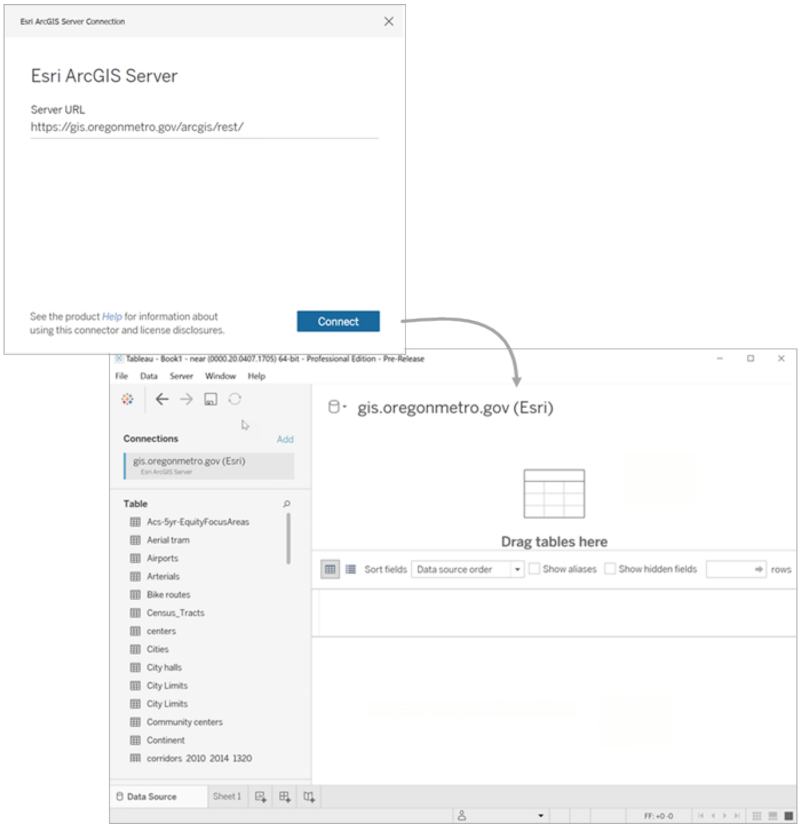
      - Use https URLs and make them as specific as possible to speed performance. Tableau's Web Data Connector (WDC) framework supports only HTTPS endpoints and scans every service in the instance.

        It's best to specify a folder, service, or service layer rather than a root URL. Examples:

        - Root: https://gis.oregonmetro.gov/arcgis/rest/
        - Folder: https://gis.oregonmetro.gov/arcgis/rest/services/OpenData
        - Service: https://gis.oregonmetro.gov/arcgis/rest/services/OpenData/BoundaryData/MapServer
        - Service Layer: https://gis.oregonmetro.gov/arcgis/rest/services/OpenData/BoundaryData/MapServer/0
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Under **Table**, select a table or use the search field to search for a table by name.

c. Drag a table to the canvas, and then select the sheet tab to start your analysis.



Troubleshooting Esri ArcGIS connections

Because the Esri ArcGIS Server connector is built around the Web Data Connector (WDC), there are several limitations to be aware of.

- The Esri ArcGIS Server connector does not support definition queries.
- The Esri ArcGIS Server connector does not support live connections, and will always create an extract. A packaged workbook with an ArcGIS connection can be opened in versions of Tableau older than 2019.4, but you will be unable to refresh the extract.
- The connector searches only for MapServer and FeatureServer service types.
- The connector ignores the **esriFieldTypeBlob** and **esriFieldTypeRaster** types.
- Esri geometry types are limited to the following:
  - esriGeometryPoint
  - esriGeometryMultipoint
  - esriGeometryLine

    - esriGeometryPath
    - esriGeometryPolyline
    - esriGeometryPolygon
    - esriGeometryEnvelope
- If the ArcGIS Server contains multiple layers or tables with identical names within a service, only one is displayed.
- Editing the connection may cause an "Invalid map <k,t>" error. If that happens, close and reopen the workbook.

License disclosures

The Esri ArcGIS Server Connector uses the arcgis-to-geojson-utils open source library under the Apache License version 2.0. A copy of the license is available in the repository’s LICENSE file.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Exasol

This article describes how to connect Tableau to data stored in the Exasol platform and set up the data source. Tableau can connect to Exasol version 4.2 and later.

Before you begin

Before you begin, gather this connection information:

- Name of the server you want to connect to
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Exasol**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of the server that you want to connect to.
    b. Enter the user name and password.
    c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    d. Select **Sign In**.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.
    c. Under **Table**, select a table or use the text box to search for a table by name.
    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

    Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Firebird 3

This article describes how to connect Tableau to a Firebird database and set up the data source.

> **Note:** In version 2020.1, the Firebird 3 connector replaced the Firebird connector. If you're using an earlier version of Tableau, go to the Tableau Help page and select the version you need from the drop-down list on the left. Then search for "Firebird" to find information on that connector.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Location of the database
- User name and password

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Firebird 3**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the database, or browse to the location of the database.
   c. Enter the user name and password, and then select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

    b. Select a table, drag it to the canvas, and then select the sheet tab to start your analysis.

Starting in 2019.2, the Firebird connector no longer supports custom SQL connections.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Upgrade a Firebird extract

Beginning with version 10.5, Tableau has changed the format of extracts to the .hyper format. This format change requires Firebird extracts be opened and saved using an earlier version of Tableau before it can either be used with the current version of Tableau or be upgraded to the .hyper format. When you open and save the Firebird extract in an earlier version of Tableau, the Firebird extract is upgraded to a .tde format. To upgrade a Firebird extract, follow the steps listed below.

1. Get an earlier version of Tableau Desktop. To get an earlier version of Tableau Desktop, see the Tableau Alternate Downloads Site page.

2. Install the earlier version of Tableau Desktop, and then use it to open the Firebird extract.

3. Save the extract.

4. Open the upgraded Firebird extract using the current version of Tableau Desktop. The extract should work as expected.

> **Note:** You can upgrade your extract from a .tde to a .hyper format by selecting **Data** > **Extract** > **Upgrade**. For more information, see **Extract Upgrade to .hyper Format** on page 1031.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Google Ads

This article describes how to connect Tableau to Google Ads and set up the data source.

> **Note:** For the most up-do-date information on this connector, see the Google Ads help topic for the current version of Tableau.

Before you begin

Before you begin, gather this connection information:

- Your email address or user ID and password

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Ads**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Sign in to Google Ads using your email or phone, and then select **Next** to enter your password. If multiple accounts are listed, select the account that has the GA data you want to access, and enter the password, if you are not already signed in.

    b. Select **Allow** to authorize Google to securely share your data with Tableau Desktop.

    c. Close the browser window when notified to do so.

    d. In Tableau Desktop, select your Account and your Client Customer ID

e. Then, select the pre-defined report and date filters

f. You also can select the columns to show for the report previously selected.

g. Select Connect.

> If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Google Ads administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. By default, the report selected will be displayed under Table and will be automatically drag it to the top of the canvas.

   c. Select the sheet tab to start your analysis.

   d. After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Google Ads. You can update the data by refreshing the extract. For more information, see Extract Your Data.

   e. Creating extracts may take some time depending on the amount of data that is included.

Key considerations

Account Requirements

To use the Google Ads connector, you must be a customer of Google Ads.

Date range selections can impact performance

When you apply a date filter, it's tempting to gather as much data as possible when you do an analysis, however, retrieving records from Google Ads can be time-consuming. Tableau doesn’t know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance.

To give you a rough idea of how much time it might take to retrieve data from Google Ads, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

Number of Records	Time to Retrieve
1,000	11 seconds
10,000	2 minutes
100,000	18 minutes

Selecting more than one segment can impact performance

While in the Google Ads UI only one segment at a time can be used for display, with the Google Ads connector you can combine multiple segments in the same report.

Keep in mind that the number of rows can increase exponentially for each additional segment field included in your report.

For technical details visit:

Segmentation

Use a calculated field to correct money values

The connector returns the data in the same format as the Google Ads API. For Money fields (such as costs, amounts, etc.), the Google Ads API returns them in micro currency units (micros). To have the correct value you need to divide by 1000000 to get the amount in the account's local currency.

Please, refer to:

Adwords Cost Format Issue

Format of various fields

Troubleshoot data access

If you see an error when you try to log in using your google account, for example, " The login information provided corresponds to a Google account that does not have Ads enabled. Make sure to login with a valid Ads account....”, contact your company's assigned Google Ads account administrator.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

5 Tips to Get More from Google Analytics - Read the Tableau whitepaper (registration or sign in required).

Google Analytics

This article describes how to connect Tableau to Google Analytics (GA) and set up the data source.

Before you begin

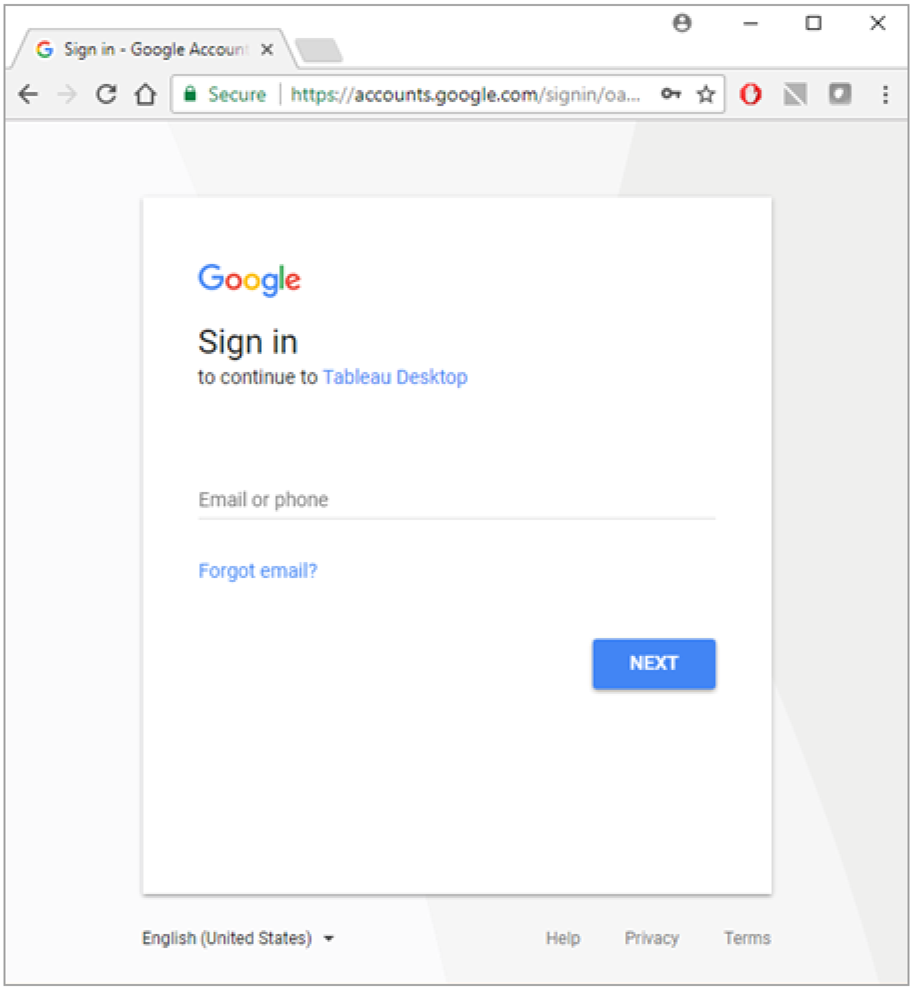
Before you begin, gather this connection information:

- GA email address and password

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Analytics**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

    a. Sign in to GA using your email or phone, and then select **Next** to enter your password. If multiple accounts are listed, select the account that has the GA data you want to access, and enter the password, if you are not already signed in.



b. Select **Allow** so that Tableau Desktop can access your GA data.

c. Close the browser window when notified to do so.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Follow the steps at the top of the data source page to complete the connection.

      Step 1 – Select an Account, Property, and Profile using the drop-down menus.

      Step 2 – Select filters for a date range and a segment.

- For **Date Range**, you can select one of the predefined date ranges or select specific dates. When selecting a date range, GA can provide complete data only up to the previous full day. For example, if you choose Last 30 days, data will be retrieved for the last 30-day period ending yesterday.
- For **Segment**, select a segment to filter your data. Segments are preset filters that you can set for a GA connection. Default Segments are defined by Google, and Custom Segments are defined by the user on the GA website. Segments also help prevent sampling to occur by filtering the data as defined by the segment. For example, with a segment, you can get results for a specific platform, such as tablets, or for a particular search engine, such as Google.

> **Note:** GA restricts the amount of data that it returns in a query. When you try to retrieve more data than GA allows in a single query, GA returns sampled data instead. If Tableau detects that your GA query might return sampled data, Tableau attempts to bypass the query restriction to return all data instead. For more information, see **All data vs. sampled data returned from a query** below below.

Step 3 – Add dimensions and measures by using the **Add Dimension** and **Add Measure** drop-down menus, or select a predefined set of measures from the **Choose a Measure Group** drop-down menu. Some dimensions and measures cannot be used together. For more information, see Dimensions & Metrics Reference Guide on the Google developer website.

c. Select the sheet tab to start your analysis. After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Google Analytics. You can update the data by refreshing the extract. For more information, see **Extract Your Data** on page 1014.

All data vs. sampled data returned from a query

GA restricts the amount of data that it returns from a query and provides sampled data instead. Sampled data is a random subset of your data. When performing analysis on sampled data, you can miss interesting outliers, and aggregations can be inaccurate. If Tableau detects that your query might return sampled data, by default, Tableau creates multiple queries from your query, and then combines the results from the queries to return all data.

You see the following message when Tableau returns all data.

If the query stays within the boundaries of the query restrictions, GA doesn't return sampled data and you do not see the above message.

Troubleshoot issues with returning all data

If your query continues to return sampled data, consider the following:

- **Missing date dimension** – You must use the date dimension in your query to return all data.
- **Too much data** – Your query might contain too much data. Reduce the date range. Note that the minimum date range is one day.
- **Non-aggregatable dimensions and measures** – Some dimensions and measures cannot be separated into multiple queries. If you suspect a problematic dimension or measure in your query, hover over the **All data** button to see the tooltip that shows which dimensions or measures to remove from your query.
- **Legacy workbooks** – Workbooks created in Tableau Desktop 9.1 and earlier cannot return all data. Open the legacy workbook in Tableau Desktop 9.2 and later and save the workbook.

Return sampled data

In cases when workbook performance is critical or there are specific dimensions and measures you want to use in your query that are not supported by Tableau's default query process, use sampled data instead. To return sampled data, select the **Sample data** button.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

5 Tips to Get More from Google Analytics - Read the Tableau whitepaper (registration or sign in required).

Google BigQuery

This article describes how to connect Tableau to Google BigQuery and set up the data source.

Before you begin

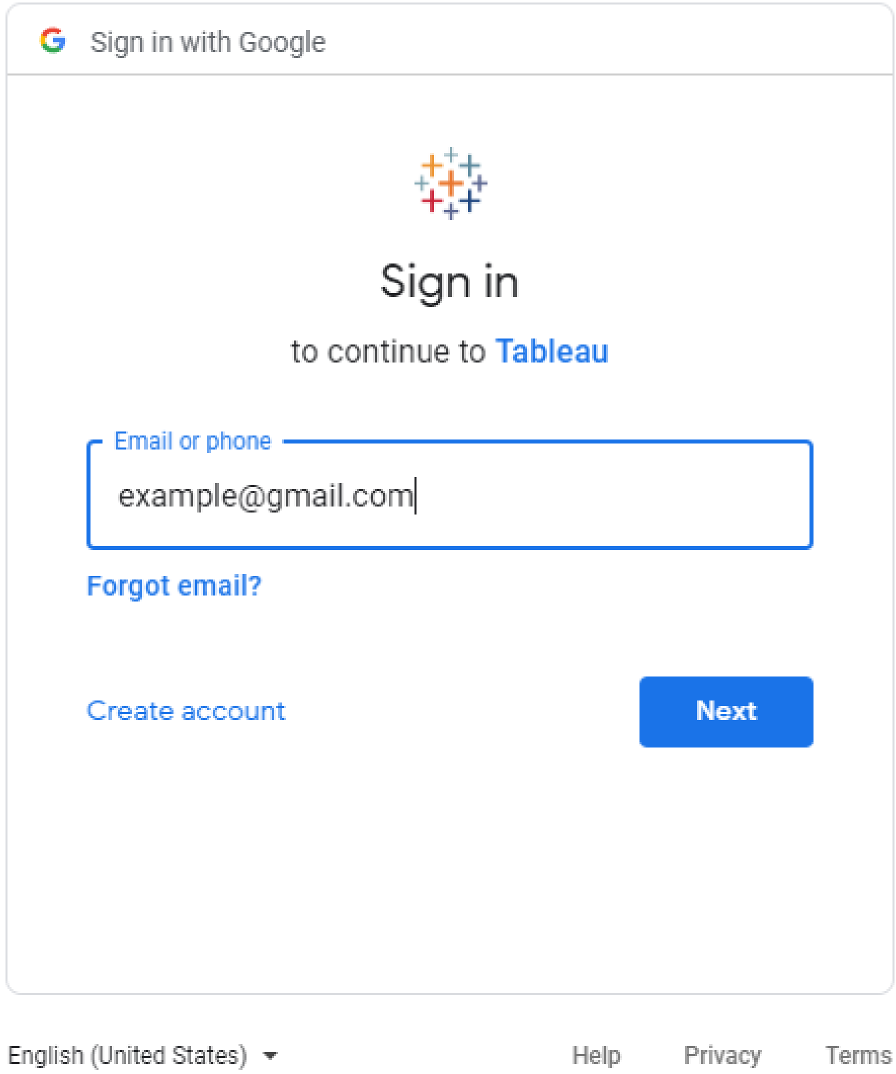
Before you begin, gather this connection information:

- Google BigQuery email or phone, and password

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google BigQuery**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

    a. Sign in to Google BigQuery using your email or phone, and then select **Next** to enter your password. If multiple accounts are listed, select the account that has the Google BigQuery data you want to access and enter the password, if you're not already signed in.



b. Select **Accept** so that Tableau can access your Google BigQuery data.

c. Close the browser window when notified to do so.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. (Optional) From the **Billing Project** drop-down list, select a billing project. If you don't select a billing project, `EmptyProject` appears in the field after you have selected the remaining fields.

c. From the **Project** drop-down list, select a project. Alternatively, select **publicdata** to connect to sample data in BigQuery.

d. From the **Dataset** drop-down list, select a data set.

e. Under **Table**, select a table.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Notes:
- Google BigQuery has changed support from BigQuery legacy SQL (BQL) to standard SQL. Your workbooks will upgrade to support standard SQL when you open them in Tableau.
- Because of the large volume of data in BigQuery, Tableau recommends that you connect live.
- The Radians() function is not supported in Google BigQuery.

Use customization attributes to improve query performance

Note: Customization attributes aren't currently supported in Tableau Prep Builder.

You can use customization attributes to improve the performance of large result sets returned from BigQuery to Tableau Online and Tableau Server, and on Tableau Desktop.

You can have the customization attributes included in your published workbook or data source, as long as you specify the attributes before you publish the workbook or data source to Tableau Online or Tableau Server.

Use Google BigQuery customization attributes

Customization attributes accept integer values and affect both live queries and extract refreshes for the specified connection.

The following attributes help the most to increase performance of large result sets:

bq-fetch-tasks	Number of parallel background tasks to use when fetching data using HTTP. The default is 10.

bq-large-fetch-rows	Number of rows to fetch in each batch for spool queries. The default is 50000.

The following attributes are also available and are primarily used for small queries:

bq-fetch-rows	Number of rows to fetch in each batch for non-spool queries. The default is 10000.
bq-response-rows	Number of rows returned in non-spool non-batched queries. The default is 10000.

This capability setting accepts yes or no values and can be useful when testing:

CAP_BIGQUERY_FORCE_SPOOL_JOB	Force all queries to use the temp table approach. The default value is “no.” Change the value to “yes” to turn this attribute on.

How Tableau returns rows from Google BigQuery

Tableau uses two approaches to return rows from BigQuery: the default non-spool approach, or the temp table (spool) approach:

- On the first attempt, queries are executed using the default, non-spool query, which uses the bq-fetch-rows setting.
- If the result set is too large, the BigQuery API returns an error and the Tableau BigQuery connector retries the query by saving the results into a BigQuery temp table. The BigQuery connector then reads from that temp table, which is a spool job that uses the bq-large-fetch-rows setting.

How to specify the attributes

You can specify attributes in one of two ways: in a Tableau Datasource Customization `.tdc` file, or in the workbook or data source XML.

Specify attributes in a `.tdc` file

To specify customization attributes during a publish workbook or publish data source operation from Tableau Desktop, follow these steps:

1. Create an XML file that contains the customization attributes.
2. Save the file with a `.tdc` extension, for example, `BigQueryCustomization.tdc`.
3. Save the file to the `My Tableau Repository\Datasources` folder.

The customization attributes in the `.tdc` file are read and included by Tableau Desktop when the data source or workbook is published to Tableau Online or Tableau Server.

> **Important:** Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

Example of a .tdc file with recommended settings for large extracts

```
<connection-customization class='bigquery' enabled='true' ver-
sion='8.0' >

  <vendor name='bigquery' />

  <driver name='bigquery' />

  <customizations>

    <customization name='bq-fetch-tasks' value='10' />

    <customization name='bq-large-fetch-rows' value='10000' />

  </customizations>

</connection-customization>
```

Manually embed attributes in the XML of the workbook or data source file

You can manually embed customization attributes inside the 'connection' tag in the workbook .twb file or the data source .tds file. The BigQuery customization attributes are bold in the following example to make them easier for you to see.

Example of manually embedded attributes

```
<connection CATALOG='publicdata' EXECCATALOG='some-project-123'
REDIRECT_URI='some-url:2.0:oob'
```

```
SCOPE='https://www.googleapis.com/auth/bigquery
https://www.googleapis.com/auth/userinfo.profile
https://www.googleapis.com/auth/userinfo.email'
authentication='yes' bq-fetch-tasks='10' bq-large-fetch-
rows='10000' bq_schema='samples' class='bigquery' connection-
dialect='google-bql' connection-protocol='native-api' login_
title='Sign in to Google BigQuery' odbc-connect-string-extras=''
project='publicdata' schema='samples'
server='googleapis.com/bigquery' server-oauth=''
table='wikipedia' username=''>
```

Check if your workbook uses standard SQL or legacy SQL

In 2016, Google updated the BigQuery APIs to support standard SQL, in addition to still supporting BigQuery SQL (now called legacy SQL). Starting in Tableau 10.1, the Google BigQuery connector has been upgraded to support standard SQL, and also still supports legacy SQL. Standard SQL enables users of the BigQuery connector to use level of detail expressions, get faster metadata validation, and select a billing project with the connection.

Now, when you create a new workbook, Tableau supports standard SQL by default. Tableau also supports legacy SQL using the **Use Legacy SQL** option on the **Data** pane. For example, when you open a workbook that was created using a previous version of Tableau Desktop, and if your workbook uses legacy SQL, the **Use Legacy SQL** option is selected.

You might want to configure the **Use Legacy SQL** option for the following reasons:

- You have an existing workbook that you want to update to use standard SQL in order to write level of detail expressions or to take advantage of other improvements. In this case, make sure that the **Use Legacy SQL** option is not selected.
- You are creating a new workbook that needs to connect to a legacy SQL view. You can’t mix legacy SQL with standard SQL, so you need to select the **Use Legacy SQL** option for the workbook to function.

In Google BigQuery, views are written in standard SQL or legacy SQL. You can join views written in standard SQL to views written in standard SQL, or views written in legacy SQL to views written in legacy SQL, and you can join views written in either version of SQL to a table. But you can’t join views written in standard SQL and views written in legacy SQL in one workbook. When you join views, you must set the **Use Legacy SQL** check box to correspond to the SQL type used in the view you are connecting to.

Note: Tableau Desktop provides limited support for nested data when you use legacy SQL or standard SQL. For example, if a table contains nested data and you're using legacy SQL or standard SQL, on the data source page, **Update Now** won't work.

For more information about migrating from legacy SQL to standard SQL, see Migrating from legacy SQL on the Google Cloud Platform website.

Troubleshoot Google BigQuery issues

Connections to multiple accounts

When using web authoring or publishing to the web, you cannot use multiple Google BigQuery accounts in the same workbook. You can have multiple Google BigQuery account connections in Desktop.

When publishing flows, the credentials that you use to connect to Google BigQuery in your Input step must match the credentials that are set up in the **Settings** tab in the **My Account Settings** page for Google BigQuery in Tableau Server or Tableau Online. If you select different credentials or no credentials in your authentication setting when publishing the flow, the flow will fail with an authentication error until you edit the connection for the flow in Tableau Server or Tableau Online to match those credentials.

Web authoring with Internet Explorer 11 and Edge

In Internet Explorer 11 and Edge, you cannot access a server using an unsecured connection (http). Use a secure connection (https) or switch to another browser.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

- Set up Oauth for Google -Configure Oauth connections for Tableau Server.
- Oauth Connections - Configure Oauth connections for Tableau Online.
- Google BigQuery & Tableau: Best Practices - Read the Tableau whitepaper (registration or sign in required)

Google Cloud SQL

This article describes how to connect Tableau to a Google Cloud SQL database instance and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Cloud SQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password, and then select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Google Drive

This article describes how to connect Tableau to Google Drive and set up the data source.

Before you begin

Before you begin, gather this connection information:

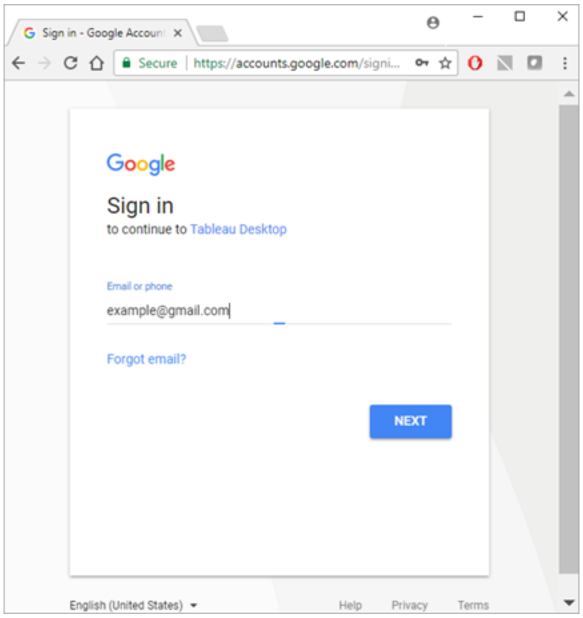
- Google email address and password

> **Note**: The Google Drive connector does not support Google shared drives.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Drive**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

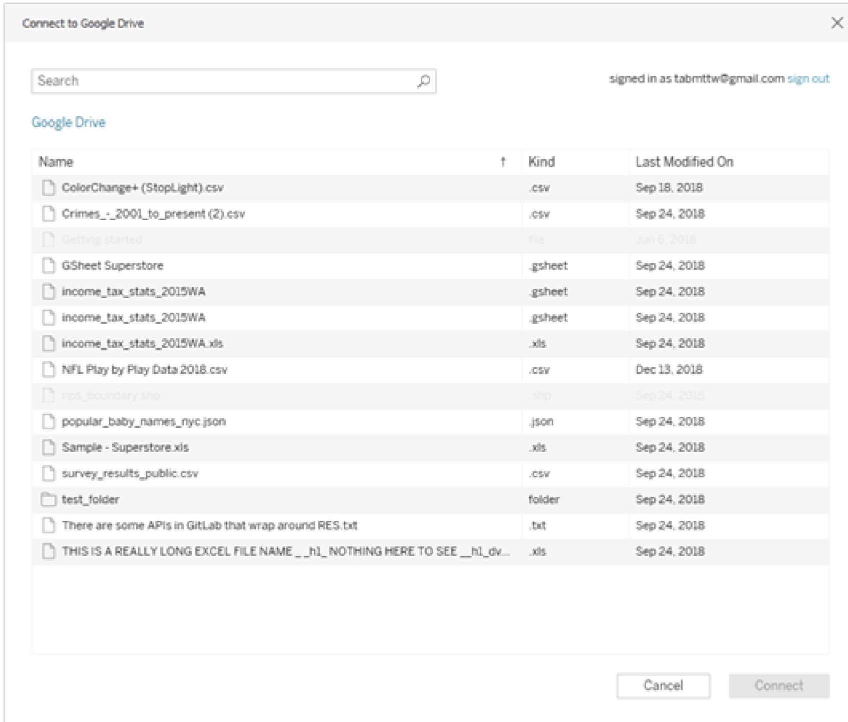
    a. Sign in to Google Drive using your email or phone, and then select **Next** to enter your password. If multiple accounts are listed, select the account that has the Google Drive data you want to access and enter the password, if you're not already signed in.



b. Select **Allow** so that Tableau Desktop can access your Google Drive data.

c. Close the browser window when notified to do so.

d. Select a file from the list or use the text box to search for a file by name, and then select **Connect**.



Note: This connector might not support all file types. Unsupported file types are grayed out.

2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data figure out which data source to connect to.

    b. If your Google Drive file has one table, select the sheet tab to start your analysis.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

Troubleshoot Google Drive issues

Data limit in Google Drive

Google Drive supports a limited number of cells (in the millions) for spreadsheets that are created in or converted to Google Sheets. For more information, see Files you can store in Google Drive in the Google Drive Help.

Connections to multiple accounts

When using web authoring or publishing to the web, you cannot use multiple Google Drive accounts, even when using different connections. You can have multiple Google Drive account connections in Desktop.

Web authoring with Internet Explorer 11 and Edge

In Internet Explorer 11 and Edge, you cannot access a server using an unsecured connection (http). Use a secure connection (https) or switch to another browser.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Google Sheets

This article describes how to connect Tableau to Google Sheets and set up the data source.

Before you begin

Before you begin, gather this connection information:

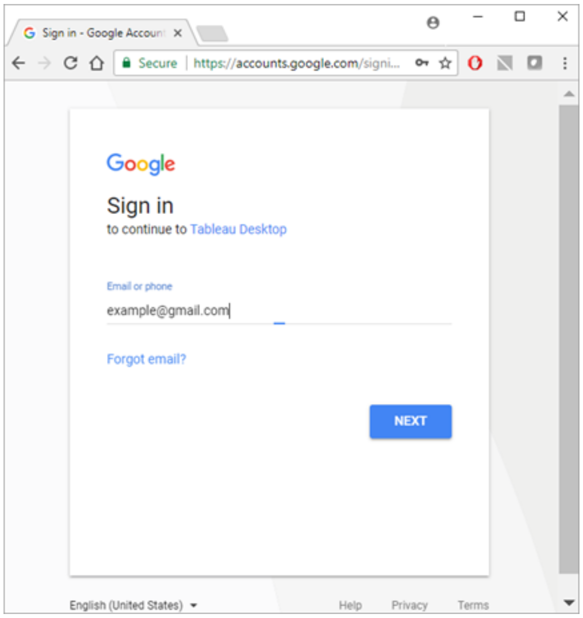
- Google email address and password

> **Note**: The Google Sheets connector does not support Google shared drives.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Google Sheets**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:

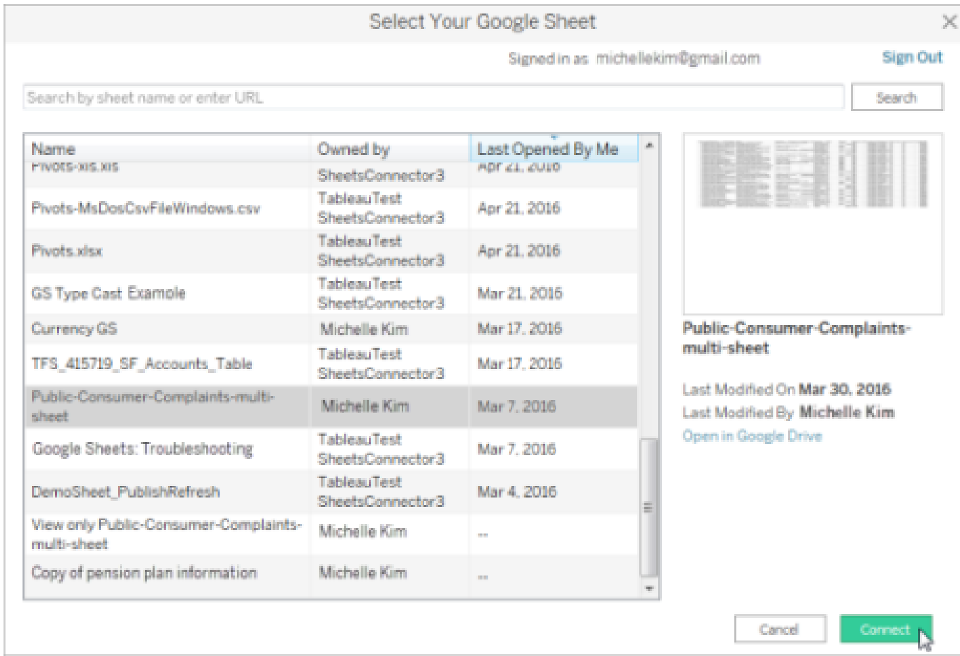
a. Sign in to Google Sheets using your email or phone, and then select **Next** to enter your password. If multiple accounts are listed, select the account that has the Google Sheets data you want to access and enter the password, if you're not already signed in.



b. Select **Allow** so that Tableau Desktop can access your Google Sheets data.

c. Close the browser window when notified to do so.

d. Select a Google Sheet from the list or use the text box to search for a Google Sheet by name or by URL, and then select **Connect**.



2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data figure out which data source to connect to.

    b. If your Google Sheets file has one table, select the sheet tab to start your analysis.

Note: Tableau doesn't support pivot tables in Google Sheets.

Select Your Google Sheet dialog box functionality

The Select Your Google Sheet dialog box includes the following functionality:

- The list of sheets that you can select from includes your private sheets, sheets shared with you, and the public sheets that you've accessed in the past.
- If you search by URL and the URL doesn't exist or you don't have access to it, an error displays.
- You can select the **Name** and **Last opened by me** column names to sort the Google Sheets, and when you select a sheet you can preview it in the right pane. You cannot sort by **Owned by**.

Connect to more data

You can connect to more than one table by using join. For more information, see **Join Your Data** on page 883.

You can also connect to a named range the same way you connect to a worksheet. The named range functions as a table in Tableau.

You create named ranges in Google Sheets by highlighting a range of cells and then selecting **Data** > **Named ranges**. When you connect to a named range in Tableau, an icon appears next to the sheet in the **Data Source** tab as shown below.

About .ttde and .hhyper files

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

Troubleshoot Google Sheets issues

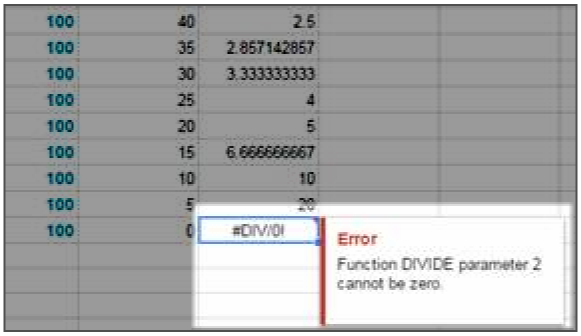
Data limit in Google Drive

Google Drive supports a limited number of cells (in the millions) for spreadsheets that are created in or converted to Google Sheets. For more information, see Files you can store in Google Drive in the Google Drive Help.

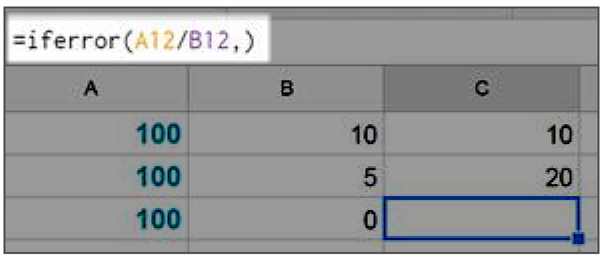
Error message: Internal Error - An unexpected error occurred and the operation could not be completed.

If there are errors in your Google Sheet, such as #DIV/0! or #N/A, Tableau is unable to create an extract and an error message will appear. To resolve this issue, wrap the function with `iferror()` and have it return a blank, or any value that's appropriate.

For example, the sheet below includes a #DIV/0! error.



The solution is to wrap the calculation in an `iferror()` calculation.



See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Hortonworks Hadoop Hive

This article describes how to connect Tableau to a Hortonworks Hadoop Hive database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Authentication method:

  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
  - Binary
  - SASL
  - HTTP
- Sign-in credentials depend on the authentication method you choose and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service name
  - HTTP path
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** Make sure you use the latest available drivers. To get the latest drivers, see Hortonworks Hadoop Hive on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Hortonworks Hadoop Hive**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of server that hosts the database.
    b. In the **Authentication** drop-down list, select the authentication method to use.
    c. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.
    d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    e. Select **Sign In**.

    Select the **Require SSL** option when connecting to an SSL server.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.
    c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.
    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

    Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

    Note: This database type only support equal (=) join operations.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see **Date Functions** on page 2170.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to **String** and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera's Impala and Hortonworks' Stringer project) are designed to address this limitation.

Truncated columns in Tableau

The default string column length for Hortonworks Hadoop Hive is 255 characters. For more information about Hortonworks Hive ODBC driver configuration options, and specifically about DefaultStringColumnLength, see the Hive ODBC Driver User Guide from Hortonworks.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

IBM BigInsights

This article describes how to connect Tableau to an IBM BigInsights database.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Database name
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

SSL requirements

This connector enables you to connect to your server using SSL authentication. If you work in an SSL environment, your computer is probably already configured to support SSL. If you have trouble making an SSL connection with Tableau, make sure that the following IBM software is installed on your computer:

- IBM Data Server Runtime Client v10.5fp3 or above
- GSK8 SSL Library

Note that the IBM Data Server Runtime Client and GSK8 Library must have the same bitness. For example, both must be either 32-bit or 64-bit. For more information, see Installation of the GSK8 SSL Library on the IBM website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **IBM BigInsights**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the name of the server that hosts the database and the port number to use.

   b. Enter the name of the database you want to connect to.

   c. Enter your user name and password.

   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

   e. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.

   c. Under **Table**, select a table or use the text box to search for a table by name.

   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.

- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Using Tableau with IBM BigInsights Hadoop - Tableau blog post.

IBM DB2

This article describes how to connect Tableau to an IBM DB2 database and set up the data source. You can also use this connector to connect to an IBM DB2 for z/OS database. See the Technical Specifications to confirm which DB2 databases are supported.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to (Host name) and port number
- Database name
- User name (User ID) and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Port settings

The port number is dependent on the type of server you are connecting to and whether you are connecting to an encrypted port. Generally, use 50000 for a non-encrypted port and 60000 for an encrypted port. It is possible that your server is configured to use a non-standard port. Contact your administrator if you don't know which port to connect to.

SSL requirements

This connector enables you to connect to your server using SSL authentication. If you work in an SSL environment, your computer is probably already configured to support SSL. If you have trouble making an SSL connection with Tableau, make sure that the following IBM software is installed on your computer:

- IBM Data Server Runtime Client v10.5fp3 or above
- GSK8 SSL Library

Note that the IBM Data Server Runtime Client and GSK8 Library must have the same bitness. For example, both must be either 32-bit or 64-bit. For more information, see Installation of the GSK8 SSL Library on the IBM website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **IBM DB2**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of the server that hosts the database (the Host name), the port number, and the name of the database that you want to connect to.
    b. Enter your user name (User ID) and password.

       If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
    c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    d. Select **Sign In**.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.
    c. Under **Table**, select a table or use the text box to search for a table by name.
    d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: Tableau doesn't support the DECFLOAT data type.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Connecting Tableau to a Db2 database - Review connection instructions on the IBM website.

IBM PDA (Netezza)

This article describes how to connect Tableau to an IBM PDA (PureData System for Analytics) database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **IBM PDA (Netezza)**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the name of the database that you want to connect to.
   c. Enter the user name and password.

      Select the **Require SSL** option when connecting to an SSL server.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   e. Select **Sign In**.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table or use the text box to search for a table by name.
   c. Drag a table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Impala

This article describes how to connect Tableau to an Impala database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name and port of the server that hosts the database you want to connect to
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
- Transport type (Username and Password authentication only):
  - Binary
  - SASL
- Sign-in credentials depend on the authentication method you choose and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service name
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** Make sure you use the latest available drivers. To get the latest drivers, see Impala on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Impala**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Enter the name of the server that hosts the database and the port number to use. Port 21050 is the default .

    b. In the **Authentication** drop-down list, select the authentication method to use.

    c. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.

    d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

    e. Select **Sign In**.

    Select the **Require SSL** option when connecting to an SSL server.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

    b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

    c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

    Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

    Note: This database type supports only equal (=) join operations.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Intuit QuickBooks Online

This article describes how to connect Tableau to Intuit QuickBooks Online data and set up the data source.

> **Note:** For the most up-do-date information on this connector, see the Intuit QuickBooks Online help topic for the current version of Tableau.

Before you begin

Before you begin, gather this connection information:

- Your email address or user ID and password for your Quickbooks Online account.

> **Note:** To connect Tableau to QuickBooks Online data, you must be a Company Administrator on the QuickBooks Online company. Only one administrator at your company can connect Tableau to QuickBooks Online.

QuickBooks Online doesn't support retrieving the Inventory Adjustment transactions. For information about when Inventory Adjustment transactions will be available in the QuickBooks Online API, see the responses to this question on the Intuit Developer website.

Make the connection and set up a data source

1. Start Tableau and under **Connect**, select **Intuit QuickBooks Online**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:
   a. Enter your email address or user ID and password for your QuickBooks Online account.
   b. Select **Sign In**.

      QuickBooks Online periodically prompts for two-step verification. If this happens, follow the prompts to get a code to confirm your Intuit account.
   c. If more than one company is associated with your account, select the company you want to connect to.
   d. Select **Authorize** to authorize Intuit to securely share your data with Tableau Desktop.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table and drag it to the top of the canvas.
   c. Select the Sheet 1 tab to start your analysis.

      After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for QuickBooks Online. You can update the data by refreshing the extract. For more information, see **Extract Your Data** on page 1014.

      Creating extracts may take some time depending on the amount of data that is included.

Use the Sales and Expenses Line Items table to create accounting reports

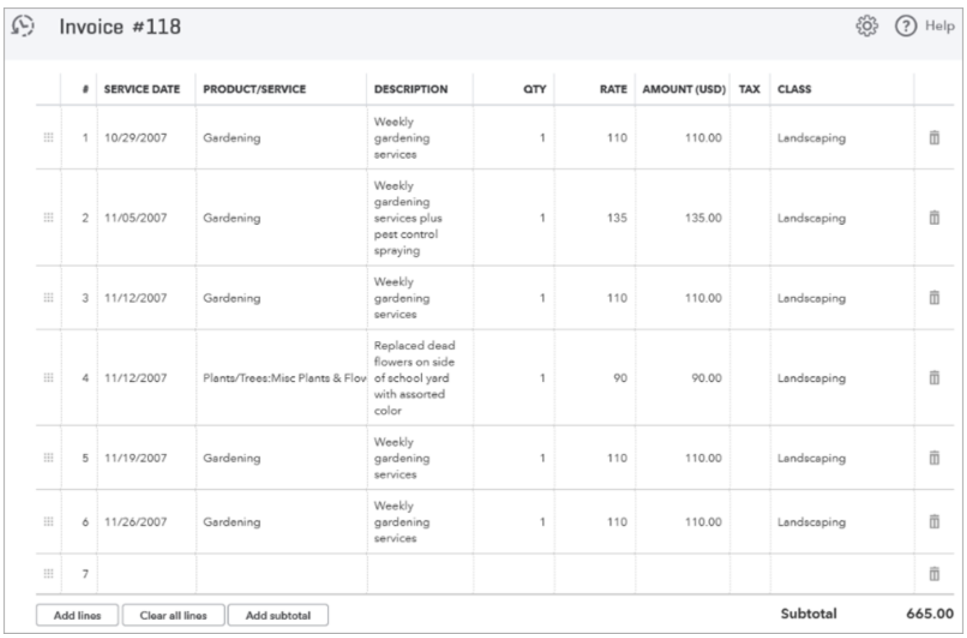
QuickBooks Online aggregates the information needed to create a variety of reports, such as profit and loss (P&L) statements and balance sheets. Because the data is aggregated by QuickBooks Online, you can’t create these kinds of reports in Tableau. To remedy this, we’ve added a table in Tableau, Sales and Expenses Line Items, that includes line item details with all the fields you need to create these reports.

Sales and Expenses Line Items table structure

The Sales and Expenses Line Items table contains information for the following QuickBooks Online transactions (or documents):

Bill	Estimate	Purchase Order
Bill Payment	Invoice	Refund Receipt
Credit Memo	Journal Entry	Sales Receipt
Deposit	Payment	Vendor Credit
	Purchase	

Most of the transactions have children (or sub-items), for example, an invoice has detail items. Like this example:



Invoice #118

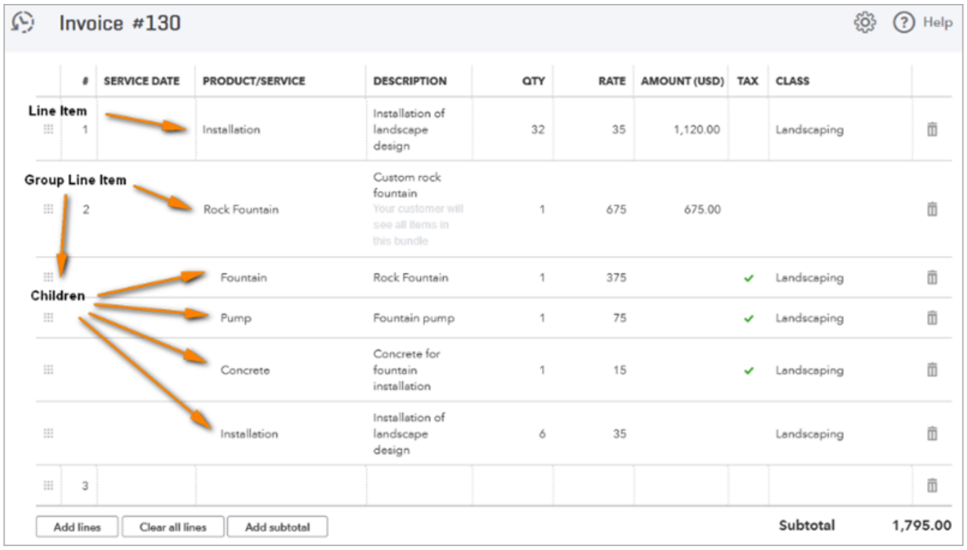
Help

#	SERVICE DATE	PRODUCT/SERVICE	DESCRIPTION	QTY	RATE	AMOUNT (USD)	TAX	CLASS
1	10/29/2007	Gardening	Weekly gardening services	1	110	110.00		Landscaping
2	11/05/2007	Gardening	Weekly gardening services plus pest control spraying	1	135	135.00		Landscaping
3	11/12/2007	Gardening	Weekly gardening services	1	110	110.00		Landscaping
4	11/12/2007	Plants/Trees:Misc Plants & Flov	Replaced dead flowers on side of school yard with assorted color	1	90	90.00		Landscaping
5	11/19/2007	Gardening	Weekly gardening services	1	110	110.00		Landscaping
6	11/26/2007	Gardening	Weekly gardening services	1	110	110.00		Landscaping
7								

Add lines | Clear all lines | Add subtotal

Subtotal 665.00

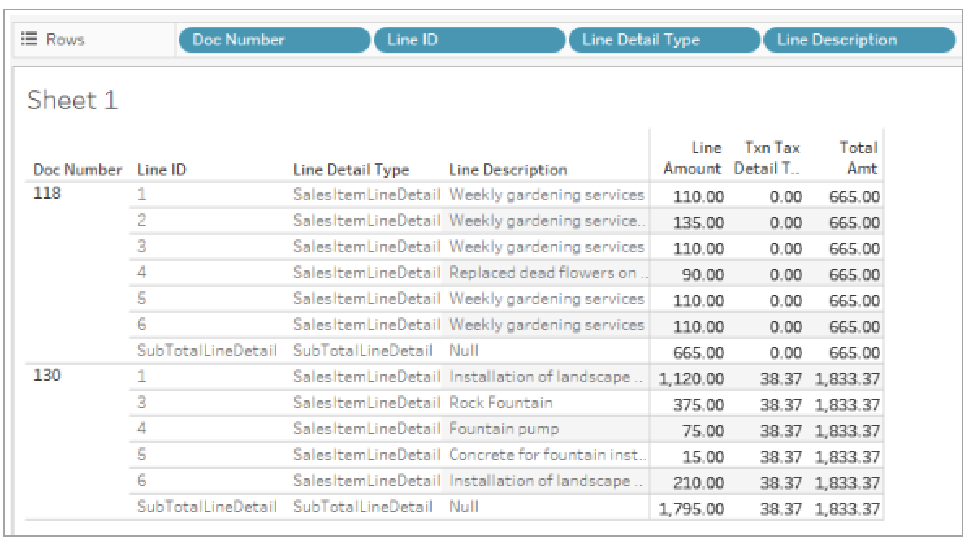
Some transactions have group items, which consist of a group of other line items. Like this example:


Invoice #130

#	SERVICE DATE	PRODUCT/SERVICE	DESCRIPTION	QTY	RATE	AMOUNT (USD)	TAX	CLASS
1		Installation	Installation of landscape design	32	35	1,120.00		Landscaping
2		Rock Fountain	Custom rock fountain Your customer will see all items in this bundle	1	675	675.00		
		Fountain	Rock Fountain	1	375		✓	Landscaping
		Pump	Fountain pump	1	75		✓	Landscaping
		Concrete	Concrete for fountain installation	1	15		✓	Landscaping
		Installation	Installation of landscape design	6	35			Landscaping
3								

The **Sales and Expenses Line Items** table denormalizes the "transaction to items" relationship by including one row per item and duplicating the transaction fields on each row. In the case of a group line item with children, the table includes the group child items instead of the group.

The following image shows how these two invoices appear in Tableau with the denormalized data:



Sheet 1

Doc Number	Line ID	Line Detail Type	Line Description	Line Amount	Txn Tax Detail T..	Total Amt
118	1	SalesItemLineDetail	Weekly gardening services	110.00	0.00	665.00
	2	SalesItemLineDetail	Weekly gardening service..	135.00	0.00	665.00
	3	SalesItemLineDetail	Weekly gardening services	110.00	0.00	665.00
	4	SalesItemLineDetail	Replaced dead flowers on ..	90.00	0.00	665.00
	5	SalesItemLineDetail	Weekly gardening services	110.00	0.00	665.00
	6	SalesItemLineDetail	Weekly gardening services	110.00	0.00	665.00
	SubTotalLineDetail	SubTotalLineDetail	Null	665.00	0.00	665.00
130	1	SalesItemLineDetail	Installation of landscape ..	1,120.00	38.37	1,833.37
	3	SalesItemLineDetail	Rock Fountain	375.00	38.37	1,833.37
	4	SalesItemLineDetail	Fountain pump	75.00	38.37	1,833.37
	5	SalesItemLineDetail	Concrete for fountain inst..	15.00	38.37	1,833.37
	6	SalesItemLineDetail	Installation of landscape ..	210.00	38.37	1,833.37
	SubTotalLineDetail	SubTotalLineDetail	Null	1,795.00	38.37	1,833.37

In addition to the transactions tables, the **Sales and Expenses Line Items** table contains information for the **Account** table.

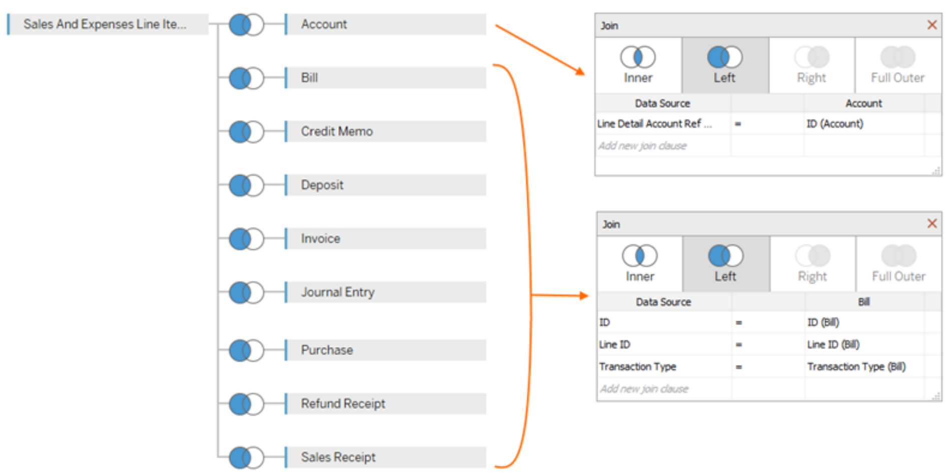
Join tables with the Sales and Expenses Line Items table

Using the **Sales and Expenses Line Items** table, you can join the **Account** and transactions tables with it to get the detailed information you need for the report you want to create.

To join the **Account** table with each line in a transaction, join [Line Detail Account Ref Value] in the **Sales and Expenses Line Items** table to [ID Account] in the **Account** table.

For the QuickBooks Online transactions tables (for example, **Bill**, **Estimate**, **Purchase Order** and so on), join [Transaction Type] + [ID] + [Line ID] in the **Sales and Expenses Line Items** table to the equivalent fields in the transactions table. For example, if you create a join with the **Bill** transactions table, the [ID] field to join on for Bill is [(ID) Bill]; the [Line ID] field to join on for Bill is [Line ID (Bill)], and so on.

The following image shows multiple joins with the **Sales and Expenses Line Items** table on the left, as well as examples of the join clauses for the **Account** table and the **Bill** table on the right.



Create the QuickBooks Online data source

These are the general steps to get you started with making a data source that you can use to create various reports, such as P&L reports:

1. Connect to QuickBooks Online.
2. On the **Data Source** page, under **Table**, select the **Sales And Expenses Line Items** table.

3. Join one or more of the tables listed in **Sales and Expenses Line Items table structure** on page 541 to the **Sales And Expenses Line Items** table to include the data you need for the report you want to create.
4. After you create your data source, you can build your statements and analyze your data.

Troubleshoot QuickBooks Online errors

You might see one of the following errors when you try to connect Tableau to your QuickBooks Online data.

Workbooks saved in previous versions of Tableau

Workbooks and data sources created prior to version 2018.2 of Tableau will not be able to access tables or fields added in later versions of the connector. To resolve this, open a new workbook and create a new connection to your QuickBooks Online data source. Copy and paste the worksheets you want to keep from the previous connection, or recreate them in Tableau.

Sorry, only administrators can buy apps…

QuickBooks Online requires that each user who connects an app to QuickBooks Online be a Company Administrator. The following Intuit error appears if you are not an administrator on the company account:

> Sorry, only administrators of <email address>'s Company can buy apps in the company. Please contact administrator in the company or else choose another company.

To resolve this issue, ask your company account owner to make you a Company Administrator on the account.

Error Code: app_already_purchased

Only one Company Administrator is allowed to connect per application, for example to Tableau Desktop. If someone in your company has already connected Tableau to your QuickBooks Online account, the following Intuit error appears:

> Oops! An error has occurred. Please close this window and try again.
>
> Error Code: app_already_purchased

Message: The application has already been subscribed to by another user for this company. Please contact <email address> to make changes to this subscription.

To resolve this issue, you need to ask that Company Administrator to give up the application privileges for Tableau Desktop and provide them to you. If that's not possible, for example, because the administrator actively uses Tableau, another option is for the administrator to publish the data source to Tableau Server or Tableau Online, so that anyone with permission can access the data.

Changing a Connection Opens the My Apps Page

If you need to access another QuickBooks Online company, after you sign in (and possibly go through the two-step authentication process), Intuit might show the My Apps page, rather than the "select the company to connect to" page. To resolve this issue, follow these steps:

1. Close the My Apps page.
2. On the Tableau **Connect** pane, select Intuit QuickBooks Online.
3. If more than one company is associated with your account, select the company you would like to connect to.
4. Select **Authorize** to open the Tableau data source page.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Kognitio

This article describes how to connect Tableau to a Kognitio database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password

- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Kognitio**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of the server that hosts the database you want to connect to.
    b. Enter the user name and password.
    c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    d. Select **Sign In**.

    Select the **Require SSL** check box when connecting to an SSL server.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.
    c. Under **Table**, select a table or use the text box to search for a table by name.
    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Kyvos

This topic describes how to connect Tableau to a Kyvos database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Port number
- HTTP path to the server
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Kyvos**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the **Server** name.
    b. Enter the **Port** number.
    c. Enter the **HTTP Path** to the data.
    d. Enter the **Username** and **Password**
    e. Select the **Require SSL** check box when connecting to an SSL server.
    f. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    g. Select **Sign In**.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. Under **Schema**, select a schema or use the text box to search for a schema by name.
    c. Under **Table**, select a table or use the text box to search for a table by name.
    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

    Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

LinkedIn Sales Navigator

This topic describes how to connect Tableau to a LinkedIn Sales Navigator database and set up the data source.

> **Note:** For the most up-do-date information on this connector, see the LinkedIn Sales Navigator help topic for the current version of Tableau.

Before you begin

Before you begin, get the email address or phone number, and password for the LinkedIn Sales Navigator account.

You need a Sales Navigator Enterprise plan and Sales Navigator administrator or reporting permissions. Note that a reporting user doesn't take up a Sales Navigator seat on the contract.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **LinkedIn Sales Navigator**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Select **Sign In** to authenticate to LinkedIn in your default browser.
   b. Enter your email or phone number and your password for the LinkedIn Sales Navigator account.
   c. Select **Sign in**.
   d. Close the browser window when notified to do so.
   e. In the LinkedIn Sales Navigator Connection dialog box, select the following:
      - **Contracts** : Custom or all
      - **Date Range**: Relative or fixed
   f. Select **Connect**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble accessing LinkedIn. Contact your network or database administrator.
2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
b. Select a standard connection or create your own custom connection.

   Under **Standard Connection**, you can select from a list of predefined selections of tables and joins that represent common scenarios and use cases. Or under **Table**, you can select a table or search for a table by name.
c. Drag the standard connection or table to the canvas, and then select the sheet tab to start your analysis.

Use Dashboard Starters

You can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics on Tableau Desktop and Tableau Online. For more information, see **Dashboard Starters for Cloud-based Data** on page 2750.

Date range selections can impact performance

It’s tempting to gather as much data as possible when you run an analysis. Tableau doesn’t know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance.

Refresh LinkedIn Sales Navigator data

After you click the sheet tab, Tableau imports the data by creating an extract. Tableau Desktop supports only extracts for LinkedIn Sales Navigator. Note that the initial extract may take some time.

To refresh, select Data> [data source name] > Extract> Refresh.

MapR Hadoop Hive

This article describes how to connect Tableau to a MapR Hadoop Hive database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service (starting in version 10.2.1)
- Transport options depend on the authentication method you choose and can include the following:
  - Binary
  - SASL
  - HTTP
- Sign-in credentials depend on the authentication method you choose and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service Name
  - HTTP Path
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** Make sure you use the latest available drivers. To get the latest drivers, see MapR Hadoop Hive on the Tableau Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MapR Hadoop Hive**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of the server that hosts the database.
    b. In the **Authentication** drop-down list, select the authentication method to use.
    c. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.
    d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    e. Select **Sign In**.

        Select the **Require SSL** option when connecting to an SSL server.

        If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.
    c. In the **Table** text box, select the search icon or enter the table name and select the search icon.
    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: This database type supports only equal (=) join operations.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Work with Hadoop Hive data

Work with date/time data

Tableau supports TIMESTAMP and DATE types natively. However, if you store date/time data as a string in Hive, be sure to store it in ISO format (YYYY-MM-DD). You can create a calculated field that uses the DATEPARSE or DATE function to convert a string to a date/time format. Use DATEPARSE() when working with an extract, otherwise use DATE(). For more information, see **Date Functions** on page 2170.

For more information about Hive data types, see Dates on the Apache Hive website.

NULL value returned

A NULL value is returned when you open a workbook in Tableau 9.0.1 and later and 8.3.5 and later 8.3.x releases that was created in an earlier version and has date/time data stored as a string in a format that Hive doesn't support. To resolve this issue, change the field type back to **String** and create a calculated field using DATEPARSE() or DATE() to convert the date. Use DATEPARSE() when working with an extract, otherwise use the DATE() function.

High latency limitation

Hive is a batch-oriented system and is not yet capable of answering simple queries with very quick turnaround. This limitation can make it difficult to explore a new data set or experiment with calculated fields. Some of the newer SQL-on-Hadoop technologies (for example, Cloudera's Impala and Hortonworks' Stringer project) are designed to address this limitation.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

MariaDB

This article describes how to connect Tableau to a MariaDB database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MariaDB**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database that you want to connect to.
   b. Enter the user name and password, and then select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data

source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Database** drop-down list, select a database.

c. Under **Table**, select a table or use the search field to search for a table by name.

d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Marketo

This article describes how to connect Tableau to Marketo data and set up the data source.

> **Note:** For the most up-do-date information on this connector, see the Marketo help topic for the current version of Tableau.

Before you begin

Before you begin, gather this connection information:

- Custom Service Endpoint
- Client ID
- Client Secret

For details about how to create the custom service information you need from Marketo to connect, see the Custom Service topic on the Marketo website.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Marketo**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter your Custom Service Endpoint, Client ID, and Client Secret.
   b. Select **Sign In**.
   c. Select **Filter Type**: **Relative date range** or **Fixed date range**, and then select or specify the range.
   d. Select **Connect** to authorize Marketo to securely share your data with Tableau Desktop.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Marketo administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table and drag it to the top of the canvas.
   c. Select the sheet tab to start your analysis.

      After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Marketo. You can update the data by refreshing the extract. For more information, see **Extract Your Data** on page 1014.

      Creating extracts may take some time depending on the amount of data that is included.

Bulk API retrieves data for some tables

Tableau uses the bulk API to retrieve data for the **All Activities**, **Leads**, and **Leads by List - <list name>** tables, thereby reducing the number of API calls. For more information, see the Bulk Extract documentation on the Marketo website.

You can use the Daily API Usage and Weekly API Usage tables to monitor your API usage.

Use Dashboard Starters

You can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics on Tableau Desktop and Tableau Online. For more information, see **Dashboard Starters for Cloud-based Data** on page 2750.

Date range selections can impact performance

It’s tempting to gather as much data as possible when you do an analysis, however, retrieving records from Marketo can be time-consuming. Tableau doesn’t know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance.

To give you a rough idea of how much time it might take to retrieve data from Marketo, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

Number of records	Time to retrieve
1,000	4.5 seconds
10,000	45 seconds
100,000	8 minutes
1,000,000	75 minutes

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

MarkLogic

This article describes how to connect Tableau to a MarkLogic database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Port number for the ODBC server process

- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MarkLogic**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of the server that hosts the database you want to connect to.
    b. Enter the port number for the ODBC server process of the database you want to connect to.
    c. Enter your user name and password.
    d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    e. Select **Sign In**.

    If the connection is unsuccessful, verify that your user name and password are correct and that the port number correctly identifies the MarkLogic database configured by your database administrator. If the connection continues to fail, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Schema** drop-down list, select a schema or use the text box to search

for a schema by name.

c. Under **Table**, select a table or use the text box to search for a table by name.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Tableau and MarkLogic Visualize Unstructured Data - Read the Tableau blog post.

Microsoft Analysis Services

This topic describes how to connect Tableau to a Microsoft Analysis Services database and set up a data source.

Before you begin

Before you begin, gather this connection information:

- For a remote cube file: Server name, or URL if connecting to the server using HTTP
- For a local cube file: File name
- Authentication method: Windows Authentication or user name and password

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Microsoft Analysis Services**. For a complete list of data connections, select **More** under **To a Server**. Then do the

following:

a. Select whether to connect to a remote cube file on a server or to a local cube file.

   To connect to a remote cube file, select **Server** and enter the name of the server in the text box. If you are connecting to the server using HTTP, you can enter the URL as the server name.

   To connect to a local cube file, select **Local cube file** and select **Browse** to navigate to the cube file on your computer.

b. Select how you want to sign in to the server. Specify whether to use Windows Authentication or a specific user name and password. If the cube is password protected, and you are not in a Kerberos environment, you must enter your user name and password.

   Specify whether to use Windows Authentication or a specific user name and password.

c. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Select a database.

   c. Select a cube from the database.

   d. Select the sheet tab to start your analysis.

Incompatible Measures and Dimensions

When you build views in Tableau using a Microsoft Analysis Services Cube you might see some fields highlighted in gray, or you might see a caution symbol on a field in the view with the message, "This measure is incompatible with one or more dimensions in this view." This occurs because it is possible to have measures and dimensions that don’t make a lot of sense when placed in the view together. For example, you may have a measure for Sales Quota. It won't make sense to place that measure against a dimension containing products if products don't have sales quotas.

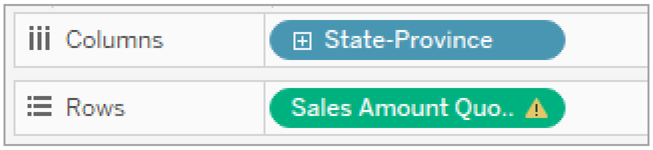


Tableau helps you figure out the dimensions and measure that can be used together in meaningful ways by highlighting unrelated dimensions and measures in gray. So in the last example, when we place Sales Quota onto a shelf, the products dimensions are highlighted in gray. Highlighted dimensions are not disabled and can still be added to the view. When you add an incompatible measure to the view, the measure is marked with a caution symbol. For more information on working with cubes, see **Cube Data Sources** on page 1085.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Microsoft PowerPivot

This article describes how to connect Tableau to a Microsoft PowerPivot database.

Before you begin

Before you begin, gather this connection information:

- SharePoint URL, or SharePoint, UNC, or local Excel file name

  If you plan to publish the workbook to Tableau Server, make sure to connect to a PowerPivot file on SharePoint. Tableau Server does not support connections to local PowerPivot files.

> **Note:** Tableau Desktop supports Microsoft PowerPivot 2010 and 2013.

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Microsoft PowerPivot**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Select whether to connect to a PowerPivot file using a SharePoint URL, a SharePoint UNC (file path), or a local Excel file.
   b. Select **Sign In**.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. The file name appears under **Select a File**. Note that there is one file per connection. Search for or select a perspective available in that file.
   c. Select the sheet tab to start your analysis.

Use PowerPivot perspectives

PowerPivot data may contain one or more perspectives. Perspectives are subsets of objects from the model that define sets of data. Typically perspectives are defined for a particular group of users or business scenario. Using perspectives can make it easier to navigate large data sources.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Microsoft SQL Server

This article describes how to connect Tableau to a Microsoft SQL Server database and set up the data source.

Note: Use the Microsoft SQL Server connector to connect to Microsoft SQL Server Parallel Data Warehouse (PDW), Microsoft Azure Synapse Analytics (formerly Azure SQL Data Warehouse), or Microsoft Azure SQL Database.

Before you begin

Before you begin, gather this connection information:

- Name of the server you want to connect to
- (Optional) Port number if you want to connect to a non-default port
- (Optional) Database name if you want to connect to a contained database
- Authentication method: Windows Authentication or user name and password
- Are you connecting to an SSL server?
- Do you want to set the database isolation level to read uncommitted data?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Microsoft SQL Server**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the name of the server you want to connect to.

       To connect to a non-default port, use the `<server name>, <port number>` format when entering the server name. For example, `ExampleServer, 8055`.
    b. (Optional) Enter a database name if you want to connect to a contained database.

c. Select how you want to sign in to the server. Specify whether to use Windows Authentication or a specific user name and password. If the server is password protected, and you are not in a Kerberos environment, you must enter the user name and password.

   Select the **Require SSL** check box when connecting to an SSL server.

d. Specify whether to **Read uncommitted data**. This option lets Tableau running at the read-uncommitted isolation level. Long queries from other transactions, including extract refreshes, can lock the database and delay Tableau's transactions. Select this option to allow queries to read rows that have been modified by other transactions even when they have not been committed yet. When this option is cleared, Tableau uses the default isolation level specified by the database.

e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

f. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.

   c. Under **Table**, select a table or use the text box to search for a table by name.

      You can also specify a stored procedure in the database. For more information about stored procedures, including a list of constraints specific to SQL Server databases, see **Use a Stored Procedure** on page 972.

Starting with Tableau version 2018.1, you can connect to spatial columns. For more information, see **Connect to Spatial Data in a Database** on page 1799.

d. Drag the table or stored procedure to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959. Starting with Tableau version 2018.1, you can use Custom SQL to perform advanced spatial analysis on spatial columns in Microsoft SQL Server. For more information, see **Use Custom SQL and RAWSQL to perform advanced spatial analysis** on page 1801.

Note: Tableau Desktop does not support the Microsoft SQL Server TIME data type. Fields of this data type are not imported and do not appear in Tableau Desktop. If included in stored procedures, TIME data type fields will not appear in Tableau Desktop. For more information, see **Use a Stored Procedure** on page 972.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Resolve "Worksheet Unavailable" error message

When you open a workbook, you may see a “Worksheet Unavailable” error message and then, when you select **Edit Connection**, you are prompted to provide your sign-in credentials. This error typically displays under one of two circumstances:

- You don't have access to a data connection used in the workbook.
- Your credentials are invalid, for example, your password has expired.

To resolve, verify that your sign-in credentials are correct. If they are, contact the Tableau administrator to ask about access to the data connection.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Connect to Spatial Data in a Database on page 1799

MonetDB

This article describes how to connect Tableau to a MonetDB database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MonetDB**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.

   b. Enter the user name and password.

   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

d. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select a schema.

c. Under **Table**, select a table or use the text box to search for a table by name.

d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

MongoDB BI Connector

This article describes how to connect Tableau to MongoDB Business Intelligence (BI) data and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password, if MongoDB authentication is enabled
- Database name associated with the user's credentials, if MongoDB authentication is enabled

- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

You must install and run the MongoDB Connector for BI, version 2.1 or later, before you can connect Tableau to your MongoDB BI data.

To get the MongoDB Connector for BI, sign in to the MongoDB website and download the connector. Follow the Installation instructions on the MongoDB website.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MongoDB BI Connector**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server.
   b. If MongoDB authentication is enabled, enter your user name with associated database name, and password. For information about the correct format to use, see **Sign in options** on the facing page. If MongoDB authentication isn't enabled, proceed to step c.

      Select the **Require SSL** check box when connecting to an SSL server. For more information about connecting with SSL to MongoDB, see Connecting Tableau to MongoDB on the MongoDB website.
   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   d. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Select the sheet tab to start your analysis.

Sign in options

User accounts in MongoDB are associated with specific logical databases in MongoDB. So when users log into MongoDB, they need to specify the database name associated with the user's credentials. This is done by passing special options in the username string. For example, if user `henrywilson` is associated with the `example` database, he enters the following in the **Username** field.

```
henrywilson?source=example
```

You can use the `mechanism` option to define the authentication mechanism if you want to use a mechanism other than the default, SCRAM-SHA-1. For example, if user `henrywilson` wants to connect to the `example` database using challenge/response as the authentication mechanism, he enters the following in the **Username** field:

```
henrywilson?source=example,mechanism=MONGODB-CR
```

For more information about authentication options, see the MongoDB website.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

MySQL

This article describes how to connect Tableau to a MySQL database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **MySQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password.

      Select the **Require SSL** option when connecting to an SSL server.
   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   d. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.

c. Under **Table**, select a table or use the text box to search for a table by name.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

OData

This article describes how to connect Tableau to an OData data source. Tableau connects to OData V1, V2, and V4.

Before you begin

Before you begin, gather this connection information:

- Server URL for the data you want to connect to
- Sign-in credentials (user name and password), if required

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **OData**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. In **Server**, type one of the following:
      - A service URL to access all feeds inside the service.
      - A feed URL to access a specific data feed.

b. From the **Authentication** dropdown, select the authentication mode for this connection.
   - If you select **Username and Password**, also type a username and password.
   - Otherwise, select **No Authentication**.

c. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   > **Note:** If you publish the data source to Tableau Server or Tableau Online, the data source name is limited to 255 characters. Longer names cause a `RepositoryException` error.

   b. Select the sheet tab to start your analysis.

   After you select the sheet tab, Tableau imports the data by creating an extract. For more information about extracts, see **Extract Your Data** on page 1014. Note that Tableau Desktop supports only extracts for OData.

OData connector support

- The OData connector supports only extracts, which means that some OData V4 functionality is not supported; for example, interactive query arguments such as `$expand` or `$select`.
- Tableau supports these OData versions and formats:
  - OData V1 and V2 – We support OData XML/Atom format for feed responses.
  - OData V4 and V4.01 – We support OData-JSON format for feed responses.
- Tableau assumes metadata documents for all supported OData services are in XML format.
- Collection-valued properties are not supported. Tableau interprets them as nulls.

Note:

Tableau can't guarantee that using a third-party OData service or feed with will result in successfully connecting and extracting data. Tableau provides a basic level of customer support with troubleshooting third-party OData connections. Tableau customer support doesn't have access to Apache Olingo library so you will need to see your system administrator for in-depth OData troubleshooting.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

OneDrive

This article describes how to connect Tableau to OneDrive data and set up the data source.

Before you begin

Before you begin, get the email address and password for your OneDrive account.

Note: The OneDrive connector supports only files owned by you, not those shared with you by another owner.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **OneDrive**. For a complete list of data connections, select **More** under **To a Server**. In the tab Tableau opens in your default browser, do the following:
    a. Enter your email address and then select **Next**.
    b. Enter your password and then select **Sign In**.
    c. Select **Yes** to grant Tableau access to your OneDrive files.
    d. Close the browser window when notified to do so.
    e. Search for or select the file to connect to, and then select **Connect**.

       Note: This connector might not support all file types. Unsupported file types are grayed out.
2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Select the sheet tab to start your analysis.

> **Note:** Tableau doesn't support pivot tables in OneDrive.

Use Data Interpreter to clean your data

If Tableau detects that it can help optimize your data source for analysis, it prompts you to use Data Interpreter. Data Interpreter can detect sub-tables that you can use and remove unique formatting that might cause problems later on in your analysis. For more information, see **Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter** on page 993.

Troubleshoot OneDrive issues

Connections to multiple accounts

When using web authoring or publishing to the web, you cannot use multiple OneDrive accounts in the same workbook. You can have multiple OneDrive account connections in Desktop.

Web authoring with Internet Explorer 11 and Edge

In Internet Explorer 11 and Edge, you cannot access a server using an unsecured connection (http). Use a secure connection (https) or switch to another browser.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Oracle

This article describes how to connect Tableau to an Oracle database and set up the data source.

> **Note:** Beginning with Tableau 2020.2, if you want to publish a data source or workbook using net_service_names to Tableau Server, be sure to review the changes described in the Knowledge Base article, Setting an Oracle Connection to Use TNSNames.ora or LDAP.ora.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and the Oracle service name and port, or the TNS name
- Authentication method: Integrated Authentication or user name and password
- Are you connecting to an SSL server? You must configure the Oracle client before you can use SSL with Tableau.

  If you are using version 2020.2 or later, see Require SSL for Oracle JDBC Connections.
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

JDBC driver versus OCI driver

Beginning with version 2020.2, Tableau recommends that you use the JDBC driver (available from the Driver Download page) with this connector. The newer JDBC driver resolves some limitations found with the OCI driver, such as no spatial support or Kerberos delegation on Linux.

> **Note:** If you experience issues using the JDBC driver for this connector, an OCI driver is still available and can be used instead. See the Driver Download page for instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Oracle**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the server name or the TNS name.
      - If you enter the server name, you must enter the Oracle service name and port number.

        > **Note:** To find the correct service name, open the tnsnames.ora file (in the ORACLE_HOME/network/admin directory by default) and search for the **net_service_name** section. Use the value listed for **service_name**.

      - If you enter the TNS name, do **not** enter the Oracle service name and port number.

   b. Select how you want to sign in to the server. Specify whether to use Integrated Authentication or a specific user name and password.

      Select the **Require SSL** check box when connecting to an SSL server.

   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

   d. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

Note: Search is case-sensitive.

c. Under **Table**, select the search icon or enter the table name and select the search icon, and then select the table.

> **Note:** Search is case-sensitive.

Tableau Desktop also supports connecting to Oracle table functions, which appear under **Stored Procedures** in the left pane of the Data Source page.

Starting with Tableau version 2020.2, you can connect to spatial columns. For more information, see **Connect to Spatial Data in a Database** on page 1799.

d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Use net services definitions in Tableau

To use your net services definitions in Tableau, you must set either `TNS_ADMIN` or `ORACLE_HOME` as an environment variable. To set `TNS_ADMIN` as the environment variable use the full path of the directory that contains the `tnsnames.ora` file. To set `ORACLE_HOME` as an environment variable use the path to the main Oracle directory.

Migrate workbook from Oracle OCI connector to Oracle JDBC connector

If you create a workbook using the Oracle OCI connector with SSL and later try to open it using the JDBC-based Oracle connector, Tableau won’t be able to find the corresponding SSL certificate.

To migrate a workbook

1. Convert the certificate from PKCS12 format (used by Oracle Wallet) to JKS format. Use this command:

   keytool -importkeystore -srckeystore PFX_P12_FILE_NAME -srcstoretype pkcs12 -srcstorepass PFX_P12_FILE -srcalias SOURCE_ALIAS -destkeystore KEYSTORE_FILE -deststoretype jks -deststorepass PASSWORD -destalias ALIAS_NAME
2. Modify oracle.properties to customize this connection as follows:

   javax.net.ssl.trustStore = <new jks file path>
   javax.net.ssl.trustStoreType = JKS
   javax.net.ssl.trustStorePassword = <new jks file password>

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

- Tableau and JDBC - Find tips for using and troubleshooting JDBC drivers with Tableau.
- Require SSL for Oracle JDBC Connections - Add trusted SSL certificates to your Oracle JDBC connection.
- Troubleshooting Oracle Connection Errors - Review the Tableau Knowledge Base article about how to troubleshoot errors.

Oracle JDBC Connections with SSL

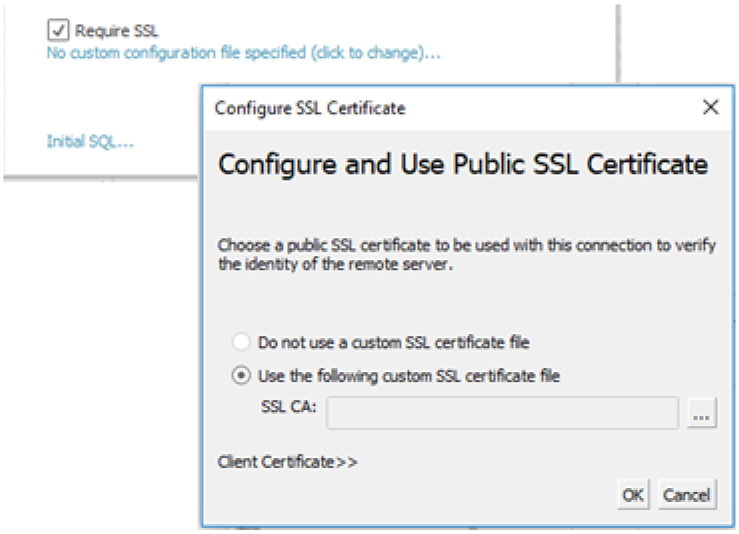
If you are connecting to an SSL (security socket layer) server, check the **Require SSL** check box.

Beginning in 2020.2, Oracle uses a JDBC connector. If your Oracle data source is configured to use SSL one-way or two-way authentication, you can include the appropriate certificates and/or key to connect.

Require SSL

If your connection requires SSL, do the following:

1. From the connector dialog, click **Require SSL**.



2. Configure SSL as appropriate for your connection:
   - If you are using self-signed certificates or a certificate from signed by a private CA, you may need to configure trust for one-way SSL connections. See Configure one-way SSL for Oracle.
   - If you are using two way SSL, you will need to configure your client keys. See Configure two-way SSL.

Configure one-way SSL for Oracle

To configure trust for your server certificate, see general instructions for JDBC connectors here: Configure One-Way SSL for JDBC Connections.

If you decide to use properties files to configure trust, then use the following properties for the Oracle JDBC driver.

javax.net.ssl.trustStore=C:\\My_Folder\\truststore.jks
javax.net.ssl.trustStoreType=JKS
javax.net.ssl.trustStorePassword=*password*

> **Note:** To publish a workbook to Tableau Server, both the Tableau Desktop and Tableau Server computers must have copies of the same Oracle properties file and truststore.jks file.

Configure two-way SSL

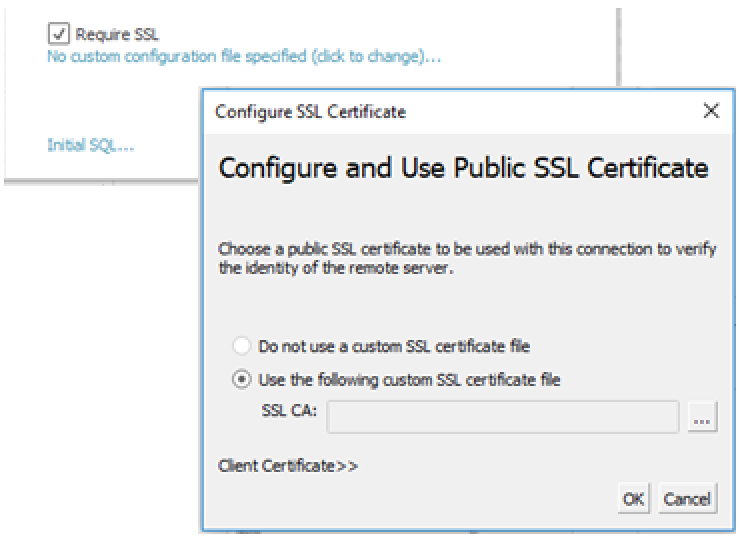
There are two options for configuring two-way SSL support with an Oracle connection:

- Embed client certificates and private key in the data source
- Use a properties file to configure the client certificate and private key

Option 1: Embed client certificates and private key in the data source

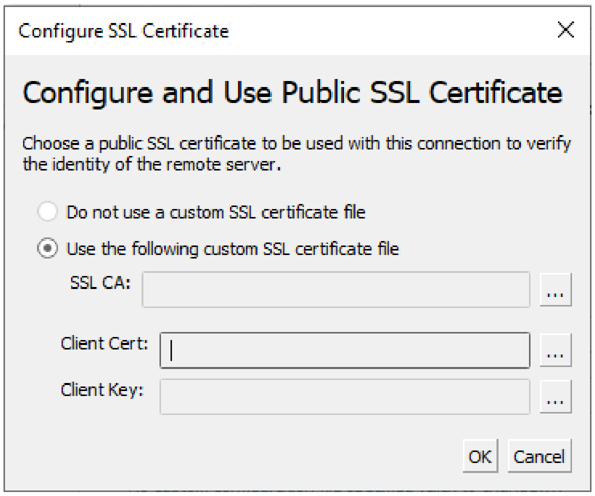
To embed an SSL certificate, client certificate, and client key:

1. Follow the steps in **Configure one-way SSL for Oracle** on the previous page to configure server trust, if necessary.
2. Click **Client Certificate >>**.



3. Specify the locations for the client certificate and client key.
   The certificate and private key should be in unencrypted PEM format. See instructions

later in this document for suggestions on converting to the correct format.



Option 2: Use a properties file to configure the client certificate and private key

To use a properties file for certificate and key configuration:

1. Modify the oracle.properties file to reflect the key store and trust store settings. Use the following as an example. Be sure to replace “My_Folder” for the location of your files and “<password>” with your own password. Paths on OSX and Linux need only a single "/" to separate.

   javax.net.ssl.trustStore=C:\\My_Folder\\truststore.jks
   javax.net.ssl.trustStoreType=JKS
   javax.net.ssl.trustStorePassword=<password>
   javax.net.ssl.keyStore=C:\\My_Folder\\keystore.jks
   javax.net.ssl.keyStoreType=JKS
   javax.net.ssl.keyStorePassword=*password*
2. Save oracle.properties to the <Document>\MyTableau Respository\Datasources folder.
3. Copy oracle.properties, truststore.jks, and keystore.jks to the Tableau Server computer. This allows the workbook to be published to the Tableau Server.

If you use this method, you don't need to upload certificates and key via the SSL dialog. Instead, simply click **Require SSL**. This causes the Oracle JDBC connector to read the locations of the key store and trust store from oracle.properties file.

Publish a workbook

To publish a workbook to Tableau Server, be sure the following are true:

- If you use a properties file, both the Tableau Desktop and Tableau Server computers have copies of the same Oracle properties file, truststore.jks, and keystore.jks.
- The server uses the same Oracle JDBC connector as the Tableau Desktop computer.
- If you used the embedding option, select “embed password” option when you publish.

Convert key formats

There are several formats for storing SSL keys, including:

- Oracle Wallet
- JKS - Use this format if you want to configure SSL trust using properties files.
- PEM - Use this format if you want to embed the keys in the data source using Tableau interface.

The following are some suggested commands you can type at the command-line prompt to convert keys to the correct format. See the Oracle documentation for the more details.

To export server certificate from an Oracle wallet to PEM file:

orapki wallet export -wallet wallet_location -dn certificate_dn -cert output_certificate_filename.pem

To export server certificate and client keys from Oracle wallet to JKS:

orapki wallet pkcs12_to_jks -wallet ./ -pwd mypasswd -jksKeyStoreLoc ./ewalletK.jks -jksKeyStorepwd *password* -jksTrustStoreLoc ./ewalletT.jks -jksTrustStorepwd *password*

To export server certificate from JKS trust store to PEM file using keytool and OpenSSL:

1. Export JKS format to PKCS12 format:

   keytool -importkeystore -srckeystore truststore.jks -destkeystore truststore.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass "*password*" -deststorepass "*password*"
2. Get the PEM file from PKCS12:

   openssl pkcs12 -in truststore.p12 -out serverca.pem

To export server certificate and client keys from JKS trust store to PEM file using keytool and OpenSSL:

1. Export the certificate from the trust store as follows.
   a. Export JKS format to PKCS12 format:

      keytool -importkeystore -srckeystore truststore.jks -destkeystore truststore.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass "*password*" -deststorepass "*password*"
   b. Get the PEM file from PKCS12:

      openssl pkcs12 -in truststore.p12 -out serverca.pem
2. Export the client key and client certificate from the key store as follows.
   a. Use keytool to transform JKS format to PCKS12 format:

      keytool.exe -importkeystore -srckeystore keystore.jks -destkeystore keystore.p12 -srcstoretype JKS -deststoretype PKCS12 -srcstorepass "*password*" -deststorepass "*password*"
   b. Use OpenSSL to export the client certificate:

      openssl pkcs12 -in keystore.p12 -nokeys -out clientcert.pem
   c. Use OpenSSL to export the client key:

      openssl pkcs12 -in keystore.p12 -nocerts -out clientkeyEncrypted.pem
   d. Use OpenSSL to transform the encrypted version to PKCS8 format:

      openssl pkcs8 -topk8 -nocrypt -in clientkeyEncrypted.pem -out clientkey.pem

See also

- Configure SSL for JDBC Connections - More general information about installing trusted SSL certificates for JDBC connections.

Oracle Eloqua

This topic describes how to connect Tableau to Oracle Eloqua data and set up the data source. It also describes how date range selections can impact performance.

> **Note:** For the most up-do-date information on this connector, see the Oracle Eloqua help topic for the current version of Tableau.

Before you begin

Before you begin, gather and confirm this information for connecting:

- You must be a member of the following Eloqua security groups:
  - Advanced Users - Marketing
  - API Users
- Company Name
- Username
- Password

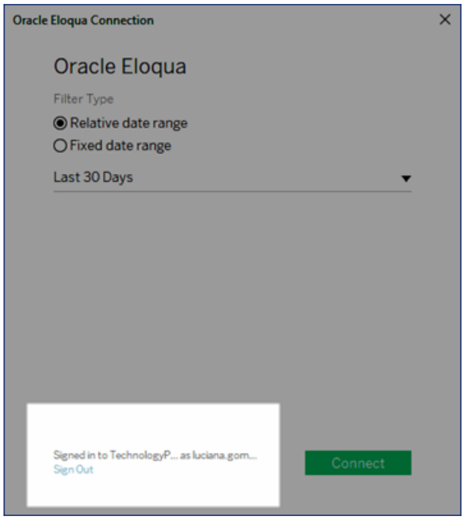
Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Oracle Eloqua**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter **Company Name**, **Username**, and **Password**, and then select **Sign In**.
   b. Select **Filter Type**: **Relative date range** or **Fixed date range**, and then select or specify the range.
   c. Select **Connect**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or Eloqua administrator.

   > After your credentials are verified, the Eloqua connector will only prompt you to authenticate when necessary. The **Sign Out** option will start the

authentication process again.



2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under **Table**, select a table and drag it to the top of the canvas.

   c. Select the sheet tab to start your analysis.

      After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for Eloqua. You can update the data by refreshing the extract. For more information, see **Extract Your Data** on page 1014.

      Creating extracts may take some time depending on the amount of data that is included.

Date range selections can impact performance

It's tempting to gather as much data as possible when you do an analysis, however, retrieving records from Eloqua can be time-consuming. Tableau doesn't know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance. You can modify the date range on the Data Source page. Under **Connections**, select the drop-down arrow for the connection and select **Edit connection**.

To give you a rough idea of how much time it might take to retrieve data from Eloqua, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records. The connector retrieves 1000 records per request.

Number of records	**Time to retrieve**
50,662 records	3 minutes 17 seconds
95,214 rows	6 minutes 14 seconds
194,679 rows	12 minutes 55 seconds

Eloqua limits

Field limits

Some Eloqua tables can be customized to have more than 250 fields. Tableau currently retrieves the first 250 fields and ignores the rest. When you select an Events, Custom Objects, Contracts, or Accounts table, Tableau will display a warning if there are more fields than Tableau can retrieve.

Daily table request limits

Each table requested by Tableau counts against a daily Eloqua request limit of 2000. When the limit is exceeded, Eloqua operations might choose to disable Tableau, in which case, they will contact the account owners. You can monitor your daily usage in Eloqua using the Eloqua Marketing Operations Center.

Incremental Refresh

For all the activity labels (with the exception of "Activities- all"), you can now implement incremental refreshes. For more information, please see **Refresh Extracts** on page 1041.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Oracle Essbase

This article describes how to connect Tableau to an Oracle Essbase database.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

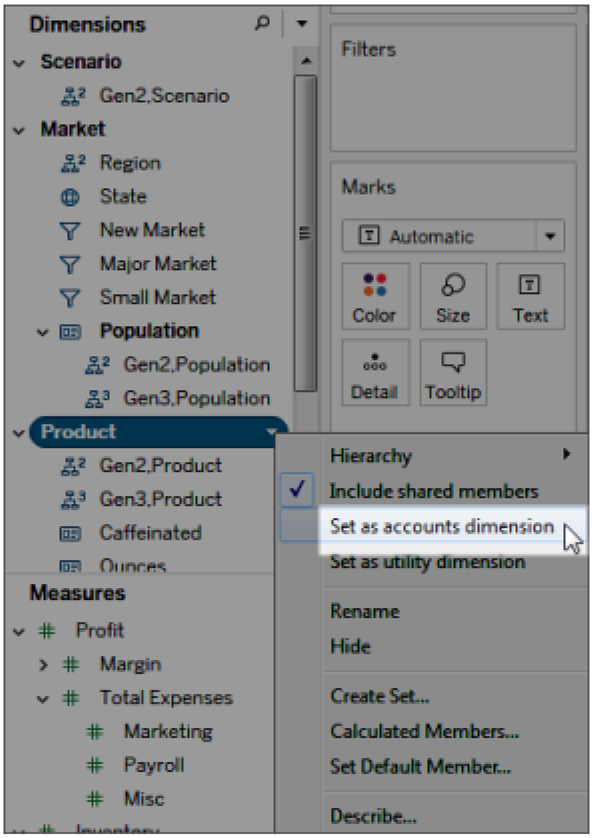
Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Oracle Essbase**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter your user name and password to sign in to the server, and then select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Search for or select an application.
   c. Search for or select a database from your application.
   d. Select the sheet tab to start your analysis.

Set an accounts dimension

In some case, the accounts dimension for your data source can appear in the Dimensions area of the Data pane. This might occur if there is an error in the cube and another field is identified as the accounts dimension or there is no accounts dimension set at all. The accounts dimension defines the fields that are included as measures. To correct this error, right-click the field and then select **Set as Accounts Dimension** from the context menu.



Build views with Oracle Essbase

When Tableau is connected to an Oracle Essbase data source, there are three important features that you should know about:

Generations and Levels

In Tableau, you can work with either the generations or the levels of a dimension. The generations of a dimension are all members that are an equal distance from the root of the dimension. The levels are all members that are an equal distance from the leaves of the dimension. For balanced dimensions, you'll typically want to work with generations. However, if your dimension is ragged, then it may make more sense to navigate using levels.

By default, the generations of each dimension are listed in the Data pane. When you drag a dimension to a shelf, all generations that are ancestors of the selected generation (all generations that are above it in the hierarchy) are automatically included in the placement.

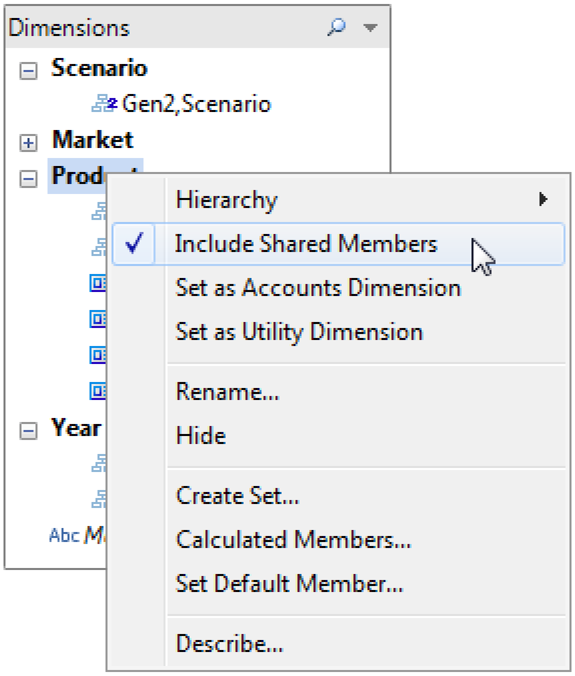
If you would rather navigate using the levels of a dimension, right-click the name of the dimension and then select **Hierarchy** > **Levels**.

> If you are using the same dimension in multiple worksheets, you can use levels in one worksheet and generations in another worksheet simultaneously. Furthermore, you can mix generations and levels from different dimensions in the same worksheet.

Shared Members

Shared members are dimension members that appear in more than one place in a hierarchy. For example, Diet Coke might be part of the product generation. But it might be shared by both the diet colas branch and the colas branch of the product hierarchy above it. In the database, however, the data about Diet Coke is stored just once.

By default, Tableau includes shared members in all generations (or levels) of a dimension. This means that a shared member might appear multiple times in a table. If you choose to exclude shared members, they will appear only once in a table. By default, shared members are included for all dimensions. To exclude shared members for a given dimension hierarchy, right-click the dimension name in the Data pane and select **Include Shared Members** from the menu.



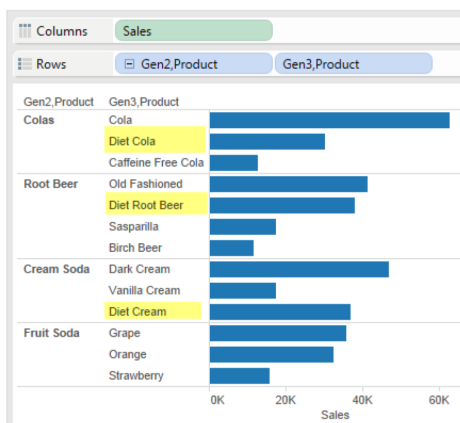
The figure below shows part of a data view where shared members are included (left) and excluded (right). Notice that diet drinks are shared members.

Includes Shared Members

The diet drinks are listed in both the Diet Drinks hierarchy and their respective drink hierarchies.

Does Not Include Shared Members

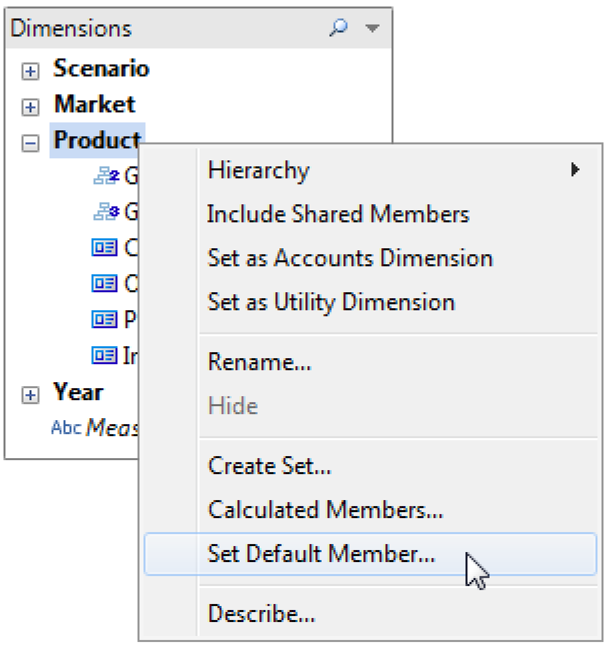
The diet drinks are only listed once, in their respective drink hierarchies.



Setting the Default Member

All multidimensional data sources have default members that are set when the data source is first built. If you find that you are creating filters all the time to look at the same specific data, you may find it useful to change the default member. For example, if you are the regional manager for the Western region in a company and you only want to look at your region’s numbers, you can set the default member to the Western region.

To change the default member in Tableau, right-click a dimension hierarchy and select **Set Default Member**.



In the subsequent dialog box, select from the following options:

- Default member defined on cube – uses the default member that was defined when the cube was built. This is the default setting in Tableau.
- (All) member for the hierarchy – uses the ALL member for the selected hierarchy as the default member.
- Selected member – uses the member that you select in the bottom half of the dialog box as the default member.

The default member determines how you view the cube and so is much more powerful than applying filters. All fields will be calculated based on the default member you select. In addition, these default member settings are saved with the connection.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Oracle NetSuite

This article describes how to connect Tableau to Oracle NetSuite and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Environment Account ID.
- Authentication information

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Oracle NetSuite**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the Application ID.
   b. Enter the Authentication information.
   c. Select **Sign In**.
      Note: If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, contact your database or network administrator.
2. On the data source page, complete the following steps:

   a. (Optional) Select the data source name at the top of the page, and then enter a unique data source name.

b. From the Database drop-down list, select a database.

c. Under Table, select a table or use the search field to search for a table by name.

d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Pivotal Greenplum Database

This article describes how to connect Tableau to a Pivotal Greenplum Database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Pivotal Greenplum Database**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the name of the database you want to connect to.
   b. Enter the user name and password.
   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in

to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

d. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Select and drag a table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Support for spatial columns

Starting with Tableau version 2019.2, you can connect to spatial columns. For more information, see **Connect to Spatial Data in a Database** on page 1799.

For Pivotal Greenplum Databases with spatial columns, we recommend that you use the PostgreSQL driver that is installed on Windows by default with Tableau Desktop. (Tableau already uses the PostgreSQL driver on macOS.) To use the PostgreSQL driver, you must uninstall the Pivotal Greenplum Database driver on your Windows computer, if it's installed.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

PostgreSQL

This article describes how to connect Tableau to a PostgreSQL database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- Authentication method: Integrated Authentication or user name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** Beginning with Tableau 2020.4, you can connect to PostgreSQL 12 (or higher) databases from Tableau. A new JDBC driver for PostgreSQL supports these connections and is included with all Tableau products on Windows.
> On Linux and Mac computers, you'll need to install a driver for this connector when you use it with Tableau Desktop or Tableau Prep Builder. See instructions on the Driver Download page.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **PostgreSQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the name of the server that hosts the database that you want to connect to.

   b. Enter the name of the database.

   c. Select how you want to sign in to the server. Specify whether to use **Integrated Authentication** or **User Name and Password**. If the server is password protected, and you are not in a Kerberos environment, you must enter the user name and password.

   > **Note:** If you're using a Mac, and it is not attached to the domain correctly, the Mac won't know that Kerberos is being used in the domain, and the **Authentication** drop-down list won't be available.

   Select the **Require SSL** check box when connecting to an SSL server.

   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

   e. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. Under **Table**, select a table or use the text box to search for a table by name.

   Starting with Tableau version 2019.2, you can connect to spatial columns. For more information, see **Connect to Spatial Data in a Database** on page 1799.

   c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Presto

This article describes how to connect Tableau to a Presto database and set up the data source.

Tableau connects to Presto 141t from Teradata and to Presto on-premises and Amazon EMR Presto instances for Presto version 0.148.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Catalog
- Authentication method:
  - Username
  - LDAP

> **Note:** LDAP authentication is available from Tableau Desktop version 10.3.2 forward.

- Sign-in credentials. Your choices depend on the authentication method you choose, and can include the following:
  - User name
  - Password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Presto**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server you want to connect to.
   b. Enter the catalog name.
   c. Select the **Authentication** method: **Username** or **LDAP**, and enter the credentials you're prompted for.

      Select the **Require SSL** check box when connecting to an SSL server.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   e. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data

source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- One-Way SSL for JDBC Connections - Configure trust for a self-signed or CA certificate when you use one-way SSL with a JDBC connector.

Progress OpenEdge

This article describes how to connect Tableau to a Progress OpenEdge database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Progress OpenEdge**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database you want to connect to.
   b. Enter the name of the database.
   c. Enter the user name and password.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   e. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table or use the text box to search for a table by name.

c. Drag the table to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Qubole Presto

This article describes how to connect Tableau to Qubole Presto and set up the data source.

> **Note:** As of 2020.3, the Qubole Presto connector is JDBC-based.

Before you begin

Before you begin, gather this connection information:

- End Point- end point URL based on the region where you have the Qubole account
- Catalog name ('hive' is the default)
- Cluster label
- Password (API token)
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Qubole Presto**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

a. Enter the **End Point**.

> **Note:** Advanced user can use additional properties in the **Endpoint** field; for example: `https://api.qubole.com;`
> `LogLevel=6;LogPath=C:\\Users\\User\\Desktop`
> For more information, see the "Additional Properties (Optional)" section of the Setting the JDBC Connection String topic on the Qubole website.

b. Enter the **Catalog** name.
c. Enter the **Cluster Label**.
d. Enter the **Password**, which is an API token.
e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
f. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Schema**, select a schema or use the text box to search for a schema by name.
   c. Under **Table**, select a table or use the text box to search for a table by name.
   d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Salesforce CDP

This article describes how to connect Tableau

to Salesforce CDP.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection

1. Start Tableau and under **Connect**, select **Salesforce CDP**.
2. In the Authentication field, select **OAuth**.
3. Click **Sign In**.
4. If prompted, enter your user name and password.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Salesforce

This article describes how to connect Tableau to Salesforce.com data and set up a data source.

Notes:
- Beginning with Tableau Desktop version 2020.4, OAuth is used for all sign-ins. Signing in with username and password is not an available option.
- Tableau Prep Builder version 2020.2.1 supports connecting to Salesforce.com data. However, not all connector options described in this article are supported. For information about what is supported by Tableau Prep Builder, see Connect to Salesforce data in the Tableau Prep Builder help.

Before you connect

Before you begin, gather this connection information:

- Credentials for signing in to Salesforce.com using OAuth

Make the connection and set up the data source

1. Start Tableau and under Connect, select **Salesforce**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Enter your sign-in credentials for Salesforce.com.

    b. Select **Log In**.

    c. In the Allow Access dialog box, select **Allow**.

    If the connection is unsuccessful, verify that the authentication information is correct. If the connection continues to fail, your computer is having trouble locating the server or you may not have permission to access the data. Contact your network administrator or Salesforce administrator.

2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

    b. Select a standard connection or create your own custom connection.

    Under **Standard Connection**, you can select from a list of predefined queries, which represent commonly used objects, and drag it to the canvas. Or, under **Table**, you can select a single table, which includes any custom objects your company has created, and drag it to the canvas.

c. Select the sheet tab to start your analysis.

Work with a subset of columns

Beginning with Tableau Desktop versions 2020.1, you can work with a subset of columns.

If you want to select a subset of columns to work with, double-click **New Custom SQL** and use this syntax:

```
SELECT <column1, column2, […]columnN> FROM <table>
```

For example, to work with three columns from the Org table — Account Rep, Department, and Cost Code — you could type:

```
SELECT AccountRep, Department, CostCode from Org
```

> **Note:** This connector doesn’t currently support SOQL containing relationship queries. Any SOQL containing relationship queries will fail.

For more information, see **Connect to a Custom SQL Query** on page 959.

Refresh Salesforce data

After you click the sheet tab, Tableau imports the data by creating an extract. Tableau Desktop supports only extracts for Salesforce.

> **Note:** The initial extract may take some time depending on the amount of data that is included.

After the initial extract, you can do a full or incremental refresh of the extract. Incremental refreshes for Salesforce data include any Salesforce object that has been added or changed since the last incremental refresh. Results returned are limited to the previous 30 days.

To refresh, select **Data**> [data source name] > **Extract**> **Refresh (Incremental)** | **Refresh (Full)**.

Use Dashboard Starters

You can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics on Tableau Desktop and Tableau Online. For more information, see **Dashboard Starters for Cloud-based Data** on page 2750.

Use joins in Salesforce

When you set up the data source on the data source page, when you drag a table to the canvas, the list of tables you can select from to join is filtered. You can create joins on any string fields and on fields that are constrained references between tables. (Only left and inner joins are supported.) You can also add joins later. For more information, see **Join Your Data** on page 883.

Cross-database join limitations with Salesforce

Salesforce also supports combining tables from different databases using a cross-database join. For more information, see the "Combine tables from different databases" section in **Join Your Data** on page 883.

When a Salesforce connection is included in a multi-connection data source, the data source won't support the following features:

- Live connections
- Extract filters
- Incremental updates to the extract
- Viewing your data in the data grid

Troubleshoot Salesforce connections

Tableau leverages the various force.com APIs to connect to Salesforce.com, Force.com, and Database.com data. These services have some restrictions on the type of data and the amount of data that you can access at a given time. If you are having trouble connecting, the following list of common restrictions may help you find a solution.

Company account and user profile configuration

In order for Tableau to connect to Salesforce data, all of the following must be enabled on both your company's account and your user profile:

- SOAP API for signing in
- REST API for getting meta data
- BULK API for downloading objects
- REST API for downloading objects that the BULK API does not support
- Replication SOAP APIs for retrieving changes in the data

To optimize for performance and ensure the APIs are available for all of their customers, Salesforce.com balances the load by limiting the number of concurrent API requests as well as limiting the total number of API requests overall. An error occurs if these limits are reached while connected using Tableau. See the Salesforce.com developer documentation to learn more about these limits.

> **Note:** Some editions of Salesforce may not allow API access at all. Contact your Salesforce administrator to verify that your account has access to the above list of APIs.

Errors during extract

When you connect to Salesforce using Tableau, the data is automatically extracted into a Tableau Data Extract file. In some cases, certain fields cannot be extracted because of character limits. Specifically, text fields that are greater than 4096 characters and calculated fields will not be included in the extract. If you have calculated fields in your data, you will need to recreate them in Tableau after creating the extract.

In addition, the Force.com API restricts queries to 10,000 total characters.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- 5 Ways to Maximize Your Salesforce Data - Read the Tableau whitepaper (registration or sign in required).

Splunk

This article describes how to connect Tableau to Splunk data and set up a data source.

> **Note:** The Splunk ODBC connector is compatible with Splunk Enterprise 5.x and 6.

Before you begin

Before you begin, gather this connection information:

- Server URL, including the name of the database, if there are multiple databases
- User name and password

- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Splunk**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the server URL. You should use HTTPS to connect to Splunk data.

       If your server contains multiple databases, enter the name of a specific database on the server.

       > **Note:** Database names are case sensitive.

    b. Enter your user name and password.
    c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    d. Select **Sign In**.

       If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. Under **Saved Search**, enter a saved search name in the text box or select a

saved search from the list, and drag it to the box at the top of the data source page.

c. Select the sheet tab to start your analysis.

Join limitations with Splunk

While you cannot create a join between Splunk tables, you can combine Splunk data from multiple tables by doing one of the following:

- **Set up and connect to a saved search:** Set up a saved search in Splunk that returns all the tables you need in a single search result. Then connect to this saved search from Tableau Desktop.
- **Use data blending:** Set up a data source for each Splunk table you need, then use data blending to combine the data. For more information, see **Blend Your Data** on page 910.
- **Create an extract:** Create an extract of each Splunk table and then join the extracts together using a cross-database join. For more information, **Extract Your Data** on page 1014 and **Join Your Data** on page 883.

However, you can create a cross-database join between Splunk data and data from different databases (or files) with one exception. After adding a new connection to the Tableau data source, you must double-click the table in the left pane to add it to the canvas. You cannot drag the table from the left pane to the canvas as you normally would when creating a cross-database join. For more information about cross-database joins, see **Join Your Data** on page 883.

Note: Creating a cross-database join between Splunk tables is not supported. A cross-database join between Splunk tables causes all the connections in the Tableau data source to disappear except for the last Splunk connection.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

SAP HANA

This article describes how to connect Tableau to an SAP HANA database and set up the data source.

Note: This connector uses JDBC drivers for Mac, Linux, and Windows. The JDBC driver for Windows is new in Tableau version 2021.1. JDBC drivers for Mac and Linux were introduced with Tableau version 2020.2. JDBC driver version 2.3.4 is the recommended version.

Before you begin

Before you begin, gather this connection information:

- Are you making a single-node or multi-node connection?
  - SingleNode: Name of server that hosts the database you want to connect to, and the port number if you're using a non-default port
  - MultiNode: Names of servers and port numbers that host the databases you want to connect to
- Authentication method:
  - Windows: Windows Authentication or user name and password
  - Mac: Kerberos or user name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SAP HANA**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Select the type of connection:
        - **SingleNode** - Enter the name of the server that hosts the database you want to connect to, and the port number if you're using a non-default port.
        - **MultiNode** - Enter the host name and port number of each server, separated by a comma. For example:

```
host1:30015,host2:30015,host3:30015
```

For more information about SAP HANA support for failover, see Configuring Clients for Failover on the SAP Help Portal.

b. Specify how you want to sign in to the server:

On Windows:

- Select **Use Windows Authentication** if your environment supports single sign-on (SSO). For more information about single sign-on and environment requirements, see **Support for SAP HANA single sign-on (SSO)** on page 615.
- Or, select **Use a specific username and password**. If the server is password protected and you are not in a Kerberos environment, you must enter the user name and password.

On Mac: Select **Kerberos** or select **Use a specific username and password**.

Select the **Require SSL** option when connecting to an SSL server.

c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

d. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.

c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

Tableau Desktop also supports connecting to SAP HANA table functions, which appear under **Stored Procedures** in the left pane of the Data Source page.

d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

By default, column labels are displayed instead of column names.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

> **Note:** When you create a self-join with an analytic view, if one of the tables contains variables, make sure that the table with variables is on the left. Otherwise, the join may not return the expected results.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Convert an SAP HANA level hierarchy to a user hierarchy

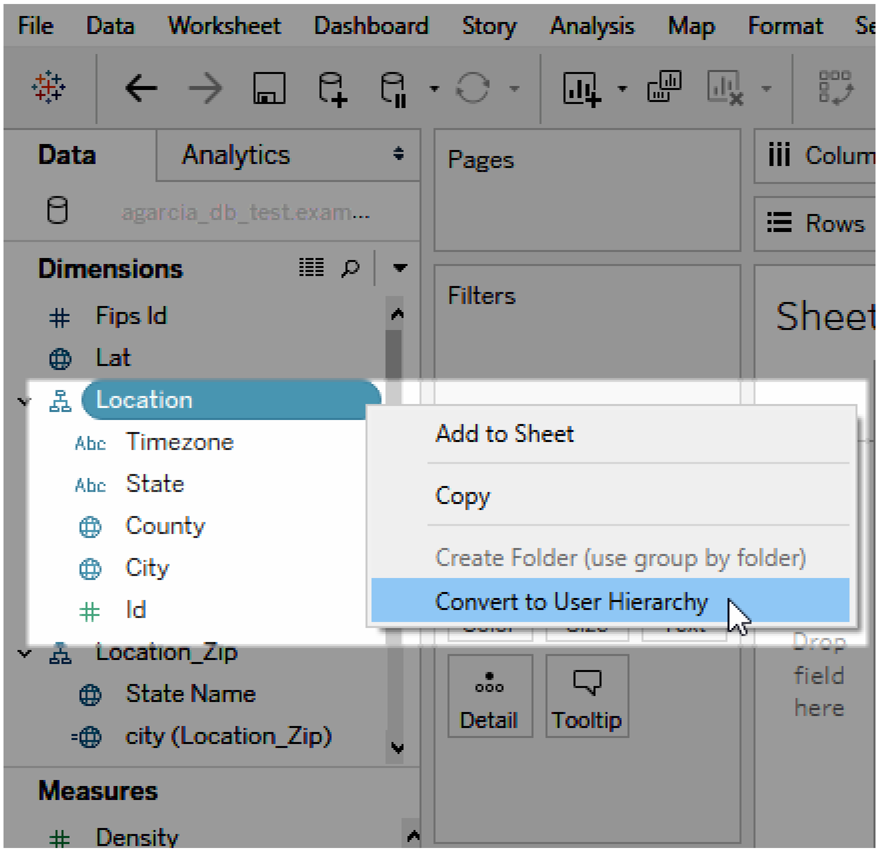
In all Tableau versions, when you create an SAP HANA data source or workbook, you can create your own hierarchy in Tableau that's solely for use in Tableau.

Starting in Tableau 2019.4, in addition to support for creating your own hierarchy for use in Tableau, Tableau now supports level hierarchies created in SAP HANA. This means that when you connect Tableau to an SAP HANA calculation view with a level hierarchy, the structure of the SAP HANA level hierarchy appears in the Data pane.

If you want to modify the SAP HANA level hierarchy in Tableau, you need to convert the SAP HANA level hierarchy to a user hierarchy. After you convert an SAP HANA level hierarchy to a user hierarchy, you can't reverse it. If you decide you want to use the SAP HANA level hierarchy instead of the user hierarchy, you need to start over.

To convert to a user hierarchy, follow these steps:

1. To the left of the sheet, in the **Data** pane, select the hierarchy, and then select the drop-down menu.
2. In the drop-down menu, select **Convert to User Hierarchy**.



Customize JDBC connections

On Mac and Linux, SAP HANA uses a JDBC connection, which you can customize using a properties file. For more information, see Customize JDBC Connections Using a Properties File in Tableau Community.

Install trusted SSL certificates on a Mac

If you want to use SSL/TLS to encrypt communications between Tableau Desktop and your database, you may need to add certificates to the Java certificate trust store. This is required if your database uses self-signed certificates.

> **Note:** Beginning with Tableau version 2020.2, you can import a custom certificate into the "System" keychain. For details, see One-Way SSL for JDBC Connections.

To connect Tableau Desktop to your database using SSL/TLS, follow these steps:

1. Start the Terminal application, and navigate to your Java Home directory, which is usually located at /Library/Internet Plug-Ins/JavaAppletPlugin.plugin/Contents/Home, unless you have installed the JDK or set the JAVA_HOME environment variable.

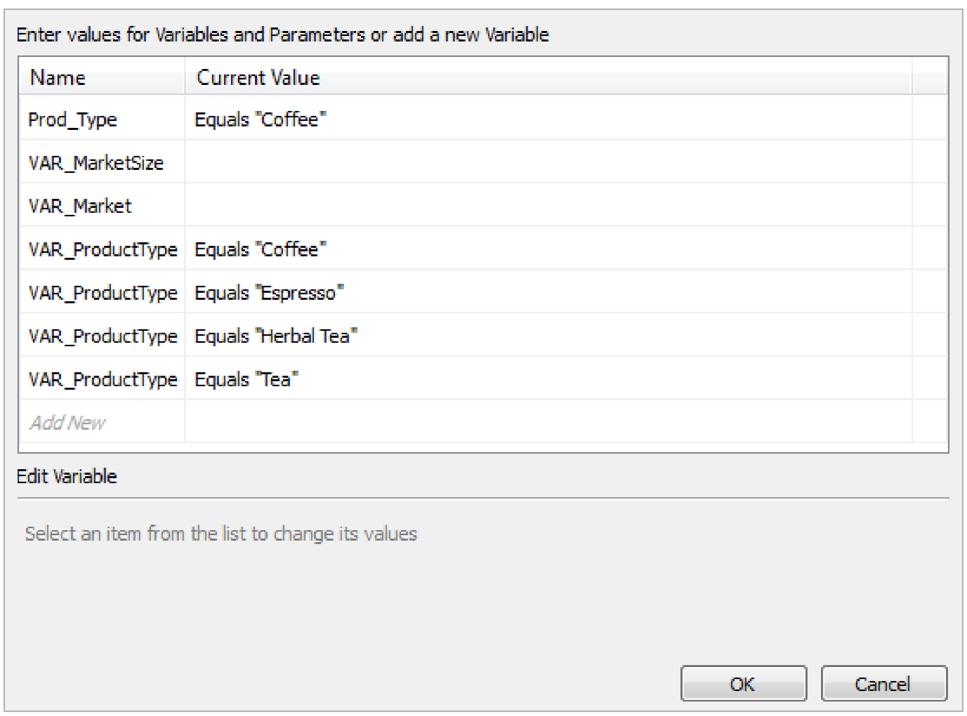
2. Run the following command. (You might need to use the `sudo` Terminal command, which is used by administrators to execute commands as a different user, such as root).

   ```
   bin/keytool -importcert -keystore cacerts -alias <cer-
   tificate name>  -file <path to certificate file>
   ```

For more information on installing trusted certificates, consult the documentation for your Java Runtime Environment.

Select variables and input parameters

If the table you use includes required or optional variables or parameters, the Variables and Input Parameters dialog box opens.



- Required variables and parameters display their current value or ***Required**.
- Optional variables and parameters display their current value or are blank.

- Prompt for the variable when the workbook is opened by selecting the check box under **Prompt**.

Select a variable or parameter and type or select a value for it. Repeat for all required values and any optional values that you want to include, and then select **OK**.

> **Note:** You can use SQL query-based SAP HANA prompts.

Support for SAP HANA single sign-on (SSO)

When SAP HANA is configured to support single sign-on (SSO), after you sign in to the SAP HANA server, you can access data, and publish data sources and workbooks to Tableau Server, without having to re-enter your user name and password. And, you can publish a data source or workbook so that other users with SSO can access the published data sources and workbooks without having to enter their user names and passwords.

To use SSO, you need to publish data sources and workbooks to Tableau Server with **Viewer credentials** authentication. To enable **Viewer credentials** authentication, you must connect to SAP HANA using Windows Authentication on a Windows computer or Kerberos on a Mac. For more information, see **Publish to the Server** below.

Important: Your environment must be correctly configured to support SSO for SAP HANA:

- Tableau Desktop requires SAP HANA driver version 1.00.85 and later.
- Tableau Server must be configured to support SSO for SAP HANA. For information, see Configure SAP HANA for Single Sign-On in the Tableau Server Help.

Sign in to the server

When SSO is set up in your environment, to sign in to SAP HANA, simply enter the server name, select **Use Windows Authentication** on a Windows computer or **Kerberos** on a Mac, and then click **Sign In**.

Publish to the Server

In an SSO environment, users don’t have to enter their credentials to access a data source or a workbook if you publish it to Tableau Server with the correct authentication mode.

To enable SSO when you publish a data source or workbook to Tableau Server, follow these steps:

1. Select **Server**> **Publish Data Source** or **Server** > **Publish Workbook**.
2. Under Data Sources, click **Edit**.
3. In the Manage Data Sources dialog box, do the following:
    1. Select the **Publish Type**: **Embedded in workbook** or **Published separately**.
    2. Select **Viewer credentials** for **Authentication**.
4. Click **Publish**.

How to refresh HANA extracts when you use SSO

When you use single sign-on (SSO) to connect to SAP HANA, because of authentication restrictions, you can't schedule an extract refresh when you publish the data source or workbook to a server. You can, however, use the Tableau Data Extract Utility to automate the extract refresh task.

For example, the following command refreshes an SAP HANA extract named "mydatasource" that has been published to Tableau Server. This command specifies the following:

- The name of your Tableau Server
- The name of the data source to refresh

```
C:\Program Files\Tableau\Tableau 10.2\bin>tableau refreshextract
--server https://mytableauserver --datasource mydatasource
```

For more information about the utility, see **Tableau Data Extract Command-Line Utility** on page 1049.

Performance tips

You can improve performance by enabling one or both of these capabilities:

- Query parameter binding
- Shared metadata caching

The following sections describe how to use these features.

Use bind variables

Tableau can use bind variables for your SAP HANA queries. This improves performance:

- For queries — HANA can compile a query once and execute it many times.
- For multi-node HANA deployments — Parameterized queries are optimally routed to the proper node.

To enable this feature, your Tableau Server Administrator can add the CAP_ODBC_QUERY_USE_PREPARE_PARAMETER_MARKER capability in a TDC file.

> **Notes:** This feature requires Tableau 2018.3 or later.
> Despite its name, CAP_ODBC_QUERY_USE_PREPARE_PARAMETER_MARKER can be used with JDBC as well as ODBC.

Use column metadata collection

Tableau supports column metadata collection, resulting in great load-time improvements for some views.

When Tableau reads the column metadata for a table or view, it uses a prepared query on the table by default. This is usually the most accurate option, but may not be the most efficient.

Optionally, you can use column metadata collection. That is, you can disable prepared query metadata using a TDC file. When you do this, Tableau uses the ODBC API, SQLColumns. While this method is faster, it may not be as accurate. That is why you should use this option cautiously and only when it seems like a good approach for your data.

Using column metadata via SQLColumns is enabled using these capabilities:

- CAP_ODBC_METADATA_SUPPRESS_EXECUTED_QUERY
- CAP_ODBC_METADATA_SUPPRESS_PREPARED_QUERY

> **Note:** To apply these capabilities to temporary tables, you must install Tableau Server 2019.3.6 or later. Then you can add those capabilities to the TDC file.

TDC example for SAP HANA

Your TDC file should look like the following example. In most cases, you will need no other capabilities. In particular, be sure to avoid capabilities such as CAP_ODBC_METADATA_SUPPRESS_SQLPRIMARYKEYS_API and CAP_ODBC_METADATA_SUPPRESS_SQLFORIEGNKEYS_API. These may speed up the initial connection time, but will cause us to generate less efficient queries later.

```
<?xml version='1.0' encoding='utf-8' ?>
<connection-customization class='saphana' enabled='true' ver-
sion='9.3'>
<vendor name='saphana' />
<driver name='saphana' />
<customizations>
<customization name='CAP_ODBC_QUERY_USE_PREPARE_PARAMETER_MARKER'
value='yes' />
</customizations>
</connection-customization>
```

For more information about using TDC files, see "Create a TDC file" in the Customize and Tune ODBC Connections topic. Also see the Knowledge Base article called Using a TDC File with Tableau Server.

> **Important:** Be sure enabled='true' is part of your connection-customization line.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

- Configure SSL for JDBC Connections - Get more general information about installing trusted SSL certificates for JDBC connections.
- Customize SAP HANA Connections to Enable SAML Encryption - Read thisTableau Community article on enabling encryption by creating a TDC file.
- Tableau and JDBC - Find tips for using and troubleshooting JDBC drivers with Tableau.
- Customize and Tune a Connection - Describes how to use TDC and PROPERTIES files to customize connections.
- Using a TDC File with Tableau Server - Read this Tableau Knowledge Base article on using a TDC file with Tableau Server.

SAP NetWeaver Business Warehouse

This article describes how to connect Tableau to SAP NetWeaver Business Warehouse (BW) data and set up the data source.

Use this connector with Tableau Desktop on a Windows computer.

Install SAP software

Before you can connect Tableau to SAP BW data, you must install SAP software on your computer. For instructions on how to install SAP software, see Connecting to SAP BW on Tableau Community.

Before you connect

Before you connect, gather this connection information:

- Connection name
- Authentication: User name and password, or integrated authentication
- (Optional) Client for BW system
- (Optional) Language

If you want to use single sign-on, be sure that your SAP personal security environment (PSE) is configured for single sign-on.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SAP NetWeaver Business Warehouse**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Select a connection from the drop-down list.

       A connection appears in the drop-down list if a system entry exists for it. You can create a new system entry using the SAP Logon utility, or contact your database administrator.

    b. Select **Authentication** method:

       Username and Password - Enter the user name and password to sign in to the server.

Integrated Authentication - Select this option if your environment is configured for single sign-on (SSO). This option is configured by the Tableau Server administrator. For more information, see Enable Single Sign-On for SAP BW in the Tableau Server Help.

c. Optionally, enter the **Client ID** for the BW system and the **Language**.

d. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. Search for or select a catalog or InfoProvider.

c. Search for or select a cube or query.

d. Select the sheet tab to start your analysis.

Note: Tableau doesn’t support link nodes.

Support for SAPUILandscape.xml

Prior to version 7.40 of the SAP GUI tools, SAP BW connections are defined on client computers in the saplogon.ini file. When you add a connection definition using SAP Logon, that definition is added to the saplogon.ini file. Tableau reads from that file to get the list of SAP BW systems to show in the **Select connection** drop-down list in the Tableau connection dialog box.

Starting in version 7.40 of the SAP GUI tools, the format of the connection definition file has changed to XML, and this connection information is now stored in SAPUILandscape.xml and SAPUILandscapeGlobal.xml. Note that Tableau doesn't support using the SAPLOGON_LSXML_FILE environment variable to specify the location of these files.

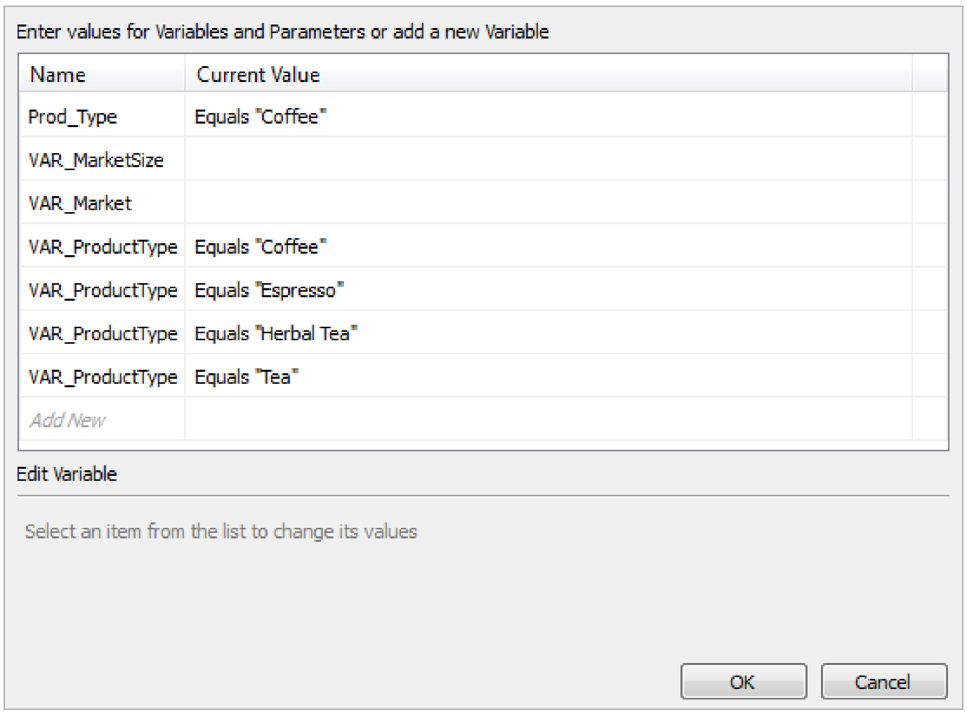
Tableau supports using landscape XML files located on a remote HTTP server.

Connections in drop-down list might not work

If you update to version 7.40 of the SAP GUI tools and then have issues with the **Select connection** drop-down list where connections don't work in Tableau, it might be because Tableau doesn't support `Messageserver` elements in the SAPUILandscape.xml and SAPUILandscapeGlobal.xml file. If a connection definition references a message server, the connection will display in the **Select connection** drop-down list, but it won't connect.

Select variables and input parameters

If the table you use includes required or optional variables or parameters, the Variables and Input Parameters dialog box opens.

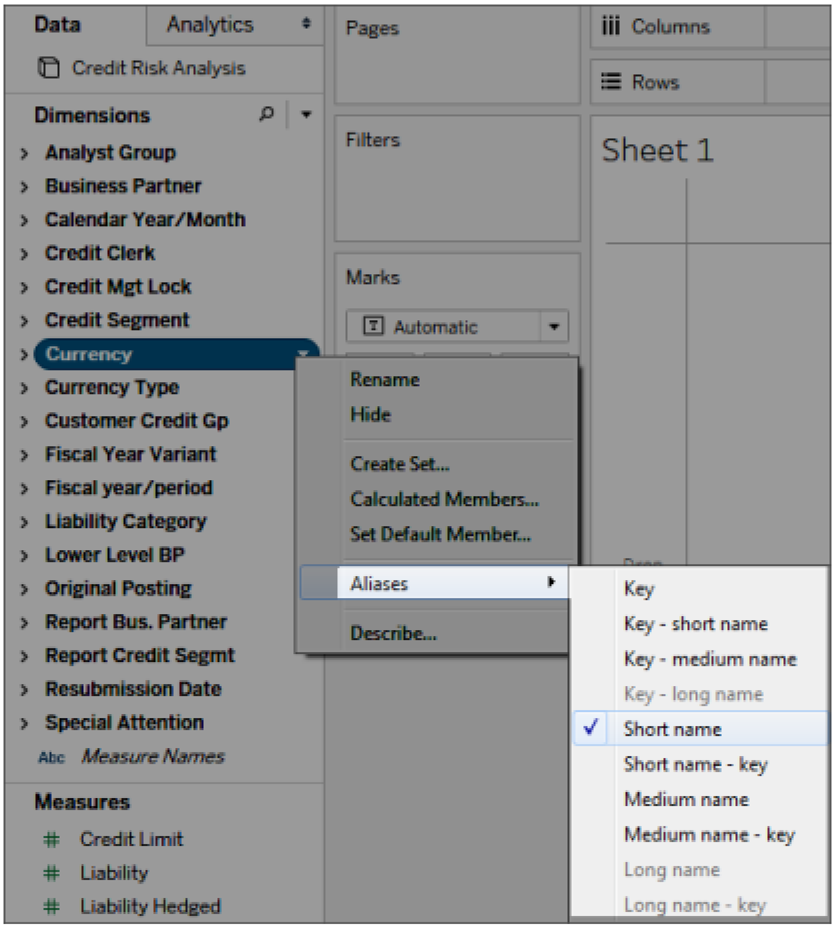


- Required variables and parameters display their current value or ***Required**.
- Optional variables and parameters display their current value or are blank.
- Prompt for the variable when the workbook is opened by selecting the check box under **Prompt**.

Select a variable or parameter and type or select a value for it. Repeat for all required values and any optional values that you want to include, and then select **OK**.

Select the alias for a dimension

When you connect to SAP NetWeaver Business Warehouse (SAP BW) databases, you have the option to select which alias to use for dimensions. SAP BW offers a number of different options including long name, short name, medium name, and key. Sometimes the alias used by default is not very useful. You can right-click the dimension and change it to another option such as **Short name**, which changes the labels to something more meaningful.



Support for SAP BW extracts

Starting in 10.4, you can create an extract of SAP BW data without the need for a special product key from Tableau. Prior to 10.4, SAP BW extracts were an unsupported beta feature. If you're using this beta feature, product keys will be extended through 2017, however, once the keys expire or to get support for this feature, you must upgrade to version 10.4.

When you create an extract from an SAP BW data source, Tableau converts data from the cube into a relational extract. This conversion can sometimes take a while based on the size of the cube. In addition, much of the cube metadata cannot be represented in the relational data model, so there are some limitations when working with extracts created from an SAP BW data source in Tableau Desktop. (Note that beta extract functionality was not supported in Tableau Desktop 9.0.)

SAP BW extract features

The following features are included in SAP BW extract functionality:

- The behavior of empty user filters has changed to be consistent with the behavior on other data sources. Using an empty user filter no longer hides the dimension.
- You can use **Hide** to hide dimensions before you extract data or refresh an extract. This reduces the size of the extract and the amount of time to perform an extract.
- Hiding a dimension takes priority over creating a user filter on the dimension.
- You can rename dimensions and measures in the data grid before extracting data.
- Tableau modifies the extract query slightly to fix some of the `[cube contains no data to transform]` errors.

SAP BW extract limitations

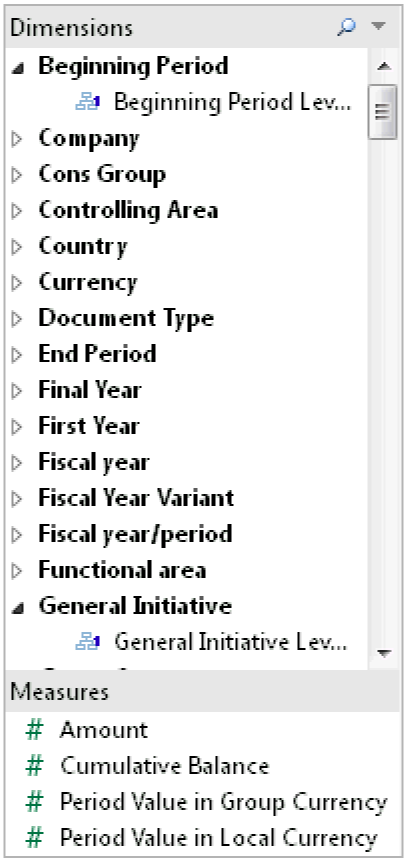
- Tableau supports only infocubes, or InfoProviders that contain only infocubes. A composite source that contains other BW object types is not supported.
- For large extracts, Tableau automatically partitions the query if the query fails or times out.
- BEx queries that contain more than 49 dimensions are not supported by SAP BW.
- You can't switch between a live connection and the extract. When you connect to an SAP BW data source, you'll have the option to Connect Live or Import the data. If you choose to Connect Live, the Extract options that are typically available for a connection are not an option. Further, if you choose to Import that data and create an extract, you cannot switch back to the live connection using the **Use Extract** command.
- You can't replace an extract with a live connection to SAP BW.
- Incremental extracts are not supported.
- When creating the extract, you can add filters to limit the data included in the extract, but these filters can be based only on hierarchies.
- Tableau doesn't support extracting the following SAP BW Objects:

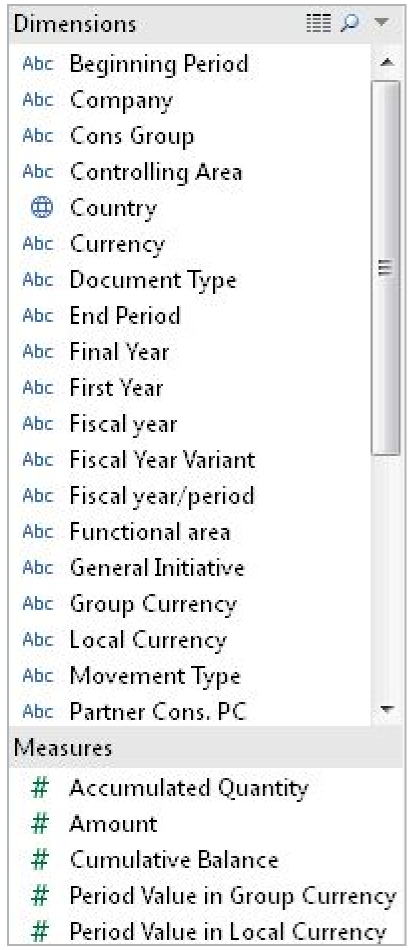
- Calculated sets.
- Custom formatting and cell properties, except aliases.
- User hierarchies. Hierarchies will collapse.
- Pre-aggregated data.
- Exception aggregation.
- Compounding attribute.

Data displays differently for extract and live connections

In addition to the above limitations, you may notice some differences between an extract connection and live connection to SAP BW data. Specifically, when you connect to SAP BW directly, the dimensions, hierarchies, and levels are represented in the Data pane. After you create an extract, all of the fields are shown in a flat list.

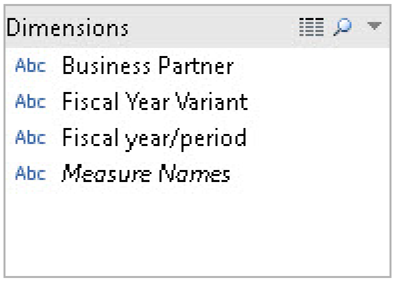
Live connection	Extract connection

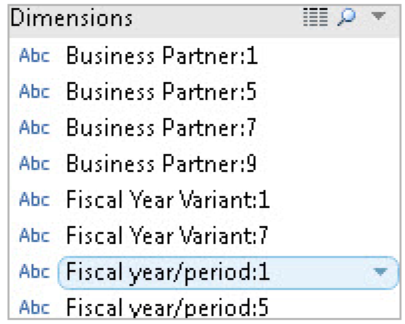




When you connect to an SAP BW data source and create an extract or connect to a Tableau Data Source file (.tds) that connects to an SAP BW extract, the fields display in the Data pane as expected. However, if you connect directly to the extract file, you will see extra fields that represent the alias options. All aliases are included in the extract and are typically hidden. You can hide the alias variants that you are not using by right-clicking the field and selecting **Hide**.

Aliases hidden	**Aliases showing**





Frequently asked questions

Requirements

What versions do I need?

- For the client machine where Tableau Desktop or Tableau Server is installed: SAP GUI 7.30 or later for Windows.
- For the server machine: SAP BW 7.40 or later.
- Tableau Desktop or Tableau Server, any supported version. For information about which versions are supported, see Supported Versions on the Tableau website.

Do I need any special licenses like OpenHub to use Tableau with SAP BW?

Tableau uses the SAP OLE DB for OLAP provider (part of the Open Analysis Interfaces) to interact with SAP BW. This provider is a mechanism to do 'reporting' and retrieving 'reasonable' amounts of data. Unlike OpenHub, the interface is not designed for moving large amounts of data. You should verify with SAP that you have licensed SAP BW correctly for your environment and that your licensing terms permit the use of the OLE DB for OLAP provider for tools like Tableau.

Does the SAP BW connector require HANA?

No. The Tableau SAP BW connector does not require HANA. The connector will work with any of the supported SAP BW backend databases such as Oracle and SQL Server. However, you may use HANA if you prefer.

Can I connect to SAP BW using Logon Groups?

No. Tableau doesn't support connecting to SAP BW using Logon Groups.

Connection

Is the connection between Tableau and SAP BW a live connection or is it a Tableau extract?

Tableau supports live and extract connections to SAP BW data.

With a live connection you will always see the latest data in your workbooks and dashboards.

Does Tableau support connecting directly to the underlying database tables that support the BW instance?

No. Tableau does not support this method for connecting to the SAP BW system.

Performance

How does performance compare to existing tools we use?

System performance should be comparable to existing tools. If performance is an issue in your environment, you might want to consider adopting SAP HANA in your environment. For more information, see SAP HANA on the SAP website.

Security

How does Tableau handle user security?

Tableau leverages all the security policies already implemented in the SAP BW server. There is no need for you to duplicate this in Tableau. All user authentication and authorization is performed by SAP BW. Tableau passes your credentials to the SAP BW system to be authenticated. Once authenticated, you will see only the objects and data that your profile or role on the SAP BW system has been enabled to access.

How does user security work with Tableau Server when I publish a dashboard?

By default, worksheets and dashboards connected to SAP BW data require you to log into SAP BW with your SAP user name and password. However, Tableau Server can be configured to embed credentials so that you are taken directly to the published views and dashboards. In addition, Tableau Server offers an additional layer of security if desired. Tableau Server administrators can configure user authentication and authorization, independent and in addition to the SAP BW authentication and authorization. For more information about Tableau Server security, refer to the Security section of the Tableau Server Online Help.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

SAP SuccessFactors

This article describes how to connect Tableau to the SAP SuccessFactors application and set up the data source.

Before you begin

Before you begin, gather the followingconnection information:

- Environment Account ID
- Authentication information

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SuccessFactors**. For a complete list of data connections, select **More,** under **To a Server**. Then do the following:

    a. Enter the application ID.

    b. Enter the authentication information.

    c. Select **Sign In.**

Note: If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users figure out which data source to connect to.
   b. From the **Database** drop-down list, select a database.
   c. Under **Table**, select a table or use the search field to search for a table by name.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

SAP Sybase ASE

This article describes how to connect Tableau to an SAP Sybase ASE database and set up the data source.

Tableau connects to Sybase ASE version 15.2 and later.

Before you begin

Before you begin, gather this connection information:

- Name of the server you want to connect to, port number, and if you use single sign-on, service name
- Authentication method: integrated authentication (single sign-on), or user name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SAP Sybase ASE**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

   a. Enter the name of the server you want to connect to, and specify the port to use. Optionally, enter the service name. If you use single sign-on, you must enter the service name.

   b. Select the **Authentication** method:

      Integrated Authentication - Select this option if your environment is configured for single sign-on (SSO).

      Username and Password - Enter the user name and password to sign in to the server.

   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

   d. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.

   c. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.

   d. Under **Table**, select a table or use the text box to search for a table by name.

      You can also specify a stored procedure in the database. To access stored procedures, the database must be appropriately configured with a linked server, which could be a remote server or a pointer to the database itself (known as *loopback*). For more information, see **Use a Stored Procedure** on page 972.

e. Drag a table or stored procedure to the canvas, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: Kerberos authentication for SAP Sybase ASE doesn't support delegation. This means that if you selected **Integrated Authentication**, you can't use **Viewer credentials** as the **Authentication** method when you publish a workbook or data source to Tableau Server; you can only use **Server Run As account**, and the Tableau Server Run As User account must have an account in the SAP Sybase ASE database.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

SAP Sybase IQ

This article describes how to connect Tableau to an SAP Sybase IQ database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the host that hosts the database you want to connect to
- (Optional) Server name. Tableau supports Sybase IQ multiplexes, so a server name is not required.
- Authentication method: Windows Authentication or user name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SAP Sybase IQ**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. Enter the host name that hosts the database that you want to connect to.
    b. (Optional) Enter the server name.
    c. Select how you want to sign in to the server. Specify whether to use Windows Authentication or a specific user name and password. If the server is password protected, you must enter the user name and password.
    d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
    e. Select **Sign In**.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. From the **Database** drop-down list, select a database or use the text box to search for a database by name.
    c. Under **Table**, select a table or use the text box to search for a table by name.
    d. Drag a table to the top area of the data source page, and then select the sheet tab to start your analysis.

Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

ServiceNow ITSM

This article describes how to connect Tableau to ServiceNow IT Service Management (ITSM) data and set up the data source.

> **Note:** For the most up-do-date information on this connector, see the ServiceNow ITSM help topic for the current version of Tableau.

Before you begin

Before you begin, gather this connection information:

- ServiceNow instance you want to connect to: cloud-based or on premises
- User name and password for a local ServiceNow account

Connection requirements

To connect Tableau to ServiceNow, make sure that the following requirements are met:

- **Use a local ServiceNow account.** The ServiceNow API doesn’t support single sign-on (SSO), so the Tableau connector can only authenticate using a local ServiceNow account. For more information, see **Use a local ServiceNow account** on page 635.
- **Have the correct permissions.** Access control lists (ACLs) must provide the user the permission required to access the data. For more information see **Troubleshoot data access** on page 636.
- **For on-premises, CORS is configured.** Before you connect to an on-premises instance of ServiceNow, you must configure CORS for Tableau in ServiceNow. For more information, see Configure CORS for Tableau on ServiceNow in Tableau Community.
- **For cloud instances, proxy connection is required.** The native ServiceNow connector routes through "servicenow-auth-proxy.connectors.tableau.com" instead of directly from the local user's IP. The proxy is necessary due to the way OAuth 2.0 is

designed. It stores the unique client ID and secret required by OAuth to permit Tableau Desktop access to ServiceNow.

Note: Don't apply IP address access controls in ServiceNow when using a proxy connection. If you do need to apply IP address access controls, use Find denied IP addresses to find the denied IPs, and then add Allow rules for them. Be aware of that an IP associated with "servicenow-auth-proxy.connectors.tableau.com" may change. This is a cloud service and does not have static IP. Tableau does not provide or update information about the IP.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **ServiceNow ITSM**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Select **Instance Location**: **Cloud-based** or **On Premise**.
   b. Enter your ServiceNow instance, user name, and password for a local ServiceNow account.
      Note: When you select **On Premise**, you must enter a full URL for the ServiceNow instance.
   c. Select **Sign In**.
   d. Select **Filter Type**: **Relative date range** or **Fixed date range**, and then select or specify the range.
   e. Select **Connect**.
      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network or ServiceNow administrator.
2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. Under **Table**, select a table and drag it to the top of the canvas.
   c. Select the sheet tab to start your analysis.
      After you select the sheet tab, Tableau imports the data by creating an extract. Note that Tableau Desktop supports only extracts for ServiceNow. You can update the data by refreshing the extract. For more information, see **Extract**

Your Data on page 1014.

Creating extracts might take some time depending on the amount of data that is included.

Use Dashboard Starters

You can use a Dashboard Starter to quickly build informative dashboard designs made specifically for key business metrics on Tableau Desktop and Tableau Online. For more information, see **Dashboard Starters for Cloud-based Data** on page 2750.

Key considerations

ServiceNow enforces an account's access permissions

ServiceNow accounts that have the Admin role can retrieve encrypted password values from the User table. This file will be null for accounts with less permissions. Because ServiceNow enforces the access permissions assigned to the account, be aware that there might be other sensitive fields in your data that an Admin can access.

Custom tables not supported

The Tableau connector for ServiceNow ITSM supports the ITSM service, which doesn't include custom tables.

ServiceNow provides table names

When you view tables in the Tableau connector, the tables are labeled according to their ServiceNow label, not the table name. For example, the table "sys_user_role" is shown by its label, "User Role". This label is determined by the ServiceNow connector.

Display value fields aren’t included in the data source

When you connect to a table that contains fields with display values (fields with a dv_ prefix), those fields aren’t included when you create a data source. You can add the display values by joining to the appropriate related tables that contain those values.

Use a local ServiceNow account

You must use a local account to connect Tableau to your data. If you don’t have a local ServiceNow account, we recommend that you ask your administrator to implement one of the following solutions:

- Create a local account and share it.
- Create a local account and temporarily share it long enough to extract the data and publish it to Tableau Online or Tableau Server. Then the administrator can save the new password and schedule refreshes.
- Create a local account on a non-production ("staging") instance and use it to create a data source. Give the data source to the administrator to edit the connection to point to the production instance, publish the data source to Tableau Online or Tableau Server, and schedule for data refreshes.
- Ask the administrator to publish the data sources you need.

Date range selections can impact performance

When you apply a date filter, it's tempting to gather as much data as possible when you do an analysis, however, retrieving records from ServiceNow ITSM can be time-consuming. Tableau doesn’t know how much data there is in a particular date range until it retrieves the data. For this reason, you should restrict your date range at first, and then expand after you evaluate performance. Date filters are applied to these tables: metric_instance, change_request, incident, task, problem.

To give you a rough idea of how much time it might take to retrieve data from ServiceNow ITSM, tests were conducted using a high-speed connection. This table shows how long it took in the test environment to retrieve a given number of records.

Number of records	**Time to retrieve**
1,000	11 seconds
10,000	2 minutes
100,000	18 minutes

Troubleshoot data access

For ServiceNow ITSM users

If you see an error when you try to access data, for example, "You are not authorized to access this resource...," contact your ServiceNow administrator.

For ServiceNow administrators

ServiceNow uses access control lists (ACLs) to specify who has access to which data. Often the default permissions that are set for ServiceNow fulfillers are adequate for Tableau users to access the content they need to analyze ServiceNow ITSM data.

Tableau users need to access metadata about tables in ServiceNow, which is available in the following tables:

- sys_dictionary table (metadata about columns)
- sys_db_object table (metadata about tables)
- sys_glide_object table (metadata about field types)

If a Tableau user can't access ServiceNow ITSM data, make sure that the ACLs provide the user table level and field level read access to the tables above, and that there are no scripts that are locking down access.

To grant access to the metadata tables, as a best practice, you might consider creating a role with the ACLs needed to access the metadata tables, and then assign the role to a Tableau User group. Then, you can assign Tableau users to the group so that users can access the data needed for analysis in Tableau.

For information on the creation of an ACL, please see the ServiceNow Access control list rules on the ServiceNow website.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

SharePoint Lists

This article describes how to connect Tableau to SharePoint lists and set up the data source.

> **Note:** Multi-factor authentication (MFA) is not supported by the drivers currently available for Sharepoint Lists.

Before you connect

Before you begin, gather this connection information:

- The URL to the SharePoint site that contains the lists you want to connect to
- The edition of your SharePoint server: on premises or SharePoint Online
- The method of authentication: third-party SSO or integrated authentication
- The user name or email address, password, and (optionally) SSO domain, if needed

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SharePoint Lists**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
    a. In the **SharePoint Site** text box, enter the SharePoint URL. Don't include the list name in the URL.
    b. Select the **Edition**: **On Premise** or **SharePoint Online**.
    c. Select the **Authentication** method: **Username and Password**, **Third-party SSO**, or **Integrated Authentication**.
    d. If the fields are available, enter the **Username** and **Password**, and optionally, enter **SSO Domain**.
    e. Select **Sign In**.
2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. Under **Table**, select a table and drag it to the top of the canvas.
    c. Select the sheet tab to start your analysis.

Connection tips

- If you use SharePoint Online, select **Third-party SSO** as your **Authentication** type.
- If you use Kerberos, select **Integrated Authentication** as your **Authentication** type.
- If you use NTLM, select **Username and Password** as your **Authentication** type.
- If you select **Third-party SSO** as your **Authentication** type, enter your credentials, which might be your user name or email address, and your password. Optionally, enter

the **SSO Domain**.

> **Note**: Tableau currently supports only **Okta** and **OneLogin** for Third-party SSO connections.

- Any list you connect to will be the existing default SharePoint view only. This means that if any rows of data are filtered on the default view in SharePoint, you will not be able to retrieve the filtered data. If your goal is to access everything in the SharePoint list then you will need to adjust the default view to show all rows.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Both **Authentication** methods on a Mac require you to enter your user name and password to connect.

Refresh your data

Tableau supports only extract connections for SharePoint lists, not live connections. You can update the data by refreshing the extract; incremental refreshes are not supported. For more information, see **Refresh Extracts** on page 1041. For information about refresh schedules, see **Schedule Extract Refreshes as You Publish a Workbook** on page 3086.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

SingleStore

This article describes how to connect Tableau to a SingleStore database and set up the data source.

Note: SingleStore was formerly named MemSQL.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- User name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **SingleStore**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database.
   b. Enter the user name and password.
   c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   d. Select **Sign In**.

      Select the **Require SSL** check box when connecting to an SSL server.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:

a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

b. From the **Database**drop-down list, select a database or use the text box to search for a database by name.

c. Under **Table**, select a table or use the text box to search for a table by name.

d. Drag a table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Snowflake

This article describes how to connect Tableau to a Snowflake data warehouse and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server name that you want to connect to
- Authentication method:
  - OAuth
  - Username and Password
  - Okta Username and Password
- Sign in credentials depend on the authentication method you choose
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

> **Note:** To support using OAuth with Snowflake, you must install the latest ODBC driver from Snowflake (currently version 2.21).

Make the connection and set up the data source

Connect Tableau to your data

1. Start Tableau and under **Connect**, select **Snowflake**. For a complete list of data connections, select **More** under **To a Server**.
2. Enter the name of the server that you want to connect to.
3. Select the **Authentication** method: **Username and Password**, **SAML IdP**, or **Sign in using OAuth**.
4. Enter the information that you are prompted to provide. The information you are prompted for depends on the authentication method you choose.
   - If you select **SAML IdP**, enter the **Username** and **Password**, then in the **SAML IdP (Okta)** field, enter the URL for the SAML IdP server. Proceed to step 5.
   - If you select **Sign in using OAuth** follow these steps:
     a. Select **Initial SQL** if you want to run a SQL command at the beginning of every connection. For more information, see **Run Initial SQL** on page 736.
     b. Select **Sign In**.

   c. In the web page that opens, log in to Snowflake by entering your **User Name** and **Password** and selecting **Log In**. Or select **Single Sign On** if Snowflake is configured to support SSO.
   d. Click **OK** to confirm authorization to access your Snowflake data.
   e. Close the browser window and return to Tableau.
   f. Proceed to the next section, "Set up the data source."
5. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
6. Select **Sign In**.

   If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

Customize the connection using driver parameters

In some cases, you may want to modify the connection made with the Snowflake connector. You can do this by appending driver parameters to the connection string in the **Advanced** tab.

> **Note:** You can’t change the driver parameters that Tableau generates; you can only append parameters.

For example, this syntax specifies a proxy server to use when the `no_proxy` values are not met. `No_proxy` indicates which host names are allowed to bypass the proxy server:

```
Proxy=http://proxyserver.company:80;no_proxy=.trustedserver.com;
```

For another custom driver parameters example, see **Use a proxy for Snowflake** on page 645.

For details about driver parameters, see ODBC Configuration and Connection Parameters on the Snowflake website.

Set up the data source

On the data source page, do the following:

1. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
2. From the **Warehouse** drop-down list, select the warehouse or use the text box to search for a warehouse by name.

   Note: If you leave this blank and do not select a warehouse, Tableau uses your default Snowflake virtual warehouse.

   For more information, see **Default virtual warehouses in Snowflake** below.
3. From the **Database** drop-down list, select a database or use the text box to search for a database by name.
4. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.
5. Under **Table**, select a table or use the text box to search for a table by name.
6. Drag a table to the canvas, and then select the sheet tab to start your analysis. Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Default virtual warehouses in Snowflake

When you create a Snowflake data source, you are accessing both a Snowflake warehouse (for computational resources) and Snowflake databases (for data resource). Both warehouses and databases require permissions to access them. You can use different warehouses to access the same database, depending on the computation power you need, assuming you have permissions for each warehouse and database.

When you create a data source, you have the option to choose a warehouse from a drop-down list. If you leave this blank and do not select a warehouse, Tableau uses your default

Snowflake virtual warehouse. If you subsequently publish that data source or workbook, when a user connects, Tableau will use that person's default Snowflake virtual warehouse.

Each user should be assigned a default warehouse so that Tableau can use that one as a default to open a workbook from a Snowflake data source.

Configure Snowflake server support for OAuth

For information about how to configure OAuth for connections between Tableau and Snowflake, see Configure OAuth for Snowflake Connections.

Use a proxy for Snowflake

You can connect from Tableau through a proxy to Snowflake. If you are using Tableau 2019.4 or later, you can configure this with the connector dialog by typing the required parameters in the **Advanced** tab.

For example:

`proxy=http://proxyserver.company:80`

For information about the parameters you should use, see the Snowflake documentation.

> **Note:** Tableau doesn't use a DSN to connect to Snowflake, so you can ignore the instructions in the Snowflake documentation about using a DSN.

If you are using a version of Tableau before 2019.4, you can configure this by entering the parameters in a TDC file using `odbc-extras`.

For example:

```
<connection-customization class='snowflake' enabled='true' version='19.1'>
<vendor name='snowflake' />
<driver name='snowflake' />
  <customizations>
  <customization name='odbc-connect-string-extras'
value='proxy=http://proxyserver.company:80' />
  </customizations>
</connection-customization>
```

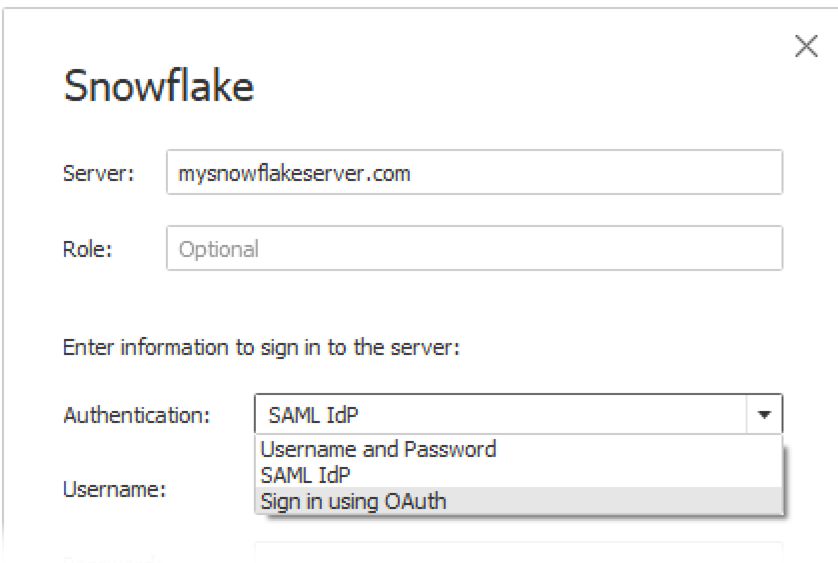
For more information, see the Customize the Connection String for a Native Connector article.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Best Practices for Using Tableau with Snowflake on the Snowflake website (registration required).

Configure OAuth for Snowflake Connections

When you connect to your Snowflake data, you have three authentication options to choose from. In most cases, we recommend using OAuth. This option offers the best combination of functionality and security.



With OAuth, you can:

- Leverage an identity provider (IdP) to facilitate access.
- Configure it to provide a single sign-on (SSO) experience.
- Enforce multi-factor authentication (MFA).

OAuth 2.0 is an industry-standard protocol for authorization. It is configured at the browser level and displays a sign-in dialog in a browser window to the user.

> **Note:** Be careful not to confuse this with Tableau's **SAML IdP** authentication option in the connection dialog. You should connect using the **Sign in using OAuth** option.

When you use OAuth, a key consideration is maintaining access for content published to Tableau Server or Tableau Online. When Tableau content connects live to Snowflake via OAuth, owners must reauthenticate the workbook connection each time the access token expires (every 90 days by default).

See the Snowflake help topic, Configure Snowflake OAuth for Partner Applications, for details on setting the access token expiration limit. If you have a business need to extend this period to prevent errors on your Tableau content, contact Snowflake Support for assistance. If you do not manually refresh your content before this time period, it can cause an error when the Tableau workbook tries to load.

Configure OAuth between Snowflake and Tableau

In a connection between Tableau and Snowflake, each must have OAuth configured.

- **Tableau:** Tableau includes credentials for Snowflake to allow access. This happens automatically when you use the Snowflake connector in Tableau. No additional OAuth configuration is required in Tableau. For more information, see OAuth Connections in Tableau Help.
- **Snowflake:** In Snowflake, you will enable OAuth to grant access to Tableau. Follow steps in this Snowflake help topic: Configure Snowflake OAuth for Partner Applications.

Note: Beginning with version 2020.4, your OAuth connection can use AWS PrivateLink or Azure Private Link. For more information, see Change Snowflake OAuth to Private Link with Saved Credentials.

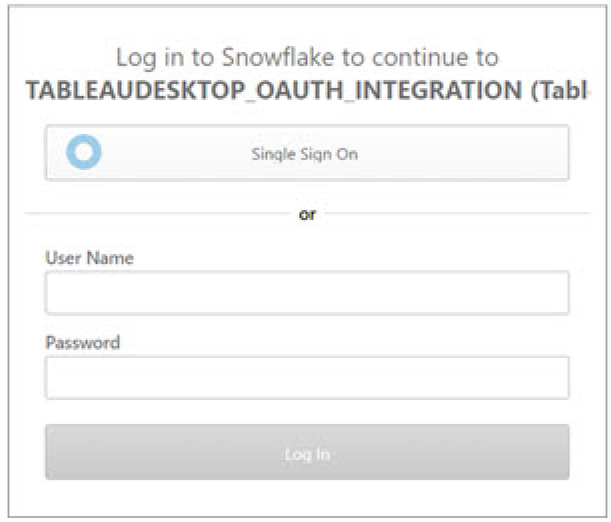
About using SSO with OAuth

Single Sign-On (SSO) adds another layer of security on top of OAuth authentication. A separate IdP configured for SSO manages authentication for all access activity across applications for your organization. All sign-in requests are routed to the SSO server, which displays a common sign-in dialog and checks the user’s credentials against a centralized database.

Tip: You can use saved credentials to avoid getting reprompted for your password. For more information, see Manage Saved Credentials for Data Connections.

Configure OAuth between Okta and Snowflake

Snowflake uses Okta as the default identity provider (IdP) that provides access tokens and authenticates identities. You'll need to configure settings in Snowflake and Okta for OAuth and single sign-on (SSO) capabilities.



In Okta, you will define Okta as an OAuth authentication server and identify Snowflake as an OAuth resource. Follow steps in this Snowflake help topic: Configuring an Identity Provider for Snowflake.

About using MFA with OAuth

Multi-factor authentication (MFA) introduces yet another layer of security. It requires two or more different methods of identification before the user can access a resource. Methods might include:

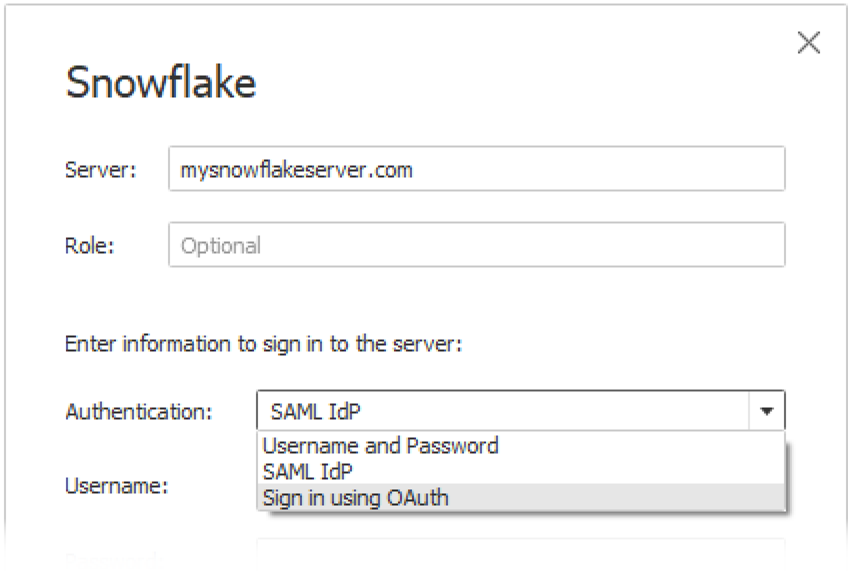
- A password
- A token from a second device
- Biometrics (fingerprint or eye scan, for example)
- Answer to a security question

You can optionally set up multi-factor authentication (MFA) with Okta or other an IdP for your connections between Tableau and Snowflake. For more on configuring MFA with Okta, see Okta Help.

Other connection options

When you connect to Snowflake from Tableau Desktop, you have two other options:

- SAML IdP
- Username and Password



The **SAML IdP** option works only if Okta is your identity provider and if MFA is disabled for users in Okta. The **SAML IdP** option supports SSO but does not support MFA. In this case, publishing with embedded credentials will use a specific user, but you can't employ per-user "viewer credentials" when you use Okta SAML.

> **Note:** In the past, some customers have used the "external browser" option with SAML IdP as a workaround to achieve SSO between Tableau Desktop and Snowflake. It will not work for Tableau Server. We recommend that you use the OAuth connection instead.

The **Username and Password** option uses the password stored by Snowflake. This option requires users to re-authenticate with their credentials whenever they connect to Snowflake or to embed these credentials.

Frequently asked questions

Why do my published Snowflake extracts fail after a period of time?

It may be that your OAuth access token has expired. You will need to manually re-authenticate to the data source to refresh the token. If you need to extend the lifetime of these tokens in the future, you can contact Snowflake Support.

How do I choose between "Prompt User" and "Embed Credentials" options when publishing a Snowflake data source?

If you would like any user accessing the data source to use their own credentials when connecting to the data source and associated content, use the **Prompt User** option. You can choose to embed credentials so everyone who accesses that data source uses those specific credentials. This method is functionally similar to using a "service account".

How do I leverage the row-level security that I've set up on Snowflake?

When users are prompted to enter their own credentials to access Snowflake from Tableau, those credentials map to the privileges they have in the Snowflake account.

See also

- OAuth Connections — Learn more about OAuth support with Tableau.
- OAuth (Snowflake Help) — Learn about Snowflake with OAuth.
- Summary of Security Features (Snowflake Help) — Find information about how Snowflake supports OAuth, SSO, and other security features.
- Manage Your Account Settings — Read about how to create and revoke personal access tokens.

Spark SQL

This article describes how to connect Tableau to a Spark SQL database and set up the data source. Tableau can connect to Spark version 1.2.1 and later.

You can use the Spark SQL connector to connect to a Spark cluster on Azure HDInsight, Azure Data Lake, Databricks, or Apache Spark.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to and port number
- Authentication method:
  - No Authentication
  - Kerberos
  - User Name
  - User Name and Password
  - Microsoft Azure HDInsight Service
- Transport. Your choices depend on the authentication method you choose, and include the following:

  - Binary
  - SASL
  - HTTP
- Sign-in credentials. Your choices depend on the authentication method you choose, and can include the following:
  - User name
  - Password
  - Realm
  - Host FQDN
  - Service Name
  - HTTP Path
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Spark SQL**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database and the port number to use.
   b. Connect to the database using SparkThriftServer. Note that the legacy SharkServer and SharkServer2 connections are provided for your use, but are not supported by Tableau.
   c. Select the **Authentication** method. Then, based on your selection, enter the information you are prompted for.

      If the server is password protected, and you are not in a Kerberos environment, you must enter the user name and password.
   d. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   e. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Schema** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the schema.
   c. In the **Table** text box, select the search icon or enter the table name and select the search icon, drag the table to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Note: Starting in 2018.3, Kerberos authentication for Spark SQL supports delegation. In 2018.2 and earlier, delegation isn't supported, which means that in earlier versions you can't use **Viewer credentials** as the **Authentication** method when you publish a workbook or data source to Tableau Server; you can only use **Server Run As account**.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.

- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Databricks and Tableau - Read the Tableau user guide on the Databricks website.

Teradata

This topic describes how to connect Tableau to a Teradata database or a Teradata Unity server and set up the data source.

Before you begin

Before you begin, gather this connection information:

- For Teradata database: Name of the server you want to connect to
- For Teradata Unity server: URL of the server you want to connect to
- Authentication method: Teradata Database, LDAP, or Integrated Authentication
- Depending on the environment, the user name and password
- Do you need an encrypted connection?
- (Optional) Query banding and initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Teradata**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the database you want to connect to.

      Note: You can connect to a Teradata Unity server by entering the appropriate URL in the **Server** text box.
   b. Select how you want to sign in to the server. Specify whether to use the built-in **Teradata Database** authentication, **LDAP**, or **Integrated Authentication**.

If the server is password protected, and you are not in a Kerberos environment, you must select **Teradata Database** or **LDAP** and enter the user name and password.

c. Select the **Require Encryption** check box if you need an encrypted connection.

d. (Optional) Select **Query Banding and Initial SQL**.

e. Select **Sign In**.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

   b. From the **Database** drop-down list, select the search icon or enter the schema name in the text box and select the search icon, and then select the database.

   c. In the **Table** text box, select the search icon or enter the table name and select the search icon, and then select the table.

   You can also specify a stored procedure in the database. For more information and for a list of constraints specific to Teradata databases, see **Use a Stored Procedure** on page 972.

   d. Drag the table or stored procedure to the canvas, and then select the sheet tab to start your analysis.

   Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

Use query bands to increase performance

When you connect to a Teradata database, you can optionally define query band statements that run during connection. These statements can increase performance, and they take advantage of the built-in security rules of the database.

Using query bands, you can pass parameters into the Teradata environment. Use these to set up a workbook to filter the data based on security rules that exist in the database. For example, you can pass in the Tableau Server user name for the current user so that when the view is loaded it only shows the data specific to that user. Query bands can also be used to improve performance. When connecting to Teradata, you can define a map between the name of the attributes passed into the query band and the corresponding values from Tableau.

> **Note:** This functionality is not currently supported for web authoring.

To set up query banding:

1. On the data source page, select **Data** > **Query Banding and Initial SQL**.
2. In the subsequent dialog box, specify name/value pairs in the top text box labeled Query Banding. You can use the **Insert** drop-down menu to add Tableau values. The Tableau values are described in the table below.

Value	Description	Example
`<TableauMode>`	The mode Tableau is operating in when generating queries. This value will either be “Connect” when retrieving metadata or “Analytical” when retrieving actual data.	Connect or Analytical
`<LoginUser>`	The user name of the person signed in to the database.	jsmith
`<ServerUser>`	The signed-in Server user. Does not include domain name. **Note:** `ProxyUser` returns the same value as `ServerUser`.	jsmith

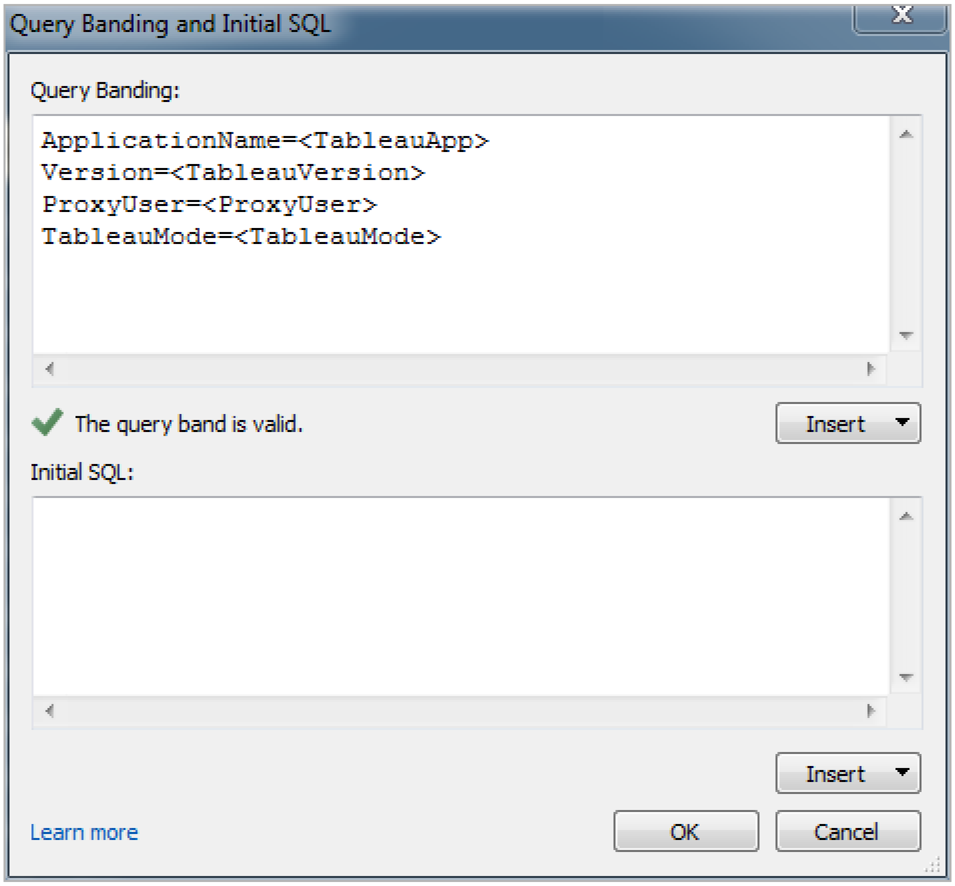
Value	Description	Example
	However, `ProxyUser` sets up impersonation and stores the Tableau Server user in the query band parameter. If you are using query banding for security purposes, you should use `ProxyUser` instead. This will ensure that query results are not shared between different users. `ServerUser` should only be used for auditing purposes.	
`<ServerUserFull>`	The signed-in server user including the domain name (if the server is using Active Directory). **Note:** `ProxyUserFull` returns the same value as `ServerUserFull`. However, `ProxyUserFull` sets up impersonation and stores the Tableau Server user in the query band parameter. If you are using query banding for security purposes, you should use `ProxyUserFull` instead. This will ensure that query results are not shared between different users. `ServerUserFull` should only be used for auditing purposes.	domain.lan\jsmith

Value	Description	Example
`<ProxyUser>`	Used when setting up impersonation on the server. Provides the username of the current server user.	jsmith
`<ProxyUserFull>`	Used when setting up impersonation on the server. Provides the username and domain name of the current server user.	domain.lan\jsmith
`<TableauApp>`	The name of the Tableau application.	Tableau Desktop Professional or Tableau Server
`<TableauVersion>`	The version of the Tableau application	6100.11.0428.0300
`<WorkbookName>`	The name of the workbook. Note that this only works for data sources embedded in the workbook.	Financial-Analysis

An example query band statement is shown below. This example passes the user name for the current server user.

```
ApplicationName=<TableauApp>Version=<TableauVersion>ProxyUser=<ProxyUser>TableauMode=<TableauMode>
```

Tableau checks the statement for errors as you type. When it is valid a green check mark displays at the bottom of the text box.



Use initial SQL commands

When you connect to a Teradata database, you can optionally specify a SQL command that will be run once upon connection. See **Run Initial SQL** on page 736 to learn more about adding these commands to your connection.

> **Note:** If Tableau detects that the Initial SQL payload cannot be delivered in one query, it attempts to split the query into separate statements, where each statement must end with a semicolon followed by a newline character, with no additional characters (such as an extra space) in between. This requirement is important because if a semicolon exists within the interior of a statement, for example within a string literal, it may not be safe to split the statement at that point. If the Initial SQL code that you specify in Tableau returns an error, and the same Initial SQL is deemed valid by the Teradata databases, it may be that Tableau has split the query incorrectly. If this happens, try reformatting the SQL code in Tableau manually.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Tableau and Teradata: The visual approach to the active warehouse - Read the Tableau whitepaper (registration or sign in required).

Teradata OLAP Connector

This article describes how to establish a Teradata OLAP connection.

Before you begin

Before you begin, gather this connection information:

- Connection name
- User name and password

Use this connector with Tableau Desktop on a Windows computer.

When you connect to Teradata OLAP data, you select from a list of connections, each representing a system data source name (DSN). A connection appears in the drop-down list if a system DSN exists for that connection. You can use the Windows ODBC Data Source Administrator utility to create a system DSN, or contact your database administrator.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Teradata OLAP Connector**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Select a connection from the drop-down list.
   b. Enter your user name and password to log on to the server.

If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

    c. Select **Sign In**.

2. On the data source page, do the following:
    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    b. Search for or select a catalog.
    c. Search for or select a cube from the catalog.
    d. Select the sheet tab to start your analysis.

Work with Teradata OLAP data

Named sets from a Teradata OLAP data source are displayed in the **Sets** area of the **Data** pane in Tableau. You can interact with these named sets in the same way you interact with other custom sets in Tableau. For more information, see **Create Sets** on page 1186. You can view underlying data for Teradata OLAP data sources, provided the database administrator has enabled this functionality. For more information, see **View Underlying Data** on page 2728.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

TIBCO Data Virtualization

This article describes how to connect Tableau to a TIBCO Data Virtualization (formerly called Cisco Information Server) virtual database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the data you want to connect to
- (Optional) Domain name
- Datasource name
- Authentication method: Windows Authentication, Kerberos, or user name and password
- Are you connecting to an SSL server?
- (Optional) Initial SQL statement to run every time Tableau connects

Use this connector with Tableau Desktop on a Windows computer.

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **TIBCO Data Virtualization**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:
   a. Enter the name of the server that hosts the data you want to connect to.
   b. (Optional) Enter the name of the domain.
   c. Enter the name of the **Datasource** you want to connect to.
   d. Select how you want to sign in to the server. Specify whether to use Windows Authentication, Kerberos, or a user name and password.

      Select the **Require SSL** check box when connecting to an SSL server.
   e. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.
   f. Select **Sign In**.

      If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:
   a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
   b. From the **Catalog** drop-down list, enter the catalog name in the text box, or select the catalog from the list.
   c. From the **Schema** drop-down list, enter the schema name in the text box, or select the schema from the list.
   d. Under **Table**, enter the table name in the text box, or select the table from the list.
   e. Drag the table to the canvas, and then select the sheet tab to start your analysis.

      Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Vertica

This article describes how to connect Tableau to a Vertica database and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Name of the server that hosts the database you want to connect to
- Database name
- User name and password
- (Optional) Initial SQL statement to run every time Tableau connects

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Vertica**. For a complete list of data connections, select **More** under **To a Server**. Then do the following:

    a. Enter the name of the server that hosts the database and the name of the database that you want to connect to.

    b. Enter the user name and password.

    c. (Optional) Select **Initial SQL** to specify a SQL command to run at the beginning of every connection, such as when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. For more information, see **Run Initial SQL** on page 736.

    d. Select **Sign In**.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.

2. On the data source page, do the following:

    a. (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.

    b. From the **Schema** drop-down list, select a schema or use the text box to search for a schema by name.

    c. Under **Table**, select a table or use the text box to search for a table by name.

    d. Drag the table to the canvas, and then select the sheet tab to start your analysis.

    Use custom SQL to connect to a specific query rather than the entire data source. For more information, see **Connect to a Custom SQL Query** on page 959.

Sign in on a Mac

If you use Tableau Desktop on a Mac, when you enter the server name to connect, use a fully qualified domain name, such as mydb.test.ourdomain.lan, instead of a relative domain name, such as mydb or mydb.test.

Alternatively, you can add the domain to the list of Search Domains for the Mac computer so that when you connect, you need to provide only the server name. To update the list of Search Domains, go to **System Preferences** > **Network** > **Advanced**, and then open the **DNS** tab.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Vertica & Tableau Solution Overview - Read the Tableau whitepaper (registration or sign in required).

Vertica Integration with Tableau: Connection Guide - Read the Vertica knowledge base article on the Vertica website.

Web Data Connector

This article describes web data connectors, what to do before you use a connector, and how to connect Tableau to a web data connector.

Use a web data connector

You can use a web data connector to connect to data that is accessible over HTTP and that doesn't already have a connector. A web data connector is an HTML file that includes JavaScript code. You can create your own web data connector or use one that has been created by someone else. The web data connector must be hosted on a web server running locally on your computer, on a web server in your domain, or on a third-party web server.

Test and vet the web data connector

You should use a web data connector that you trust. If you are unfamiliar with the web data connector, you or your Tableau Server administrator should test and vet the web data connector before you use it. For more information, see Testing and Vetting Web Data Connectors in the Tableau Server Help.

How to use a web data connector on Tableau Server and Tableau Online

When you use a web data connector, Tableau creates an extract of the data that the connector accesses. You can refresh the extract in Tableau Desktop.

When you publish the data source or workbook to Tableau Server, you cannot refresh the extract on Tableau Server unless the web data connector is added to the safe list on Tableau Server. If the connector requires credentials to sign in to the web-based data source, you need to ensure that the credentials are embedded with the data source. For more information, see Web Data Connectors in Tableau Server in the Tableau Server Help.

After the web data connector has been tested and vetted, the server administrator can add the URL for the connector to the safe list on the server. The server administrator should give this URL to you and to anyone else who wants to use the connector in a workbook.

If you open a workbook on Tableau Server that was created using a web data connector, but the connector has not been added to the safe list on Tableau Server, and you want to be able to refresh the extract on Tableau Server, follow the process for testing, vetting, and adding the connector to the safe list. If the connector requires credentials to sign in, you need to ensure that the credentials are embedded with the data source. You can then refresh your data on Tableau Server.

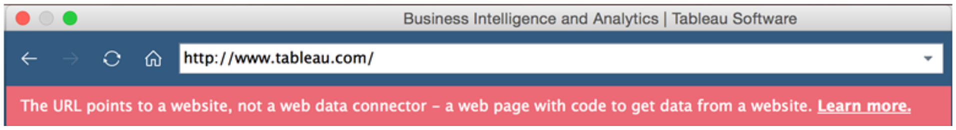
When you publish to Tableau Online, as a security measure, Tableau Online can't connect to or refresh an extract created by a web data connector. To refresh some web data connector extracts, you can use Tableau Bridge. For more information, see When to use Tableau Bridge to keep data fresh in the Tableau Online Help.

When you publish to Tableau Public, because you can't add a web data connector to Tableau Public, you can't refresh web data connector extracts directly on Tableau Public.

Connect to the data source

1. Start Tableau and under **Connect**, select **Web Data Connector**. For a complete list of data connections, select **More** under **To a Server**.
2. Enter the URL of the web data connector that you want to connect to, and then press Enter.

   Note: If you enter a URL to a website, rather than to a web data connector, an error message will appear.



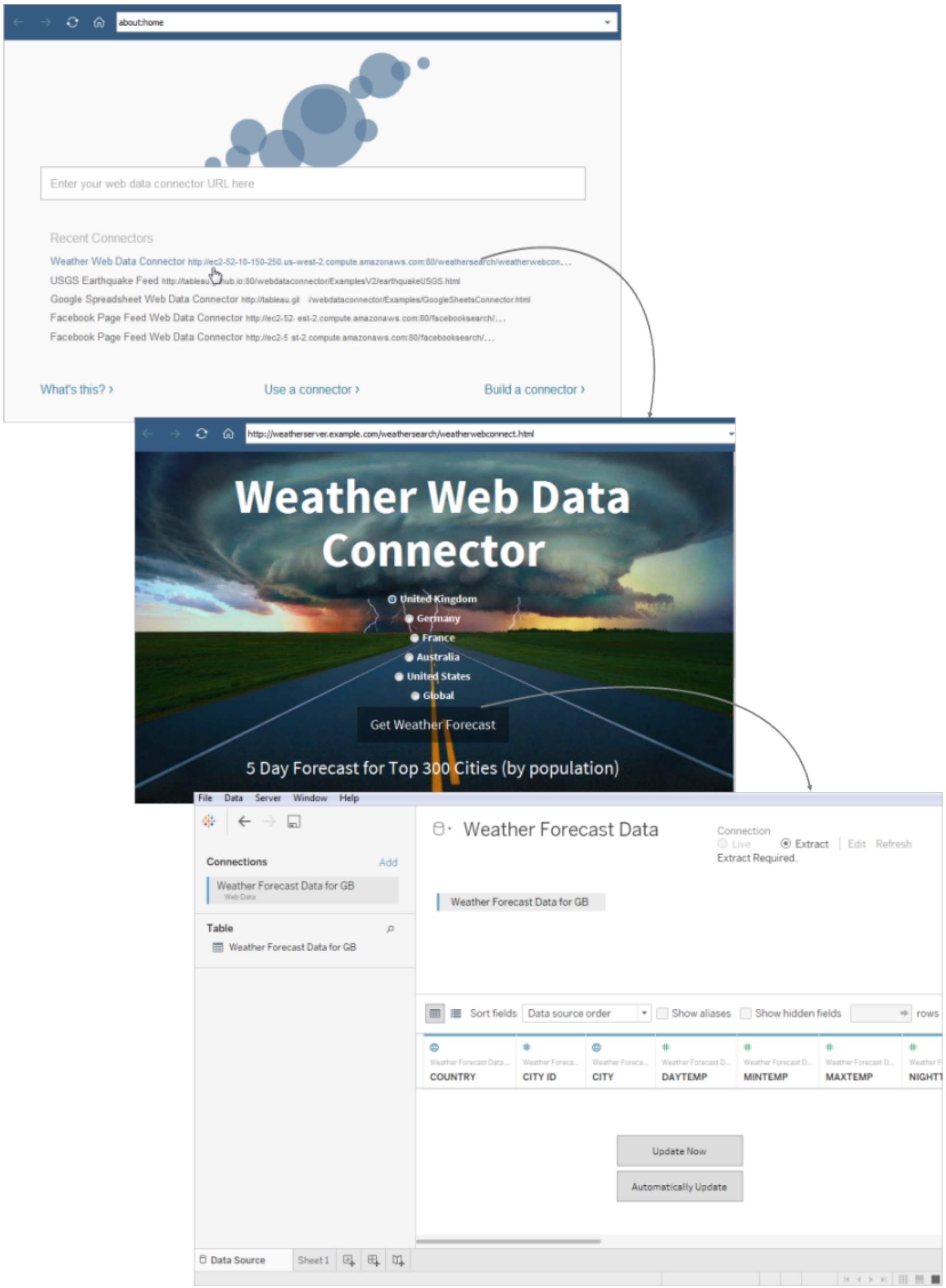
3. If the connector displays a web page, enter any information that you're prompted for, and then submit the page.
4. Wait while the connector retrieves your data and imports it into Tableau as an extract.
5. Select the sheet tab to start your analysis.

Web data connector data source example

An example of a web data connector data source is shown below. The first time you connect to a web data connector, the connection page outlines how to connect to the web data connector.

After you connect to a web data connector, on the connection page, you can do the following:

- View a history of all the web data connectors that you've connected to, or clear the history, in the drop-down list in the address bar.
- View your five most recently used connectors under Recent Connectors.
- Select links at the bottom on the window to learn more about web data connectors:
    - **What's this?** links to an overview of this feature.
    - **Use a connector** links to this topic.
    - **Build a connector** links to the Web Data Connector SDK, which provides tools and documentation for building a web data connector.



Optional settings

After you connect, you can make the following configuration changes to the data source before you start your analysis.

- **Sort fields and rows** – From the **Sort fields** drop-down list, select how you want to sort the columns in the grid or metadata grid. Sort the row values by selecting the sort button next to the column name.

- **Rename or hide columns** – Select the column header drop-down arrow and select the option you want.
- **Split columns** – Depending on how the data is structured, you can split the columns in your data into new fields. For more information, see **Split a Field into Multiple Fields** on page 1001.
- **Create calculations** – Create a new calculation based on an existing field in the data source.
- **Copy values** – Copy values in the grid by selecting the values and then pressing Ctrl+C. Alternatively, to copy values in the metadata grid, select the values, right-click, and then select **Copy**.
- **Add new or edit other data sources** – Select the drop-down arrow next to the data source.

Tableau support for Web Data Connector

Tableau provides limited customer support for connections using the **Web Data Connector**. Tableau does not provide support for connectors or for other programs written to interface with the WDC API. However, you can submit questions and ask for help on the Tableau developer community forums.

Tableau *does* provide support for the WDC library and SDK though. If you find an issue with the WDC library, the simulator, or any of the developer samples, submit an issue on Github.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Zendesk

This article describes how to connect Tableau to Zendesk and set up the data source.

Before you begin

Before you begin, gather this connection information:

- Support site URL for the application instance.
- Authentication information

Driver required

This connector requires a driver to talk to the database. You might already have the required driver installed on your computer. If the driver is not installed on your computer, Tableau displays a message in the connection dialog box with a link to the Driver Download page where you can find driver links and installation instructions.

Make the connection and set up the data source

1. Start Tableau and under **Connect**, select **Zendesk** . For a complete list of data connections, select, **More** under **To a Server**. Then do the following:
    a. Enter the support site URL.
    b. Enter the Authentication information.
    c. Select **Sign In**.

    If Tableau can't make the connection, verify that your credentials are correct. If you still can't connect, your computer is having trouble locating the server. Contact your network administrator or database administrator.
2. On the data source page, do the following:

    (Optional) Select the default data source name at the top of the page, and then enter a unique data source name for use in Tableau. For example, use a data source naming convention that helps other users of the data source figure out which data source to connect to.
    a. Under **Table**, select a table or search for a table by name.
    b. Drag the table to the canvas, and then select the sheet tab to start your analysis.
    c. Use custom SQL to connect to a specific query rather than the entire data source.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.

Other Databases (JDBC)

When you start Tableau, under **Connect**, you can see the file and database types that are supported by Tableau Desktop. Select **More** to see the complete list. For supported files and databases, Tableau provides built-in connectors that are built for and optimized for those data sources.

> **Important:** If your file or database type is listed under **Connect**, use that named connector to connect to your data.

If your file or database type is not listed on under **Connect**, you might be able to connect to it using the **Other Databases (JDBC)** connector. JDBC (Java Database Connectivity) is a standard way to connect to a database. If the database driver you want to connect with implements the JDBC standard, you can connect Tableau to your data using the JDBC driver for your database and the Tableau **Other Databases (JDBC)** connector.

When you use the **Other Databases (JDBC)** connector, you might notice significant performance differences between this "generic" connector and a named Tableau connector because the generic JDBC connection doesn't take advantage of the many connection-specific features used in a named connector to optimize performance.

Before you begin

Specify the right JDBC driver

JDBC drivers are self-contained .jar files. Place the .jar files in the folder for your operating system. (You need to create the folder if it doesn't already exist.)

- Windows: C:\Program Files\Tableau\Drivers
- Mac: ~/Library/Tableau/Drivers
- Linux: /opt/tableau/tableau_driver/jdbc

> **Note:** If the driver includes only a single .jar file, simply copy it to C:\Program Files\Tableau/Drivers. If the driver includes multiple files, create a subfolder "A" under C:\Program Files\Tableau/Drivers, then copy all files to folder "A".

Also note these requirements:

- You must have read permissions on the .jar file.
- Tableau requires a JDBC 4.0 or later driver.

- Tableau requires a Type 4 JDBC driver.
- Install the latest 64-bit version of Java 8.

Dialect support

The **Other Databases (JDBC)** connector supports the following dialects, which are used to establish how Tableau translates the SQL queries that are sent to the database:

- MySQL
- PostgreSQL
- Generic SQL-92

Gather connection information

Gather the following connection information:

- JDBC connection string to enter in the URL field when you connect. Refer to the driver's documentation to make sure you use the correct format for your JDBC driver. Here is one example:

  `jdbc:postgresql://www.exampledatabase.com:5432/databasename`
  - `jdbc` - always use the jdbc prefix, or else you'll disable the sign in button.
  - `postgresql` - defines the class of the JDBC driver you're using. Tableau checks for a matching driver in the Tableau driver folder. The JDBC driver has a sub-protocol associated with it, such as postgresql for PostgreSQL or mysql for MySQL.
  - `www.exampledatabase.com` - the network address to your database server. You can use a host name or an IP address.
  - `5432` – the port that the database is responding to at the network address specified.
  - `databasename` - the name of the database or schema on your database server.
- The dialect you're using: MySQL, PostgreSQL, or SQL-92.
- Credentials for signing in to the server: user name and password.
- (Optional) JDBC properties file to customize the driver behavior. For more information, see Customize JDBC Connections Using a Properties File in Tableau Community.

Make the connection

To connect to your data using the JDBC driver, do the following:

1. Start Tableau and under **Connect**, select **Other Databases (JDBC)**. For a complete list of data connections, select **More** under **To a Server**.

2. Enter the JDBC connection string in **URL** field.
3. Select the **Dialect** from the drop-down list.
4. Enter the user name and password for signing in to the server.
5. (Optional) Browse to the JDBC properties file. When you use a properties file, it overrides class-level properties.
6. Select **Sign in**.

> **Important:** When you use **Other Databases (JDBC)** to connect to a database, the outcome may vary and compatibility with Tableau Desktop features is not guaranteed.

If your connection works as you need it to, you can begin your analysis. If, however, you need to customize your JDBC connection to improve functionality and performance, refer to the articles under **See also** on page 674.

Configure for publishing and cross-platform portability

When you work with a generic JDBC connection, additional configuration is required to make sure that the workbooks and data sources you create can be used on different computers and platforms.

For publishing and for cross-platform portability, the driver must be installed to match the workbook or data source connection.

As an example, Ashley Garcia creates an **Other Databases (JDBC)** data source that connects using the associated JDBC driver installed on her Windows computer.

Ashley can do a couple of things with this data source to make it available for others to use.

- She can publish it to Tableau Server. When she does this, Ashley needs to ask the Tableau Server administrator to install the associated JDBC driver on the Tableau Server computer so that the data source can be accessed by Ashley and by others.
- She can email the data source to coworkers. Ashley needs to tell the coworkers to install the associated JDBC driver on their computers so that they can open the data source in Tableau Desktop.

When you try to open a workbook or data source created using the **Other Databases (JDBC)** connector and you don't have the JDBC driver installed, you'll see an error that includes the message `No suitable driver`. To resolve this error, contact the workbook or data source publisher or the Tableau Server administrator to get the JDBC driver that matches the workbook or data source. Then install the driver on your computer.

Review your data

Tableau probably hasn't tested the database your connecting to using the **Other Databases (JDBC)** connector, so it's important that you take some time to review your data to ensure that it reflects what you expect.

To check the metadata, review the data tab and the data grid on the data source page to ensure that you see the right tables and columns, and that the data types are correct.

Change your database

When you connect to data using **Other Databases (JDBC)**, you specify the database that you are connecting to in the JDBC connection string in the **URL** field. After you connect, if you want to change to another database, you need to edit the connection string or add a new connection with the new string. Otherwise you will receive an error and see results like the following examples:

Tableau Desktop

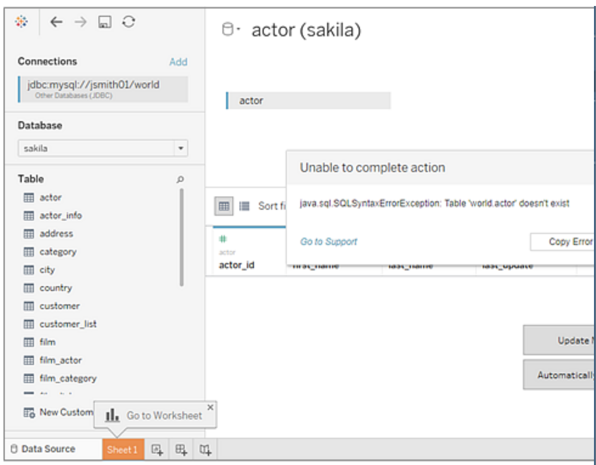


Tableau Prep Builder

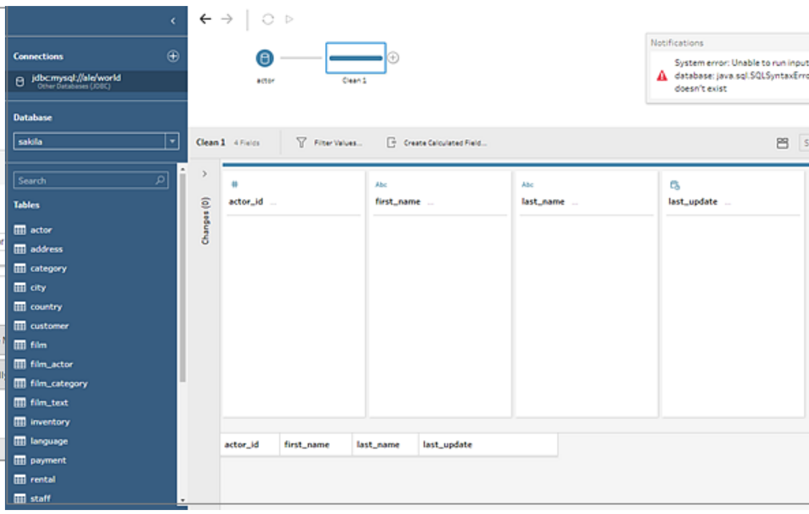


Tableau support for Other Databases (JDBC) connections

Tableau provides limited customer support for connections using **Other Databases (JDBC)**. For example, Support will help you if your connection works in Tableau Desktop, but doesn't work in Tableau Server. However, Tableau does not support creating or debugging JDBC properties files. For help, you can review the documentation listed under **See also** on the next page, and you can submit questions and ask for help on the Tableau Developer Community forums.

See also

Tableau and JDBC – Provides tips for improving performance and solving common problems.

Customize JDBC Connections Using a Properties File in Tableau Community

Customizing JDBC Connections in Tableau Knowledge Base

Tableau and JDBC

Tableau supports JDBC (Java Database Connectivity) drivers for several connectors. This includes both official named connectors, such as Athena, and generic connector for use with other JDBC-based drivers, called **Other Databases (JDBC)**.

JDBC requires Java, which makes it platform-independent. Starting with version 2020.2, all Tableau applications are bundled with the Java Runtime, so you don't have to install it separately.

> **Note:** If you are using a version of Tableau Desktop before 2020.2, see the Driver Downloads page for instructions on installing Java.

Tune JDBC connector performance

Tableau supports the ability to customize your JDBC data connection, which can improve the connection experience. For more information, see Customize and Tune Connections.

Tableau support for JDBC connections

Tableau provides no guarantee or warranty that using the **Other Databases (JDBC)** connector with any particular JDBC driver or database will be able to successfully connect and query data. Some JDBC drivers will support the full Tableau interactive experience, while others may work only for creating extracts. Some JDBC drivers might not work with Tableau.

> **Note:** Tableau will provide reasonable levels of customer support to assist in troubleshooting connections with JDBC drivers, but can't create or customize a connector to work with a specific JDBC driver.

Frequently asked questions

Here is a list of some common problems and steps to resolution.

I'm seeing a "Missing Java" message. What now?

This sometimes happens if you are using a version of Tableau before 2020.2 or using Kerberos on Windows. See the Driver Downloads page for instructions on installing the correct version of Java for your connector.

Starting with version 2020.2, all Tableau applications are bundled with the Java Runtime, so you won't have this error.

I'm seeing a "Missing Driver " message. What should I do?

Install the driver, and any libraries it depends on, in one of the following paths:

- Windows: C:\Program Files\Tableau\Drivers
- Mac: /Library/JDBC or ~/Library/JDBC
- Linux: /opt/tableau/tableau_driver/jdbc

Ensure that the driver is readable by the user running Tableau, or the Tableau service user.

I have multiple JDBC drivers installed for Tableau, but I'm having trouble connecting. What should I do?

If you have multiple JDBC drivers installed for Tableau, they may use different version of the same third-party library. This may cause problems in unpredictable ways. If you have strange errors while connecting, and you have multiple JDBC drivers installed in the JDBC drivers folder, then you can use the following steps to isolate them.

Suppose you have these libraries installed in the JDBC folder:

- athena-driver.jar
- athena-dependency.jar
- athena-dependency.so
- postgres-driver.jar

To isolate the Athena driver from the Postgres driver, install them in separate subdirectories:

- Athena/athena-driver.jar
- Athena/athena-dependency.jar
- Athena/athena-dependency.dll
- Postgres/postgres-driver.jar

After you have moved the drivers, restart Tableau.

I'm seeing a "ClassNotFoundException" (or similar) message. What should I do?

If you are running a JDBC driver that is not Java 11 compatible, you may experience problems running Tableau.

If you think your issues might be caused by Java 11:

1. Download Java 64-bit JRE (Java Runtime Environment) or JDK (Java Development Kit) from AdoptOpenJDK:
   a. Go to https://adoptopenjdk.net/releases.html.
   b. Select **OpenJDK 11**, and select **Hotspot**.
   c. Select **Other Platforms** to view a full list of options.
   d. Download the JRE or JDK for your operating system.
2. On the computer running Tableau, set an environment variable like one of the following:

   TABLEAU_JAVA_HOME=C:\Program Files\AdoptOpenJDK\jre-8.0.232.09-hotspot

   or

   TABLEAU_JAVA_HOME=C:\Program Files\AdoptOpenJDK\jdk-8.0.252.09-hotspot

How do I resolve proxy problems?

Starting with version 2019.4, when Tableau runs on Windows, it reads proxy settings by enabling the `useSystemProxies` option for Java.

In some cases, this can cause problems. For example, if you select "Use the same proxy server for all protocols," the Java Virtual Machine tries to use a SOCKS proxy.

To find instructions for resolving this issue, see the Tableau Knowledge Base article called Tableau Fails To Connect to Amazon Athena through Proxy With Error Code 37CE01A3. This article refers to Athena specifically, but this can apply to any JDBC connector.

I'm having trouble connecting to a server with SSL. What can I try?

If you see an error message like "Could not find SSL verification path," you may need to configure a certificate or private keys.

See One-Way SSL for JDBC Connections for instructions on using a certificate. If you are using an Oracle JDBC driver, see Oracle JDBC Connections with SSL for information on using the appropriate certificates and/or key to connect.

Using Other Database (JDBC) connector with a Postgres JDBC driver, attempts to create extracts sometimes fail. What can I try?

You may run into errors if the Postgres driver fetch size isn't set and if auto-commit is enabled. The driver will attempt to handle too much data at a time, and won't be able to.

Try the following:

1. Set the fetch size for your data using defaultRowFetchSize.

   See the JDBC Driver page on the PostgreSQL site for details.
2. Disable auto-commit by creating (or editing an existing) TDC file. Include the following in the TDC file:

```
<?xml version='1.0' encoding='utf-8' ?>
<connection-customization class="genericjdbc" enabled="true"
version="10">
<vendor name="genericjdbc"/>
<driver name="postgresql"/>
<customizations>
<customization name="CAP_JDBC_QUERY_DISABLE_AUTO_COMMIT"
value="yes"/>
</customizations>
</connection-customization>
```

   Note: While this step is required when you use the Other Database (JDBC) connector, for named Tableau connector it is not because auto-commit is already set by default.

   For more information on using a TDC file for JDBC connections, see Customize and Tune Connections.

See also

Other Databases (JDBC) – Describes how to connect to your data using the generic JDBC connector.

Customize and Tune Connections - Describes how to use TDC and PROPERTIES files to customize connections.

One-Way SSL for JDBC Connections – Provides instructions on using a certificate with a JDBC driver to access a server with SSL.

Tableau JDBC Capability Customizations Reference – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

One-Way SSL for JDBC Connections

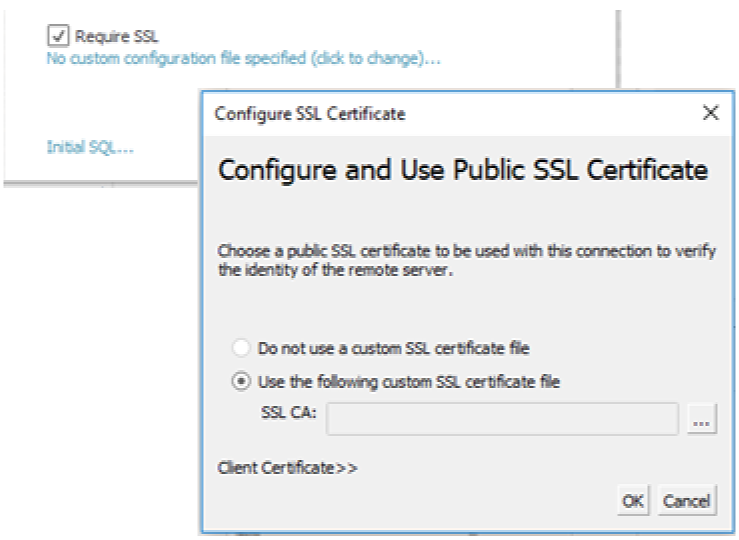
If you are using regular (one-way) SSL with a JDBC based connector, and you have self-signed certificates, or certificates that are signed by a non-public certificate authority (CA), then you will need to configure trust for your certificate.

With Tableau, you can configure one-way SSL trust for JDBC connections using one of these methods:

- **Embed a certificate** below
- **Install a certificate in the system trust store** below
- **Use custom driver properties** on page 681

Embed a certificate

Some connectors support embedding the certificate in the workbook or data source. If that is available, then you can use Tableau Desktop to embed the CA certificate.



Install a certificate in the system trust store

If embedding is not available, you will need to configure the Java Runtime to trust your certificate. Also, it may be easier to do this than to embed the certificate in every workbook.

For Windows:

You can install your CA or self-signed certificate in the Windows root CA trust store. The Java Runtime looks for trusted CAs in the system root trust store. It does not look in the intermediate certificate storage.

> **Note:** If you have installed your root CA, but are still having trouble making connections, it may be caused by missing intermediate certificates. While the TLS standard requires that servers send all certificates in their chain except the root certificate, not all servers are compliant. If you server doesn't send the intermediate certificates, you can either fix the server to properly forward intermediate certificates or install the intermediate certificates in the root trust store. Alternatively, you could choose to embed certificates in the data source or configure a trust store with driver properties.

1. In Windows, search for "certificates".
2. Select **Manage computer certificates**.
3. From the **Action** menu, select **All Tasks**, and then, depending on Windows version, do one of the following:
   - Select **Import** and then select **Local Machine**.
   - Select **Find Certificates**.
4. Browser to find your certificate file.
5. Import into "Trusted Root Certificate Authorities".

For Mac:

To install a custom certificate on a Mac, follow these steps to import the certificate into the "System" keychain.

> **Note:** Loading certificates from keychain on Mac works for most, but not all, drivers. In a small number of cases, you may need to use a PROPERTIES file to configure truststore. For more information, see Customize and Tune Connections.

1. Go to https://support.apple.com/guide/keychain-access/add-certificates-to-a-keychain-kyca2431/mac.
2. Import the certificate into the "System" keychain (not "System Roots").

3. Enable trust as follows:
   a. In the Keychain App, right-click the new certificate.
   b. Select **Get Info**.
   c. In the dialog, open the **Trust** section, and then select **When using this certificate always trust**.

> **Note:** For SAP HANA connections with Tableau versions before 2020.2, you will add the certificate to the JRE instead on a Mac. For details, see the "Install trusted SSL certificates on the Mac" section in the SAP HANA connector Help topic.

For Linux:

Many Linux distributions will generate a trust store in Java format from the system certificates. You may need to install Java from the package manager for this file to be created.

This allows the JRE to use the same certificates as the operating system.

> **Note:** Tableau Server looks for this file in the standard locations:
> `/etc/ssl/certs/java/cacerts`
> `/etc/pki/java/cacerts`

To configure a different location, run:

```
tsm configuration set -k native_api.ConnectivityTrustStore -v
<path-to-cacerts> --force-keys
```

This file should:

- Contain all trusted CAs and self-signed certificates.
- Contain only public keys.
- Be in JKS format.
- Be readable by the Tableau unprivileged user ("run as user").
- Use default JKS password "changeit"

To install a custom CA or self-signed certificate, see the documentation for your distribution. Run the appropriate commands to generate the key store. For example:

```
update-ca-certificates
```

Use custom driver properties

You can customize JDBC connection options, including the location of the trust store, with a properties file. This is a plain-text file containing key-value pairs for each connection parameter. For details on specific property settings, see the documentation for your driver.

For example, lines in this properties file are being used to configure trust settings:

javax.net.ssl.trustStore=C:\\My_Folder\\truststore.jks
javax.net.ssl.trustStoreType=JKS
javax.net.ssl.trustStorePassword=*password*

When you create the file and save it to the correct location, the properties in the file are applied to all JDBC connections to the same data source type.

If you use the generic "Other Database (JDBC)" connector, you can specify a properties file directly in the connection dialog.

For more information, see Customize and Tune Connections.

See also

- Require SSL for Oracle JDBC Connections - Installation instructions for adding trusted SSL certificates to Oracle JDBC connections.

Other Databases (ODBC)

When you start Tableau, under **Connect**, you can see the file and database types that are supported by Tableau Desktop. Select **More** to see the complete list. For supported files and databases, Tableau provides built-in connectors that are built for and optimized for those data sources.

> **Important:** If your file or database type is listed under **Connect**, use that named connector to connect to your data.

If your file or database type is not listed on under **Connect**, you might be able to connect to it using the **Other Databases (ODBC)** connector. ODBC (Open Database Connectivity) is a standard way to connect to a database. If the database driver you want to connect with implements the ODBC standard, you can connect Tableau to your data using the ODBC driver for your database and the Tableau **Other Databases (ODBC)** connector.

Note: Tableau Prep Builder version 2019.2.2 supports connecting to data using the Other Databases ODBC connector. However, it doesn't support the same functionality as Tableau Desktop. For Tableau Prep Builder, follow the instructions in Connecting using ODBC in the Tableau Prep Builder online help.

Make the connection

To connect to your data using the ODBC driver, do the following:

1. Start Tableau and under **Connect**, select **Other Databases (ODBC)**. For a complete list of data connections, select **More** under **To a Server**.
2. Enter the information required to make the connection.

   If table or schema names are not listed, click the search icon or enter the name in the text box and click the search icon, and then select the name.

> **Note:** When you use **Other Databases (ODBC)** to connect to a database, the outcome may vary and compatibility with Tableau Desktop features is not guaranteed.

If your connection works as you need it to, you can begin your analysis. If, however, you need to customize your ODBC connection to improve functionality and performance, refer to the articles under **Other Databases (ODBC)** on the previous page.

Configure for publishing and cross-platform portability

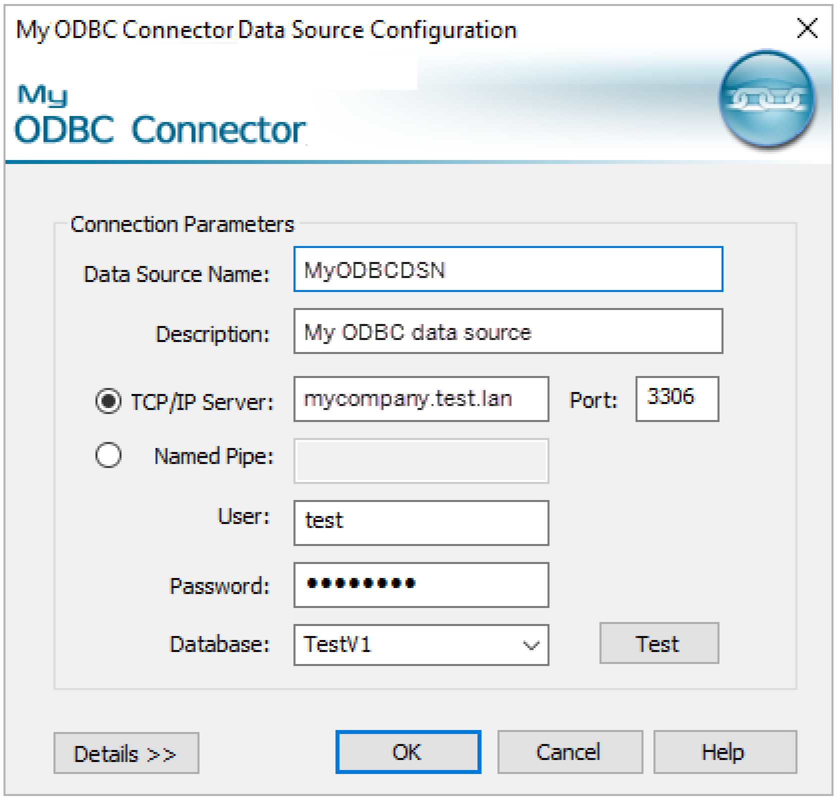
When you work with a generic ODBC connection, additional configuration is required to make sure that the workbooks and data sources you create can be used on different computers and platforms.

For publishing and for cross-platform portability:

- We recommend that you use data source names (DSNs) to accommodate different drivers and configurations between systems.
- The driver and DSN must be installed and configured to match the workbook or data source connection.

Important: When you set up a DSN, be aware that sensitive data (for example, user name and password) is plaintext in the logs and readable by anyone with access to the logs.

As an example, Ashley Garcia creates an **Other Databases (ODBC)** data source that connects using a DSN for which she has the associated ODBC driver installed on her Windows computer. Here is an example of the data source configuration, including the data source name, that Ashley created:



Ashley can do a couple of things with this data source to make it available for others to use.

- She can publish it to Tableau Server. When she does this, Ashley needs to ask the Tableau Server administrator to create the same DSN and install the associated ODBC driver on the Tableau Server computer so that the data source can be accessed by Ashley and by others.
- She can email the data source to coworkers. Ashley needs to tell the coworkers to install the associated ODBC driver and create the same DSN on their computers so that they can open the data source in Tableau Desktop.

When you try to open a workbook or data source created using the **Other Databases (ODBC)** connector and your workstation isn't configured correctly (you don't have the workbook or data source's DSN configured, or you don't have the ODBC driver installed), you'll see an error message that begins `Generic ODBC requires additional configuration`. To resolve this error, contact the workbook or data source publisher or the Tableau Server administrator to get the DSN and ODBC driver information that matches the workbook or data source. Then install the driver and configure the DSN on your computer.

Create a DSN

The method you use to create a DSN depends on your operating system.

- **Windows.** Create a DSN using the ODBC Driver Manager.
- **OS X.** Create a DSN using an ODBC Manager utility. If you don't have the utility installed your Mac, you can download one from, for example, http://www.odbcmanager.net/. Or you can manually edit the odbc.ini file.

Review the documentation for your operating system for specific instructions on how to create a DSN.

Tableau support for Other Databases (ODBC) connections

Tableau provides limited customer support for connections using **Other Databases (ODBC)**. For example, Support will help you with the sample code provided in Tableau documentation, or if your connection works in Tableau Desktop, but doesn't work in Tableau Server. However, Tableau does not support creating or debugging custom TDC files. For help, you can review the other topics in this section, and you can submit questions and ask for help on the Tableau Developer Community forums.

Tableau and ODBC

Tableau includes built-in connectors for many databases. These connectors take advantage of capabilities and optimizations specific to these data sources, and they're developed and tested to ensure they are robust and provide good performance.

But you might be working with a database that Tableau doesn't have a connection for. In that case, you still might be able to connect Tableau to the database. Tableau has a connector that uses the ODBC standard. By using ODBC, you can access any data source that supports the SQL standard and implements the ODBC API.

ODBC basics

ODBC (Open Database Connectivity) is an industry standard that enables a wide variety of software to access data. The basis of ODBC is a standard syntax for SQL queries that is used by software applications to connect to and request data from databases. The ODBC driver, usually provided by the database vendor, accepts requests in this standard syntax and converts the request to the native format that the target database prefers. In effect, the ODBC driver is a translation layer to convert from a general-purpose request to the database-specific request.

For example, the ODBC specification details that dates should be specified in SQL statements by using the syntax: `{d 'yyyy-mm-dd'}`. This format is then translated by the driver to the proper date syntax of the actual database. If you use the ODBC connector to connect to an Oracle database, the ODBC connector sends a request to the Oracle ODBC driver in this format:

```
select name from emp where birthdate > {d '1987-12-29'}
```

The driver converts your request to the actual format the Oracle database requires, which is this:

```
select name from emp where birthdate > '29-DEC-87'
```

The ODBC specification includes syntax for making function requests, join syntax, data types, and data type conversions. The SQL language itself supports complex concepts such as nested queries, correlated subqueries, temporary tables, and a variety of functions that can be used in select, where, group by, join clauses, and more. The ODBC driver is responsible for the proper conversion of all these requests to the syntax of the target database.

How Tableau determines the capabilities of the ODBC driver

There are differences in how each database vendor implements capabilities of the ODBC standard. Tableau uses a feature-discovery API in ODBC to question a database driver about the functionality it supports. One example of how Tableau changes behavior based on what the driver reports is the list of functions available when creating a calculated field. Fewer driver limitations means that more functions are available.

In some cases Tableau will require you to create a data extract from the data returned by the ODBC connector. And there are some ODBC drivers and databases that Tableau can't connect to.

ODBC discovery

During the initial connection, Tableau performs requests to the driver and SQL discovery queries in order to determine the capabilities of the driver.

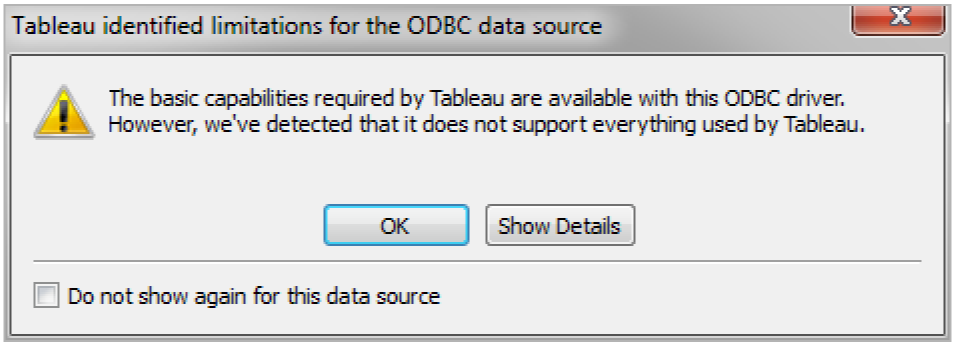
Some examples of the capabilities discovery that Tableau performs include:

- What scalar and aggregate functions are available.
- What date and time manipulation functions are available.
- Can temporary tables be created and can if select into statements can be used.
- Are subqueries supported.
- Are top and limit queries supported.
- What join styles are supported (outer, inner, full).
- What data types are supported.

Based on the outcome of the capabilities discovery, Tableau will classify the current connection as being in one of four categories:

- **Fully Functional.** This is a driver that supports all the functions and capabilities that Tableau uses.
- **Minor limitations.** This is a driver that has a small number of non-critical limitations. An example would be a driver that doesn't support the full set of numeric, string, or date functions that Tableau normally makes available. Tableau will modify its behavior to account for these limitations and proceed.
- **Major limitations.** This is a driver that has a large number of limitations or critical capabilities that Tableau relies upon but that are not supported. However, even with these major restrictions, Tableau might be able to connect well enough to create a Tableau extract to retrieve the data into Tableau for further work offline from the database. In this case, Tableau will recommend that you create an extract. When you create the extract, you might not be able to create a filter on the data source to reduce the amount of data in the extract. (For more information, see **Filter Data from Data Sources** on page 1006.) After the data has been extracted, full Tableau functionality is available when working with the extract.
- **Fatal limitations.** This is a driver that does not support even the most minimal set of capabilities that Tableau needs in order to connect and run the basic queries to create extract files. Therefore, Tableau is unable to proceed any further with this driver.

After the connection is established, if Tableau has determined that the capabilities available classify this connection as anything less than fully functional, a message is displayed to inform you of the limitations that have been detected. For example, a connection to an ODBC driver with minor limitations displays this message:



For more critical limitations, the warning dialog box will advise you to create a Tableau extract in order to proceed.

Tune ODBC connector performance

Tableau supports the ability to customize your ODBC data connection, which can improve the connection experience. For more information, see **Customize and Tune a Connection** on page 693.

Tableau support for ODBC connections

Tableau provides no guarantee or warranty that using Tableau with any particular ODBC driver or database will be able to successfully connect and query data. Some ODBC drivers will support the full Tableau interactive experience, while others may work only for creating extracts. Some ODBC drivers might not work with Tableau.

> **Note:** Tableau will provide reasonable levels of customer support to assist in troubleshooting connections with ODBC drivers, but can't create or customize a connector to work with a specific ODBC driver.

ODBC FAQ

What's a typical use case for the ODBC connector?

The most common use case for an ODBC connection is to access data specifically to retrieve it into a Tableau extract. Bringing this data into a Tableau extract then allows the full Tableau capabilities to work with the data. Many ODBC drivers support the needed functionality to connect and perform the simpler queries required to perform an extract. Whether you extract all of the data or just select a few columns and set some filters to retrieve a relevant subset, this use of ODBC is a good scenario to keep in mind.

Where do I get ODBC drivers for my database?

Most database vendors distribute ODBC drivers for use with their databases. Contact your database vendor to obtain the drivers. Additionally there are a number of third party manufacturers of ODBC drivers that can provide drivers for a variety of common databases.

What ODBC version driver do I need?

Be sure your ODBC driver is a version 3 or higher driver, meaning it implements the ODBC specification version 3. Each driver provider will have their own version numbering system for their driver versions, which will likely be different from the ODBC version number it implements. The ODBC version 3 specification was introduced in 1995, so it is likely your database provider has a driver that implements this level of compliance. We have seen a number of Tableau customers with drivers that are not version 3 compliant (which fail to connect), who then upgraded to a newer driver and were able to proceed.

Has Tableau tested database [x]?

Tableau has tested ODBC connections with many data sources, but due to the volume of ODBC drivers available on the market we haven't tested against every possible driver, nor can Tableau perform this breadth of testing. The best recommendation for any particular driver is to give it a try, and let us know how it works.

What do I do if it doesn't work?

First check the driver version. If you look at the Tableau logs we will echo back the ODBC version level the driver reports. Search the logs for an entry like the following:

```
ODBCProtocol: driver ODBC version: 03.52
```

The number at the end indicates the ODBC version level. If it is less than 03.00, the driver needs to be upgraded.

If you get warnings when you connect about capabilities not supported, check with your database vendor to see if they have an updated driver that supports more capabilities. Not all drivers will work with Tableau.

Should I use the named database connector or the ODBC connector?

If you are connecting to a database for which Tableau has a named connection option, then use the named connector. The named connectors are optimized for the particular database.

See also

Other Databases (ODBC) on page 681 – Describes how to connect to your data using the ODBC connector.

Customize and Tune a Connection on page 693 – Describes how to customize connection information for improved functionality and performance.

Tableau Capability Customizations Reference on page 713 – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

ODBC/SQL Customizations Reference on page 731 – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

Use Partner-Built Connector from the Extension Gallery

Tableau Desktop has dozens of supported connectors you can choose from. You can also find more connectors built by partners in the Extension Gallery.

> **Note:** Connectors from the Extension Gallery are not currently supported for use with Tableau Bridge.

For information about creating your own connector and submitting it to the Extension Gallery, see Connectors Built with the Tableau Connector SDK.

Choose and install a partner-built connection

1. Go to the Extension Gallery Connectors page at https://extensiongallery.tableau.com/connectors.
2. Click a tile for the connector you want to use.
3. Follow the on-screen instructions to install the connector. (Note that each connector page also has a **Support** link in case you need help.)
4. Restart Tableau Desktop.

Now the new connector will appear in the list of connectors in Tableau Desktop.

Make the connection

To connect to your data using the connector you installed from the Extension Gallery, do the following:

1. Start Tableau and under **Connect**, select the name of the connector you installed. For a complete list of data connections, select **More** under **To a Server**.
2. Enter the information required to make the connection.

   If table or schema names are not listed, click the search icon or enter the name in the text box and click the search icon, and then select the name.

Work with your data

When you connect to data with a partner-built connector, it offers the same features as Tableau-supported connectors. Once you are connected, you can:

- Define a data source
- Visualize your data
- Publish to a server (when the server has the connector installed)

> **Note:** Publishing to Tableau Online is not currently supported with partner-built connectors.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Connectors Built with the Tableau Connector SDK - Create your own connector and submit it to the Extension Gallery

Connectors Built with the Tableau Connector SDK

Tableau has great connectivity that allows you to visualize data from virtually anywhere. In addition to its dozens of built-in connectors and additional partner-built connectors (available on Tableau Extension Gallery), Tableau gives you the tools to build a customized connector with the Tableau Connector SDK.

> **Note:** Connectors built using the Tableau Connector SDK are sometimes called plugins because they “plug into” Tableau so you can connect to your data.

About Tableau Connector SDK and connector files

You can use the Tableau Connector SDK to build and custom a connector. Partner-built connectors (whether built by you or someone else) typically support most of the same features as built-in Tableau connectors. These include defining a data source, visualizing your data, publishing to a server (if the server has the connector), and so on.

Each connector is made up of a set of XML and JavaScript files, zipped into a single packaged .jar file, and give a ".taco" file extension. This TACO file is signed by the developer using a trusted public certificate authority. The XML and JavaScript files are configuration files that describe:

- UI elements needed to collect user input for creating a connection to a data source
- Any dialect or customizations needed for the connection
- How to connect using the ODBC or JDBC driver

A connector that is developed using the Tableau Connector SDK is appropriate for connecting to an ODBC or JDBC driver which interfaces using SQL. The underlying technology works well with relation databases.

> **Note:** Partner-built connectors are not supported by Tableau. However, if you use a partner-built connector from the Extension Gallery, click the **Support** link on that connector's page for help.

Build your own connector

To build your own customized connector, use the tools and instructions in the Tableau Connector SDK repository on GitHub.

> **Note:** If you find an issue with the SDK or any of the developer samples, submit an issue on GitHub.

Submit a connector to the Extension Gallery

If you would like to submit your connector to the Tableau Extension Gallery, follow steps outlined in the Tableau Connector SDK (https://tableau.github.io/connector-plugin-sdk/docs/gallery-submission).

Use a connector built with Tableau Connector SDK

After you build a connector with the Tableau Connector SDK, follow these steps to use it with Tableau:

1. Download the connector file, [connector name].taco.
2. Place the .taco file in the following directory (default location):

For Tableau Desktop	**For Tableau Prep**	**For Tableau Server**
• Windows - C:\Users\[Windows User]\Documents\My Tableau Repository\Connectors • macOS - /Users/[user]/Documents/My Tableau Repository/Connectors	• Windows: C:\Users\[Windows User]\Documents\My Tableau Prep Repository\Connectors • MacOS: /Users//Documents/My Tableau Prep Repository/ConnectorsFor Tableau For flow web authoring, on Tableau Server: • /data/tabsvc/flowqueryservice/Connectors For Tableau Prep Conductor, on Tableau Server: • /data/tabsvc/flowprocessor/Connectors	• C:\ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Connectors

3. Before you connect, install the ODBC or JDBC driver. Follow the driver installation instructions provided by the source of your data.

Once you've installed the connector, it will appear in the list of connectors in Tableau.

Connect Tableau to the data

1. Start Tableau, and under **Connect**, select the name of the connector you just installed. For a complete list of data connectors, select **More** under **To a Server**.
2. Enter the information that you're prompted for, and then select **Sign In**.
3. Wait while the connector retrieves your data into Tableau.
4. Select the sheet tab to start your analysis.

About load order for connectors

Tableau loads connectors by directory in the following order:

1. Built-in Tableau connectors
2. Connectors located in "My Tableau Repository/Connectors" (including those added manually by the user and those downloaded from the Tableau Extension Gallery)
3. Any connectors in the dev path specified by "-DConnectPluginsPath"

If a connector has the same class as a connector that is registered already, the new connector will be rejected. This means that connectors loaded first have precedence when two connectors share the same class name.

See also

- **Set Up Data Sources** on page 741 – Add more data to this data source or prepare your data before you analyze it.
- **Build Charts and Analyze Data** on page 1109 – Begin your data analysis.
- Tableau Connector SDK - Find information and tools to build and test your own customized connector.
- **Use Partner-Built Connector from the Extension Gallery** on page 689 - Find and install a connector from the Extension Gallery.

Customize and Tune a Connection

You can make customizations to a connection using a TDC file.

If you are customizing a JDBC-based connection, you can also make customizations in a PROPERTY file. For details, see Use a PROPERTIES file to customize a JDBC connection.

Use a TDC file to customize a connection

TDC (Tableau Datasource Customization) files are XML files with a “.tdc” filename extension. These files are used to customize Tableau-specific settings. For ODBC-based connection, you can also use a TDC file to define parts of the ODBC and SQL standards that the ODBC driver supports.

A TDC file contains a vendor name, driver name, and `<connection-customization>` section. For a built-in connector, the vendor name and driver name must match the connector’s class name. In most cases, the vendor name matches the driver name. Two generic vendor names are used for the Other Databases connectors:

- For the Other Databases (JDBC) connector, the required vendor name is "genericjdbc".
- For Other Databases (ODBC) connector, the required vendor name is "genericodbc".

If an existing workbook or data source file has an enabled customization section already, Tableau uses only the customizations it defines and not any defined in the TDC file.

For more information, see Using a TDC File with Tableau Server in Tableau Knowledge Base.

Important: Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

Structure of a TDC file

Each TDC file follows this basic structure:

```
<connection-customization class=DSCLASS enabled='true' ver-
sion='10.0'>
    <vendor name=VENDOR />
    <driver name=DRIVER />
    <customizations>
        <customization name='CAP_FAST_METADATA' value='yes'/>
        ...
    </customizations>
</connection-customization>
```

A TDC file has three main sections:

- Connection customization variables
- Vendor and driver names
- The customizations themselves

The `connection-customization` section includes the following:

- **Class** The data source you want to connect to and customize.
- **Enabled** Specifies whether the TDC file connection customizations are applied. In the TDC file, always set to “true”.
- **Version** Tableau ignores the version number.

The next section is crucial because it identifies the vendor name and driver name of the database provider for this TDC file's data source. Each TDC file can bind to only one data source type. For our native data sources, these names must both match the data source class name; for example, “teradata” for our Teradata connections. For ODBC data sources, the vendor name and driver name of the TDC file must match what is reported to Tableau by the database and driver; for example, “SQLite” and “SQLite3 ODBC Driver”, respectively.

The final section lists the actual connection customizations.

For more information, see Tableau JDBC Capability Customizations.

Use a PROPERTIES file to customize a JDBC connection

PROPERTIES files are used by JDBC and passed directly through to the JDBC driver. These are plain-text files with a “.properties” filename extension. They contain key-value pairs for each connection parameter.

Using a PROPERTIES file with a JDBC connection is similar to using the `odbc-connect-string-extras` parameter in a TDC file for ODBC connections.

> **Note:** A PROPERTIES file should be in Latin-1 format (see https://en.wikipedia.org/wiki/.properties). However, as long as it contains only ASCII characters, you can safely save the file in UTF-8 format without a BOM (byte order mark).

In the file, place each key-value pair on its own line with no delimiter. Give the file the same name as its corresponding connector, but with a “.properties” filename extension. For example, the PROPERTIES file for the SAP HANA connector would be “saphana.properties”.

For Tableau Desktop, place PROPERTIES files in this folder:

```
My Tableau Repository/Datasources
```

To enable a PROPERTIES file for Tableau Server, place it in this folder for each Server node:

- Windows: `ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Datasources`
- Linux: `/var/opt/tableau/tableau_server/data/tabsvc/vizqlserver/Datasources/`

Once you save the file to the correct location, the properties will be applied to all JDBC connections for that same data source type.

Publish a workbook using a JDBC-based connection

To publish a workbook from a JDBC-based connection, you must include a copy of the PROPERTIES file on both Tableau Desktop and Tableau Server computers.

See also

Customize the Connection String for a Native Connector – This article in Tableau Community describes how to modify the connection made with a native connector using the `odbc-connect-string-extras` customization.

Example: Customize an ODBC Connection

When you use a connector for a database that supports SQL, Tableau generates SQL statements that are tuned for that database. Because Tableau has no representation of the SQL dialect used by an ODBC data source, it must infer the supported syntax through a variety of tests. If a driver reports incorrect or incomplete information about the SQL dialect it supports, you can use Tableau customizations to fine tune this connection information in order to improve functionality and performance.

In this article, you will make an ODBC connection, examine the resulting Tableau Data Source (TDS) file and use part of it to create a Tableau Data source Customization (TDC) file that you can then use to customize your ODBC connection. Before you begin, you should be familiar with the content in **Tableau and ODBC** on page 684.

> **Note:** Tableau will provide reasonable levels of customer support to assist in troubleshooting connections with ODBC drivers, but can't create or customize a connector to work with a specific ODBC driver.

Make an ODBC connection

This section shows you how to create an ODBC connection using an example. In the example, you connect to a SQLite database using the SQLite ODBC driver.

Prerequisites

The ODBC connection in this article is based on SQLite (http://www.sqlite.org/), an open-source database.

You need to download the following two items:

> **Disclaimer:** This information refers to a third-party product. This example is not an endorsement of this product over any other competing products.

- The SQLite 32-bit ODBC driver (required for 32-bit or 64-bit Windows). Download and install the following (select the SQLite 2 drivers):

  http://www.ch-werner.de/sqliteodbc/sqliteodbc.exe

- A sample SQLite database created from the Microsoft example Northwind database. Download and expand the following:

  http://download.vive.net/Northwind.zip

Create a connection

To create an ODBC connection, you connect to the Northwind database using the SQLite3 ODBC driver, and then save the connection as a Tableau Data Source (TDS) file.

1. Open Tableau Desktop.
2. On the start page under **Connect**, click **Other Databases (ODBC)**.

   Note: For a database that already has a Tableau-supported connector, you can click the name of that connector to create a connection.

3. Under **Connect Using**, select **Driver**, and then from the drop-down list, select **SQLite3 ODBC Driver**.
4. Click **Connect**.
5. Next to the **Database Name** text box, click **Browse**, navigate to the location of your **Northwind.sl3** file, and then click **Open**.
6. Click **OK** to close the dialog box, and then click **Sign In**.

7. On the data source page, in the **Table** text box, enter `Orders`.

8. Drag the **Orders** table to the canvas, and then click the sheet tab.

   A "Tableau identified limitations for the ODBC data source" dialog box opens. You can review the details or dismiss the dialog box and continue. For more information, see How Tableau determines the capabilities of the ODBC driver.

9. Select **Data** > **Orders [path to Northwind.sl3]** > **Add to Saved Data Sources**.

10. Click **Save**.

11. Close the workbook. When asked to save changes, click **No**.

Now you can open the TDS file to examine the SQLite connection.

Review the XML structure of a TDS file

Open the saved Tableau Data Source (TDS) file in a text editor to view its XML structure. By default, the file you created above is named `Northwind.sl3.tds` and is in the following path:

```
Users\[your name]\Documents\My Tableau Repository\Datasources
```

or

```
Users\[your name]\Documents\My Tableau Repository (Beta)\Data-
sources
```

This is an XML document describing the SQLite connection to the Northwind Orders table. Within the `<connection>` section, find the `<connection-customization>` element, which contains `<customization>` elements that you can edit.

Vendor and driver name

The connection customization section begins with the names Tableau detected for the ODBC driver and the database vendor. Tableau uses this information to associate a specific connection customization with a single type of ODBC data source. The section looks like this:

```
<connection-customization class='genericodbc' enabled='false' ver-
sion='10.1'>
<vendor name='SQLite' />
<driver name='SQLite3 ODBC Driver' />
<customizations>
```

Types of customizations

Tableau permits two types of customizations: Tableau-specific capabilities, and ODBC API calls to `SQLGetInfo`. These customizations are made of name/value pairs, with the names following a convention of `CAP_` for Tableau capabilities and `SQL_` for the SQLGetInfo API calls.

The saved data source file contains examples of both types of customizations. These customizations represent the values that Tableau was able to detect by querying the driver when you connected. The list of customizations might be incomplete or incorrect. You can use customization to modify and shape Tableau's behavior when connecting to an ODBC data source.

The following articles contain a complete reference for the customizations. You should have an idea of how you need to adjust your ODBC connection to perform as you expect in your environment. Review the lists to get an idea of the customizations you can use to tune your ODBC connection.

- **Tableau Capability Customizations Reference** on page 713
- **ODBC/SQL Customizations Reference** on page 731

Format of customization values

- The customization values are all represented as strings for each name/value pair.
- Tableau capabilities are all Boolean values represented by yes or no.
- The SQLGetInfo values can be long-integer, short-integer, or string data, depending on the expected return value of SQLGetInfo for the given identifier.
- Many of the integer fields are bitmasks that represent a collection of capabilities.

Make customizations global with a TDC file

To make your customization changes apply to all connections for a given ODBC data source, you need to create a Tableau Data source Customization (TDC) file. This file contains only the `<connection-customization>` section and is applied to any new Tableau connection that matches the database vendor name and driver name described in the TDC file (as noted in **Vendor and driver name** on the previous page). Any existing workbook or data source file that already has an enabled customization section will use only the customizations that it supplies, not the TDC file.

Important: Tableau does not test or support TDC files. These files should be used as a tool to explore or occasionally address issues with your data connection. Creating and maintaining TDC files requires careful manual editing, and there is no support for sharing these files.

Create a TDC file

When you create a TDC file, you need to save it to the right location so that it can be used by your connection. If you create a workbook that you want to publish to Tableau Server, then you need to savethe TDC file to the server as well. For more information, see Using a TDC File with Tableau Server in Tableau Knowledge Base.

Structure of a TDC file

Each TDC file follows this basic structure:

```
<connection-customization class=DSCLASS enabled='true' ver-
sion='10.0'>
    <vendor name=VENDOR />
    <driver name=DRIVER />
    <customizations>
        <customization name='CAP_FAST_METADATA' value='yes'/>
        ...
    </customizations>
</connection-customization>
```

A TDC file has three main sections:

- Connection customization variables
- Vendor and driver names
- The customizations themselves

The `connection-customization` section includes the following:

- **Class** The data source you want to connect to and customize.
- **Enabled** Specifies whether the TDC file connection customizations are applied. In the TDC file, always set to “true”.
- **Version** Tableau ignores the version number.

The next section is crucial because it identifies the vendor name and driver name of the database provider for this TDC file's data source. Each TDC file can bind to only one data source type. For our native data sources, these names must both match the data source class name; for example, “teradata” for our Teradata connections. For ODBC data sources, the vendor name and driver name of the TDC file must match what is reported to Tableau by the database and driver; for example, “SQLite” and “SQLite3 ODBC Driver”, respectively.

The final section lists the actual connection customizations.

Use the TDC file with Tableau Desktop

1. Using a text editor, copy and paste the entire `<connection-customization>` section of your saved data source file. For reference, see the Sample SQLite TDC file.
2. Name the file `odbc-sqlite.tdc` and save it to `Documents\My Tableau Repository\Datasources`.

   Note: The file must be saved with a `.tdc` extension, but the name does not matter.

3. Restart Tableau Desktop to apply the change.
4. Create a new connection to SQLite as described in Create a connection.
5. Go to and open the Tableau Desktop log file, and look for a record similar to the example below to verify that this customization file was applied to your new connection.

Log File Location	Example of Record
By default, C:\Users\<user>\Documents\My Tableau Repository\Logs	`Found matching TDC 'C:\\Users\\<name>\\Documents\\My Tableau Repository\\Datasources\\odbc-sqlite.tdc' for class='genericodbc', vendor='SQLite', and driver='SQLite3 ODBC Driver'`

Use the TDC file with Tableau Server

1. Using a text editor, copy and paste the entire `<connection-customization>` section of your saved data source file. For reference, see the Sample SQLite TDC file.
2. Name the file `odbc-sqlite.tdc` and save it to the following location:

- For Windows:

  In Tableau Server versions prior to 2018.2:

  `Program Files\Tableau\Tableau Server\<version>\bin`

  Starting in Tableau Server version 2018.2:

  `Program Files\Tableau\Tableau Server\packages\bin.<build number>`

  Or for all Tableau Server versions:

  `ProgramData\Tableau\Tableau Server\data\tabsvc\vizqlserver\Datasources`

  To save the file, you must be a Windows Administrator on the server computer.
- For Linux:

  `/var/opt/tableau/tableau_server/data/tabsvc/vizqlserver/Datasources/`

> **Note:** The file must be saved with a `.tdc` extension, but the name does not matter. The TDC file must be saved on all server nodes for consistent behavior.

3. Restart Tableau Server to apply the change.
4. Go to and open the Tableau Server log file, and look for a record similar to the example below to verify that this customization file was applied to your new connection.

Log File Location	Example of Record
<install path>\Tableau Server\data\tabsvc\log\vizqlserver <install path>\Tableau Server\data\tabsvc\log\backgrounder <install path>\Tableau Server\data\tabsvc\log\dataserver <install path>\Tableau	`Found matching TDC 'C:\\ProgramData\\Tableau\\Tableau Server\\data\\tabsvc\\vizqlserver\\Datasources\\odbc-sqlite.tdc for class='genericodbc', vendor='SQLite', and driver='SQLite3 ODBC Driver'`

Log File Location	Example of Record
Server\data\tabsvc\log\vizportal	

Sample SQLite TDC file

```
<?xml version='1.0' encoding='utf-8' ?>
<connection-customization class='genericodbc' enabled='true' ver-
sion='7.8'>
<vendor name='SQLite' />
<driver name='SQLite3 ODBC Driver' />
       <customizations>
               <customization name='CAP_CREATE_TEMP_TABLES' value='yes' />
               <customization name='CAP_QUERY_BOOLEXPR_TO_INTEXPR' value='yes'
/>
               <customization name='CAP_QUERY_GROUP_BY_ALIAS' value='no' />
               <customization name='CAP_QUERY_GROUP_BY_DEGREE' value='yes' />
               <customization name='CAP_QUERY_JOIN_ACROSS_SCHEMAS' value='no'
/>
               <customization name='CAP_QUERY_JOIN_REQUIRES_SCOPE' value='no'
/>
               <customization name='CAP_QUERY_SUBQUERIES' value='yes' />
               <customization name='CAP_QUERY_SUBQUERIES_WITH_TOP' value='yes'
/>
               <customization name='CAP_SELECT_INTO' value='no' />
               <customization name='CAP_SELECT_TOP_INTO' value='yes' />
               <customization name='SQL_AGGREGATE_FUNCTIONS' value='127' />
               <customization name='SQL_SQL_CONFORMANCE' value='4' />
       </customizations>
</connection-customization>
```

Customize your ODBC connection

Let's assume that you've made an ODBC connection to your data source and verified that you can get the metadata and data you need. Now you need to determine if creating an extract or if using the live connection as is gives you the functionality you need. If it does, the TDC file you created is sufficient, and there is no need to customize your connection.

If the TDC file is not performing the way you want it to, you can customize the connection. You should have an idea of how you need to adjust your ODBC connection to perform as you expect in your environment. Review the lists to see the customizations you can use to tune your ODBC connection.

- **Tableau Capability Customizations Reference** on page 713
- **ODBC/SQL Customizations Reference** on page 731

Common customizations for improving functionality

The following customizations can help improve functionality for data sources with default capabilities that are not optimal.

- `CAP_SUPPRESS_DISCOVERY_QUERIES` – Setting this value to `true` prevents Tableau from performing any SQL queries at the beginning of the connection to determine the driver's capabilities. You will need to consider which other capabilities should be explicitly enabled, since Tableau will not be able to determine those automatically.
- `SQL_SQL_CONFORMANCE` – This setting indicates which level of the SQL standard is fully supported by the data source. Tableau works best with at least entry-level conformance, so this value should be at least `1` if the data source supports it.
- `SQL_AGGREGATE_FUNCTIONS` – This setting indicates which aggregate functions are supported, such as `MIN`, `MAX`, `SUM`. A value of `'127'` indicates support for all standard SQL aggregate functions.

Handle severely limited data sources

Some data sources are so severely limited that Tableau is unable to complete the steps of creating a connection. Occasionally this is due to crashes within the driver, which cause Tableau to cease working. You can use a global TDC file to prevent Tableau from issuing queries or checking for capabilities that might be associated with the instability.

To create this file, you need to know the database vendor name and ODBC driver name.

1. Create a new connection in Tableau, open your log file, and then look for a line like the following:

```
GenericODBCProtocol::Connect: Detected vendor: 'SQLite' and
driver: 'SQLite3 ODBC Driver'
```

2. Create a TDC file with the listed `vendor` name and `driver` name. See **Vendor and driver name** on page 698 for an example.
3. Use customizations, such as `CAP_SUPPRESS_DISCOVERY_QUERIES`, to constrain Tableau's interaction with a limited data source.

Configure ODBC driver settings

ODBC drivers provide dialog boxes for supplying connection details such as the server, user name, and password. Many offer advanced options for controlling the connection behavior. You should consider these options when exploring ways to improve the functionality or performance of your ODBC connection. In particular, look for settings that control the items listed below, as these have been the cause of past issues with ODBC connections in Tableau:

- Transaction Isolation - Prefer `READ`, `COMMITTED`, or `SERIALIZABLE` to ensure that queries do not include data from pending transactions in their results.
- Cursors - Prefer Holdable or Scrollable over Streaming cursors. Since Tableau closes a transaction after every read-only analytical query, streaming cursors may become truncated and lead to incomplete results displayed in Tableau.
- Row/Resultset Buffers - Prefer larger buffer sizes to improve the performance of fetching numerous rows, which can greatly improve the speed of creating extracts. This is sometimes called the cache size or response size.
- Character Set - In decreasing order of preference: UTF-16, UTF-8, ASCII.

Advanced SQLite customizations

Tableau has special customizations for SQLite that are built in to the product. These customizations take precedence over any connection customizations for SQLite in the workbook file, data source file, or global TDS file. To make advanced changes to SQLite connection customizations, you need to trick Tableau into ignoring its own SQLite dialect. You can do so by changing the reported name of the database vendor (`<customization name='SQL_DBMS_NAME' value='SQLite' />`) to a different value such as `SQLite-Tableau`.

ODBC documentation reference

Microsoft's MSDN has documentation for the ODBC standard. The ODBC Appendixes are the most applicable as resources for the article information, specifically:

- Appendix C: SQL Grammar - especially the SQL Minimum Grammar and ODBC Escape Sequences
- Appendix D: Data Types
- Appendix E: Scalar Functions

See also

Other Databases (ODBC) on page 681 – Describes how to connect to your data using the ODBC connector.

Tableau and ODBC on page 684 – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.

Tableau Capability Customizations Reference on page 713 – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

ODBC/SQL Customizations Reference on page 731 – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

Customize the Connection String for a Native Connector – This article in Tableau Community describes how to modify the connection made with a native connector using the `odbc-connect-string-extras` customization.

Tableau JDBC Capability Customizations Reference

You can set the following capability customizations in the Tableau Datasource Customization (TDC) file to define which Tableau capabilities are supported by the JDBC connection.

Many of these customizations influence the type of SQL queries that Tableau issues. For settings that are not defined, Tableau attempts to determine the proper values for each capability by issuing various forms of SQL queries to experimentally verify which forms are supported.

CAP_CREATE_TEMP_TABLES	Set to 'yes' if Tableau can create temporary tables needed for certain complex or optimized queries. See also: CAP_SELECT_INTO.

CAP_CONNECT_STORED_ PROCEDURE	Set to 'yes' to allow support for connecting to a stored procedure.
CAP_FAST_METADATA	Set to 'yes' if you have small to moderate size schemas. This capability controls whether Tableau should enumerate all of the objects immediately when you connect. Set the value to “yes” to enable this capability for better performance when creating new connections. Disable this capability to allow search for specific schemas or tables instead of retrieving all objects. You can search for all objects by using an empty string. This capability is available in 9.0 and later.
CAP_ISOLATION_LEVEL_READ_ COMMITTED	Set to 'yes' to force the transaction isolation level to Read Committed if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. See also: CAP_SET_ ISOLATION_LEVEL_VIA_SQL.
CAP_ISOLATION_LEVEL_READ_ UNCOMMITTED	Set to 'yes' to force the transaction isolation level to Read Uncommitted if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. This capability can improve speed by reducing lock contention, but may result in partial or inconsistent data in query results. See also: CAP_SET_ISOLATION_LEVEL_VIA_ SQL.
CAP_ISOLATION_LEVEL_ REPEATABLE_READS	Set to 'yes' to force the transaction isolation level to Repeatable Reads if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. See also: CAP_SET_ ISOLATION_LEVEL_VIA_SQL.
CAP_ISOLATION_LEVEL_ SERIALIZABLE	Set to 'yes' to force the transaction isolation level to Serializable if the data source supports it. Only

	one of the four transaction isolation levels should be set to 'yes'. This is a very conservative setting that may improve stability at the expense of performance. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL.
CAP_JDBC_BIND_DETECT_ALIAS_CASE_FOLDING	Set to ‘yes’ to allow Tableau to detect and recover from a JDBC data source that reports the field names in a result set using only uppercase or lowercase characters, instead of the expected field names.
CAP_JDBC_EXPORT_DATA_BATCH	Set to 'no' to disable the use of JDBC batch operations for data insert.
CAP_JDBC_METADATA_GET_INDEX_INFO	Set to ‘no’ to disable reading index information.
CAP_JDBC_METADATA_READ_FOREIGNKEYS	Set to ‘no’ to disable reading foreign key metadata.
CAP_JDBC_METADATA_READ_PRIMARYKEYS	Set to ‘no’ to disable reading primary key metadata.
CAP_JDBC_METADATA_USE_RESULTSET_FOR_TABLE	Set to 'yes' to get column metadata from the result set of a select * query. Available in Tableau 2020.4 and newer.
CAP_JDBC_QUERY_ASYNC	Set to ‘yes’ to run queries on another thread.
CAP_JDBC_QUERY_CANCEL	Set to ‘yes’ if driver can cancel queries.
CAP_JDBC_QUERY_DISABLE_AUTO_COMMIT	Set to 'yes' to disable the default auto-commit mode when running query. Available in Tableau 2020.4 and newer.
CAP_JDBC_QUERY_FORCE_PREPARE	Set to 'yes' to always prepare the query before execution. Available in Tableau 2020.4 and newer.

CAP_JDBC_SUPPRESS_EMPTY_CATALOG_NAME	Set to 'yes' to ignore missing catalog.
CAP_JDBC_SUPPRESS_ENUMERATE_DATABASES	Set to ‘yes’ to disable database enumeration.
CAP_JDBC_SUPPRESS_ENUMERATE_SCHEMAS	Set to ‘yes’ to disable schema enumeration.
CAP_JDBC_METADATA_SUPPRESS_PREPARED_QUERY	If CAP_JDBC_METADATA_USE_RESULTSET_FOR_TABLE is enabled, set this capability to 'yes' to disable preparing the query used for reading the table metadata. We will execute the query wrapped with a where-false clause.
CAP_JDBC_USE_ADAPTIVE_FETCH_SIZE	Set to 'yes' to use ResultSet metadata to determine optimal fetch size. May require CAP_JDBC_QUERY_FORCE_PREPARE to be enabled to work properly. Available in Tableau 2020.4 and newer.
CAP_MULTIPLE_CONNECTIONS_FROM_SAME_IP	Set to 'no' to prevent Tableau from creating more than one active connection to the database. This is a conservative setting that may increase stability at the expense of performance.
CAP_QUERY_BOOLEXPR_TO_INTEXPR	Set to 'yes' if Tableau must coerce any Boolean expressions to an integer value in order include in a result set.
CAP_QUERY_FROM_REQUIRES_ALIAS	Set to 'yes' if the FROM clause must provide an alias for the given table.
CAP_QUERY_GROUP_ALLOW_DUPLICATES	Set to 'no' if SQL queries cannot contain duplicate expressions in the GROUP BY clause (this is uncommon).
CAP_QUERY_GROUP_BY_ALIAS	Set to 'yes' if SQL queries with aggregations can reference the grouping columns by their

	corresponding alias in the SELECT list, e.g. GROUP BY "none_ShipCountry_nk".
CAP_QUERY_GROUP_BY_DEGREE	Set to 'yes' if SQL queries with aggregations can reference the grouping columns by the ordinal position of each column, e.g. GROUP BY 2, 5. See also: CAP_QUERY_SORT_BY_DEGREE
CAP_QUERY_HAVING_REQUIRES_GROUP_BY	Set to 'yes' if Tableau must use an artificial grouping field for any query which has a HAVING clause but no grouping columns.
CAP_QUERY_HAVING_UNSUPPORTED	Set to 'yes' if the SQL syntax for HAVING is unsupported. Tableau may be able to work around this using subqueries. See also: CAP_QUERY_SUBQUERIES.
CAP_QUERY_INCLUDE_GROUP_BY_COLUMNS_IN_SELECT	Set to 'yes' to require all GROUP BY expressions to also appear in the SELECT expression list.
CAP_QUERY_JOIN_ACROSS_SCHEMAS	Set to 'yes' if SQL queries can express joins between tables located in different schemas.
CAP_QUERY_JOIN_ASSUME_CONSTRAINED	Set to ‘yes’ to cull inner joins even if the database tables does do not have FK-PK relationships.
CAP_QUERY_JOIN_PUSH_DOWN_CONDITION_EXPRESSIONS	Set to 'yes' to rewrite joins to simplify the ON clause conditions to simple identifier comparisons.
CAP_QUERY_JOIN_REQUIRES_SCOPE	Set to 'yes' if SQL queries must scope each join clause within parentheses to ensure a proper order of evaluation.
CAP_QUERY_JOIN_REQUIRES_SUBQUERY	Set to ‘yes’ to force join expressions involving more than two tables to be composed with subqueries.
CAP_QUERY_NULL_REQUIRES_CAST	Set to 'yes' if the data source requires that all NULL literals are cast to an explicit data type.

CAP_QUERY_SELECT_ALIASES_SORTED	Set to 'yes' if Tableau must impose a deterministic order on the SELECT expressions (sorted by alias) to ensure that query results can be properly matched with each field in the Tableau visualization. This is only required for data sources which do not preserve the aliases of the SELECT expressions when returning metadata with the query results.
CAP_QUERY_SORT_BY_DEGREE	Set to 'yes' if SQL queries can reference the sorting columns by the ordinal position of each column, e.g. ORDER BY 2, 5. See also: CAP_QUERY_GROUP_BY_DEGREE.
CAP_QUERY_SUBQUERIES	Set to 'yes' if the data source supports subqueries.
CAP_QUERY_SUBQUERIES_WITH_TOP	Set to 'yes' if the data source supports a TOP or LIMIT row-limiting clause within a subquery.
CAP_QUERY_SUBQUERY_DATASOURCE_CONTEXT	Set to 'yes' to use subquery filtered query context to implement data source filters. This capability is available in Tableau 8.0 through Tableau 9.3 only.
CAP_QUERY_SUBQUERY_QUERY_CONTEXT	Set to 'yes' to force Tableau to use a subquery for context filters instead of a temporary table or locally cached results.
CAP_QUERY_TOP_N	Set to 'yes' if the data source supports any form of row-limiting clause. The exact forms supported are described below.
CAP_QUERY_TOPSTYLE_LIMIT	Set to 'yes' if the data source uses LIMIT as the row-limiting clause.
CAP_QUERY_TOPSTYLE_ROWNUM	Set to 'yes' if the data source supports an Oracle-style filter on ROWNUM as the row-limiting clause.

CAP_QUERY_TOPSTYLE_TOP	Set to 'yes' if the data source uses TOP as the row-limiting clause.
CAP_QUERY_USE_QUERY_FUSION	Set to ‘no’ to prevent Tableau from combining multiple individual queries into a single combined query. Turn off this capability for performance tuning or if the database is unable to process large queries. This capability is enabled by default and is available in Tableau 9.0 and later for all data sources except Tableau data extracts. Support for this capability in Tableau data extracts is available in Tableau 9.0.6.
CAP_SELECT_INTO	Set to 'yes' if Tableau can create a table on the fly from the resultset of another query. See also: CAP_CREATE_TEMP_TABLES.
CAP_SELECT_TOP_INTO	Set to 'yes' if Tableau can use a TOP or LIMIT row-limiting clause when creating a table from a query result set.
CAP_SET_ISOLATION_LEVEL_VIA_SQL	Set to 'yes' to force Tableau to set the transaction isolation level for the data source using a SQL query. CAP_SET_ISOLATION_LEVEL_VIA_SQL must be set to 'yes' when any one of the four CAP_ISOLATION_LEVEL capabilities has been set to 'yes'.
CAP_STORED_PROCEDURE_PREFER_TEMP_TABLE	Set to 'yes' to use a temporary table to support remote queries over the stored procedure result set.
CAP_STORED_PROCEDURE_REPAIR_TEMP_TABLE_STRINGS	Set to 'yes' to attempt to compute actual string widths if metadata indicates no width or non-positive width.
CAP_STORED_PROCEDURE_TEMP_TABLE_FROM_BUFFER	Set to 'yes' to populate the temporary table from a result set buffered in entirety.

CAP_STORED_PROCEDURE_TEMP_TABLE_FROM_NEW_PROTOCOL	Set to 'yes' to populate the temporary table from a separate protocol created for just this operation.
CAP_SUPPRESS_DISCOVERY_QUERIES	Set to 'yes' to prevent Tableau from detecting the supported SQL syntax for a variety of clauses.
CAP_SUPPRESS_DISPLAY_LIMITATIONS	Set to 'yes' to suppress displaying any warnings about limitations for this data source.

See also

Other Databases (JDBC) on page 670 – Describes how to connect to your data using the Other Databases (JDBC) connector.

Tableau Capability Customizations Reference

You can set the following capability customizations in the Tableau Datasource Customization (TDC) file to define which Tableau capabilities are supported by the ODBC connection. For more information, see **Customize and Tune a Connection** on page 693.

Many of these customizations influence the type of SQL queries that Tableau issues. For settings that are not defined, Tableau attempts to determine the proper values for each capability by issuing various forms of SQL queries to experimentally verify which forms are supported, as described in How Tableau determines the capabilities of the ODBC driver.

CAP_CREATE_TEMP_TABLES	Set to 'yes' if Tableau can create temporary tables needed for certain complex or optimized queries. See also: CAP_SELECT_INTO.
CAP_CONNECT_STORED_PROCEDURE	Set to 'yes' to allow support for connecting to a stored procedure.
CAP_FAST_METADATA	Set to 'yes' if you have small to moderate size schemas. This capability controls whether Tableau should enumerate all of the objects immediately when you connect. Set the value to "yes" to enable this capability for better performance when creating new connections.

	Disable this capability to allow search for specific schemas or tables instead of retrieving all objects. You can search for all objects by using an empty string. This capability is available in 9.0 and later.
CAP_ISOLATION_LEVEL_READ_COMMITTED	Set to 'yes' to force the transaction isolation level to Read Committed if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.
CAP_ISOLATION_LEVEL_READ_UNCOMMITTED	Set to 'yes' to force the transaction isolation level to Read Uncommitted if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. This capability can improve speed by reducing lock contention, but may result in partial or inconsistent data in query results. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.
CAP_ISOLATION_LEVEL_REPEATABLE_READS	Set to 'yes' to force the transaction isolation level to Repeatable Reads if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.
CAP_ISOLATION_LEVEL_SERIALIZABLE	Set to 'yes' to force the transaction isolation level to Serializable if the data source supports it. Only one of the four transaction isolation levels should be set to 'yes'. This is a very conservative setting that may improve stability at the expense of performance. See also: CAP_SET_ISOLATION_LEVEL_VIA_SQL, CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API.
CAP_SET_ISOLATION_LEVEL_VIA_	Set to 'yes' to force Tableau to set the transaction

ODBC_API	isolation level for the data source using the ODBC API. CAP_SET_ISOLATION_LEVEL_VIA_ODBC_API must be set to 'yes' when any one of the four CAP_ISOLATION_LEVEL capabilities has been set to 'yes'.
CAP_SET_ISOLATION_LEVEL_VIA_SQL	Set to 'yes' to force Tableau to set the transaction isolation level for the data source using a SQL query. CAP_SET_ISOLATION_LEVEL_VIA_SQL must be set to 'yes' when any one of the four CAP_ISOLATION_LEVEL capabilities has been set to 'yes'.
CAP_MULTIPLE_CONNECTIONS_FROM_SAME_IP	Set to 'no' to prevent Tableau from creating more than one active connection to the database. This is a conservative setting that may increase stability at the expense of performance.
CAP_ODBC_BIND_DETECT_ALIAS_CASE_FOLDING	Set to 'yes' to allow Tableau to detect and recover from an ODBC data source that reports the field names in a result set using only upper-case or lower-case characters, instead of the expected field names.
CAP_ODBC_BIND_BOOL_AS_WCHAR_01LITERAL	Set to 'yes' to bind a Boolean data type as a WCHAR containing values '0' or '1'.
CAP_ODBC_BIND_BOOL_AS_WCHAR_TFLITERAL	Set to 'yes' to bind a Boolean data type as WCHAR containing values 't' or 'f'.
CAP_ODBC_BIND_FORCE_DATE_AS_CHAR	Set to 'yes' to force the Tableau native ODBC protocol to bind date values as CHAR.
CAP_ODBC_BIND_FORCE_DATETIME_AS_CHAR	Set to 'yes' to force the Tableau native ODBC protocol to bind datetime values as CHAR.
CAP_ODBC_BIND_FORCE_MAX_STRING_BUFFERS	Set to 'yes' to force the Tableau native ODBC protocol to use maximum-sized buffers (1MB) for strings instead of the size described by metadata.

CAP_ODBC_BIND_FORCE_MEDIUM_STRING_BUFFERS	Set to 'yes' to force the Tableau native ODBC protocol to use medium-sized buffers (1K) for strings instead of the size described by metadata.
CAP_ODBC_BIND_FORCE_SMALL_STRING_BUFFERS	Set to 'yes' to force the Tableau native ODBC protocol to use small buffers for strings instead of the size described by metadata.
CAP_ODBC_BIND_FORCE_SIGNED	Set to 'yes' to force binding integers as signed.
CAP_ODBC_BIND_PRESERVE_BOM	Set to 'yes' to preserve BOM when present in strings. Hive will return BOM and treat strings containing it as distinct entities.
CAP_ODBC_BIND_SKIP_LOCAL_DATATYPE_UNKNOWN	Set to ‘yes’ to prevent the native ODBC Protocol from binding to columns having local data type DataType::Unknown in the expected metadata.
CAP_ODBC_BIND_SPATIAL_AS_WKT	Set to ‘yes’ to force binding Spatial data as WKT (Well Known Text)
CAP_ODBC_BIND_SUPPRESS_COERCE_TO_STRING	Set to 'yes' to prevent the Tableau native ODBC protocol from binding non-string data as strings (i.e. requesting driver conversion).
CAP_ODBC_BIND_SUPPRESS_INT64	Set to 'yes' to prevent the Tableau native ODBC protocol from using 64-bit integers for large numeric data.
CAP_ODBC_BIND_SUPPRESS_PREFERRED_CHAR	Set to 'yes' to prevent the Tableau native ODBC protocol from preferring a character type that differs from the driver default.
CAP_ODBC_BIND_SUPPRESS_PREFERRED_TYPES	Set to 'yes' to prevent the Tableau native ODBC protocol from binding any data according to its preferred wire types. With this capability set, Tableau will only bind according to the data types described by the ODBC driver via metadata.

CAP_ODBC_BIND_SUPPRESS_WIDE_CHAR	Set to 'yes' to prevent the Tableau native ODBC protocol from binding strings a WCHAR. Instead they will be bound as single-byte CHAR arrays, and processed locally for any UTF-8 characters contained within.
CAP_ODBC_CONNECTION_STATE_VERIFY_FAST	Set to ‘yes’ to check if a connection is broken with a fast ODBC API call.
CAP_ODBC_CONNECTION_STATE_VERIFY_PROBE	Set to ‘yes’ to check if a connection is broken with a forced probe.
CAP_ODBC_CONNECTION_STATE_VERIFY_PROBE_IF_STALE	Set to ‘yes’ to check if a connection is broken with a forced probe only if it is "stale" (i.e., unused for about 30 minutes).
CAP_ODBC_CONNECTION_STATE_VERIFY_PROBE_PREPARED_QUERY	Set to ‘yes’ to check if a connection is broken using a prepared query.
CAP_ODBC_CURSOR_DYNAMIC	Set to 'yes' to force the Tableau native ODBC protocol to set the cursor type for all statements to Dynamic (scrollable, detects added/removed/modified rows).
CAP_ODBC_CURSOR_FORWARD_ONLY	Set to 'yes' to force the Tableau native ODBC protocol to set the cursor type for all statements to Forward-only (non-scrollable).
CAP_ODBC_CURSOR_KEYSET_DRIVEN	Set to 'yes' to force the Tableau native ODBC protocol to set the cursor type for all statements to Keyset-driven (scrollable, detects changes to values within a row).
CAP_ODBC_CURSOR_STATIC	Set to 'yes' to force Tableau to set the cursor type for all statements to Static (scrollable, does not detect changes).
CAP_ODBC_ERROR_IGNORE_FALSE_ALARM	Set to 'yes' to allow the Tableau native ODBC protocol to ignore SQL_ERROR conditions where

	SQLSTATE is '00000' (meaning "no error").
CAP_ODBC_ERROR_IGNORE_SQLNODATA_FOR_COMMAND_QUERIES	Set to ‘yes’ to ignore when SQLExecDirect returns SQL_NO_DATA even when data is not expected back.
CAP_ODBC_EXPORT_ALLOW_CHAR_UTF8	Set to 'yes' to allow the use of single-byte char data type for binding Unicode strings as UTF-8.
CAP_ODBC_EXPORT_BIND_FORCE_TARGET_METADATA	Set to 'yes' to force binding for export based on all of the metadata from the target table instead of the ODBC metadata for the parameterized insert statement.
CAP_ODBC_EXPORT_BIND_PREFER_TARGET_METADATA	Set to 'yes' to prefer binding for export based on specific types of metadata from the target table instead of the ODBC metadata for the parameterized insert statement.
CAP_ODBC_EXPORT_BUFFERS_RESIZABLE	Set to 'yes' to allow export buffers to be reallocated after the first batch to improve performance.
CAP_ODBC_EXPORT_BUFFERS_SIZE_FIXED	Set to 'yes' to ignore the width of a single row when computing the total rows to insert at a time.
CAP_ODBC_EXPORT_BUFFERS_SIZE_LIMIT_512KB	Set to 'yes' to limit export buffers to 512 KB. This is an uncommon setting.
CAP_ODBC_EXPORT_BUFFERS_SIZE_MASSIVE	Set to 'yes' to force the use of large buffers for insert. If CAP_ODBC_EXPORT_BUFFERS_RESIZABLE is not set or disabled, a fixed row count is used.
CAP_ODBC_EXPORT_BUFFERS_SIZE_MEDIUM	Set to 'yes' to force the use of medium-sized buffers for insert. If CAP_ODBC_EXPORT_BUFFERS_RESIZABLE is not set or disabled, a fixed row count is used.

CAP_ODBC_EXPORT_BUFFERS_SIZE_SMALL	Set to 'yes' to force the use of small buffers for insert. If CAP_ODBC_EXPORT_BUFFERS_RESIZABLE is not set or disabled, a fixed row count is used.
CAP_ODBC_EXPORT_CONTINUE_ON_ERROR	Set to 'yes' to continue data insert despite errors. Some data sources report warnings as errors.
CAP_ODBC_EXPORT_DATA_BULK	Set to 'yes' to allow the use of ODBC bulk operations for data insert.
CAP_ODBC_EXPORT_DATA_BULK_VIA_INSERT	Set to 'yes' to allow the use of ODBC bulk operations based on 'INSERT INTO' parameterized queries.
CAP_ODBC_EXPORT_DATA_BULK_VIA_ROWSET	Set to 'yes' to allow the use of ODBC bulk operations based on a rowset cursor.
CAP_ODBC_EXPORT_FORCE_INDICATE_NTS	Set to 'yes' to force the use of indicator buffers for identifying null-terminated strings (NTS).
CAP_ODBC_EXPORT_FORCE_SINGLE_ROW_BINDING	Set to 'yes' to force the use of a single row for binding export buffers to insert data.
CAP_ODBC_EXPORT_FORCE_SINGLE_ROW_BINDING_WITH_TIMESTAMPS	Set to 'yes' to force the use of a single row for binding export buffers when dealing with timestamp data. This is required for some versions of Teradata.
CAP_ODBC_EXPORT_FORCE_STRING_WIDTH_FROM_SOURCE	Set to 'yes' to force the use of the source string width (from Tableau metadata), overriding the destination string width (from insert parameter metadata).
CAP_ODBC_EXPORT_FORCE_STRING_WIDTH_USING_OCTET_LENGTH	Set to 'yes' to force the use of the source string width from the octet length.
CAP_ODBC_EXPORT_SUPPRESS_STRING_WIDTH_VALIDATION	Set to 'yes' to suppress validating that the target string width can accommodate the widest source

	strings.
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_BATCH_ MASSIVE	Set to ‘yes’ to commit in massive batches of INSERT statements (~100,000). This may be useful with single-row export binding.
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_BATCH_ MEDIUM	Set o 'yes' to commit in medium-sized batches of INSERT statements (~50). A single statement may be bound to multiple records.
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_BATCH_ SMALL	Set to 'yes' to commit in small batches of INSERT statements (~5). A single statement may be bound to multiple records.
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_BYTES_ MASSIVE	Set to 'yes' to commit in massive batches of data (~100 MB).
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_BYTES_ MEDIUM	Set to 'yes' to commit in medium batches of data (~10 MB).
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_BYTES_ SMALL	Set to 'yes' to commit in small batches of data (~1 MB).
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_EACH_ STATEMENT	Set to 'yes' to commit after executing each INSERT statement. A single statement may be bound to multiple records.
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_ INTERVAL_LONG	Set to 'yes' to commit in long intervals of elapsed time (~100 seconds).
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_ INTERVAL_MEDIUM	Set to 'yes' to commit in medium intervals of elapsed time (~10 seconds).
CAP_ODBC_EXPORT_ TRANSACTIONS_COMMIT_	Set to 'yes' to commit in short intervals of elapsed time (~1 seconds).

INTERVAL_SHORT	
CAP_ODBC_EXPORT_TRANSACTIONS_COMMIT_ONCE_WHEN_COMPLETE	Set to 'yes' to commit only once at the end after the export is complete.
CAP_ODBC_EXPORT_TRANSLATE_DATA_PARALLEL	Set to 'yes' to use parallel loops to translate Tableau DataValues to wire buffers on export.
CAP_ODBC_FETCH_ABORT_FORCE_CANCEL_STATEMENT	Set to 'yes' to cancel the statement handle upon interrupting SQLFetch with a cancel exception.
CAP_ODBC_FETCH_BUFFERS_RESIZABLE	Set to 'yes' to allow buffers to be reallocated after fetch to improve performance or handle data truncation.
CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED	Set to 'yes' to ignore the width of a single row when computing the total rows to fetch.
CAP_ODBC_FETCH_BUFFERS_SIZE_MASSIVE	Set to 'yes' to force the use of large buffers. If CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED is enabled, a fixed row count is used.
CAP_ODBC_FETCH_BUFFERS_SIZE_MEDIUM	Set to 'yes' to force the use of medium-sized buffers. If CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED is enabled, a fixed row count is used.
CAP_ODBC_FETCH_BUFFERS_SIZE_SMALL	Set to 'yes' to force the use of small buffers. If CAP_ODBC_FETCH_BUFFERS_SIZE_FIXED is enabled, a fixed row count is used.
CAP_ODBC_FETCH_CONTINUE_ON_ERROR	Set to 'yes' to allow the Tableau native ODBC protocol to continue resultset fetch despite errors (some data sources report warnings as errors).
CAP_ODBC_FETCH_IGNORE_FRACTIONAL_SECONDS	Set to 'yes' to allow the Tableau native ODBC protocol to ignore the fractional seconds component of a time value when fetching query result set data. This is useful when working with

	data sources that do not follow the ODBC specification for fractional seconds, which must be represented as billionths of a second.
CAP_ODBC_FETCH_RESIZE_BUFFERS	Set to 'yes' to allow the Tableau native ODBC protocol to automatically resize buffers and fetch again if data truncation occurred.
CAP_ODBC_FORCE_SINGLE_ROW_BINDING	Set to 'yes' to force the Tableau native ODBC protocol to use a single row for result set transfers instead of the more efficient bulk-fetch.
CAP_ODBC_IMPORT_ERASE_BUFFERS	Set to 'yes' to reset the contents of data buffers before fetching each block.
CAP_ODBC_IMPORT_TRANSLATE_DATA_PARALLEL	Set to ‘no’ to disable decoding data locally in parallel.
CAP_ODBC_METADATA_FORCE_LENGTH_AS_PRECISION	Set to 'yes' to force the Tableau native ODBC protocol to use the column "length" as the numeric precision. This is an uncommon setting.
CAP_ODBC_METADATA_FORCE_NUM_PREC_RADIX_10	Set to 'yes' to force the Tableau native ODBC protocol to assume the numeric precision is reported in base-10 digits. This is an uncommon setting.
CAP_ODBC_METADATA_FORCE_UNKNOWN_AS_STRING	Set to 'yes' to force the Native ODBC Protocol to treat unknown data types as string instead of ignoring the associated column.
CAP_ODBC_METADATA_FORCE_UTF8_IDENTIFIERS	Set to 'yes' to force the protocol to treat identifiers as UTF-8 when communicating with the driver.
CAP_ODBC_METADATA_SKIP_DESC_TYPE_NAME	Set to 'yes' to remove the check for the SQL_DESC_TYPE_NAME attribute with the SQLColAttribute API.
CAP_ODBC_METADATA_STRING_LENGTH_UNKNOWN	Set to 'yes' to prevent Tableau from allocating

	memory based on the driver-reported string length, which may not be known or reported properly. Instead, Tableau will use a fixed-sized string length, and will reallocate as needed to handle string data that is too large for the fixed-size buffer.
CAP_ODBC_METADATA_STRING_TRUST_OCTET_LENGTH	Set to 'yes' to use the octet length reported by the driver for strings instead of computing it from the number of characters.
CAP_ODBC_METADATA_SUPPRESS_EXECUTED_QUERY	Set to 'yes' to prevent Tableau from executing a query as a means of reading metadata. While Tableau typically includes a row-limiting clause in such metadata queries (e.g., 'LIMIT', or 'WHERE 1=0'), this may not help when used with a Custom SQL connection for database systems with poor query optimizers. Note that this capability may prevent Tableau from determining the connection metadata properly.
CAP_ODBC_METADATA_SUPPRESS_PREPARED_QUERY	Set to 'yes' to prevent Tableau from using a prepared query as a means of reading metadata. A prepared query is often the fastest way to accurately read metadata. However, not all database systems are capable of reporting metadata for a prepared query without actually executing the query. Note that certain metadata -- for example from connections using Custom SQL-- cannot be retrieved if this capability and CAP_ODBC_METADATA_SUPPRESS_EXECUTED_QUERY are both set.
CAP_ODBC_METADATA_SUPPRESS_READ_IDENTITY_COLUMNS	Set to ‘no’ to prevent reading identity column metadata.
CAP_ODBC_METADATA_	Set to 'yes' to prevent reading metadata using a

SUPPRESS_SELECT_STAR	'select *' query.
CAP_ODBC_METADATA_ SUPPRESS_SQLCOLUMNS_API	Set to 'yes' to prevent Tableau from using older, less accurate API for reading metadata from ODBC data sources. Setting this capability allows Tableau to read metadata by issuing a full 'select *' query, which is expensive but may enable connectivity for extremely limited or unstable data sources.
CAP_ODBC_METADATA_ SUPPRESS_SQLFOREIGNKEYS_API	Set to 'yes' to prevent Tableau from attempting to read metadata describing foreign key constraints. Despite the simple nature of this ODBC API, some drivers may have unstable behavior or produce inaccurate results. Setting this capability may force Tableau to generate less efficient queries involving multi-table joins.
CAP_ODBC_METADATA_ SUPPRESS_SQLPRIMARYKEYS_API	Set to 'yes' to prevent Tableau from reading primary key metadata using the SQLPrimaryKeys API or an equivalent query. This capability is available in Tableau 9.1 and later.
CAP_ODBC_METADATA_ SUPPRESS_SQLSTATISTICS_API	Set to 'yes' to prevent reading unique constraints and table cardinality estimates using the SQLStatistics API or an equivalent query. This capability is available in Tableau 9.0 and later.
CAP_ODBC_QUERY_USE_ PREPARE_PARAMETER_MARKER	Enable to use prepared statements with parameter markers instead of literal values. Applies only for floating point, integer, and string values.
CAP_ODBC_REBIND_SKIP_UNBIND	Set to 'yes' to force the Tableau native ODBC protocol to rebind a column directly and skip unbinding, which reduces ODBC API calls when resizing buffers to refetch truncated data.

CAP_ODBC_SUPPORTS_LONG_DATA_BULK	Set to ‘yes’ if driver can fetch multiple long-data rows at a time.
CAP_ODBC_SUPPORTS_LONG_DATA_ORDERED	Set to ‘yes’ if driver requires long-data columns to come after non-long-data ones.
CAP_ODBC_SUPPRESS_INFO_SCHEMA_STORED_PROCS	Set to 'yes' to prevent the INFORMATION.SCHEMA schema from being queried when enumerating stored procedures.
CAP_ODBC_SUPPRESS_INFO_SCHEMA_TABLES	Set to ‘yes’ to prevent tables from “information_schema” schema from being returned by EnumerateTables.
CAP_ODBC_SUPPRESS_PG_TEMP_SCHEMA_TABLES	Set to ‘yes’ to prevent tables from “pg_temp” schema from being returned by EnumerateTables.
CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_ALL_COMMAND_QUERIES	Set to 'yes' to execute all commands directly (i.e., no prepared statement).
CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_DDL_COMMAND_QUERIES	Set to 'yes' to execute DDL commands (e.g. CREATE TABLE) directly (i.e., no prepared statement).
CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_DML_COMMAND_QUERIES	Set to 'yes' to execute DML commands (e.g. INSERT INTO) directly (i.e, no prepared statement).
CAP_ODBC_SUPPRESS_PREPARED_QUERY_FOR_NON_COMMAND_QUERIES	Set to ‘yes’ to execute all non-command queries directly (no prepared statement).
CAP_ODBC_SUPPRESS_SYS_SCHEMA_STORED_PROCS	Set to 'yes' to explicitly add the "SYS" schema to the schema exclusions when enumerating stored procedures.
CAP_ODBC_TRANSACTIONS_COMMIT_INVALIDATES_	Set to ‘yes’ to indicate that a transaction will invalidate all prepared statements and close any

PREPARED_QUERY	open cursors.
CAP_ODBC_TRANSACTIONS_SUPPRESS_AUTO_COMMIT	Set to 'yes' to prevent the Native ODBC Protocol from using default auto-committing transaction behavior in ODBC. This capability cannot be used with CAP_ODBC_TRANSACTIONS_SUPPRESS_EXPLICIT_COMMIT.
CAP_ODBC_TRANSACTIONS_SUPPRESS_EXPLICIT_COMMIT	Set to 'yes' to prevent the Native ODBC Protocol from explicitly managing transactions. This capability cannot be used with CAP_ODBC_TRANSACTIONS_SUPPRESS_AUTO_COMMIT.
CAP_ODBC_TRIM_CHAR_LEAVE_PADDING	Set to 'yes' to leave whitespace padding at the end of a character or text data type. Most data sources will trim this whitespace automatically, but the behavior depends on the driver.
CAP_ODBC_TRIM_VARCHAR_PADDING	Set to 'yes' to force the Tableau native ODBC protocol to trim trailing whitespace from VARCHAR columns which the driver has erroneously padded.
CAP_ODBC_UNBIND_AUTO	Set to 'yes' to force the Tableau native ODBC protocol to unbind and deallocate columns automatically, which can reduce ODBC API calls.
CAP_ODBC_UNBIND_BATCH	Set to 'yes' to force the Tableau native ODBC protocol to unbind and deallocate columns in a single batch operation, which can reduce ODBC API calls.
CAP_ODBC_UNBIND_EACH	Set to 'yes' to force the Tableau native ODBC protocol to unbind and deallocate columns individually, which may improve stability.
CAP_ODBC_UNBIND_PARAMETERS_BATCH	Set to ‘yes’ to unbind all parameters in a single batch operation.

CAP_ORACLE_SHOW_ALL_ SYNONYM_OWNERS	Set to 'yes' to list all the owners in the all_ synonyms view for Oracle. This capability is available in 9.0 and later.
CAP_QUERY_BOOLEXPR_TO_ INTEXPR	Set to 'yes' if Tableau must coerce any Boolean expressions to an integer value in order include in a result set.
CAP_QUERY_FROM_REQUIRES_ ALIAS	Set to 'yes' if the FROM clause must provide an alias for the given table.
CAP_QUERY_GROUP_ALLOW_ DUPLICATES	Set to 'no' if SQL queries cannot contain duplicate expressions in the GROUP BY clause (this is uncommon).
CAP_QUERY_GROUP_BY_ALIAS	Set to 'yes' if SQL queries with aggregations can reference the grouping columns by their corresponding alias in the SELECT list, e.g. GROUP BY "none_ShipCountry_nk".
CAP_QUERY_GROUP_BY_DEGREE	Set to 'yes' if SQL queries with aggregations can reference the grouping columns by the ordinal position of each column, e.g. GROUP BY 2, 5. See also: CAP_QUERY_SORT_BY_DEGREE
CAP_QUERY_HAVING_REQUIRES_ GROUP_BY	Set to 'yes' if Tableau must use an artificial grouping field for any query which has a HAVING clause but no grouping columns.
CAP_QUERY_HAVING_ UNSUPPORTED	Set to 'yes' if the SQL syntax for HAVING is unsupported. Tableau may be able to work around this using subqueries. See also: CAP_ QUERY_SUBQUERIES.
CAP_QUERY_INCLUDE_GROUP_ BY_COLUMNS_IN_SELECT	Set to 'yes' to require all GROUP BY expressions to also appear in the SELECT expression list.
CAP_QUERY_JOIN_ACROSS_ SCHEMAS	Set to 'yes' if SQL queries can express joins between tables located in different schemas.

CAP_QUERY_JOIN_ASSUME_CONSTRAINED	Set to ‘yes’ to cull inner joins even if the database tables does do not have FK-PK relationships.
CAP_QUERY_JOIN_PUSH_DOWN_CONDITION_EXPRESSIONS	Set to 'yes' to rewrite joins to simplify the ON clause conditions to simple identifier comparisons.
CAP_QUERY_JOIN_REQUIRES_SCOPE	Set to 'yes' if SQL queries must scope each join clause within parentheses to ensure a proper order of evaluation.
CAP_QUERY_JOIN_REQUIRES_SUBQUERY	Set to ‘yes’ to force join expressions involving more than two tables to be composed with subqueries.
CAP_QUERY_NULL_REQUIRES_CAST	Set to 'yes' if the data source requires that all NULL literals are cast to an explicit data type.
CAP_QUERY_SELECT_ALIASES_SORTED	Set to 'yes' if Tableau must impose a deterministic order on the SELECT expressions (sorted by alias) to ensure that query results can be properly matched with each field in the Tableau visualization. This is only required for data sources which do not preserve the aliases of the SELECT expressions when returning metadata with the query results.
CAP_QUERY_SORT_BY_DEGREE	Set to 'yes' if SQL queries can reference the sorting columns by the ordinal position of each column, e.g. ORDER BY 2, 5. See also: CAP_QUERY_GROUP_BY_DEGREE.
CAP_QUERY_SUBQUERIES	Set to 'yes' if the data source supports subqueries.
CAP_QUERY_SUBQUERIES_WITH_TOP	Set to 'yes' if the data source supports a TOP or LIMIT row-limiting clause within a subquery.
CAP_QUERY_SUBQUERY_	Set to 'yes' to use subquery filtered query context

DATASOURCE_CONTEXT	to implement data source filters. This capability is available in Tableau 8.0 through Tableau 9.3 only.
CAP_QUERY_SUBQUERY_QUERY_CONTEXT	Set to 'yes' to force Tableau to use a subquery for context filters instead of a temporary table or locally cached results.
CAP_QUERY_TOP_0_METADATA	Set to ‘yes’ if the data source can handle a “TOP 0” request for retrieving metadata.
CAP_QUERY_TOP_N	Set to 'yes' if the data source supports any form of row-limiting clause. The exact forms supported are described below.
CAP_QUERY_TOPSTYLE_LIMIT	Set to 'yes' if the data source uses LIMIT as the row-limiting clause.
CAP_QUERY_TOPSTYLE_ROWNUM	Set to 'yes' if the data source supports an Oracle-style filter on ROWNUM as the row-limiting clause.
CAP_QUERY_TOPSTYLE_TOP	Set to 'yes' if the data source uses TOP as the row-limiting clause.
CAP_QUERY_USE_QUERY_FUSION	Set to ‘no’ to prevent Tableau from combining multiple individual queries into a single combined query. Turn off this capability for performance tuning or if the database is unable to process large queries. This capability is enabled by default and is available in Tableau 9.0 and later for all data sources except Tableau data extracts. Support for this capability in Tableau data extracts is available in Tableau 9.0.6.
CAP_QUERY_WHERE_FALSE_METADATA	Set to ‘yes’ if the data source can handle a “WHERE <false>” predicate for retrieving metadata.
CAP_SELECT_INTO	Set to 'yes' if Tableau can create a table on the fly from the resultset of another query. See also:

	CAP_CREATE_TEMP_TABLES.
CAP_SELECT_TOP_INTO	Set to 'yes' if Tableau can use a TOP or LIMIT row-limiting clause when creating a table from a query resultset.
CAP_STORED_PROCEDURE_PREFER_TEMP_TABLE	Set to 'yes' to use a temporary table to support remote queries over the stored procedure result set.
CAP_STORED_PROCEDURE_REPAIR_TEMP_TABLE_STRINGS	Set to 'yes' to attempt to compute actual string widths if metadata indicates no width or non-positive width.
CAP_STORED_PROCEDURE_TEMP_TABLE_FROM_BUFFER	Set to 'yes' to populate the temporary table from a result set buffered in entirety.
CAP_STORED_PROCEDURE_TEMP_TABLE_FROM_NEW_PROTOCOL	Set to ‘yes’ to populate the temporary table from a separate protocol created for just this operation.
CAP_SUPPRESS_DISCOVERY_QUERIES	Set to 'yes' to prevent Tableau from detecting the supported SQL syntax for a variety of clauses.
CAP_SUPPRESS_DISPLAY_LIMITATIONS	Set to 'yes' to suppress displaying any warnings about limitations for this data source.

See also

Other Databases (ODBC) on page 681 – Describes how to connect to your data using the ODBC connector.

Tableau and ODBC on page 684 – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.

Customize and Tune a Connection on page 693 – Describes how to customize connection information for improved functionality and performance.

ODBC/SQL Customizations Reference on the facing page – Lists customizations that represent the parts of the ODBC and SQL standards that the ODBC driver reports supporting.

ODBC/SQL Customizations Reference

You can set the following customizations in the Tableau Datasource Customization (TDC) file to define parts of the ODBC and SQL standards that the ODBC driver supports.

Documentation resources for SQLGetInfo

The names of these customizations come from the identifiers used as parameters to SQLGetInfo.

For more information, see the MSDN documentation and the source code header file `sqlext.h` for the numeric and bit-mask values associated with each customization.

- The Microsoft documentation for the ODBC API function SQLGetInfo:

  http://msdn.microsoft.com/en-us/library/ms711681%28VS.85%29.aspx

- A sample header file, `sqlext.h`, for the ODBC API identifier values used with SQLGetInfo:

  http://cpansearch.perl.org/src/MIMER/DBD-Mimer-1.00/sqlext.h

SQLGetInfo long-integer values

SQL_ODBC_INTERFACE_CONFORMANCE	Currently unused.
SQL_SQL_CONFORMANCE	Integer bitmask. Defines the level which the data source conforms to the SQL standard: '1' for entry-level SQL-92 conformance, '2' for FIPS 127-2 transitional, '4' for Intermediate and '8' for Full conformance.
SQL_CATALOG_USAGE	Integer bitmask. Defines the SQL statements in which a catalog identifier can be used.
SQL_SCHEMA_USAGE	Integer bitmask. Defines the SQL statements in which a schema identifier can be used.
SQL_AGGREGATE_FUNCTIONS	Integer bitmask. Defines which standard SQL aggregation forms are supported.
SQL_NUMERIC_FUNCTIONS	Integer bitmask. Defines which SQL scalar numeric functions are supported.

SQL_STRING_FUNCTIONS	Integer bitmask. Defines which SQL scalar string functions are supported.
SQL_TIMEDATE_FUNCTIONS	Integer bitmask. Defines which SQL scalar date / time functions are supported.
SQL_TIMEDATE_ADD_INTERVALS	Integer bitmask. Defines which date / time intervals are supported with the TIMESTAMPADD scalar function.
SQL_TIMEDATE_DIFF_INTERVALS	Integer bitmask. Defines which date / time intervals are supported with the TIMESTAMPDIFF scalar function.
SQL_DATETIME_LITERALS	Integer bitmask. Defines which SQL-92 literals are supported for representing DATE / TIME constants and INTERVALs.
SQL_SYSTEM_FUNCTIONS	Integer bitmask. Defines support for special SQL system scalar functions: IFNULL, DBNAME and USERNAME.
SQL_SQL92_VALUE_EXPRESSIONS	Integer bitmask. Defines which logical functions are supported for testing and manipulating values: CASE, CAST and NULLIF.
SQL_SQL92_NUMERIC_VALUE_FUNCTIONS	Integer bitmask. Defines which functions can produce a numeric value from non-numeric data, including: EXTRACT (for date / time part extraction), CHAR_LENGTH, CHARACTER_LENGTH and POSITION(.. IN ..).
SQL_SQL92_STRING_FUNCTIONS	Integer bitmask. Defines which string manipulation functions are supported.
SQL_SQL92_DATETIME_FUNCTIONS	Integer bitmask. Defines which date / time manipulation functions are supported for determining the current date, time or timestamp.
SQL_OJ_CAPABILITIES	Integer bitmask. Defines which type of outer joins are supported.
SQL_SQL92_RELATIONAL_JOIN_OPERATORS	Integer bitmask. Defines which types of JOIN operators are supported, e.g. INNER, OUTER.

SQL_SQL92_PREDICATES	Integer bitmask. Defines which predicates are supported for logical tests of values, e.g. IS NULL, LIKE, IN.
SQL_CONVERT_FUNCTIONS	Integer bitmask. Defines which ODBC scalar functions are supported for CASTing or CONVERTing one data type to another.
SQL_CONVERT_TINYINT	Integer bitmask. Determines which other data types that this named type can be converted to using the ODBC scalar function CONVERT.
SQL_CONVERT_SMALLINT	Integer bitmask. Same as above.
SQL_CONVERT_INTEGER	Integer bitmask. Same as above.
SQL_CONVERT_BIGINT	Integer bitmask. Same as above.
SQL_CONVERT_REAL	Integer bitmask. Same as above.
SQL_CONVERT_FLOAT	Integer bitmask. Same as above.
SQL_CONVERT_DOUBLE	Integer bitmask. Same as above.
SQL_CONVERT_CHAR	Integer bitmask. Same as above.
SQL_CONVERT_VARCHAR	Integer bitmask. Same as above.
SQL_CONVERT_LONGVARCHAR	Integer bitmask. Same as above.
SQL_CONVERT_DECIMAL	Integer bitmask. Same as above.
SQL_CONVERT_NUMERIC	Integer bitmask. Same as above.
SQL_CONVERT_BIT	Integer bitmask. Same as above.
SQL_CONVERT_GUID	Integer bitmask. Same as above.
SQL_CONVERT_BINARY	Integer bitmask. Same as above.
SQL_CONVERT_VARBINARY	Integer bitmask. Same as above.

SQL_CONVERT_LONGVARBINARY	Integer bitmask. Same as above.
SQL_CONVERT_DATE	Integer bitmask. Same as above.
SQL_CONVERT_TIME	Integer bitmask. Same as above.
SQL_CONVERT_TIMESTAMP	Integer bitmask. Same as above.
SQL_CONVERT_INTERVAL_DAY_TIME	Integer bitmask. Same as above.

SQLGetInfo short-integer values

SQL_CURSOR_COMMIT_BEHAVIOR	Short integer value. “0” if the prepared statements are closed upon commit for the data source.
SQL_MAX_IDENTIFIER_LEN	Integer value. Defines the maximum number of characters that can be used in an identifier. Tableau leaves room for one extra character as the string terminator.
SQL_TXN_CAPABLE	Short integer value. “0” if the data source does not support the transaction.
SQL_QUOTED_IDENTIFIER_CASE	Integer bitmask.

SQLGetInfo string values

SQL_COLUMN_ALIAS	Boolean value. "Y" if the data source supports using aliases for columns listed in the SELECT clause.
SQL_IDENTIFIER_QUOTE_CHAR	String value. Indicates the character which can be used for quoting identifiers. Because the connection customization is an XML document, any entities must properly be encoded. For example, double-quotes will be '"'. Additionally this character is assumed to work as the opening and closing character around

	identifiers, so some data sources which require '[]' to enclose identifiers will not be supported.
SQL_CATALOG_NAME_SEPARATOR	Character value. Indicates the separator character to use between identifiers when qualifying them with a catalog, schema or table name. This is typically the period character.
SQL_SPECIAL_CHARACTERS	String value. Indicates the special characters which are allowed in identifier strings.
SQL_CATALOG_TERM	String value. This is the descriptive term for a database catalog, which appears in the Tableau connection dialog for this ODBC data source.
SQL_SCHEMA_TERM	String value. This is the descriptive term for a database schema, which appears in the Tableau connection dialog for this ODBC data source.
SQL_TABLE_TERM	String value. This is the descriptive term for a database table, which appears in the Tableau connection dialog for this ODBC data source.
SQL_DRIVER_NAME	String value. This is the name of the database ODBC driver.
SQL_DRIVER_VER	String value. This is the version number of the ODBC driver.
SQL_DRIVER_ODBC_VER	String value. This is the version of the ODBC API which the driver supports.
SQL_ODBC_VER	String value. This is the version of ODBC which the Windows ODBC Driver Manager supports. This should not need to be customized.
SQL_DBMS_NAME	String value. This is the name of the database vendor.
SQL_DBMS_VER	String value. This is the version of the database system.

SQL_SERVER_NAME	String value. This is the named network address of the database server.
SQL_USER_NAME	String value. This is the name of the currently authenticated user.

See also

Other Databases (ODBC) on page 681 – Describes how to connect to your data using the ODBC connector.

Tableau and ODBC on page 684 – Provides background information about ODBC, describes how Tableau determines the functionality of an ODBC driver, and lists frequently asked questions.

Customize and Tune a Connection on page 693 – Describes customize connection information for improved functionality and performance.

Tableau Capability Customizations Reference on page 713 – Lists customizations you can use to define which Tableau capabilities are supported by the data source.

Run Initial SQL

Note: Tableau Prep Builder version 2019.2.2 and later supports using Initial SQL, but doesn't yet support all of the same options supported by Tableau Desktop . For information about using Initial SQL with Tableau Prep Builder, see Use Initial SQL to query your connections in the Tableau Prep Builder online help.

When connecting to some databases, you can specify an initial SQL command that will run when a connection is made to the database, for example, when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. Initial SQL is not run when your refresh your view. Note that this initial SQL is different than a custom SQL connection. A custom SQL connection defines a relation (or table) to issue queries against. For more information, see Connect to a Custom SQL Query.

You can use this command to:

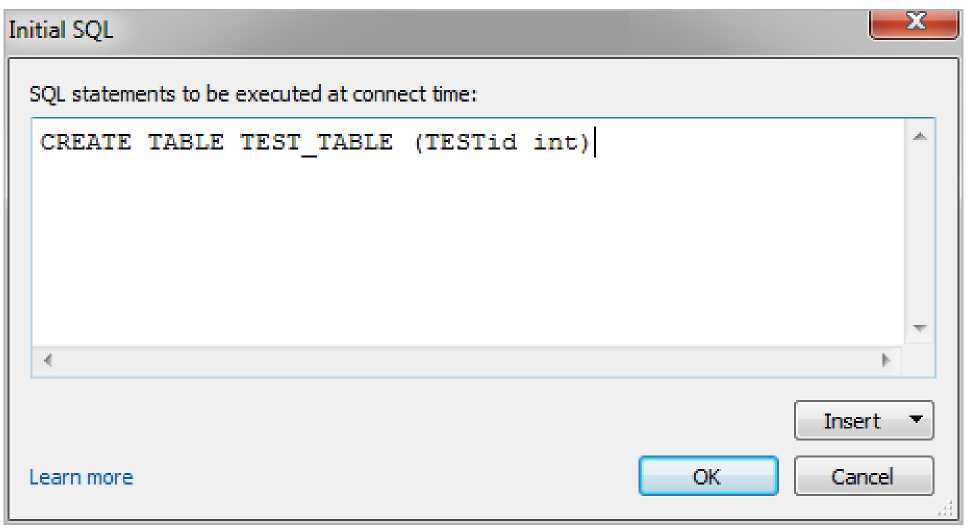
- Set up temporary tables to use during the session.
- Set up a custom data environment.

You have the option to add an initial SQL command in the Server Connection dialog box or on the Data Source page.

Note: If your data source supports running an initial SQL statement, an **Initial SQL** link appears in the lower-left corner of the Server Connection dialog box. For information about your data source, see Supported Connectors.

To use initial SQL

1. In the Server Connection dialog box, click **Initial SQL**. Or, on the Data Source page, select **Data** > **Initial SQL** or **Data** > **Query Banding and Initial SQL** depending on the database you connect to.
2. Enter the SQL command into the Initial SQL dialog box. You can use the **Insert** drop-down menu to pass parameters to your data source.



Note: Tableau does not examine the statement for errors. This SQL statement is simply sent to the database when you connect.

Your software license may restrict you from using initial SQL with your connection. If you publish to Tableau Server, the server must be configured to allow Initial SQL statements. By default, the server software is configured to allow these statements to run when the workbook is loaded in a web browser.

Administrators can configure server to ignore initial SQL statements by using the `tsm configuration set` command:

```
tsm configuration set -k vizqlserver.initialsql.disabled -v true
```

If the server doesn't allow initial SQL statements, the workbook opens, but the initial SQL commands are not sent.

For more information about the `tsm configuration set` command, see the Tableau Server Help.

Parameters in an initial SQL statement

You can pass parameters to your data source in an initial SQL statement. There are several reasons why this is useful:

- You can configure impersonation using the **TableauServerUser** or **TableauServerUserFull** parameters.
- If your data source supports it, you can set up row-level security (for example, for Oracle VPD or SAP Sybase ASE) to make sure that users see only the data that they are authorized to see.
- You can provide more details in logging, for example, the Tableau version or the workbook name.

The following parameters are supported in an initial SQL statement:

Parameter	Description	Example of returned value
TableauServerUser	The user name of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.	jsmith
TableauServerUserFull	The user name and domain of the current server user. Use when setting up imper-sonation on the server. Returns an empty string if the user is not signed in to Tableau Server.	domain.lan\jsmith
TableauApp	The name of the Tableau application.	Tableau Desktop

		Professional Tableau Server
TableauVersion	The version of the Tableau application.	9.3
WorkbookName	The name of the Tableau workbook. Use only in workbooks with an embedded data source.	Financial-Analysis

Examples

The following examples show different ways you can use parameters in an initial SQL statement.

- This example sets the security context on Microsoft SQL Server:

  ```
  EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
  ```

- This example shows how, on a DataStax data source, you can use parameters to add detail to logging or to set up a session variable to keep track of the data:

  ```
  SET TABLEAUVERSION [TableauVersion];
  ```

- This example can be used to help set up row-level security for Oracle VPD:

  ```
  begin
   DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);
  end;
  ```

  Note: Oracle PL/SQL blocks require a trailing semicolon to terminate the block. Consult Oracle documentation for the proper syntax.

Defer execution to the server

You can defer an initial SQL statement so that it is executed only on the server. One reason to defer execution to the server is if you don’t have permission to execute the commands that set up impersonation. Use <ServerOnly></ServerOnly> tags to enclose the commands to be executed only on the server.

Example:

```
CREATE TEMP TABLE TempTable(x varchar(25));
INSERT INTO TempTable VALUES (1);
<ServerOnly>INSERT INTO TempTable Values(2);</ServerOnly>
```

Security and impersonation

If you use the **TableauServerUser** or **TableauServerUserFull** parameter in an initial SQL statement, you will create a dedicated connection that can’t be shared with other users. This will also restrict cache sharing, which can enhance security, but may also slow performance.

Troubleshoot 'create table' for MySQL and Oracle connections

For MySQL connections, tables are not listed after using initial SQL to create table

When you connect to MySQL, if you run an initial SQL statement like the following, tables might not show because of the way Tableau constructs the query:

```
CREATE TABLE TestV1.testtable77(testID int);
```

To resolve this issue, add `IF NOT EXISTS` to the SQL statement:

```
CREATE TABLE IF NOT EXISTS TestV1.TestTable(testID int);
```

For Oracle connections, using initial SQL to create table causes Tableau to stall

When you connect to Oracle and run an initial SQL statement like the following, Tableau is stalled with a spinning wheel because of the way Tableau constructs the query:

```
CREATE TABLE TEST_TABLE (TESTid int)
```

To resolve this issue, use the following SQL statement:

```
BEGIN
EXECUTE IMMEDIATE 'create table test_table(testID int)';
EXCEPTION
WHEN OTHERS THEN NULL;
END;
```

Set Up Data Sources

After you connect to your data, use the Data Source page to set up the data source and prepare your data for analysis. There are many optional configurations that you can make before you begin your analysis. The configurations that you make on the Data Source page creates the data source that Tableau uses to interpret and interact with your data.

The topics in this section describe how to use these configurations in order to optimize your data source for analysis.

Plan the Data Source

At the center of Tableau is your data. How successful you are with exploring data, answering questions, and building visualizations for yourself or others to consume all depend on the underlying data.

If your goal is to do some quick exploration or ad-hoc analysis, you might be able to hop in, connect to some data, drag and drop a bit to build some vizzes, and hop out with the information you need. But if your goal is to create an analysis or data source that will be used more than once, it's best to think through and plan your data source.

Tableau data sources

A Tableau data source is the link between your source data and Tableau. It is essentially the sum of your data (either as a live connection or an extract), the connection information, the names of tables or sheets containing data, and the customizations that you make on top of data to work with it in Tableau. Those customizations include things like how the data is combined and metadata such as calculations, renamed fields, and default formatting.

A Tableau *data source* may contain multiple *data connections* to different databases or files. Connection information includes where the data is located, such as a file name and path or a network location, and details on how to connect to your data, such as database server name and server sign-in information.

> **Note**: The terms *data source* and *data connection* have been used indiscriminately in the past and may lack clear distinction in older materials.
>
> It’s also important to bear in mind the difference between the Tableau data source (Tableau’s access to your data and all the customizations that may be involved) and the

source data itself (such as a database or Excel spreadsheet). Planning a data source in the sense of designing a database is not the scope of this topic.

Tableau data sources can remain embedded in the workbook where they were originally created, or they can be published separately. Published Data Sources allow for centralization and scaling of curated data sources. See **Best Practices for Published Data Sources** on page 3065 for more information.

Considerations for a Tableau data source

Before anything else, make sure you're clear on the purpose of the data source. If you're creating a catch-all data source for a wide variety of users with broad questions, you'd go about it differently than if you were constructing a niche data source optimized for performance. It's important to know your goal from the outset.

With that goal in mind, there are several items to consider and plan before you build a data source in Tableau:

- Location and access
- Shape and cleanliness
- The data model and combining data
- Metadata and customization
- Scalability, security, and discoverability
- Performance and freshness

Location and access

The location of your data can be as simple as a single excel file on your computer or a Google sheet you've created, or as complex as multiple databases using various technologies. Access can also involve specific drivers or connectors for a database or login information for a database server.

Questions to consider:

- Do the correct Tableau users have access and permissions to the data and to Tableau?
- What user account should be used to log into the database?
- Will user filtering or row level security be needed?
- Is the source data on premises or in the cloud?
  - This may have implications if the data is published to Tableau Online.
- Are there **Supported Connectors** on page 393?

- Are there any limitations for that database (does it support all the functionality you hope to use)?

Shape and cleanliness

Your organization may already have well-structured data that you can connect to or ETL processes you can leverage, or you may need to use Tableau Prep Builder to perform cleaning and shaping operations to get your data into a useful format.

Questions to consider:

- Is the data well-structured for use with Tableau?
- Should Tableau Prep Builder and Tableau Prep Conductor be used to clean and automate data prep flows?
- What calculations or manipulations would be best done natively in Tableau?
- Is any of this data unnecessary? Is any data missing?

The data model and combining data

Tableau needs to know how to query the data, which is conveyed by **The Tableau Data Model** on page 810. If the data comes from multiple tables, the data needs to be combined. Methods for combining data include relationships, joins, unions, and blends.

Questions to consider:

- Is the data spread across multiple systems?
- What is the best way to combine each table of data?

Metadata and customization

Data is often stored in a way that isn't easy for a person to use comfortably. For example, default field names can be unwieldy, custom formatting may be desired, definitions of abbreviations or codes may be missing, or common calculations may not be present in the underlying data. Creating a metadata layer with those pieces of information can make the data much easier to understand and use.

Questions to consider:

- What calculations should be added?
- Are the table and field names understandable?
- Are the default settings for the fields useful or should they be customized?
- Are there any adjustments that should be made to support Ask Data or Tableau Catalog?

  Note that Explain Data does not support multi-table data sources at this time.

Scalability, security, and discoverability

A data source may remain in the workbook where it was created—an embedded data source—or it may be published to Tableau Server or Tableau Online independently of any analysis as a content asset all its own. This centralization of published data sources has the added benefit of security and scaling, allowing multiple users to access a consistent data source. Published data sources can also take advantage of Tableau Catalog and recommendations and provide a single source of truth for an organization.

Questions to consider:

- Should the data source stay embedded in the workbook?

(The following questions assume the data source is published)

- Where will the data source be published?
- Who will own the data source?
- What authentication and permissions should it have?
- Will it be certified?

Performance and freshness

In addition, Tableau data sources can be live connections to the source data or the data can be copied out as a refreshable extract. Extracts can boost the analysis performance or prevent resource drains against database systems.

Questions to consider:

- Should the data connection be live or an extract?
- If extracted, will there be a refresh schedule? (Tableau Server | Tableau Online)
- If published to Tableau Online, will Tableau Bridge need to be used?

Build quality data sources

There's a lot that goes into a well-designed and performant data source. You don't have to publish your first attempt, though. Like creating a good visualization, building a data source can be an ongoing process. Try your hand at combining the data you think you need and then see if you can build the viz you want. You can always go back and write a new calculation or bring in another table of data. Do some user testing and ask a colleague if they can figure out your field names.

Of course, it's best to only publish data sources you're confident in and only certify ones that have passed quality assurance. By the time your data source is in front of users, they should be able to trust it.

Combine Data

If your data comes from one table, you can connect to your data to create the data source, drag the table onto the canvas, and then start building your view. But if your data is spread across multiple tables—or across multiple databases—you'll need to combine it. Combining data happens on the Data Source Page.

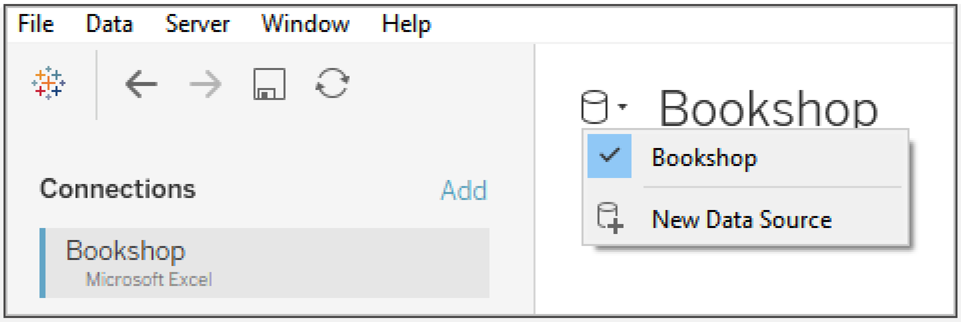
Single database

1. Connect to your data. This creates a data source.
2. Drag a table or sheet of data to the canvas and release it.
3. Drag a second table or sheet to the canvas and release it. Either a noodle will form, indicating an automatic relationship has formed, or the Edit Relationship dialog will open, prompting you to tell Tableau which field those tables relate the two tables.
4. Continue adding tables.

Multiple databases

1. Connect to your data. This creates a data source.
2. Drag a table or sheet of data to the canvas and release it.
3. After bringing out the first table of data, click the **Add** link to the right of the Connections heading in the Left pane.
4. Switch between data connections in the Left pane, then drag out the desired table to the canvas and release it. Either a noodle will form, indicating an automatic relationship has formed, or the Edit Relationship dialog will open, prompting you to tell Tableau which field those tables relate the two tables.
5. Continue adding tables, switching between data connections as relevant.

Note: In order to relate tables across different databases, they must be added as **data connections** within the same data source, not a new **data source**. Data connections are added from the Add link in the Left pane. New data sources can be created from the Data menu or the drop down menu from the icon next to the Data Source's name, as seen in the image below.



Methods of combining data

Relationships are the default way to combine data in Tableau. Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. If necessary, tables can also be joined or unioned. Data sources can also be blended. Let's take a look the options for how data can be combined in Tableau and some of the situations when each method is useful.

Relationships	Establish the potential for joins between two tables based on related fields. Does not merge data together to create a new, fixed table. During analysis, queries the relevant tables automatically using the contextually-appropriate joins to generate a custom table of data for that analysis. Maintains the appropriate level of detail, does not lose data, keeps appropriate aggregations, and handles nulls.
Joins	Merge two tables of data based on a join clause and join type to form a new, fixed table of data. Often used to add new columns of data across the same basic row structure. May cause data loss with some join types if fields are not present in all tables. May cause data duplication if tables are at different levels of detail.
Unions	Merge two or more tables of data to form a new, fixed table of data. Used to append (add new rows of) data across the same basic column structure.
Blends	Work across two or more separate Tableau data sources. Data remains separate. Tableau queries the data sources independently and visualizes the results together in the view, based on the linking fields established for

that sheet. Mimics the behavior of a left join and may filter data from secondary data sources.

Note: It's also possible to combine data with Custom SQL. Most of the time, it's best to combine data directly in the canvas or with data blending. This allows Tableau the greatest flexibility for optimization. However, in certain scenarios, there may be reasons to use Custom SQL. Keep in mind doing so can have performance implications because Tableau is forced to execute the Custom SQL statement as written.

Relationships

Relating is a method for working with data from multiple tables based on shared fields—columns—between those tables. Establishing a relationship informs Tableau how rows can be connected across tables. Tableau holds that information but does not immediately bring the rows together. Instead, when a visualization is created, the fields involved in the analysis are traced back through the relationship and the appropriate data is returned from its original table.

Because the data is never fully brought together into a single table outside the context of a visualization, relationships are useful when the data is at different levels of detail, or granularity. For example, if you need to work with data about daily rainfall in one table but hourly temperatures in another.

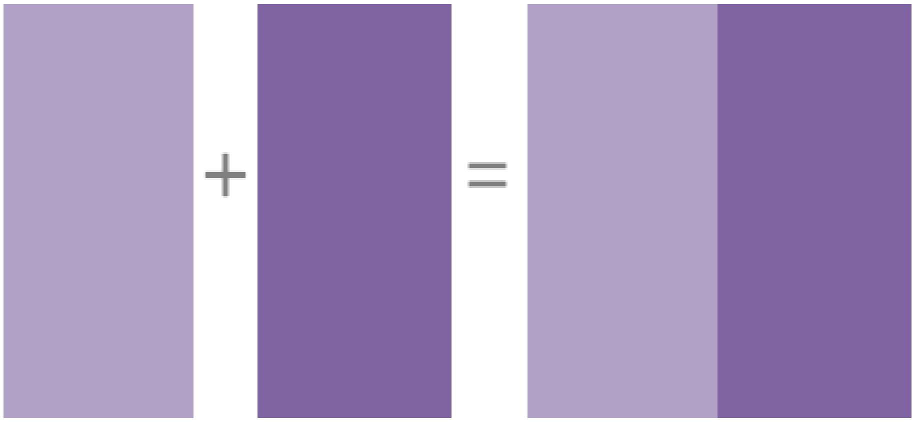
Tableau supports relationships between tables in the same database and between tables in different databases.

For more information about how to combine tables using relationships, see **Relate Your Data** on page 766 and these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Join

Joining is a method for combining tables related by common fields. The result of combining data using a join is a table that extends horizontally by adding columns of data.



For more information about how to join data in Tableau, see **Join Your Data** on page 883.

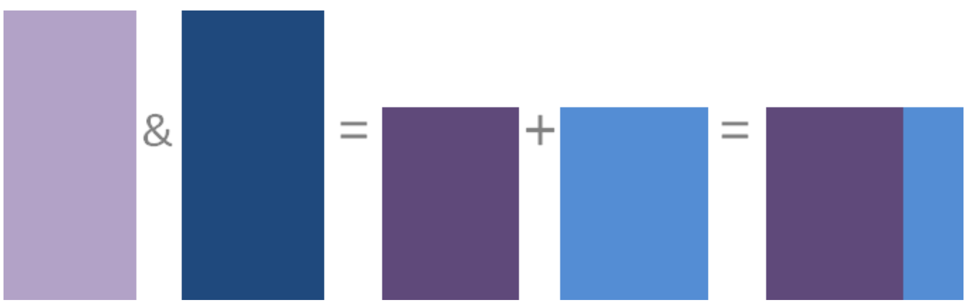
Data blending

When you use data blending to combine your data, you combine data in what is called a primary data source with common fields from one or more secondary data sources.

Data blending is useful when you need to change how the data source is configured on a sheet-by-sheet basis, when you want to combine databases that don't allow relationships or joins

such as cube data sources or Published Data Sources.

The result of combining data using data blending is a virtual table that extends horizontally by adding columns of data. The data from each data source will be aggregated to a common level before being displayed together in the visualization.



For more information about how data blending works and how to blend your data in Tableau, see **Blend Your Data** on page 910.

Union

Unioning is a method for appending values (rows) to tables. You can union tables if they have the same columns. The result of combining data using a union is a virtual table that has the same columns but extends vertically by adding rows of data.

You can union tables in one of two ways: manually or using wildcard search. For more information about how to union data using one of these methods in Tableau, see **Union Your Data** on page 942.

Structure Data for Analysis

There are certain concepts that are fundamental to understanding data prep and how to structure data for analysis. Data can be generated, captured, and stored in a dizzying variety of formats, but when it comes to analysis, not all data formats are created equal.

Data preparation is the process of getting well formatted data into a single table or multiple related tables so it can be analyzed in Tableau. This includes both the structure, i.e. rows and columns, as well as aspects of data cleanliness, such correct data types and correct data values.

Tip: It may help to go through the following topic with a data set of your own. If you do not already have a data set you can use, see our tips for finding good data sets.

How structure impacts analysis

The structure of your data may not be something you can control. The rest of this topic assumes you have access to the raw data and the tools needed to shape it, such as Tableau Prep Builder. However, there may be situations when you can't pivot or aggregate your data as desired. It is often still possible to perform the analysis but you may need to change your calculations or how you approach the data. For an example of how to perform the same analysis with different data structures, see Tableau Prep Day in the Life Scenarios: Analysis with the Second Date in Tableau Desktop. But if you can optimize the data structure it will likely make your analysis much easier.

Data Structure

Tableau Desktop works best with data that is in tables formatted like a spreadsheet. That is, data stored in rows and columns, with column headers in the first row. So what should be a row or column?

What is a row?

A row, or record, can be anything from information around a transaction at a retail store, to weather measurements at a specific location, or stats about a social media post.

It's important to know what a record (row) in the data represents. This is the *granularity* of the data.

Here, each record is a day

Date	Max TemperatureF	Mean Tempe
1/1/2015	42	
1/2/2015	42	
1/3/2015	41	
1/4/2015	51	
1/5/2015	54	
1/6/2015	54	
1/7/2015	46	
1/8/2015	46	
1/9/2015	50	
1/10/2015	46	

Here, each record is a month

Date	Max TemperatureF	Mean TemperatureF	Min TemperatureF
January	63	45	26
February	62	49	33
March	69	51	31
April	77	52	37
May	82	59	43
June	92	68	49
July	95	71	54
August	92	69	54
September	81	61	45
October	74	58	45
November	60	44	25

Tip:A best practice is to have a unique identifier, a value that identifies each row as a unique piece of data. Think of it like the social security number or URL of each record. In Superstore, that would be Row ID. Note that not all data sets have a UID but it can't hurt to have one.

Try to make sure you can answer the question "What does a row in the data set represent?". This is the same as answering "What does the TableName(Count) field represent?". If you can't articulate that, the data might be structured poorly for analysis.

Aggregation and Granularity

A concept related to what makes up a row is the idea of aggregation and granularity, which are opposite ends of a spectrum.

Aggregation

- refers to how multiple data values are brought together into a single value, such as counting all the Google searches for Pumpkin Spice or taking the average of all the temperature readings around Seattle on a given day.
- By default, measures in Tableau are always aggregated. The default aggregation is SUM. You can change the aggregation to options like Average, Median, Count Distinct, Minimum, etc.

Granularity

- refers to how detailed the data is. What does a row or record in the data set represent? A person with malaria? A provinces' total cases of malaria for the month? That's the granularity.
- Knowing the granularity of the data is crucial to working with level of detail (LOD) expressions.

Understanding aggregation and granularity is a critical concept for many reasons; it impacts things like finding useful data sets, building the visualization you want, relating or joining data correctly, and using LOD expressions.

Tip: For more information, see the training video Aggregation and Granularity or Data Aggregation in Tableau in the Help

What is a field or column?

A *column* of data in a table comes into Tableau Desktop as a *field* in the data pane, but they are essentially interchangeable terms. (We save the term *column* in Tableau Desktop for use in the columns and rows shelf and to describe certain visualizations.) A field of data should contain items that can be grouped into a larger relationship. The items themselves are called *values* or *members* (only discrete dimensions contain members).

What values are allowed in a given field are determined by the *domain* of the field (see the note below). For example, a column for "grocery store departments" might contain the members "deli" "bakery", "produce", etc., but it wouldn't include "bread" or "salami" because those are items, not departments. Phrased another way, the domain of the department field is limited to just the possible grocery store departments.

Additionally, a well-structured data set would have a column for "Sales" and a column for "Profit", not a single column for "Money", because profit is a separate concept from sales.

- The domain of the Sales field would be values ≥ 0, since sales cannot be negative.
- The domain of the Profit field, however, would be all values, since profit can be negative.

> **Note**: *Domain* can also mean the values present in the data. If the column "grocery store department" erroneously contained "salami", by this definition, that value would be in the domain of the column. The definitions are slightly contradictory. One is the values that could or should be there, the other is values that actually are there

Categorizing fields

Each column in the data table comes into Tableau Desktop as a field, which appears in the **Data** pane. Fields in Tableau Desktop must be either a dimension or measure (separated by a line within tables in the **Data** pane) and either discrete or continuous (color coded: blue fields are discrete and green fields are continuous).

- *Dimensions* are qualitative, meaning they can't be measured but are instead described. Dimensions are often things like city or country, eye color, category, team name, etc. Dimensions are usually discrete.
- *Measures* are quantitative, meaning they can be measured and recorded with numbers. Measures can be things like sales, height, clicks, etc. In Tableau Desktop, measures are automatically aggregated; the default aggregation is SUM. Measures are usually continuous.
- *Discrete* means individually separate or distinct. Toyota is distinct from Mazda. In Tableau Desktop, discrete values come into the view as a label and they create panes.
- *Continuous* means forming an unbroken, continuous whole. 7 is followed by 8 and then it's the same distance to 9, and 7.5 would fall midway between 7 and 8. In Tableau Desktop, continuous values come into the view as an axis.
- Dimensions are usually discrete, and measures are usually continuous. However, this is not always the case. Dates can be either discrete or continuous.
  - Dates are dimensions and automatically come into the view as discrete (aka date parts, such as "August", which considers the month of August without considering other information like the year). A trend line applied to a timeline with discrete dates will be broken into multiple trend lines, one per pane.

- We can chose to use continuous dates if preferred (aka date truncations, such as "August 2017", which is different than "August 2018"). A trend line applied to a timeline with continuous dates will have a single trend line for the entire date axis.

Tip: For more information, see the training video Understanding Pill Types or Dimensions and Measures, Blue and Green in the Help.

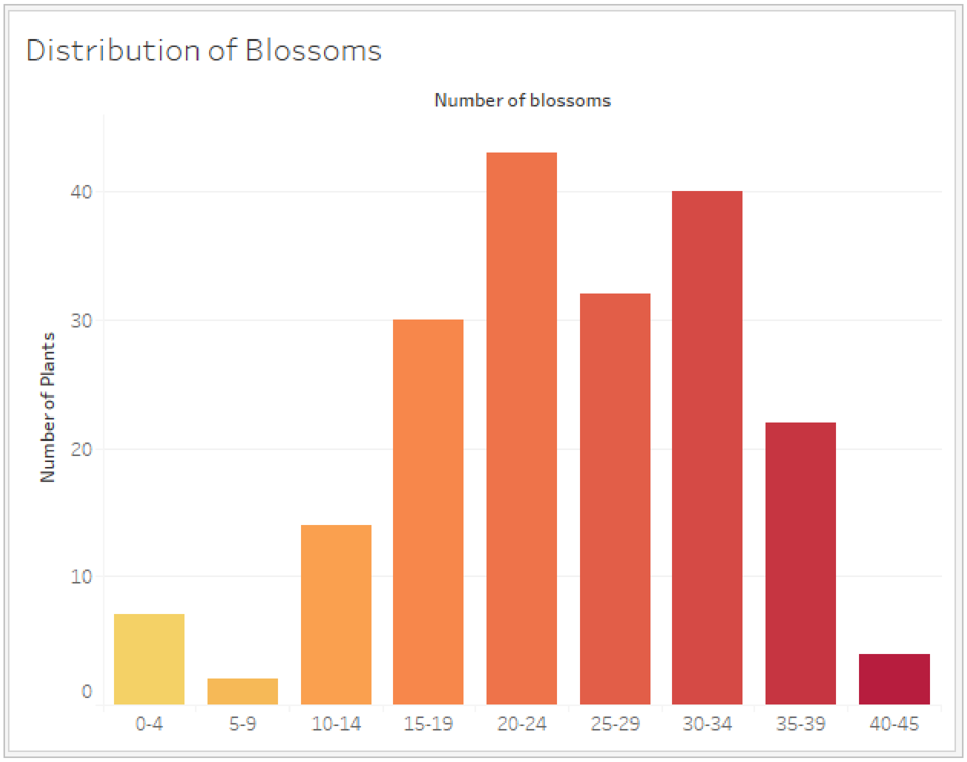
In Tableau Prep, no distinction is made for dimensions or measures. Understanding the concepts behind discrete or continuous are important, however, for things like understanding the detail versus summary presentation of data in the profile pane.

- Detail: the detail view shows every domain element as a discrete label and has a visual scrollbar to provide a visual overview of all the data.
- Summary: the summary view shows the values as binned on a continuous axis as a histogram.

Binning & Histograms

A field like age or salary is considered continuous. There is a relationship between the age 34 and 35, and 34 is as far from 35 as 35 is from 36. However, once we're past age 10 or so, we usually stop saying things like we're "9 and a half" or "7 and ¾". We’re already binning our age to neat year-sized increments. Someone who is 12,850 days old is older than someone who is 12,790 days old, but we draw a line and say they're both 35. Similarly, age groupings are often used in place of actual ages. Child prices for movie tickets might be for kids 12 and under, or a survey may ask you to select your age group, such as 20-24, 25-30, etc.

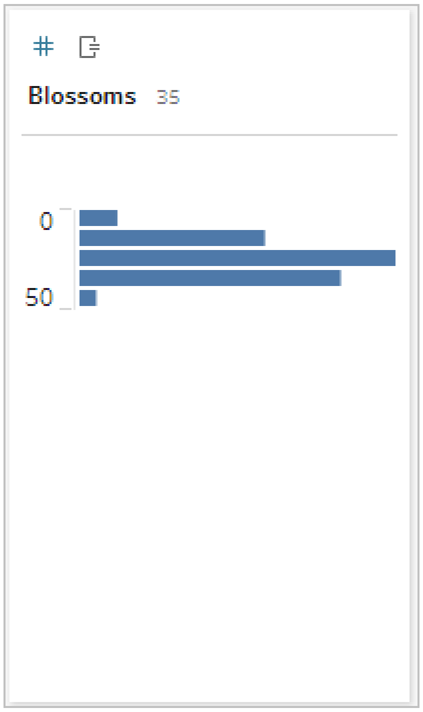
Histograms are used to visualize the distribution of numerical data using binning. A histogram is similar to a bar chart, but rather than being discrete categories per bar, the rectangles making up the histogram span a *bin* of a continuous axis, such as range of the number of blossoms (0-4, 5-9, 10-14, etc.). The height of the rectangles is determined by frequency or count of those values. Here, the y-axis is the count of plants that fall into each bin. Seven plants have 0-4 blossoms, two plants have 5-9 blossoms, and 43 plants have 20-24 blossoms.

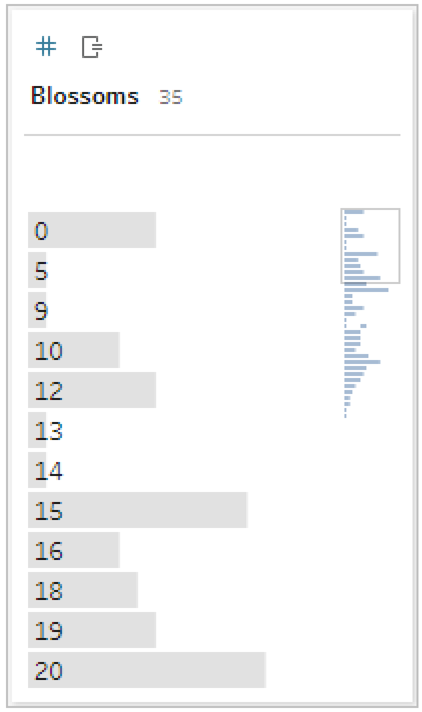


In Tableau Prep, the summary view is a histogram of binned values. The detail view shows the frequency for every value and has a visual scrollbar off to the side that shows the overall distribution of the data.

Summary view

Detail view





Distributions and outliers

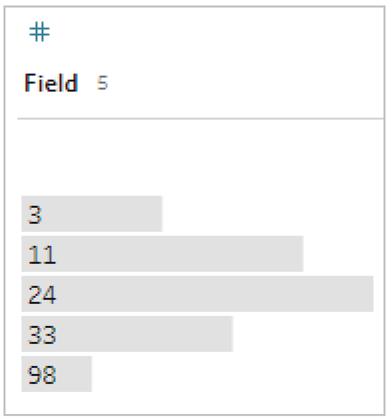
Seeing the distribution of a data set can help with outlier detection.

- *Distribution*: the shape of the data in a histogram, though this depends on the size of the bins. Being able to see all your data in a histogram view can help identify if the data seems correct and complete. The shape of the distribution will only be of use if you know the data and can interpret whether or not the distribution makes sense.
  - For example, if we were to look at a data set of the number of homes with broadband internet from 1940-2017, we'd expect to see a very skewed distribution. However, if we were to look at the number of homes with broadband internet from January 2017 to December 2017 we'd expect a fairly uniform distribution.
  - If we were to look at a data set of Google searches for "Pumpkin Spice Latte", we'd expect to see a fairly sharp peak in the fall, whereas searches for “"convert Celsius to Fahrenheit" would likely be fairly stable.
- *Outlier*: a value that is extreme compared to other values. Outliers may be correct values or they may be indicative of an error.

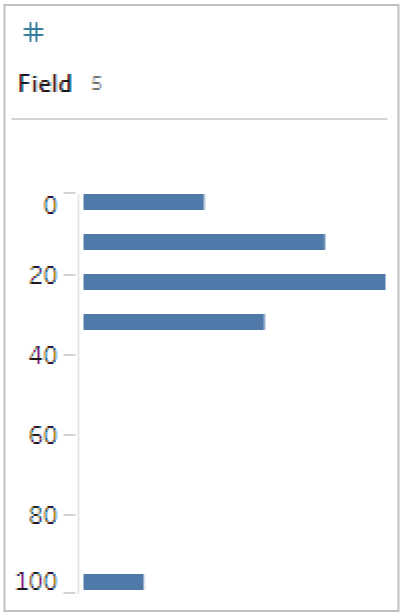
- Some outliers are correct and indicate actual anomalies; these should not be removed or modified.
- Some outliers indicate issues with data cleanliness, such as a salary of $50 instead of $50,000 because a period was typed instead of the comma.

Visual outlier detection with distributions

If you were to see a list like this:



at first glance it doesn't look odd. But if, instead of a list of labels, this was plotted on a continuous binned axis, it would look like this:



And it's much more obvious that the last observation is farther away from the first and may be an outlier due to error.

Data Types

Databases, unlike spreadsheets, usually enforce strict rules on data types. Data types classify the data in a given field and provide information about how the data should be formatted, interpreted, and what operations can be done to that data. For example, numerical fields can have mathematical operations applied to them and geographic fields can be mapped.

Tableau Desktop assigns whether a field is a dimension or measure, but fields have other characteristics that depend on their data type. These are indicated by the icon each field has (though some types share an icon). Tableau Prep uses the same data types. If data type is enforced on a column and an existing value doesn't match its assigned data type, it may be displayed as null (because "purple" doesn't mean anything as a number).

Some functions require specific data types. For example, you cannot use CONTAINS with a numerical field. Type functions are used to change the data type of a field. For example, DATEPARSE can take a text date in a specific format and make it a date, thus enabling things like automatic drill down in the view.

Icon	Data type
	Text (string) values

Icon	Data type
	Date values
	Date & Time values
	Numerical values
T|F	Boolean values (relational only)
	Geographic values (used with maps)

Tip: For more information, refer to the Help article on Data Types.

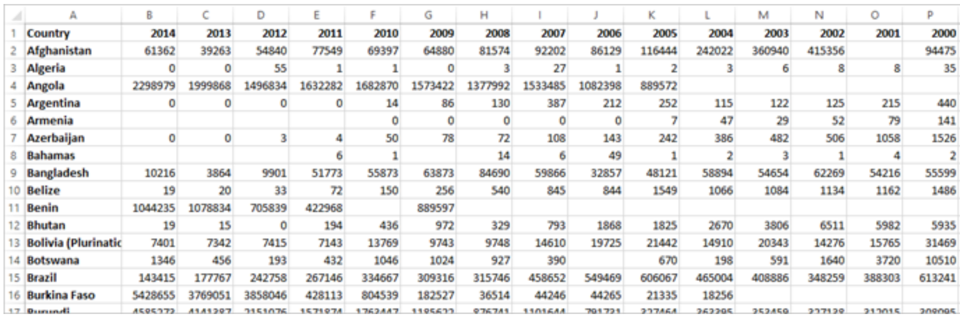
Pivot and Unpivot Data

People-friendly data is often captured and recorded in a wide format, with many columns. Machine-readable data, like Tableau prefers, is better in a tall format, with fewer columns and more rows.

> **Note**: Traditionally, pivoting data means going from tall to wide (columns to rows), and unpivoting means going from wide to tall (rows to columns). However, Tableau uses the word *pivot* to mean going from wide (people-friendly) to tall (machine-readable) by turning rows into columns. In this document, *pivot* will be refer to the Tableau sense of the word. For clarity, it can help to specify "pivot columns to rows" or "pivot rows to columns".
>
> For more information, refer to the Help articles Pivot Your Data and Tips for Working with your Data.

Wide data

In the WHO malaria data set, there is a column for country, then a column per year. Each cell represents the number of cases of malaria for that country and year. In this format we have 108 rows and 16 columns.



Country	2014	2013	2012	2011	2010	2009	2008	2007	2006	2005	2004	2003	2002	2001	2000
Afghanistan	61362	39263	54840	77549	69397	64880	81574	92202	86129	116444	242022	360940	415356		94475
Algeria	0	0	55	1	1	0	3	27	1	2	3	6	8	8	35
Angola	2298979	1999868	1496834	1632282	1682870	1573422	1377992	1533485	1082398	889572					
Argentina	0	0	0	0	14	86	130	387	212	252	115	122	125	215	440
Armenia					0	0	0	0	0	7	47	29	52	79	141
Azerbaijan	0	0	3	4	50	78	72	108	143	242	386	482	506	1058	1526
Bahamas				6	1		14	6	49	1	2	3	1	4	2
Bangladesh	10216	3864	9901	51773	55873	63873	84690	59866	32857	48121	58894	54654	62269	54216	55599
Belize	19	20	33	72	150	256	540	845	844	1549	1066	1084	1134	1162	1486
Benin	1044235	1078834	705839	422968		889597									
Bhutan	19	15	0	194	436	972	329	793	1868	1825	2670	3806	6511	5982	5935
Bolivia (Plurinatic	7401	7342	7415	7143	13769	9743	9748	14610	19725	21442	14910	20343	14276	15765	31469
Botswana	1346	456	193	432	1046	1024	927	390		670	198	591	1640	3720	10510
Brazil	143415	177767	242758	267146	334667	309316	315746	458652	549469	606067	465004	408886	348259	388303	613241
Burkina Faso	5428655	3769051	3858046	428113	804539	182527	36514	44246	44265	21335	18256				

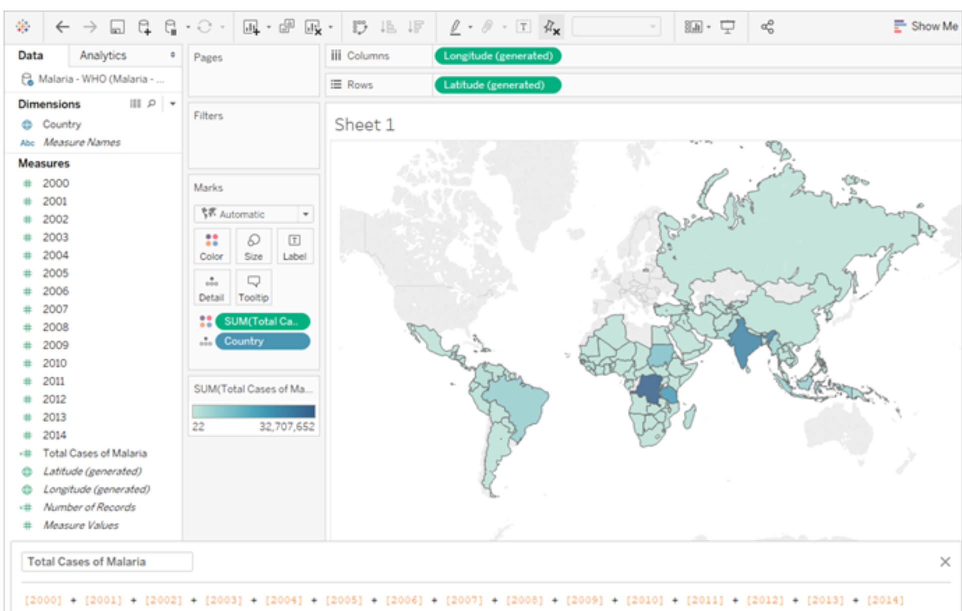
It's easy for a person to read and understand this format. However, if we were to bring this data into Tableau Desktop, we get a field per column. We have a field for 2000, a field for 2001, a field for 2002, etc.

To think of it another way, there are 15 fields that all represent the same basic thing—number of reported cases of malaria—and no single field for time. This makes it very hard to do analysis across time as the data is stored in separate fields.

Example: working with wide data

Q: How would we create a map that shows the total number of malaria cases per country from 2000 to 2014?

A: Create a calculated field to sum all the years.



Note: this image has not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

Another indication that this format isn't ideal for analysis can be seen in the fact that nowhere do we have information about what the actual values mean. For Algeria in 2012, we have the value 55. Fifty five what? It's not clear from the structure of the data.

	A	B	C	D	E
1	**Country**	**2014**	**2013**	**2012**	**2011**
2	**Afghanistan**	61362	39263	54840	77549
3	**Algeria**	0	0	55	1
4	**Angola**	2298979	1999868	1496834	1632282

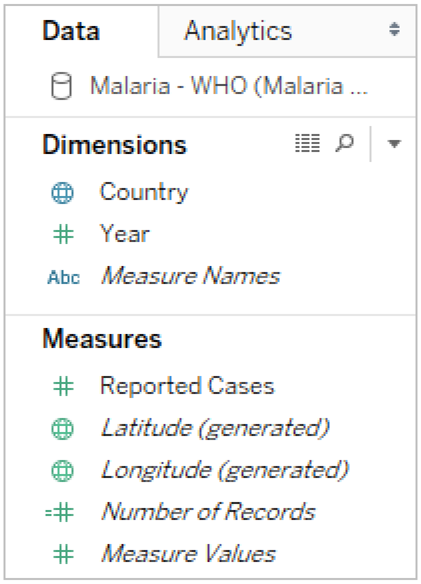
If the name of the column isn't describing what the values are but rather conveys additional information, this is a sign the data needs to be pivoted.

Tall data

If we pivot the data, we reshape the data from wide to tall. Now, rather than having a column for each year, we have a single column, Year, and a new column, Reported Cases. In this format we have 1,606 rows and 3 columns. This data format is taller rather than wider.

	A	B	C
1	Country	Year	Reported Cases
2	Afghanistan	2000	94,475
3	Afghanistan	2001	
4	Afghanistan	2002	415,356
5	Afghanistan	2003	360,940
6	Afghanistan	2004	242,022
7	Afghanistan	2005	116,444
8	Afghanistan	2006	86,129
9	Afghanistan	2007	92,202
10	Afghanistan	2008	81,574
11	Afghanistan	2009	64,880
12	Afghanistan	2010	69,397
13	Afghanistan	2011	77,549
14	Afghanistan	2012	54,840
15	Afghanistan	2013	39,263
16	Afghanistan	2014	61,362

Now in Tableau Desktop, we have a field for Year and a field for Reported Cases as well as the original Country field. It's much easier to do analysis because each field represents a unique quality about the data set—location, time, and value.

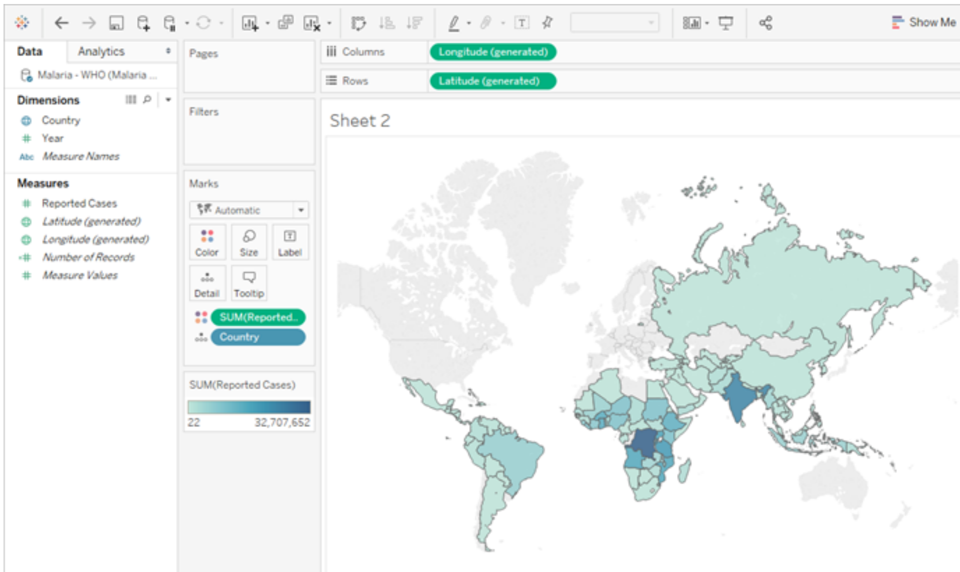


Note: this image has not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

Example: working with tall data

Q: How would we create a map that shows the total number of malaria cases per country from 2000 to 2014?

A: Use the Reported Cases field.



Note: this image has not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

Now it's easy to see that for Algeria in 2012, the 55 refers to the number of reported cases (because we could label this new column).

	A	B	C
1	Country	Year	Reported Cases
28	Algeria	2011	1
29	Algeria	2012	55
30	Algeria	2013	0

Note: In this example, the wide data consisted of a single record per country. With the tall data format, there are now 15 rows for each country (one for each of the 15 years in the data). It's important to keep in mind that there are now multiple rows per country.

If there was a column for Land Area, that value would be repeated for each of the 15 rows for each country in a tall data structure. If you created a bar chart by bringing out **Country** to **Rows** and **Land Area** to **Columns**, by default the view would sum the land area for all 15 rows per country.

For some fields it may be necessary to compensate for double counting values by aggregating with an average or minimum rather than sum or filtering.

Normalization

Relational databases are made up of multiple tables that can be related or linked together in some way. Each table contains a unique identifier, or key, per record. By relating or joining on the keys, records can be linked to provide more information than is contained in a single table. What information goes into each table is dependent on the data model used, but the general principle is around reducing duplication.

For example, consider event planning for an event like a wedding. We need to keep track of information at the level of groups (such as families or couples) as well as the level of individuals.

A table could be created that combines all the information together:

ID	Name	Group	Dietary	Seating	Attending	Address	Invitation	Gift	Bride
10	Cedar	Tree	Omnivore	1	yes	87 Forest Ave	sent	pebble	yes
11	Redwood	Tree	Omnivore	1	yes	87 Forest Ave	sent	pebble	yes
12	Fir	Tree	Vegan	1	yes	87 Forest Ave	sent	pebble	yes
13	Madrona	Tree	Omnivore		no	87 Forest Ave	sent	pebble	yes
14	Chanterelle	Mushroom	Omnivore	2	yes	3 Troll St	sent	leaf	no
15	Cremini	Mushroom	Kosher	2	yes	3 Troll St	sent	leaf	no
16	Portobello	Mushroom	Omnivore	2	yes	3 Troll St	sent	leaf	no
17	Hydrangea	Flower				652 Meadow Ln	undeliverable		yes
18	Dahlia	Flower				652 Meadow Ln	undeliverable		yes

However, if an address is incorrect and needs to be fixed, it must be fixed across multiple rows, potentially leading to errors or conflicts. A better structure is to create two tables, one for information that pertains to the group (such as address and if the invitation was sent) and one for information pertaining to the individuals (for things like seating assignments and dietary restrictions).

Group table

ID	Group	Address	Invitation
34	Tree	87 Forest Ave	sent
35	Mushroom	3 Troll St	sent
36	Flower	652 Meadow Ln	undeliverable

Individual table

ID	Name	Group	Dietary	Seating	Attending
10	Cedar	Tree	Omnivore	1	yes
11	Redwood	Tree	Omnivore	1	yes
12	Fir	Tree	Vegan	1	yes
13	Madrona	Tree	Omnivore		no
14	Chanterelle	Mushroom	Omnivore	2	yes
15	Cremini	Mushroom	Kosher	2	yes
16	Portobello	Mushroom	Omnivore	2	yes
17	Hydrangea	Flower			
18	Dahlia	Flower			

It's much easier to track and analyze group-level information in the group table and individual-level information in the individual table. For example, the number of chairs needed can be obtained from the number of Attending = Yes records in the individual table, and the number of stamps needed for thank-yous can be obtained from the number of records in the group table where Gift isn't null.

The process of breaking up all the data into multiple tables—and figuring out which table contains which columns—is called *normalization*. Normalization helps reduce redundant data and simplifies the organization of the database.

However, there may be times when information is needed that spans multiple tables. For example, what if we wanted to balance seating arrangements (individuals) such that groups from the bride's side are intermingled with groups from the groom's side? (The bridge or groom affiliation is tracked at the group level.) To achieve this, we need to relate the tables back together so the individuals are associated with information about their group. Proper normalization isn't just breaking up tables, it also requires the presence of a shared, related field or unique identifier than can be used to combine the data back together again. Here, that related field is Group. That field is present in both tables, so we can join on this field and return to our original, single table format. This is a *denormalized* structure.

So why didn't we just keep the original denormalized table? It is harder to maintain and was storing redundant information. At scale, the level of data duplication can be come massive. Storing the same information over and over isn't efficient.

Normalized tables have a few key properties:

- Each row needs a unique identifier
- Each table needs a column or columns that can be used to connect it back to other tables (key).

These shared (key) columns are used for relating or joining tables back together. For our data, the relationship or join clause would be on the Group field in each table.

Join Types

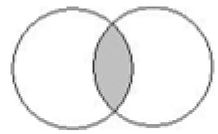
Although the default method for combing data in Tableau Desktop is relating, there are cases when you may want to join tables in Tableau Desktop or Tableau Prep Builder. For a basic overview of joins and join types, see Join Your Data.

Name	# of Siblings
Taylor	2
Alex	3
Shannon	0
Tracy	1

Name	Eye Color
Taylor	Blue
Alex	Brown
Morgan	Brown

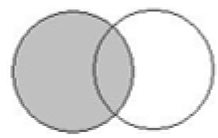
Inner Join

Name	# of Siblings	Eye Color
Taylor	2	Blue
Alex	3	Brown



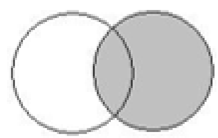
Left Join

Name	# of Siblings	Eye Color
Taylor	2	Blue
Alex	3	Brown
Shannon	0	*null*
Tracy	1	*null*



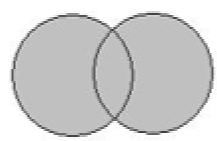
Right Join

Name	Eye Color	# of Siblings
Taylor	Blue	2
Alex	Brown	3
Morgan	Brown	*null*



Outer Join

Name	# of Siblings	Eye Color
Taylor	2	Blue
Alex	3	Brown
Shannon	0	*null*
Tracy	1	*null*
Morgan	*null*	Brown



"Tidy" Data

Hadley Wickham published an article in 2014 in the Journal of Statistical Software called "Tidy Data" (August 2014, Volume 59, Issue 10). This article does an excellent job of laying out a framework for data that is well-structured for analysis. The article can be found here (Hadley Wickham's Academic Portfolio) or here (hosted by r-project.org).

> **Note**: The article is hosted on external websites. Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the owners if you have questions regarding their content.

Relate Your Data

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. A relationship describes how two tables relate to each other, based on common fields, but does not merge the tables together. When a relationship is created between tables, the tables remain separate, maintaining their individual level of detail and domains.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

Learn more: The ability to relate your data is an important feature of Tableau's new data modeling capabilities. For more information, see **What's Changed with Data Sources and Analysis** on page 785. Learn more about how relationships work in these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Watch a video: For an overview of data source enhancements and an introduction to using relationships in Tableau, see this 5-minute video.

What are relationships?

Relationships are the flexible, connecting lines created between the logical tables in your data source. Some people affectionately call relationships "noodles", but we usually refer to them as "relationships" in our help documentation.

We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to.

Relationships provide several advantages over using joins for multi-table data:

- You don't need to configure join types between tables. You only need to select the fields to define the relationship.
- Related tables remain separate and distinct; they are not merged into a single table.
- Relationships use joins, but they are automatic. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data.

- Tableau uses relationships to generate correct aggregations and appropriate joins during analysis, based on the current context of the fields in use in a worksheet.
- Multiple tables at different levels of detail are supported in a single data source. You can build data models that contain more tables, and reduce the number of data sources needed to build a viz.
- Unmatched measure values are not dropped (no accidental loss of data).
- Avoids data duplication and filtering issues that can sometimes result from joins.
- Tableau will generate queries only for the data that is relevant to the current view.

For related information, see:

- **Use Relationships for Multi-table Data Analysis** on page 782
- **How Relationships Differ from Joins** on page 821
- **How Analysis Works for Multi-table Data Sources that Use Relationships** on page 841
- **Don’t Be Scared of Relationships** on page 861

Requirements for relationships

- When relating tables, the fields that define the relationships must have the same data type.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't edit relationships in a published data source.
- You can't define relationships between published data sources.
- Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page in Tableau Online or Tableau Server.

Factors that limit the benefits of using related tables:

- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions).

Data that can't be related

Most relational connection types are completely supported. Cubes, SAP HANA (with OLAP attribute), JSON, and Google Analytics are limited to a single logical table in Tableau 2020.2. Stored procedures can only be used within a single logical table.

Published data sources can't be related to each other. You also can't edit published data sources.

Unsupported

- Cube databases do not support the new logical layer. Connecting to a cube offers the same experience as pre-2020.2.
- Stored Procedures: Don't support federation, relationships, or joins. They are represented in a single logical table, and don't allow opening the Join/Union canvas (physical layer).
- Splunk: Doesn't support left joins (and therefore relating logical tables).
- JSON: Doesn't support federation, custom SQL, joins, or relationships (only unions).
- Datasources that do not support LOD calcs. For more information, see **Data Source Constraints for Level of Detail Expressions** on page 2471.

Limited support

- Salesforce and WDC Standard Connections: These are represented as joined tables within a logical table. Adding these connections is currently only supported for single, logical table data sources. Standard connections cannot join to an existing table.
- SAP HANA: Doesn't currently support relating logical tables when the connection has the OLAP attribute set.

Create and define relationships

After you drag the first table to the top-level canvas of the data source, each new table that you drag to the canvas must be related to an existing table. When you create relationships between tables in the logical layer, you are building the data model for your data source.

> **Note**: You can't edit the data model of a published data source.

Create a relationship

You create relationships in the logical layer of the data source. This is the default view of the canvas that you see in the Data Source page.

1. Drag a table to the canvas.
2. Drag another table to the canvas. When you see the "noodle" between the two tables, drop that table.

   The Edit Relationship dialog box opens. Tableau automatically attempts to create the relationship based on existing key constraints and matching fields to define the relationship. If it can't determine the matching fields, you will need to select them.

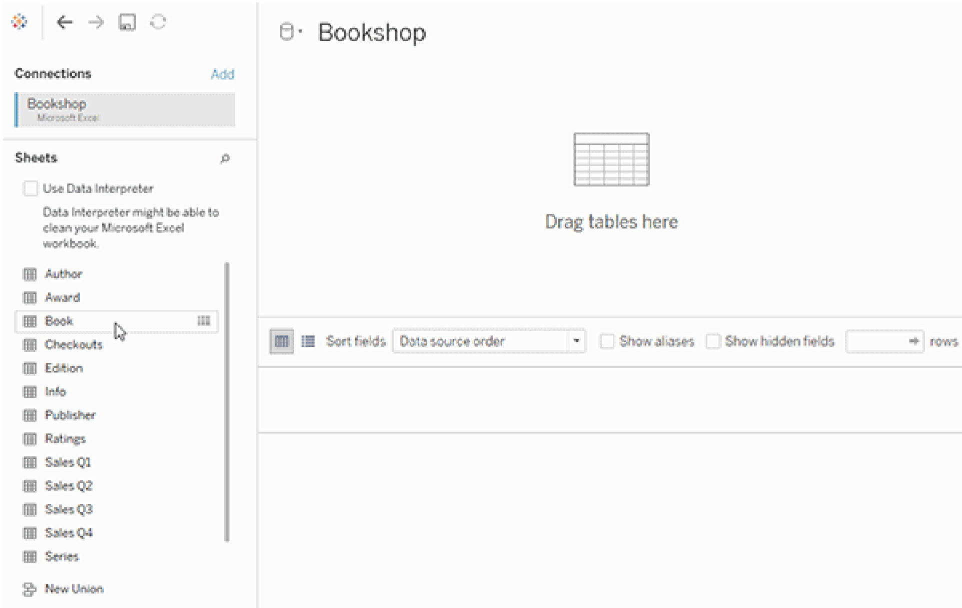
   To change the fields: Select a field pair, and then click in the list of fields below to select a new pair of matching fields.

   To add multiple field pairs: After you select the first pair, click **Close**, and then click **Add more fields**.

   > **Note**: In Tableau 2020.3 and later, you can create relationships based on calculated fields, and compare fields used for relationships using operators in the relationship definition. Note that the following connectors do not support inequality operators: Google BigQuery, MapR, Salesforce.

   If no constraints are detected, a **Many-to-many** relationship is created and referential integrity is set to **Some records match**. These default settings are a safe choice and provide the most a lot of flexibility for your data source. The default settings support full outer joins and optimize queries by aggregating table data before forming joins during analysis. All column and row data from each table becomes available for analysis.

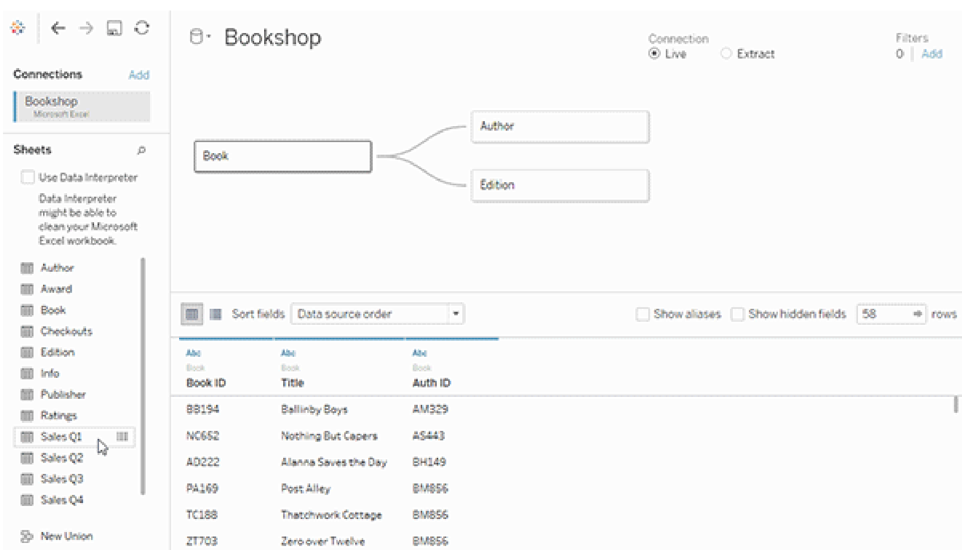
   In many analytical scenarios, using the default settings for a relationship will give you all of the data you need for analysis. Using a many-to-many relationship will work even if your data is actually many-to-one or one-to-one. If you know the particular cardinality and referential integrity of your data, you can adjust the Performance Options settings to describe your data more accurately and optimize how Tableau queries the database.
3. Add more tables following the same steps, as needed.



After you have built your multi-table, related data source, you can dive into exploring that data. For more information, see **How Analysis Works for Multi-table Data Sources that Use Relationships** on page 841 and **Troubleshooting multi-table analysis** on page 856.

Move a table to create a different relationship

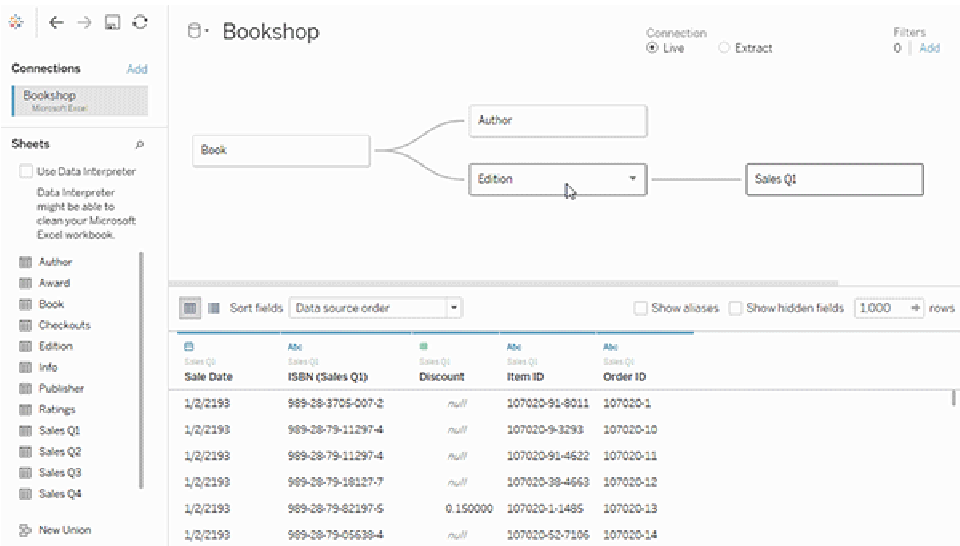
To move a table, drag it next to a different table. Or, hover over a table, click the arrow, and then select **Move**.



Tip: Drag a table over the top of another table to replace it.

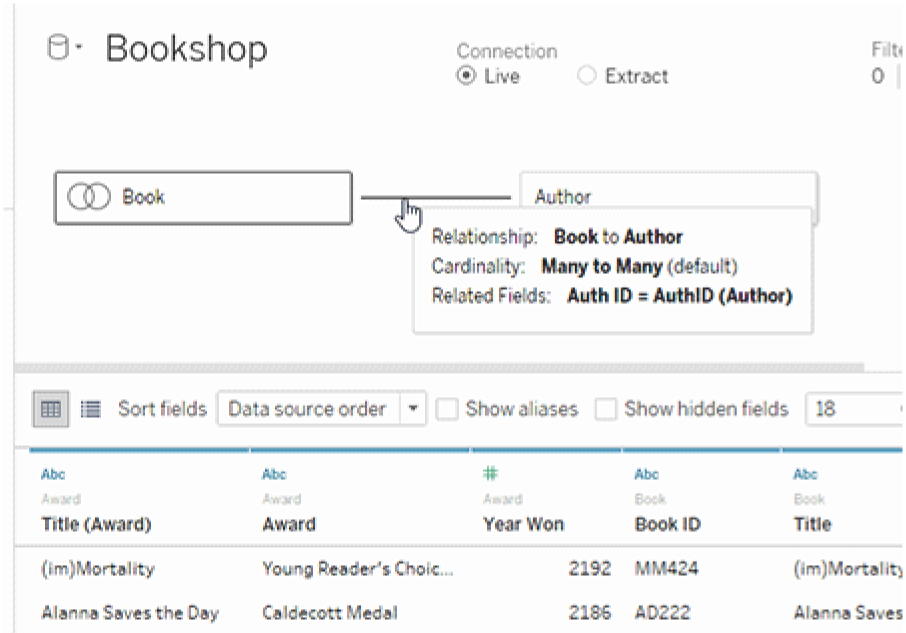
Remove a table from a relationship

To move a table, hover over a table, click the arrow, and then select **Remove**.



View a relationship

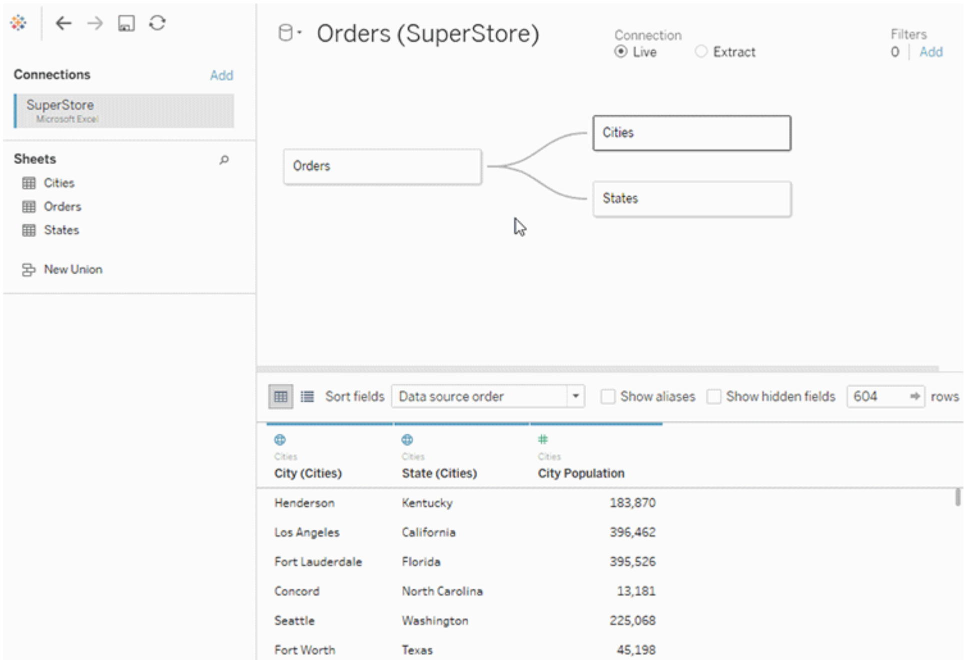
- Hover over the relationship line (noodle) to see the matching fields that define it. You can also hover over any logical table to see what it contains.



Edit a relationship

- Click a relationship line to open the **Edit Relationship** dialog box. You can add, change, or remove the fields used to define the relationship. Add additional field pairs to create a compound relationship.

  To add multiple field pairs: After you select the first pair, click **Close**, and then click **Add more fields**.



Tips on creating relationships

- The first table that you drag to the canvas becomes the root table for the data model in your data source. After you drag out the root table, you can drag out additional tables in any order. You will need to consider which tables should be related to each other, and the matching field pairs that you define for each relationship.
- Before you start creating relationships, viewing the data from the data source before or during analysis can be useful to give you a sense of the scope of each table. For more information, see **View Underlying Data** on page 2728. You can also use View Data to see a table's underlying data when a relationship is invalid.
- If you are creating a star schema, it can be helpful to drag the fact table out first, and then relate dimension tables to that table.
- Each relationship must be made of at least one matched pair of fields. Add multiple field pairs to create a compound relationship. Matched pairs must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- Relationships can be based on calculated fields. You can also specify how fields should be compared by using operators when you define the relationship.

- Deleting a table in the canvas automatically deletes its related descendants as well. If you delete the root table, all other tables in the model are also removed.

Validate relationships in your data source

You have several options for validating your data model for analysis. As you create the model for your data source, we recommend going to the sheet, selecting that data source, and then building a viz to explore record counts, unmatched values, nulls, or repeated measure values. Try working with fields across different tables to ensure everything looks how you expect it to.

What to look for:

- Are your relationships in the data model using the correct matching fields for their tables?
- What are the results of dragging different dimensions and measures into the view?
- Are you seeing the expected number of rows?
- Would compound relationships make the relationship more accurate?
- If you changed any of the Performance Options settings from the default settings, are the values that you are seeing in the viz what you would expect? If they aren't, you might want to check the settings, or reset to the default.

Options for validating relationships and the data model:

- Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. To see the count for a table, drag its Count field into the view. To see the count for all tables, select the Count field for each table in the Data pane, and then click the Text Table in Show Me.
- Click **View Data** in the Data pane to see the number of rows and data per table. Also, before you start creating relationships, viewing the data from the data source before or during analysis can be useful to give you a sense of the scope of each table. For more information, see **View Underlying Data** on page 2728.
- Drag dimensions onto rows to see the Number of Rows in the status bar. To see unmatched values, click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. You can also drag different measures to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

Tip: If you would like to see the queries that are being generated for relationships, you can use the Performance Recorder in Tableau Desktop.

1. Click the Help menu, and then select **Settings and Performance** > **Start Performance Recording**.
2. Drag fields into the view to build your viz.
3. Click the Help menu, and then select **Settings and Performance** > **Stop Performance Recording**.
4. In the Performance Summary dashboard, under Events Sorted By Time, click an "Executing Query" bar and view the query below.

Another more advanced option is to use the Tableau Log Viewer on GitHub. You can filter on a specific keyword using `end-protocol.query`. For more information, start with the Tableau Log Viewer wiki page in GitHub.

Dimension-only visualizations

When using a multi-table data source with related tables: If you build a dimension-only viz, Tableau uses inner joins and you won't see the full unmatched domain.

To see partial combinations of dimension values, you have can:

- Use Show Empty Rows/Columns to see all of the possible rows. Click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**.
- Add a measure to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

For more information, see **How Analysis Works for Multi-table Data Sources that Use Relationships** on page 841 and **Troubleshooting multi-table analysis** on page 856.

Relationships (logical tables) versus joins (physical tables)

While similar, joins and relationships behave differently in Tableau, and are defined in different layers of the data model. You create relationships between logical tables at the top-level, logical layer of your data source. You create joins between physical tables in the physical layer of your data source.

Joins merge data from two tables into a single table before your analysis begins. Merging the tables together can cause data to be duplicated or filtered from one or both tables; it can also cause NULL rows to be added to your data if you use a left, right, or full outer join. When doing analysis over joined data, you need to make sure that you correctly handle the effects of the join on your data.

Note: When duplication or the filtering effects of a join might be desirable, use joins to merge tables together instead of relationships. Double-click a logical table to open the physical layer and add joined tables.

A relationship describes how two independent tables relate to each other but does not merge the tables together. This avoids the data duplication and filtering issues that might occur in a join and can make working with your data easier.

relationships	joins
Defined between logical tables in the Relationship canvas (logical layer)	Defined between physical tables in the Join/Union canvas (physical layer)
Don't require you to define a join type	Require join planning and join type
Act like containers for tables that are joined or unioned	Are merged into their logical table
Only data relevant to the viz is queried. Cardinality and referential integrity settings can be	Run as part of every query

relationships	joins
adjusted to optimize queries.	
Level of detail is at the aggregate for the viz	Level of detail is at the row level for the single table
Join types are automatically formed by Tableau based on the context of analysis. Tableau determines the necessary joins based on the measures and dimensions in the viz.	Join types are static and fixed in the data source, regardless of analytical context. Joins and unions are established prior to analysis and don’t change.
Rows are not duplicated	Merged table data can result in duplication
Unmatched records are included in aggregates,	Unmatched records are omitted from the

relationships	joins
unless explicitly excluded	merged data
Create independent domains at multiple levels of detail	Support scenarios that require a single table of data, such as extract filters and aggregation

Relationships versus blends

While both relationships and blends support analysis at different levels of detail, they have distinct differences. One reason you might use blends over relationships is to combine published data sources for your analysis.

relationships	blends
Defined in the data source	Defined in the worksheet between a primary and a secondary data source
Can be published	Can't be published
All tables are equal	Depend on

relationships	blends
semantically	selection of primary and secondary data sources, and how those data sources are structured.
Support full outer joins	Only support left joins
Computed locally	Computed as part of the SQL query
Related fields are fixed	Related fields vary by sheet (can be customized on a sheet-by-sheet basis)

Features of different options for combining data: Relationships, joins, and blends

There are many ways to combine data tables, each with their own preferred scenarios and nuances.

Relate	Use when combining data

	from different levels of detail. • Requires matching fields between two logical tables. Multiple matching field pairs can define the relationship. • Automatically uses correct aggregations and contextual joins based on how fields are related and used in the viz. • Supports many-to-many and outer joins. • Relationships are consistent for the entire workbook and can be published. • Can be published, but you can't relate published data source, or edit relationships in published data sources. • Can't define relationships based on calculated fields or geographic fields. • Using data source filters limits join culling benefits of relationships.
Join	Use when you want to add more columns of data across the same row structure. • Requires common

	fields between two physical tables. • Requires establishing a join clause and a join type. • Can join on a calculation. • Joined physical tables are merged into a single logical table with a fixed combination of data. • May cause data loss if fields or values are not present in all tables (dependent on join types used). • May cause data duplication if fields are at different levels of detail. • Can use data source filters.
Union	Use when you want to add more rows of data with the same column structure. • Based on matching columns between two tables. • Unioned physical tables are merged into a single logical table with a fixed combination of data.
Blend	Use when combining data from different levels of detail.

- Can be used to combine published data sources, but can't be published.
- Can be used between a relational data source and a cube data source.
- Data sources can be blended on a per-sheet basis.
- Are always effectively left joins (may lose data from secondary data sources).

Use Relationships for Multi-table Data Analysis

Tables that you drag into this canvas use relationships. Relationships are a flexible way to combine data for multi-table analysis in Tableau.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to. Learn more about the basics of creating relationships in this 5-minute video.

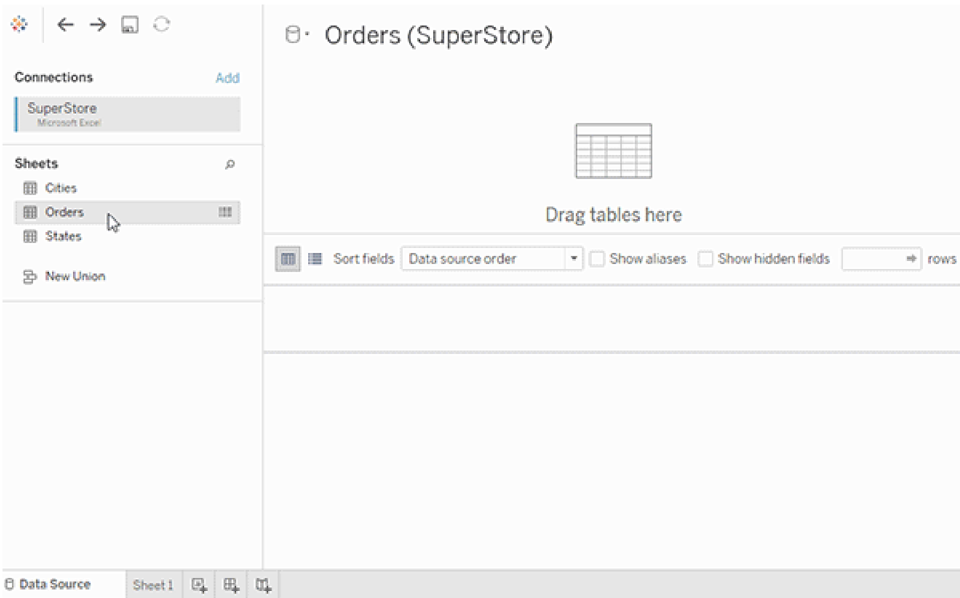
Learn more about how relationships work in these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

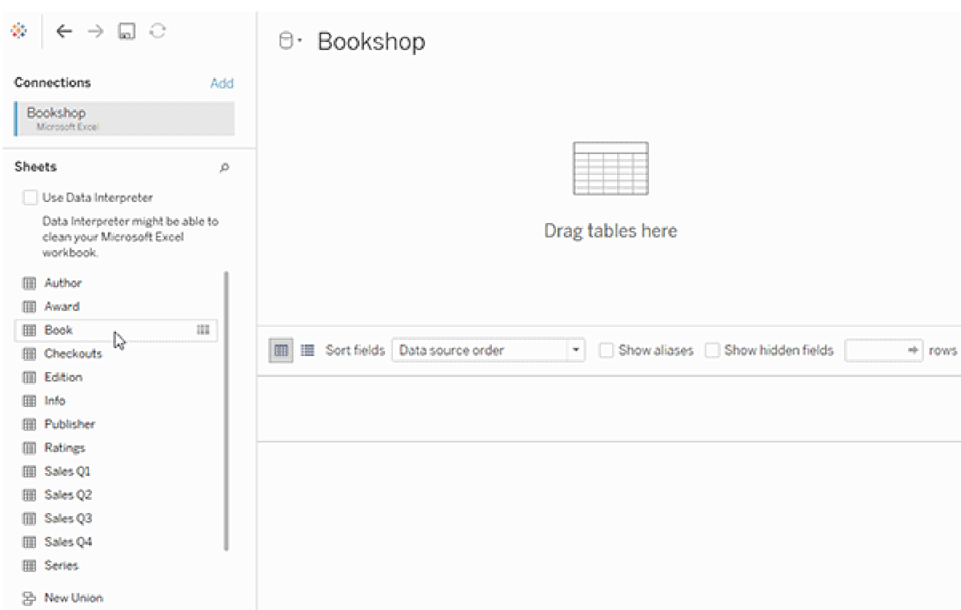
Are you building a new data source and workbook?

Drag a table to the Data Source page canvas to start building your data source.

A data source can be made of a single table that contains all of the dimension and measure fields you need for analysis...



Or, you can create a multi-table data source by dragging out more tables and defining their relationships...



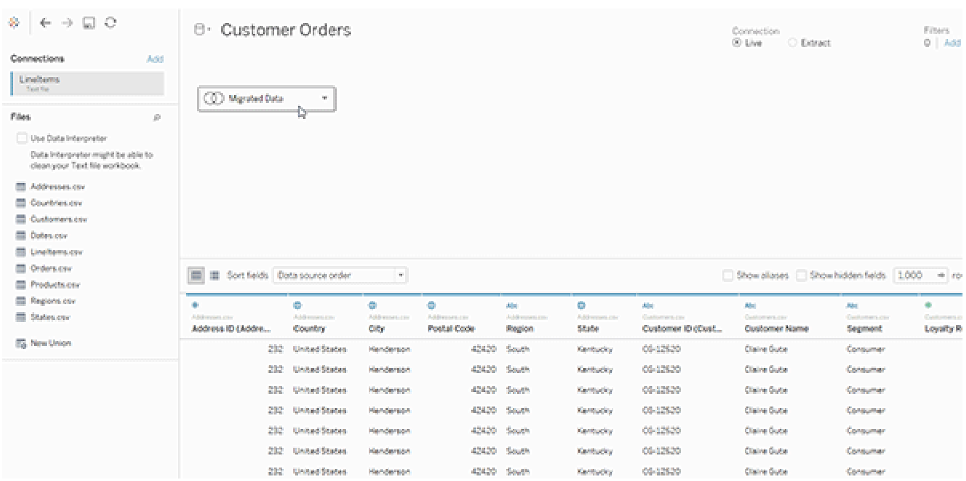
Watch this 1-minute video about getting started with using relationships.

For more information about using relationships, see Relate Your Data, **How Relationships Differ from Joins** on page 821, **The Tableau Data Model** on page 810, and Create and define relationships.

For more information on changes to data sources and analysis in Tableau 2020.2, see What's Changed with Data Sources and Analysis in 2020.2 and Questions about Relationships, the Data Model, and Data Sources in 2020.2.

Are you opening an older workbook or data source?

When you open a pre-2020.2 workbook or data source in 2020.2, your data source will appear as a single logical table in the canvas, with the name "Migrated Data" or the original table name. Your data is preserved and you can continue to use the workbook as you did before.



To see the physical tables that make up the single logical table, double-click that logical table to open it in the physical layer. You will see its underlying physical tables, including joins and unions.

For more information on changes to data sources and analysis in Tableau 2020.2, see What's Changed with Data Sources and Analysis in 2020.2 and Questions about Relationships, the Data Model, and Data Sources in 2020.2.

What's Changed with Data Sources and Analysis

Starting with Tableau 2020.2, we've made several major enhancements to make multi-table analysis easier and more intuitive. This topic describes these changes.

New data modeling capabilities in Tableau help you more easily analyze data across multiple tables. With a new logical layer and relationships in data sources, you can create multi-table data models in the form of star and snowflake schemas with multiple fact tables at different levels of detail (LOD).

- **Data sources are easier to build, maintain, and analyze**. Data sources now have a new logical layer where you can create flexible relationships between tables. Create multi-table, multi-fact data models by relating tables at different levels of detail. Bring in data from multiple tables more easily and maintain fewer data sources to meet your analytical needs.
- **Relationships make the analysis experience more intuitive**. You no longer need to engage in extensive join planning and make assumptions about what join types will be required to make your data ready for analysis. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data. You can see aggregations at the level of detail of the fields in your viz rather than having to think about the underlying joins. Relationships can be many-to-many and support full outer joins. You don't need to use LOD expressions such as FIXED to deduplicate data in related tables.
- **The Data Source page, View Data window, and Data pane have been updated to support a multi-table analysis experience**. Your first view of the Data Source page canvas is now the logical layer, where you can define relationships between tables. In support of multi-table analysis, several parts of the Tableau interface have changed. The Data Source page (canvas, data grid), View Data window, and the Data pane in the worksheet have all been updated to support a multi-table analysis experience. For more information, see Changes to different parts of the interface. The API to access View Data has also been updated to support multi-table analysis.
- **Context-aware queries bring in relevant data when it’s needed**. A data source that uses relationships makes it easier to bring more tables, more rows of data, and multiple fact tables into a single data source. Rather than querying the entire data source, Tableau brings in data from the tables that are needed for the worksheet, based on the fields at play in the visualization.

Watch a video: For an overview of data source enhancements and an introduction to using relationships in Tableau, see this 5-minute video.

Learn more about how relationship queries work in these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Changes to data sources, the data model, and query semantics

Support for multi-table data sources

Data sources in Tableau recognize and preserve normalized data. With multi-table data sources, Tableau can handle multiple levels of detail in a single data source. This support makes it easier for you to prepare and explore your data without having to write specialized calculations to control aggregations like averages and totals. Combine tables in star and snowflake schemas to seamlessly relate multiple fact tables. The new logical layer uses contextual, dynamic joins and gives you better domain control (see **Supported data model schemas** on page 817).

Tableau recognizes and maintains the underlying relationships between tables, so fewer joins and calculations should be needed. Because these enhanced data sources can answer a wider variety of questions, you can consolidate the number of distinct data sources you need to create and maintain. Fewer data sources are needed to represent the same data.

What didn't change?

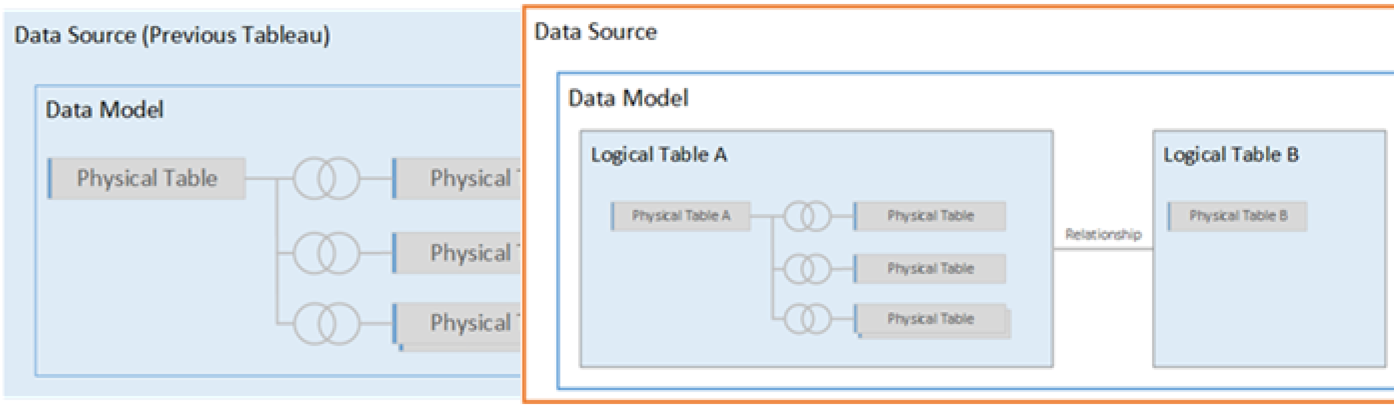
Relationships in the logical layer give you more options for combining data above in previous versions (nothing has been taken away). You can still create single-table data sources in Tableau. You can build a logical table using a combination of joins, unions, custom SQL, and so on. The behavior of single-table analysis in Tableau has not changed. Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2.

Also see **Questions about Relationships, the Data Model, and Data Sources** on page 798.

New logical layer in the data source

In previous versions of Tableau, the data model in your data source consisted of a single, physical layer where you could specify joins and unions. Tables added to the physical layer (joined or unioned) create a single, flattened table (denormalized) for analysis.

Previous versions	2020.2 and later



In versions of Tableau before 2020.2, the data model has only the physical layer

In 2020.2 and later, the data model has two layers: the logical layer and the physical layer

In Tableau 2020.2, the data model in your data source includes a new semantic layer above the physical layer—called the logical layer—where you can add multiple tables and relate them to each other based on matched field pairs. Tables at the logical layer are not merged in the data source, they remain distinct (normalized), and maintain their native level of detail.

Logical tables act like containers for physical tables. You can still use joins and unions between physical tables. Just double-click a logical table to open it and work with joins or unions.

The logical layer makes it easier for you to combine your data for a variety of analytics scenarios. You can now use and create normalized data models in the form of star and snowflake schemas, and flexible models with fact tables at different grains.

For more information, see **The Tableau Data Model** on page 810 and **Supported data model schemas** on page 817.

New query semantics based on relationships

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. Relationships can be many-to-many and support full outer joins. You don't need to use LOD expressions such as FIXED to deduplicate data in related tables.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to.

- **No up-front join type**. You only need to select matching fields to define a relationship (no join types). Because you don't specify the join type, you won't see a Venn diagram when you are setting up relationships.
- **Automatic and context-aware**. Relationships defer joins to the time and context of analysis. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data. You can see aggregations at the level of detail of the fields in your viz rather than having to think about the underlying joins.
- **Flexible**. When you combine tables using relationships, it’s like creating a custom, flexible data source for every viz, all in a single data source for the workbook. Because Tableau queries only tables that are needed based on fields and filters in a viz, you can build a data source that can be used for a variety of analytic flows.

For more information, see Relate Your Data, Don’t Be Scared of Relationships, and Relationships, part 1: Introducing new data modeling in Tableau.

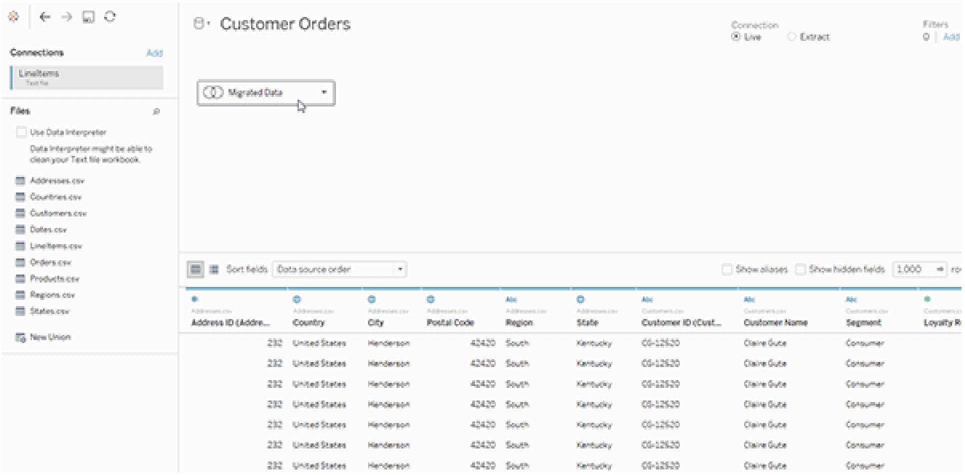
> **Note**: To go to the original join canvas, double-click a logical table in the top level view (the logical layer of the data source). You still can add joins and unions between tables in the physical layer of the data source. For more information, see **The Tableau Data Model** on page 810.

Migrated data sources

When you open a pre-2020.2 workbook or data source in a 2020.2 version of Tableau, your data source will appear as a single logical table in the canvas. You can continue to use the workbook as you did before.

If your data source contained multiple joined or unioned tables, the single logical table is displayed with the name "Migrated Data". You can rename the Migrated Data table.

To see the physical tables that make up the single logical table, double-click that logical table to open it in the physical layer. You will see its underlying physical tables, including joins and unions.



Web Authoring

Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page in web authoring in Tableau Server or Tableau Online. You cannot edit the data model of a published data source.

Extracts

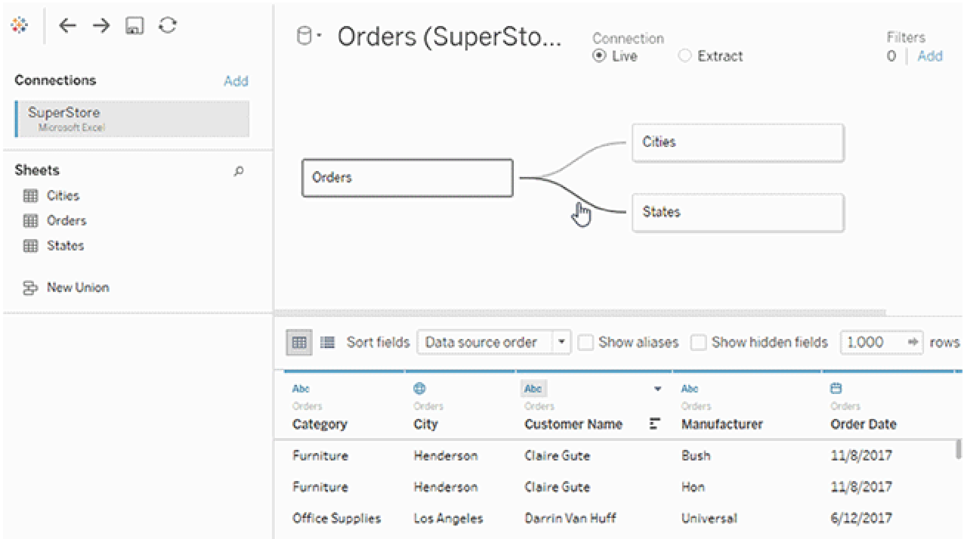
Extract data is now stored based on logical tables (replaces the Single Table option) or physical tables (replaces the Multiple Tables option). For more information, see **Create an extract** on page 1022.

Changes to different parts of the interface

Data Source page changes

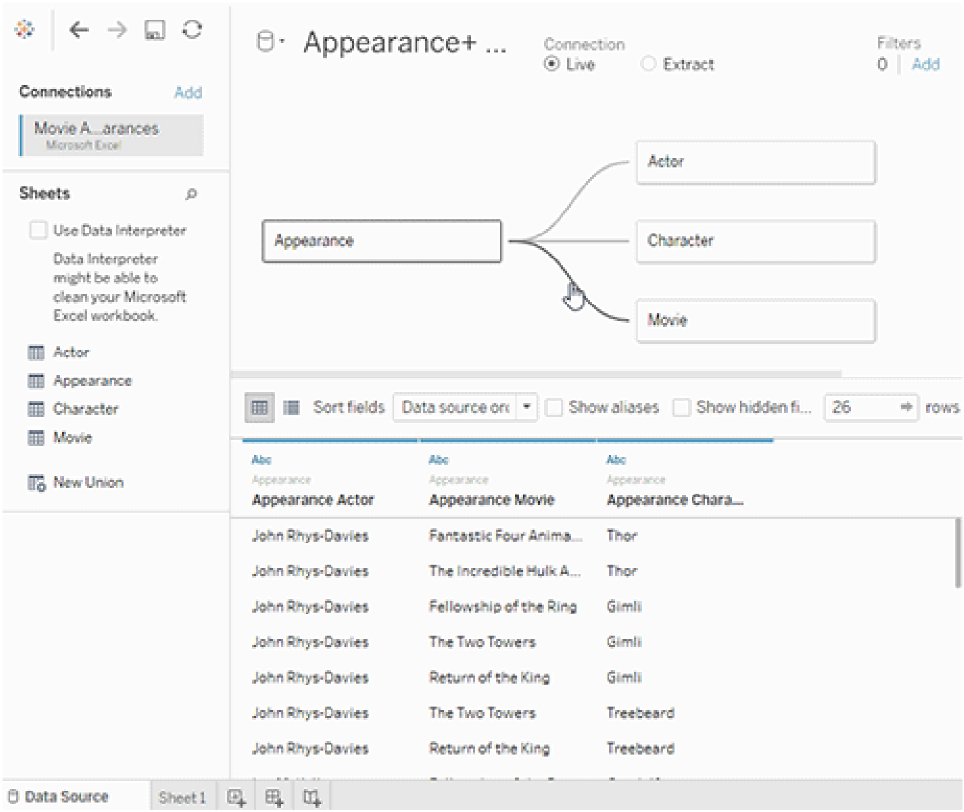
The Data Source page canvas now has two layers: a logical layer and a physical layer. The default view that you see in the Data Source page is new; it now shows the logical layer, which uses relationships ("noodles") between tables.

You still can add joins and unions between tables in the physical layer, just double-click a logical table in the logical, top-level view to switch to the physical layer. For more information, see **The Tableau Data Model** on page 810.



Data Grid

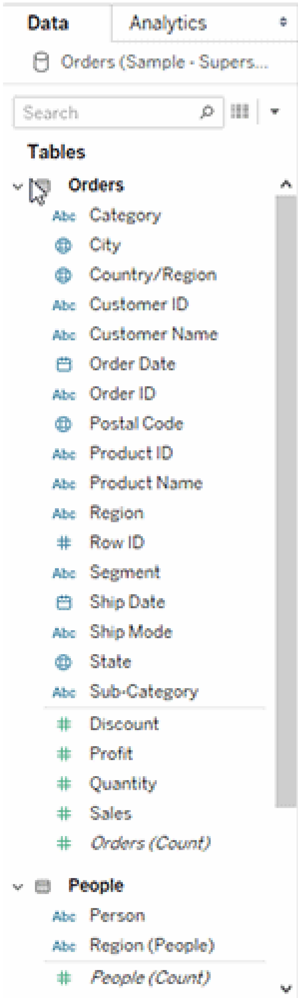
When the data source is made of multiple, related tables, selecting a logical table displays that table's data in the data grid below. With relationships, the data has not been merged, so you are seeing the values for the currently selected logical table. If you need to pivot columns, you must open a logical table and pivot the data in the physical table view of the data source.



Data pane in worksheets

To support relationships and multi-table analysis, the Data pane has been updated.

You will no longer see the Dimensions and Measures labels in the Data pane. Dimension and measure fields can be organized by table (**Group by Data Source Table**) or folder (**Group by Folder**). Dimensions are displayed above the gray line, and measures below the gray line for each table or folder. In some cases, a table might show only dimensions, or only measures to start with.



You will no longer see the Number of Records field in data sources that contain logical tables. Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. In the previous example you can see *Addresses(Count)*, *Customers (Count)*, and *LineItems(Count)*.

COUNT of table = SUM of Number of Records per table. You can't build calculations on top of a table's *Count* field. *Count* is aggregate-only.

> **Note**: You might see the *Number of Records* field in the Data pane if you open a pre-2020.2 workbook that used *Number of Records* in a view.

Calculated fields are listed with their originating field, if all of their input fields come from the same table.

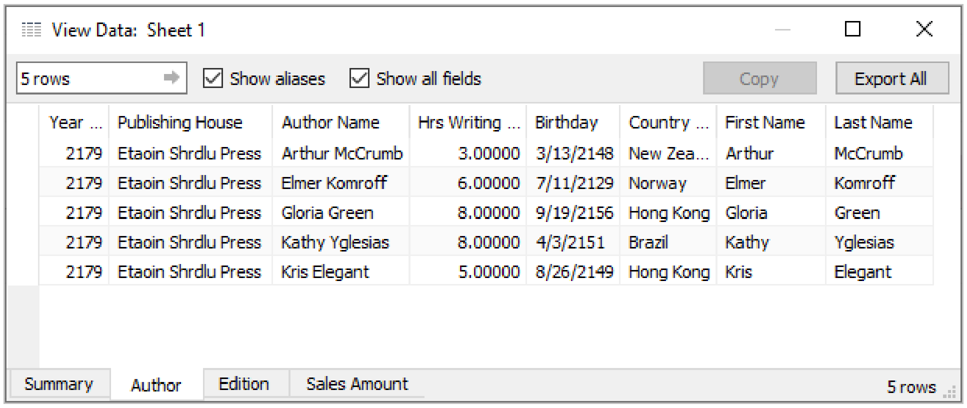
Sets and groups are displayed under the table with their originating field.

Fields that don't belong to a specific table or aren't grouped under a folder are displayed in the general area below the tables. These include: aggregated calculations, calculations that use fields from multiple tables, Number of Records when applicable, Measure Names, and Measure Values.

View Data

In the View Data window, when the data source contains multiple logical tables, the number of rows and data are listed at the native level of detail for each table in separate tabs.

For more information, see **View Underlying Data** on page 2728.



Year ...	Publishing House	Author Name	Hrs Writing ...	Birthday	Country ...	First Name	Last Name
2179	Etaoin Shrdlu Press	Arthur McCrumb	3.00000	3/13/2148	New Zea...	Arthur	McCrumb
2179	Etaoin Shrdlu Press	Elmer Komroff	6.00000	7/11/2129	Norway	Elmer	Komroff
2179	Etaoin Shrdlu Press	Gloria Green	8.00000	9/19/2156	Hong Kong	Gloria	Green
2179	Etaoin Shrdlu Press	Kathy Yglesias	8.00000	4/3/2151	Brazil	Kathy	Yglesias
2179	Etaoin Shrdlu Press	Kris Elegant	5.00000	8/26/2149	Hong Kong	Kris	Elegant

Calculated fields are listed under the table with their originating field.

LOD calculations won't necessarily appear in View Data for a logical table because an LOD calculation might be at a different granularity than the table itself. If the dimensionality of an LOD calc includes fields from a single table, that LOD calculation will appear in its owning table in the Data pane.

View Data column order changes in Tableau 2020.2 and later

When you open a data source from a previous version of Tableau in Tableau 2020.2 or later, the column order may be different. Columns may be displayed differently in the View Data window, and the column order may be different when you export it to CSV format.

This change in how column order is handled in the View Data window is required to support relationships and logical tables.

If you use scripts that depend on a customized column order, we recommend using the Tableau Extensions API for View Data to download underlying data to CSV format.

Changes to analysis with multi-table data sources

Using a data source that has multiple, related tables affects how analysis works in Tableau. Because multiple, related tables have independent domains and retain their native level of detail, when you drag fields into the view:

- Data is queried at its natural level of detail.
- Only data that is relevant to a viz is queried.
- The relationships between the tables affect the results of the query. The flow of building a viz can vary depending on how tables of fields are related to each other in the data model, or if they aren't related directly.

Benefits of using muti-table data sources with relationships:

- Relationships use joins, but they are automatic. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data.
- Tableau uses relationships to generate correct aggregations and appropriate joins during analysis, based on the current context of the fields in use in a worksheet.
- Multiple tables at different levels of detail are supported in a single data source, so fewer data sources are needed to represent the same data.
- Unmatched measure values are not dropped (no accidental loss of data).
- Avoids data duplication and filtering issues that can sometimes result from joins.

For more information, see **Relate Your Data** on page 766, **How Analysis Works for Multi-table Data Sources that Use Relationships** on page 841 and **Troubleshooting multi-table analysis** on page 856.

For related information about how relationship queries work, see these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks

What didn't change?

The behavior of single-table analysis in Tableau has not changed. Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2.

Dimension-only visualizations

When using a multi-table data source with related tables: If you build a dimension-only viz, Tableau uses inner joins and you won't see the full unmatched domain.

To see partial combinations of dimension values, you can:

- Use Show Empty Rows/Columns to see all of the possible rows. Click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. Be aware that this setting will also trigger densification for Date and Numeric Bin fields, which may be unwanted.
- Add a measure to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

Unmatched values behavior for dimensions that might surprise you

When you build a visualization that uses dimensions from multiple logical tables, the behavior can be initially surprising. By default, if the dimensions in a visualization or the inputs to a row-level calculations use fields from multiple logical tables, Tableau will only display results where all the dimensions have matching values.

For example, consider a data set that contains two logical tables, Customers and Purchases, where some Customers have made no Purchases and have no matching rows in the Purchases table. If you bring the dimension [Customers].[Customer Name] into play, you will see a list of all customers regardless of whether or not they made a Purchase. If you then add the dimension [Purchases].[Product] to the visualization, Tableau will change to just show the Customers who made purchases along with the Products they bought.

This is the default behavior. Here are several ways you can work with this behavior:

- Use Show Empty Rows/Columns to see all of the possible rows and bring back the unmatched dimension values. Click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. Be aware that this setting will also trigger densification for Date and Numeric Bin fields, which may be unwanted.
- Convert one of the dimensions to a measure. For example, switching to use COUNT ([Purchases].[Product]) will keep Customers with no purchases and display a "0" for them. This may not be a viable approach if you want to see all the Products.
- Bring a measure into the view from the table with unmatched values. For example, adding COUNT([Customers]) on the detail shelf will bring back the unmatched customers, while still showing all the Products.
- Instead of relating the two tables at the logical level, you can join them using an outer join at the physical level, as in previous versions of Tableau. This approach will show the unmatched values, with a few side effects. Queries will always be issued querying both tables, NULL values will appear in visualizations and quick filters even when no fields

from the unmatched table are used, and you will have to write LOD calculations to remove any unwanted duplication introduced by the join.

Using a table's Count field instead of Number of Records

You will no longer see the Number of Records field in data sources that contain logical tables. Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. In the previous example you can see *Addresses(Count)*, *Customers (Count)*, and *LineItems(Count)*.

COUNT of table = SUM of Number of Records per table. You can't build calculations on top of a table's *Count* field. *Count* is aggregate-only.

> **Note**: You might see the *Number of Records* field in the Data pane if you open a pre-2020.2 workbook that used *Number of Records* in a view.

Cross-table calculations

You can write calculations that span multiple tables. Be aware that cross-table calculations use inner join query semantics by default. The calculation lives on top of the join between the logical tables that it references. Also, it is now possible to write row-level calculations across tables that result in outer joins.

Calculations are evaluated after Null dimensions are added, so you can change a Null to something else using a calculation like: IFNULL([Dimension], “Missing”).

Grand totals using SUM

Grand totals using SUM reflect the actual total at the native level of detail of a field's logical table, rather than the SUM of subtotals.

Nulls and automatic joins that result from relationships

In Tableau 2020.2, when performing joins that result from relationships, Tableau does not join nulls against nulls.

Unmatched null behavior

When you add a measure to the view, a Null dimension value is added when the measure has values that are not associated with a dimension.

This behavior occurs because Tableau treats unmatched values (or non-associated dimension values) the same as true NULL values in the underlying data source. Values for both will be aggregated together or filtered together.

For example, with `SUM(Sales),[State]` every matched State represents a state. Its SUM (Sales) is the sum for a single state. For unmatched nulls, SUM(Sales) may reflect many states.

If you don't want this behavior, you can clean your data to make sure that every row in the measure table matches a row in the dimension table. You can use Tableau Prep for this. Or, within a sheet, you can filter out the Null by clicking on the Null and selecting **Exclude**. After you do this, you can edit the relationship between the measure table and the dimensions table by selecting **All Records Match** in the Performance Options settings.

Dealing with unmatched nulls in the Edit Filter dialog box

When you apply a filter to a dimension, the rules for when the Null appears are:

- On the General tab, you can explicitly include or exclude the Null row using the check box.
- On the Wildcard tab, the setting won’t impact whether or not Null appears (since it only filters out text values and Null is not treated as a text value).
- On the Condition or Top tabs, using a setting other than **None** will cause the “Null” state to be filtered out.

Validating relationships

You have several options for validating your data model for analysis. As you create the model for your data source, we recommend going to a sheet, selecting that data source, and then building a viz to explore record counts, expected data, unmatched values, nulls, or repeated measure values. Try working with fields across different tables to ensure everything looks how you expect it to.

What to look for:

- Are your relationships in the data model using the correct matching fields for their tables?
- Would adding multiple matching field pairs make the relationship more accurate?
- What are the results of dragging different dimensions and measures into the view from different tables?
- Are you seeing the expected number of rows?

- If you changed any of the Performance Options settings from the default settings, are the values that you are seeing in the viz what you would expect? If they aren't, you might want to check the settings, or reset to the default.

Options for validating relationships and the data model:

- Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. To see the count for a table, drag its Count field into the view. To see the count for all tables, select the Count field for each table in the Data pane, and then click the Text Table in Show Me.
- Click **View Data** in the Data pane to see the number of rows and data per table. Also, before you start creating relationships, viewing the data from the data source before or during analysis can be useful to give you a sense of the scope of each table. For more information, see **View Underlying Data** on page 2728.
- Drag dimensions onto rows to see the Number of Rows in the status bar. To see unmatched values, click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. You can also drag different measures to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

Tip: If you would like to see the queries that are being generated for relationships, you can use the Performance Recorder in Tableau Desktop.

1. Click the Help menu, and then select **Settings and Performance** > **Start Performance Recording**.
2. Drag fields into the view to build your viz.
3. Click the Help menu, and then select **Settings and Performance** > **Stop Performance Recording**. The Performance Recording workbook will automatically open.
4. In the Performance Summary dashboard, under Events Sorted By Time, click an "Executing Query" bar and view the query below.

Another more advanced option is to use the Tableau Log Viewer on GitHub. You can filter on a specific keyword using `end-protocol.query`. For more information, start with the Tableau Log Viewer wiki page in GitHub.

Questions about Relationships, the Data Model, and Data Sources

These are some of the questions we've heard related to data modeling and relationships in Tableau. If you have a question that isn't in this list, please share it with us by clicking the blue

Send feedback icon in the bottom right corner of this Help page, (click Yes or No, add your question in the Comment field, and then click Send).

Relationships

Is a relationship just a different name for a join?

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. A relationship describes how two independent, logical tables relate to each other, but does not merge the tables together. When a relationship is created between tables, the tables remain separate (normalized), maintaining their native level of detail and domains. You can use relationships to create multi-fact data models.

You can't set a join type for relationships. Relationships defer joins to the time and context of analysis. Tableau automatically selects what join types should be used based on the current fields in use in the viz. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data. You can see aggregations at the level of detail of the fields in your viz rather than having to think about the underlying joins.

Relationships can be many-to-many and support full outer joins. You don't need to use LOD expressions such as FIXED to deduplicate data in related tables.

Learn more: For related information on combining data using relationships, also see these topics and blog posts:

- Relate Your Data
- **How Relationships Differ from Joins** on page 821
- **Use Relationships for Multi-table Data Analysis** on page 782
- **Don't Be Scared of Relationships** on page 861
- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Can I use joins between logical tables?

You must use relationships between logical tables. You can only use joins between physical tables contained in a logical table. Double-click a logical table to open it.

We recommend using relationships as your first approach to combining your data because it makes your data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to. Situations where you might still use joins include when you want to:

- Continue to use a data source from a pre-2020.2 version to Tableau that you have upgraded to 2020.2
- Explicitly use a specific join type
- Use a data model that supports shared dimensions

Where did joins go? Can I still combine table data using joins?

Joins are still an option for combining your data. You just need to open a logical table to work with joins (double-click a logical table to open it). Tables that you join get merged into a single logical table.

Where did joins go? on page 824

Are relationships like blends? When should I use a blend?

While both relationships and blends support analysis at different levels of detail, they have distinct differences. If you want to combine data from published data sources, blends are currently your only option.

You might also choose to use blends when the fields used to join two tables are dependent on the worksheet.

Blends only support left joins, while relationships support full outer joins. Blends can be customized on a per-sheet basis.

Relationships versus blends on page 778

Data sources, data model, and connections

How have data sources changed?

In versions before 2020.2, tables that you combined in the Data Source page using joins and unions made up the data model in a data source. The data was merged into a single, flat table.

In Tableau 2020.2, a new logical layer has been added to Tableau's data model. The logical layer and relationships expand your options for bringing the data you need into Tableau. You can now create normalized data sources with multiple tables at different levels of detail. Related tables aren't merged; they remain distinct. Using relationships makes creating and analyzing data across multiple tables more flexible and more intuitive.

You can still create single-table data sources in Tableau. You can build a logical table using a combination of joins, unions, custom SQL, and so on. The behavior of single-table analysis in Tableau has not changed. Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2. When you upgrade a workbook or data source to 2020.2, will contain a single logical table that represents your pre-2020.2 data source. It will work the same as it did before 2020.2.

What are logical tables and physical tables?

Data sources in 2020.2 use a data model that has two layers: a logical layer where you can relate tables, and a physical layer where tables can be joined or unioned. Tables that you drag to the logical layer use relationships and are called logical tables. Every logical table can contain one more physical tables in the physical layer.

For more information, see **The Tableau Data Model** on page 810.

Can relationships exist between tables from different connections?

Yes. You can create a data source with relationships between tables from different connections. For more information, see **Combine Data** on page 745 in **Plan the Data Source** on page 741.

Can I use inequality operators or calculated fields to define relationships?

In Tableau 2020.3 and later, you can create relationships based on calculated fields, and compare fields used for relationships using operators in the relationship definition.

Note that the following connectors do not support inequality operators:

- Google BigQuery
- MapR
- Salesforce

Do all connection types support logical tables and relationships?

Most relational connection types are completely supported. Cubes, SAP HANA (with OLAP attribute), JSON, and Google Analytics are limited to a single logical table in 20.2. Stored procedures can only be used within a single logical table.

Published data sources can't be related to each other. You also can't edit published data sources.

Unsupported

- Cube databases do not support the new logical layer. Connecting to a cube offers the same experience as pre-2020.2.

- Stored Procedures: Don't support federation, relationships, or joins. They are represented in a single logical table, and don't allow opening the Join/Union canvas (physical layer).
- Splunk: Doesn't support left joins (and therefore relating logical tables).
- JSON: Doesn't support federation, custom SQL, joins, or relationships (only unions).
- Datasources that do not support LOD calcs. For more information, see **Data Source Constraints for Level of Detail Expressions** on page 2471.

Limited support

- Salesforce and WDC Standard Connections: These are represented as joined tables within a logical table. Adding them is currently only supported for single logical table data sources. Standard connections cannot join to an existing table.
- SAP HANA: Doesn't support relating logical tables when the connection has the OLAP attribute set.

What types of data models are supported?

See **The Tableau Data Model** on page 810 and **Supported data model schemas** on page 817

Is there a classic view of the Data Source page that I can use?

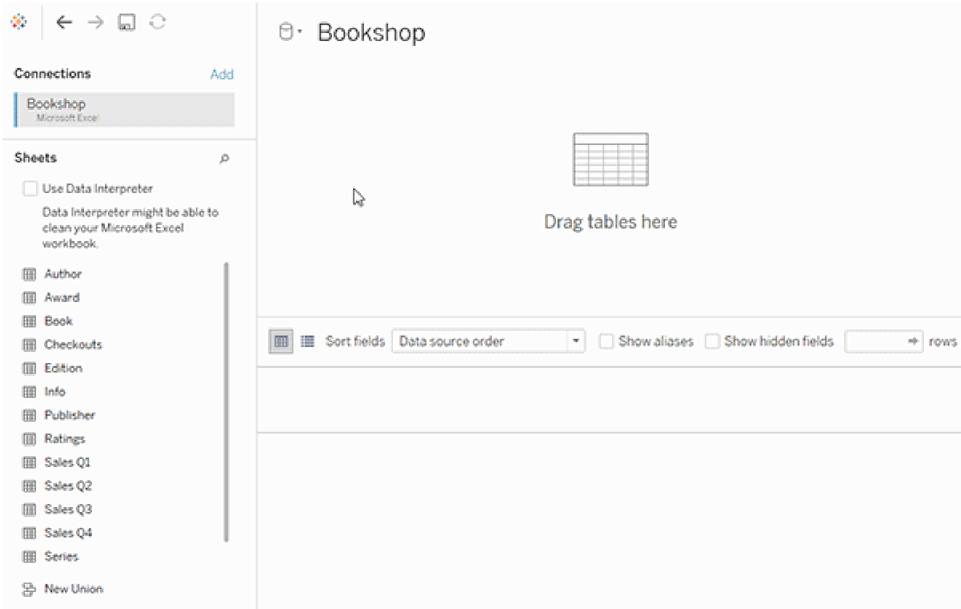
The physical layer of the Data Source page canvas is basically the "classic view" of the Data Source page in previous versions of Tableau.

The default view of Data Source canvas is the logical layer starting in 2020.2. You must double-click a logical table in the Data Source page canvas to go to the physical layer of the canvas.

How do I create a single-table data source?

Drag a single table into the canvas (logical layer) of the Data Source page. In the worksheet, use the fields from that table in the Data pane for analysis.

You can add more data inside the single, logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged into their single, logical table.



This example shows the Book table in the Relationships canvas (logical layer) of the data source. Double-clicking the Book logical table opens the Join/Union canvas (physical layer). The joins merge the Award and Info tables with the Book table.

Has row-level security changed?

Row-level security works the same. For information on row-level security, see Best Practices for Row Level Security in Tableau Server help.

How do I use custom SQL in the new data model?

Double-click the New Custom SQL option in the left pane of the Data Source page (as before). The custom SQL will be contained by a single logical table.

What happens to my older data sources when I open them in 2020.2 and later versions?

The data is migrated without changes and you can continue to use the workbook as you did before.

When you open a pre-2020.2 workbook or data source in a 2020.2 version of Tableau, your data source will appear as a single logical table in the canvas with the name "Migrated Data". You can rename the Migrated Data table.

Your original, denormalized data may have been originally built from one or more tables using joins and unions. When you open the data source in Tableau 2020.2, Tableau migrates the denormalized data model to a single logical table in the new data model to ensure that your data and workbooks are migrated without changes.

To see the physical tables that make up the single logical table, double-click that logical table to open it in the physical layer. You will see its underlying physical tables, including joins and unions.

Do I need to change my migrated data source to use relationships instead of joins?

If you have existing multi-table data sources defined using physical layer joins, they will still work as they did before. You don't have to change your data source. You can relate logical tables to your migrated table, but you can't downgrade logical tables.

You might consider changing you data source to use relationships instead of joins if you want to add more tables to your data model. This will require you to remove tables from the physical layer, and then add them to the logical layer.

How do I view or edit the data model on the web?

Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page.

You cannot edit the data model of a published data source on the web or in Tableau Desktop.

Can I use a data source on Tableau Server or Tableau Online (version 2020.2 or later) in a previous version of Tableau Desktop (version 2020.1 or earlier)?

If you attempt to open a published data source or published workbook from Tableau Server or Tableau Online version 2020.2 or later in a previous version of of Tableau Desktop, any logical tables that were related to the root table in the data source will be removed. Only the root table (the first table added to that data source model) will remain.

If you attempt to use a local data source in a previous version of Tableau:

- Tableau displays a warning that the data source is from a newer version of Tableau that isn't compatible with the previous version.
- On the Data Source page, only the root table will remain in the data source.
- In the Data pane in a Tableau Desktop worksheet, Tableau displays errors (red exclamation points) next to affected fields that are not part of the root table.

If you attempt to use a published data source in a previous version of Tableau:

- Tableau displays a warning that the data source is from a newer version of Tableau that isn't compatible with the previous version.

- In the Data pane in a Tableau Desktop worksheet, Tableau displays an error (red exclamation point) next to the selected data source and a message that the connection needs to be updated. Click **Show Details** for more information about the error.

Can published data sources be related to each other?

No. Also, you can't edit or view the data model for a published data source.

If you want to combine data from published data sources, blends are currently your only option.

Relationships versus blends on page 778

How do extracts work with related logical tables?

Extract data is now stored based on logical tables (replaces the Single Table option) or physical tables (replaces the Multiple Tables option). For more information, see **Create an extract** on page 1022.

If I need to downgrade from 2020.2 to a previous version, what happens to relationships between logical tables?

If you downgrade a 2020.2 workbook to a previous version of Tableau, any logical tables that were related to the root table in the model will be removed from the data source. Only the root table (the first table added to that model) will remain.

All sheets in the workbook used fields from logical tables (non-root) become invalid, because their tables and fields have been removed from the data model.

Downgrading works best for workbooks that contain single-table data sources.

How does the new data model work with cubes?

Cubes work the same as in previous versions of Tableau. A cube data source will appear as a single-table data source, just like it is today. You can't create relationships using cube data.

Interaction with other Tableau features and products

Does Explain Data work with multi-table data sources that use relationships?

In 2020.3, you can use Explain Data with data sources that contain multiple, related tables. Cardinality and Referential Integrity settings for relationships must be set up correctly for Explain Data to analyze multi-table, related data.

In 2020.2, you can use Explain Data with a single-table data sources only. Your data source can have a single, logical table that is defined by one or more physical tables.

Does Ask Data work with multi-table data sources?

Ask Data fully supports multi-table, normalized data sources.

How do new data modeling capabilities affect using Tableau Bridge?

You will need to update to the latest version of Tableau Bridge for full compatibility with 2020.2 data modeling functionality.

When should I use Tableau Prep vs. authoring in Tableau Desktop, Tableau Online, or Tableau Server to create a data source?

Tableau Prep cleans data, and creates flows, extracts, and published data sources that contain physical tables.

In Tableau Desktop, and in Tableau Online and Tableau Server web authoring, you can create data sources that use normalized data models. These data models can be made of logical tables and physical tables, and your data sources can be saved as live data sources or as extracts.

Only logical tables can be related. Physical tables can be joined and unioned.

Analysis with multi-table data sources

Does analysis work differently with multi-table data sources that use relationships?

Using a data source that has multiple, related tables affects how analysis works in Tableau. Because multiple, related tables have independent domains and retain their native level of detail, when you drag fields into the view:

- Data is queried at its natural level of detail.
- Only data that is relevant to a viz is queried.
- The relationships between the tables affect the results of the query. The flow of building a viz can vary depending on how tables of fields are related to each other in the data model, or if they aren't related directly.

For more information, see Don’t Be Scared of Relationships, **How Analysis Works for Multi-table Data Sources that Use Relationships** on page 841, **Unmatched values behavior for dimensions that might surprise you** on page 795, and **Troubleshooting multi-table analysis** on page 856.

Do LODs work the same with the new data model? When should I use an LOD?

LOD expressions and calculations work the same. Since Tableau now understands the level of detail (LOD) of your input tables, you shouldn't need to use LOD calculations to remove unwanted duplication due to joins.

You may still want to use LOD calcs to:

- Handle unwanted duplication in your source tables.
- Compute multi-level aggregations (e.g. an average over a sum)
- To do cohort analysis (e.g. to compute the first order date for each customer)

If the dimensionality of an LOD calculation includes fields from a single table, that LOD calculation will appear in its owning table in the Data pane.

How can I tell if I combined my data correctly with relationships?

You have several options for validating your data model for analysis. As you create the model for your data source, we recommend going to a sheet, selecting that data source, and then building a viz to explore record counts, expected data, unmatched values, nulls, or repeated measure values. Try working with fields across different tables to ensure everything looks how you expect it to.

What to look for:

- Are your relationships in the data model using the correct matching fields for their tables?
- Would adding multiple matching field pairs make the relationship more accurate?
- What are the results of dragging different dimensions and measures into the view?
- Are you seeing the expected number of rows?
- If you changed any of the Performance Options settings from the default settings, are the values that you are seeing in the viz what you would expect? If they aren't, you might want to check the settings, or reset to the default.

Options for validating relationships and the data model:

- Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. To see the count for a table, drag its Count field into the view. To see the count for all tables, select the Count field for each table in the Data pane, and then click the Text Table in Show Me.
- Click **View Data** in the Data pane to see the number of rows and data per table. Also, before you start creating relationships, viewing the data from the data source before or

during analysis can be useful to give you a sense of the scope of each table. For more information, see **View Underlying Data** on page 2728.

- Drag dimensions onto rows to see the Number of Rows in the status bar. To see unmatched values, click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. You can also drag different measures to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

Can I see the queries that Tableau is generating for relationships?

If you would like to see the queries that are being generated for relationships, you can use the Performance Recorder in Tableau Desktop.

1. Click the Help menu, and then select **Settings and Performance** > **Start Performance Recording**.
2. Drag fields into the view to build your viz.
3. Click the Help menu, and then select **Settings and Performance** > **Stop Performance Recording**. The Performance Recording workbook will automatically open.
4. In the Performance Summary dashboard, under Events Sorted By Time, click an "Executing Query" bar and view the query below.

Another more advanced option is to use the Tableau Log Viewer on GitHub. You can filter on a specific keyword using `end-protocol.query`. For more information, start with the Tableau Log Viewer wiki page in GitHub.

Changes to different parts of the interface

How has the Data Source page changed? The data grid? View Data? The Data pane?

The Data Source page (canvas, data grid), View Data, and Data pane have been updated to support a multi-table analysis experience.

Your first view of the data source canvas is now the logical layer, which is where you define relationships. The data grid shows row data for each table's level of detail.

In the Data pane, fields and calculated fields are automatically organized by their native level of detail.

The View Data window displays row level data at the correct level of detail, without replication, to make validation easier. Get row counts for each table in the Data pane using *Table(Count)* fields.

Data Source page changes on page 789

Data Grid on page 790

View Data on page 793

Data pane in worksheets on page 791

View Data column order changes in Tableau 2020.2 and later

When you open a data source from a previous version of Tableau in Tableau 2020.2 or later, the column order may be different. Columns may be displayed differently in the View Data window, and the column order may be different when you export it to CSV format.

This change in how column order is handled in the View Data window is required to support relationships and logical tables.

If you use scripts that depend on a customized column order, we recommend using the Tableau Extensions API for View Data to download underlying data to CSV format.

How does Count of Table work versus Number of Records?

You will no longer see the Number of Records field in data sources that contain logical tables. Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. In the previous example you can see *Addresses(Count)*, *Customers (Count)*, and *LineItems(Count)*.

COUNT of table = SUM of Number of Records per table. You can't build calculations on top of a table's *Count* field. *Count* is aggregate-only.

> **Note**: You might see the *Number of Records* field in the Data pane if you open a pre-2020.2 workbook that used *Number of Records* in a view.

Where are sets, groups, and calculated fields shown?

If the field belongs to a table, it is listed under the table. If it doesn't, it is listed in the general area at the bottom of the Data pane.

Calculated fields are listed with their originating field, if all of their input fields come from the same table.

Sets and groups are displayed under the table with their originating field.

Fields that don't belong to a specific table are displayed in the general area below the tables. These include: aggregated calculations, calculations that use fields from multiple tables, Measure Names, and Measure Values.

The Tableau Data Model

Every data source that you create in Tableau has a data model. You can think of a data model as a diagram that tells Tableau how it should query data in the connected database tables.

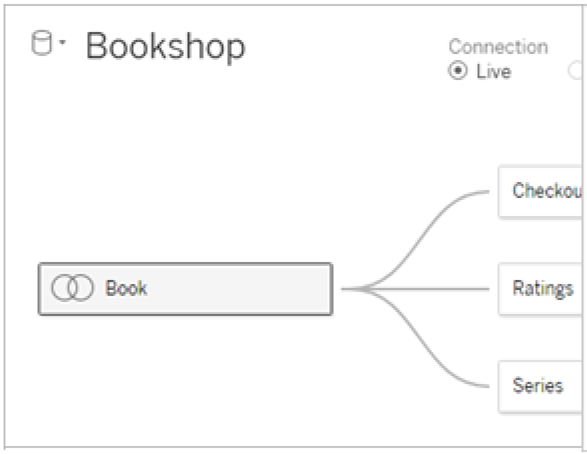
The tables that you add to the canvas in the **Data Source** page create the structure of the data model. A data model can be simple, such as a single table. Or it can be more complex, with multiple tables that use different combinations of relationships, joins, and unions.

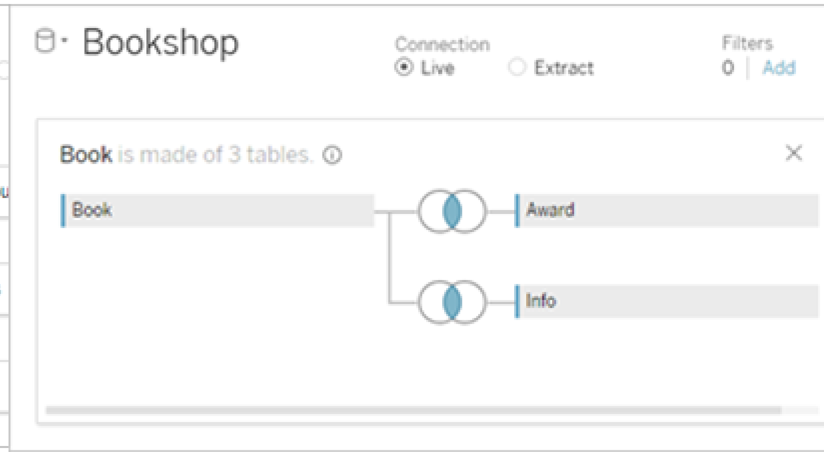
The data model has two layers:

- The default view that you first see in the Data Source page canvas is the *logical layer* of the data source. You combine data in the logical layer using relationships (or noodles). Think of this layer as the Relationships canvas in the Data Source page. For more information, see **Use Relationships for Multi-table Data Analysis** on page 782.
- The next layer is the *physical layer*. You combine data between tables at the physical layer using joins and unions. Each logical table contains at least one physical table in this layer. Think of the physical layer as the Join/Union canvas in the Data Source page. Double-click a logical table to view or add joins and unions.

Logical Layer	**Physical Layer**
Noodles = Relationships	Venn diagram = Joins

Logical Layer	**Physical Layer**





The top-level view of a data source with multiple, related tables. This is the logical layer. Logical tables can be combined using relationships (noodles). They don't use join types. They act like containers for physical tables.

Double-click a logical table to open it and see its physical tables. Physical tables can be combined using joins or unions. In this example, the Book logical table is made of three, joined physical tables (Book, Award, Info).

Logical Layer	Physical Layer
Relationships canvas in the Data Source page	Join/Union canvas in the Data Source page
Tables that you drag here are called logical tables	Tables that you drag here are called physical tables
Logical tables can be related to other logical tables	Physical tables can be joined or unioned to other physical tables
Logical tables are like containers for physical tables	Double-click a logical table to see its physical tables
Level of detail is at the row level of the logical table	Level of detail is at the row level of merged physical tables
Logical tables remain distinct (normalized), not merged in the data source	Physical tables are merged into a single, flat table that defines the logical table

Layers of the data model

The top-level view that you see of a data source is the logical layer of the data model. You can also think of it as the Relationships canvas, because you combine tables here using relationships instead of joins.

When you combine data from multiple tables, each table that you drag to the canvas in the logical layer must have a relationship to another table. You do not need to specify join types for relationships; during analysis Tableau automatically selects the appropriate join types based on the fields and context of analysis in the worksheet.

The physical layer of the data model is where you can combine data using joins and unions. You can only use pivots in this canvas. You can think of it as the Join/Union canvas. In previous versions of Tableau, the physical layer was the only layer in the data model. Each logical table can contain one or more physical tables.

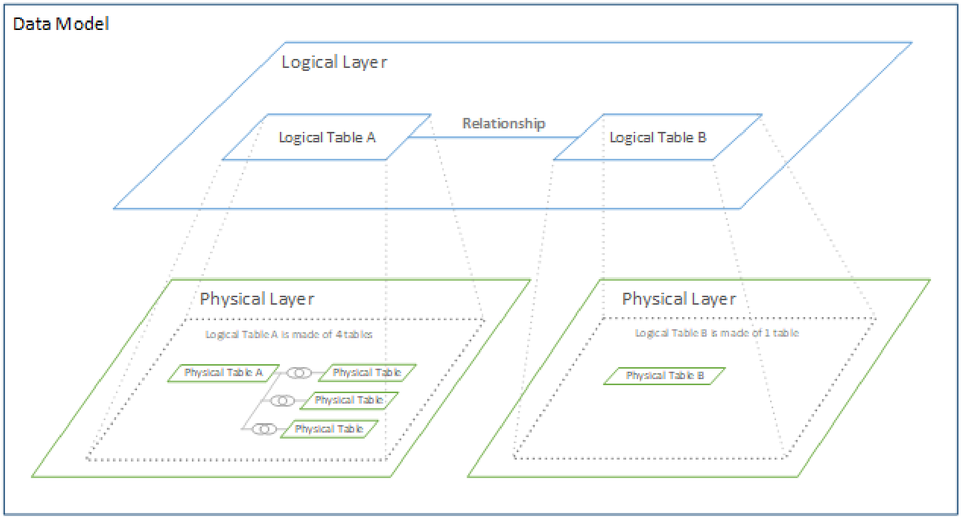
Important: You can still create single-table data sources in Tableau that use joins and unions. The behavior of single-table analysis in Tableau has not changed. Your upgraded workbooks will work the same as they did before 2020.2.

Learn more: For related information on combining data using relationships, also see these topics and blog posts:

- **How Relationships Differ from Joins** on page 821
- **Use Relationships for Multi-table Data Analysis** on page 782
- Relate Your Data
- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

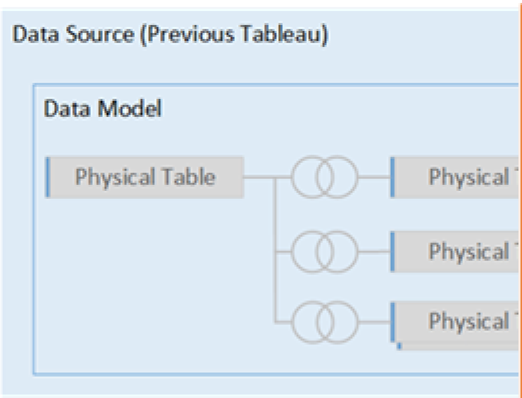
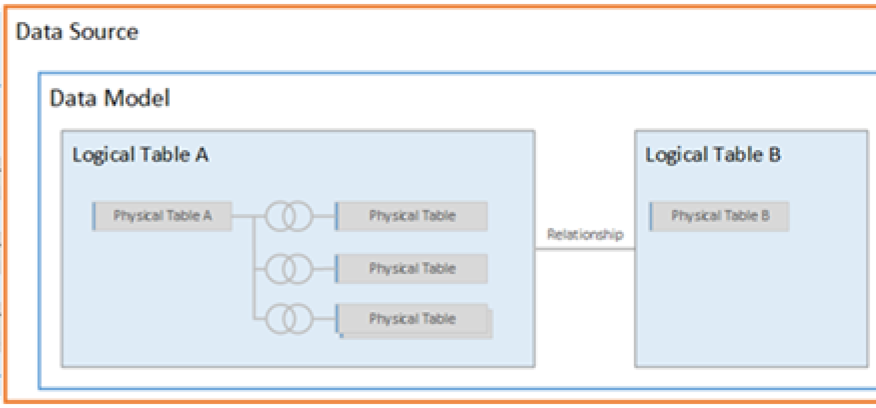
Understanding the data model

In previous versions of Tableau, the data model had only the physical layer. In Tableau 2020.2 and later, the data model has the logical (semantic) layer and a physical layer. This gives you more options for combining data using schemas to fit your analysis.



In Tableau 2020.2 and later, a logical layer has been added in the data source. Each logical table contains physical tables in a physical layer.

In previous versions of Tableau, the data model in your data source consisted of a single, physical layer where you could specify joins and unions. Tables added to the physical layer (joined or unioned) create a single, flattened table (denormalized) for analysis.

Previous versions	**2020.2 and later**
	
In versions of Tableau before 2020.2, the data model has only the physical layer	In 2020.2 and later, the data model has two layers: the logical layer and the physical layer

In Tableau 2020.2, the data model in your data source includes a new semantic layer above the physical layer—called the logical layer—where you can add multiple tables and relate them to each other. Tables at the logical layer are not merged in the data source, they remain distinct (normalized), and maintain their native level of detail.

Logical tables act like containers for merged physical tables. A logical table can contain a single, physical table. Or it can contain multiple physical tables merged together through joins or unions.

Build a new model

When you add one or more tables to the logical layer, you are essentially building the data model for your data source. A data source can be made of a single, logical table, or you can drag multiple tables to the canvas to create a more complex model.

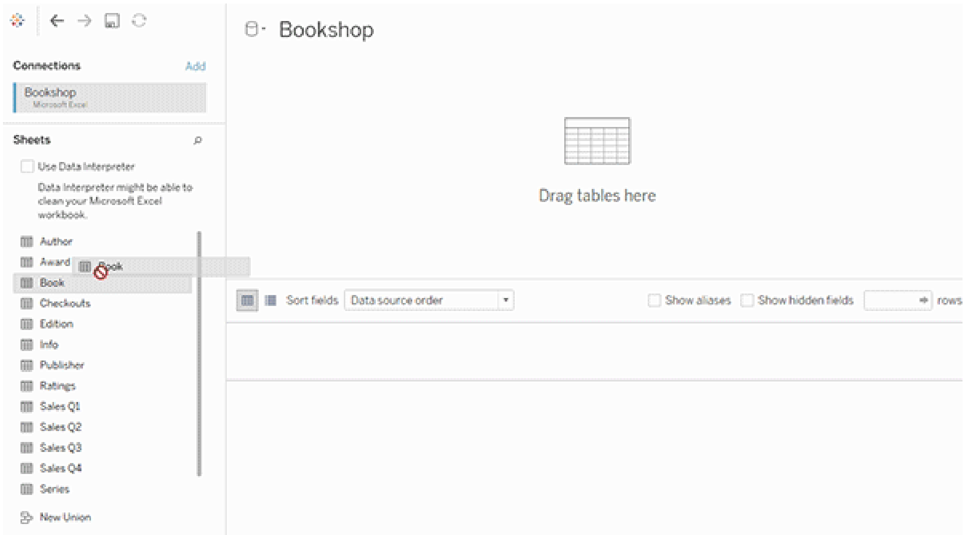
- The first table that you drag to the canvas becomes the root table for the data model in your data source.
- After you drag out the root table, you can drag out additional tables in any order. You will need to consider which tables should be related to each other, and the matching field pairs that you define for each relationship.
- If you are creating a star schema, it can be helpful to drag the fact table out first, and then relate dimension tables to that table.
- Deleting a table in the canvas automatically deletes its related descendants as well. If you delete the root table, all other tables in the model are also removed.
- Each relationship must be made of at least one matched pair of fields. Add multiple field pairs to create a compound relationship. Matched pairs must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- Relationships can be based on calculated fields.
- You can specify how fields used in the relationships should be compared by using operators when you define the relationship.

Note: You can't edit the data model of a published data source.

For more information about relationships, see Create and define relationships in Relate Your Data.

Multi-table model

- To create a multi-table model, drag tables to the logical layer of the Data Source page canvas.



Tables that you drag to the logical layer of the Data Source page canvas must be related to each other. When you drag additional tables to the logical layer canvas, Tableau automatically attempts to create the relationship based on existing key constraints and matching fields to define the relationship. If it can't determine the matching fields, you will need to select them.

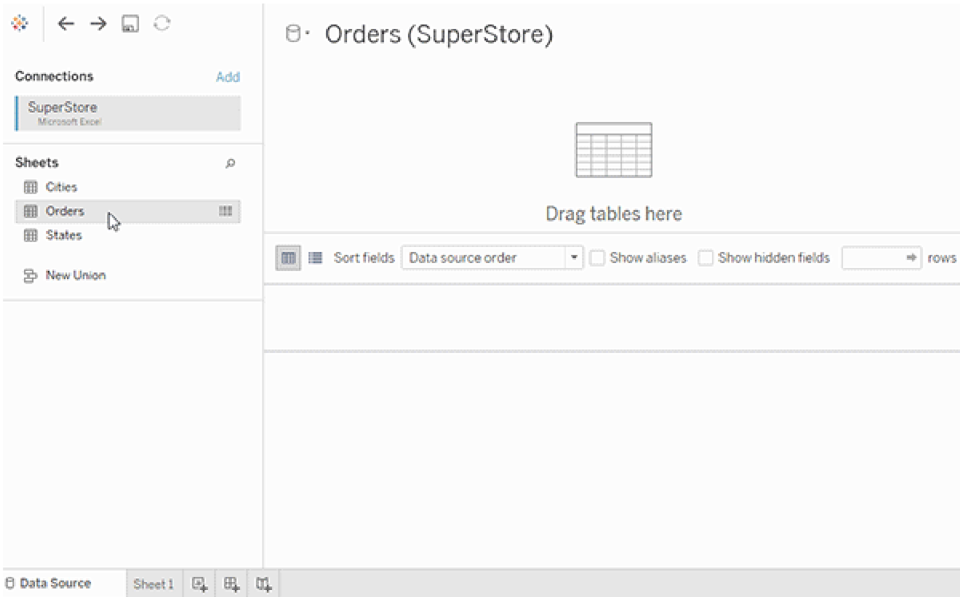
If no constraints are detected, a **Many-to-many** relationship is created and referential integrity is set to **Some records match**. These default settings are a safe choice and provide the most a lot of flexibility for your data source. The default settings support full outer joins and optimize queries by aggregating table data before forming joins during analysis. All column and row data from each table becomes available for analysis.

You can add more data inside any logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged in their logical table.

Follow the steps in Create and define relationships to combine multiple tables.

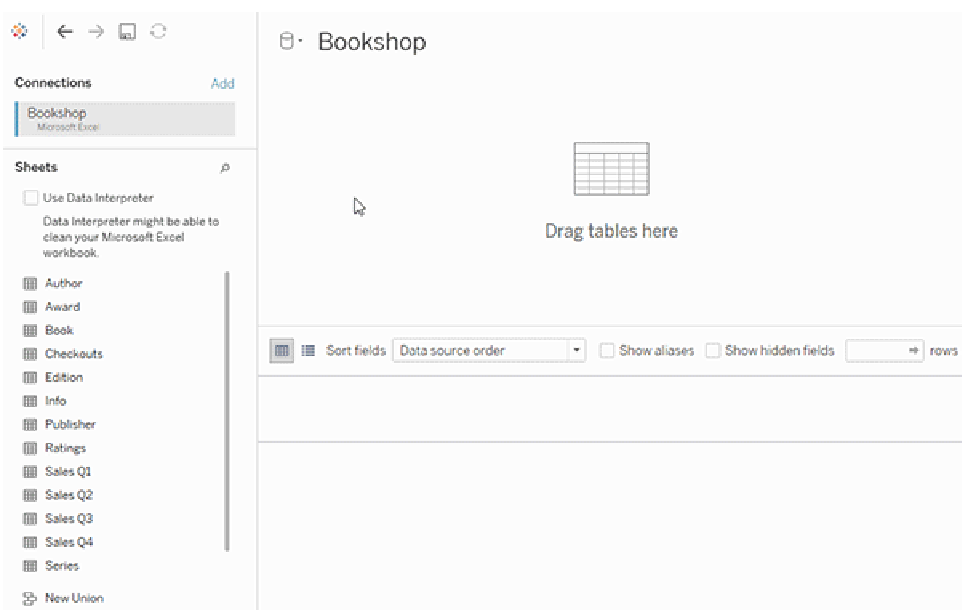
Single-table model

- To create a single-table model, drag a table into the logical layer canvas of the Data Source page. You can then use the fields from that table in the Data pane for analysis.



Single-table model that contains other tables

You can add more data inside the single, logical table by double-clicking the table. This opens the physical layer of the Data Source page canvas. If you need to use joins or unions, you can drag the tables you want to join or union into the physical layer canvas. The physical tables are merged in their logical table.



This example shows the Book table in the Relationships canvas (logical layer) of the data source. Double-clicking the Book logical table opens the Join/Union canvas (physical layer).

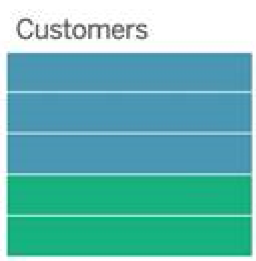
In this example, the joins merge the Award and Info tables with the Book table. In this case, the join between Book and Award will be one-to-many, at the level of detail of awards. This would duplicate measure values for Book and Info. To avoid duplication, you could relate Award and Info to Book instead of joining them inside of the Book logical table.

Supported data model schemas

The data modeling capabilities introduced to Tableau in 2020.2 are designed to make analysis over common multi-table data scenarios—including star and snowflake data models—easy. The following types of models are supported in Tableau data sources.

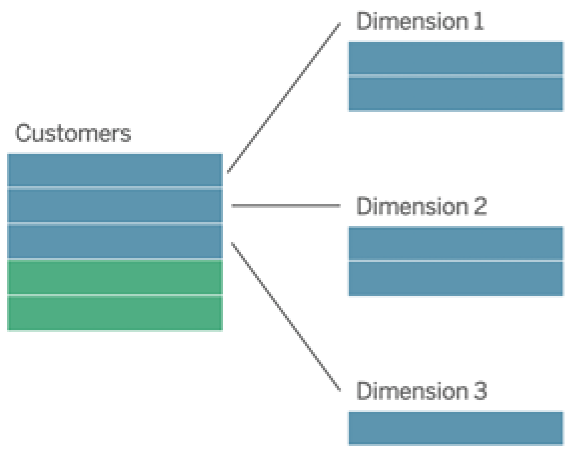
Single-table

Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2. You can build a logical table using a combination of joins, unions, custom SQL, and so on.



Star and snowflake

In enterprise data warehouses, it is common to have data structured in star or snowflake schemas where measures are contained in a central fact table and dimensions are stored separately in independent dimension tables. This organization of data supports many common analysis flows including rollup and drill down.



These models can be directly represented with relationships in the data modeling capabilities available starting with Tableau 2020.2.

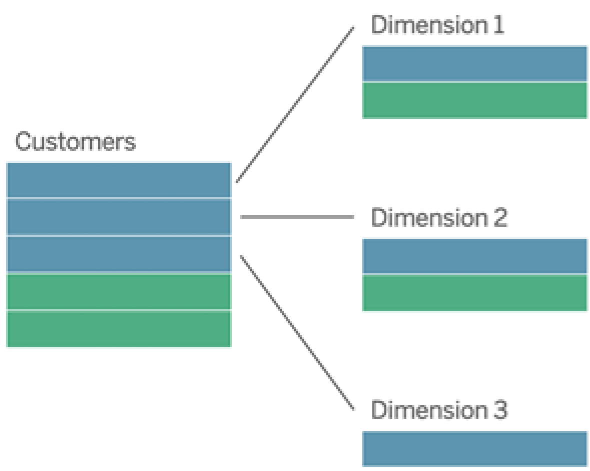
Drag the fact table into the model first and then relate the dimension tables to the fact table (in a star schema) or to other dimension tables (in a snowflake).

Typically, in a well-modeled star or snowflake schema, the relationships between the fact table and the dimension tables will be many-to-one. If this information is encoded in your data warehouse, Tableau will automatically use this to set the relationship's Performance Options. If not, you can set this information yourself. For more information, see Optimize Relationship Queries Using Performance Options.

In a well-modeled star or snowflake schema, every row in the fact table will have a matching entry in each of the dimension tables. If this is true and captured in your data warehouse integrity constraints, Tableau will automatically use this information to set the referential integrity setting in Performance Options. If some fact table rows do not have a matching row in a dimension table (sometimes called "late-arriving dimensions" or "early-arriving facts"), Tableau will default to retaining all rows when computing measures, but may drop values when showing dimension headers. For more information, see Optimize Relationship Queries Using Performance Options.

Star and snowflake with measures in more than one table

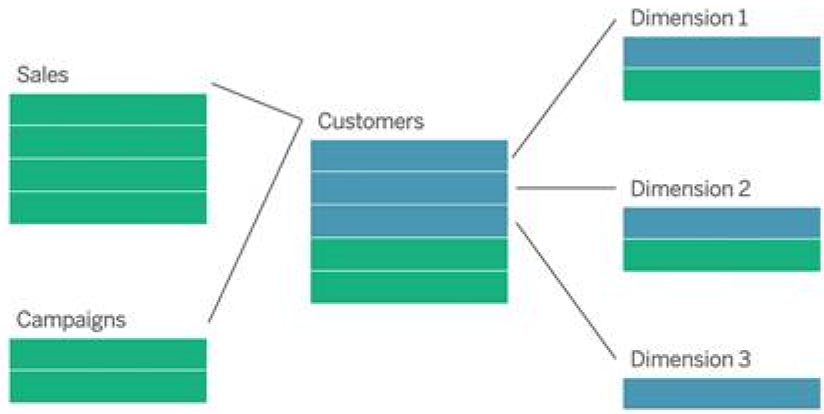
In some star or snowflake schemas, all the measures for your analysis are contained in the fact table. However, it is often true that additional measures of interest may be related to the dimension tables in your analysis. Even if the dimension tables do not contain measures, it is common in analysis to want to count or otherwise aggregate dimension values. In these cases, the distinction between fact tables and dimension tables is less clear. To create clarity when viewing your data model, we recommended adding the finest grain table to the data source canvas first, and then relating all other tables to that first table.



If you were to join these tables together into a single logical table, the measures in the dimension tables would be replicated, resulting in distorted aggregates unless you took precautions to deduplicate the values using LOD calculations or COUNT DISTINCT. However, if you instead create relationships between these tables, Tableau will aggregate measures before performing joins, avoiding the problem of unnecessary duplication. This relieves you of the need to carefully track the level of detail of your measures.

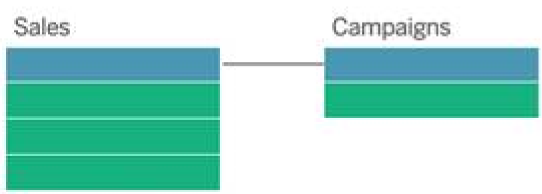
Multi-fact analysis

Tableau's data modeling capabilities support some forms of multi-fact analysis. Additional fact tables (containing measures) can be added to any of the previously mentioned models as long as they only relate to a single dimension table. For example, you can bring two or more fact tables together to analyze a shared dimension, such as in Customer 360-like analyses. These fact tables can be at a different level of detail than the dimension table, or from each other. They can also have a many-to-many relationship with the dimension table. In these scenarios, Tableau will ensure that values are not replicated before aggregation.



If you don’t have a shared dimension table that relates your fact tables, you can sometimes dynamically build one using custom SQL or by using joins or unions of other dimension tables.

Two fact tables can be related directly to each other on a common dimension. This type of analysis works best when one of the fact tables contains a superset of the common dimension.



Unsupported models

- **Multiple fact tables related to multiple shared dimension tables**. In some use cases it is common to have multiple fact tables related to multiple shared dimension tables. For example, you might have two fact tables, Store Sales and Internet Sales, related to two common dimension tables, Date and Customer. Typically, such scenarios would require creating a circular relationship in your data model. Circular relationships are not supported in 2020.2.

  You can approximate this type of model by merging some of the tables in the physical layer. For example, you might be able to union Store Sales and Internet Sales into a single table, which can then be related to Date and Customer. Alternatively, you might be able to cross-join Date and Customer to create a single dimension table which can then be related to Store Sales and Internet Sales.

- **Directly relating 3 or more fact tables on shared dimensions**. While it is possible to build this model in the logical layer, you might see unwanted results, unless you only use dimensions from a single table.

Requirements for relationships in a data model

- When relating tables, the fields that define the relationships must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't edit relationships in a published data source.
- You can't define relationships between published data sources.
- Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page in Tableau Online or Tableau Server.

Factors that limit the benefits of using related tables

- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions).

How Relationships Differ from Joins

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. You don’t define join types for relationships, so you won’t see a Venn diagram when you create them.

Think of a relationship as a contract between two tables. When you are building a viz with fields from these tables, Tableau brings in data from these tables using that contract to build a query with the appropriate joins.

- **No up-front join type**. You only need to select matching fields to define a relationship (no join types). Tableau first attempts to create the relationship based on existing key constraints and matching field names. You can then check to ensure they are the fields you want to use, or add more field pairs to better define how the tables should be related.

- **Automatic and context-aware**. Relationships defer joins to the time and context of analysis. Tableau automatically selects join types based on the fields being used in the visualization. During analysis, Tableau adjusts join types intelligently and preserves the native level of detail in your data. You can see aggregations at the level of detail of the fields in your viz rather than having to think about the underlying joins. You don't need to use LOD expressions such as FIXED to deduplicate data in related tables.
- **Flexible**. Relationships can be many-to-many and support full outer joins. When you combine tables using relationships, it's like creating a custom, flexible data source for every viz, all in a single data source for the workbook. Because Tableau queries only tables that are needed based on fields and filters in a viz, you can build a data source that can be used for a variety of analytic flows.

For more information, see Relate Your Data and Don't Be Scared of Relationships.

Joins are still available as an option for combining your data. Double-click a logical table to go to the join canvas. For more information, see **Where did joins go?** on page 824

Watch a video: For an overview of data source enhancements and an introduction to using relationships in Tableau, see this 5-minute video.

For related information about how relationship queries work, see these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Characteristics of relationships and joins

Relationships are a dynamic, flexible way to combine data from multiple tables for analysis. We recommend using relationships as your first approach to combining your data because it makes data preparation and analysis easier and more intuitive. Use joins only when you absolutely need to.

Here are some advantages to using relationships to combine tables:

- Make your data source easier to define, change, and reuse.
- Make it easier to analyze data across multiple tables at the correct level of detail (LOD).
- Do not require the use of LOD expressions or LOD calculations for analysis at different levels of detail.
- Only query data from tables with fields used in the current viz.

Relationships

- Are displayed as flexible noodles between logical tables
- Require you to select matching fields between two logical tables
- Do not require you to select join types
- Make all row and column data from related tables potentially available in the data source
- Maintain each table's level of detail in the data source and during analysis
- Create independent domains at multiple levels of detail. Tables aren't merged together in the data source.
- During analysis, create the appropriate joins automatically, based on the fields in use.
- Do not duplicate aggregate values (when Performance Options are set to Many-to-Many)
- Keep unmatched measure values (when Performance Options are set to Some Records Match)

Joins

Joins are a more static way to combine data. Joins must be defined between physical tables up front, before analysis, and can’t be changed without impacting all sheets using that data source. Joined tables are always merged into a single table. As a result, sometimes joined data is missing unmatched values, or duplicates aggregated values.

- Are displayed with Venn diagram icons between physical tables
- Require you to select join types and join clauses
- Joined physical tables are merged into a single logical table with a fixed combination of data
- May drop unmatched measure values
- May duplicate aggregate values when fields are at different levels of detail
- Support scenarios that require a single table of data, such as extract filters and aggregation

Requirements for using relationships

- When relating tables, the fields that define the relationships must have the same data type. Changing the data type in the Data Source page does not change this requirement. Tableau will still use the data type in the underlying database for queries.
- You can't define relationships based on geographic fields.
- Circular relationships aren't supported in the data model.
- You can't edit relationships in a published data source.
- You can't define relationships between published data sources.

- Your workbook must use an embedded data source for you to be able to edit relationships and performance options in the Data Source page in Tableau Online or Tableau Server.

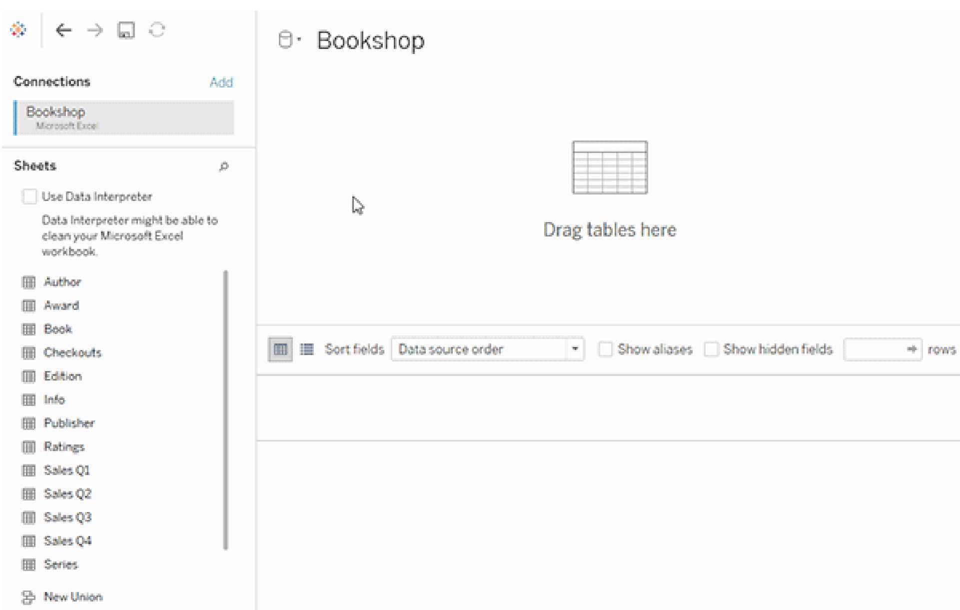
Factors that limit the benefits of using related tables

- Dirty data in tables (i.e. tables that weren't created with a well-structured model in mind and contain a mix of measures and dimensions in multiple tables) can make multi-table analysis more complex.
- Using data source filters will limit Tableau's ability to do join culling in the data. Join culling is a term for how Tableau simplifies queries by removing unnecessary joins.
- Tables with a lot of unmatched values across relationships.
- Interrelating multiple fact tables with multiple dimension tables (attempting to model shared or conformed dimensions).

Where did joins go?

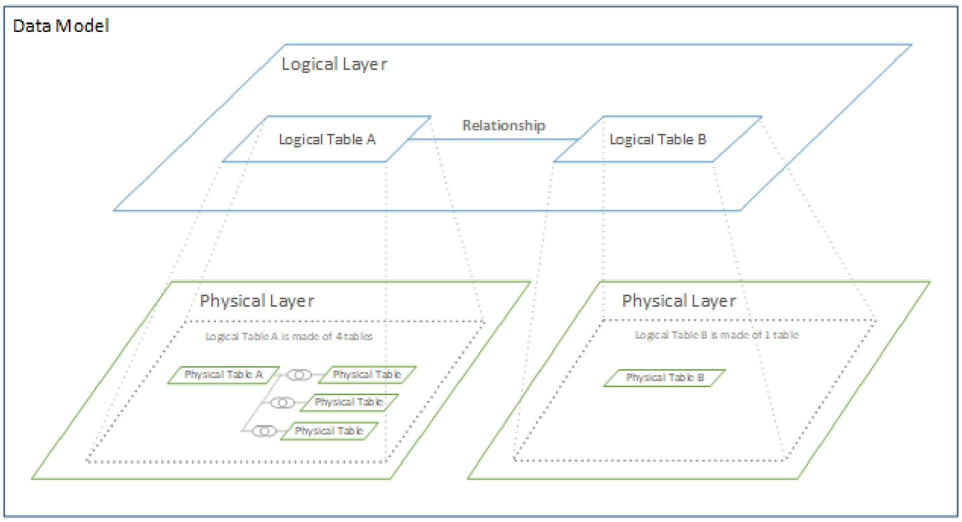
You can still specify joins between tables in the physical layer of a data source. Double-click a logical table to go to the Join/Union canvas in the physical layer and add joins or unions.

Every top-level, logical table contains at least one physical table. Open a logical table to view, edit, or create joins between its physical tables. Right-click a logical table, and then click **Open**. Or, just double-click the table to open it.



When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships.

The next layer is the physical layer of the data source. You combine data between tables at the physical layer using joins. For more information, see Logical and physical tables in the data model.



Optimize Relationship Queries Using Performance Options

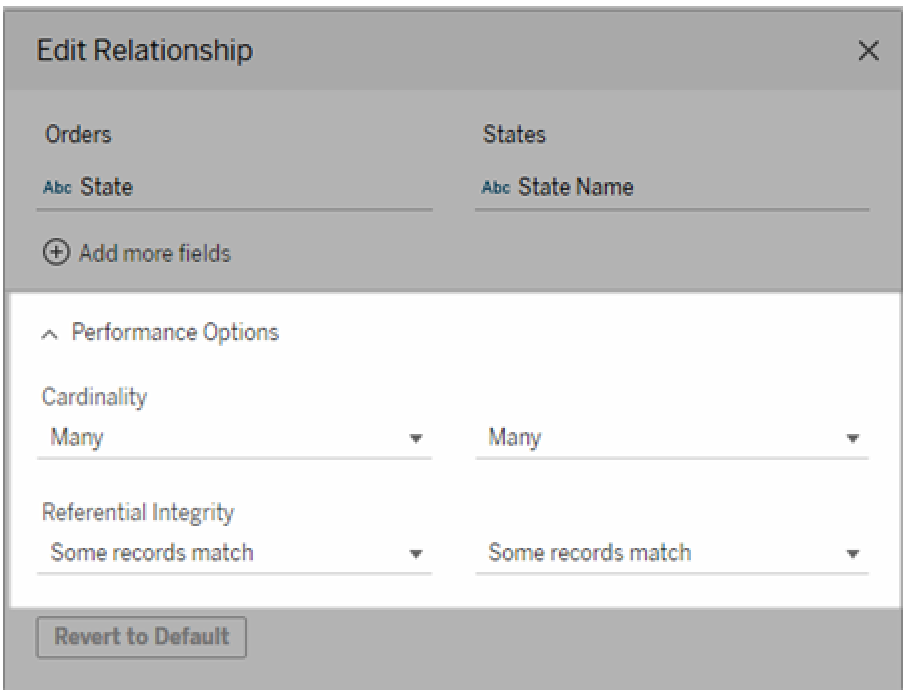
Performance Options are optional settings that define the cardinality (uniqueness) and referential integrity (matching records) between the two tables in a relationship. These settings help Tableau optimize queries during analysis.

- **If you aren't sure what to choose**, use the recommended default settings indicated by Tableau. Using the defaults is safe and will automatically generate correct aggregations and joins during analysis. If you don't know the cardinality or referential integrity, you don't need to change these settings.
- **If you know the shape of your data**, you can optionally change these settings to represent uniqueness and matching between the records in the two tables.

In many analytical scenarios, using the default settings for a relationship will give you all of the data you need for analysis. In some scenarios, you might want to adjust the Performance Options settings to describe your data more accurately. For more details about using relationships to combine and analyze data, see Relate Your Data and this Tableau blog post: Relationships, part 1: Introducing new data modeling in Tableau.

Note: In web authoring in Tableau Server or Tableau Online: Your workbook must be connected to a live data source for you to be able to edit relationships and performance options in the Data Source page. You cannot edit the data model of a published data source.

What the Cardinality and Referential Integrity settings mean



Cardinality options

Cardinality settings determine if Tableau aggregates table data before or after automatically joining the data during analysis.

- Select **Many** if the field values aren't unique, or you don't know. Tableau will aggregate the relevant data before forming joins during analysis.
- Select **One** if field values are unique. During analysis, the relevant data will be joined before aggregation. Setting this option correctly optimizes queries in the workbook when the field values in the relationship are unique. However, selecting **One** when field values aren’t unique can result in duplicate aggregate values being shown in the view.

Note: Selecting **One** treats records as if each key value is unique and there is at most only one row with a null value.

Referential Integrity options

Referential Integrity settings determine the type of join used to get the dimension values for a measure during analysis.

- Select **Some Records Match** if some values in the field don't have a match in the other table, or you don't know. During analysis, Tableau uses outer joins to get dimensions values for a measure. All measure values will be shown in the view, even unmatched measures.
- Select **All Records Match** if values in the field are guaranteed to have a match in the other table. This setting generates fewer and simpler joins during analysis, and optimizes queries. You might see inconsistent results during analysis (unmatched values removed or missing in view) if there are unmatched values in this table.

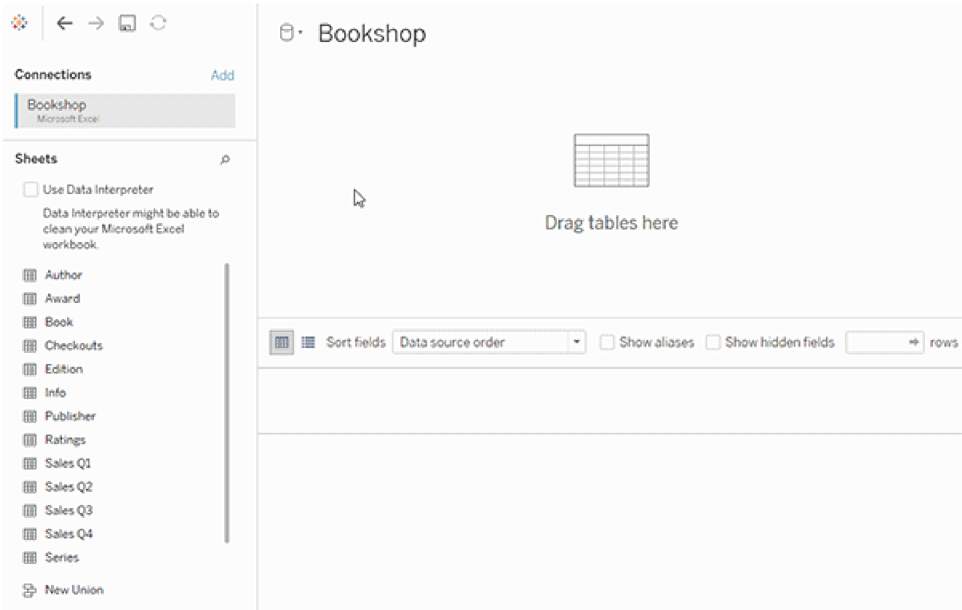
Notes: Selecting **All Records Match** treats records as if no null values exist in the fields used for the relationship. During analysis, Tableau will use inner joins to get dimension values for a measure. By default, Tableau will never join null keys.

For more information about Cardinality and Referential integrity as concepts, see Cardinality and Referential Integrity.

Where did joins go?

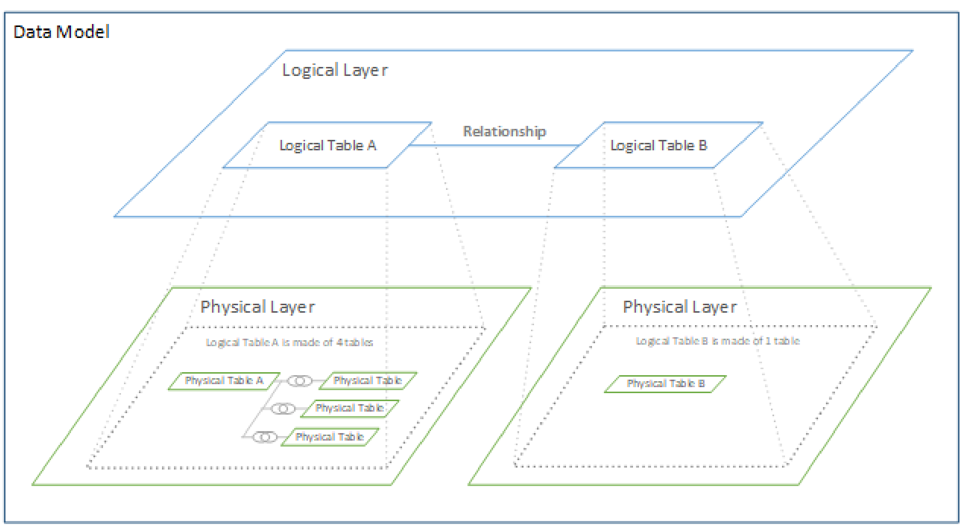
You can still specify joins between tables in the physical layer of a data source. Double-click a logical table to go to the join canvas.

Every top-level, logical table contains at least one physical table. Open a logical table to view, edit, or create joins between its physical tables. Right-click a logical table, and then click **Open**. Or, just double-click the table to open it.



When you create a data source, it has two layers. The top-level layer is the logical layer of the data source. You combine data between tables in the logical layer using relationships.

The next layer is the physical layer of the data source. You combine data between tables at the physical layer using joins. For more information, see Logical and physical tables in the data model.



Tips on using Performance Options

If you know the shape of your data, you can use the optional settings in Performance Options to establish the cardinality of the tables to each other (one-to-one, one-to-many, many-to-many) and indicate referential integrity (values from one table will always have match in the other table).

Instead of thinking of the settings in Performance Options as “yes” and “no”, think of them as “yes” and “I don’t know”. If you are sure that a table's values are unique, select **One**. If you are sure that each record in one table matches one or more records in the other table, select **All Records Match**. Otherwise, leave the default settings as they are.

If you aren't sure about the shape of your data, use the default settings. When Tableau can't detect these settings in your data, the default settings are:

- Cardinality: Many-to-Many
- Referential integrity: Some Records Match

If Tableau detects key relationships or referential integrity in your data, those settings will be used and indicated as "detected".

To reapply the default settings, click **Revert to Default**.

Terms defined

Cardinality refers to the uniqueness of data contained in a field (column) or combination of fields. When the tables you want to analyze contain many rows of data, queries can be slow (and performance of the overall data source is affected) so we recommend choosing a method for combining data based on the cardinality of the related columns between tables.

- Low cardinality: When related columns have a lot of repeated data. For example, a table called Products might contain a Category column that contains three values: Furniture, Office Supplies, and Technology.
- High cardinality: When related columns have highly unique data. For example, a table called Orders might contain an Order ID column that contains a unique value for every order of product.

Referential integrity means that one table will always have a matching row in the other table. For example, a Sales table will always have a matching row in the Product Catalog table.

Cardinality and Referential Integrity

Setting up a data source, no matter how you combine the data, requires understanding the data structure of each table and how they can be combined. There are several key elements that should be considered:

- **Level of detail**: how detailed the data is—its granularity. This can be thought of as answering the question "what defines a row?". For more information on granularity, see **Structure Data for Analysis** on page 749
- **Shared field**: there must be at least one field that can be used to form the link between tables. For a join, these fields define the *join clause*. In related tables, they establish the *relationship*.
- **Cardinality**: how many or how few unique values there are for the shared field (uniqueness). For more information, see the next section.
- **Referential Integrity**: a value in one table is guaranteed to have a match in the other table. In other words, there can't be a record in one table that does not have a corresponding record in the other table. For more information, see below.

Cardinality

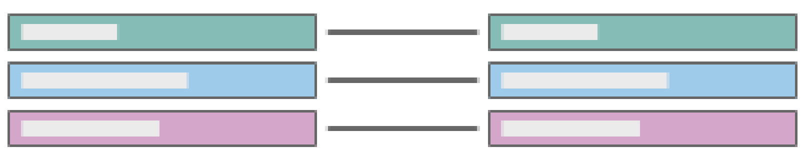
Cardinality in a single column or field refers to how unique its values are. Low cardinality means there are only a few unique values (such as in a field for eye color). High cardinality means there are a lot of unique values (such as in a field for phone numbers).

Cardinality between tables is similar, but refers to whether a row from one table *could be* linked with more than one row in another table. (It's important to remember that cardinality does not address whether there is missing data in either table. The presence of absence of missing data is referential integrity. Although these concepts work together, they are two different attributes of the relationship.)

The options are one-to-one, one-to-many, many-to-one, or many-to-many.

One-to-one

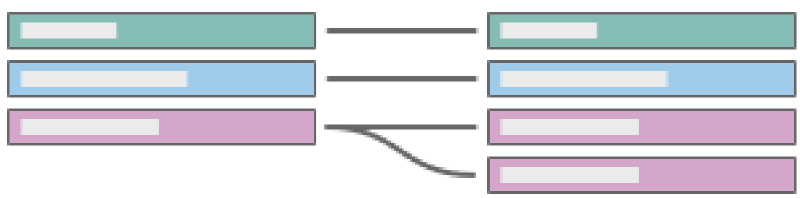
- *Definition*: Each value of the shared field in one table is related to **at most** one value in the shared field in the other table.
- *Abbreviation*: 1:1



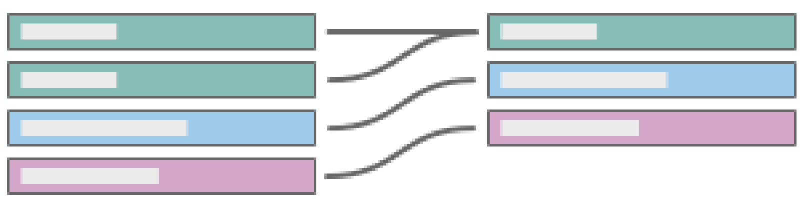
Example: Every car has its own license plate and a license plate is specific to an individual car. Car-to-license is one-to-one.

Note that even if a car is unregistered or a license plate number hasn't yet been assigned to a car, that discrepancy is described by the referential integrity. A car can only have one license plate and a license plate can only be assigned to one car, so the cardinality remains one-to-one.

One-to-many or **Many-to-one**



- *Definition*: Each value of the shared field in one table can be related to multiple rows the other table (that value is repeated in the other table).

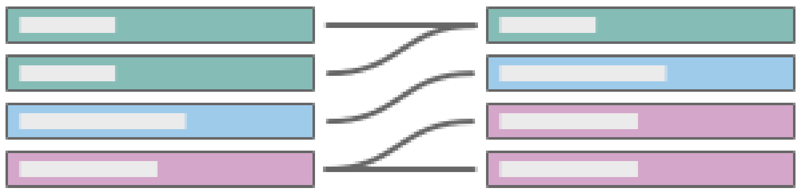


- *Abbreviation*: m:1 or 1:m
- The order of the tables determines whether it is many-to-one or one-to-many (similar to left and right joins).

Examples: Many employees have the

same manager. Employees-to-manager is many-to-one. Manager-to-employee is one-to-many.

Many-to-many



- Multiple values of the shared field in one table can be related to multiple rows in the other table (values of the shared field can be repeated in multiple rows in either table).
- Abbreviated as m:m.
- This is the default setting if no other settings are detected in the data source.

Examples: An actor is in many movies and a movie has many actors. Actor-to-movie is many-to-many. Multiple books can be purchased in the same transaction and a book can be purchased multiple times. ISBN-to-OrderID is many-to-many.

Cardinality can be specified in the Performance Options settings. For more information, see **Optimize Relationship Queries Using Performance Options** on page 825.

Referential Integrity

There's a related concept called referential integrity, which means a row in one table will always have a matching row in the other table, as determined by the value of their shared fields. If the database contains no records for cars without license plates or license plates without cars, that relationship has referential integrity.

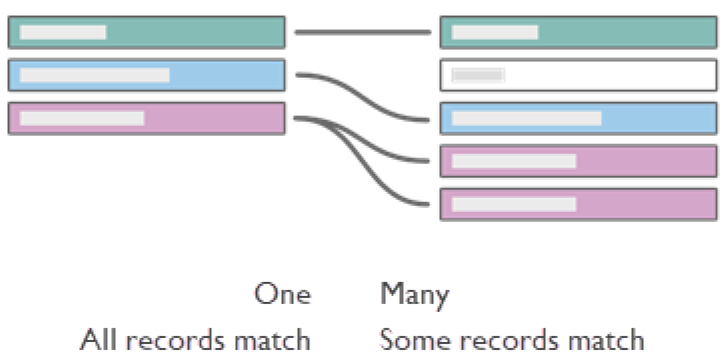
In Tableau, referential integrity is configured on each side of the relationship. In the Performance Options settings, **Some records match** means there isn't (or you don't know if there is) referential integrity. **All records match** means there is referential integrity. The default setting is to not assume referential integrity (Some records match).

For more information, see **Optimize Relationship Queries Using Performance Options** on page 825.

Test yourself

Can you define the cardinality and referential integrity of each diagram? What does this mean in words?

Example:



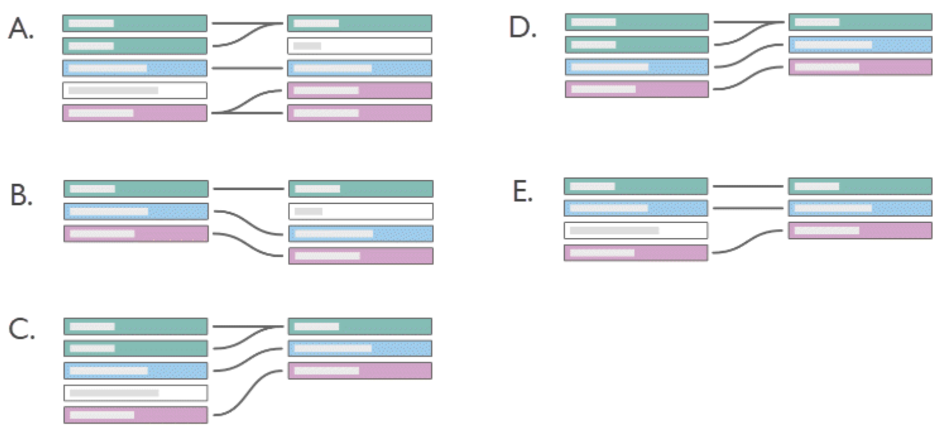
If we set the left table as books and the right table as authors linked on AuthorID, translate the diagram into words:

- **One book can have multiple authors** (the purple records show one row in the book table on the left corresponding to multiple records in the author table on the right).
- **No authors have more than one book** (each author record on the right leads only one book record on the left).

- **There are no books without authors** (no records on the left fail to correspond to a record on the right).
- **Some authors may not have books** (the gray author record on the right has no corresponding book record on the left.)

Click each section below to expand it.

Check your knowledge



Solution

Why does it matter?

Correctly configuring the cardinality or referential integrity settings can boost performance through query optimization. Incorrect configurations, however, can lead to issues of aggregation due to loss or duplication of data. The default Performance Option settings are **Many** for cardinality and **Some records match** for referential integrity. These should only be adjusted if you are sure of the correct characteristics of your data.

For more information about how Tableau handles each setting, see **What the Cardinality and Referential Integrity settings mean** on page 826.

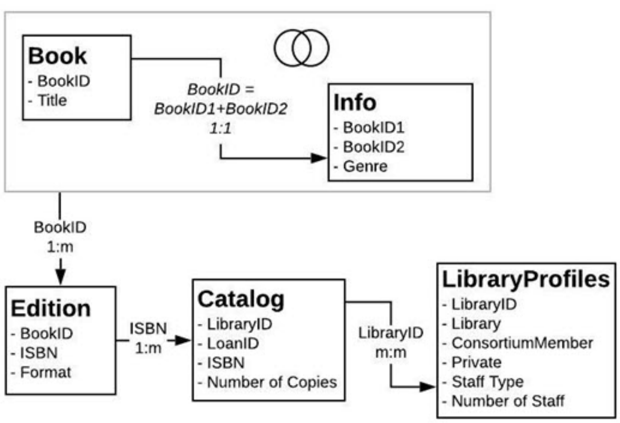
An example in Tableau

Let's explore what happens when cardinality is configured improperly.

Note: The following example uses a subset of tables from **The Bookshop data set** on page 358. You can download the workbook to follow along, or download the raw data to create the data sources yourself. The tables used are **Books**, **Info** and **Edition** from *Bookshop.xlsx* (keeping only some fields) and **LibraryProfile** and **Catalog** from *BookshopLibraries.xlsx*.

The Book and Info tables have a one-to-one relationship—Info is essentially additional columns for the Book table. Because of this, while they could be related, it makes sense to join them to create a new logical table that has all the columns. Edition has a many-to-one relationship with this combined table as there can be multiple editions for a single book, usually with different formats. (Note that the diagram below shows the relationship from the Book+Info table to Edition, so it is one-to-many.)

Edition is related to Catalog as a one-to-many relationship on ISBN. The Catalog and LibraryProfile tables are related many-to-many on Library ID. The key point is that the LibraryProfiles table has multiple rows per library, one for each staff type (Librarian, Library Assistant, Library Technician). For more information about the structure of these tables, see **The Bookshop data set** on page 358.



The right settings

When the Catalog-LibraryProfile relationship is set up correctly, we can make a simple viz that shows the number of staff for each library for several books. This is a silly viz to make, but it's useful to illustrate the point. Idle Hour Library has 130 staff regardless of which book we're talking about. There are three values for staff type, so each total is made up of three records—the number in parentheses.

Title	Bide Awhile	Idle Hour Library	IndieUnBound	Old Friend Libr..
Alanna Saves the Day	35 (3)	130 (3)	56 (3)	25 (3)
Banana Slug and the Glass Half Full	35 (3)	130 (3)	56 (3)	25 (3)
Banana Slug and the Lost Cow	35 (3)	130 (3)	56 (3)	25 (3)
Banana Slug and the Mossy Rock	35 (3)	130 (3)	56 (3)	25 (3)
Banana Slug and Xyr Friends	35 (3)	130 (3)	56 (3)	25 (3)
Heliotrope Pajamas	35 (3)	130 (3)	56 (3)	25 (3)

Staff count by library and title. (Numbers in parentheses indicate the number of records in each mark.)

The wrong settings: one-to-one

When the relationship is erroneously set as one-to-one, in the viz each title from Catalog is effectively paired with only one record from the LibraryProfile table (as indicated by the record count in parentheses).

Title	Bide Awhile	Idle Hour Library	IndieUnBound	Old Friend Libr..
Alanna Saves the Day	6 (1)	16 (1)	2 (1)	3 (1)
Banana Slug and the Glass Half Full	6 (1)	16 (1)	2 (1)	3 (1)
Banana Slug and the Lost Cow	6 (1)	16 (1)	2 (1)	3 (1)
Banana Slug and the Mossy Rock	6 (1)	16 (1)	2 (1)	3 (1)
Banana Slug and Xyr Friends	6 (1)	16 (1)	2 (1)	3 (1)
Heliotrope Pajamas	6 (1)	16 (1)	2 (1)	3 (1)

Staff count by library and title. (Numbers in parentheses indicate the number of records in each mark.)

Above, we can see that each library only displays their minimum number of staff. (Refer to the bolded numbers in the viz below. The lowest number of staff is the number reflected in the Staff Count viz.)

Staff breakdown

Staff Type	Bide Awhile	Idle Hour Library	IndieUn..	Old Friend L..
Librarians	9	53	7	3
Library Assistants	6	16	47	17
Library Technicians	20	61	2	5
Grand Total	35	130	56	25

Staff breakdown by type and library.

For more information about how relationships become contextual joins for a viz, see Introducing the new data modeling in Tableau on the Tableau blog.

The wrong setting: joining

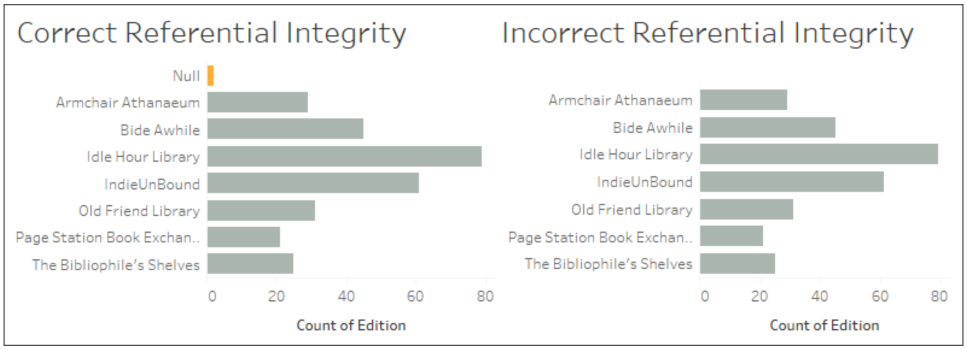
While there are ways to work around this sort of issue—Level of Detail expressions being a common one—joining tables that have different granularity or “many” in their cardinality can cause duplication. Here, the staff counts are accurate for titles that only have one format, but for the books that have two formats in the Editions table, that doubling is passed along to the staff counts as well (note the record counts in parentheses that are 6 instead of the correct 3).

Title	Bide Awhile	Idle Hour Library	IndieUnBound	Old Friend Libr..
Alanna Saves the Day	35 (3)	130 (3)	56 (3)	25 (3)
Banana Slug and the Glass Half Full	35 (3)	260 (6)	56 (3)	25 (3)
Banana Slug and the Lost Cow	35 (3)	260 (6)	56 (3)	25 (3)
Banana Slug and the Mossy Rock	35 (3)	260 (6)	56 (3)	25 (3)
Banana Slug and Xyr Friends	35 (3)	260 (6)	56 (3)	25 (3)
Heliotrope Pajamas	35 (3)	260 (6)	112 (6)	25 (3)

Staff count by library and title. (Numbers in parentheses indicate the number of records in each mark.)

The wrong setting: incorrectly assuming referential integrity

Telling Tableau there is referential integrity (all records match) when this isn’t the case can cause dropped values. Here, these two vizzes are similar but the one on the right is from a data source configured to assume referential integrity. That viz has lost the nulls. While this can be fine in some circumstances, it’s important to understand what those nulls represent. Here, where the viz shows the number of editions in each library, the nulls indicate two editions that are present in the edition table but aren't held by any library. This could be an important oversight, and it's one that incorrectly assuming referential integrity would overlook.



Explore the workbook and its data sources to see what other issues may arise from improperly combined tables.

Performance impacts

If mis-configuring these settings can cause missing or duplicated data, why does Tableau allow them to be changed at all? In many cases, you can and should leave the default settings: relate tables instead of joining, leave the cardinality as many-to-many, and don't assume referential integrity. Especially if you're not sure what the settings should be.

However, cardinality and referential integrity are Performance Options because there can be performance implications to the defaults. If you're sure about the structure of your data, configuring the correct settings can reduce the query execution to improve the speed.

Under the hood

Note: This section uses analogies to other data combination techniques to provide a conceptual framework only. It is not a technical description of how Tableau uses the performance settings for relationships.

Cardinality

The cardinality of the relationship impacts when aggregation happens. This can be thought of in terms of blending. Data blending queries two data sources independently. Each data source is aggregated as necessary to the desired level of detail for the view regardless of the other data source. For relationships, the cardinality setting impacts whether the aggregation happens before or after the join.

In the example above, the Many setting means the number of staff for each library is aggregated before combining that data with the book information, thus ensuring every book has the correct numbers. When the cardinality was incorrectly set to One, the number of staff was not aggregated before it was combined with the book data, leading to incorrect values.

Note that not only are the incorrect values shown, all the values are assigned to the staff type Librarians, despite the fact they are taken from all three staff types. Misconfiguration of this setting can cause unpredictable and incorrect values. This filtering of results only happens when a field from another table on the other side of the incorrectly set relationship is used in the view.

Staff Counts m:m

Staff Type	Bide Awhile	Idle Hour Library	IndieUnBound	Old Friend Library
Librarians	9	53	7	3
Library Assistants	6	16	47	17
Library Technicians	20	61	2	5

Staff Counts 1:1 (multi table)

Staff Type	Bide Awhile	Idle Hour Library	IndieUnBound	Old Friend Library
Librarians	6	16	2	3
Library Assistants				
Library Technicians				

Staff Counts 1:1 (single table)

Staff Type	Bide Awhile	Idle Hour Library	IndieUnBound	Old Friend Library
Librarians	9	53	7	3
Library Assistants	6	16	47	17
Library Technicians	20	61	2	5

If the values are unique, however, Tableau is free to remove the pre-join aggregation if it optimizes the query.

Referential Integrity

Although referential integrity refers to a setting for relationships, it can be thought of in terms of join types. A full outer join will retain all records, regardless of whether or not there is a match in the other table, but at a performance cost. If you're not sure if records would be lost, an outer join is safer. This is how tables are treated when there may not be referential integrity (Some records match).

An inner join will retain only those records where there is a match from both tables, dropping records that do not appear in each table. If you know an inner join won't eliminate necessary data, it is more efficient. If the Performance Options are set to All records match, referential integrity is assumed and joins are performed without considering unmatched values.

An incorrect referential integrity setting can have a filter-like effect on the combined data, removing unmatched values. For more information on the power of retaining unmatched records, see Asking questions across multiple related tables on the Tableau blog. For more information on join types, see **Join Your Data** on page 883.

Keep the defaults

If your analysis has acceptable performance, we strongly encourage leaving the default Performance Option settings of many-to-many and not assuming referential integrity. The power of relationships comes from their ability to provide accurate, contextually appropriate results based on the tables that are used in the analysis. Changing these settings removes the semantic flexibility of relationships.

How Analysis Works for Multi-table Data Sources that Use Relationships

Using a data source that has multiple, related tables affects how analysis works in Tableau. Because multiple, related tables have independent domains and retain their native level of detail, when you drag fields into the view:

- Data is queried at its natural level of detail.
- Only data that is relevant to a viz is queried.
- The relationships between the tables affect the results of the query. The flow of building a viz can vary depending on how tables of fields are related to each other in the data model, or if they aren't related directly.

For an overview of data source enhancements and an introduction to using relationships, see this 5-minute video.

Also see **Don't Be Scared of Relationships** on page 861 and these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Note: You can still create single-table data sources in Tableau. You can build a logical table using a combination of joins, unions, custom SQL, and so on. The behavior of single-table analysis in Tableau has not changed. Analysis over a single logical table that contains a mixture of dimensions and measures works just as in Tableau pre-2020.2.

Considerations for analysis

Validating relationships

You have several options for validating your data model for analysis. As you create the model for your data source, we recommend going to a sheet, selecting that data source, and then building a viz to explore record counts, expected data, unmatched values, nulls, or repeated measure values. Try working with fields across different tables to ensure everything looks how you expect it to.

What to look for:

- Are your relationships in the data model using the correct matching fields for their tables?
- Would adding multiple matching field pairs make the relationship more accurate?
- What are the results of dragging different dimensions and measures into the view?
- Are you seeing the expected number of rows?
- If you changed any of the Performance Options settings from the default settings, are the values that you are seeing in the viz what you would expect? If they aren't, you might want to check the settings, or reset to the default.

Options for validating relationships and the data model:

- Every table includes a count of its records, as a field named *TableName(Count)*, at the level of detail for that table. To see the count for a table, drag its Count field into the view. To see the count for all tables, select the Count field for each table in the Data pane, and then click the Text Table in Show Me.
- Click **View Data** in the Data pane to see the number of rows and data per table. Also, before you start creating relationships, viewing the data from the data source before or

during analysis can be useful to give you a sense of the scope of each table. For more information, see **View Underlying Data** on page 2728.
- Drag dimensions onto rows to see the Number of Rows in the status bar. To see unmatched values, click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. You can also drag different measures to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

Tip: If you would like to see the queries that are being generated for relationships, you can use the Performance Recorder in Tableau Desktop.

1. Click the Help menu, and then select **Settings and Performance** > **Start Performance Recording**.
2. Drag fields into the view to build your viz.
3. Click the Help menu, and then select **Settings and Performance** > **Stop Performance Recording**. The Performance Recording workbook will automatically open.
4. In the Performance Summary dashboard, under Events Sorted By Time, click an "Executing Query" bar and view the query below.

Another more advanced option is to use the Tableau Log Viewer on GitHub. You can filter on a specific keyword using `end-protocol.query`. For more information, start with the Tableau Log Viewer wiki page in GitHub.

Dimension-only visualizations

When using a multi-table data source with related tables: If you build a dimension-only viz, Tableau uses inner joins and you won't see the full unmatched domain.

To see partial combinations of dimension values, you can:

- Use Show Empty Rows/Columns to see all of the possible rows. Click the **Analysis** menu, and then select **Table Layout** > **Show Empty Rows** or **Show Empty Columns**. Be aware that this setting will also trigger densification for Date and Numeric Bin fields, which may be unwanted.
- Add a measure to the view, such as *<YourTable>(Count)* from one of the tables represented in your viz. This ensures that you will see all values of the dimensions from that table.

For more information, also see **Unmatched values behavior for dimensions that might surprise you** on page 795 and **Troubleshooting multi-table analysis** on page 856.

When to use LOD calculations and expressions

Because Tableau understands the level of detail (LOD) of your input tables, you shouldn't need to use LOD calcs to remove unwanted duplication due to joins.

You may still want to use LOD calcs to:

- Handle unwanted duplication in your source tables.
- Compute multi-level aggregations (e.g. an average over a sum)
- To do cohort analysis (e.g. to compute the first order date for each customer)

If the dimensionality of an LOD calc includes fields from a single table, that LOD calculation will appear in its owning table in the Data pane.

Examples of multi-table analysis

The following set of examples show how data might be queried across multiple, related tables. This multi-table data source contains a short list of actors' appearances in different movie roles.

A row in the Appearances table means that an Actor played a particular Character in a specific Movie. In this data set, an Actor can have zero or more Appearances.

Learn more about how relationships work in these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks

Example 1: Analysis of a single question in joined data versus related data

When you connected to data in Tableau before 2020.2, your data source could consist of a single table, or multiple tables that were joined or unioned together into a single, denormalized table. Starting with Tableau 2020.2, Tableau recognizes and preserves normalized data for multi-table data sources, where the table data remains separate, and each table maintains its native level of detail. The following examples show how analysis differs between single table and multi-table data sources.

This example shows three tables of movie data: Appearances, Actors, and Movies.

Movie	Character	Actor
The Fellowship of the Ring	Gimli	John Rhys-Davies
The Return of the King	Gimli	John Rhys-Davies
The Return of the King	Treebeard	John Rhys-Davies
The Two Towers	Gimli	John Rhys-Davies
The Two Towers	Treebeard	John Rhys-Davies
Casino Royale	James Bond	Peter Sellers
Dr. Strangelove	Dr. Strangelove	Peter Sellers
Dr. Strangelove	Group Captain Mandr...	Peter Sellers
Dr. Strangelove	President Muffley	Peter Sellers
From Russia With Love	James Bond	Sean Connery
The Hunt for the Red October	Marko Ramius	Sean Connery

Actor	Height (cm)
John Rhys-Davies	185
Peter Sellers	173
Sean Connery	189
Jodie Foster	161

Movie	World Premiere	Gross (USD millions)
The Fellowship of the Ring	12/19/2001	869
The Return of the King	12/18/2002	1,119
The Two Towers	12/17/2003	923
Casino Royale	4/13/1967	22
Dr. Strangelove	1/29/1964	94
From Russia With Love	5/27/1964	24
The Hunt for the Red October	3/2/1990	200
Casablanca	11/26/1942	4

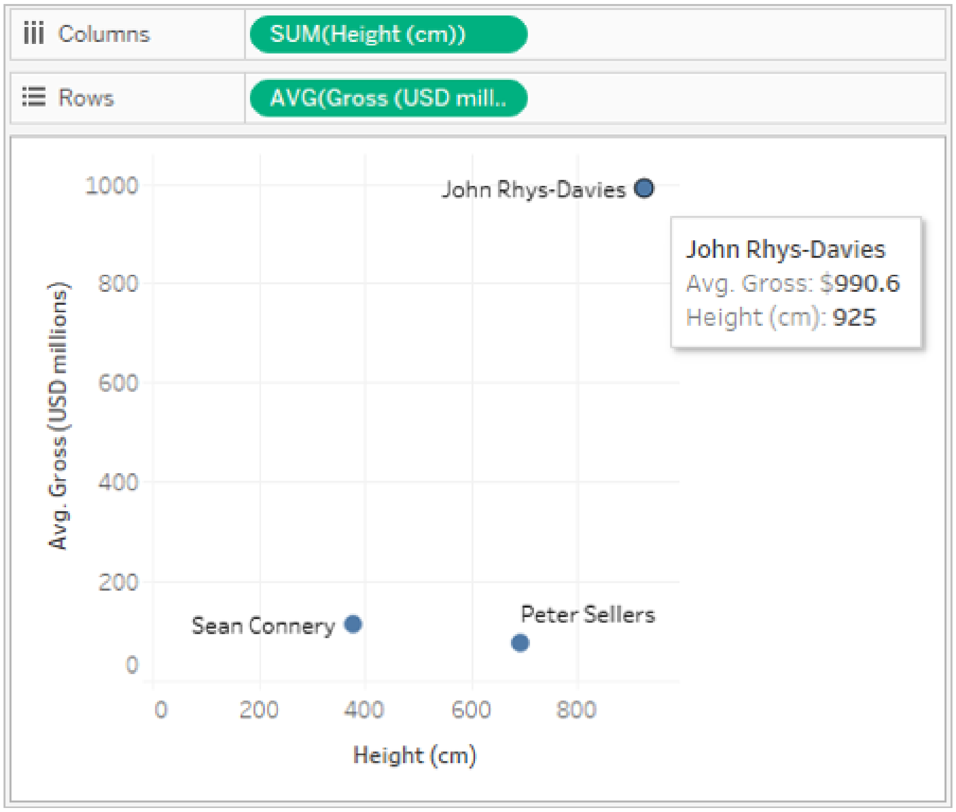
These tables can be joined together, joining the Appearances table with the Actors table on Actor = Actor and the Appearances table with the Movies table on Movie = Movie. If the joins are full outer joins so no rows are lost, the final output looks like this. Note that fields that are used in the join clauses appear twice.

Actor (Actors)	Height (cm)	Movie	Character	Actor	Movie (Movies)	World Premiere	Gross (USD millio...
John Rhys-Davies	185	The Fellowship of the Ring	Gimli	John Rhys-Davies	The Fellowship of the Ring	12/19/2001	869
John Rhys-Davies	185	The Return of the King	Gimli	John Rhys-Davies	The Return of the King	12/18/2002	1,119
John Rhys-Davies	185	The Return of the King	Treebeard	John Rhys-Davies	The Return of the King	12/18/2002	1,119
John Rhys-Davies	185	The Two Towers	Gimli	John Rhys-Davies	The Two Towers	12/17/2003	923
John Rhys-Davies	185	The Two Towers	Treebeard	John Rhys-Davies	The Two Towers	12/17/2003	923
Peter Sellers	173	Casino Royale	James Bond	Peter Sellers	Casino Royale	4/13/1967	22
Peter Sellers	173	Dr. Strangelove	Dr. Strangelove	Peter Sellers	Dr. Strangelove	1/29/1964	94
Peter Sellers	173	Dr. Strangelove	Group Captain Mandr...	Peter Sellers	Dr. Strangelove	1/29/1964	94
Peter Sellers	173	Dr. Strangelove	President Muffley	Peter Sellers	Dr. Strangelove	1/29/1964	94
Sean Connery	189	From Russia With Love	James Bond	Sean Connery	From Russia With Love	5/27/1964	24
Sean Connery	189	The Hunt for the Red October	Marko Ramius	Sean Connery	The Hunt for the Red October	3/2/1990	200
Jodie Foster	161	null	null	null	null	null	null
null	null	null	null	null	Casablanca	11/26/1942	4

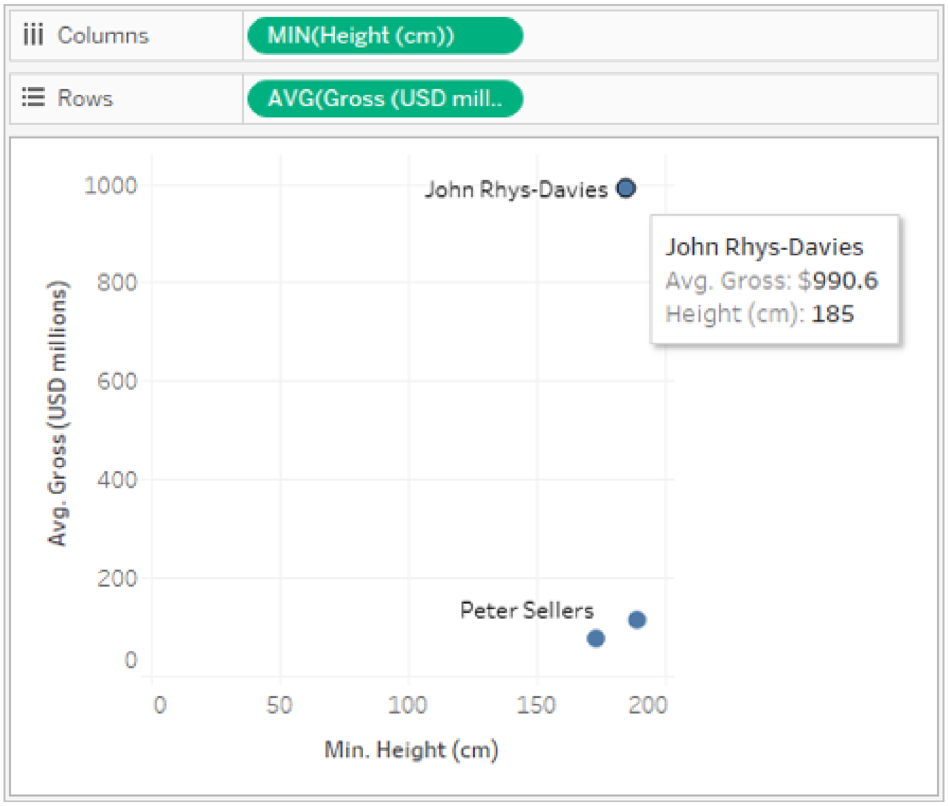
Data that is brought together into a single table in this way is called denormalized or flattened.

This joined data is flattened data. Each row consists of an actor's appearance as a character in a movie (so John Rhys-Davies has two rows for The Return of the King because he played two characters); therefore the granularity of the data is at the level of a character in a movie. Information that is relevant in multiple rows is duplicated. The Return of the King's premiere date is present twice because there are two characters from that movie in the data set. John Rhys-Davies' height is listed five times because there are 5 unique combinations of character/movie for him as an actor.

This flattened data therefore has some characteristics to be aware of. For example, if you wanted to plot actor height by the average gross of their movies, you might assume we could bring out Height to columns and Gross to Rows, then take the average of the Gross. But if you do that the default view doesn't seem right. Here John Rhys-Davies is supposed to be 925cm tall, which is over 30 feet!



This is because the default aggregation is SUM. There are 5 rows in the data for him, so we get his true height of 185cm five times. You could fix this by changing the aggregation on Height, say to average or minimum. This effectively returns the value for only one row (since they are all the same).

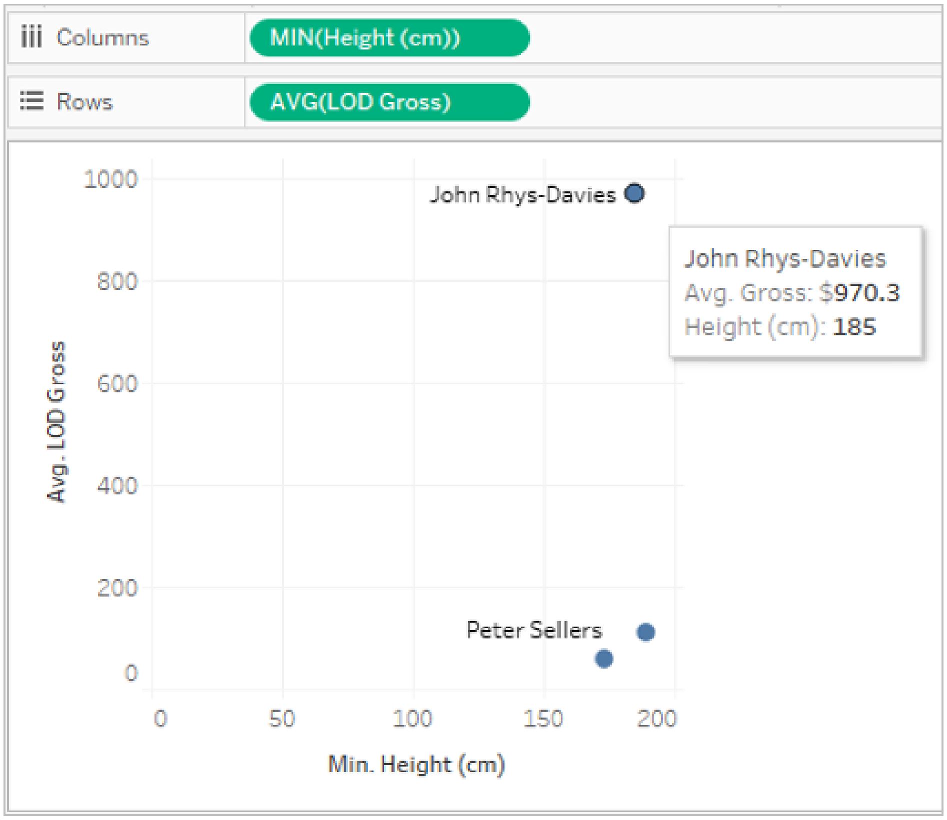


When you change the aggregation the heights are much more realistic. But now you need to pay attention to average gross. Remember that Tableau considers all five rows for John Rhys-Davies' height. When you think about the average gross for the movies he was in, it shouldn't be the average of five rows, but rather the average of three movies. You don't want to count the gross for Return of the King twice just because he played two characters in it. But is that what is happening?

Doing some quick math, the Lord of the Rings movies should have an average of (869 + 923 + 1119)/3, or $970.3. However, the value in the scatterplot is $990.6. The current average is from the five rows (869 + 923 + 923 + 1119 + 1119)/5.

This isn't as easy to fix as the height issue with changing the aggregation. You need to use a Level of Detail (LOD) expression to alter which level of detail Tableau looks at, going from the default—Appearance—up to the level of Movie. You could create a calculation for LOD Gross as {FIXED [Movie] : MIN([Gross (USD millions)])} and then take the average of that new LOD Gross field for our view.

The LOD expression can be read as “for each movie, return its minimum Gross”. This removes the duplication problem because the gross is always returned per movie, even though the view is constructed with movie and actor.

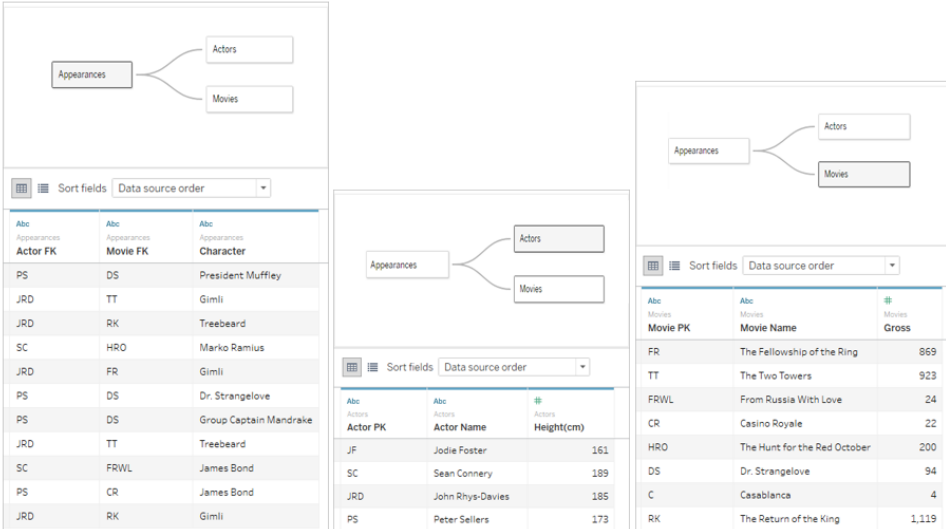


Now the numbers are correct. John Rhys-Davies is 185cm tall, and the average gross for his movies in this data set is 970.3. You would need to understand how data was being replicated and how Tableau aggregated the data to display it before you could ensure the correct values were returned.

Normalized data in multiple tables

Creating relationships between logical tables may seem similar to creating joins, but instead of flattening the data into a single table, with all the replications that can entail, Tableau stays aware of the relationships between the tables. Information is brought out of each table at the appropriate level of detail and related to other data.

On the **Data Source** page, you won't see the “full” grid view of the flattened table. It doesn’t exist. Tableau keeps all three tables as they are and merely establishes the relationships, pulling together the required data as needed by the view.

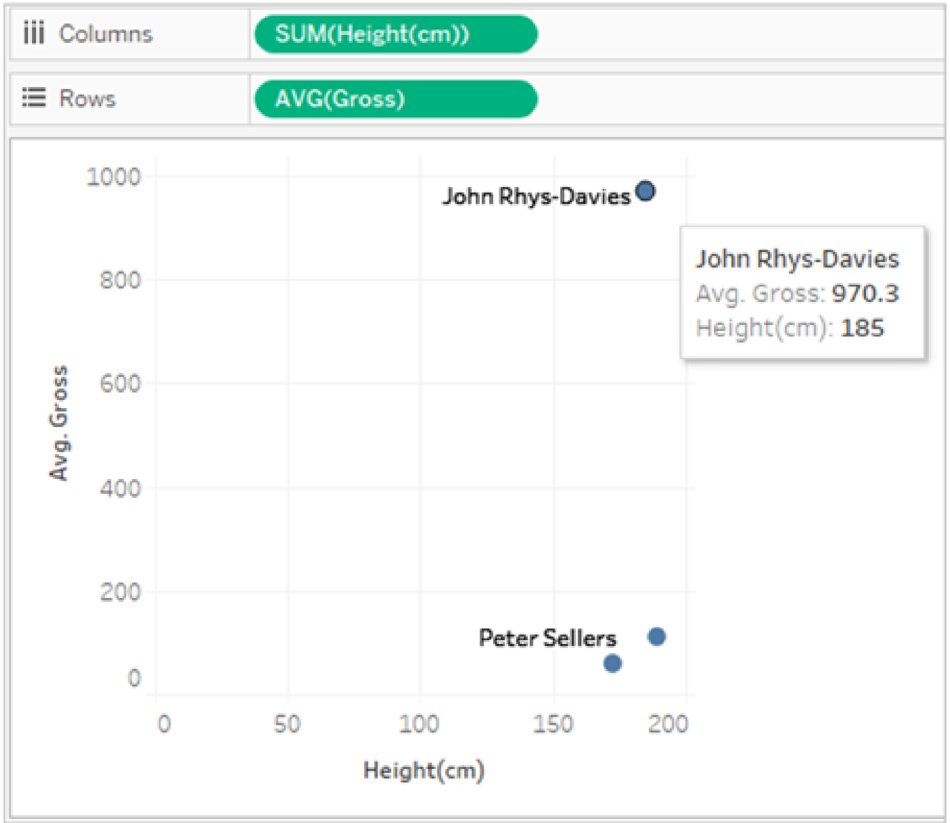


Actor FK	Movie FK	Character
PS	DS	President Muffley
JRD	TT	Gimli
JRD	RK	Treebeard
SC	HRO	Marko Ramius
JRD	FR	Gimli
PS	DS	Dr. Strangelove
PS	DS	Group Captain Mandrake
JRD	TT	Treebeard
SC	FRWL	James Bond
PS	CR	James Bond
JRD	RK	Gimli

Actor PK	Actor Name	Height(cm)
JF	Jodie Foster	161
SC	Sean Connery	189
JRD	John Rhys-Davies	185
PS	Peter Sellers	173

Movie PK	Movie Name	Gross
FR	The Fellowship of the Ring	869
TT	The Two Towers	923
FRWL	From Russia With Love	24
CR	Casino Royale	22
HRO	The Hunt for the Red October	200
DS	Dr. Strangelove	94
C	Casablanca	4
RK	The Return of the King	1,119

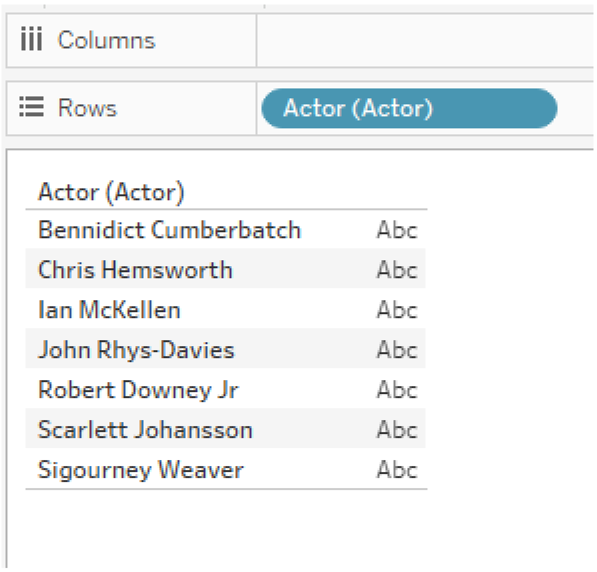
To create the same scatterplot, drag Height and Gross into the view, and set Gross to average. That's it! Tableau looks at the way each table's data is related to data from other related tables and infers how the height should be displayed (by actor) and how the average gross should be computed (by movie).



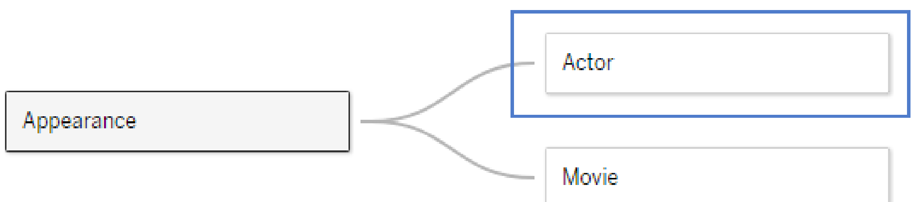
Example 2: Dimension from a single table

If the dimensions in your visualization come from a single table, Tableau queries just the one table and shows results for the entire domain. You can add measures and still see the entire domain.

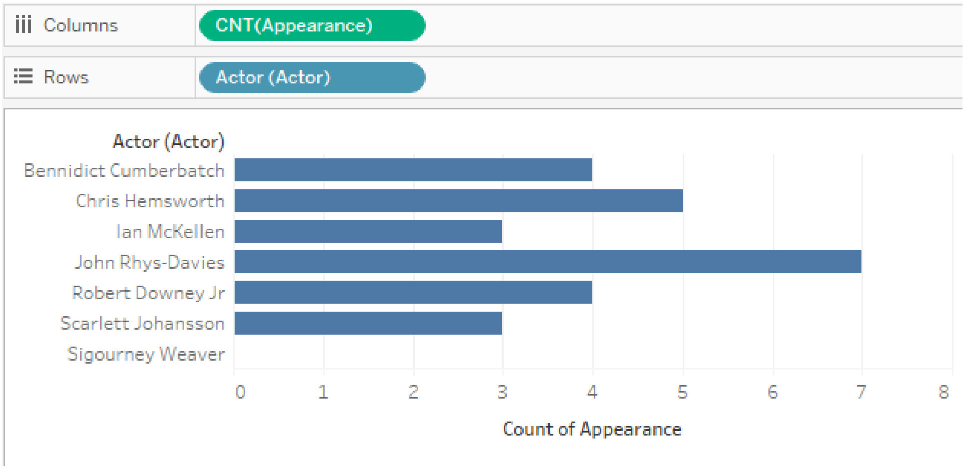
For example, using the Movie Appearances data source introduced above, adding the Actor field to a visualization results in the following visualization:



Because the only dimension in the viz came from the Actors table, Tableau will run a query against just the Actors table. All actors that appear in the Actors table are displayed in the viz, regardless of whether they have any Appearances.



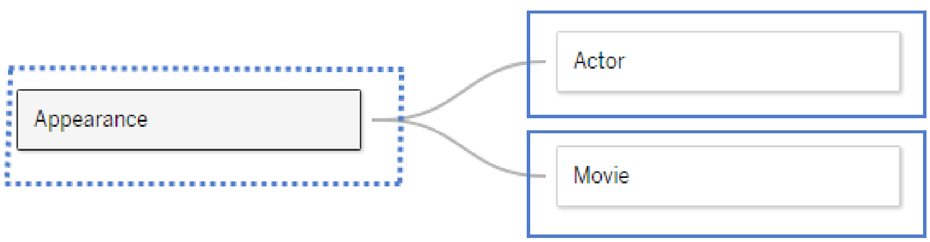
Bringing the Appearance Actor field into the view as a measure, and then applying a COUNT aggregation creates a view that shows the number of appearances by actors. Notice that Sigourney Weaver does not have any appearances, but her name is still in the view.



Example 3: Dimensions from multiple tables

If the dimensions in the view are from multiple tables, Tableau finds the table that relates all the dimensions and shows the domain from that table. As a result, some dimension values that you saw in Example 1 change.

For example, dragging a field from the Movies table into the viz changes the query. Because the Movies and Actors tables are related by the Appearances table, the query returns only the Actor/Movie pairs that exist in the Appearances table.

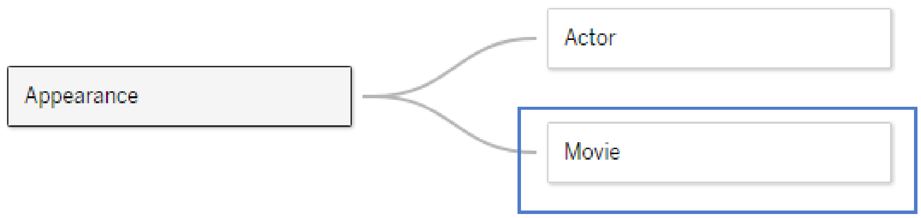


Because Sigourney Weaver does not have any Appearances in this data set (and, thus, is not related to any of the Movies in the data set), the viz of Actor/Movie pairs does not show her:

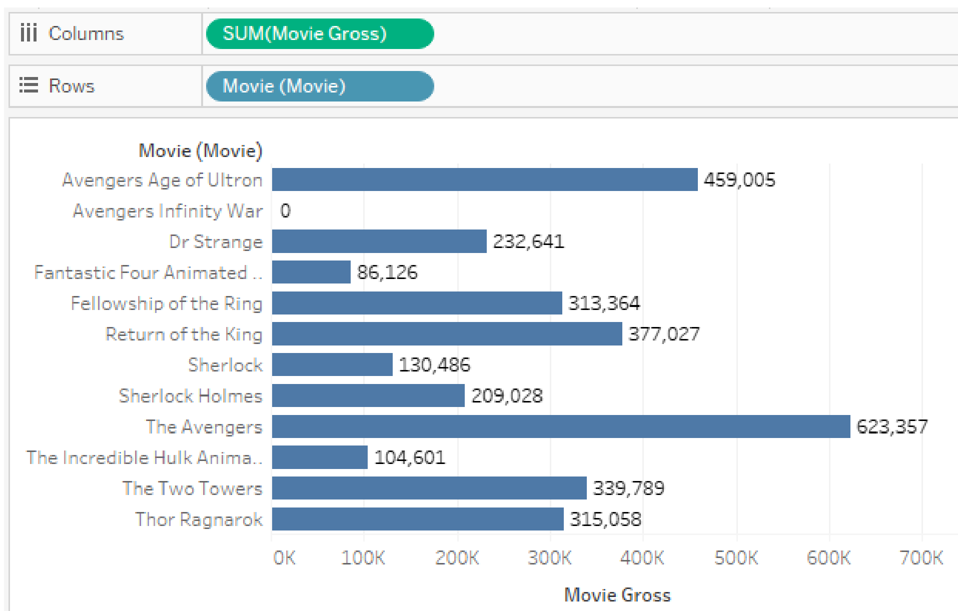
Example 4: Measures that can’t be split by a dimension

If a measure can't be split by a dimension, Tableau replicates the measure across that dimension.

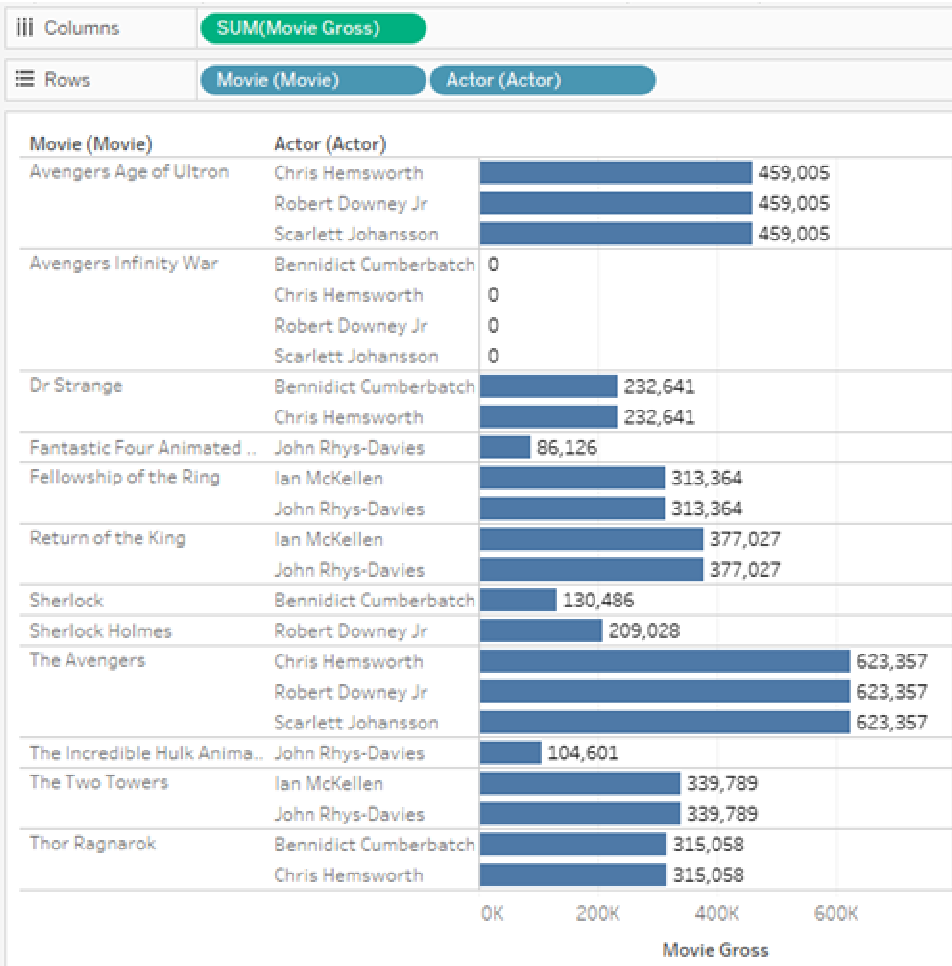
The next visualization shows the total amount grossed by Movie. Since both fields come from the Movies table, Tableau will query only the Movies table.



The Movies table includes the already aggregated Gross receipts for each movie, as shown in the following visualization (the gross for Infinity War was not available at the time this data set was created and is shown as zero).



If you add Actor to this visualization, Tableau knows that it cannot break down the Movie Gross receipts by Actor, since this more granular information is not available in the data model. Instead, Tableau shows the Movie Gross receipts for each movie, replicated across actors.

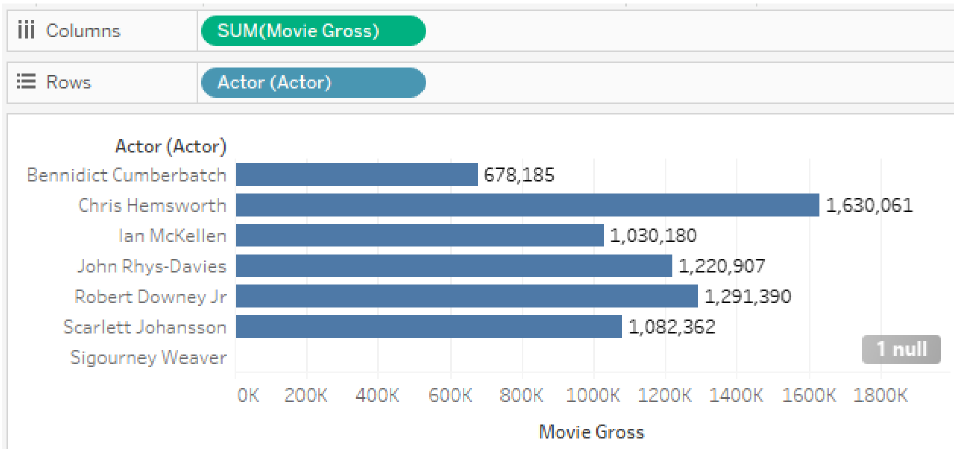


Example 5: Dimension that isn't hierarchically related to a measure

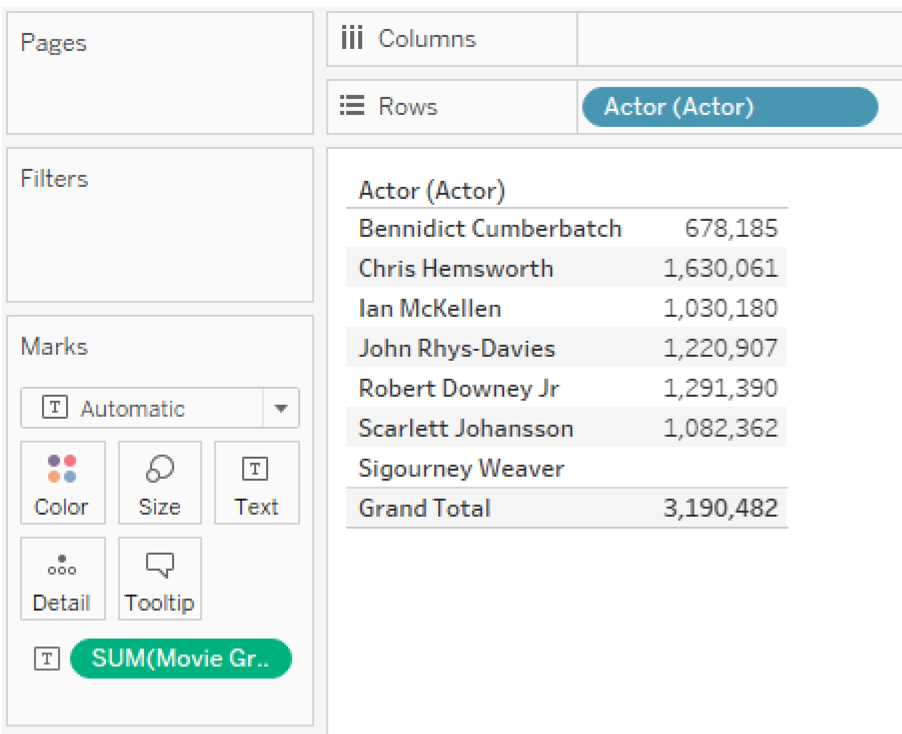
Removing the Movie dimension from the previous visualization queries the sum of the Movie Gross measure for each Actor. The result is the aggregated Movie Gross for every movie the actor appeared in.

In this case there isn't a hierarchical relationship between the dimension, Actor, and the measure, Movie Gross—multiple actors may have appeared in the same movie. For example, Benedict Cumberbatch and Chris Hemsworth both appeared in Dr. Strange. In this case, Tableau includes the Movie Gross for Dr. Strange in both actors' totals.

Because the same Movie Gross value is included in multiple actors' totals, Tableau won't directly sum over these values.



However, when totals are shown for this visualization, note that Tableau correctly computes the total Movie Gross without including duplicate movies.



Troubleshooting multi-table analysis

The following situations are possible with multi-table, related tables. This table describes known scenarios and high-level descriptions for how to troubleshoot your analysis.

Analysis Situation	Description
Inner join for dimension-only vizzes	You may not immediately see all of the values you were expecting when you add multiple dimensions to the same view. Or, you might notice that adding a new dimension to the viz from a different table causes some values to disappear from the viz. Tableau will use queries that preserve the combination of values that actually exist in the data. This means that you will see rows that would be produced by an inner join of the tables providing dimensions to the viz. If you want to see partial combinations of the dimension values, you can either turn on Show Empty Rows/Columns to see all of the possible rows, or you can add a measure, such as <*MyTable*>(Count), from one of the tables represented in the view to ensure you see all values of the dimensions from that table.
Constants in calculations	In a multi-table datasource, constant values behave as though they come from their own table with a single row. If you aggregate a constant value, it will behave as though the aggregate was on a single row. Sum(10) will always equal 10. Avg(10) will also always equal 10. Count (10) will always equal 1. To ensure backwards compatibility, constant values on a single-logical table data source will behave as though the constant value is replicated for each value in the table. Constants in row level calculations do not change the row level of detail of the calculation. The calculation [Sales] + 10 will behave as though it came from the same table as the [Sales] field.
Forcing outer join	Tableau ensures that all measure values are represented in the viz (along with all combinations of dimen-

Analysis Situation	Description
	sion values that actually occur in the data) so if you want to ensure you are seeing all of the possible values in your data including "unmatched nulls" you can do so by bringing measures into the view from each table in the worksheet.
I'm not seeing the measure values I would expect from a cross-table calculation	The domain of a calculation is the inner join of its inputs. If there are not matching values from all of the inputs of the measure calculation they will not be included in the measure calculation. Consider using an LOD calculation to move the measure values to the same object before creating the row-level calculation.
Calculation to switch between fields from different logical tables gives unexpected results	If you have a calculation that switches between row level fields, either using a case statement, if statement, or a function like "IFNULL", you may see unexpected results, because this calculation is being computed for every row, where row is the inner join between the inputs of the calculation. A better approach is to switch between Aggregated values, rather than trying to switch within the row level calculation. This will also produce better performance in single table scenarios. Alternately, this is only an issue for calculations that span between tables, so using LOD calculations to bring all of the fields to the same table will work as well. Don't do this: `SUM( IF [Parameter] == "Foo" THEN [Field 1] ELSE [Field 2] END )` Do this:

Analysis Situation	Description
	`IF [Parameter] == "Foo" THEN SUM([Field 1]) ELSE SUM([Field 2]) END`
Unexpected unmatched nulls	You may see measure values associated with a null dimension value that you didn't expect. This can possibly be a sign that the relationships in the data source were incorrectly configured. It may also indicate that there are actually unmatched values from the table that contains the measure, that have no corresponding row in the dimension table. In the past, this data might have been lost if the wrong join type was selected. When you use relationships, these unmatched values are preserved. If you don't want to see the unmatched values, you can exclude them using a filter.
Incorrect aggregate values	Are you using relationships or joins? With relationships, aggregates are computed correctly by default. When using joins, you may need to write LOD calcs to deduplicate values. Have you incorrectly set Performance Options on relationships? Try resetting the Performance Options to the default and see if that produces the correct aggregations.
Dimensions replicate measure values rather than partitioning them. Dimension filters don't subset measures.	Check that the fields used to define relationships are correct.
Tableau is generating too many queries or queries with many left joins	Check the logs or performance recording to see how many queries are generated and how many left joins are used. With the new data modeling capabilities, Tableau

Analysis Situation	Description
	generates queries with left joins and/or additional queries to make sure that unmatched measure values are always included in the visualization. If you don't need to see the unmatched values, use filters to remove the unmatched (NULL) values from your visualization. This should result in fewer queries. If you know that your data doesn't have any unmatched values, you can set the Referential Integrity setting for each relationship to "All Values Match" in Performance Options. This will also result in fewer queries. You can also reduce the complexity of your visualization to reduce the number of queries generated. Removing measures and hiding Filter controls are key ways to simplify queries multi-table, related data.
Queries have many subqueries	Check the logs or performance recording to see the complexity of the queries Tableau generates. Tableau automatically generates subqueries to deduplicate data, when necessary, to generate correct aggregates. This is similar to queries generated by LOD calculations. If you know the relationships between the logical tables in your data have many-to-one or one-to-one cardinality, then you can set this cardinality information in the relationship Performance Options. This will allow Tableau to eliminate unnecessary subqueries since it will know that no duplication can happen.
I was using joins to filter data	In 2020.2, Tableau will work hard to recover unmatched values. Sometimes that means it will use a left join where you might have specified an inner join to intentionally filter out data.

Analysis Situation	Description
	If you filter out the unmatched values that are brought in by this join, Tableau will be able to optimize the query back to an inner join. Depending on your specific scenario, it might make sense to model this inner join as a physical join within a logical table. This is especially powerful if a table containing measures is being used to filter a dimension table, as it will not introduce additional measure replication.

Don’t Be Scared of Relationships

The new data modeling capabilities create flexible data sources built around relationships.

Relationships combine data from different tables by looking at what columns (fields) those tables have in common and using that information to bring information from each table together in the analysis.

Unlike joins or unions, relationships form a data source without flattening multiple tables into a single table. Because of this, related data sources know which table each field is from. That means each field keeps its context, or level of detail. Related data sources can therefore handle tables with different granularity without issues of duplication or data loss.

In a related data source, the joins aren’t fixed up front. Instead of merging all the data (and having to work with all the data regardless of what each viz requires), only the relevant data is combined—per sheet and as necessary. As you drag and drop, Tableau evaluates the relationships of the relevant fields and tables. Those relationships are used to write queries with the correct join types, aggregations, and null handling.

You can think about how the data fits together and what questions you want to answer, rather than how to combine the data or compensate for artifacts from the data source.

Relationships don’t replace the previous ways of combining data: joins, unions, and blends. Rather, relationships are the new, flexible way to bring data together from multiple sources. The existing methods aren’t going away, and are still useful in specific scenarios.

Note: For related information on how relationship queries work, see these Tableau blog posts:

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Use Relationships

This topic walks through building a related data source and using one for analysis. If you would like to follow along, you can download the Bookshop data set. Choose Bookshop.xlsx for the raw data to start from scratch, or MinimalBookshop.tdsx to start with the related data source basics configured for you.

> **Note:** Relationships are available in Tableau 2020.2 and later.

Video: Migrated Data

If you open an existing workbook from a previous version of Tableau (version 2020.1 or earlier) in Tableau 2020.2 or later, your data source tab might show a **Migrated Data** table.

There are multiple videos embedded in this topic. For your convenience, a transcript is provided in the expandable section underneath each video.

Migrated Data video transcript

If you open a workbook created before relationships were available (before version 2020.2), your data source may look different. If there were joined or unioned tables, they appear as a single logical table named Migrated Data. Double click to see the underlying tables—they're still there, and your analysis will still behave as expected.

Note that when you're only using the Migrated Data table, the Data pane behaves more like previous versions of Tableau Desktop.

You can even relate new tables to the directly to the Migrated Data table to extend the data model.

Once you've added relationships to the data model, the Data pane updates to the new layout, with dimensions and measures maintained within each table, and the Migrated Data table is displayed as a single, merged table.

Video: Relate tables

Relate tables video transcript

To build a data source of related tables, connect to your data and drag a table to the canvas. Drag out a second table. A relationship will automatically form if it can. If Tableau cannot detect the related fields, you will be prompted to select them yourself.

The canvas you're seeing is a new layer of the data model where you can relate tables together. Dragging out additional tables adds them to the data model. The noodle shows which tables are related.

Relationships can be based on calculated fields. You can also specify how fields should be compared by using operators when you define the relationship.

Add joins

You can create a join by double clicking on a table to open the join canvas (also referred to as the physical layer), and then dragging out a table to create a join. Creating joins is the same experience as previous versions of Tableau—you can set a join clause, including a calculation if necessary, and configure the join type.

Once you close the join canvas, the table on the relationship canvas (also referred to as the logical layer) has a join icon to indicate the underlying join. This is because a join merges its tables together to form a single, new table. Related tables remain distinct.

Add unions

Add unions by dragging out New Union. Unions, like joins, actually merge the data, and show an icon on the logical layer. We recommend always renaming unions—or any tables with unclear names—to something more useful.

Video: Performance Options

Performance Options video transcript

For any relationship, there are performance options as well as related fields. Tableau automatically chooses safe defaults for these options. However, if you are sure of the characteristics of your data, changing these options can give Tableau more information about how to best configure automatic joins during analysis. Setting the cardinality and referential integrity options gives Tableau more information about how to optimize queries.

Cardinality

Cardinality indicates whether or not the linking field records are unique. If multiple books can have the same author, but each author is listed only once in the author table, the cardinality for AuthID would be Many for Books and One for Author between those two tables. This setting impacts how Tableau handles aggregation before or after the join.

Referential integrity

Referential integrity indicates whether or not a record in one table is guaranteed to have a relationship in another table. If every book has an author, Book has referential integrity to Author. If not every author has a book, Author does not have referential integrity to Book. This setting impacts how Tableau chooses a join type and handles unmatched records.

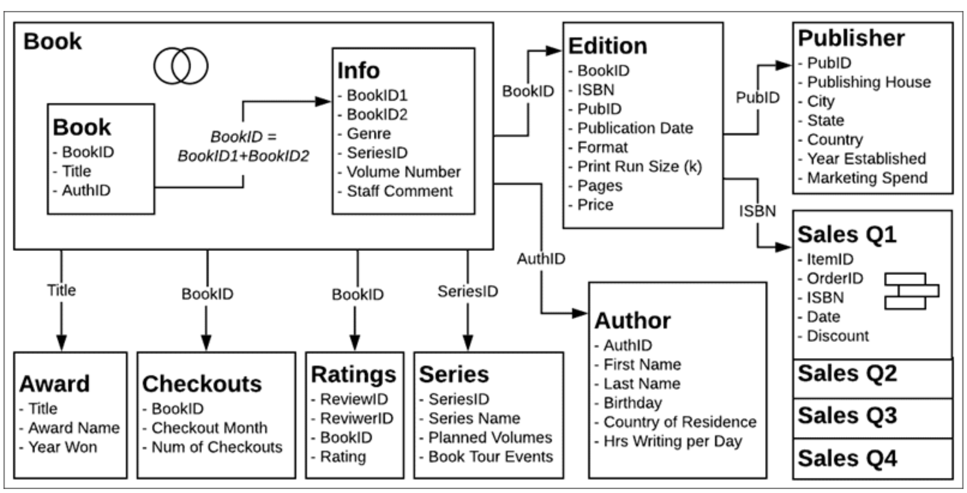
For more information, see **Cardinality and Referential Integrity** on page 829.

Example: Bookshop Data

We'll do some analysis with this data source in a moment, so let's talk about the data.

The data set is around (fictitious) books, and it's important to consider the distinction between a book and an edition. A book is the conceptual work itself, with attributes such as the title, author, and genre. There are also editions of a book, with attributes such as a price and format (hardcover or paperback), that are identified with an ISBN. An edition of a book has a publisher and a page count, whereas a book may have won an award or be in a series.

You can follow along by downloading MinimalBookshop.tdsx, or be adventurous and build the model yourself from Bookshop.xlsx. Note that you can hide many of the ID fields once the tables are combined.



Video: Work with multiple tables

Work with multiple tables video transcript

The Data Source page

Before we go to a sheet tab, notice that the data grid view, below, shows only the fields from one table at a time. If we click to another table, we'll see the data it contains. There is no overall view because related data is not brought together until it's used in an analysis. For now, the data remains in each table and is therefore displayed by table.

If we click on a table with an underlying join or union, we do see that merged data, with any nulls that may involve.

The Data pane

When we click onto a sheet, you may notice that the Data pane looks different than in previous versions of Tableau. Instead of the primary organization of dimensions and measures, it's now focused on tables. Each table has its own breakdown for dimensions and measures, indicated by this line. It's possible to still group by folders instead, if you choose.

Number of Records and Count

There is also no **Number of Records** field. This is because the concept of Number of Records has changed. There isn't a single Number of Records for the data source. Each table has a **Count** field, which can be thought of as a local Number of Records for that table.

To get a sense of the number of records for multiple tables at the same time, you can use Measure Names and Measure Values or multi-select the count fields and use Show Me to create a viz.

Video: Basic calculations

Basic calculations video transcript

There are a few calculations we can do to make this data set a bit friendlier. First, we can create a calculation for the full author name that brings together first name and last name. We'll create a calculation called **Author Name** that is simply first name, a space, and last name. Note that this calculation stays in the Author table because it uses fields only from that table.

```
[First Name] + " " + [Last Name]
```

When we make a sale, the sales amount is a combination of the price—fixed by the publisher and specific to the edition, or the ISBN—and any discount at the time of the sale. So the calculation **Sales Amount** will be price times one minus the discount. Note that we're using the ZN—or Zero Null—function to handle sales without discounts so we don't get nulls. This calculation goes to the shared area at the bottom of the Data pane as it uses fields from multiple tables. Aggregated calculations will also go to the bottom of the Data pane.

```
[Price] * (1-ZN([Discount]))
```

Video: Sets and groups

Sets and groups video transcript

We'll create a set for books to identify if they're in a series. In the Book table, right click on **Title** and select Create > Set. We'll name it **In a series**. On the **Condition** tab, choose **By field** with **Series Name**, **Count**, as >= 1.

In the Edition table, the Format field is quite detailed, including specifying mass market and trade paperback. Sometimes that detail is useful, but sometimes we want to generalize between hardcover and paperback. Right click on **Format** field in the Data pane and select Create > Group. We'll name it **Cover**. Control-click to select everything except Hardcover and then click **Group**. The name of the group is automatically highlighted, and we can call it **Paperback**.

Like with the Author Name calculation, these items we just created are relevant within a single table and therefore stay within those tables in the Data pane.

Analyze related data

Let's do some comparisons of how analysis looks between relationships and joins. We recommend that you download the companion workbooks for Question 1 and Question 2 rather than continue with your own data source from above. If you identify issues with the data source configuration or steps for Question 1, note that they are intentional for the purposes of the scenario.

Question 1: How many editions are there for each book?

Important: The video for Question 1 was recorded on beta software. The released version of 2020.2 correctly handles our attempt to introduce an error in the related data source. If you follow along, your workbook will not match the video—the relationships values will be correct. Please bear with us while we attempt to outsmart our product and force an error for educational purposes.

We'll set this up by doing some basic analysis with both data sources. We'll build a bar chart with each one, looking at the number of editions by title.

Related

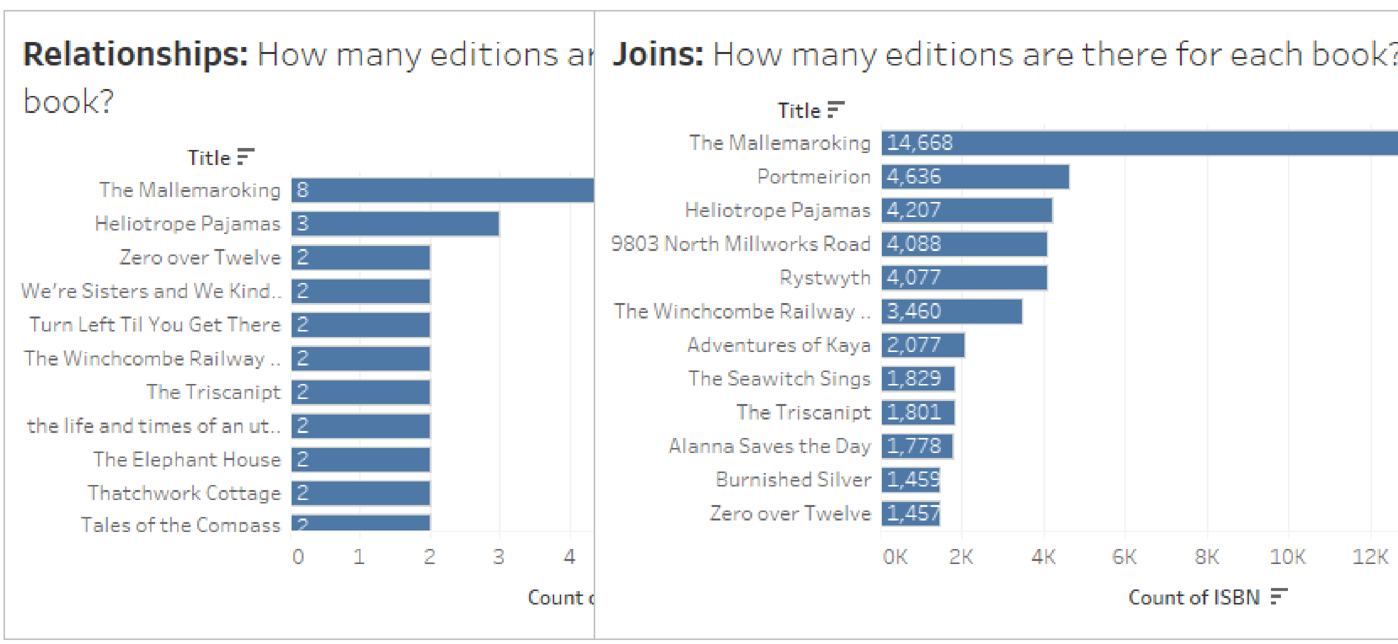
Using the related data source:

1. Bring Title to Rows
2. Bring Edition (Count) to Columns
3. Click the Label icon in the toolbar to turn on labels
4. Click the Sort Descending icon in the toolbar

Joined

Using the joined data source:

1. Bring Title to Rows
2. Bring ISBN to Columns
3. Right Click on the pill and select Measure > Count
4. Click the Label icon in the toolbar to turn on labels
5. Click the Sort Descending icon in the toolbar



It’s immediately clear that Count is not the right aggregation for the joined data. However, something isn’t right with the related data, either. Let’s view the data and see what’s going on and how to fix it. Watch the video (or read its transcript) to see how.

Video: Data validation and fixing the data model

Question 1 video transcript

Joined

Here we’re looking at our joined data. On the joined viz, we’ll take a look at a mark’s underlying data.

This view data dialog has two tabs, Summary and Full Data.

- There are tons of rows for the same book—this is likely duplicated because of all the joins.

To handle the duplication, we’ll change the aggregation on the ISBN field in the view to Count Distinct. Remember that we’ll need to think about this duplication in almost every analytic scenario and address it with the correct aggregation, filtering, or how we structure the viz.

Related

Let’s take a look at the structure of our related data source, and we’ll take a peek at what’s going on with our join as well. On the related viz, we’ll take a look at a mark’s underlying data.

And we see that the view data dialog again has two tabs, but this time it’s the Edition tab instead of Full Data. Every table used in the viz will have its own tab.

- There’s a doubling of rows for this book—and we know that joins can cause duplication, and that the Book table in this data source is really the Book table joined with Info and Award.

We can create a viz with Title and Award Name and scroll down to investigate. And sure enough, The Mallemaroking has won two awards. It would be better to relate these tables than to join them.

We’ll go back to the Data Source tab and remove the Awards table from the join and relate it instead. And now we don’t have to worry about the duplication issue any more for this data source.

Question 2: For authors with books in a series, who is has the most book tour events?

Video: Work with different levels of details

Question 2 video transcript

Let's look at which authors we'll be working with. Using the related data source, we'll bring **Author Name** and **Series Name** to Rows.

Because related data sources pare down unmatched data in tables that only contain dimensions, we're focusing on the data that is relevant to us. We can see that there are seven authors, two of whom have written for two series. If you don't want to see only matched values, you can restore those unmatched values (which is more like the legacy behavior) by going to the Analysis menu > Table Layout and checking Show Empty Rows.

Related	Joined
Using the related data source: 1. Bring the set **In a series** from the Book table to the filter shelf. The default is to filter only to members in the set. 2. Bring **Author Name** to Rows. 3. Bring **Book Tour Events** to Columns.	Using the joined data source: 1. Bring the set **In a series** to the filter shelf. 2. Bring **Author Name** to Rows. 3. Bring **Book Tour Events** to Columns.

Our numbers here look a bit off.

For the joined data, we know there's duplication because of the join, and we also know some authors span series. Because of this, we can't just change the aggregation to, say, MIN or MAX, because we lose information for authors with works in multiple series.

What we really want is the number of events per series, visualized by author. This is a classic case for Level of Detail (LOD) expressions. We'll create a calculation **Series Events**:

```
{FIXED [Series Name] : MIN ([Book Tour Events])}
```

Note that the MIN is to handle duplication of events for a single series.

Now if we bring this new field to Columns instead of the original events field, we'll get the correct values.

For the related data, we didn't need to do any of that. Relationships are smart enough to understand the native level of detail and the way the Author table relates through the Book and Info tables through to the Series table, and to correctly join and aggregate the Events measure back to Author Name—all without having to write LOD calculations.

So don't be scared of relating your data. See for yourself! You can download the Bookshop data or use some of your own. Try filtering, using table calculations, building a variety of chart types, configuring performance options, and pushing relationships as far as you can.

Tips when working with related data

- **View Underlying Data** on page 2728 often to verify what data a mark represents.
- If you aren't required to join—and there are reasons you may need to—using a relationship provides more flexibility.
- If you don't want to see only matched values, you can restore those unmatched values by going to the Analysis menu > Table Layout > Show Empty Rows.

Continued reading

Ready to tackle calculations with Relationships? Check out **Don't Be Scared of Calculations in Relationships** on the facing page.

Ready to keep exploring how to do complex analysis with Relationships? Check out **Don't Be Scared of Deeper Relationships** on page 874.

For more information on the technical underpinnings of relationships straight from the Product Management team, check out the series on relationships on the Tableau Blog.

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Don't Be Scared of Calculations in Relationships

Calculations can be intimidating. Calculations in a data source that uses relationships can seem more intimidating. But there's no reason to be scared of calculations in relationships.

> **Note**: if you aren't yet comfortable with the principles behind data sources that use relationships, it might help to read **Don't Be Scared of Relationships** on page 861 before diving into this topic.

Calculation types

A **row-level calculation** is something that is computed for each record in the data. For example, each sale transaction has a sales amount, the price, potentially adjusted by the discount percentage: `Price * (1- ZN([Discount]))`. This is computed row-by-row for each transaction—the results can be thought of as adding a new column in the data source for Sales Amount.

There are also **aggregate calculations**. Aggregate calculations are computed at the level of detail of the visualization and their value depends on the structure of the view. Counts are an example of an aggregate calculation. The value of `Count([Title])` depends on whether we're looking at the count by format, author, or sales day.

See our blog for more information about types of calculations.

Table level of detail

Because each table in the Tableau data source can have its own level of detail, which table a calculation is associated with has a large impact. For example, historically in Tableau (pre 2020.2, before relationships), it was possible to count the number of records in the data source by creating a calculation with the constant value `1` and summing it. `1` was assigned to each row in the data source so the sum was equal to the number of rows.

Now, however, that calculation would have a value of `1`. There is no global level of detail for the entire data source, and a constant calculation lives in the unassigned area at the bottom of the Data pane. It has its own level of detail. Summing `1` is simply `1`.

As mentioned previously, calculations that have the same level of detail as a table—usually because they contain a field from the table—belong in that table in the Data pane. For example, the concatenation of First Name and Last Name to get the full name of each author (that is, `[First Name] + " " + [Last Name]`)is placed in the Author table.

However, if we were to use fields from multiple tables in one calculation, the calculation would go to the unassigned area at the bottom of the Data pane. This is called a cross-table row-level calculation; it requires a row-level join of the tables involved, which can impact performance. (Make sure the correct cardinality is set in the Performance Options of the relationship).

Assigning the level of detail

Because the level of detail of a table controls what a calculation's results represent, it's important to have calculations in the appropriate tables. A FIXED Level of Detail expression can be used to pull a calculation into a specific table. The field the expression is fixed to (the dimension declaration) determines the level of detail of the results.

Example: Author book tours

Q: How many book tour events is each author participating in?

A: This number should "belong" to each author, so it should go in the author table.

- **Author Book Tour Events** = `{FIXED [Author Name] : SUM([Book Tour Events])}`
- (This can be thought of in plain language as "for each author, total the number of book tour events associated with them")

Q: How many authors are not doing any book tours?

A: It's tempting to try a calculation like

- `COUNTD(IF ISNULL([Book Tour Events]) THEN ([Author Name]) END).`
- (This can be thought of in plain language as "if Book Tour Events is null, then return the author's name. Count each unique author name")

However, this is a cross-table row-level calculation because Book Tour Events is from the Series table and Author Name is from the Author table. And cross-table calculations like that use inner joins, which means when there's no corresponding value from both tables, the row is dropped from the join results. Which in turn means we're asking Tableau to count something that doesn't exist. Note that if the two fields were in the same table, however, this would be a fine way to do it. There's nothing wrong with the calculation as it stands other than the fact it doesn't play well with the structure of the data source.

Instead, we need to indicate the presence or absence of book tour events for each author—and keep those results in the Author table. Once each author is flagged as "participating in a book tour" or not, we can count the number who aren't doing book tours.

- **Book tour participant?** = `IF {FIXED [Author Name]: COUNT([Book Tour Events])} = 0 THEN "No" ELSE "Yes" END`
- (This can be thought of in plan language as "For each author, count the number of book tour events. If that count is zero, label the author 'No', otherwise label them 'Yes'". By naming the field "Book tour participant?" the no's and yes's will line up.)

Now we can count the number of Nos and answer the original question.

If our calculation is **Authors without book tours** = `IF [Book tour participant?]= "No" THEN ([Author Name]) END`, we'll get a list of authors who aren't doing book tours. (This can be thought of as "for every author where Book tour participant? is no, list the author's name.")

If our calculation is `COUNTD(IF [Book tour participant?]= "No" THEN ([Author Name]) END)`, we'll get a numeric answer that will live in the unassigned area at the bottom of the data pane. Why? Because this calculation is aggregated.

Bonus Question:

It's also possible to use the structure of the Book tour participant? calculation directly to return the list of author names. `IF {FIXED [Author Name]: COUNT([Book Tour Events])} = 0 THEN ([Author Name]) END`. Where would this calculation go in the Data Pane? Why? Try it in your own copy of Tableau Desktop, or download this workbook to see. (Requires Tableau Desktop 2020.2 or higher. Once in the workbooks, cight click the calculation in the Data pane and select Edit to open the calculation editor and see an explanation in the calculation comments.)

Continued reading

Feeling a bit overwhelmed and want to go back a step? Try **Don't Be Scared of Relationships** on page 861.

Ready to keep exploring how to do complex analysis with Relationships? Check out **Don't Be Scared of Deeper Relationships** on the next page.

For more information on the technical underpinnings of relationships straight from the Product Management team, check out the series on relationships on the Tableau Blog.

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Don’t Be Scared of Deeper Relationships

Because of the nature of relationships, understanding where fields come from, their domain, and what nulls represent is more important than ever. Remember that anyone viewing the viz without access to the Data source page or the Data pane will be reliant on the title or other information presented by the author to interpret the viz correctly.

> **Note**: if you aren’t yet comfortable with the principles behind data sources that use relationships, it might help to read **Don’t Be Scared of Relationships** on page 861 before diving into this topic.

Why aren’t all date fields equal?

It’s important to remember that just because two field names contains the word “Date” doesn’t mean the contents of those fields are identical. Let’s look at an example that sticks with the level of Year. That is, we don’t care about the values for months and days, we’re only interested in what years the fields cover.

There are multiple fields in **The Bookshop data set** on page 358 with a data type of Date.

- Birthday
- Publication Date
- Sales Date
- And Year Won and Year Established also contain date information, though they are numeric fields.
    - NB: If the icon in the data pane is green, this field is continuous. Right click on the field name in the Data pane and choose Convert to Discrete. The icon should turn blue. Do this for both “Year Won” and “Year Established” if necessary.

But the domain—the values in each field—vary. Bring each field in turn to the rows shelf and look at the number of marks in the bottom left corner, removing each field before bringing out the next.

- YEAR(Birthday) has 27 marks, meaning the domain contains 27 different years.
- YEAR([Publication Date]) has 15 marks, so its domain consists of 15 years.
- Year Won has 11 marks
- Year Established has four marks
- YEAR([Sales Date]) has one mark

This difference in domain is very important to keep in mind. If we related the Edition and Award table on YEAR([Publication Date]) to Year Won, we could wind up with analyses that use an inner join and are therefore trimmed down to just the years when an award was won—thus dropping all books from years without awards. (Note that this is not the same thing as filtering out books without awards. Non-award winning books that were published in the same year as an award-winning book will be preserved. All books from years with no awards will be dropped entirely. The filtering action from the join is at the level of year, not book.)

Example: Publication Date and Birthday

If we bring Publication Date to Columns and Birthday to Rows, we get a table of Abcs. You can download the workbook and explore this table on your own. (Requires Tableau Desktop 2020.2 or later.)

	Publication Date														
Year of Birthday	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192
2120										Abc	Abc		Abc		
2122								Abc							
2129		Abc			Abc	Abc				Abc	Abc			Abc	
2130						Abc		Abc							
2133				Abc	Abc		Abc	Abc							
2141				Abc					Abc	Abc	Abc				
2143														Abc	Abc
2145	Abc	Abc						Abc	Abc	Abc					
2146					Abc	Abc									
2147	Abc	Abc	Abc	Abc											
2148		Abc													Abc
2149	Abc	Abc	Abc				Abc	Abc						Abc	Abc
2150	Abc					Abc		Abc	Abc	Abc	Abc				Abc
2151		Abc			Abc									Abc	Abc
2153					Abc										
2154										Abc	Abc				
2155				Abc	Abc	Abc		Abc	Abc		Abc	Abc	Abc	Abc	
2156		Abc		Abc		Abc	Abc		Abc	Abc	Abc	Abc			Abc
2157														Abc	Abc
2160												Abc	Abc		
2161													Abc		
2163												Abc			
2169								Abc	Abc						

Abc is simply a placeholder because there is no data other than the years. Tableau has no values to make marks with, but the Abc shows where marks could go.

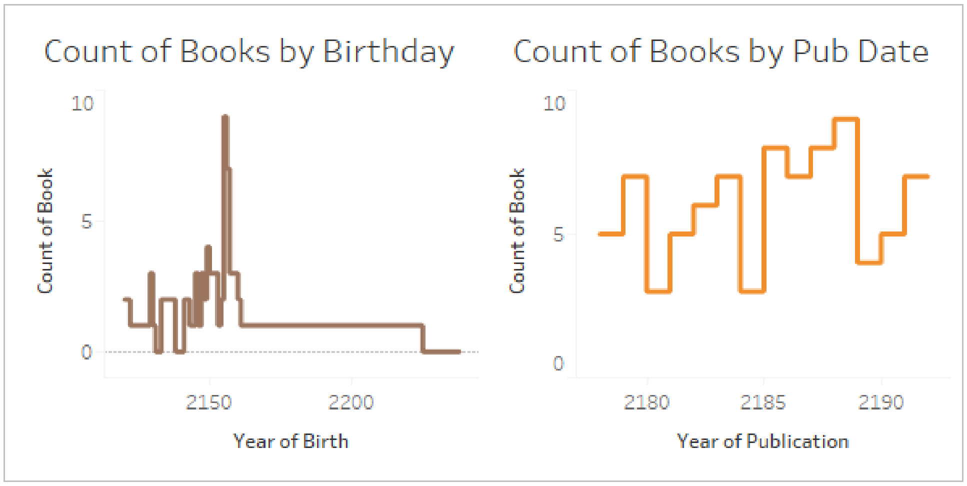
Notice that there is a lot of blank space in this table. There is no placeholder for the Publication year 2180 and the Birthday 2133. This means the author born in 2133 did not publish in the year 2180.

Sure. But why do we care?

Imagine you wanted to do some analysis over time. Are Hardcover sales increasing over time? You might imagine your graph as a timeline that filters Format to only Hardcover with Sales (Count) on Rows and date on Columns. But which date field? Publication Date? Birthday? Sales Date? These fields are well named and it's fairly obvious that if we have a question about sales we should use Sales Date. Not all data sets have clear names, though. If the fields were something like "Date1" and "Date3" it would be much harder to tell. Especially if the analytical question isn't clearly thought through.

What table a field comes from and what the domain of the field represents have a crucial impact on analysis.

By changing the date field, we can change the analysis entirely. Consider these two vizzes:



The first could be interpreted as "How many books were written by authors born in each year?". It answers questions like "Have older authors written more books?" (No) or "Author(s) born in which year are the most prolific?" (2155).

The second could be interpreted as "How many books were published each year?". It answers questions like "Which year had the most published books?" (2188) or "Is book publication steady over time?" (No).

It's awkward to phrase the question for the viz that uses birthday because it's an awkward combination of concepts. But Tableau doesn't judge and will let you ask whatever question you like—whether you mean to or not. That's a vital takeaway. If you bring out Date3 when you should have used Date1, Tableau will give you a viz. But not all date fields mean the same thing and it's up to the author to know which field to use for the correct analysis.

For more information about the importance of what table a field comes from, check out this blog post.

What does missing data mean?

There’s a difference between zero and null.

Zero means we measured and something isn’t there. We know the value and it is zero. If I have zero speeding tickets, I probably don’t speed much.

- NB: 0 and 1 are often used interchangeably with True/False, or other binomial values like Yes/No or Pass/Fail. In this case, zero is being used as a label, not a numeric value.

Null means we don’t know: we haven’t measured or we didn’t record the data. If my driving record is simply blank for speeding tickets, we have no idea if I speed or not.

Nulls can represent either missing data or nonexistent data.

- For my driving record, if the speeding tickets value is null, it’s conceivable that I could have a speeding ticket that wasn’t recorded. We should assume that data is missing.
- Nonexistent data could be recorded as N/A, but is often simply not recorded because we don’t need to track impossible things. How many speeding tickets did I get while riding a bus? That data likely won’t be present in the data set. In a matrix of “Modes of transportation” and “speeding tickets”, there are combinations that simply don’t make sense. We may assume that data is nonexistent.

When nulls carry meaning

It takes knowledge of the data to be able to recognize if a null is truly unknown (lack of information for number of speeding tickets) or if it represents nonexistent data (lack of information about speeding tickets as a bus passenger). Is the data trustworthy enough that a lack of information can be taken as nonexistent data rather than missing data? When subject matter expertise is applied, nulls can be meaningful.

Looking again at the table of Abcs, we can analyze the spaces without placeholders. We’ll assume this data is reliable and a null means the data is nonexistent rather than incomplete.

	Publication Date														
Year of Birthday	2178	2179	2180	2181	2182	2183	2184	2185	2186	2187	2188	2189	2190	2191	2192
2120										Abc	Abc		Abc		
2122								Abc							
2129		Abc			Abc	Abc				Abc	Abc			Abc	
2130						Abc		Abc							
2133				Abc	Abc		Abc	Abc							
2141				Abc					Abc	Abc	Abc				
2143														Abc	Abc
2145	Abc	Abc						Abc	Abc	Abc					
2146					Abc	Abc									
2147	Abc	Abc	Abc	Abc											
2148		Abc													Abc
2149	Abc	Abc	Abc				Abc	Abc						Abc	Abc
2150	Abc					Abc		Abc	Abc	Abc	Abc				Abc
2151		Abc			Abc									Abc	Abc
2153					Abc										
2154										Abc	Abc				
2155				Abc	Abc	Abc		Abc	Abc		Abc	Abc	Abc	Abc	
2156		Abc		Abc		Abc	Abc		Abc	Abc	Abc	Abc			Abc
2157														Abc	Abc
2160												Abc	Abc		
2161													Abc		
2163												Abc			
2169								Abc	Abc						

The lack of a placeholder means an author was born that year (so the row exists), and a book was published that year (so the column exists), but the published book was not written by an author born in that year (so the cell is empty). We can meaningfully assign a label to that empty space: Nothing Published. We could even do analysis on those nulls—for example, for the dates present in the data, how many years went by for each author without publishing a book?

> **Note**: There are also gaps in the domain of Birthdays. No authors were born in 2131 or 2132, so Year goes from 2130 to 2133 (the rows for 2131 and 2132 do not exist). These missing years can be interpreted as “no books in this data set had authors born in those years”. However, as pointed out above when talking about the domain of the date fields, the fact there are missing values in the domain is important information to consider when building relationships or choosing which field to use for a header or axis in a viz.

For more information about working with nulls in relationships, check out this blog post.

Hands-on exercises

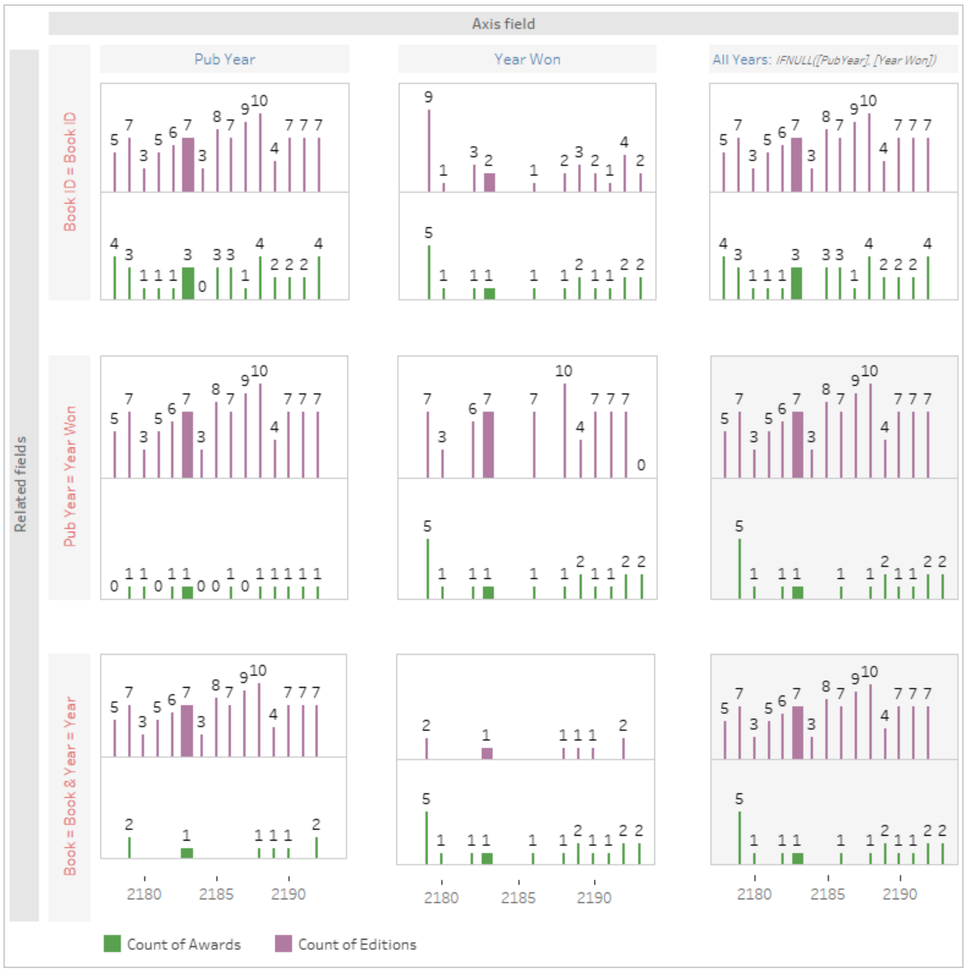
Write a title for each viz. Can you describe what is happening in plain language? Download the workbook to explore the live vizzes. (This workbook uses a slightly tweaked version of **The Bookshop data set** on page 358 and uses only two tables. The date fields have been adjusted so they can be used as the related field.)

To read the matrix of vizzes:

- The bars show the number of editions (purple) and number of awards (green) per year.
- Each column has a different date field for the axis. The left-hand column is Publication Year from the Edition table, the center column is Year Won from the Award table, and

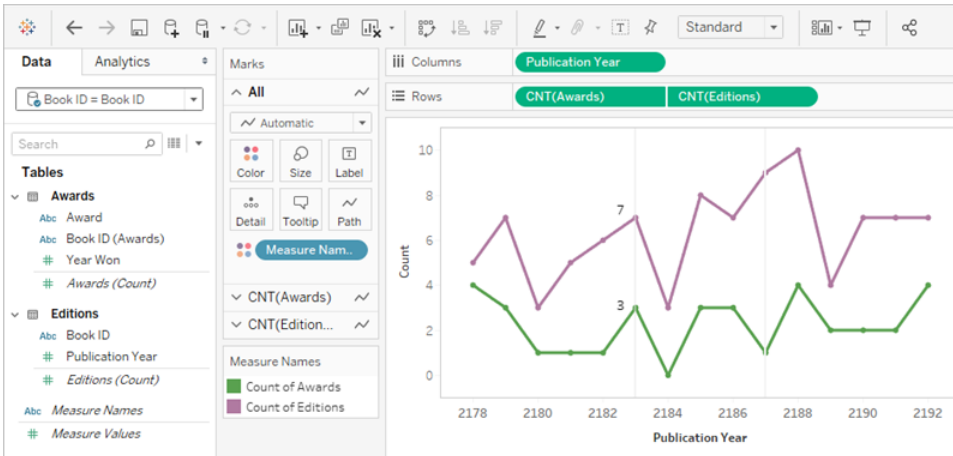
the right-hand column is a calculated field that used publication date but if that field is null uses Year won (this calculation is used to ensure that both domains are fully represented).

- Each row is a different relationship for how the Award and Edition table are combined. The top row relates the tables on Book ID, the middle row relates them on Year, and the bottom row relates them on both Book ID and Year.
- The bars for the year 2183 are thicker to make comparison easier. In the solution below, that is the year whose values are looked at in detail.
- Note that the two vizzes that are shaded have identical values.



If you get stuck

Walk through each part of the viz step-by-step. Look at the data source structure, the axis and headers, and the fields used for the marks (and which tables they come from). Think about nulls and why they might be there. View the data for a specific mark or two to see which records it contains.



- The data source is Awards related to Editions on Book ID.
- The date axis is Publication Year
- The values are Count of Awards and Count of Editions.

Try to pick a single mark in the viz and define it. For the year 2183, the viz shows there are seven editions and three awards. Use **View Underlying Data** on page 2728 to investigate which records are represented in each mark.

Editions

7 rows ☑ Show aliases

Publication Year	Book ID
2183	WG715
2183	DY800
2183	TM925
2183	QQ737
2183	IA439
2183	WO506
2183	DE571

Awards

3 rows ☑ Show aliases ☑ Show all fields

Publication Year	Award	Book ID (Awards)	Year Won
2183	Nebula Award	TM925	2179
2183	Hugo Award	TM925	2179
2183	Pulitzer	WO506	2183

Together, this can be interpreted as “return the seven books published in 2183, then for those books, list any awards they won *regardless of when they won the award*”. TM925 was originally published in 2179 and the hardcover edition won two awards in that year. In 2183, a different edition of the book was published, perhaps in paperback. The value for Count of Award is tied to the book, not the year.

So the viz overall could be interpreted as “the number of editions published each year, and how many awards were won by books published in that year” or “the number of editions published each year and the number of awards won by those books”.

Solution to the hands-on exercise

The year 2183—the thicker bars—is the year focused on for interpretation. Information about editions is in purple and information about awards is in green. The year 2187 is also called out in the descriptions because it is a year when no awards were won but books were published. As such, it is a good illustration of the power of which date field is used for the axis. This nuance is addressed in blue in the description. The importance of the field(s) used to set up the relationship is called out in pink.

This visual is provided for your convenience, but a better experience may be to download the workbook and open it in Tableau Desktop 2020.2 or later so you can make use of interactivity such as tooltips and View Data. If you prefer to look at the sheets more closely, you can right click on either dashboard tab at the bottom and choose Unhide all Sheets. This will make all the individual vizzes available, and from each sheet you can see the Data pane and the authoring environment, including which fields are on which shelves. Note that in order to achieve the three different relationships (the rows of the matrix above), there are three different data sources.

Continued reading

Feeling a bit overwhelmed and want to go back a step? Try **Don’t Be Scared of Relationships** on page 861.

Ready to tackle calculations with Relationships? Check out **Don’t Be Scared of Calculations in Relationships** on page 871.

For more information on the technical underpinnings of relationships straight from the Product Management team, check out the series on relationships on the Tableau Blog.

- Relationships, part 1: Introducing new data modeling in Tableau
- Relationships, part 2: Tips and tricks
- Relationships, part 3: Asking questions across multiple related tables

Join Your Data

It is often necessary to combine data from multiple places—different tables or even data sources—to perform a desired analysis. Depending on the structure of the data and the needs of the analysis, there are several ways to combine the tables.

Relationships vs Joins

The default method in Tableau Desktop is to use relationships. Relationships preserve the original tables' level of detail when combining information. Relationships also allow for context-based joins to be performed on a sheet-by-sheet basis, making each data source more flexible. Relationships are the recommended method of combining data in most instances. For more information, see **How Relationships Differ from Joins** on page 821.

However, there may be times when you want to directly establish a join, either for control or for desired aspects of a join compared to a relationship, such as deliberate filtering or duplication.

Note: Relationships eventually leverage joins (just behind the scenes). For example, a relationship across data sources will produce a cross-database join when the viz uses fields from tables in different data sources. As such, **Improve Performance for Cross-Database Joins** on page 901 may be relevant.

Common issues

- To view, edit, or create joins, you must open a logical table in the relationship canvas—the area you see when you first open or create a data source—and access the join canvas.
- **Published Tableau data sources cannot be used in joins.** To combine published data sources, you must edit the original data sources to natively contain the join or use a data blend.

- When joining tables, the fields that you join on must be the same data type. If you change the data type after you join the tables, the join will break.
- Fields used in the join clause cannot be removed without breaking the join. To join data and be able to clean up duplicate fields, use Tableau Prep Builder instead of Desktop

Tip: While Tableau Desktop has the capability to create joins and do some basic data shaping, Tableau Prep Builder is designed for data preparation. If you need to do multiple joins, clean up field names, change data types, perform multiple pivots, or other sorts of involved data prep, consider using Tableau Prep Builder.

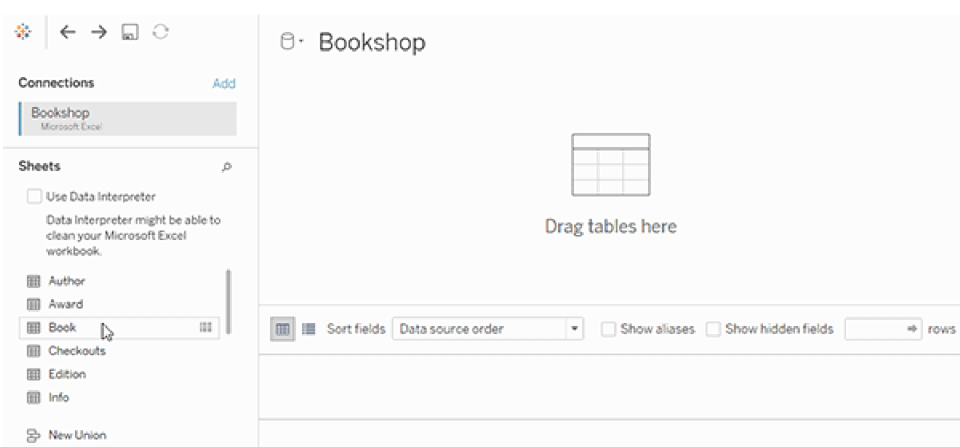
Create a join

1. To create a join, connect to the relevant data source or sources. See **Connect to Your Data** on page 391.

   These can be in the same data source (such as tables in a database or sheets in an Excel spreadsheet) or different data sources (this is known as a cross-database join). If you combined tables using a cross-database join, Tableau colors the tables in the canvas and the columns in the data grid to show you which connection the data comes from.

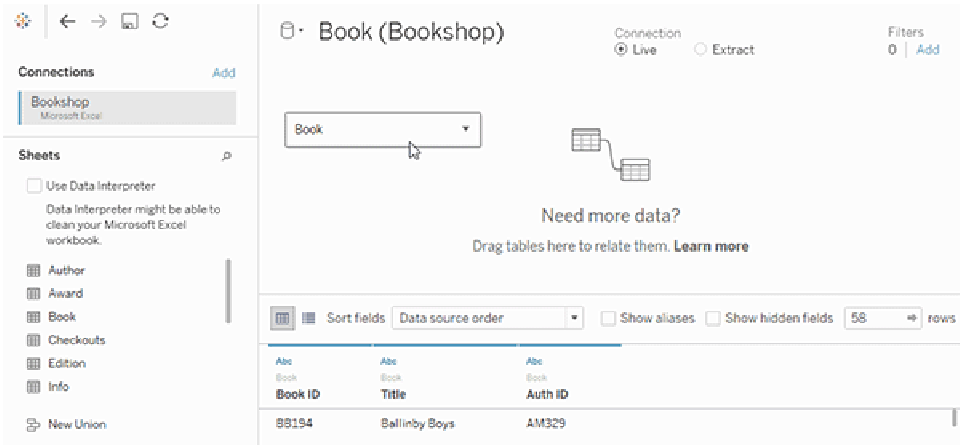
   > **Note**: Not all data sources support cross-database joins, including published Tableau data sources. To combine published data sources, edit the original data sources to natively contain the join or use a data blend.

2. Drag the first table to the canvas.



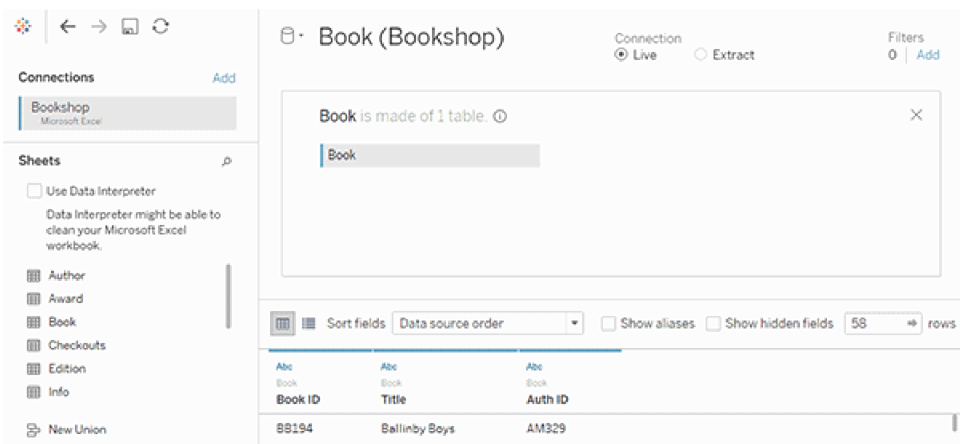
Note that if you drag out a second table at this point a relationship will form.

3. Select Open from the menu or double-click the first table to open the join canvas (physical layer).



Double-clicking a table on the relationship canvas opens the join canvas

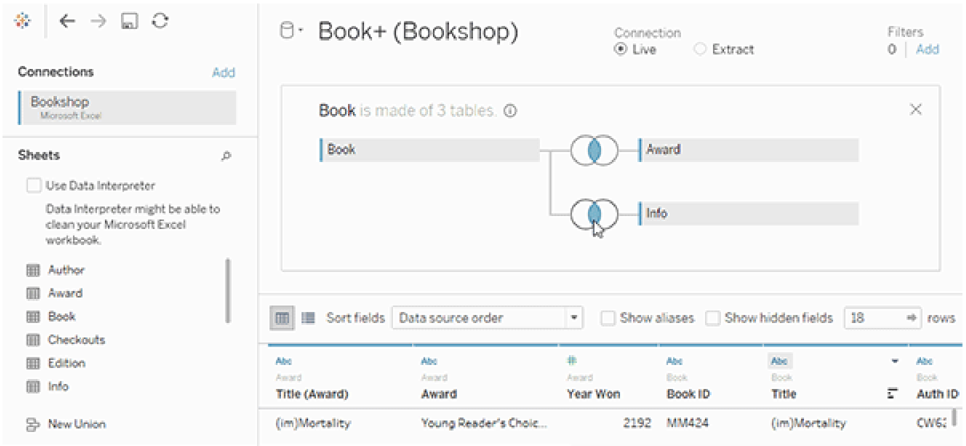
4. Double-click or drag another table to the join canvas.



If your next table is from another data source entirely, in the left pane, under

Connections, click the **Add** button (in web authoring) to add a new connection to the Tableau data source. With that connection selected, drag the desired table to the join canvas.

5. Click the join icon to configure the join. Add one or more join clauses by selecting a field from one of the available tables used in the data source, choosing a join operator, and a field from the added table.



> **Note:** You can delete an unwanted join clauses by clicking the "x" that displays when you hover over the right side of the join clause.

6. When finished, close the join dialog and join canvas.

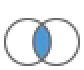
After you've created a join, **Join Your Data** on page 883. To troubleshoot your join, see **Join Your Data** on page 883.

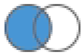
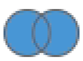
Anatomy of a join

Joins are defined by their type as well as the join clause.

Join types

In general, there are four types of joins that you can use in Tableau: inner, left, right, and full outer. If you aren't sure what join type you want to use to combine data from multiple tables, you should use relationships.

Join Type	Result
Inner 	When you use an inner join to combine tables, the result is a table that contains values that have matches in both tables. When a value doesn't match across both tables, it is dropped entirely.
Left	When you use a left join to combine tables, the result is a table that contains all values from the left table and corresponding matches from the right table.

	When a value in the left table doesn't have a corresponding match in the right table, you see a null value in the data grid.
Right 	When you use a right join to combine tables, the result is a table that contains all values from the right table and corresponding matches from the left table. When a value in the right table doesn't have a corresponding match in the left table, you see a null value in the data grid.
Full outer 	When you use a full outer join to combine tables, the result is a table that contains all values from both tables. When a value from either table doesn't have a match with the other table, you see a null value in the data grid.
Union 	Though union is not a type of join, union is another method for combining two or more tables by appending rows of data from one table to another. Ideally, the tables that you union have the same number of fields, and those fields have matching names and data types. For more information about union, see **Union Your Data** on page 942.

Not all databases support all join types. If an option is unavailable in the join dialog, it is likely due to a constraint from your data source.

Join Clauses

A join is performed by setting up one or more join clauses. The join clause tells Tableau which fields are shared between the tables and how to match the corresponding rows. For example, rows with the same ID are aligned in the results table.

Join clauses most often use the equality operator (=) which matches rows with the same values. It is also possible to perform non-equi joins, such as less than (<) and not equal (<>).

A join can also have multiple join clauses. For example, if First name and Last name are stored in separate columns, it may be beneficial to join only if “First name = First name” and “Last name = Last name”. Both conditions will have to be true for rows to be joined. Alternatively, if the goal was to return results when the last name is shared but the first name is not, the join clauses could be “First name <> First name” and “Last name = Last name”.

Join clauses can also contain calculations. For example, the join clause could be the concatenation of the name fields “[First name] + [Last name] = [First name] + [Last name]”. Note that not all data source connections support calculations in join clauses.

About null values in join keys

In general, joins are performed at the database level. If the fields used to join tables contain null values, most databases return data without the rows that contain the null values. However, for certain single-connection data sources, Tableau provides an additional option to allow you to join fields that contain null values with other fields that contain null values.

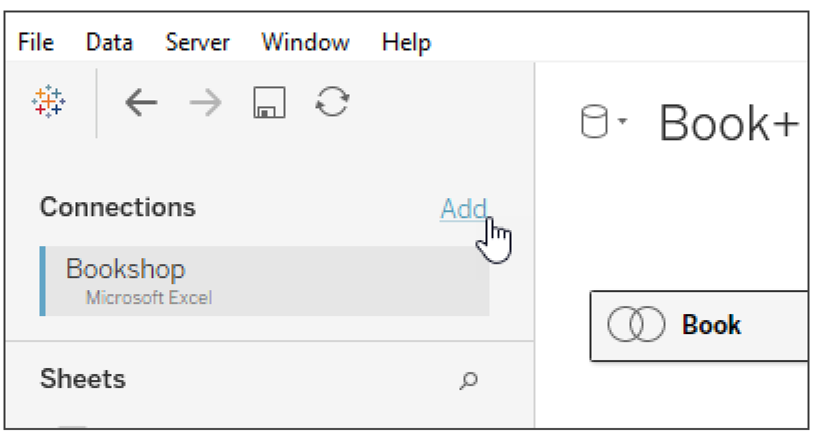
After you've set up your data source, on the data source page, select **Data** > **Join null values to null values**.

If the option is greyed out, it is not available for your data source. Note that if you add a second connection to a data source that uses this option, the join reverts back to the default behavior of excluding rows with null values.

Cross-database joins

Tableau allows joins from tables in different data sources, albeit with some limitations from the database side on which platforms are compatible. Cross-database joins require a multi-connection data source—that is, you create a new *connection* to each database before you join the tables.

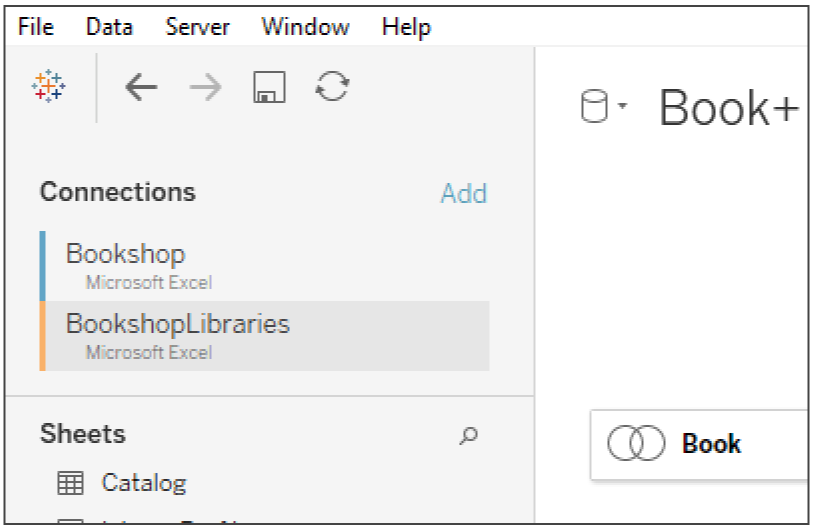
1. Once you've connected to the first source of data, use the Add option in the data pane to add another connection.



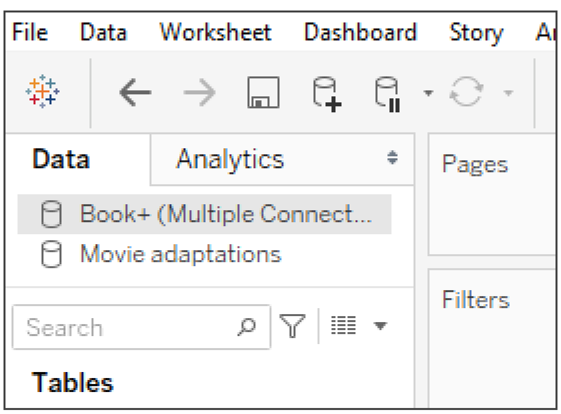
> **Note:** If the connector you want is not available from the **Connect** list when you're trying to add another connection, cross-database joins are not supported for the combination of sources that you want to join. This includes connections to

cube data (e.g., Microsoft Analysis Services), most extract-only data (e.g., Google Analytics and OData), and published Tableau Server data sources.

2. This creates a second connection rather than an entirely different data source. You can switch between the two (or more) connections while on the data source tab.



3. Once you move to a worksheet and begin analysis, the data source functions as a single, combined data source. This is in contrast to two independent data sources that can be toggled between on a worksheet.



The **Book+** data source contains two connections with a cross-database join. It functions as a single data source. The **Movie adaptations** data source is an independent data source.

Note: Typically, joining tables from the same database yields better performance. This is because querying data that is stored on the same database takes less time and leverages the native capabilities of the database to perform the join. For more information on cross-database join performance, see **Improve Performance for Cross-Database Joins** on page 901.

Review Join Results in the Data Grid

After you have created a join on the canvas, review the data grid to make sure the join produces the results that you expect. If the data grid displays data that you don't expect, you might need to modify the join.

Results in the data grid

- **No data:** If no data displays in the data grid, you might need to change the join type or a join field used in the join condition. If you suspect a mismatch between fields in the join, use a calculation instead. For more information, see **Use calculations to resolve mismatches between fields in a join** below.
- **Duplicate data:** If you see duplicate data, there a few things you can do. Consider relating the tables instead of joining. For more information, see Combine Data Using Relationships.
- **Missing data:** If some data is missing from the data grid, you might need to change the join type or a join field used in the join condition. Again, if you suspect a mismatch between fields in the join, use a calculation instead. For more information, see **Use calculations to resolve mismatches between fields in a join** below.
- **Many null values:** If you see many null values that you do not expect, you might need to change the join type from the full outer type to the inner type.
- **All null values for one table:** If all values for one table is null, there are no matches between the tables that you are joining. If this is not expected, consider changing the join type.

Use calculations to resolve mismatches between fields in a join

When the fields in a join condition don't match—that is a mismatch between the values in the fields used in a join condition, the data grid can show little or no data at all. A mismatch between fields can occur for several reasons but often caused by the differences in format of the string values or date values in the fields. In many cases, you can resolve mismatches between the fields in your join by using a calculation.

Most functions are available for you to use in a calculation to create and replace a field in the join condition, with the exception of aggregate functions and table calculation functions.

> **Note:**Join calculations are not supported for QuickBooks Online, Marketo, Oracle Eloqua, Anaplan, ServiceNow ITSM, and web data connectors.

String mismatch

A common mismatch scenario when working with string data occurs when one of the fields on one side of the join condition is equivalent to two or more fields on the other side of the join condition. In this case, you can use a calculation to combine the two fields so that its format matches the other field in the join condition.

For example, suppose you want to join two tables that contain the following data:

Patron

First name	Last name	Branch	Member since	Units borrowed	Fees	Suggested limit
Alan	Wang	North	2000	1	0	15
John	Smith	North	2000	36	3.50	15
Ashley	Garcia	South	2000	243	11.30	15
Fr-	Su-	Nor-	200-	52	.9-	15

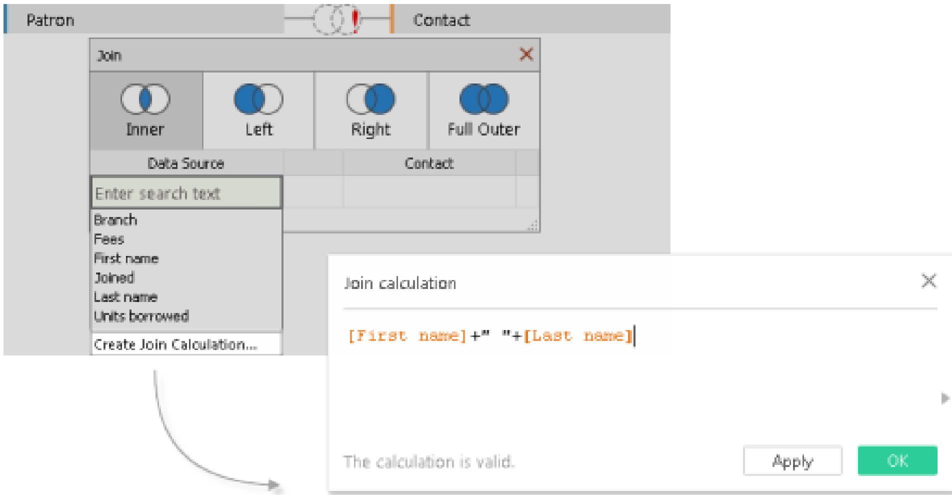
Contact

Name	Member number	Emergency contact	Relationship	Emergency number
Adam Davis	555-0324	Ellen Davis	Partner	555-0884
Alan Wang	555-0356	Jean Wilson	Mother	555-0327
Fre-	555-	Jim Suzuki	Brother	555-3188

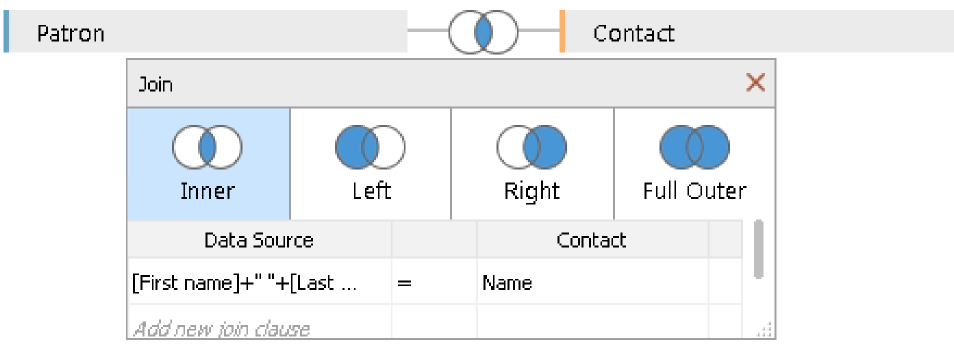
First name	Last name	Branch	Member since	Units borrowed	Fees	Suggested limit
ed	zuki	th	0		0	

Name	Member number	Emergency contact	Relationship	Emergency number
d Suzuki	018-8			
Henry Wilson	555-010-0	Laura Rodriquez	Partner	555-0103
Michelle Kim	555-019-9	Steven Kim	Partner	555-0125

The common fields between the two tables appear to be name. However, in the Patron table the first and last names are in separate columns and in the Contact table the first and last names are in the same column. To join the tables on names, you can use a calculation in the left side of the join condition to merge the first name and last name columns together.



The result is a calculated field on the left side of the join condition that is accessible only from the join dialog. This calculation converts the field in the Patron table into a format that now matches the format of the field in the Contact table on the right side of the join condition.



Using the calculation in the join produces the following combined table:

First name	Last name	Branch	Member since	Units borrowed	Fees	Suggested limit	Name	Phone number
Alan	Wang	North	2000	1	0	15	Alan Wang	555-0356
Fred	Suzuki	North	2000	52	.90	15	Fred Suzuki	555-0188

Date mismatch

A common mismatch scenario when working with date data occurs when the date values in one field of the join condition are captured at a different level of detail than the other field in the join condition. In this case you can use a calculation in the join condition to change the format of the field on one side of the join condition so that its format matches the other field in the join condition.

For example, suppose you have the following two tables of data:

Projector rental

Date	Reservation type	Requester ID
1/1/2000	Individual	23344-5589
1/28/2002	K-12	36594-8999
1/29/2002	Non-profit	23344-8888
12/5/2002	K-12	36594-8999
5/5/2003	Non-profit	33401-5476
3/12/2004	Non-profit	33401-5476
3/15/2006	City	21189-6980
7/8/2007	K-12	33401-5476
1/4/2008	Individual	56049-5523

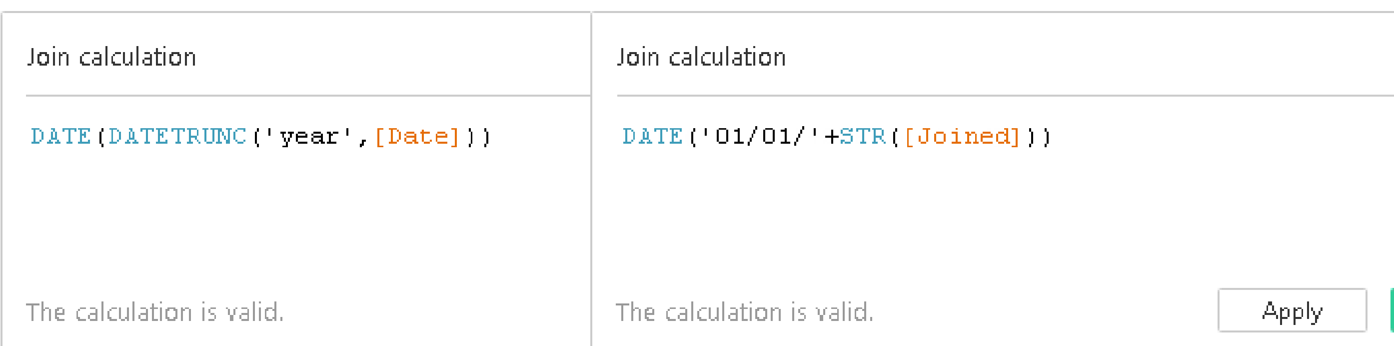
Patron

ID	First name	Last name	Branch	Member since	Units borrowed	Fees	Suggested limit
45461-3981	Adam	Davis	West	2012	25	0	10
23250-2870	Alan	Wang	North	2000	1	0	15
29800-0916	Amanda	Smith	North	2001	54	6.4	15
23344-5566	Ashley	Garcia	South	2000	243	11.30	15
90000-5122	Brian	Frank	East	2011	12	.10	10
92149-1769	Elizabeth	Johnson	West	2010	19	.5	10
23344-5589	Fred	Suzuki	North	2000	52	.90	15
34455-	Henr-	Wils-	Sou-	200-	3	.2	15

Date	Reservation type	Requester ID
3/8/2009	Non-profit	23344-5566
2/14/2014	Non-profit	23344-5566
12/21/2015	Non-profit	23344-5566
2/10/2016	Non-profit	23344-5566

ID	First name	Last name	Branch	Member since	Units borrowed	Fees	Suggested limit
6677	y	on	th	5			
93950-2870	Jane	Johnson	West	2017	0	0	10
23344-8978	John	Smith	North	2000	36	3.50	15

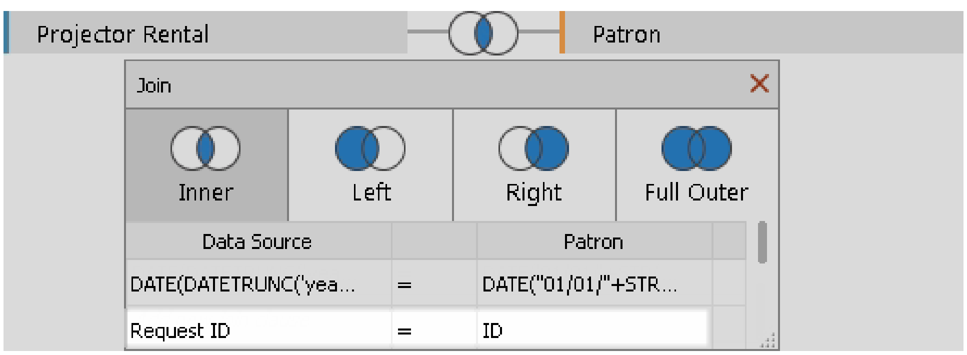
To find out more information about new patron behavior, joining the Patron table to the Projector Rental table might provide some insight about which library services motivate new memberships. The common fields between the two tables appear to be "Date" and "Member since." However, the date values in each field are captured at different levels of detail. To join these tables on their respective date fields, use a combination of DATE functions in a calculation on each side of the join condition to make the level of detail in each field match.



Using the calculation in the join produces the following combined table:

Date	Reservation type	Requester ID	ID	First name	Last name	Branch	Member since	Units borrowed	Fees	Suggested limit
1/1/2000	Individual	233445589	232502870	Alan	Wang	North	2000	1	0.00	15
1/1/2000	Individual	233445589	233445589	Fred	Suzuki	North	2000	52	0.90	15
1/1/2000	Individual	233445589	233445566	Ashley	Garcia	South	2000	243	11.30	15
1/1/2000	Individual	233445589	233448978	John	Smith	North	2000	36	3.50	15

To determine if a patron rented the projector in the same year he or she started his or her membership, add one more clause to the join based on ID.



The result of the additional join condition shows that only one patron might have started his membership to rent a projector.

Date	Reservation type	Requester ID	ID	First name	Last name	Branch	Joined	Units borrowed	Fees	Suggested limit
1/1/2000	Individual	233445589	233445589	Fred	Suzuki	North	2000	52	0.90	15

Troubleshoot Joins

Note: If your join is mis-counting data, it could be a sign you should be using a relationships instead. For more information about relationships, see **Relate Your Data** on page 766. For information about how Tableau handles combining data from different levels of detail (such as many-to-many relationships), see Cardinality and Referential Integrity.

Over-counting values or duplicated data

When you connect to and join multiple tables together, you set up a *denormalized* version of the data. In some cases, Tableau treats the multiple tables as one table. When multiple tables are treated as one table, after the tables are joined, *all* tables are queried. This can cause values to be over-counted.

For example, suppose you have two tables. The first table contains employee information, such as employee ID and salary. The second table contains organizational information, such as department names.

Table 1

Employee ID	Name	Salary
20106	Kim, Michelle	63,000
20107	Wilson, Henry	61,000
20108	Smith, John	65,000

Table 2

Department	Hire Date	Employee ID	Organization
Development	6/16/2016	20289	Strategy
Support	7/28/2015	20107	Operations
Support	8/05/2016	20299	Operations
Sales	8/25/2016	20107	Operations

If you join these tables on **Employee ID**, an employee's salary is counted more than once because the employee is associated with multiple departments.

Employee ID	Name	Salary	Department	Hire Date	Organization
20107	Wilson, Henry	61,000	Support	7/28/2015	Operations
20107	Wilson, Henry	61,000	Sales	8/25/2016	Operations

This is an example of one-to-many cardinality (one employee ID in Table 1 has multiple results in Table 2). For more information on cardinality between tables, see **Optimize Relationship Queries Using Performance Options** on page 825.

To help resolve this issue, you can do one of the following:

- **Relate the data:** Instead of creating a join, which might duplicate your data, you can use relationships. For more information, see **Relate Your Data** on page 766.

  Change the aggregation: Depending on your analysis, you can use the **MIN** or **AVG** aggregation to remove over-counting.

  For example, if you change the aggregation of **Salary** from SUM to MIN or from SUM to AVG.

- **Create a calculated field that removes the duplicate values:** You can create a calculation that divides the sum of the field being duplicated by the number of instances of the field that is causing the duplication.

  For example, the Salary values are being duplicated by the number of instances of Employee ID for Wilson, Henry. In this case, in the view replace Salary with a calculated field that uses the following formula: `SUM ([Salary])/ COUNT ([Employee ID])`.

  You could also use a Level of Detail Expression to delete the duplicate data. For more information, see Removing Duplicate Data with LOD Calculations in the Tableau Knowledge Base.

- **Use custom SQL:** Tableau supports using custom SQL for connecting to multiple tables in a data source. You can pre-aggregate the tables before joining them with a GROUP BY clause. This option requires some knowledge about writing SQL queries and assistance from a database expert if possible. For more information about how to connect to a custom SQL query from Tableau, see **Connect to a Custom SQL Query** on page 959.

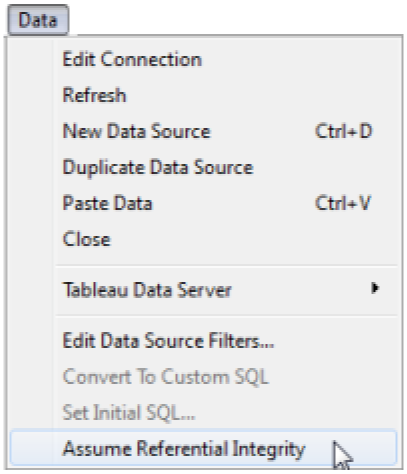
Broken table joins and cross-database joins

Before you join tables (from the same data source or using a cross-database join), make sure that the data types of the join keys match. If the data types of the join keys don't match, the join breaks, which is indicated by a red exclamation point and the error text "type mismatch between join fields". To fix a broken join, use one of the following suggestions:

- You can change the format of a field (including its data type) in the join dialog to make the join keys match by creating a join calculation. A join calculation supports a subset of calculations that lets you modify the format of the join key in one or more tables that you want to combine. To create a join calculation, click the join icon between the tables that have a broken join, click the field whose format needs to be modified, and then select **Create Join Calculation**. For more information, see **Troubleshoot Joins** on page 897.
- For text or Excel-based data, modify the data type of one of the text or Excel fields in the join key using the data type menu in the data grid.
- **Tableau Desktop only**: For most connections, you can use the **Convert to custom SQL** option to change the data type of one of the fields in the join key, using a function like CAST (). The **Convert to custom SQL** option is available only when the data source contains only one connection. In this case, remove the second connection and then select **Data** > **Convert to custom SQL**.

Assuming Referential Integrity for Joins

In some cases, you can improve query performance for joins by selecting the option to **Assume Referential Integrity** from the **Data** menu. When you use this option, Tableau will include the joined table in the query only if it is specifically referenced by fields in the view.



Using this setting is appropriate when you know that your data has referential integrity (see definition below) but your database is not enforcing or cannot enforce referential integrity. If you have the option of configuring referential integrity in your database that is a better option than using this setting because it can improve performance both in the database and in Tableau. The **Assume Referential Integrity** option in Tableau can only affect performance on Tableau's end. If your data does not have referential integrity and you turn on this setting, query results may not be reliable.

To understand what referential integrity is, imagine connecting to sales data that has two tables: Sales and Product Catalog. These two tables are shown below:

Sales

Product ID (Foreign Key)	Sale Amount	Transaction Date
1	100	10/1/2012
1	2000	10/2/2012
2	50	9/30/2012
3	10	8/21/2012

Product Catalog

Product ID (Primary Key)	Product Name
1	10 Inch Tablet
2	Smart Phone
3	Desk Lamp

4	Memor-y Stick

Because all products that are sold must have a listing in the Product Catalog, every row in the Sales table has a matching row in the Product Catalog table. When these two tables are joined on Product ID, you end up with a table that looks like this:

Product ID	Product Name	Product ID	Sale Amount	Transaction Date
1	10 Inch Tablet	1	100	10/1/2012
1	10 Inch Tablet	1	2000	10/2/2012
2	Smart Phone	2	50	9/30/2012
3	Desk Lamp	3	10	8/21/2012

Now let's say you build a view to look at Sale Amount by Region. By default, dragging the Sale Amount field to the view may create a query like this:

```
SELECT SUM([Sales Amount]) FROM [Sales] S INNER JOIN [Product
Catalog] P ON S.ProductID = P.ProductID
```

By selecting **Assume Referential Integrity**, you tell Tableau that the joined tables have referential integrity. In other words, you are confirming that the Sales table will always have a matching row in the Product Catalog table. Because that is true, Tableau doesn't need any information from the Product Catalog table in order to return these results. When you drag the Sales Amount field into the view, Tableau can simplify the query to:

```
SELECT SUM([Sales Amount]) FROM [Sales]
```

This simplified query can often return quicker results because it removes the join operation. This option impacts only inner joins and does not affect data sources with a single table.

Improve Performance for Cross-Database Joins

Important: This feature temporarily moves data outside of Tableau. Be sure that the database you are connected to is from a trusted source.

Beginning with Tableau Desktop and Tableau Server version 2019.3, you can improve performance when joining data from a single file connection and a single database connection by allowing Tableau to perform the join using the database that you are connected to instead of Hyper. When this option is enabled, Tableau chooses the fastest option (Hyper or the connected database). If Tableau uses the connected database, the data from the file connection is moved into temporary tables in the database and the join is performed there.

Feature conditions

This option is only available if the following conditions are met:

- The data source consists of one or more file-based connections and a single SQL-based connection.
- The file is a Microsoft Excel, PDF or Text (.csv, .txt, .tsv or .tab) file type.
- The connected database is one of the following:
  - Microsoft SQL Server
  - Oracle
  - PostgreSQL
  - Vertica
  - Teradata
- The join type is an inner join.
- In **web authoring**: The **Allow users to use web authoring** option is enabled.

  Site Administrators can also configure the options available for users by setting the **Cross-Database Joins** options on the **Settings** > **General** tab in Tableau Server. For more information about these settings, see Set a Sites Web Authoring Access in the Tableau Server help.

Overriding feature conditions

Starting in version 2020.3, as an administrator, you can override the file size, join type, and single file connection limitations and force Tableau to use the live database connection to perform the join. This enables you to experiment and determine optimal performance configurations. The single database connection requirement still applies.

To enable this option enter the following information from the command line:

- **Tableau Desktop**: Enter the command `tableau.exe -DForceAlternativeFederationEngine=true`.
- **Tableau Server**: Enter the tsm configuration command `tsm configuration set`

```
-k native_api.force_alternative_federation_engine -v true
```

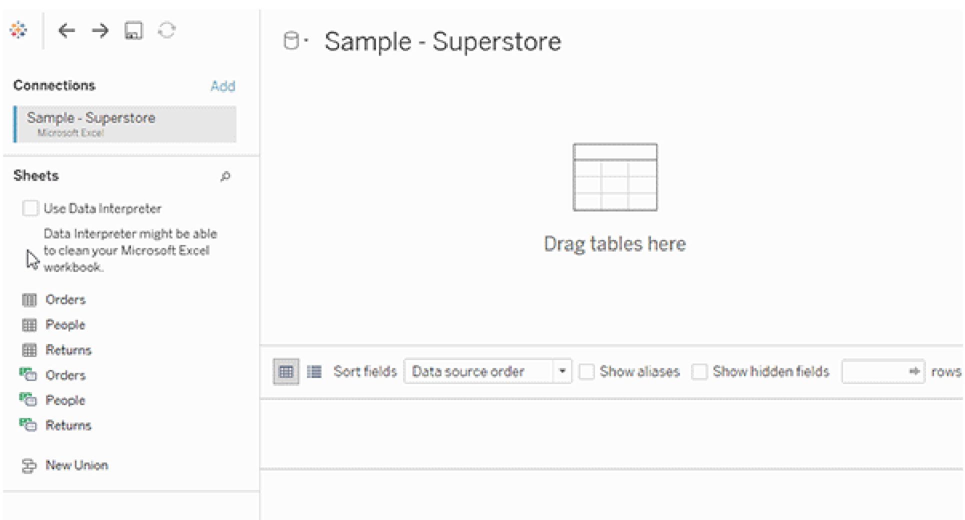
For more information about setting configuration values in Tableau Server, see tsm configuration set Options in the Tableau Server help.

Enable the performance option for cross-database joins

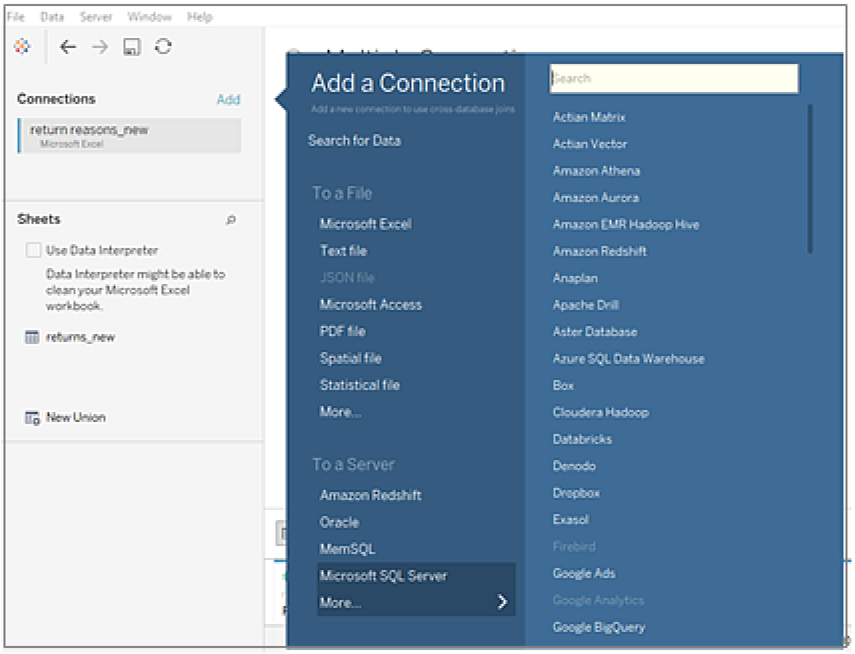
1. In **Tableau Desktop**: On the Start page, under **Connect**, connect to a supported file type or supported database type. This step creates the first connection in the Tableau data source.

   In **web authoring**: From the **Home** or **Explore** page, click **Create**> **Workbook** to start a new workbook and then connect to your data. This step creates the first connection in the Tableau data source.

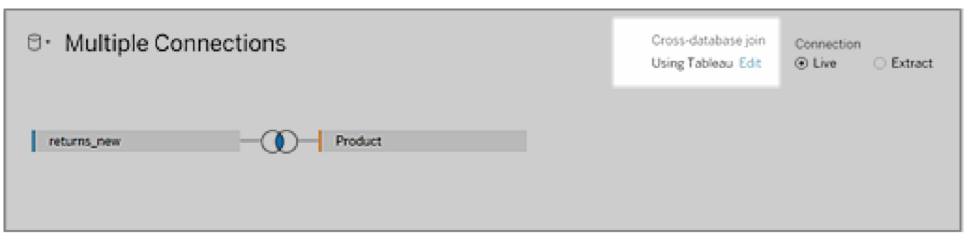
2. Select the file or database that you want to connect to, then double-click or drag a table to the canvas.



3. In the left pane, under **Connections**, click the **Add** button (in web authoring) to add your second connection to the Tableau data source.



The **Cross-database join** option is displayed.



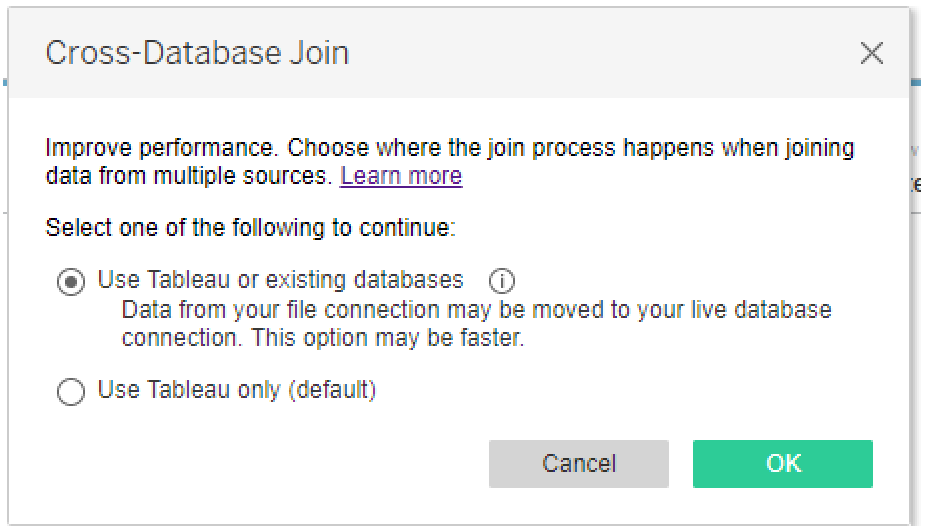
> **Note**: If you don't see this option, check that you are using only supported data sources and that you have only two data sources (one file and one database type). Otherwise, the Site Administrator may have set the **Cross-Database Joins** configuration option to **Tableau only**.

4. To change how Tableau performs the join, next to the **Cross-database join** option, click **Edit**.

5. In the in the **Cross-Database Join** dialog, select one of the following options, then click **OK**:

   - **Use Tableau or existing databases**. This option allows Tableau to choose the fastest option to perform the join - either Hyper or the database you're connected to.

- **Use Tableau only**. This option is the default and always uses Hyper to perform the join.



The **Cross-database join** option changes from the default option, **Using Tableau** (using Hyper), to the new option **Using your database**, depending on what you choose.

Important: If you select **Use Tableau or existing databases**, Tableau chooses the fastest option when performing the join. This behavior is pre-determined by a set of criteria including join types. For instance, Tableau will always choose Hyper for non-inner joins.

If Tableau uses Hyper to perform the join, this process happens in the background and no indicator is shown to identify where the join was performed.

6. Add one or more join clauses by selecting a field from one of the available tables used in the data source, a join operator, and a field from the added table. Inspect the join clause to make sure it reflects how you want to connect the tables.

About working with multi-connection data sources

Working with multi-connection data sources is just like working with any other data source, with a few caveats, discussed in this section.

Union data from within a connection

To union data, you must use text tables or Excel tables from the same connection. That is, you cannot union tables from different databases. In Tableau Desktop, you can union tables across different Excel workbooks and files in different folders. For more information, see the **Union tables using wildcard search (Tableau Desktop)** on page 946.

If you need to union data from different databases, use Tableau Prep.

Collation

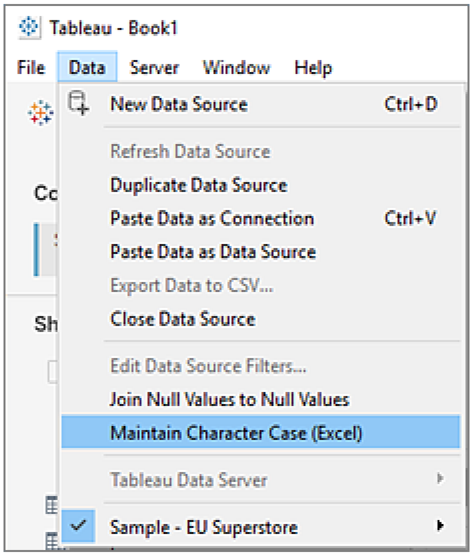
Collation refers to the rules of a database that determine how string values should be compared and sorted. In most cases, the collation is handled by the database. However, when you work with cross-database joins, you might join columns that have different collations.

For example, suppose your cross-database join used a join key comprised of a case-sensitive column from SQL Server and a case-insensitive column from Oracle. In cases like this, Tableau maps certain collations to others to minimize interpreting values incorrectly.

The following rules are used in cross-database joins:

- If a column uses collation standards of the International Components for Unicode (ICU), Tableau uses the collation of the other column.
- If all columns use collation standards of the ICU, Tableau uses the collation of the column of the left table.
- If no columns use collation standards of the ICU, Tableau uses a binary collation. A binary collation means the locale of the database and data type of the columns determine how string values should be compared and sorted.

When working with Microsoft Excel data if you need to maintain case sensitivity for your data when performing joins, for example Japanese data that is Kana-sensitive, you can enable the **Maintain Character Case (Excel)** option from the Data menu.



When this option is selected, Tableau maintains the casing of the values from your Excel data source and uniquely identifies values with different casing instead of combining them, resulting in a different number of rows.

For example, you have one worksheet with the value "House" and another worksheet with the values "House", "house" and "houSe". By default, when you join the two tables, Tableau ignores the character casing differences in the second worksheet and joins "House", "house" and "houSe" from the second worksheet with "House" from the first worksheet.

With the **Maintain Character Case (Excel)** option enabled, when you join your tables, Tableau recognizes the character casing differences. The join results show fewer results because the join will only match "House" from the first worksheet with the value "House from the second worksheet and read "house" and "houSe" as unique separate values.

> **Note:** This option is available for all Tableau supported languages and is not dependent on the locale of your operating system. This option is only available for Microsoft Excel data sources.

Calculations and multi-connection data sources

Only a subset of calculations can be used in a multi-connection data source.

In Tableau Desktop: You can use a specific calculation if it is both:

- Supported by all the connections in the multi-connection data source
- Supported by Tableau extracts.

In web authoring (Tableau Online and Tableau Server): You can use a specific calculation if it is supported by all the connections in the multi-connection data source.

Stored procedures

Stored procedures are not available for multi-connection data sources.

Pivot data from within a connection

To pivot data, you must use text columns or Excel columns from the same connection. That is, you cannot include columns from different databases in a pivot.

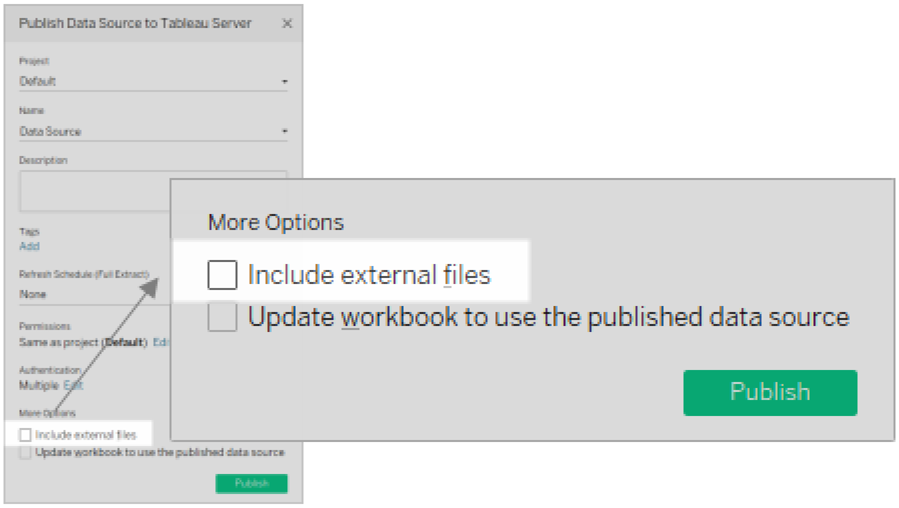
Make extract files the first connection (Tableau Desktop only)

When connecting to extract files in a multi-connection data source, make sure that the connection to the extract (.tde or .hyper) file is the first connection. This preserves any customizations that might be a part of the extract, including changes to default properties, calculated fields, groups, aliases, etc.

Note: If you need to connect to multiple extract files in your multi-connection data source, only the customizations in the extract in the first connection are preserved.

Extracts of multi-connection data sources that contain connections to file-based data (Tableau Desktop only)

If you're publishing an extract of a multi-connection data source that contains a connection to file-based data such as Excel, selecting the **Include external files** option puts a copy of the file-based data on the server as part of the data source. In this case, a copy of your file-based data can be downloaded and its contents accessed by other users. If there is sensitive information in the file-based data that you have intentionally excluded from your extract, *do not* select **Include external files** when you publish the data source.

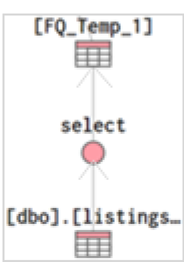
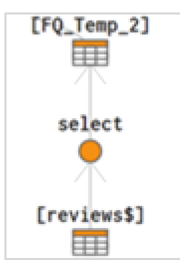


For more information about publishing data sources, see **Publish a Data Source** on page 3071.

About queries and cross-database joins

For each connection, Tableau sends independent queries to the databases in the join. The results are stored in a temporary table, in the format of an extract file.

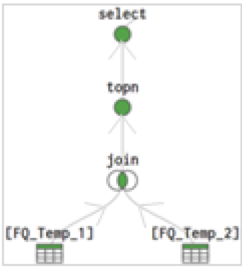
For example, suppose you create connections to two tables, dbo.listings and reviews$. These tables are stored in two different databases, SQL Server and Excel. Tableau queries the database in each connection independently. The database performs the query and applies customizations such as filters and calculations, and Tableau stores the results for each connection in a temporary table. In this example, FQ_Temp_1 is the temporary table for the connection to SQL Server and FQ_Temp_2 is the temporary table for the connection to Excel.

SQL Server table	Excel table
	

When you perform a cross-database join, the temporary tables are joined together by Tableau Desktop. These temporary tables are necessary for Tableau to perform cross-database joins.

After the tables have been joined, "topn" filter is applied to limit the number of values shown in the data grid to the first 1,000 rows. This filter is applied to help maintain responsiveness of the data grid and the overall performance of the Data Source page.

Joined tables



Blend Your Data

Data blending is a method for combining data from multiple sources. Data blending brings in additional information from a secondary data source and displays it with data from the primary data source directly in the view.

There are several ways to combine data, each with their own strengths and weaknesses.

Relationships are the default method and can be used in most instances, including across tables with different levels of detail. Relationships are flexible and are adaptable to the structure of the analysis on a sheet by sheet basis. However, relationships cannot be formed between tables from data sources published to Tableau Server or Tableau Online.

Joins combine tables by adding more columns of data across similar row structures. This can cause data loss or duplication if tables are at different levels of detail, and joined data sources must be fixed before analysis can begin.

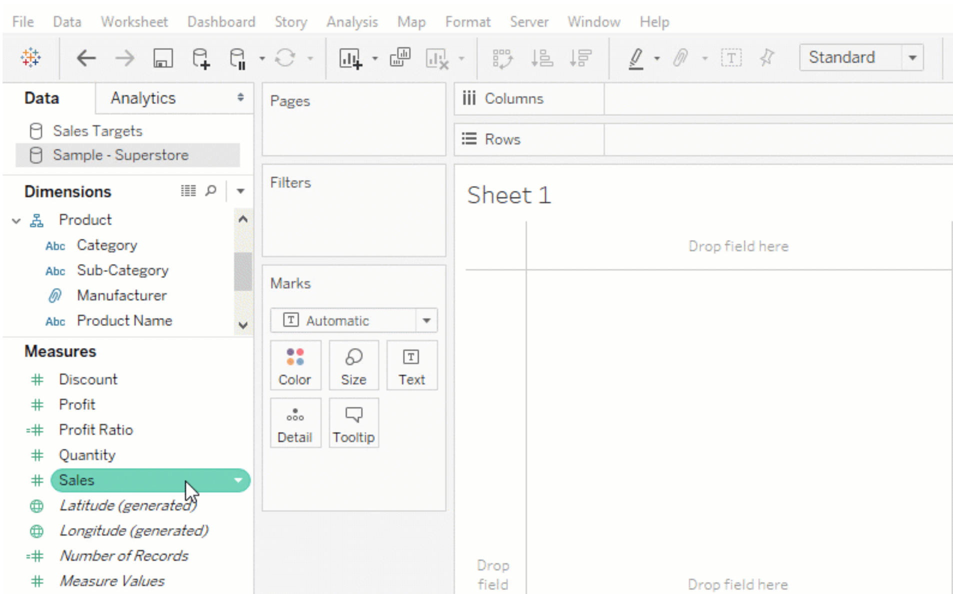
Blends, unlike relationships or joins, never truly combine the data. Instead, blends query each data source independently, the results are aggregated to the appropriate level, then the results are presented visually together in the view. Because of this, blends can handle different levels of detail and working with published data sources. Blends are also established individually on every sheet and can never be published, because there is no true “blended data source”, simply blended results from multiple data sources in a visualization.

Data blending is particularly useful when the blend relationship—linking fields—need to vary on a sheet-by-sheet basis, or when combining published data sources.

Important: Prior to version 2020.2, data blending was often the best way to handle data sources at different levels of detail. These can now be combined with relationships. Relationships have fewer technical limitations than data blending and are the recommended way of combining data when possible. Blending is only encouraged when it is the best method for your data or relationships are not available.

Steps for blending data

Data blending is performed on a sheet-by-sheet basis and is established when a field from a second data source is used in the view. To create a blend in a workbook already connected to at least two data sources, bring a field from one data source to the sheet—it becomes the primary data source. Switch to the other data source and use a field on the same sheet—it becomes a secondary data source. An orange linking icon will appear in the data pane, indicating which field(s) are being used to blend the data sources.



Note: images in this topic have not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

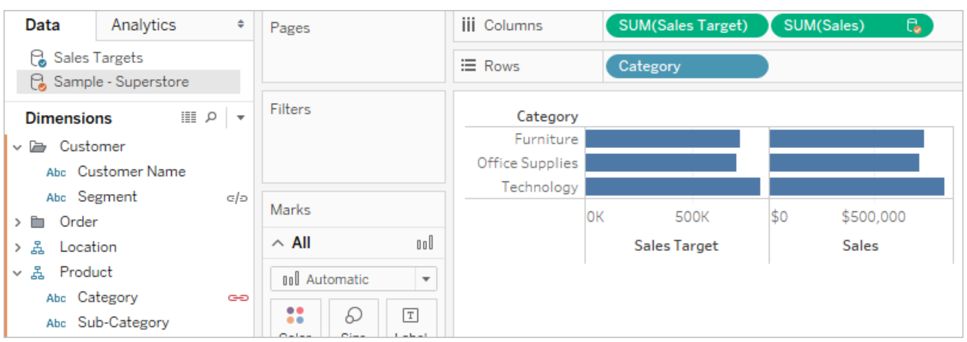
1. Ensure that the workbook has multiple data sources. The second data source should be added by going to **Data** > **New data source**.

   Tip: Adding another *connection* to the first data source enables relationships and joins on the Data Source page. Blending requires two or more distinct data sources, listed independently in the Data pane.

2. Drag a field to the view. Whichever data source this first field comes from will become the primary data source.
3. Switch to another data source and make sure there is a blend relationship to the primary data source.
    - If there is an orange linking field icon ( ), the data sources are automatically linked. As long as there is at least one active link, the data can be blended.
    - If there are gray, broken link icons ( ), click the icon next to the field that should link the two data sources. It will turn orange, representing an active link.
    - If a link icon does not appear next to the desired field, see **Define blend relationships for blending** on page 915.
4. Drag a field into the view from the secondary data source.

As soon as this second data source is used in the same view, a blend is established. In the example below, our primary data source is **Sales Targets** and the secondary data source is **Sample - Superstore**

- The primary data source is indicated with a blue check mark on the data source. Fields from the primary data source used in the view have no indication.
- The secondary data source is indicated with an orange check mark on the data source and an orange bar down the side of the Data pane. Fields from the secondary data source used in the view have an orange check mark.



Understand primary and secondary data sources

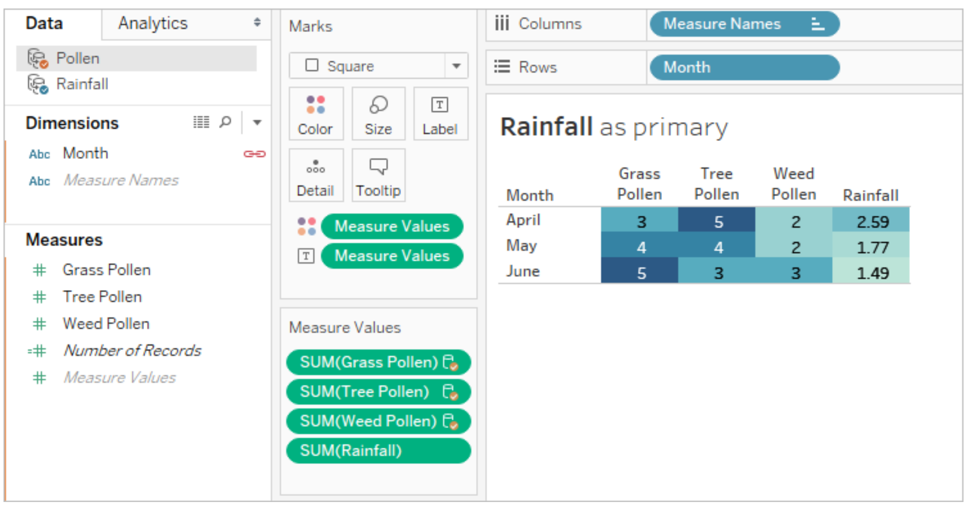
Data blending requires a primary data source and at least one secondary data source. The first data source used in the view becomes the primary data source and defines the view. This can restrict values from the secondary data source—only values that have corresponding matches in the primary data source appear in the view. This is comparable to a left join.

For example, if the primary data source has a Month field that only contains April, May, and June, any view built around months will only display April, May, and June, even if the secondary data source has values for twelve months. If the desired analysis involves all twelve months, try switching which data source is primary by rebuilding the sheet and using the other data source first.

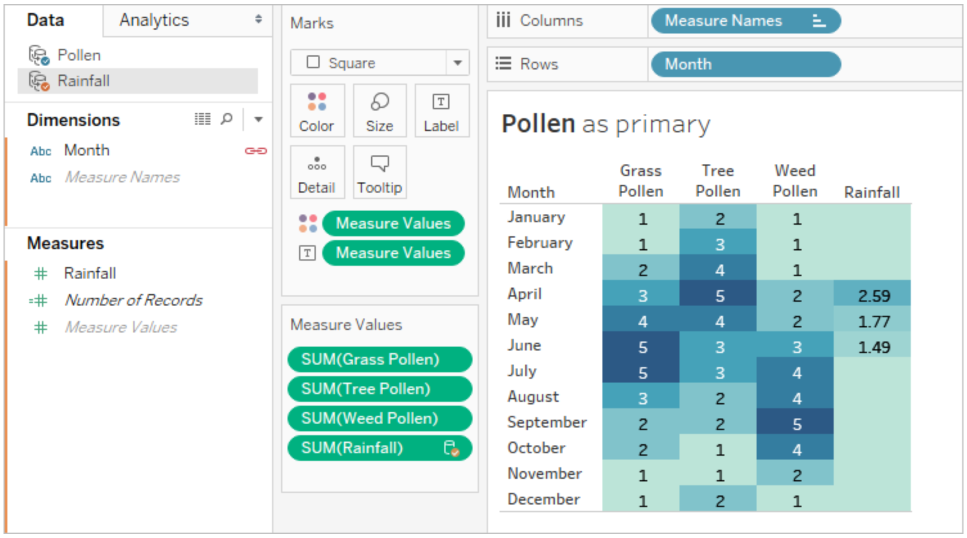
Learn more: the effects of data source order

The following examples below use the same data sources linked on the same field, and the viz is built the same way both times. The difference between the results is due to which data source is designated as the primary.

1. Here, the Month field from the **Rainfall** data source is brought into the view first. Because **Rainfall** only contains three months, when the **Pollen** data set is added as a secondary, only three months are brought into the view.



2. On another sheet, the Month field from the **Pollen** data set is brought into the view first. All twelve months are displayed. When the **Rainfall** data set is added as a secondary, **Rainfall** is only available for the three months in that data set.



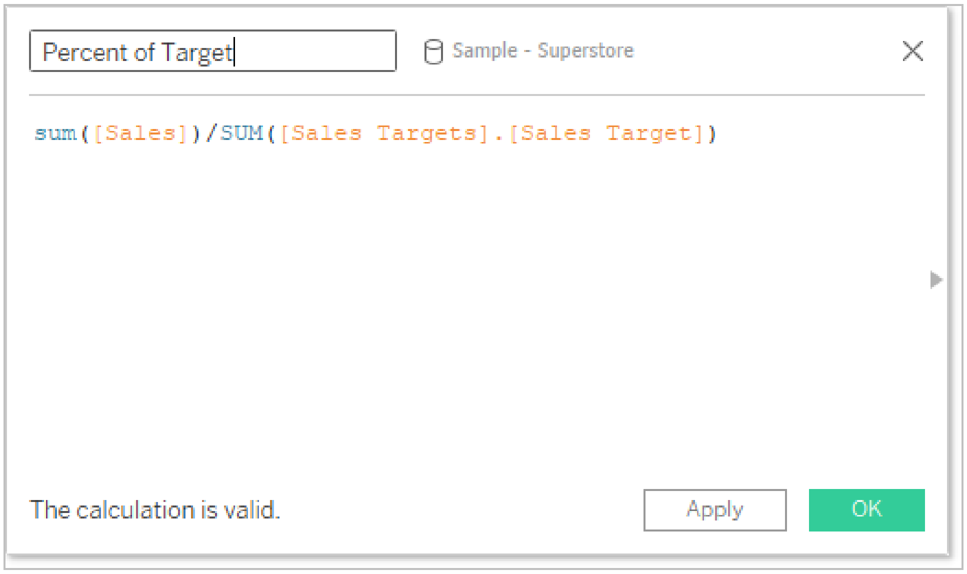
Work across blended data sources

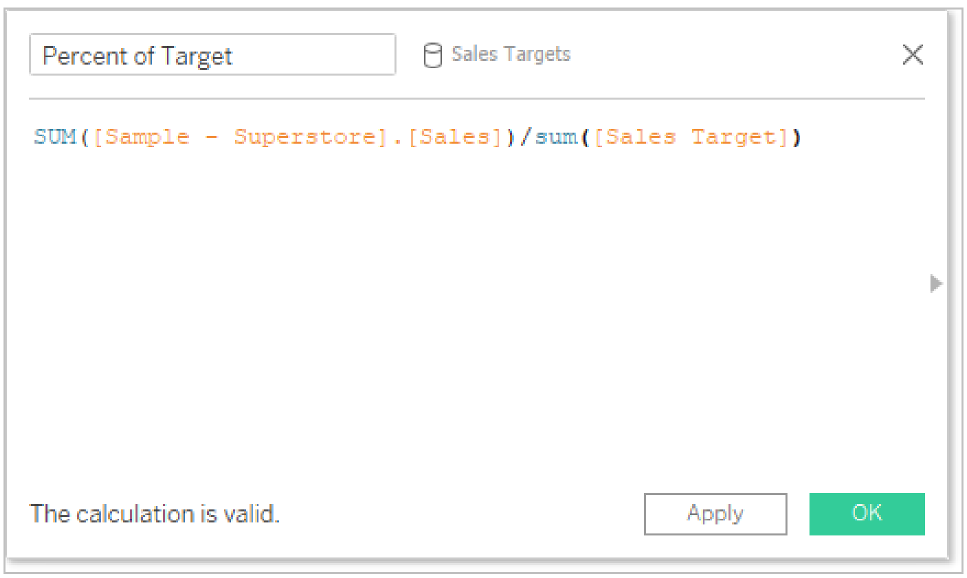
Due to the nature of a data blend, there are some things to keep in mind when working across blended data sources.

Performing calculations with fields from more than one data source can be slightly different than an ordinary calculation. A calculation must be created in one data source; this is indicated at the top of the calculation editor.

- **Aggregation**. Any fields used from another data source will come in with an aggregation—by default, SUM, but this can be changed. Because calculations cannot mix aggregate and non-aggregate arguments, fields from the data source where the calculation is being made must also be aggregated. (In the images below, the **SUM** aggregation was added automatically and the **sum** aggregation was added manually.)
- **Dot notation**. Any field referenced in the calculation that belong to another data source will refer to its data source using dot notation. (In the images below, for the calculation built in **Sample - Superstore**, the Sales Target field becomes **[Sales.Targets].[Sales Target]**. When the calculation is built in **Sales Targets**, the Sales field becomes **[Sample - Superstore].[Sales]**.)

- These are equivalent versions of the same calculation built in each data source. In both cases, this is SUM(Sales) / SUM(Sales Target).





In addition to handling calculations slightly differently, there are some limitations on secondary data sources. You may not be able to sort by a field from a secondary data source, and action filters may not work as expected with blended data. For more information, see **Other data blending issues** on page 936.

Define blend relationships for blending

In order for Tableau to know how to combine the data from multiple sources, there must be a common dimension or dimensions between the data sources. This common dimension is called the linking field. Active linking fields are identified in the Data pane for the secondary data source with an active link icon ( ) and potential linking fields are identified with a broken link icon ( ).

For example, in a blend of transactional and quota data, a geographic field might be the desired the linking field so you can analyze a region's quota and performance towards that quota.

> **Note**: In order for the blend to function, the linking fields must also share values or members. Tableau builds the view of blended data based on the shared values. For example, if **Color** is the linking field in both data sources, Tableau will match data on "Purple" from the primary and "Purple" from the secondary. But "Lt. Blue" will not map correctly to "Light Blue", so one of them would need to be realiased. Like renaming fields to help Tableau identify linking fields, you can edit aliases for members in those fields. For more information, see **Create Aliases to Rename Members in the View** on page 1161.

Establish a link

If the linking field in the primary and secondary data sources have the same name, Tableau automatically creates the relationship. When a primary data source has been established (that is, a field is in use in the view) and the secondary data source is selected in the Data pane, any fields with the same name between the two data sources will display a link icon (or) in the *secondary* data source. If the related field from the primary data source is used in the view, the link becomes active automatically.

If there are no link icons on the secondary data source, you may need to help Tableau establish the link in one of two ways:

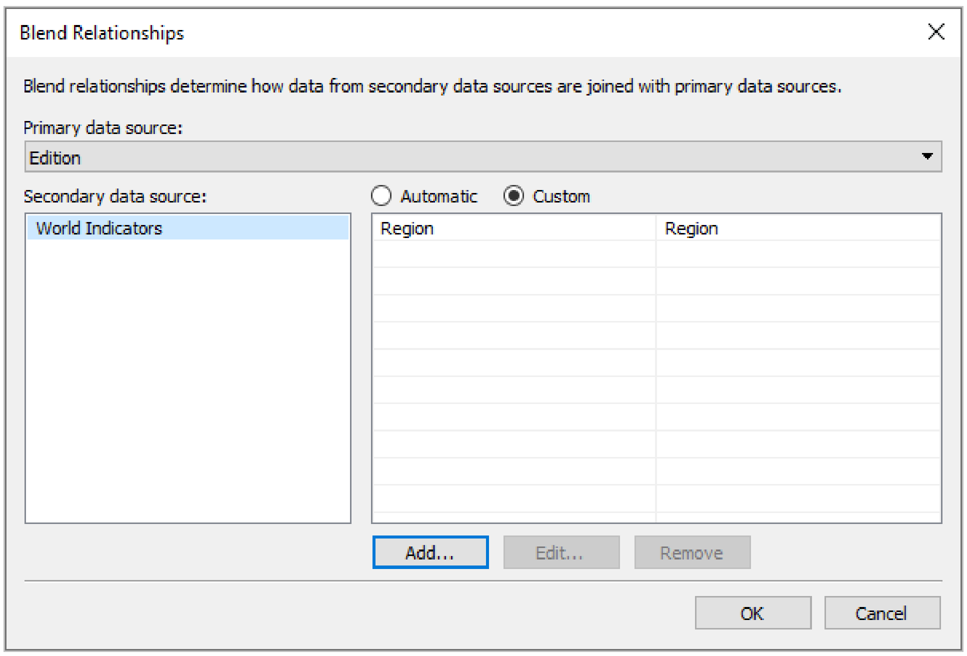
1. If common dimensions don’t have the same name (such as "Region" and "Sales Region"), renaming one will let Tableau identify them as common dimensions and establish the link.
2. Alternatively, you can manually define a relationship between the fields in the primary and secondary data sources. See below for more information on creating manual link relationship

There can be as many active or potential linking fields as necessary. Click the broken link icon ( ) in the data pane to make the relationship active.

Manually define a link relationship

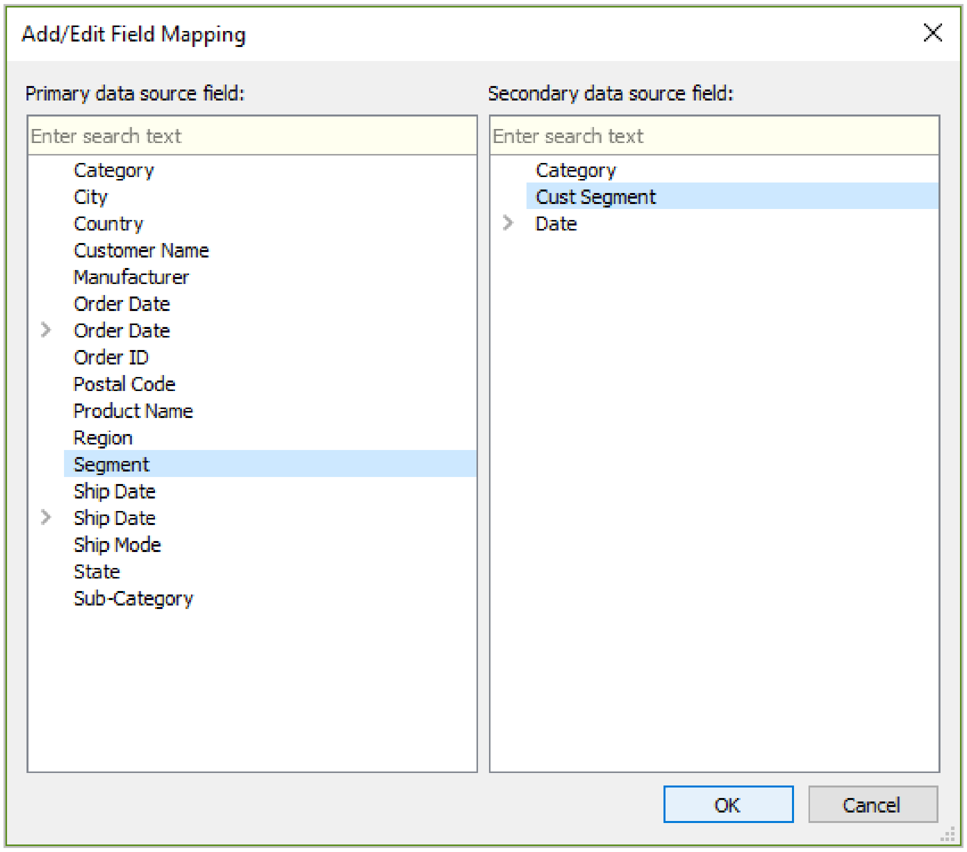
If your common dimensions do not share the same name, you can map the relationship between them manually.

1. Select **Data** > **Edit Blend Relationships...**.
2. In the Blend Relationships dialog box, verify that the primary data source is selected from the **Primary data source** drop-down list.
3. Select the secondary data source in the **Secondary data source** pane. Any existing, automatic blend relationships will be visible. Select **Custom** in the relationships list, and then click **Add**.



4. In the Add/Edit Field Mapping dialog box, do the following:
   a. Select a field from the primary data source.
   b. Select a field from the secondary data source to establish the linking field or the blend relationship between the data sources even though the fields do not have the same name.
   c. Click **OK**.

In this example, a mapping between **Segment** and **Cust Segment** is created.



Tip: For dates, the relationship can be specified precisely. Expand the date field and select the desired aspect of the date, such as exact date, month, year, etc.

5. Create as many field mappings as desired, then click **OK**.

Multiple links

As with relationships or joins, there are times when the links between the data sources are defined by more than one field. For example, if regional sales quotas are monthly, a blend between transactional sales data and quota data needs to be established on both region and month for the correct data to be brought together in the view. Multiple links can be active at the same time.

Learn more: the impact of multiple linking fields

When data is blended based on multiple fields, values are included in the view only where the combination of data from those fields matches across both data sets. Let's look at an example to understand this.

We have two tables, one for birds that were actually seen by birdwatchers, and one for birds that were reported as seen.

Birds seen

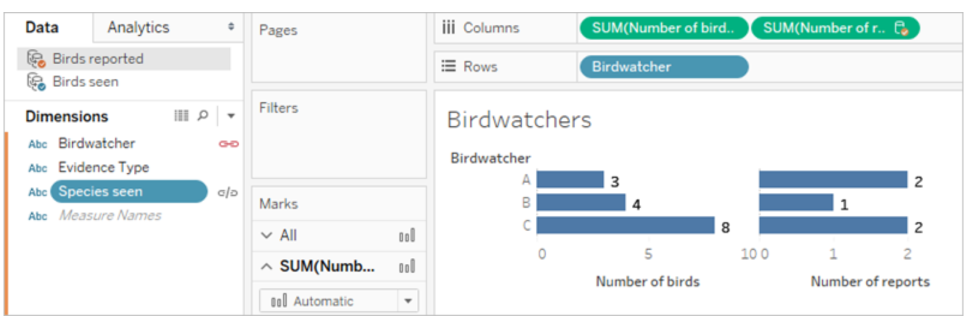
Birdwatcher	Species seen	Location	Number of birds
A	Grackle	Beach	1
	Oriole	Pier	2
B	Kiskadee	Beach	4
C	Kiskadee	Beach	5
	Oriole	Tree	3

and

Birds reported

Birdwatcher	Species seen	Evidence Type	Number of reports
A	Kiskadee	journal entry	1
	Oriole	photograph	1
B	Grackle	journal entry	1
C	Grackle	journal entry	1
	Kiskadee	photograph	1

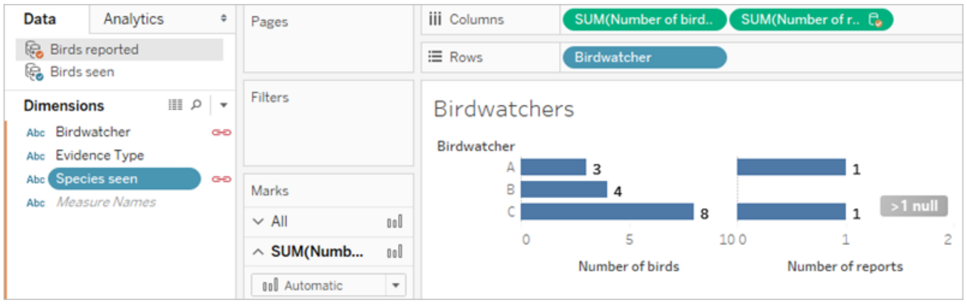
If we set up a blended view with the fields **Birdwatchers** and **Number of birds** from the primary data source (**Birds seen**) and bring in the field **Number of reports** from the secondary data source (**Birds reported**), Tableau automatically blends on **Birdwatcher**.



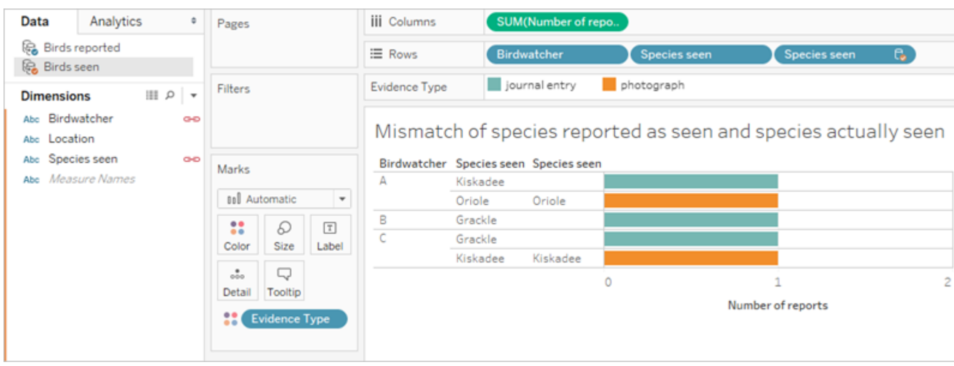
Note: images in this topic have not been updated to reflect the most current UI. The Data pane no longer shows Dimensions and Measures as labels.

We see that birdwatcher A saw three birds and made two reports, B saw four birds and made one report, and C saw 8 birds and made two reports.

But there's another possible linking field, **Species seen**. Why not blend on this, too? Will it make a difference?



It makes quite a dramatic difference. Now we see that there is only one report each for birdwatchers A and C, and B has a null. What's going on?



It turns out that these birdwatchers aren't very honest. When they only reported their sightings based on a journal entry (blue bars in the image above), the species they reported seeing didn't match what they actually saw (note the nulls in the second column, from the **Birds seen** secondary data source). When they backed up the report with a photograph (orange bars), they were honest (both columns of **Species seen** match). Because three reports did not match on species, those rows of data were dropped when the **Birdwatcher** and **Species seen** fields were used as a linking field. The view only shows data where values for both linking fields match.

Be careful when linking on multiple fields. Although it can be very easy to click the icon and establish an active link, over-linking or linking on undesired fields can have a serious impact on the analysis.

Differences between joins and data blending

Data blending simulates a traditional left join. The main difference between the two is when the aggregation is performed. A join combines the data and then aggregates. A blend aggregates and then combines the data.

Left join

When you use a left join to combine data, a query is sent to the database where the join is performed. A left join returns all rows from the left table and any corresponding rows from the right table. The results of the join are then sent back to Tableau and aggregated for display in the visualization.

A left join takes all rows from the left table. The common columns are **User ID** and **Patron ID**; where there is corresponding information from the right table, that data is returned. Otherwise, there is a null.

User ID	District	Level	Type
1	2	3	G
2	3	4	J
4	5	6	M
1	2	3	W

Branch	Patron ID	District	Level
A001	1	2	3
B001	2	3	4
C001	1	2	3

User ID	District	Level	Branch	Type
1	2	3	A001	G
1	2	3	A001	W
1	2	3	C001	G
1	2	3	C001	W
2	3	4	B001	J
4	5	6	*null*	M

Suppose you have the same tables, but flip the order. This new left join produces different results. Again, a left join takes all the data from the new left table, but essentially ignores a row from the right table. The row of data for User ID = 4 is not included because there is no row for Patron ID = 4 in the left table.

Branch	Patron ID	District	Level
A001	1	2	3
B001	2	3	4
C001	1	2	3

User ID	District	Level	Type
1	2	3	G
2	3	4	J
4	5	6	M
1	2	3	W

User ID	District	Level	Branch	Type
1	2	3	A001	G
1	2	3	A001	W
1	2	3	C001	G
1	2	3	C001	W
2	3	4	B001	J

Data blending

When you use data blending to combine data, a query is sent to the database for each data source that is used on the sheet. The results of the queries are sent back to Tableau as aggregated data and presented together in the visualization.

> **Note**: Aggregating measures is straightforward—we can take the sum, average, maximum, or other aggregation of a number with ease. Measure values are aggregated based on how the field is aggregated in the view. However, all fields from a secondary data source must be aggregated. How does that work for dimensions? Dimension values are aggregated using the **ATTR** aggregate function, which returns a single value for all rows in the secondary data source. If there are multiple values contained in those rows, an asterisk (*) is shown. This can be interpreted as "there are multiple values in the secondary data source for this mark in the view".

The view uses all values from the primary data source (functioning as the left table) and the corresponding rows from the secondary data source (the right table) based on the linking field (s).

Suppose you have the following tables. If the linking fields are **User ID** and **Patron ID**, not all values can be a part of the resulting table because of the following:

- A row in the left table does not have a corresponding row match in the right table, as indicated by the null value in the results.
- There are multiple corresponding values in the rows in the right table, as indicated by the asterisk (*) in the results.

User ID	District	Level	Type
1	2	3	G
2	3	4	J
4	5	6	M
1	2	3	W

Branch	Patron ID	District	Level
A001	1	2	3
B001	2	3	4
C001	1	2	3

User ID	District	Level	Branch	Type
1	2	3	*	G
2	3	4	B001	J
4	5	6	*null*	M
1	2	3	*	W

When measures are involved, they are also aggregated, as seen below:

Branch	Patron ID	District	Level	Fines
A001	1	2	3	10.00
B001	2	3	4	20.00
C001	1	2	3	30.00

User ID	District	Level	Type
1	2	3	G
2	3	4	J
4	5	6	M
1	2	3	W

Branch	Patron ID	District	Level	Fines
*	1	2	3	40.00
B001	2	3	4	20.00
*	1	2	3	40.00

User ID	District	Level	Type	Branch	Fines
1	2	3	G	*	40.00
2	3	4	J	B001	20.00
4	5	6	M	*null*	*null*
1	2	3	W	*	40.00

Important: an asterisk (*) in a view with blended data indicates multiple values. This can be resolved by ensuring there is only one matching value in the secondary data source for each mark in the primary data source, potentially by swapping the primary and secondary data sources. For more information, see **Troubleshoot Data Blending** on page 930.

Data blending at a glance

- Data blending occurs on a sheet-by sheet basis.
- The order in which fields are used determines which data source is the primary versus the secondary data source.

- The primary data source is indicated with a blue check mark, any secondary data sources and fields from secondary data sources have an orange check mark.
- Linking fields can be automatically determined based on shared field names or the blend relationship can be manually created.
- Data blending behaves similarly to a left join, which may result in missing data from the secondary data source.
- Asterisks (*) may appear. This indicates multiple dimension values in a single mark, because data blending takes aggregated results and combines them in the view.
- A secondary data source can be used to re-alias the field values in a primary data source. For more information, see **Alias Field Values Using Data Blending** below.

Data blending limitations

- There are some data blending limitations around non-additive aggregates, such as COUNTD, MEDIAN, and RAWSQLAGG. For more information, see **Troubleshoot Data Blending** on page 930
- Blended data sources cannot be published as a unit. Instead, publish each data source separately (to the same server) and then blend the published data sources.
- Data from secondary data sources must always be aggregated in calculations.
- If you are blending a cube data source, it must be the primary data source.

Alias Field Values Using Data Blending

Data blending is a method for combining data. Data blending works by supplementing the data in the primary data source with the data in the secondary data source.

Aliasing is the alternative name that you can assign to a value in a dimension field. You can use aliases to rename specific values within a dimension. This can be useful when you want to show more relevant or descriptive dimension values in your view than what the original data provides.

You can use data blending as a method for re-aliasing values in a data source by using field values from another data source. To do this, there must be a field in the secondary data source that contains aliases for a field in the primary data source.

For example, suppose you have a primary data source that contains information on fruit, including fruit names and code values that the fruit are associated with. A secondary data source contains a field with more descriptive code values.

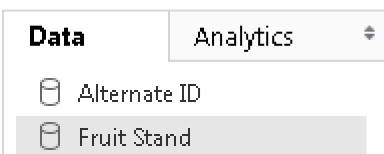
Primary data source		Secondary data source
Fruit	**Fruit ID**	**Alternate ID**
Apple	A	APP
Banana	B	BAN

When you re-alias a field in your primary data source, you can create a view that replaces the original code values with the more descriptive code values.

Note: After you have used the secondary data source to provide aliases for the primary data source, you no longer need the secondary data source and can close it.

To alias field values

1. Set up the primary and secondary data sources and establish a relationship between the data sources. For more information, see **Blend Your Data** on page 910. The data sources used in this example are Fruit Stand and Alternate ID.

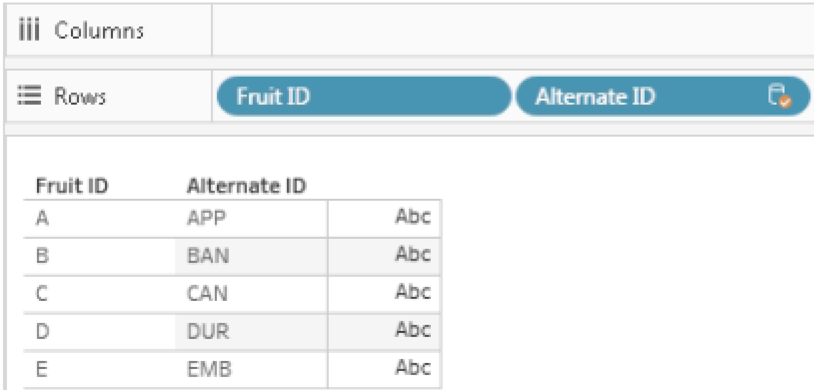
2. Select the primary data source in the **Data** pane, and then drag the field that you want to alias to the view.

   For this example, Fruit Stand is the primary data source.

3. Select the secondary data source in the **Data** pane, and drag the field that contains the alias information to the view.

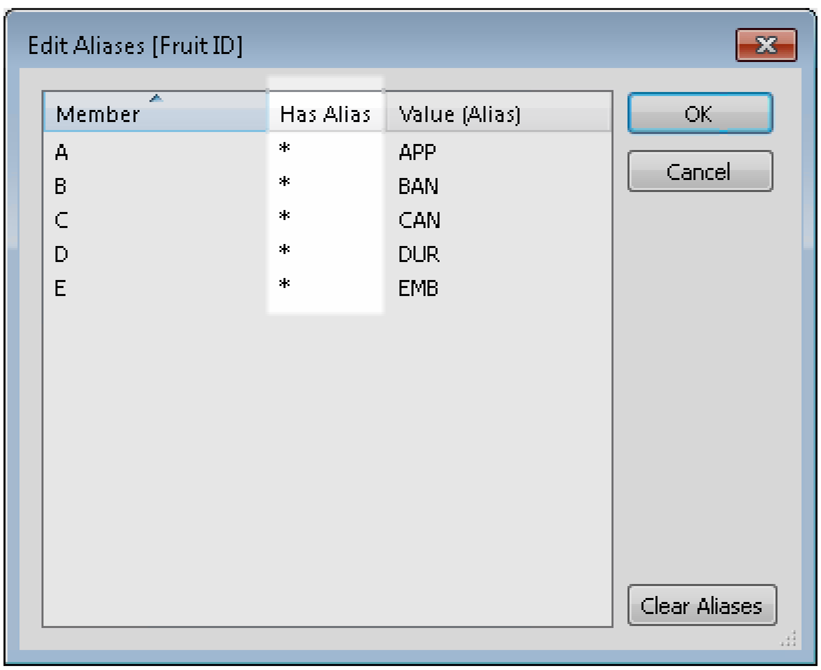
   For this example, Alternate ID is the secondary data source. **Note:** The field you are using to alias the field values in the primary data source cannot be the linking field you are using to link the primary and secondary data sources.

   In this example, Alternate ID field in the secondary data source contains the aliasing information needed by the Fruit Stand field in the primary data source.



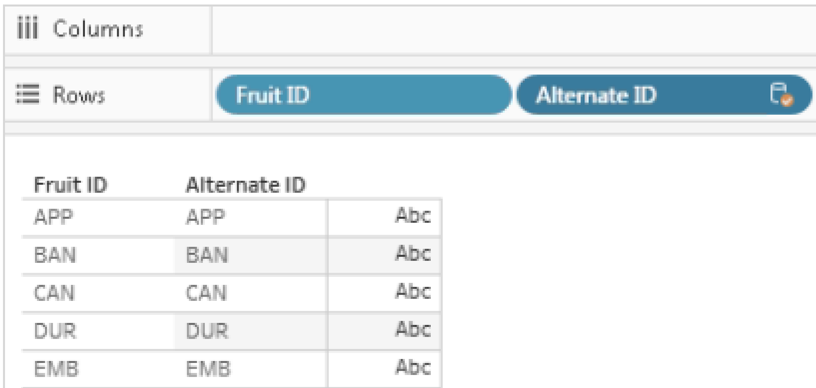
4. Right-click (control-click on a Mac) the field from the secondary data source and select **Edit Primary Aliases**.

   Values in the primary data source must have corresponding alias values in the secondary data source. An asterisk under **Has Alias** confirms that the data sources have corresponding alias values .



Note: If asterisks don't appear under **Has Alias**, then either the secondary data source doesn't have the corresponding aliases to match the field values in the primary data source, or there is more than one field value in the primary data source that has the same alias in the secondary data source. In cases like this, aliasing field values in the primary data source with data from the secondary data source is not possible.

5. Click **OK**. The alias values in the secondary data source replace the field values in the primary data source. In this example, the more descriptive code values from Alternate ID field replaces the values in the Fruit ID field.



6. (Optional) Select the secondary data source from the **Data** menu, and select **Close**.

To watch a video that demonstrates aliasing field values using data blending, see Cleaning Data by Bulk Re-Aliasing.

Bring a Field into the Primary Data Source

Data blending is a method for combining data. Data blending works by supplementing the data in the primary data source with the data in the secondary data source.

In cases where you might only need just *some* of the data in the secondary data source, you can create a primary group to bring only the fields that you need into the primary data source. This eliminates the need for the secondary data source, which reduces the size of the workbook and can improve workbook performance.

For example, suppose you have two data sources: Superstore and Population. The Superstore data source contains lots of data, but most importantly, it contains data about states and the region each of the states is associated with. The Population data source contains state and population data, but doesn't contain any data about regions. In this case, you can blend the Population and Superstore data sources to create a primary group. Then so that you can pull in the Region field from the Superstore data source into the Population data source, thereby eliminating the need for the Superstore data source all together.

To blend two data sources and create a primary group

1. Download and open the following workbook from Tableau Public.

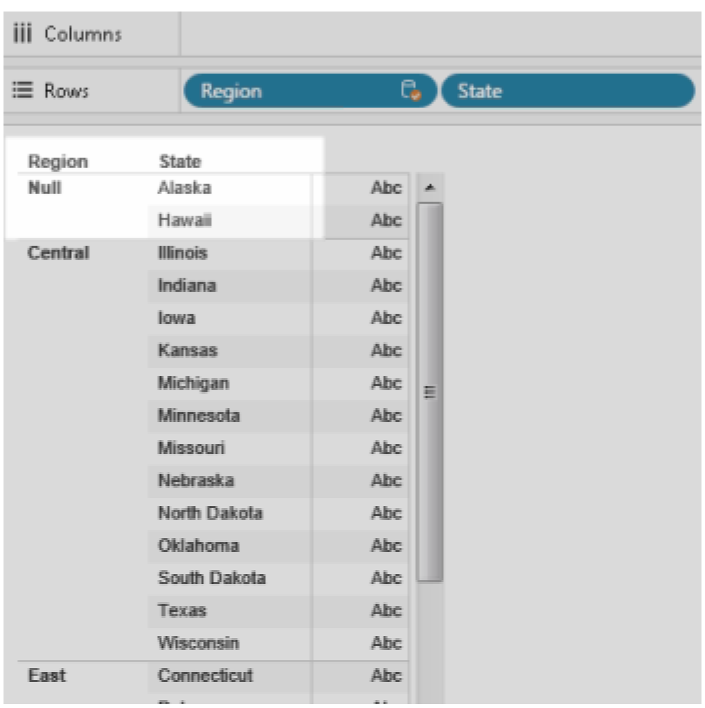
    a. Go to https://public.tableau.com/profile/tableau.docs.team#!/vizhome/BringaFieldintothePrimaryDataSource/Sheet1.

    b. Click **Download Workbook** in the upper-right corner and then open the workbook.

    In the workbook, the **State** dimension is already on the **Rows** shelf, making it the first field in the view and establishing the Population data source as the primary data source for the sheet. The Superstore data source is also included in the workbook and functions as the secondary data source.

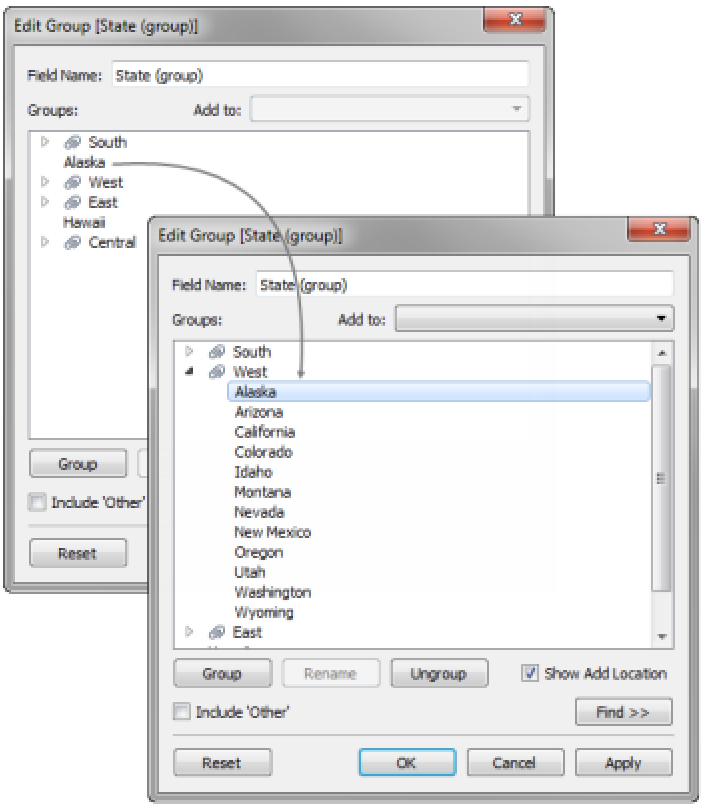
2. In the **Data** pane, select Superstore, and then drag the **Region** dimension to the **Rows** shelf and place it in front of **State**.

    The view shows the regions from the Superstore data source (secondary) and the states in the Population data source (primary) that are associated with those regions. There are two states, Alaska and Hawaii, from Population data source (primary) that are not associated with a specific region. These two states are assigned to a "null" region.



3. Right-click the **Region** field in the **Rows** shelf from the Superstore data source (secondary), and select **Create Primary Group**.

4. In the Edit Group dialog box, drag Alaska and Hawaii (from the "null" region) to the **West** region. This task assigns the states in the Population data source (primary) in the "null" region into a region.



5. Click **OK** to close the Edit Group dialog box.

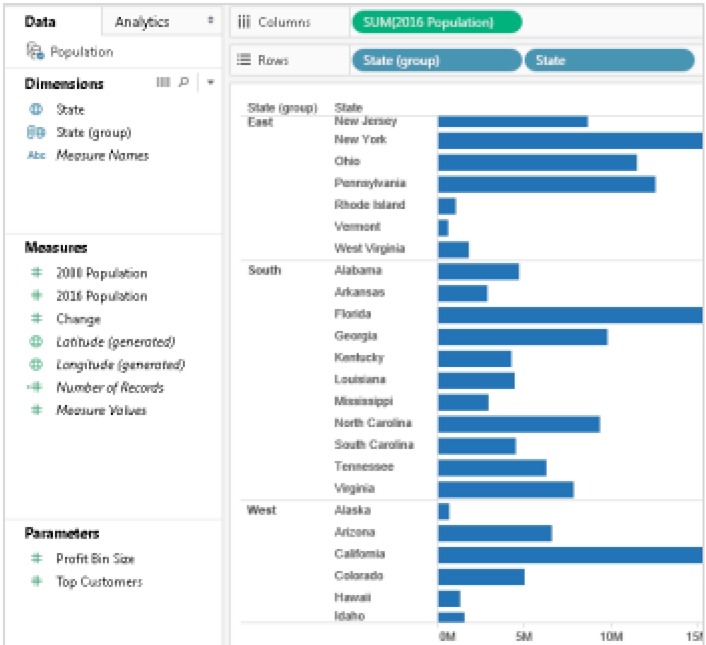
6. In the **Data** pane, select the State data source (secondary). Now you see a new group field listed in the Population data source (primary) called **State (group)**.

7. Drag the **State (group)** dimension over the Region dimension on the **Rows** shelf to replace it.

8. Drag the **2016 Population** measure to the **Columns** shelf to see Population based on the regions from the Superstore (secondary) data source.

9. (Optional) You can close the Superstore data source (secondary), and then continue with your analysis. You can also publish the data source or workbook without the secondary

data source. This is because the field you were using from the secondary data source, **Region**, is replaced by the primary group field that you created in the primary data source.



To watch a video that demonstrates another example of the Create a Primary Group functionality, see Cleaning Data by Bulk Re-Aliasing.

Troubleshoot Data Blending

When you use data blending to combine your data from different data sources, there are a few common issues that might arise. This topic lists them and describes how you can respond to each situation.

Common warnings and errors when blending data sources

No relationship to the primary data source

When you drag a field from a secondary data source to the view, you might see a warning that says:

> *Fields cannot be used from the [name of secondary data source] data source, because there is no blend relationship to the primary data source. In the Data pane, switch to the [name of secondary data source] data source, and click at least one link icon to blend these data sources.*

This warning occurs when you have no active links in the secondary data source

For example, suppose you have two data sources that are related by the **State** and **Date** fields.

At least one of these fields must have the active link icon ( ) next to it in the secondary data source. You can make the link active by clicking the link icon in the **Data** pane or by using the linking field from the primary data source in the view.

The secondary data source may not have any links to the primary data source. Look in the **Data** pane for the link icon. Tableau automatically links fields that have the same name. If your fields do not have the same name, you must define a link between them. For more information, see **Blend Your Data** on page 910.

Primary and secondary connections are from tables in the same data source

When you drag a field from a secondary data source to the view, you might see a warning that says:

> *The primary and secondary connections are from tables in the same data source. Instead of linking the connections, use the Data menu to join the data. Joins can integrate data from many tables and may improve performance and filtering.*

This warning occurs when the workbook contains separate data sources that connect to the same database. Though you can combine data in this way, Tableau recommends that you use a join or relationship to combine data from the same database instead, as they can leverage some of the database's native optimizations in ways blending can't.

Cannot blend the secondary data source because one or more fields use an unsupported aggregation

Data blending has some limitations regarding non-additive aggregates such as COUNTD, MEDIAN, and RAWSQLAGG. Non-additive aggregates are aggregate functions that produce results that cannot be aggregated along a dimension. Instead, the values have to be calculated individually. All Number functions, except for MAX and MIN, are non-additive aggregates. For more information, see **Tableau Functions (by Category)** on page 2267.

These limitations cause certain fields in the view to become invalid under certain circumstances. If you hover your mouse cursor over one of these invalid fields, you see the following error:

> *Cannot blend the secondary data source because one or more fields use an unsupported aggregation.*

This error can occur for one of the following reasons:

- **Non-additive aggregates from the primary data source:** Non-additive aggregates are only supported in the primary data source if the data in the data source comes from a relational database that allows the use of temporary tables. To work around this issue, consider creating an extract of your data source. Extracts support temporary tables.
- **Non-additive aggregates from the secondary data source**: Non-additive aggregates are only supported in the secondary data source if the linking field from the primary data source is included in the view and no non-aggregated dimensions from the secondary data source is included in the view. Some number functions can still be used if they include an additive aggregation. For example, use ROUND(SUM([Sales]),1) instead of ROUND([Sales],1).
- **Non-additive aggregates from a multi-connection data source that uses a live connection:** Multi-connection data sources that connect to data using a live connection do not support temporary tables. Therefore, using a multi-connection data source that connects to data using a live connection prohibits the use of blending functionality with non-additive aggregates. To work around this issue, consider creating an extract of your multi-connection data source. Extracts support temporary tables.
- **LOD expressions from the secondary data source:** This error can also appear

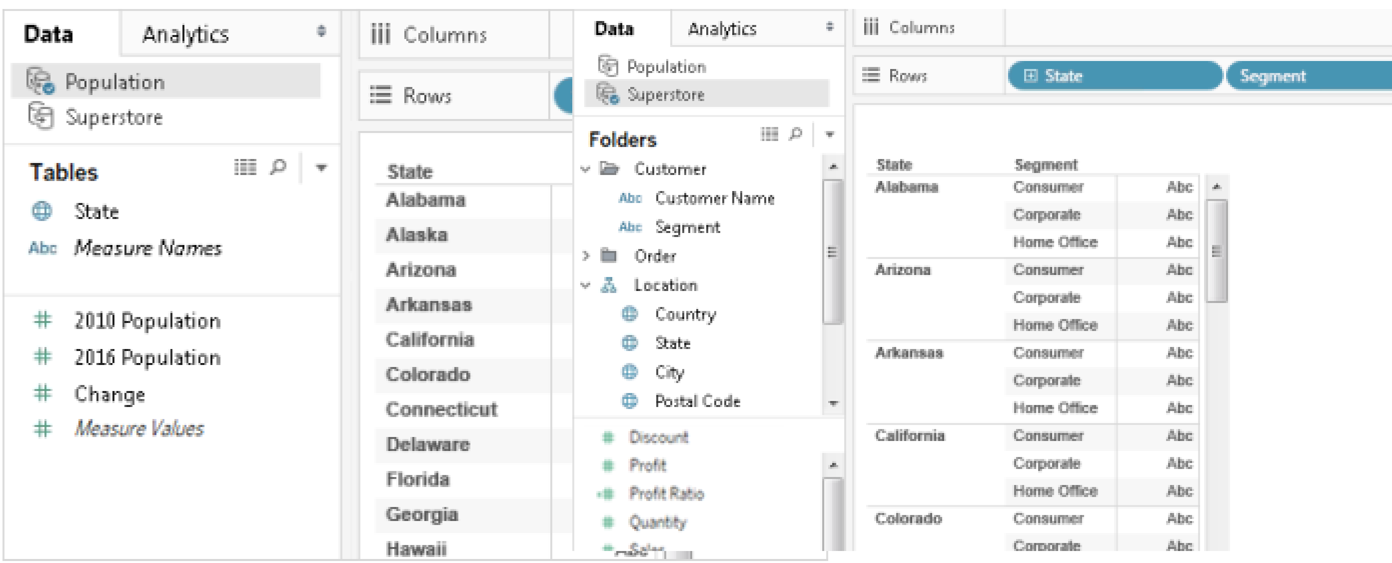
when you use a level of detail expression in a view that uses data blending. To resolve the error, make sure the linking field in the primary data source is in the view before you use an LOD expression from the secondary data source and remove any dimensions, including dimension filters, from the secondary data source.

Asterisks show in the sheet

When you blend data, make sure that there is only *one* matching value in the secondary data source for each mark in the primary data source. If there are multiple matching values, you see an asterisk in the view that results after you blend data.

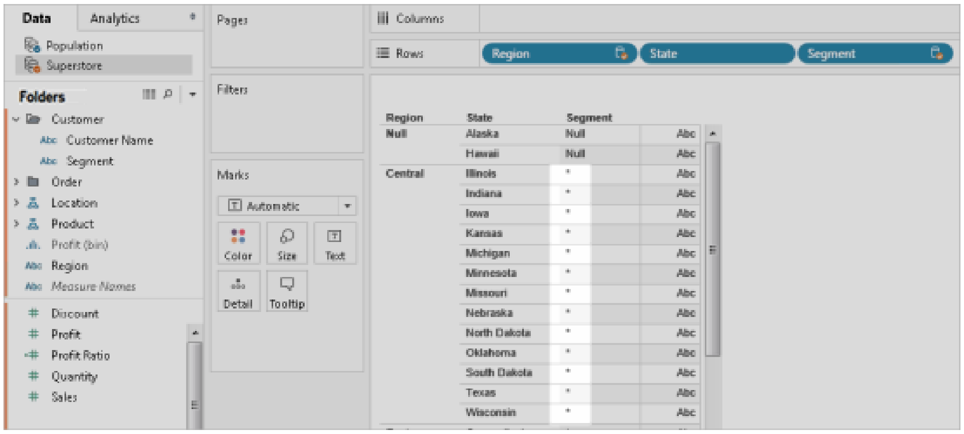
For example, suppose you have two data sources: Population and Superstore. The primary data source, Population, has a field called **State**. The secondary data source, Superstore, contains fields called **State** and **Segment**. In the secondary data source, each state has multiple segments. Alabama, for example, has three segments: Consumer, Corporate, and Home Office.

Primary data source **Secondary data source**



When you blend the two data sources on the **State** field, you create a link where individual state values (in the primary data source) can have multiple segment values (in the secondary data source). In this case, multiple values for segments in the secondary data source for each corresponding state value in the primary data source cause asterisks to appear in the view, as demonstrated by the images below.

Blended data



Resolve asterisks in the sheet

Avoid asterisks in the sheet by making sure that there is only *one* matching value in the secondary data source for each mark in the primary data source. Though the way you ensure this depends on the data and the view you're trying to create, consider one of the following suggestions to resolve asterisks in the sheet.

- Add field with a higher level of granularity from the primary data source to the sheet.

  For example, suppose you have a view similar to the example used above. Consider adding a field such as City into the sheet because its values are at a higher level of granularity than a field like State.

- Rebuild the view to switch the primary and secondary data sources with each other. In general, you should make the data source whose values have a higher level of granularity the primary data source.

  For example, suppose you have a view similar to the example used above. Consider making Superstore the primary data source and Population the secondary data source.

Null values appear after blending data sources

Null values can sometimes appear in place of the data you want in the view when you use data blending. Null values can appears for a few reasons:

- The secondary data source does not contain values for the corresponding values in the primary data source.

- The data types of the fields you are blending on are different.
- The values in the primary and secondary data sources use different casing.

Data blending works by supplementing the data in the primary data source with data from the secondary data source based on the linking field. This means Tableau takes all the data in the primary data source, and only the corresponding matches in the secondary data source. If Tableau cannot identify at least one matching value in the secondary data source for the corresponding value in the primary data source, null values appear.

You can resolve this issue by doing the following:

- **Insert data in the secondary data source:** Insert missing data in the secondary data source so that all records in the primary data source have a match. For example, suppose your primary data source contains values for all 50 U.S. states, but your secondary data source only has values for 30 U.S. states. To resolve this issue, you can add values for the remaining 20 U.S. states to the underlying data set of the secondary data source. Alternatively, you might be able to **Bring a Field into the Primary Data Source** on page 927 or **Alias Field Values Using Data Blending** on page 924 to work around null values if you cannot modify the underlying data.
- **Verify data types in the primary and secondary data sources match:** For example, both data sources should use string data for the fields you are blending on, or both fields should contain numbers. You can't blend fields where one field contains numbers and the other contains strings.
- **Verify the casing of the values in the primary and secondary data sources match:** If one data source uses a mix of upper- and lowercase names, and the other uses only uppercase, you'll see null values. For more information, see Nulls Show When Data Blending.

Blending issues after publishing data sources

When you make a field-level customization that blends two data sources, the customization is based on one of the data sources, the primary data source. Then, when the primary and secondary data sources are published to Tableau Server or Tableau Online, the primary data source, which contains the customization (i.e., the reference to the secondary data source) no longer works. The customization no longer works because the blend relationship between the primary and secondary data sources is lost. This is because the blend relationship between the primary and secondary data sources is defined at the workbook level.

For example, suppose you have two data sources: Store - Main and Store - West. You create a field-level customization, like a calculation, that combines the sales totals from the primary data source, Store - Main, with the sales totals from the secondary data source, Store - West. As soon as you publish the data sources to Tableau Server or Tableau Online, the calculation in Store - Main no longer works. This is because the calculation refers to the secondary data source, Store - West, whose location is now undetermined.

To work around this scenario, do the following:

1. Before creating any field-level customizations, publish each data source first.
2. Connect to each data source.
3. Create a workbook that contains the field-level customizations that you need.
4. Publish the workbook to Tableau Server.

   You can use this workbook as a "template" that can be shared and downloaded by you and others.

Blending with a cube (multidimensional) data source

Cube data sources can only be used as the primary data source for blending data in Tableau. They cannot be used as secondary data sources.

Other data blending issues

For other issues that emerge while blending data, see the following links.

Sort is not available

- For issues sorting on a calculated field, see Sorting by Fields is Unavailable for Data Blended Measures.
- For issues with a computed sort, see Sort Options Not Available from Toolbar When Data Blending.

Actions do not behave as expected

- Fields from the secondary data source cannot be added to a URL action. See Fields from Blended Data Source Unavailable for URL Actions.

- Action filters are not behaving as expected. See Action Filters with Blended Data Not Working as Expected.

Unexpected values and field changes

- Duplicate totals after every date value in the view. See Issues with Blending on Date Fields.
- Underlying data shows different values than blended data. See Underlying Data from Secondary Data Source Not Displayed or Consistent with Blended Data.

Blend on Summary Data

Data blending is a method for combining data. Data blending works by supplementing the data in the primary data source with the data in the secondary data source.

When one of the data sources that you're working with is large, you can reduce query time by using data blending to combine data in the primary data source with summary data from the secondary (large data source).

For example, suppose you have two data sources: Store - North and Store - South. Store - North is the primary data source and Store - South is the large secondary data source. To reduce query times when working with a large data source like Store - South, you can use data blending to blend only on the summary data of the large secondary data source (Store - South) with the data in the smaller and presumably faster primary data source (Store - North).

To blend on summary data, you can do the following:

1. Create a relationship between data sources.
2. Bring summary data from the secondary data source into the primary data source.
3. Compare primary data source data to summary data in the secondary data source.

Step 1: Create a relationship between the primary and secondary data sources

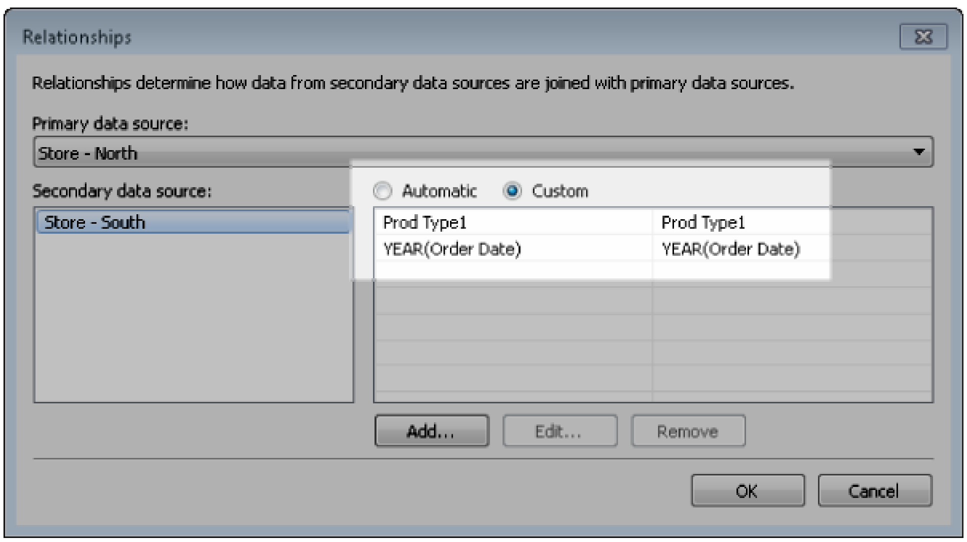
1. Download and open the following from Tableau Public.
    a. Go to https://public.tableau.com/profile/tableau.docs.team#!/vizhome/BlendonSummar

yData/Sheet1.

b. Click **Download Workbook** in the upper-right corner.

In the workbook, the Order Date dimension is already on the Columns shelf, making it the first field in the view and establishing *Store - North* as the primary data source for the sheet. The *Store - South* data source is also included in the workbook and functions as the large secondary data source.

2. Select **Data** > **Edit Relationships**.
3. In the Relationships dialog box, select **Custom**, and then click **Add**.
4. In the Add/Edit Field Mapping dialog box, in both lists, click the arrow next to Order Date, select **Year(Order Data)**, and then click **OK**.
5. In the Relationships dialog box, click **Add** again.
6. In the Add/Edit Field Mapping dialog box, in both lists, click **Prod Type 1**, and then click **OK**.



Step 2: Bring summary data from the secondary data source into the primary data source

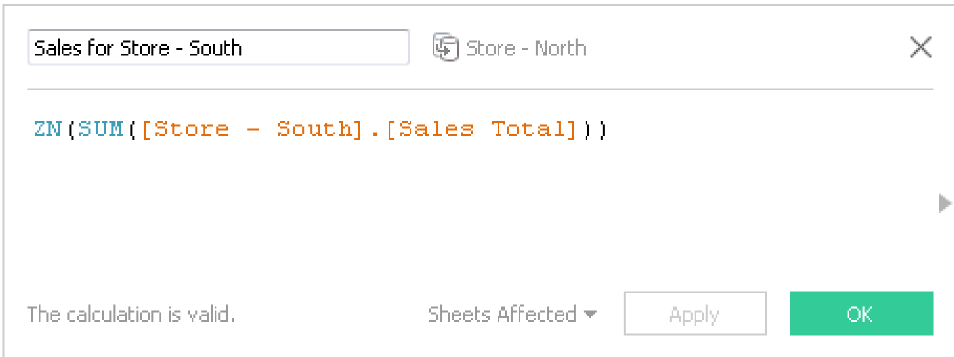
1. In the Data pane, select the Store - North data source.
2. Select **Analysis** > **Create Calculated Field**.
3. In the calculation dialog box, do the following:

a. In the name text box, enter **Sales for Store - South**.

b. In the formula text box, type the ZN function, the SUM function, and the Sales Totals field from the Store - South data source.

Your formula should look like the following: `ZN(SUM([Store - South].[Sales Total]))`

This formula pulls in just the sales data from the large secondary data source.



4. Select **Analysis** > **Create Calculated Field**.

5. In the calculation dialog box, do the following:

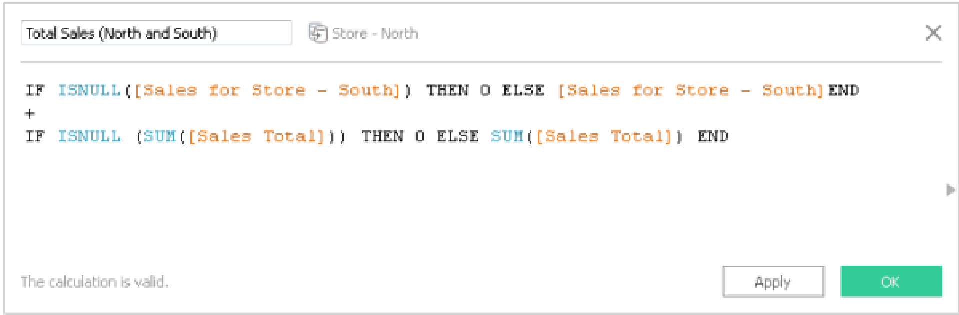
a. In the name text box, enter **Total Sales (North and South)**.

b. In the formula text box, type the following:

```
IF ISNULL([Sales for Store - South]) THEN 0 ELSE [Sales for Store - South]END

+

IF ISNULL (SUM([Sales Total])) THEN 0 ELSE SUM([Sales Total]) END
```

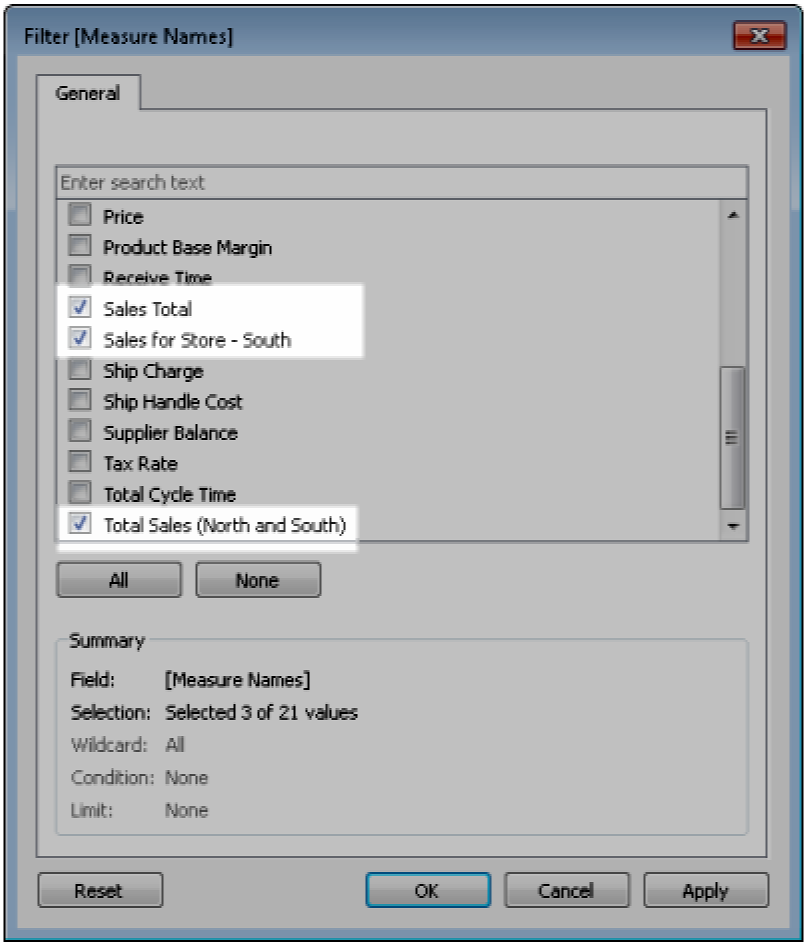
This formula adds the sales data from the large secondary data source to the sales data in the primary data source to get combined sales for Store - South and Store -

North.



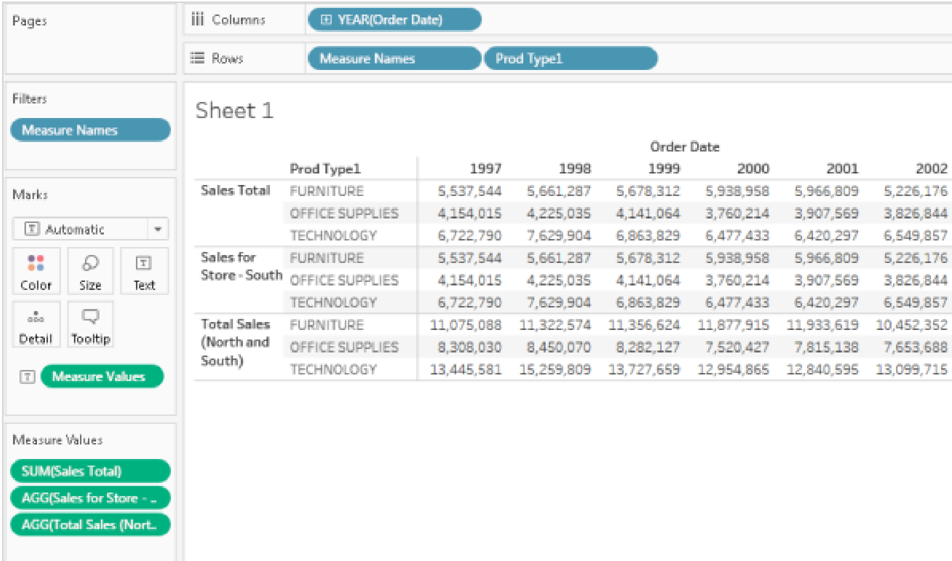
Step 3: Compare primary data source data to summary data of secondary data source

1. From the Data pane, drag **Order Date** to the Columns shelf and **Prod 1** to Rows shelf.
2. Drag **Measure Names** to the Rows shelf, and then to the Filters shelf.
3. In the Filter dialog box, click **None**, and then select **Sales Total**, **Sales for Store - South**, **Total Sales (North and South)** check boxes.



4. From the Data pane, drag **Measure Values** to Text.

The final result is a view that summarizes primary data source and secondary data source data on the Order Year and product category.



	Prod Type1	1997	1998	1999	2000	2001	2002
Sales Total	FURNITURE	5,537,544	5,661,287	5,678,312	5,938,958	5,966,809	5,226,176
	OFFICE SUPPLIES	4,154,015	4,225,035	4,141,064	3,760,214	3,907,569	3,826,844
	TECHNOLOGY	6,722,790	7,629,904	6,863,829	6,477,433	6,420,297	6,549,857
Sales for Store - South	FURNITURE	5,537,544	5,661,287	5,678,312	5,938,958	5,966,809	5,226,176
	OFFICE SUPPLIES	4,154,015	4,225,035	4,141,064	3,760,214	3,907,569	3,826,844
	TECHNOLOGY	6,722,790	7,629,904	6,863,829	6,477,433	6,420,297	6,549,857
Total Sales (North and South)	FURNITURE	11,075,088	11,322,574	11,356,624	11,877,915	11,933,619	10,452,352
	OFFICE SUPPLIES	8,308,030	8,450,070	8,282,127	7,520,427	7,815,138	7,653,688
	TECHNOLOGY	13,445,581	15,259,809	13,727,659	12,954,865	12,840,595	13,099,715

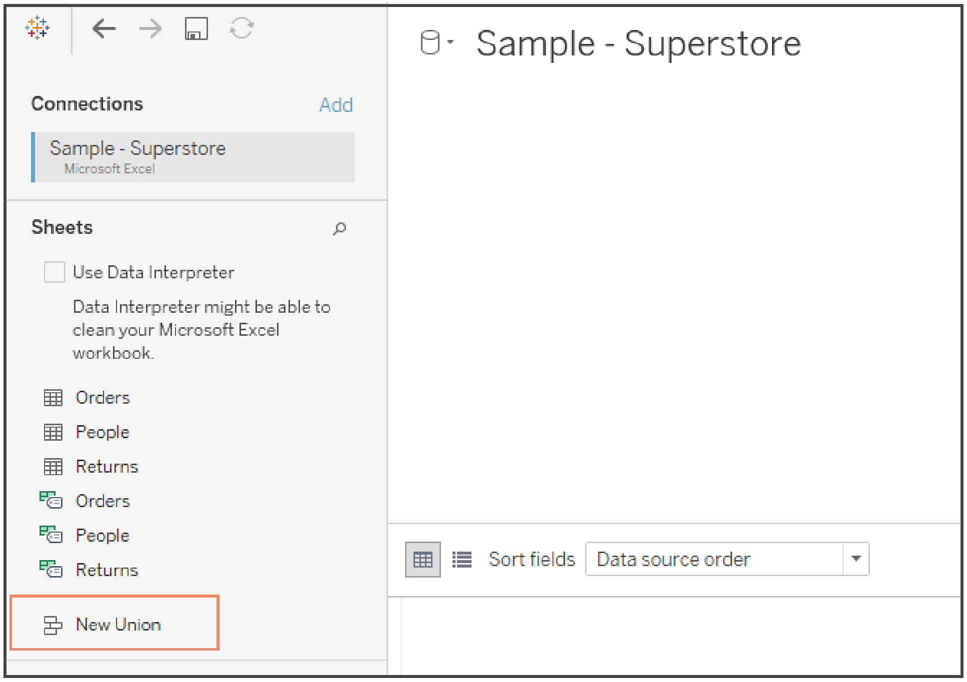
Note: The example sales data in both Store - North and Store - South are identical and therefore yield same results for both **Sales Total** and **Sales for Store - South** fields.

Union Your Data

You can union your data to combine two or more tables by appending values (rows) from one table to another. To union your data in Tableau data source, the tables must come from the same connection.

Supported connectors

If your data source supports union, the **New Union** option displays in the left pane of the data source page after you connect to your data. Supported connectors may vary between Tableau Desktop and Tableau Server and Tableau Online.



For best results, the tables that you combine using a union must have the same structure. That is, each table must have the same number of fields, and related fields must have matching field names and data types.

For example, suppose you have the following customer purchase information stored in three tables, separated by month. The table names are "May2016," "June2016," and "July2016."

May2016

Day	Customer	Purchases	Type
4	Lane	5	Credit
10	Chris	6	Credit
2-	Juan	1	Cr-

June2016

Day	Customer	Purchases	Type
1	Lisa	3	Credit
28	Isaac	4	Cash
28	Sam	2	Credi-

July2016

Day	Customer	Purchases	Type
2	Mario	2	Credit
15	Wei	1	Cash
21	Jim	7	Cash

D-a-y	Cus-tomer	Purch-ases	Ty-pe
8			edi-t

D-a-y	Cus-tomer	Purch-ases	Ty-pe
			t

A union of these tables creates the following single table that contains all rows from all tables.

Union

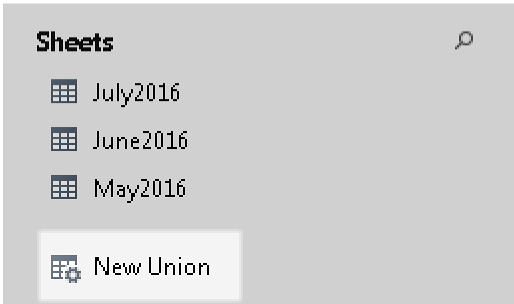
Day	Customer	Purchases	Type
4	Lane	5	Credit
10	Chris	6	Credit
28	Juan	1	Credit
1	Lisa	3	Credit
28	Isaac	4	Cash
28	Sam	2	Credit
2	Mario	2	Credit
15	Wei	1	Cash
21	Jim	7	Cash

Union tables manually

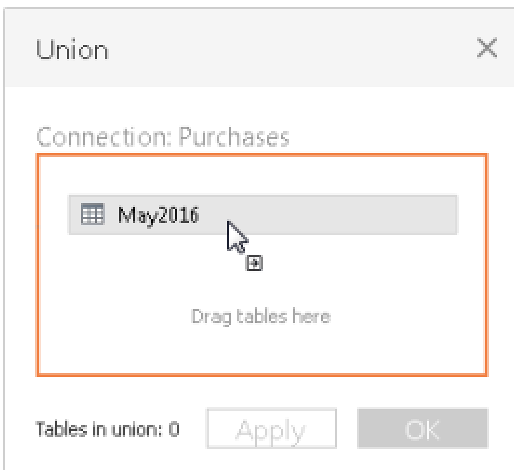
Use this method to manually union distinct tables. This method allows you to drag individual tables from the left pane of the Data Source page and into the Union dialog box.

To union tables manually

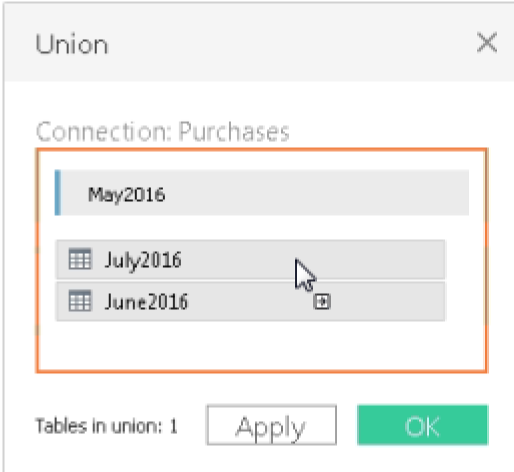
1. On the data source page, double-click **New Union** to set up the union.



2. Drag a table from the left pane to the Union dialog box.



3. Select another table from the left pane and drag it directly below the first table.



> **Tip:** To add multiple tables to a union at the same time, press **Shift** or **Ctrl** (**Shift** or **Command** on a Mac), select the tables you want to union in the left pane, and then drag them directly below the first table.

4. Click **Apply** or **OK** to union.

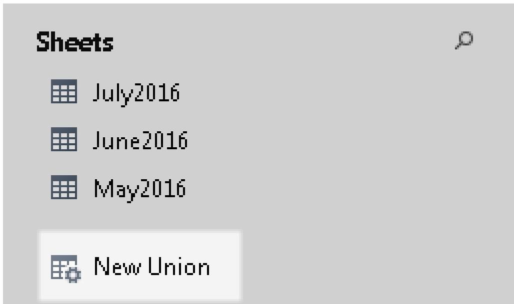
Union tables using wildcard search (Tableau Desktop)

Use this method to set up search criteria to automatically include tables in your union. Use the wildcard character, which is an asterisk (*), to match a sequence or pattern of characters in the Excel workbook and worksheet names, Google Sheets workbook and worksheet names, text file names, JSON file names, .pdf file names, and database table names.

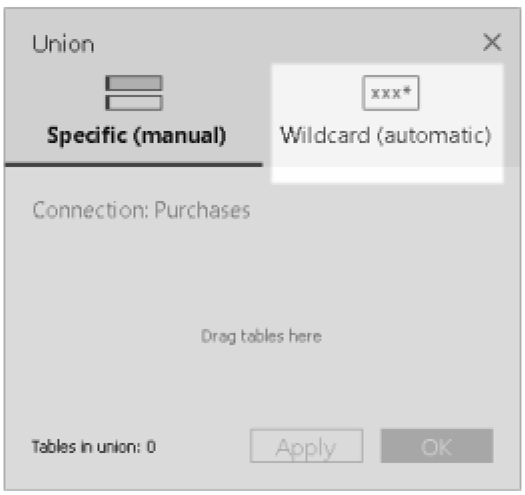
When working with Excel, text file data, JSON file, .pdf file data, you can also use this method to union files across folders, and worksheets across workbooks. Search is scoped to the selected connection. The connection and the tables available in a connection are shown on the left pane of the Data source page.

To union tables using wildcard search

1. On the data source page, double-click **New Union** to set up the union.



2. Click **Wildcard (automatic)** in the Union dialog box.



3. Enter the search criteria that you want Tableau to use to find tables to include in the union.

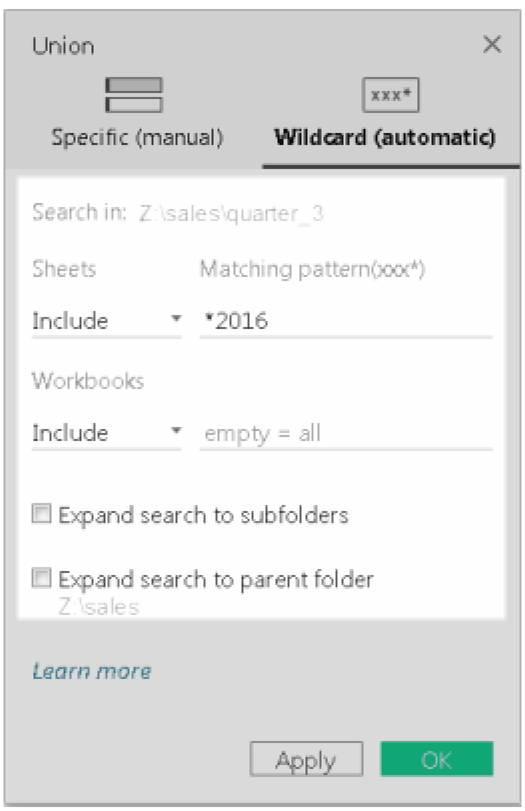
For example, you can enter ***2016** in the **Include** text box to union tables in Excel worksheets that end with "2016" in their names. Search criteria like this will result in the union of May2016, June2016, and July2016 tables (Excel worksheets), from the selected connection. In this case, the connection is called Sales, and the connection made to the Excel workbook containing the worksheets you wanted was in the quarter_3 folder in the sales directory (e.g., Z:\sales\quarter_3).

4. Click **Apply** or **OK** to union.

Expand search to find more Excel, text, JSON, .pdf data

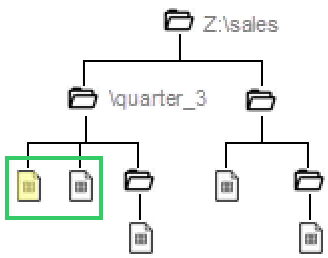
The tables initially available to union are scoped to the connection you've selected. If you want to union more tables that are located outside of the current folder (for Excel, text, JSON, .pdf files) or in a different workbook (for Excel worksheets), select one or both check boxes in the Union dialog box to expand your search.

For example, suppose you want to union all Excel worksheets that end with "2016" in its name outside of the current folder. The initial connection is made to an Excel workbook located in the same directory in the above example, Z:\sales\quarter_3.



- **Include:** If you enter ***2016** in the **Include** text box and leave the remaining search criteria of the dialog as is, Tableau looks for all Excel worksheets that end with "2016" in its name inside the current folder.

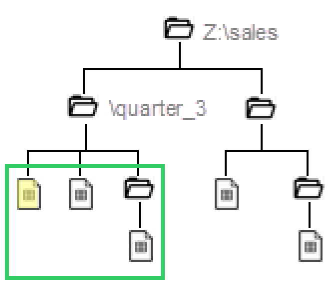
  In the diagram below, the yellow highlighted item represents the current location, that is, the Excel workbook that you created a connection to in the "quarter_3". The green box represents the tables belonging to workbooks and sheets that are unioned as result of this search criteria.

- **Include + Expand search to subfolders:** If you enter ***2016** in the **Include** text box and select the **Expand search to subfolders** check box, Tableau does the following:

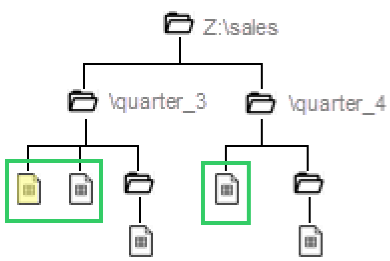
  - Looks for all Excel worksheets that end with "2016" in their names inside the current folder.
  - Looks for additional Excel worksheets that end with "2016" in their names that are located in Excel workbooks in subfolders of the "quarter_3" folder.

  In the diagram below, the yellow highlighted item represents the current location, that is, the Excel workbook that you created a connection to in the "quarter_3" folder. The green box represents the tables belonging to workbooks and worksheets that are unioned as a result of this search criteria.

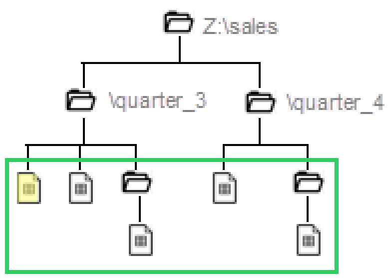
- **Include + Expand search to parent folder:** If you enter ***2016** in the **Include** text box and select the **Expand search to parent folder** check box, Tableau does the following:
  - Looks for all Excel worksheets that end with "2016" in their names inside the current folder, "quarter_3."
  - Looks for additional Excel worksheets that end with "2016" in their names that are located in parallel folders of the "quarter_3" folder. In this example, "quarter_4" is the parallel folder.

  In the diagram below, the yellow highlighted item represents current location, that is, the Excel workbook that you created a connection to in the "quarter_3" folder. The green boxes represent the tables belonging to the workbook and worksheets that are unioned as a result of this search criteria.

- **Include + Expand search to subfolders + Expand search to parent folder:** If you enter ***2016** in the **Include** text box and select both the **Expand search to subfolders** and **Expand search to parent folder** check boxes, Tableau does the following:
  - Looks for all Excel worksheets that end with "2016" in their namesinside the current folder, "quarter_3."
  - Looks for additional Excel workbooks that are located in the subfolders of the current folder, "quarter_3."
  - Looks for additional Excel workbooks that are located in parallel folders and subfolders of the "quarter_3" folder. In this example, "quarter_4" is the parallel folder.

  In the diagram below, the yellow highlighted item represents the current location, that is, the Excel workbook that you created a connection to. The green box represents the tables belonging to the workbook and worksheets that are unioned as a result of this search criteria.



Note: When working with Excel data, wildcard search includes named ranges but excludes tables found by Data Interpreter.

Rename, modify, or remove unions

Perform basic union tasks directly in the canvas of the Data Source page.

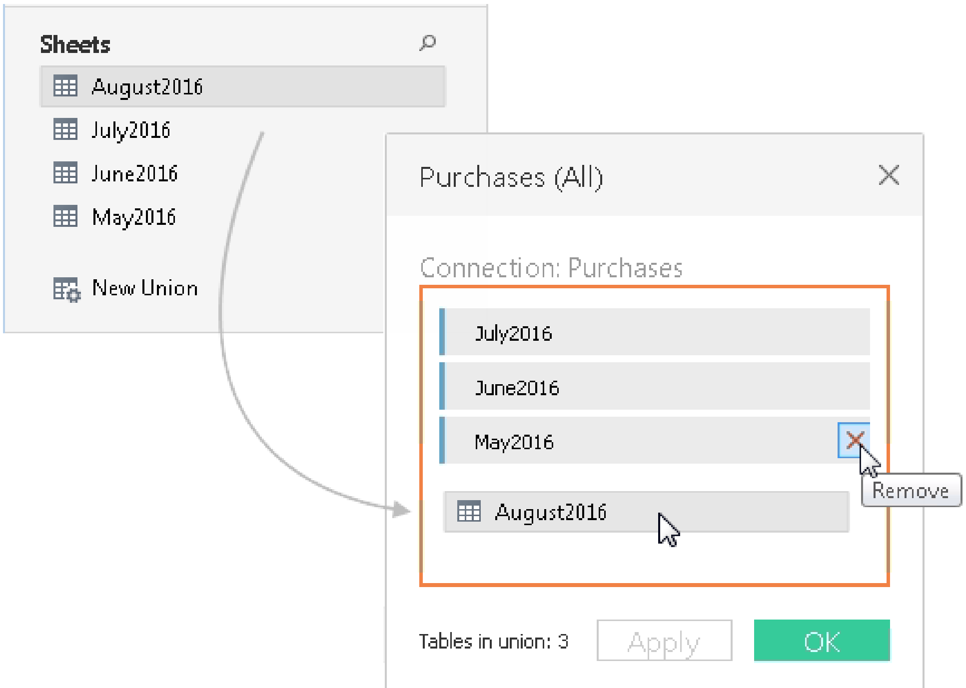
To rename a union

1. Double-click the logical table that contains unioned physical tables.
2. Double-click the union table on the physical layer canvas.
3. Enter a new name for the union.

To add or remove tables in the union

1. Double-click the logical table that contains unioned physical tables.
2. Click the union drop-down arrow and then select **Edit Union**.
3. You can drag additional tables that you want to union from the left pane, or hover over a

table until the remove icon displays and then click the icon to remove the table.



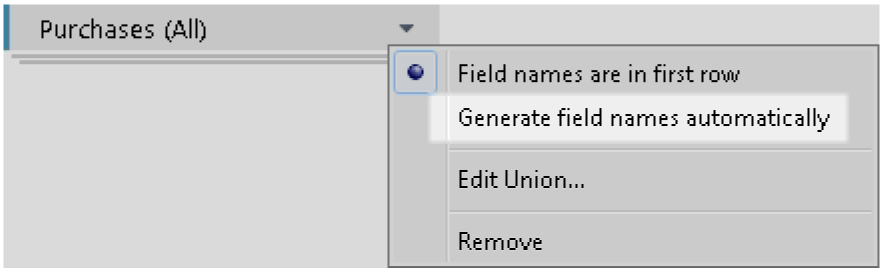
4. Click **Apply** or **OK** to complete the task.

To remove a union

- Double-click the logical table that contains unioned physical tables, and then click the union drop-down arrow and select **Remove**.

Matching field names or field ordering

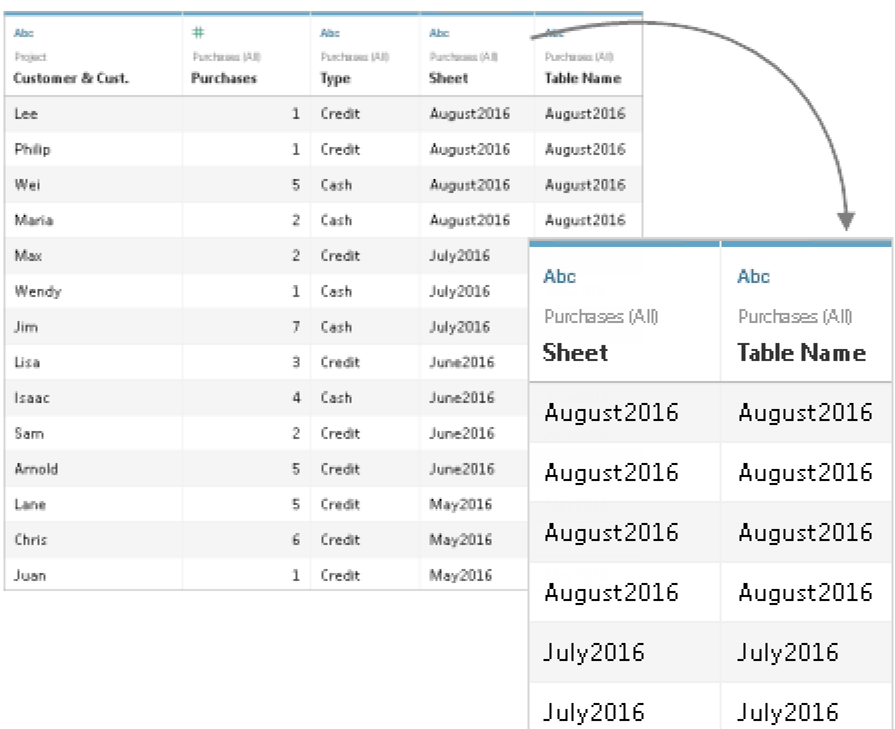
Tables in a union are combined by matching field names. When working with Excel, Google Sheets, text file, JSON file or .pdf file data, if there are no matching field names (or your tables do not contain column headers), you can tell Tableau to combine tables based on the order of the fields in the underlying data by creating the union and then selecting **Generate field names automatically** option from the union drop-down menu.



Metadata about unions

After you create a union, additional fields about the union are generated and added to the grid. The new fields provide information about where the original values in the union come from, including the sheet and table names. These fields are useful when unique information that is critical to your analysis is embedded in the sheet or table name.

For example, the tables used in the example above have unique month and year information stored in the table name instead of in the data itself. In this case, you can use the **Table Name** field that is generated by the union to access this information and use it in your analysis.



If a named range is used in a union, null values display under the **Sheet** field.

> **Note:** You can use the fields generated by a union, such as **Sheet**or **Table Name**, as join keys. You can use a unioned table in a join with another table or unioned table.

Merge mismatched fields in the union

When field names in the union do not match, fields in the union contain null values. You can merge the non-matching fields into a single field using the merge option to remove the null values. When you use the merge option, the original fields are replaced by a new field that displays the first non-null value for each row in the non-matching fields.

You can also create your own calculation or, if possible, modify the underlying data to combine the non-matching fields.

For example, suppose a fourth table, "August2016", is added to the underlying data. Instead of the standard "Customer" field name, it contains an abbreviated version called "Cust."

August2016

Day	Cust.	Purchases	Type
7	Maria	2	Credit
9	Kathy	1	Credit
18	Vijay	7	Cash

A union of these tables creates a single table that contains all rows from tables, with several null values. You can use the merge option to combine the related customer fields into a single field.

Union (with null values)

Union (with columns that have been merged)

Day	Customer	Purchases	Type	Cust.
4	Lane	5	Credit	null
10	Chris	6	Credit	null
28	Juan	1	Credit	null
1	Lisa	3	Credit	null
28	Isaac	4	Cash	null
28	Sam	2	Credit	null
2	Mario	2	Credit	null
15	Wei	1	Cash	null
21	Jim	7	Cash	null
7	null	2	Credit	Maria
9	null	1	Credit	Kathy
18	null	7	Cash	Vijay

Day	Purchases	Type	Customer, Cust.
4	5	Credit	Lane
10	6	Credit	Chris
28	1	Credit	Juan
1	3	Credit	Lisa
28	4	Cash	Isaac
28	2	Credit	Sam
2	2	Credit	Mario
15	1	Cash	Wei
21	7	Cash	Jim
7	2	Credit	Maria
9	1	Credit	Kathy
18	7	Cash	Vijay

After you merge fields, you can use the field generated from the merge in a pivot or split, or use the field as a join key. You can also change the data type of the field generated from a merge.

To merge mismatched fields

1. Select two or more columns in the grid.
2. Click the column drop-down arrow, and then select **Merge mismatched fields**.

To remove a merge

- Click the column drop-down arrow of the merged field and select **Remove merge**.

At a glance: Working with unions

Tableau Desktop and web authoring (Tableau Online and Tableau Server)

- A unioned table can be used in a join.
- A unioned table can be used in a join with another unioned table.
- The fields generated by a union, **Sheet** and **Table name**, can be used as the join key.
- If a named range is used in union, null values display under the **Sheet** field.
- The field generated from a merge can be used in a pivot.
- The field generated from a merge can be used as a join key.
- The data type of the field generated from a merge can be changed.
- Union tables from within the same connection. That is, you cannot union tables from different databases.

Tableau Desktop only

- When working with Excel data, wildcard search includes named ranges but excludes tables found by Data Interpreter.
- The field generated from a merge can be used in a pivot or split.
- To union a JSON file, it must have a .json, .txt, or .log extension. For more information about working with JSON data, see **JSON File** on page 409.

- When using wildcard search to union tables in a .pdf file, the result of the union is scoped to the pages that were scanned in the initial .pdf file you connected to. For more information about working with data in .pdf files, see **PDF File** on page 416.
- Stored procedures cannot be unioned.
- When working with database data, you can convert your union into custom SQL.

Create a Data Source or Add a New Connection with Clipboard Data

Sometimes you want to pull in data from an outside source for some quick analysis. Rather than create a whole data source and then connect in Tableau, you can copy and paste the data directly into your workbook. Tableau automatically creates a data source that you can begin analyzing.

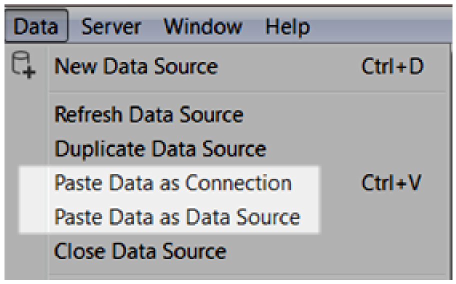
When you paste data on the data source page, Tableau creates a new connection in the existing data source.

When you paste data on the sheet, Tableau creates a new data source that you can begin analyzing. When you paste the data as a data source, the data source is saved as a text file to your Tableau Repository when you save the workbook.

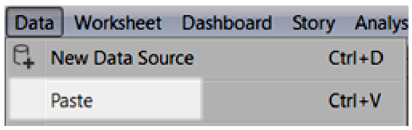
You can copy and paste data from a variety of office applications including Microsoft Excel and Microsoft Word. You can also copy and paste HTML tables from web pages. Tables that are copied as comma separated values or tab delimited can be pasted into Tableau.

Note: Not all applications use these formats when copying.

1. Select the data you want and copy it to the clipboard.
2. Open Tableau Desktop and do one of the following:
    1. On the data source page, select **Data** > **Paste Data as Connection** or **Paste Data as Data Source**.

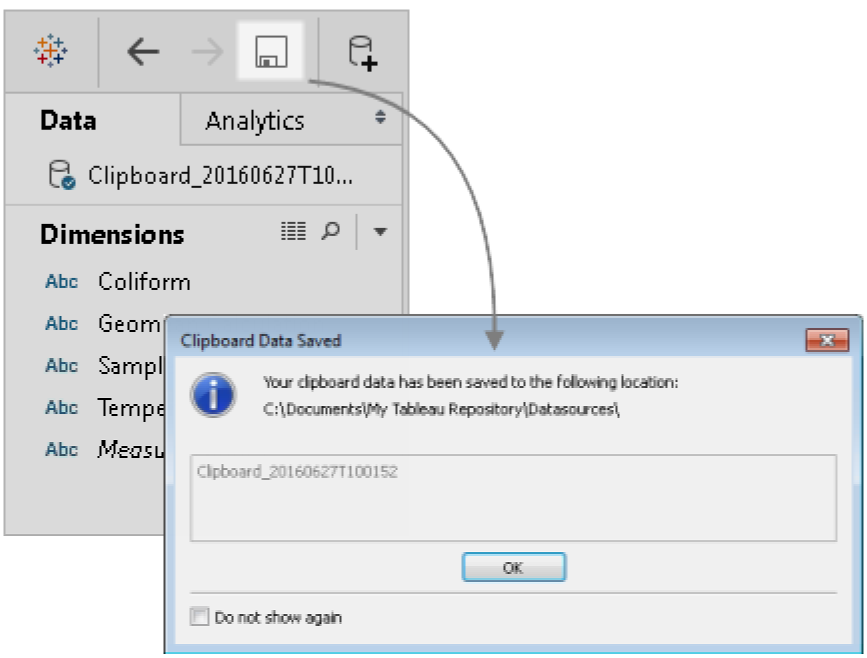


2. On the sheet, select **Data** > **Paste** to paste the data as a data source.



3. Select **File** > **Save** to save the data source.

   When you save the workbook, the data source either becomes a part of the existing data source or is added to your repository, depending on which of the methods you choose. If you paste the data as a data source, the data source is saved with the workbook when you save the workbook as a packaged workbook (.twbx).



Connect to a Custom SQL Query

For most databases, you can connect to a specific query rather than the entire data set. Because databases have slightly different SQL syntax from each other, the custom SQL you use to connect to one database might be different from the custom SQL you might use to connect to another. However, using custom SQL can be useful when you know exactly the information you need and understand how to write SQL queries.

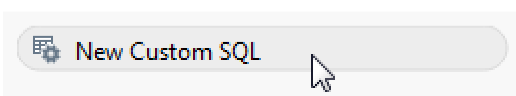
Though there are several common reasons why you might use custom SQL, you can use custom SQL to union your data across tables, recast fields to perform cross-database joins, restructure or reduce the size of your data for analysis, etc.

For Excel and text file data sources, this option is available only in workbooks that were created before Tableau Desktop 8.2 or when using Tableau Desktop on Windows with the legacy connection. To connect to Excel or text files using the legacy connection, connect to the file, and in the Open dialog box, click the **Open** drop-down menu, and then select **Open with Legacy Connection**.

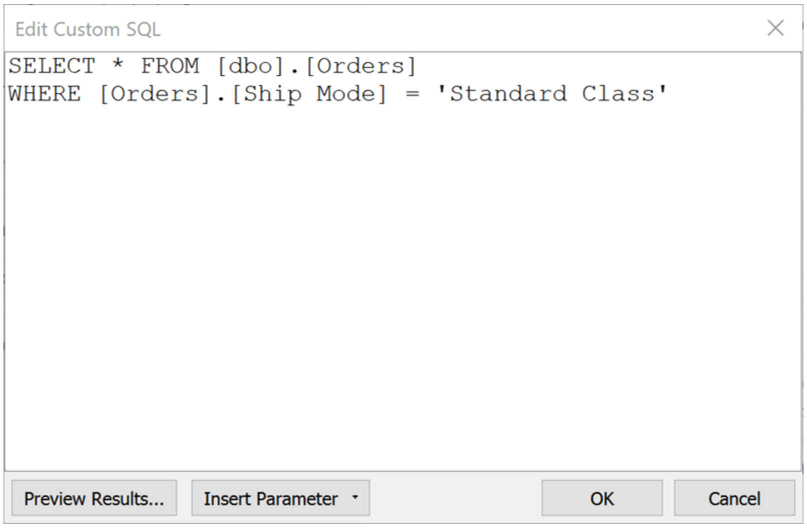
> **NOTE:** Beginning with Tableau 2020.2, legacy Excel and Text connections are no longer supported. See the Legacy Connection Alternatives document in Tableau Community for alternatives to using the legacy connection.

Connect to a custom SQL query

1. After connecting to your data, double-click the **New Custom SQL** option on the Data Source page.

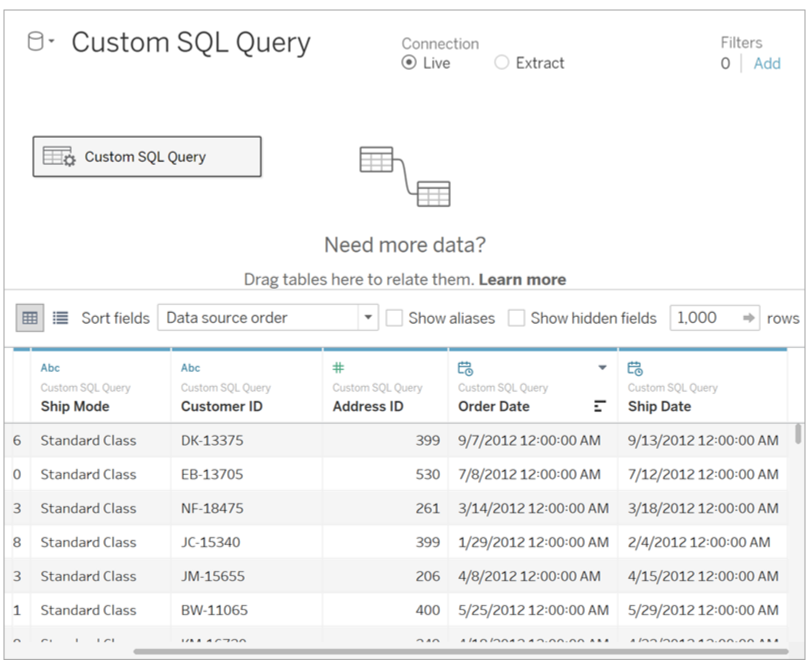


2. Type or paste the query into the text box.



3. When finished, click **OK**.

When you click OK, the query runs and the custom SQL query table appears in the logical layer of the canvas. Only relevant fields from the custom SQL query display in the data grid on the Data Source page.



For more information about the logical and physical layers of the canvas, see **The Tableau Data Model** on page 810.

Examples of custom SQL queries

Combine your tables vertically (union)

If you need to append data to each other, you can use the union option in the physical layer of the canvas in Tableau. In some cases your database does not support this option, so you can use custom SQL instead.

For example, suppose you have the following two tables: November and December.

November

Guest ID	Party Size
3005	4
3006	2
3007	2
3008	2

December

Guest ID	Party Size
3009	2
3010	2
3011	4
3012	5
3013	3
3014	4
3015	2

You can use the following custom SQL query to append the second table, December, to the first table, November:

```
SELECT * FROM November UNION ALL SELECT * FROM December
```

The result of the query looks like this in the data grid:

Abc Custom SQL Query Guest ID	# Custom SQL Query Party Size
3005	4.00000
3006	2.00000
3007	2.00000
3008	2.00000
3009	2.00000
3010	2.00000
3011	4.00000
3012	5.00000
3013	3.00000
3014	4.00000
3015	2.00000

For more information about the union option, see **Union Your Data** on page 942.

Change the data type of a field to do a cross-database join

When you want to perform a join between two tables in the physical layer of the canvas, the data type of the fields you join on must be the same. In cases when the data type of the fields are not the same, you can use custom SQL to change the data type (cast) the field before performing the join.

For example, suppose you want to join two tables, Main and Sub, using the Root and ID fields, respectively. The Root field is a number type and the ID field is a string type. You can use the following custom SQL query to change the data type of Root from a number to a string so that you can join the Main and Sub tables using the Root and ID fields.

```
SELECT [Main].[Root] AS [Root_Number]
CAST([Main].[Root] AS INT] AS [Root_String]
FROM [Main]
```

The result of this query shows the original Root field and the Root field cast as a string.

# Custom SQL Query Root_Number	Abc Custom SQL Query Root_String
7,981.00	7981
740.00	740
741.00	741
742.00	742
1,760.00	1760

For more information about joins and cross-database joins, see **Join Your Data** on page 883.

Reduce the size of your data

When working with very large data sets, sometimes you can save time while working with your data if you reduce its size first.

For example, suppose you have a large table called FischerIris. You can use the following custom SQL query to retrieve the specified columns and records thereby reducing the size of the data set that you connect to from Tableau.

```
SELECT
 [FischerIris].[Species] AS [Species],
 [FischerIris].[Width] AS [Petal Width],
 COUNT([FischerIris].[ID]) AS [Num of Species]
FROM [FischerIris]
WHERE [FischerIris].[Organ] = 'Petal'
AND [FischerIris].[Width] > 15.0000
GROUP BY [FischerIris].[Species], [FischerIris].[Width]
```

Restructure your data (pivot)

In some cases, you might be working with a table that needs to be restructured before analysis. Though this type of task can be done in the physical layer of the canvas in Tableau by using options like pivot, your database might not support it. In this case, you can use custom SQL instead.

For example, suppose you have the following table:

Season ID	Items - Don't like	Items - Defective	Items - Too small	Items - Too big
R000151493		1	3	20
R000151493		2	44	1
R000151495		0	0	0
R000151495		0	17	1
R000151497		1	0	0
R000151497		0	15	0
R000151789		0	0	0
R000151789		0	12	4
R000151813	3	0	0	3
R000151813	3	0	53	1
R000151815	0	0	0	0
R000151815	0	0	0	0
R000151855	0	0	0	3
R000151855	59	0	22	0
R000151857	4	0	14	18
R000151857	0	0	0	33
R000153013	0	0	21	0
R000153013	9	1	19	0

To change its structure and optimize your data for analysis in Tableau, you can use the following custom SQL query:

```
SELECT Table1.Season ID AS [Season ID],
Table1.Items - Don't like AS [Quantity],
"Don't Like" AS [Reason]
FROM Table1
UNION ALL
SELECT Table1.Season ID AS [Season ID],
Table.Items - Defective AS [Quantity],
"Defective" AS [Reason]
FROM Table1
UNION ALL
SELECT Table1.Season ID AS [Season ID],
Table1.Items - Too big AS [Quantity],
"Too Big" AS [Reason]
FROM Table1
UNION ALL
SELECT Table1.Season ID AS Season ID,
Table1.Items - Too small AS [Quantity]
"Too Small" AS [Reason]
FROM Table1
```

The result of the query looks like this in the data grid:

Abc Custom SQL Query **Season ID**	Abc Custom SQL Query **Quantity**	Abc Custom SQL Query **Reason**
R000151493	*null*	Don't Like
R000151493	*null*	Don't Like
R000151495	*null*	Don't Like
R000151495	*null*	Don't Like
R000151497	*null*	Don't Like
R000151497	*null*	Don't Like
R000151789	*null*	Don't Like
R000151789	*null*	Don't Like
R000151813	3	Don't Like
R000151813	3	Don't Like
R000151815	0	Don't Like
R000151815	0	Don't Like
R000151855	0	Don't Like
R000151855	59	Don't Like

For more information about the pivot option, see **Pivot Data from Columns to Rows** on page 988.

Combine (join) and aggregate your data

If you need to combine tables and aggregate your data, you can use both a join and default aggregation type options in the physical layer of the canvas in Tableau. In some cases you might need to use custom SQL instead.

For example, suppose you have the following two tables: Orders and Vendors.

Orders

Order	CustomerID	VendorID
10248	32	1
10249	90	3
10250	81	2

Vendors

VendorID	Name
1	Tminus Shipping
2	Packing You
3	ShipWise Express

You can use the following custom SQL query to find a count on the number of orders and do a left join on the Orders and Vendors tables:

```
SELECT Vendors.Name,COUNT(Orders.Order) AS Number Of Orders
FROM Orders
LEFT JOIN Vendors
ON Orders.VendorID=Vendors.VendorID
GROUP BY Name;
```

The result of the query looks like this:

Abc Custom SQL Query Name	# Custom SQL Query Number Of Orders
Packing You	1
ShipWise Express	1
Tminus Shipping	1

For more information about joins, see **Join Your Data** on page 883.

Errors when duplicate columns are referenced

If your custom SQL query references duplicate columns, you may get errors when trying to use one of the columns in your analysis in Tableau. This will happen even if the query is valid. For example, consider the following query:

```
SELECT * FROM authors, titleauthor WHERE authors.au_id =
titleauthor.au_id
```

The query is valid, but the **au_id** field is ambiguous because in this case it exists in both the “authors” table and the “titleauthor” table. Tableau will connect to the query but you will get an error anytime you try to use the **au_id** field. This is because Tableau doesn't know which table you are referring to.

> **Note:** It is a best practice to define column aliases with an AS clause whenever possible in a Custom SQL Query. This is because each database has its own rules when it comes to automatically generating a column name whenever an alias is not used.

Edit a custom SQL query

To edit a custom SQL query

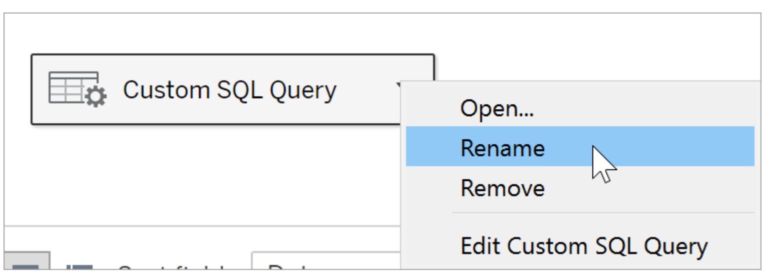
1. On the data source page, in the canvas, double-click the custom SQL query in the logical layer.
2. Hover over the custom SQL table in the physical layer until the arrow displays.

3. Click the arrow and then select **Edit Custom SQL Query**.
4. In the dialog box, edit the custom SQL query.

To change a custom SQL query name

When you drag a custom SQL query to the logical layer of the canvas, Tableau gives it a default name: Custom SQL Query, Custom SQL Query1, and so on. You can change the default name to something more meaningful.

1. On the data source page, in the logical layer of the canvas, select the drop-down arrow in the custom SQL query table and select **Rename**.



2. Enter the name you want to use for your custom SQL query.

Use parameters in a custom SQL query

You can use parameters in a custom SQL query statement to replace a constant value with a dynamic value. You can then update the parameter in the workbook to modify the connection. For example, you may connect to a custom SQL query that provides web traffic data for a particular page that is specified by a pageID. Instead of using a constant value for the pageID value in the SQL query, you can insert a parameter. Then after finishing the connection, you can show a parameter control in the workbook. Use the parameter control to switch out the pageID and pull in data for each page of interest without having to edit or duplicate the connection.

In **Tableau Desktop**, you can create a parameter directly from the Custom SQL dialog box or use any parameters that are part of the workbook. If you create a new parameter, it becomes available for use in the workbook just like any other parameter. See **Create Parameters** on page 1218 to learn more.

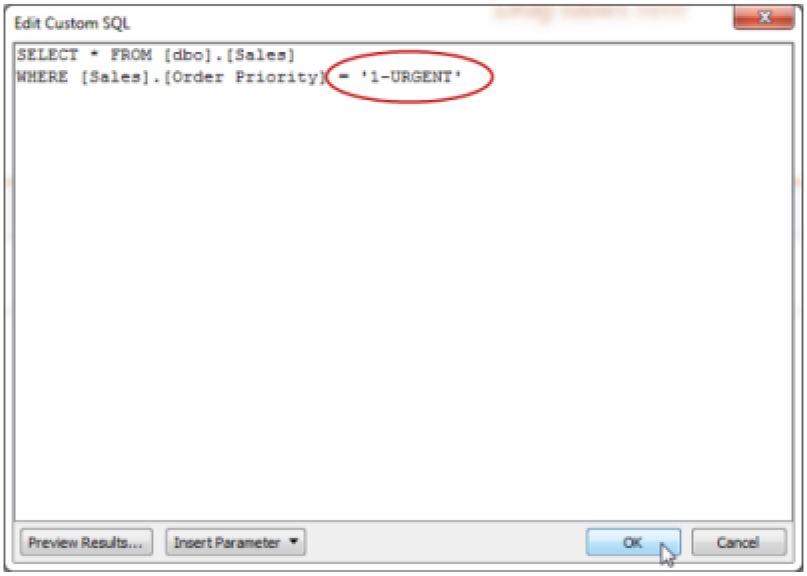
For **web authoring (in Tableau Online or Tableau Server)**, you can use an existing parameter published from Tableau Desktop. You cannot create a new parameter in web authoring.

To add a parameter to a custom SQL query

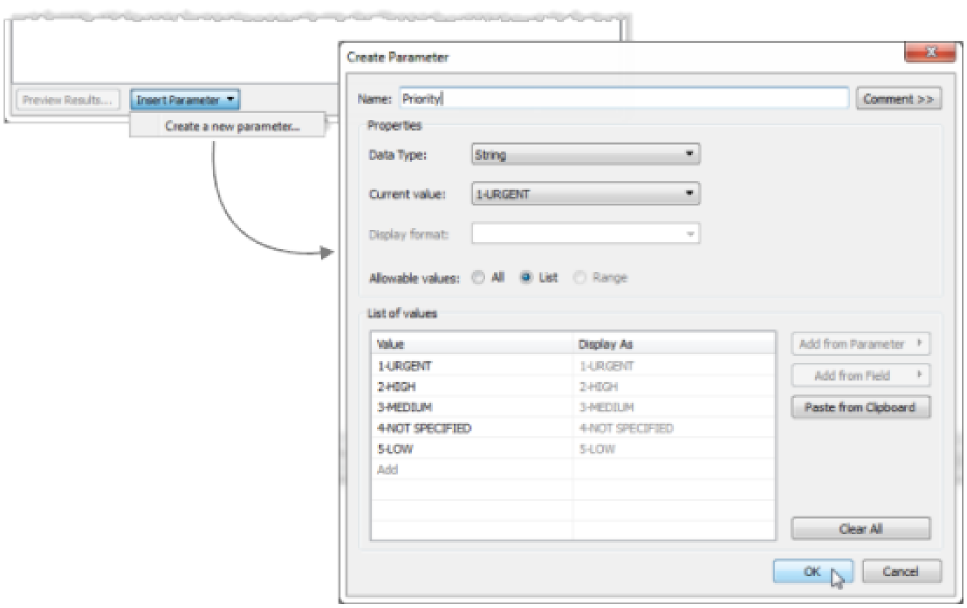
1. On the data source page, in the canvas, hover over the table until the edit icon displays, and then click the edit button.
2. At the bottom of the dialog box, click **Insert Parameter**.
3. Select a constant value in the SQL statement and then, from the **Insert Parameter** drop-down menu select the parameter you want to use instead. If you have not created a parameter yet, select **Create a new parameter**. Follow the instructions in **Create Parameters** on page 1218 to create a parameter.

> **Note:** Parameters can only replace literal values. They cannot replace expressions or identifiers such as table names.

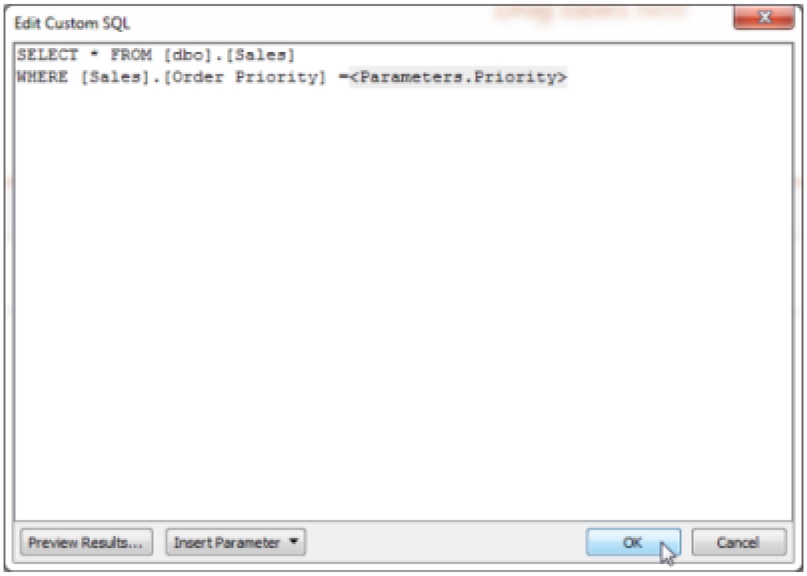
In the example below, the custom SQL query returns all orders that are marked as Urgent priority. In the custom SQL statement, the order priority is the constant value. If you want to change the connection to see the High priority orders, you would have to edit the data source.



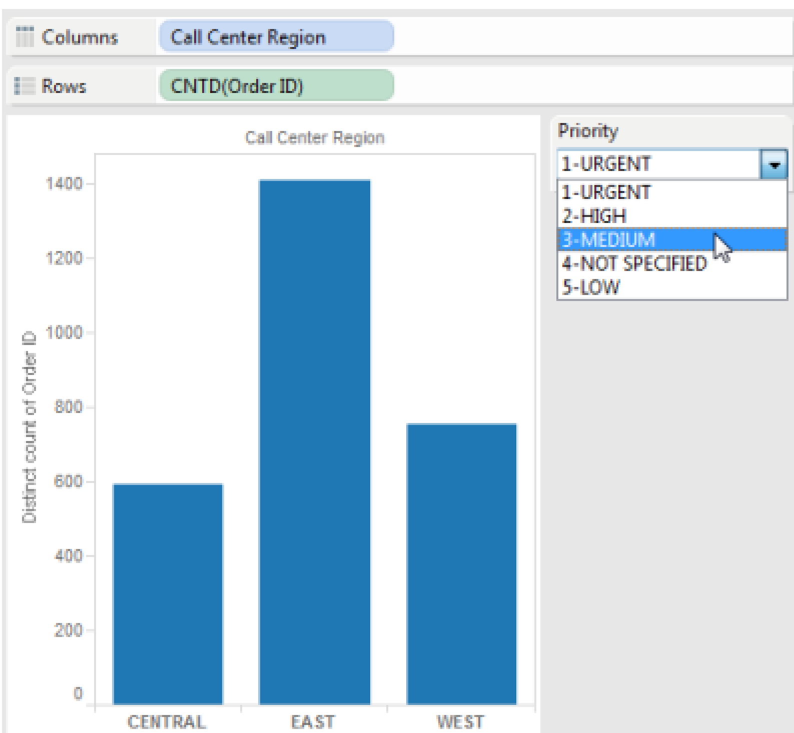
Instead of creating and maintaining many variations of the same query, you can replace the constant order priority value with a parameter. The parameter should contain all of the possible values for Order Priority.



After you create a parameter, you can insert it into the SQL statement to replace the constant value.



After you finish editing the connection, the new parameter is listed in the Parameters area at the bottom of the Data pane and the parameter control displays on the right side of the view. As you select different values, the connection updates.



Note: If you are using an extract, you must refresh the extract in order to reflect changes to the parameter. Publishing a data source that uses Custom SQL parameters includes the parameters. The parameters are transferred to any workbooks that connect to the data source.

Tableau Catalog support for custom SQL

Starting in 2019.3, Tableau Catalog is available as part of the Data Management Add-on to Tableau Server and Tableau Online. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Supported queries

Catalog currently supports custom SQL queries that meet the ANSI SQL-2003 standard, with three known exceptions:

- Time zone expressions
- Multiset expressions
- Tableau parameters

Supported features and functions

Catalog supports the following additional functionality for data sources, workbooks, and flows with connections that use the Microsoft SQL Server, MySQL, or PostgreSQL drivers, for example, Amazon Aurora for MySQL, Amazon RedShift, Pivotal Greenplum Database, MemSQL, Denodo, and others.

- SQL Server temp tables
- SQL Server table variables
- MySQL GROUP_CONCAT function
- PostgreSQL arrays
- PostgreSQL EXTRACT() function

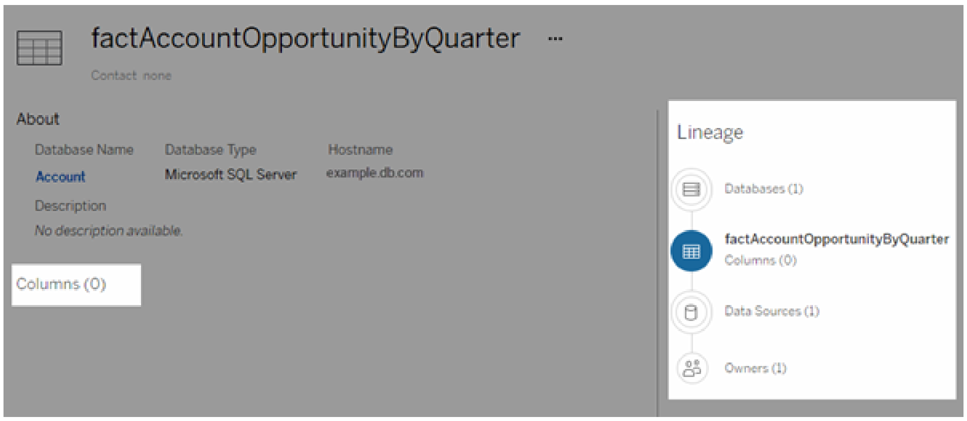
Other custom SQL scenarios and functionality might work, but Tableau doesn't specifically test for or support them.

Supported lineage

To see the lineage of a connection that uses custom SQL, from the External Assets page, navigate to and select a table from the list. A page opens and shows information about the table, including name, type, and description.

To the right of the table information is the Lineage pane, which shows the table's relationship to upstream and downstream assets. Note that Catalog doesn't support showing column information in the lineage for table metadata gathered using custom SQL.

In the following screenshot, the factAccountOpportunityByQuarter table was indexed because it's used by a data source. However, because it's referenced by a custom SQL query, the column information isn't available.



In a case where more than one data source, workbook, or flow uses a table, any of the assets downstream from that table that use a custom SQL query are excluded when column-level filters are applied. As a result, fewer downstream assets show in the lineage than are actually used.

See Also

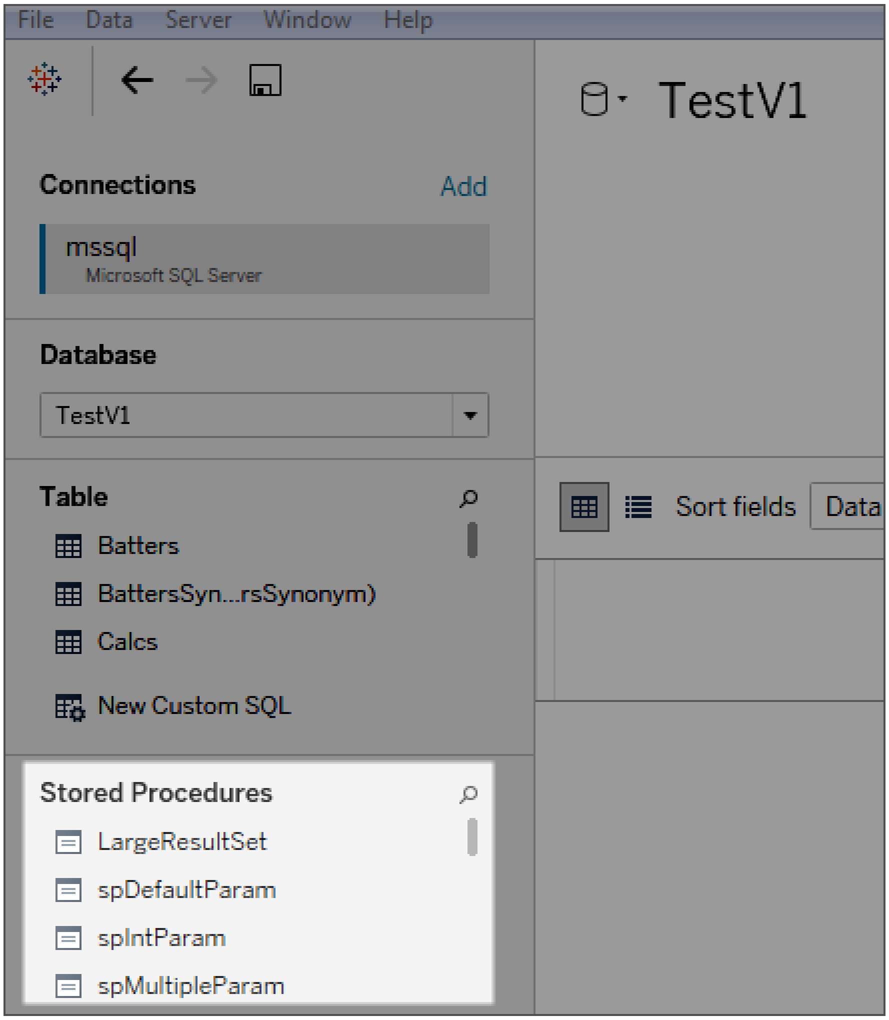
Use Custom SQL and RAWSQL to perform advanced spatial analysis on page 1801

Use a Stored Procedure

A stored procedure is a subroutine available to applications that access a relational database system. When you connect to a SAP Sybase ASE, Microsoft SQL Server, or Teradata database with Tableau, you can use a stored procedure to define the connection.

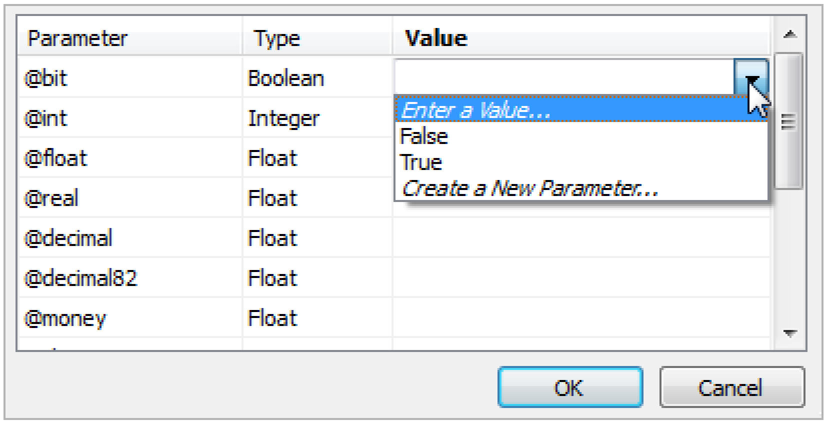
For Oracle, you can use a table function to define the connection. Oracle table functions act similarly to stored procedures and are listed under **Stored Procedures** on the Data Source page.

When you create a data source using one of these data types, available procedures are listed under **Stored Procedures**, as shown in the Microsoft SQL Server example:



From the left pane, drag the procedure to the canvas or double-click one of the listed procedures. If parameters are available in the procedure, the Parameters dialog box automatically displays.

Instead of entering a value, you can use an existing Tableau parameter, or create a new Tableau parameter for the value:



If you then expose the Tableau parameter in the view, users are able to change the value of the parameter in the procedure interactively.

Stored procedures don't support federation, relating, or joining. They are represented in a single logical table in the logical layer, and don't allow opening the Join/Union canvas (physical layer).

Notes on Stored Procedures

If you use stored procedures to define a data source for Tableau, keep the following in mind:

- If a stored procedure returns more than one result set, Tableau reads the first one and ignores the rest.
- If a stored procedure has output parameters, Tableau filters out the stored procedure.
- Stored procedures that have parameters of a non-scalar type are excluded.
- Result set columns that don't have matching types in Tableau (such as varbinary, geometry, and hierarchyid) are logged. If all result set columns map to unknown data types, Tableau displays a message:

  "The result set... has no usable columns."
- Stored procedures that return no result sets are listed on the data source page but fail if selected.
- If no value is provided for a parameter that the stored procedure requires, an error occurs. Tableau cannot determine in advance whether parameters are required.
- Tableau does not perform any transaction management for stored procedures. That is, stored procedure writers must not depend on Tableau to start transactions before invoking stored procedures, or to commit them afterward.

- Column names must be unique for stored procedures to work. If two columns have the same name, or if no name is provided, the procedure can result in an error.
- If there are multiple queries in a stored procedure (for example, to read values from another table or to hold temporary combinations) each of the queries must return the same sets of columns in the same order (same names and data types). To ensure that column order and names match in the query results, you may need to explicitly `CAST` to ensure the data type is correct, for example `CAST(Username as VARCHAR(20))`, and explicitly name the columns. If a stored procedure does not follow these guidelines, an error message can result:

  "InsertData: unbound column error"
- If there are multiple queries in a stored procedure (for example, to read values from another table or to hold temporary combinations) and the procedure is generating an error, try adding `SET NOCOUNT ON` to the top of the procedure.This prevents the message which shows the count of number of rows affected by a Transact-SQL statement from being returned as part of the result set for a query.

In addition, the following constraints apply for specific databases.

Stored Procedure Constraints for Teradata Databases

The following constraints apply for stored procedures on Teradata databases.

- Values must be provided for every parameter. If the user does not provide a value for one or more parameters, Tableau displays a Teradata database error stating there are too few values provided for the stored procedure.

Stored Procedure Constraints for SQL Server Databases

The following constraints apply for stored procedures on SQL Server databases.

- If the result set for a stored procedure contains columns of type IMAGE or TEXT, the stored procedure will fail with an "Incorrect syntax" error message.
- If the total width of the result set (number of bytes in each row) exceeds 8060, the stored procedure fails. This can occur with very wide tables (hundreds of columns) or with tables having large text columns, intended to hold thousands of characters of text.
- Tableau does not display stored procedures from schema "sys".
- If the user does not provide a value for one or more parameters that the procedure requires, Tableau displays a SQL Server database error in the form "The procedure

requires a value for parameter @x but one was not provided."

- Stored procedures that contain multiple queries should follow the guidelines listed in Notes on Stored Procedures (above).
- Tableau Desktop does not support the Microsoft SQL Server TIME data type. When fields of this type are included in a stored procedure on a Microsoft SQL Server database Tableau Desktop will not import them.

Stored Procedure Constraints for SAP Sybase ASE Databases

The following constraints apply for stored procedures on SAP Sybase ASE databases (Windows only).

- The database must have a properly configured remote server.
- If the user does not provide a value for one or more parameters that the procedure requires, Tableau displays a Sybase ASE database error in the form "The procedure requires a value for parameter @x but one was not provided."

Use Certified and Recommended Data Sources and Tables

To speed up the process of validating and setting up data, use certified and recommended data sources and tables. Here's how these options differ:

- *Certified data sources* are carefully chosen by site administrators and project leaders.
- *Recommended data sources* include personally certified ones and others automatically picked for you based on usage patterns at your organization.
- *Recommended tables* are database tables that are used frequently in data sources and workbooks published to your Tableau server.

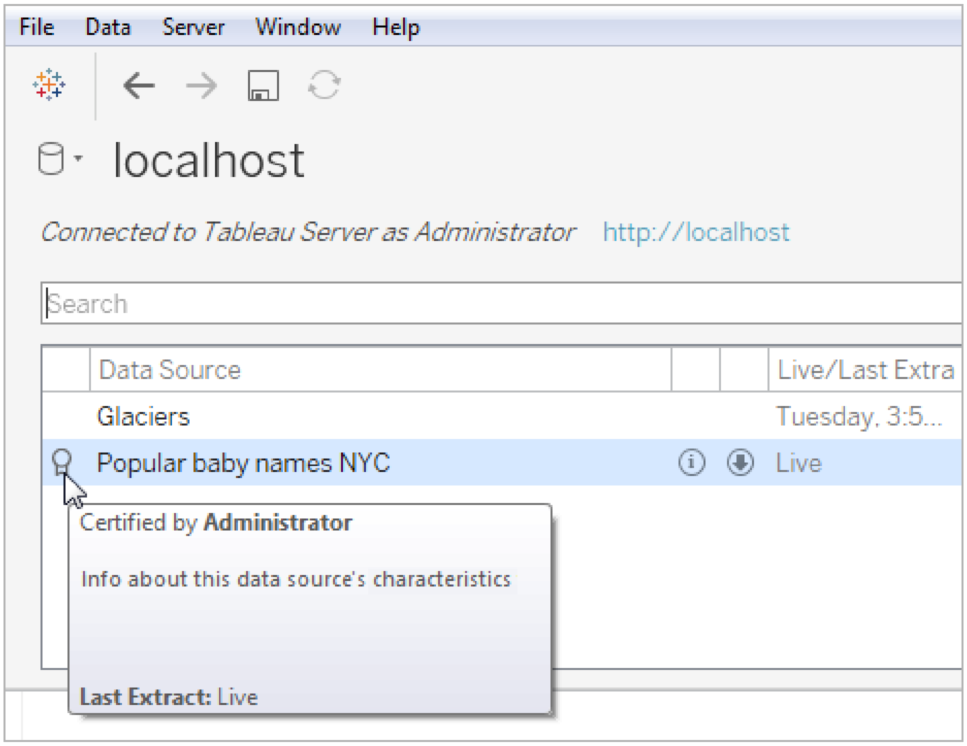
By taking advantage of relevant content that's already available, you can avoid duplicating work and spend more time on your analysis.

Note: To access these features in Tableau Desktop, you need to sign in to Tableau Server or Tableau Online from the **Server** menu.

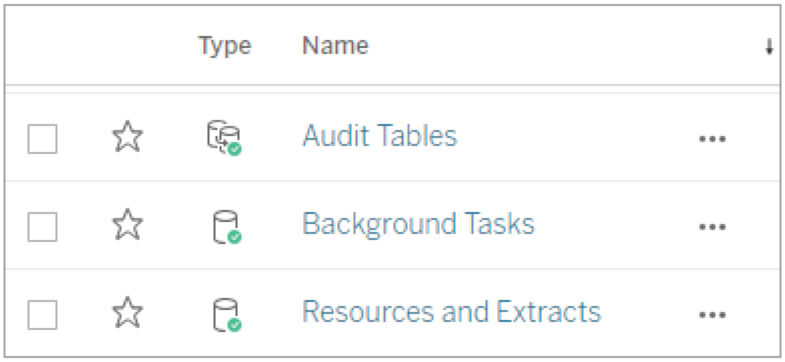
Certified data sources published to Tableau Server or Tableau Online

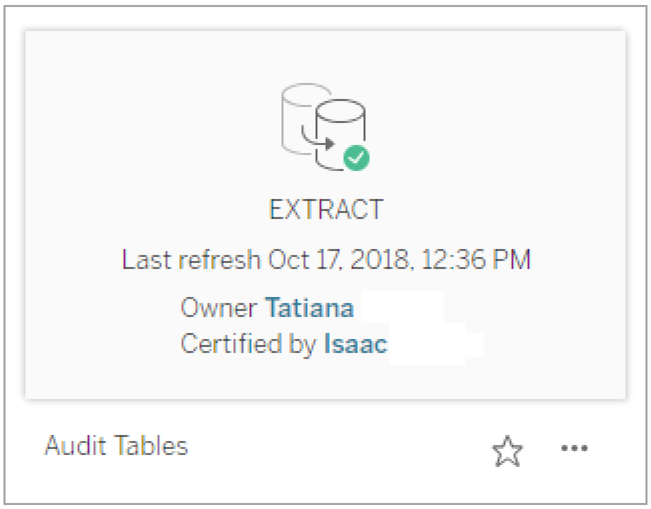
Certified data sources appear with a unique certification badge. Hover over the badge to learn who certified the data source, read any descriptive notes they've provided, and confirm the connection type.

Here's how certified data sources look when you connect to data in Tableau Desktop.



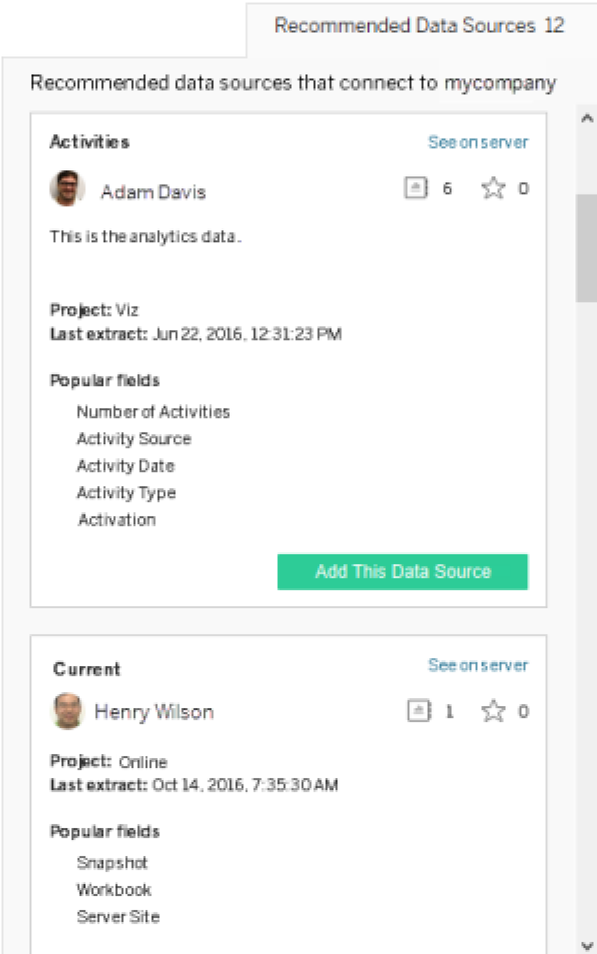
And here's how you can identify certified data sources in Tableau Server or Tableau Online.





Recommended data sources published to Tableau Server or Tableau Online

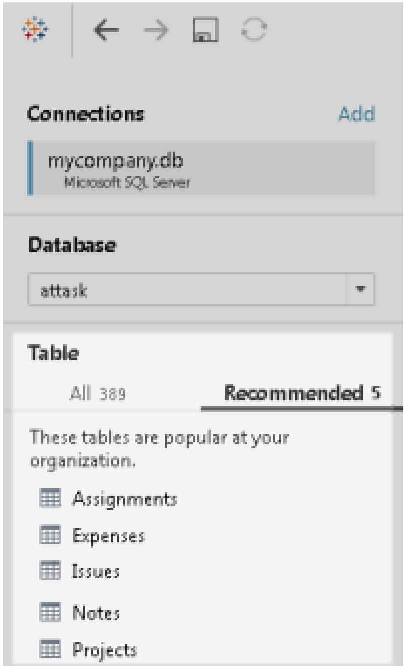
On the right side of the data source setup page, click the **Recommended Data Sources** button to quickly access popular data sources at your organization.



Note: Tableau Server administrators can turn off recommendations features.

Recommended tables

Recommended tables appear in the left pane of the data source page. After you add a recommended table to the canvas, the Recommended list shows tables frequently joined to the table you added. If you join a recommended table, Tableau uses the most popular fields and join type by default. (Any foreign keys specified in a database override recommended joins).



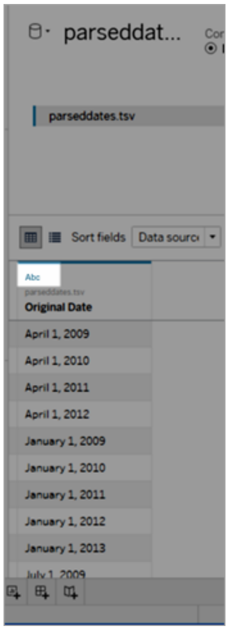
Recommended data sources and tables are supported for a subset of data connectors. If you don't see the Recommended list while signed into Tableau Online or Tableau Server from Tableau Desktop, the current data connector doesn't support recommendations.

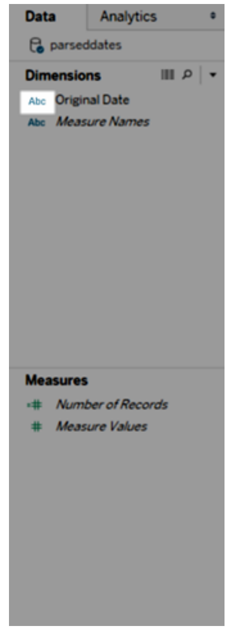
Convert a Field to a Date Field

Each database stores Date fields in slightly different ways. Tableau does its best in interpreting the date fields, but sometimes a field will be imported to Tableau as a text string or numerical field. If this happens, there are a few steps we can take to resolve the issue.

Verifying Date Fields

A date field might appear as a string in the Data Source page	Or as a string in the Data pane





When dates are interpreted as strings, you will lose all of the features and conveniences of working with date fields, such as drilling down, using date calculations, and switching between Continuous and Discrete measures.

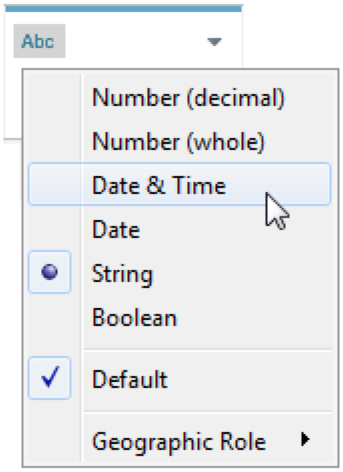
If your dates aren't interpreted correctly, take the following steps in order:

1. **Change the data type of the field** below
2. **Create a calculation using the DATEPARSE function** on the next page
3. **Create a calculation using the DATE function** on page 986

Change the data type of the field

The first step in resolving a date field interpretation issue is to make sure the data type is set to **Date** or **Date & Time**.

1. In the **Data Source** or **Data** pane, click the data type icon and change the data type to **Date** or **Date & Time**



2. Inspect the data in the view or **Data Source** pane. If you see many Null values, return the data type to **String** and proceed to **Create a calculation using the DATEPARSE function** below to correct the issue.

Create a calculation using the DATEPARSE function

Dates are stored in a nearly infinite array of formats. Some date fields have years before months, others separate the parts of the date with periods, and still others use a combination of formats. When Tableau cannot interpret a date field, it might be because the particular format cannot be translated.

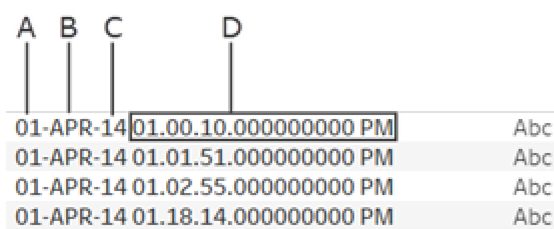
The DATEPARSE function lets you clearly define which parts of your field are which parts of a date. In essence, you are creating a map that Tableau can use to translate the string into a date field. This map is referred to as the **format**.

> **Note:** This function is available through the following connectors: non-legacy Excel and text file connections, Amazon EMR Hadoop Hive, Cloudera Hadoop, Google Sheets, Hortonworks Hadoop Hive, MapR Hadoop Hive, MySQL, Oracle, PostgreSQL, and Tableau extracts. Some formats may not be available for all connections.

> **Note**: DATEPARSE is not supported on Hive variants. Only Denodo, Drill, and Snowflake are supported.

Creating the DATEPARSE calculation

1. Inspect the current format of your date field. Note where months, years, hours, and days are displayed in your field. You will need this information for the DATEPARSE function.



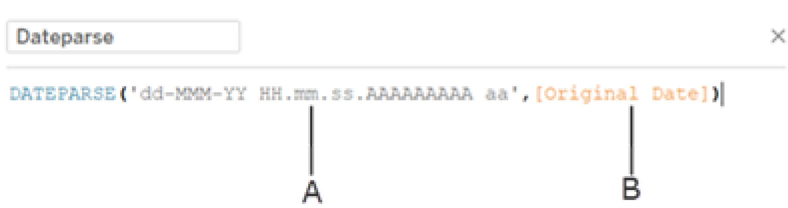
A. Day of month

B. Month

C. Year

D. Hour, minute, second, millisecond, period

2. Right click your date field in the Dimensions pane and select **Create** > **Calculated Field**.
3. In the dialog, write the DATEPARSE function. The DATEPARSE function has two parts: the **format** and the **string**. The **string** is field you wish to convert, which must be a string data type.



A. Format

B. String

The **format** is the guide Tableau will use to interpret the string as a date. Each part of a date or time string has a matching symbol, as seen in the table below. The format must exactly duplicate the way that the date is displayed. For example, if a single year code ("Y") is used when the string has a 2-digit year code ("97"), the calculation might return null values.

Note: The exception to the format exactly duplicating the displayed string is that long form parts of dates ("September") can be formatted with four symbols ("September" = "MMMM").

The format must include all spaces, hyphens, and other non-alphanumeric symbols in order to interpret the string correctly.

Date Field Symbols

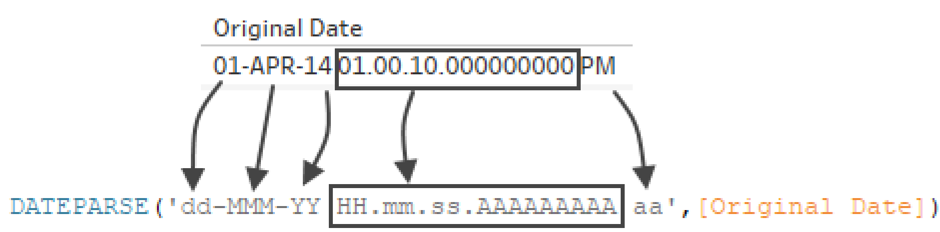
Date Part	Symbol	Example String	Example Format
Year	y	97, 2017	yy, y or yyyy
Year of "Week of year"	Y	1997	Y
Time Zone	Z, z	-0800, PST-08:00, PST	Z, ZZZZ, zzz
Month	M	9, 09, Sep, Septem-ber	M, MM, MMM, MMMM
Week of year (1-52)	w	8, 27	w, ww
Quarter of year	Q	Q2, 2nd quarter	QQQ, QQQQ
Day of week	E	Mon, Monday	EEE, EEEE
Day of month	d	1, 15	d, dd
Day of year (1-365)	D	23, 143	D, DD, DDD

Period (am/pm marker)	a	AM, am, PM	aa, aaaa
Hour (1-12), Hour (0-23)	h, H	1, 10, 16	h, hh, HH
Minute	m	8, 59	m, mm
Second in a minute	s	5, 05	s, ss
Fractional second	S	2, 23, 235, 2350	S, SS, SSS, SSSS
Milliseconds in day	A	34532	AAAAA

For a more complete list of date symbols, please see the Formatting Date and Times Overview on the International Components for Unicode page.

> **Note:** Some date formats are not supported by all databases and file types.

4. Check your format against the string displayed in Tableau. If the symbols and formatting is correct, select OK to create the new calculated field. The calculated field will act as a date field in your view.



Hyper extracts

For .hyper extract data sources, the symbols are defined by the Unicode Consortium. For more information, go to **Format syntax in DATEPARSE function for extract data sources** on page 1270 to see a the subset of field types and symbols that can be used with the DATEPARSE function in .hyper extracts.

Locale considerations

The DATEPARSE function relies on the locale specified by your computer settings to interpret and then display the strings that you want to convert. More specifically, the locale will affect whether a certain format can be recognized. This means that if a format is not supported by the locale, then you might see a null value or no value returned. For example, suppose you have the following string in your data:

12Sep2016:9:8:8.6546

The values returned from the DATEPARSE function for this string are different based on locale. In the English locale you will get a certain value, but for the Japanese locale you will get no value. In this case, no value is returned because the Japanese locale does not recognize "Sep."

Datetime value

English locale	Japanese locale
#9/12/2016 9:08:09 AM#	-

As with all dates, after you have used the DATEPARSE function to convert a string to a datetime type, by default Tableau will display the datetime value in the default format of your locale. If the locale changes, the result of the DATEPARSE function might display your new datetime value in a different format.

> **Note:** Because of a Jet limitation, the DATEPARSE function cannot correctly identify locale information specified by your computer settings in extracts created from Access data sources. To resolve this issue, consider exporting your Access data to Excel.

Create a calculation using the DATE function

If the DATEPARSE function is not available for the data that you're working with, or the field you are trying to convert is a number data type, you can use the DATE function instead.

The DATE function converts a number, string or date expression to a date type. When you create a calculation that uses the DATE function, Tableau creates a new field in your Tableau data source that allows you to interact with your date data as a date. To successfully produce date values from a number, string, or date expression using the DATE function, Tableau needs to be able to interpret the components of the string into date parts. After the components of the date are identified, Tableau uses the computer locale to determine the default format of the date.

For example, suppose the table you're working with contains a column of date data that is called "Original Date." The "Original Date" column is a string type.

Original Date
03Jan2017
05Jan2017
07Mar2017
19Mar2017
30Apr2017

In this case, you can create a calculated field called "New Date" that uses an expression in a DATE function to convert the string values in the "Original Date" field into date values.

For this example, the date expression is comprised of the LEFT function to isolate the day component, the MID function to isolate the month component, and the RIGHT function to isolate the year component.

```
DATE (LEFT([Original Date], 2) + "/" + MID([Original Date],3,3) +
"/" + RIGHT([Original Date],4))
```

The "New Date" calculation produces the following column:

New Date
1/3/2017
1/5/2017
3/7/2017
3/19/2017
4/30/2017

In this example, the new date values are based on an English locale and default formatting.

See Also

Custom Date Formats on page 1261

Format Dates Using ISO-8601 Weeks and Years on page 1279

Date Functions on page 2170

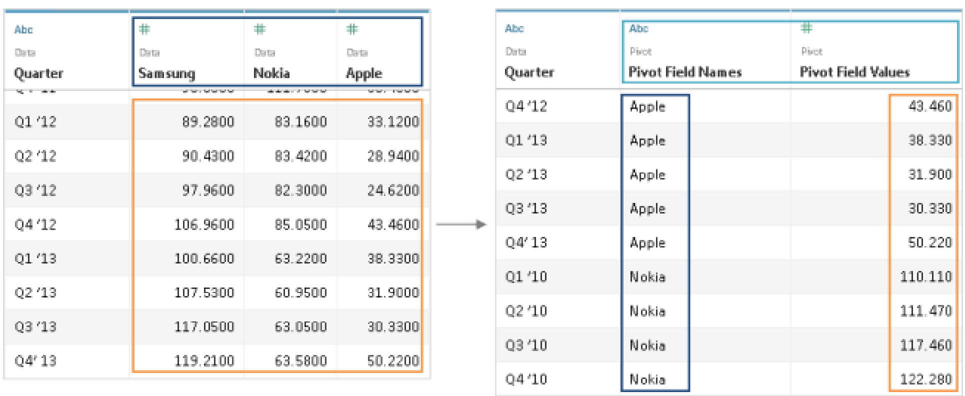
All Functions (by Category): Date

STR() Function Ignores the Default Date and Number Formatting (Tableau Knowledge Base)

Pivot Data from Columns to Rows

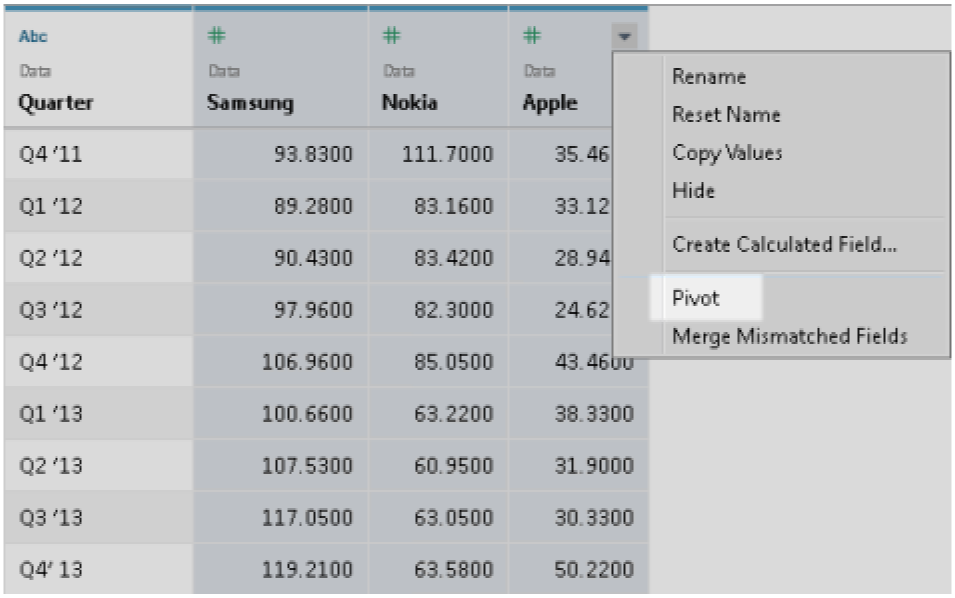
Sometimes, analyzing data that is stored in a crosstab format can be difficult in Tableau. When working with Microsoft Excel, text file, Google Sheets, and .pdf data sources, you can pivot your data from crosstab format into columnar format. If you are working with other data sources, you can **Pivot using custom SQL (Tableau Desktop)** on page 990.

For example, suppose you have the number of devices sold by quarter for three vendors in three separate fields. You can pivot your data so that the vendor is in one field and the number of devices sold is in another field.



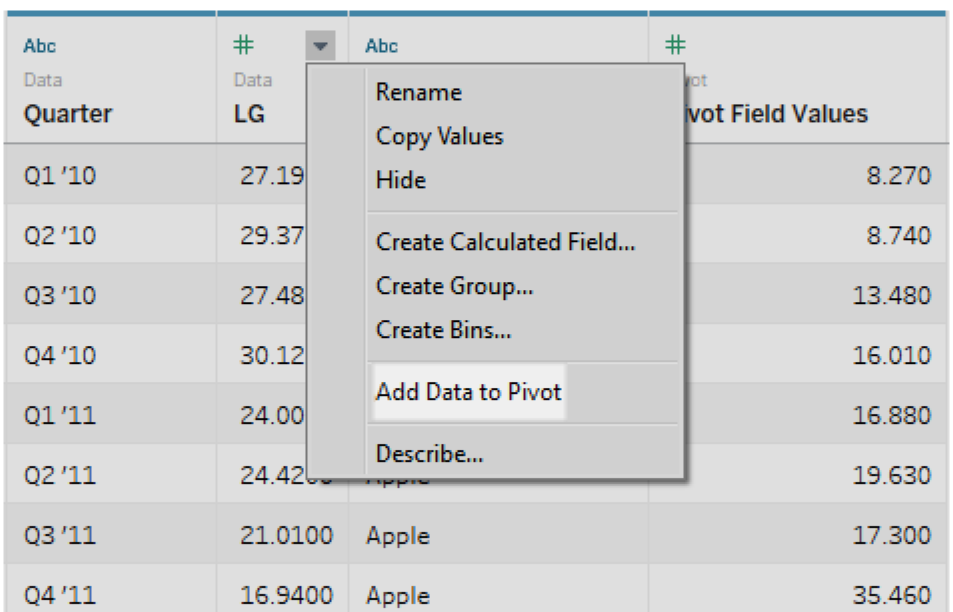
Pivot the data

After you have set up the data source, in the grid, select two or more columns. Click the drop-down arrow next to the column name, and then select **Pivot**. New columns called "Pivot field names" and "Pivot field values" are created and added to the data source. The new columns replace the original columns that you selected to create the pivot.



Add to the pivot

To add more data to the pivot, select another column, click the drop-down arrow next to the column name, and then select **Add Data to Pivot**. Make sure that the pivot columns and values look as expected before you begin your analysis.



To remove a pivot, click the drop-down arrow next to the name of a pivot column, and then select **Remove Pivot**.

Troubleshooting pivots

- **Red fields in the view and fields with exclamation points in the Data pane**: Because the original fields are replaced with new pivot fields, any references to the original fields in the view will no longer work. They cause fields to become red in the view or show a red exclamation point next to the field in the **Data** pane.
- **Null values in the grid:** If all of the original fields used in the pivot are removed, for example in an extract refresh, null values display in the pivot fields.
- **No pivot option**: Pivot appears when you select two or more columns in a single Microsoft Excel, text file, Google Sheets, and .pdf data source. If using a different data source in Tableau Desktop, you can use custom SQL to pivot.

Pivot using custom SQL (Tableau Desktop)

You can also use custom SQL to pivot your data, even if you aren't working Excel, text file, Google Sheets, and .pdf data sources. When you use the UNION ALL operator in a custom SQL query, you can take values from distinct columns and put them into a new column.

For example, suppose you have a table called **Contest**.

Contest

Runner	Start Time	End Time
Amanda	9/3/2016 3:04 PM	9/3/2016 3:25 PM
Oscar	9/3/2016 3:04 PM	9/3/2016 3:21 PM
William	9/3/2016 3:04 PM	9/3/2016 3:16 PM

To optimize your analysis of this data in Tableau, you can use the following custom SQL query to pivot the "Start Time" and "End Time" columns so that their values are in a single column.

```
Select [Runner]

, 'Start' as [Action]

, [Start Time] as [Time]

From [Contest]

Union ALL
```

```
Select [Runner]

, 'End' as [Action]

, [End Time] as [Time]

From [Contest]
```

The above custom SQL query does the following:

- Pivots the **Start Time** column header into a string value called **Start** and adds that value to a new column called **Action**.
- Pivots the **End Time** column header into a string value called **End** and adds that value to a new column called **Action**.
- Pivots the **Start Time** and **End Time** columns so that their values are in a new column called **Time**.

The following table shows the results of this custom SQL query.

Runner	Action	Time
Amanda	Start	9/3/2016 3:04 PM
Oscar	Start	9/3/2016 3:04 PM
William	Start	9/3/2016 3:04 PM
Amanda	End	9/3/2016 3:25 PM
Oscar	End	9/3/2016 3:21 PM
William	End	9/3/2016 3:16 PM

To pivot data using custom SQL

1. Connect to your data.
2. Double-click the **New Custom SQL** option in the left pane. For more information, see **Connect to a Custom SQL Query** on page 959.
3. In the **Edit Custom SQL** dialog box, copy and paste the following custom SQL query and replace the contents with information about your table:

```
Select [Static Column]

, 'New Value (from Column Header 1)' as [New Column Header]

, [Pivot Column Values 1] as [New Values]

From [Table]

Union ALL

Select [Static Column]

, 'New Value (from Column Header 2' as [New Column Header]

, [Pivot Column Values 2] as [New Values]

From [Table]

Union ALL

Select [Static Column]

, 'New Value (from Column Header 3' as [New Column Header]

, [Pivot Column Values 3] as [New Values]

From [Table]
```

Where the following is true:

- *Static Column*: A comma-delimited list of the columns from *Table*, both dimensions and measures, which should not be included in the pivot.
- *New Value (from Column Header 1-3)*: New names that you give to the original column headers, which are used as row values in the pivot.
- *Pivot Column Values 1-3*: The columns whose values need to be pivoted into a single column.
- *New Column Header*: The name you give the new column that contains the new row values from *New Value (from Column Header 1-3)*.
- *New Values*: The name give the new column that contains the original values

from *Pivot Column Values 1-3*.

- *Table*: The table that you connected to.

4. Click **OK**.

Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter

When you track data in Excel spreadsheets, you create them with the human interface in mind. To make your spreadsheets easy to read, you might include things like titles, stacked headers, notes, maybe empty rows and columns to add white space, and you probably have multiple tabs of data too.

When you want to analyze this data in Tableau, these aesthetically pleasing attributes make it very difficult for Tableau to interpret your data. That’s where Data Interpreter can help.

> **Tip:** Though Tableau's Excel add-in is no longer supported, Data Interpreter can help you reshape your data for analysis in Tableau.

What does Data Interpreter do?

Data Interpreter can give you a head start when cleaning your data. It can detect things like titles, notes, footers, empty cells, and so on and bypass them to identify the actual fields and values in your data set.

It can even detect additional tables and sub-tables so that you can work with a subset of your data independently of the other data.

After Data Interpreter has done its magic, you can check its work to make sure it captured the data that you wanted and identified it correctly. Then, you can make any necessary adjustments.

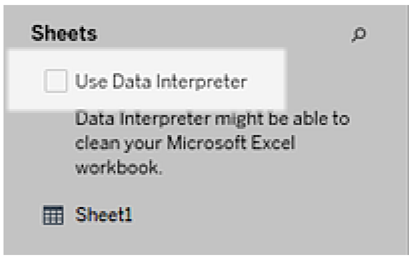
After you select the data that you want to work with, you might also need to do some additional cleaning steps like pivoting your data, splitting fields, or adding filters to get the data in the shape you want before starting your analysis.

> **Note**: If your data needs more cleaning than what Data Interpreter can help you with, try Tableau Prep.

Turn on Data Interpreter and review results

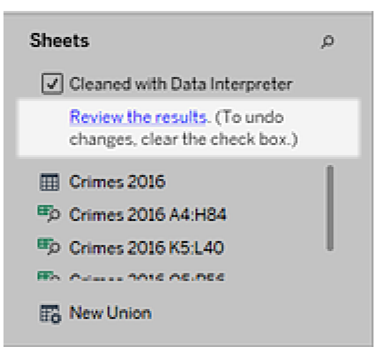
1. From the **Connect** pane, connect to an Excel spreadsheet or other connector that supports Data Interpreter such as Text (.csv) files, PDF files or Google sheets.

2. Drag a table to the canvas (if needed), then on the **Data Source** page, in the left pane, select the **Use Data Interpreter** check box to see if Data Interpreter can help clean up your data.

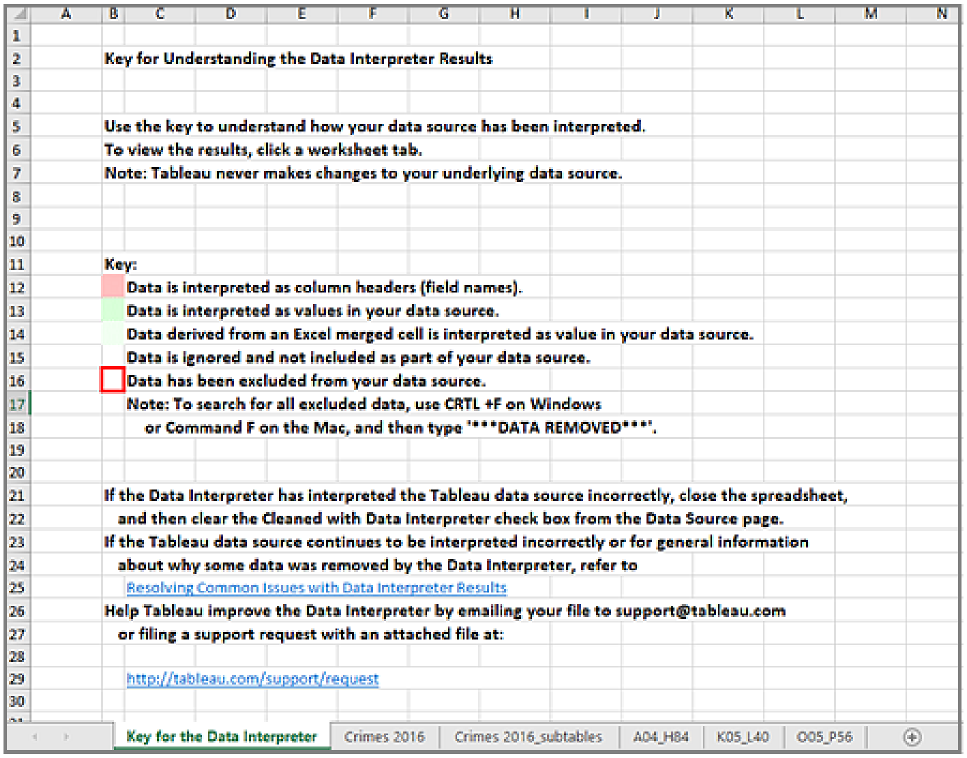
   

   > **Note:** When you clean your data with Data Interpreter, Data Interpreter cleans all the data associated with a connection in the data source. Data Interpreter does not change the underlying data.

3. In the Data pane, click the **Review the results** link to review the results of the Data Interpreter.

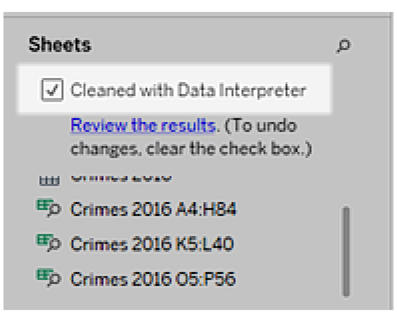
   A copy of your data source opens in Excel on the **Key for the Data Interpreter** tab. Review the key to find out how to read the results.



4. Click each tab to review how Data Interpreter interpreted the data source.

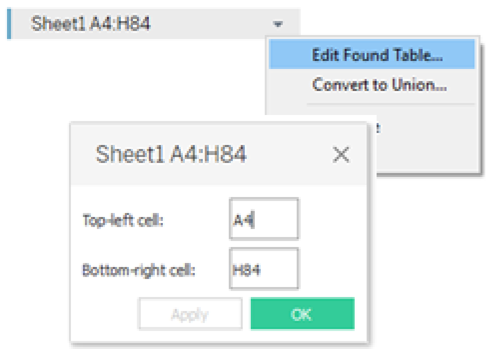
   If Data Interpreter found additional tables, also called found tables or sub-tables, they are identified in the <sheet name>_subtables tab by outlining their cell ranges. A separate tab is also included for each sub-table, color coded to identify the header and data rows.

   If Data Interpreter does not provide the expected results, clear the **Cleaned with Data Interpreter** check box to use the original data source.



5. To replace the current table with any of the found tables, drag the current table off the canvas and then drag the found table that you want to use to the canvas.

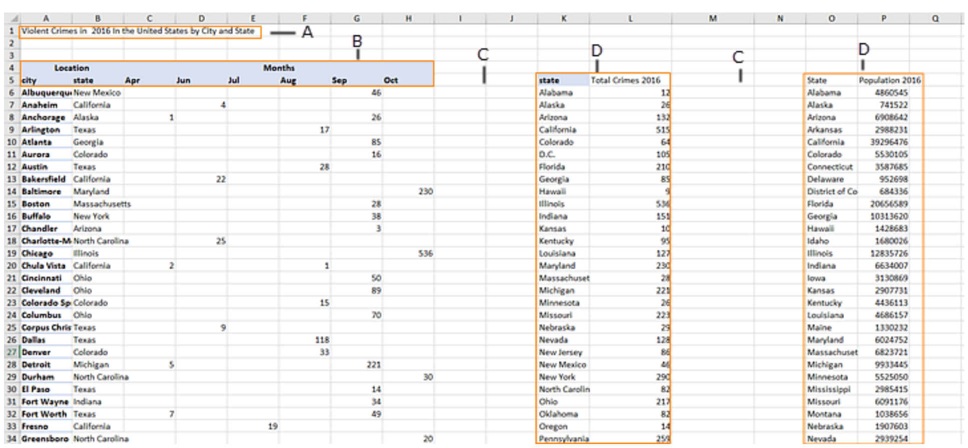
If Data interpreter has misidentified the range of the found table, after you drag the found table to the canvas, click the drop-down arrow on that table, and then select **Edit Found Table** to adjust the corners of the found table (the top-left cell and bottom-right cell of the table).



6. After you have the data that you want to work with, you can apply any additional cleaning operations to your data so that you can analyze it.

Data Interpreter Example

In this example we are connecting to an Excel spreadsheet with violent crime data by city and state for the year 2016. This spreadsheet includes multiple tables on one sheet and some extra formatting.



A. Title

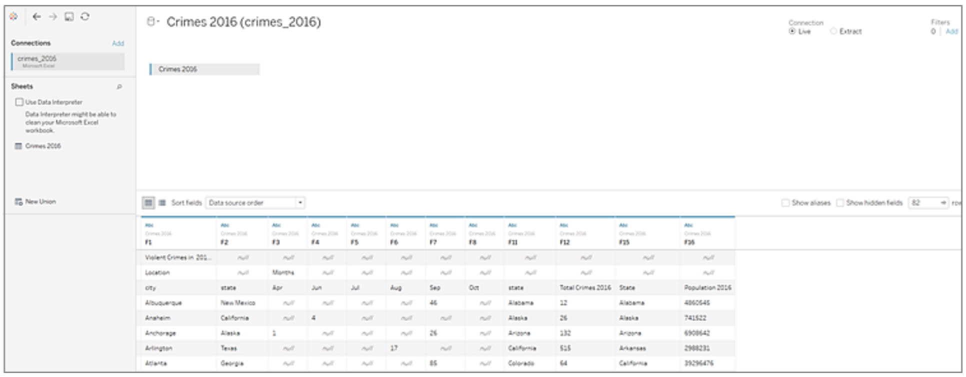
B. Merged header cells

C. Extra white space

D. Sub-tables

The extra formatting in this spreadsheet makes it difficult for Tableau to determine what the field headers and values are.

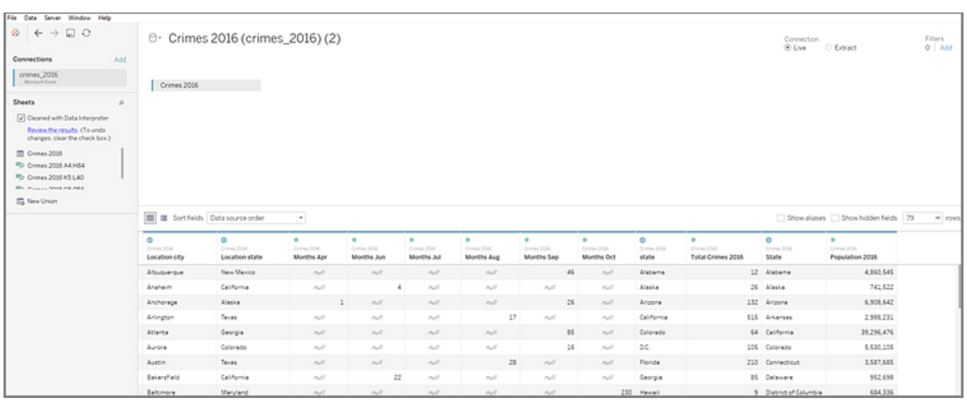
Instead, it reads the data vertically and assigns each column the default value F1, F2, F3 (Field 1, Field 2, Field 3) and so on. Blank cells are read as null values.



To see if Data Interpreter can help clean this data set, we select **Use Data Interpreter**.

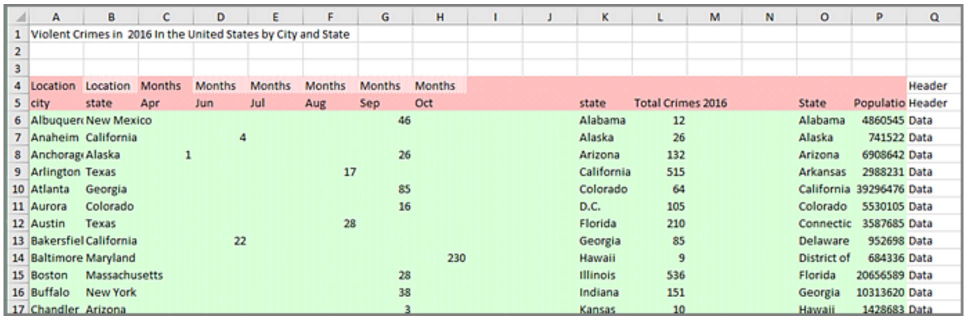
Data Interpreter detected the proper headings for the fields, removed the extra formatting and found several sub-tables. The sub-tables are listed in the **Sheets** section in the Data pane and are named using the original sheet name and the cell ranges for each sub-table.

In this example there are three sub-tables: **Crimes 2016 A4:H84**, **Crimes 2016 K5:L40**, and **Crimes 2016 O5:P56**.



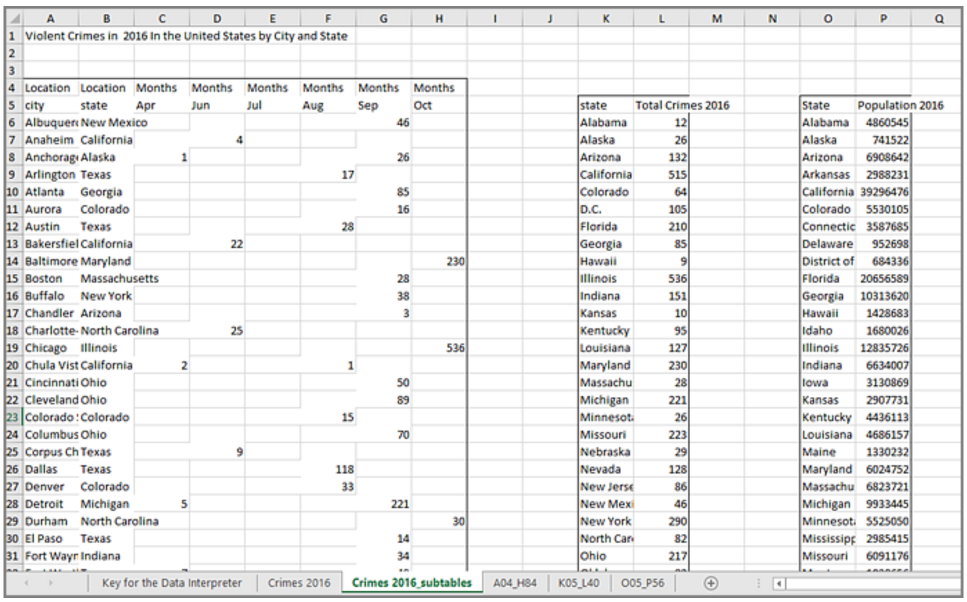
To examine the results of the Data Interpreter more closely, we click the **Review the results** link in the Data pane to view an annotated copy of the spreadsheet.

Here we see a copy of the original data, color coded to identify which data was identified as header data and which data was identified as field values.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Violent Crimes in 2016 In the United States by City and State																
2																	
3																	
4	Location	Location	Months	Months	Months	Months	Months	Months									Header
5	city	state	Apr	Jun	Jul	Aug	Sep	Oct			state	Total Crimes 2016			State	Populatio	Header
6	Albuquer	New Mexico					46				Alabama	12			Alabama	4860545	Data
7	Anaheim	California		4							Alaska	26			Alaska	741522	Data
8	Anchorag	Alaska	1				26				Arizona	132			Arizona	6908642	Data
9	Arlington	Texas				17					California	515			Arkansas	2988231	Data
10	Atlanta	Georgia					85				Colorado	64			California	39296476	Data
11	Aurora	Colorado					16				D.C.	105			Colorado	5530105	Data
12	Austin	Texas				28					Florida	210			Connectic	3587685	Data
13	Bakersfiel	California		22							Georgia	85			Delaware	952698	Data
14	Baltimore	Maryland						230			Hawaii	9			District of	684336	Data
15	Boston	Massachusetts					28				Illinois	536			Florida	20656589	Data
16	Buffalo	New York					38				Indiana	151			Georgia	10313620	Data
17	Chandler	Arizona					3				Kansas	10			Hawaii	1428683	Data

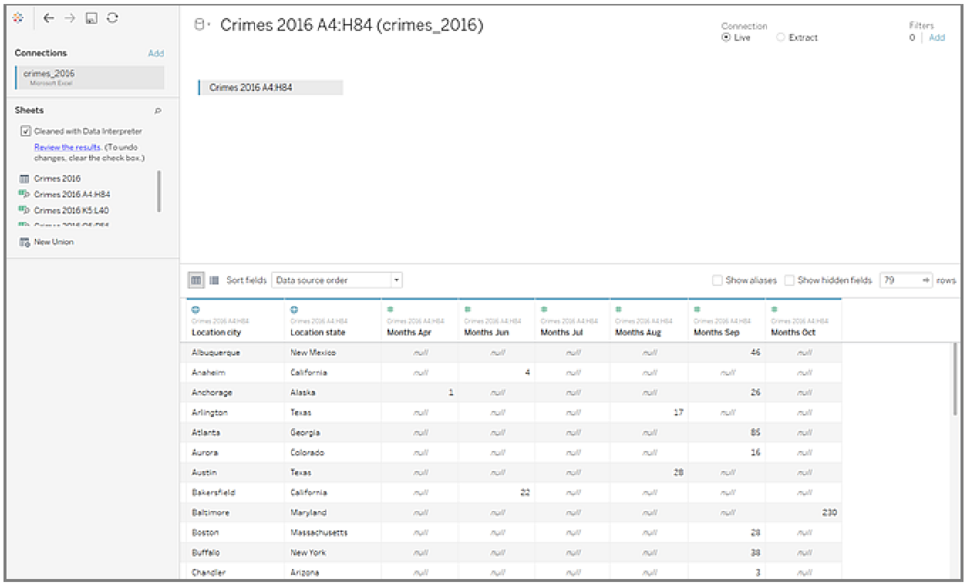
The next tab shows us the sub-tables that Data Interpreter found, outlined by the cell ranges.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	Violent Crimes in 2016 In the United States by City and State																
2																	
3																	
4	Location	Location	Months	Months	Months	Months	Months	Months									
5	city	state	Apr	Jun	Jul	Aug	Sep	Oct			state	Total Crimes 2016			State	Population 2016	
6	Albuquer	New Mexico					46				Alabama	12			Alabama	4860545	
7	Anaheim	California		4							Alaska	26			Alaska	741522	
8	Anchorag	Alaska	1				26				Arizona	132			Arizona	6908642	
9	Arlington	Texas				17					California	515			Arkansas	2988231	
10	Atlanta	Georgia					85				Colorado	64			California	39296476	
11	Aurora	Colorado					16				D.C.	105			Colorado	5530105	
12	Austin	Texas				28					Florida	210			Connectic	3587685	
13	Bakersfiel	California		22							Georgia	85			Delaware	952698	
14	Baltimore	Maryland						230			Hawaii	9			District of	684336	
15	Boston	Massachusetts					28				Illinois	536			Florida	20656589	
16	Buffalo	New York					38				Indiana	151			Georgia	10313620	
17	Chandler	Arizona					3				Kansas	10			Hawaii	1428683	
18	Charlotte-	North Carolina		25							Kentucky	95			Idaho	1680026	
19	Chicago	Illinois						536			Louisiana	127			Illinois	12835726	
20	Chula Vist	California	2			1					Maryland	230			Indiana	6634007	
21	Cincinnati	Ohio					50				Massachu	28			Iowa	3130869	
22	Cleveland	Ohio					89				Michigan	221			Kansas	2907731	
23	Colorado :	Colorado				15					Minnesot	26			Kentucky	4436113	
24	Columbus	Ohio					70				Missouri	223			Louisiana	4686157	
25	Corpus Ch	Texas		9							Nebraska	29			Maine	1330232	
26	Dallas	Texas				118					Nevada	128			Maryland	6024752	
27	Denver	Colorado				33					New Jerse	86			Massachu	6823721	
28	Detroit	Michigan	5				221				New Mexi	46			Michigan	9933445	
29	Durham	North Carolina						30			New York	290			Minnesot	5525050	
30	El Paso	Texas					14				North Car	82			Mississipp	2985415	
31	Fort Wayn	Indiana					34				Ohio	217			Missouri	6091176	

Key for the Data Interpreter | Crimes 2016 | Crimes 2016_subtables | A04_H84 | K05_L40 | O05_P56

In this example the first sub-table, **Crimes 2016 A4:H84**, has the main data that we want to work with. To use this table as our data table, we can simply drag the original table off the canvas and then drag the new table to the canvas.



Once we have the data that we want to work with in the canvas, we can do some additional clean up on the data. For example we can:

- Change the field names so that they represent city, state, and month names.
- Pivot the months fields.
- Drag in the third sub-table **Crimes 2016 o5:P56** and join it to our first sub-table on the **State** field to include state populations for our analysis.
- Hide any duplicate fields that were added as a result of the join.

The results might look something like this:

Sort fields Data source order

Crimes 2016 A4:H84 City	Crimes 2016 A4:H84 State	Crimes 2016 O5:P56 Population 2016	Pivot Months	Pivot Crimes
Phoenix	Arizona	6,908,642	August	111
Pittsburgh	Pennsylvania	12,787,085	August	null
Plano	Texas	27,904,862	August	5
Portland	Oregon	4,085,989	August	null
Raleigh	North Carolina	10,156,689	August	null
Riverside	California	39,296,476	August	7
Sacramento	California	39,296,476	August	null
San Antonio	Texas	27,904,862	August	null
San Diego	California	39,296,476	August	30
San Francisco	California	39,296,476	August	null
San Jose	California	39,296,476	August	35
Santa Ana	California	39,296,476	August	null
Seattle	Washington	7,280,934	August	14
St. Louis	Missouri	6,091,176	August	133
St. Petersburg	Florida	20,656,589	August	14

Now we are ready to start analyzing our data in Tableau.

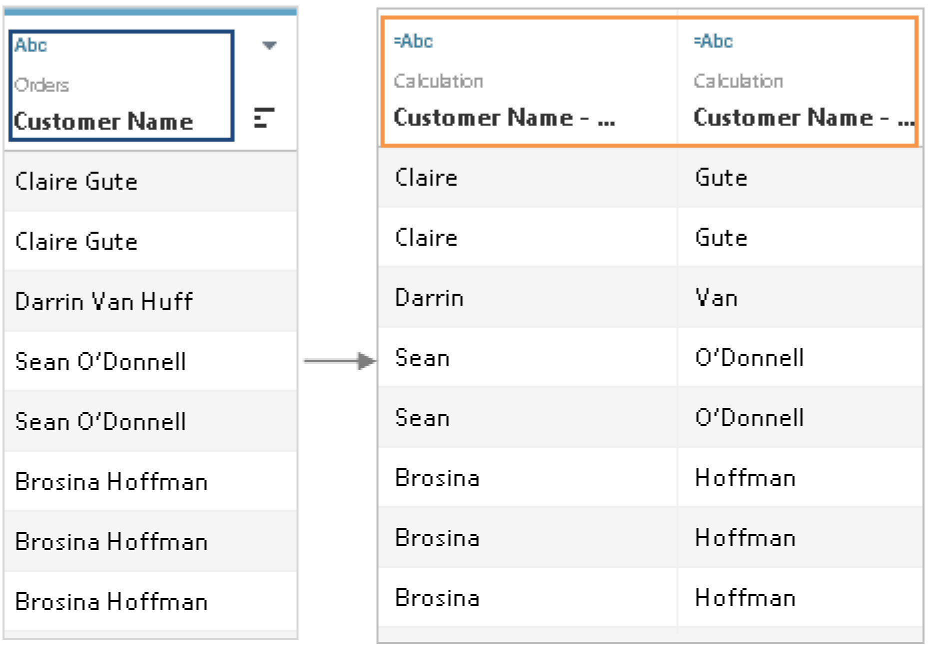
When Data Interpreter is not available

The Data Interpreter option might not be available for the following reasons:

- **The data source is already in a format that Tableau can interpret:** If Tableau Desktop doesn't need extra help from Data Interpreter to handle unique formatting or extraneous information, the Data Interpreter option is not available.
- **Many rows or many columns:** The Data Interpreter option is not be available when your data has the following attributes:
  - Data contains more than 2000 columns.
  - Data contains more than 3000 rows and more than 150 columns.
- **The data source is not supported:** Data Interpreter is only available for Microsoft Excel, Text (.csv) files, PDF files and Google Sheets. For Excel, your data must be in the .xls or .xlsx format.

Split a Field into Multiple Fields

If you have string fields in your data that contain multiple units of information, for example, the first and last name of a customer, it might be easier to analyze the data if you split the values in that field into separate fields. You can use split or custom split options in Tableau to separate the values based on a separator or a repeated pattern of values present in each row of the field. In this example, the common separator is a space character ( ).



The new fields created by the split or custom split are added to the data source as calculated fields.

Splits and custom splits are based on the SPLIT string function. For more information, see **String Functions** on page 2160.

You can tell if your data supports the SPLIT function by checking the field drop-down menu:

- On the Data Source page, check the menu for **Split** and **Custom Split**.
- On the Data pane in the worksheet, check the menu for **Transform** > **Split** and **Custom Split**.

Because different connection types support different functionality in Tableau, you may need to check both places to determine if your data supports the split and custom split options.

Split fields automatically

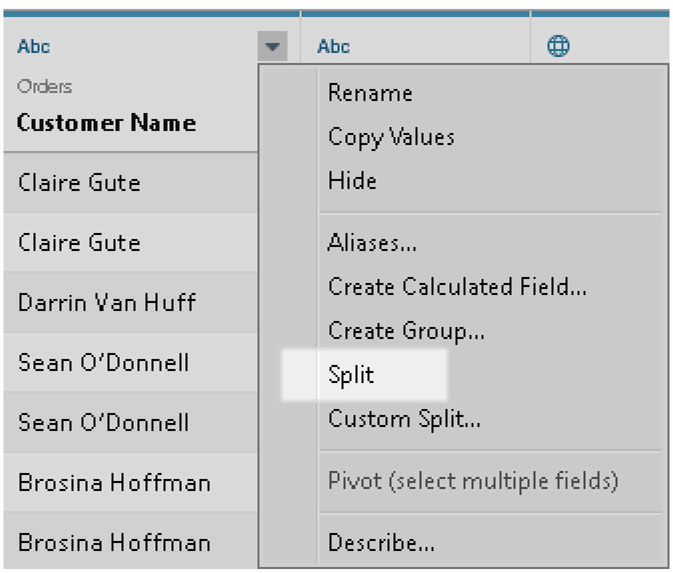
A string field can be split automatically based on a common separator that Tableau detects in the field.

Depending on the connection type, a split can automatically separate a field's values in up to ten new fields. The data type of the new fields generated by the split can vary depending on the pattern combination that Tableau detects in the separator that is used to split the original field.

> **Note:** In some cases, if the split takes too long to generate new fields or Tableau cannot find a common separator, a custom split dialog box displays. For more information, see the **Custom split** on the facing page section.

To split a field automatically

1. On the Data Source page, in the grid, click the drop-down arrow next to the field name.
2. Select **Split**.



> **Note:** You can also use the split option from the Data pane in the worksheet. In the Data pane, right-click the field you want to split, and then select **Transform** > **Split**.

If you do not like the results of the split, you can go to the Data pane and edit the calculated fields that are created by the split. Alternatively, you can click undo in the Tableau Desktop toolbar or remove the split.

To remove fields created by a split

1. On the Data Source page, in the grid, click the drop-down arrow next to the field name.
2. Select **Delete**.
3. Repeat steps 1-2 to remove all the fields created by the split.

Alternatives to automatic split

Sometimes, using automatic split is not the best option. The following are example of cases when you should not split fields automatically:

- **Values contain different number of separators:** Fields cannot be split automatically if the number of separators varies from value to value. For example, suppose a field has the following values:

  jsmith| accounting | north

  dnguyen | humanresources

  lscott | recruiting| west

  karnold |recruiting |west

  In cases like this, consider using a custom split. For more information, see the **Custom split** below section.

- **Values contain mixed separators:** Fields cannot be split automatically if the separator types are different. For example, suppose a field contains the following values:

  smith.accounting

  dnguyen-humanresources

  lscott_recruiting

  karnold_recruiting

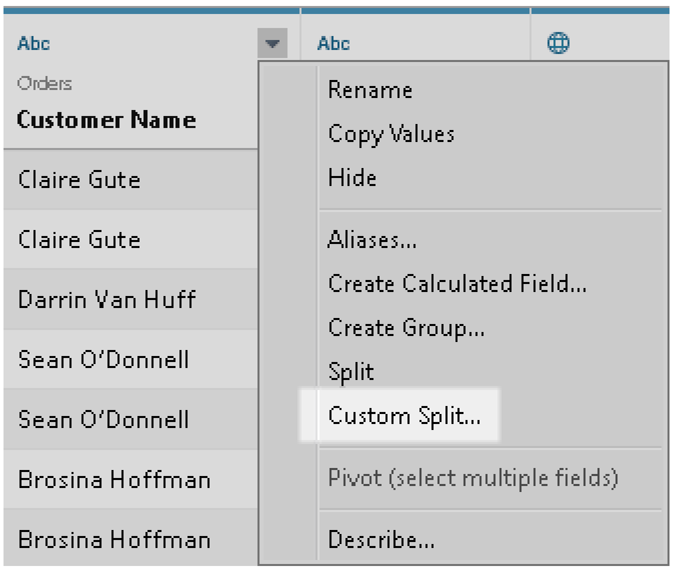
  In cases like this, consider using regular expressions to create new fields. For more information, see **Additional Functions** on page 2260.

Custom split

You can use the custom split option to specify a common separator for the split. Like the split option, a custom split can separate a field's values in up to ten new fields. In addition, you can choose to split the values at the first n occurrences of the separator, the last n occurrences of the separator, or at all occurrences of the separator. The data type of the new fields generated by the custom split always results in a string data type.

To use a custom split

1. On the Data Source page, in the grid, click the drop-down arrow next to the field name.
2. Select **Custom Split**.



> **Note:** You can also access the custom split option from the Data pane. In the Data pane, right-click the field you want to split, and then select **Transform** > **Custom Split**.

If you do not like the results of the split, you can go to the Data pane and edit the calculated fields that are created by the split. Alternatively, you can click undo in the toolbar or remove the split.

To remove fields created by a custom split

1. On the Data Source page, in the grid, click the drop-down arrow next to the filed name.
2. Select **Delete**.
3. Repeat steps 1-2 to remove all the fields created by the custom split.

At a glance: Working with splits and custom splits

The following is a list of additional notes you might need to know about splits and custom splits.

- New fields generated from a split or custom split cannot be used as keys to join tables, but can be used to blend data sources.
- New fields generated from a split or custom split cannot be used in a pivot.
- Split and custom split options are not supported for sets, groups, parameters, and bins.
- Microsoft SQL Server only allows up to four split fields.
- To generate more than ten new fields, consider using a split or custom split on the field that was previously generated by a split or custom split.

Troubleshooting splits and custom splits

The following is a list of issues you might experience when using splits and custom splits:

- **Split and custom split options missing for a supported data source type:** Split and custom split options are available only for fields that are a string data type.
- **Null values or empty cells:** After creating a split or a custom split, new fields might contain null values or no values at all. Null values or empty cells occur when there are no values for all of the expected new fields.
- **Data has been removed:** Tableau might use portions of the field’s values as a separator. If a portion of a field’s values is used as a separator, those values no longer appear in the new fields. For example, suppose a field contains the following values:

  ZIP-98102

  ZIP-98109

  ZIP-98119

  ZIP-98195

  In this case, the split will create a new field with the following values:

  98102

  98109

  98119

  98195

The split will not create a separate field for “ZIP-“ because the split uses it as a separator.

Filter Data from Data Sources

You can create filters on a data source, thereby reducing the amount of data in the data source. This feature is available to Creators when authoring in Tableau Online and Tableau Server.

If you create an extract from a data source that already has data source filters in place, those filters are automatically recommended as extract filters, and will appear in the Extract dialog. Those recommended filters are not required to be part of the Extract filter list, and can safely be removed without affecting the existing set of data source filters.

Data source filters can be useful for restricting the data users can see when you publish a workbook or data source. When you publish a data source to Tableau Server, the data source and any associated files or extracts are transported in entirety to the Server. As you publish a data source you can define access permissions for downloading or modifying the data source, and you can also choose the users and groups who can remotely issue queries through Tableau Server against that data source. When users have query permission and no download permission, you can share a rich data model having calculated fields, aliases, groups, sets and more—but only for querying.

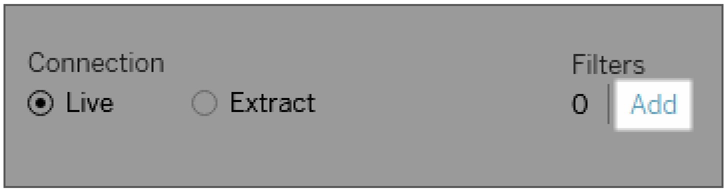
Furthermore, users who query published data source will never be able to see or modify any data source filters present on the originally published data source, but all of the users' queries will be subject to those data source filters. This is a great way to offer a restricted subset of your data, for example by filtering dimensions for specific users and groups, or by defining data source filters based on a fixed or relative date range. This is often useful for data security, and it also allows you to manage performance of the remote database which Tableau Server will ultimately query on a user's behalf. For systems that rely heavily on partitions or indexing, data source filters may yield tremendous control over the performance of queries issued by Tableau.

Create a data source filter

The primary way to create a data source filter is from the data source page.

To create a data source filter

1. On the data source page, click **Add** in the Filters section in the upper-right corner of the page.



   To create a data source filter on a worksheet, right-click (control-click on a Mac) the data source and choose **Edit Data Source Filters**.

   Whether you start from the Data Source page or from a worksheet, you see an Edit Data Source Filter dialog box, listing any existing data source filters.

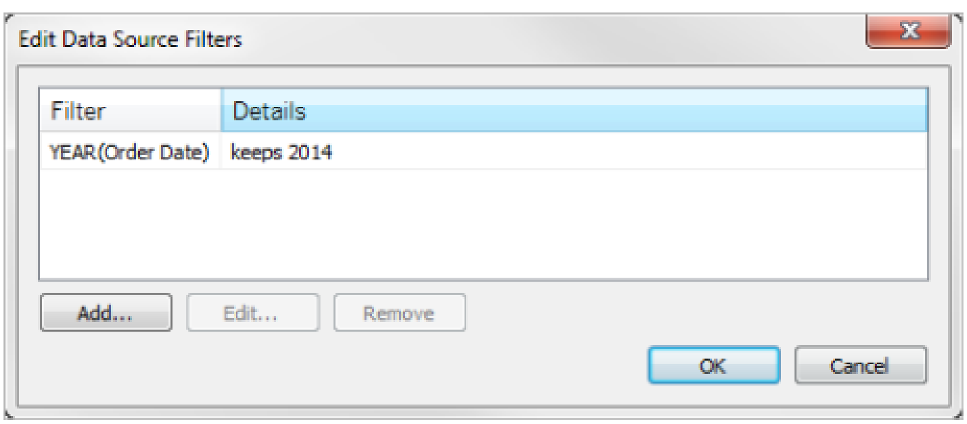
2. Click **Add** to open an Add Filter dialog box listing all fields in the data source.
3. Click to select a field to filter; then specify how the field should be filtered, just as you would for a field on the Filters shelf.

   To add an additional data source filter, repeat this procedure.

Global filters and data source filters

When you create a data source filter, any global filters that use that data source are displayed automatically in the **Edit Data Source Filters** dialog box.

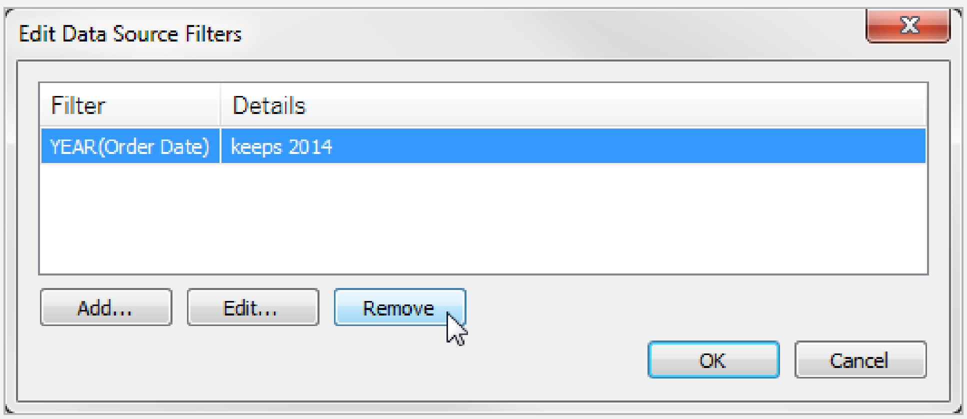
In Tableau Desktop, you can promote a global filter to be a data source filter. To promote the global filter to be a data source filter, click **OK**.



If you promote a global filter to be a data source filter in Tableau Desktop, that global filter will no longer be visible in worksheets of the workbook (because it becomes a data source filter).

Important: Be aware that you do not need to select a global filter in the **Edit Data Source Filters** dialog box to promote it. When you click **OK**, all global filters in the list will be promoted.

To prevent a global filter from being promoted to a data source filter, select the global filter in the **Edit Data Source Filters** dialog box, and then click **Remove**.



Understand Field Type Detection and Naming Improvements

Tableau uses a collection of commonly used patterns to detect and transform your data to make it easier for you to interact with. For example, when you work with fields in the Data pane, Tableau takes the data from your data source, detects its type, and divides it into dimensions and measures.

Note: Tableau never changes your underlying data.

Tableau can also detect fields that should be treated as dimensions and clean up field names by making them more readable. You see these improvements only when the field names, which come from column headers in your data source, meet the patterns or conditions described below.

New fields added to underlying data

If you add new fields to your underlying data, and then refresh the data source in Tableau, the newly added fields will retain their original database names. If you would like Tableau to automatically rename those fields, you will need to create a new connection to that same data source.

If you prefer to maintain the field types and field names from the underlying data, see the Undo changes made by Tableau section at the bottom of this article.

Fields that are treated as a dimension

Field names that contain certain keywords are treated as dimensions, even if the values for those fields are numeric.

Keywords Code, Key, and ID

Field names that contain the following keywords and meet the conditions listed are treated as dimensions instead of measures.

Conditions:

- Contains keywords Code, ID, or Key.
- Keywords are either separated from other text in the field name by non-letter characters, all capitalized, or the first letter is capitalized in a field name that otherwise has mixed casing.
- Keywords are at the beginning or end of the field name, with leading or trailing non-letter characters. In Chinese, Japanese, and Korean, the key word must be located at the end of the field name.

Language	**Keywords**
English (UK and US)	Code, Id, Key
Chinese (Simplified and Traditional)	Id
French	Id
German	Schlüssel, Schlussel
Italian	Chiave, Id, Codice
Japanese	Id

Language	Keywords
Korean	Id
Portuguese	Chave, Código, Codigo, Identificação, Identificador
Spanish	Código, Codigo

Keywords Number, Num, and Nbr

Field names that end with Number, Num, or Nbr are treated as a dimension instead of a measure. For example, “Record Number” is treated as a dimension but “Number of Records” is not. In addition, the Korean field name must be four or fewer characters long.

Language	Keywords
English (UK and US)	Number, Num, Nbr
Chinese (Simplified and Traditional)	-
French	Num
German	Nr, Nummer
Italian	Numero, N°
Japanese	番号
Korean	번호
Portuguese	Num
Spanish	Num

Keywords related to dates

Field names containing keywords that are recognized as date parts are treated as dimensions. These field names can only contain only one additional word unrelated to dates to qualify. For example, “Fiscal Year” is treated as a dimension but “Fiscal Year Information” is not. In addition, Chinese field names must be four or fewer characters long and cannot contain digits. Japanese and Korean field names must be four or fewer characters long.

Language	Keywords
English (UK and US)	Year, Yr, Day, Day of Week, Week, Wk, Month, Quarter, Qtr, FY
Chinese (Simplified and Traditional)	年, 月
French	Annee, Année, Jour, Mois, Semaine
German	Jahr, Monat, Woche
Italian	Anno, A, Giorno, Giorno della Settimana, Settimana, Sett, Mese, Trimestre, Trim, AF
Japanese	年度
Korean	년도
Portuguese	Ano, Dia, Semana, Mês, Trimestre
Spanish	Año, Ano, Día, Dia, Mes, Semana, Trimestre

Field name clean-up

Field names that contain specific characters or capitalized in a certain way are renamed. Field values that include square brackets are automatically converted to parentheses.

Field names with underscore and space characters

Non-leading and trailing underscore (_) characters in field names are converted into space ( ) characters. Carriage return or line feed characters in a field name are removed. However, field names that contain leading underscore characters remain unchanged. In addition, leading and trailing spaces in field names are removed. This type of field name clean up applies only to English, French, German, Portuguese, and Spanish languages.

> **Example 1:** The field name “Country_Name” is converted to “Country Name”.
> **Example 2:** The field name “_Days_On_Market” is converted to “_Days On Market”.

Field names with multiple underscore characters in a row remain unchanged. For example, “Country__Name” remains as “Country__Name”.

Field names and capitalization

Field names that use all capital letters with non-letter characters are converted to all lower-case letters except for the characters immediately after the non-letter character.

Additionally, field names that use all lower-case letters are converted so the first letter in the field name is capitalized.

This type of field name clean up applies only to English, French, German, Portuguese, and Spanish languages.

> **Example 1:** The field name “PC1” is converted to “Pc1.” The field name “COUNTRY_NAME” is converted to “Country Name”. However, “Budget COGS” remains as “Budget COGS” .
> **Example 2:** The field name “FDA” does not change. However, the field name “FDA_Sales” is converted to “Fda Sales”.
> **Example 3:** The field name “age” is converted to “Age”.

Space characters are added to the case boundaries of field names that contain mixed casing with non-letter characters.

Example: The field name “ThisCase” is converted to “This Case”.

Field names that are two or three letters long

Field names that are two or three letters long that are part of a multiple word phrase are converted so that each letter in the word is capitalized, unless the two or three letter word contains a vowel (i.e., a, e, i, o, or u).

> **Example:**The field name "Unit Qty" is converted to "Unit QTY". However, the field name "Sales Amt" remains as "Sales Amt".

Several three-letter acronyms are converted to all uppercase. Examples of three-letter acronyms include the following:

CIF	FDA	MPI
DMA	FOB	MSA
DOB	FTE	SKU
EIN	KPI	UPC

ESP	LOB	URL
		USD

The following field names are converted to all lowercase letters unless the name occurs at the beginning of the field name:

1st	as	down	of	th (suffix following a number)
2nd	at	for	on	than
3rd	but	in	off	the
a	by	inclu.	or	to
an	de	into	over	vs
and	da	la	per	with

Note: The casing of field names that are one letter long remain unchanged.

Undo changes made by Tableau

If you prefer to maintain the field types and field names specified by the underlying data, you can undo changes made by Tableau either manually, at the data source level, or automatically, at the application level.

To undo changes manually

1. Open your workbook and go to the Data Source page.
2. Click the metadata grid button.
3. Press **Ctrl+Click** (Windows) or **Command+Click** (Mac) to select the rows with changes you want to undo.
4. Click the drop-down arrow for one of the rows selected, and select **Reset name**.

To automatically undo changes made by Tableau at the application level, follow the procedure in the Overriding Automatic Field Renaming article. This method affects all data sources that you access from Tableau Desktop.

Extract Your Data

Extracts are saved subsets of data that you can use to improve performance or to take advantage of Tableau functionality not available or supported in your original data. When you create an extract of your data, you can reduce the total amount of data by using filters and configuring other limits. After you create an extract, you can refresh it with data from the original data. When refreshing the data, you have the option to either do a full refresh, which replaces all of the contents in the extract, or you can do an incremental refresh, which only adds rows that are new since the previous refresh.

Extracts are advantageous for several reasons:

- **Supports large data sets:** You can create extracts that contain billions of rows of data.
- **Help improve performance:** When you interact with views that use extract data sources, you generally experience better performance than when interacting with views based on connections to the original data.
- **Support additional functionality:** Extracts allow you to take advantage of Tableau functionality that's not available or supported by the original data, such as the ability to compute Count Distinct.
- **Provide offline access to your data:** If you are using Tableau Desktop, extracts allow you to save and work with the data locally when the original data is not available. For example, when you are traveling.

Latest changes to extracts

Extracts in the web

Beginning with version 2020.4, extracts are available in web authoring and content server. Now, you no longer have to use Tableau Desktop to extract your data sources. For more information, see Create Extracts on the Web.

Logical and physical table extracts

With the introduction of logical tables and physical tables in the Tableau data model in version 2020.2, extract storage options have changed from Single Table and Multiple Tables, to

Logical Tables and Physical Tables. These options better describe how extracts will be stored. For more information, see **Decide how the extract data should be stored** on page 1022.

Beginning with version 10.5, when you create a new extract it uses the .hyper format. Extracts in the .hyper format take advantage of the improved data engine, which supports faster analytical and query performance for larger data sets.

Similarly, when an extract-related task is performed on a .tde extract using version 10.5 and later, the extract is upgraded to a .hyper extract. After a .tde extract is upgraded to a .hyper extract, it can't be reverted back to .tde extract. For more information, see **Extract Upgrade to .hyper Format** on page 1031.

Changes to values and marks in the view

To improve extract efficiency and scalability, values in extracts can be computed differently in versions 10.5 and later compared to versions 10.4 and earlier. Changes to how the values are computed can affect the way marks in your view are populated. In some rare cases, the changes can cause your view to change shape or become blank. These changes can also apply to multi-connection data sources, data sources that use live connections to filed-based data, data sources that connect to Google Sheets data, cloud-based data sources, extract-only data sources, and WDC data sources.

To get an idea of some of the differences you might see in your view using version 2021.2, see the sections below.

Format of date and date time values

In versions 10.5 and later, extracts are subject to more consistent and stricter rules around how date strings are interpreted through the DATE, DATETIME, and DATEPARSE functions. This affects how dates are parsed, or the date formats and patterns that are allowed for these functions. More specifically, the rules can be generalized as the following:

1. Dates are evaluated and then parsed by column, not by row.
2. Dates are evaluated and then parsed based on the locale of where the workbook was created, not on locale of the computer where the workbook is opened.

These new rules allow extracts to be more efficient and to produce results that are consistent with commercial databases.

However, because of these rules, particularly in international scenarios where the workbook is created in a locale different from the locale that the workbook is opened in or the server that the workbook is published to, you might see that 1.) date and datetime values change to different date and datetime values or 2.) date and datetime values change to *Null*. When your date and datetime values change to different date and datetime values or become *Null*, it's often an indication that there are issues with the underlying data.

Here are some common reasons why you might see changes to your date and datetime values in your extract data source using version 10.5 and later.

Common causes of changes to date/datetime values	Common causes of null values
• When a function has to parse multiple date formats in a single column. Where the date is ambiguous and can be interpreted in several different ways, the date will be interpreted based on the format Tableau has determined for that column. For some examples, see **Date scenario 1** on the facing page and **Date scenario 2** on page 1018 below. • When a function has to parse a YYYY-MM-DD (ISO) format. For an example, see **Date scenario 3** on page 1019. • When a function doesn't have enough information to derive the time, it can interpret a value as "00:00:00.0", using "0" for hour, minute, second, and millisecond. • When a function doesn't have enough information to derive the day, it can interpret a value as "1" or "January" for month. • When a function parses years, it is interpreted as the following: • Year "07" is interpreted	• When a function has to parse multiple date formats in a single column. After Tableau determines the date format, all other dates in the column that deviate from the format become null values. For some examples, see **Date scenario 1** on the facing page and **Date scenario 2** on page 1018 below. • When a function has to parse a YYYY-MM-DD (ISO) format. Values that exceed what is allowed for "YYYY," or "MM," or "DD" cause null values. For an example, see **Date scenario 3** on page 1019. • When a function has to parse date values that contain trailing characters. For example, time zone and daylight savings suffixes and keywords, such as "midnight" cause null values.

Common causes of changes to date/datetime values	Common causes of null values
as "2007" • Year "17" is interpreted as "2017." • Year "30" is interpreted as "2030." • Year "69" interpreted as "2069." • Year "70" is interpreted as "1970."	• When a function has to parse an invalid date or time. For example, 32/3/2012 causes a null value. In another example, 25:01:61 causes a null value. • When a function has to parse contradicting inputs. For example, suppose the pattern is 'dd.MM (MMMM) y' and the input string is '1.09 (August) 2017', where both "9" and "August" are months. The result is a null value because the month values are not the same. • When a function has to parse contradicting patterns. For example, a pattern that specifies a mix of Gregorian year (y) and ISO week (ww) causes null values.

Date scenario 1

Suppose you have a workbook created in an English locale that uses .tde extract data source. The table below shows a column of string data contained in the extract data source.

10/31/2018
31/10/2018

12/10/2018

Based on the particular English locale, the format of the date column was determined to follow the MDY (month, day, and year) format. The following tables show what Tableau displays based on this locale when the DATE function is used to convert string values into date values.

October 31, 2018
October 31, 2018
December 10, 2018

If the extract is opened in a German locale, you see the following:

31 Oktober 2018
31 Oktober 2018
12 Oktober 2018

However, after the extract is opened in a German locale using version 10.5 and later, the DMY (day, month, and year) format of the German locale is strictly enforced and causes a *Null* value because one of the values doesn't follow DMY format.

Null
October 31, 2018
October 12, 2018

Date scenario 2

Suppose you have another workbook created in an English locale that uses a .tde extract data source. The table below shows a column of numeric date data contained in the extract data source.

1112018
1212018
1312018
1412018

Based on the particular English locale, the format of the date column was determined to follow the MDY (month, day, and year) format. The following tables show what Tableau displays based on this locale when the DATE function is used to convert the numeric values into date values.

11/1/2018
12/1/2018
Null
Null

Date scenario 3

Suppose you have a workbook that uses a .tde extract data source. The table below shows a column of string data contained in the extract data source.

2018-10-31
2018-31-10
2018-12-10
2018-10-12

Because the date uses the ISO format, the date column always follows the YYYY-MM-DD format. The following tables show what Tableau displays when the DATE function is used to convert string values into date values.

October 10, 2018
Null

December 10, 2018
October 12, 2018

Note: In versions 10.4 (and earlier), ISO format and other date formats could have produced differing results depending on the locale of where the workbook was created. In an English locale for example, both 2018-12-10 and 2018/12/10 could produce December 12, 2o18. However, in a German locale 2018-12-10 could produce December 12, 2018 and 2018/12/10 could produce October 12, 2018.

Sort order and case sensitivity

Extracts have collation support and therefore can more appropriately sort string values that have accents or are cased differently.

For example, suppose you have a table of string values. In terms of sort order, this means that a string value like Égypte is now appropriately listed after Estonie and before Fidji.

About Excel data:

With regard to casing, this means that how Tableau stores values have changed between version 10.4 (and earlier) and version 10.5 (and later). However, the rules for sorting and comparing values haven't. In version 10.4 (and earlier), string values like "House," "HOUSE," and "houSe" are treated the same and stored with one representative value. In version 10.5 (and later), the same string values are considered unique and therefore stored as individual values. For more information, see **Changes to the way values are computed** on page 397.

Breaking ties in Top N queries

When a Top N query in your extract produces duplicate values for a specific position in a rank, the position that breaks the tie can be different when using version 10.5 and later. For example, suppose you create a top 3 filter. Positions 3, 4, and 5 have the same values. When using version 10.4 and earlier, the top filter can return 1, 2, and 3 positions. However, when using version 10.5 and later, the top filter can return 1, 2, and 5 positions.

Precision of floating-point values

Extracts are better at taking advantage of the available hardware resources on a computer and therefore able to perform mathematical operations in a highly parallel way. Because of this, real numbers can be aggregated by .hyper extracts in different order. When numbers are aggregated in different order, you might see different values in your view after the decimal point each time the aggregation is computed. This is because floating-point addition and multiplication is not necessarily associative. That is, (a + b) + c is not necessarily the same as a + (b + c). Also, real numbers can be aggregated in different order because floating-point multiplication is not necessarily distributive. That is, (a x b) x c is not necessarily the same as a x b x c. This type of floating-point rounding behavior in .hyper extracts resemble that of floating-point rounding behavior in commercial databases.

For example, suppose your workbook contains a slider filter on an aggregated field comprised of floating point values. Because the precision of floating-point values have changed, the filter might now exclude a mark that defines the upper or lower bound of the filter range. The absence of these numbers could cause a blank view. To resolve this issue, move the slider on the filter or remove and add the filter again.

Accuracy of aggregations

Extracts optimize for large data sets by taking better advantage of the available hardware resources on a computer and therefore able to compute aggregations in a highly parallel way. Because of this, aggregations performed by .hyper extracts can resemble the results from commercial databases more than the results from software that specializes in statistical computations. If you’re working with a small data set or need a higher level of accuracy, consider performing aggregations through reference lines, summary card statistics, or table calculation functions like variance, standard deviation, correlation, or covariance.

About the Compute Calculations Now option for extracts

If the **Compute Calculations Now** option was used in a .tde extract using an earlier version of Tableau Desktop, certain calculated fields were materialized and therefore computed in advance and stored in the extract. If you upgrade the extract from a .tde extract to a .hyper extract, the previously materialized calculations in your extract are not included. You must use the **Compute Calculations Now** option again to ensure that materialized calculations are a part of the extract after the extract upgrade. For more information, see **Materialize Calculations in Your Extracts** on page 1047.

New Extract API

You can use the Extract API 2.0 to create .hyper extracts. For tasks that you previously performed using the Tableau SDK, such as publishing extracts, you can use the Tableau Server REST API or the Tableau Server Client (Python) library. For refresh tasks, you can use the Tableau Server REST API as well. For more information, see **Tableau Hyper API** on page 1059.

Create an extract

Though there are several options in your Tableau workflow for creating an extract, the primary method is described below.

1. After you connect to your data and set up the data source on the Data Source page, in the upper-right corner, select **Extract**, and then click the **Edit** link to open the Extract Data dialog box.

2. (Optional) Configure one or more of the following options to tell Tableau how to store, define filters for, and limit the amount of data in your extract:

   - **Decide how the extract data should be stored**

     You can choose to have Tableau store the data in your extract using one of two structures (schemas): logical tables (denormalized schema) or physical tables (normalized schema). For more information about logical and physical tables, see **The Tableau Data Model** on page 810.

     The option you choose depends on what you need.

     - **Logical Tables**

       Stores data using one extract table for each logical table in the data source. Physical tables that define a logical table are merged and stored with that logical table. For example, if a data source was made of a single logical table, the data would be stored in a single table. If a data source was made of three logical tables (each containing multiple physical tables), the extract data would be stored in three tables—one for each logical table.

       Select **Logical Tables** when you want to limit the amount of data in your extract with additional extract properties like extract filters, aggregation, Top N, or other features that require denormalized data. Also use when your data uses pass-through functions (RAWSQL). This is the default structure Tableau uses to store extract data. If you use this option when your extract contains joins, the joins are applied when the extract is created.

- **Physical Tables**

  Stores data using one extract table for each physical table in the data source.

  Select **Physical Tables** if your extract is comprised of tables combined with one or more equality joins and meets the **Conditions for using the Physical Tables option** below listed below. If you use this option, joins are performed at query time.

  This option can potentially improve performance and help reduce the size of the extract file. For more information about how Tableau recommends you use the Physical Tables option, see **Tips for using the Physical Tables option** on page 1030. In some cases, you can also use this option as a workaround for row-level security. For more information about row-level security using Tableau, see **Restrict Access at the Data Row Level** on page 3088.

  ### Conditions for using the Physical Tables option

  To store your extract using the Physical Tables option, the data in your extract must meet all of the conditions listed below.

  - All joins between physical tables are equality (=) joins
  - Data types of the columns used for relationships or joins are identical
  - No pass-through functions (RAWSQL) used
  - No incremental refresh configured
  - No extract filters configured
  - No Top N or sampling configured

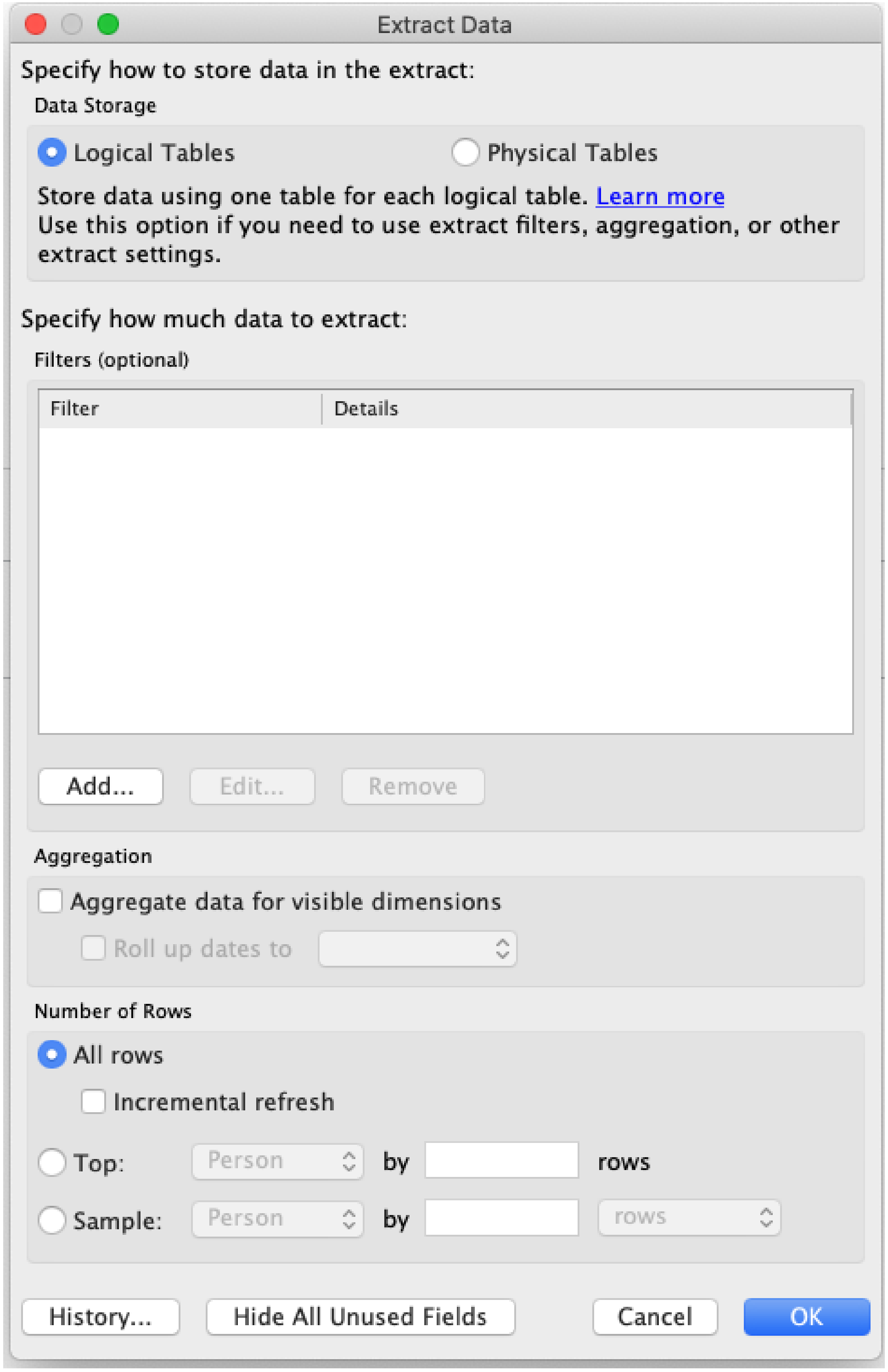
  When the extract is stored as physical tables, you cannot append data to it. For logical tables, you can't append data to extracts that have more than one logical table.

Note: Both the Logical Tables and Physical Tables options only affect how the data in your extract is stored. The options do not affect how tables in your extract are displayed on the Data Source page.

For example, suppose your extract is comprised of one logical table that contains three physical tables. If you directly open the extract (.hyper) file that has been configured to use the default option, Logical Tables, you see one table listed on the Data Source page. However, if you open the extract using the packaged data source (.tdsx) file or the data source (.tdsx) file with its corresponding extract (.hyper) file, you see all three tables that comprise the extract on the Data Source page.

- **Determine how much data to extract**

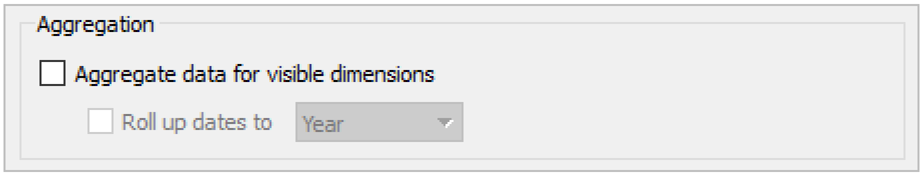
  Click **Add** to define one or more filters to limit how much data gets extracted based on fields and their values.



- **Aggregate the data in the extract**

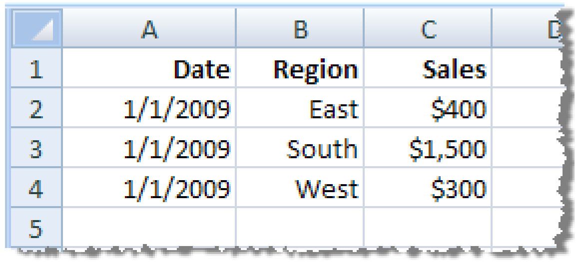
Select **Aggregate data for visible dimensions** to aggregate the measures using their default aggregation. Aggregating the data consolidates rows, can minimize the size of the extract file, and increase performance.

When you choose to aggregate the data, you can also select **Roll up dates** to a specified date level such as Year, Month, etc. The examples below show how the data will be extracted for each aggregation option you can choose.



<table>
<tr>
<td>Original data</td>
<td>
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th></tr>
<tr><td>1</td><td>Date</td><td>Region</td><td>Sales</td></tr>
<tr><td>2</td><td>1/1/2009</td><td>South</td><td>$500</td></tr>
<tr><td>3</td><td>1/1/2009</td><td>West</td><td>$200</td></tr>
<tr><td>4</td><td>1/1/2009</td><td>West</td><td>$100</td></tr>
<tr><td>5</td><td>1/1/2009</td><td>East</td><td>$300</td></tr>
<tr><td>6</td><td>1/2/2009</td><td>South</td><td>$600</td></tr>
<tr><td>7</td><td>1/2/2009</td><td>South</td><td>$400</td></tr>
<tr><td>8</td><td>1/2/2009</td><td>East</td><td>$100</td></tr>
<tr><td>9</td><td></td><td></td><td></td></tr>
</table>
</td>
<td>Each record is shown as a separate row. There are seven rows in your data.</td>
</tr>
<tr>
<td>Aggregate data for visible dimensions

(no roll up)</td>
<td>
<table>
<tr><th></th><th>A</th><th>B</th><th>C</th></tr>
<tr><td>1</td><td>Date</td><td>Region</td><td>Sales</td></tr>
<tr><td>2</td><td>1/1/2009</td><td>East</td><td>$300</td></tr>
<tr><td>3</td><td>1/1/2009</td><td>South</td><td>$500</td></tr>
<tr><td>4</td><td>1/1/2009</td><td>West</td><td>$300</td></tr>
<tr><td>5</td><td>1/2/2009</td><td>East</td><td>$100</td></tr>
<tr><td>6</td><td>1/2/2009</td><td>South</td><td>$1,000</td></tr>
<tr><td>7</td><td></td><td></td><td></td></tr>
</table>
</td>
<td>Records with the same date and region have been aggregated into a single row. There are five rows in the extract.</td>
</tr>
</table>

<table>
<tr>
<td>Aggregate data for visible dimensions
(roll up dates to Month)</td>
<td>
<table>
<tr><td></td><td>A</td><td>B</td><td>C</td><td>D</td></tr>
<tr><td>1</td><td>Date</td><td>Region</td><td>Sales</td><td></td></tr>
<tr><td>2</td><td>1/1/2009</td><td>East</td><td>$400</td><td></td></tr>
<tr><td>3</td><td>1/1/2009</td><td>South</td><td>$1,500</td><td></td></tr>
<tr><td>4</td><td>1/1/2009</td><td>West</td><td>$300</td><td></td></tr>
<tr><td>5</td><td></td><td></td><td></td><td></td></tr>
</table>
</td>
<td>Dates have been rolled up to the Month level and records with the same region have been aggregated into a single row. There are three rows in the extract.</td>
</tr>
</table>

- **Choose the rows to extract**

  Select the number of rows you want to extract.

  You can extract **All rows** or the **Top** *N* rows. Tableau first applies any filters and aggregation and then extracts the number of rows from the filtered and aggregated results. The number of rows options depend on the type of data source you are extracting from.

  Notes:

  - Not all data sources support sampling. Therefore, you might not see the **Sampling** option in the Extract Data dialog box.
  - Any fields that you hide first in the Data Source page or on the sheet tab will be excluded from the extract. Click the **Hide All Unused Fields** button to remove these hidden fields from the extract.

3. When finished, click **OK**.
4. Click the sheet tab. Clicking the sheet tab initiates the creating of the extract.
5. In the subsequent dialog box, select a location to save the extract, give the extract file a name, and then click **Save**.

   If the Save dialog box does not display, see the **Troubleshoot extracts** on page 1031 section, below.

General tips for working with extracts

Save your workbook to preserve the connection to the extract

After you create an extract, the workbook begins to use the extract version of your data. However, the connection to the extract version of your data is not preserved until you save the workbook. This means if you close the workbook without saving the workbook first, the workbook will connect to the original data source the next time you open it.

Toggle between sampled data and entire extract

When you're working with a large extract, you might want to create an extract with a sample of the data so you can set up the view while avoiding long queries every time you place a field on a shelf on the sheet tab. You can then toggle between using the extract (with sample data) and using the entire data source by selecting a data source on the **Data** menu and then selecting **Use Extract**.

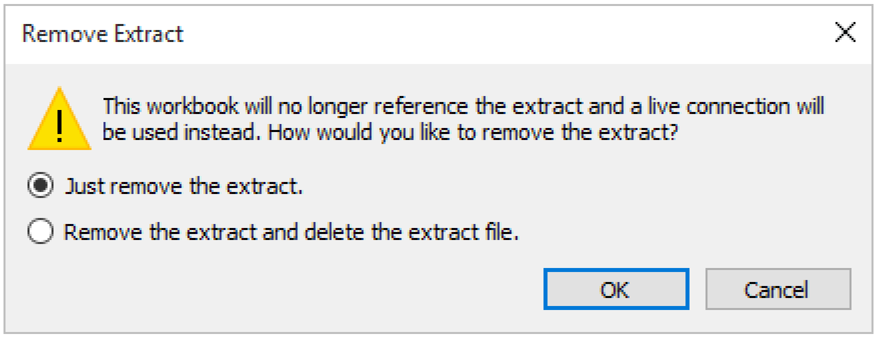
Don't connect directly to the extract

Because extracts are saved to your file system, it is possible to connect directly to them with a new Tableau Desktop instance. This is not recommended for a few reasons:

- **The table names will be different**. Tables stored in your extract use special naming to guarantee name uniqueness, and it may not be human-readable.
- **You cannot refresh the extract**. When connecting directly to an extract, Tableau treats that file as the true source, as opposed to a clone of underlying data. So, it's not possible to relate it back to your source data.
- **The data model and relationships will be lost**. The data model and relationships between the tables is stored in the .tds file and not in the .hyper file, so this information is lost when connecting directly to the .hyper file. Additionally, if you extract using logical tables storage, you will not see any references to the original underlying physical tables.

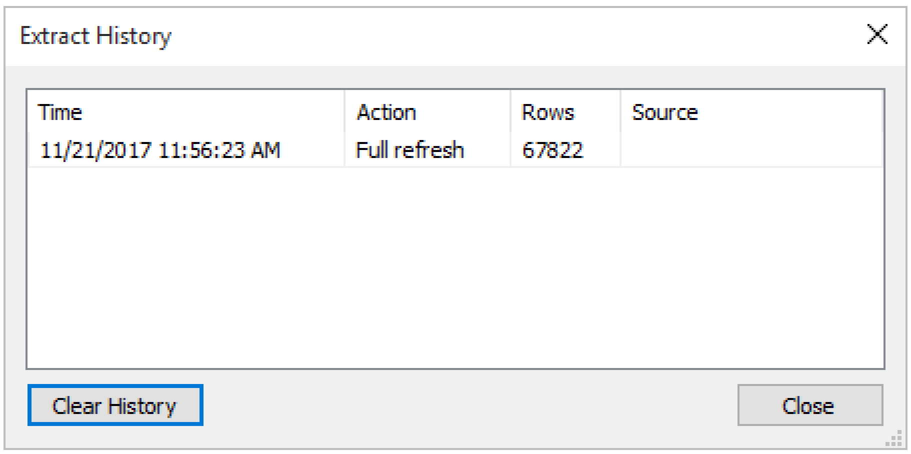
Remove the extract from the workbook

You can remove an extract at anytime by selecting the extract data source on the **Data** menu and then selecting **Extract** > **Remove**. When you remove an extract, you can choose to **Remove the extract from the workbook only** or **Remove and delete the extract file**. The latter option will delete the extract from your hard drive.



See extract history

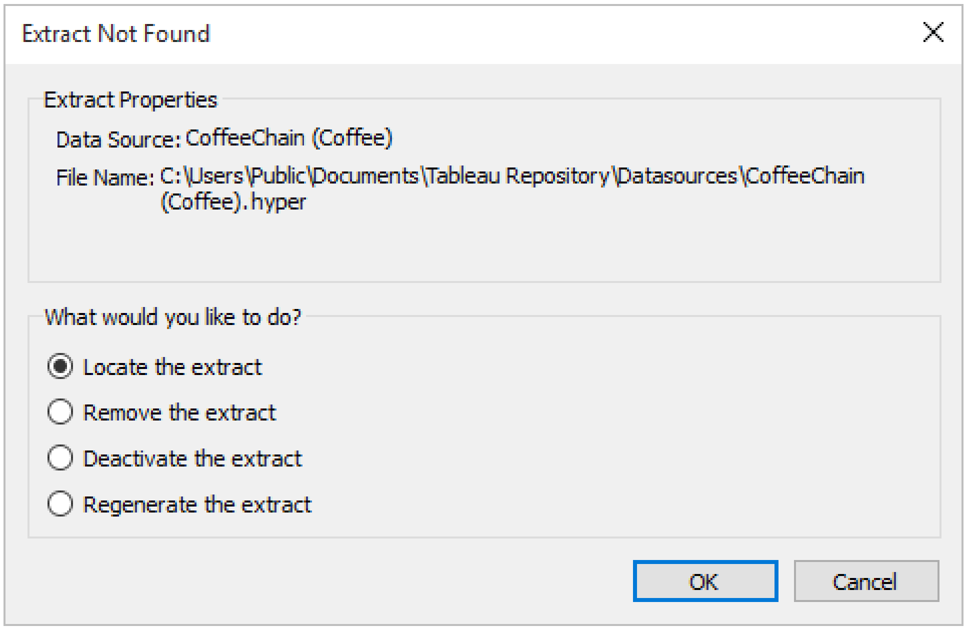
You can see when the extract was last updated and other details by selecting a data source on the **Data** menu and then selecting **Extract** > **History**.



If you open a workbook that is saved with an extract and Tableau cannot locate the extract, select one of the following options in the Extract Not Found dialog box when prompted:

- **Locate the extract:** Select this option if the extract exists but not in the location where Tableau originally saved it. Click **OK** to open an Open File dialog box where you can specify the new location for the extract file.
- **Remove the extract:** Select this option if you have no further need for the extract. This is equivalent to closing the data source. All open worksheets that reference the data source are deleted.
- **Deactivate the extract:** Use the original data source from which the extract was created, instead of the extract.

- **Regenerate the extract:** Recreates the extract. All filters and other customizations you specified when you originally created the extract are automatically applied.



Tips for using the Physical Tables option

Tableau generally recommends that you use the default data storage option, Logical Tables, when setting up and working with extracts. In many cases, some of the features you need for your extract, like extract filters, are only available to you if you use the Logical Tables option.

Physical Tables option for extracts that are larger than expected

The Physical Tables option should be used sparingly to help with specific situations such as when your data source meets the **Conditions for using the Physical Tables option** on page 1023 and the size of your extract is larger than expected. To determine if the extract is larger than it should be, the sum of rows in the extract using the Logical Tables option must be higher than the sum of rows of all the combined tables before the extract has been created. If you encounter this scenario, try using the Physical Tables option instead.

Alternative filtering suggestions when using the Physical Tables option

When using the Physical Tables option, other options to help reduce the data in your extract, like extract filters, aggregation, Top N and Sampling are disabled. If you need to reduce the data in an extract that uses the Physical Tables option, consider filtering the data before it is brought into Tableau Desktop using one of the following suggestions:

- **Connect to your data and define filters using custom SQL:** Instead of connecting to a database table, connect to your data using custom SQL instead. When creating your custom SQL query, make sure that it contains the appropriate level of filtering that you need to reduce the data in your extract. For more information about custom SQL in Tableau Desktop, see **Connect to a Custom SQL Query** on page 959.
- **Define a view in the database:** If you have write access to your database, consider defining a database view that contains just the data you need for your extract and then connect to the database view from Tableau Desktop.

Row-level security with extracts

If you want to secure extract data at the row level, using the Physical Tables option is the recommended way to achieve this scenario. For more information about row-level security in Tableau, see **Restrict Access at the Data Row Level** on page 3088.

Troubleshoot extracts

- **Creating an extract takes a long time:** Depending on the size of your data set, creating an extract can take a long time. However, after you have extracted the data and saved it to your computer, performance can improve.
- **Extract is not created:** If your data set contains a really large number of columns (e.g., in the thousands), in some cases Tableau might not be able to create the extract. If you encounter problems, consider extracting fewer columns or restructuring the underlying data.
- **Save dialog does not display *or* extract is not created from a .twbx:** If you follow the above procedure to extract data from a packaged workbook, the Save dialog does not display. When an extract is created from a packaged workbook (.twbx), the extract file is automatically stored in the package of files associated with the packaged workbook. To access the extract file that you created from the packaged workbook, you must unpackage the workbook. For more information, see **Packaged Workbooks** on page 3022.

Extract Upgrade to .hyper Format

New .hyper extract format

Beginning in version 10.5, when you create a new extract, it uses the .hyper format instead of the .tde format. Extracts in the .hyper format take advantage of the improved data engine, which supports the same fast analytical and query performance as the data engine before it, but for even larger extracts.

Although there are many benefits of using .hyper extracts, the primary benefits include the following:

- **Create larger extracts:** You can create extracts with billions of rows of data. Because .hyper extracts can support more data, you can consolidate .tde extracts that you previously had to create separately into a single .hyper extract.
- **Create and refresh extracts faster:** While Tableau has always optimized performance for creating and refreshing extracts, version 2021.2 supports faster extract creation and refreshes for even larger data sets.
- **Experience better performance when interacting with views that use extract data sources:** Although smaller extracts continue to perform efficiently, larger extracts perform more efficiently.

What causes an extract upgrade?

Although you can continue to open and interact with .tde extracts in version 2021.2, when you perform an extract task on a .tde extract, the .tde extract is upgraded to a .hyper extract. After the extract is upgraded, you should be aware of the following backward limitations:

- You can't convert the upgraded extract back to a .tde extract.
- You can't open the upgraded extract in an earlier version of Tableau Desktop. This might affect how you and other users might interact with .tde extracts.
- You can't use the **Export as Version** in Tableau Desktop to downgrade a workbook that contains a .hyper extract.
- You can't publish the extract as an older version from Tableau Desktop to Tableau Server.
- You can't open a workbook in Tableau Desktop 10.4 and earlier that is downloaded from Tableau Server or Tableau Online using the **Download Tableau Workbook** option if the workbook contains a .hyper extract.

Tasks that cause an extract upgrade

There are three distinct ways a .tde extract can get upgraded to a .hyper extract: 1.) during an extract refresh (full or incremental), 2.) when appending data to an extract, and 3.) when an extract is upgraded manually. After an extract has been upgraded using Tableau Desktop, it's not automatically removed in case the .tde extract is being referenced by other workbooks.

In addition to the tasks that upgrade an extract, extract tasks performed outside of Tableau Desktop can cause an extract upgrade to the .tde extract you interact with. This includes the following tasks:

- Manual refresh (Tableau Server and Tableau Online)
- Incremental refresh on a published extract data source from Tableau Bridge (Tableau Online)
- A scheduled full or incremental extract refresh data source (Tableau Server and Tableau Online)
- A scheduled incremental extract refresh from Tableau Bridge (Tableau Online)

Impact of extract upgrade

When you're working in an environment where the version number of Tableau Desktop doesn't match the Tableau Desktop version that others are using, or doesn't match the version of Tableau Server or Tableau Online, you might encounter some extract compatibility issues depending on the task you want to perform.

Review a detailed explanation of the extract compatibility scenarios on the Tableau Support page or a summary explanation below to better understand when extract upgrades can occur and potential compatibility issues you or others might experience.

Note: In the tables below, "10.4" represents Tableau 10.4 and earlier and "10.5" represents Tableau 10.5 and later.

Tasks using Tableau Desktop

You should be aware of the following extract-related compatibility scenarios around tasks performed on different versions of Tableau Desktop.

Tasks		10.4 workbook	10.5 workbook	
		.tde extract	.tde extract	.hyper extract
Create new	in 10.4	√	Not possible	Not possible
	in 10.5	Not possible	Not possible	√
Open	in 10.4	√	Can't open workbook; you're asked to open the workbook in 10.5 or later	Can't open workbook; you're asked to open the workbook in 10.5 or later
	in 10.5	√	√	√
Refresh or append	in 10.4	√	Not possible	Not possible
	in 10.5	Workbook version remains unchanged, but extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√
Publish workbook	from 10.5 to Tableau Server 10.4	√	Can't publish; in the Publish dialog, you see a "server is running an older version than your copy of Tableau Desktop" message, then you see a	Can't publish; in the Publish dialog, you see a "server is running an older version than your copy of Tableau Desktop" message, then you see a

Tasks		10.4 workbook	10.5 workbook	
		.tde extract	.tde extract	.hyper extract
			"workbook cannot be downgraded" error message	"workbook cannot be downgraded" error message
	from 10.5 to Tableau Server 10.5 or Tableau Online	√	√	√
	from 10.4 to Tableau Server 10.5 or Tableau Online	√		
Publish extract data source	from 10.5 to Tableau Server 10.4	√	Can't publish; you see a "cannot publish, make sure you are connected to a compatible version" error message	Can't publish; you see a "cannot publish, make sure you are connected to a compatible version" error message

Tasks		10.4 workbook	10.5 workbook	
		.tde extract	.tde extract	.hyper extract
	from 10.5 to Tableau Server 10.5 or Tableau Online	√	√	√
Connect to a published data source	from 10.4 to Tableau Server 10.5 or Tableau Online	√	√	√
	from 10.5 to Tableau Server 10.5 or Tableau Online	√	√	√

Tasks on Tableau Server 10.5 or Tableau Online

When working with extracts created in an earlier version of Tableau Desktop, you should be aware of the following extract-related compatibility scenarios around tasks performed on Tableau Server 2021.2 or Tableau Online.

Scenario		**10.4 workbook**	**10.5 workbook**	
		.tde extract	**.tde extract**	**.hyper extract**
Edit/save in web authoring	on Tableau Server 10.4	√	Not possible	Not possible
	on Tableau Server 10.5 or Tableau Online	Workbook changes to version 10.5, extract remains in .tde format	√	√
Edit/save in web authoring then refresh or append	on Tableau Server 10.4	√	√	√
	on Tableau Server 10.5 or Tableau Online	Workbook changes to version 10.5, and extract upgrades to .hyper format	Workbook version remains unchanged, but extract upgrades to .hyper format	√

Scenario		10.4 workbook	10.5 workbook	
		.tde extract	.tde extract	.hyper extract
Download and open	in Tableau Desktop 10.4	√	Can't open workbook; you see a "this workbook uses a .hyper extract and is not compatible with this version; open the workbook in version 10.5 or later" error message, and then asked to locate the extract	Can't open workbook; you see a "this file was created by a newer version; upgrade Tableau" error message
	in Tableau Desktop 10.5	√	√	√
Export As Version	from Tableau Desktop 10.5	√	Option is greyed out	Option is greyed out

Automating extract creation, refresh, and append tasks

When working with extracts created in earlier versions of Tableau, you should be aware of the following extract-related compatibility scenarios when automating extract creation, refresh, and append tasks.

Scenario		.tde extract	.hyper extract
Create new	using Tableau SDK	√	Not possible

Scenario		.tde extract	.hyper extract
extract			
	using Extract API 2.0	Not possible	√
Refresh or append	using 10.4 Tableau Command-Line utility	√	Not possible
	using 10.5 Tableau Command-Line utility	Workbook version remains unchanged, but extract upgrades to .hyper format	√

Why keep an extract in .tde format?

If you cannot upgrade Tableau Desktop, you'll need to keep your extract in .tde format.

How to keep an extract in .tde format

To keep an extract in the .tde format, do not upgrade the extract. To avoid upgrading the extract, do not perform any of the tasks listed in the **Tasks that cause an extract upgrade** on page 1032 section to a .tde extract using Tableau Desktop 2021.2. Then, consider the following suggestions if you need to maintain a .tde version of an extract:

- Perform extract refresh and append data tasks using Tableau Desktop 2021.2, but maintain an earlier version of Tableau Desktop and the ability to connect to the original data in case you need to recreate the .tde extract.
- Use an earlier version of Tableau Desktop to perform extract tasks, such as extract refresh or append data.
- If possible, disable existing extract refresh schedules on Tableau Server, Tableau Online, or Tableau Bridge until you can identify which extracts should and shouldn't be upgraded.

What to expect after an extract upgrade

After your extract has been upgraded, you can expect some additional changes when working in version 2021.2.

Changes to opening extracts

After a .tde extract is upgraded to a .hyper extract, you can't open an upgraded extract using an earlier version of Tableau Desktop. Consider some of the suggestions below to help resolve issues you have with opening extracts.

- **Upgrade Tableau Desktop to 2021.2:** If possible, consider upgrading Tableau Desktop to 2021.2 so that you can experience the benefits of .hyper extracts. For more information, see Upgrade Tableau Desktop in the Tableau Desktop Deployment Guide.
- **Recreate the extract using an earlier version of Tableau Desktop:** To recreate your .tde extract, you must be able to connect to the original data using an earlier version of Tableau Desktop. For more information, see **Connect to and Prepare Data** on page 391 and Downloading Previous Versions of Tableau in the Tableau Knowledge Base.
- **Retrieve earlier versions of a published extract data source or workbook:** If revision history is enabled on Tableau Server or Tableau Online, you can download an earlier version of a published extract data source or a published workbook that uses an extract data source. **Note:** Earlier workbooks that you retrieve that use an extract data source are saved in a .twb format and therefore will require you to recreate the extract after you open the workbook in an earlier version of Tableau Desktop. For more information, see **Work with Content Revisions** on page 3217.

Changes to sharing extracts

When you need to share the data in the extract data source with someone directly (for example, through email or a file share) be sure to save the extract as a .tdsx file. Alternatively, you can save the workbook with the extract data as a .twbx. If you only share the .hyper file instead of the .tdsx or .twbx, metadata about the .hyper extract like connection information and column name changes will be missing. For more information, see **Tableau File Types and Folders** on page 275.

Potential differences in extract file size

While .tde extract file sizes can vary depending on complexity of your data, after an extract upgrade, corresponding .hyper extract file sizes can also vary for the same reason. There may be some cases when some .hyper extracts become bigger after an extract upgrade and other cases when .hyper extracts become smaller after an extract upgrade.

Changes to the Export as Version option

When your workbook contains a .hyper extract, the **Export as Version** option in the File menu is not available. If you need to open a workbook that contains a .hyper extract using an earlier version of Tableau, you must first remove the extract from the workbook before you can downgrade it using the **Export as Version** option. After you've downgraded the workbook, you must recreate the .tde extract.

Other issues

For errors that you see after an extract upgrade, refer to the following articles in the Tableau Knowledge Base:

- Error when opening .hyper extract in Tableau Desktop 10.4 or earlier
- "Export as Version" grayed out for workbooks with .hyper extract
- Error "The workbook cannot be downgraded because it contains a .hyper extract" when publishing to Tableau Server 10.4 and earlier

Changes to your view after an extract upgrade

After an extract has been upgraded, you might notice some changes to a view that uses the extract data source. This is because values in your extract can be computed differently in version 2021.2. Changes can include fewer marks in the view, more null values when you inspect the summary data, or the view itself has a different shape. In some rare cases, the view might even be blank. For more information about the type of changes you might see when interacting with extracts with this version of Tableau, see **Changes to values and marks in the view** on page 1015.

Manually upgrade your extract

If you manage extracts locally, you can manually upgrade your .tde extract to a .hyper extract using Tableau Desktop.

1. In Tableau Desktop, open a workbook that uses a .tde extract.
2. Select the extract data source from the **Data** menu and then select **Extract** > **Upgrade**
3. Select **File** > **Save**, which saves the workbook and also completes the extract upgrade.

Refresh Extracts

When the original data changes, you can refresh the extract using Desktop by selecting a data source on the **Data** menu and then selecting **Extract** > **Refresh**. Extracts can be configured to be fully refreshed, replacing all of the data with what's in the original data source, or incrementally refreshed, adding just the new rows since the previous refresh.

Note: You can also automate extract refreshes using the Tableau Data Extract Command Line Utility. For more information, see **Tableau Data Extract Command-Line Utility** on page 1049.

Before you refresh extracts

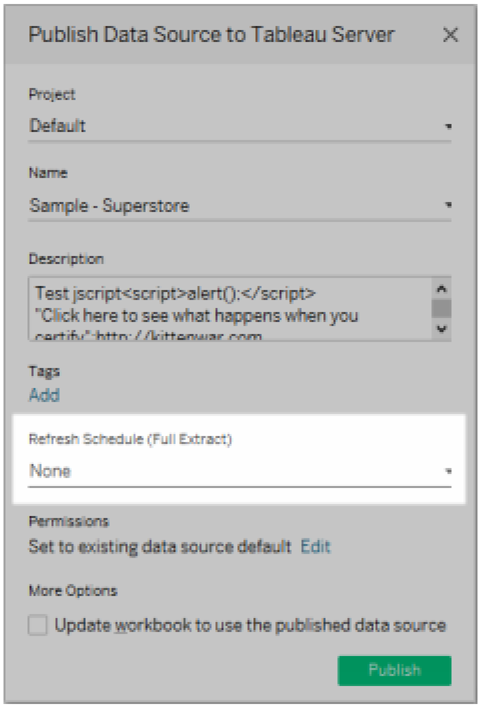
If you want to refresh an extract, make note of the file format of the extract before you perform an extract refresh. If you perform a refresh on an .tde extract using version 2021.2, the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop. For more information, see **Extract Upgrade to .hyper Format** on page 1031.

Configure a full extract refresh

By default, extracts are configured to fully refresh. This means that every time you refresh the extract, all of the rows are replaced with the data in the original data source. While this kind of refresh ensures that you have an exact copy of what is in the original data, depending on the size of the extract, a full refresh can sometimes take a long time and be expensive on the database. If an extract is not configured for an incremental refresh, selecting to refresh the extract will perform a full refresh of the extract.

Publish to Tableau Server

If you plan to publish the extract as a data source to Tableau Server, you can specify a schedule for the extract refresh during publishing.



Similarly, if you are publishing the extract in a workbook to Tableau Server, you can also specify a schedule for the extract refresh during publishing. For more information, see **Schedule Extract Refreshes as You Publish a Workbook** on page 3086.

Publish to Tableau Online

If you plan to publish the extract as a data source to Tableau Online, your options for refreshing your data depend on the characteristics of your data sources. For more information on refreshing data from specific data sources, see Extract refresh options by data source.

Configure an incremental extract refresh

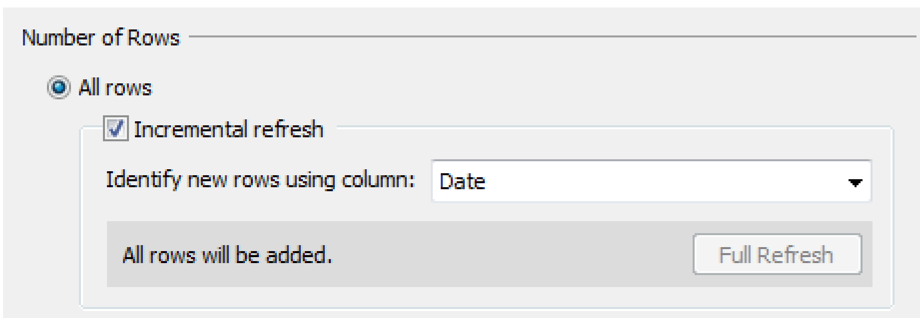
Most data sources support an incremental refresh. Rather than refreshing the entire extract, you can configure a refresh to add only the rows that are new since the previous time you extracted the data. For example, you may have a data source that is updated daily with new sales transactions. Rather than rebuild the entire extract each day, you can just add the new transactions that occurred that day. Then once a week you might want to do a full refresh just to be sure you have the most up to date data.

Note: If the data structure of the source data changes (for example, a new column is added), you will need to do a full extract refresh before you can start doing incremental refreshes again.

You can follow the steps below to set up an extract to be refreshed incrementally.

1. Select a data source on the **Data** menu and then select **Extract Data**.
2. In the Extract Data dialog box, select **All rows** as the number of Rows to extract. Incremental refresh can only be defined when you are extracting all rows in the database. You cannot increment a sample extract.
3. Select **Incremental refresh** and then specify a column in the database that will be used to identify new rows. For example, if you select a Date field, refreshing will add all rows whose date is after that last time you refreshed. Alternatively, you can use an ID column that increases as rows are added to the database.

   Note: A Full Refresh replaces all of the rows with the data in the original data source every time you refresh the extract. A Full Refresh can take longer and be expensive on the database.

   Note: The data engine, which is the underlying mechanism that Tableau uses to create extracts, stores time values with a precision of up to 3 decimal places. If you specify a datetime or timestamp column for **Identify new rows using column**, and your database uses a higher precision than Tableau, you can end up with duplicate rows after an incremental refresh. For example, if the database has two rows, one with a datetime value of `2015-03-13 17:30:56.502352` and one with a datetime value of `2015-03-13 17:30:56.502852`, Tableau will store both rows using a datetime value of `2015-03-13 17:30:56.502` thereby creating duplicate rows.

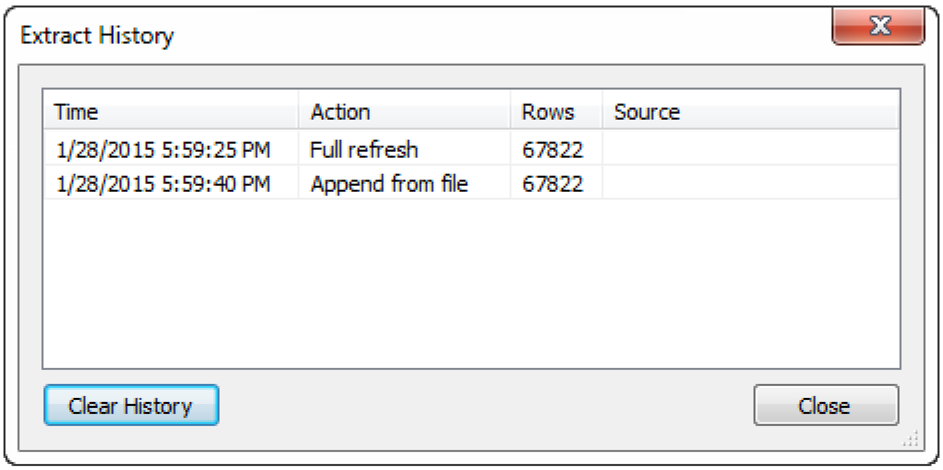
4. When finished, click **Extract**.

The steps above can be used to define a new extract or edit an existing extract for an incremental refresh. If you are editing an existing extract, the last refresh is shown so you can be sure you are updating the extract with the correct data.

See extract history

You can see a history of when the extract was refreshed by selecting a data source on the **Data** menu and then select **Extract** > **History**.

The Extract History dialog box shows the date and time for each refresh, whether it was full or incremental, and the number of rows that were added. If the refresh was from a file, it also shows the source file name.



Add Data to Extracts

There are two ways you can add new data to an extract: from a file or from a data source. However, to add new data, you must connect first connect to data and create a new data source, and the columns in the file or data source must match the columns in the extract.

Before you add data to extracts

If you want to add data to an extract, make note of the file format of the extract before you perform the task. If you add data to a .tde extract using version 2021.2, the extract is upgraded to .hyper extract. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract using previous versions of Tableau Desktop. For more information, see **Extract Upgrade to .hyper Format** on page 1031.

Add data from a file

You can add new data to an extract from a file-based data source. Use this option when the file type of the extract is the same as the file type of the data that you want to add. Alternatively, you can add data from an extract (.tde or .hyper) file. For example, you may have text files that are generated for a task that is performed every day. To add each day's worth of information to your extract whose original data source is also a text file, use the **Append Data from File** command.

1. On the **Data** menu, select a data source, and then select **Extract** > **Append Data from File**.
2. Browse to and select the file that has the new data.

   Note: By default, the file format of the extract's original data source is used. To add data from a Tableau data extract, click the file format drop-down list, and then select **Tableau Data Extract (*.tde)** or **Hyper Extracts (*.hyper)**.

3. When finished, click **OK**.

Add data from a data source

You can also add new data to an extract from another data source that's already open in the workbook. Use this option when the data from the data source that you want to add matches the extract data source before it was extracted. For example, you created an extract from a SQL Server database that has the past ten years worth of data. However, the maintained version of that data is stored on a different SQL Server database. You can add new data to the extract by using the **Append Data from Data Source** command.

Note: Joins or custom SQL should be specified in the data source before adding data to the extract.

1. On the **Data** menu, select a data source, and then select **Extract** > **Append Data from Data Source**.
2. In the dialog box, select the data source that you want to append.
3. When finished, click **OK**.

Using either option will add new rows to the extract. To see a summary of the number of rows that were added, select a data source on the **Data** menu and then select **Extract** > **History**.

Note: When you refresh this extract, the data will be replaced with the data from the original data.

Materialize Calculations in Your Extracts

You can use the **Compute Calculations Now** option to materialize calculations in your extract. When calculations are materialized, certain calculations are computed in advance and its values stored in the extract. Depending on the complexity of the calculations used in your extract, this can potentially speed up future queries by allowing Tableau to compute certain calculations in advance. Tableau can then use the precomputed value rather than computing the value each time a query against a calculation is made.

If the formula for a calculation that Tableau has already materialized changes or the calculation is deleted from the data source, the materialized calculation is removed from the extract until you use the **Compute Calculations Now** option again.

When to materialize calculations

You should not use the **Compute Calculations Now** option for all your extracts. Instead, try this option only when the query performance of your extract is slow as a result of complex calculations, such as string manipulations and regular expressions.

Materialize calculations

To materialize calculations in your extract, do the following:

1. In Tableau Desktop, select the extract data source from the **Data** menu.
2. Select **Extract** > **Compute Calculations Now**.

Materialize calculations in extracts already published to Tableau Online or Tableau Server

If you need to retroactively materialize calculations in extracts that are already published to Tableau Online or Tableau Server, you have the following options:

- You can download the extract, open the extract in Tableau Desktop, use the **Compute Calculations Now** option (as described above), and then republish the extract.
- As a site administrator or data source owner, you can use tabcmd to refresh the published extract with the `--addcalculations` option to materialize the calculations. For more information about the tabcmd command option, see refreshextracts in the Tableau Online Help or refreshextracts in the Tableau Server Help.

  Note: Materializing calculations can increase the size of the extract.

Calculations that cannot be materialized

The following calculations cannot be materialized:

- Calculations that use unstable functions such as NOW() and TODAY()
- Calculations that use external functions such as RAWSQL and R
- Table calculations
- Level of detail (LOD) calculations

Update Server Data Sources That Are Using Extracts

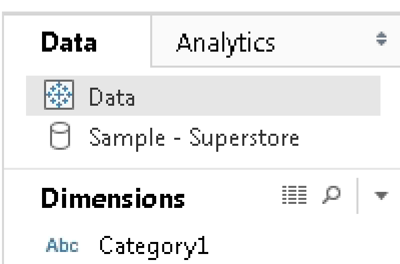
You have the following options for updating extract data sources that are published to Tableau Server or Tableau Online:

- **Option 1:** You can add the data source or a workbook that connects to it to a refresh schedule in Tableau Server or Tableau Online (cloud-based data sources only).
- **Option 2:** You can update the data source in Tableau Desktop and then republish it.
- **Option 3:** You can add to or refresh the data source on Tableau Server or Tableau Online, from Tableau Desktop, without first adding to or refreshing the extract in Tableau Desktop.

The remainder of this topic describes option 3.

Update Server or Online extracts from Tableau Desktop

Before you attempt to update an extract data source on Tableau Server or Tableau Online, verify first that Tableau Desktop is connected to a published data source. You can verify that Tableau Desktop is connected to a published data source by the Tableau Server icon next to the data source name in the Data pane:



Note: If you want to refresh an extract data source, make note of the file format of the extract before you perform an extract refresh. If you perform a refresh on an .tde extract using version 2021.2, the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of

Tableau Desktop. For more information, see **Extract Upgrade to .hyper Format** on page 1031**Latest changes to extracts** on page 1014

To update the published data source, do the following:

1. In Tableau Desktop, right-click (control-click on a Mac) the data source in the Data pane
2. Select **Tableau Data Server**, and choose one of the following options:

- **Edit Server and Site Path**

  If you have changed the location of the data source in Tableau Server or Tableau Online, choose this option to point Tableau Desktop to the new, correct location.

- **Refresh from Source**

  Refreshes the extract (full or incremental) using the original data.

  This command is available only for extracts that include a connection to the original data. If you connected directly to an extract file (.hyper or .tde) and then published it, the connection to the original data is not included and this option is therefore unavailable.

- **Append from Data Source**

  Updates the extract from another open (and compatible) data source. If you connected directly to an extract file (.hyper or .tde) and then published it, the connection to the original data is not included and this option is therefore unavailable.

Note: If you see the Tableau Data Server option, but the commands in the submenu are unavailable, the data source exists on the server, but is not an extract.

It is also possible to update an extract on Tableau Server using a command-line utility. For more information, see **Tableau Data Extract Command-Line Utility** below.

Tableau Data Extract Command-Line Utility

You can automate extract refresh tasks using the Tableau Data Extract Command-Line Utility. This is a command-line utility that comes with Tableau Desktop, through which you can refresh published extract data sources or append data to them from a file.

Requirements for using the Tableau Data Extract Command-Line Utility include the following:

- It is available with Tableau Desktop on Windows and can run only on a Windows system.
- It is not available with the trial version of Tableau Desktop.

- You can use it for extract data sources that don't use OAuth.
- You can use it to refresh single-connection data sources only. It does not work for multi-connection data sources.

Before you run the utility

When an extract refresh or append data is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop.

For more information, see **Extract Upgrade to .hyper Format** on page 1031.

Run the utility

1. Open the Command Prompt as an administrator and change to the Tableau Desktop bin directory. For example:

   `cd C:\Program Files\Tableau\Tableau 2021.2\bin`

2. Use either of the following commands, adding parameters described in the tables below.

   - `tableau refreshextract`
   - `tableau addfiletoextract`

> **Note:** When using the utility, always specify `tableau` on the command line or in scripts, never `tableau.exe`.

> **Note:** To troubleshoot, check the logs in the tableaucom.txt file in the \My Tableau Repository\Logs folder.

> **Note:** In a multiple-site environment, specify the site to which the command applies.

Syntax and parameters for the tableau refreshextract command

Use `tableau refreshextract` to refresh an extract on Tableau Server or Tableau Online. Refreshing an extract updates an existing extract with any modifications that have been made to the data source since the last refresh.

To see help for this command, at the Windows command prompt, type the following command:

```
tableau help refreshextract
```

Using parameters

- All options have a full form that you use with a double hyphen (for example, `--server`).
- Some options also have a short form that you use with a single hyphen (for example, `-s`).
- If the value for an option contains spaces, enclose it in quotation marks.
- The source being refreshed must be the original data source, not a .tde.

tableau refreshextract command options

Short Form	Full Form	Description
	--source-username <user name>	A valid user name for the data source connection. Use this option with --source-password, or use --original-file instead of the user name and password options. **Note**: You must provide the user name and password when refreshing a published extract, even if the data source was originally published with embedded credentials.
	--source-password "<password>"	The password for the data source user.
	--original-file <path and file name> or --original-file <path	Path and file name for the data source to be refreshed on the server. For example: --original-file c:\folder\file.csv To refresh a multi-file data source, pass the path to a folder that contains the data files. For example: --original-file c:\folder. If the file is on a network share, use the UNC format for

	`and folder name>`	the path: `\\server\path\filename.csv`
	`--force-full-refresh`	If the data source is set up for incremental refreshes, use this option to force a full extract refresh. If this option is not included, an incremental refresh is performed. Not all data sources support incremental refresh.
`-s <server http address>`	`--server <URL>`	The URL for the Tableau server on which the data is published. For Tableau Online, specify `https://online.tableau.com`.
`-t <site id>`	`--site <siteid>`	In a multiple-site environment, specifies the site to which the command applies. For Tableau Online, use this argument if your user name is associated with more than one site. For Tableau Server, if you do not specify a site, the default site is assumed. The site id is independent of the site name, and it is indicated in the URL when you view the site in a browser. For example, if the URL for the page you see after signing in to Tableau Online is `https://online.tableau.com/t/vernazza/views` the site id is *vernazza*.
	`--datasource <datasource>`	The name of the data source, as published to Tableau Server or Tableau Online.
	`--project <projectname>`	The project to which the data source belongs. If this option is not included, the default project is assumed. If the project you want to specify is a child project nested within a project hierarchy, you must use this parameter along with the `--parent-project-`

		path parameter.
	--parent-project-path path/to/project	If a project to which the data source is published is not at the top level of a project hierarchy, use this parameter along with the --project parameter to specify the path to a nested project. Use the forward slash character (/) to delimit project levels in the hierarchy. Use the backward slash (\\), to escape instances of forward or backward slash characters in project names. For example, for a project named Sandbox, in project Social, under top-level Marketing: --project Sandbox --parent-project-path Marketing/Social
-u <user name>	--username <username>	Valid Tableau Server or Tableau Online user.
-p "<password>"	--password "<password>"	The password for the specified Tableau Server or Tableau Online user.
	--proxy-username <username>	The user name for a proxy server.
	--proxy-password "<password>"	The password for a proxy server.
-c "<path and file name>"	--config-file "<path and file name>"	Path and file name information for a file containing configuration options for the command. Always enclose the path in double quotation marks. For more information, see **Using a config file** on page 1057 below.

Sample tableau refreshextract command

The following command refreshes an extract named CurrentYrOverYrStats that has been published to Tableau Server on-premises. This command specifies the following:

- The name of your Tableau Server.
- Server user name and password.
- Project name.
- The name of the data source to refresh, along with the data source username and password.

```
C:\Program Files\Tableau\Tableau 2021.2\bin>tableau
refreshextract --server https://our_server_name --username
OurServerSignIn --password "OurServerPwd" --project "New
Animations" --datasource "CurrentYrOverYrStats" --source-
username OurDatabaseSignIn --source-password
"OurDatabasePassword"
```

The following command refreshes an extract named CurrentYrOverYrStats that has been published to Tableau Online. This command specifies the following:

- Tableau Online user and password.
- Tableau Online site and project names.
- The data source, which in this case is hosted by a cloud-based data source provider (for example, Salesforce.com), and the username and password to sign in to the hosted data source.

```
C:\Program Files\Tableau\Tableau 2021.2\bin>tableau
refreshextract --server https://online.tableau.com --
username email@domain.com --password "OurServerPwd" --site
vernazza --project "New Animations" --datasource
"CurrentYrOverYrStats" --source-username database_
user@hosted_datasource_provider.com --source-password "db_
password"
```

To refresh an extract of file-based data source, provide the path to the original file from which you created the extract. If the file is on a network share, use the UNC format instead of a mapped drive.

```
C:\Program Files\Tableau\Tableau 2021.2\bin>tableau
refreshextract --server https://online.tableau.com --username
email@domain.com --password "OurServerPwd" --site vernazza --
project "New Animations" --datasource "CurrentYrOverYrStats"
--original-file "\\server\path\filename.csv"
```

Syntax for tableau addfiletoextract

Use `tableau addfiletoextract` to append file content to an extract that has been published to Tableau Server or Tableau Online. This command combines the two files.

If you want simply to update an existing extract with the latest changes, use the `refreshextract` command instead. Using `addfiletoextract` to update an existing extract will duplicate data instead.

To see help for this command, at the Windows command prompt, type the following command:

```
tableau help addfiletoextract
```

All options have a full form that you use with a double hyphen (for example, `--server`). Some options also have a short form that you use with a single hyphen (for example, `-s`). If the value for an option contains spaces, enclose it in quotation marks.

tableau addfiletoextract command options

Short Form	Full Form	Description
	`--file <path and file name>`	Path and file name information for the data file containing data to append. The file can be from Excel, Access, a Tableau data extract, or a delimited text file. It cannot be password protected. Use UNC format if the file is on a network share. For example, `\\server\path\filename.csv`
`-s <server http address>`	`--server <URL>`	The URL for the Tableau server on which the data is published. For Tableau Online, specify

		`https://online.tableau.com`.
`-t <site id>`	`--site <site id>`	In a multiple-site environment, specifies the site to which the command applies. For Tableau Online, you must include this argument if your user name is associated with more than one site. For Tableau Server, if you do not specify a site, the default site is assumed.
	`--datasource <datasource>`	The name of the data source, as published to Tableau Server or Tableau Online.
	`--project <pro-jectname>`	The project to which the data source belongs. If this option is not included, the default project is assumed. If the project you want to specify is a child project nested within a project hierarchy, you must use this parameter along with the `--parent-project-path` parameter.
	`--parent-pro-ject-path path/to/project`	If a project to which the data source is published is not at the top level of a project hierarchy, use this parameter along with the `--project` parameter to specify the path to a nested project. Use the forward slash character (`/`) to delimit project levels in the hierarchy. Use the backward slash (`\`), to escape instances of forward or backward slash characters in project names. For example, for a project named Sandbox, in project Social, under top-level Marketing: `--project Sandbox --parent-project-path Marketing/Social`
`-u <user-name>`	`--username <username>`	Valid Tableau Server or Tableau Online user.

`-p "<password>"`	`--password "<password>"`	The password for the specified Tableau Server or Tableau Online user.
	`--proxy-username <username>`	The user name for a proxy server.
	`--proxy-password "<password>"`	The password for a proxy server.
`-c "<path and file-name>"`	`--config-file "<path and file-name>"`	Path and file name information for a file containing configuration options for the command. Always enclose the path in double quotation marks. For more information, see **Using a config file** below below.

Sample tableau addfiletoextract command

```
C:\Program Files\Tableau\Tableau 2021.2\bin>tableau
addfiletoextract --server https://our_server_name --username
OurServerSignIn --password "OurServerPwd" --project "New
Animations" --datasource "CurrentYrOverYrStats" --file
"C:\Users\user1\Documents\DataUploadFiles\AprMay.csv"
```

```
C:\Program Files\Tableau\Tableau 2021.2\bin>tableau
addfiletoextract --server https://online.tableau.com --
username email@domain.com --password "OurServerPwd" --site
vernazza --project "New Animations" --datasource
"CurrentYrOverYrStats" --file
"C:\Users\user2\Documents\DataUploadFiles\AprMay.csv"
```

Using a config file

You can use a plain text editor, such as Notepad or Text Edit, to create a config (configuration) file that you can use with `tableau refreshextract` or `tableau addfiletoextract`. A config file can be useful if you expect to update the same data source regularly over time. Instead of having to type the same options each time you run a command, you specify the config file. A config file also has the advantage of not exposing user names and passwords on the command line.

Create the config file

For example, say you created a file called config.txt and saved it to your Documents folder. And in the file, you included the parameter information shown below.

For an extract published to Tableau Server:

```
server=https://our_server_name
username=OurServerSignIn
password=OurServerPwd
project=New Animations
datasource=CurrentYrOverYrStats
```

For an extract from a hosted data source, published to Tableau Online, where **server** is **https://online.tableau.com**:

```
server=https://online.tableau.com
site=vernazza
username=email@domain.com
password=OurPassword
project=New Animations
datasource=CurrentYrOverYrStats
source-username=database_user@hosted_datasource_provider.com
source-password=db_password
```

Reference the Config File from the Command Line

After you create the config file, you run the `tableau refreshextract` or `tableau addfiletoextract` command, pointing to the config file as the only option you use on the command line, and enclosing the config file’s path in double quotation marks. The syntax is as follows:

```
tableau refreshextract --config-file "<path>"
```

For example, to refresh the extract specified in the sample in the **Create the config file** on the previous page section, you would run the following command (making sure that you are working in the bin directory for your version of Tableau Desktop):

```
C:\Program Files\Tableau\Tableau 2021.2\bin>tableau
refreshextract --config-file
"C:\Users\user1\Documents\config.txt"
```

Syntax Differences for Config Files

The syntax for specifying options inside a config file differs from the syntax you use on the command line in the following ways:

- Option names do not begin with dashes or hyphens.
- You use an equals sign (with no spaces) to separate option names from option values.
- Quotation marks are not necessary (or allowed) around values, even when they include spaces (as for the `project` option in the example shown earlier).

Use Windows Task Scheduler to Refresh Extracts

You can use Windows Task Scheduler, in combination with the Tableau Data Extract Command-Line Utility, to automate regular updates to Tableau Desktop data sources from within your corporate firewall. You can configure a task to occur once per day, week, or month, or after a specific system event. For example, run the task when the computer starts.

To learn more, see the Task Scheduler How To... page in the Microsoft TechNet library.

Tableau Hyper API

You can use the Tableau Hyper API to create `.hyper` extract files (supported in Tableau 10.5 and later). Using the Hyper API you can build applications that can insert, read, update and delete data from those files.

- For more information about the Tableau Hyper API, see the Hyper API documentation.
- For more information about the `.hyper` format, see **Extract Upgrade to .hyper Format** on page 1031.

If you are using Tableau 10.5 and later, you can still use the Extract API 2.0 to create `.hyper` files. However, the Hyper API provides you more options for working with the extract files and offers improved performance.

If you're using Tableau 10.4 and earlier, you can continue to use the Tableau SDK to create `.tde` extracts and publish those extracts to Tableau Server. For more information, see the Tableau SDK documentation.

The Hyper API runs on Windows, Linux, and macOS and supports multiple programming languages, including C++, Java, and Python, and .NET (C#).

Manage Data Sources

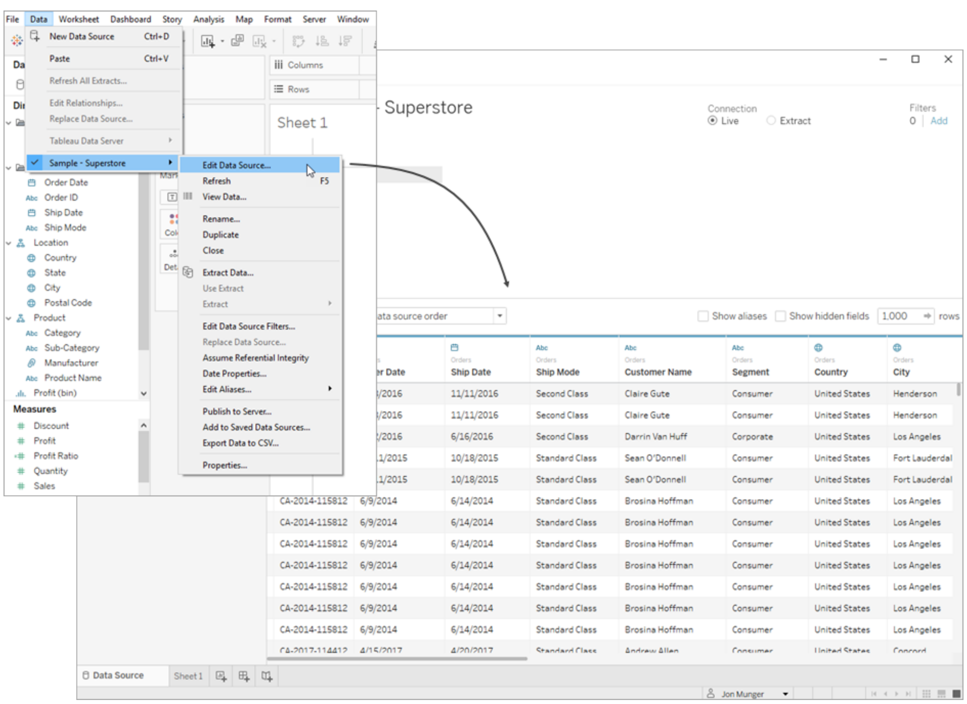
The topics in this section describe the tasks you can perform after you have set up data sources.

Edit Data Sources

At anytime during your analysis you can edit the data source used in the workbook.

How to edit the data source

1. On the **Data** menu, select a data source, and then select **Edit Data Source**.
2. On the data source page, make the changes to the data source.



You might want to edit the data source to:

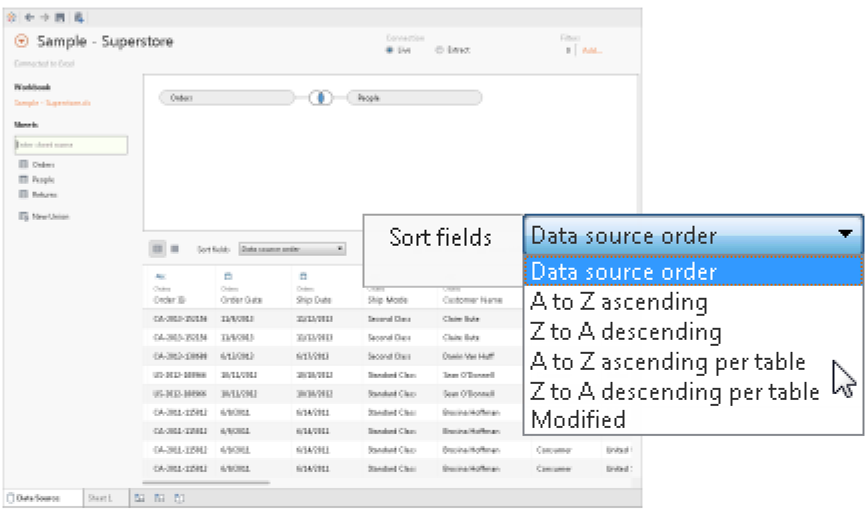
- **Join Your Data** on page 883
- **Connect to a Custom SQL Query** on page 959 (Tableau Desktop)
- **Use a Stored Procedure** on page 972 (Tableau Desktop)

Navigate the Data Grid

The following enhancements made to the grid on the **Data Source** page help you better see the data in your data source and prepare it for analysis.

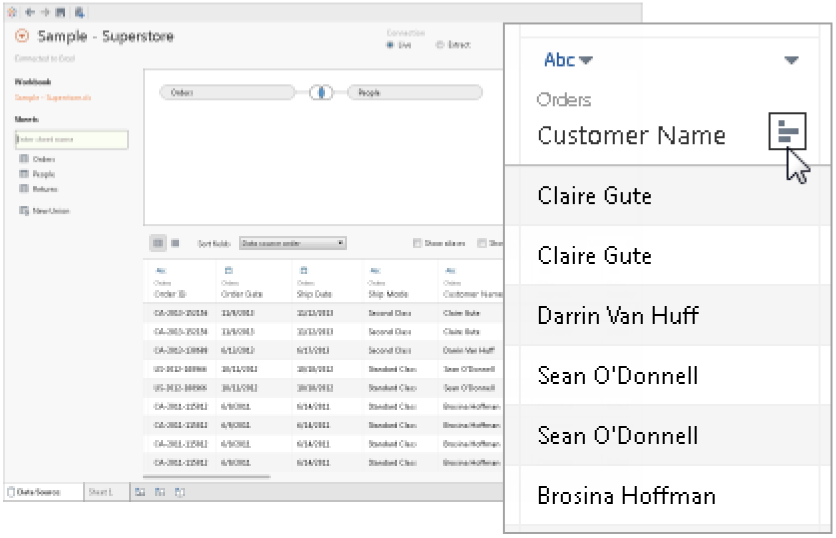
Sort columns and rows

Sort columns: Sort columns in the grid and metadata grid by selecting a sort option from the **Sort fields** drop-down list. Sort the columns by table or data source order.



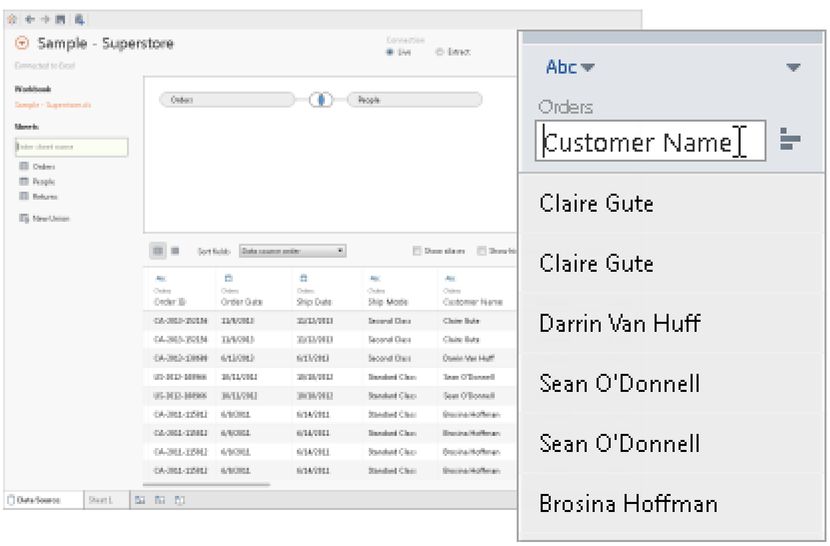
A **Modified** sort state can occur when some tasks cause new columns to be added to the grid.

Sort rows: Sort rows by clicking the sort button. Click the sort button once to sort rows in ascending order, click the sort button again to sort rows in descending order, and then click the sort button a final time to clear the sort.

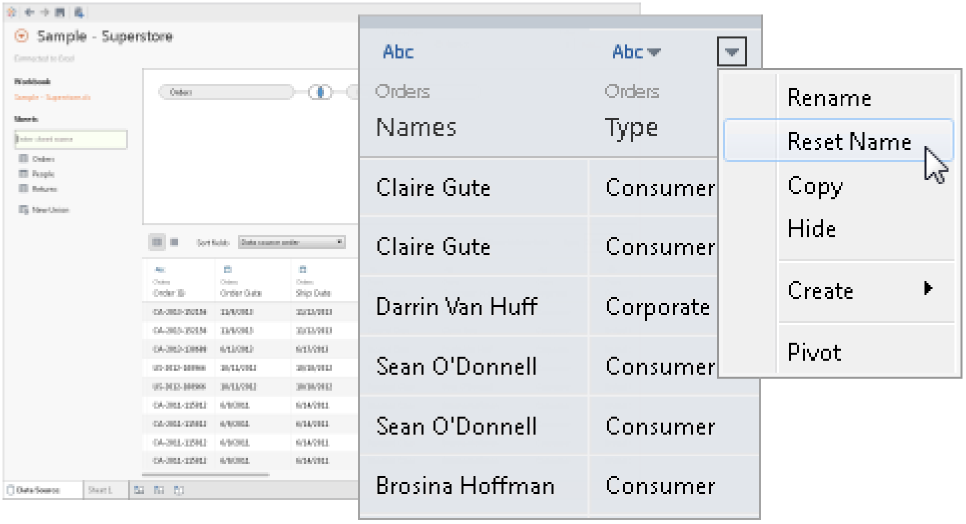


Change or reset field names

Rename column: Double-click the name of the column to rename the field.



Reset name: If you've renamed a field, click the column drop-down menu, and then select **Reset Name** to revert back to the original name of the field. You can also select multiple columns and perform the same action.

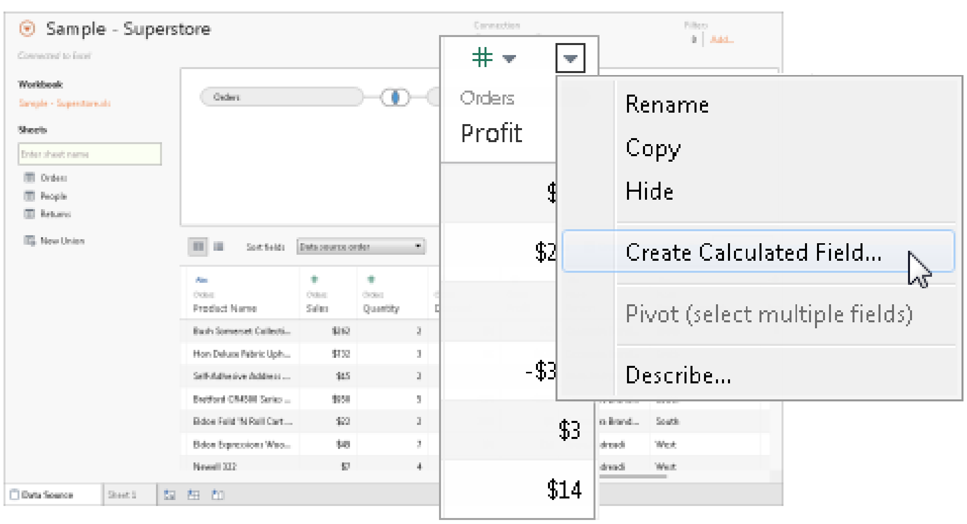


Original name indicates the name specified in the underlying data. You can use the **Revert** command to reset field names that have changed as a result of naming improvements Tableau has automatically made to the data source. For more information, see **Understand Field Type Detection and Naming Improvements** on page 1008.

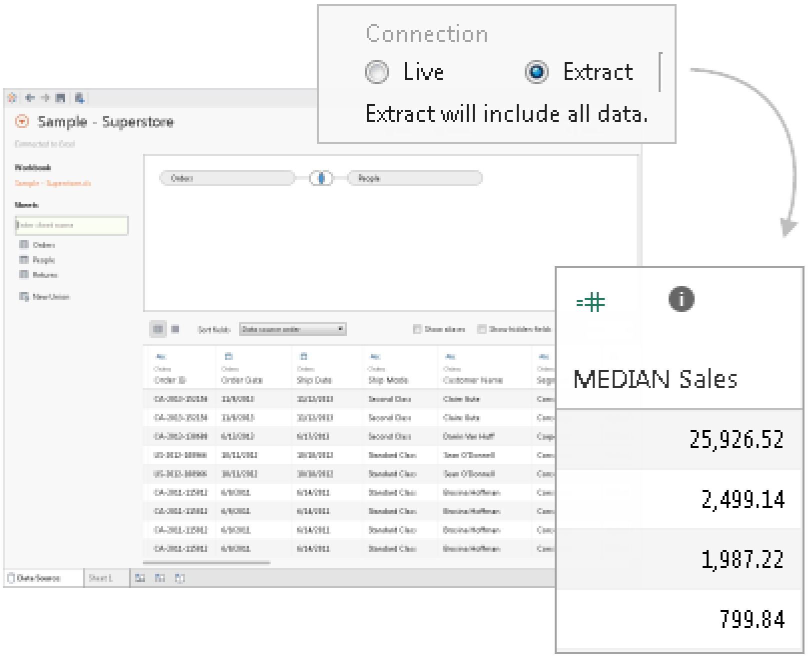
Create new columns and see extract data

Create new calculations: Create new calculations, groups, or bins based on existing fields in the data source. Click the column drop-down arrow and select one of the respective options.

> If you create an aggregate calculation, the values displayed in the grid are not meaningful until the calculation is used in context of the view's level of detail.



See extract data: For the Web Data Connector, file- and relational-based data sources in Extract mode, you can see extract data in the grid, including extract only calculations like MEDIAN.



When in Extract mode, the row order of the values in the data source might be different than the row order of values in Live mode.

Copy values

Copy values: Copy values in the grid by selecting the values and then pressing Ctrl+C (Command-C on a Mac). Alternatively, to copy values in the metadata grid, select the values, right-click, and then select **Copy**.

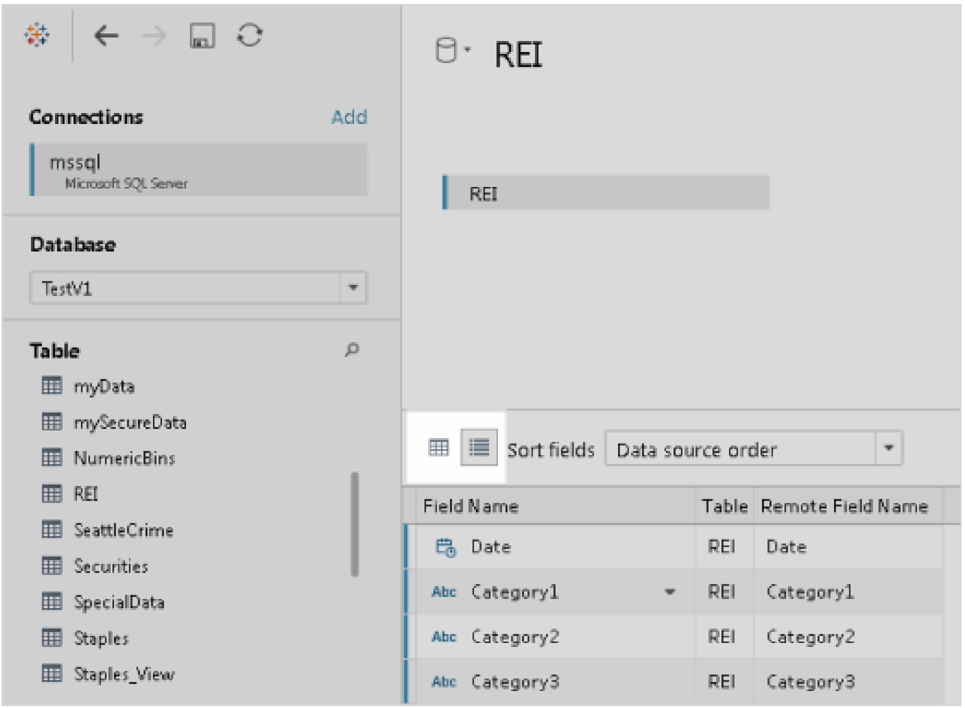
Examine your Metadata

Use the metadata area to quickly examine and change the structure of the Tableau data source. You can use the metadata area to review the columns in your data and their data types, do routine management tasks such as renaming a column, hiding a column, changing the data type of a column, and changing the geographical role of the column.

> **Note:** Metadata management tasks vary when working with a multi-dimensional data source.

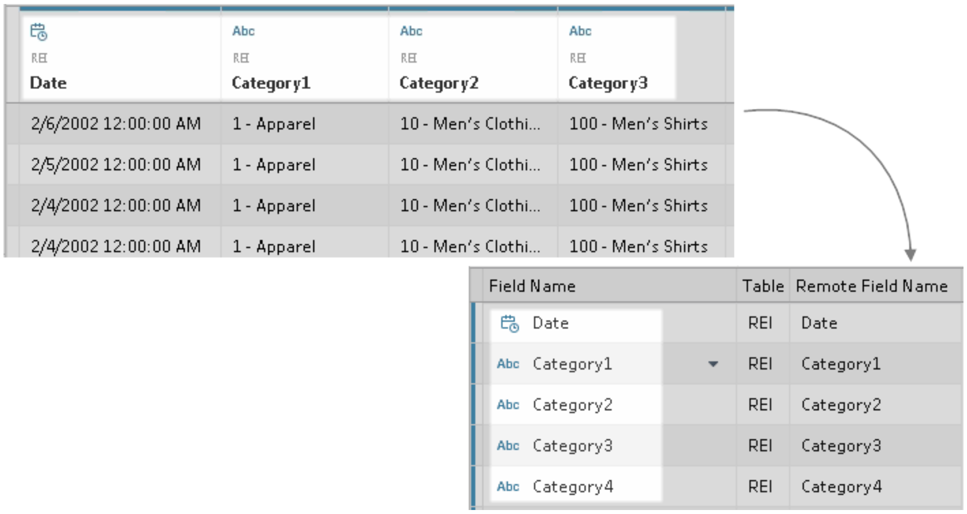
Go to the metadata area

After you have set up your data source, click the metadata area button. If you are connected to a cube (multidimensional) data source, the metadata area is shown by default. For other data source types, such as Salesforce or Tableau data extracts (TDE), the default view will vary.



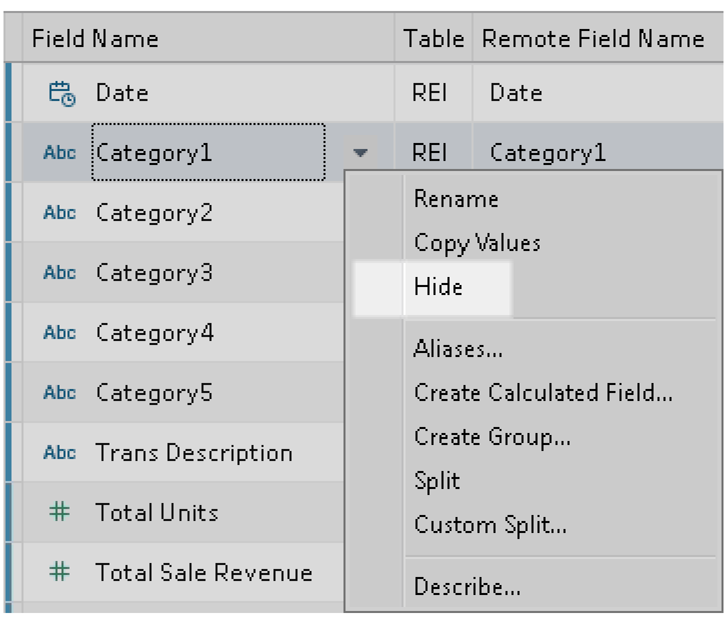
Review metadata

The columns of the data source are represented as rows in the metadata area. Each row shows the data type of the column, the column name in the Tableau data source, the name of the table in the underlying data from which the column comes, and the name of the column in the underlying data.



Manage metadata

Depending on the type of data you're connected to, you can perform routine management tasks like hiding multiple columns at once or quickly renaming columns. For example, to hide multiple columns at once, press the CTRL or Command key, select the columns you want to hide, click the drop-down arrow next to a column name, and then select **Hide**. Changes that you make in the metadata area of the data source do not modify your underlying data.



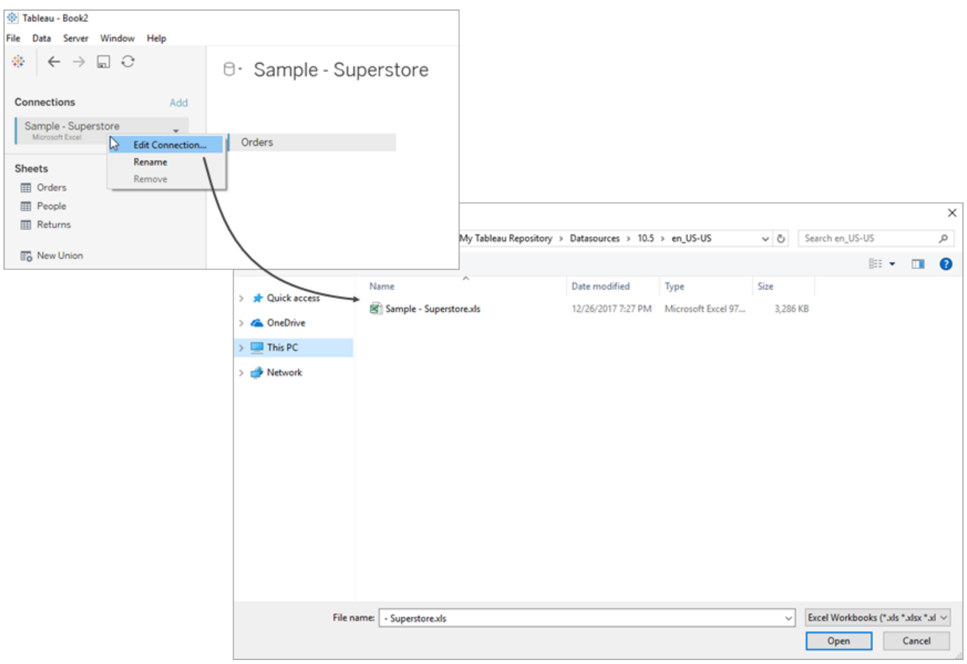
Change the Location of the Data Source

When you edit a data source, you have the option of changing the location of the data source. For example, suppose the name or location of a data source you were using has changed and is no longer available using the previous connection information. In this case, you can direct the workbook to the correct location without losing your work.

You can use this same method to apply analyses performed on one data source to a similar data source. Suppose you create a workbook containing several views involving markets, products, sales, and profits and you want to apply the analyses to a new data source. Instead of recreating each view, you can edit the original data connection and specify a new data source.

To change the location of a data source

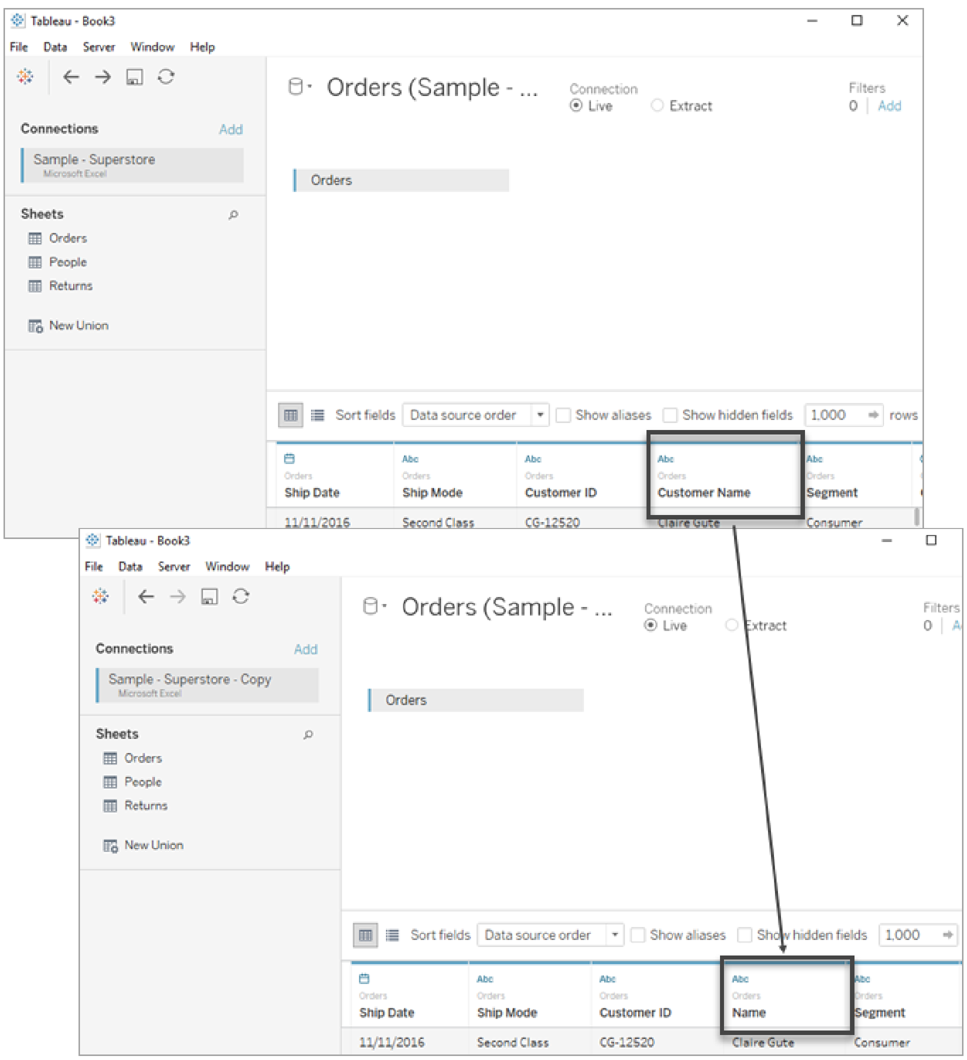
1. Right-click (control-click on a Mac) the data source in the Data pane and then select **Edit Connection**.
2. In the Edit Connection dialog box, navigate to the location of the data source or select a new data source.



Replace Field References

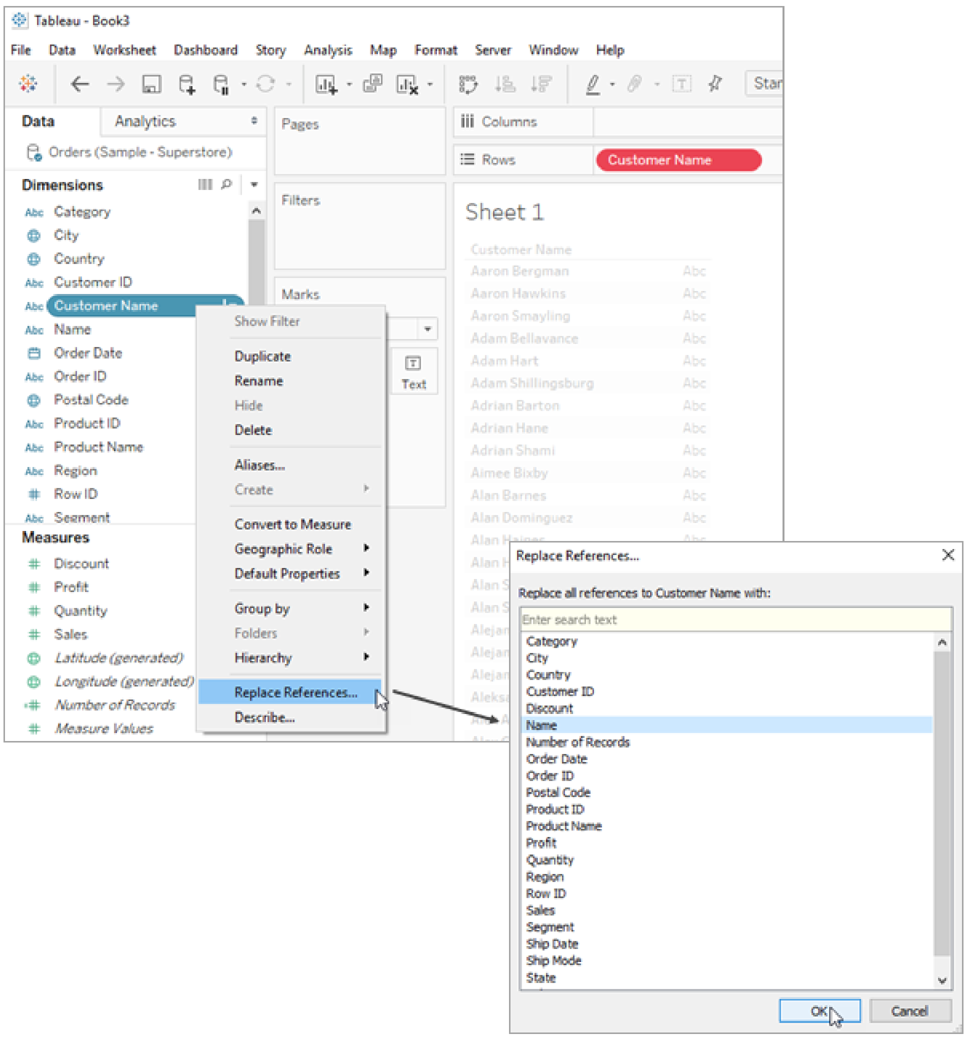
When you successfully connect to a new data source, all worksheets in the workbook that previously referred to the original data source now refer to the new data source. If the new data source does not have the same field names as the original workbook, the fields become invalid and are marked with an exclamation point  . You can quickly resolve the problem by replacing the field's references.

For example, say you have a workbook connected to a data source that contains a Customer Name field. Then you edit the data source to point to a new data source that has all the same data but instead of Customer Name, the field name has been changed to Name. The Customer Name field remains in the Data pane but is marked as invalid. To make the field valid, you can replace the references, which means you can map the invalid field to a valid field in the new data source (for example, Customer Name corresponds to Name).



To replace field references

1. Right-click (control-click on a Mac) the invalid field in the Data pane and then select **Replace References**.
2. In the Replace References dialog box, select a field from the new data source that corresponds to the invalid field.



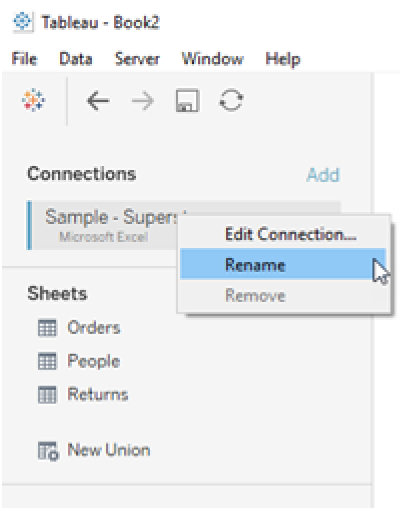
Rename the Data Source

When you connect to a data source, you are given the option to give it a name for use in Tableau Desktop.

To rename the data source

- On the **Data** menu, select **Rename**.

Naming a connection is useful when you have a single workbook connected to many data sources. The name you assign can help you keep track of the specifics of the connection. You can also review the connection properties by selecting a data source on the **Data** menu, and then selecting **Properties**.



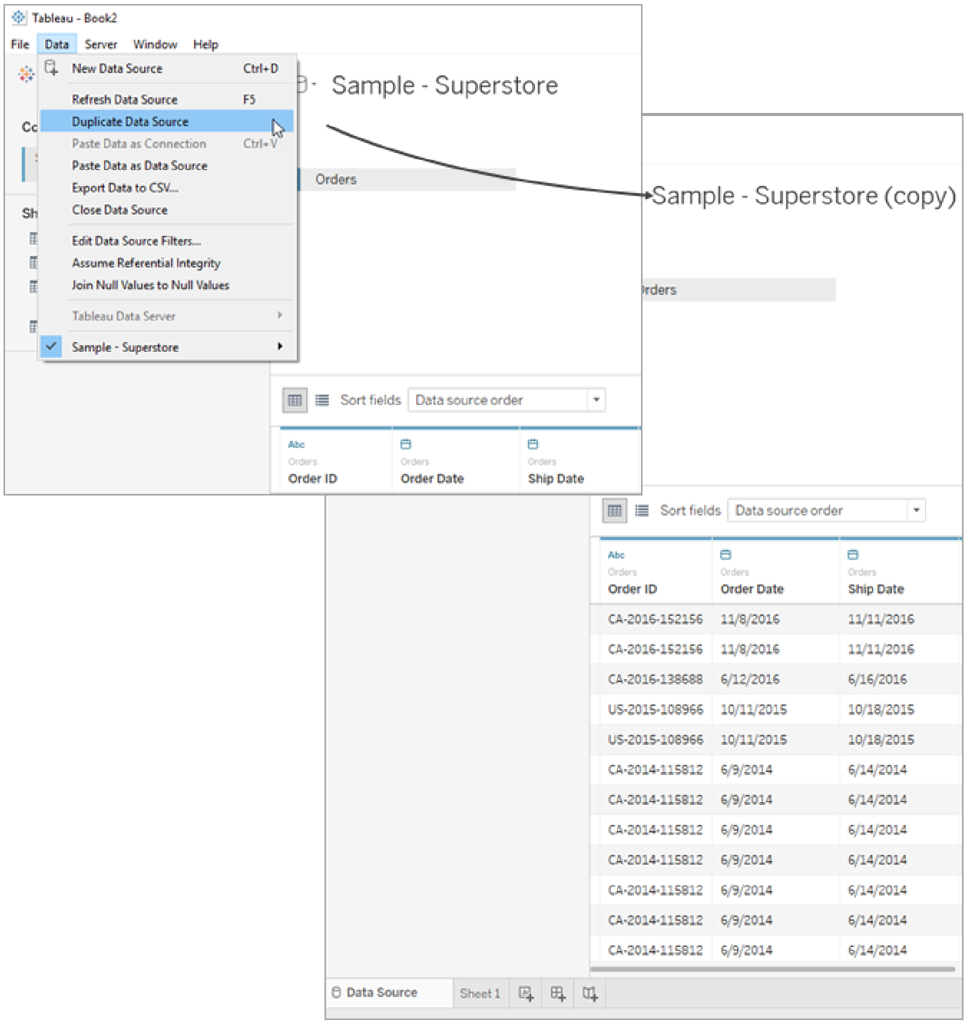
Duplicate the Data Source (Tableau Desktop)

Sometimes you'll want to make changes to a data source such as add more tables, hide and show fields, set field defaults, and so on. When you make these changes it affects all sheets that use the data source. You can duplicate the data source so that you can make the changes without affecting the existing sheets.

To duplicate a data source

- On the **Data** menu, select a data source , and then select **Duplicate**.

When you duplicate a data source, the duplicate source name has "(copy)" appended to the end.



Refresh Data Sources

If you are connected to a data source that has been modified, you can immediately update Tableau Desktop with the changes by selecting a data source on the **Data** menu and then selecting **Refresh**.

Not every connection will behave in the same way. There are three primary connection types: Live Connection, Extract, and Published Data Source. The type of connection is displayed by the icon next to your data source in the Data Pane.

Live Connection

A live connection sends queries to the database and updates the view depending on the results. However, the specific fields queried are defined when the connection is initially created. Refreshing the data source will update any new or changed fields. For more information, see **Connect to Your Data** on page 391.

Extract

Refreshing an extract will query the data source the extract was created from and rebuild the extract. This process might take some time, depending upon the size of the extract. For more information, see **Extract Your Data** on page 1014.

Published Data Source

When connected to a Published Data Source, the data source can be either a live connection or an extract. Selecting the **Data Source** tab will display whether the Published Data Source is a live connection or an extract. If the data source is an Extract, all refreshes of the extract are managed by Tableau Server and can only be refreshed by the server.

For more information, see **Best Practices for Published Data Sources** on page 3065.

Changes to underlying data

If a field that is used in a Tableau worksheet is removed from the underlying data of the data source and then the data source is refreshed, a warning message displays indicating that the field will be removed from the view and the worksheet will not display correctly because of the missing field.

If the underlying data changes—for example, if new fields or rows are added, data values or field names are changed, or data is deleted, Tableau will reflect those changes the next time you connect to the data source. However, because Tableau Desktop queries the data and does not import the data, you can immediately update Tableau to reflect the data modifications without disconnecting, provided the changes have been saved in the underlying data first.

> **Note about web authoring (Tableau Online and Tableau Server):** If you connect to a published **flat file** (Excel or text) directly in Tableau Online or Tableau Server, that data will not be refreshed even if it's modified. If your data is in an **on-premises server**

and is published to the web through Tableau Desktop, it will be rendered as an extract and won’t be refreshed. If you need to keep data published through Tableau Desktop fresh on the web, you can use Tableau Bridge. To learn more about these requirements, see Keep Data Fresh and **Use Bridge to Keep Data Fresh** on page 3084

Replace Data Sources

There are times when you may want to update a workbook or sheet to use a different data source. Rather than rebuild your workbook using a new data source, you can replace the data source.

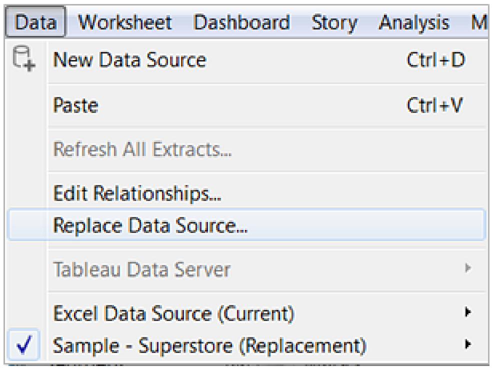
Tip: All sheets that use the data source will be updated when a data source is replaced. If you wish to replace the data source for a single sheet, you’ll need to use a workaround. (1) Copy the sheet to a new workbook and delete it from the old workbook. (2) In the new workbook, replace the data source. (2) Copy the sheet back to the original workbook. The data source and all relevant items for the sheet such as parameters and sets will be copied as well. For more information, see **Copy and paste sheets between workbooks** on page 3041.

Replacing a data source does not merge or edit the data sources. Rather, replacing a data source simply redirects fields used in the worksheet to map to corresponding fields in the new data source. Any fields that the user creates in the *original* data source (for example, calculated fields, groups, sets, etc.), that are included in the view are successfully copied over to the new data source. Fields that are not included in the view that don't exist in the *new* data source are not automatically copied and will need to be manually copied and pasted to the *new* data source before replacing the old one. Folder structures will also not be copied over.

The two data sources do not have to be identical, however, any differences between the workbooks will affect the sheets in the workbook and the fields in the view. More specifically, any fields, groups, sets, and calculated fields that do not exist in the new data source (or have a different name) are removed from the Data pane. For more information about how to replace field references to fix invalid fields, see **Edit Data Sources** on page 1060.

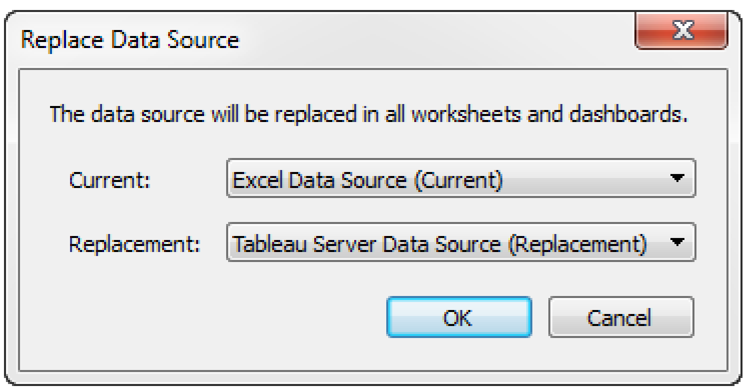
1. Open a workbook that connects to the *original* data source.
2. Select **Data** > **New Data Source** and then connect to the *new* data source.
3. On the Data Source page, drag a table to the canvas to set up the data source (if this is not automatically done for you).

4. Go to the sheet tab and select **Data** > **Replace Data Source**.



> **Note:** You must have at least one field in the view to make the **Replace Data Source** option available.

5. In the Replace Data Source dialog box, select the **Current** data source and the **Replacement** data source.



6. When finished, click **OK**.

All worksheets, dashboards, and stories that used the *original* data source are updated to use the *new* data source. You can click **Undo** on the toolbar to revert the change and return to the original data source.

You can only replace one data source with another data source when both data sources are relational data sources, or when both data sources are cube (multidimensional) data sources. In Tableau, cube data sources are supported for Windows only.

Published data sources

When you publish a data source from a workbook, the current workbook in Tableau Desktop will automatically connect to the published data source, and automatically close the local data source. So it is not necessary to replace the data source when you publish a data source. For more information, see **Publish a Data Source** on page 3071.

In addition, replacing a data source is not currently available for web authoring in Tableau Online or Tableau Server.

Save Data Sources

If you've created a data connection that you might want to use with other workbooks or share with colleagues, you can export (save) the data source to a file. You might want to do this also if you've added joined tables, default properties, or custom fields—such as groups, sets, calculated fields, and binned fields—to the Data pane.

> **Note:** This topic describes how to save a data source that you reuse locally, if you do not publish to Tableau Server or Tableau Online. For information about sharing data sources on a server, see Publish Data Sources and Workbooks.

Options for saving a local data source

You can save a data source to either of the following formats:

 **Data Source (.tds)** - contains only the information you need to connect to the data source, including the following:

- Data source type
- Connection information specified on the data source page; for example, database server address, port, location of local files, tables
- Groups, sets, calculated fields, bins
- Default field properties; for example, number formats, aggregation, and sort order

Use this format if everyone who will use the data source has access to the underlying file or database defined in the connection information. For example, the underlying data is a CSV file on your computer, and you are the only person who will use it; or the data is hosted on a cloud platform, and your colleagues all have the same access you do.

 **Packaged Data Source (.tdsx)** - contains all information in the data source (.tds) file, as well as a copy of any local file-based data or extracts.

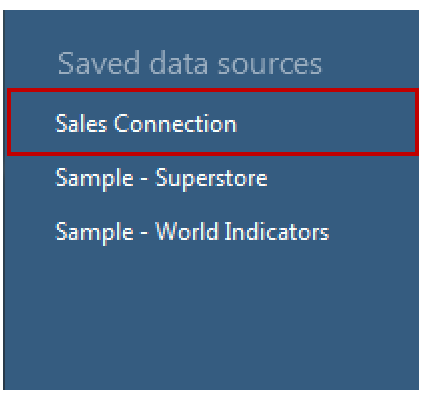
A packaged data source is a single zipped file. Use this format if you want to share your data source with people who do not have access to the underlying data that is defined in the connection information.

Save a data source

1. In Tableau Desktop, open the workbook that has the connection to the data you want to save as a file.
2. At the top of the **Data** pane, right-click (Control-click on Mac) the name of the data source, and then select **Add to Saved Data Sources**.
3. Enter a file name, select the file type (.tds or .tdsx), and then click **Save**.

Connect to your data source

By default, Tableau saves .tds and .tdsx files to the **Datasources** folder under your Tableau repository. When you use the default location, you can connect to the data source on the **Connect** pane.



If you specified a different location, you can connect to the data source by selecting **File** > **Open** and navigating to it.

Updating a data source after the underlying data location changes

If you move a local data file that a .tds file contains a reference to, when you try to open the data source in Tableau, you will be prompted to locate or replace the original data source. If you replace the original data source, the replacement data must be of the same type (for example, Excel or MySQL). To avoid saving a specific file path, save the data source as a .tdsx file.

Upgrade Data Sources

If you have workbooks that were created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, or you are using the Excel or text file legacy connection option, you have the option to upgrade the data sources in your workbook. By upgrading your Excel and text file data sources you can take advantage of 1.) better data interpretation, and 2.) compatibility on the Mac. If you are considering upgrading an existing workbook to use the new connection, review the tables below to compare how your data is processed and displayed between the legacy and default connection types.

If there are certain capabilities of the Jet driver that your data relies on, you might need to use the legacy connection to get the expected results. Workbooks that were created before Tableau Desktop 8.2 that use Excel and text file data sources automatically use the legacy connection. To create a new workbook that uses the legacy connection, in Tableau Desktop, navigate to your Excel or text file data, click the drop-down arrow on **Open**, and then select **Open with Legacy Connection**.

NOTE: Beginning with Tableau 2020.2, legacy Excel and Text connections are no longer supported. See the Legacy Connection Alternatives document in Tableau Community for alternatives to using the legacy connection.

File formats and attributes

The following tables show some examples of the types of Excel and text files, and tables that you can connect to in Tableau when using the legacy and default connections types.

Excel

Your Data	Legacy Connection	Default Connection
.xlsb file	Allows connections to Excel data in	Does not allow connections to Excel data in

Your Data	Legacy Connection	Default Connection
format	.xlsb file format.	.xlsb format. Connections to Excel data in .xls or .xlsx file formats can be used instead.
Excel built-in named ranges	Allows connections to built-in named ranges.	Built-in named ranges are hidden.
Hidden sheet	Allows connections to a table in a hidden sheet.	Hides a table in a hidden sheet.
Very hidden sheet	Allows connections to a table in a very hidden sheet.	Hides a table in a very hidden sheet.
Table that contain charts	Allows connections to tables that contain Excel charts. However, the table does not contain any values.	Hides tables that contain Excel charts.
Empty table	Allows connections to an empty table.	Hides an empty table.
Table headers	Treats the first row of a table as a header.	Automatically detects whether the first row of a table is a header.
Table width	Limits the table width to 255 columns.	No table width constraints.
Empty columns and rows	Empty columns and rows are visible.	Empty columns and rows are hidden because they do not contain any values.

Text

Your Data	Legacy Connection	Default Connection
File name length	Does not allow connections to a file whose file name is more than 64 characters long.	No length constraints for file names.
Multiple period in the file name	Does not allow connections to a file whose file name contains multiple periods.	No period constraints for file names.
Number of columns in the file	Uses the number of fields present in the first row to determine the number of columns in the file.	Scans the entire file and uses the most common number of fields in a row to determine the number of columns in the file.
Headers rows	Does not automatically detect whether the first row in the file is a header. **Note:** You can manually specify that the first row in the file is a header.	Automatically detects whether the first row in the file is a header. **Note:** You can manually override the detection.
Empty columns	Empty columns are visible.	Empty columns are hidden because they do not contain any values.

Characters and formatting

The following tables show some examples of how your Excel and text file data might be displayed in Tableau when using the legacy and default connection types.

Excel

Your Data	Legacy Connection	Default Connection
Number of	Field names are truncated to 64 characters.	No constraint on the

Your Data	Legacy Connection	Default Connection
characters in the field name		number of characters in a field name.
Special characters in field names	Special characters are not allowed in field names. For example, special characters like quotation marks and periods are converted to number signs. Brackets are converted to parentheses.	Special characters are allowed in field names.
Leading and trailing spaces in field names	Leading and trailing spaces in field names are allowed.	Leading and trailing spaces in field names are automatically removed from column names.
Duplicate field names	For duplicate field names, an index number is appended onto the field name. For example, Test Scores1.	For duplicate field names, a space and an index number are appended to the field name. For example, Test Scores 1.
Excel cell formatting	Supports cell formatting configured using the Format Cells option in Excel.	Does not support cell formatting configured using the Format Cells option in Excel.
Precision with currency values	Fields that contain currency values are represented with a maximum precision of four digits	Full precision of currency values is represented.

Text

Your Data	Legacy Connection	Default Connection
Special characters in file names	Special characters are not allowed in file names. For example, special characters like periods are converted to number signs.	Special characters are allowed.
Characters in header field names	Automatically removes leading spaces from header field names.	Automatically removes leading and trailing spaces from header field names.
Field separators	Recognizes only commas as field separators. **Note:** You can manually override the detection.	Automatically detects the field separator. **Note:** You can manually override the detection.
Text enclosed with quotation marks	Quotation marks used to enclose text values remain visible.	Quotation marks used to enclose text values are automatically hidden.
ANSI and OEM code pages	ANSI and OEM code pages are supported.	ANSI an OEM code pages are automatically converted to be cross-platform compatible.
BOM code pages	Inconsistently supports BOM code pages.	Supports BOM code pages.

Data type detection

The following tables show some examples of how data type is detected and how certain values are displayed in Tableau when using the legacy and default connection types.

Excel

Data Type Detection	Legacy Connection	Default Connection
Columns	Data type of a column is	Data type of a column is

Data Type Detection	Legacy Connection	Default Connection
	determined by the first 8 rows. **Note:** After the data type of a column in the table has been determined, it cannot be changed.	determined by the 95% of the first 10000 rows. **Note:** You can manually override the detection.
Date values without time	Date values are assigned a datetime data type.	Date values without time are assigned a date data type. Date values with time are assigned a datetime data type.
Number values	All number values are represented as real numbers.	Number values without decimal points are represented as integers.
Null values	If a column contains a null cell, the data type for the column is automatically designated as a string data type.	Null cells do not affect data type detection.
Reference errors or empty cells	If a column contains cells with reference errors or empty cells, the entire column is interpreted as a string data type.	Reference errors or cells with no values do not affect the data type detection.
Overriding data type detection	After the data type of a column in the table has been determined, it cannot be changed.	Data type of a column can be changed after the automatic detection.
Time value precision	The smallest measurement of time values is whole seconds.	The smallest measurement of time values is fractional seconds.

Data Type Detection	Legacy Connection	Default Connection
DecimalSymbol and CurrencyDecimalSymbol schema.ini values	Both DecimalSymbol and CurrencyDecimalSymbol schema.ini values are recognized.	DecimalSymbol and CurrencyDecimalSymbol scheam.ini values are recognized. However if both values are used, DecimalSymbol takes precedence.
Cells formatted as text	The data type of a column is detected as a string data type when cells are formatted as text using the Format Cells option in Excel.	Does not support cell formatting configured using the Format Cells option in Excel.

Text

Data Type Detection	Legacy Connection	Default Connection (8.2 and later)
Columns	Data type of a column is determined by the first 25 rows.	Data type of a column is determined by the first 1024 rows.
Boolean (True/False) values	Boolean values are assigned the string data type.	Boolean values are assigned the boolean data type.
Values that become null	Spaces in a cell, whether enclosed in quotation marks or not, are treated as null values. Columns with null values are detected as the string data type.	Two field separators in a row are treated as a null value. Null values are ignored during data type detection.

Data source connection properties

The following table shows examples of the differences in data source connection properties in Tableau when using the legacy and default connection types.

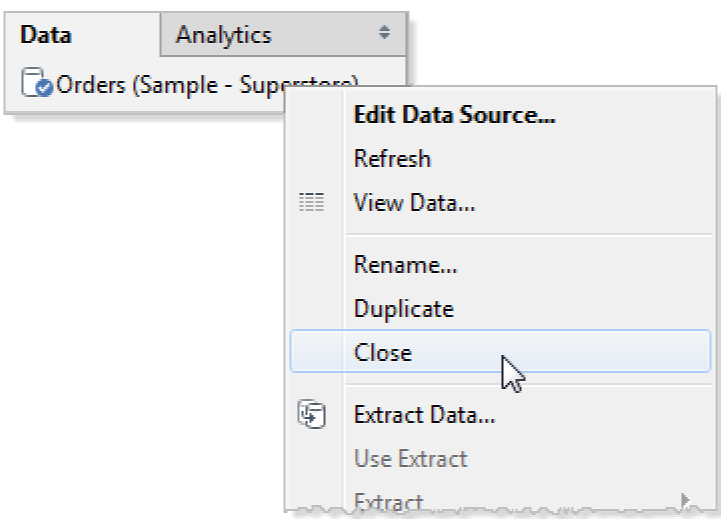
Property	Legacy Connection	Default Connection
Custom	Custom SQL is allowed.	Does not allow the

Property	Legacy Connection	Default Connection
SQL		use of Custom SQL.
Join type	Allows left, right, and inner join types.	Allows left, right, inner, and full outer join types.
Join operators	Allows equal to (=), greater than (>), greater than or equal to (>=), less than (<), less than or equal to (<=), and not equal to (<>) join operators.	Allows equal to (=) join operators.

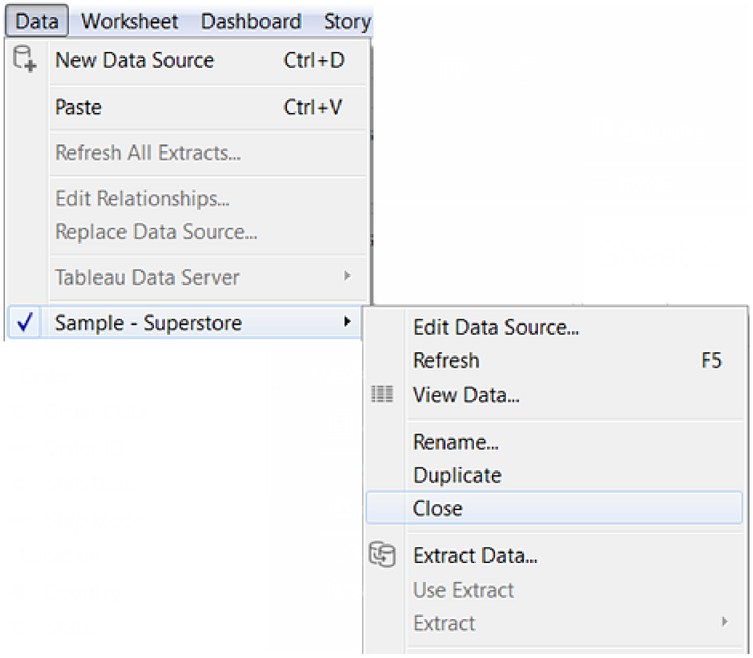
Close Data Sources

You can close a data source at any time. Doing so does not modify the data source. Instead, it disconnects Tableau from the data so that you can no longer query it. Additionally, the data source is cleared from the Data pane and all open worksheets associated with the data source are cleared. If you accidentally close a data source, use the Undo button to reopen it. Close a data source by doing one of the following:

- Right-click (control-click on a Mac) the data source at the top of the Data pane and select **Close**.



- Select a data source on the **Data** menu and then select **Close**.



Cube Data Sources

Cube data sources (also known as multidimensional or OLAP data sources) have certain characteristics that differentiate them from relational data sources when you work with them in Tableau. This topic describes these differences, and also identifies some Tableau features that are not available when you connect to a cube data source. In many cases there are alternative approaches that you can use to compensate for the unavailability of these features with cube data sources, but you may also have the option to connect directly to a relational database that was used as the source for a cube data source. Talk to your database administrator to find out if this is an option.

> **Note**: Cube data sources are supported only for Tableau Desktop on Windows—not for the Mac.

What are cube data sources?

A cube data source is a data source in which hierarchies and aggregations have been created by the cube's designer in advance.

Cubes are very powerful and can return information very quickly, often much more quickly than a relational data source. However, the reason for a cube's speed is that all its aggregations and hierarchies are pre-built. These definitions remain static until the cube is rebuilt. Thus, cube data sources are not as flexible as relational data sources if the types of questions you need to ask were not anticipated by the original designer, or if they change after the cube was built.

The cube data sources supported in Tableau are

- Oracle Essbase
- Teradata OLAP
- Microsoft Analysis Services (MSAS)
- SAP NetWeaver Business Warehouse
- Microsoft PowerPivot

Watch a video: To see related concepts demonstrated in Tableau, watch Analysis with Cubes and MDX, a 5-minute free training video. Use your tableau.com account to sign in.

Create calculated members using MDX formulas

When working with a cube data source, you can create calculated members using MDX formulas instead of creating Tableau formulas. MDX, which stands for Multidimensional Expressions, is a query language for OLAP databases. With MDX calculated members, you can create more complex calculations and reference both measures and dimensions. A calculated member can be either a calculated measure, which is a new field in the data source just like a calculated field, or a calculated dimension member, which is a new member within an existing hierarchy. For details, see **How to Create a Calculated Member** on page 1094.

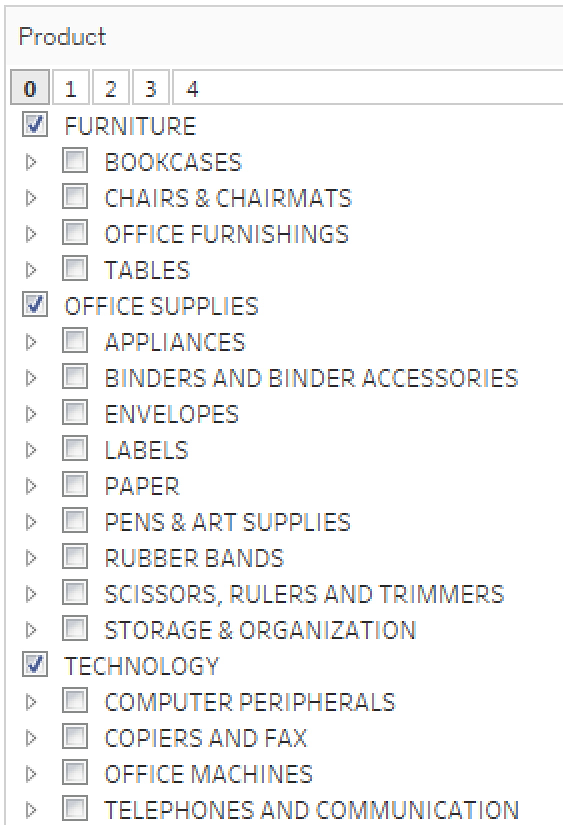
Tableau features that are affected when you use a cube data source

When you use a cube data source, not all Tableau features work the same as with relational data sources, or are even available. The following table details the differences.

Feature	**Status for Cube Data Sources**
Actions	With Microsoft Analysis Services connections, drill-down actions

	defined in the cube are not available in Tableau. Cube data sources do not accept actions from relational or other cube data sources. For example, suppose you have a workbook with a view that uses a MySQL data source, a second view that uses cube data source A, and a third view that uses cube data source B. Actions in the view using the MySQL data source won't affect the views using the cube data sources, and actions in the view for one cube data source won’t affect the other. But actions in the views using the cube data sources will affect the view using the MySQL data source.
Advanced analytics features	Level of detail expressions, trend lines, forecasting, and clustering are not supported for cube data sources.
Aggregate calculation functions	Cube data sources are pre-aggregated and thus do not support aggregation functions, such as `SUM()`, `AVG()`, and `CNT()`. It may be possible to use Table Calculations to perform aggregation operations on the cell-level results from the cube in Tableau.
Aliases	Aliases for cube databases are created by the cube's designer and can be activated in Tableau by selecting the data source from the Data menu and then choosing **Alias File**. Talk to your database administrator to find out whether your database has aliases available. Aliases are not supported by Microsoft Analysis Services databases. By default, the alias for every member of every dimension is initially defined to be the original member name.
Bins	For cube data sources, the **Create Bin** command is not available for measures. You can, however, write a calculation that takes cube cell results and bins them. For example: `str((INT([Internet Sales Amount]/1000)) * 1000)`
Cube KPI data type	When connected to Microsoft Analysis Services, any KPI calculations defined in the cube are not available.

	However, you can write your own KPI calculations in Tableau. You can also use Tableau parameters to create highly flexible what-if KPI analysis. For more information, see **Visualize Key Progress Indicators** on page 1684.
Cube lag functions	For cube data sources, you cannot use cube lag functions in the Tableau calculation editor. You can use Tableau Table Calculations to calculate certain percentages and totals instead. For more information, see **Transform Values with Table Calculations** on page 2406. Alternatively, you can use an MDX Lag function directly in Tableau using a calculated member. For example: `Avg ( { [Date].[Calendar].CurrentMember.Lag(4) : [Date].[Calendar].CurrentMember } , [Measures].[Internet Sales Amount] )`
Custom multidimensional expression (MDX) statements	When connected to a cube, you cannot make a connection to a custom MDX statement. All sub-cube definitions need to be created on the server by implementing necessary cube perspectives, partitions, dimensions, or cell security.
Data blending	Cube data sources can only be used as the primary data source for blending data in Tableau. They cannot be used as secondary data sources. For details, see **Troubleshoot Data Blending** on page 930.
Date dimensions	For cube data sources, date dimensions are typically organized into hierarchies that contain levels such as year, quarter, and month. In addition, some multidimensional data sources have time intelligence enabled, which makes it possible to look at data levels different ways, such as Months by Year, Months by Quarter, Weekends, etc. These levels are represented as attributes of the hierarchy. For details, see **Dates and Times** on page 1245.
Data source filters	Data source filters are not available for cube data sources. All field values must be defined in the cube prior to analysis in Tableau.
Extracts	You cannot create extracts from most cube data sources. Cube and relational data sources have incompatible data structures, which makes extracting data from a cube and storing it in a relational data source,

<table>
<tr>
<td></td>
<td>
such as the data engine, impossible in most cases.
Starting in 10.4, you can create SAP BW extracts without the need for a special product key from Tableau. For information about support for and limitations of SAP BW extracts, see **SAP NetWeaver Business Warehouse** on page 618.
</td>
</tr>
<tr>
<td>Filters</td>
<td>
When you display a filter for a cube dimension, all levels of the hierarchy for that dimension are included in the filter. For example:


You can use cube attributes as filters, to show just a single level instead of a hierarchy. In the Data pane, attributes appear in the Dimensions section and are indicated by this icon:
Slicing filters behave differently with a cube data source than with a relational data source. See **Create Slicing Filters** on page 1091 for details.
</td>
</tr>
</table>

	Some dimension filter card options are not available. For example, single value (drop-down), multiple values (custom list), etc. Instead, dimensional filters shown in the view retain their hierarchical look and feel, and cannot be changed into specific list types. You can create a Tableau set containing certain values from the hierarchy and then use the set as a filter in the view with the expected filter options (right-click the set in the Data pane, and then click **Show Filter**). In addition, see **Filter Your Data Carefully** on page 3000 for information on performance considerations when filtering data from cube data sources.
Groups	You cannot create groups when working with a cube data source. Any concept of groupings should be pre-defined in the cube as dimensional attributes or cube sets. However, you can write multidimensional expressions (MDX) directly in Tableau using a calculated member to create a group. For example: `[Customer].[Customer Geography].[France] + [Customer].[Customer Geography].[Germany]` For a demonstration of this process, see the Tableau on-demand video Analysis with Cubes and MDX.
Hierarchies	For cube data sources, hierarchies must be defined in the cube prior to analysis.
Parameters	For cube data sources, you cannot use parameter values to filter dimensions in an MDX calculation.
Publishing	Workbooks using cube data source can be published to Tableau Server, but do not support pass-through connections. This means that you cannot make a connection from Tableau Server using such a data source. It also means you cannot create a workbook using the data source in Tableau Server. For details, see Cube Data Sources in the Tableau Server help. Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it

	locally. Workbooks that use cube data source cannot be published to Tableau Online.
Sets	Cube data sources support hierarchical sets, which filter data to the selected members and all of their descendants. For details, see **Examples of sets** on page 1199.
Tableau string functions	With cube data sources, dimensions are not available in the calculation editor. However, you can write MDX inside Tableau using a calculated member to manipulate dimensional values. For example: `LEFT([Product].[Product Categories].DataMember.MemberValue,LEN([Product].[Product Categories].DataMember.MemberValue)-5)`
Type conversions	For cube data sources, some type conversion functions are not available in the calculation editor. As a rule, data type conversions should be defined in the cube prior to analysis. In particular, changing the data type of a cube dimension to a date in Tableau sometimes gives incorrect information. This feature is supported for some cube dimensions, depending on how the dates are formatted in the cube. You can write Multidimensional Expressions (MDX) directly in Tableau using a calculated member to change the data type of a cube dimension to a date. For example: `CDATE([Date].[Date].CurrentMember.MemberValue)`

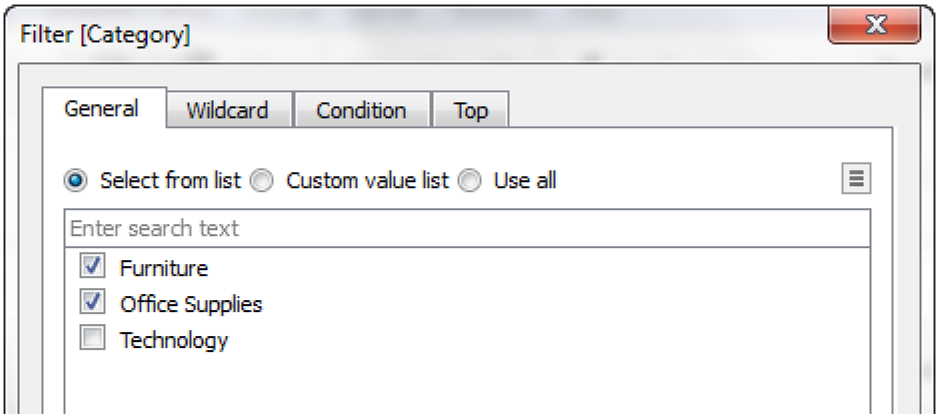
Create Slicing Filters

Slicing filters, also known as slicers or calculation filters, are dimension filters that behave differently for multidimensional data sources than they do for relational data sources.

> In Tableau, multidimensional data sources are supported only in Windows.

A slicing filter exists when:

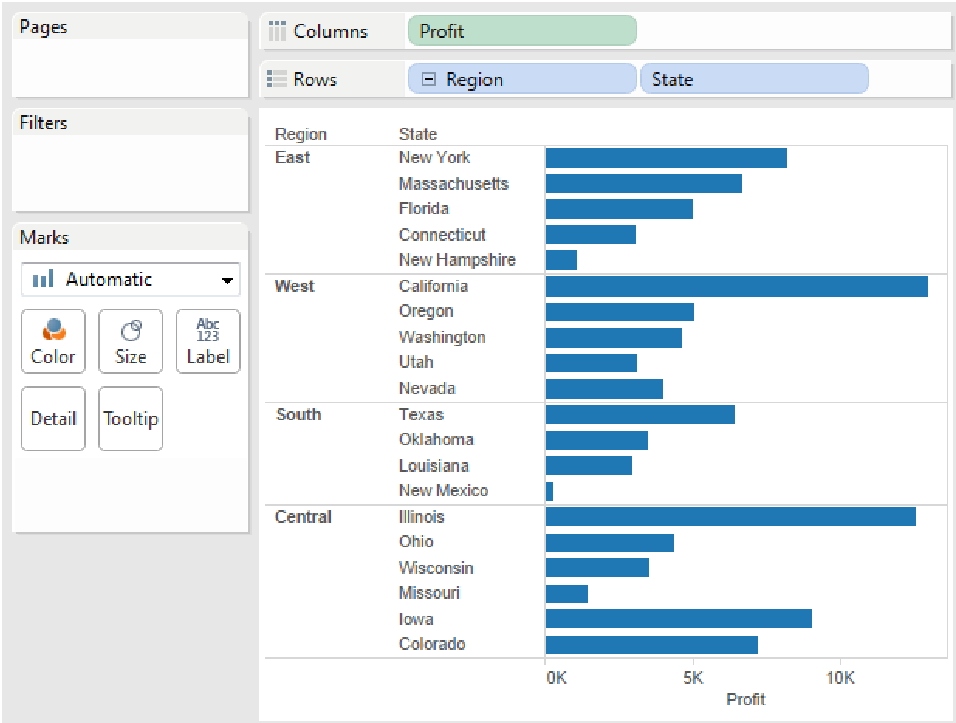
- The dimension that you place on the Filters shelf is not also on any other shelf in the view.
- The filter is defined to include multiple values. For example:



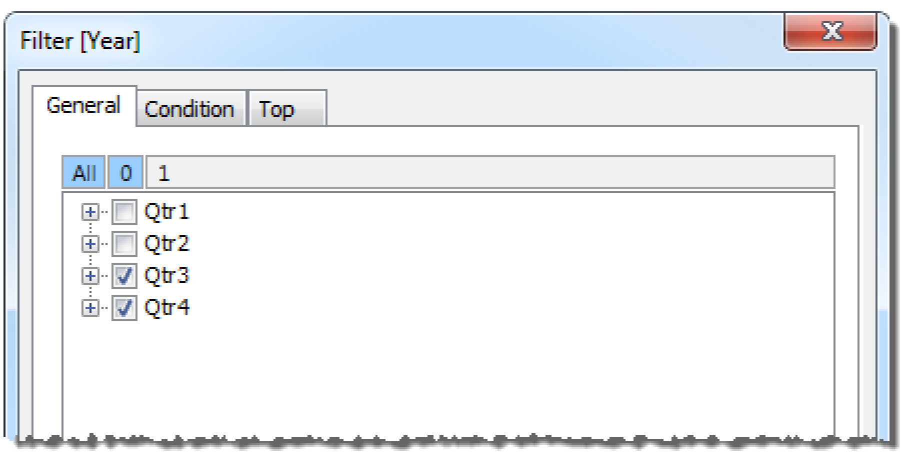
For relational data sources, there is nothing remarkable about a slicing filter. Tableau recalculates values for each measure in the view using the configured aggregations.

But because multidimensional data sources contain data that is aggregated in the data source, no aggregation can be defined for the “slice” that the dimension filter cuts through the values of individual measures. Tableau therefore automatically performs a summation. And because multiple values are defined, the result is the sum of a group of sums.

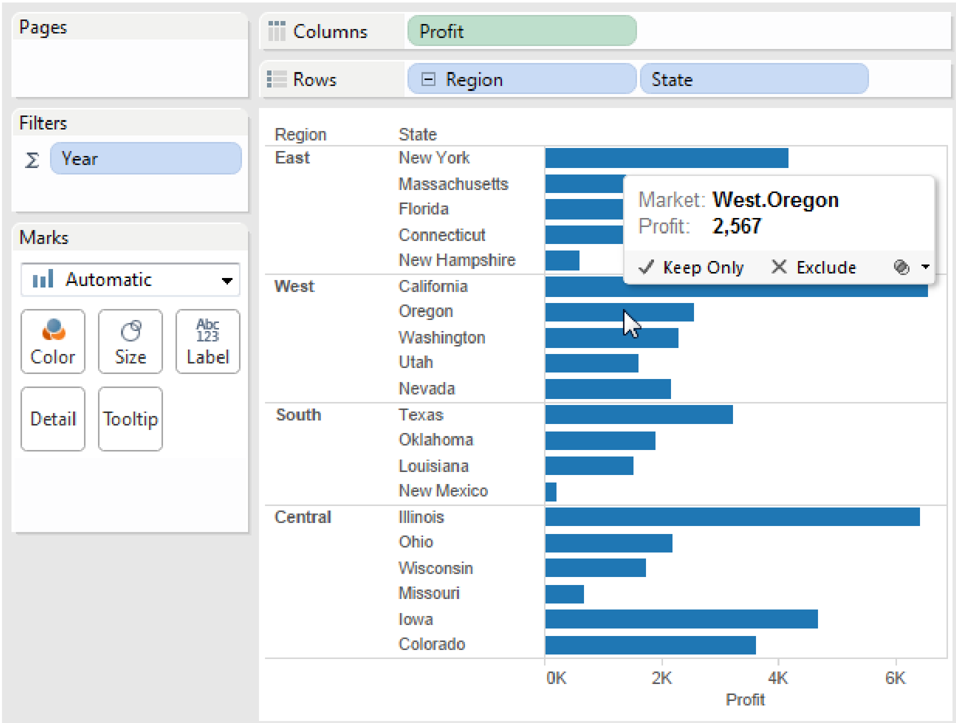
For example, consider the view shown below, which shows profit by region and state.



A slicing filter that includes the Q3 and Q4 members of the Quarter dimension is then applied to the data.



In the resulting view, the profit for Oregon in the West region is 2,567. This number was calculated by summing data values for Qtr3 and Qtr4.



The summation symbol (Σ) you see on the Filters shelf in the view above indicates that this is a slicing filter. If the original aggregation for a field is non-additive (for example, Count Distinct) the values you see after applying a slicing filter may be unexpected.

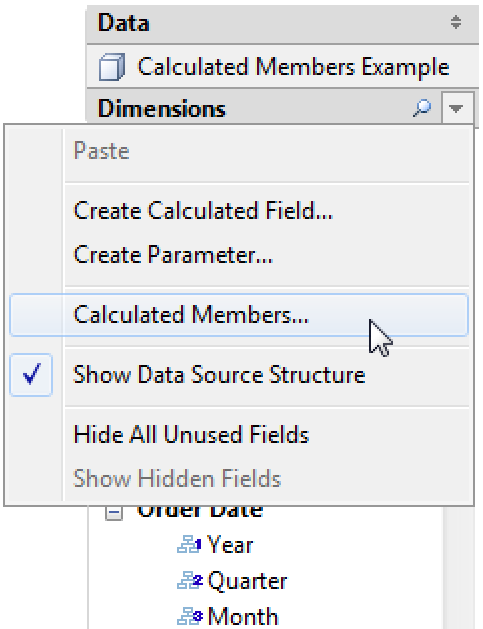
How to Create a Calculated Member

If you are using a multidimensional data source, you can create calculated members using MDX formulas instead of Tableau formulas. A calculated member can be either a calculated measure, which is a new field in the data source just like a calculated field, or a calculated dimension member, which is a new member within an existing hierarchy. For example, if a dimension Product has three members (Soda, Coffee, and Crackers) you can define a new calculated member Beverages that sums the Soda and Coffee members. When you then place the Products dimension on the **Rows** shelf it displays four rows: Soda, Coffee, Crackers, and Beverages.

Note: In Tableau, multidimensional data sources are supported only in Windows.

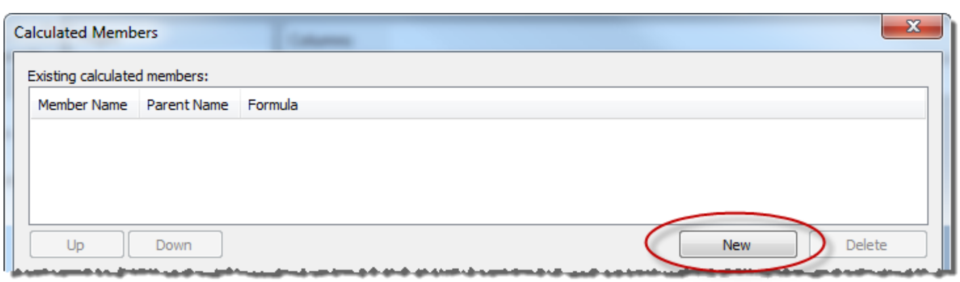
Defining Calculated Members

You can define a calculated dimension member by selecting **Calculated Members** from the Data pane menu. In the Calculated Members dialog box that opens, you can create, delete, and edit calculated members.

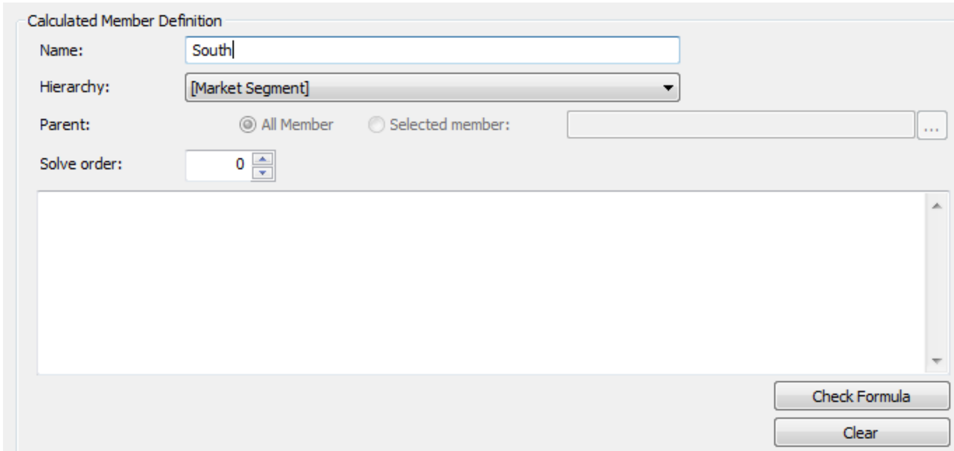


To create a new calculated member do the following:

1. Click New to add a new row to the list of calculated members at the top of the dialog box.



2. Type a Name for the new calculated member in the Member Definition area of the dialog box.



3. Specify the **Parent** member for the new calculated member. **All Member** is selected by default. However, you can choose **Selected Member** to browse the hierarchy and select a specific parent member.

> **Note**: Specifying a parent member is not available if you are connected to Oracle Essbase.

4. Give the new member a solve order.

   Sometimes a single cell in your data source can be defined by two different formulas. The solve order defines the precedence given to each formula. Formulas with a lower solve order are solved first. The default solve order is zero.

5. If you are connected to a Microsoft Analysis Services data source, the calculation editor contains a **Run before SSAS** check box. Choose this option to execute the Tableau calculation before any Microsoft Analysis Services calculations. For information on connecting to Microsoft Analysis Services data sources, see **Microsoft Analysis Services** on page 559

6. Type or paste an MDX expression into the large white text box.

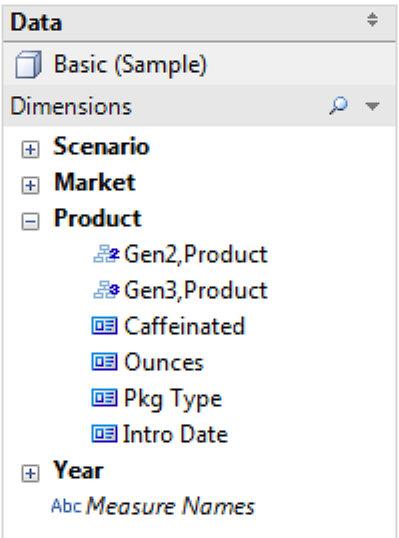
7. Click **Check Formula** to verify that the formula is valid.

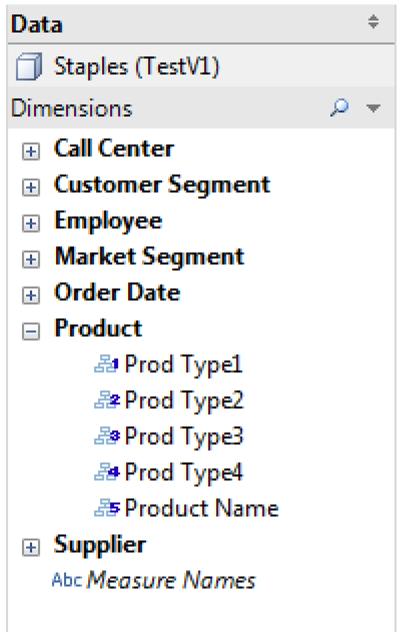
8. When finished, click **OK**.

The new member displays in the **Data** pane either in the Measures area, if you chose [Measures] as the parent member, or in the Dimensions area under the specified parent member. You can use the new member just like any other field in the view.

Multidimensional Hierarchies

Multidimensional data sources contain hierarchies. For example, your database might contain a Product dimension that includes members such as product family, product department, and so on organized into a hierarchy, or you might have a Time dimension that includes years, quarters, and months.





Note: In Tableau, multidimensional data sources are supported only in Windows.

Drilling Down and Up in a Cube Data Source Hierarchy

Note: The following applies only to certain multidimensional data sources.

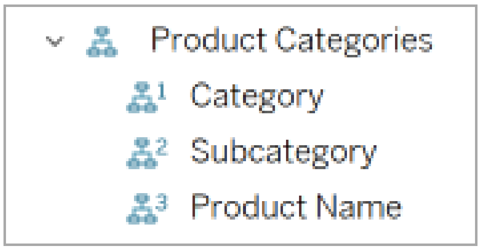
Multidimensional (cube) data sources contain hierarchies. One of the most useful ways to navigate hierarchies is to drill down or drill up. For example, if you are examining the sales totals for various years, you can then drill down and view sales for all of the months within each year. Alternatively, if you are examining sales totals for all months, you can then drill up and view the sales for each year.

Hierarchies in cube data sources

When connected to cube data sources, you cannot create or customize hierarchies in Tableau. Hierarchies must be created in the cube **before** you connect to it in Tableau.

Hierarchies appear with the following icon in the Data pane: 

You can see an example of a hierarchy below:

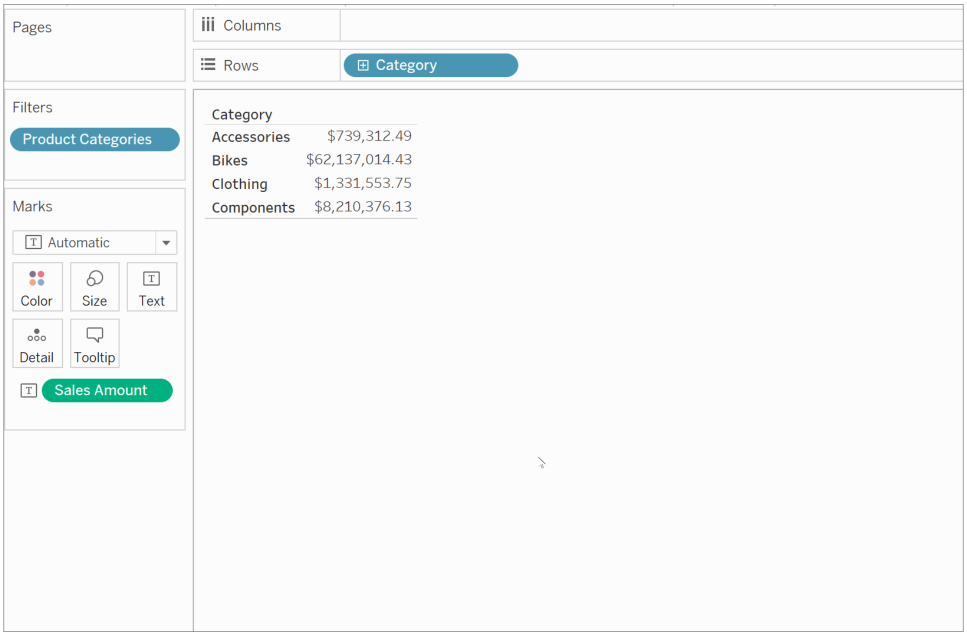


> **Note**: You can only drill up or down in fields that are part of a hierarchy.

Drill up and down for **all** dimension members in a hierarchy

To drill up or down for all dimension members in a hierarchy:

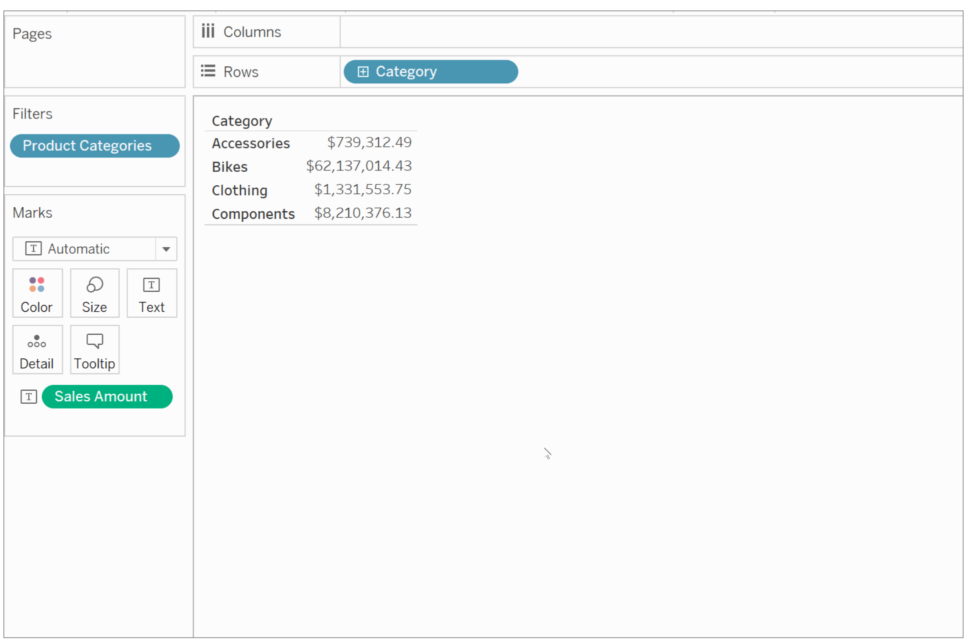
- On the Columns or Rows shelf, or the Marks card, click the plus sign (+) on a field to drill down, or the minus sign (-) to drill up.



Drill up and down for **individual** dimension members in a hierarchy

To drill down and drill up for individual dimension members in a hierarchy:

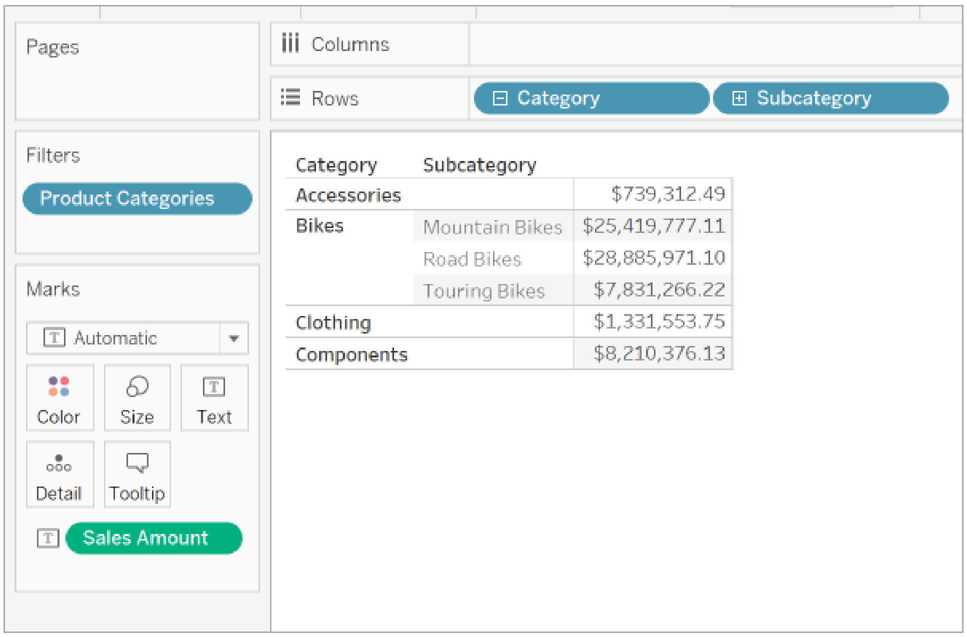
- Right-click a table header and select **Drill Down** or **Drill Up** .



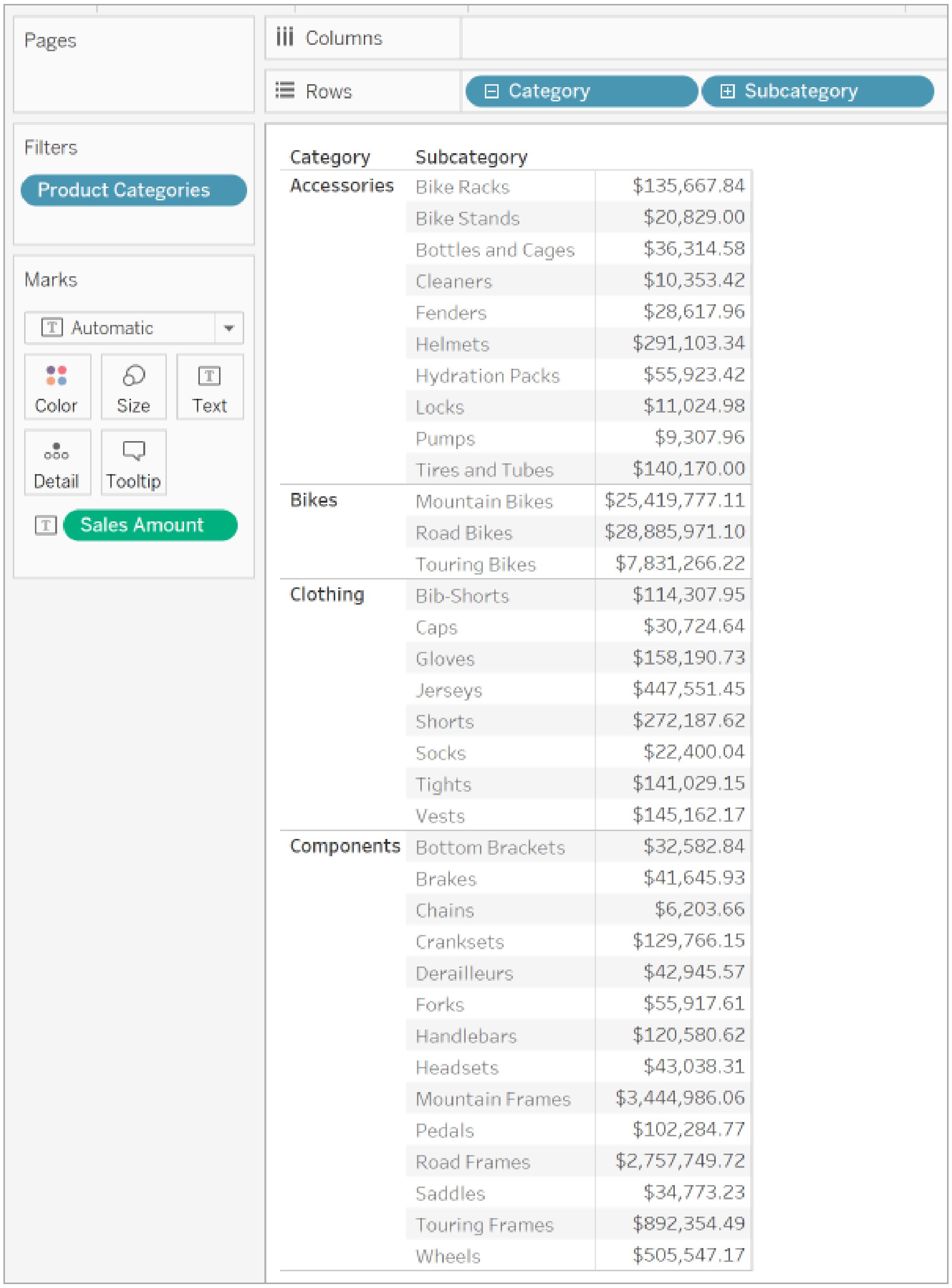
This is often referred to as non-uniform drill down because you expose only the members of interest instead of exposing all the members of a given level.

For example, you can see the difference between drilling down for all dimension members and drilling down for individual dimension members below.

Individual dimension member: Bikes



All dimension members



Category	Subcategory	
Accessories	Bike Racks	$135,667.84
	Bike Stands	$20,829.00
	Bottles and Cages	$36,314.58
	Cleaners	$10,353.42
	Fenders	$28,617.96
	Helmets	$291,103.34
	Hydration Packs	$55,923.42
	Locks	$11,024.98
	Pumps	$9,307.96
	Tires and Tubes	$140,170.00
Bikes	Mountain Bikes	$25,419,777.11
	Road Bikes	$28,885,971.10
	Touring Bikes	$7,831,266.22
Clothing	Bib-Shorts	$114,307.95
	Caps	$30,724.64
	Gloves	$158,190.73
	Jerseys	$447,551.45
	Shorts	$272,187.62
	Socks	$22,400.04
	Tights	$141,029.15
	Vests	$145,162.17
Components	Bottom Brackets	$32,582.84
	Brakes	$41,645.93
	Chains	$6,203.66
	Cranksets	$129,766.15
	Derailleurs	$42,945.57
	Forks	$55,917.61
	Handlebars	$120,580.62
	Headsets	$43,038.31
	Mountain Frames	$3,444,986.06
	Pedals	$102,284.77
	Road Frames	$2,757,749.72
	Saddles	$34,773.23
	Touring Frames	$892,354.49
	Wheels	$505,547.17

One reason to use a non-uniform drill down is if your data source has a ragged hierarchy (asymmetric layout). You also might want to view the children for just the member of interest.

> **Note**: Drilling down and drilling up results in filtering the data.

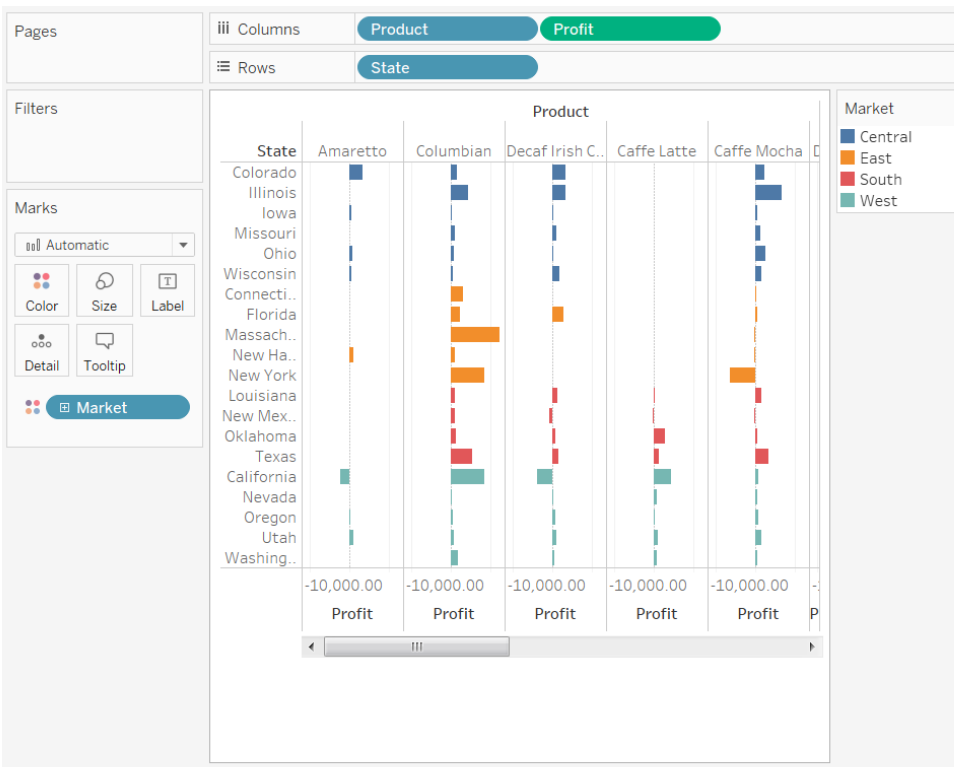
Perfect Pivoting

Note: The following applies only to certain multidimensional data sources.

In Tableau, perfect pivoting refers to working with hierarchies in these ways:

- Using varying levels of detail including skipping levels (for example, Country and City, but not State).
- Using varying levels of detail on different worksheet shelves simultaneously (for example, Product Family on the Columns shelf and Product Department on Color).
- Using varying levels of detail out of order (for example, Quarter before Year).

For example, in the following view the Market hierarchy is broken up to show the State level as Rows and the Region level as Color.



Defining Unique Values

Note: The following applies only to certain multidimensional data sources.

Sometimes, when you are building views in Tableau, a field will have multiple members with the same name. For example, you may have a view showing the average profit by month over several years. The month February appears multiple times (once for each year).

While the name, February, is repeated, each instance of February can either be considered similar or unique. If you consider them similar, they will appear in the same column if you decided to move the Year field to the Color. However, if you consider them unique, they will be treated as two different values.

It is generally okay to consider repeated names within date and time fields (like in the previous example) similar but if there are repeated names in the Customer Name field, you won't want to consider the two customers as the same person.

To define how you want Tableau to determine whether repeated values are unique, right-click (control-click on Mac) on the dimension in the **Data** pane and select one of the following on the **Unique Values** context menu:

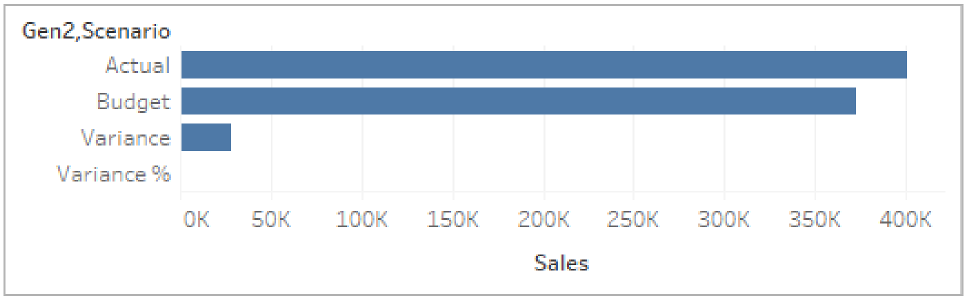
- By Key: each member is considered unique based on the key given to it by the system administrator when the database is set up. Members with the same name but different keys are treated as unique values.
- By Name: each member is considered unique based on the member name. Members with the same name (regardless of their keys) are treated as if they are the same.

By default, unique date and time values are determined by name and all other values are determined by key.

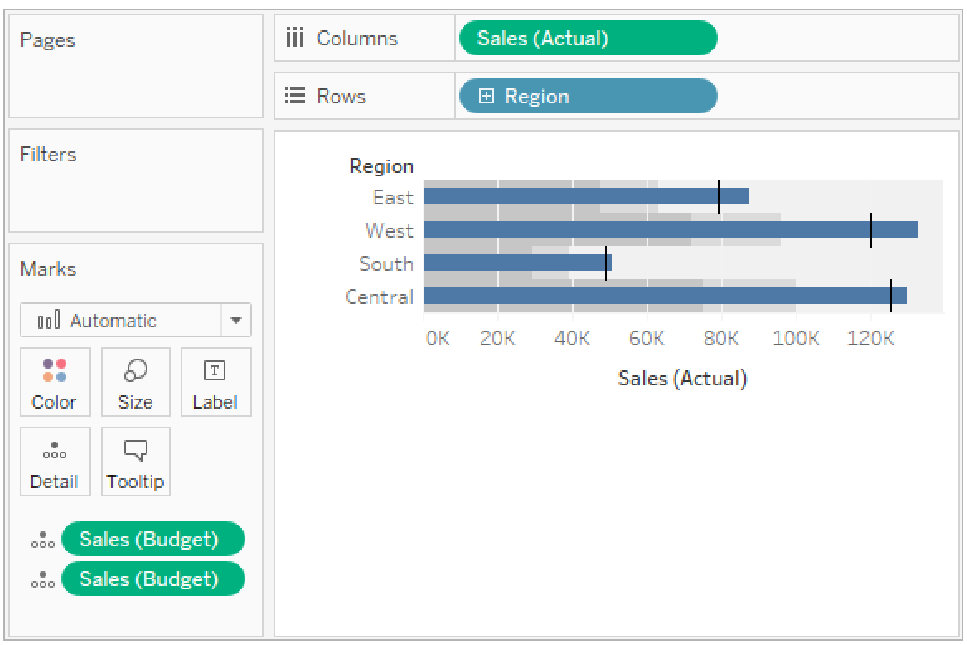
Utility Dimensions

Note: The following applies only to certain multidimensional data sources.

Oracle Essbase databases sometimes have special dimensions used to model comparative values such as Actual vs. Budgeted or Current Year vs. Previous Years. These dimensions are the utility dimensions and are often set up as Scenario or Years. For example, the members of a Scenario dimension are shown below.



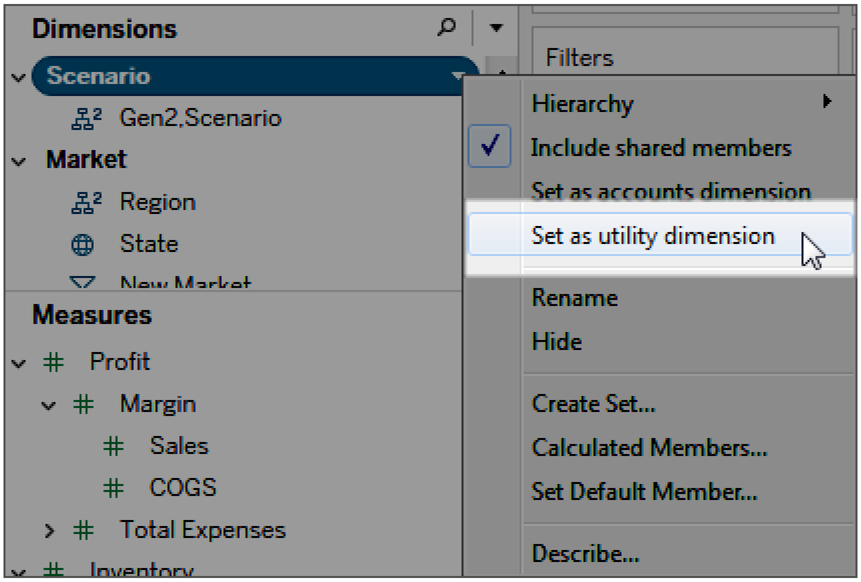
In the above view, you can see Actual Sales, Budgeted Sales, and so on. However, what if you wanted compare Actual Sales to Budgeted Sales in a bullet graph? In that case you need to set the Scenario dimension to be used as the utility dimension. When you set a dimension as the utility dimension you can then specify which member of the utility dimension to use for each measure in the view. For example, below is a bullet graph showing actual sales to budgeted sales by region.



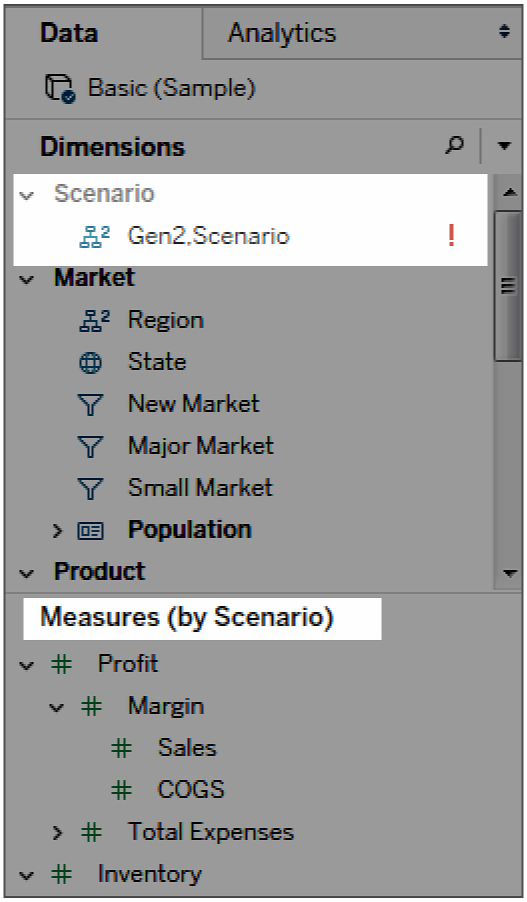
You can see that the Sales measure is used twice in the view: once to show actual and once to show budgeted.

To use a dimension as the utility dimension:

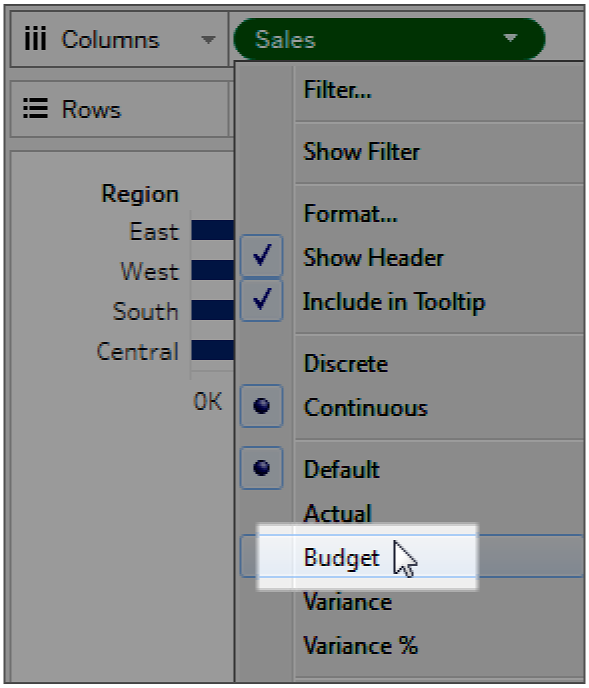
1. Right-click the dimension in the Data pane and select **Set as utility dimension**.



The dimension in the Data pane (in this example, Scenario) can no longer be used as a dimension field in the view. The Measures area of the Data pane indicates that there is a utility dimension (in this example, (by Scenario) indicates the utility dimension).

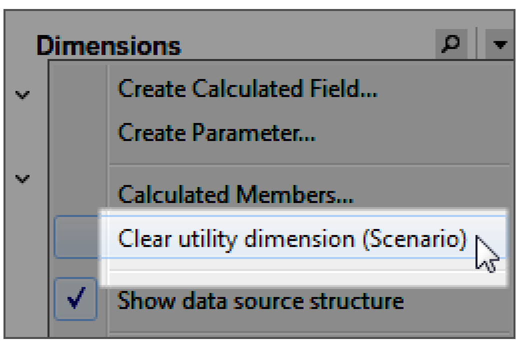


2. Drag a measure to the view.
3. Open the field menu for the measure in the view and select the member of the utility dimension you want to anchor the measure to.



To remove a utility dimension:

Open the drop-down menu at the top of the Data pane and select **Clear utility dimension**.



When you remove the utility dimension, measures that reference it in the view are no longer valid.

Build Charts and Analyze Data

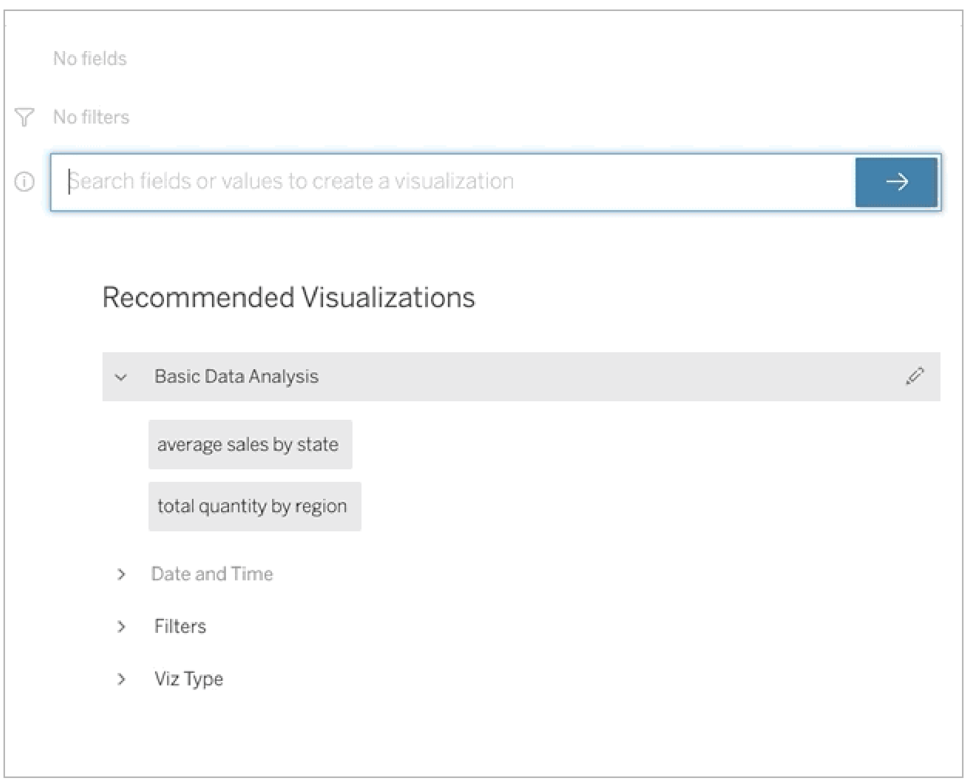
Tableau Desktop is a kind of laboratory in which you can discover the meaning that lies hidden in your data.

In this section you can discover the various features at your disposal as you build views, and learn the basic skills you need to create elegant, insightful views, dashboards, and stories.

Automatically Build Views with Ask Data

Ask Data lets you type a question in common language and instantly get a response right in Tableau. Answers come in the form of automatic data visualizations, with no need to manually drag-and-drop fields or understand the nuances of your data's structure.

Ask Data lets you ask sophisticated questions naturally, with support for key analytical concepts like time series and spatial analysis, and an understanding of conversational phrases like "last year", "earliest", and "most popular".

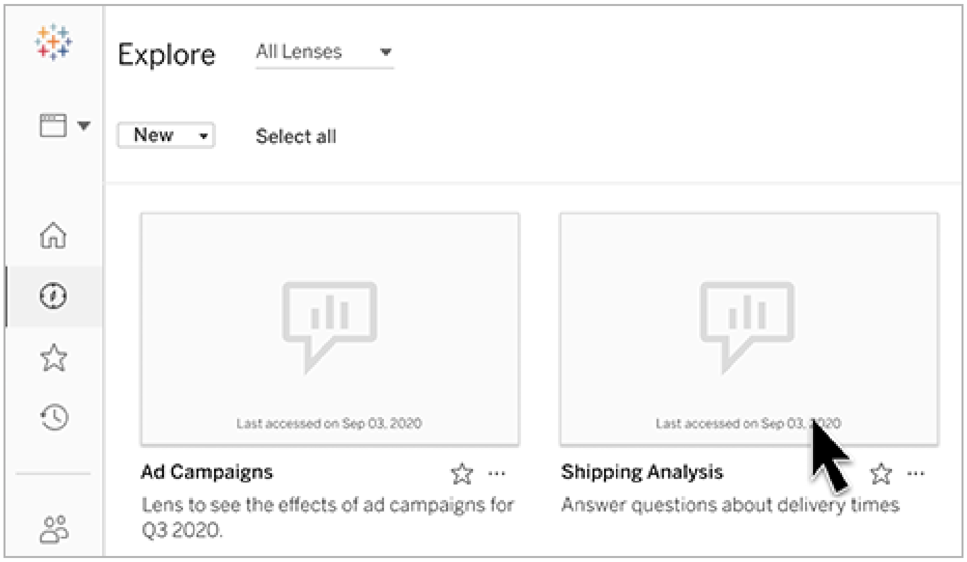


Navigating to Ask Data lenses

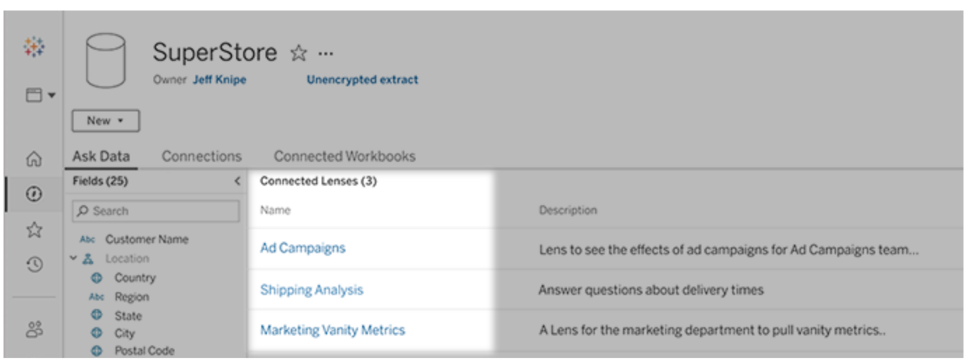
Before you can query a data source with Ask Data, a Tableau author must first create a lens that specifies the subset of data fields the lens uses.

In Tableau, here are all the places where you can access an Ask Data lens:

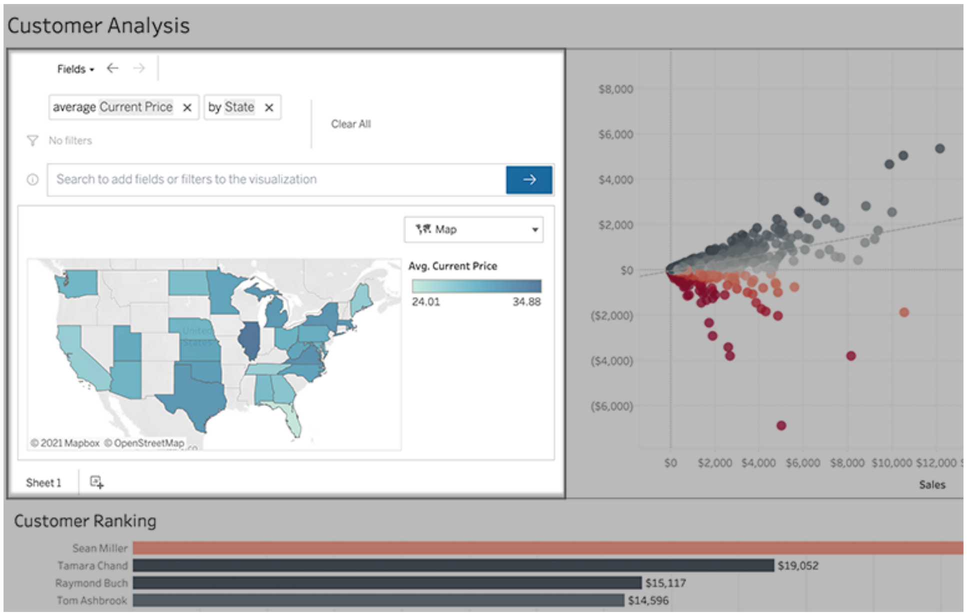
- On the All Lenses page at the top level of your Tableau Online or Tableau Server site.



- On the Ask Data tab for a data source for which lenses have been created.



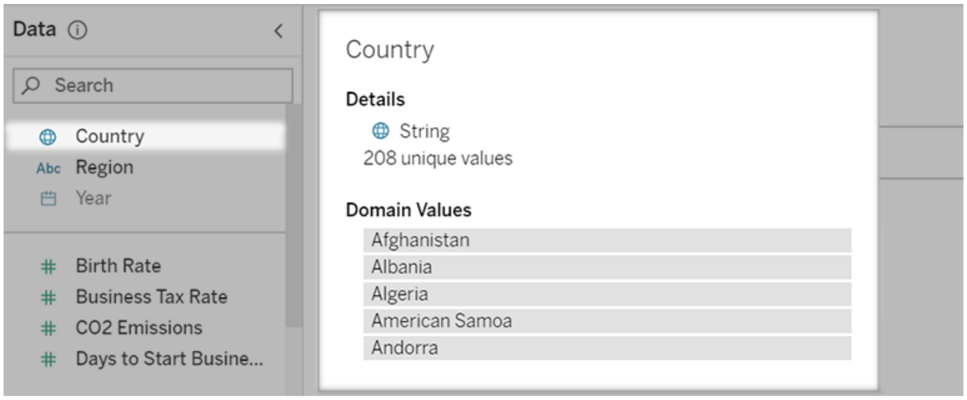
- In an Ask Data object on a dashboard.



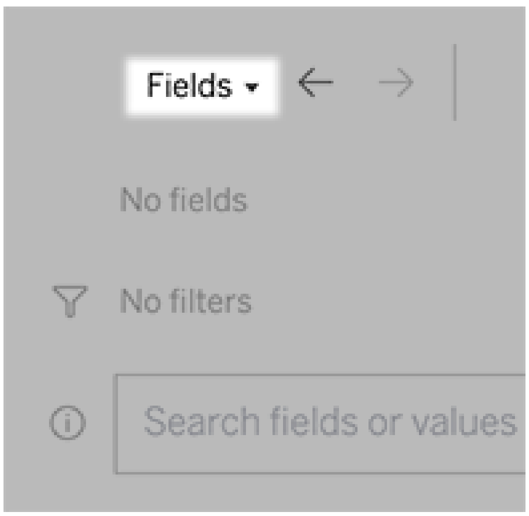
Ask Data from a lens page or dashboard object

1. Navigate to a lens via the All Lenses page for your Tableau site, the Ask Data tab for a data source, or an Ask Data object on a dashboard.
2. (Optional) Under **Recommended Visualizations**, click an entry to quickly see visualizations the lens author has created for your organization.

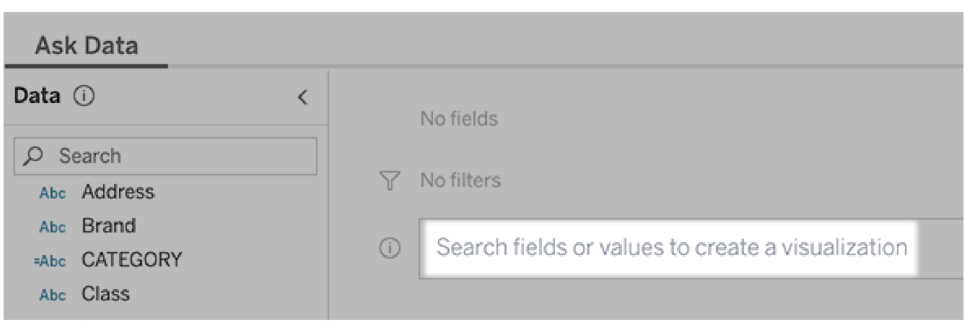
   If a recommendation addresses your data analysis needs, you can stop here. Otherwise, continue on to quickly create your own question.
3. (Optional) In the Data pane at left, briefly hover over each field to learn more about the data it contains.



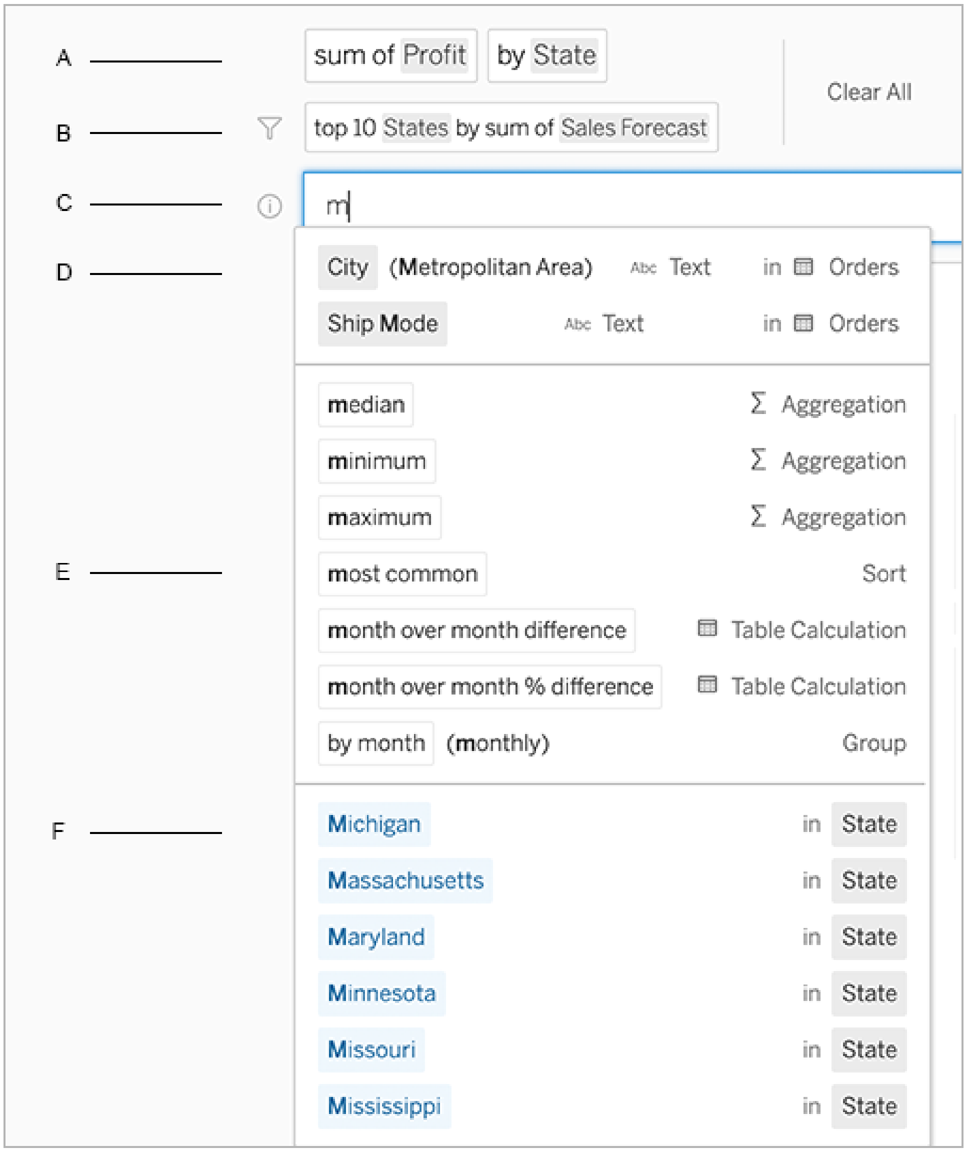
In a narrower dashboard object, the Data pane may be hidden, but you can see the same information by clicking the **Fields** drop-down menu.



4. In the box reading "**Search fields or values to create a visualization**", enter your query about the data.



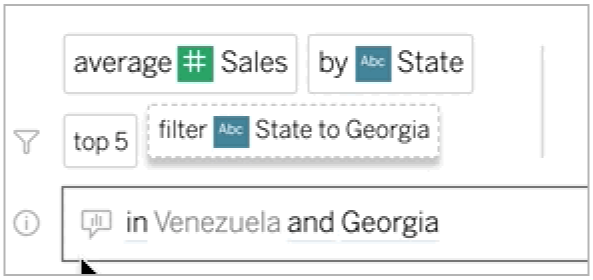
5. As you type, Ask Data searches data fields, functions, and string values and displays results in a drop-down list. Click items in the list to add them to your current entry, shown above the search box. To automatically create a viz using the current entry, press **Enter** at any time.



Modifying a query by searching for fields and analytical functions

A. Current entry B. Current filters C. Search box D. Returned fields E. Returned analytical functions F. Returned field values

> **Tip:** (Tableau Online only) Hover over elements in the text box to see how they're incorporated in the interpretation above. Words that aren't used are grayed out, helping you rephrase your query in a way that's clearer to Ask Data.



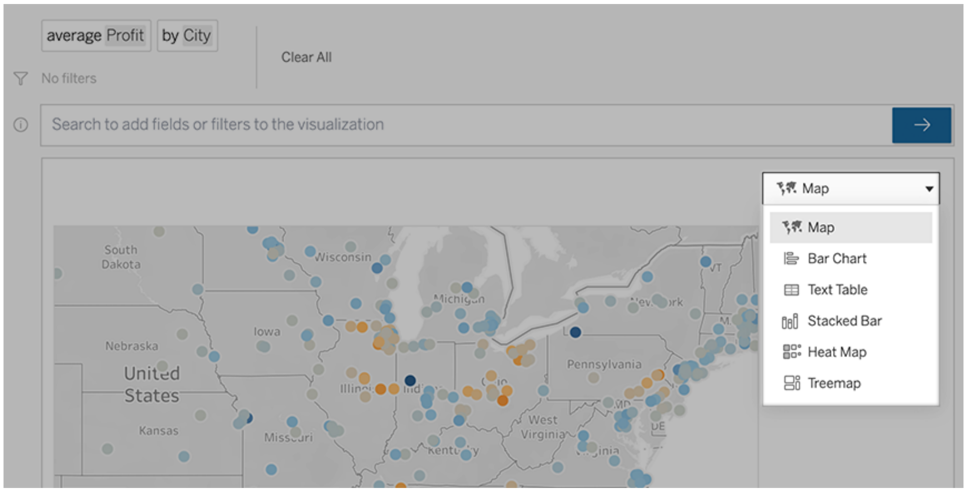
Rephrase your question

You can rephrase questions by clicking options, data fields, and filters in the user interface.

Change the viz type

If the default viz doesn't fully reveal your data, click the menu at upper right, and choose from these supported viz types:

- Bar Chart
- Gantt Bar
- Heat Map
- Histogram
- Line Chart
- Map
- Pie Chart
- Scatter
- Stacked Bar Chart
- Text Table
- Treemap



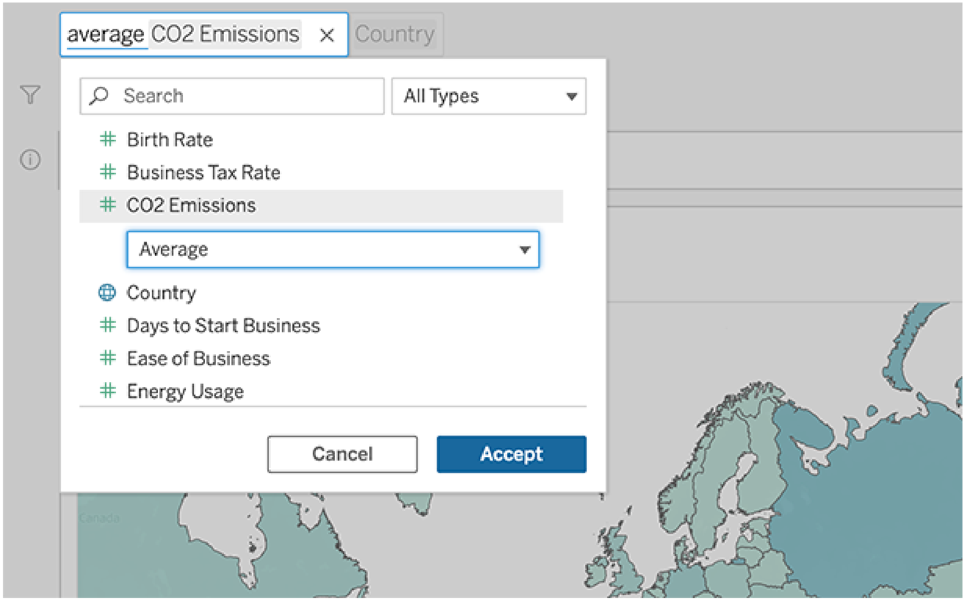
Note: To automatically create certain viz types, Ask Data sometimes adds fields like "Number of Records" to your entries.

Change fields, filters, and displayed data

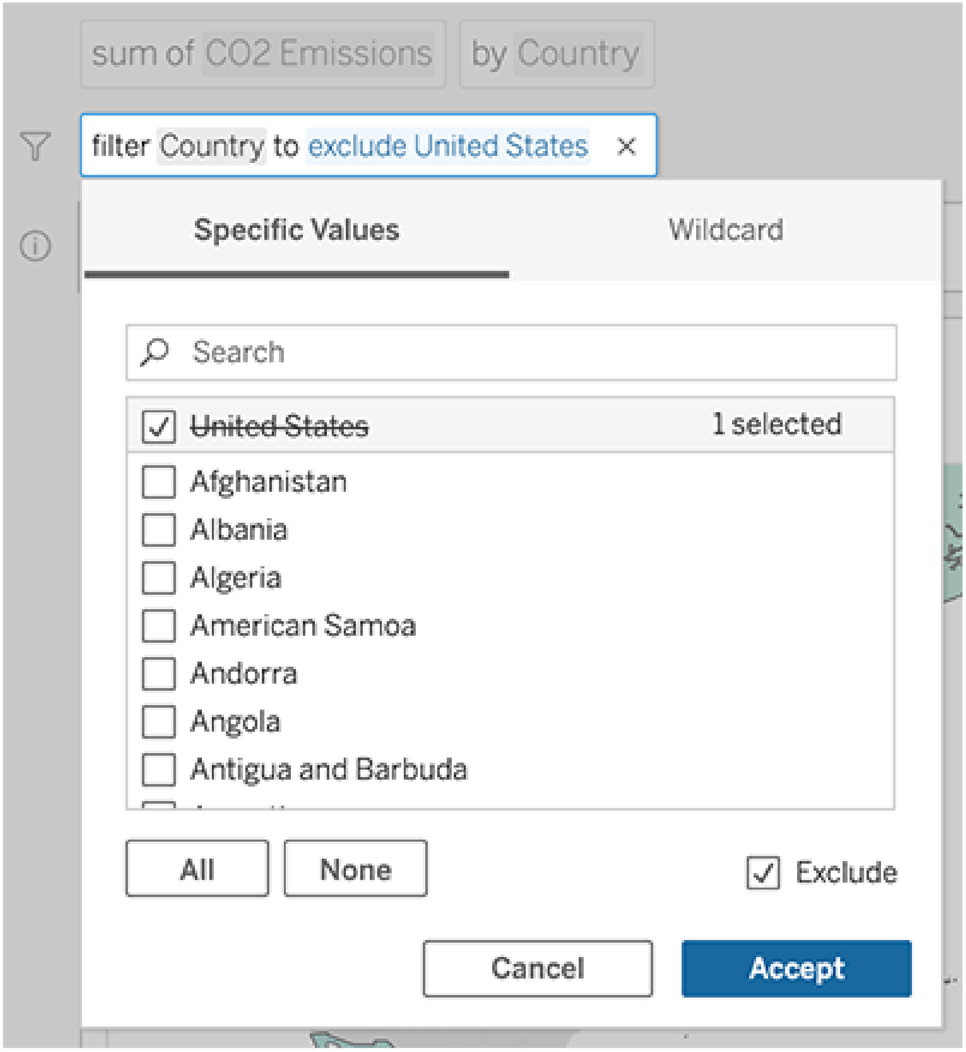
Ask Data gives you several ways to fine-tune how field values are displayed.

- To switch the fields used for the vertical and horizontal axes, click the Swap Axes button  to the left of the viz selection menu:

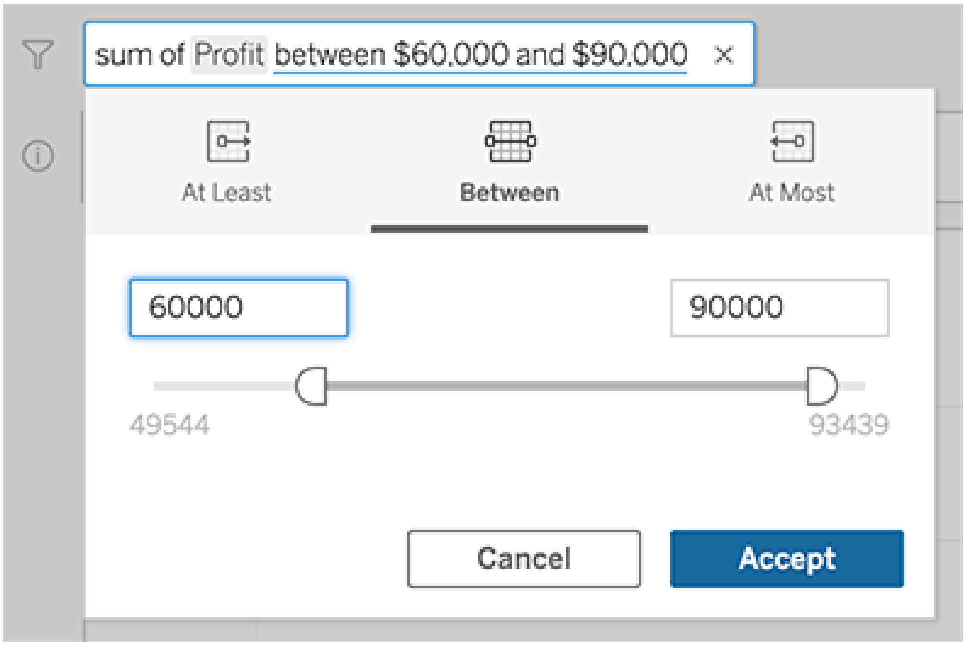
- To change a field or its aggregation or grouping type (for example, from sum to average), click the field name in the text box.



- For categorical filters, click values (like "exclude United States" in the example below) to change specific values or enter wildcard parameters.


sum of CO2 Emissions
by Country
filter Country to exclude United States
Specific Values
Wildcard
Search
United States
1 selected
Afghanistan
Albania
Algeria
American Samoa
Andorra
Angola
Antigua and Barbuda
All
None
Exclude
Cancel
Accept

- To adjust a numeric range, click words like “high”, “between”, or “cheap”.



- To delete a field or filter, hover over it and click the **X**.

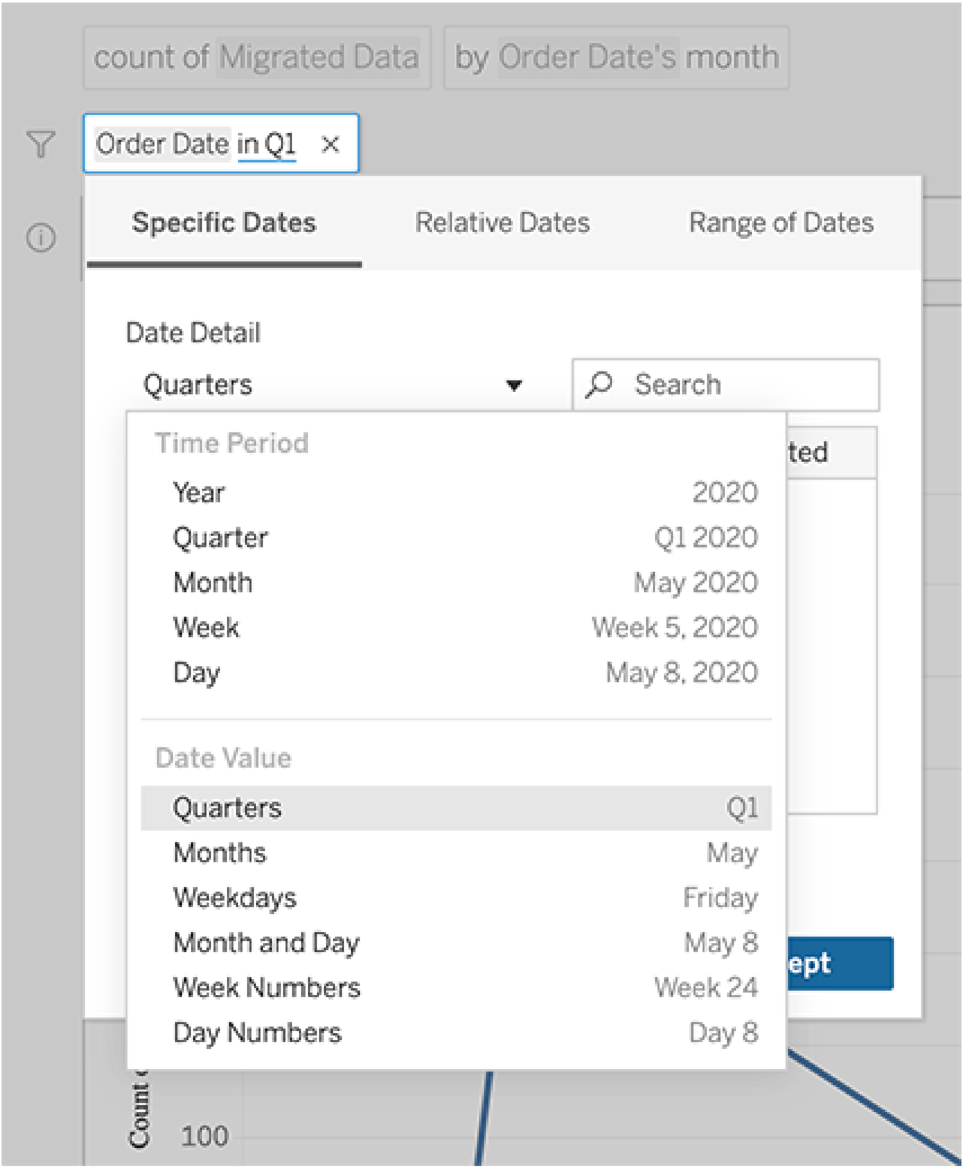
Adjust date filters

To adjust a date filter, click words like “in”, “last”, “previous”, or “between”. Then click one of the following:

- **Specific Dates** to enter a specific time period or date value
- **Relative Dates** to show a date range relative to the present day
- **Range of Dates** to enter specific start and end points

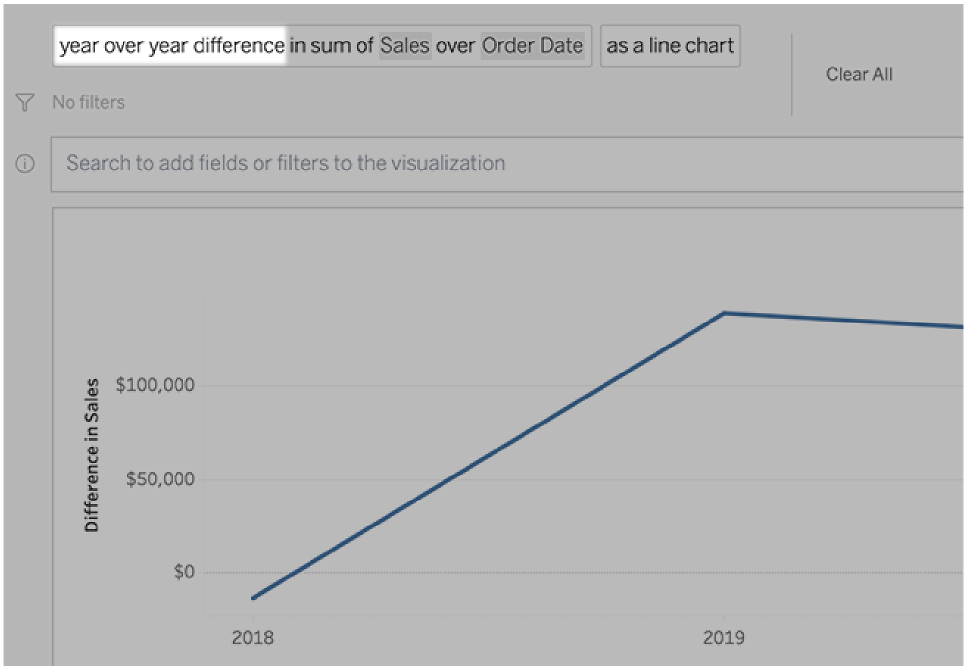
Specific Dates offers some unique options in the **Date Detail** menu:

- **Time Period** options show a single continuous date range
- **Date Value** options show ranges that can repeat in multiple time periods. For example, to see combined sales performance for Q1 across multiple years, under Date Value, you would choose Quarters.

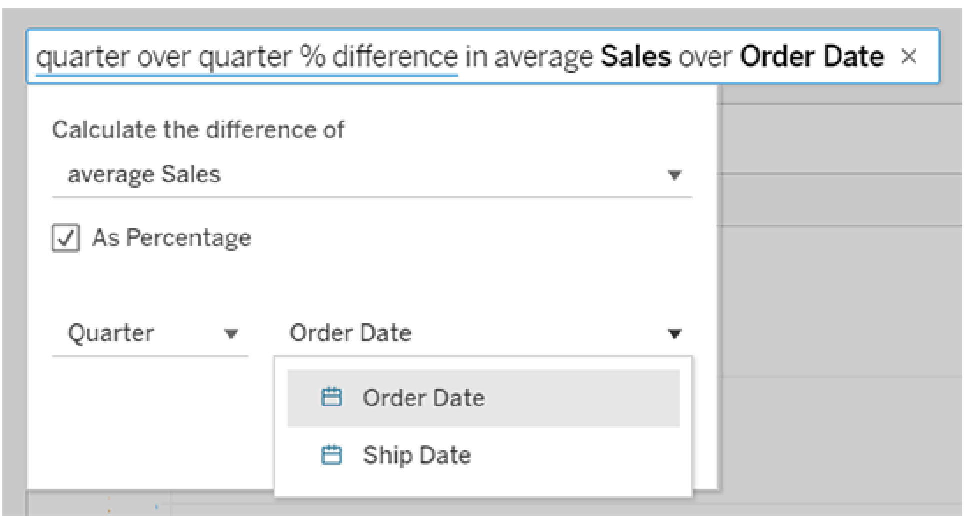


Compare differences over time

Ask Data lets you compare time periods with phrases like “year over year”, “quarter over quarter”, or similar comparisons by month, week, or day. The results appear as difference or percent difference table calculations in workbooks you save from Ask Data.



In the text box, click a difference calculation to choose other fields, aggregation methods, and time periods.



Apply simple calculations

Ask Data supports simple calculations between two measures, which you can apply using these symbols:

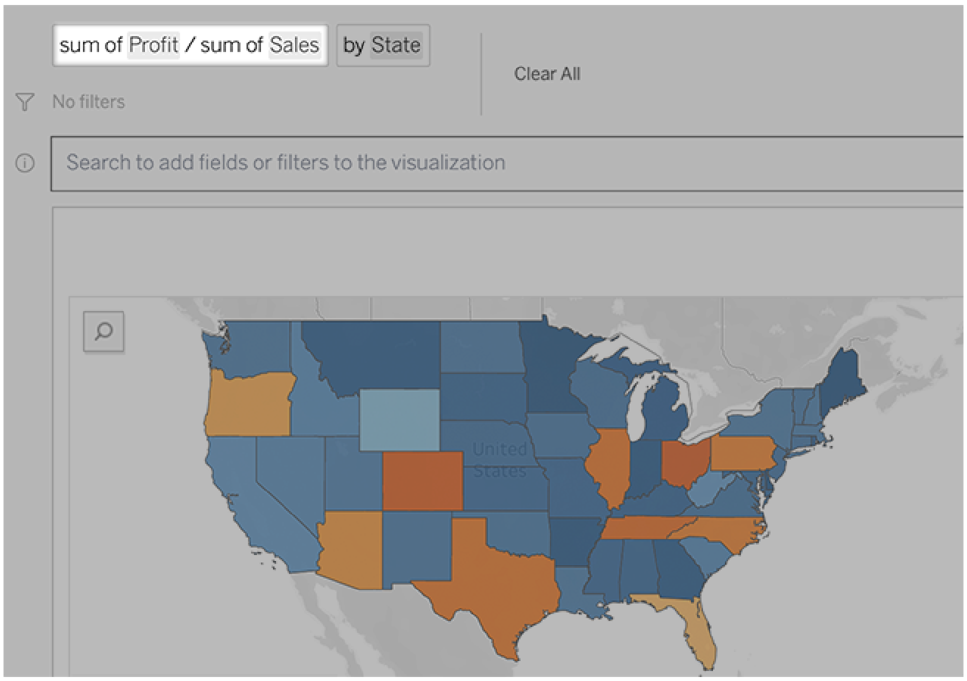
+ sums the measures

- produces the difference between them

***** multiplies

/ divides

In workbooks you save from Ask Data, these calculations don't become calculated fields but instead ad hoc calculations on the Columns, Rows, or Marks shelves. For more information, see **Ad-Hoc Calculations** on page 2497.

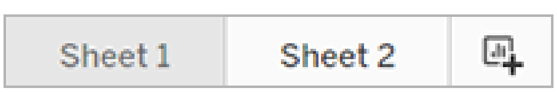


Add sheets with other vizzes

To quickly create multiple different vizzes from a lens, add sheets in Ask Data.

At the bottom of the web page, do any of the following:

- Click the **Add Sheet** icon to the right of named sheets.



- Right-click a sheet name, and choose either **Duplicate** or **Delete**.

  (To rename sheets from Ask Data, you need to save them in a new workbook.)

Share links to Ask Data vizzes

You can quickly share Ask Data vizzes. Simply click the link icon above the query box, and copy a link viewable by anyone with access to the lens.

Tips for success

As you structure questions for Ask Data, apply these tips to get better results.

- **Use keywords** — For example, instead of “I want to see all the countries that these airports are in”, try “by airport and country”.
- **Use exact wording for field names and values** — For example, if your lens includes Airport Code, Airport Name, and Airport Region fields, specify those by name.
- **See a ranked list** — Ask Data maps terms like “best” and “worst” to Top 1 and Bottom 1, respectively. If you want to see broader rankings, use “high” and “low” instead. For example, enter “houses with low sale prices”.
- **Surround unusually long values with quotation marks** — To analyze long field values that contain line returns, tabs, more than five words, or more than 50 characters, surround them with quotation marks. To improve performance, Ask Data doesn't index fields of that length.

Embed and Save Vizzes from Ask Data

If you're a Tableau Creator, after you create vizzes with Ask Data, you can embed and save them.

Embed an Ask Data lens in a web page

After you ask a question with an Ask Data lens, an **Embed Code** icon appears in the toolbar. Click it to copy code you can incorporate into any web page.

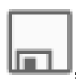
When web pages with a lens embedded in them load, the lens opens without a viz, waiting for users to ask a question.

To see a lens embedded in a web page, users must have a Tableau license and permission to connect to the lens. They can interact with Ask Data just as they would in the Tableau Online or Tableau Server interface, but they can't save resulting vizzes.

Save vizzes in a new workbook

Save vizzes from Ask Data to combine them with other Tableau views and dashboards in a workbook, where you can access Tableau's full range of data analysis and visualization tools.

Tip: To incorporate an interactive Ask Data experience into workbooks instead of a finished viz, see **Add an Ask Data lens to a dashboard** on page 1129.

1. Above the viz, click the save icon , and choose **Save as** if you're saving the viz for the first time, or **Save** to overwrite a previously saved version.
2. If you chose **Save as**, select **Embed password for data source** if people who lack access will be viewing the viz.

   To quickly navigate to the saved workbook, click the save icon again, and choose **Edit workbook**.

Supported Analytical Functions for Ask Data

Ask Data supports the core analytical functions of Tableau, including aggregations, grouping, sorting, and numeric and categorical filtering.

Note: Ask Data requires English analytical phrases but works great with non-English data. For example, you could ask “how many pedidos where fruta is manzanas” if you have Spanish data about orders of fruit for a grocery.

Aggregations

When using Ask Data, example aggregation phrases include “Sum of Sales”, “Average Profit”, or “Count of Customers”. You can also use common synonyms for these terms, such as “mean” for average or “cnt” for count.

- Sum
- Average
- Median
- Count
- Distinct count
- Minimum
- Maximum

Grouping

Grouping phrases include the word “by”, with examples like “by Region” and “by Sales”.

This expression is equivalent to placing a dimension or unaggregated measure on the Columns or Rows shelf while authoring a standard Tableau viz.

Sorting

Examples of sorting phrases include “sort Products in ascending order by sum of Profit” or “sort Customer Name in alphabetical order”. You can also use synonyms like “from smallest” for ascending order and “from largest” for descending.

- Ascending
- Descending
- Alphabetical

Numerical filters

Ask Data lets you filter data down to the numerical range you're interested in and then fine-tune that range as needed. Example phrasing includes “sum of Sales at least $2,000” or “prices between $10 and $20”.

- At least
- At most
- Between

Numerical limits (top and bottom values)

You can quickly see the top and bottom values of a field with phrases like “top 10 cars sold last year” or “bottom Region by average Profit”.

- Top
- Bottom

Ask Data also supports common terms that people tend to use with their data, like “low”, “high”, “lowest”, and “highest”. With currency-based fields, you can use synonyms for these terms, like “cheap” for low and “expensive” for high (for example, “show me the cheapest houses in Ballard”).

Categorical filters for text values

Categorical filters let you narrow data down to the specific text values you're interested in. Examples for each type of filter appear below:

- Is (for example, State is “Ohio”)
- Is not (State is not “Ohio”)
- Starts with (State starts with “O”)
- Ends with (State ends with “O”)
- Contains (State contains “O”)
- Does not contain (State does not contain “hi”)

Time filtering

Ask Data supports time filtering with phrases like “Sales in the last 10 years” or “Profit between 2012 and 2014”. You can also use synonyms of phrases below, like “starts in” instead of “starting at”.

- In (followed by specific date)
- Previous
- Last

- Following
- Next
- Between
- Starting at
- Ending at

Tip: You can also compare differences over time with table calculations.

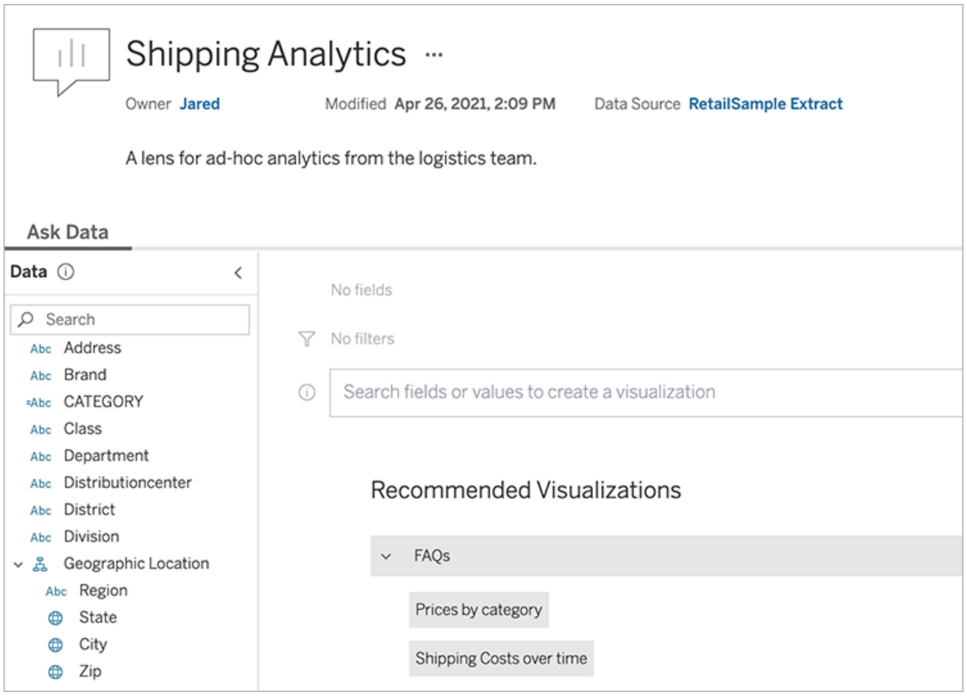
Create Lenses that Focus Ask Data for Specific Audiences

Most people don't need information from an entire data source but instead want data visualizations relevant to their job function, like sales, marketing, or support. To optimize Ask Data for different audiences like these, Tableau authors create separate Ask Data *lenses*, which query a selected subset of fields. For the selected fields, authors can specify synonyms for field names and values, reflecting terms the lens audience uses in common language (for example, "SF" for "San Francisco"). Lens authors then customize the recommended visualizations that appear below the Ask Data query box, which provide answers to users with a single click.

Note: Ask Data lenses can be created only for data sources published separately to a Tableau site, not data sources embedded in workbooks.

Create or configure a lens page on your Tableau site

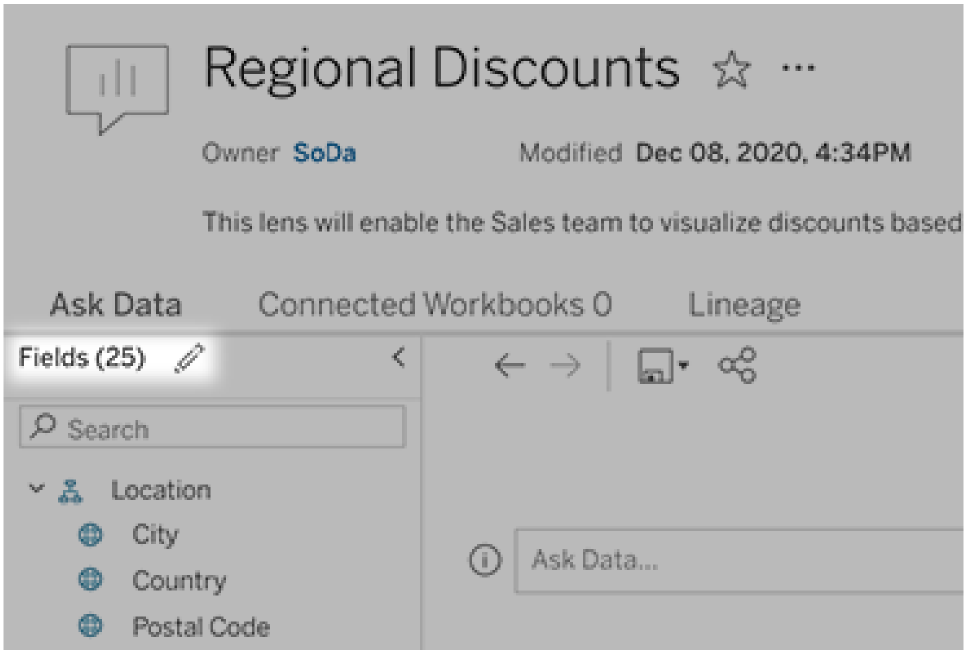
On your Tableau site, each lens has a separate page where users can query Ask Data and authors can configure lens fields, synonyms, and suggested questions.



A lens page on a Tableau site

1. To create a lens page on your Tableau site, go to a data source page, and choose **New** > **Ask Data Lens**.

   To configure an existing lens, go to the lens page on your site. (From an Ask Data object in a dashboard, you can click the pop-up menu in the upper corner and choose **Go to Lens Page**.)
2. If you're creating a new lens, enter a name, description, and project location, and then click **Publish Lens**.
3. At the top of the Fields pane at left, click the pencil icon. Then select the relevant fields for lens users, and click **Save**.



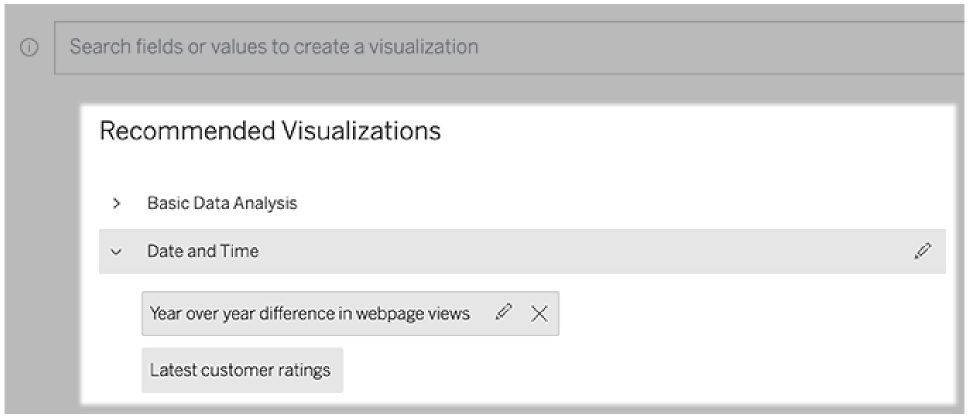
4. At left, hover over individual fields, and click the pencil icon to edit either of the following:
   - Common synonyms for field names and values that lens users may enter in their queries. For detailed instructions, see Add Synonyms for Ask Data.
   - (Tableau Online only) Descriptions that appear when users hover over fields

5. To edit recommended visualizations below the query box so they address common queries from lens users, do the following:

   - To edit a section title, click the pencil icon  to the right of the title.
   - To add a recommendation to a section, click the plus sign (+) to the right of the title. Then enter Text to Display and the related query in the Matching Result field.

   Tip: Ask Data creates a viz for the matching result when users either click the displayed text or enter it as a query.

- To change an existing recommendation, hover over it and click the pencil icon . To remove a question, click the X.

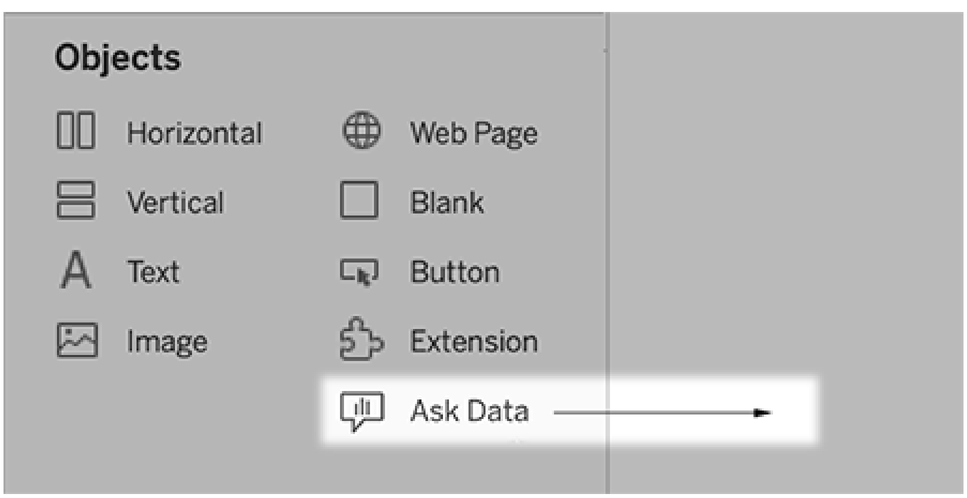


Add an Ask Data lens to a dashboard

On a dashboard, you can add an Ask Data object that lets users query a published data source via a lens on your Tableau site.

1. While editing a dashboard in Tableau Online, or Tableau Server, drag the Ask Data object to the canvas.

   Note: In Tableau Desktop, you can also drag an Ask Data object to the canvas for placement purposes. But to select a lens, you will need publish to Tableau Online or Tableau Server and edit the object there.

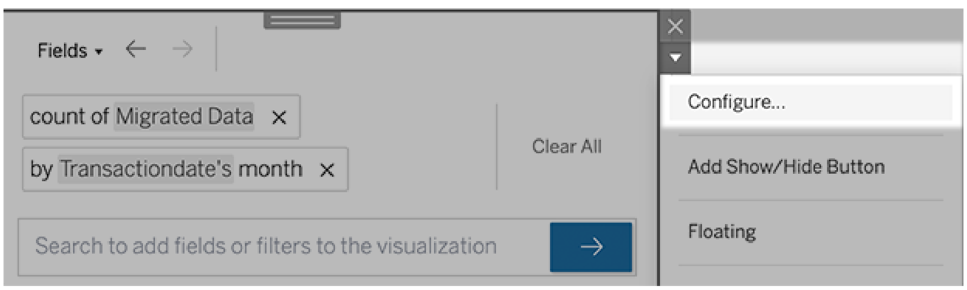
   

2. Select a published data source previously connected to the workbook.
3. To use an existing lens, select it, and click **Use Lens**. Then skip remaining steps.

   Or, to create a new lens, do one of the following:
   - If there are no lenses for the data source, click **Go to Data Source Page**.
   - If lenses already exist, click the data source name at the bottom of the dialog.
4. Complete the steps in **Create or configure a lens page on your Tableau site** on page 1126.
5. After you finish creating the lens, return to the Lens object in your dashboard, and click **Refresh**. Then select the new lens, and click **Use Lens**.

Apply a different lens to an Ask Data dashboard object

1. From the pop-up menu at the top of the object, choose **Configure**.



2. Go to **Add an Ask Data lens to a dashboard** on the previous page, and repeat steps 2 onward.

Change a lens name, description, or project location

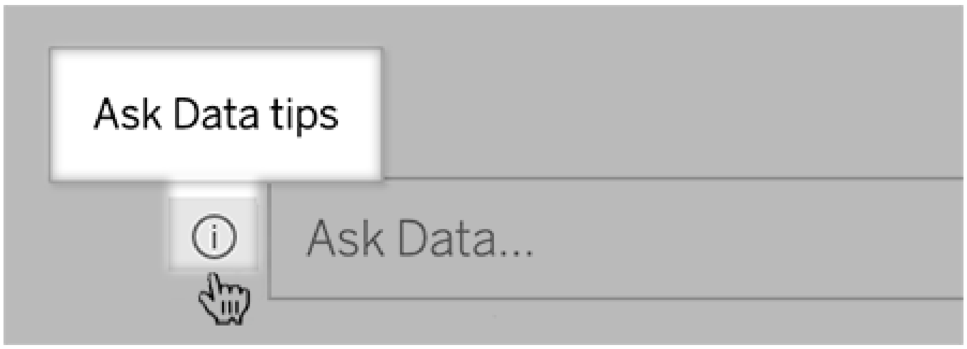
1. Navigate to the lens page on Tableau Online or Tableau Server.
2. To the right of the lens name at the top of the page, click the three dots (...), and choose **Edit Workbook**.
3. Click **Edit Lens Details**.

See how people use Ask Data with a lens

For data source owners and lens authors, Ask Data provides a dashboard that reveals the most popular queries and fields, the number of visualization results that users clicked, and other helpful information. Filters let you narrow data down to specific users and time ranges. These stats help you further optimize a lens to increase the success of your users.

Note: If you use Tableau Server, you can access this data in the Tableau Server Repository to create custom dashboards.

1. In Tableau Server or Tableau Online, navigate to a lens page.
2. To the left of the Ask Data text box, click the "i".



3. In the lower-left corner of the tips dialog, click **Usage Analytics**.

Permissions for publishing and viewing lenses

For Ask Data objects in dashboards, no change to permissions should be required: by default, existing workbook authors can create lenses, and existing dashboard audiences can view them. But for reference, here's a detailed outline of required lens permissions for both dashboards and direct access via a data source page.

To create and publish a lens, a user needs:

- The Creator or Explorer user role
- Lens Creation permission for the data source (inherited by default from the Connect permission)
- Write permission for the parent project to which the lens is published

To access and interact with a published lens, a user needs:

- The Viewer role or above
- Connect permission for the data source
- View permission for the lens

Note: By default, lens permissions like View reflect a project's permissions for workbooks. If Tableau administrators want to change default lens permissions, they can do so either individually for each project, or in bulk using the permissions API.

Optimize Data for Ask Data

If you manage and publish data sources, here are some tips to help make users of Ask Data more successful. By spending a little extra time on this process, you'll open up data analysis to a wider range of people at your organization, helping them independently answer questions and gain deeper insights.

Optimize data in Ask Data

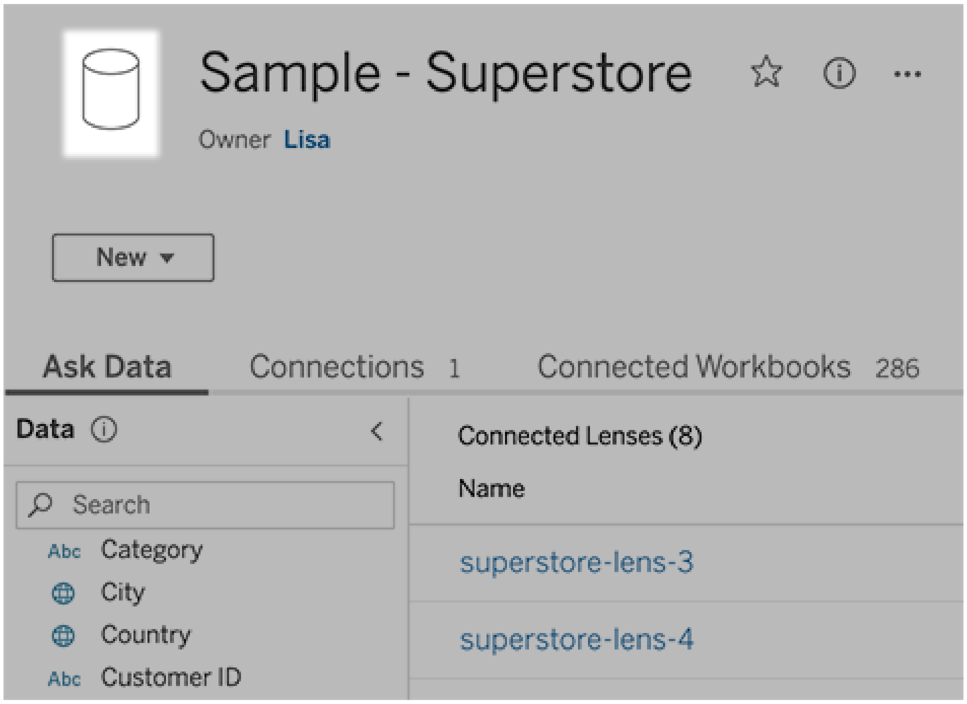
In the Data pane on the left of the Ask Data interface, data source owners can add synonyms for fields and exclude irrelevant values.

Changing settings at the data source or lens level

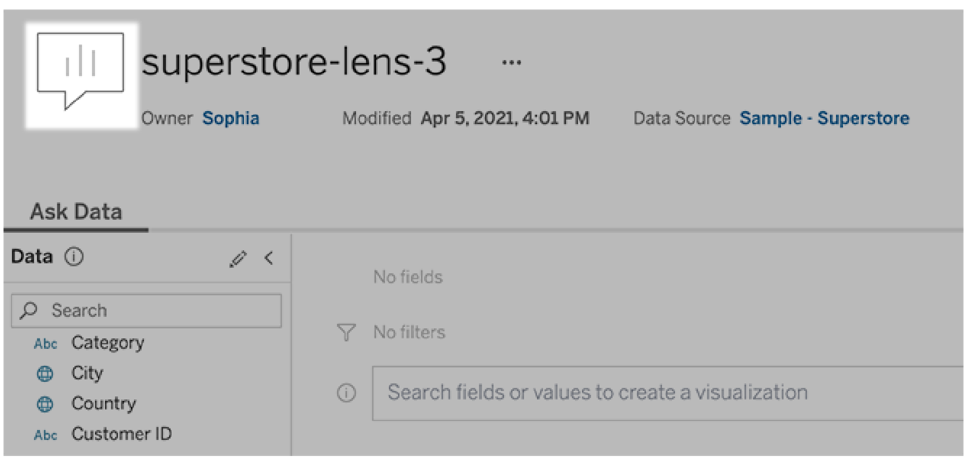
When changing settings in the Data pane for Ask Data, pay close attention to whether you're at the data source or lens level. (For more information, see Create Lenses that Focus Ask Data for Specific Audiences.)

- At the data source level, you'll see the cylindrical data source icon in the upper left corner. Here, changes you make in the Data pane will apply by default to all subsequently created lenses.

  > **Note:** For extracts, two cylinders will appear.



- For an individual lens, you'll see the quotation icon in the upper left corner. Here, changes you make in the Data pane will apply to this lens alone.



Add synonyms for field names and values

People may not use the same terminology found in your data source, so data source owners and Tableau administrators can add synonyms for specific data field names and values. Synonyms you enter are available throughout your organization, making data analysis quicker and easier for everyone.

Exclude values of specific fields from search results

To improve the usability of search results in Ask Data, you can exclude the values of specific fields from indexing. Though Ask Data doesn't add non-indexed values to search results, the values still appear in visualization results when relevant. For example, if you don't index values from a "Product" field because they add unnecessary detail to search results, Ask Data can still display values such as "iPhone 12" in resulting data visualizations. And users can manually add non-indexed values to queries by surrounding them with quotation marks (for example, "Sales for Product containing "iPhone 12"").

Note: This field-level setting is ignored if the value indexing setting for the data source is set to Disabled. Field names and related synonyms are always indexed.

1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data source field at left, and click the **Edit Field Details** icon (the pencil).

3. Deselect **Index field values.**

   Either click the text box that appears to reindex the data source now, or let it reindex based on its regular indexing schedule.

Optimize data sources

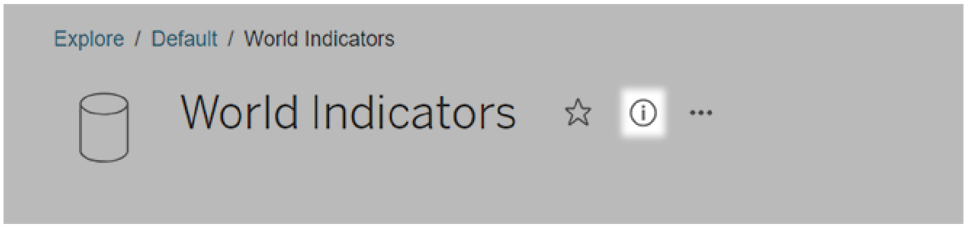
To create the best experience for Ask Data users, optimize the original data source.

Note: Ask Data doesn't support multidimensional cube data sources, or non-relational data sources like Google Analytics.

Optimize indexing for Ask Data

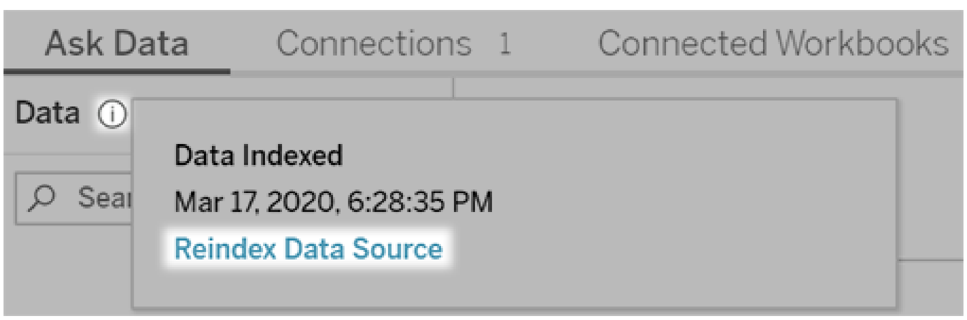
Data source owners can change how often field values are indexed for Ask Data, optimizing system performance.

1. At the top of a data source page, click the Details icon:



2. In the Ask Data section, click **Edit**.
3. Choose an indexing option for field values:
   - **Automatic** checks for changes every 24 hours and analyzes the data source if it is live, has had an extract refreshed, or has been republished. Choose this option for a data source frequently used with Ask Data, so it will be ready before users query it.
   - **Manual** analyzes the data source only when Tableau creators manually trigger indexing on the data source page. Choose this option if the data source changes frequently but users query it with Ask Data only occasionally.

     To trigger manual indexing, go to the data source page, click the circled “i” in the Data pane at left, and then click **Reindex Data Source**.



   - **Disabled** analyzes only field names, not values.
4. Click **Save**

Use data extracts for faster performance

For improved performance and support for large data sets, use Ask Data with published extracts rather than live data sources. For more information, see Create an extract.

Ensure that users can access the data source

To use Ask Data, users must have permission to connect to the individual data source. If a data source has row-level permissions, those permissions also apply to Ask Data, which won't recognize secure values or make related statistical recommendations.

Be aware of unsupported data source features

Ask Data supports all Tableau data source features except the following. If your data source contains these, Ask Data users won't be able to query related fields.

- Sets
- Combined fields
- Parameters
- Table calculations in calculated fields

Anticipate user questions

Anticipate the kinds of questions your users will ask, and then optimize your data source for those questions using these techniques:

- Clean and shape data in Tableau Prep or a similar tool.
- Join data to include all fields users may have questions about in one table, improving performance.
- Add calculated fields that answer common user questions.
- Create bins with appropriate sizes for quantitative variables that users are likely to want to see as a histogram or another binned form.

Simplify the data

To make data easier to understand by both users and Ask Data, simplify the data source as much as possible during the data prep process.

1. Remove any unnecessary fields to improve performance.
2. Give each field a unique and meaningful name.
    - For example, if there are five field names that start with "Sales …", better distinguish them so Ask Data can properly interpret the term "sales".
    - Rename "Number of records" to something more meaningful. For example, use "Number of earthquakes" in a data source where each record is an earthquake.
    - Avoid field names that are numbers, dates, or boolean ("true" or "false") values.
    - Avoid names which resemble analytical expressions such as "Sales in 2015" or "Average Products Sold".

3. Create meaningful aliases for field values, reflecting terms people would use in conversation.

Set appropriate field defaults

To help Ask Data analyze data correctly, ensure that default field settings reflect the content of each field.

- Set data types for text, time, date, geographic, and other values.
- Assign the proper data role: dimension or measure, continuous or discrete.
- For each measure, assign appropriate default settings in Tableau Desktop, such as color, sort order, number format (percentage, currency, etc.), and aggregation function. For example, SUM may be appropriate for “Sales”, but AVERAGE might be a better default for “Test Score”.

> **Tip:** It's particularly helpful to set a default comment for each field, because these comments appear as informative descriptions when users hover over fields in Ask Data.

Create hierarchies for geographic and categorical fields

For time data, Tableau automatically creates hierarchies, which let users quickly drill up and down in vizzes (for example, from day to week to month). For geographic and categorical data, however, we recommend that you create custom hierarchies to help Ask Data produce visualizations that reflect the relationships between fields. Be aware that Ask Data won't show the hierarchies in the data pane.

Ask Data doesn't index hierarchy names, only names of fields within hierarchies. For example, if a geographic hierarchy named “Location” contains “Country” and “City” fields, users should enter “Country” and “City” in their questions for Ask Data.

Avoid field names that Ask Data uses for analytical functions

To let users ask questions about data in a conversational style, Ask Data reserves the following words for analytical functions. If any of these words appear as field names in your data, we recommend that you revise them.

> **Note:** In addition to the terms listed below, Ask Data reserves all conjunctions, prepositions, and visualization types such as “Bar chart”.

Aggregation terms

- aggregate
- average
- avg
- maximum
- max
- mean
- median
- minimum
- min
- most
- sum
- count
- distinct count

Grouping terms

- group
- by
- per

Filtering and limiting terms

- filter
- limit
- at least
- at most
- between
- top
- bottom
- contains
- starts with
- ends with
- include

Sorting terms

- sort
- alphabetical

- ascending
- descending

Time terms

- date
- second
- minute
- hour
- day
- week
- month
- quarter
- year
- second over second (sos)
- minute over minute
- hour over hour (hoh)
- day over day (dod)
- week over week (wow)
- month over month (mom)
- quarter over quarter (qoq)
- year over year (yoy)
- today
- tomorrow
- yesterday
- next
- last

Qualifying terms

- best
- expensive
- most expensive
- cheap
- cheapest
- big
- small

Add Synonyms for Ask Data

People may not use the same terminology found in your data source, so data source owners and Tableau administrators can add synonyms for specific data field names and values. Synonyms you enter are available throughout your organization, making data analysis quicker and easier for everyone.

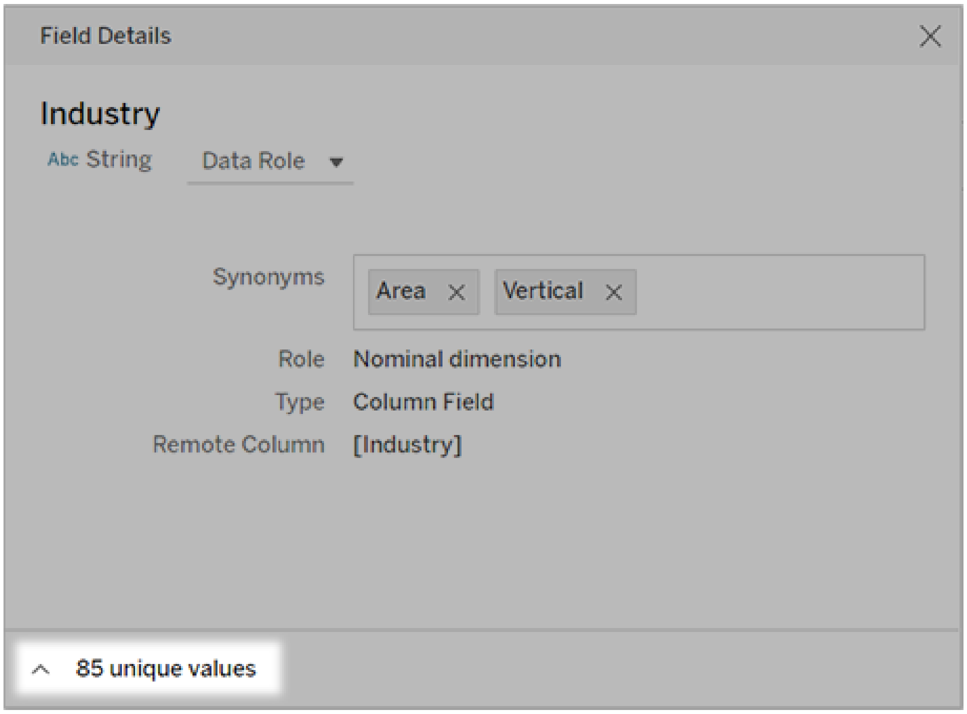
> **Note:** You can't use synonyms that reflect standard analytical terms like “by” or “max”. Likewise, you can't create synonyms for these special types of data: sets, bins, clusters, hierarchies, combined fields, table calculations, or latitude and longitude.

Add field and value synonyms for a specific data source or lens

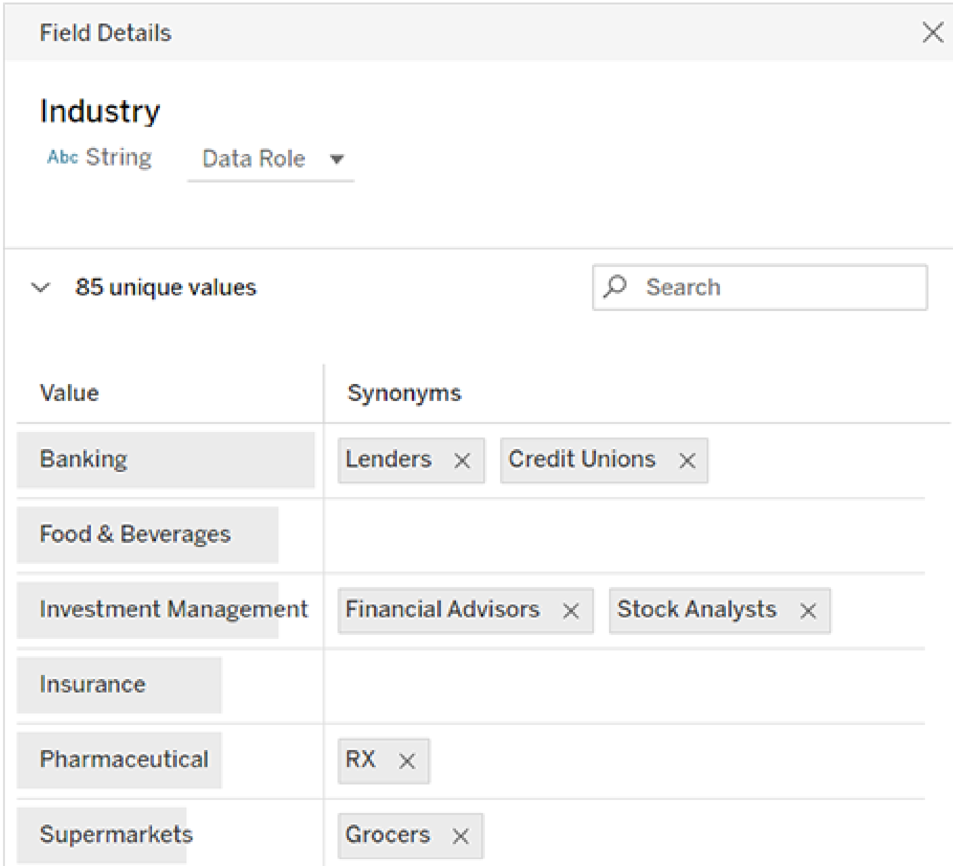
1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data source field at left, and click the **Edit Field Details** icon (the pencil).

3. To enter synonyms for the field name, use the **Synonyms** text box.

4. To enter synonyms for field values, click the arrow in the lower left corner.



Then use the search box to find specific values, and enter synonyms for them at right.



Tip: To help Ask Data better interpret users' questions, make synonyms for field values unique and short.

Add synonyms for multiple data sources

When a field or value has similar usage throughout your organization, use data roles to apply the same synonyms across multiple data sources. For example, data roles are a great choice if every subsidiary uses the same names for business departments, or if product lines have a consistent set of sub-products.

Using data roles for synonyms saves you a lot of time, because you can manage them centrally in Tableau Online or Tableau Server. From Ask Data, you can publish new data roles and apply existing ones created in Ask Data or Tableau Prep.

Publish synonyms to a new data role

1. Follow the steps for adding field name and value synonyms for a specific data source.
2. From the **Data Role** menu, choose **Publish Data Role**.
3. Enter a name for the role, and then click **Publish**.

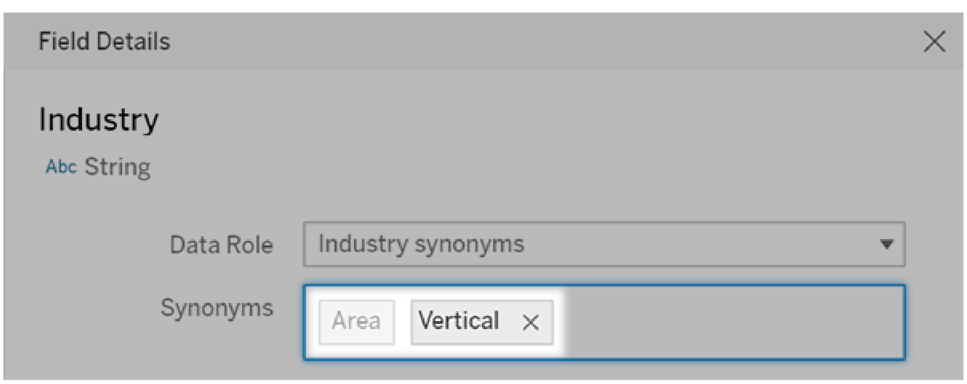
Apply synonyms from an existing data role

> **Note:** When applying existing data roles, you can select only roles you have permission to access.

1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data field at left, and click the **Edit Field Details** icon (the pencil).

3. From the **Data Role** menu, choose **Apply Data Role**.
4. Enter search terms to narrow the list of roles. Then select a role from the list, and click **Apply**.

   Synonyms from the role appear grayed out, but you can edit them on the central page for the data role.

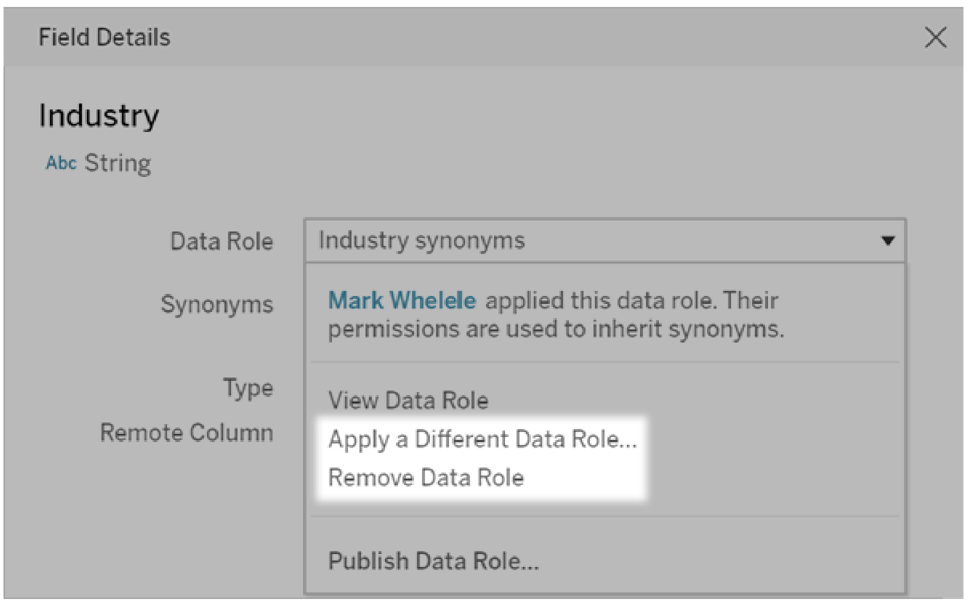


Grayed-out synonyms must be edited on the data role page. In this example, “Area” comes from a data role, while “Vertical” is specific to the current data source.

Change or remove an applied data role

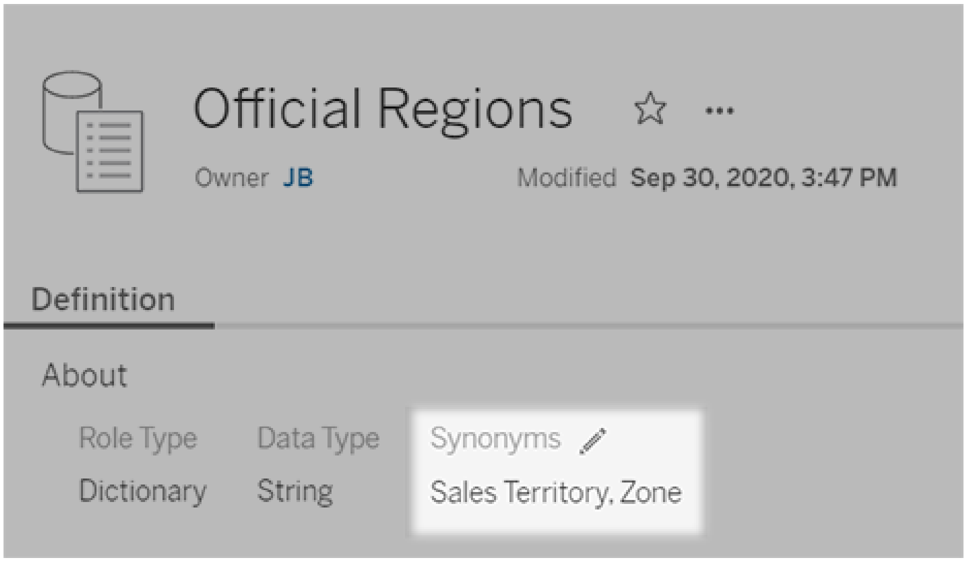
Note: You cannot change built-in data roles for geographic fields like country, state, province, or city.

1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data source field at left, and click the **Edit Field Details** icon .
3. Click the name of the currently applied data role, and choose either **Apply a Different Data Role** or **Remove Data Role**.

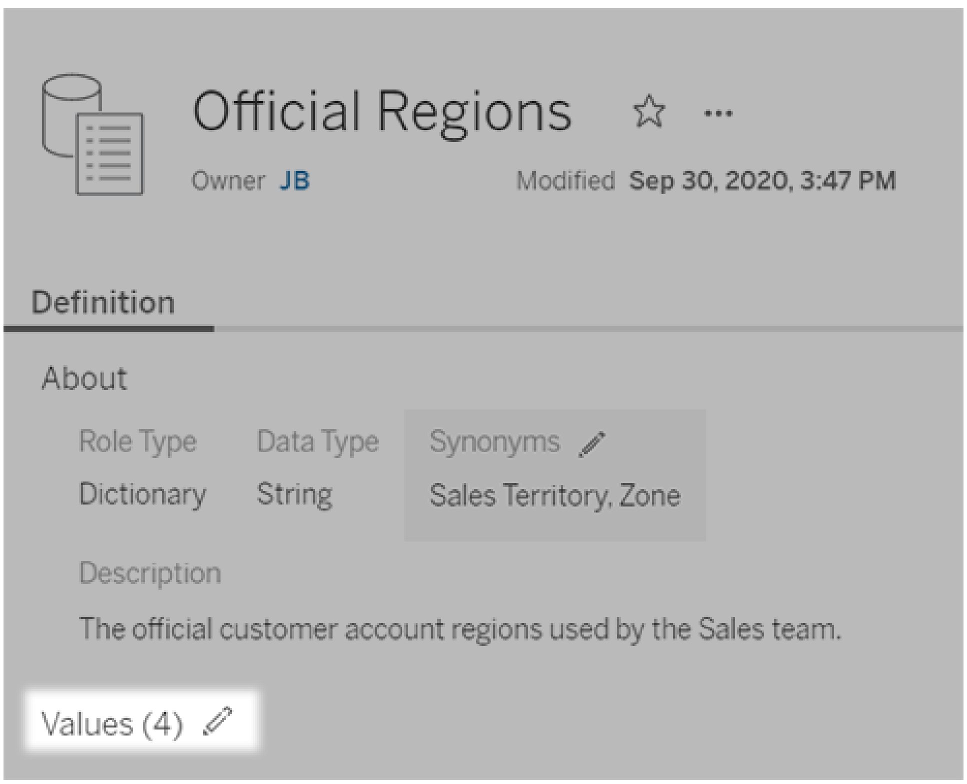


Edit field name or value synonyms for a data role

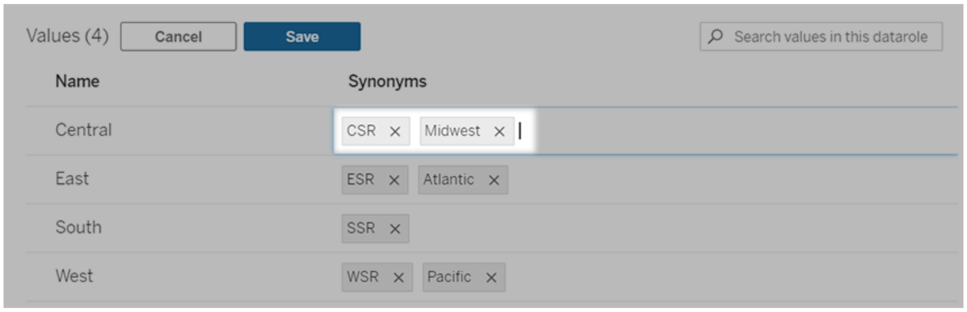
1. Go to the Ask Data tab for a data source or individual lens.
2. Hover over a data source field at left, and click the **Edit Field Details** icon .
3. Click the name of the currently applied data role, and choose **View Data Role**.
4. On the central page for the data role, do the following:
   - To edit synonyms for the field name, click the **Synonyms** section at top. Then add or remove synonyms in the text box, and click **Save**.



- To edit synonyms for field values, click the pencil icon next to **Values**.



Then edit values in the **Synonyms** column, and click **Save**. For fields with long lists of values, use the search box at right to narrow things down.



Disable or Enable Ask Data for a Site

Ask Data is enabled for sites by default, but Tableau administrators may disable it.

1. Go to the **General** site settings.
2. (Tableau Server only) In the **Web Authoring** section, select **Let users edit workbooks in their browser**.
3. In the **Availability of Ask Data** section, choose from these options:
   - **Enabled** enables creation of Ask Data lenses for all published data sources.
   - **Disabled** hides Ask Data throughout the site, while preserving information about previously created lenses so they can be restored if Ask Data is re-enabled.

Organize and Customize Fields in the Data Pane

The Data pane has many features and functions to help you organize and customize your data fields, find fields, and hide fields.

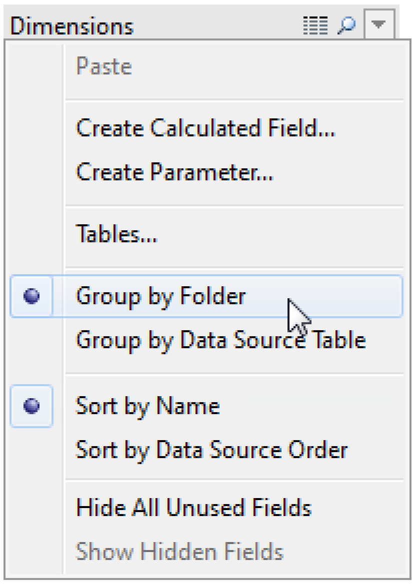
Organize the Data Pane

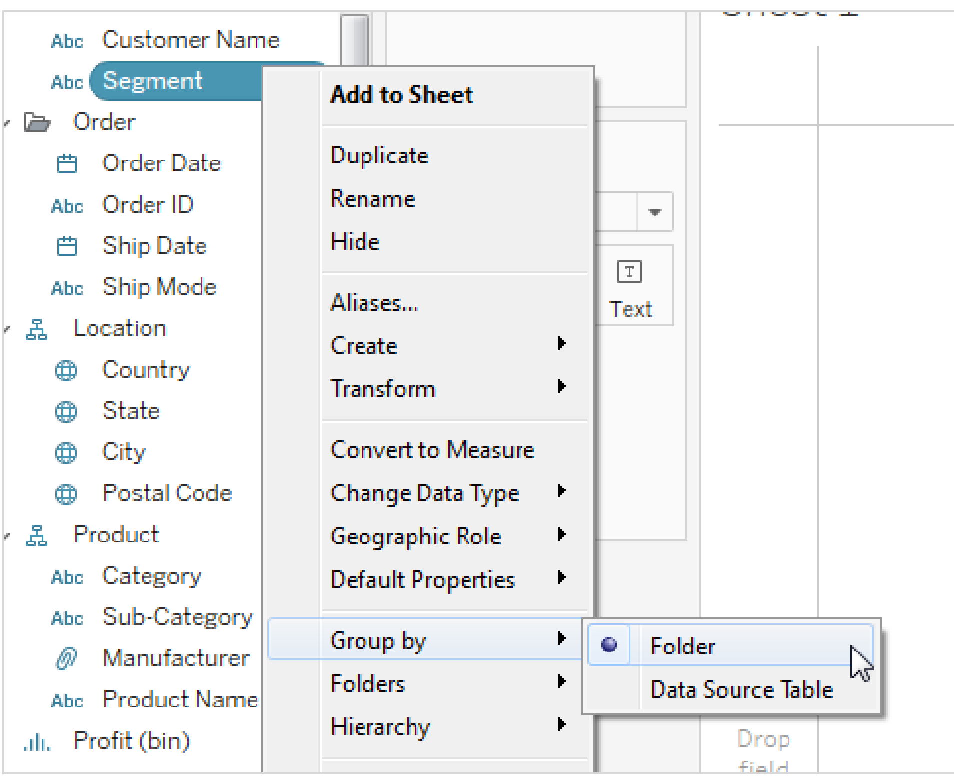
You can reorganize the items in the Data pane from its default layout using folders or through sorting.

Group fields in folders

To make data sources with many fields easier to work with, you can organize the Data pane items into folders. Items like fields, parameters, and sets can be grouped into folders.

Click the **Group by Folder** option in the Data pane menu, or in a field's context menu.



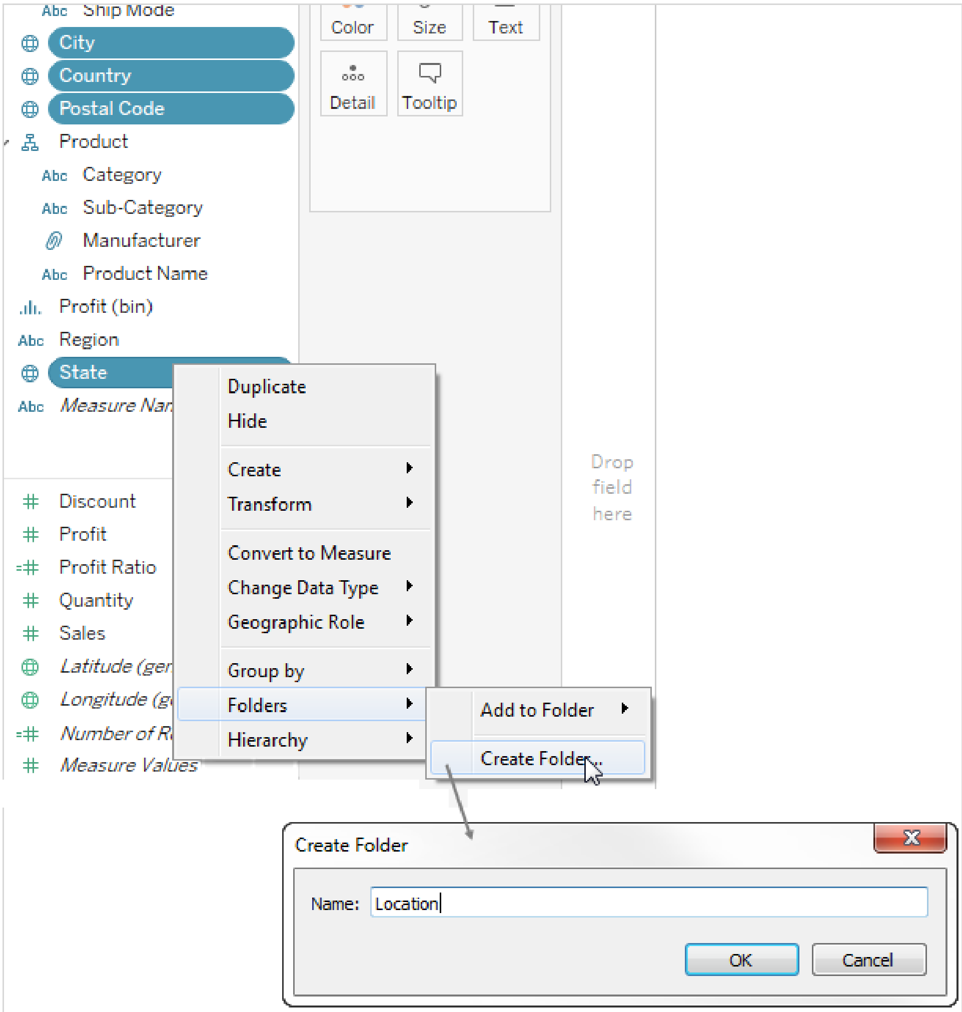


Notes on grouping options

- When you connect to a single table in your data source, grouping by folder is enabled by default.
- When you connect to a data source with multiple tables, grouping by table is enabled.
- When the **Group by Data Source Table** option is selected, the dimensions and measures are grouped according to the database table they belong to. This is especially useful when you have several joined tables.
- "Group by" options are only available for relational data sources—not for multidimensional (cube) data sources.

Group fields into folders

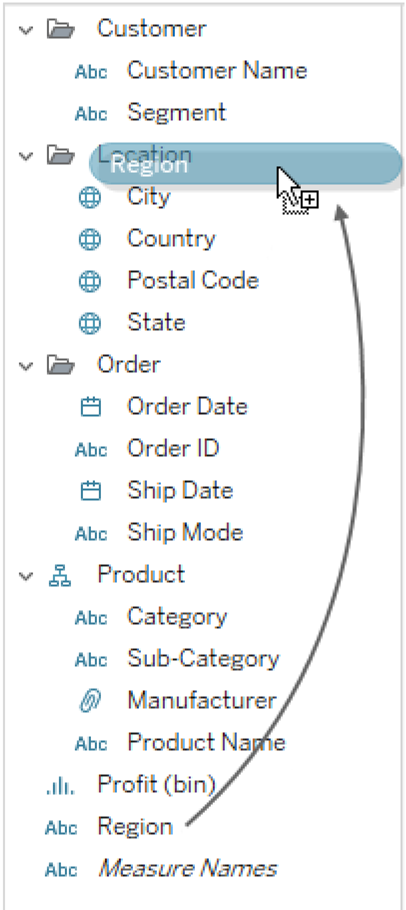
1. In the Data pane, select the fields you want to group together or right-click (control-click on Mac) an empty area in the Data pane.
2. Click **Folders** > **Create Folder**.
3. When prompted, name the new folder.



After you create a folder structure, you can drag fields from one folder to another or duplicate a field you want to have available in more than one folder.

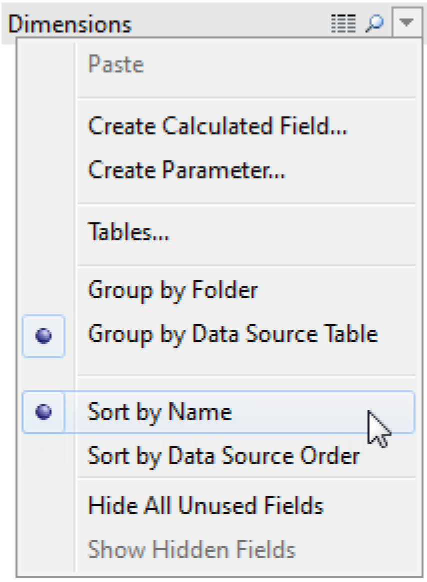
Add a field to a folder

- Drag a field on top of the folder name to add the field to the folder. If the folder is expanded, you can drag the field into the general area of the folder.



Sort fields in the Data pane

When organizing the Data pane with or without folders, you can have Tableau sort the items. These **Sort by** options are also located in the Data pane menu.



Sort options are only available for relational data sources—not for multidimensional (cube) data sources. You can sort by one of the following options:

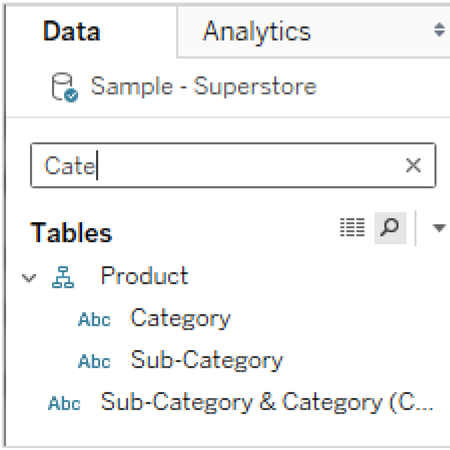
- **Sort by Name** – lists the dimensions and measures in alphabetical order according to their field aliases.
- **Sort by Data Source Order** – lists the dimensions and measures in the order they are listed in the underlying data source.

Find fields

You can search for fields, folders and hierarchies in the Data pane. When there are many fields in your data source it can be difficult to find a specific one like “Date” or “Customer” or “Profit,” or to find all fields that end in "xyz." To find an item, do the following:

Click the **Find Field** icon  at the top of the Data pane (**Ctrl + F** in Windows, **Command-F** on a Mac) and enter the name of the item you want to search for. You can also enter a string of characters, to search for all item names that contain that string.

As you type in the search box, search filters the contents of the Data pane to show all fields, folders or hierarchies that contain the typed string.



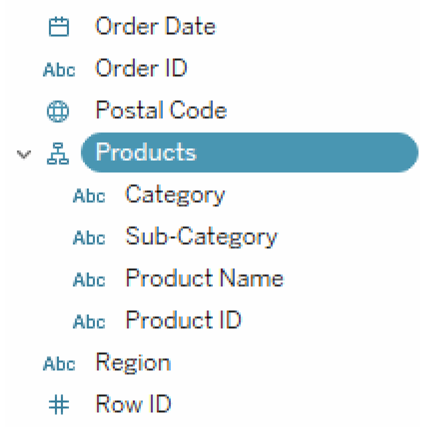
Search remains open until you click the **Find Field** icon  or press **Ctrl + F** again.

Rename Fields

You can rename fields in the Data pane. For example, you could rename a field named **Customer Segment** in the data source to be **Business Segment** in Tableau. You can also rename user-created fields. Renaming a field does not change the name of the field in the underlying data source, rather it is given a special name that appears only in Tableau workbooks. The changed field name is saved with the workbook as well as when you export the data source. You can rename any type of field: dimensions, measures, sets, or parameters.

Rename a field

1. Click field name in the Data pane and hold the mouse button down until the field name is shown in an edit box:



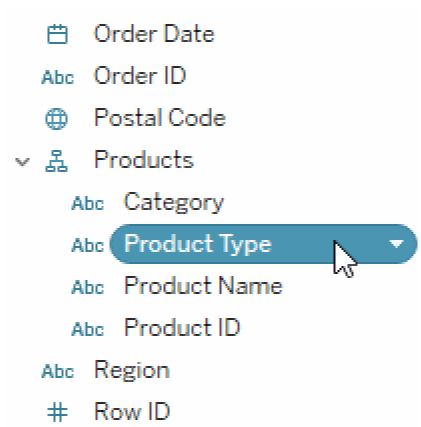
You can also press F2 or Ctrl + Enter to make the field name editable.

2. Type the new name for the field and press Enter.

The field displays with the new name in the Data pane.

Revert to the Default Field Name

If the field you renamed was from the original data source you can click the field name in the Data pane and hold the mouse button down until the field name is shown in the box. At the right of the edit field is a small circular arrow that you can click to restore the original data source field name:



To revert the names of multiple fields that were in the original data source, select them all, right-click, and then choose **Reset Names**.

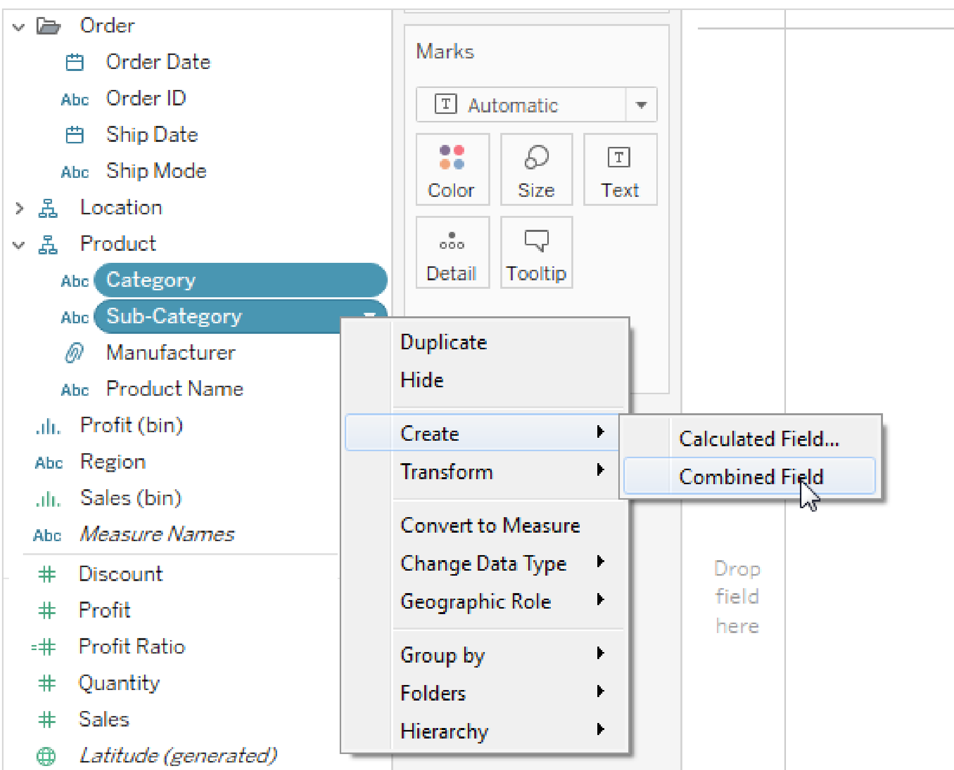
Combine Fields

Combine fields to create a cross product of members from different dimensions. You can combine dimensions if you want to encode a data view using multiple dimensions.

To combine the fields, select multiple dimensions in the Data pane and then right-click (control-click on a Mac) the fields and select **Create** > **Combined Field**.

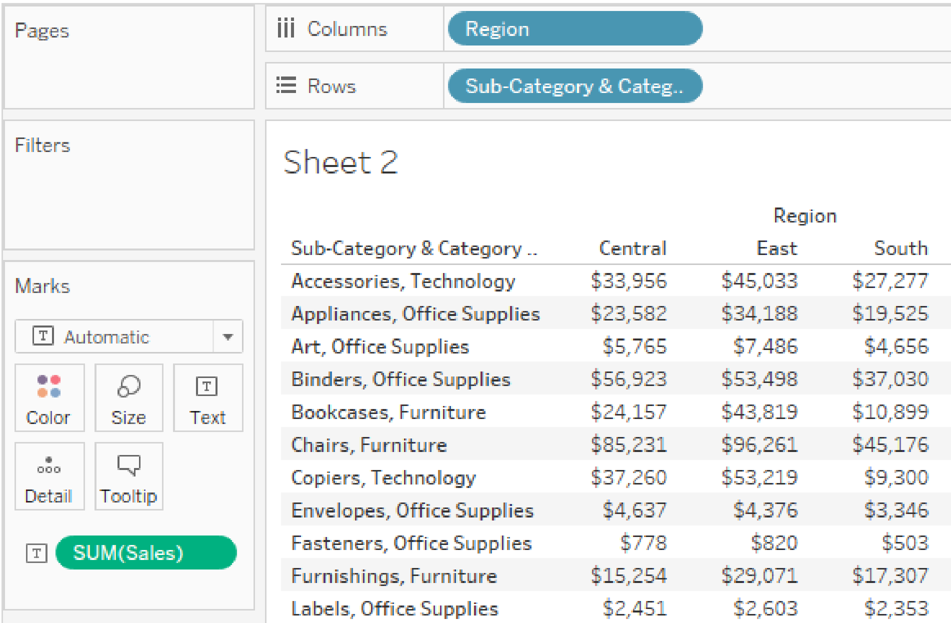
> **Note:** For cube (multidimensional) data sources, you must select levels from different hierarchies. In Tableau, cube data sources are supported only in Windows.

For example, the selections shown below will produce a new field that consists of the Category and Sub-Category dimensions.



The two dimensions are combined into a new dimension. The name of the field is automatically created from the names of the original fields. Right-click (control-click on a Mac) the new field and select **Rename** to change the name.

When you use the new field in a view, a header is created for each combination of the two original dimensions. For example, the view below shows the members of the combined Category and Sub-Category fields.

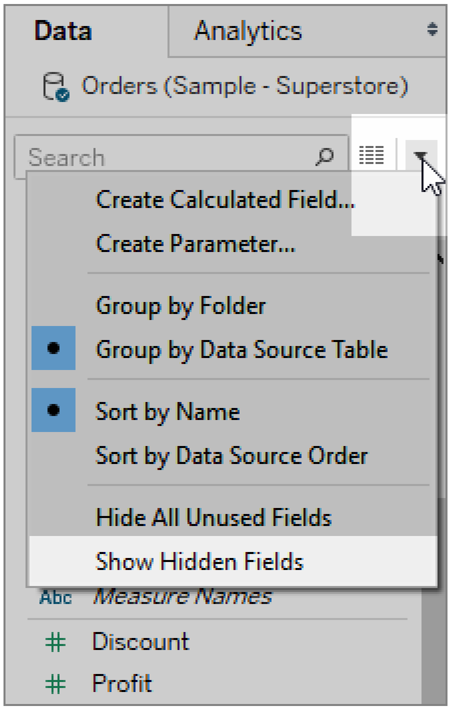


Note: For cube data sources, to choose to display the fully qualified name, right-click (control-click on a Mac) the combined field in the Data pane and select **Qualify Member Names**.

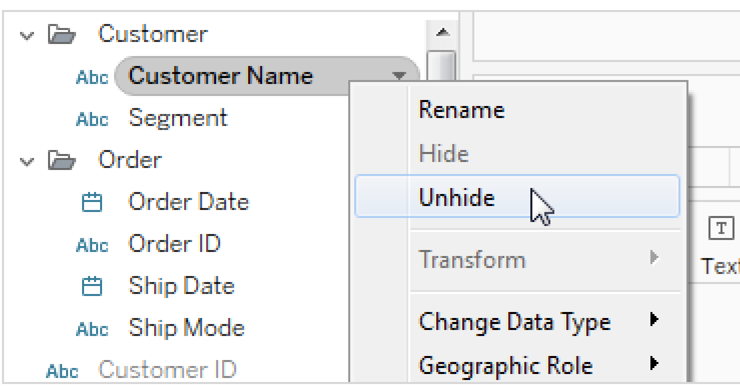
Hide or Unhide Fields

You can selectively hide or show fields in the Data pane. To hide a field, right-click (control-click on a Mac) the field and select **Hide**.

When you want to change your fields from hidden to visible, select **Show Hidden Fields** on the Data pane menu.



The hidden fields are then shown in gray in the Data pane. You can then select one or more hidden fields, right-click (control-click on a Mac) and select **Unhide**.

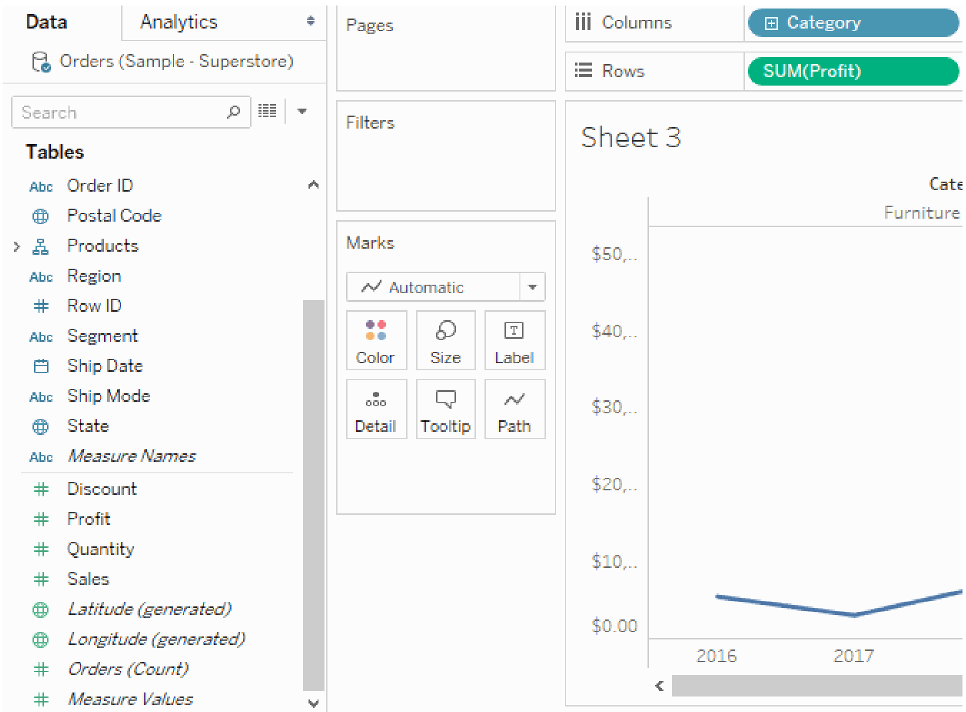


Select **Hide All Unused Fields** on the Data pane menu to quickly hide all of the fields that are not being used in the workbook.

Note: Hiding fields can be a good way to decrease the size of a data extract file because hidden fields are automatically excluded from the extract.

Add Calculated Fields to the Data Pane

You can create calculated fields that appear in the Data pane. These new computed fields can be used like any other field. Select **Create Calculated Field** on the Data pane menu. Alternatively, select **Analysis** > **Create Calculated Field**.



Edit Default Settings for Fields

When you drag fields to shelves, the data is represented as marks in the view. The fields and their marks are displayed initially based on their default settings. You can control these default settings by clicking the drop-down arrow on a field.

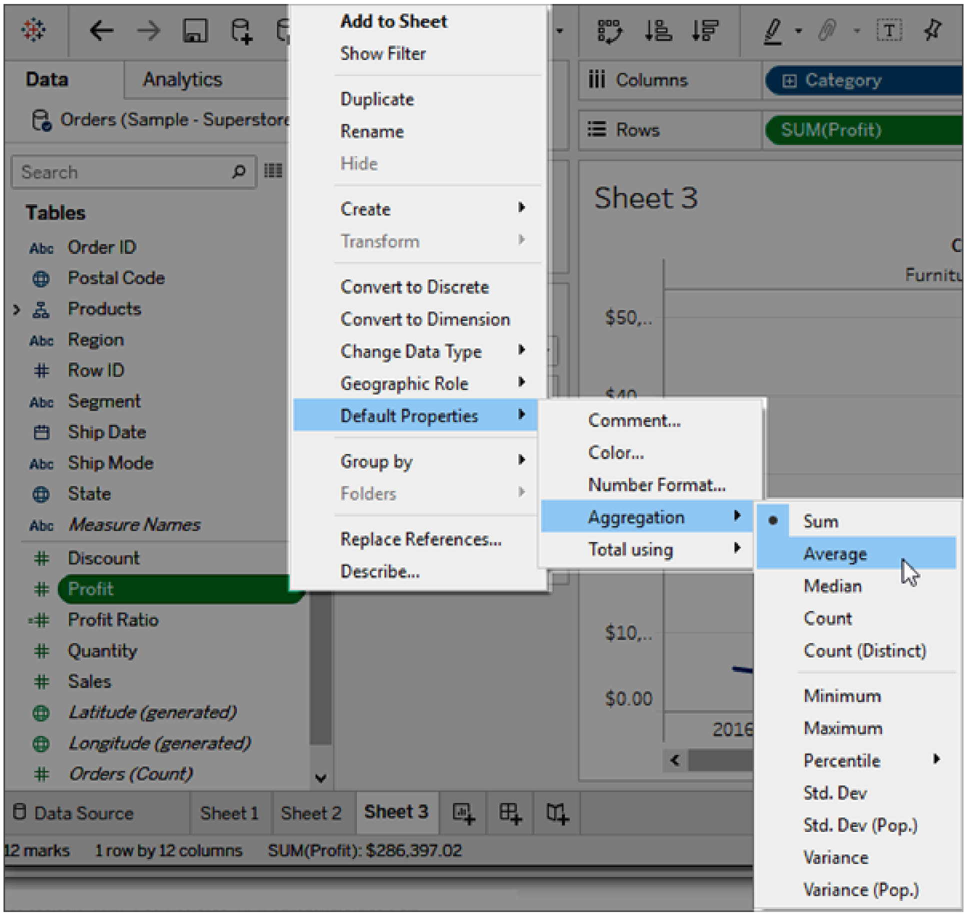
The **Default Properties** menu includes default settings for aggregation, comments, number formatting, color, shape, and totals (based on the type of field).

Set the default aggregation for a measure

You can specify a default aggregation for any measure. The default aggregation will be used automatically when the measure is first totaled in the view.

1. Right-click (control-click on a Mac) any measure in the Data pane and select **Default Properties** > **Aggregation**.

2. In the Aggregation list, select an aggregation.



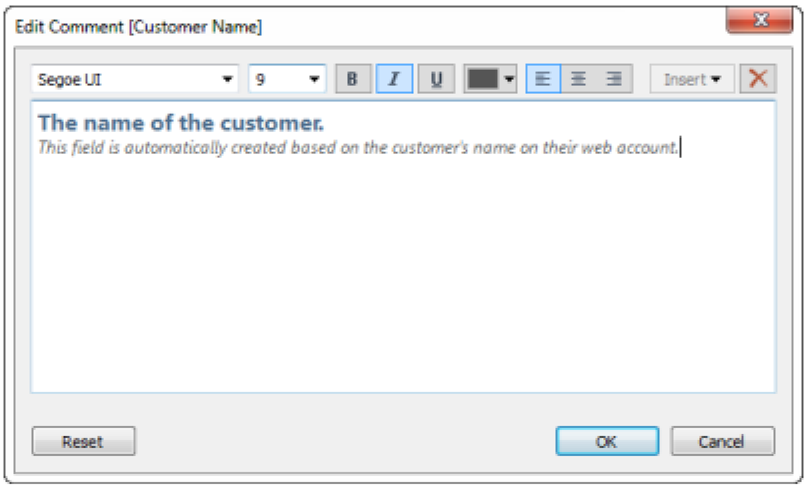
Whether you are specifying the aggregation for a field on a shelf or the default aggregation in the Data pane, you can select from several aggregations. See **Data Aggregation in Tableau** on page 341 to learn about each type of aggregation.

Add default comments for specific fields

Fields can have comments that describe them. The comments display in a tooltip in the Data pane and in the Calculated fields dialog box. Field comments are a good way to give more context to the data in your data source. Comments are especially useful when you are building a workbook for others to use.

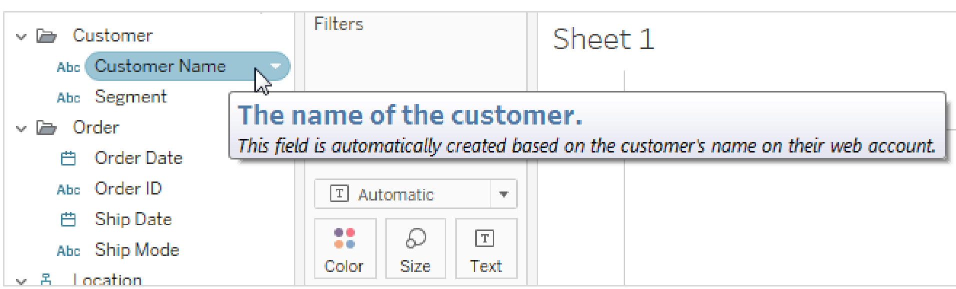
To add a default comment for a field

1. Right-click (control-click on a Mac) a field in the Data pane and select **Default Properties** > **Comment**.
2. Write a comment in the subsequent dialog box. Comments support rich text formatting that will be represented in the tooltip.



3. When finished, click **OK**.

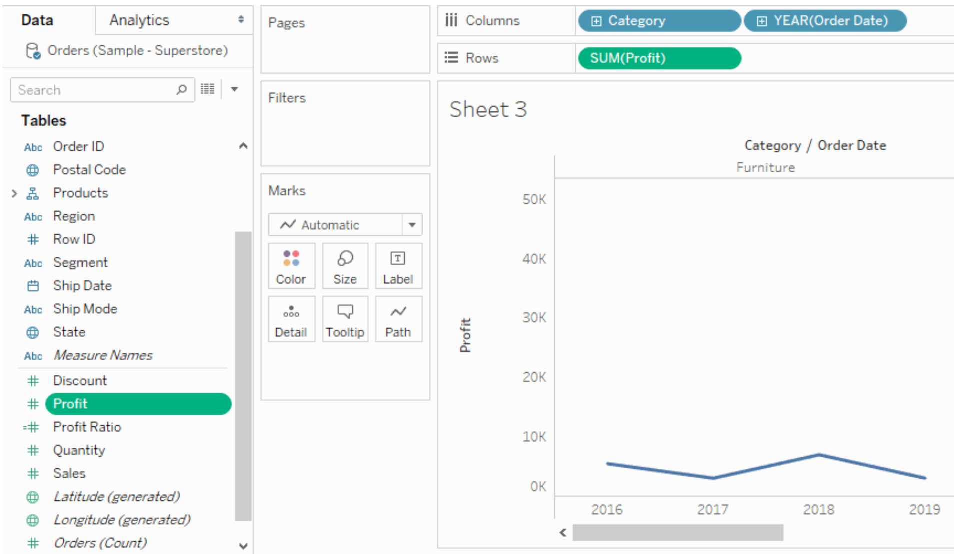
   Now when you hover the cursor over the field in the Data pane, you see the comment.



Set the default number format

You can set the default number format for date and number fields. For example, you may want to always show the Sales values as currency using the U.S. dollar sign and two decimal places. Or you may want to always show Discount as a percentage.

To set the default formats, right-click (control-click on Mac) a date or number field and select either **Date Format** or **Number Format** on the Default Properties menu. A dialog box opens where you can specify a default format.



Set the default color

When you use a dimension to color encode the view, default colors are assigned to the field’s values. Color encodings are shared across multiple worksheets that use the same data source to help you create consistent displays of your data. For example, if you define the Western region to be green, it will automatically be green in all other views in the workbook. To set the default color encodings for a field, right-click (control-click on Mac) the field in the Data pane and select **Default Properties** > **Color**.

For information about color properties, and how to configure and customize colors in Tableau, see **Color Palettes and Effects** on page 1394,

Set the default shape

When you use a dimension to shape encode the view, default shapes are assigned to the field’s values. Shape encodings are shared across multiple worksheets that use the same data source to help you create consistent displays of your data. For example, if you define that Furniture products are represented with a square mark, it will automatically be changed to a square mark in all other views in the workbook.

To set the default shape encodings for a field, right-click (control-click on Mac) the field in the Data pane and select **Default Properties** > **Shape**.

Set the default sort order for the values within a categorical field

You can set a default sort order for the values within a categorical field so that every time you use the field in the view, the values will be sorted correctly. For example, let's say you have an Order Priority field that contains the values High, Medium, and Low. When you place these in the view, by default they will be listed as High, Low, Medium because they are shown in alphabetical order. You can set a default sort so that these values are always listed correctly.

To set the default sort order, right-click (control-click on a Mac) a dimension and select **Default Properties** > **Sort**. Then use the sort dialog box to specify a sort order.

Note: The default sort order also controls how the field values are listed in a filter in the view.

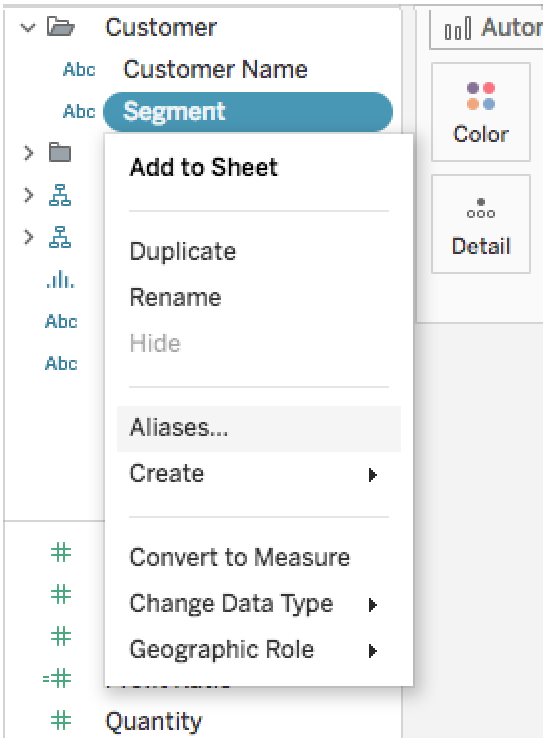
Create Aliases to Rename Members in the View

You can create aliases (alternate names) for members in a dimension so that their labels appear differently in the view.

Aliases can be created for the members of discrete dimensions only. They cannot be created for continuous dimensions, dates, or measures.

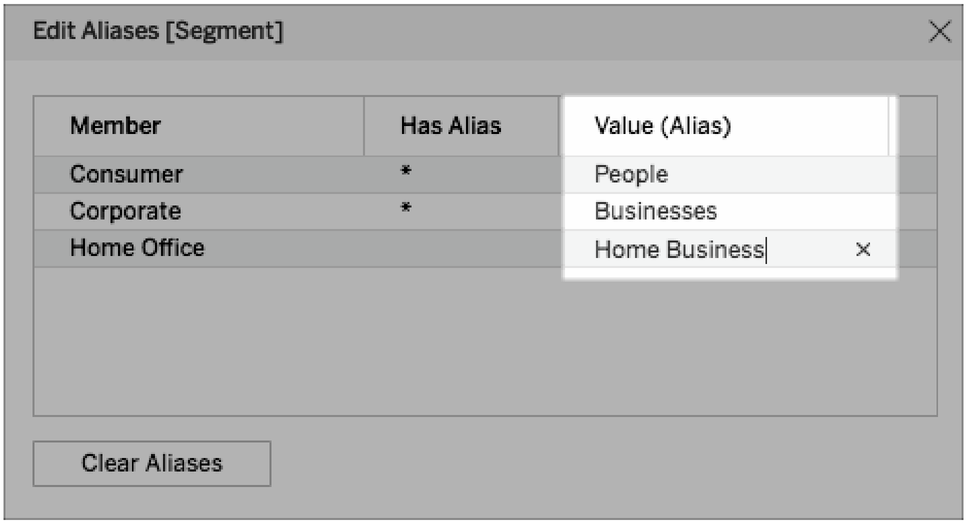
Create an alias

1. In the Data pane, right-click a dimension and select **Aliases**.



Tip: If **Aliases** does not appear in the dialog, verify that you are not connected to a published data source and the field you are trying to alias is a discrete dimension.

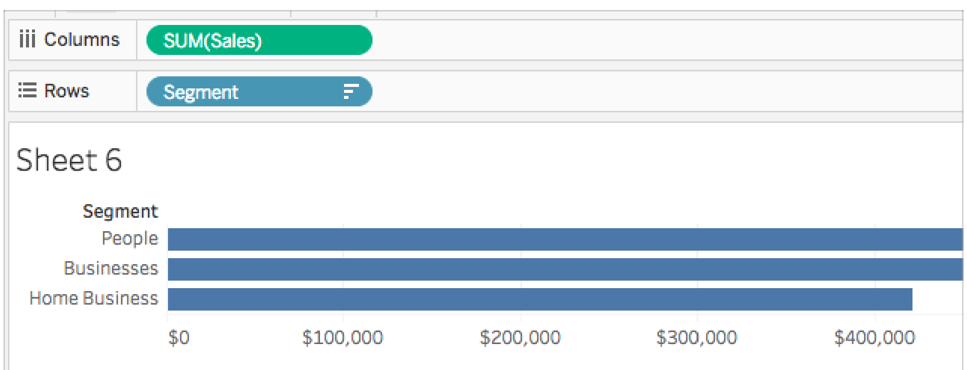
2. In the Edit Aliases dialog box, under **Value (Alias)**, select a member and enter a new name.



Tip: To reset the member names back to their original names, click **Clear Aliases**.

3. To save your changes:
   - In Tableau Desktop, click **OK**.
   - On Tableau Server or Tableau Online, click the **X** icon in the top-right corner of the dialog box.

When you add the field to the view, the alias names appear as labels in the view. For example,



When the Alias option is not available

Measures can't be re-aliased

Tableau does not permit re-aliasing measures as this would involve modifying data values themselves. If you have a field in your data that contains values such as *0* and *1* but actually encodes information such as *no* and *yes*, Tableau will interpret this as a continuous measure.

You can convert it to a discrete dimension and then re-alias the values. For more information about discrete vs continuous and dimensions vs measures, see:

- **Dimensions and Measures, Blue and Green** on page 314
- **Convert a Measure to a Dimension** on the facing page
- **Convert Fields between Discrete and Continuous** below

Published data sources

Alternatively, the Alias option may be missing for a discrete dimension because you are connected to a published data source. When using a published data source, you cannot create or edit aliases. You can only change aliases on fields that you create in your workbook. If you didn't publish the data source, you can duplicate the field (right click > **Duplicate**) and edit aliases for the duplicate field without restriction.

Bulk re-aliasing

Watch a Video: To see how to bulk re-alias all members of a field, watch Cleaning Data by Bulk Re-aliasing, a 3-minute free training video. Use your tableau.com account to sign in.

Convert Fields between Discrete and Continuous

You can convert measures from discrete to continuous or from continuous to discrete. And you can convert date dimensions and other numeric dimensions to be either discrete or continuous. For more information on continuous and discrete fields, see **Dimensions and Measures, Blue and Green** on page 314.

You can also **Convert a Measure to a Dimension** on the facing page.

Convert measures

You can convert measures from discrete to continuous or from continuous to discrete. Click the field and choose **Discrete** or **Continuous**. The field is green when it is continuous, and blue when it is discrete.

For measures in the **Data** pane, right-click the field and choose **Convert to Discrete** or **Convert to Continuous**. The color of the field changes accordingly.

Convert date fields

You can convert Date fields between discrete and continuous. Discrete dates act as labels and continuous dates will have an axis similar to a measure.

Click any Date field in the view and choose one of the options on the context menu to change it from discrete to continuous or from continuous to discrete.

To convert a Date field in the **Data** pane (and thus to determine the default result when you drag it into a view), right-click the field and choose **Convert to Discrete** or **Convert to Continuous**.

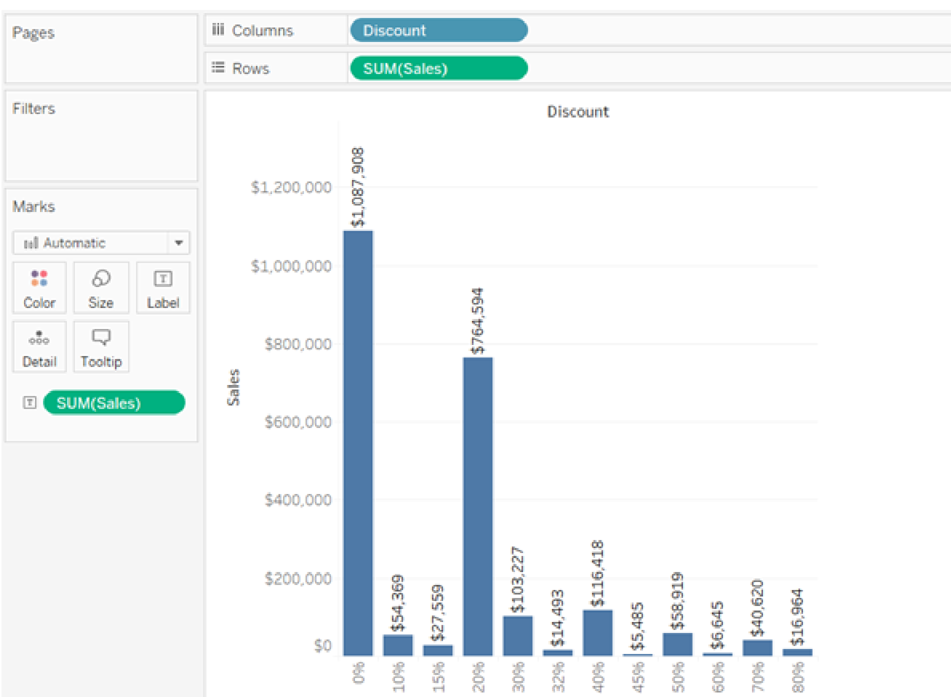
Convert a Measure to a Dimension

You can convert a field from a measure to a dimension in the current view. Or, if you want the change to affect all future uses of the field in the workbook, you can convert a field in the **Data** pane from a measure to a dimension. For more information on dimensions and measures, see **Dimensions and Measures, Blue and Green** on page 314.

You can also **Convert Fields between Discrete and Continuous** on the previous page.

Convert a measure in the view into a discrete dimension

You can drag a measure field from the **Data** pane but then use it as a dimension in the view. For example, suppose you want to know the aggregated sales totals for each possible discount rate. The view you are aiming for looks like this:



The **Discount** field contains numeric data, so when you connect to the data source, Tableau assigns it to the **Measures** area in the **Data** pane. In the **Sample - Superstore** data source, which is included with Tableau Desktop, the values for **Discount** range from 0% to 80%.

Here are the steps to create the view shown above:

1. Drag **Sales** to **Rows** and **Discount** to **Columns**. Tableau shows you a scatter plot—this is the default chart type when you put one measure on **Rows** and another on **Columns**.

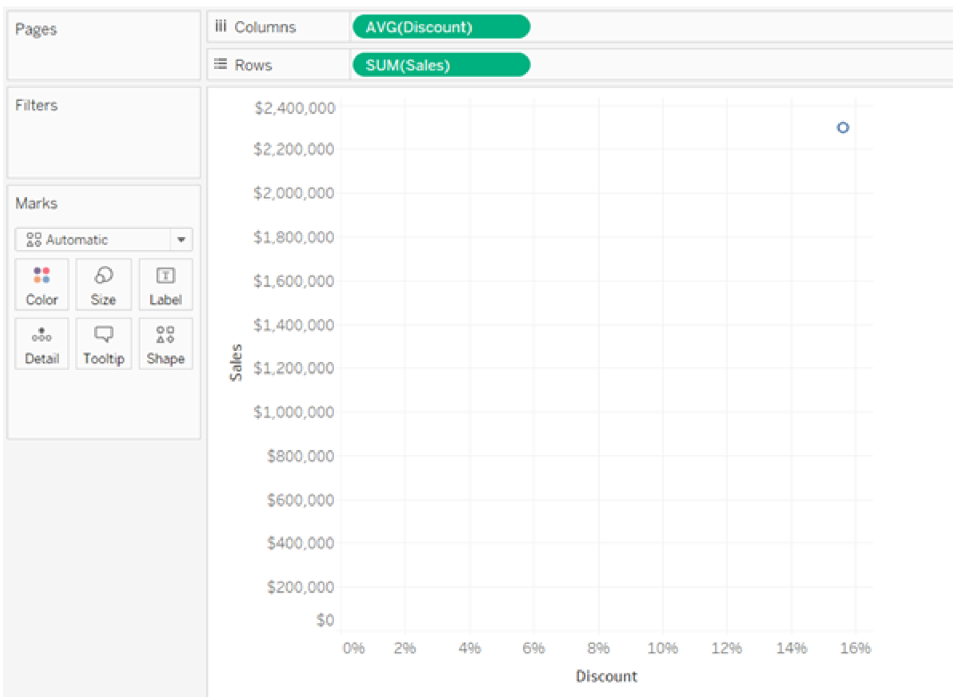
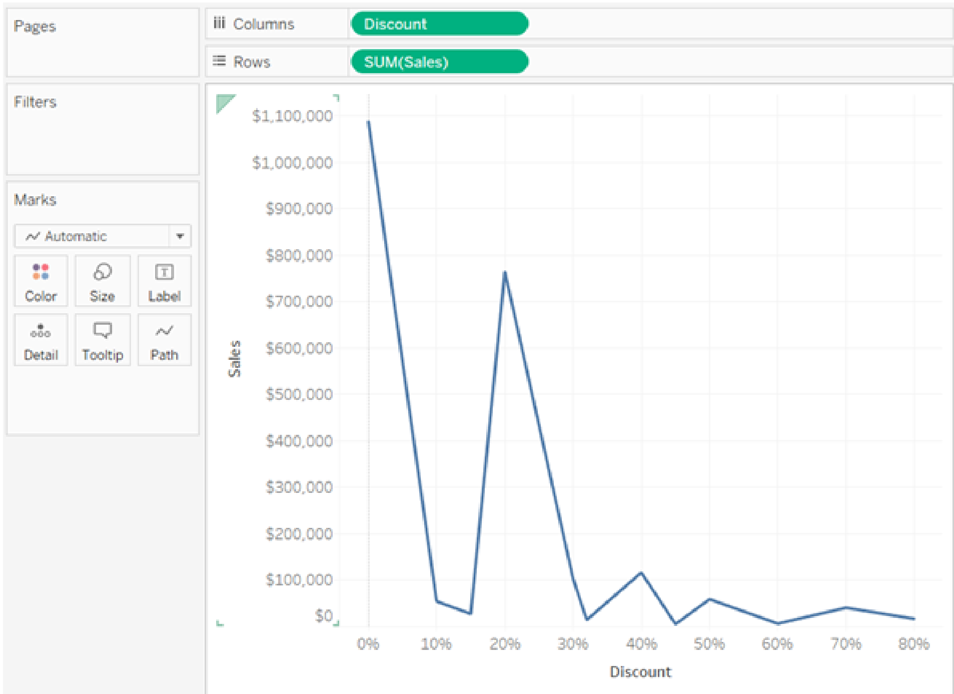
   

   Tableau aggregates **Discount** as AVG, and **Sales** as SUM. The fields are both continuous, so along the bottom and left side of the view Tableau displays axes (and not column or row headers).

2. To treat **Discount** as a dimension, click the drop-down arrow on the field (on the **Columns** shelf) and select **Dimension** from the context menu. Tableau no longer aggregates the values for **Discount**, so what you see now is a line. But the values for **Discount** are still continuous, so Tableau still shows continuous axes for both fields:



3. To complete the process, click the drop-down arrow on the **Discount** again and select **Discrete** from the context menu. The transformation of **Discount** is now complete. You now see the bar chart as in the initial image at the beginning of this topic. Across the bottom, you now see column headers (0%, 10%, 20%, etc.) instead of an axis.

   Let's review how we got to this point:

Action	Result
Convert **Discount** from a measure to a dimension...	Sales values are no longer aggregated according to discount rate, resulting in a line chart instead of a scatter plot.
Convert **Discount** from continuous to discrete...	Tableau shows headers at the bottom of the view, instead of a continuous axis.

   The only thing left to do is to drag **Sales** to Label and then format the labels for readability.

The resulting chart is somewhat useful because there are only 12 unique values for **Discount** in the data source. Had there been a unique value for each row, which would not have been unusual for a numeric field, the number of individual bars in the resulting view would have been equal to the number of rows in the data source, which would probably not result in a useful visualization.

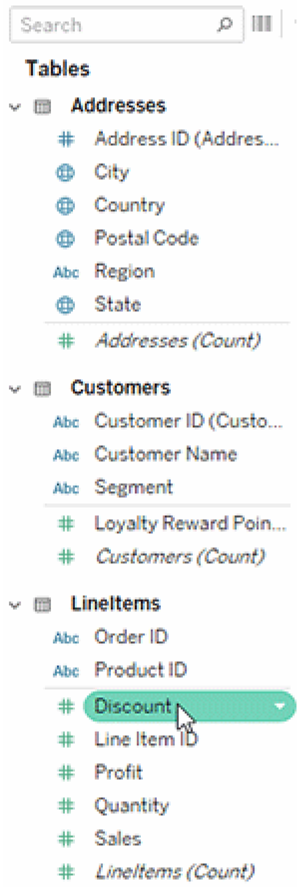
Converting a measure in the Data pane into a dimension

When you first connect to a data source, Tableau assigns most fields that contain quantitative, numerical information (that is, fields where the values are numbers) as measure fields in the **Data** pane. The exception is for fields where the name suggests the data type, such as Year or Month (which Tableau would identify as Date dimensions) or fields containing words like “ID” and “Key,” which Tableau would categorize as dimensions, even when they are numeric.

However, you might decide that some of these fields that Tableau has categorized as measures should actually be dimensions. Postal codes are a classic example—they often consist entirely of numbers, but the information is categorical and not continuous—you would never want to aggregate postal codes by adding or averaging them. Similarly, a field containing individuals' ages may be categorized as a measure by default in Tableau because it contains numeric data. In some cases you may want to add or average ages, but you might also want to look at each individual age as a bin or category, in which case you want Tableau to create headers for this field rather than an axis. If this is how you want to use age in your view, you can convert the field to a dimension.

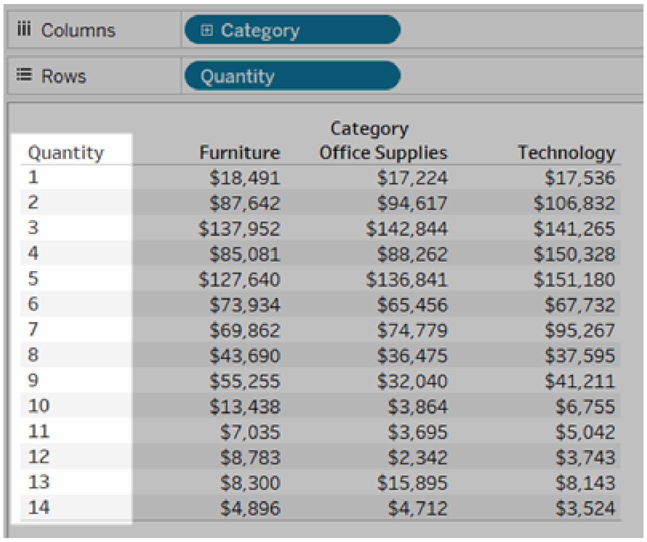
To convert a measure to a dimension in the **Data** pane, do either of the following.

- Click and drag the field from a measures area in the **Data** pane and drop it into a dimensions area (above the line).



- Right-click (control-click on a Mac) the measure in the **Data** pane and select **Convert to Dimension**.

If you place a field that you converted from a measure to a dimension on a shelf, it now produces headers instead of an axis.



Quantity	Category Furniture	Office Supplies	Technology
1	$18,491	$17,224	$17,536
2	$87,642	$94,617	$106,832
3	$137,952	$142,844	$141,265
4	$85,081	$88,262	$150,328
5	$127,640	$136,841	$151,180
6	$73,934	$65,456	$67,732
7	$69,862	$74,779	$95,267
8	$43,690	$36,475	$37,595
9	$55,255	$32,040	$41,211
10	$13,438	$3,864	$6,755
11	$7,035	$3,695	$5,042
12	$8,783	$2,342	$3,743
13	$8,300	$15,895	$8,143
14	$4,896	$4,712	$3,524

Create Hierarchies

When you connect to a data source, Tableau automatically separates date fields into hierarchies so you can easily break down the viz. You can also create your own custom hierarchies. For example, if you have a set of fields named Region, State, and County, you can create a hierarchy from these fields so that you can quickly drill down between levels in the viz.

> **Watch a video**: To see related concepts demonstrated in Tableau, watch Drill Down and Hierarchies, a 5-minute free training video. Use your tableau.com account to sign in.

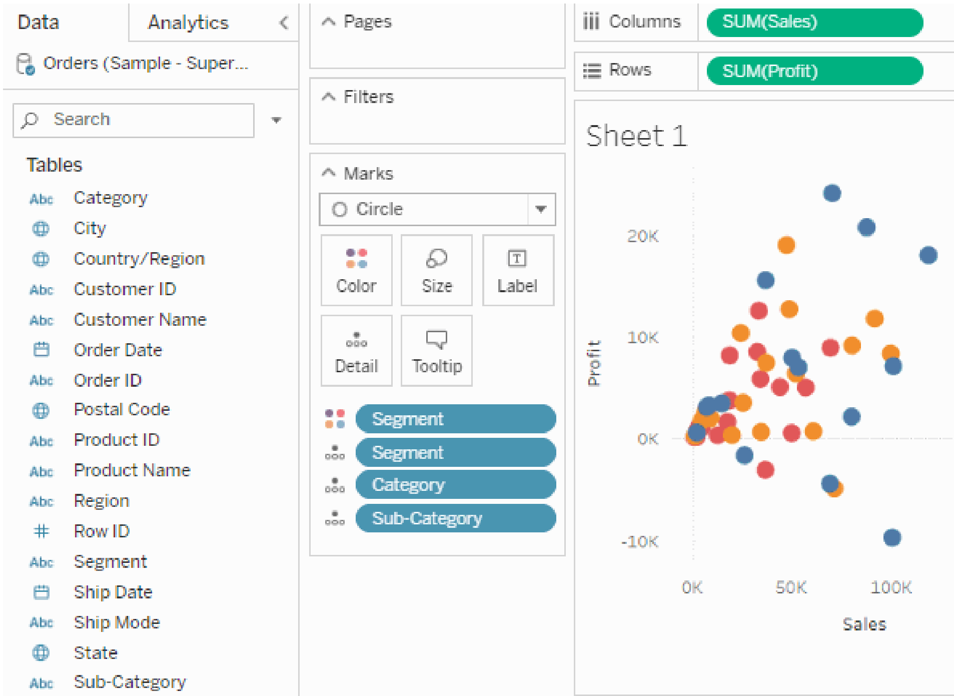
Create a hierarchy

To create a hierarchy:

1. In the **Data** pane, drag a field and drop it directly on top of another field.

   > **Note:** When you want to create a hierarchy from a field inside a folder, right-click (control-click on a Mac) the field and then select **Create Hierarchy**.

2. When prompted, enter a name for the hierarchy and click **OK**.
3. Drag additional fields into the hierarchy as needed. You can also re-order fields in the hierarchy by dragging them to a new position.

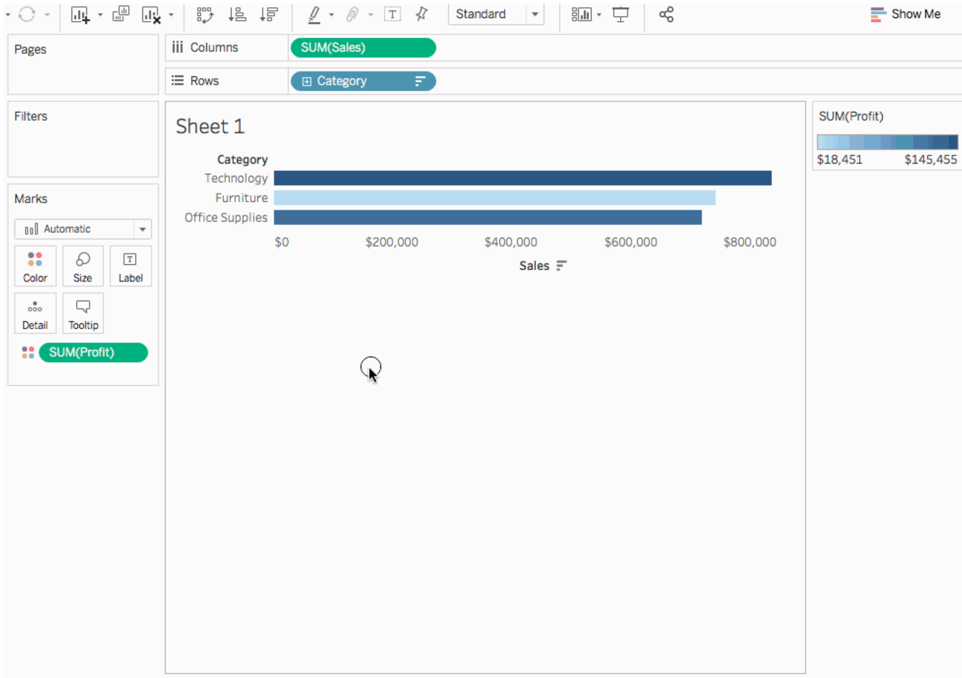


Drill up or down in a hierarchy

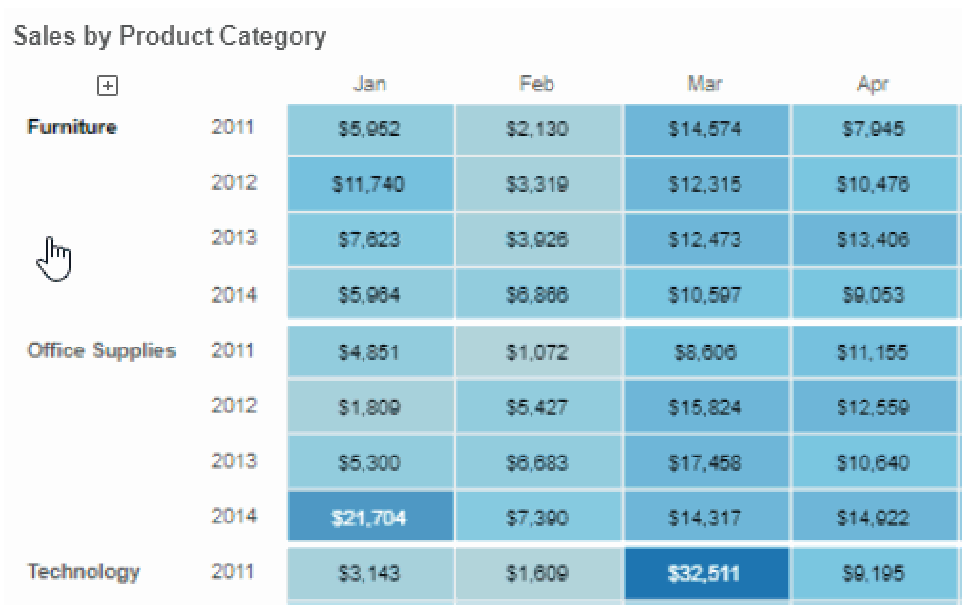
When you add a field from a hierarchy to the visualization, you can quickly drill up or down in the hierarchy to add or subtract more levels of detail.

To drill up or down in a hierarchy in Tableau Desktop or in web authoring:

- In the visualization, click the **+** or **-** icon on the hierarchy field.



When you are editing or viewing the visualization on the web, you have the option of clicking the **+** or **-** icon next to a field label.



Remove a hierarchy

To remove a hierarchy:

- In the **Data** pane, right-click (control-click on a Mac) the hierarchy and select **Remove Hierarchy**.

  The fields in the hierarchy are removed from the hierarchy and the hierarchy disappears from the Data pane.

Group Your Data

You can create a group to combine related members in a field. For example, if you are working with a view that shows average test scores by major, you might want to group certain majors together to create major categories. English and History might be combined into a group called Liberal Arts Majors, while Biology and Physics might be grouped as Science Majors.

Groups are useful for both correcting data errors (e.g., combining CA, Calif., and California into one data point) as well as answering "what if" type questions (e.g., "What if we combined the East and West regions?).

Create a group

There are multiple ways to create a group. You can create a group from a field in the **Data** pane, or by selecting data in the view and then clicking the group icon.

Create a group by selecting data in the view

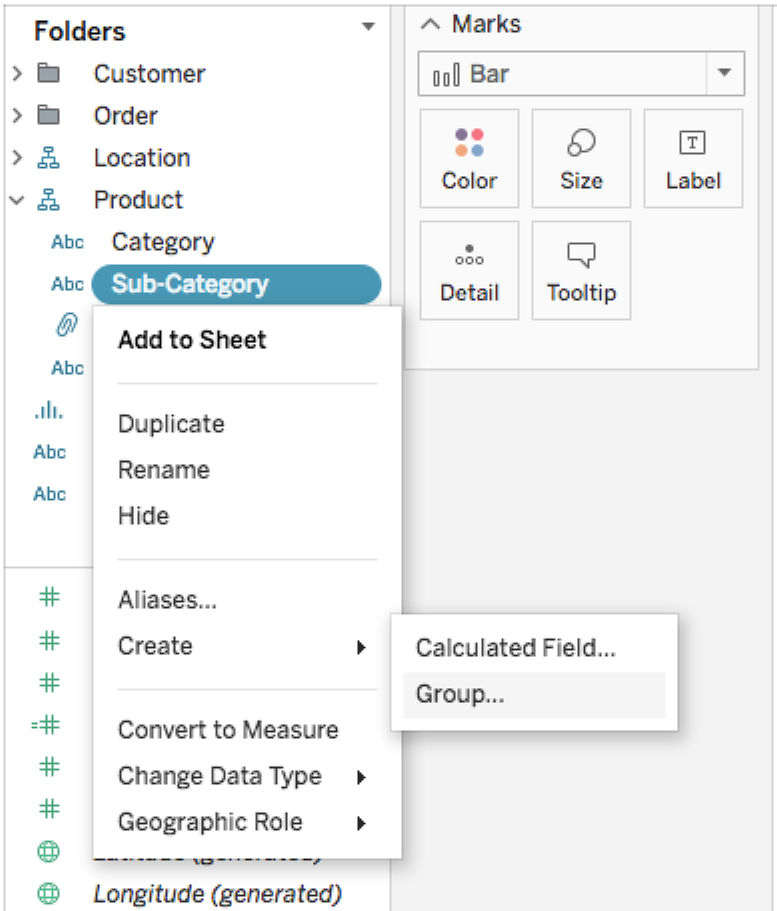
1. In the view, select one or more data points and then, on the tooltip that appears, click the group icon .

   Note: You can also select the group icon on the toolbar at the top of the workspace.

   If there are multiple levels of detail in the view, you must select a level to group the members. You can select to group all dimensions, or just one.

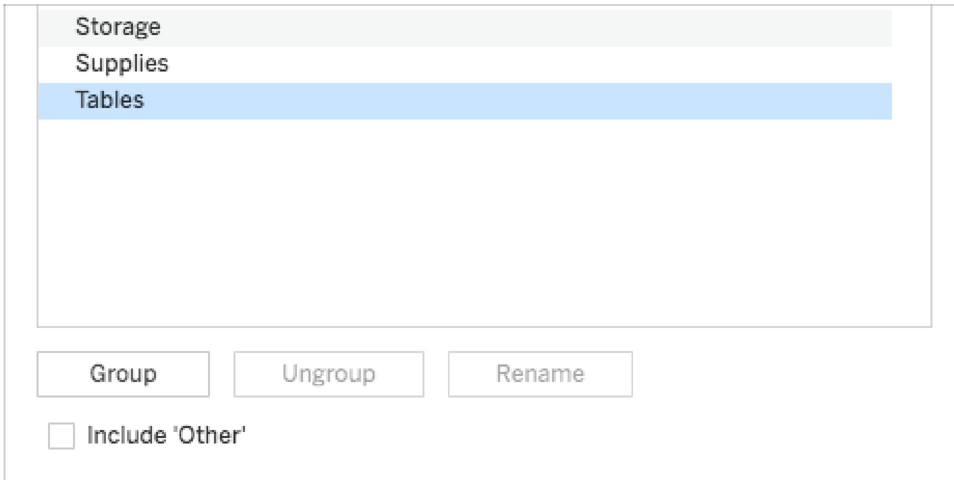
Create a group from a field in the Data pane

1. In the **Data** pane, right-click a field and select **Create** > **Group**.



2. In the Create Group dialog box, select several members that you want to group, and

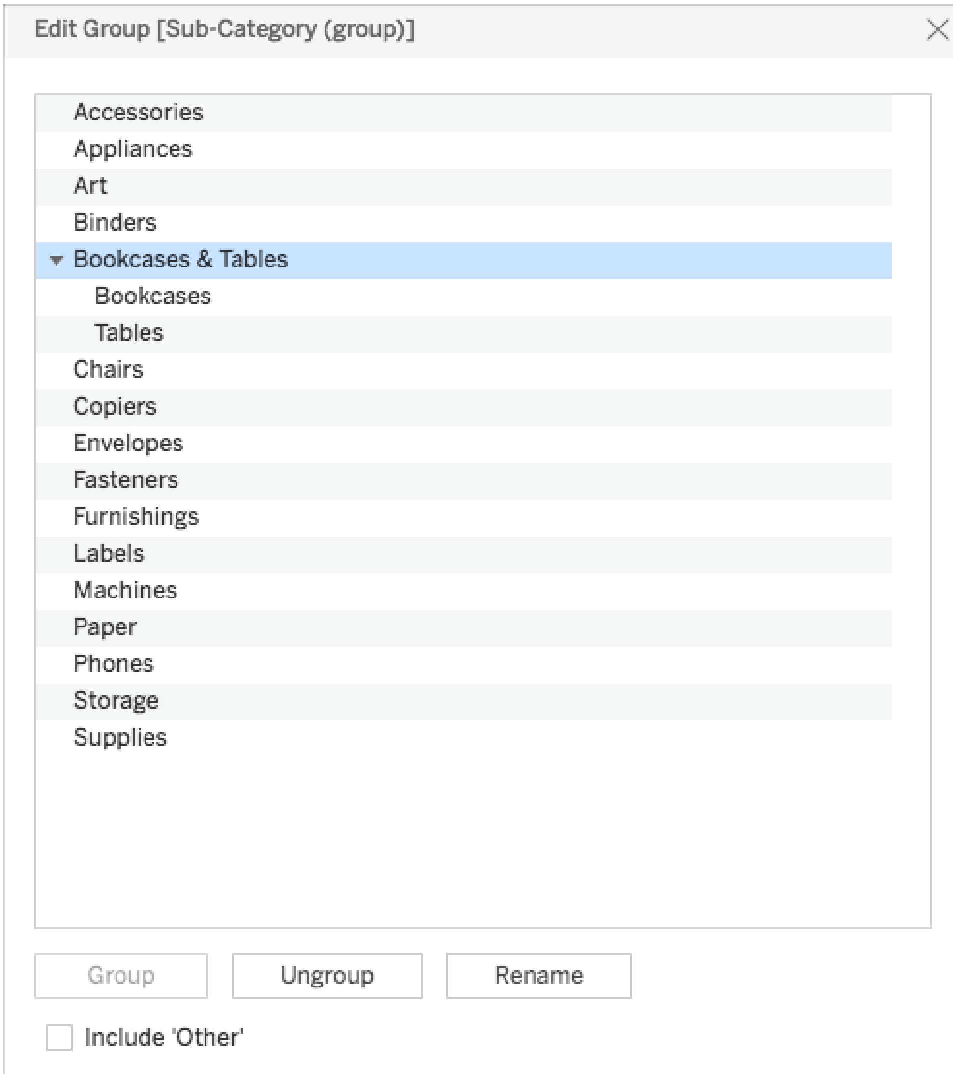
then click **Group**.



The selected members are combined into a single group. A default name is created using the combined member names.

To rename the group, select it in the list and click **Rename**.

Tip: You can search for members using the **Find** option near the bottom-right of the dialog box. (Tableau Desktop only)

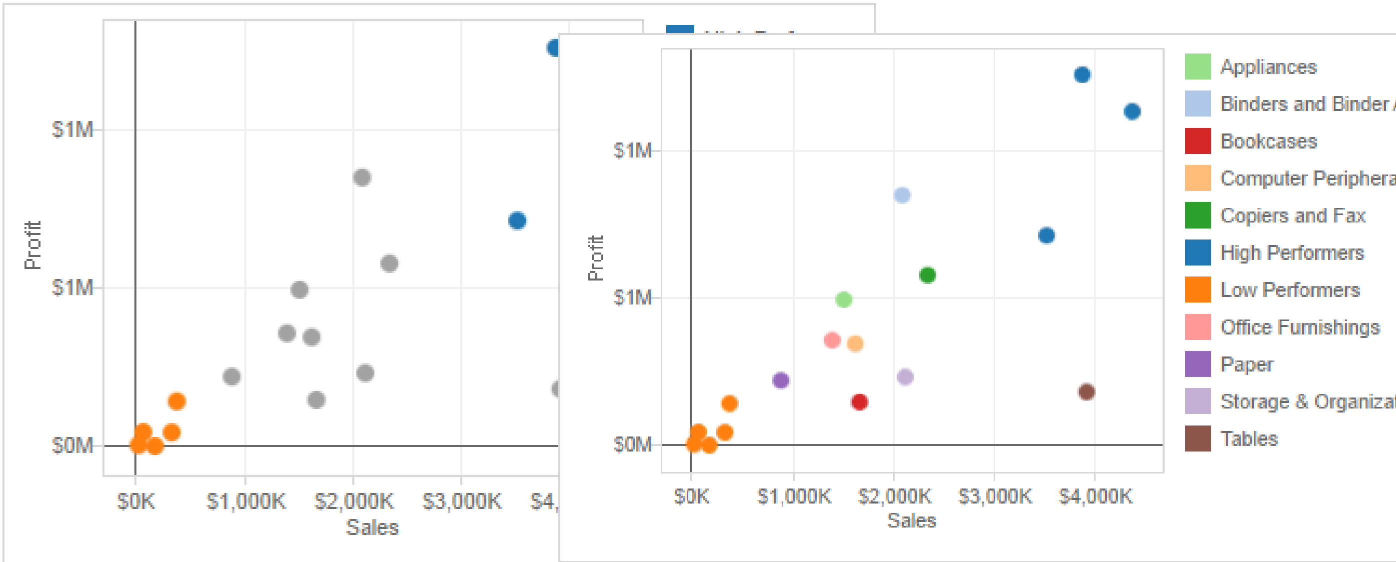


Include an Other Group

When you create groups in Tableau, you have the option to group all remaining, or non-grouped members in an Other group.

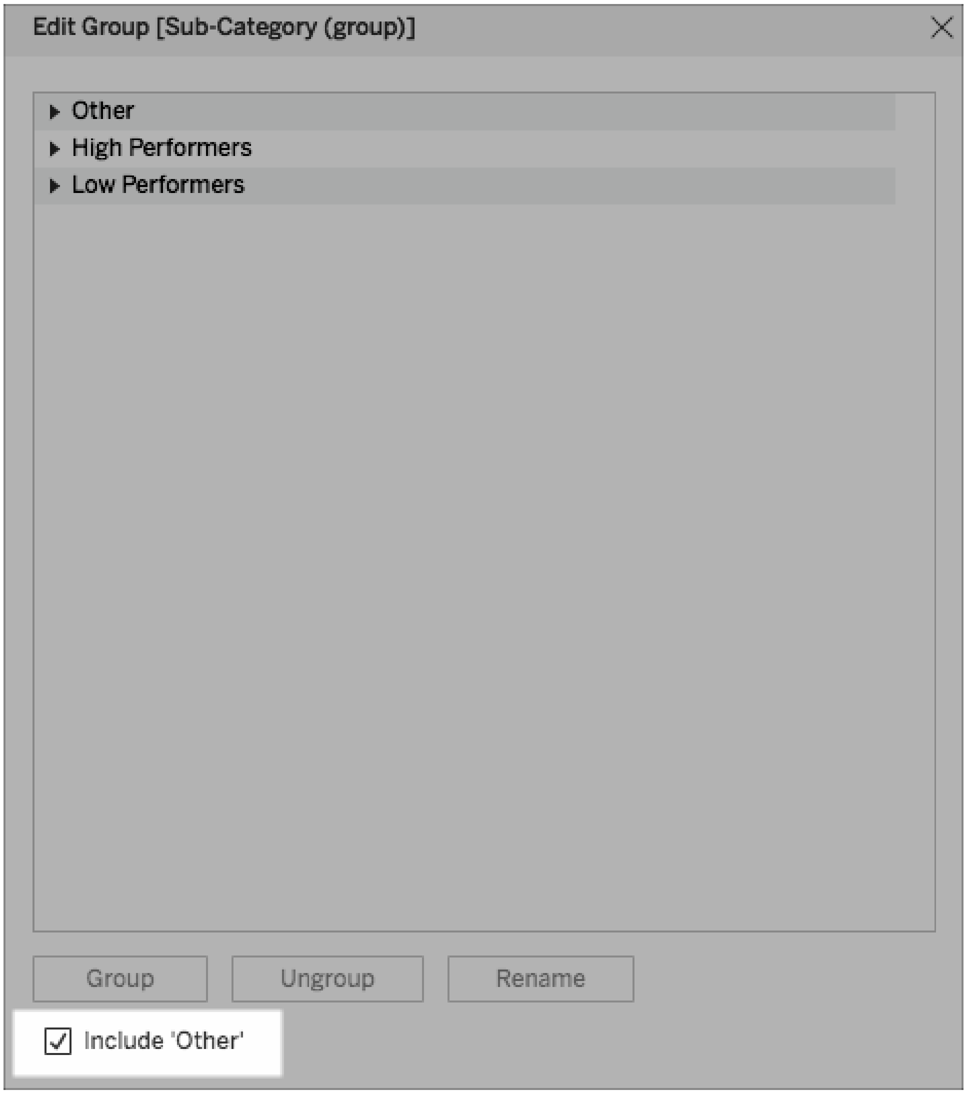
The Include Other option is useful for highlighting certain groups or comparing specific groups against everything else. For example, if have a view that shows sales versus profit product category, you might want to highlight the high and low performing categories in the view, and group all the other categories into an "Other" group.

Includes Other	Does not include Other



To include an Other group:

1. In the **Data** pane, right-click the group field and select **Edit Group**.
2. In the Edit Group dialog box, select **Include 'Other'**.

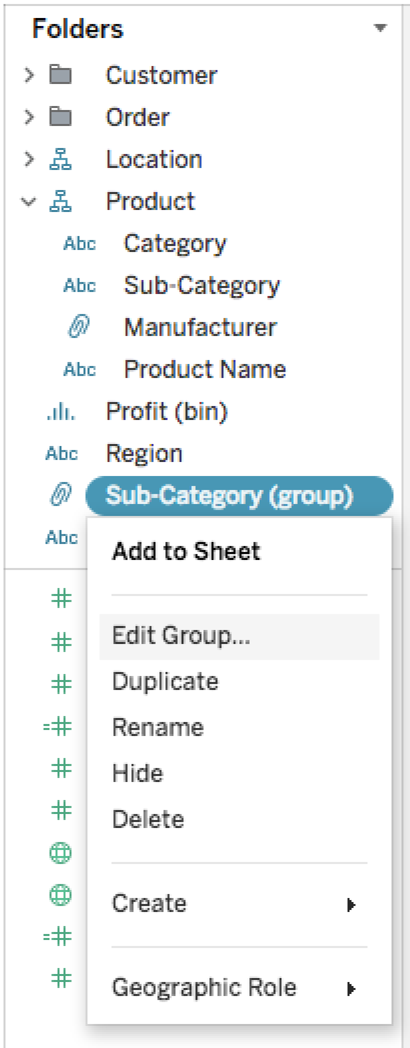


Edit a Group

After you have created a grouped field, you can add and remove members from the groups, create new groups, change the default group names, and change the name of the grouped field. You can make some changes directly in the view, and others through the Edit Group dialog box.

To add members to an existing group:

- In the **Data** pane, right-click the group field, and then click **Edit Group**.



- In the Edit Group dialog box, select one or more members and drag them into the group you want.
- Click **OK**.

To remove members from an existing group:

- In the **Data** pane, right-click the group field, and then click **Edit Group**.
- In the Edit Group dialog box, select one or more members, and then click **Ungroup**.

  The members are removed from the current group. If you have an Other group, the members are added to it.

- Click **OK**.

To create a new group in a group field:

- In the **Data** pane, right-click the group field, and then click **Edit Group**.
- In the Edit Group dialog box, select one or more members, and then click **Group**.
- Click **OK**.

> **Note:** To rename a group, select the group in the Edit Group dialog box, and then click **Rename**.

See also

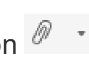
Color a View Using Groups below

Correct Data Errors or Combine Dimension Members by Grouping Your Data on page 1184

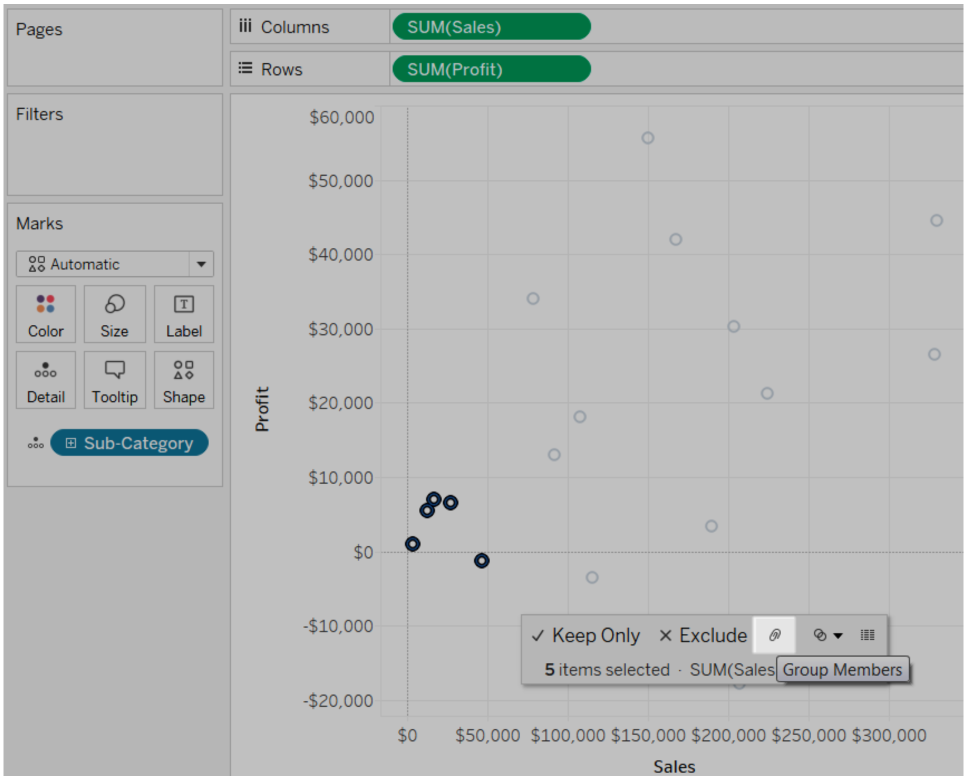
Color a View Using Groups

In addition to correcting data errors and combining dimensions, you can visually identify groups of related marks by selecting marks directly. This technique is especially useful when working with a scatter plot or a view that doesn't have headers that you can select when defining the group.

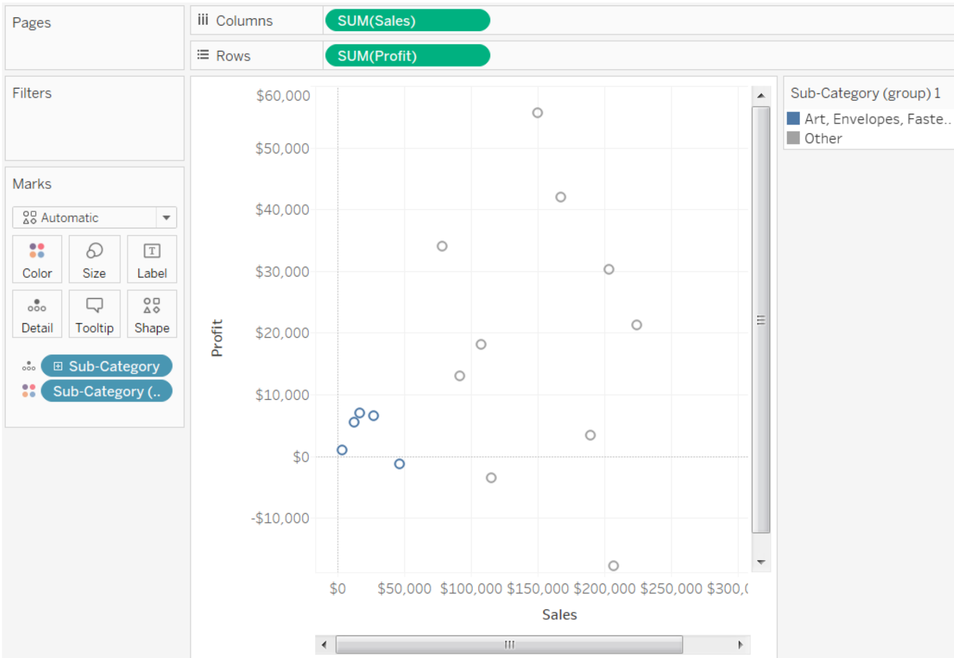
1. Press and hold the **CTRL** or **Shift** key on the keyboard to select one or more marks in the view. On a Mac, use the **Shift** key.
2. To create a group do one of the following:

    - For Tableau Desktop, click the **Group** button  on the toolbar or right-click and select **Group**.

- For Tableau Server or Tableau Online, click the **Group** icon on the tooltip.

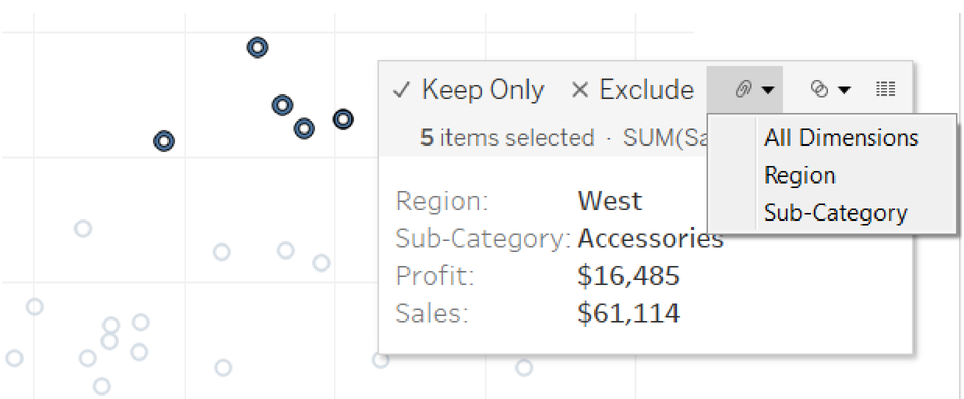


The selected marks are grouped and all other members are combined into an "Other" category. The new group field is automatically added to Color. For more information about grouping fields using Other, see **Include an Other Group** on page 1176.



Note: If there were already fields on **Color**, they are moved to **Detail** and replaced with the group field.

When you create groups by selecting marks, it's possible that the marks will represent multiple dimensions. For example, you may have a scatter plot that shows the Sales vs. Profit by Region and Sub-Category. In this case, a selection of marks will represent members of both the Region and Sub-Category dimensions. When the selection represents multiple dimensions, the **Group** menu lets you choose to group on **All Dimensions** or on a particular dimension.

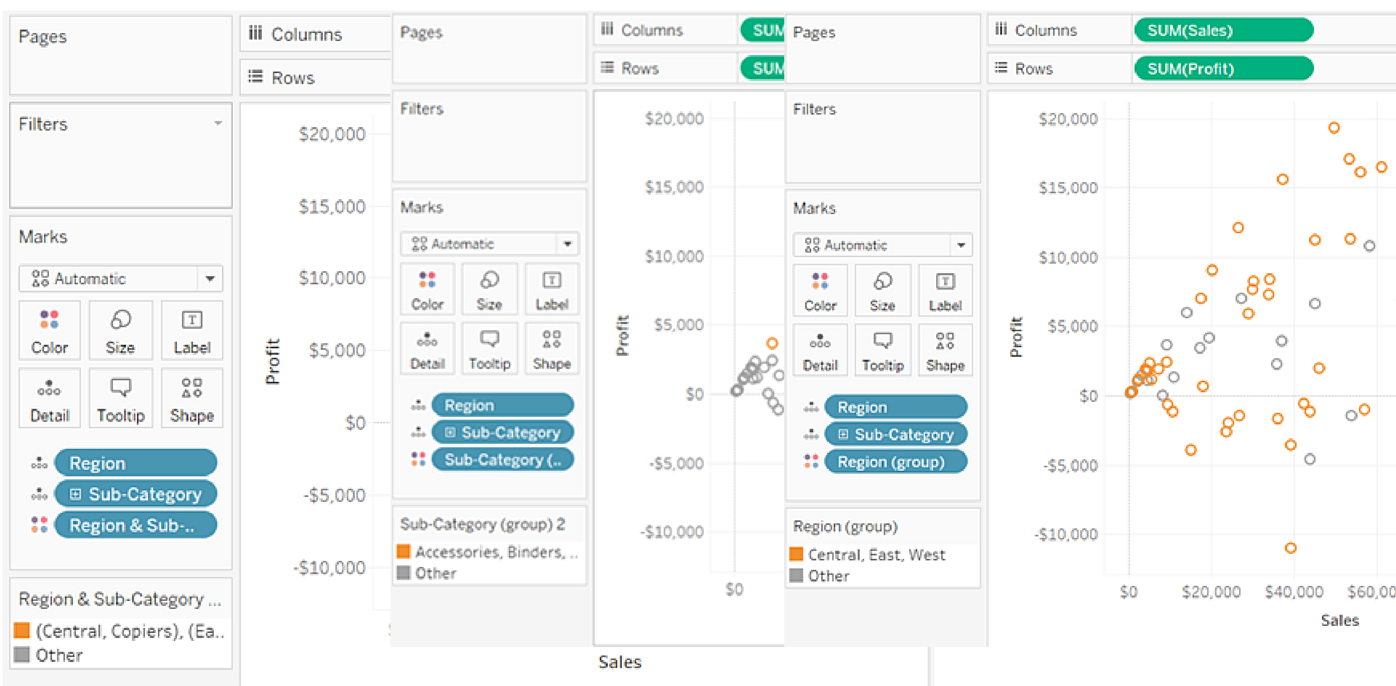


In the example above, the 5 selected marks represent a region and the following sub-category combinations .

- West, Copiers
- West, Binders
- West, Accessories
- East, Copiers
- Central, Copiers
- South, Art
- South, Envelopes
- South, Labels
- West, Envelopes
- West, Labels

The views below show the results of grouping these measures on All Dimensions, Sub-Category, and Region.

All Dimensions	**Sub-Category**	**Region**



The 5 marks are combined and the rest of the marks are added to an "Other" cat-	All marks associated with any of the three sub-categories are combined and everything	All marks associated with the four regions are combined and everything else is added

egory.	else is added to an "Other" category.	to an "Other" category.

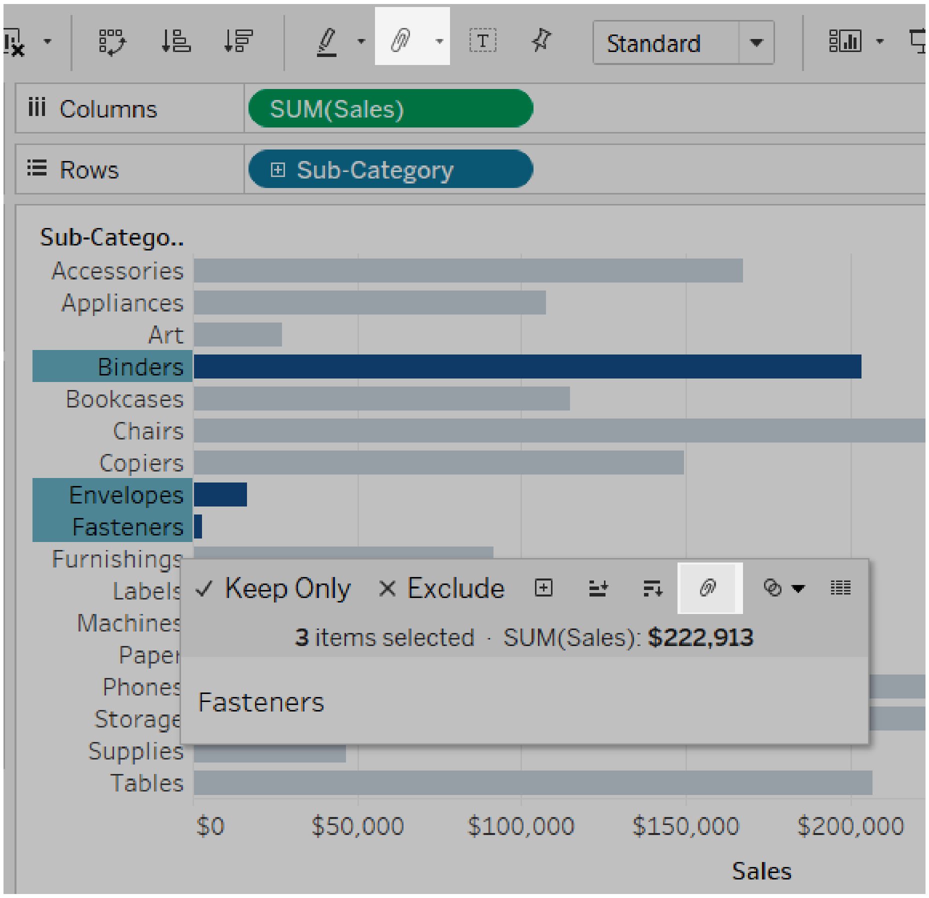
Correct Data Errors or Combine Dimension Members by Grouping Your Data

If you are using groups to correct data errors (for example, to combine "CA" and "California") or to combine dimensions members (for example, to combine "East" and "West" regions), the easiest option is to select headers in the view.

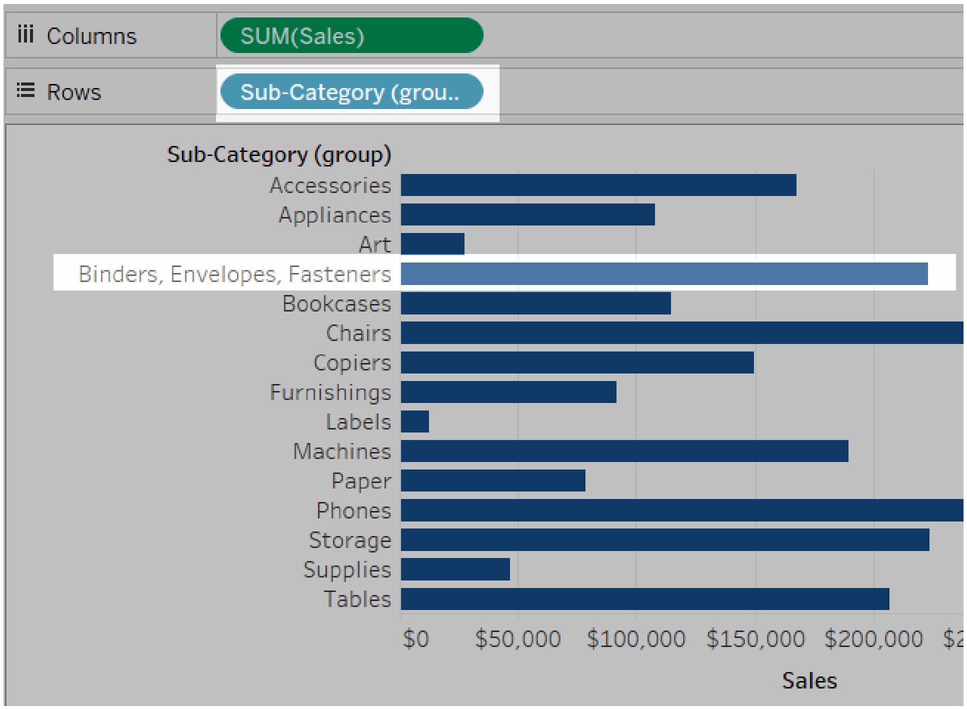
1. Press and hold the **CTRL** or **Shift** key on the keyboard to multi-select headers in the view. On a Mac, press and hold the Command key.
2. To create a group, do one of the following:

    - On Tableau Desktop, click the **Group** button  on the toolbar, right-click and select **Group** or click the **Group** icon on the tooltip.

- On Tableau Server or Tableau Online, click the **Group** icon on the tooltip.



The selected members are combined into a single member. In this example, the view now shows the SUM(Sales) across all binders, envelopes and fasteners as a single mark. A default is automatically constructed using the combined member names. The dimension on the Rows or Columns shelf is replaced with the new grouped field.



Create Sets

You can use sets to compare and ask questions about a subset of data. Sets are custom fields that define a subset of data based on some conditions.

Watch a video: To see related concepts demonstrated in Tableau, watch the free training videos Creating Sets (6:54 minutes) and Working with Sets (4 minutes). Use your tableau.com account to sign in.

You can make sets more dynamic and interactive by using them in **Set Actions** on page 1540. Set actions let your audience interact directly with a viz or dashboard to control aspects of their analysis. When someone selects marks in the view, set actions can change the values in a set.

In addition to a Set Action, you can also allow users to change the membership of a set by using a filter-like interface known as a Set Control, which makes it easy for you to designate inputs into calculations that drive interactive analysis. For details, see **Show a set control in the view** on page 1196.

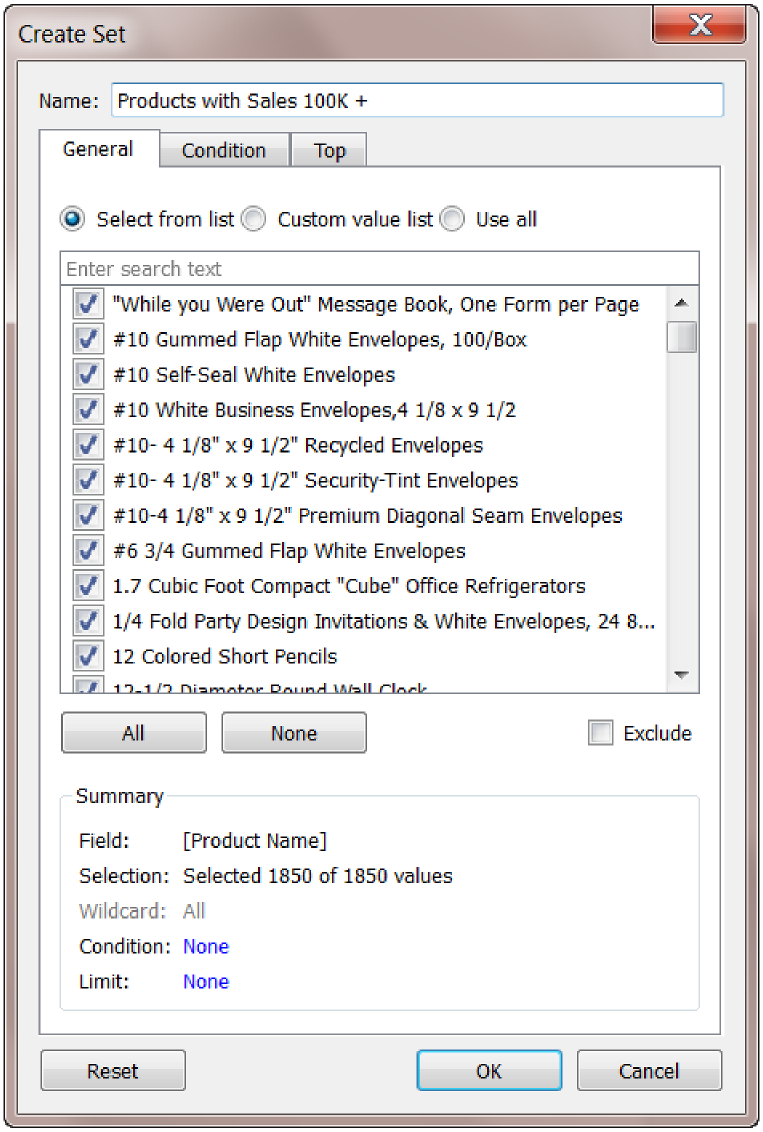
Create a dynamic set

There are two types of sets: dynamic sets and fixed sets. The members of a dynamic set change when the underlying data changes. Dynamic sets can only be based on a single dimension.

To create a dynamic set:

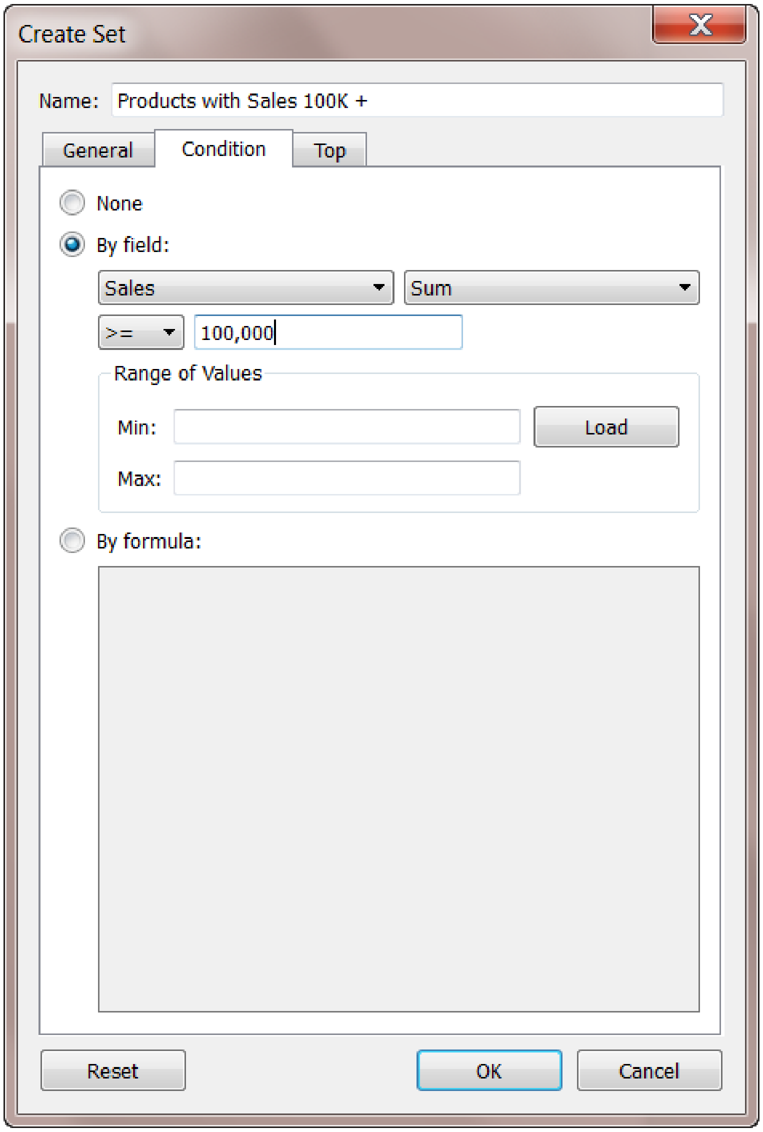
1. In the Data pane, right-click a dimension and select **Create** > **Set**.
2. In the Create Set dialog box, configure your set. You can configure your set using the following tabs:
    - **General**: Use the General tab to select one or more values that will be considered when computing the set.

      You can alternatively select the **Use all** option to always consider all members even when new members are added or removed.



- **Condition**: Use the Condition tab to define rules that determine which members to include in the set.

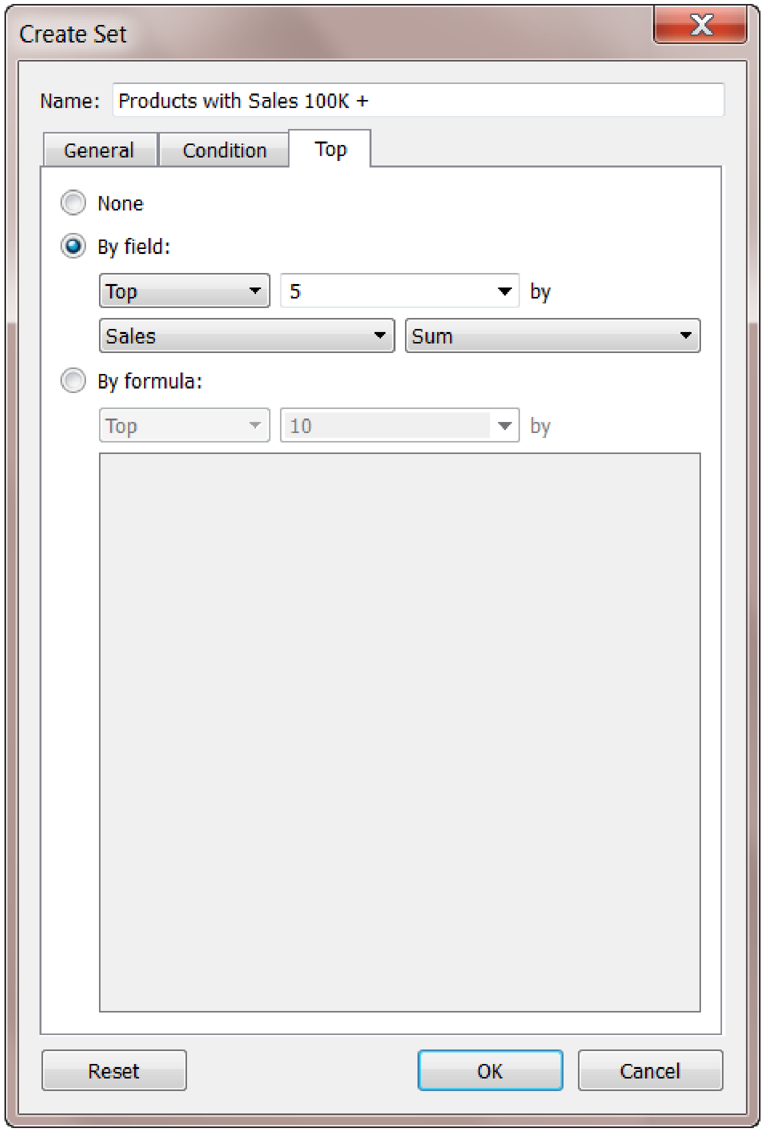
  For example, you might specify a condition that is based on total sales that only includes products with sales over $100,000.



Note: Set conditions work the same as filter conditions. See **Filter Data from Your Views** on page 1411 to learn more.

- **Top**: Use the Top tab to define limits on what members to include in the set.

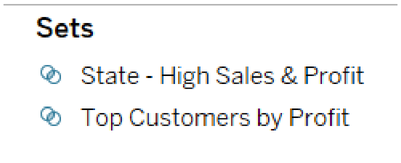
  For example, you might specify a limit that is based on total sales that only includes the top 5 products based on their sales.



Note: Set limits work the same as Filter limits. See **Filter Data from Your Views** on page 1411 to learn more.

3. When finished, click **OK**.

   The new set is added to the bottom of the Data pane, under the Sets section. A set icon indicates the field is a set.



Create a fixed set

The members of a fixed set do not change, even if the underlying data changes. A fixed set can be based on a single dimension or multiple dimensions.

To create a fixed set:

1. In the visualization, select one or more marks (or headers) in the view.

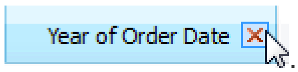
2. Right-click the mark(s) and select **Create Set**.

3. In the Create Set dialog box, type a name for the set.

4. Optionally complete any of the following:

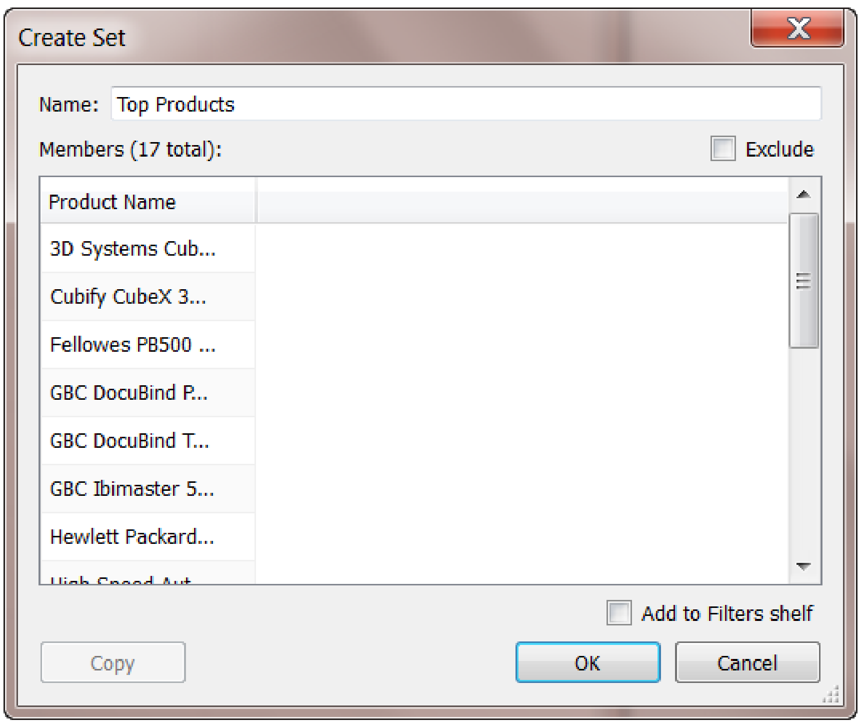
   - By default, the set includes the members listed in the dialog box. You can select the option to **Exclude** these members instead. When you exclude, the set will include all of the members you didn't select.

   - Remove any dimensions that you don't want to be considered by clicking the red "x" icon that appears when you hover over a column heading



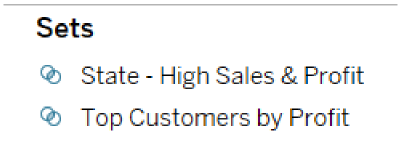
.

- Remove any specific rows that you don't want to include in the set by clicking the red "x" icon that appears when you hover over the row .
- If the marks you selected represent multiple dimensions, each member of the set will be a combination of those dimensions. You can specify the character that separates the dimension values. To do so, for **Separate members by**, enter a character of your choice.
- Select **Add to Filters shelf** to automatically move the set to the Filters shelf once it is created.



5. When finished, click **OK**.

   The new set is added to the bottom of the Data pane, under the Sets section. A set icon indicates the field is a set.

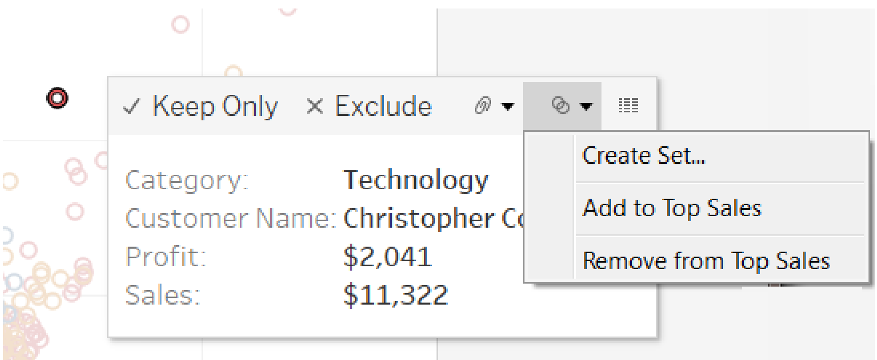


Add or remove data points from sets

If you created a set using specific data points, you can add more data to or subtract data from the set.

To add or remove data points from a set:

1. In the visualization, select the data points you want to add or remove.
2. In the tooltip that appears, click the Sets drop-down menu icon, and then select **Add to [set name]** or **Remove from [set name]** to add or remove data from a particular set.



Use sets in the visualization

After you create a set, it displays at the bottom of the Data pane in the Sets section. You can drag it into the viz like any other field.

When you drag a set to the viz in Tableau Desktop, you can choose to show the members of the set or aggregate the members into In/Out categories.

In Tableau Server or Tableau Online you can only aggregate the members of the set into In/Out categories.

Show In/Out members in a set

In most cases, when you drag a set to the viz, Tableau displays the set using the In/Out mode. This mode separates the set into two categories:

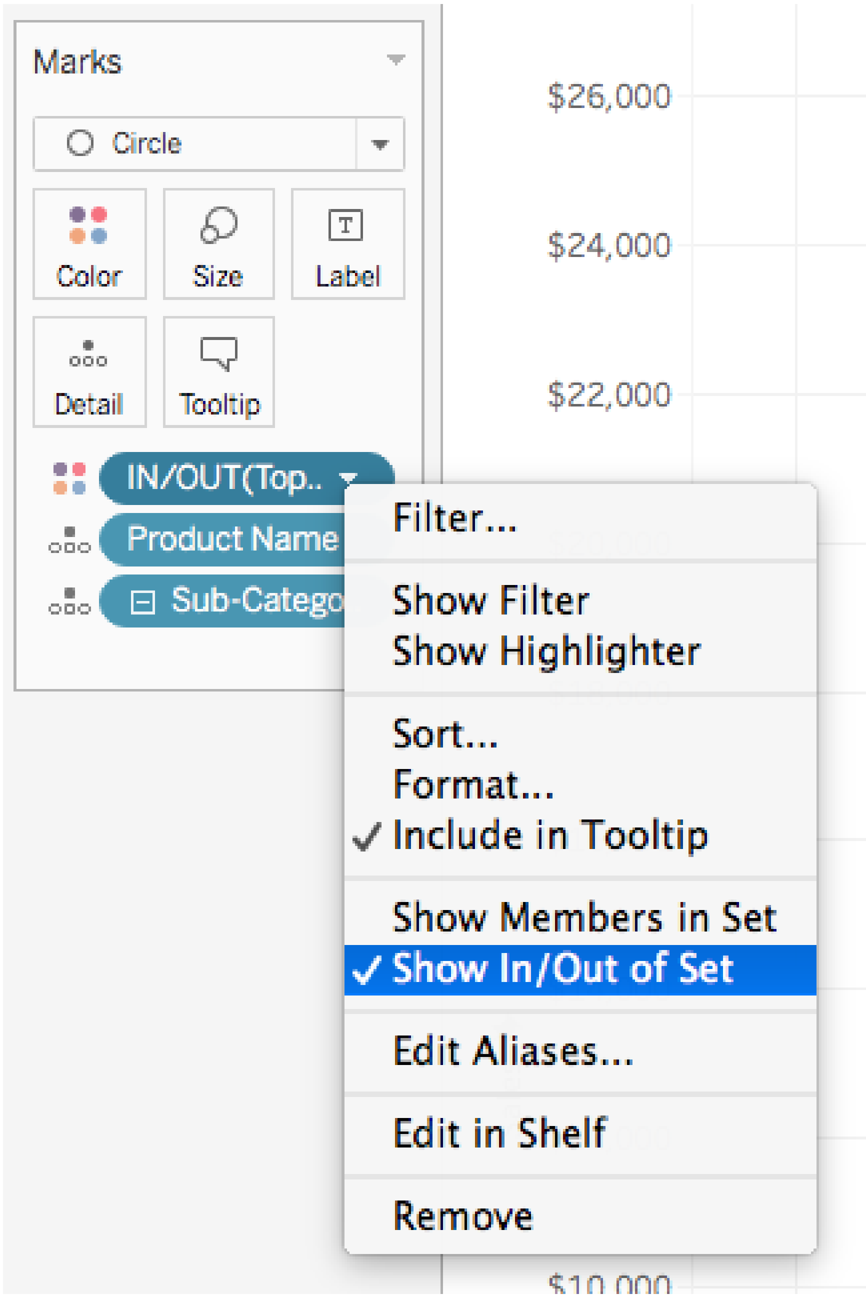
- In - The members in the set.
- Out - The members that aren't part of the set.

For example, in a set defined for the top 25 customers, the top customers would be part of the In category and all other customers would be part of the Out category.

Using the In/Out mode makes it easy to compare the members in the set to everything else.

To show In/Out members in the visualization:

- In Tableau Desktop, right-click the set in the visualization workspace and select **Show In/Out of Set**.



When a set is in In/Out mode, the field on the shelf is prefaced by the text, "IN/OUT", followed by the set name.

Note: In/Out mode is not available in workbooks created before version 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, or workbooks that use Microsoft Access data sources.

Show members in a set

As an alternative to showing the set using In/Out mode, you can list the members in the set. Showing the members in the set automatically adds a filter to the view that includes only the members of the set.

To switch a set to list the individual members:

- In the visualization workspace, right-click the set and select **Show Members in Set**.

> **Note:**To display the fully qualified member names for cubes, right-click the set in the Data pane and select **Qualify Member Names**.

Let users change set values

Add a set action

You can use set actions to give your audience more control over their analysis of your visualization.

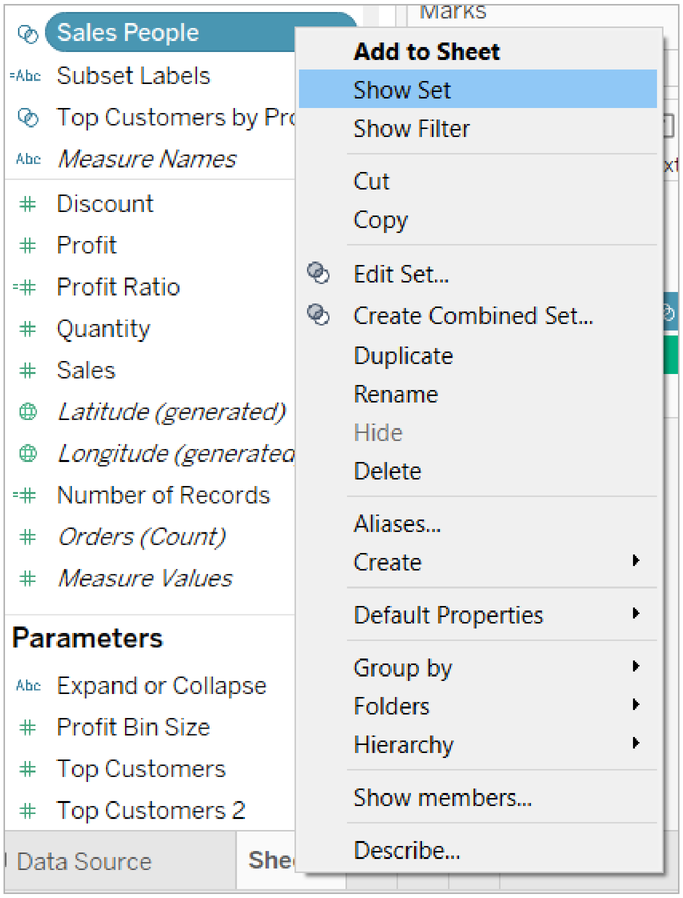
Set actions take an existing set and update the values contained in that set based on a user’s action in the viz. As the author, you can use a set or sets that you have already created to define the scope of the set action.

For details on how to create and use set actions, see **Set Actions** on page 1540.

Show a set control in the view

To give your audience the ability to quickly modify members of a set, you can also display a Set Control. A set control is a worksheet card that is very similar to a parameter control or filter card. You can add set controls to worksheets and dashboards and they are included when you publish to Tableau Server or Tableau Online, or save to the web on Tableau Public.

To display the set control, right-click (Control-click) the set in the Data pane and select **Show Set**.



Like other cards, set controls have a menu that you can open using the drop-down arrow in the upper right corner of the card. Use this menu to customize the display of the control, which supports both single-value and multiple-value selection modes. For example, you can show radio buttons for individual selection, or a drop-down list that supports multiple selections.

> **Note:** You can only display a set control for dynamic sets—not fixed sets. This is by design, as fixed sets aren't meant to change in membership. In addition, if the dynamic set is not in play in the view (that is, if it's not referenced in a calculation or instantiated on the sheet), the context menu item will be disabled, reminding you to add the set to the view.

Combine sets

You can combine two sets to compare the members. When you combine sets you create a new set containing either the combination of all members, just the members that exist in both, or members that exist in one set but not the other.

Combining sets allows you to answer complex questions and compare cohorts of your data. For example, to determine the percentage of customers who purchased both last year and this year, you can combine two sets containing the customers from each year and return only the customers that exist in both sets.

To combine two sets, they must be based on the same dimensions. That is, you can combine a set containing the top customers with another set containing the customers that purchased last year. However, you cannot combine the top customers set with a top products set.

To combine sets:

1. In the Data pane, under Sets, select the two sets you want to combine.
2. Right-click the sets and select **Create Combined Set**.
3. In the Create Set dialog box, do the following
    - Type a name for the new combined set.
    - Verify that the two sets you want to combine are selected in the two drop-down menus.
    - Select one of the following options for how to combine the sets:
        - **All Members in Both Sets** - the combined set will contain all of the members from both sets.
        - **Shared Members in Both Sets** - the combined set will only contain members that exist in both sets.
        - **Except Shared Members** - the combined set will contain all members from the specified set that don't exist in the second set. These options are equivalent to subtracting one set from another. For example, if the first set contains Apples, Oranges, and Pears and the second set contains Pears and Nuts; combining the first set except the shared members would contain just Apples and Oranges. Pears is removed because it exists in the second set.
    - Optionally specify a character that will separate the members if the sets represent multiple dimensions.
4. When finished, click **OK**.

Note: This functionality is not available in workbooks created before version 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, or workbooks that use Microsoft Access data sources.

Examples of sets

There are many ways you can use sets to answer complex questions and compare cohorts of data. Below are some examples of ways you can use sets to define and compare subsets of data.

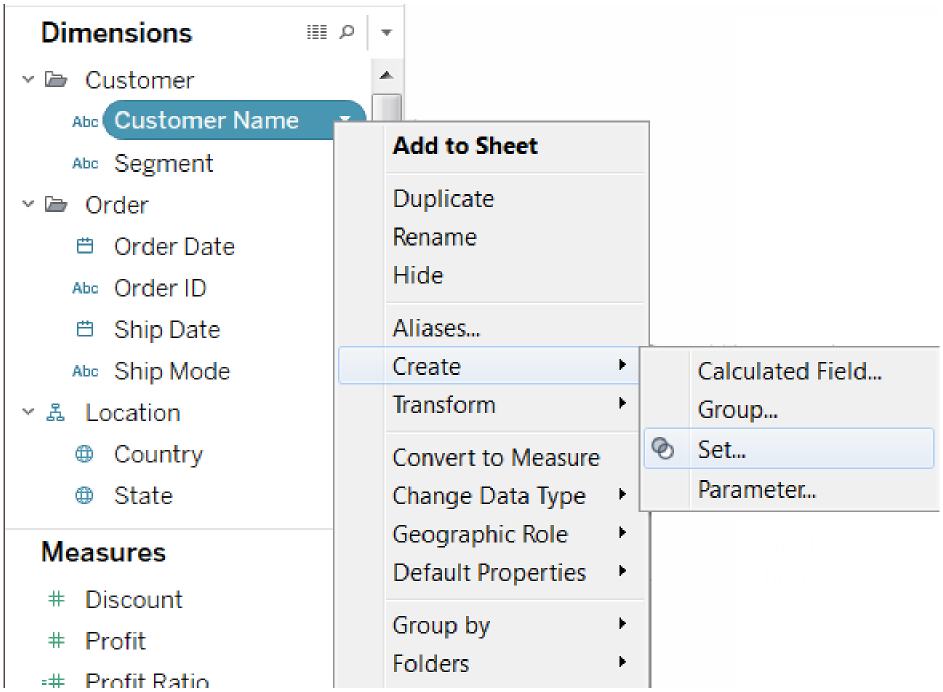
How do members of a set contribute to the total?

You may have all kinds of questions surrounding how the members in a set contribute to the overall total. For example, what percent of total sales come from repeat customers? You can answer these types of questions using the IN/OUT mode for a set.

The example below uses sales data to create a set for customers who have purchased 5,000 USD or more in products.

Create the set

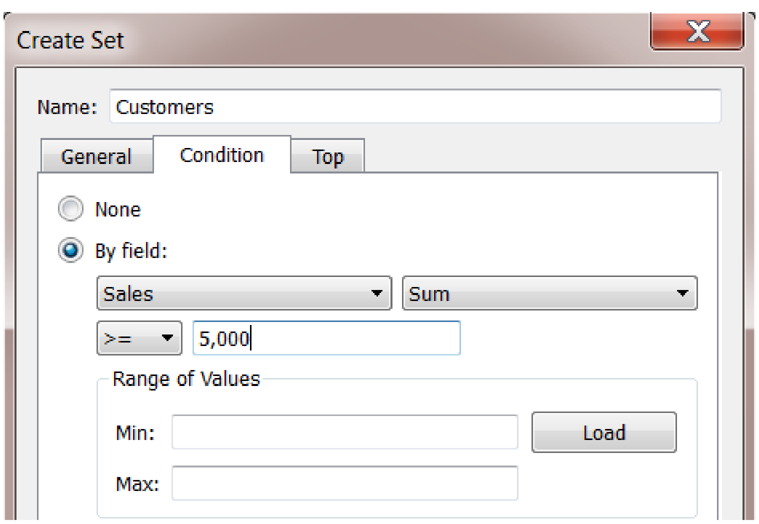
1. Right-click (control-click on Mac) the **Customer Name** dimension in the Data pane and select **Create** > **Set**.



2. In the Create Set dialog box, type a name for the set. In this example, we'll call the set, "Customers"

3. Select the **Use all** option so the condition always applies to all values even when new customers are added.

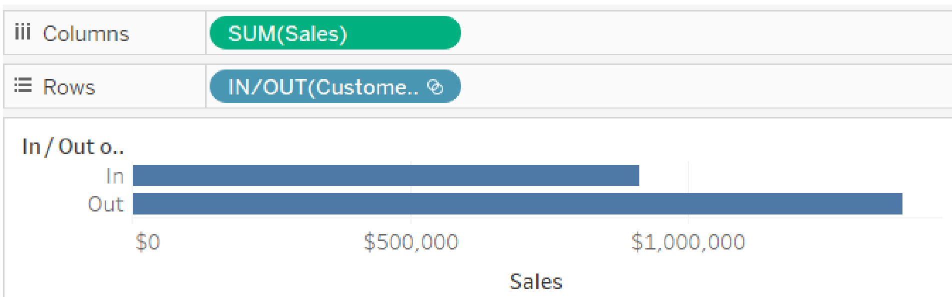
4. On the Condition tab, click **By field**, and then define a condition that only includes customers when **Sum** of **Sales** is greater than or equal to **5,000**.



5. Click **OK**.

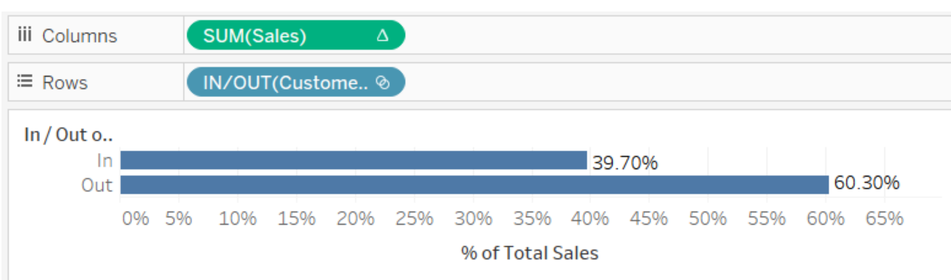
Create the visualization

1. Drag the new set from the **Sets** area at the bottom of the **Data** pane to the **Rows** shelf.
2. Drag **Sales** to the **Columns** shelf. The view now shows the total sales for customers who have purchased more than 5,000 USD of product and the total sales for all other customers.



3. Finally, click on the drop-down arrow on the **Sum (Sales)** field on the Column shelf and select **Quick Table Calculation** > **Percent of Total** on the context menu.

The view now shows that customers with sales greater than or equal to 5,000 make up about 39% of the overall sales.

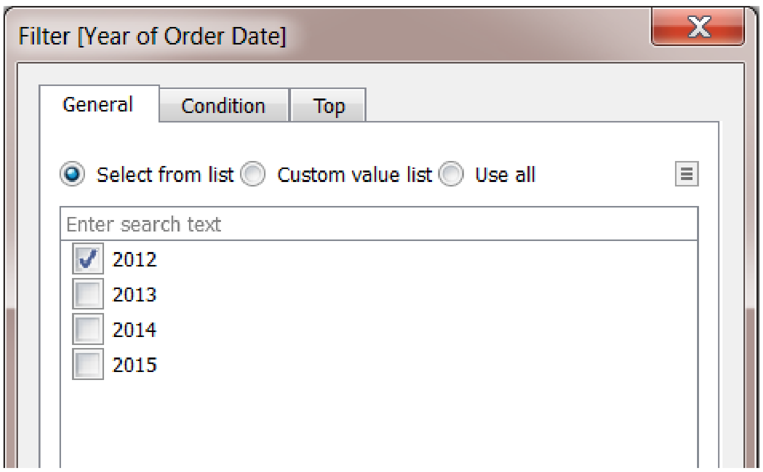


How many members of a set exist in another set?

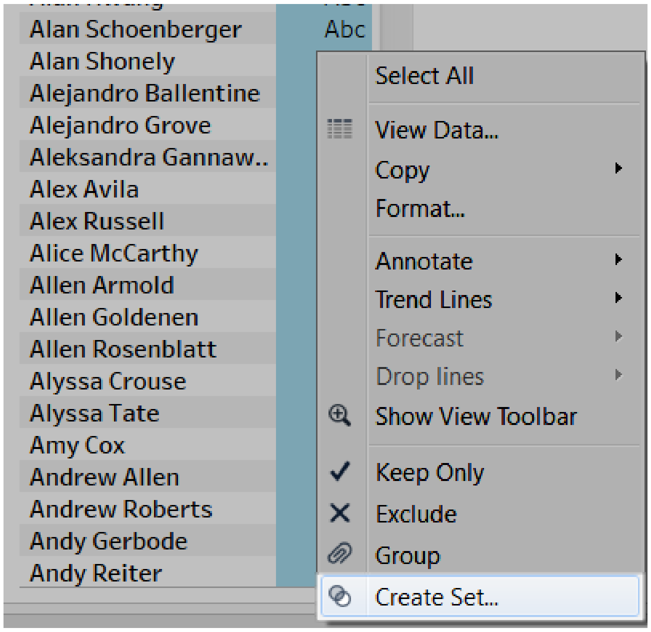
Another common use of sets is to compare subsets of data or cohorts. For example, you may wonder how many customers that purchased last year also purchased this year. Or if a customer purchased a specific product, what other products did they buy? You can answer these types of questions by creating multiple sets and combining them. The example below uses sales data to determine how many customers who purchased in 2012 also purchased in 2013.

Create a combined set

1. Drag the **Customer Name** field to the Rows shelf.
2. Drag the **Order Date** field to the Filters shelf.
3. In the Filter Field dialog box, select **Years** and click **Next**.
4. In the Filter dialog box, select **2012** and click **OK**.



5. Back in the view, press CTRL + A (Command-A on a Mac) on your keyboard to select all of the customers.
6. Right-click (control-click on Mac) the selection and select **Create Set**.

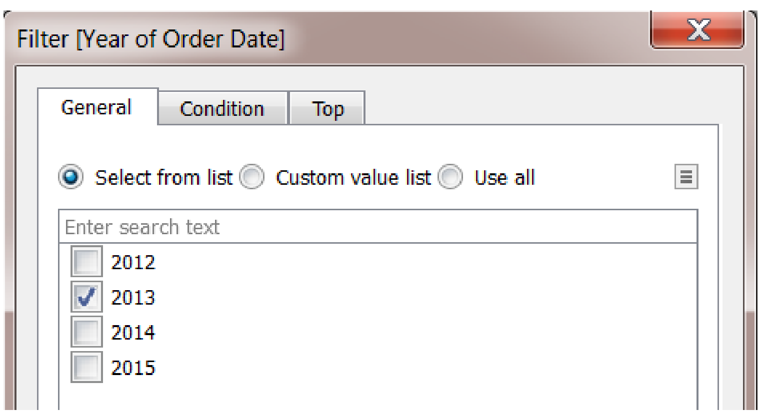


7. In the Create Set dialog box that opens, type a name for the set. In this example, we'll call the set "Customers (2012)".

8. Click **OK**.

9. On the **Filters** shelf, right-click (control-click on Mac) **Order Date** and select **Edit Filter**.

10. In the Filter dialog box, change the filter to only include **2013** instead of **2012**, and then click **OK**.



11. Again, press CTRL + A (Command-A on a Mac) on your keyboard to select all of the customers.

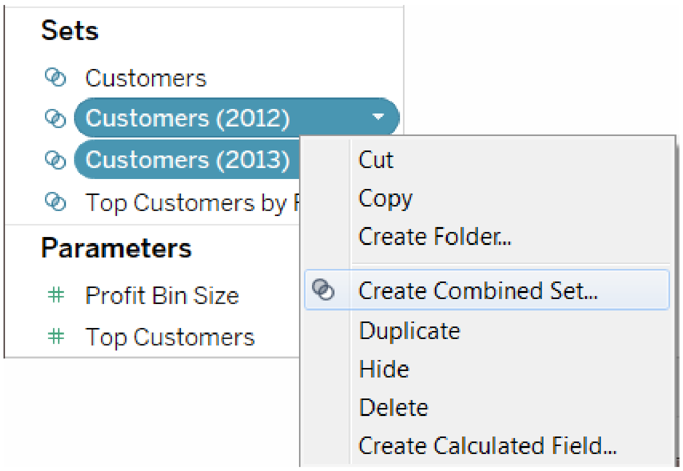
12. In the view, right-click (control-click on Mac) the selection and select **Create Set**.

13. In the Create set dialog box that opens, type a name for the set. This set will be called "Customers (2013)".

14. Click **OK**.

15. In the **Data** pane, select both the **Customers 2012** and **Customers 2013** by holding the Ctrl key (Command key on a Mac) on your keyboard as you select.

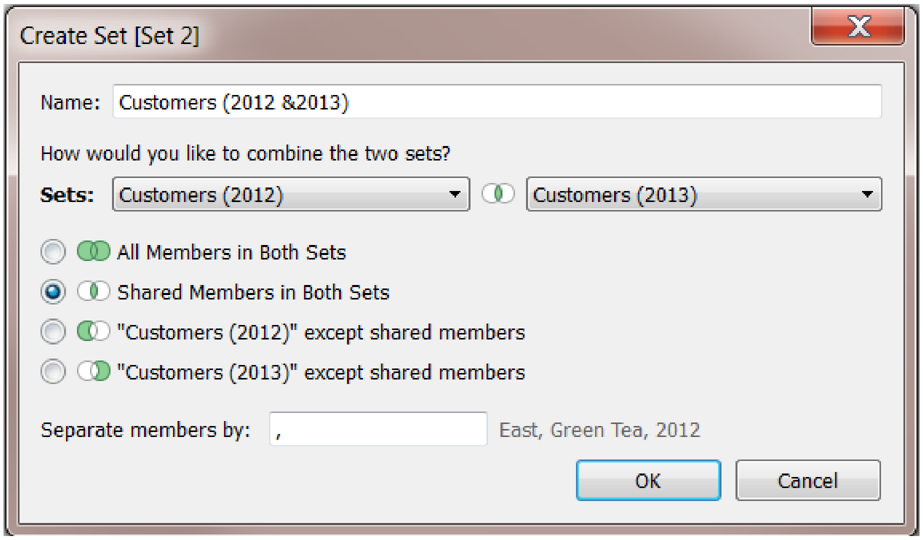
16. Right-click (control-click on Mac) the selection and select **Create Combined Set**.



17. In the Create Set dialog box, type a name for the new set. In this example, we'll call the set "Customers (2012 & 2013)".

18. Make sure the correct two sets are selected in the drop-down menus.

19. Select the option to include **Shared Members in Both Sets**.

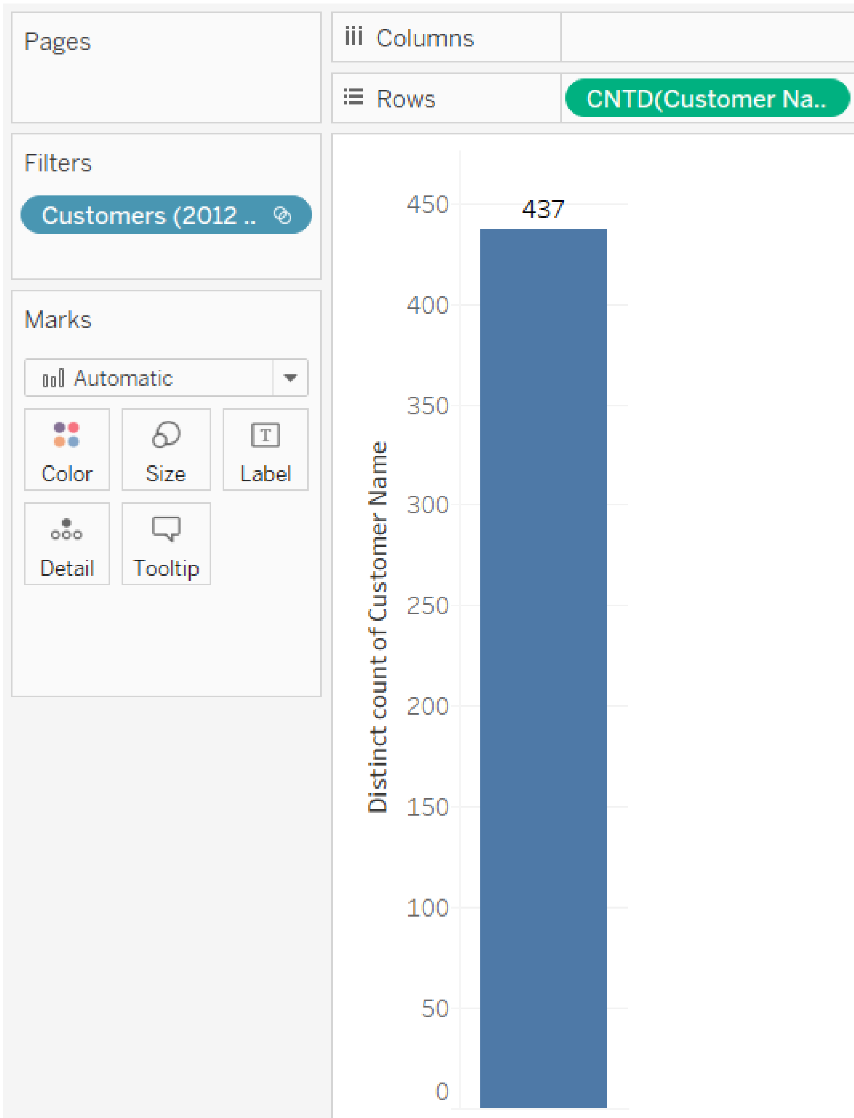


20. Click **OK**.

Create the visualization

1. At the bottom of the workbook, click the New Worksheet icon.
2. In the new worksheet, drag the **Customer Name** dimension to the **Rows** shelf.
3. Click the drop-down arrow on the Customer Name field on the Rows shelf and select **Measure** > **Count (Distinct)** from the context menu.
4. Finally, from the **Sets** area of the **Data** pane, drag the **Customers (2012 & 2013)** field to the **Filters** shelf. You can see that 437 customers purchased products in both 2012

and 2013.

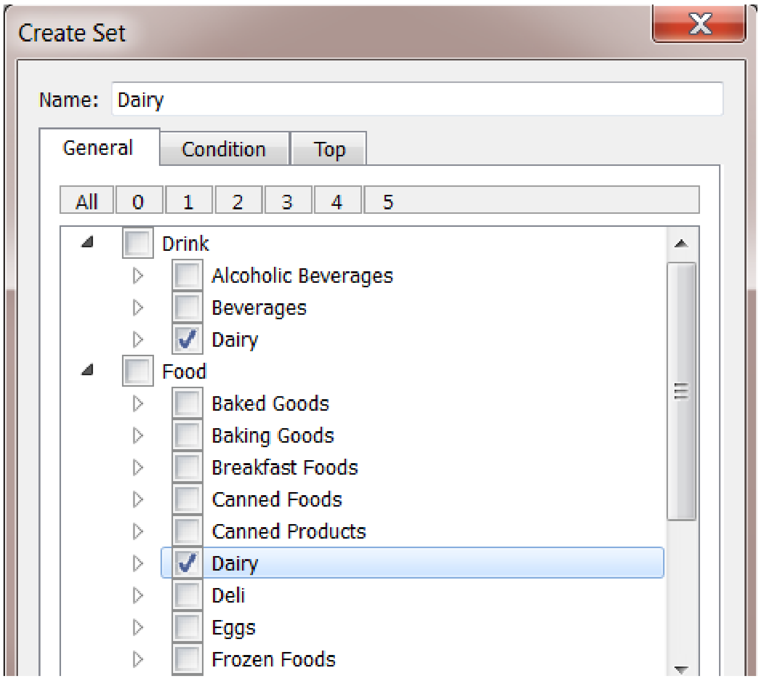


Hierarchical sets and descendants

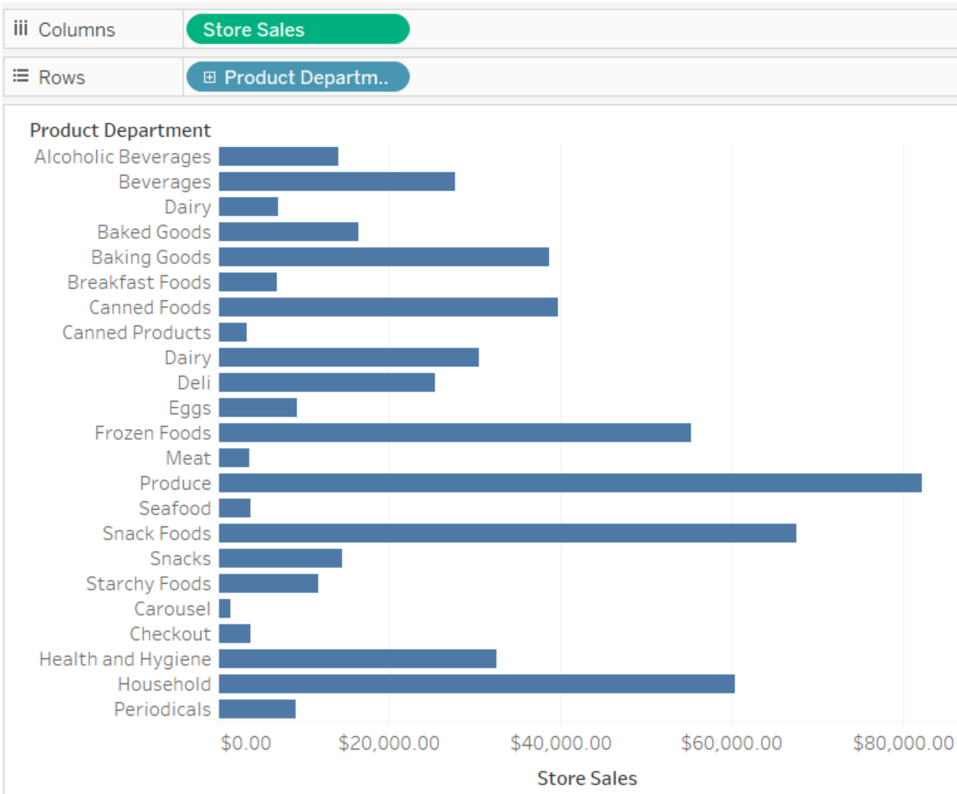
A hierarchical set filters data to the selected members and all of their descendants. They are unique to multidimensional (cube) data sources and are defined within the data source prior to connecting to Tableau Desktop.

When you create sets in Tableau from a cube data source, descendants and any hierarchical structures are automatically included with the selected members.

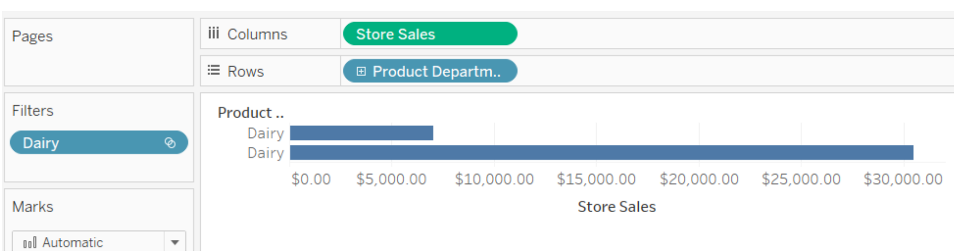
For example, a set named **Dairy** is created from the **Product** hierarchy. As shown below, it includes only the Dairy product department.



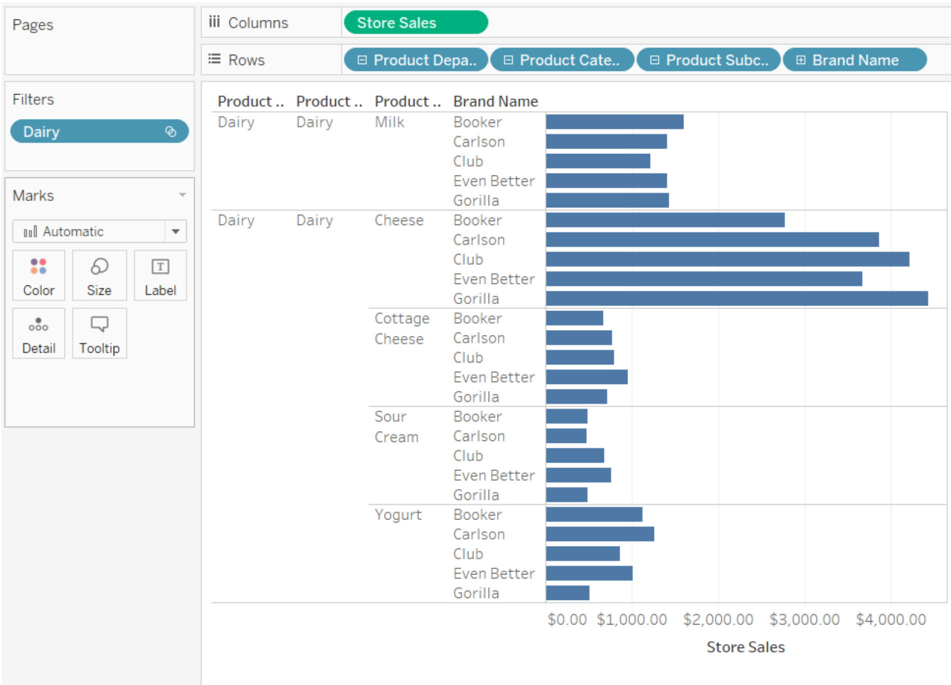
Consider the following view. The **Product Department** dimension is placed on the **Rows** shelf and the **Store Sales** measure is placed on the **Columns** shelf.



If you place the **Dairy** set on the **Filters** shelf, you can see that the view is filtered to include only the Dairy product categories.



As shown below, you can drill down into **Product Department** to reveal the **Product Category**, **Product Subcategory**, and **Brand Name** levels. As these descendants are revealed, row headers are added to the view. This is because a set filter allows you to view the levels of detail contained within the filtered members.



Sets for Top N and Others

If you collect large sets of data that you want to visualize, you might find that limiting the amount of information displayed to an important subset of records helps you work with and answer questions about the data more effectively.

This article describes how to create an interactive view that separates your customers into two dynamic groups:

- The top N customers
- All other customers

The view includes a control that your users can adjust to change the number of customers included in the top customers group. When they change the number, the view updates accordingly.

Sets and supported data sources

The method described in this article for creating a view of the top customers uses the In/Out functionality of sets.

Sets were introduced with Tableau Desktop version 8.0.

For live connections, the In/Out functionality requires a relational or multidimensional data source.

If you use a file-based data source, such as a Microsoft Excel workbook or text file, you can take an extract with which you can create sets.

Step 1: Create the parameter

1. In Tableau Desktop, open a new workbook and connect to the **Sample-Superstore** data source.
2. Open a new worksheet.
3. In the Data pane, click the drop-down arrow in the upper right corner and select **Create Parameter**.
4. In the Create Parameter dialog box, do the following:
    - In the **Name** text box, type **Top Customers 2**.
    - For **Data type**, select **Integer**.
    - For **Current value**, type **5**.
    - For **Allowable values**, click **Range**.
    - Under Range of values, do the following:
        - Click **Minimum** and type **5**.
        - Click **Maximum** and type **20**.
        - Click **Step size** and type **5**.

This parameter will be used, in combination with the top N set you will create in the next step, to quickly adjust the top N value in the view.

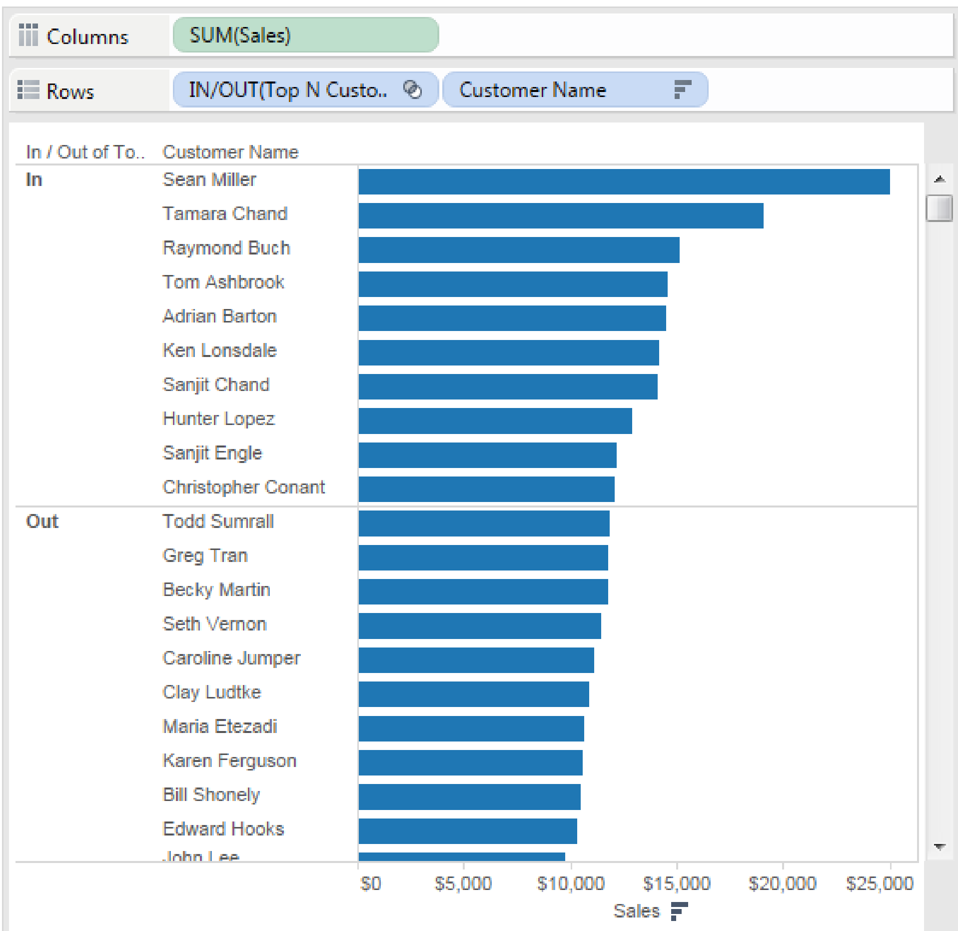
Step 2: Create the top N customers set

1. In the Data pane, right-click the **Customer Name** dimension and select **Create** > **Set**.
2. In the Create Set dialog box that opens, do the following:
    - In the **Name** text box, type **Top N Customers by Sales**.
    - Click the **Top** tab.
    - Select **By Field**.
    - From the field drop-down list (Category), select **Sales**.

- From the aggregation drop-down list, select **Sum**.
- When finished, click **OK**.

Step 3: Set up the view

1. From the Data pane, under **Sets**, drag **Top N Customers by Sales** to the **Rows** shelf.
2. Drag the **Customer Name** dimension to the **Rows** shelf, positioning it to the right of the set.
3. Drag the **Sales** measure to the **Columns** shelf.
4. On the toolbar, click the Sort Descending button to make sure that the set is working.



5. In the Data pane, right-click the **Top N Customers by Sales** set, and then click **Create Calculated Field**.

6. In the Calculated Field dialog box that opens, complete the following steps:

   - In the **Name** text box, type **Subset Labels**.
   - In the **Formula** text box, type the following formula to create dynamic labels for the customers in the set:

     ```
     IF [Top N Customers by Sales]
      THEN "Top " + str([Top Customers 2]) + " Customers"
      ELSE "Others"
      END
     ```

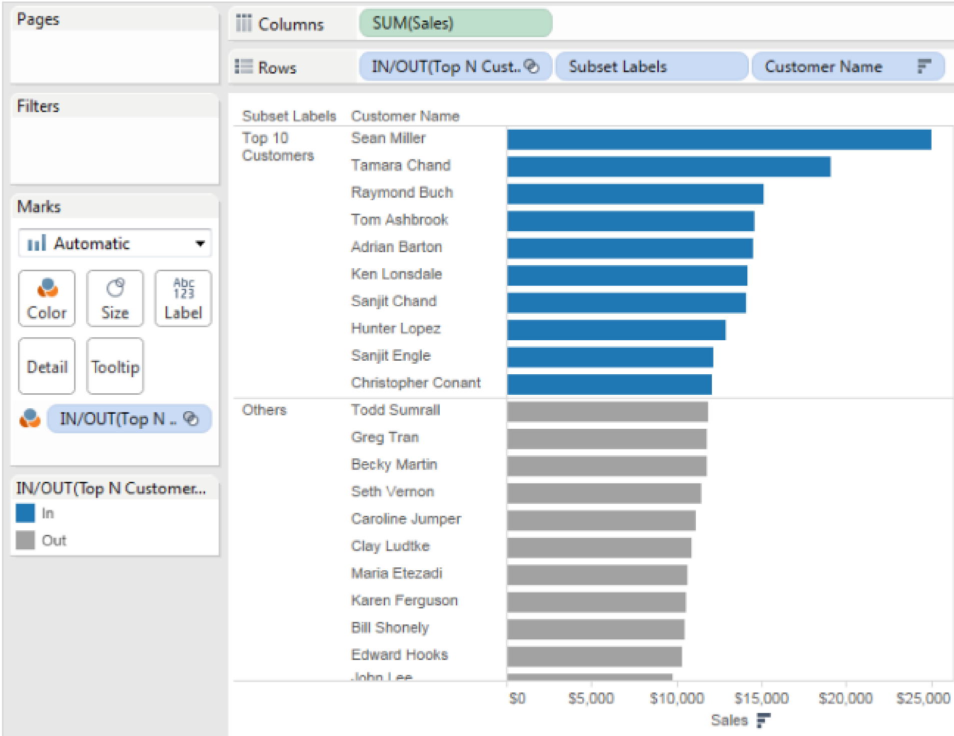
   - When finished, click **OK**.

7. From the Data pane, drag **Subset Labels** to the **Rows** shelf, placing it between the Top N set and the **Customer Name** dimension.

8. On the **Rows** shelf, right-click the **IN/OUT(Top N Customers by Sales)** set, and then clear **Show Header**.

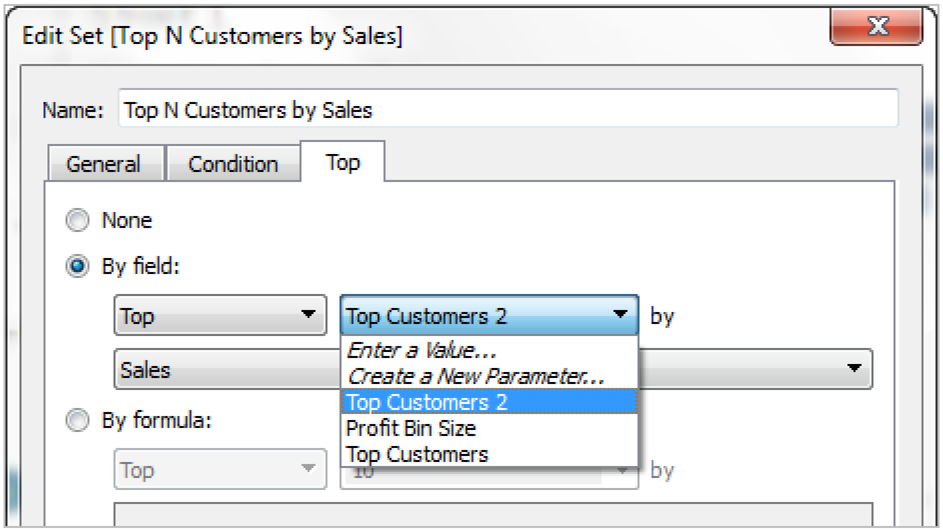
   This hides the In/Out labels while retaining the sort order so that your top N subset always appears at the top of the view.

9. From the Data pane, drag the **Top N Customers by Sales** set to **Color** on the **Marks** card.



Step 4: Combine the Top N set with a dynamic parameter

1. In the Data pane, right-click **Top N Customers by Sales**, and then select **Edit Set**.
2. In the Edit Set dialog box, do the following:
   - Select the **Top** tab.
   - Click the value drop-down menu, and select the **Top Customers 2** parameter.
   - Click **OK**.

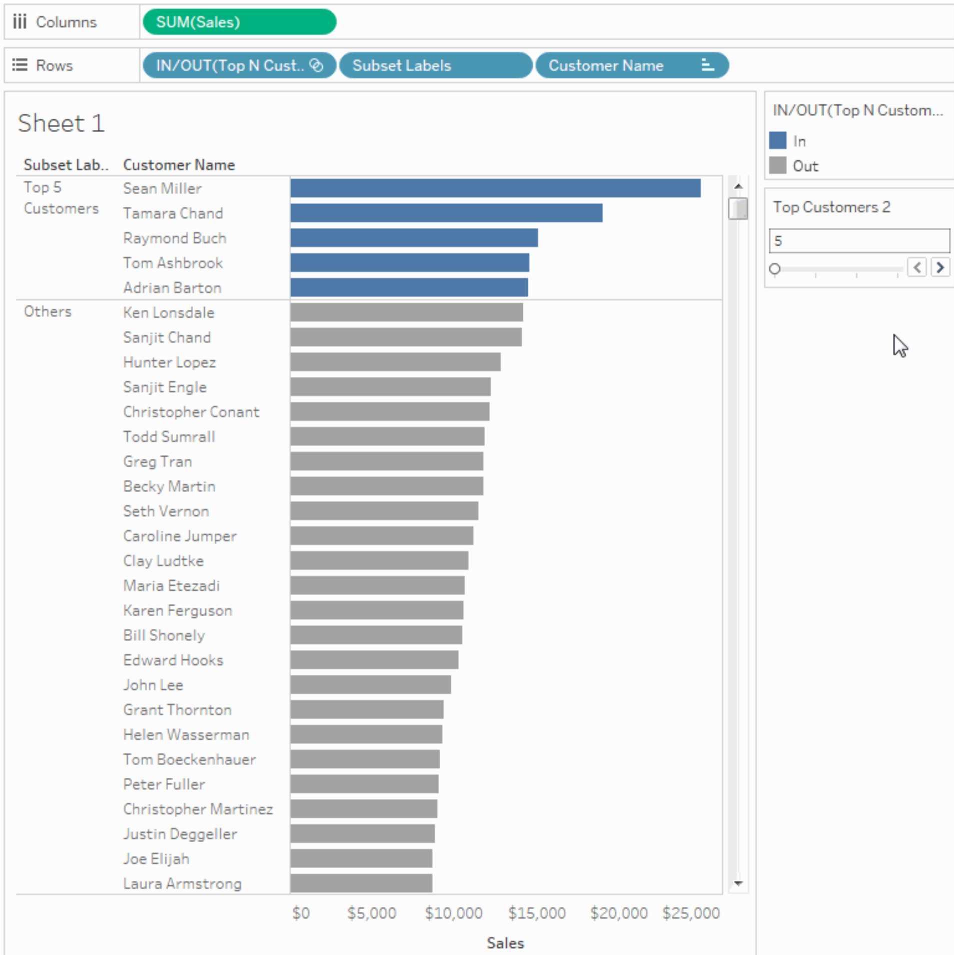


This links the **Top N Customers by Sales** set to the **Top Customers 2** dynamic parameter, instead of to a static list of 10.

This parameter will be used in combination with the Top N Customers by Sales set, to adjust the top N value in the view.

3. In the Data pane, under Parameters, right-click the **Top Customers 2** parameter, and select **Show Parameter**.

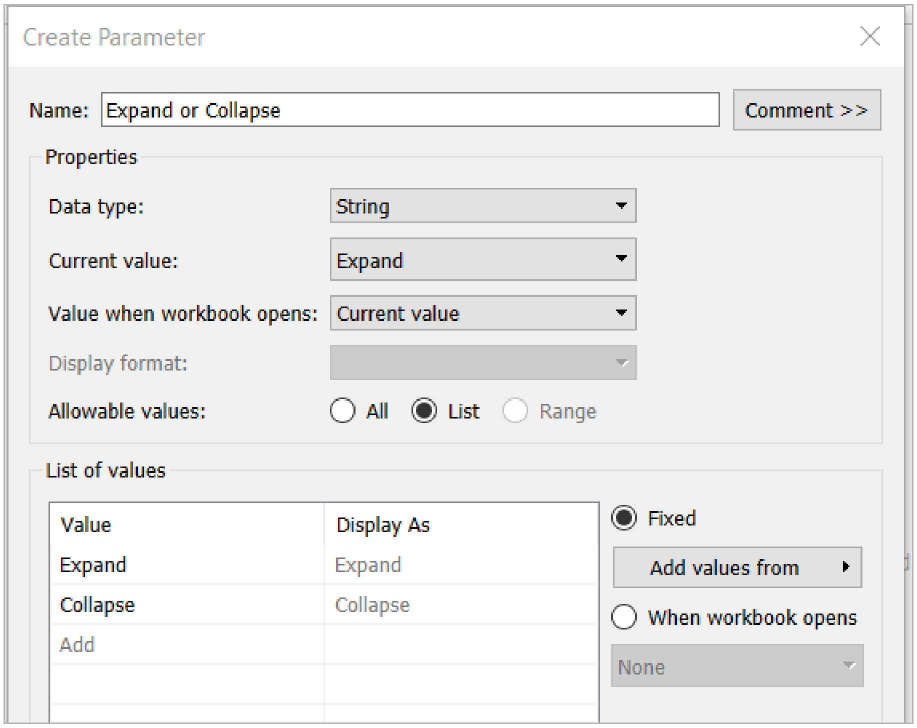
   You can control the top N value by using the **Top Customers 2** parameter control that appears in the view.



Additional tips for improving the view's functionality

Here are some additional steps you can take to give your viewers more flexibility in displaying the customer subsets.

1. From the Data pane drop-down menu, select **Create Parameter**.
2. In the Create Parameter dialog box that opens, do the following:
   - For **Name**, type **Expand or Collapse**.
   - For **Data type**, select **String**.
   - For **Allowable values**, select **List**.
   - In the **List of values**, type the values **Expand** and **Collapse**.



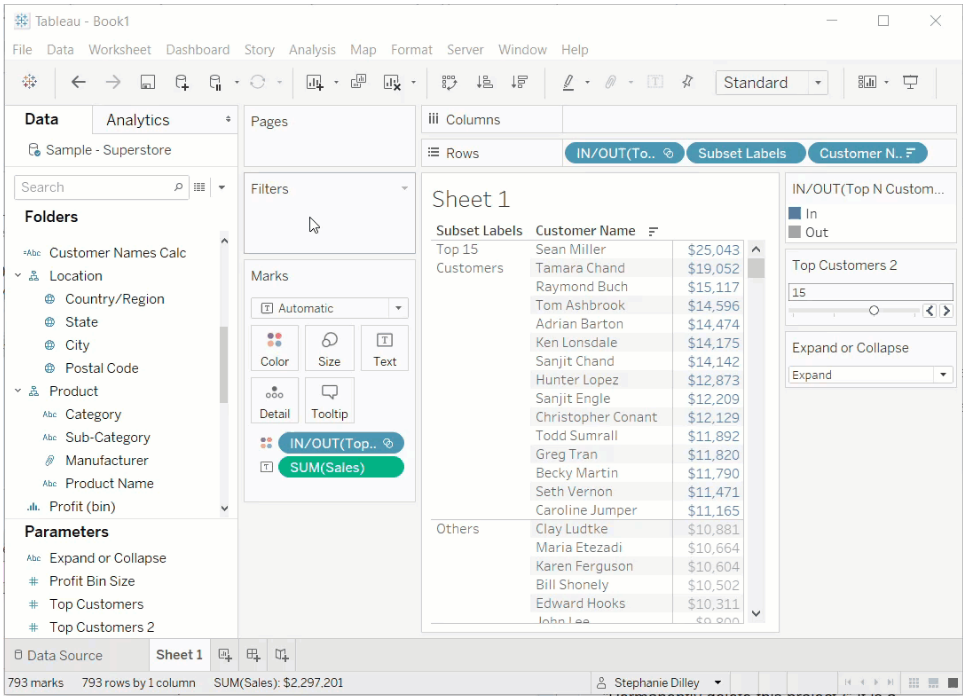
- When finished, click **OK**.

3. Select **Analysis** > **Create Calculated Field**.
4. In the Create Calculated Field dialog box that opens, do the following to create a calculation that uses the parameter you just created. This calculation enables viewers to specify how to view customers in the Others subset:
   - For **Name**, enter **Customer Names Calc**.
   - In the formula box, type the following formula, and then click **OK**:

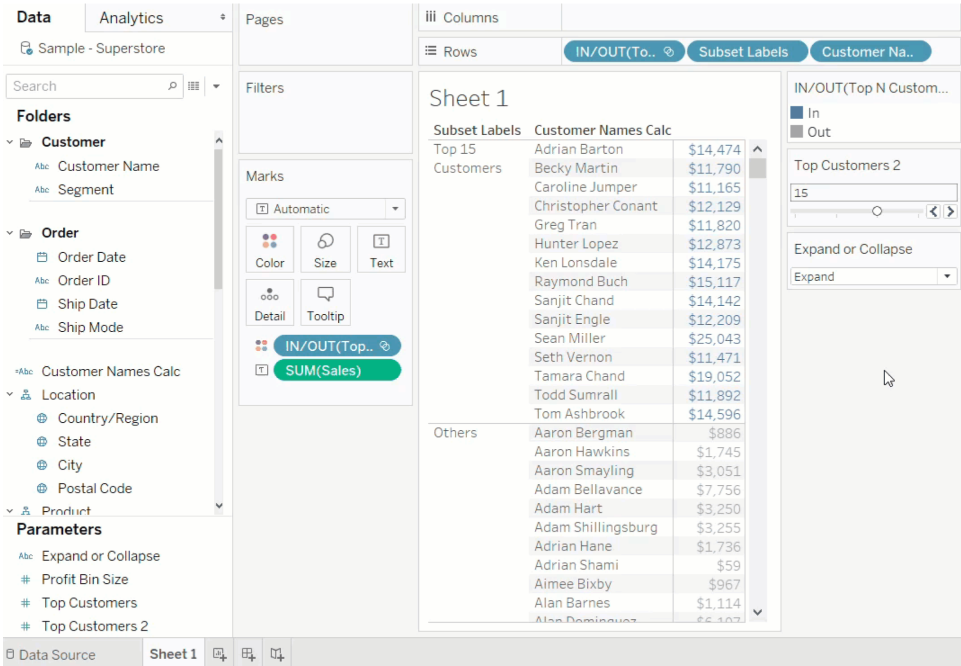
```
IF [Expand or Collapse]="Collapse" THEN
 IF [Top N Customers by Sales]
 THEN [Customer Name]
 ELSE "Others" END
 ELSE [Customer Name] END
```

5. In the Data pane, under Parameters, right-click the **Expand or Collapse** parameter, and select **Show Parameter**.
6. From the **Columns** shelf, drag the **SUM(Sales)** measure to **Label** on the **Marks** card.
7. From the Data pane, drag **Customer Names Calc** directly on top of the **Customer**

Name field on the **Rows** shelf, so that it replaces it.



Now you can use the **Expand or Collapse** parameter control to see the list of names in the top N customers and the remaining customers rolled up into a single **Others** entry.



Create Parameters

A parameter is a workbook variable such as a number, date, or string that can replace a constant value in a calculation, filter, or reference line.

For example, you may create a calculated field that returns True if Sales is greater than $500,000 and otherwise returns False. You can replace the constant value of “500000” in the formula with a parameter. Then, using the parameter control, you can dynamically change the threshold in your calculation.

You can even create a *dynamic* parameter that’s set to automatically refresh its current value (to the result of a single-value, view-independent calculation), list of values (based on a data source column), or range of values. This will happen each time the workbook is opened and Tableau connects to the data source referenced by the parameter, or whenever you select **Refresh** from the data source’s context menu..

You can make your parameters more dynamic and interactive by using them in **Parameter Actions** on page 1516. Parameter actions let your audience change a parameter value through direct interaction with a viz, such as clicking or selecting a mark.

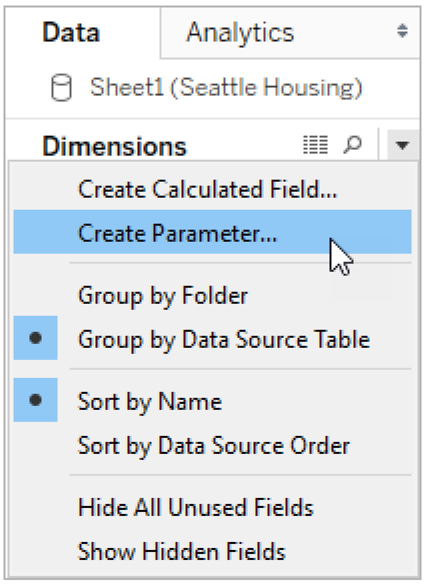
Watch a video: To see related concepts demonstrated in Tableau, watch the free training video Parameters (5 minutes). Use your tableau.com account to sign in. For a more in-depth presentation, see Parameters | Oh, the places you'll go! (45 minutes).

Learn more: See Use Parameters to Add Multiple Views to Your Viz on the Tableau blog and multiple practical topics on how to use parameters on the Data School website, such as Four Steps to Creating a Parameter.

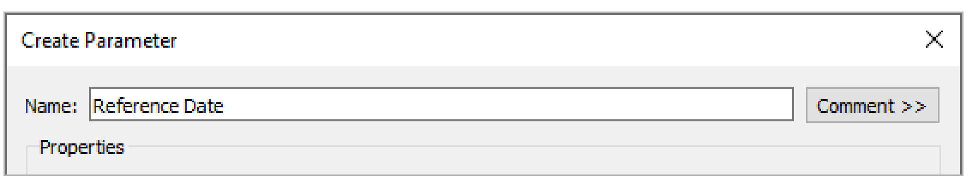
Create a parameter

Follow the instructions below to create a new parameter from the Data pane.

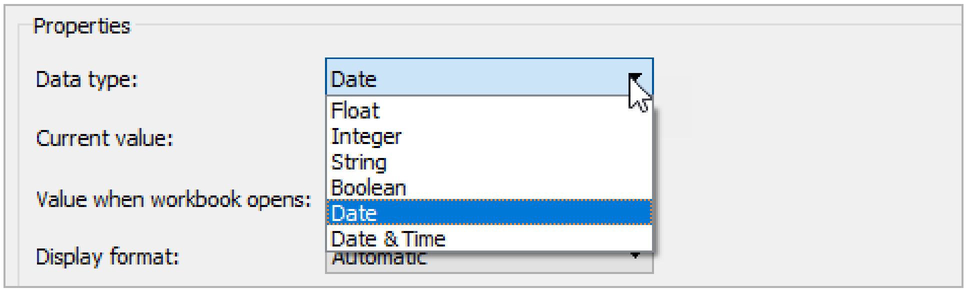
1. In the Data pane, click the drop-down arrow in the upper right corner and select **Create Parameter**.



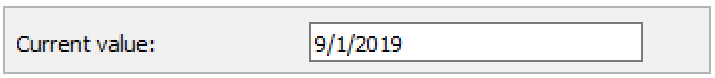
2. In the Create Parameter dialog box, give the field a **Name**.



3. Specify the data type for the values it will accept:

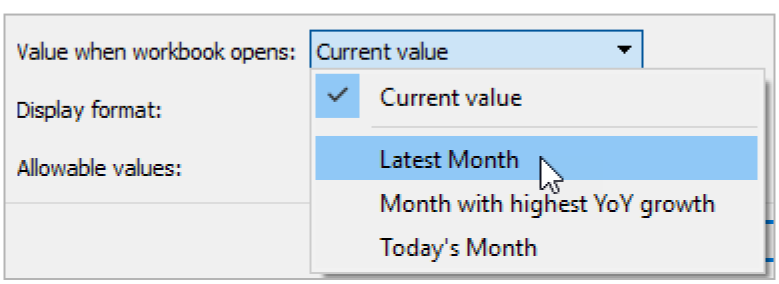


4. Specify a current value. This is the default value for the parameter. In this case, let’s leave the field as is because we’ll be using the latest data, which we’ll configure in the next step.

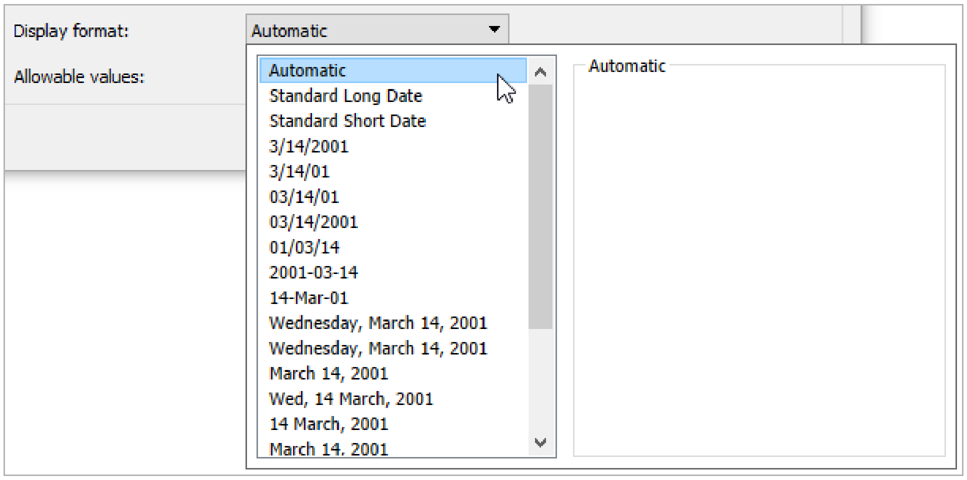


5. Specify a value when the workbook opens. In this case, let’s create a dynamic parameter by setting the parameter’s default value to the result of a single-value, view-independent calculation.
Note: If there is more than one value, the workbook would not be able to choose a default value. The calculation must also be view-independent so that the value won’t change as the viz changes. To do this, you can use a FIXED level of detail (LOD) expression that is not dependent on the structure of the viz. All parts of the calculation must be inside the FIXED LOD expression. If you use a FIXED LOD expression as the default value and are using context filters, the dynamic parameter will not reflect any context filters. For more information on LOD expressions, see **Create Level of Detail Expressions in Tableau** on page 2449.

   For this dynamic parameter, let's use **Latest Month**. This means that if the connected data source is updated and the workbook is opened, the parameter will automatically update when the workbook is opened.



6. Specify the display format to use in the parameter control (Tableau Desktop only).

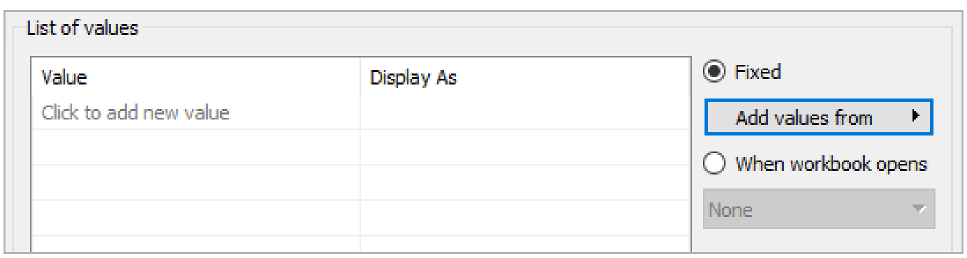


7. Specify how the parameter will accept values. You can select from the following options:

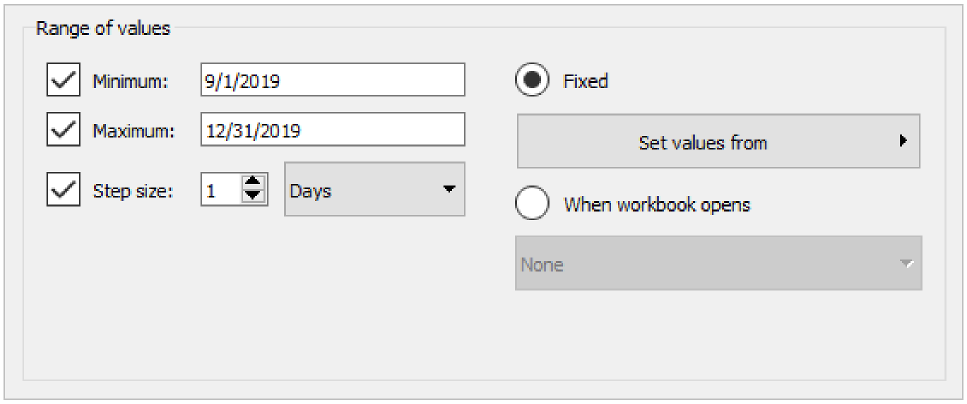
   - **All** - The parameter control is a simple text field.
   - **List** - The parameter control provides a list of possible values for you to select from.
   - **Range** - The parameter control lets you select values within a specified range.

   The availability of these options is determined by the data type. For example, a string parameter can only accept all values or a list. It does not support a range.

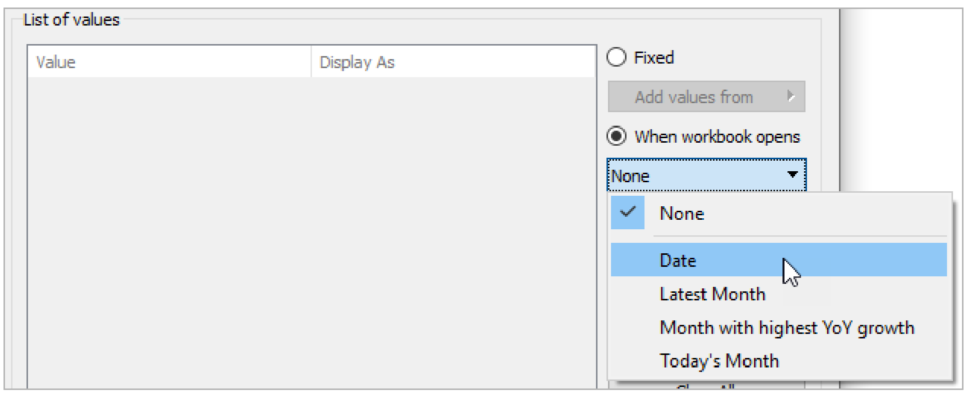
   If you select List, you must specify the list of values. Click in the left column to type your list of values, or you can add members of a field by selecting **Add values from**.



   If you select Range, you must specify a minimum, maximum, and step size. For example, you can define a date range between January 1, 2019 and December 31, 2019, with the step size set to 1 month to create a parameter control that lets you select each month in 2019.

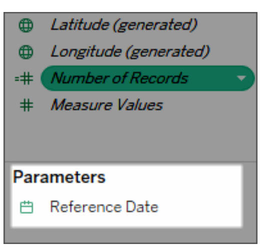


In this case, to refresh the parameter's list of values (or domain) whenever the workbook opens, select **List**, and then select **When workbook opens**. Notice that the list of values on the left is grayed out because the workbook is dynamically pulling values from the data source.



8. When finished, click **OK**.

   The parameter is now listed in the Parameters section at the bottom of the **Data** pane.



The parameter is also available everywhere else you can use a parameter—for example, on the Top tab in the Filter dialog box, or in the Reference Line dialog box. Parameters are global across the workbook and can be used in any worksheet.

When the parameter value or list of values can't refresh

Below are a few scenarios in which a default parameter value or a refreshable list of parameter values (domain) will not update as expected:

- The default field returns a value whose data is incompatible with the parameter's data type.
- The default field doesn't return a single value (for the parameter's current value).
- The default field returns null.
- The default field is in a data source that's not yet connected.
- The default field is no longer found in the workbook's namespace (i.e. it's been deleted).
- The user cancels the query to the data source while Tableau is attempting to connect.

Note: On Desktop, these queries are initially evaluated when the workbook is opened and Tableau connects to the fields' data sources for the first time. You can also evaluate the queries by refreshing the data source. To do this, press F5, or open the data source's context menu and select **Refresh**. On Server and Online, you can refresh the data source by clicking the Data Source Refresh button in the toolbar. In this case, however, remember that the value returned depends on the server's cache policies.

If the default fields fail to return values, the parameter will behave as follows:

- The current value will keep the latest valid value.
- The list of values will be empty because Tableau doesn't serialize the values returned from the query in the workbook.

The current value must be in the list of values for it to be assigned to the parameter. If the list of values is empty, the parameter will get assigned a fallback value according to the data type (1 for integer, 1.0 for float, "" for string, and the current date for date and datetime).

Edit a parameter

You can edit parameters from the Data pane or the parameter control. Follow the instructions below to edit a parameter:

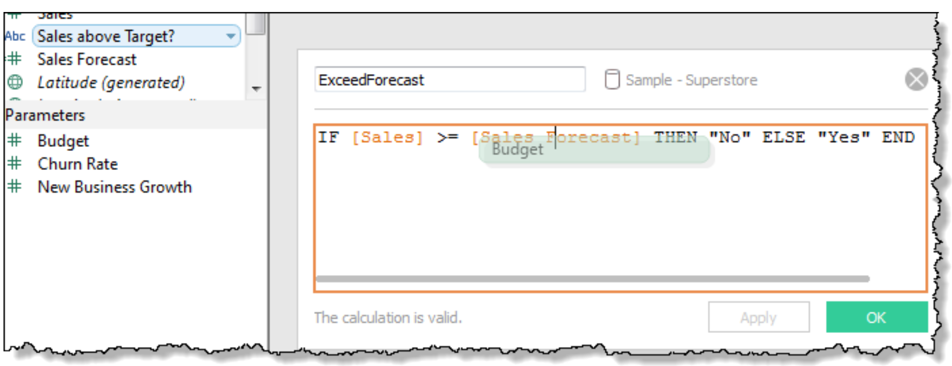
1. Do one of the following:
    - Right-click (Control-click on a Mac) the parameter in the **Data** pane and select **Edit**.
    - Select **Edit Parameter** on the parameter control card menu.
2. In the Edit Parameter dialog box, make the modifications as necessary.
3. When finished, click **OK**. The parameter is updated along with any calculations that use it.

> To delete a parameter, right-click it in the Data pane and select **Delete**. Any calculated fields that use the deleted parameter will become invalid.

Use a parameter in a calculation

Parameters give you a way to dynamically modify values in a calculation. Rather than manually editing the calculation (and all dependent calculations), you can use a parameter. Then when you want to change the value, you open the parameter control, change the value, and all of the calculations that use that parameter are updated.

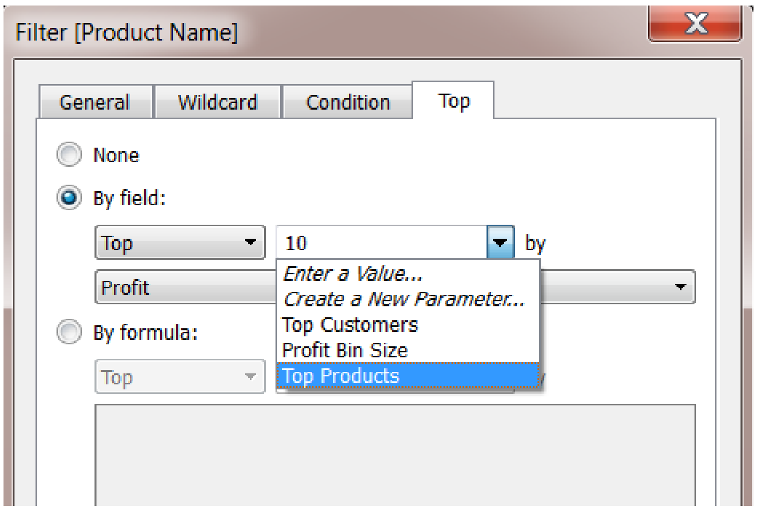
To use a parameter in a calculation, drag the parameter from the Data pane and drop it in the calculation editor, either at a new location in the formula or to replace a part of the current formula:



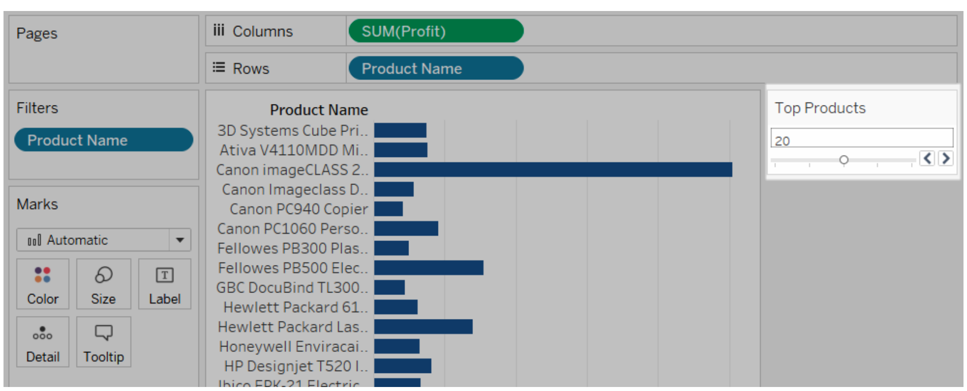
Use a parameter in a filter

Parameters give you a way to dynamically modify values in a Top N filter. Rather than manually setting the number of values you want to show in the filter, you can use a parameter. Then when you want to change the value, you open the parameter control and the filter updates. For example, when creating a filter to show the Top 10 products based on total profit, you may want to use a parameter instead of the fixed “10” value. That way, you can quickly update the filter to show the top 10, 20, or 30 products.

A list of parameters is available in the drop-down lists on the **Top** tab of the Filter dialog box. Select the parameter you want to use in the filter.



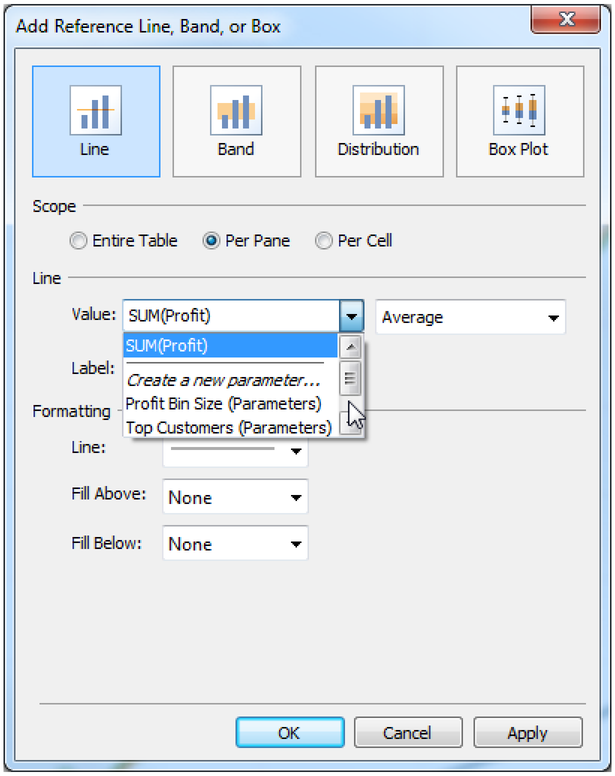
To show the parameter control, right-click the parameter in the **Data** pane and select **Show Parameter**. Use the parameter control to modify the filter to show the top 10 products, 15 products, 20 products, and so on.



Use a parameter in a reference line

Parameters give you a way to dynamically modify a reference line, band, or box. For example, instead of showing a reference line at a fixed location on the axis, you can reference a parameter. Then you can use the parameter control to move the reference line.

A list of parameters is available in the Value drop-down list in the Add Reference Line, Band, or Box dialog box. Select the parameter you want to use.

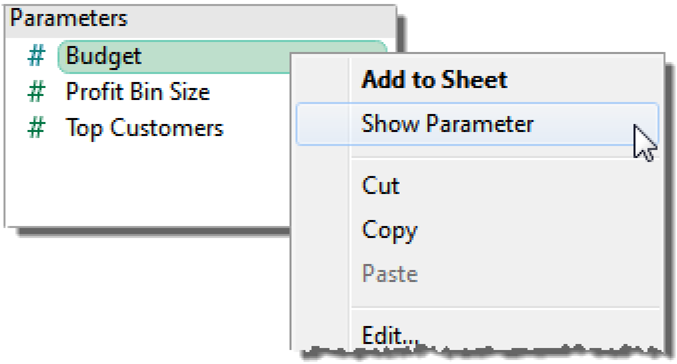


The reference line is drawn at the Current Value specified by the parameter. To open the parameter control, right-click (Control-click on a Mac) the parameter in the **Data** pane and then select **Show Parameter**. Use the parameter control to change where the reference line is drawn.

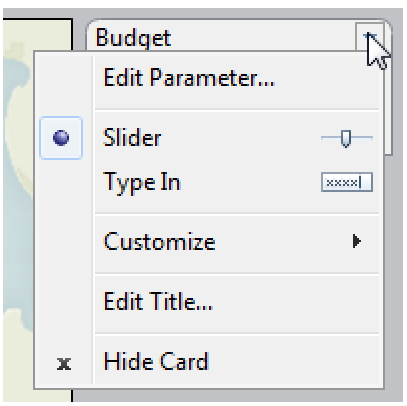
Show a parameter control in the viz

The parameter control is a worksheet card that lets you modify the parameter value. Parameter controls are very similar to filter cards in that they contain controls that modify the view. You can open parameter controls on worksheets and dashboards and they are included when you save to the web or publish to Tableau Server.

To open the parameter control, right-click (Control-click) the parameter in the **Data** pane and select **Show Parameter**.



Like other cards, parameter controls have a menu that you can open using the drop-down arrow in the upper right corner of the card. Use this menu to customize the display of the control. For example, you can show a list of values as radio buttons, a compact list, a slider, or a type in field. The options available on this menu depend on the data type of the parameter as well as whether it accepts all, a list, or a range of values.



Use Parameters to Make Views More Interactive

Parameters are useful when you want to add interactivity and flexibility to a report, or to experiment with what-if scenarios. Suppose you are unsure which fields to include in your view or which layout would work best for your viewers. You can incorporate parameters into your view to let viewers choose how they want to look at the data.

When you use parameters, you need to tie them to the view in some way:

- You can use parameters in calculations and calculated fields that are used in the view.
- You can display the parameter control in the view for users to select parameters.
- You can reference parameters in parameter actions.

Before you begin, decide which fields you want to make interactive. For example, you could allow users to view the categories within a dimension by color, or to view sales data over a period of time that they choose, and so on. The example described here sets up a table for which users can select the dimensions to display in the columns and rows.

Watch a video: To see related concepts demonstrated in Tableau, watch the free training video Parameters (5 minutes). Use your tableau.com account to sign in. For a more in-depth presentation, see Parameters | Oh, the places you'll go! (45 minutes).

Learn more: See Use Parameters to Add Multiple Views to Your Viz on the Tableau blog and multiple practical topics on how to use parameters on the Data School website.

Create the parameters

These steps use the Superstore sample to create new parameters.

1. In the Data pane, click the drop-down arrow in the upper right corner and select **Create Parameter**.
2. In the Create Parameter dialog box, complete the following steps:
    a. Name the parameter so that viewers can tell what changing it will do. This example uses **Select Column 1 Heading**.

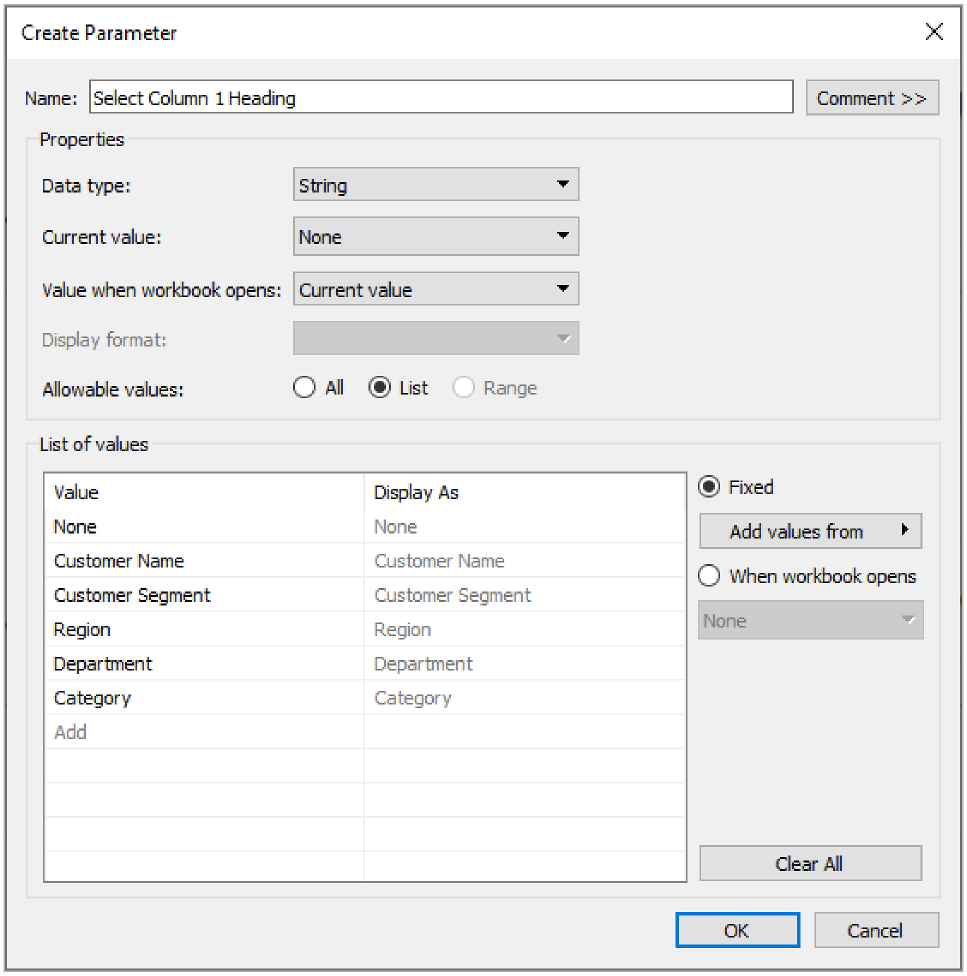
    b. For **Data type**, select **String**.

    c. For **Allowable Values**, select **List**, type **None** as the first value in the list, and then press **Enter**.

    d. Complete the list by typing the names of the additional dimension fields that you want to expose through the parameter.

    Note: This example uses the customer name, customer segment, region, department, and category fields. These are all dimensions of the same data type (string). If you wanted to include a measure such as profit in this list, one option would be to convert the measure to a string value. You would do this when you build the calculated field, using the `STR()` function. This article covers only the single data type scenario.

    The **Display As** aliases default to the field name, and for this exercise you can leave them as they are.



e. Click **OK** to return to the Calculated Field dialog box.

3. Repeat the previous step to create the following additional parameters:
   - Select Column 2 Heading
   - Select Row 1 Heading
   - Select Row 2 Heading

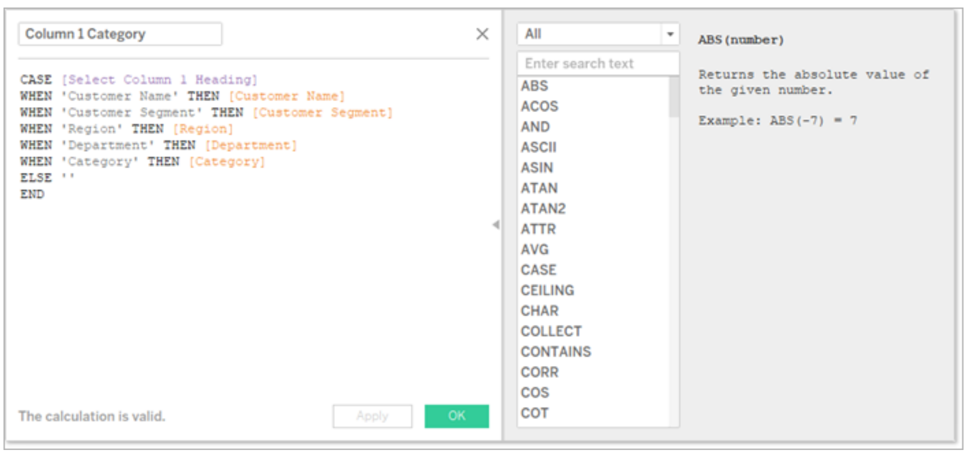
   Tip: Instead of typing each value in the list, click **Add values from > Parameters** to add them from **Select Column 1 Heading**.

Create the calculated fields

These steps use the Superstore sample to build the calculated fields that will take advantage of your parameters.

1. In the Data pane, click the drop-down arrow in the upper right corner and select **Create Calculated Field**.
2. In the Calculated Field dialog box, for **Name**, type **Column 1 Category**.
3. In the Calculated Field dialog box, for **Formula**, build the following calculation:

```
CASE [Select Column 1 Heading]
WHEN 'Customer Name' THEN [Customer Name]
WHEN 'Customer Segment' THEN [Customer Segment]
WHEN 'Region' THEN [Region]
WHEN 'Department' THEN [Department]
WHEN 'Category' THEN [Category]
ELSE ''
END
```



Confirm that the status message indicates that the formula is valid, and then click **OK**.

Note: ELSE accounts for the **None** value that you included in the parameter, and it returns an empty string.

4. Create three more calculated fields, one for each of the additional parameters you created:

Parameter name	Calculated field name
Select Column 2 Heading	Column 2 Category

Select Row 1 Heading	Row 1 Category
Select Row 2 Heading	Row 2 Category

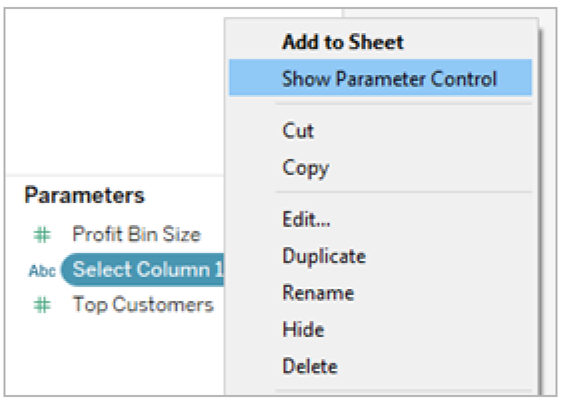
The basic formula for each calculated field is the same as in the previous step, except that you reference a different parameter in each `CASEstatement`.

Give viewers a way to interact with views

Next, expose the parameter control so users can select the categories they want to display.

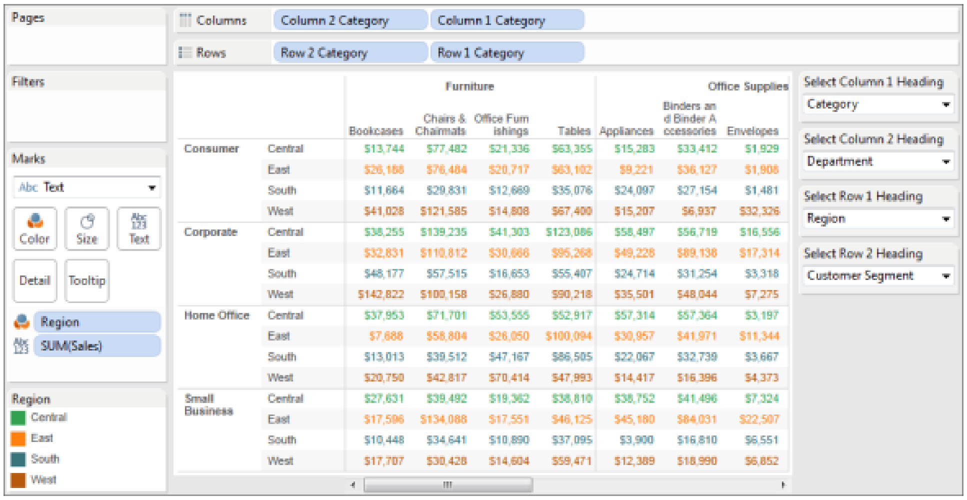
1. For each parameter you created, do the following:

   Under Parameters, right-click the parameter and select **Show Parameter Control**.

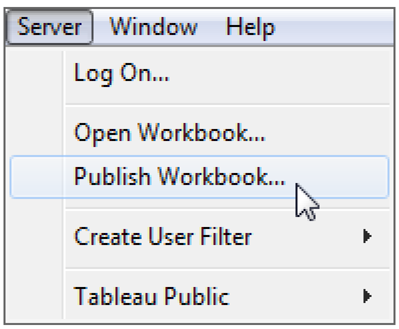


2. From the Data pane, drag the calculated fields you created to the Columns and Rows shelves.
3. From the Data pane, drag a measure to the view. In this example, **Sales** is placed on **Label** on the Marks card.
4. Test your parameters by selecting fields in the parameter controls.

   Tips:
   - Sort the dynamic dimension fields alphabetically.
   - Hide field labels for rows and columns.



5. Reset all parameters to **None** and publish the workbook to Tableau Server.



Viewers can set up their own reports, save their parameter settings, and share views with others.

Additional information

For information about building views dynamically, see Swap Measures Using Parameters.

For information about the different areas in the Tableau interface in which you can create and incorporate parameters, see **Create Parameters** on page 1218 and its related topics in the Tableau Help.

> **Disclaimer**: This topic includes information about parameters from a third-party blog, The Information Lab: Data School. Please note that while we make every effort to keep references to third-party content accurate, the information we provide here might change without notice as content on their website changes.

Example: Add a Parameter to a Map View

This example uses the World Indicators sample data source to demonstrate the following:

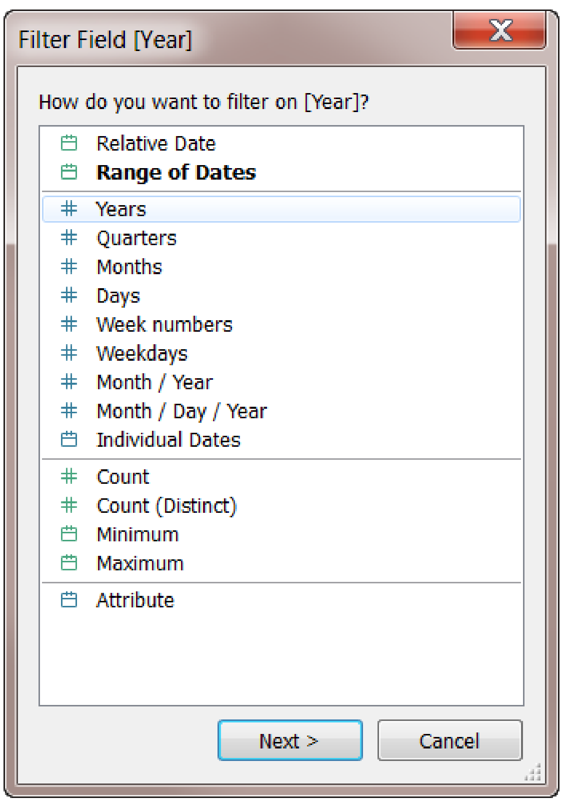
- How to build a map view that shows the birth rate for each country in the world.
- How to create a calculated field that distinguishes countries/regions with a low birth rate from those with a high birth rate.
- How to create and display a parameter so that users can set the threshold for low vs. high birth rate.

Build a map view

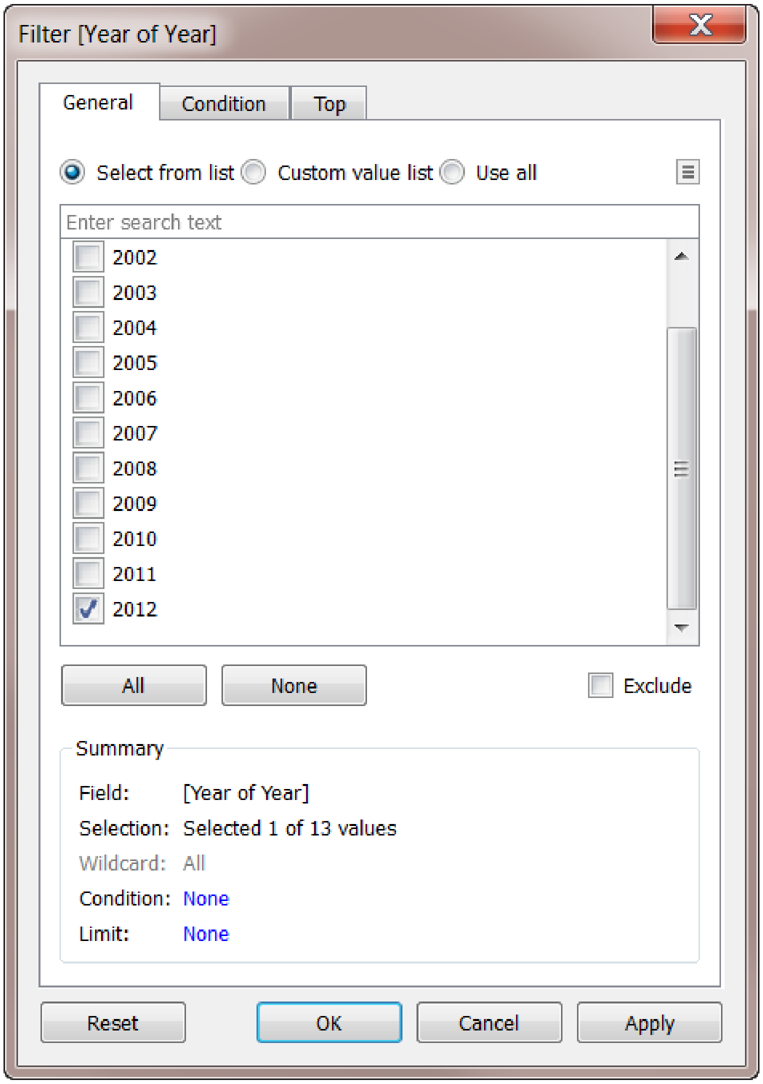
1. In the **Data** pane, double-click **Latitude** and then **Longitude**.

   Tableau puts **Longitude** on **Columns**, **Latitude** on **Rows**, and displays a map of the world.

2. Drag the **Year** dimension to **Filters**.
3. In the Filter Field [Year] dialog box, choose **Years** and then click **Next**:

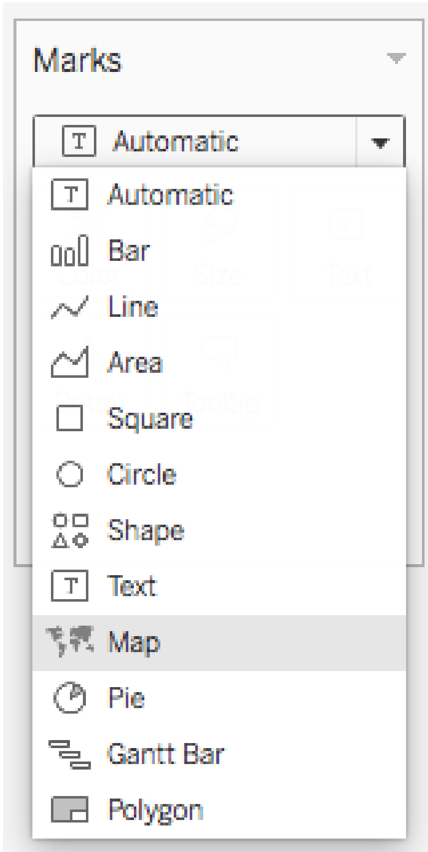


4. In the Filter [Year of Year] dialog box, select **2012** and then click **OK**:



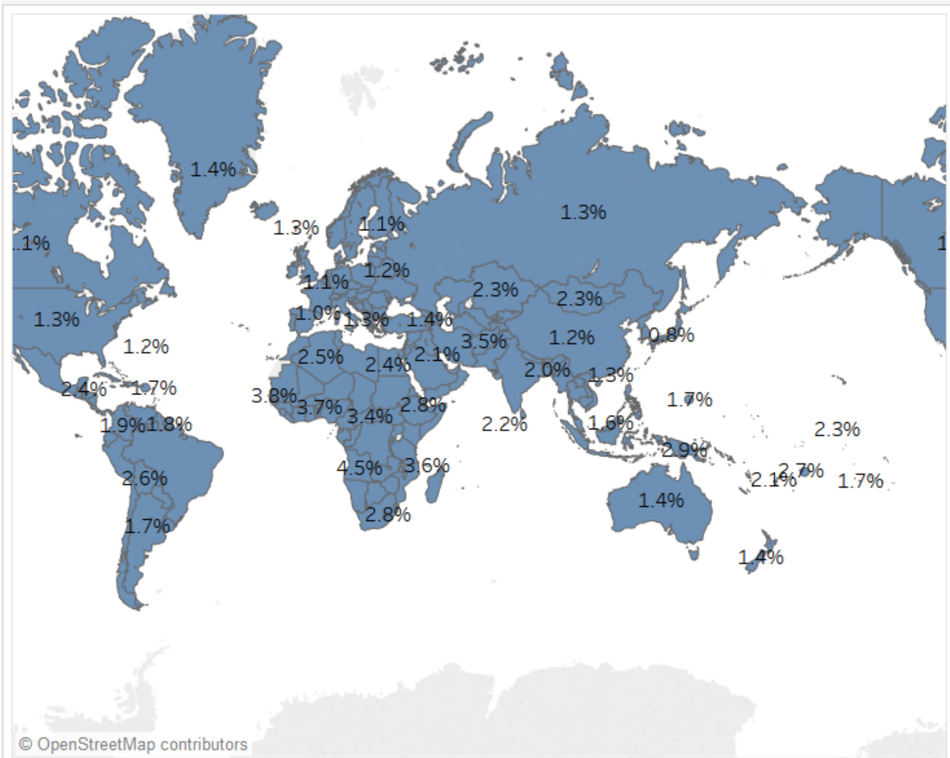
5. Drag the **Country** dimension to **Detail**.

6. Set the Marks type to Map:



7. Drag the **Birth Rate** measure to **Label**.

You now have a map that shows birth rates for countries/regions around the world:



You can zoom the map or hover to see a tooltip for any country.

Create a calculated field to set a threshold

Next, you'll distinguish a low birth rate from a high birth rate.

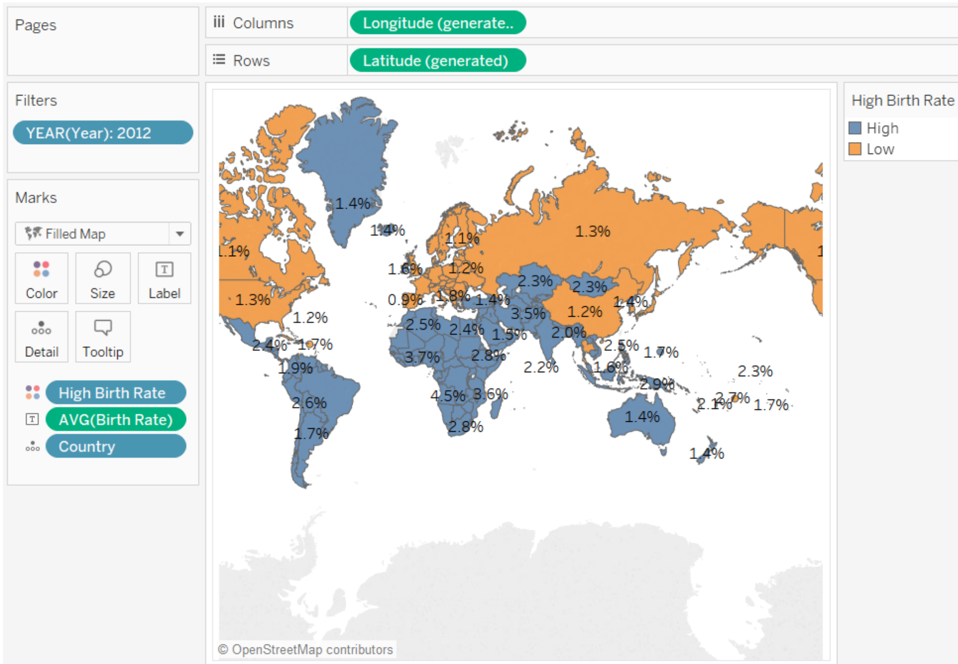
1. From the top menu, select **Analysis** > **Create Calculated Field** .
2. Name the field `High Birth Rate` and type or paste this calculation in the formula field:

   ```
   IF ([Birth Rate]) >= 0.014 THEN "High" ELSE "Low" END
   ```

   The value 0.014 is equivalent to 1.4%. The range of actual values that we can see on the map range from below 1% up to nearly 5%.

   When you click **OK** to apply and save this calculation, Tableau categorizes it as a dimension.
3. Drag **High Birth Rate** to **Color**.

   The map now shows low birth rate countries/regions in one color and high birth rate countries/regions in another:



But the definition of high birth rate as anything equal to or greater than 1.4% is arbitrary—that value was chosen because it divided the world's countries/regions about evenly.

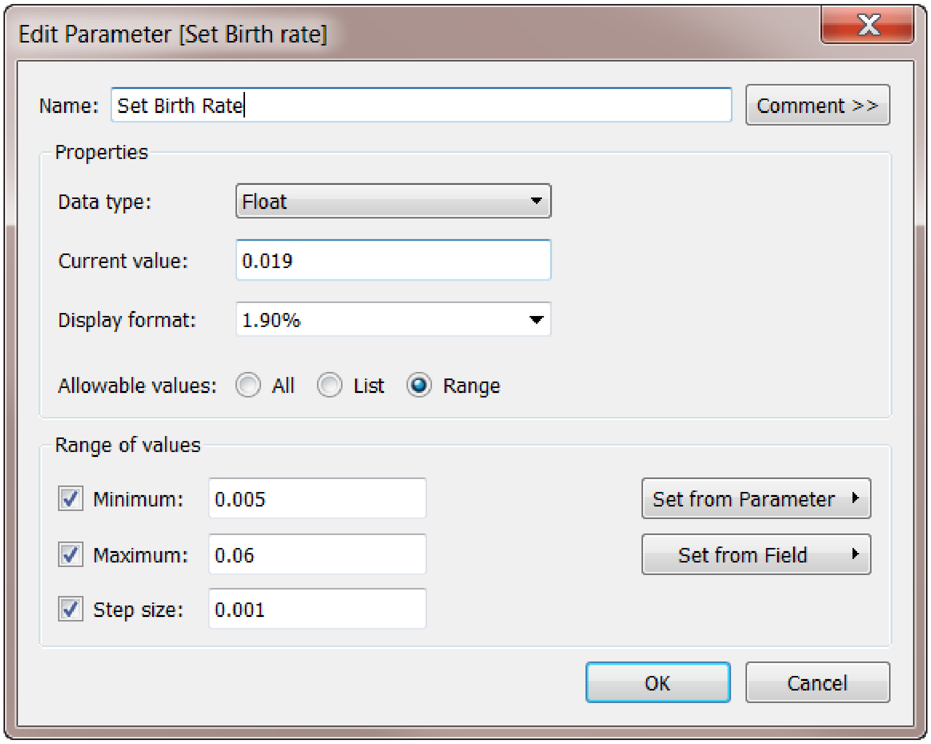
Instead, you can let users define that threshold, or give them a control that they can use to see how changing the threshold changes the map. To do this, you create a parameter.

Create a parameter

1. Right-click (control-click on a Mac) in the **Data** pane and select **Create** > **Parameter**.

   > **Note**: If you clicked on a field on the Data pane, some fields might be filled. You can just change the fields as needed.

2. In the Create Parameter dialog box, name the new parameter `Set Birth Rate` and configure it as shown:



For information on the fields in the Create Parameter dialog box, see **Create Parameters** on page 1218.

Because the **Data Type** is **Float**, the parameter control, when you display it in the next procedure, will be in the form of a slider. This is because floating point values are continuous—there are an infinite number of possible values.

The **Current value** sets the default for the parameter: 0.019 is 1.9%. The **Range of values** section sets the minimum and the maximum values and the step size—that is, the least amount by which the value can change.

3. Click **OK** .

Create and display the parameter control

Now you must connect the parameter to the **High Birth Rate** field.

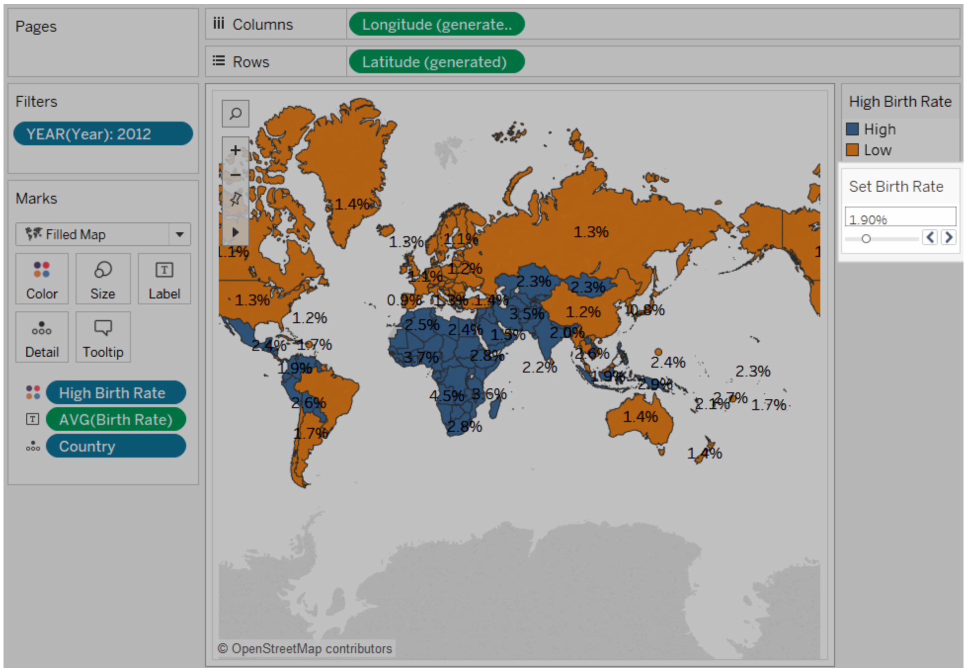
1. Right-click **High Birth Rate** in the **Data** pane and select **Edit**.
2. Replace the hard-coded 0.014 value in the field definition with the parameter name:

```
IF ([Birth Rate]) >= [Set Birth Rate] THEN "High" ELSE "Low"
END
```

Then click OK.

3. Right-click the **Set Birth Rate** parameter in the **Data** pane and select **Show Parameter Control**.

By default, the parameter control is shown on the right. Now you and users of your view can raise or lower this value incrementally to see how changing the definition of "high birth rate" affects the map.



Example: Swap Measures Using Parameters

You may want to create a view that shows how the values of two measures compare against each other. But what if you also wanted to be able to choose which measures were being compared—or better yet, add a control to the view that would let any user select the measures to be compared. You can create such a view using parameters and calculated fields. The calculated fields replace the measures in the view and can be set interactively by the user with parameter controls or parameter actions.

General steps

1. Create the parameters.

2. Create calculated fields to change the measures in the view.

3. Set up the view.

This article includes an additional section that you can use as an alternative to step 2:

- Create calculated fields to change measures and specify aggregations

The following sections break these procedures down into specific instructions.

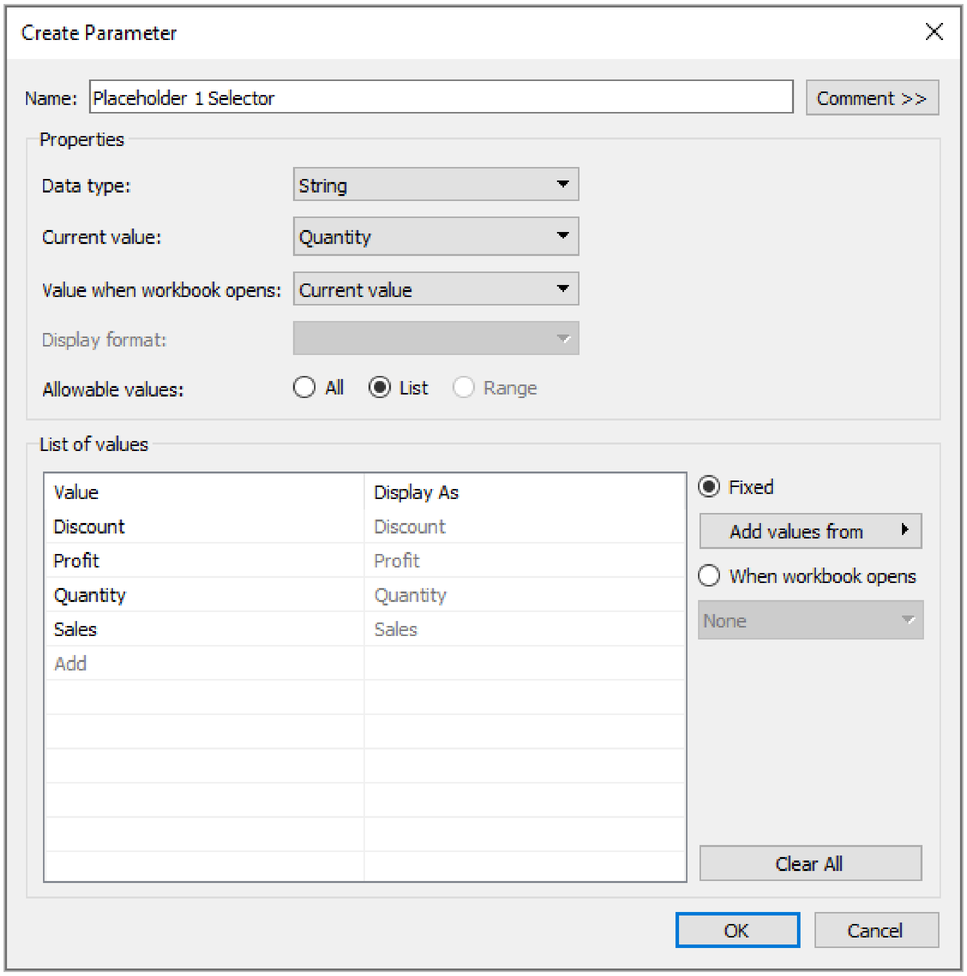
The scenario uses the **Sample - Superstore** data source provided with Tableau Desktop.

Create the parameters

Follow these steps.

1. In the Data pane, click the drop-down arrow in the upper right corner and select **Create Parameter**.
2. In the Create Parameter dialog box, do the following:
   a. Name the parameter **Placeholder 1 Selector**.
   b. Set the **Data type** to **String**.
   c. Skip down to the **Allowable values** field, and choose **List**.
   d. Type individual measure names in the **List of values** area: **Discount**, **Profit**, **Quantity**, and **Sales**.

      The Create Parameter dialog box should now look like this:



e. Click **OK** to close the Edit Parameter dialog box.

3. Create a second parameter, **Placeholder 2 Selector**, with the exact same configuration.

   There are different ways you can do this. The easiest way is to click **Placeholder 1 Selector** in the **Data** pane, choose **Duplicate**, and then change the name of the duplicated parameter to **Placeholder 2 Selector**.

Create calculated fields to change the measures in the view

Follow these steps.

1. Choose **Analysis** > **Create Calculated Field** to open the calculation editor. Name the calculation **Placeholder 1** and type or paste the following in the formula area:

```
CASE [Placeholder 1 Selector]
```

```
WHEN "Discount" THEN [Discount]

WHEN "Profit" THEN [Profit]

WHEN "Quantity" THEN [Quantity]

WHEN "Sales" THEN [Sales]

END
```

> **Note**: The calculated field must reference the **Value** entry for a given row, and not the **Display As** value.

2. Click **OK** to exit the calculation editor.
3. Create a second calculated field, **Placeholder 2**, with the same definition. Again, the easiest way to do this is to click **Placeholder 1** in the **Data** pane, choose **Duplicate**, and then change the name of the duplicated field to **Placeholder 2** .

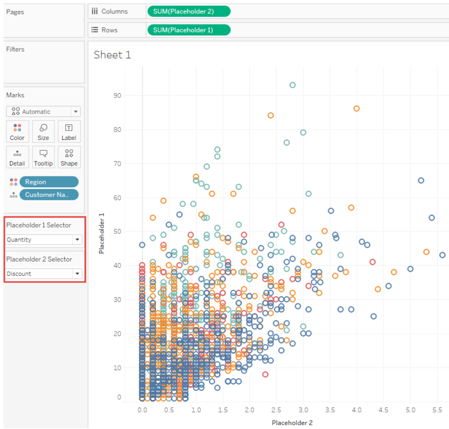
Set up the view

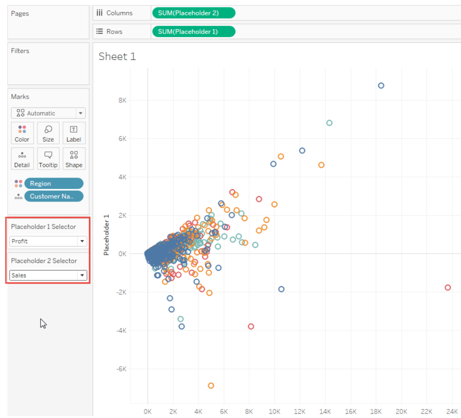
Follow these steps.

1. Drag **Placeholder 2** to **Columns** and **Placeholder 1** to **Rows**.

   Because you dragged measures to both shelves, the default view is a scatter plot. For more on why Tableau does this, see **Example: Scatter Plots, Aggregation, and Granularity** on page 351.
2. Drag **Customer Name** to **Detail** and **Region** to **Color**.
3. In the Parameters area of the Data pane, click each **Placeholder 1 Selector** and choose **Show Parameter Control**. Then do the same for **Placeholder 2 Selector**.
4. Tableau displays the parameter controls beyond the right side of the view by default. Drag them over to the left side to make them easier for your users to see.

Your view is now complete. The parameter controls let users select the measures to be used on the X and Y axes. For example, the view below on the left shows **Quantity** vs. **Discount**, while in the view the right the parameter controls have been used to show **Profit** vs. **Sales**.





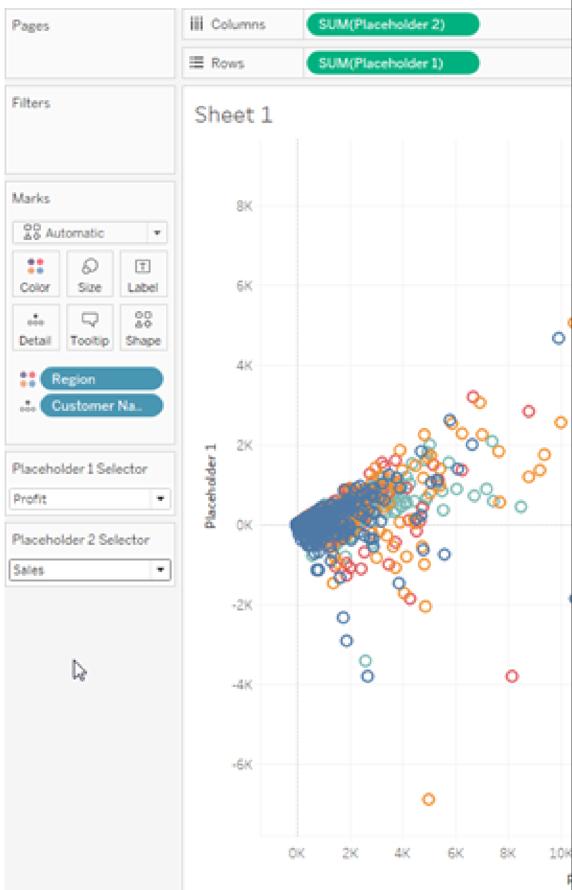
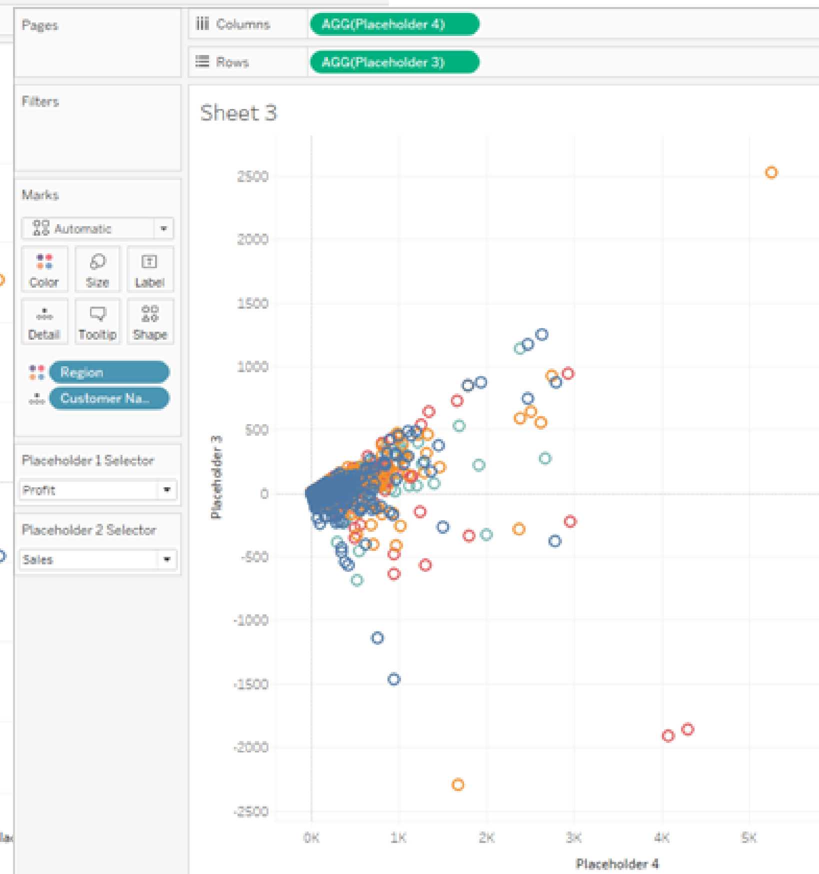
Create calculated fields to change measures and specify aggregations

As an alternative to the **Create calculated fields to change the measures in the view** section, above, consider creating calculated fields that specify aggregations for individual measures. As written above, the calculated fields do not specify aggregations. Notice in the image above that Tableau automatically assigns an aggregation (SUM) to the **Placeholder 1** and **Placeholder 2** fields. But you know your data, and you may want to dictate which aggregation Tableau uses for your measures. So instead of the calculated field definition above, consider a definition like the following:

```
CASE [Placeholder 1 Selector]

WHEN "Discount" THEN SUM([Discount])

WHEN "Profit" THEN AVG([Profit])

WHEN "Quantity" THEN SUM([Quantity])

WHEN "Sales" THEN AVG([Sales])

END
```

It's up to you to decide whether to explicitly aggregate measures in your field definitions. The only thing you cannot do is mix-and-match: that is, you cannot define aggregations for some meaures, but not for others.

Here's how the **Profit** vs. **Sales** scatter plot changes when you specify AVG as the aggregation for these fields, as opposed to not specifying an aggregation and letting Tableau default to SUM:

Profit vs. **Sales** with default aggregation	**Profit** vs. **Sales** with explicit aggregation
	

Similar, but different.

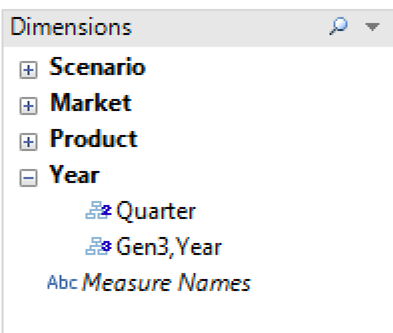
Dates and Times

How you work with dates in Tableau depends on whether you are using a relational or cube (multidimensional) data source. This section discusses the differences.

Dates in Cube (Multidimensional) Data Sources

> In Tableau Desktop, cube (multidimensional) data sources are supported only in Windows.

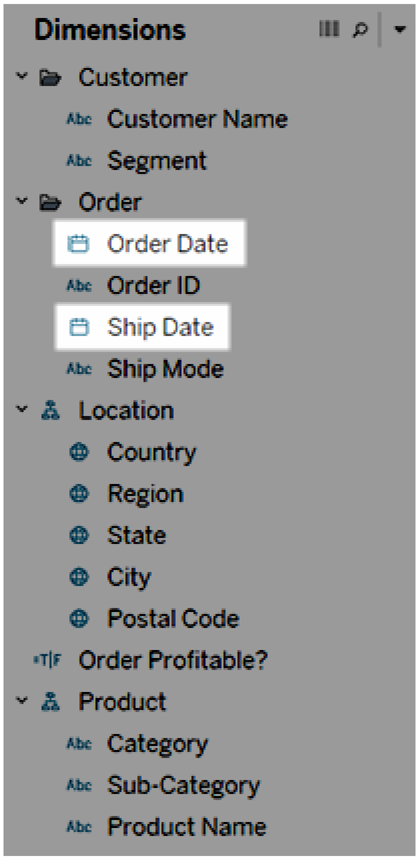
For cube data sources, dates dimensions are usually organized into hierarchies that contain levels such as year, quarter, and month. In addition, some multidimensional data sources have time intelligence enabled, which makes it possible to look at data levels different ways, such as Months by Year, Months by Quarter, Weekends, etc. These levels are represented as attributes of the hierarchy. Hierarchies and attributes are defined when the cube is created and you cannot modify them in Tableau. For example, the **Year** dimension from an Oracle Essbase data source is shown below.



When you place a multidimensional date on a shelf, the field is treated like any other dimension. For example, you can drill down, drill up, and so on.

Dates in Relational Data Sources

For relational data sources, dates and times are automatically placed in the Dimensions area of the **Data** pane and are identified by the date or date-time icon. For example, the Order Date and Ship Date dimensions from an Excel data source are shown below.



When you place a relational date on a shelf, the field name is automatically modified to reflect the default date level. Tableau defines the default date level to be the level at which there are multiple instances. For example, if the date field includes multiple years, the default level is year. However, if the date field contains data for just one year but includes multiple months, then the default level is month.

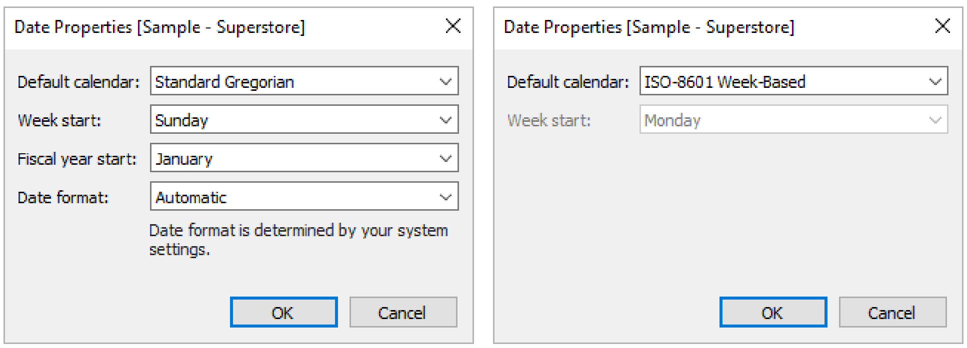
If you don't want Tableau to automatically select a date level and would rather have a date dimension be a continuous field, you can right-click (control-click on Mac) the field in the **Data** pane and select **Convert to Continuous**. The dimension then turns green in the **Data** pane; now when you use that dimension in a view, it will be continuous. You can easily revert back by selecting **Convert to Discrete** from the field's context menu in the **Data** pane. You can also convert a field in the view to continuous while it is on a shelf by selecting **Continuous** on its context menu (which you can see when you right-click (control-click on Mac) the field). The field on the shelf turns green but the field in the **Data** pane is still discrete.

Date Properties for a Data Source

You can set date properties for a data source. To do so, right-click (control-click on Mac) a data source in the **Data** pane and choose Date Properties:

The options in the Date Properties dialog box are:

- **Default calendar** - Specifies which calendar system to use: **Standard Gregorian** or **ISO-8601 Week-Based**. You can freely switch between the calendar types with the dates in your view, but setting the default calendar will determine how dates are used upon initially dragging them into your view. See **Date Functions** on page 2170



- **Week start** - Specifies which day is considered the first day of the week. You can override the **Week start** value you set at the data source level by including a `start_of_week` value with certain date functions (`DATEDIFF`, `DATENAME`, `DATEPART`, and `DATETRUNC`). See **Date Functions** on page 2170.

  The initial value for this setting is in some cases determined by your data source. If the data source does not initialize the start-of-week day, then Tableau consults the system's locale settings.The day which is considered the first day of the week varies from region to region. For example, Sunday is the first day of the week in the US, while Monday is the first day in the EU.

  This setting is disabled for multidimensional data sources because this information is defined when cube designer creates the date/time dimension.

  After a data source is created, the **Week start** value does not automatically change if for any reason the default start-of-week setting changes in the host operating system (for example, if the data source is uploaded to Tableau Server in a different country). Calendar controls reflect the workbook locale rather than the data source's **Week start** setting. The exception is that a week trunc start of week that shows the week number in it will use the data source's **Week start** setting in order to provide a consistent week number value in the calendar.

- **Fiscal year start** - Specifies which month is considered the first month of the fiscal year. To specify whether a date dimension uses the standard calendar (Jan. 1 - Dec. 31) or the fiscal calendar, right-click (control-click on Mac) the field in the Data pane and select **Default Properties** > **Calendar Type** and then select **Standard Calendar** or **Fiscal Calendar**. For any date field in the view, you can then specify whether to use the standard calendar or the fiscal calendar. For details, see **Fiscal Dates** on page 1258.
- **Date format** - Specifies the default format for data dimensions, as displayed, for example, in tooltips.To override the default date format for a date dimension, right-click (control-click on Mac) the field in the Data pane and select **Default Properties** > **Date Format** and then select one of the available formats.

Day of the week sorting

If you are working in a language for which Tableau does not provide a local version, set your workbook locale to assure that Tableau can sort the days of the week in the correct chronological order. Otherwise, Tableau will sort the names of the days alphabetically. For information on setting the locale, see **Language and Locale** on page 276.

If none of the supported locales is appropriate, you can sort the days of the week manually. See **Sort Data in a Visualization** on page 1447.

Supported date formats

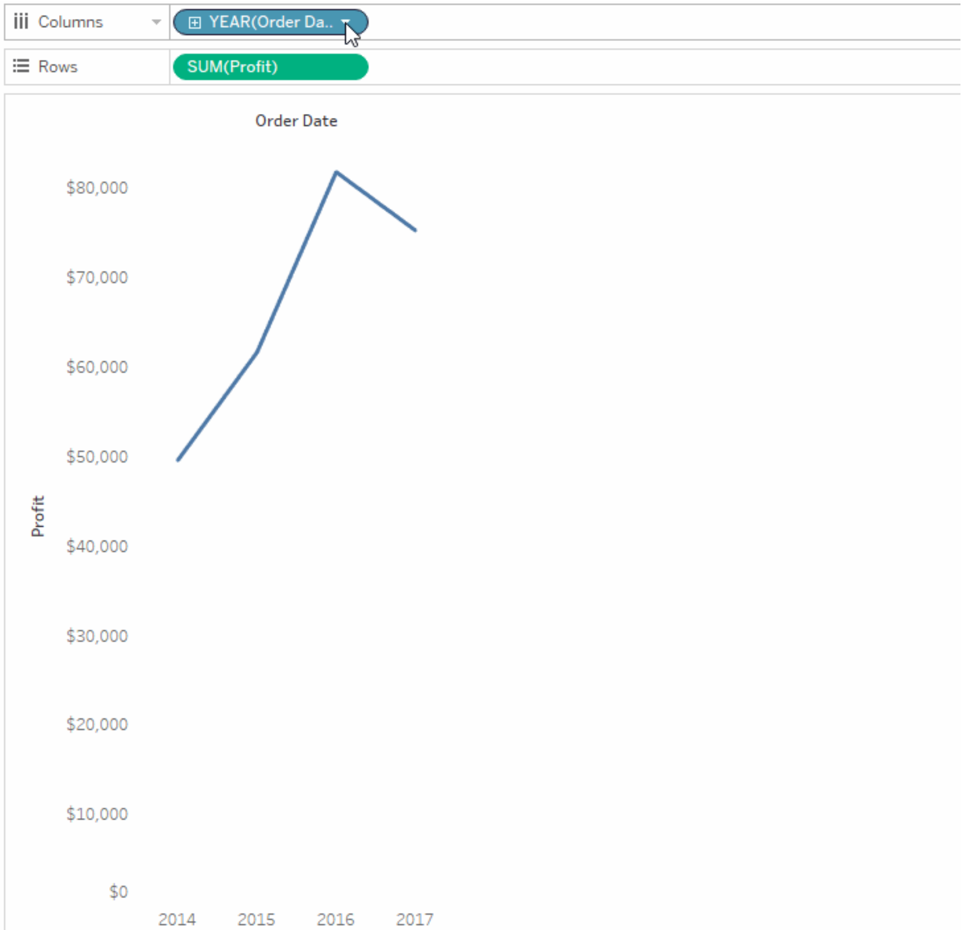
When working with dates, Tableau retrieves date formats automatically from the data source.

For a list of supported date formats, see the table of supported date format symbols in **Custom Date Formats** on page 1261.

Change Date Levels

To change the date level of a field, drag it to Rows or Columns (or another shelf), click its context menu, and then select a level.

Each level has a preview to indicate the effect of the setting.



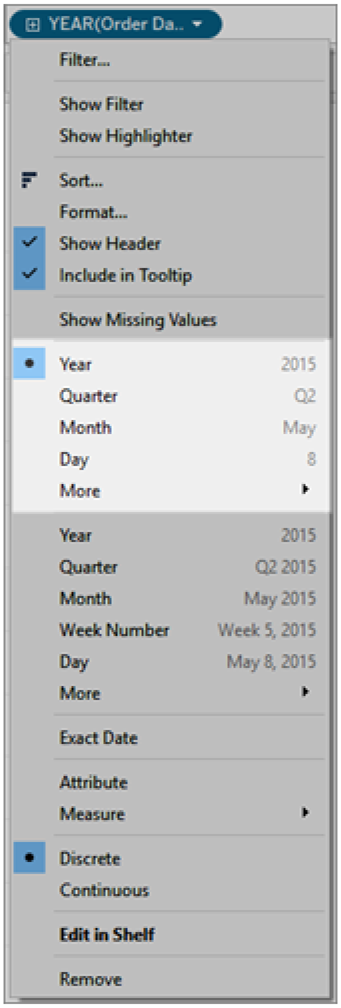
Learn more: For related concepts about dates, see Dates in Tableau make me want to tear my hair out! in the VizPainter blog by Tableau Zen Master, Josh Milligan. For an in-depth demonstration of how date parts and date values work, see the free video presentation Greatest of All Time Analytics: Conquering Date Calculations (58 minutes).

In Tableau Desktop, cube (multidimensional) data sources are supported only in Windows. For cube (multidimensional) dates, the levels available in the context menu are given by the levels defined in the date hierarchy.

Date level settings

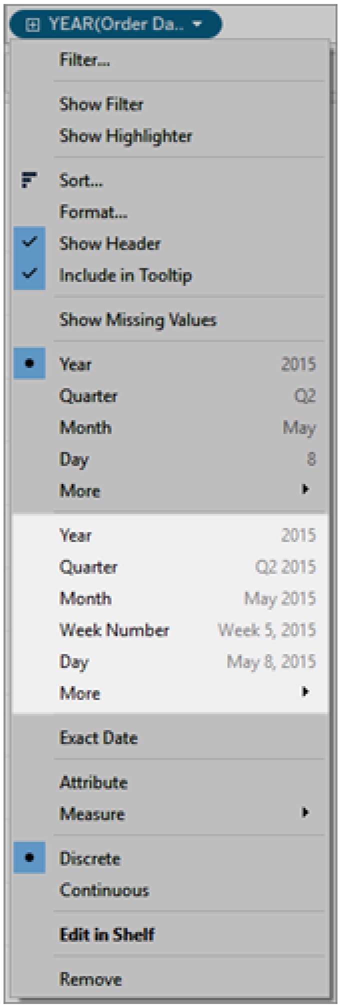
The date level settings are organized by sections for date parts and date values.

Date parts give you data aggregated to the level of a specific part of a date, such as the month of May for any year, or the 8th day of any month.



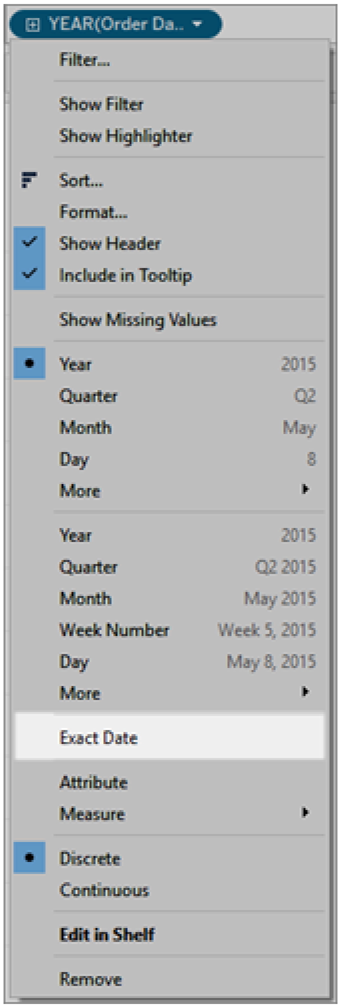
Date Parts

Date values give you the actual date, truncated to a specific level, such as May 2015 or May 8, 2015.



Date Values (also known as truncated dates or Date Truncs)

To get the most granular level of detail, which is also known as row-level or record-level data, you can select **Exact Date**.



Exact Date

When you select a particular level, Tableau performs a computation on the date field. Suppose a particular row in your data source has a date entry of 01/23/16. The year is 2016, the quarter is 1 because January falls in the first quarter, and the week number is 4 because January 23rd falls in the fourth week.

How the date level is computed depends on how Date Properties are configured for the data source. For more information, see **Date Properties for a Data Source** on page 1247.

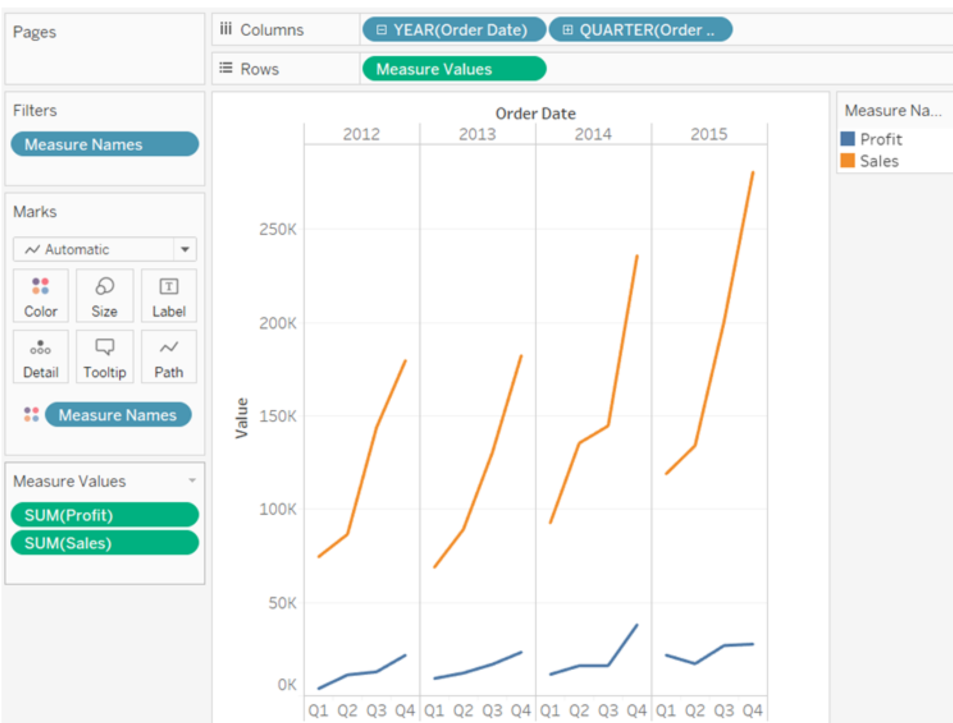
> **Note:** Some date levels might not make sense for your relational data source. For example, if the date format does not include time information such as hour, minute, or second, then selecting one of these options will not add any data to your view.

Use multiple date levels in the same viz

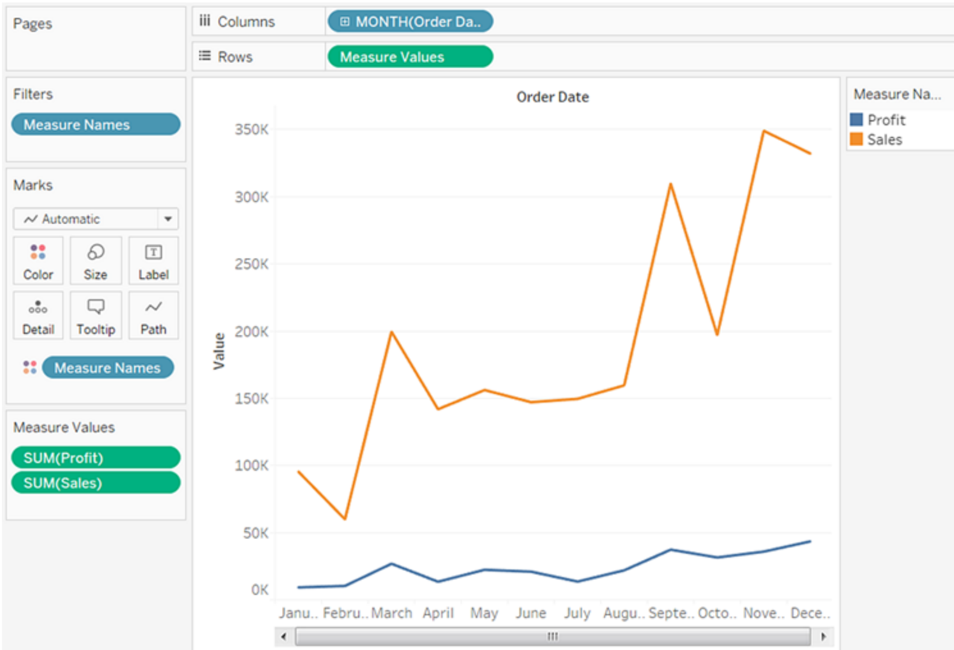
You can work with dates at varying levels of detail simultaneously. You do this by clicking the ⊞ control on a date field in the view. This is known as drilling into a date. For example, if you drill into a discrete (blue) date field **YEAR(Order Date)**, Tableau adds a second field to the right of the initial date, **QUARTER(Order Date)**, as in the image below. You can continue drilling down until you reach the finest available granularity for the date field.

You can also drag date fields to the **Rows** or **Columns** shelf multiple times in order to nest them and to drill down into them at varying levels of detail.

For example, the view shown below drills down from the year level to display the quarter level as well.



By removing one of the date fields from the Columns shelf, and then selecting **Month** from the remaining date field's context menu, you can display the data for each month across all years.



To display even finer granularity, select Month from the lower, continuous section of the field's context menu.

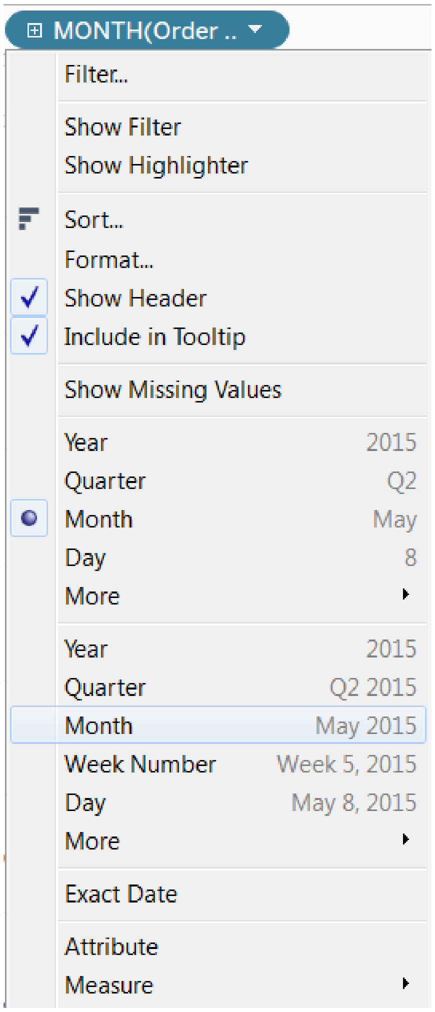
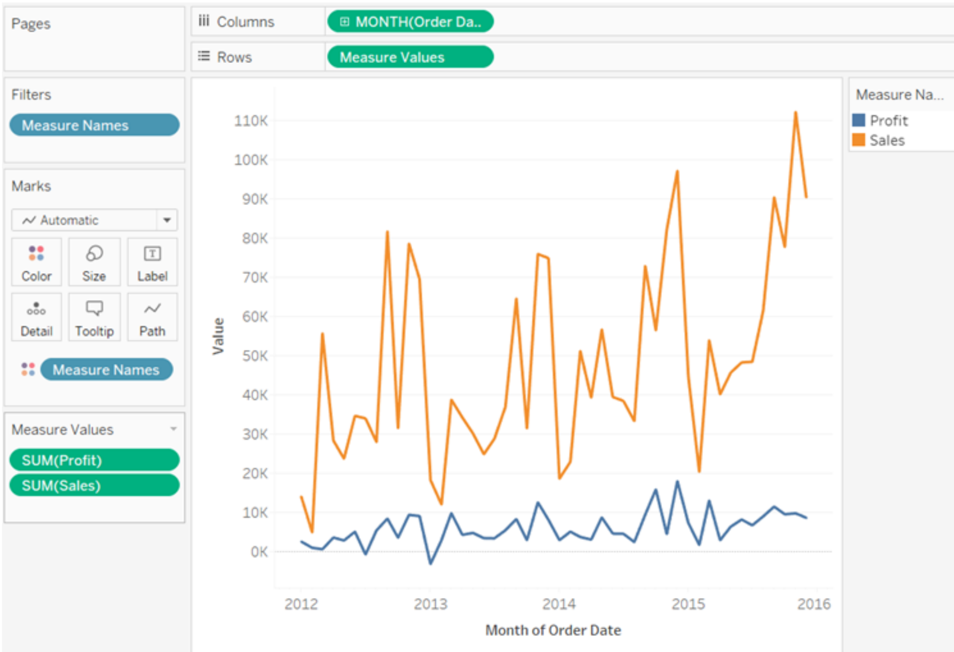


Tableau then displays the dates across the range of available years, at a month-by-month level of granularity.



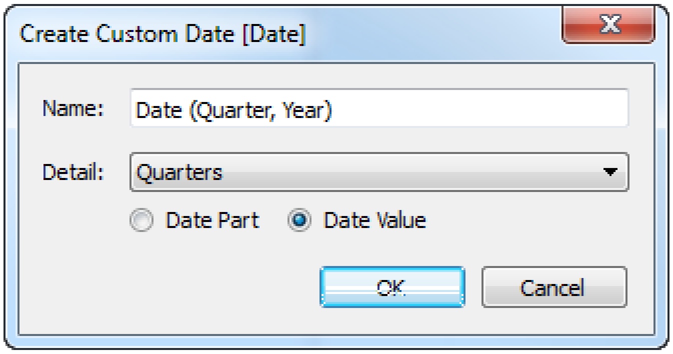
Custom Dates

You might want to create a custom date if you always use a date at a specific level, or if you want to create calculations that rely on a binned or truncated date.

You can create a custom date by working in the **Data** pane or by using the `DATEPART` and `DATETRUNC` functions in a calculated field.

The custom date you create becomes a new field in the **Data** pane.

1. Right-click (Control-click on Mac) a date field in the **Data** pane and select **Create** > **Create Custom Date**.
2. In the Create Custom Date dialog box, type a name for the custom date, such as Date (Quarter, Year).
3. From the **Detail** list, select the level you want to show the date at.
4. Specify whether you want the custom date to be discrete (select **Date Parts**) or continuous (select **Date Value)**.



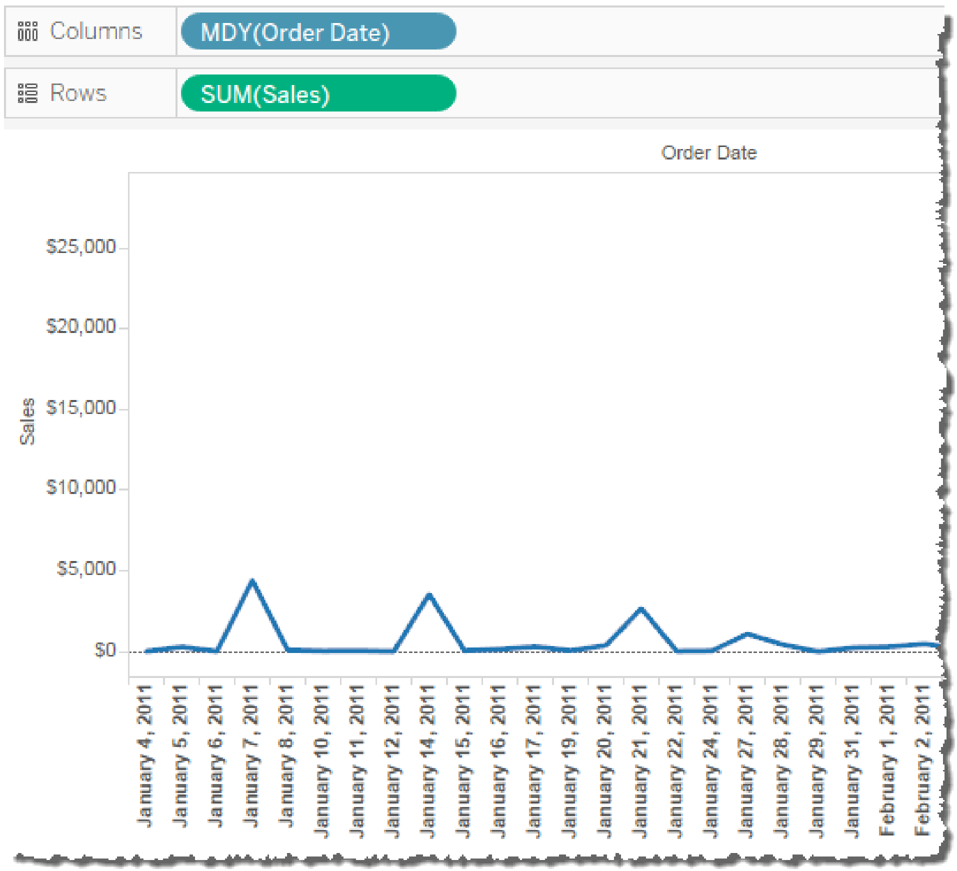
5. When you are finished, click **OK**.

Example - format column headers as Month, Day, Year

To format column headers as Month, Day, Year (for example, January 1, 2011), follow the steps below. This example uses the Sample - Superstore data source provided with Tableau Desktop.

1. Drag **[Order Date]** to **Columns**.
2. Drag **Sales** to **Rows**.
3. Right-click (Control-click on a mac) **[Order Date]** on the **Columns** shelf and choose **More** > **Custom**.
4. In the Custom Date dialog box, choose **Month / Day /Year** from the **Detail** drop-down list.

The view now has the column headers in the correct format:



Fiscal Dates

In some situations, a date field needs to be expressed in terms of an organization's fiscal year. Calendar years run from January 1st until December 31st, but an organization's fiscal year might start in a different month. For example, a company's fiscal year might run from June 1st through May 31st. In such cases, it's appropriate to display some date values in a view using fiscal equivalents (Fiscal Year, Fiscal Quarter, and Fiscal Week Number) rather than their calendar equivalents.

To set the fiscal year start month for a data source, follow these steps:

1. Right-click (Control-click on Mac) the data source in the **Data** pane to open the Date Properties dialog box.
2. Set the **Fiscal year start** field to the appropriate month.

For each date dimension, you set the fiscal year start-month separately. In the **Data** pane, right-click a date dimension (Ctrl-click on Mac), and select **Default Properties** > **Fiscal Year Start**.

Whether a given level of a date dimension is affected by the use of the Fiscal Calendar depends on the specific case.

Date Level	When Converted to Fiscal
Year	Reflects the fiscal year. For example, if **Fiscal year start** is set to April, the year for the date June 1, 2004 would be shown as FY 2005.
Quarter	The Quarter reflects the fiscal quarter. For example, if **Fiscal year start** is set to April, the quarter for the date June 1, 2004 would be Q1.
Month	No change in behavior. The calendar month is the same as the fiscal month.
Day	No change in behavior. The calendar day is the same as the fiscal day.
Hour	No change in behavior. The calendar hour is the same as the fiscal hour.
Minute	No change in behavior. The calendar minute is the same as the fiscal minute.
Second	No change in behavior. The calendar second is the same as the fiscal second.
Week Number	The Week Number reflects the fiscal week number. For example, if **Fiscal year start** is set to April, the week number for the date April 1, 2004 would be 1.
Weekday	No change in behavior. The calendar weekday is the same as the fiscal week-day.
MM/YYYY	No change in behavior. This date format always displays calendar dates, even when a fiscal year has been assigned.
M/D/Y	This date format always displays Calendar dates, even when a fiscal year has been assigned.

The only date levels that explicitly indicate that the fiscal calendar is in use are the Year and Quarter level. Specifically, fiscal years and quarters are shown with the FY prefix. This is not true for fiscal months or week numbers, however, which are not shown with any special fiscal markings.

Fiscal year designations for any given date dimension are applied to all instances of the field in the Tableau workbook. Fiscal dates can only be applied to dimensions in a relational data source.

Fiscal year formatting is applied to all date formats that include a year, or a year and a quarter. In particular, if you apply a custom date format, and only use the “y” and “q” placeholders, then `FY` will be prepended to each year.

Date Levels

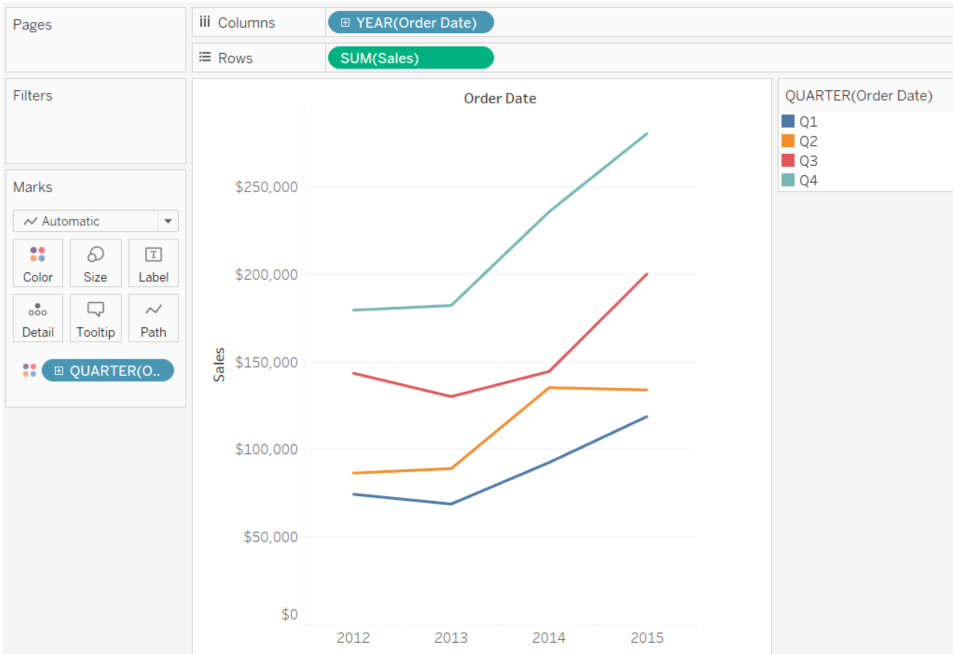
When a date dimension is using a fiscal calendar, only the following date parts and truncations present in the view will reflect the fiscal calendar:

Date part or truncation	**When using a fiscal calendar**
Year date part	**Fiscal year start** is set to April, the year for the date June 1, 2004 would be shown as FY 2005.
Quarter date part	**Fiscal year start** is set to April, the quarter for the date June 1, 2004 would be Q1.
Week Number date part	**Fiscal year start** is set to April, the week number for the date April 1, 2004 would be 1.
Year date truncation	**Fiscal year start** is set to May, the date June 1, 2004 would become May 1, 2004.
Quarter date truncation	**Fiscal year start** is set to July, the date June 1, 2004 would become April 1, 2004.

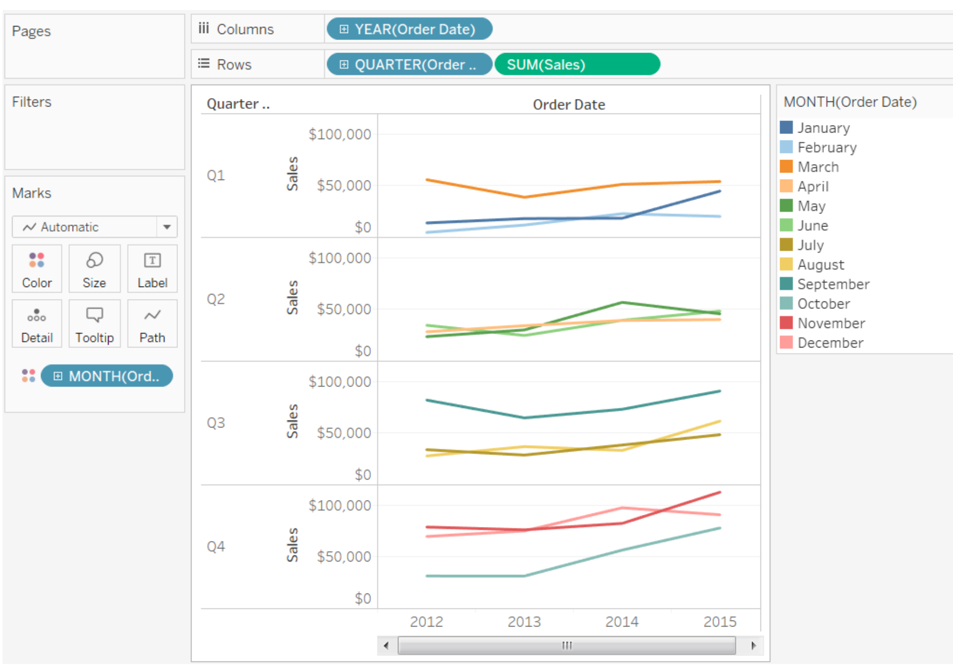
Perfect Pivoting with Dates

You can perfect pivot dates by placing different date levels on different worksheet shelves simultaneously. Place the date field on a variety of shelves and then select the desired date level from the fields’ context menus.

For example, the following line chart displays years as column headers and then color-encodes the marks by quarter.



You can separate the marks by month and by quarter as shown below.



Custom Date Formats

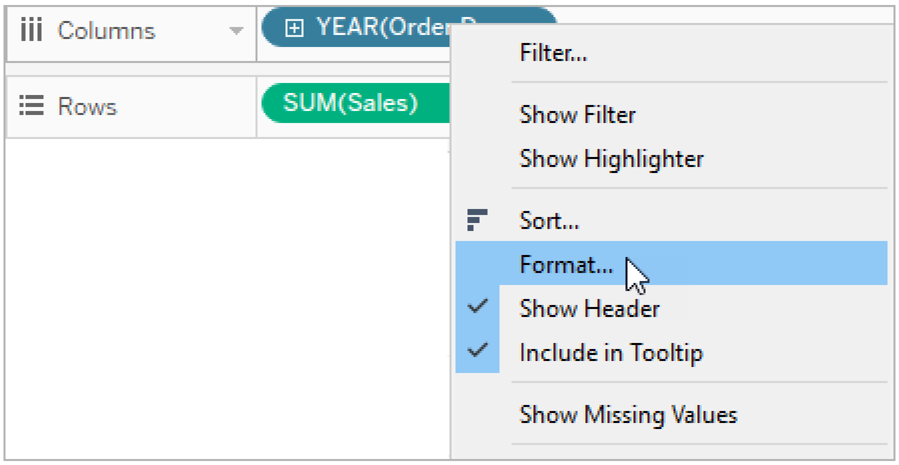
This article discusses using the custom date format field to format dates in a view. For an overview of how Tableau works with dates, see **Dates and Times** on page 1245, or **Change**

Date Levels on page 1248. For setting date properties for a **data source**, see **Date Properties for a Data Source** on page 1247.

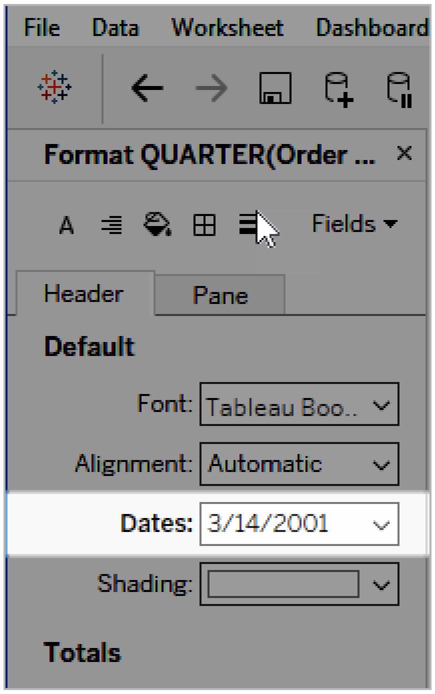
How to find the custom date format field

Format a date field in a view

To format a date field in the view, right-click (Control-click on a Mac) the field and choose **Format**.



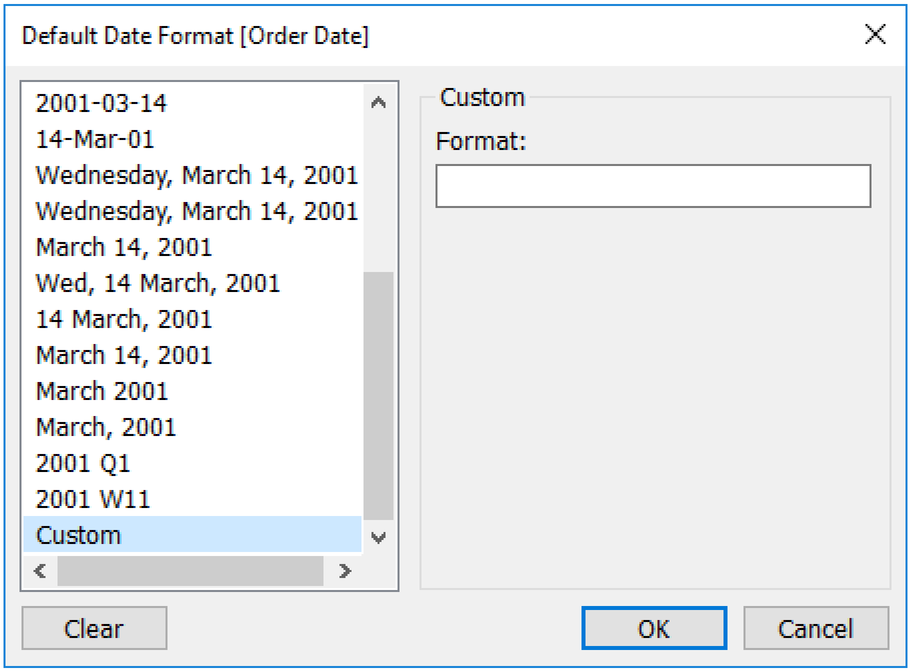
This will open the **Format** panel to the left of your view. Select the **Dates** field.



When you format dates, Tableau presents a list of available formats. In most cases, the last item in the list is **Custom**. You specify a custom date using format symbols listed in the **Supported date format symbols** on the next page table, either alone or in combination.

Format a date field in the Data pane

To format a date field in the **Data** pane, right-click the field and choose **Default Properties** > **Date Format**.



The date formats in the table are supported when your workbook is connected to a Tableau data extract or has a live connection to a data source that also supports the date format. (Refer to your data source's documentation to verify that the date format you want is supported.)

Tableau retrieves date formats from the data source. Tableau Server can also retrieve date formats from the Run As user account on the server that is running Tableau Server.

> **Note:** The following date formats might not be the same as those used with the **DATEPARSE** on page 2184 function. See **Convert a Field to a Date Field** on page 980 for more information.

Supported date format symbols

Use the following symbols to construct a custom date format.

Symbol	Description
(:)	Time separator. In some locales, a different character is used to represent the time separator. The time separator separates hours, minutes, and seconds when time values are formatted. The actual character used as the time separator in formatted output is determined by your system settings.

(/)	Date separator. In some locales, oa different character is used to represent the date separator. The date separator separates the day, month, and year when date values are formatted. The actual character used as the date separator in formatted output is determined by your system settings.
c	Display the date as `ddddd` and display the time as `ttttt`, in that order. Display only date information if there is no fractional part to the date serial number; display only time information if there is no integer portion.
d	Display the day as a number without a leading zero (1 31).
dd	Display the day as a number with a leading zero (01 31).
ddd	Display the day as an abbreviation (Sun Sat).
dddd	Display the day as a full name (Sunday Saturday).
ddddd	Display the date as a complete date (including day, month, and year), formatted according to your system's short date format setting. The default short date format is `m/d/yy`.
dddddd	Display a date serial number as a complete date (including day, month, and year) formatted according to the long date setting recognized by your system. The default long date format is `mmmm dd, yyyy`.
aaaa	The same as dddd, only it's the localized version of the string.
w	Display the day of the week as a number (1 for Sunday through 7 for Saturday).
ww	Display the week of the year as a number (1 54).
M	Display the month as a number without a leading zero (1 12). If `m` immediately follows `h` or `hh`, the minute rather than the month is displayed.
MM	Display the month as a number with a leading zero (01 12). If `m` immediately follows `h` or `hh`, the minute rather than the month

	is displayed.
MMM	Display the month as an abbreviation (Jan Dec).
MMMM	Display the month as a full month name (January December).
MMMMM	Display the month as a single letter abbreviation (J D)
oooo	The same as MMMM, but localized.
q	Display the quarter of the year as a number (1 4).
y	Display the day of the year as a number (1 366).
yy	Display the year as a 2-digit number (00 99).
yyyy	Display the year as a 4-digit number (100 9999).
h	Display the hour as a number without leading zeros (0 23).
Hh	Display the hour as a number with leading zeros (00 23).
N	Display the minute as a number without leading zeros (0 59).
Nn	Display the minute as a number with leading zeros (00 59).
S	Display the second as a number without leading zeros (0 59).
Ss	Display the second as a number with leading zeros (00 59).
000	Display milliseconds. Use a period character as a separator before specifying milliseconds.
t t t t t	Display a time as a complete time (including hour, minute, and second), formatted using the time separator defined by the time format recognized by your system. A leading zero is displayed if the leading zero option is selected and the time is before 10:00 A.M. or P.M. The default time format is `h:mm:ss`.
AM/PM	Use the 12-hour clock and display an uppercase AM with any hour before noon; display an uppercase PM with any hour between noon and 11:59 P.M.
am/pm	Use the 12-hour clock and display a lowercase AM with any hour

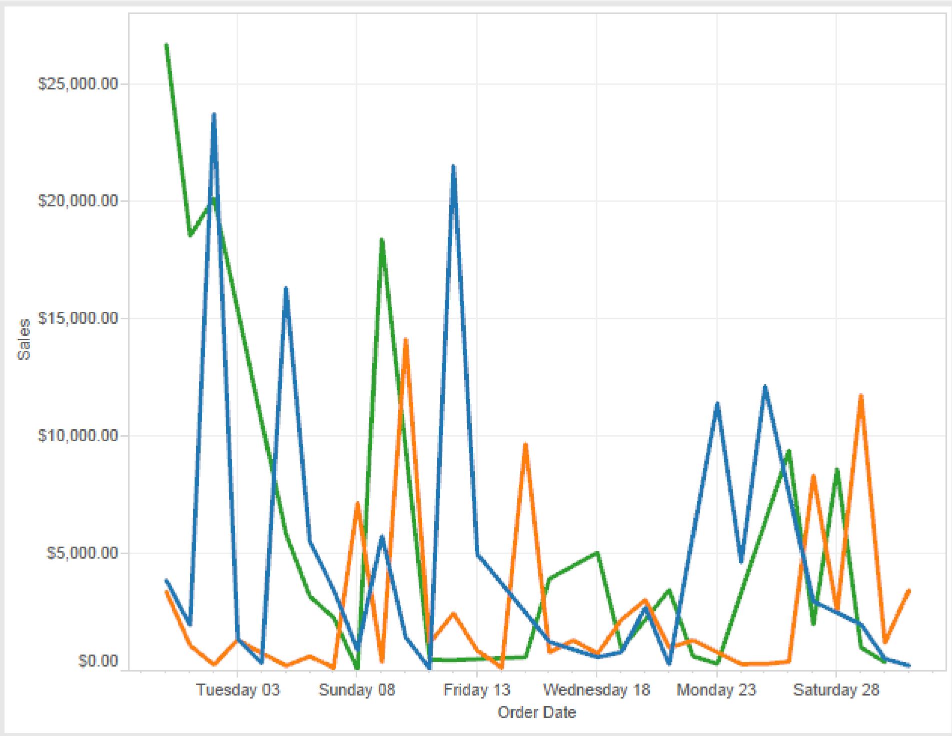
	before noon; display a lowercase PM with any hour between noon and 11:59 P.M.
A/P	Use the 12-hour clock and display an uppercase A with any hour before noon; display an uppercase P with any hour between noon and 11:59 P.M.
a/p	Use the 12-hour clock and display a lowercase A with any hour before noon; display a lowercase P with any hour between noon and 11:59 P.M.
AMPM	Use the 12-hour clock and display the AM string literal as defined by your system with any hour before noon; display the PM string literal as defined by your system with any hour between noon and 11:59 P.M. AMPM can be either uppercase or lowercase, but the case of the string displayed matches the string as defined by your system settings. The default format is AM/PM.

Custom date format examples

Any of the date formats symbols in the table above can be used alone or in combination.

Specifying a custom format yyyy-MM-dd HH:mm:ss.000 would produce dates in the format 2015-05-10 11:22:16.543. Such a format might be appropriate for scientific data.

Specifying a custom format DDDD DD would produce dates that show the Weekday and the Day, as shown below.



Specifying a custom format yy-mm-dd (dddd) would produce dates in the format 18-01-04 (Thursday).

Specifying a custom format "Q"1 YYYY would produce dates that show Q1 2018.

Support for Japanese era-based date formats

Starting with version 9.3, Tableau supports Japanese emperor-era-based date (Wareki) formats. Here's how to apply an era-based date format to a field in your view:

1. Set your workbook locale to Japanese.
2. Right-click the field in the view for which you want to set a date format.
3. Choose **Format**.
4. In the **Format** pane, from the **Dates** drop-down list, select a format.

If the format that you want isn't listed, you can construct your own date format. To do this, choose **Custom format** in the **Dates** box, then type your format using the Tableau date placeholders. The following era-based year placeholders are available:

Symbol	Description

g	Short era name (such as H for the Heisei era).
gg	Era name (such as 平成).
ggg	Long era name (for Japanese, this is currently the same as the regular era name).
e	Era-based year, such as 1 for the first year of an era.
ee	Era-based year, such as 01 for the first year of an era. If there is only one digit, then the era-based year will have a zero added to the front.

If your workbook locale is not Japanese, you can create a custom date format, then insert the language code !ja_JP! in front of your format, so that it looks like this:

```
!ja_JP! gg ee"年"mm"月"dd"日"
```

The language code forces the date to be treated as if it is a Japanese date.

Era-based dates are not fully supported by the Tableau Server browser view. In particular, if you publish a workbook that contains a quick filter, the e and g placeholders will not be filled in:

To avoid this issue, do not show era-based dates in quick filters if your workbook will be viewed in a browser.

Using literal text in a date format

You may want your date format to include some words or phrases, such as **Fiscal Quarter q of yyyy**. However, if you type that text directly into the Tableau format box, it may treat the letters like date parts:

Quarter of Order Date
Fi01/1/2010al 1uarter 1 of 2010
Fi04/1/2010al 2uarter 2 of 2010
Fi07/1/2010al 3uarter 3 of 2010
Fi010/1/2010al 4uarter 4 of 2010

To prevent Tableau from doing this, put double quotes around the letters and words that should not be treated as date parts: **“Fiscal Quarter” q “of” yyyy**.

If you want a literal quote inside of a quoted section, insert this code: “\””. For example, the format **“Fiscal “\”” Quarter”** would be formatted as **Fiscal “ Quarter**.

Format syntax in DATEPARSE function for extract data sources

If you're using the DATEPARSE function in an extract , use the syntax defined by the Unicode Consortium.

The following table lists the field types that can be represented in the format parameter of the DATEPARSE function. Click the field type to get information about the symbols, field patterns, examples, and descriptions from the Unicode Consortium website.

Unit of time	Notes
Era	n/a
Year	All symbols are supported in .hyper extracts with the exception of "U." **Notes:** • Negative values denote a year before Christ (BC). For example, DATEPARSE('y',''-10') returns the first January of 11BC and DATEPARSE('y,'-0') returns the first January of 1BC. • When working with calendar year "y," the pattern "yy" requests the two low-order digits of the year. For numbers < 70, the DATEPARSE function returns the year 2000+x. For numbers >=70, the DATEPARSE function returns the year 1900+x. • When working with "Y" in "ISO week date" based calendars, the year transition occurs on a week boundary and may differ from calendar year transition. The "Y" designation is used in conjunction with pattern character "w" in ISO year-week calendar. The ISO week date system is effectively a leap week

Unit of time	Notes
	calendar system that is part of the ISO 8601 date and time standard. Similar to "y," negative values for "Y" denote a year before Christ (BC).
Month	All symbols are supported in .hyper extracts with the exception of "l." **Notes:** • The month designations is used in conjunction with "d" for day number. • In contrast to ICU, .hyper extracts allow values 1-12. Other values will cause an error.
Week	All symbols are supported in .hyper extracts with the exception of "W." **Notes:** • When working with "w," in contrast to ICU, .hyper extracts allow only valid weeks. A year has 52 or 53 weeks (ISO 8601). The DATEPARSE function validates the input. For example, an error occurs for the 53rd week of 2016 because the 53rd week does not exist for 2016. • When working with "W," ICU does not support this designation, but it's useful for dates like 1st Monday of September.
Day	**Notes:** • When working with "d," in contrast to ICU, .hyper extracts only allow valid day numbers. For example, an error occurs for the 31st of February. • When working with "D," in contrast to ICU, .hyper extracts only allow valid day numbers. For example, an

<table>
<tr><th>Unit of time</th><th>Notes</th></tr>
<tr><td></td><td>error occurs for the 366th day of 2017.</td></tr>
<tr><td>Hour</td><td>Only "h" and "H" symbols are supported in .hyper extracts.
Notes:
When working with "h," .hyper extract don't allow negative values for this field. Negative values cause an error.
When working with "H," .hyper extracts don't allow negative values for this field. Negative values will cause an error.</td></tr>
<tr><td>Minute</td><td>Note: In contrast to ICU, .hyper extracts don't allow negative values for this field. Negative values will cause an error.</td></tr>
<tr><td>Second</td><td>Notes:
In contrast to ICU, .hyper extracts don't allow negative values for this field. Negative values will cause an error.
When working with "S," DATEPARSE('ss.SSSS', '12.3456') returns 1990-01-01 00:00:12:3456 AD.</td></tr>
<tr><td>Quarter</td><td>Note: In contrast to ICU, .hyper extracts only allow values 1-4. All other values will cause an error.</td></tr>
<tr><td>Weekday</td><td>Notes:
When working with "e and "ee," in contrast to ICU, .hyper extracts only allow values 1-7. All other values will cause an error.
When working with "c..cc," in contract to ICU, .hyper extracts only allow values 1-7. All other values will cause an error.</td></tr>
<tr><td>Period</td><td>n/a</td></tr>
</table>

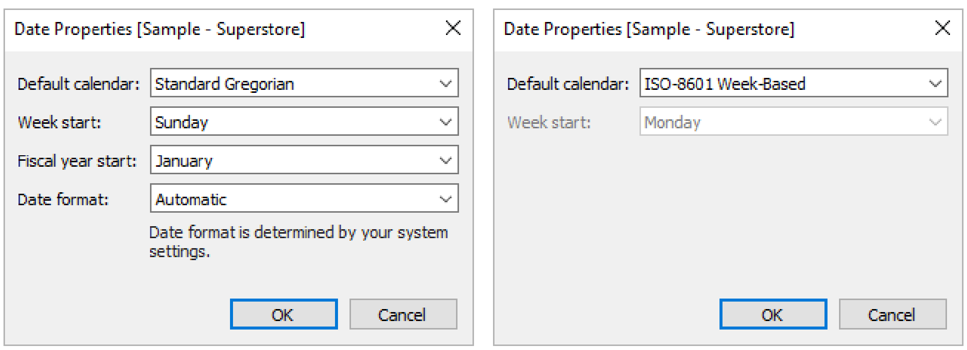
ISO-8601 Week-Based Calendar

The ISO-8601 Week-Based Calendar is an international standard for date-related data. The purpose of the ISO-8601 calendar is to provide a consistent and clear method to represent and calculate dates. ISO-8601 calendars divide dates into Years, Quarters, Weeks, and Week Days. Unlike the Gregorian calendar, ISO-8601 calendars have a consistent number of weeks in each quarter and a consistent number of days each week, making the ISO-8601 calendar popular when calculating retail and financial dates.

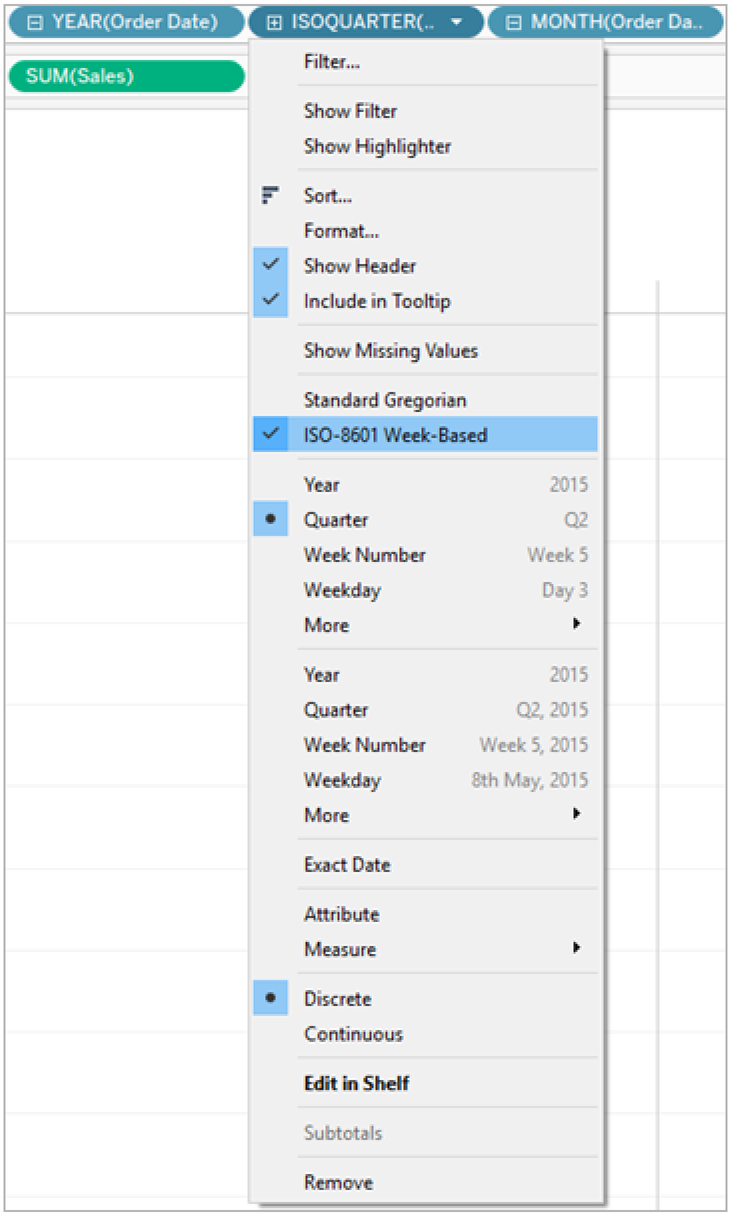
Set ISO-8601 as the default calendar

To set the ISO 8601 calendar for a data source, follow these steps:

1. Right-click (Control-click on Mac) the data source in the **Data** pane to open the Date Properties dialog box.
2. Set the **Default calendar** field to ISO-8601 Week-based .



For each date field on the shelf, you can set the calendar type separately. In the shelf, right-click a date dimension (Ctrl-click on Mac), and select **ISO-8601 Week-Based**.



Whether a given level of a date dimension is affected by the use of the Fiscal Calendar depends on the specific case.

Gregorian Date Level	ISO-8601 Date Level	Differences between ISO-8601 and Standard Gregorian calendars
Year	ISOYEAR	ISO-8601 years always start on the first Monday closest to January 1st. This may mean that the year does not start until

Gregorian Date Level	ISO-8601 Date Level	Differences between ISO-8601 and Standard Gregorian calendars
		January 4th in some cases, or may start in late December in others. Gregorian calendars always start the year on the 1st of January. This can cause some discrepancy in years between the two calendar systems when very close to the beginning of January.
Quarter	ISOQUARTER	The first three quarters in the ISO-8601 always have 13 weeks in them, with the last Quarter having either 13 or 14 weeks in it, depending upon the start of the next ISO-8601 year.
Month		The ISO-8601 Week-Based calendar does not use Months. Many retail and financial systems divide ISO-8601 Quarters into three segments of 4-4-5 weeks, though other segment systems also exist.
Week Number	ISOWEEK	All weeks in the ISO-8601 Week-Based calendar have exactly 7 days, start on a Monday, and each week belongs to single year. Unlike the Gregorian calendar, there are no weeks that extend across years. Each ISO-8601 year is either a Long or a Short year, with 52 or 53 weeks depending on when the ISO-8601 year begins.
Day		The ISO-8601 Week-Based calendar does not use Days.
Weekday	ISOWEEKDAY	ISO-8601 Weekdays always start on Monday.

Date Functions that support ISO-8601

When a date dimension is using the ISO-8601 Week-Based calendar, only the following date functions will reflect the ISO-8601 Week-Based calendar.

DATEADD and DATEDIFF

If the DATEADD and DATEDIFF calculations are called with a non- ISO-8601 date value (such as `day` or `month`), the resulting field will truncate an ISO-8601 date to the nearest ISO level. For example, adding one month to an ISO-8601 date will be reflected in the `iso-week` changing.

DATENAME

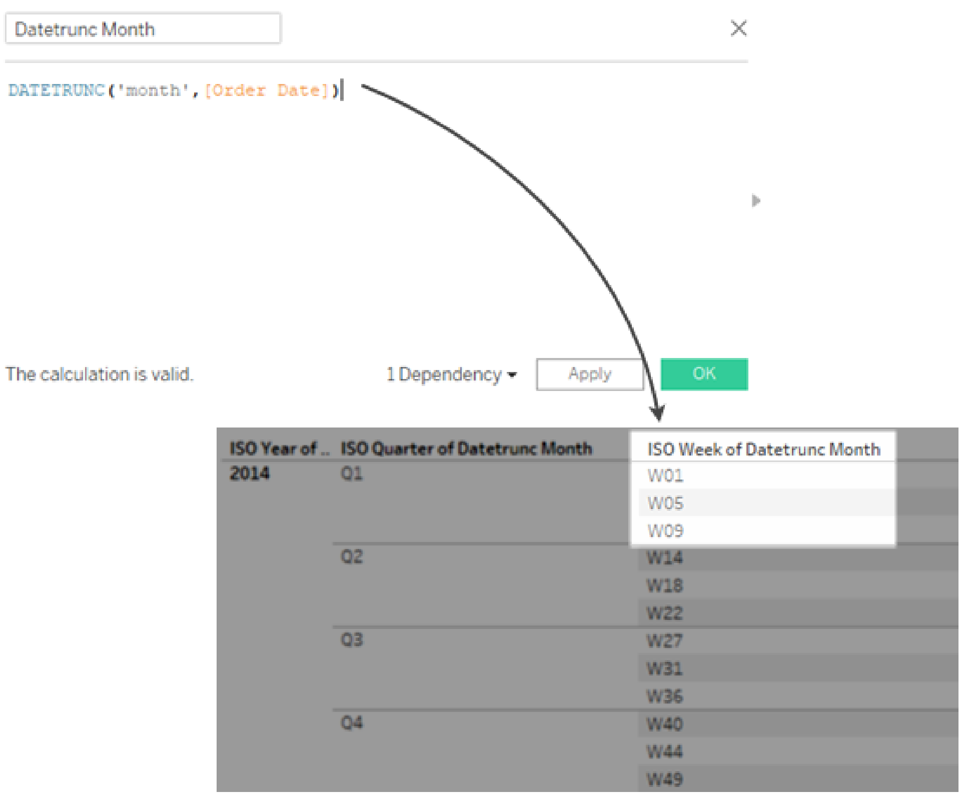
When using DATENAME, the resulting calculation will return Gregorian dates unless you use the `iso-year`,`iso-quarter`, `iso-week`, or `iso-weekday` date parts.

DATEPART

When using DATEPART, the resulting calculation will return Gregorian dates unless you use the `iso-year`,`iso-quarter`, `iso-week`, or `iso-weekday` date parts.

DATETRUNC

The DATETRUNC calculation will truncate an ISO-8601 date to the nearest ISO level. For example, as you can see in the image below, even though the ISO-8601 does not use Months, truncating the ISO date at 'month' will display the first `iso-week` of the Gregorian month.



Create a 4-4-5 calendar with ISO-8601 date functions

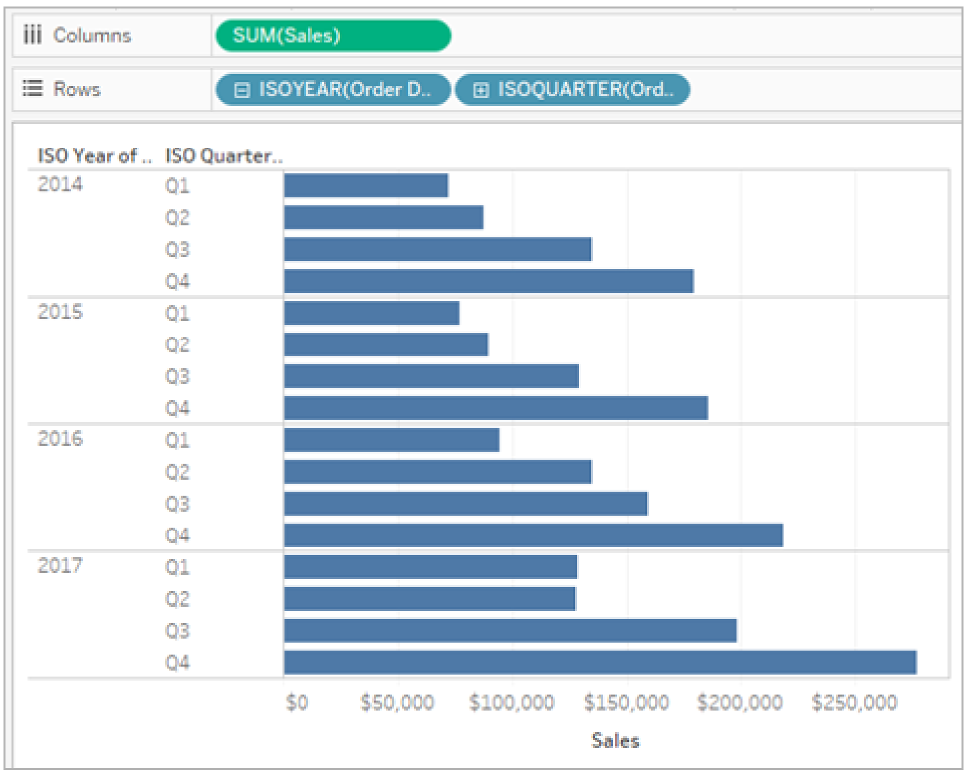
The ISO-8601 Week-Based calendar does not use Months. Many retail and financial systems divide ISO-8601 Quarters into three segments of 4-4-5 weeks, though other segment systems also exist.

Follow along with these steps to learn how to implement a 4-4-5 calendar in Tableau Desktop.

Step 1: Set up the Visualization

1. Open Tableau Desktop and connect to the **Sample-Superstore** saved data source.
2. **Set ISO-8601 as the default calendar** on page 1273.
3. Navigate to a new worksheet.
4. From the **Data** pane, under **Dimensions**, drag **Order Date** to the **Rows** shelf.
5. On the Rows shelf, click the '+' icon on ISOYEAR(Order Date).

   ISOQUARTER(Order Date) is added to the shelf.
6. From the **Data** pane, under **Measures**, drag **Sales** to the **Columns** shelf.



Step 2: Find the number of weeks in each quarter

In a 4-4-5 segment calendar, each quarter is divided into 3 segments with 4 or 5 weeks in them. We must find how many weeks are in each quarter in order to divide them into segments.

1. Select Analysis > Create Calculated Field.
2. Name the calculated field Weeks in Quarter
3. Enter the following calculation in the calculation dialog

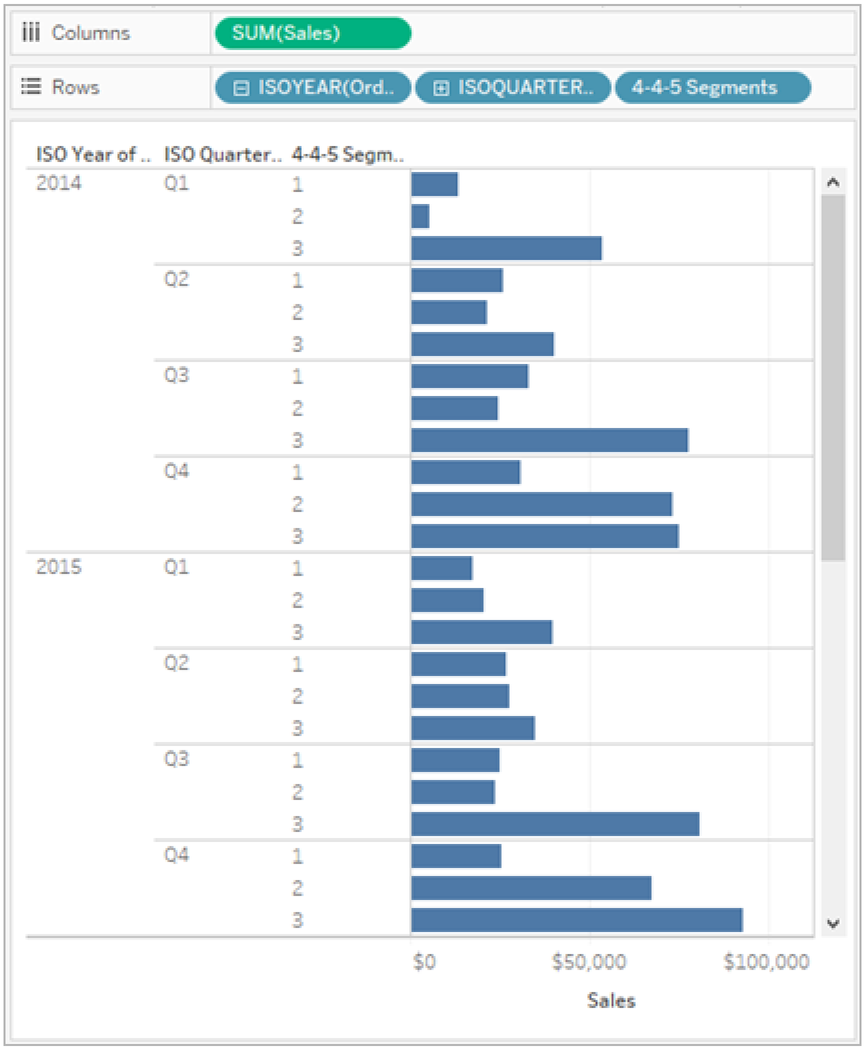
```
DATEPART('iso-quarter', [Order Date])*13 - DATEPART('iso-week', [Order Date])
```

Step 3: Create the 4-4-5 calendar calculation

1. Select Analysis > Create Calculated Field.
2. Name the calculated field 4-4-5 Segment
3. Enter the following calculation in the calculation dialog

```
IF([Weeks in Quarter] > 8) THEN "1" ELSEIF ([Weeks in Quarter] > 4) THEN "2" ELSE "3" END
```

4. From the data pane, drag the 4-4-5 Segment calculated field to the Rows shelf.



See Also

Dates and Times on page 1245

Date Functions on page 2170

Format Dates Using ISO-8601 Weeks and Years

For many European locales (listed below), Tableau allows you to format dates using ISO-8601 weeks and years.

Expand to see European locales that support ISO-8601 in

Tableau

da-DK, Danish (Denmark)

se-SE, Sami, Northern (Sweden)

rm-CH, Romansh (Switzerland)

sms-FI, Sami, Skolt (Finland)

is-IS, Icelandic (Iceland)

de-DE, German (Germany)

km-KH, Khmer (Cambodia)

nl-NL, Dutch (Netherlands)

smj-SE, Sami, Lule (Sweden)

fi-FI, Finnish (Finland)

nn-NO, Norwegian Nynorsk (Norway)

it-IT, Italian (Italy)

it-CH, Italian (Switzerland)

nb-NO, Norwegian Bokmål (Norway)

pl-PL, Polish (Poland)

sv-SE, Swedish (Sweden)

et-EE, Estonian (Estonia)

lv-LV, Latvian (Latvia)

hsb-DE, Upper Sorbian (Germany)

kl-GL, Greenlandic (Greenland)

mk-MK, Macedonian (North Macedonia)

fo-FO, Faroese (Faroe Islands)

se-NO, Sami, Northern (Norway)

as-IN, Assamese (India)

fy-NL, Frisian (Netherlands)

se-FI, Sami, Northern (Finland)

ps-AF, Pashto (Afghanistan)

prs-AF, Dari (Afghanistan)

de-CH, German (Switzerland)

sv-FI, Swedish (Finland)

dsb-DE, Lower Sorbian (Germany)

de-LU, German (Luxembourg)

fr-CH, French (Switzerland)

smj-NO, Sami, Lule (Norway)

de-LI, German (Liechtenstein)

sma-NO, Sami, Southern (Norway)

sma-SE, Sami, Southern (Sweden)

smn-FI, Sami, Inari (Finland)

The week and year numbering in an ISO-8601 calendar is different from a standard Gregorian calendar. Here’s how January 2nd, 2011 would be represented:

Calendar System	Week Number	Year Number
Standard Gregorian	1	2011
ISO-8601	52	2010

When formatting dates, it’s important to ensure that your month, week and year numbers all come from the same calendar system. Otherwise, the date might not make sense. There are two ways to do this:

- Let Tableau guess which calendar system you want to use for each placeholder. For example, if you type `mm yyyy` (a month followed by a year), Tableau uses the standard Gregorian year. In this case, it makes sense to use that year with a month number.

- In your format string, place an annotation after each week (ww) or year (yyyy) placeholder, to indicate what kind of placeholder it should be:

Annotation	Calendar to Use	Example Format String
[Y]	ISO-8601	ww[Y] yyyy[Y]
[y]	Standard Gregorian	ww[y] yyyy[y]

Important: To get correct ISO-8601 week numbering, your computer's location must be set to one of the above locales, and you must set your data source start of week to Monday. (For information on how to do this, see **Date Properties for a Data Source** on page 1247.)

Otherwise, Tableau will number weeks using your data source start of week setting, which may result in partial weeks at the beginning and end of years being created.

Note: ISO week numbering will not occur if you are using a fiscal year that doesn’t start in January, regardless of the locale you choose.

Sample format strings for the date December 31, 2013

Here are sample format strings for the date December 31, 2013, where m stands for a month or day-of-month placeholder. Time placeholders, such as hours, minutes, and seconds, don’t influence whether or not a year is week-based or year-based.

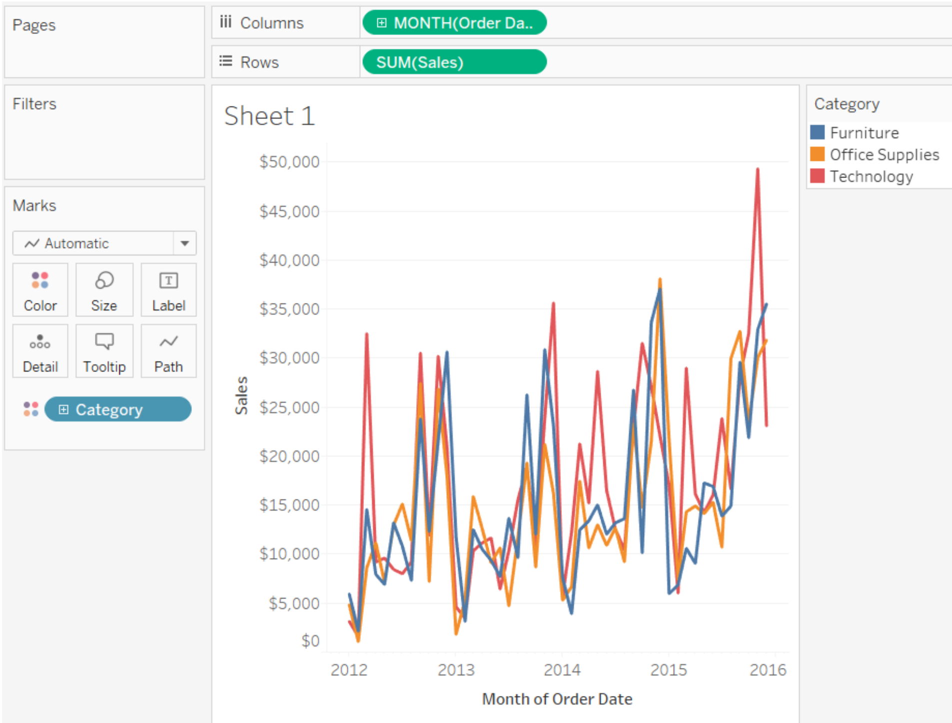
Format string	Formatted output
yyyy	2013
yyyy[Y]	2014
yyyy ww	2014 1
yyyy mm	2013 12
ww yyyy	1 2014
mm yyyy	12 2013
ww yyyy[y]	1 2013
mm yyyy[Y]	12 2014

yyyy mm ww	2013 12 1
mm yyyy ww	12 2013 1
ww mm yyyy	1 12 2013
ww mm yyyy[Y]	1 12 2014
ww yyyy mm	1 2014 12 (format is ambiguous)
ww yyyy[y] mm	1 2013 12
ww yyyy mm yyyy	1 2014 12 2013
ww yyyy yyyy mm	1 2014 2013 12
yyyy mm ww yyyy	2013 12 1 2014

Continuous Dates

You can treat a date as a continuous quantity after placing the field on a shelf. You do this by selecting one of the Continuous date options on the field’s context menu (lower list of date levels). Continuous dates draw a quantitative axis for the date values.

For example, the view below displays the sales as a function of a continuous order date and is color-encoded by category. As you can see, the color of the **Order Date** field changes from blue to green after it is converted to a continuous quantity.



Treating dates as a continuous quantity is particularly useful when you use Gantt bars or want to see trends using line charts as shown above.

By default, date dimensions are discrete fields for which Tableau automatically selects a date level when it is placed on a shelf. To make a date dimension continuous by default, right-click (control-click on Mac) the field in the **Data** pane and select **Convert to Continuous**. The field turns green and is automatically converted to a continuous field when you drag it to a shelf. To revert to discrete again, right-click (control-click on Mac) the field in the **Data** pane and select **Convert to Discrete**.

Build Data Views from Scratch

This section describes the various features at your disposal and tasks related to building visualizations in Tableau.

Every view that you build in Tableau should start with a question. What do you want to know? Every time you drag a field into the view or onto a shelf, you are asking a question about the data. The question will vary depending on where you drag various fields, the types of fields, and the order in which you drag fields into the view.

For every question you ask about your data, the marks in your viz update to represent the answer visually—with shapes, text, size, color, headers, axes, hierarchies, or table structures.

For information on building and viewing visualizations on the web, see **Use Tableau on the Web** on page 3101.

Looking for a data set to work with? Check out our tips to find good data sets.

For a 6-minute walkthrough on building data views, see the Getting Started with Visual Analytics free training video. Use your tableau.com account to sign in.

For an in-depth, 25-minute walkthrough of the Tableau environment, see the Getting Started free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Start Building a Visualization by Dragging Fields to the View

The basic structure for every view is made of dimension and measure fields placed on the Rows and Columns shelves, and on different properties in the Marks card.

You build visualizations by adding fields from the Data pane to the view. As you build a visualization, you can add as many fields as necessary to different areas of the view, and then move those fields around as your explore your data.

Watch a video: To see related concepts demonstrated in Tableau, watch Getting Started with Visual Analytics, a 6-minute free training video. Use your tableau.com account to sign in.

For an in-depth, 25-minute walkthrough of the Tableau environment, see the Getting Started free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Different ways to start building a view

When you build a view, you add fields from the **Data** pane. You can do this in different ways. For example:

- Drag fields from the **Data** pane and drop them onto the cards and shelves that are part of every Tableau worksheet.
- Double-click one or more fields in the **Data** pane.
- Select one or more fields in the **Data** pane and then choose a chart type from **Show Me**, which identifies the chart types that are appropriate for the fields you selected. For details, see **Use Show Me to Start a View** on page 1323.
- Drop a field on the **Drop field here** grid, to start creating a view from a tabular perspective.

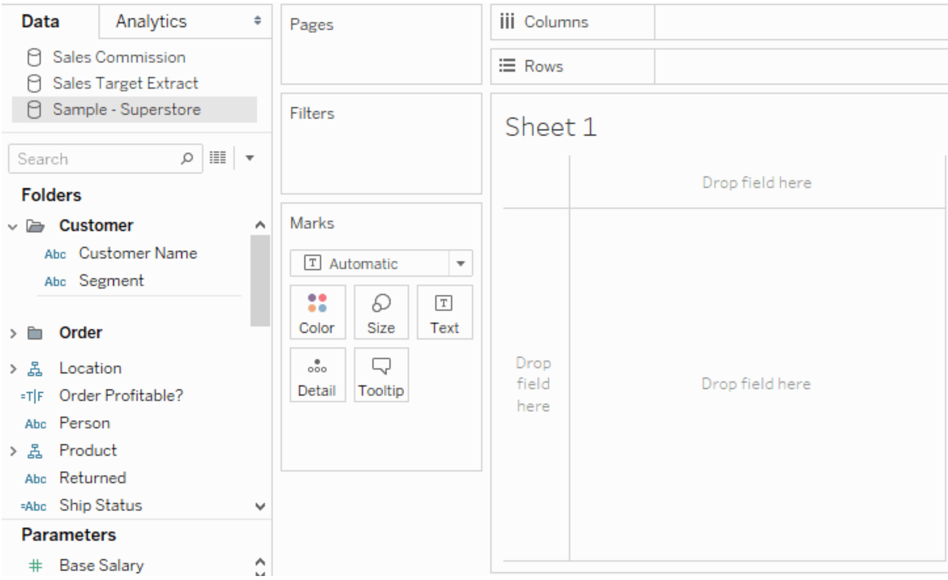
As you start exploring data in Tableau, you will find there are many ways to build a view. Tableau is extremely flexible, and also very forgiving. As you build a view, if you ever take a path that isn't answering your question, you can always undo to a previous point in your exploration.

- To undo or redo, click undo ← or redo → on the toolbar.

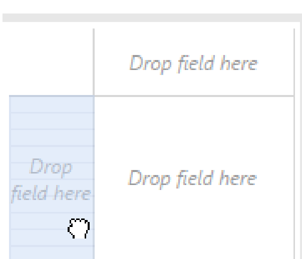
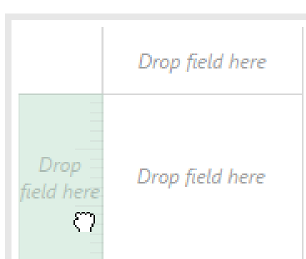
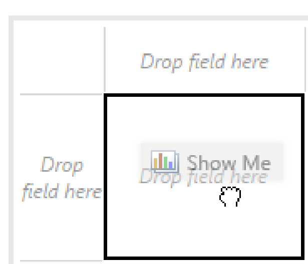
You can undo all the way back to the last time you opened the workbook. You can undo or redo an unlimited number of times.

Dragging basics

When you begin creating a new data view on a blank worksheet, you can drag a field from the **Data** pane and drop it directly into the view, rather than to a shelf.

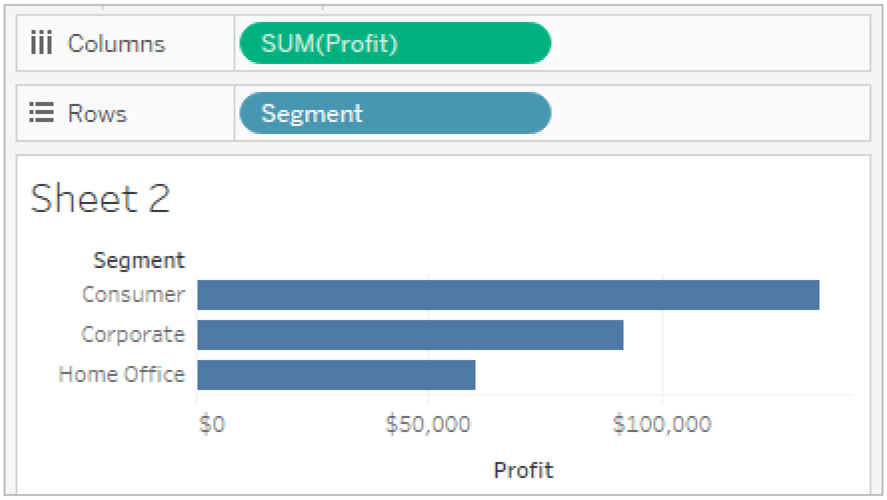


In Tableau Desktop, while dragging fields, you can hover over the different areas in the view to see how the field will be incorporated into the structure of the view. For example, dimensions typically add a row and column headers to the view, while measures add continuous axes. Below are some examples of how fields can be added to the view.

Dimensions add headers	Measures add axes	Show Me automatically adds the field according to best practices
		

For a more advanced discussion of dimensions and measures, see **Dimensions and Measures, Blue and Green** on page 314.

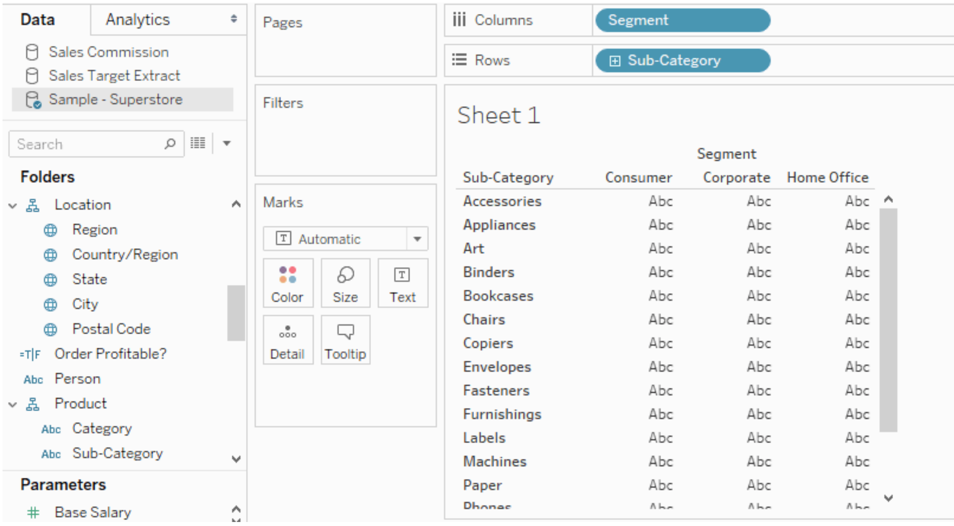
When you drop a field on one of the active areas in the view, the field is added to the view and also to one of the shelves or cards. For example, in the view below the **Segment** dimension was added to the **Rows** shelf and the **Profit** measure was added to the **Columns** shelf—and automatically aggregated so that the values in the data are summed.



You can, of course, also drag fields directly to the shelves instead of to the active areas in the view. You can also drag fields from one shelf to another shelf. To remove a field from a shelf, drag it off the worksheet or select **Remove** from the field's menu (available when you right-click (control-click on Mac) a field in the view. In Tableau Desktop, you can also quickly remove multiple fields from a shelf by right-clicking (control-click on Mac) the shelf and selecting **Clear Shelf**.

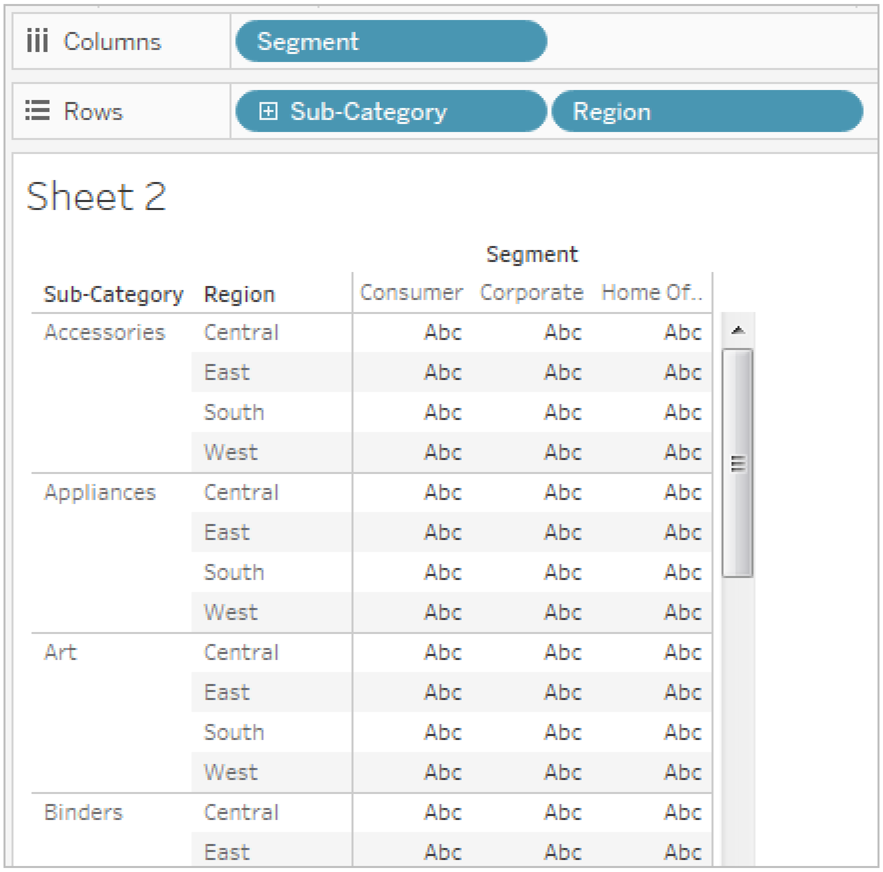
Add headers

You can add headers to a view by dragging a discrete (blue) dimension and dropping it to either side of an existing header, or to the left of an axis. For example, in the view below you can add the **Region** dimension by dragging it and dropping it to the right of the Sub-Category names.



As you hover over the view, a dotted black line indicates active areas where you can add headers.

If you drop **Region** there, it adds a column for Region along the top of the table, and slices the data so that you see results for each region within each sub-category.

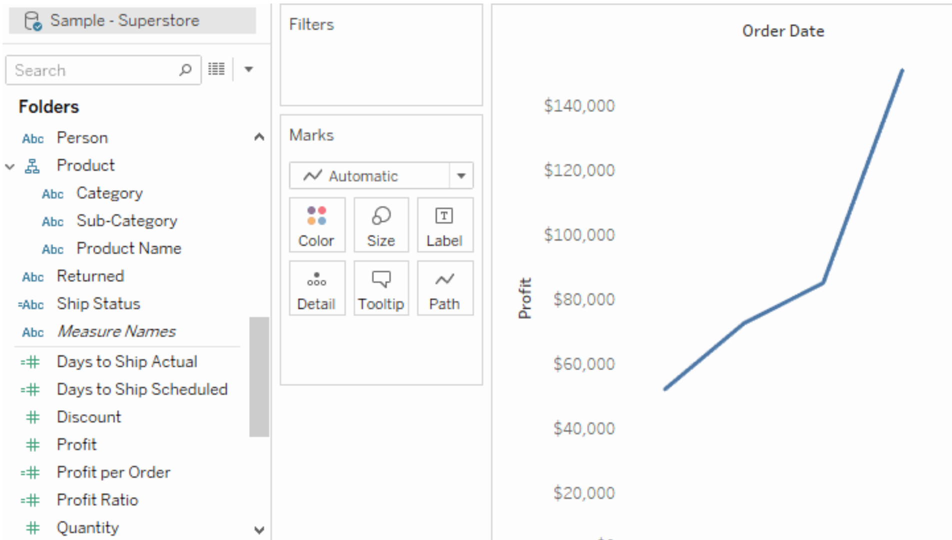


Add axes

You can add axes by dragging a continuous (green) measure and dropping it on an active area in the view. If an axis already exists in the view you can replace the existing axis, blend the new measure with the existing axis, or add a secondary axis.

Replace the Existing Axis

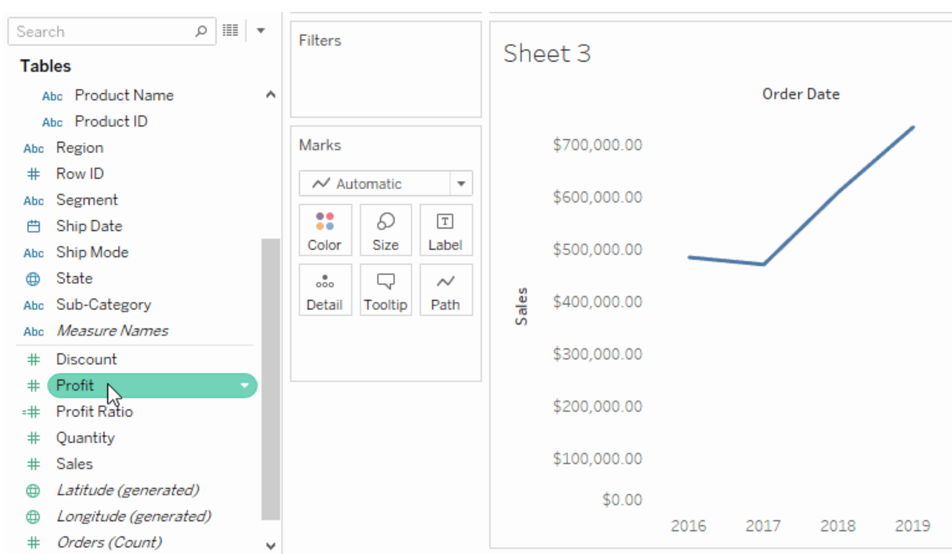
Drag the new measure to the top left portion of the axis in the view. A small square drop zone appears and a single axis icon displays to indicate that a single axis will be left when you drop the measure. In this case you are replacing one measure (**Sales**) with another (**Profit**).



Blend the Measures on Single Axis

You can show multiple measures on a single axis by dragging the new measure directly on top of the existing axis. Blending measures uses the Measure Names and Measure Values fields. For more information, see **Measure Values and Measure Names** on page 1310.

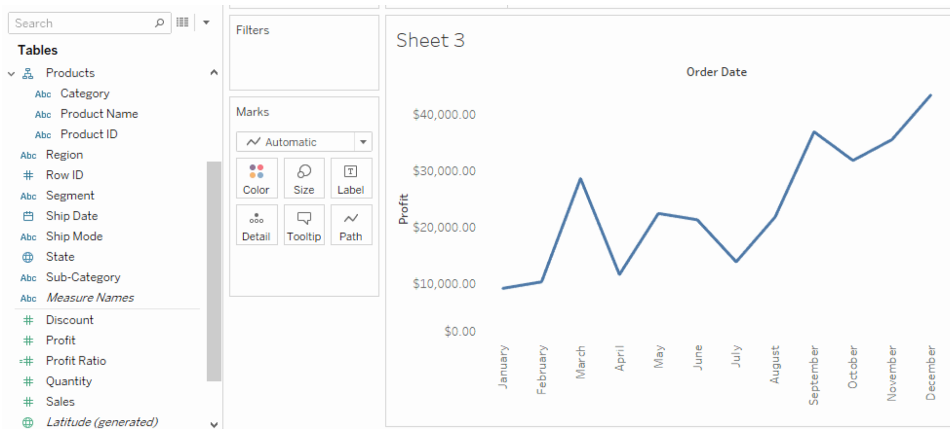
In the image below you are adding a second measure (**Profit**) to the existing measure (**Sales**) on the axis.



The result is a view with both measures plotted along a single axis. In this case, Sales is shown in orange and Profit is shown in blue.

Add a Dual Axis

Drag the field to the right side of the view to add the measure as a dual axis. Dual Axes are useful when you want to compare two fields that have different scales. In this case, blending the two axes would distort the view. Instead you can add a dual axis.

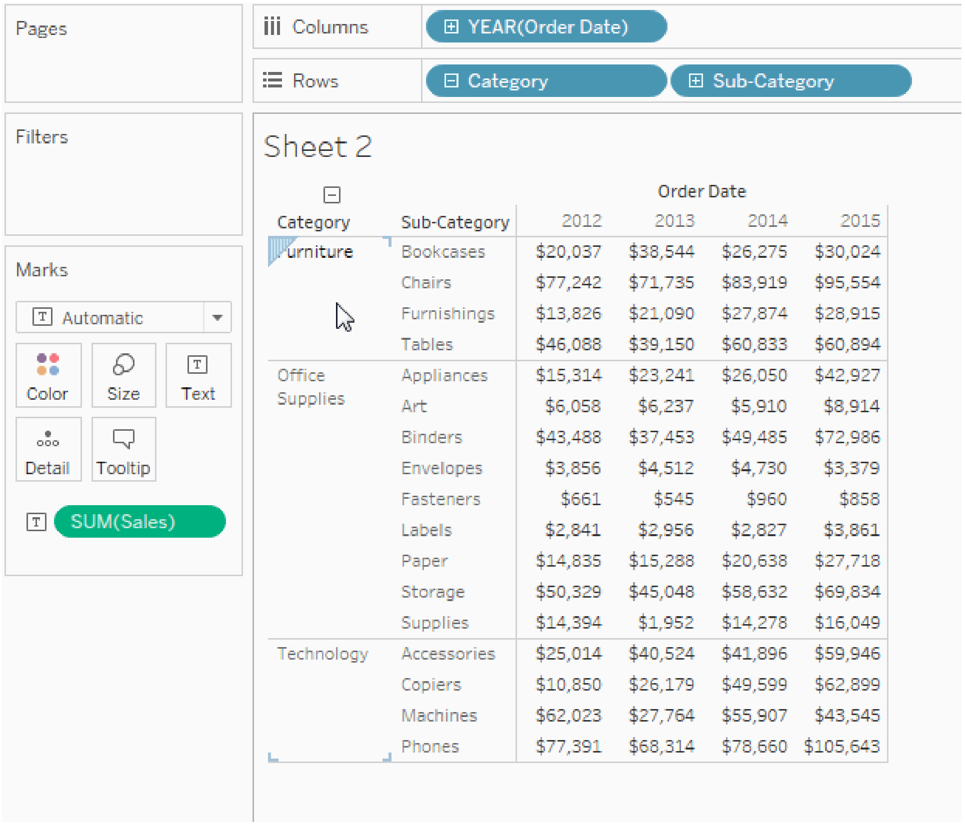


The result is a dual axis view where the Profit axis corresponds to the blue line and the Sales axis corresponds to the orange line.

You can add up to four axes to the view: two on the Columns shelf and two on the Rows shelf. To turn a measure into a dual axis, right-click (control-click on Mac) a field on the **Rows** or **Columns** shelf and select **Dual Axis**.

Rearrange rows and columns

You can rearrange the rows and columns in the view by changing the order of the fields in the Rows and Columns shelves, or by dragging the selection border for headers or an axis.

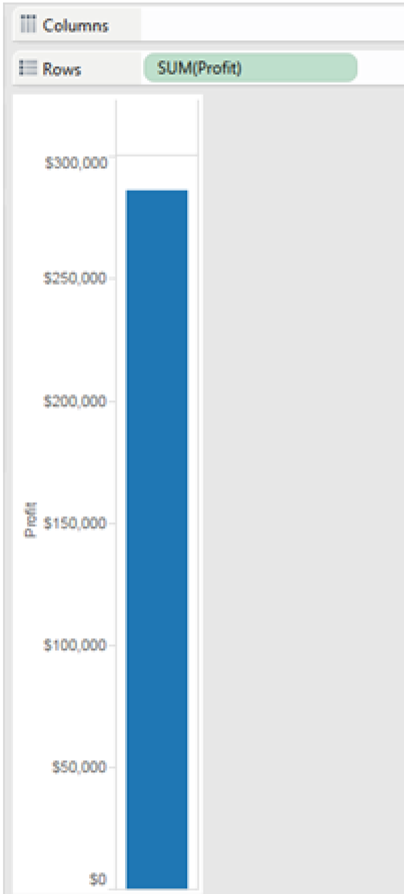


Category	Sub-Category	2012	2013	2014	2015
urniture	Bookcases	$20,037	$38,544	$26,275	$30,024
	Chairs	$77,242	$71,735	$83,919	$95,554
	Furnishings	$13,826	$21,090	$27,874	$28,915
	Tables	$46,088	$39,150	$60,833	$60,894
Office Supplies	Appliances	$15,314	$23,241	$26,050	$42,927
	Art	$6,058	$6,237	$5,910	$8,914
	Binders	$43,488	$37,453	$49,485	$72,986
	Envelopes	$3,856	$4,512	$4,730	$3,379
	Fasteners	$661	$545	$960	$858
	Labels	$2,841	$2,956	$2,827	$3,861
	Paper	$14,835	$15,288	$20,638	$27,718
	Storage	$50,329	$45,048	$58,632	$69,834
	Supplies	$14,394	$1,952	$14,278	$16,049
Technology	Accessories	$25,014	$40,524	$41,896	$59,946
	Copiers	$10,850	$26,179	$49,599	$62,899
	Machines	$62,023	$27,764	$55,907	$43,545
	Phones	$77,391	$68,314	$78,660	$105,643

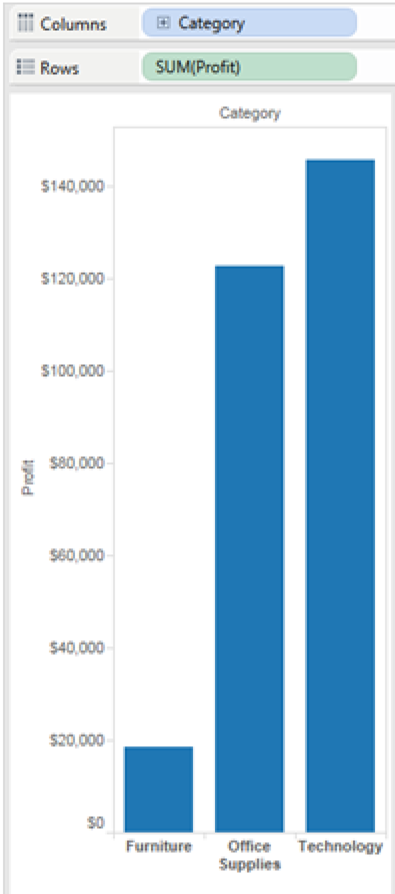
Automatically add fields to the view with double-click

Tableau supports an additional method for automatically generating views of data called Automatic Double-Click. To use this method, double-click fields in the Data pane you are interested in. Tableau automatically adds each field to the view. That is, each double-click results in an additional field added to a shelf in an intelligent way. Like **Show Me**, this function leverages Tableau's ability to make an intelligent "best guess" of how the data should be displayed. Double-clicking sometimes creates a view that you can use without further modification. More often, it can provide a starting point, which you can then modify to get the ideal result.

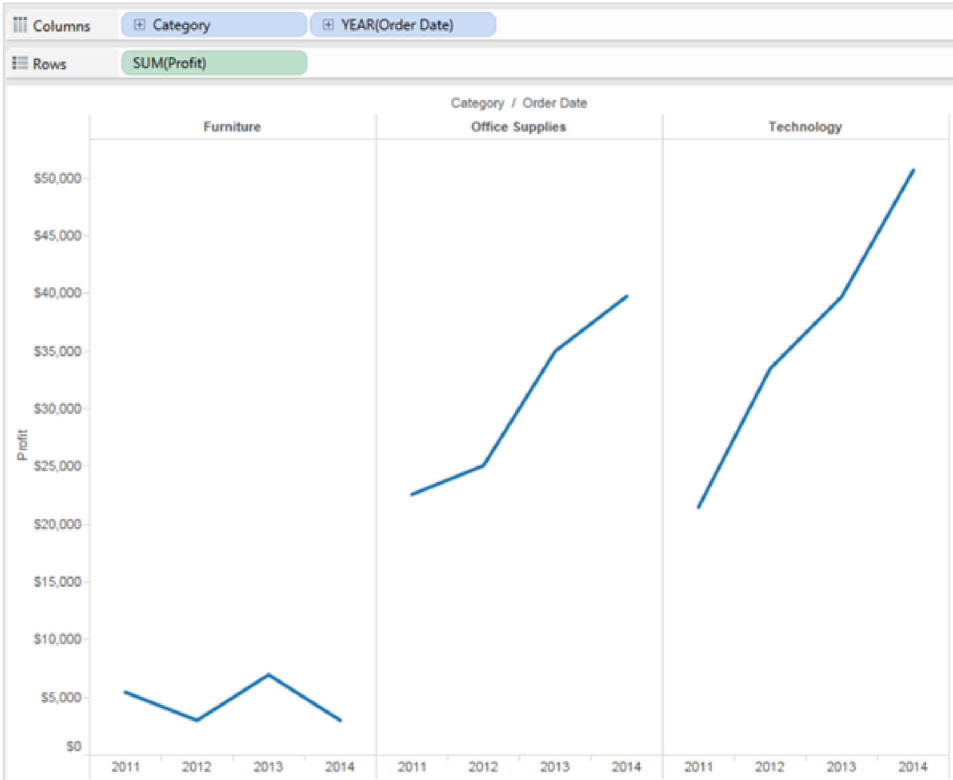
1. Double-clicking the **Profit** measure in the **Data** pane automatically adds that field to the view in an intelligent way.



2. Double-clicking the **Category** dimension in the **Data** pane automatically adds that field to the view based on the fact that Profit is already on the Rows shelf.



3. Double-clicking the **Order Date** dimension in the **Data** pane automatically adds that field to the **Columns** shelf based on the placement of other fields in the view. As you double-click fields they are successively added to the view. The order in which you click fields determines the type of view created.



The following table describes some of the rules used in creating automatic views by double-clicking fields in the **Data** pane.

Text Table	Adding a dimension first produces a text table (or cross-tab). All subsequent clicks on fields result in refinement of the text table.
Bars	Adding a measure first and then a dimension produces a bar view. All subsequent clicks result in refinement of the bar view, unless a date dimension is added, at which time the view is changed to a line.
Line	Adding a measure and then a date dimension produces a line view. All subsequent clicks result in refinement of the line view.
Continuous Line	Adding a continuous dimension and then a measure produces a continuous line view. Subsequent dimensions result in refinement of the continuous line view. Subsequent measures add quantitative axes to the view.
Scatter	Adding a measure and then another measure produces a scatter view. Subsequent dimensions result in refinement to the scatter view. Subsequent measures will create a scatter matrix.

Maps	Adding a geographic field produces a map view with latitude and longitude as axes and the geographic field on the Level of Detail shelf. Subsequent dimensions add rows to the view while subsequent measures further refine the map by adding size and color encoding.

Other resources

Build a Basic View to Explore Your Data on page 169

Build Common Chart Types in Data Views on page 1577

Add Axes for Multiple Measures in Views

There are several different ways to compare multiple measures in a single view. You can:

- Create individual axes for each measure.
- Blend two measures to share an axis.
- Add dual axes where there are two independent axes layered in the same pane.

In any of these cases you can customize the marks for each axis to use multiple mark types and add different levels of detail. Views that have customized marks are called combination or combo charts.

Note: For details on how to edit axes, see **Edit Axes** on page 2900.

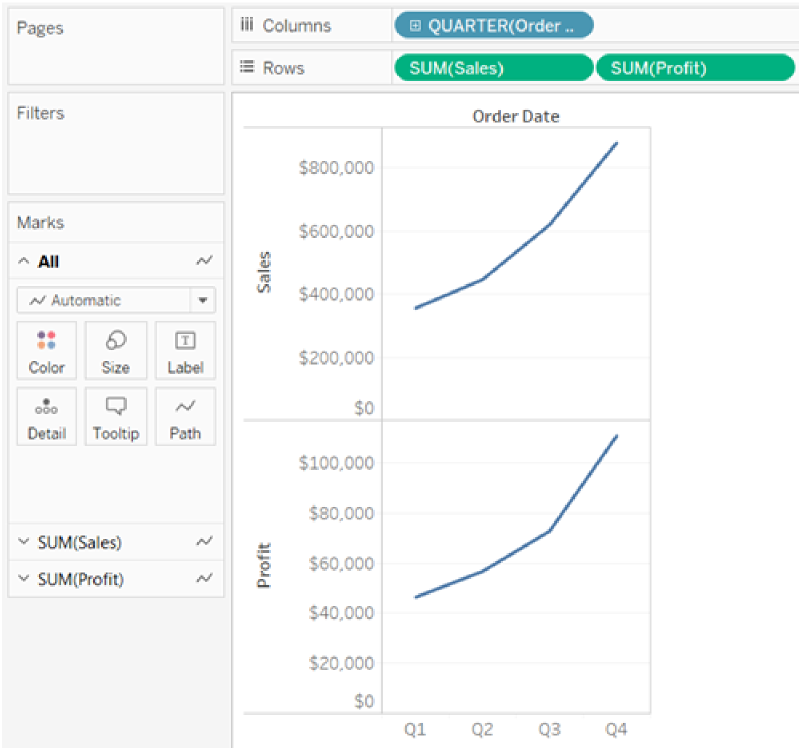
Add individual axes for measures

To add individual axes for each measure, drag measures to the **Rows** and **Columns** shelves.

- Adding a continuous field on the Rows shelf adds an additional axis to the rows of the table.
- Adding a continuous field on the Columns shelf adds an additional axis to the columns of the table.

The example below shows quarterly sales and profit. The Sales and Profit axes are individual rows in the table and have independent scales.

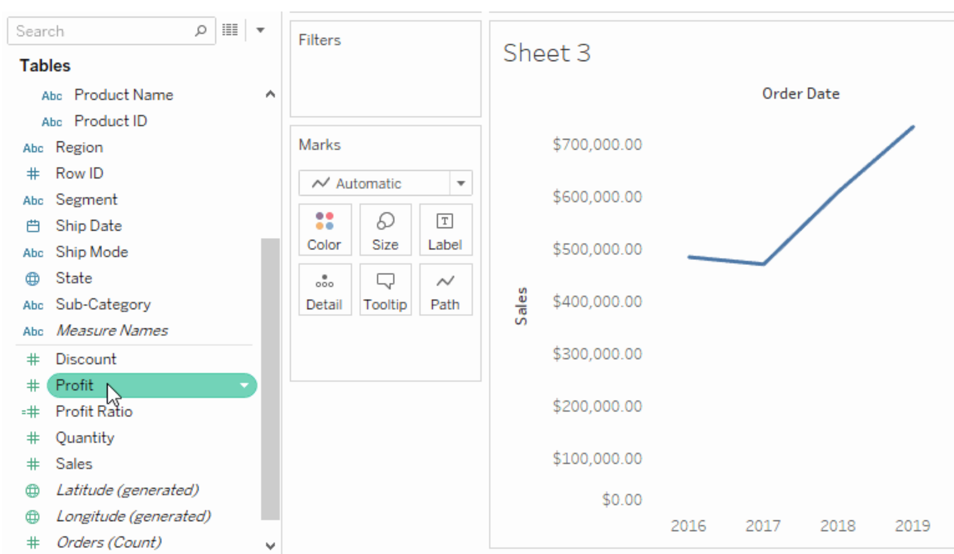
The Order Date field on the Columns shelf is a discrete date dimension. Because it is discrete, it creates headers rather than an axis. For details on how to edit axes, see **Edit Axes** on page 2900.



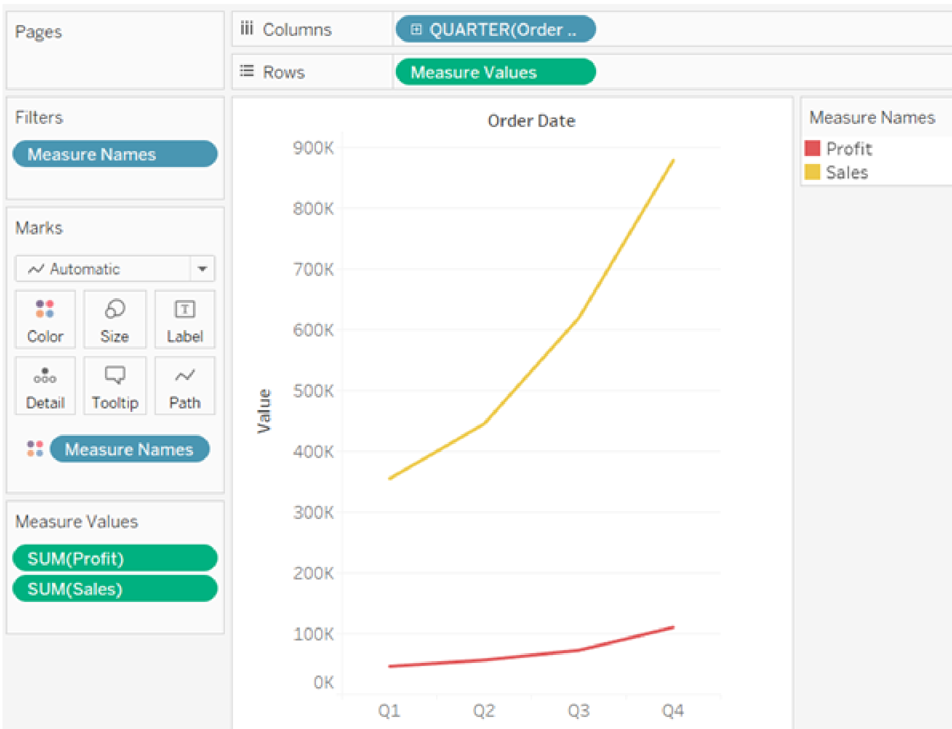
Blend axes for multiple measures into a single axis

Measures can share a single axis so that all the marks are shown in a single pane.

To blend multiple measures, drag one measure or axis and drop it onto an existing axis.



Instead of adding rows and columns to the view, when you blend measures there is a single row or column and all of the values for each measure is shown along one continuous axis. For example, the view below shows quarterly sales and profit on a shared axis.



> **Note:** If you drag a measure on to the canvas and only see a single ruler indicator instead of the double ruler indicator shown below, Tableau creates dual axes instead of a blended axis. For more information about how to create dual axes, see **Compare two measures using dual axes** on the next page.

Blending measures uses the **Measure Names** and **Measure Values** fields, which are generated fields that contain all of the measure names in your data source and all of the measure values. The shared axis is created using the **Measure Values** field. The **Measure Names** field is added to **Color** on the Marks card so that a line is drawn for each measure. Finally, the **Measure Names** field is filtered to only include the measures you want to blend.

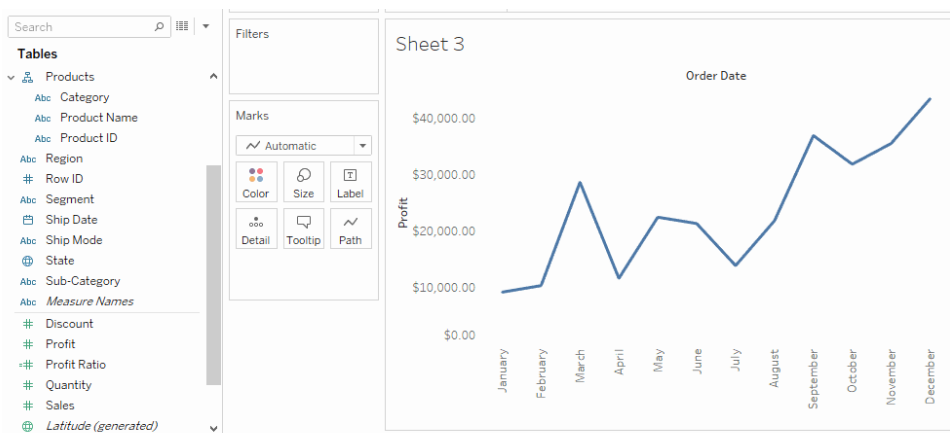
> **Note:** Blending axes is most appropriate when comparing measures that have a similar scale and units. If the scales of the two measures are drastically different, the trends may be distorted.

Compare two measures using dual axes

You can compare multiple measures using dual axes, which are two independent axes that are layered on top of each other. Dual axes are useful for analyzing two measures with different scales.

To add a measure as a dual axis, drag the field to the right side of the view and drop it when you see a black dashed line appear.

You can also right-click (control-click on Mac) the measure on the Columns or Rows shelf and select **Dual Axis**.



The result is a dual axis view where the Profit axis corresponds to the blue line and the Sales axis corresponds to the orange line.

You can add up to four layered axes: two on the Columns shelf and two on the Rows shelf.

Synchronize axes to use the same scale

To align the two axes in a dual axes chart to use the same scale, right-click (control-click on Mac) the secondary axis, and select **Synchronize Axis**. This aligns the scale of the secondary axis to the scale of the primary axis.

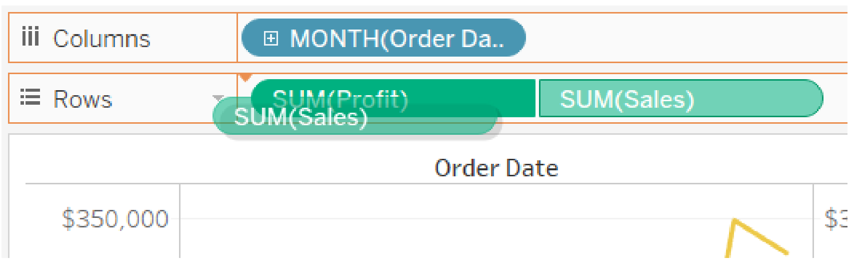
In this example, the Sales axis is the secondary axis and the Profit axis is the primary axis.

Note: To synchronize axes, the data types for both measures must be the same. If the data types for your measures are different, see the section below.

You can synchronize dual axes for numeric data types that don't match. For example, you can synchronize an axis that uses an integer data type and an axis that uses decimal data type.

If you would like to change which axis is the primary, and which axis is the secondary, select the field on the Columns or Rows shelf that is the secondary, and drag it in front of the primary field on the shelf until you see an orange triangle appear.

In this example, you can select the **SUM(Sales)** field on the **Rows** shelf, and drag it in front of the **SUM(Profit)** field. The Sales axis is now the primary and the Profit axis is the secondary.



Synchronize axes with measures of different data types

The **Synchronize Axis** option ensures that you make a scaled and correct comparison in a dual axes chart. However, sometimes this option may not be available (grayed out). This is because the data type of one of the axes is different from the other.

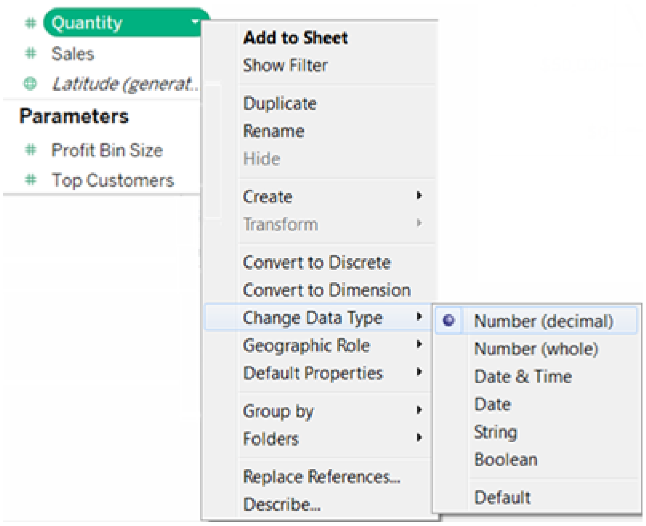
Note: In a chart with dual axes, starting with version 2018.1, you can synchronize dual axes for numeric data types that don't match. For example, you can synchronize an axis that uses an integer data type and an axis that uses decimal data type.

To resolve this issue, you must change the data type of one of the axes. Follow the example below to change the data type for an axis.

1. Click the new Worksheet  icon to open a new worksheet.
2. Drag **Order Date** to Columns, then click the drop-down arrow on the field on the shelf and select **Month**from the context menu.
3. Drag **Sales** to Rows, then drag **Quantity** to the right side of the view and drop it when you see a black dashed line appear to create a dual axis.

If you right-click on the **Quantity** Axis, you can see that the **Synchronize Axis** option is grayed out

4. In the Data pane, click on the drop-down arrow on the **Quantity** field, and select **Change Data Type** > **Number (decimal)** in the context menu. This changes the data type for this field.

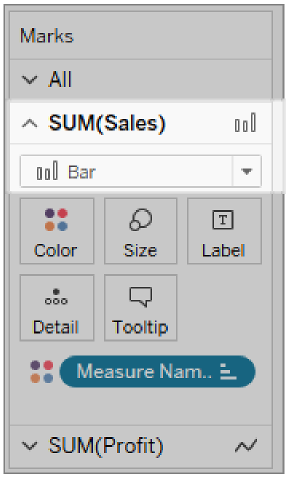


5. Replace the **Quantity** measure on the view.

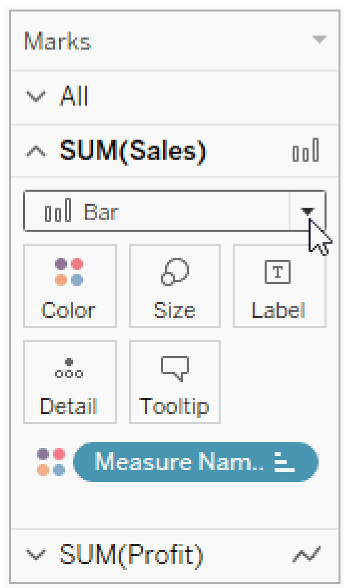
6. Right-click the secondary axis, and then select **Synchronize Axis**.

Customize the marks for a measure

1. Select the Marks card for the measure that you want to customize. There is a Marks card for each measure on the Rows and Columns shelves.

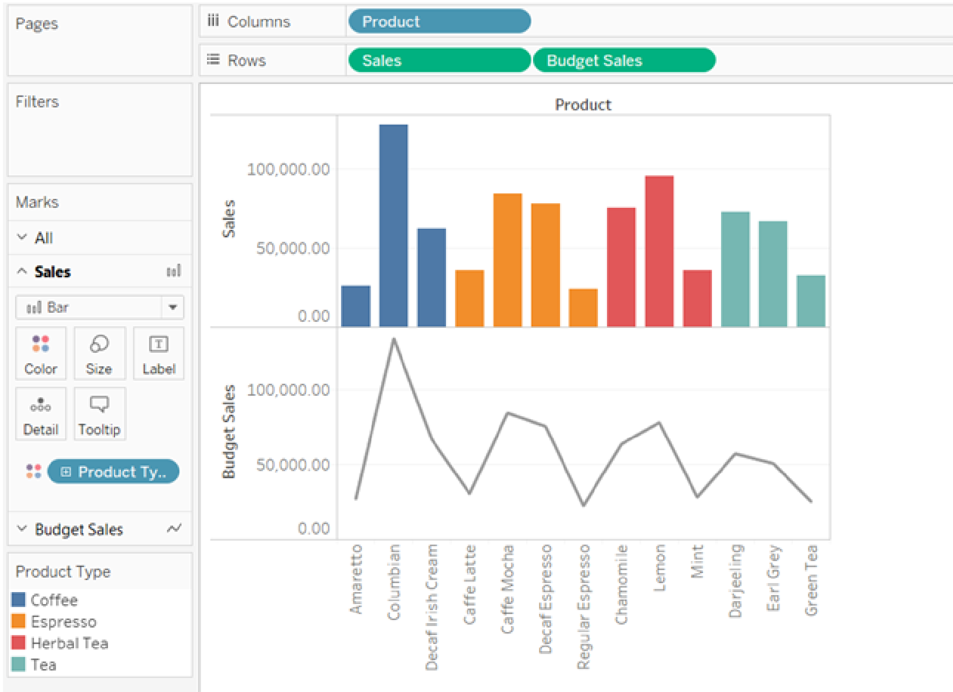


2. Select a new mark type for the measure.



Any changes to the mark type, shape, size, color, detail and other mark properties will be applied to the selected measure.

For example, in the view below the **Sales** Marks card is active. The Mark Type has been changed to **Bar** and when **Product Type** is placed on **Color** on the**Sales** Marks card, the encoding and level of detail is only applied to the Sales marks. The Budget Sales mark is not broken down by Product Type.

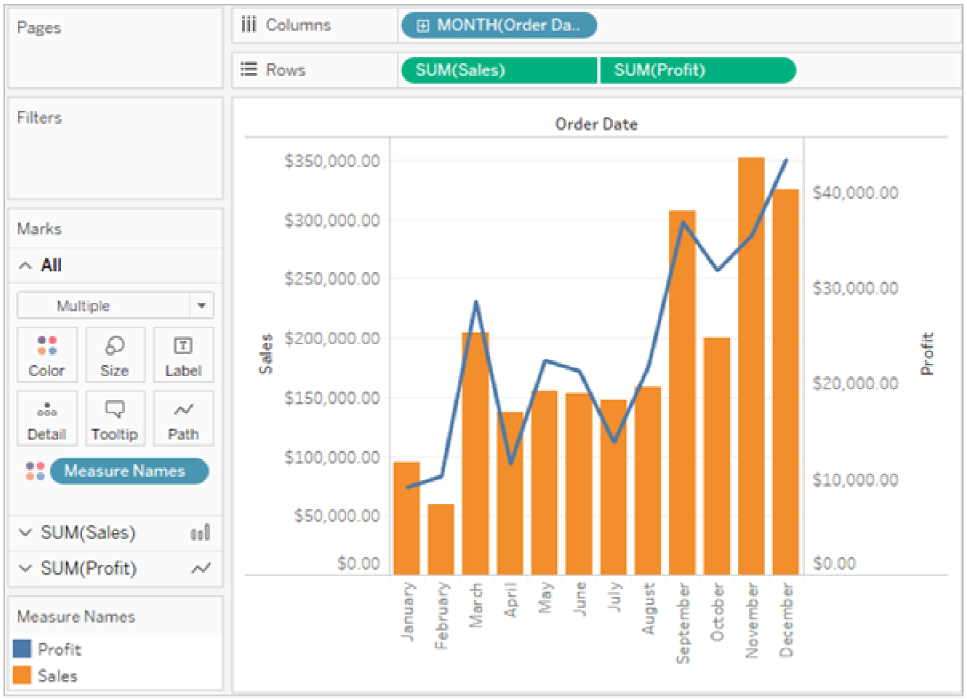


Tips

- Select a field in the Marks card to modify only its properties.
- Select the **All** Marks card to modify properties for all measures at once.
- To change the order of a field's marks in the view, right-click its axis, and then select **Move marks to front**.

Create a combo chart (assign different mark types to measures)

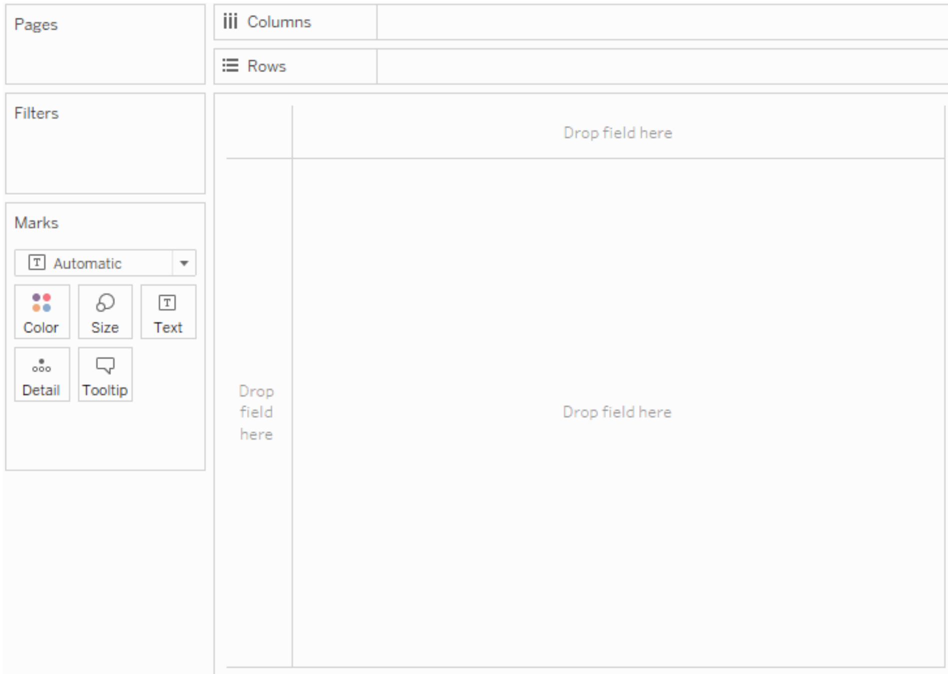
When working with multiple measures in a view, you can customize the mark type for each distinct measure. Because each measure can have customized marks, you can customize the level of detail, size, shape, and color encoding for each measure too.



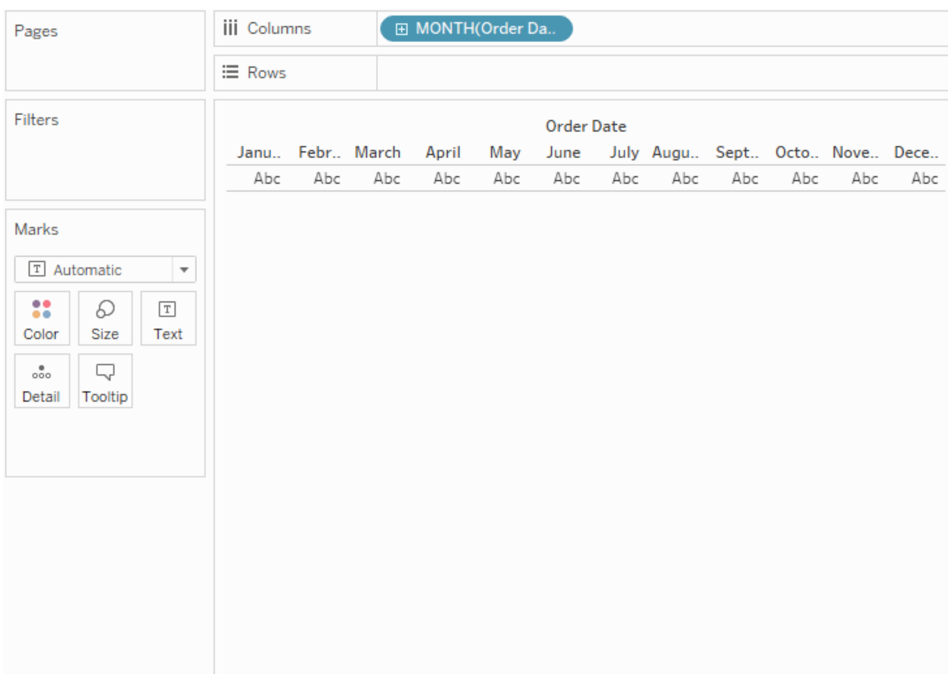
For example, you could create a view that uses two measures in dual axes. One measure shows Profit with a line mark, and the other measure shows Sales with bar marks. You could also choose to display the measures as individual axes or blended axes.

To create a combo chart like this example

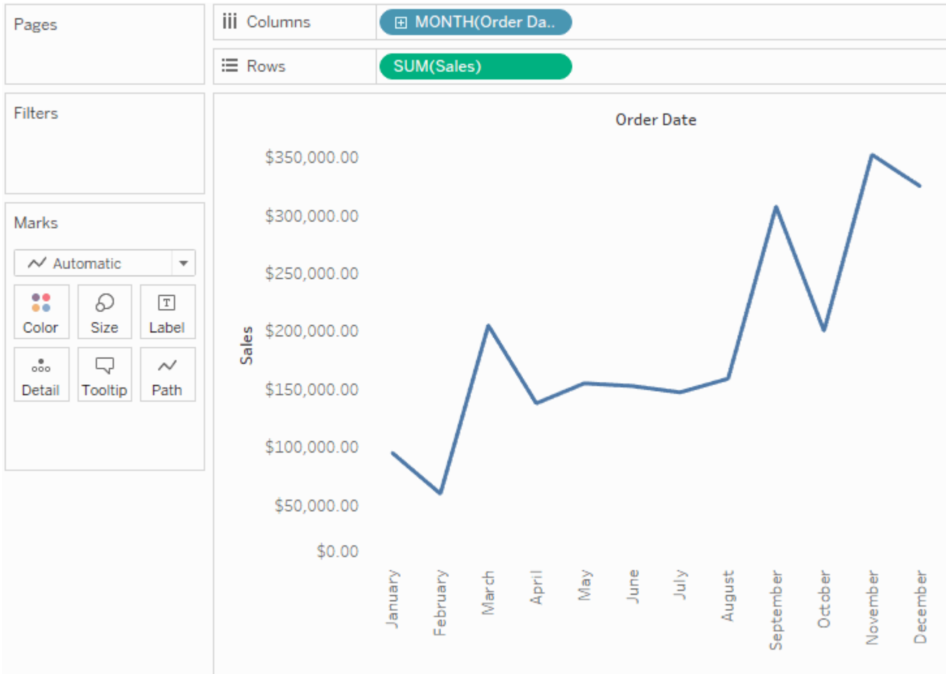
1. Connect to the Sample - Superstore data source.
2. Drag the Order Date measure to the Columns shelf, and then click Order Date change Year to Month.



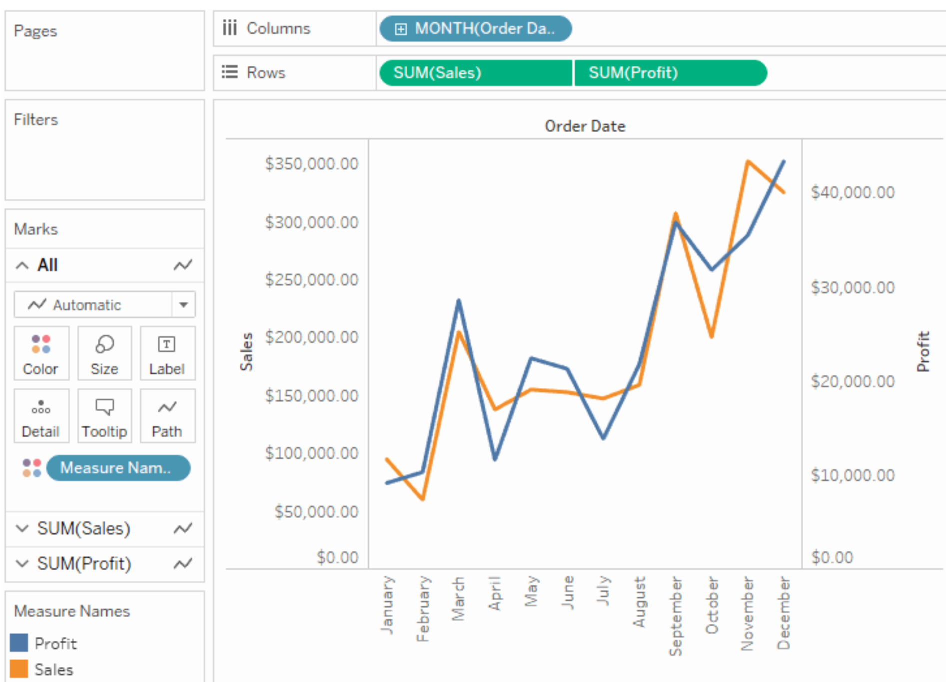
3. Drag the Sales measure to the Rows shelf.



4. Drag the Profit measure to the right side of the view and drop it when you see a black dashed line appear.



5. Click the Sales card. Change the Sales mark type to **Bar**.



Optional: To adjust the colors used for the Sales and Profit marks:

1. Click the **All** card, click **Color**, and then click **Edit Colors**.
2. Click the Profit data item, and then click a different color in the palette. Click the Sales data item, and then click a different color in the palette.
3. Click **Apply**, and then click **OK**.

Measure Values and Measure Names

The Data pane always contains a number of fields that do not come from your original data, two of which are **Measure Values** and **Measure Names**. Tableau automatically creates these fields so that you can build certain types of views that involve multiple measures.

- The **Measure Values** field contains all the measures in your data, collected into a single field with continuous values. Drag individual measure fields out of the Measure Values card to remove them from the view.
- The **Measure Names** field contains the names of all measures in your data, collected into a single field with discrete values.

Watch a video: To see related concepts demonstrated in Tableau, watch Measure Names and Measure Values, a 5-minute free training video. Use your tableau.com account to sign in.

How you can use Measure Names and Measure Values

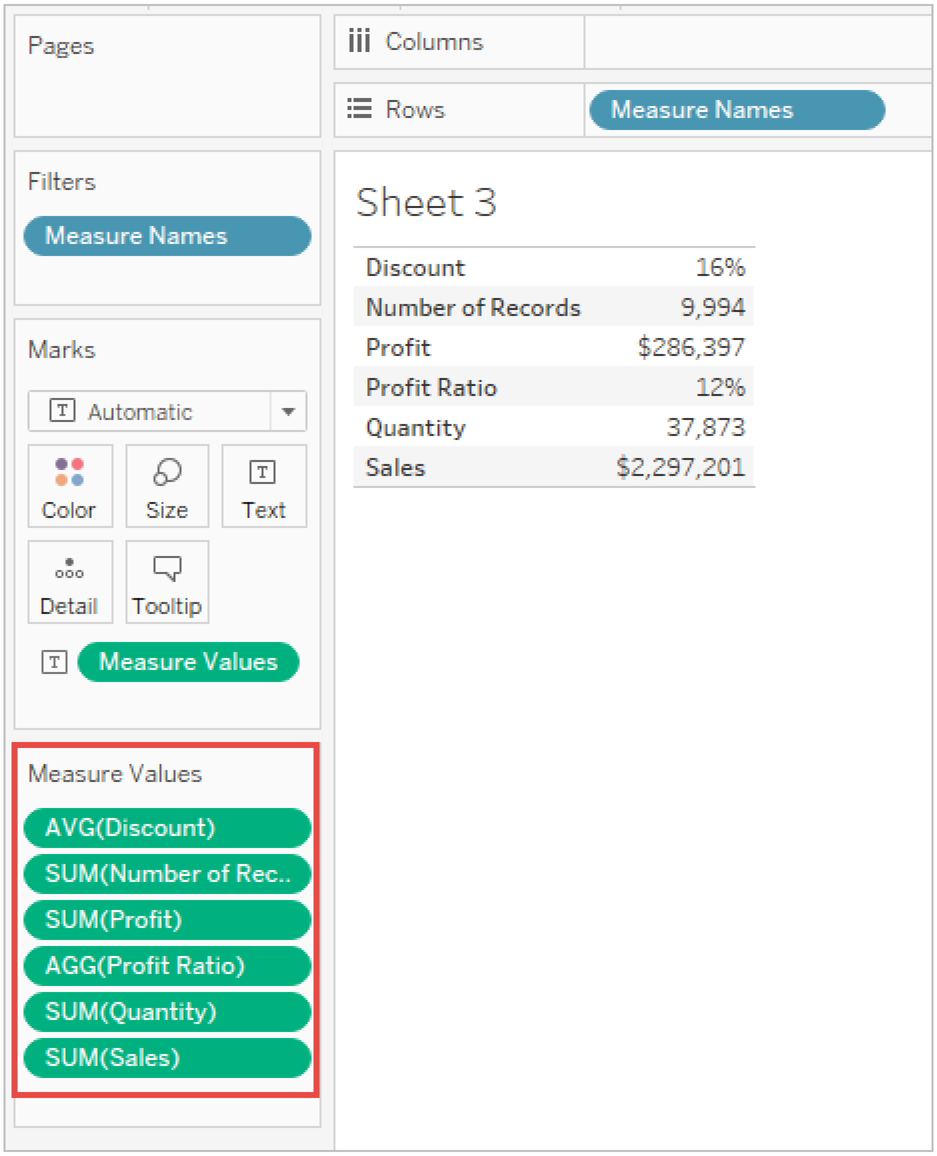
Tableau automatically creates these fields so that you can build certain types of views that involve multiple measures.

Measure Values and **Measure Names** are in most respects like other fields in Tableau:

- You can display a filter for **Measure Names**.
- You can format **Measure Values**—the formatting is then inherited as the default formatting for all measures.
- You can drop **Measure Values** and **Measure Names** on shelves.
- You can assign aliases to the values for **Measure Names**.
- You can change the manual sort order for the values for **Measure Names**.

The Measure Values card

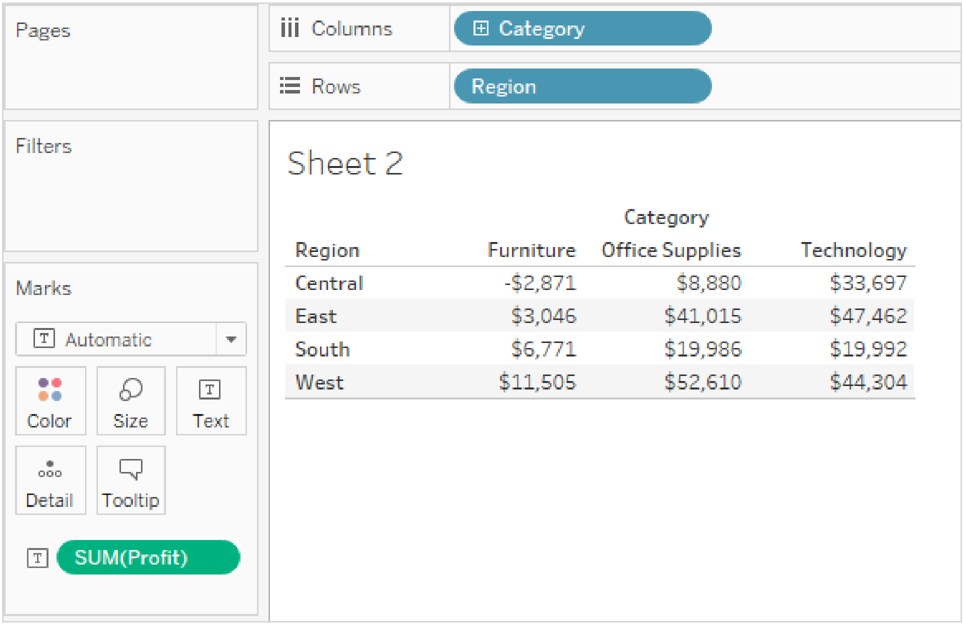
When **Measure Values** is in the view, Tableau creates a **Measure Values** card, outlined in red, below, that lists the measures in the data source with their default aggregations.



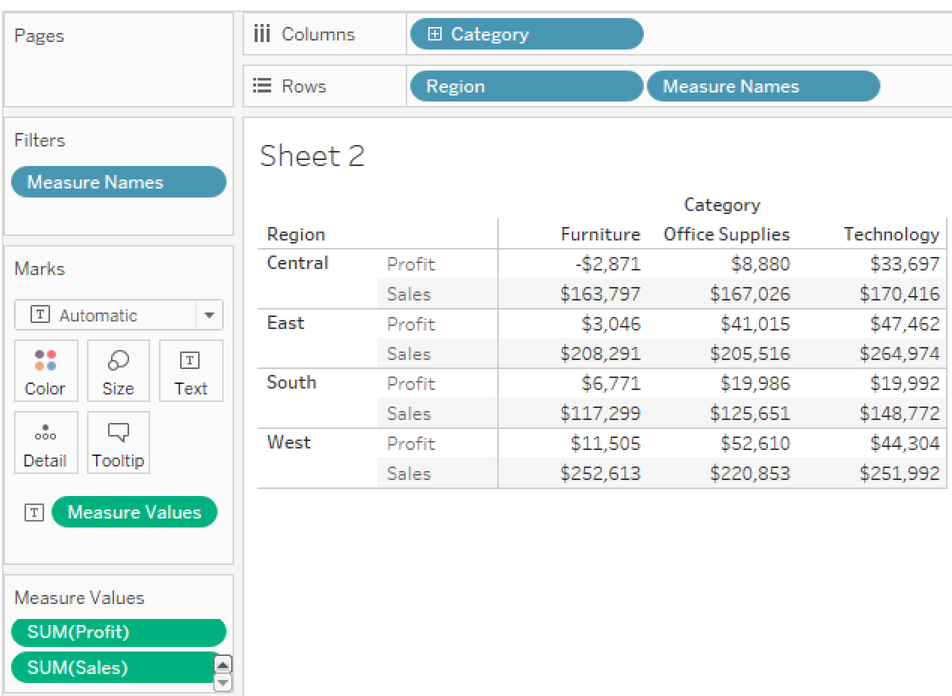
You can remove individual measures from the view by dragging them out of the **Measure Values** card.

About Measure Names

When you want to show multiple measures in a view, you can use the **Measure Values** and the **Measure Names** fields. When you add **Measure Names** to a view, all of the measure names appear as row or column headers in the view. The headers include each measure name. This feature becomes useful when you are working with a text table that shows multiple measures. For example, suppose you have a text table containing the aggregated profit of each product category by region.



Now suppose you want to show both the **Profit** and the **Sales** for each category and region. When you add **Sales** to the text table (by dragging it and dropping it in the view), the measures are combined and the **Measure Values** field is added to Text. The **Measure Names** field is automatically added to the **Rows** shelf.



Notice how the header names removed the aggregation label by default. You may want to include the aggregations or call it "Total Profit" and "Total Sales." To change the measure names, right-click (control-click on Mac) the Measure Names field on the Rows shelf and select **Edit Aliases**. Make the changes and click **OK**.

		Category		
Region		Furniture	Office Supplies	Technology
Central	Total Profit	-$2,871	$8,880	$33,697
	Total Sales	$163,797	$167,026	$170,416
East	Total Profit	$3,046	$41,015	$47,462
	Total Sales	$208,291	$205,516	$264,974
South	Total Profit	$6,771	$19,986	$19,992
	Total Sales	$117,299	$125,651	$148,772
West	Total Profit	$11,505	$52,610	$44,304
	Total Sales	$252,613	$220,853	$251,992

Create a visualization using Measures Names and Measure Values

You can use **Measure Values** and **Measure Names** to display the values for all measures in you data source, using their default aggregations, simultaneously. The view below was created with the Sample - Superstore data source. It shows the values of all measures in the Data pane, using their default aggregations.

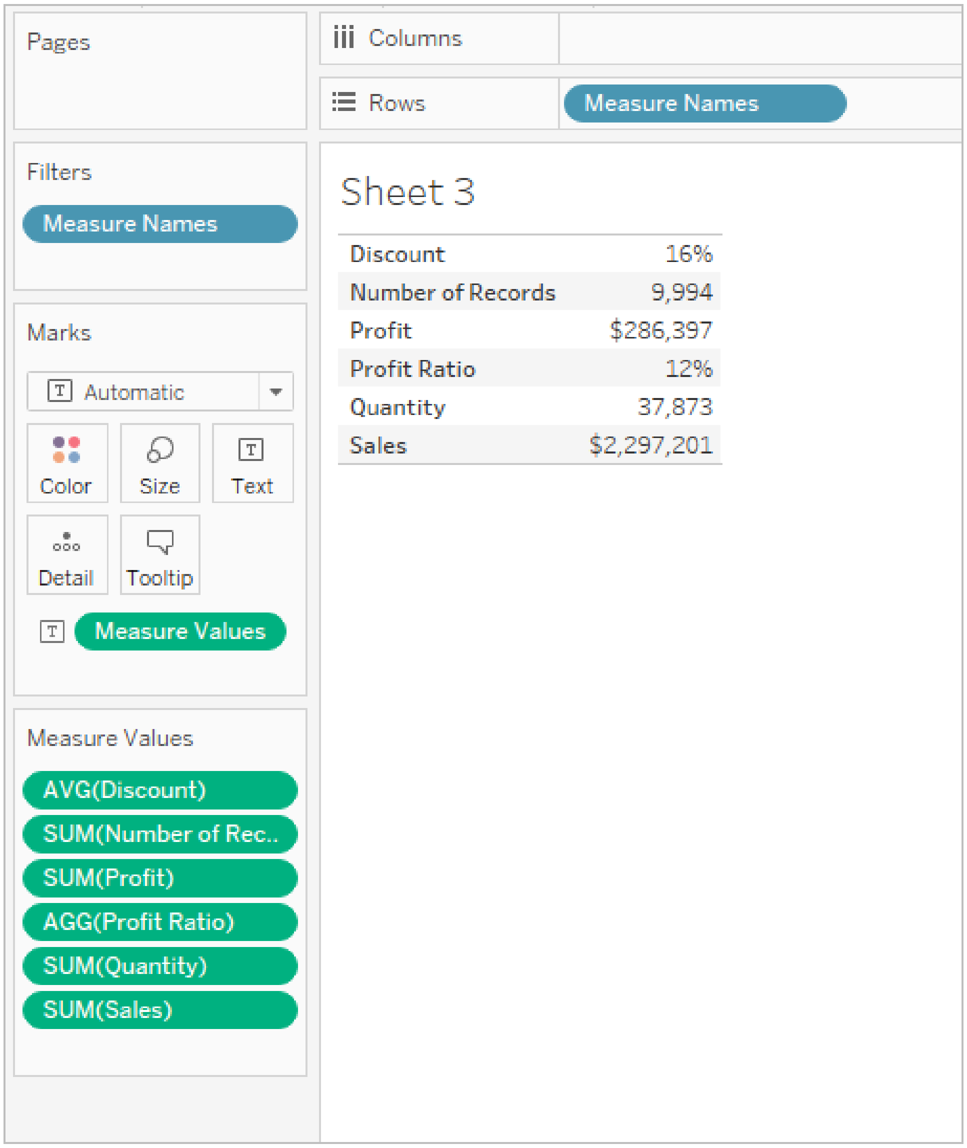
To create this view

Create a new worksheet using the Superstore example workbook.

Drag **Measure Values** to Rows

Click Text Table in Show Me.

Text Table from Show Me.



Of course, this view is not well suited for any more than a quick survey of your data because you can't generate a great deal of insight by comparing numbers that aren't measuring the same thing—**Profit** and **Sales** are both in dollars, but **Discount** and **Profit** Ratio are percentages.

When Measure Names and Measure Values automatically get added to the view

Tableau will automatically add **Measure Names** and **Measure Values** fields to the view in response to certain actions on your part.

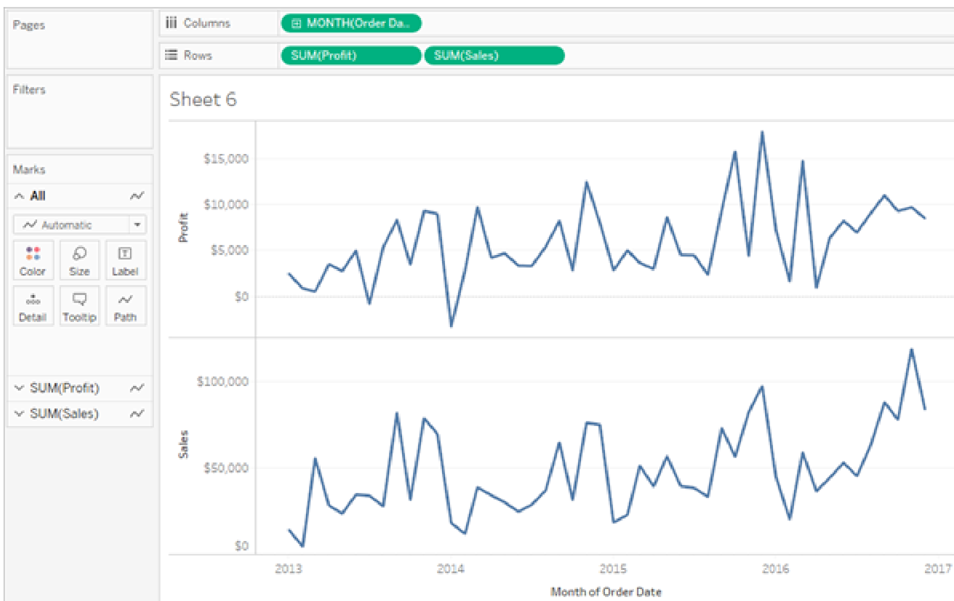
Blending Axes

You can show multiple measures on the same continuous axis by blending axes. For details, see **Blend axes for multiple measures into a single axis** on page 1300. When you use this technique, Tableau automatically adds Measure Names and Measure Values to the view, allowing you add additional measures to the blended axis.

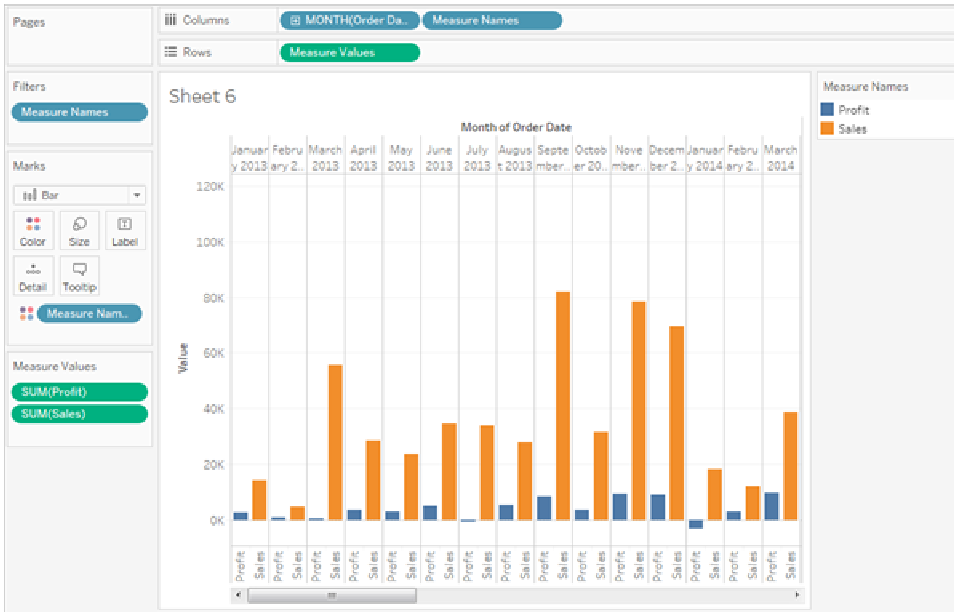
Using Show Me When Multiple Measures Are in the View

When there are multiple measures in the view, Tableau will automatically add **Measure Names** and **Measure Values**—or just **Measure Names**—when you choose certain visualization types from Show Me.

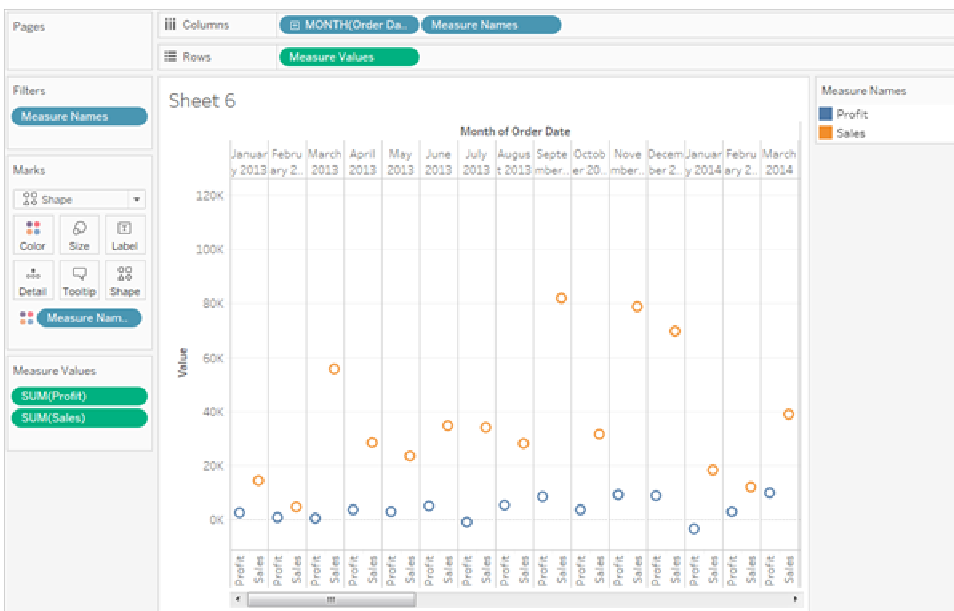
- When you choose side-by-side bars when there are multiple measures in the view, Tableau adds both **Measure Values** and **Measure Names**. For example, your original view might use lines to show **Sales** and **Profit** over time:



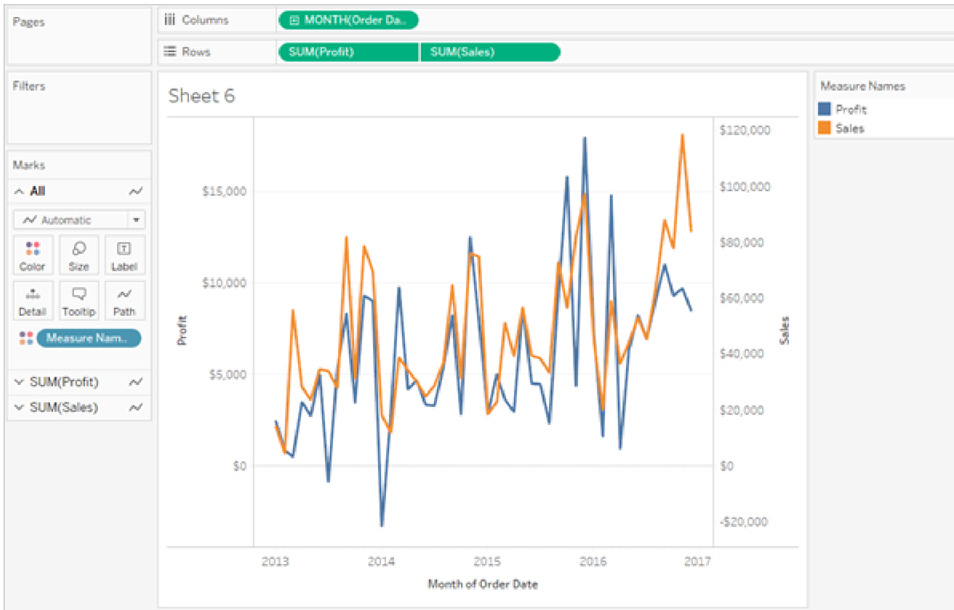
- If you choose side-by-side lines from Show Me, the view updates to include **Measure Names** and **Measure Values**:



- When you choose side-by-side circles from Show Me when there are multiple measures in the view, Tableau adds **Measure Values** and **Measure Names**:



- When you choose dual lines from Show Me when there are multiple measures in the view, Tableau adds **Measure Names** but not **Measure Values**:



This is similar to blending axes, except that Tableau uses separate value ranges (one on the left of the view and one on the right) to make the lines align as closely as possible. With blended axes, Tableau uses a single value range and the lines may or may not align well.

How Measure Names and Measure Values work with Legends Per Measure

When **Measure Values** and **Measure Names** are in the view, you can color code data for each individual measure. For details, see **Measure values and color legends** on page 257.

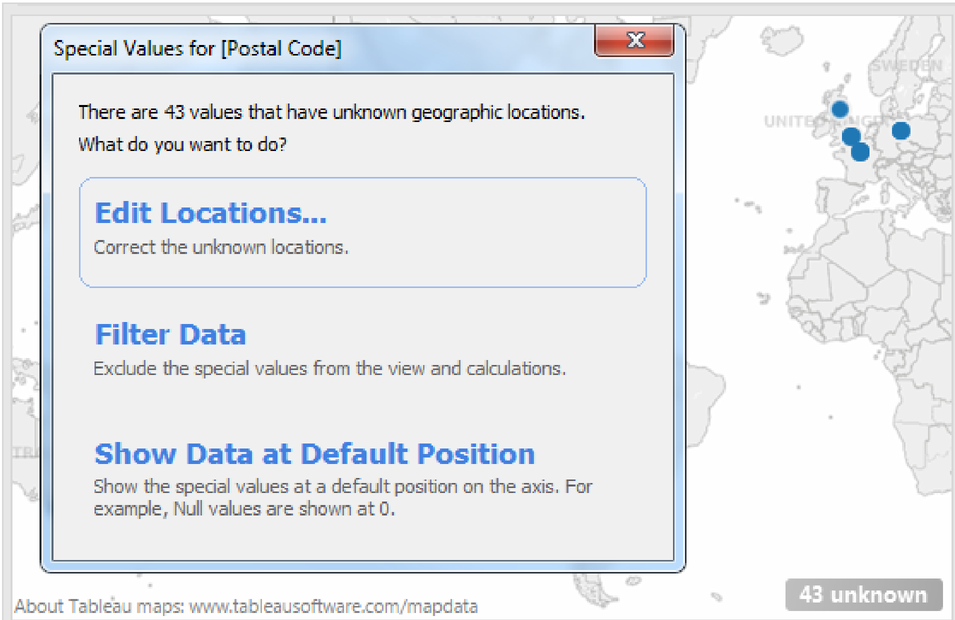
Handling Null and Other Special Values

Some data requires special handling in Tableau. Specifically:

- null values
- unrecognized or ambiguous geographic locations
- negative or zero values when working with a logarithmic scale
- negative or zero values when working with treemaps

Special values indicator

When your data contains any of these special values, Tableau cannot plot them in the view. Instead, it displays an indicator in the lower right corner of the view. Click the indicator to see options for how to handle these values.



If you don't know how to handle the values, you can choose to leave the special values indicator. Generally, you should continue to show the indicator so that you know there is data that is not being shown in the view. However, to hide the indicator, right-click (control-click on a Mac) it and select **Hide Indicator**.

Null numbers and dates, and negative values on log axes

A null value is a field that is blank, and signifies missing or unknown values.

When you drag a measure or continuous date to the view, the values are shown along a continuous axis. If the field contains null values, or if there are zeroes or negative values on a logarithmic axis, Tableau cannot plot them. When these values exist, Tableau displays an indicator in the lower right corner of the view that says unknown values exist. Click the indicator and choose from the following options:

- **Filter Data** - exclude the null values from the view using a filter. When you filter data, the null values are also excluded from any calculations used in the view.
- **Show Data at Default Position** - show the data at a default location on the axis. The null values will still be included in calculations. The default position depends on the data type. The table below defines the defaults.

Type of Data	Default Position
Numbers	0

Type of Data	Default Position
Dates	12/31/1899
Negative Values on a Log Axis	1
Unknown Geographic Location	(0,0)

Hiding null values

- To filter null dimensions or discrete measures, drag the pill to the Filter shelf and deselect **Null**. The null value will appear in the list with discrete values, where you can then remove it.
- When a measure contains null values, they are usually plotted in a view as zero. To see other options, including how to suppress these null values altogether, see **Format null values** on page 2955.

Note: See **Logical Functions** on page 2186 for functions, such as ISNULL() and IFNULL(), which you can use to handle null values with a calculated field.

Unknown geographic locations

When working with maps and geographic fields, unknown or ambiguous locations are identified by the indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Edit Locations** - correct the locations by mapping your data to known locations.
- **Filter Data** - exclude the unknown locations from the view using a filter. The locations will not be included in calculations.
- **Show Data at Default Position** - show the values at the default position of (0, 0) on the map.

Zero and negative values in treemaps

When working with treemaps, any null or zero values display in the indicator in the lower right corner of the view. Click the indicator and choose from the following options:

- **Filter Data** - exclude the unknown locations from the view using a filter. The locations will not be included in calculations.

- **Use Absolute Values** - use the absolute value to determine the size of the corresponding area in the view. For example, both values of 5 and -5 are shown as the same size.

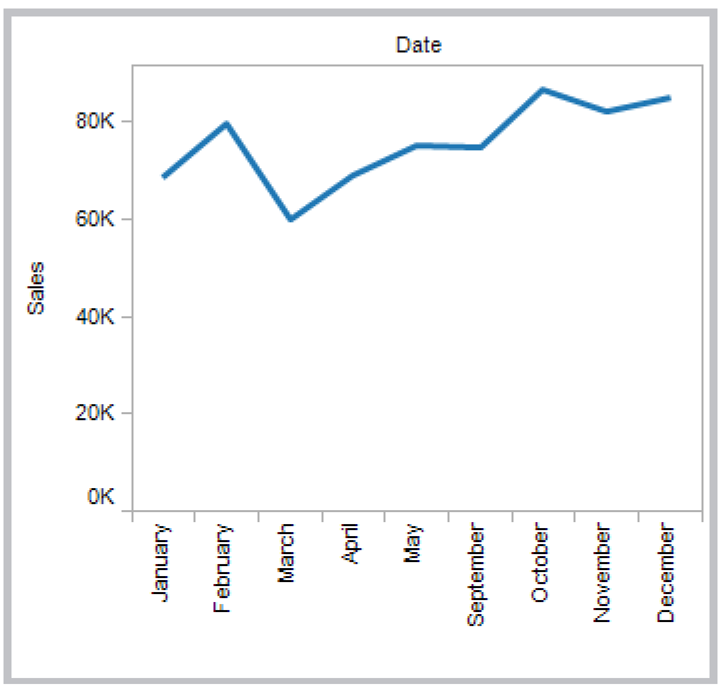
Show or Hide Missing Values or Empty Rows and Columns

When you’re working with dates or numeric bins, Tableau only shows the values that are represented in your data. If your data does not contain the complete range of values, the missing values will not be shown. For example, your data may contain data for January through May and September through December. However, there was no data recorded for June, July, and August. If you create a line chart in Tableau, the missing months will not be shown. You can optionally show the missing months to make it clear that there was no data recorded during that time.

> **Note**: You can replace missing values with a calculated field using the ZN() function. See **Number Functions** on page 2147 for details.

Missing Values Hidden (Default)

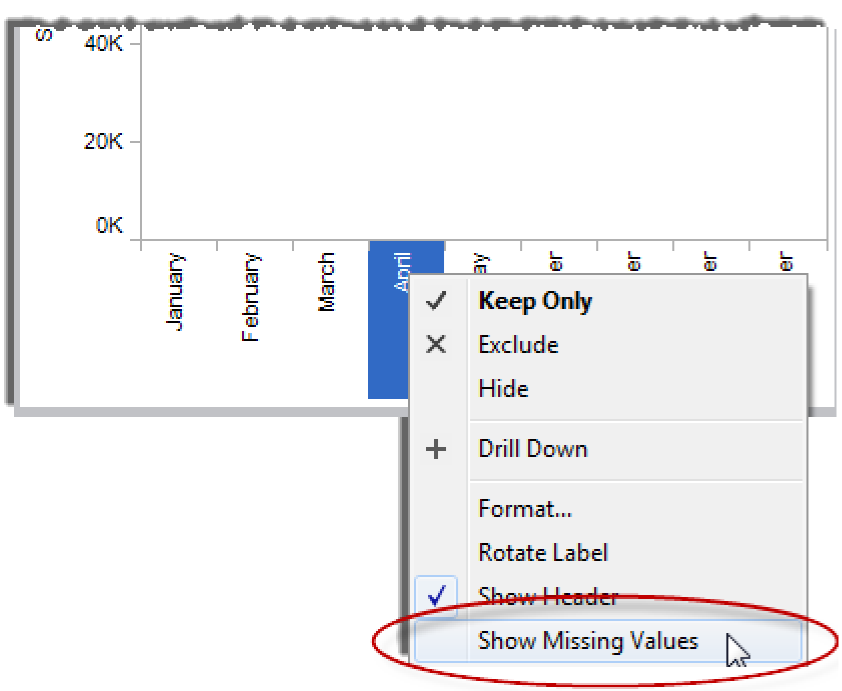
By default, missing values in a date range or numeric bins are not shown.



Missing Values Shown

You can show the missing values to indicate incomplete data.

To show missing values in a range, right-click (control-click on Mac) the date or bin headers and select **Show Missing Values**.

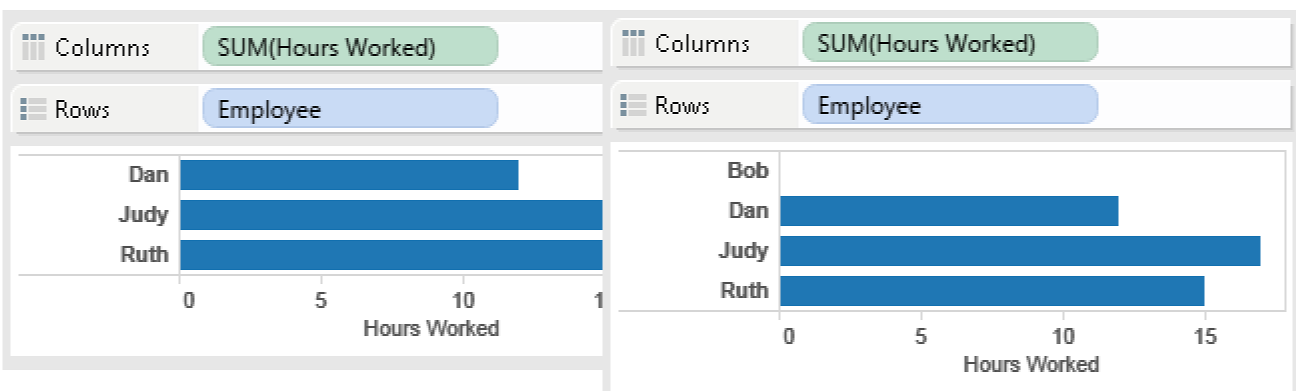


> **Note**: You can also perform calculations on missing values that are shown in the view. To do this, open the **Analysis** menu at the top, and then select **Infer Properties from Missing Values**. For an example of this, see **Predictive Modeling with Generated Marks** on page 2654.

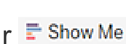
Show and Hide Empty Rows and Columns

When you are working with fields that are not dates or numeric bins, Tableau hides missing values by default. For example, you may be showing workers and the number of hours worked in a particular month. If a worker didn't work at all that month, there may not be a row in the database for that worker. When you drag the Worker field to the Rows shelf, the workers that didn't work are hidden by default. You can show the empty rows by selecting **Analysis** > **Table Layout** > **Show Empty Rows**. Similarly, show the empty columns by selecting **Analysis** > **Table Layout** > **Show Empty Columns**.

Empty Rows Hidden (default)	**Empty Rows Shown**
Bob did not work in January so there are no records in the database for him. By default he is not listed.	Even though Bob did not work in January, he is listed but no marks are drawn.



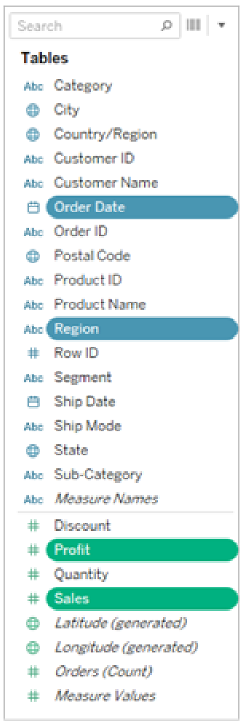
Use Show Me to Start a View

Show Me creates a view based on the fields already used in the view and any fields you've selected in the Data pane. Open Show Me by clicking **Show Me** on the toolbar 
.

Start a view based on fields that you select

To use Show Me, select the fields you want to analyze in the Data pane, and then select the type of view you want to create in the Show Me pane. Tableau automatically evaluates the selected fields and gives you the option of several types of views that would be appropriate for those fields. Show Me also highlights the visualization type that best matches the data.

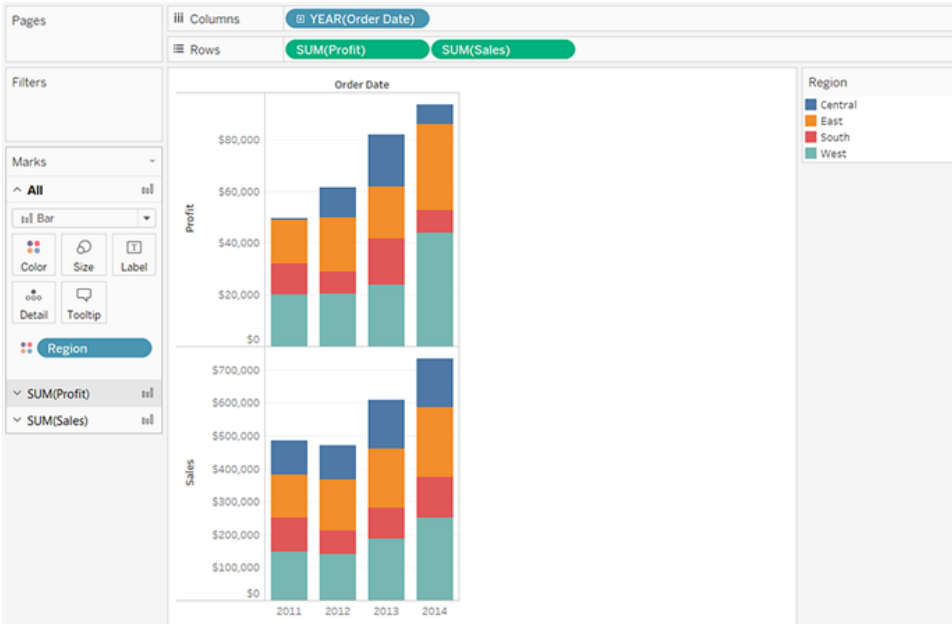
1. In the **Data** pane, select the fields that you want to analyze. Hold the Ctrl key (Command key on a Mac) to make multiple selections.



2. Click **Show Me** on the toolbar and then select the type of view you want to create.

Any view type that is not gray will generate a view of your data. As you hover over each view type, the description at the bottom shows the minimum requirements. In this example, Stacked Bars is selected.

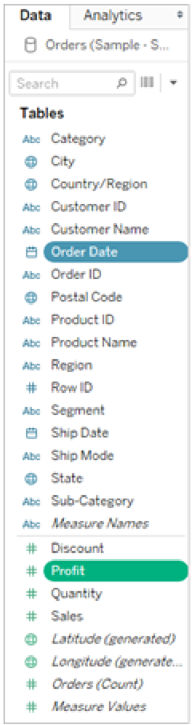
3. View the Result. Tableau automatically creates a view of the data.



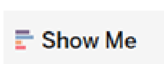
Create a Line Chart with Show Me

In this example, you use **Show Me** to create a line chart that displays profit as a function of time.

1. In the **Data** pane, select **[Order Date]** and **Profit**. Hold down the Ctrl key (or the command key on a Mac) as you select the fields.



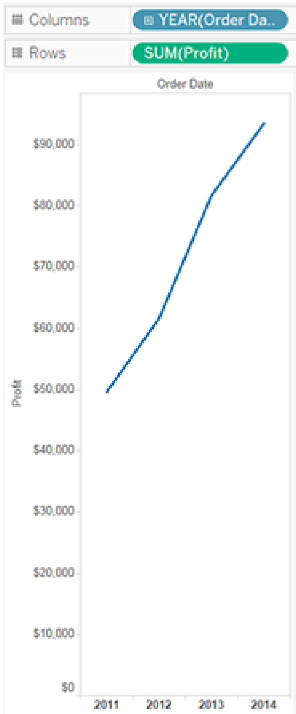
2. Click **Show Me** on the toolbar to display the **Show Me** card, if it isn't already displayed.



3. In the **Show Me** pane, see what type of view Tableau suggests.

Because a date dimension and a measure are selected, Tableau suggests you build a line view. A line is often the optimal way to look at how measure values change over time.

The view below shows the resulting chart. Each point on the line represents the sum of profit for the corresponding year.

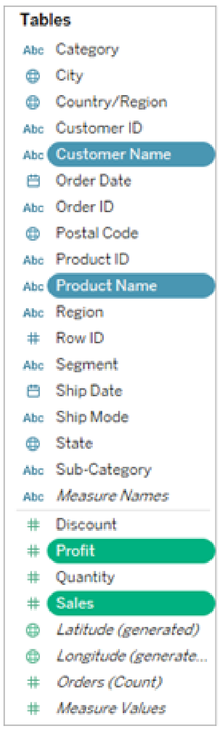


You can more easily see the points by clicking the **Show Mark Labels** button on the toolbar.

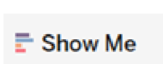
Create a Scatter Plot with Show Me

In this example you use **Show Me** to build a scatter plot to show sales versus profit for each product and customer.

1. In the **Data** pane, select the **Customer Name** and **Product Name** dimension fields, and the **Profit** and **Sales** measure fields. To select multiple fields, hold down the Ctrl key (or the command key on a Mac) as you click fields.

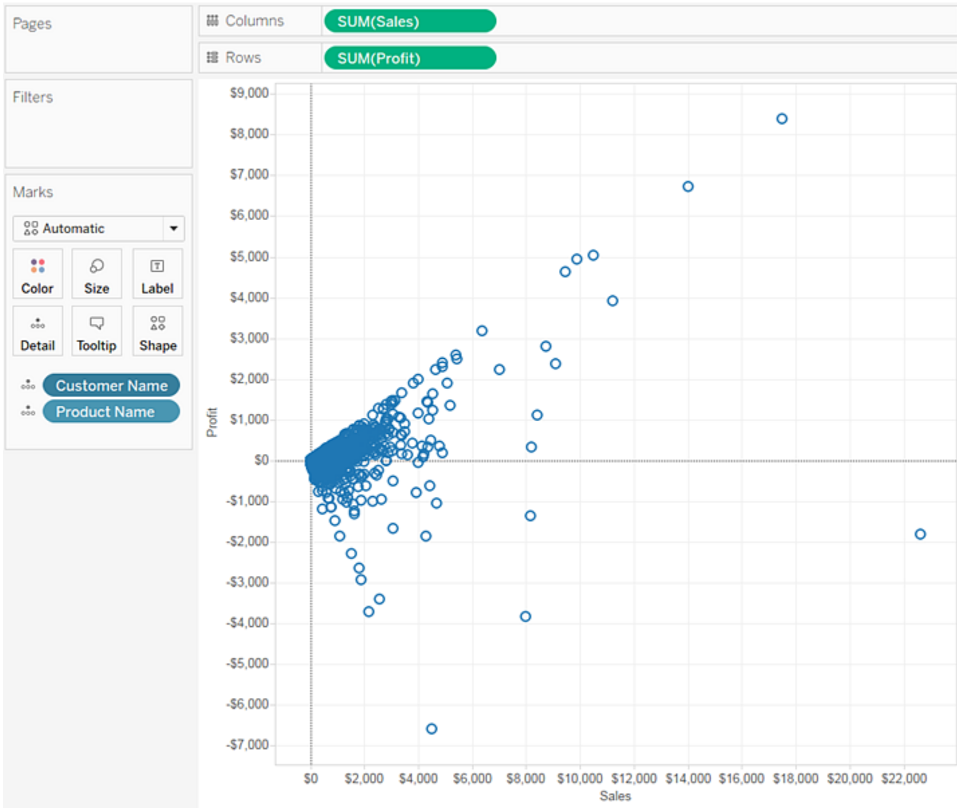


2. Click **Show Me** on the toolbar.



3. Select the scatter plot chart type from **Show Me**.

Tableau creates a scatter plot using the fields you selected. You can drag fields to further refine the view.



Add Detail

The Marks card is a key element for doing visual analysis in Tableau. When you drag measures and dimensions to the Marks card, you encode your data with color, size, shape, text, and other properties that add context, detail, and meaning to the marks in the view.

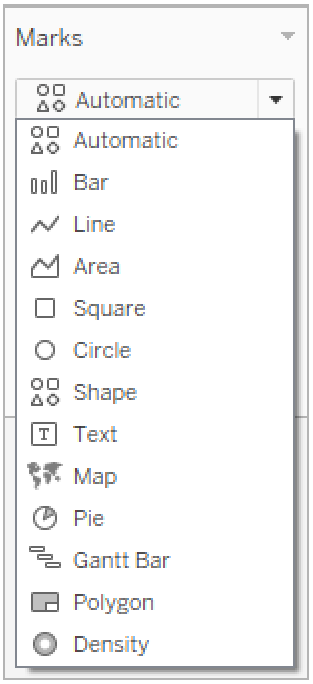
Change the Type of Mark in the View

You can change the type of marks displayed in the view to fit your analysis. For example, you can change the marks from bars to lines. This article describes how to change the type of marks used in the view, and explains the different types of marks available. For related information on marks, see **Control the Appearance of Marks in the View** on page 1356 and **Marks card** on page 227.

Change the mark type

To change the mark type:

- On the **Marks card** on page 227, click the Mark Type drop-down and select an option from the list.



For descriptions of each mark type, see the following sections:

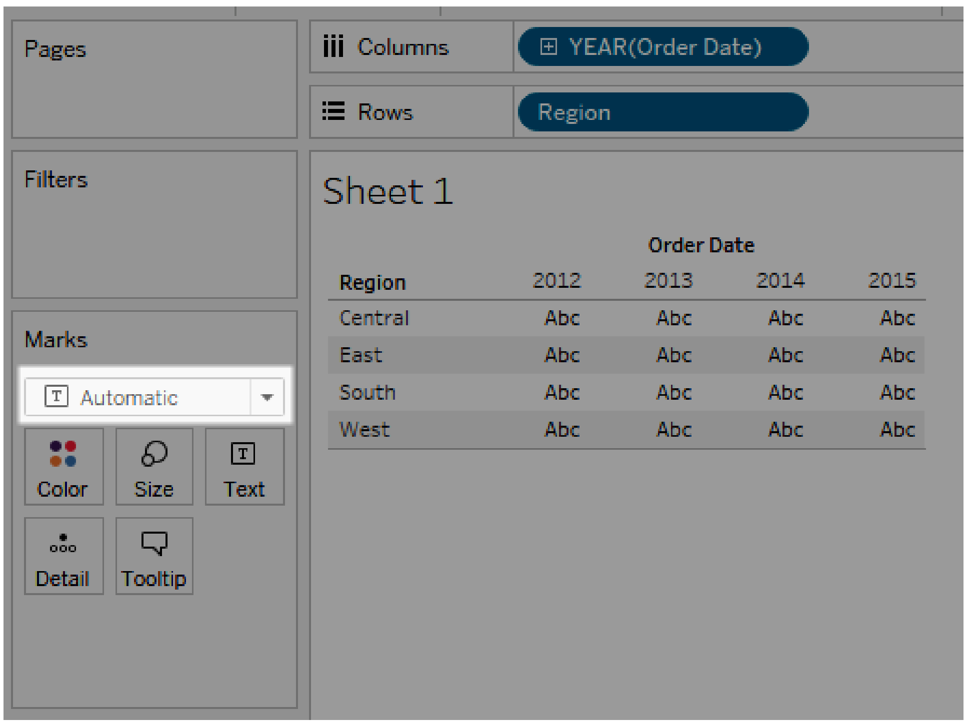
- Automatic mark
- Bar mark
- Line mark
- Area mark
- Square mark
- Circle mark
- Shape mark
- Text mark
- Map mark
- Pie mark
- Gantt Bar mark
- Polygon mark
- Density mark

Automatic mark

When the Marks card drop-down menu is set to **Automatic**, Tableau automatically selects the best mark type for your data view. The mark type that is automatically selected is determined by the inner fields on the **Rows** and **Columns** shelves. The icon in the Marks card drop-down menu indicates which type of mark was automatically selected. The mark types below are automatically selected for the following scenarios.

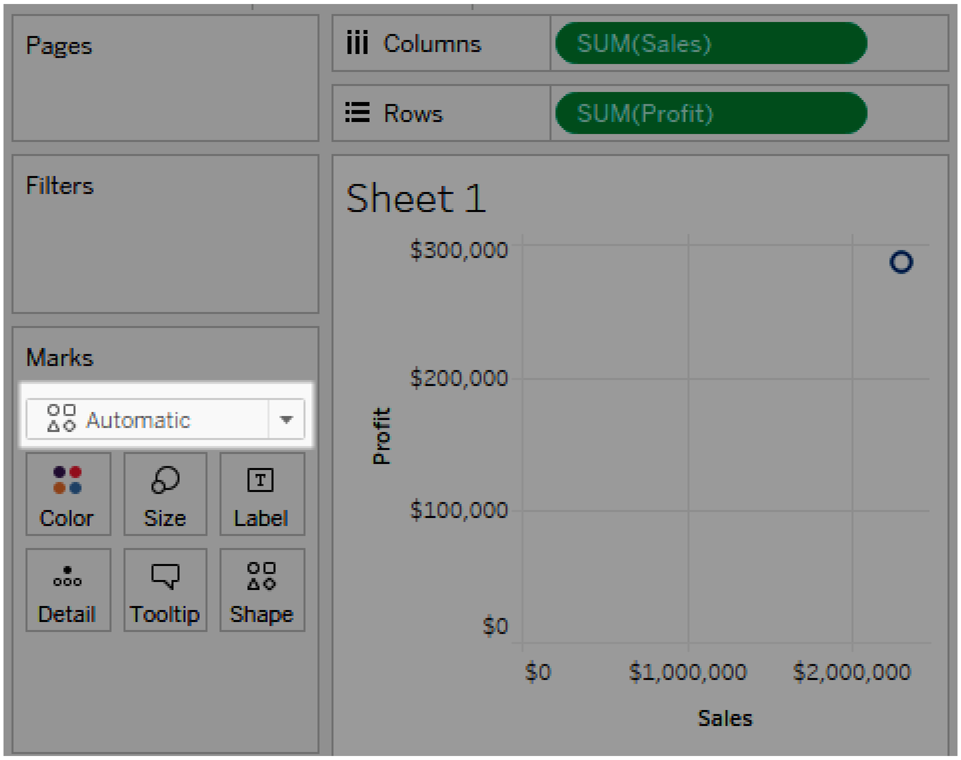
Text

The Text mark type is automatically selected when there are dimensions as inner fields on both the **Rows** and **Columns** shelves.



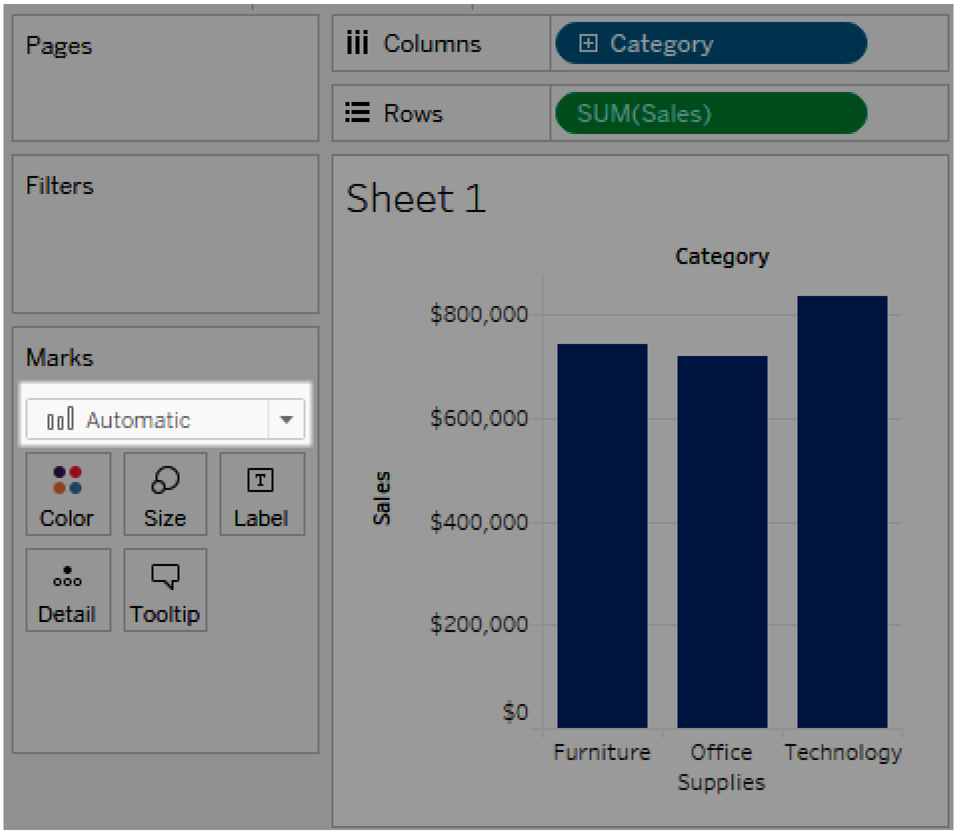
Shape

The Shape mark type is selected when there are measures as inner fields on both the **Rows** and **Columns** shelves.



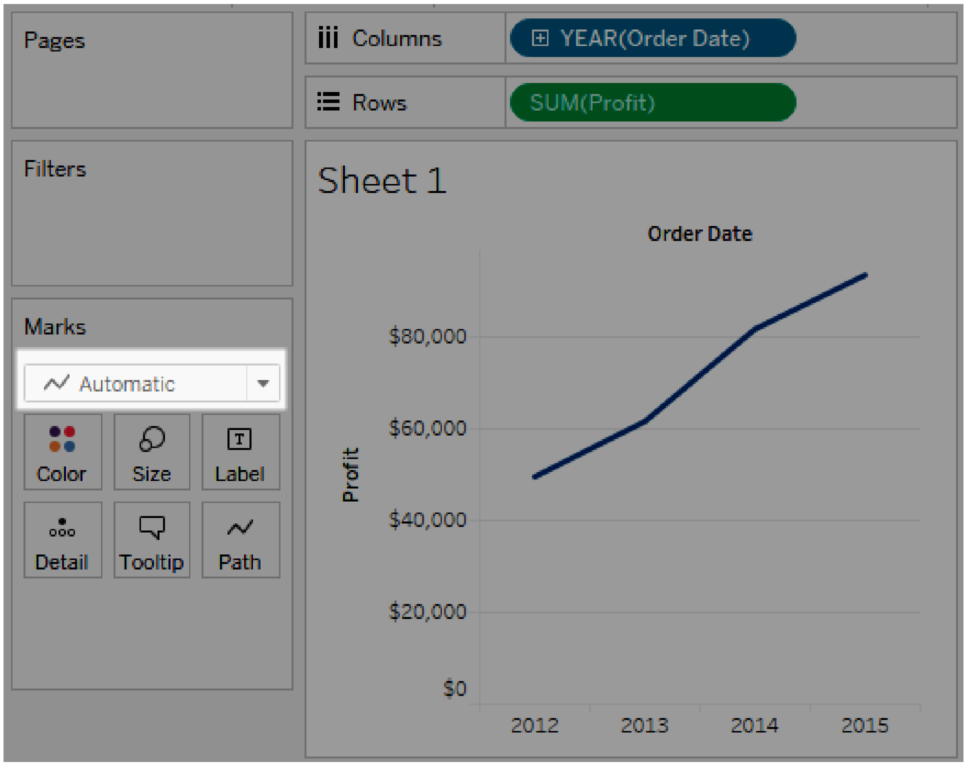
Bar

The Bar mark type is selected when there is a dimension and a measure as inner fields on **Rows** and **Columns** shelves.



Line

The Line mark type is selected when there is a date field and a measure as the inner fields on the **Rows** and **Columns** shelves.



Note: You can override the default selection and use any mark type that provides insight into your data. However, you should exercise some caution when manually selecting a mark type because the resulting view might hide important information about your data.

Bar mark

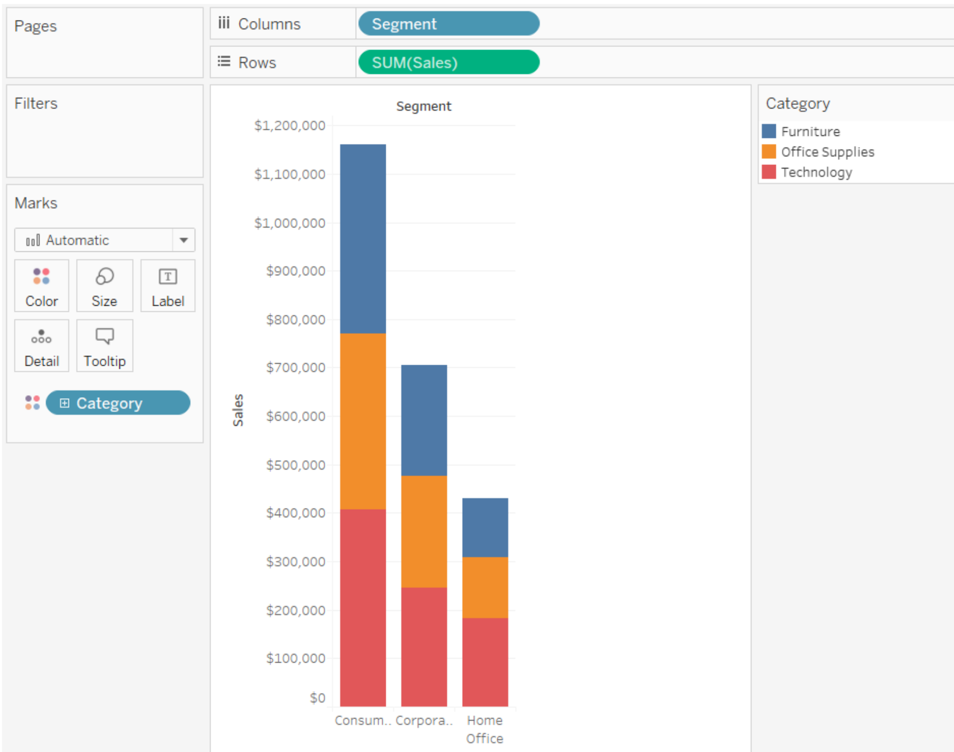
The Bar mark type is useful when you want to compare measures across categories, or when you want to break data down into stacked bars. Tableau displays your data using bars when:

- The Marks card drop-down menu is set to **Automatic**, and you place a dimension and a measure as the inner fields on the **Rows** and **Columns** shelves. If the dimension is a date dimension, the Line mark is used instead.
- You select **Bar** from the Marks card drop-down menu.

Marks are automatically stacked.

Note: For views where the mark type is Bar and there are continuous (green) fields on both **Rows** and **Columns**, Tableau supports additional options and defaults for sizing the bar marks on the axis where the bars are anchored. See the Continuous axis mark sizing section in **Change the size of marks** on page 1358 for details.

The data view shown below displays a dimension as the columns of the table, and a measure as the rows of the table. It is also color-encoded by a dimension (**Category**). Because the Marks card drop-down menu is set to **Automatic**, data is displayed using bars.



To try some hands-on exercises for building bar charts, see **Build a Bar Chart** on page 1579.

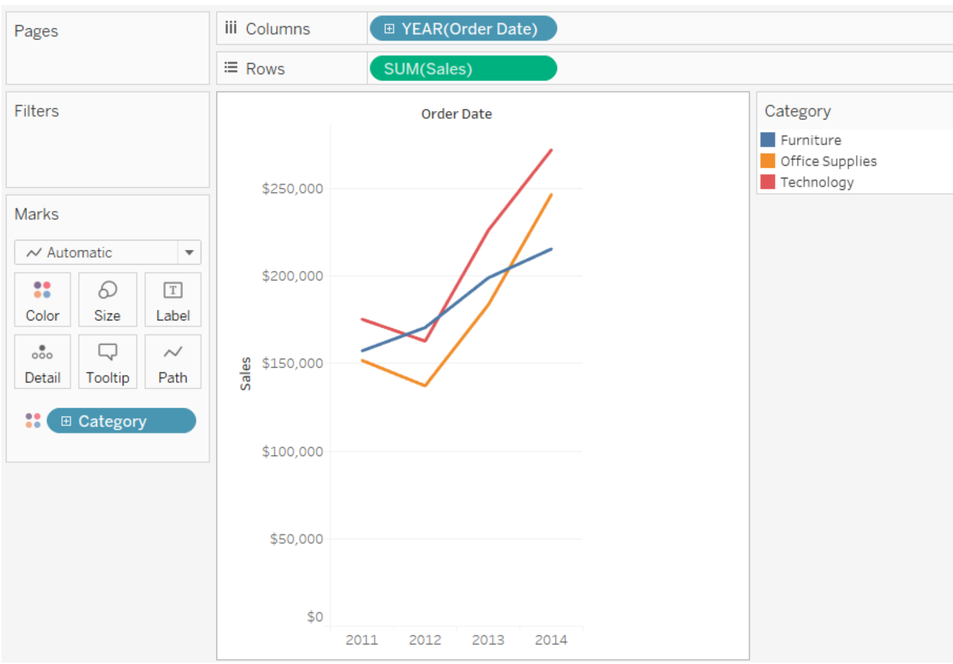
Line mark

The line mark type is useful when you want to see trends in data over time, your data are ordered, or interpolation makes sense. Tableau displays data using lines when:

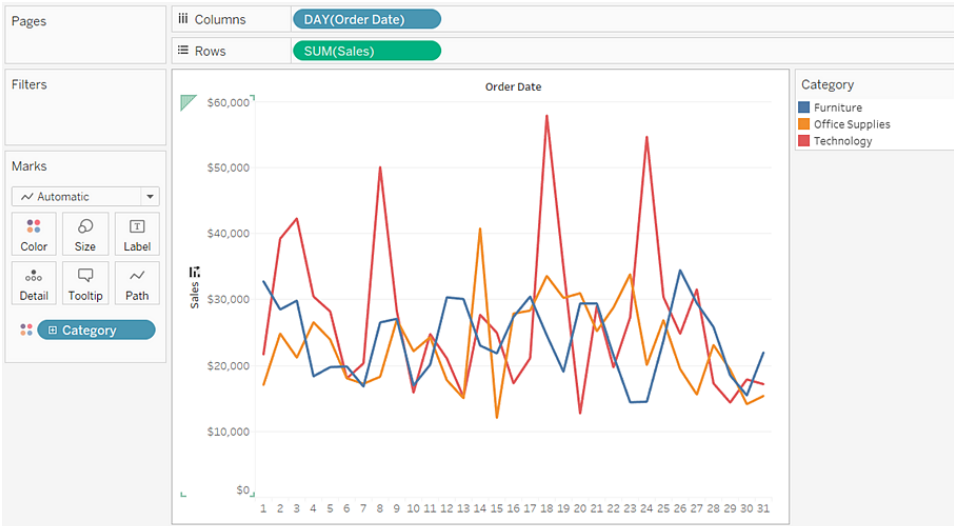
- The Marks card drop-down menu is set to **Automatic**, and you place one or more measures on either the **Columns** shelf or the **Rows** shelf, and then plot the measures against a date dimension or a continuous dimension.
- You select **Line** from the Marks card drop-down menu.

Note: In views that use the line mark type, you can use the **Path** property in the Marks card to change the type of line mark (linear, step, or jump), or to encode data by connecting marks using a particular drawing order. For details, see Path properties in the **Control the Appearance of Marks in the View** on page 1356

The data view shown below displays a date dimension as the columns of the table and a measure as the rows of the table.



As the density of data increases, trends are often easier to see when using lines. This view shows 93 data points.



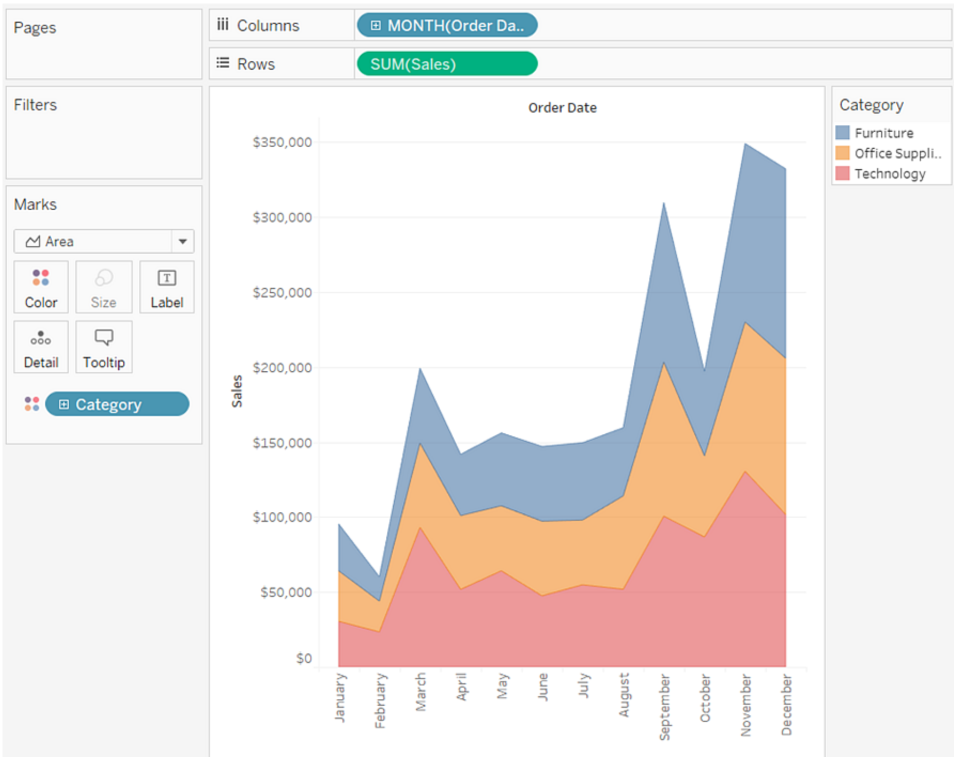
To try some hands-on exercises for building line charts, see **Building Line Charts** on page 1631.

Area mark

The Area mark type is useful when you have a view where the marks are stacked and do not overlap. For more information about stacked marks, see **Stack Marks** on page 1392. In an area chart, the space between each line and the next line is filled with a color. This type of chart is not the best way to show specific values along the line but it can clearly show the total values so you can get an idea of how a dimension is contributing to an overall trend. Tableau displays your data using areas when:

- The Marks card drop-down menu is set to **Automatic** and one or more measures is plotted against a date dimension or continuous dimension. The **Analysis** > **Stack Marks** option must also be selected.

The view below shows a date dimension on the **Columns** shelf and a measure on the **Rows** shelf. Each line represents the sales for a product category. The lines have been stacked so that they do not overlap.

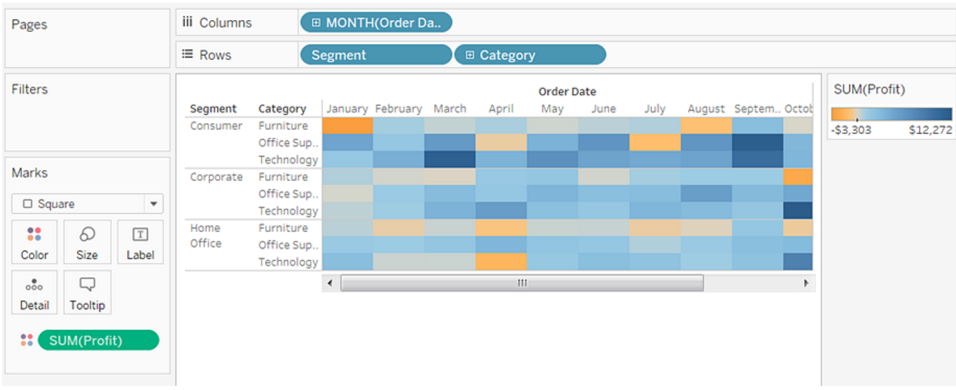


Note: You can add labels, edit the color legend, and highlight areas. In addition, you can turn on color effects, such as mark borders. For more information, see **Control the Appearance of Marks in the View** on page 1356.

Square mark

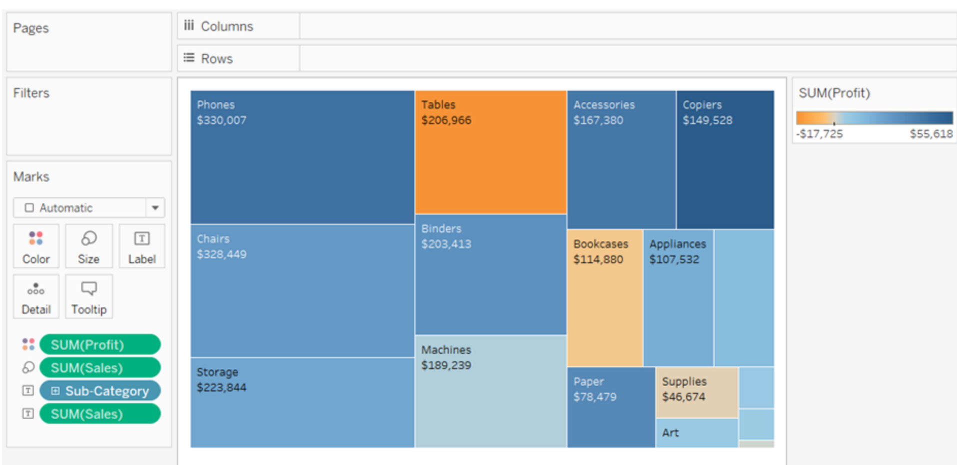
The square mark type is useful when you want to clearly see individual data points. When you select **Square** from the **Marks** card drop-down menu, Tableau displays your data using squares.

The view shown below displays several dimensions on both the rows and columns shelves. By default, the data is displayed using text. However, when you place a measure on **Color**, Tableau converts the marks to squares and creates a heat map. To try some hands-on exercises for building highlight tables, see **Build a Highlight Table or Heat Map** on page 1616.



When you add additional levels of detail by placing a dimension on **Detail**, **Color**, **Shape**, **Size**, or **Label** on the **Marks** card, the squares are displayed side-by-side and wrap to fill the cell. If the window is too small to show all of the squares, an ellipses indicates that there are more values than you can see.

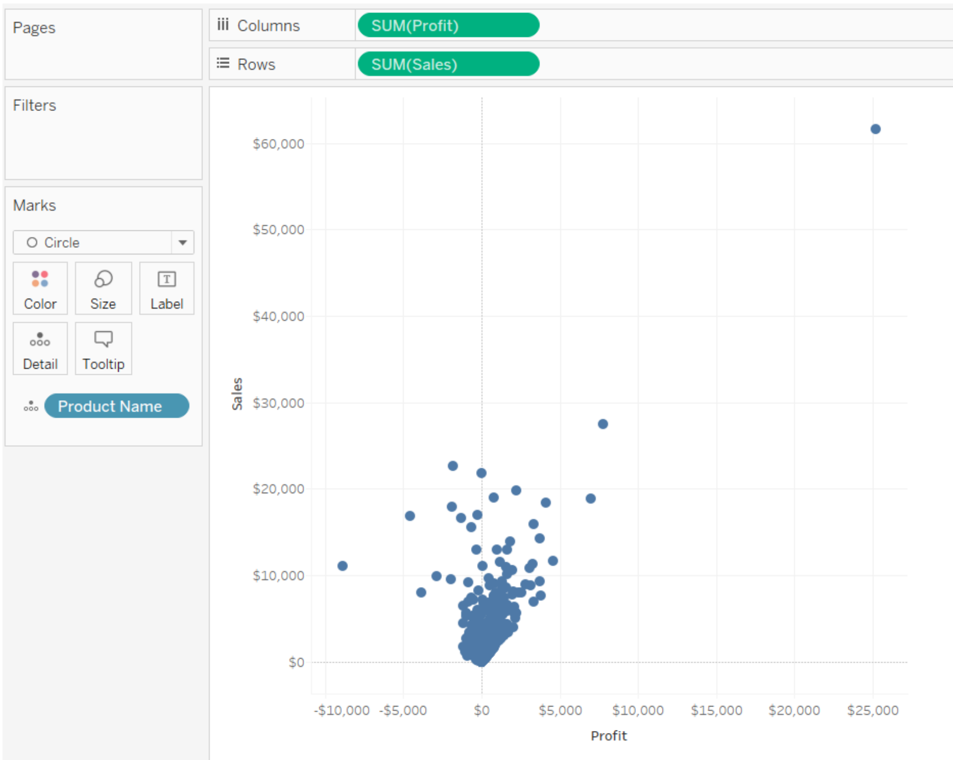
You can also use the square mark to create treemaps. A treemap displays hierarchical data as a set of nested rectangles. To try a hands-on exercise for building a treemap, see **Build a Treemap** on page 1664.



Circle mark

When you select **Circle** from the Marks card drop-down menu, Tableau displays your data using filled circles.

If the mark type was set to **Automatic**, Tableau would display the data using a shape (that is, an open circle).

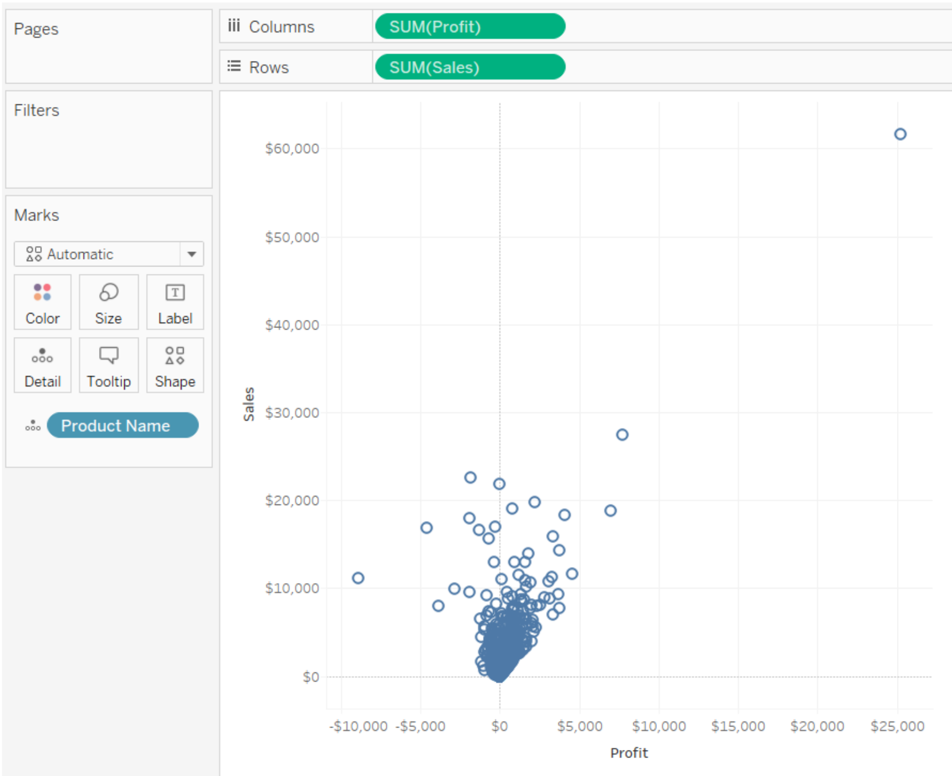


Shape mark

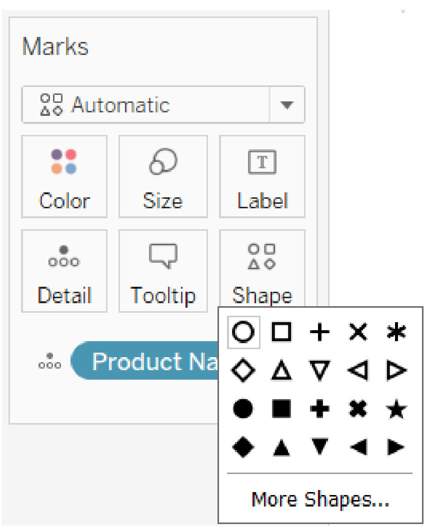
The shape mark type is useful when you want to clearly see individual data points while also viewing categories associated with those points. Tableau displays your data using a shape when:

- The Marks card drop-down menu is set to **Automatic**, and you place one or more measures on both the **Rows** and the **Columns** shelves.
- You select **Shape** from the Marks card drop-down menu.

The view shown below displays the data from two measures. Because the Marks card drop-down menu is set to **Automatic**, the data are displayed using a shape.

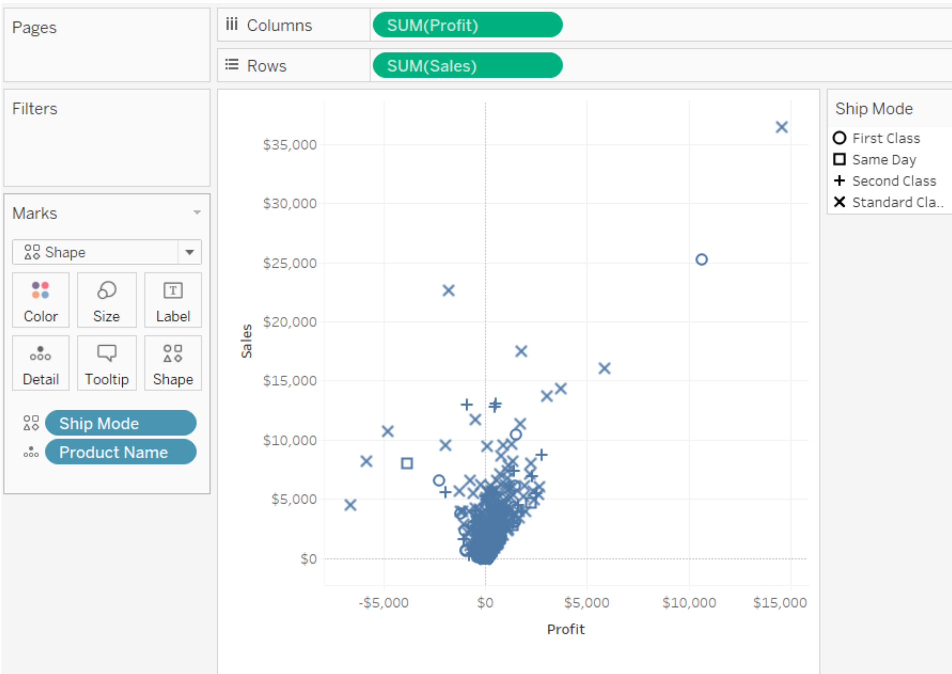


By default, the shape used is an open circle. To select a different shape, click **Shape** on the Marks card. Twenty unique shapes are available:



To add more data to the view, you can place a field on **Shape** on the Marks card. Tableau separates the marks according to the values in the field. If the field is a dimension, each member is assigned a unique shape. If the field is a measure, the measure is automatically binned into distinct buckets and each bucket is assigned a unique shape. The shape legend displays how shapes are distributed.

As shown below, the **Ship Mode** dimension is placed on **Shape** on the Marks card to encode each mark with information about how the order was shipped.



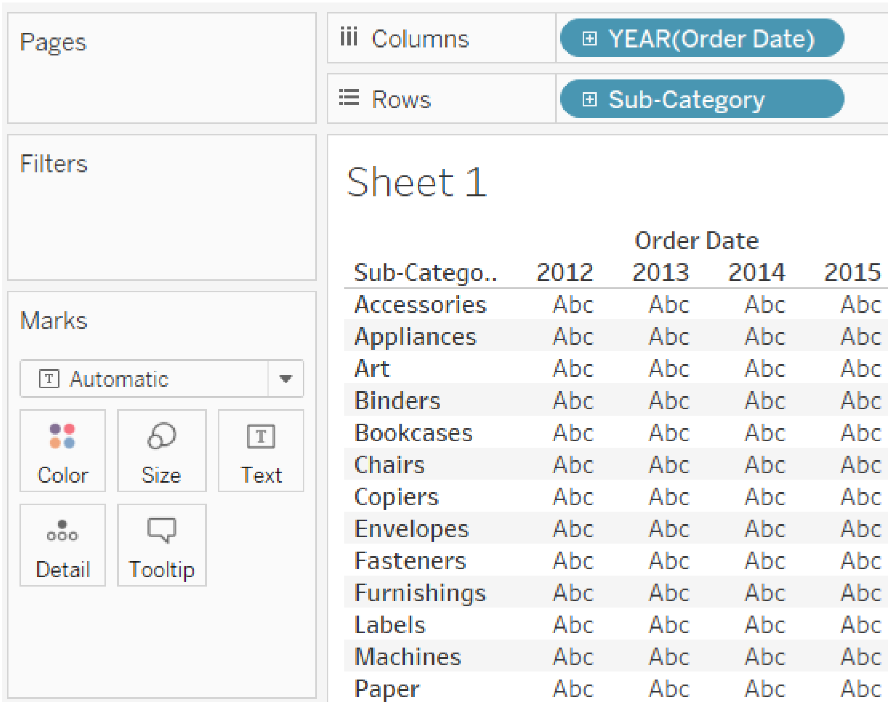
For details on how to build a scatterplot, see **Build a Scatter Plot** on page 1651.

Text mark

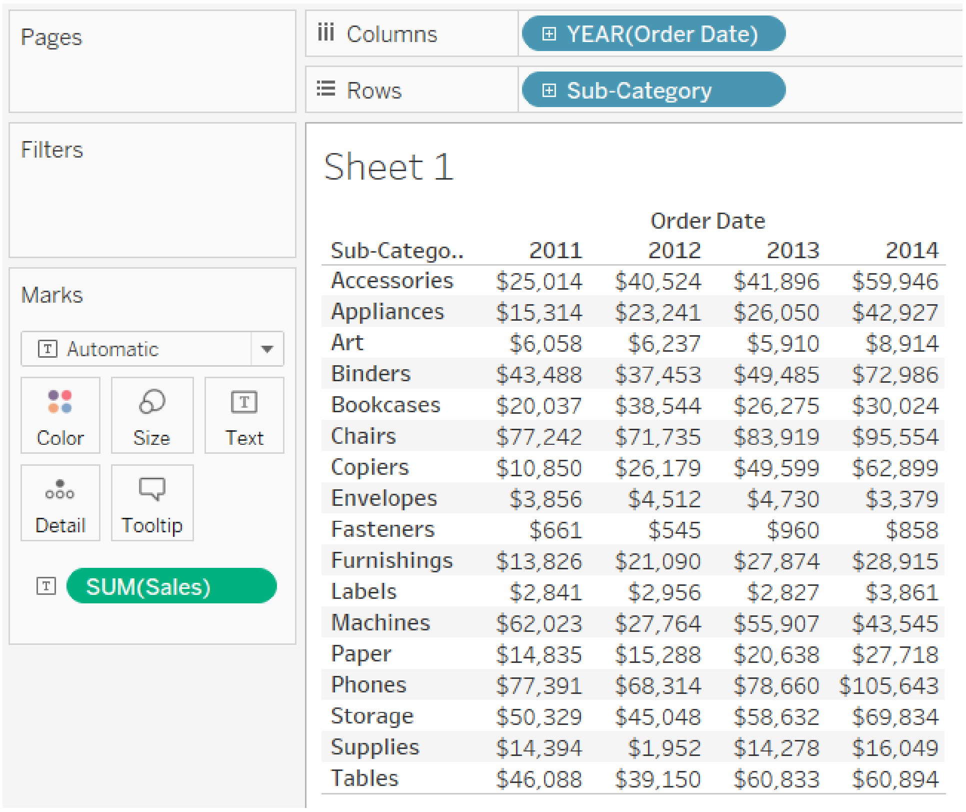
The Text mark type is useful when you want to display the numbers associated with one or more dimension members. This type of view is often called a text table, a cross-tab, or a pivot table. Tableau displays your data using text when:

- The Marks card drop-down menu is set to **Automatic**, and you place one or more dimensions as the inner fields on both the **Rows** and the **Columns** shelves.
- You select **Text** from the Marks card drop-down menu.

Initially, the data is displayed as **Abc**.



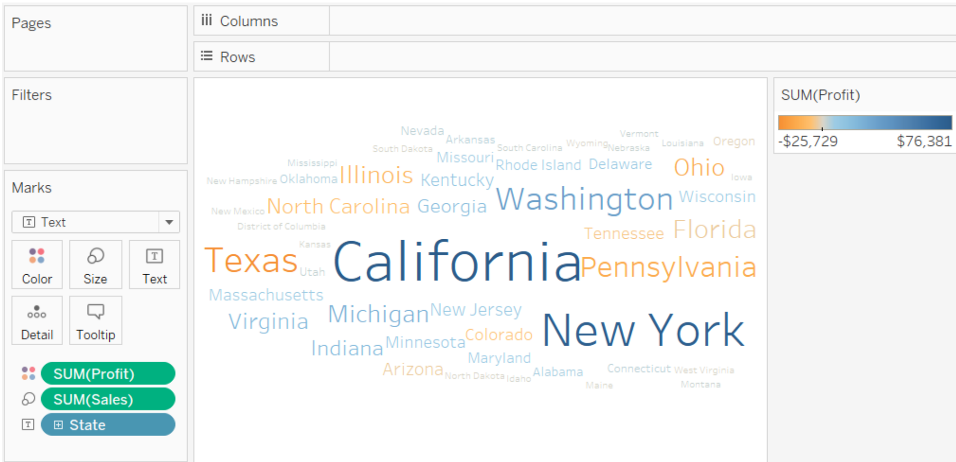
To complete the view, you must place a field (typically a measure) on **Text** on the Marks card. In the view below, the **Sales** measure, which is aggregated as a sum, is used to complete the table.



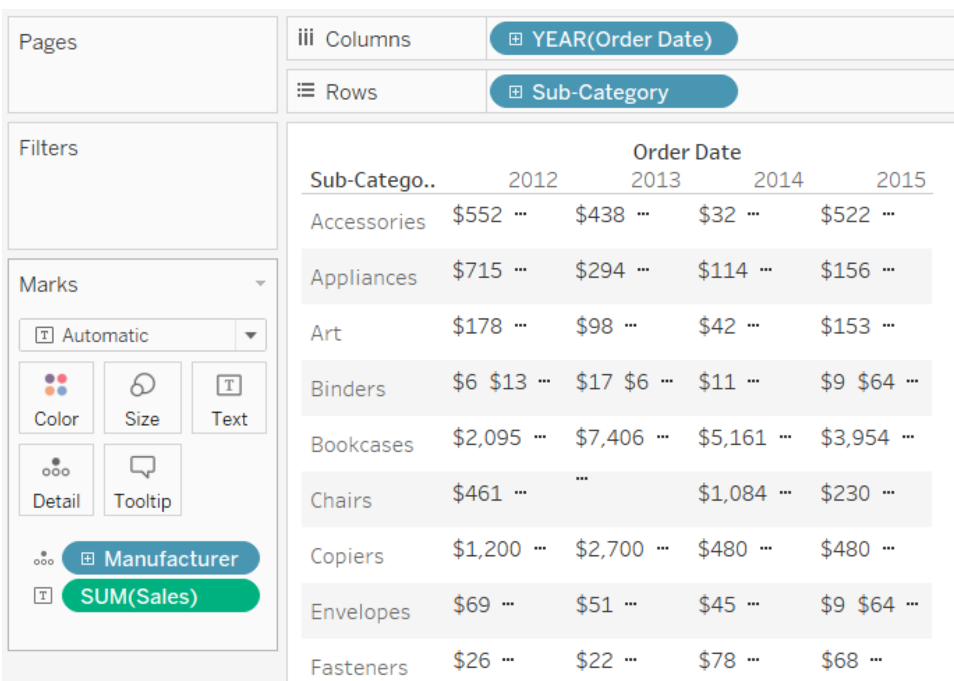
	Order Date			
Sub-Catego..	2011	2012	2013	2014
Accessories	$25,014	$40,524	$41,896	$59,946
Appliances	$15,314	$23,241	$26,050	$42,927
Art	$6,058	$6,237	$5,910	$8,914
Binders	$43,488	$37,453	$49,485	$72,986
Bookcases	$20,037	$38,544	$26,275	$30,024
Chairs	$77,242	$71,735	$83,919	$95,554
Copiers	$10,850	$26,179	$49,599	$62,899
Envelopes	$3,856	$4,512	$4,730	$3,379
Fasteners	$661	$545	$960	$858
Furnishings	$13,826	$21,090	$27,874	$28,915
Labels	$2,841	$2,956	$2,827	$3,861
Machines	$62,023	$27,764	$55,907	$43,545
Paper	$14,835	$15,288	$20,638	$27,718
Phones	$77,391	$68,314	$78,660	$105,643
Storage	$50,329	$45,048	$58,632	$69,834
Supplies	$14,394	$1,952	$14,278	$16,049
Tables	$46,088	$39,150	$60,833	$60,894

Note: To create a cross-tab of any data view, select **Worksheet** > **Duplicate as Crosstab**.

When you add additional levels of detail by placing a dimension on **Detail**, **Color**, **Shape**, **Size**, or **Text** on the Marks card, the values are shown side-by-side and wrap to fill the cell. Allowing the text to stack up makes it possible to create word cloud visualizations like the one shown below.



If the application window is too small for all of the text values, some of the displayed text values will be truncated.



If the application window is too small for all of the text values, the cell displays an ellipses to indicate that there are more values than can be displayed. Number values that can't be displayed are indicated with pound signs (#).

For details on how to build a text table, see **Build a Text Table** on page 1656.

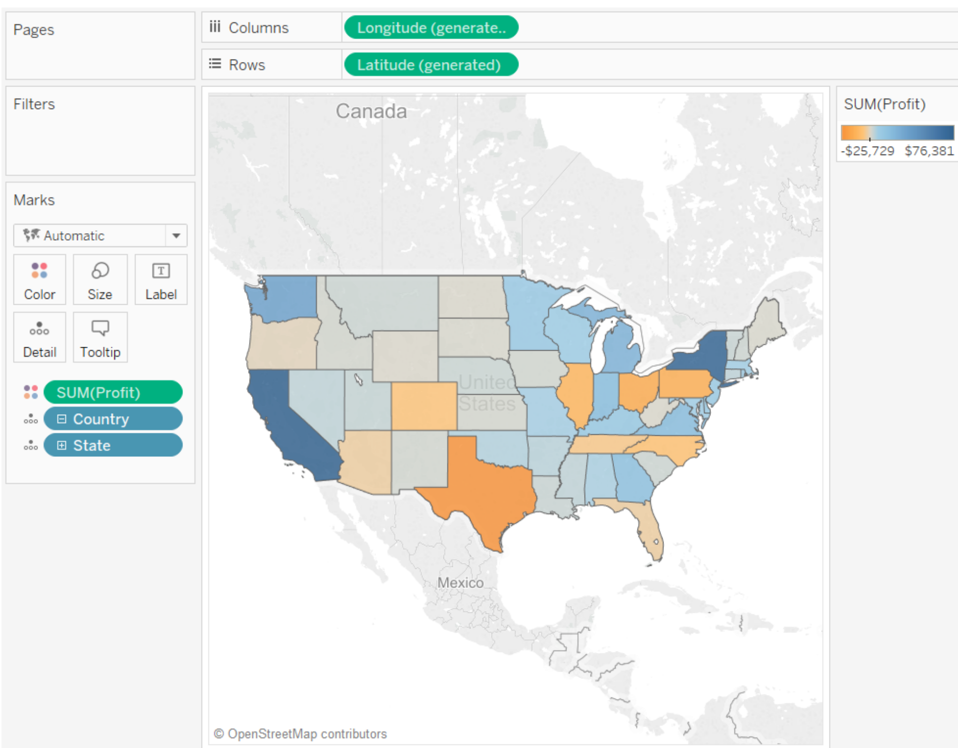
Map mark

The Map mark type uses geocoding to fill a polygon or line with a color based on data. The primary use of the map mark type is for creating polygon or line maps. The area to be filled is defined by the geographic fields used in the view. Tableau displays data using a filled map when:

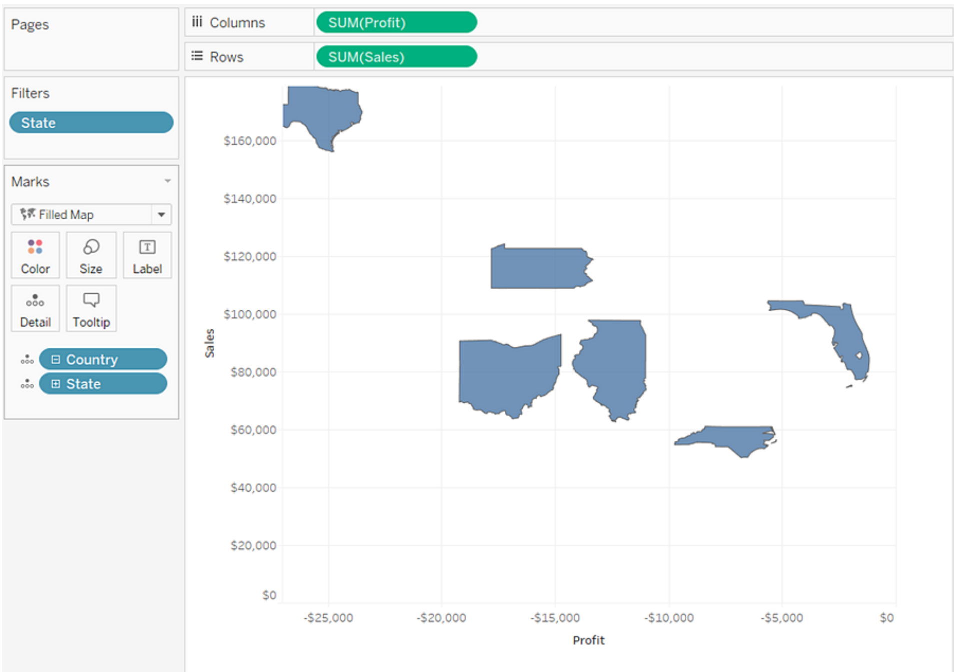
- A geographic dimension is on **Detail** on the Marks card, along with **Latitude** and **Longitude** on the **Rows** and **Columns** shelves. A measure or continuous dimension is added to **Color** on the Marks card.
- You select **Map** from the Marks card drop-down menu.

The data view below shows a map with the **State** geographic dimension on **Detail** on the Marks card. The **Profit** measure is on **Color** on the Marks card.

The polygon for each state is filled with a color based on the profit for that state. The profit in Texas is low (in fact, negative) while the profit in California is high.



You can also use the Map mark type when you are plotting two measures against each other. By default, placing measures on both the **Rows** and **Columns** shelves creates a scatter plot. However, adding a geographic dimension allows you to change the mark type to Map. Each mark becomes the area defined by the geographic field. For example, the view below shows **Sales** versus **Profit** for a selection of states. Rather than showing a round mark with a label for each state, the view uses the Map mark type to draw the outline of each state. While this works well for recognizable areas, it is not the best choice for areas that are similar in shape or difficult to recognize.



For more details on building maps, see **Get Started Mapping with Tableau** on page 1755and **Maps and Geographic Data Analysis in Tableau** on page 1755.

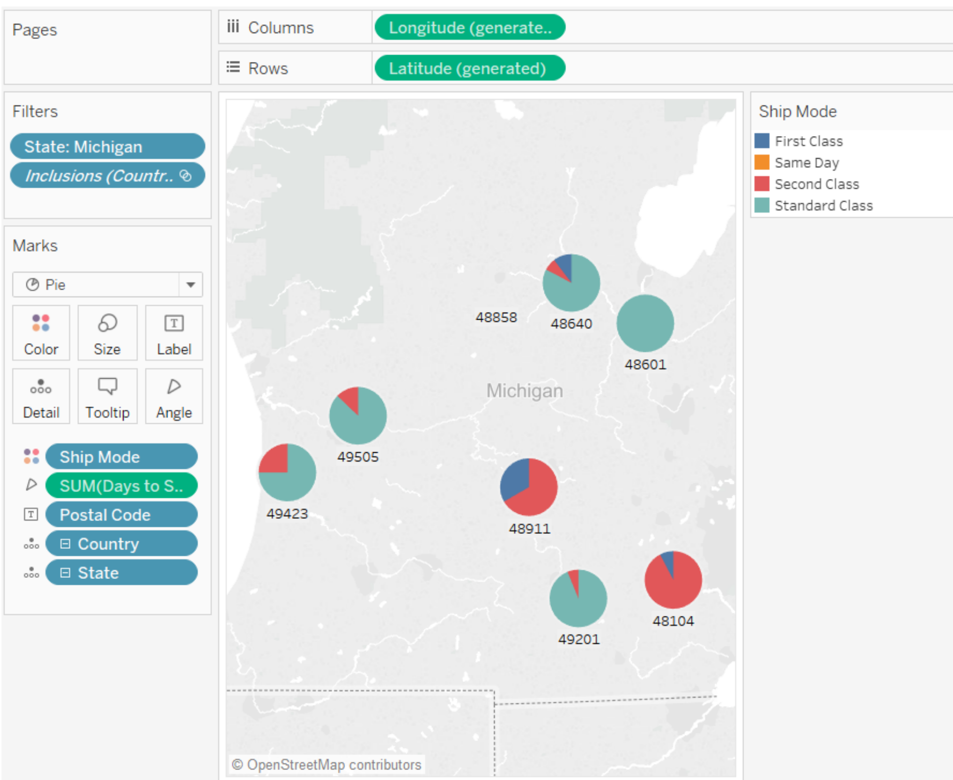
Pie mark

You can use the Pie mark to show proportions. Although generally this type of information can be better shown using stacked bar charts, there are cases where pie marks can be very effective—for example, when trying to convey the percentage allocation of marketing expenses by state where the spending of geographically close states are very relevant.

Tableau will never use the Pie mark as an automatic mark type, but you can select **Pie** from the Marks card drop-down menu.

When you select the Pie mark type, an additional target named Angle is displayed on the Marks card. The Angle target determines the angular measure of the pie wedges. For example, if you place a measure such as **Sales** on **Angle** on the Marks card, the 360 degrees of the pie corresponds to the total sum of sales, and each wedge is divided by the values of the field on **Color** on the Marks card.

The view below shows the time it took to ship products by various ship modes. The data overlays a map and shows data by state. We can see that standard class shipping takes the longest in most areas.



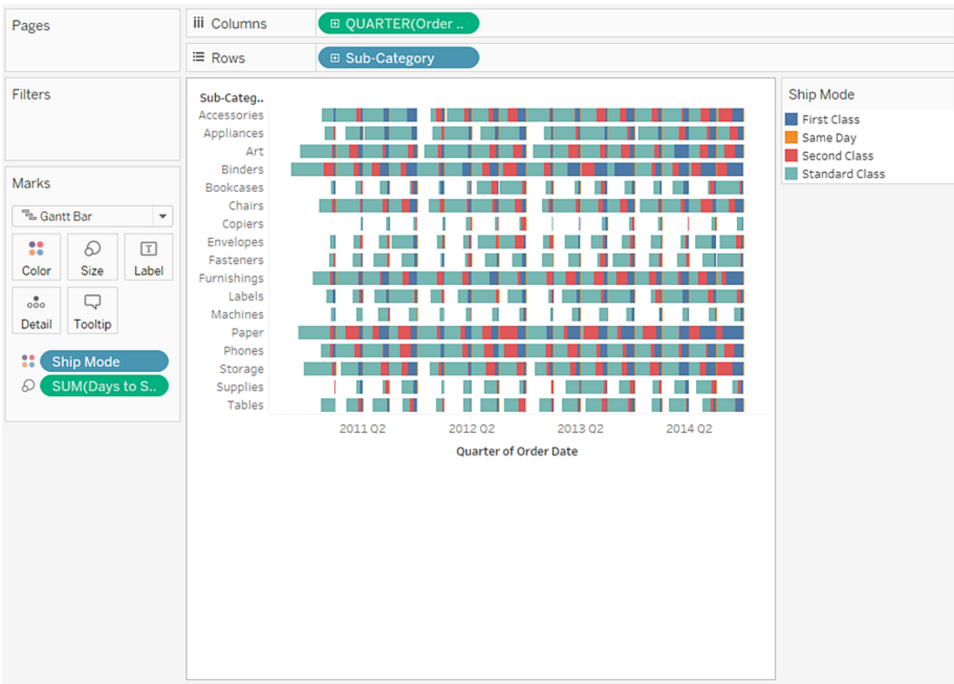
For details on how to build and use pie charts, see **Build a Pie Chart** on page 1645 and **Get Started Mapping with Tableau** on page 1755.

Gantt Bar mark

- The Marks card drop-down menu is set to **Automatic** and you place one or more dimensions on either the **Columns** shelf or the **Rows** shelf, and then plot the dimensions against a continuous quantity.
- You select **Gantt Bar** from the Marks card drop-down menu.

The distinguishing characteristic of Gantt Bars is that the length of every mark is proportional to the measure placed on **Size** on the Marks card.

The view below displays a dimension as a function of a continuous date. If the Marks card drop-down menu is set to **Automatic**, the data would be displayed using bars. Selecting **Gantt Bar** and adding additional fields will display a view like the following.



In particular, placing the **Days to Ship** measure on **Size** on the Marks card causes every bar in the view to be drawn with a length that indicates the delivery time of an order. Additionally, placing the **Ship Mode** dimension on **Color** on the Marks card causes each bar to be colored according to the ship mode.

For details on how to build Gantt Bar chart, see **Build a Gantt Chart** on page 1609.

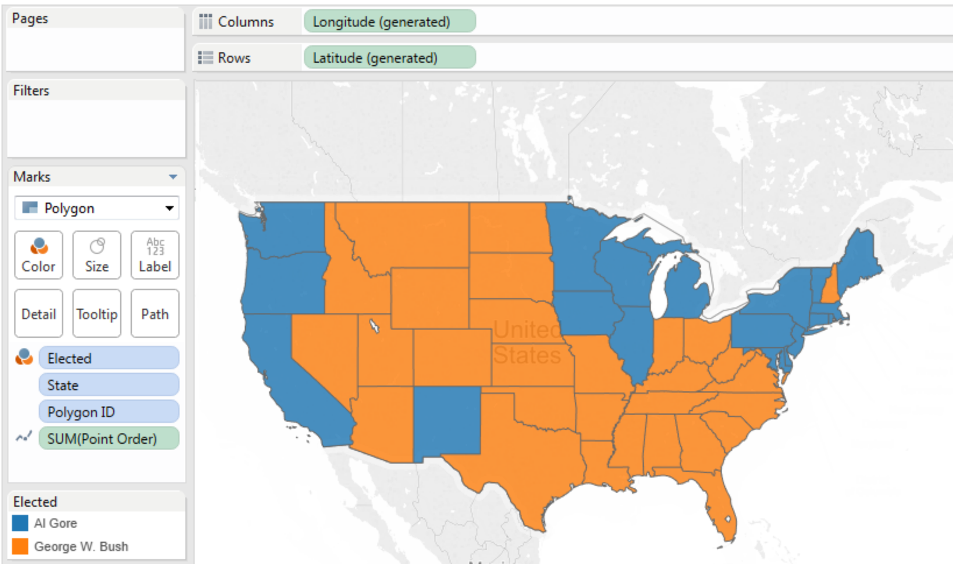
Polygon mark

Polygons are points connected by lines enclosing an area. The polygon mark type is useful when you want to connect points to create data areas. Tableau displays data using polygons when you select Polygon from the **Marks** menu.

> The polygon mark is not commonly used and often requires a specially constructed data source.

The view shown below comes from a specially constructed data source that holds geographic and election data. It displays the 48 contiguous US states as a function of latitude and longitude and color-encodes each state by the 2000 presidential election results.

If **Marks** is set to Automatic, the data will be displayed using a shape. Manually selecting Polygon and adding additional fields to the view causes a different view to be created.



Every state is considered to be a polygon in the data source. The **PolygonID** field on the **Detail** target is distinct for each US state. You can remove states from the view by filtering this field.

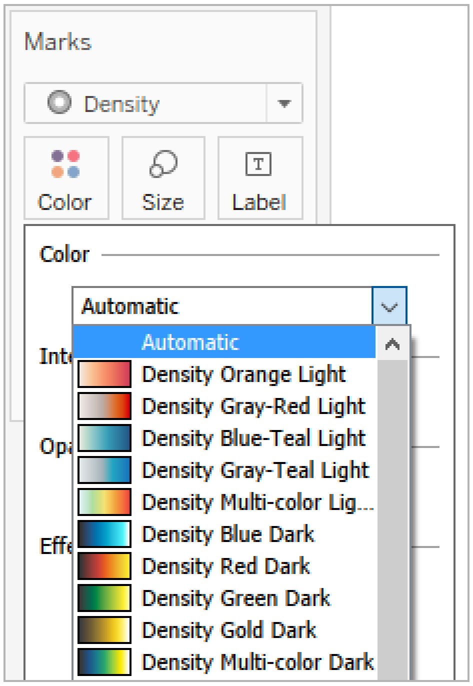
Additionally, you can specify the drawing order of the lines that constitute each polygon by placing a field on the **Path** target. In this example, the **PointOrder** measure is used to draw each state.

Density mark (Heatmap)

You can use the density mark to visualize patterns or trends in dense data with many overlapping marks. Although generally this type of information can be shown using a circle mark, there are cases where density marks can be very effective—for example, when there are many data points in an area, and you need to understand where those marks are most concentrated. The density mark can be used effectively with maps or charts that use point-based marks (scatterplots, unit histograms, box plots over individual marks, or point maps).

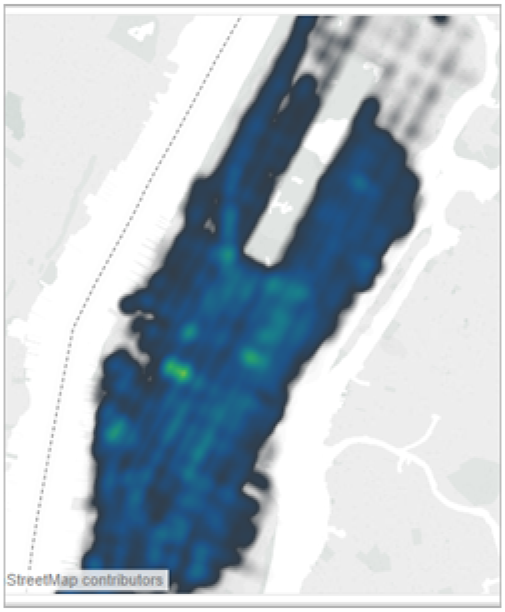
Tableau does this by grouping overlaying marks, and color-coding them based on the number of marks in the group.

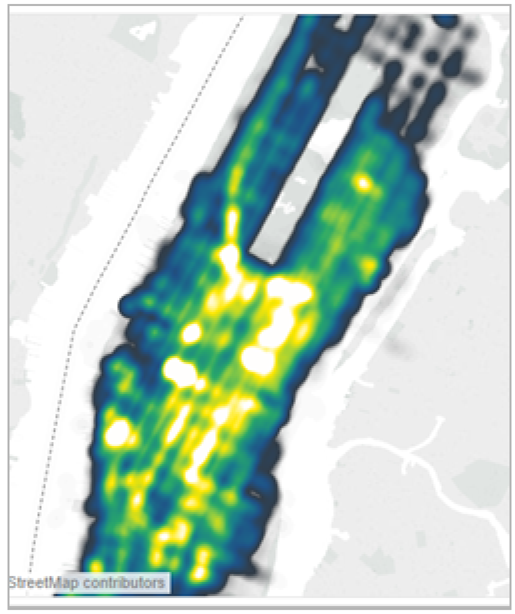
Density marks work by using color to show the relative intensity of data in a given area of the chart. You can adjust the colors of the density mark by selecting Color from the Marks card. Choose from ten density color palettes, or any of the existing color palettes.



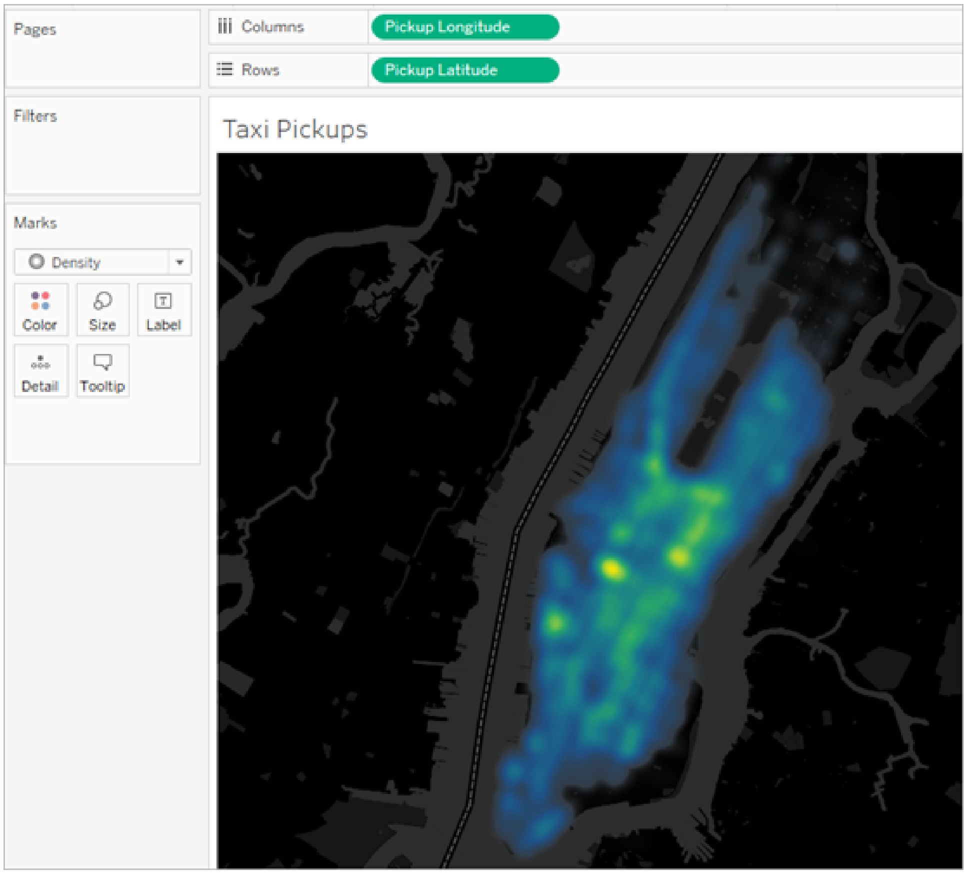
In the Color menu, use the Intensity slider to increase or decrease the vividness of the density marks. For example, increasing intensity, or vividness, lowers the "max heat" spots in your data, so that more appear.

Low Intensity (50%) High Intensity (80%)





The view shown below comes from a data source that contains many geographic points for one month of taxi pickups in Manhattan. It displays the location and frequency of taxi pickups, with lighter colors being most frequent.



You can select **Density** from the Marks card drop-down menu.

For details on how to build a density map or a density chart, see **Create Heatmaps that Show Trends or Density in Tableau** on page 1910 and **Build with Density Marks (Heatmap)** on page 1600.

Control the Appearance of Marks in the View

Control color, size, shape, detail, text, and tooltips for marks in the view using the Marks card. Drag fields to buttons on the Marks card to encode the mark data. Click the buttons on the Marks card to open Mark properties. For related information on marks, see **Change the Type of Mark in the View** on page 1332 and **Marks card** on page 227.

> **Note**: The order of dimension fields on the Marks card is hierarchical from top to bottom, and affects sorting in the view. Tableau first considers the topmost dimension field when ordering marks in the view, and then considers the dimensions beneath it on the Marks card.

Assign colors to marks

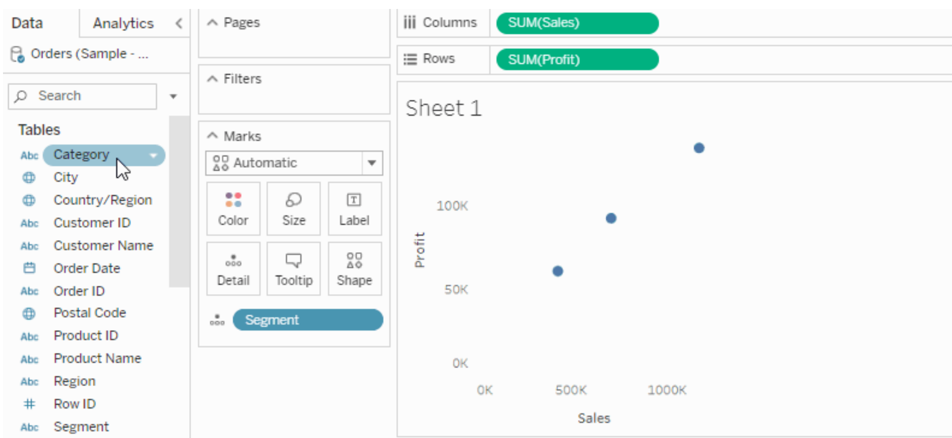
To assign a color to marks in the view, do one of the following:

- On the Marks card, click **Color**, and then select a color from the menu.

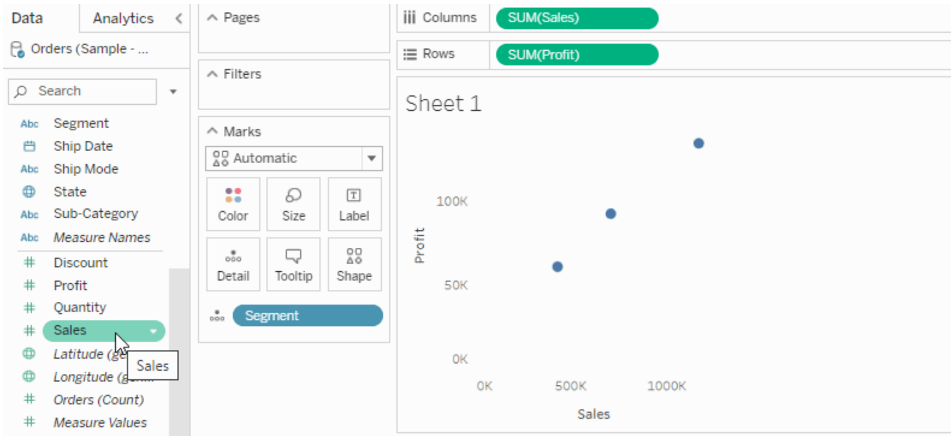
  This updates all marks in the view to the color you choose. All marks have a default color, even when there are no fields on **Color** on the **Marks** card. For most marks, blue is the default color; for text, black is the default color.

- From the **Data** pane, drag a field to **Color** on the Marks card.

  Tableau applies different colors to marks based on the field's values and members. For example, if you drop a discrete field (a blue field), such as Category, on Color, the marks in the view are broken out by category, and each category is assigned a color.



If you drop a continuous field, such as SUM(sales), on Color, each mark in the view is colored based on its sales value.



Edit colors

To change the color palette or customize how color is applied to your marks:

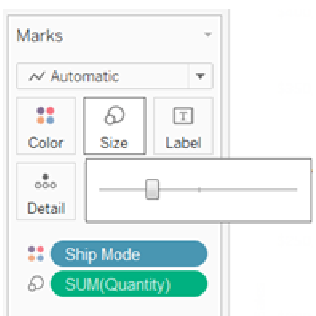
- On the Marks card, click **Color** > **Edit Colors**.

For more information, see **Color Palettes and Effects** on page 1394.

Change the size of marks

To change the size of marks in the view, do one of the following:

- On the Marks card, click **Size**, and then move the slider to the left or right.

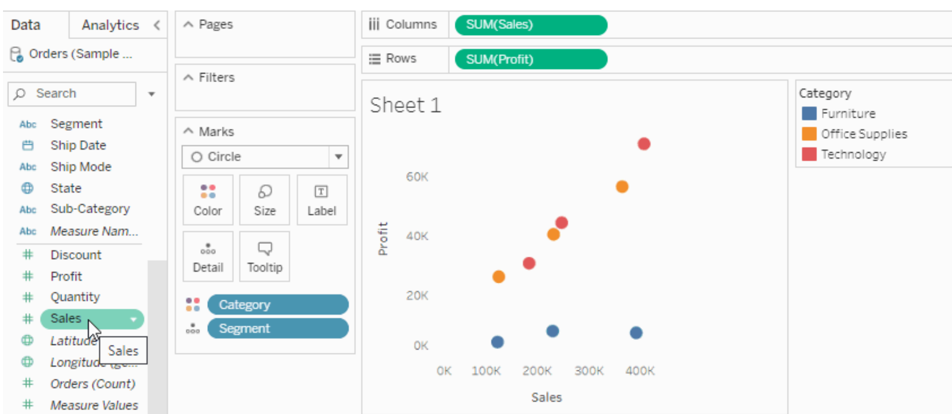


The Size slider affects different marks in different ways, as described in the following table.

Mark Type	Description
Circle, Square, Shape, Text	Makes the mark bigger or smaller.
Bar, Gantt Bar	Makes bars wider or narrower.
Line	Makes lines thicker or thinner.
Polygon	You cannot change the size of a polygon.
Pie	Makes the overall size of the pie bigger and smaller.

The size of your data view is not modified when you change marks using the **Size** slider. However, if you change the view size, the mark size might change to accommodate the new formatting. For example, if you make the table bigger, the marks might become bigger as well.

- From the **Data** pane, drag a field to **Size** on the Marks card.



When you place a discrete field on **Size** on the **Marks** card, Tableau separates the marks according to the members in the dimension, and assigns a unique size to each member. Because size has an inherent order (small to big), categorical sizes work best for ordered data like years or quarters.

Size-encoding data with a discrete field separates the marks in the same way as the **Detail** property does, and then provides additional information (a size) for each mark. When you add categorical size encoding to a view, Tableau displays a legend showing the sizes assigned to each member in the field on the **Size** target. You can modify how these sizes are distributed using the Edit Sizes dialog box.

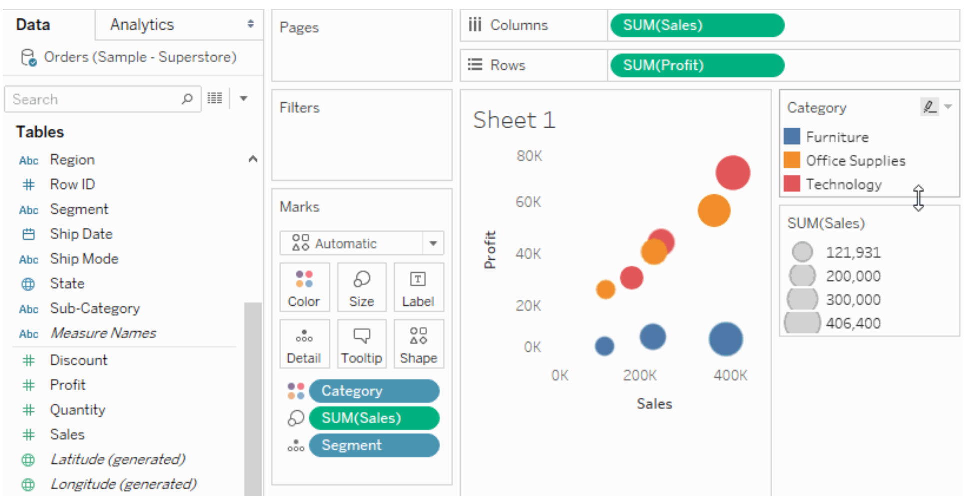
When you place a continuous field on **Size** on the **Marks** card, Tableau draws each mark with a different size using a continuous range. The smallest value is assigned the smallest sized mark and the largest value is represented by the largest mark.

When you add quantitative size encoding to the view, Tableau displays a legend showing the range of values over which sizes are assigned. You can modify how these sizes are distributed using the Edit Sizes dialog box.

Edit marks sizes

To edit the size of marks, or change how size is being applied to marks in the view:

1. On the Size legend card (which appears when you add a field to Size on the Marks card), click the drop-down arrow in the right-hand corner and select **Edit Sizes**.



2. In the Edit Sizes dialog box that appears, make your changes and then click **OK**.

   The options available depend on whether the field being applied to Size is a continuous or discrete field.

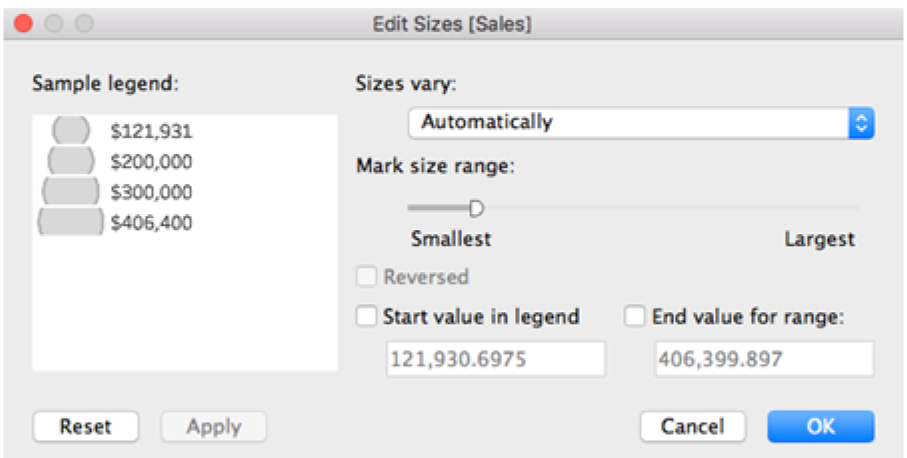
   For continuous fields, you can do the following:

   - For **Sizes vary**, click the drop-down box and select one of the following:
     - **Automatically** - Selects the mapping that best fits your data. If the data is numeric and does not cross zero (all positive or all negative), the From zero mapping is used. Otherwise, the By range mapping is used.
     - **By range** - Uses the minimum and maximum values in the data to

determine the distribution of sizes. For example, if a field has values from 14 to 25, the sizes are distributed across this range.

- **From zero** - Sizes are interpolated from zero, assigning the maximum mark size to the absolute value of the data value that is farthest from zero.

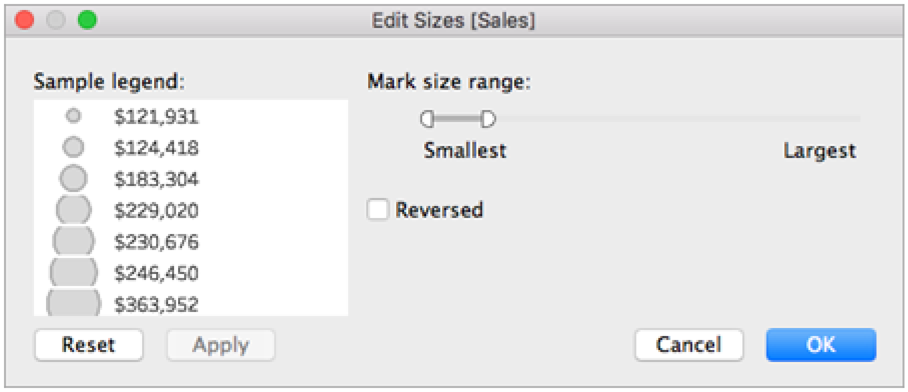
- Use the range slider to adjust the distribution of sizes. When the From zero mapping is selected from the Sizes vary drop-down menu, the lower slider is disabled because it is always set to zero.
- Select **Reversed** to assign the largest mark to the smallest value and the smallest mark to the largest value. This option is not available if you are mapping sizes from zero because the smallest mark is always assigned to zero.
- To modify the distribution of sizes, select the **Start value in legend** and **End value for range** check boxes and type beginning and end values for the range.



For discrete fields, you can do the following:

- Use the range slider to adjust the distribution of sizes.
- Select **Reversed** to assign the largest mark to the smallest value and the

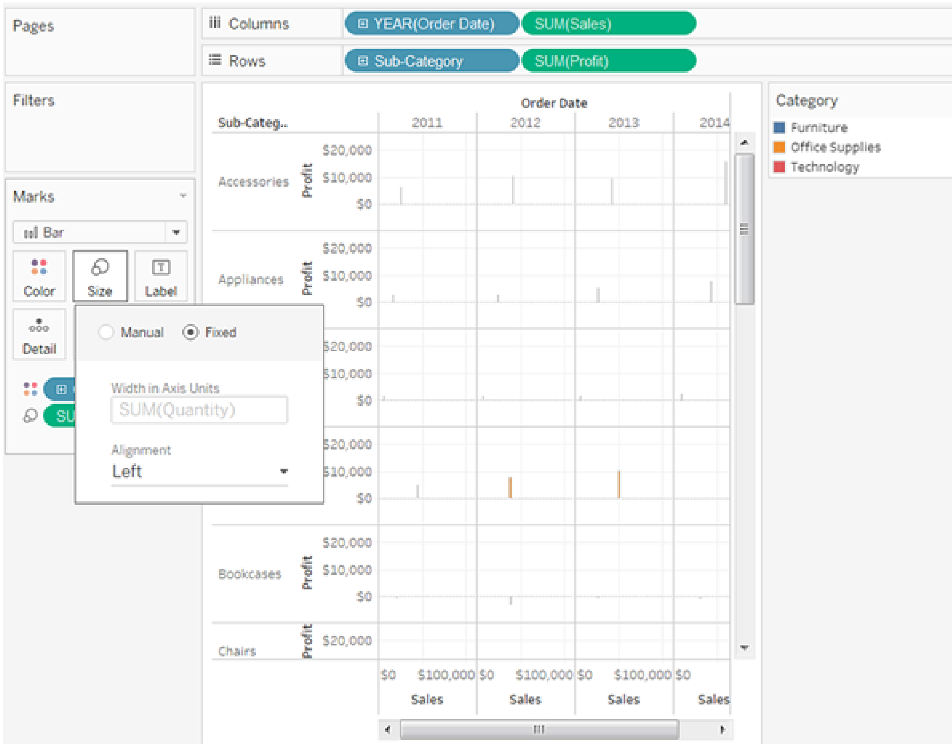
smallest mark to the largest value.



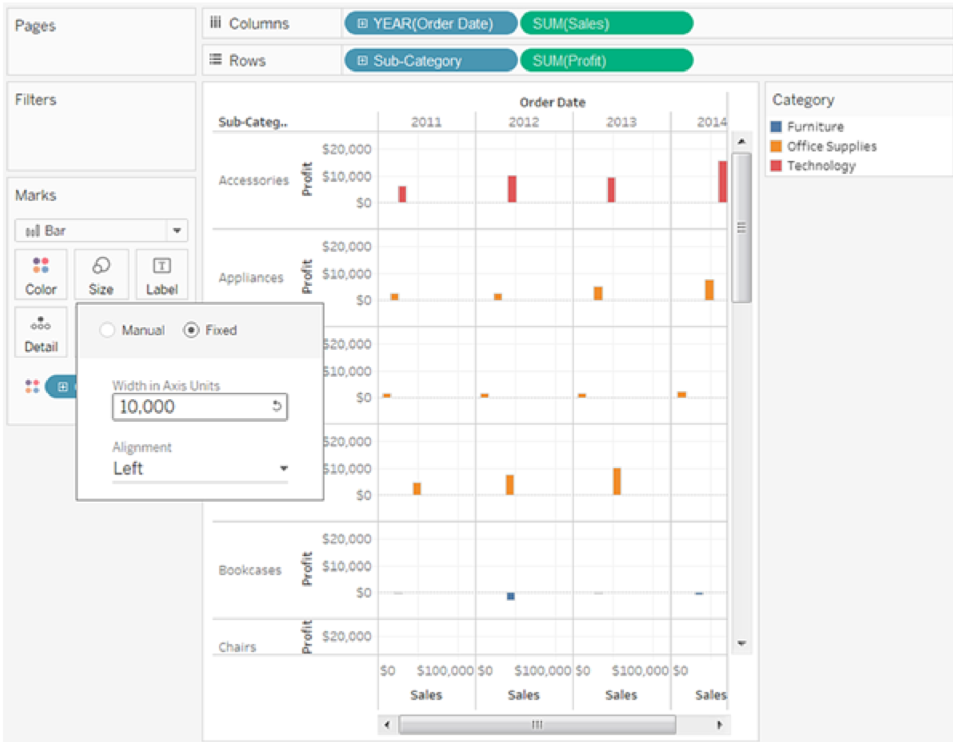
Continuous axis mark sizing

For views where the mark type is **Bar** and there are continuous (green) fields on both **Rows** and **Columns**, Tableau supports additional options and defaults for sizing the bar marks on the axis where the bars are anchored.

- The bar marks in histograms are continuous by default (with no spaces between the marks), and are sized to match the size of the bins. See **Build a Histogram** on page 1623 for an example.
- When there is a field on **Size**, you can determine the width of the bar marks on the axis where the bars are anchored by using the field on **Size**. To do this, click the **Size** card and select **Fixed**.



- When there is no field on **Size**, you can specify the width of the bar marks on the axis where the bars are anchored in axis units. To do this, click the **Size** card, choose **Fixed**, and then type a number in the **Width in axis units** field.



- When there is a continuous date field on the axis where the bars are anchored, the width of the marks is set to match the level of the date field. For example, if the level of the continuous date field is MONTH, the bars are exactly one month wide—that is, slightly wider for 31-day months than for 30-day months. You can configure the width of the bars by clicking the **Size** card, choosing **Fixed**, and then typing a number in the **Width in days** field, but the resulting bar widths don't take into account the varying lengths of time units such as months and years.

Add labels or text for marks

To add mark labels or text to the visualization:

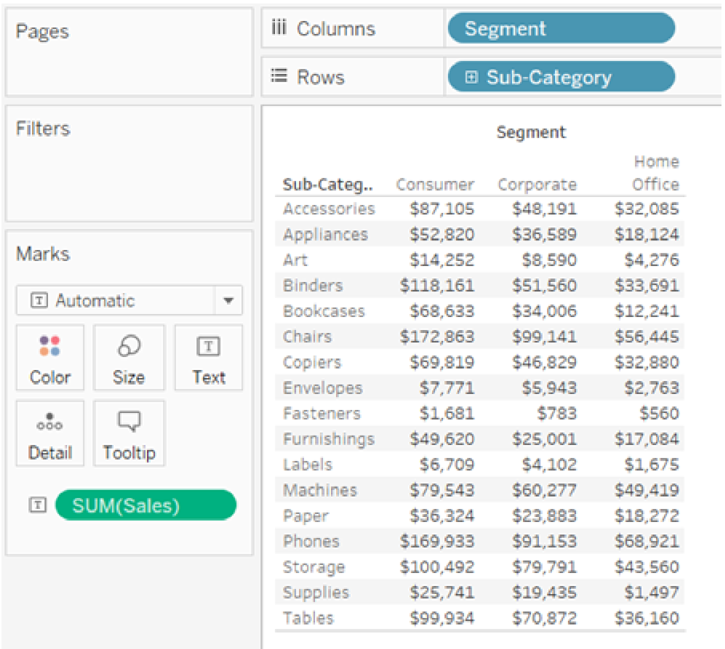
- From the **Data** pane, drag a field to **Label** or **Text** on the Marks card.

When working with a text table, the Label shelf is replaced with Text, which allows you to view the numbers associated with a data view. The effect of text-encoding your data view depends on whether you use a dimension or a measure.

- Dimension – When you place a dimension on **Label** or **Text** on the Marks card, Tableau separates the marks according to the members in the dimension. The text labels are driven by the dimension member names.

- Measure – When you place a measure on **Label** or **Text** on the Marks card, the text labels are driven by the measure values. The measure can be either aggregated or disaggregated. However, dis-aggregating the measure is generally not useful because it often results in overlapping text.

Text is the default mark type for a text table, which is also referred to as a cross-tab or a PivotTable.

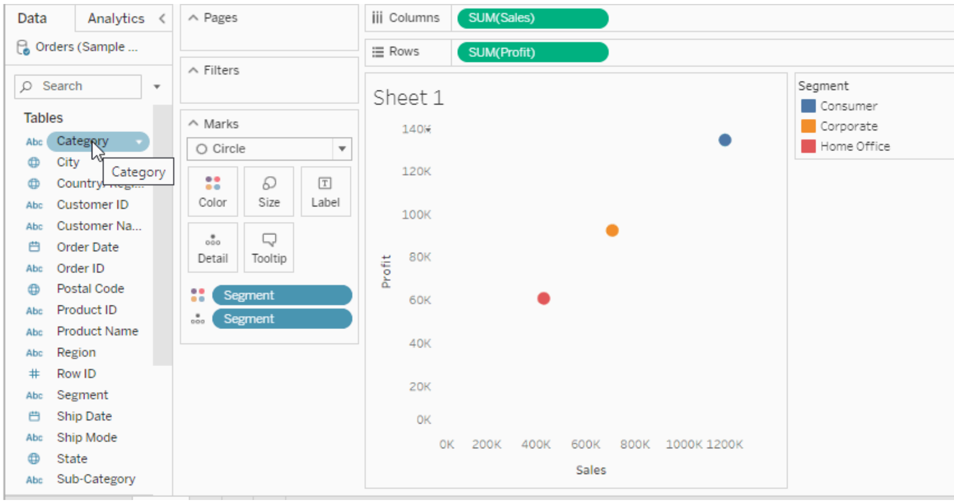


Sub-Categ..	Consumer	Corporate	Home Office
Accessories	$87,105	$48,191	$32,085
Appliances	$52,820	$36,589	$18,124
Art	$14,252	$8,590	$4,276
Binders	$118,161	$51,560	$33,691
Bookcases	$68,633	$34,006	$12,241
Chairs	$172,863	$99,141	$56,445
Copiers	$69,819	$46,829	$32,880
Envelopes	$7,771	$5,943	$2,763
Fasteners	$1,681	$783	$560
Furnishings	$49,620	$25,001	$17,084
Labels	$6,709	$4,102	$1,675
Machines	$79,543	$60,277	$49,419
Paper	$36,324	$23,883	$18,272
Phones	$169,933	$91,153	$68,921
Storage	$100,492	$79,791	$43,560
Supplies	$25,741	$19,435	$1,497
Tables	$99,934	$70,872	$36,160

Separate marks in the view by dimension members

To separate marks in the view (or add more granularity):

- From the **Data** pane, drag a dimension to **Detail** on the Marks card.



When you drop a dimension on **Detail** on the Marks card, the marks in a data view are separated according to the members of that dimension. Unlike dropping a dimension on the **Rows** or **Columns** shelf, dropping it on **Detail** on the Marks card is a way to show more data without changing the table structure.

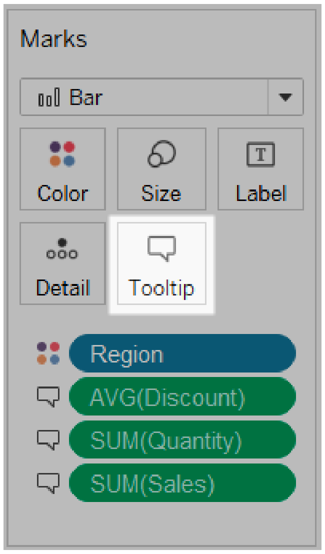
Add tooltips to marks

Tooltips are details that appear when you hover over one or more marks in the view. Tooltips are also convenient for quickly filtering or removing a selection, or viewing underlying data. You can edit a tooltip to include both static and dynamic text. You can also modify which fields are included in a tooltip and whether you want to be able to use those fields to select marks in the view.

For details on showing a visualization from a worksheet in a tooltip (Viz in Tooltip), **Create Views in Tooltips (Viz in Tooltip)** on page 1475.

Add a tooltip

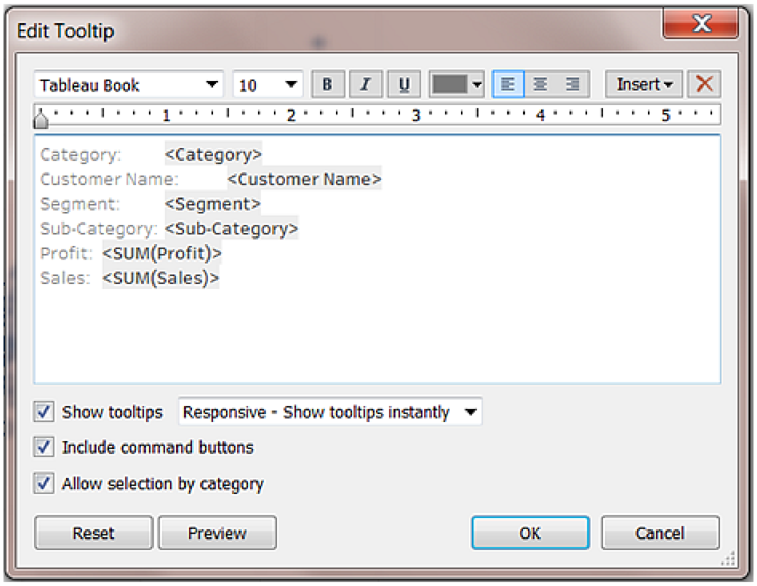
1. Drag a field to **Tooltip** on the **Marks** card.
2. Click **Tooltip** on the **Marks** card to open the Edit Tooltip dialog box, where you can add text, rearrange the tooltip contents, and insert more fields.



Note: Dimensions are added to the tooltip using the ATTR aggregation, which means the tooltip may sometimes display as an asterisk. The asterisk indicates that there are multiple dimension members that apply to the mark you are pointing at. For example, a mark may represent the aggregated sales for all regions. Adding the Region field to the tooltip results in an asterisk because the mark represents more than one region. To avoid showing an asterisk, add the dimension to Detail on the Marks card or use it elsewhere in the view to ensure the marks are at the same level of detail.

Tooltip options

After you open the Edit Tooltip dialog box, there are several options that you can choose from to format the tooltips in your view and configure their behavior. You can choose from the options below.



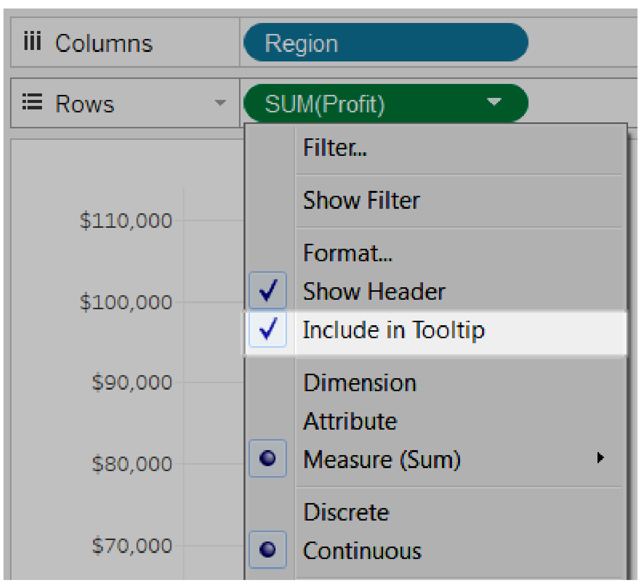
- **Format tooltips:** Tooltips are specified on a per-sheet basis and can be formatted using the tools on the top of the Edit Tooltip dialog box.
- **Adding dynamic text:** Use the **Insert** menu at the top of the dialog box to add dynamic text such as field values, sheet properties, and more. The **All Fields** command on the **Insert** menu adds all field names and values that are used in the view to the tooltip for any mark in the view.
- **Show tooltips:** Tooltips are shown by default. If you prefer to hide automatic tooltips, clear the **Show Tooltips** check box.
- **Configure tooltip behavior:** Configure how tooltips behave by selecting one of the following options in the drop-down list next to the **Show Tooltips** check box:
  - **Responsive - Show tooltips instantly**: Select this option to show tooltips instantly as you move the cursor over the marks in the view. This is the default for all views.

    With this option, tooltips appear without command buttons. You must first click a mark in the view to see the command buttons.
  - **On Hover - Show tooltips on hover**: Select this option to show tooltips only after you rest the cursor on a mark. However, with this option, command buttons appear on the tooltip without any further action from you (unless you specify otherwise in the Edit Tooltips dialog box).
- **Show commands:** Select the **Include command buttons** check box to add **Keep**

Only, **Exclude**, **Group Members**, **Create Set** and **View Data** buttons at the top of the tooltip. These buttons show both in Tableau Desktop and when the view is published to the web or viewed on a mobile device.

For more information about grouping members, see **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1184. For more information about creating sets, see **Create Sets** on page 1186.

- **Select marks in a view from the tooltip**. Select the **Allow selection by category** check box to be able to select marks in a view that have the same value by clicking on a discrete field in a tooltip. If this check box is selected, you can use this feature to select marks in a view in Tableau Desktop, when the view is published to the web or viewed on a mobile device.
- **Add or remove fields:** To add and remove fields when using the automatic tooltip, right-click (control-click on Mac) the field on one of the shelves in the view and select **Include in Tooltip**.



Note: The **Include in Tooltip** option is only available if you have not customized the tooltip. If you have customized the tooltip, you can return to the automatic tooltip by clicking **Tooltip** on the Marks card and then clicking **Reset** in the Edit Tooltip dialog box.

Change the shape of marks

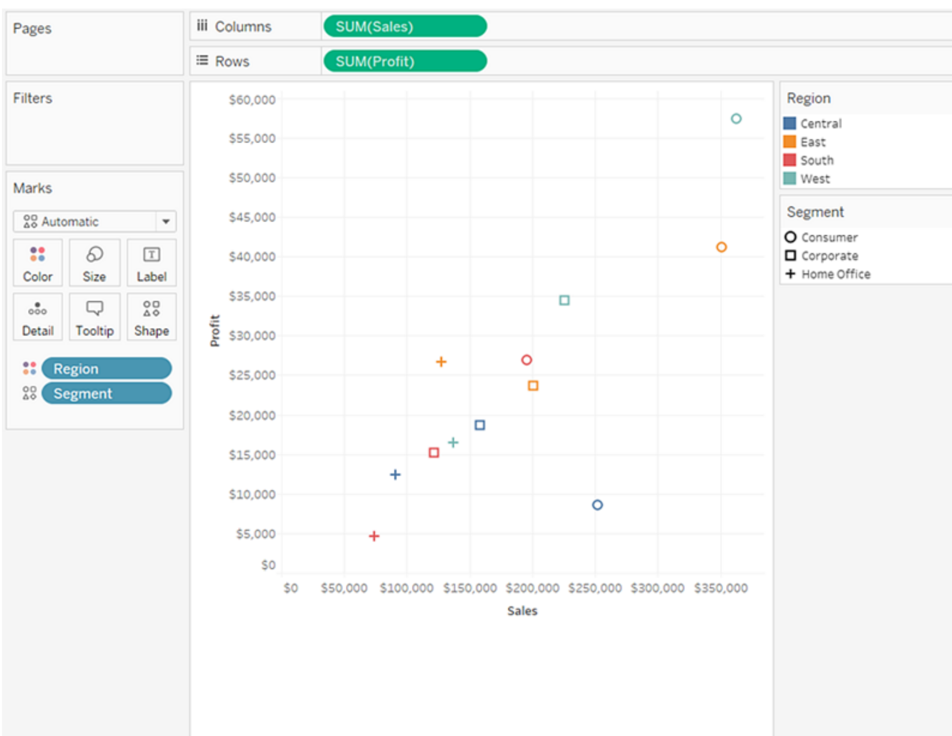
To change the shapes of marks:

- From the **Data** pane, drag a field to **Shape** on the Marks card.

When you place a dimension on **Shape** on the **Marks** card, Tableau separates the marks according to the members in the dimension, and assigns a unique shape to each member. Tableau also displays a shape legend, which shows each member name and its associated shape. When you place a measure on **Shape** on the **Marks** card, the measure is converted to a discrete measure.

Shape-encoding data separates the marks in the same way as the Detail property does, and then provides additional information (a shape) for each mark. Shape is the default mark type when measures are the inner most fields for both the **Rows** shelf and the **Columns** shelf.

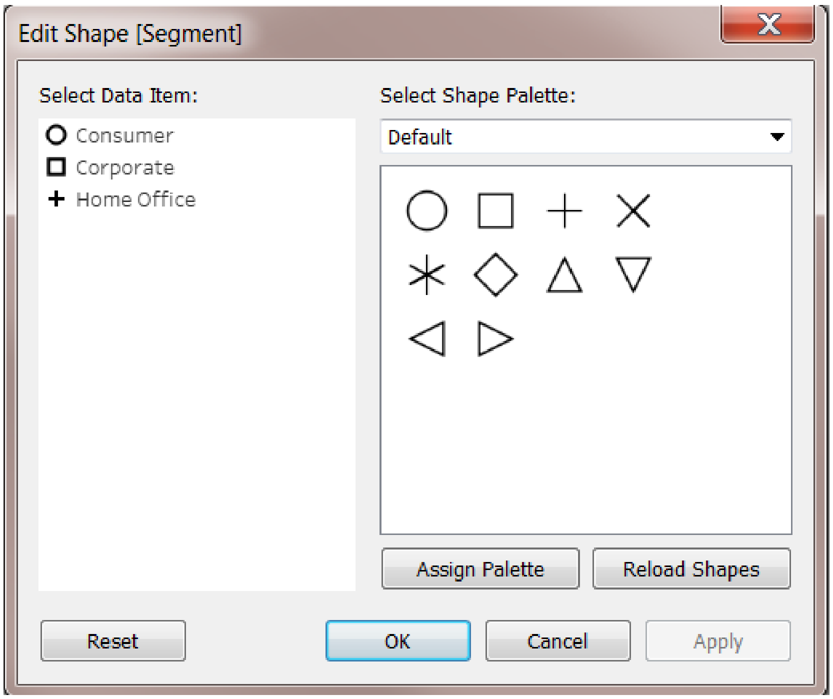
In the view below, the marks are separated into different shapes according to the members of the **Customer Segment** dimension. Each shape reflects the customer segment's contribution to profit and sales.



Edit shapes

By default, ten unique shapes are used to encode dimensions. If you have more than 10 members, the shapes repeat. In addition to the default palette, you can choose from a variety of shape palettes, including filled shapes, arrows, and even weather symbols.

1. Click **Shape** on the **Mark**s card, or select **Edit Shape** on the legend's card menu.
2. In the Edit Shape dialog box, select a member on the left and then select the new shape in the palette on the right. You can also click **Assign Palette** to quickly assign the shapes to the members of the field.



Select a different shape palette using the drop-down menu in the upper right.

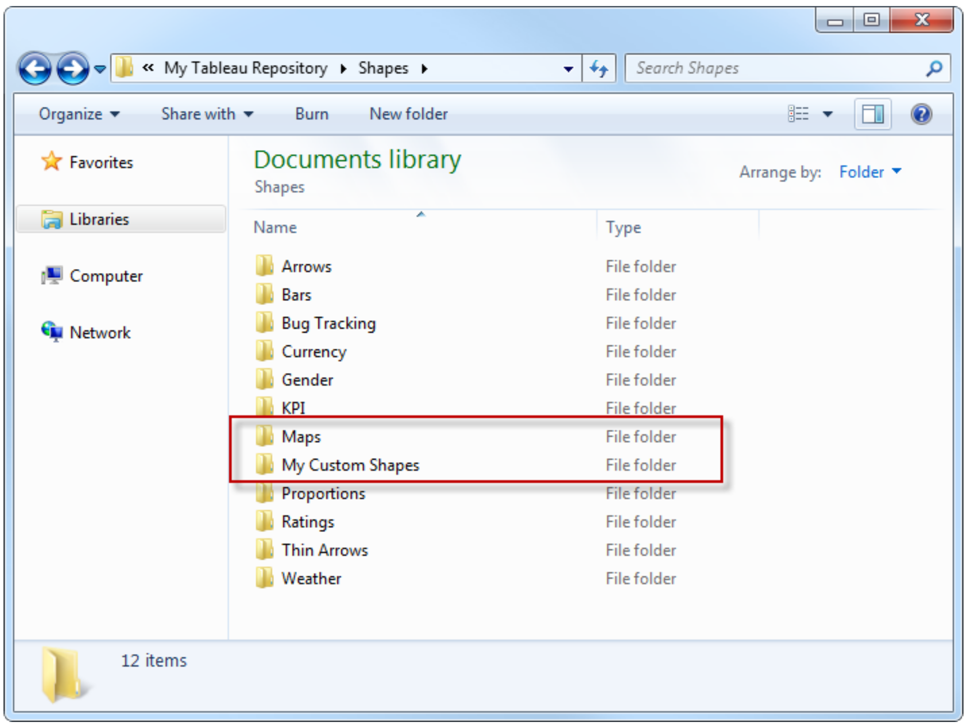
Note: Shape encodings are shared across multiple worksheets that use the same data source. For example, if you define Furniture products to be represented by a square, they will automatically be squares in all other views in the workbook. To set the default shape encodings for a field, right-click (control-click on Mac) the field in the **Data** pane and select **Default Properties** > **Shape**.

Use Custom shapes

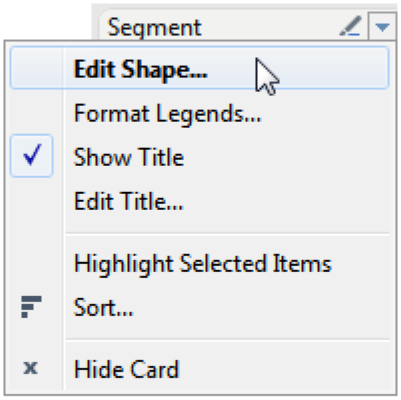
You can add custom shapes to a workbook by copying shape image files to the Shapes folder in your Tableau Repository, which is located in your Documents folder. When you use custom shapes, they are saved with the workbook. That way the workbook can be shared with others.

1. Create your shape image files. Each shape should be saved as its own file and can be in any of several image formats including bitmap (.bmp), portable network graphic (.png), .jpg, and graphics interchange format (.gif).

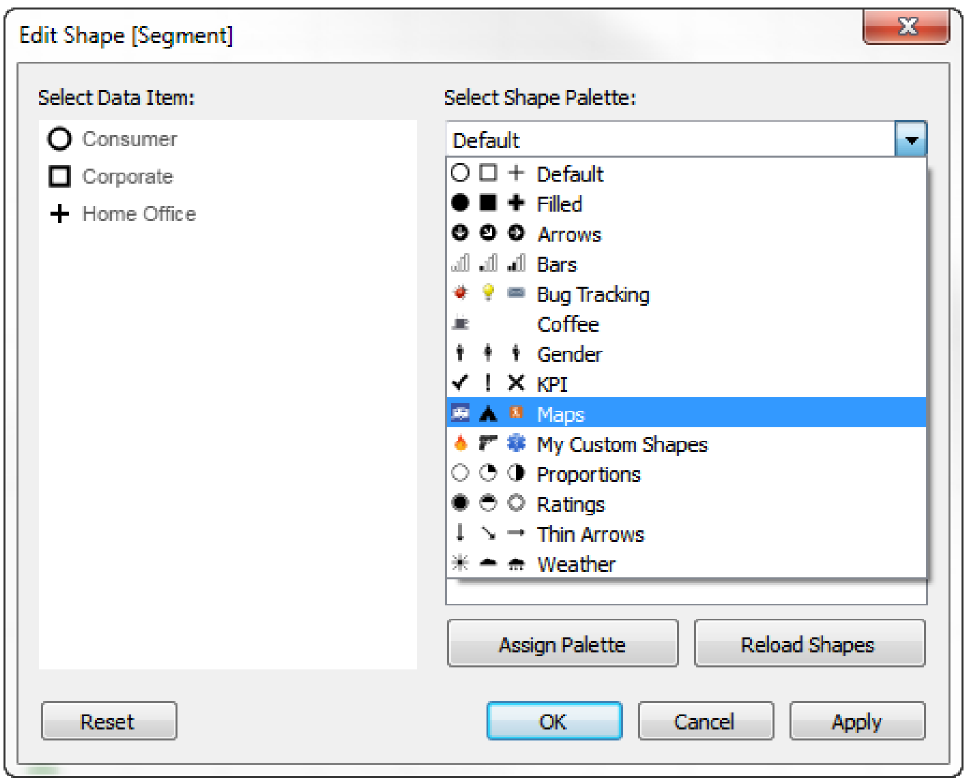
2. Copy the shape files to a new folder in the My Tableau Repository\Shapes folder in your Documents folder. The name of the folder will be used as the name of the palette in Tableau. In the example below, two new palettes are created: Maps and My Custom Shapes.



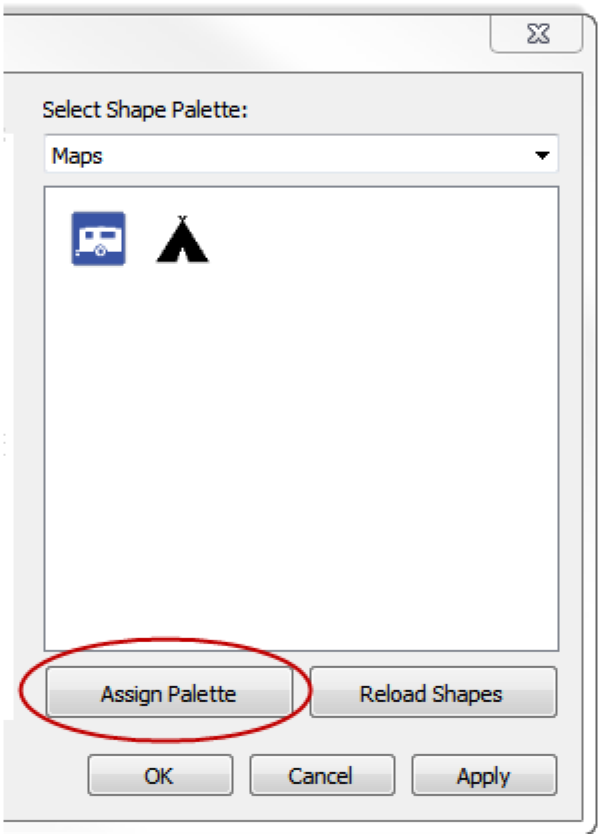
3. In Tableau, click the drop-down arrow on the shape legend, and select **Edit Shape**.



4. Select the new custom palette in the drop-down list. If you modified the shapes while Tableau was running, you may need to click **Reload Shapes**.

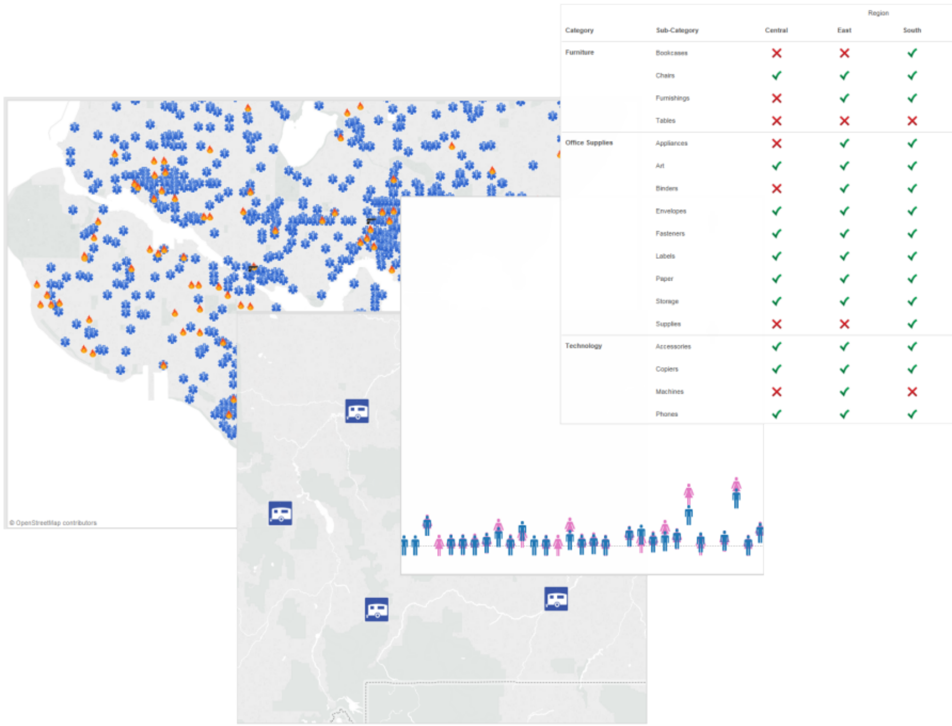


5. You can either assign members shapes one at a time, or click **Assign Palette** to automatically assign the shapes to the members.



Note: You can return to the default palette by clicking the **Reset** button. If you open a workbook that uses custom shapes that you don't have, the workbook will show the custom shapes because the shapes are saved as part of the workbook. However, you can click **Reload Shapes** in the Edit Shapes dialog box to use the ones in your repository instead.

Below are some examples of views that use both the default and custom shape palettes.



Tips for creating custom shapes

When you create custom shapes there are a few things that you can do to improve how your shapes look and function in the view. If you are creating your own shapes, we recommend following general guidelines for making icons or clip art.

- **Suggested size** - Unless you plan on using Size to make the shapes really large, you should try to make your original shape size close to 32 pixels by 32 pixels. However, the original size depends on the range of sizes you want available in Tableau. You can resize the shapes in Tableau by clicking Size on the **Marks** card, or by using the cell size options on the **Format** menu.
- **Adding color encoding** - If you plan to also use Color to encode shapes, you should use a transparent background. Otherwise, the entire square of the image will be colored rather than just the symbol. GIF and PNG file formats both support transparency. GIF files support transparency for a single color that is 100% transparent, while PNG files support alpha channels with a range of transparency levels available on every pixel in the image. When Tableau color encodes a symbol, the amount of transparency for each pixel won't be modified, so you can maintain smooth edges

> **Note**: Avoid including too much transparency around an image. Make the size of the custom shape as close to the size of the image as possible. Extra transparent pixels around the edges of the image can negatively effect the hover or click behavior near the image, especially when custom shapes overlap each other. When the actual shape area is bigger than what is visible, it can make hovering and clicking the shape more difficult and less predictable for users.

- **File formats** - Tableau doesn't support symbols that are in the Enhanced Meta File format (.emf). The shape image files can be in one of the following formats: .png, .gif, .jpg, .bmp, and .tiff.

Draw paths between marks

The Path property is available when the **Line** or **Polygon** mark type is selected in the **Marks** card drop-down menu. For more information about how to change the type of mark displayed in your viz, see **Change the Type of Mark in the View** on page 1332.

You can use the Path property in the Marks card to change the type of line mark (linear, step, or jump), or to encode data by connecting marks using a particular drawing order. You can path-encode your data using either a dimension or a measure.

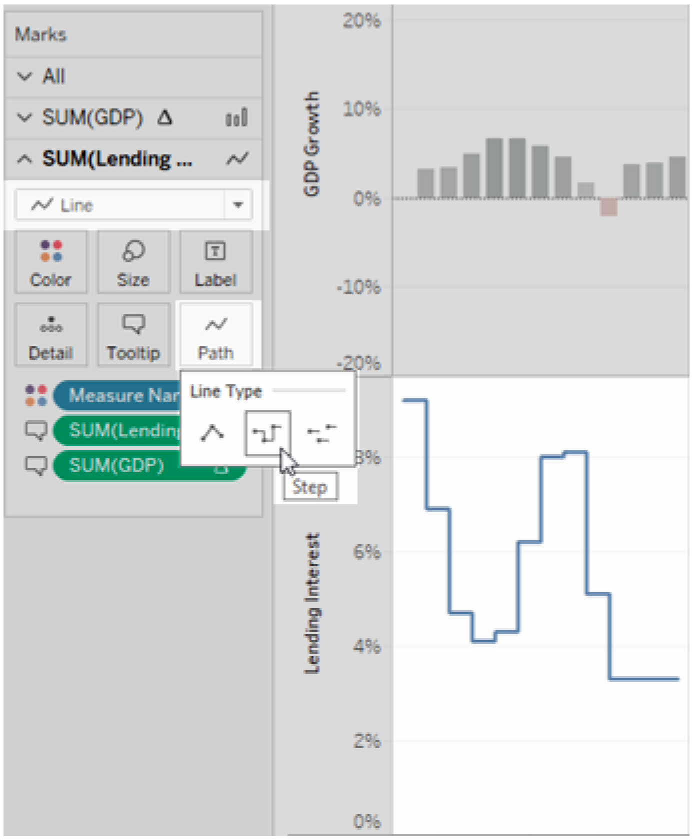
- Dimension – When you place a dimension on **Path** on the Marks card, Tableau connects the marks according to the members in the dimension. If the dimension is a date, the drawing order is driven by the date order. If the dimension includes words such as customer names or product types, the drawing order is driven by the order of the members in the data source. You can change the order that data points are connected by changing the sort order of the members. For more information, see **Sort Data in a Visualization** on page 1447.
- Measure – When you place a measure on **Path** on the **Marks** card, Tableau connects the marks according to the values of the measure. The measure can be aggregated or disaggregated.

Change the line type (linear, step, jump)

When the mark type is set to be a line (Automatic or Line), you can click the **Path** property in the Marks card to change the line type.

Use these line types for numeric data that remains constant for periods of time, with noticeable changes or deltas – such as account balances, inventory levels, or interest rates. Step lines work well for emphasizing the magnitude of change. Jump lines help to emphasize the duration of change between data points.

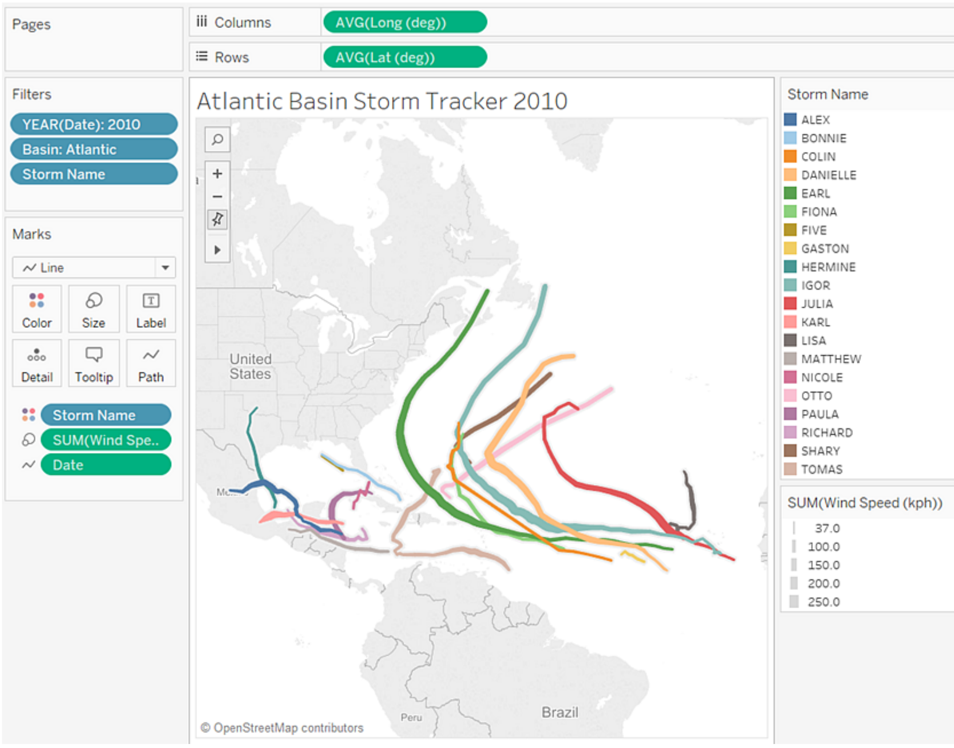
1. Click **Path** in the Marks card.
2. Select **Linear**, **Step**, or **Jump** to change the line type.



Create a path-encoded view

To create a useful path-encoded view, your data table should contain at least one measure. You can't create a path that connects only categorical data (dimensions).

The view below was created using storm data from the Atlantic basin in 2010. The view uses line marks with the path determined by the date of the storm. This lets you see the path of the storm. By placing the continuous date on **Path** on the **Marks** card, this tells Tableau to draw the lines in chronological order. For more information, see **Create Maps that Show a Path Over Time in Tableau** on page 1903 and **Create Maps that Show Paths Between Origins and Destinations in Tableau** on page 1917.



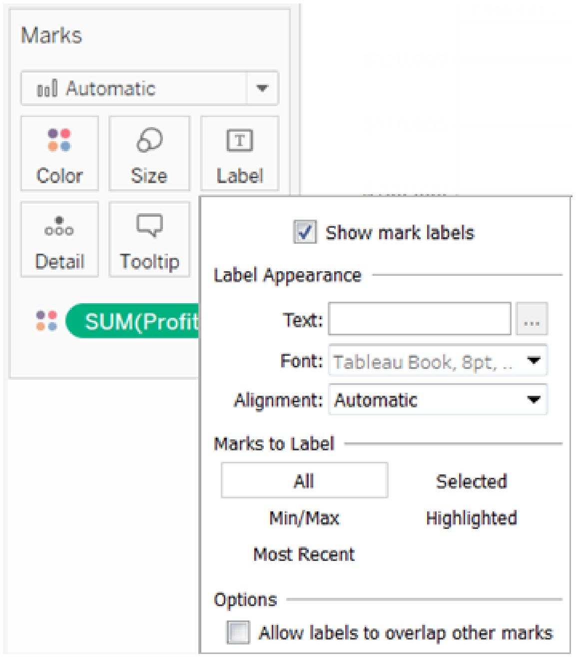
Show, Hide, and Format Mark Labels

You can add labels to the data points in your visualization. For example, in a view that shows product category sales over time as a line, you can label sales next to each point along the lines.

Show mark labels

To show mark labels in a viz:

- On the Marks card, click **Label**, and then select **Show mark labels**.



- To add another field to the mark labels, drag that field to **Label** on the Marks card.

If the marks are dense, you may not see labels for all the marks unless you check the option **Allow labels to overlap other marks**. This is not on by default as it can often be illegible if there are too many marks in close proximity.

Hide mark labels

To hide mark labels in a viz:

- On the Marks card, click **Label**, and then clear **Show mark labels**.

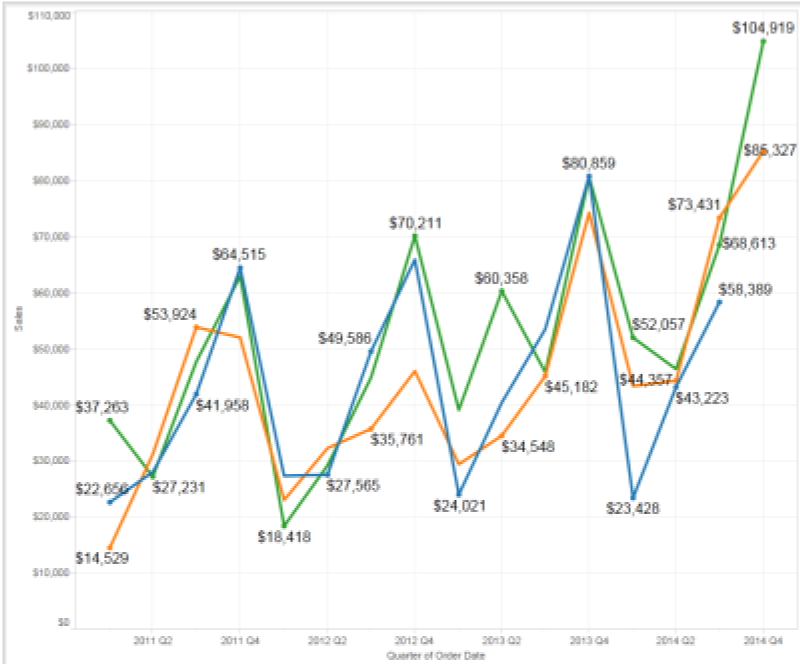
Choose which marks to label

After you show mark labels in a worksheet, you can specify which marks to label.

To specify which marks to label:

- On the Marks card, click **Label**.
- In the dialog box that opens, under Marks to Label, select one of the following options:
  - **All**

    Label all marks in the view.



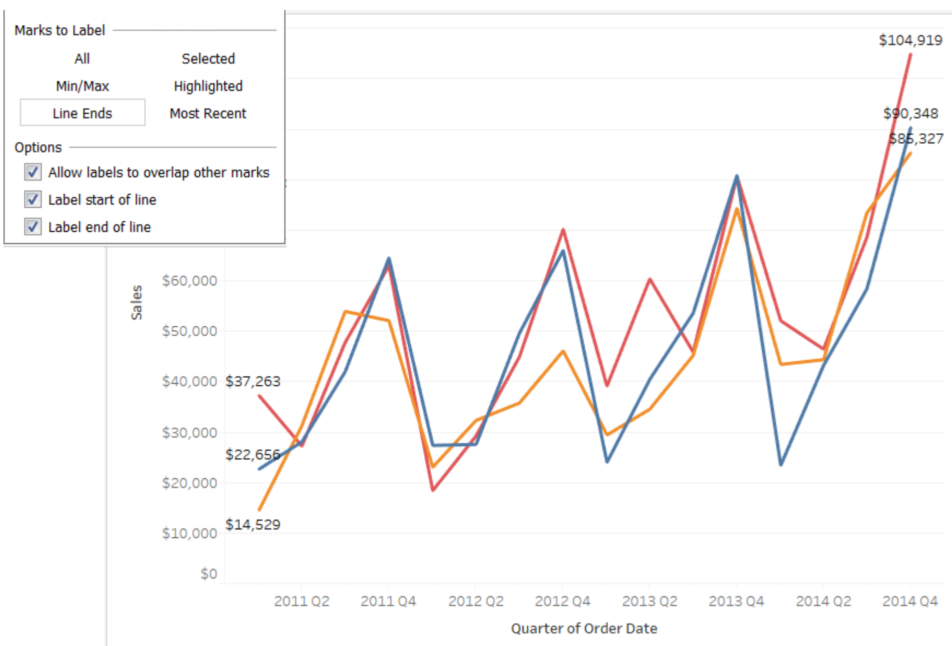
- **Min/Max**

  Label only the minimum and maximum values for a field in the view. When you select this option, you must specify a scope and field to label by.

  For example, the image below shows the mark labels scoped to pane, and specifies to show the minimum and maximum sum of sales values. This means that the minimum and maximum sales are labeled for each pane in the view. In this case, there are four panes in the view.

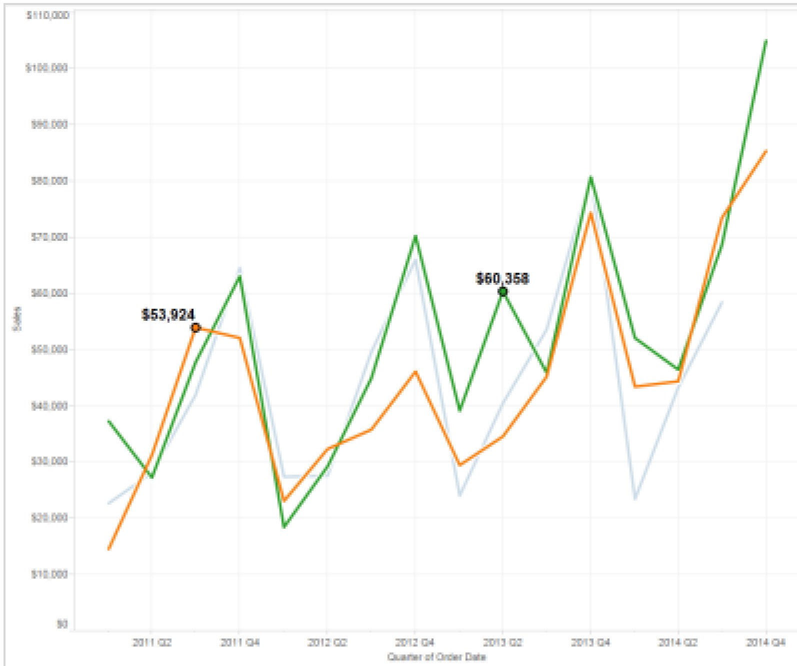
- **Line Ends**

  You can label the start or the end of all lines in the view, or both.



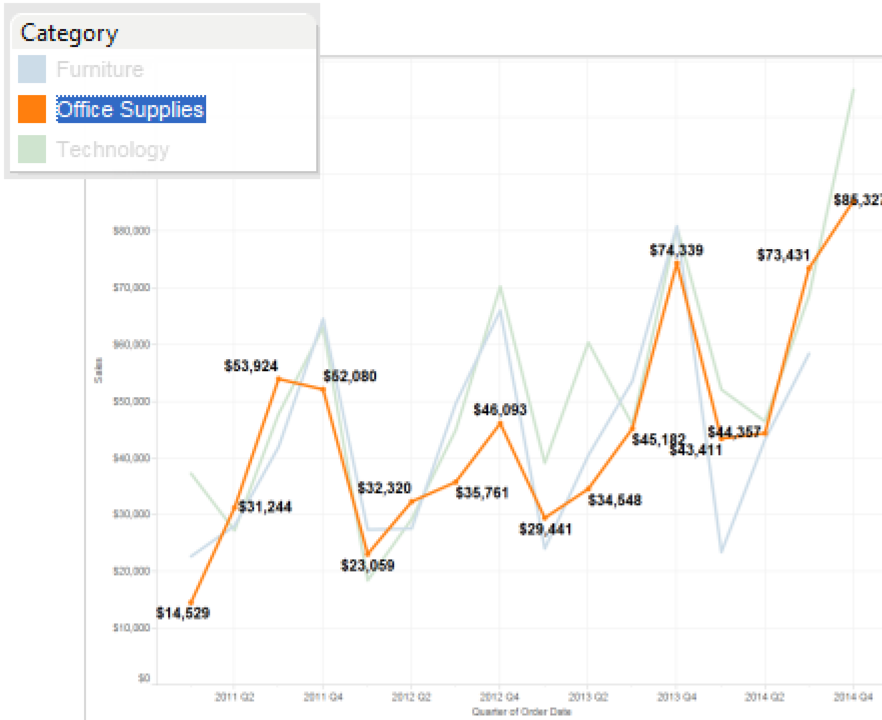
- **Selected**

Label only the selected marks in the view. Labels appear when you select one or more marks in the view.



- **Highlighted**

  Label only the highlighted marks in the view. Labels appear when you select a member in a legend (for example, Office Supplies, in the following image), when you select one or more marks in the view, or when you select marks using the Highlighter. For more information about how to use the Highlighter, see **Highlight Data Points in Context** on page 1502.

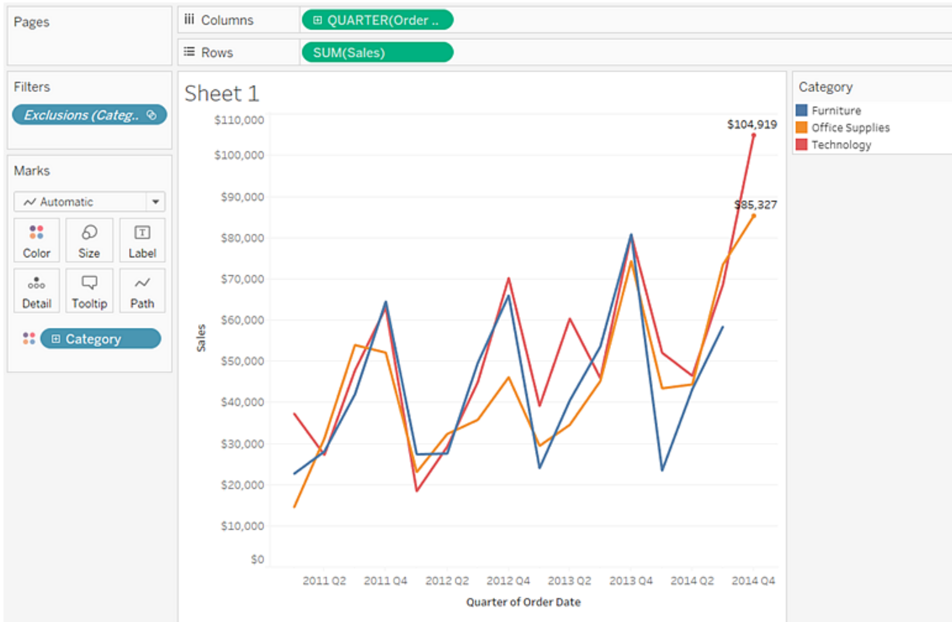


- **Most Recent**

  Label the most recent marks in the view. When there is a time or date field in the view, you can label all marks that correspond to the most recent date or time in the view. When you select this option, you can specify a scope for the labels in the view.

  Marks labeled are determined by the level of detail of the view.

  For example, the following image shows labels for each category, because the **Category** field is on **Color** on the **Marks** card. There is no label for the Furniture category, because the most recent mark in that category is excluded from the view.



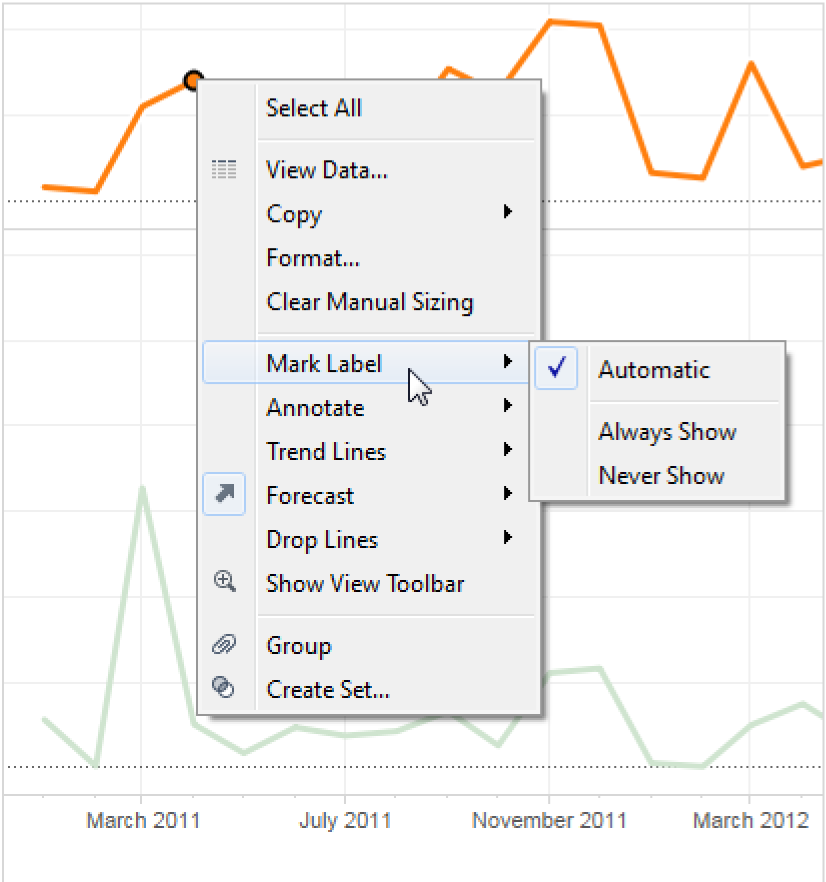
Show and hide individual mark labels

Rather than showing all mark labels or dynamically showing labels based on the view, you might want to show labels for only individual marks. You can use mark labels to call out the values of specific marks of interest, as well as hide overlapping mark labels.

To show or hide individual mark labels:

- In a worksheet, right-click (control-click on Mac) the mark you want to show or hide a mark label for, select **Mark Label**, and then select one of the following options:
  - **Automatic** - select this option to turn the label on and off depending on the view and the settings in the Label drop-down menu.
  - **Always Show** - select this option to show the mark label even when it would otherwise be hidden (based on the settings in the Label drop-down menu).
  - **Never Show** - select this option to hide the mark label even when it would

otherwise be shown (based on the settings in the Label drop-down menu).

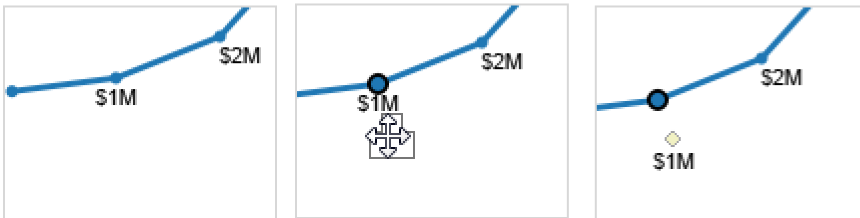


Move mark labels

After you show a mark label in a view, you can reposition it to best fit your view and presentation. For example, in a stacked bar chart, the mark labels are automatically placed in the center of each bar. However, you may want to stagger the labels so that the longer ones don't overlap.

To move a mark label:

1. Select the mark whose mark label you want to move.
2. Click and drag the move handle to a new location.



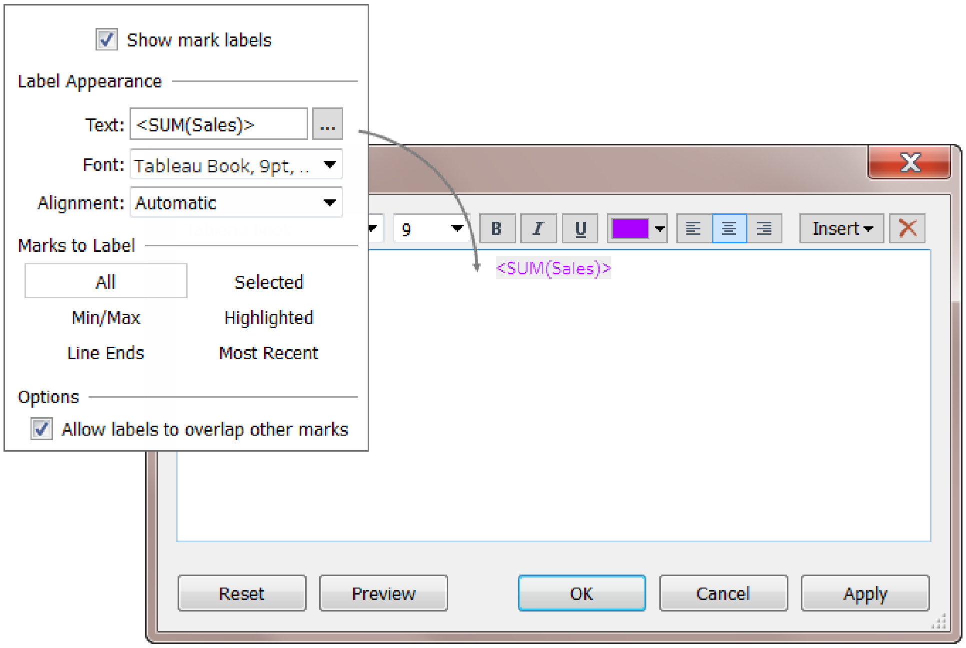
You can also change the alignment of marks. See **Edit the label alignment:** on page 1388in the next section on formatting mark labels.

Format mark labels

When you select to show mark labels in the view, there are several formatting options to help you adjust the appearance of the labels. You can customize the text, adjust the font properties, and set an alignment for all labels.

Edit the label text:

1. On the Marks card, click **Label**.
2. In the dialog box that opens, under Label Appearance, click the text button.



3. Edit the text and then click **OK**.

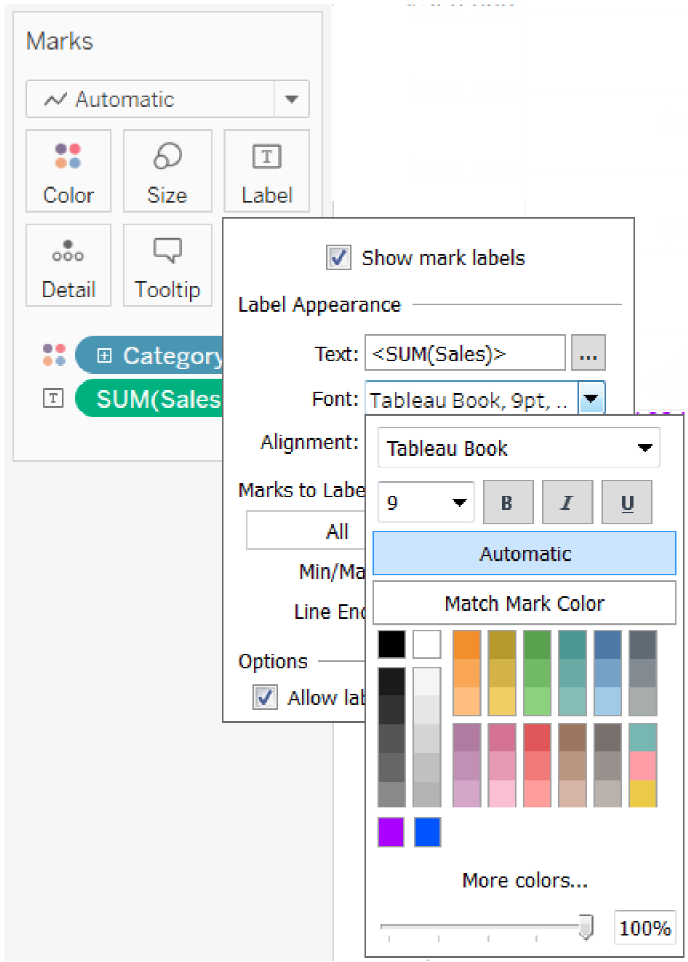
To include other fields in the label, drag them to **Label** on the Marks card.

Edit the label font:

1. On the Marks card, click **Label**.
2. In the dialog box that opens, under Label Appearance, click the Font drop-down.
3. In the Font drop-down menu, you can do the following:
   - Select a font type, size, and emphasis.
   - Adjust the opacity of the labels by moving the slider at the bottom of the menu.
   - Select a color for the labels

     To select a specific color, Click a color from the options.

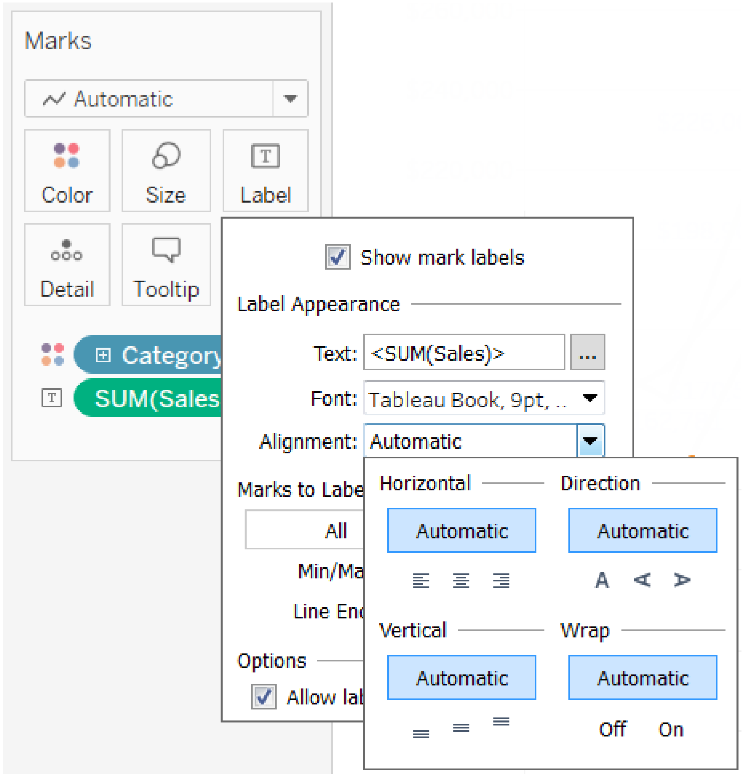
To match labels to the color of their marks, click **Match Mark Color**.



4. When finished, click **OK**.

Edit the label alignment:

1. On the Marks card, click **Label**.
2. In the dialog box that opens, under Label Appearance, click the Alignment drop-down.
3. In the Alignment drop-down menu, you can do the following:
    - Select to align the label horizontally (left, center, right).
    - Select to change the direction of the text.
    - Select to align the label vertically (bottom, middle, top).
    - Select to wrap the text for all labels in the viz.



4. When finished, click **OK**.

Overlap other marks and labels

By default, mark labels do not overlap other marks or labels in the view. You can show all labels in the view, even the ones that overlap other marks and labels.

To overlap other marks in the view:

- On the Marks card, click **Label** , and then, under the Options section, select **Allow labels to overlap other marks**.

See Also

Add Annotations on page 1465

Create Aliases to Rename Members in the View on page 1161

Move Marks

In a dual-axis view that contains many marks, some marks can become hidden behind others, making it difficult to see details in your view. Tableau provides a **Move Marks** option to move selected marks forward or backward, depending on the axis that you select.

To move marks forward or backward, right-click on one of your axes and select one of the following options:

- **Move Marks to Front**
- **Move marks to Back**

Example - Move marks forward

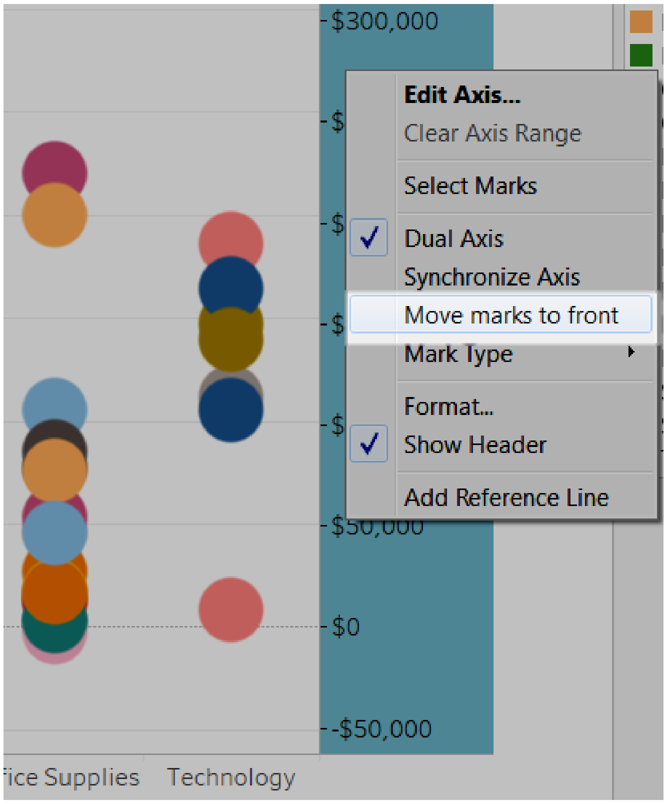
In this example, suppose you have a view that uses the **Circle** mark type on a dual axis and shows department **Sales** and **Profit** for each **Category**.

The shape of each mark represents **Sales** and **Profit**, while **Category** is encoded as **Color**. The right axis represents the **Sales** mark, while the left represents the **Profit** mark.

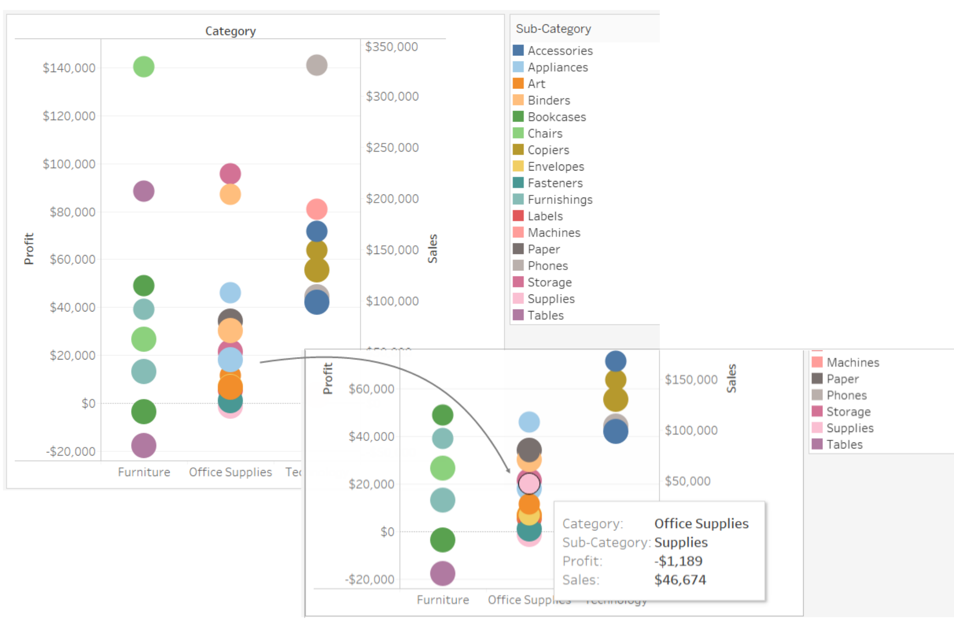
Because the profit marks are in front, it is difficult to see the **Sales** marks in the **Office Supplies** category.

To move the **Sales** marks in front of the **Profit** marks, do the following:

1. Right-click on the **Sales** axis.
2. Select **Move marks to front** from the context menu.



By moving the **Sales** mark to the front, you can now see that rubber bands are below $100,000 in sales, whereas they were nearly invisible before.



Stack Marks

Stacking marks is relevant when your data view includes numeric axes. That is, at least one measure has been placed on the **Rows** or **Columns** shelves. When marks are stacked, they are drawn cumulatively along an axis. When marks are not stacked, they are drawn independently along an axis. That is, they are overlapping.

Stacking marks is particularly useful for bar charts which is why Tableau automatically stacks bars. You might find that stacking marks is useful for other marks such as lines as well.

You can control whether marks are stacked or overlapping in any given view by selecting the **Analysis** > **Stack Marks** menu item.

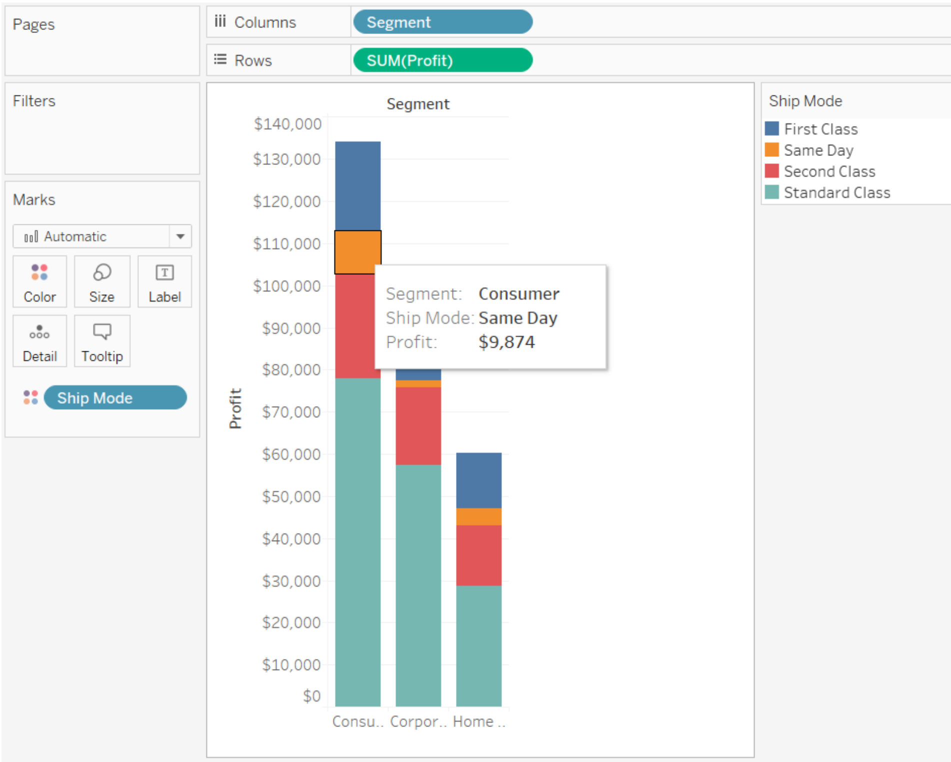
You can either allow Tableau to automatically select whether the marks are stacked or you can specify **On** or **Off**. The default setting is **Automatic**. When you are in automatic mode, the Stack Marks menu shows whether stacked marks is on or off.

Selecting **On** or **Off** on the **Stack Marks** menu, switches into manual mode. Your selection remains throughout any changes you make to the view.

The following examples illustrate stacking marks.

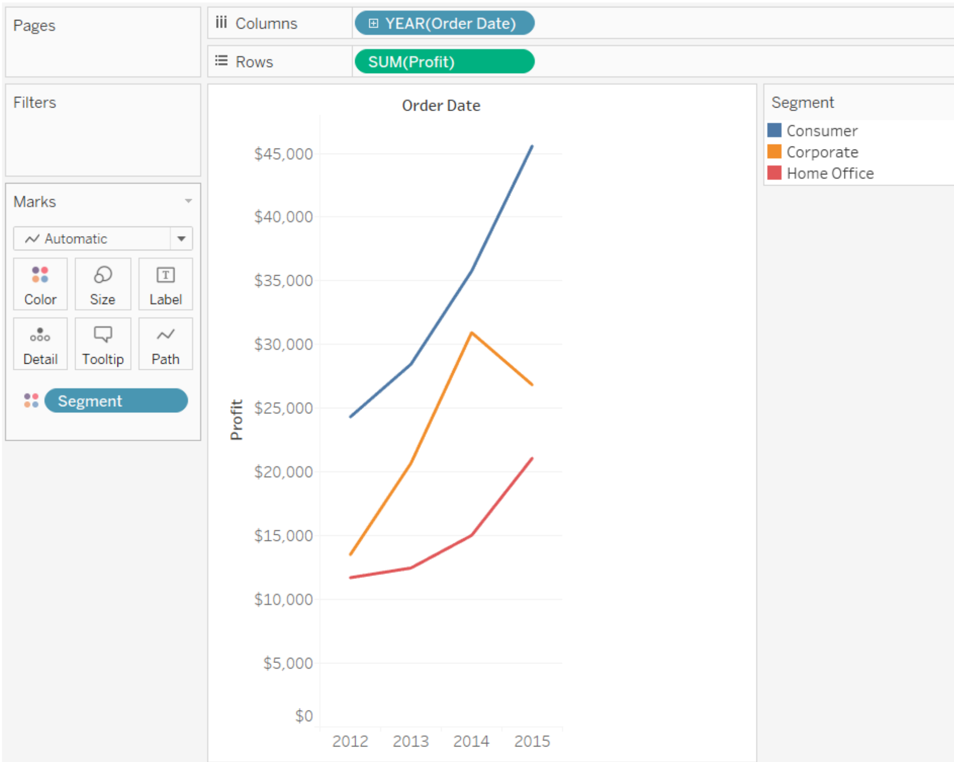
Example: Stacking Bars

Consider the stacked bars view shown below. It was created by placing a dimension on the **Columns** shelf, placing a measure on the **Rows** shelf, and color-encoding the data by a dimension (that is, dropping a dimension on **Color** on the Marks card).



Example: Stacking Lines

Consider the data view shown below. It was created by placing a date dimension on the **Columns** shelf, placing a measure on the **Rows** shelf, and color-encoding the data by a dimension (that is, dropping a dimension on **Color** on the Marks card). Because the mark type is Line, the marks are not automatically stacked. Instead, they are drawn independently from the horizontal axis.



Color Palettes and Effects

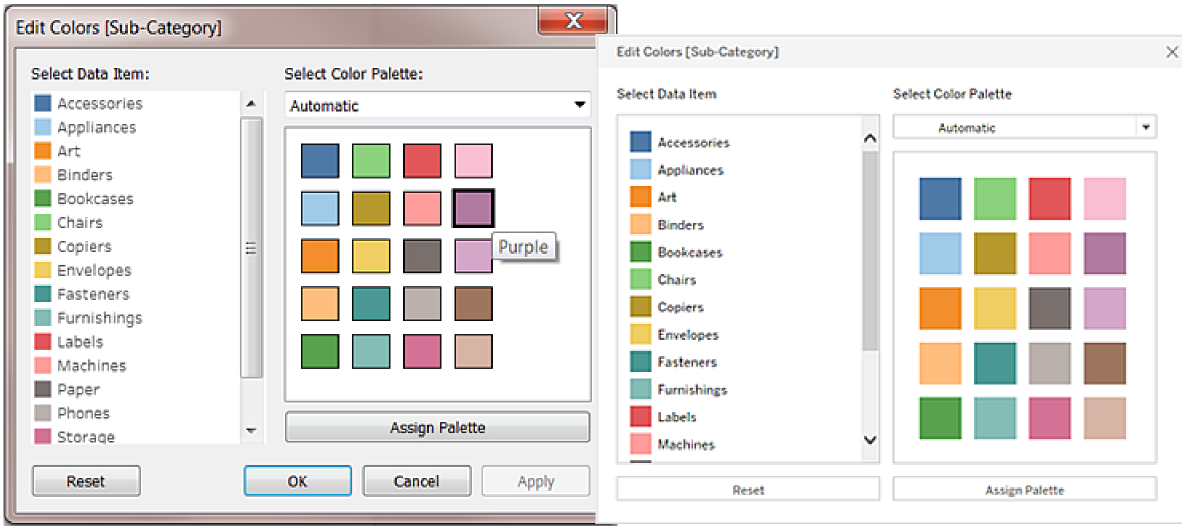
All marks have a default color, even when there are no fields on **Color** on the **Marks** card. For most marks, blue is the default color; for text, black is the default color. Also see **Assign colors to marks** on page 1357 and **Example – Multiple Fields on Color** on page 1405.

Categorical Palettes

When you drop a field with discrete values (typically a dimension) on **Color** on the **Marks** card, Tableau uses a categorical palette and assigns a color to each value of the field. Categorical palettes contain distinct colors that are appropriate for fields with values that have no inherent order, such as departments or shipping methods.

To change colors for values of a field, click in the upper-right corner of the color legend. In Tableau Desktop, select **Edit Colors** from the context menu. In Tableau Server or Tableau Online, the Edit Colors dialog opens automatically.

Tableau Desktop version	**Web version**



Change the color for a value

1. Click on an item on the left, under **Select Data Item**.
2. Click a new color in the palette on the right. In Tableau Desktop you can hover over a swatch to identify the color.
3. Repeat for as many values that you want to change.
4. In Tableau Desktop, click **OK** to exit the Edit Colors dialog box. In Tableau Server or Tableau Online, simply close the dialog box.

Select a different palette

The **Select Color Palette** drop-down list in the Edit Colors dialog box provides color palettes that you can use for discrete fields. The list contains both categorical and ordinal palettes.

At the top of the list are categorical palettes, such as *Tableau 10*. As noted above, categorical palettes are appropriate for discrete fields with no inherent order.

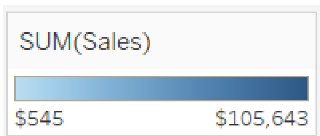
At the bottom of the list are ordinal palettes such as *Orange*. Ordinal palettes contain a range of related colors and are appropriate for fields that have an associated order, such as dates or numbers.

After you select a palette, click **Assign Palette** to automatically assign the new palette colors to the members in the field.

To return to the Automatic palette and the default color assignments, click **Reset** in the Edit Colors dialog box.

Quantitative Palettes

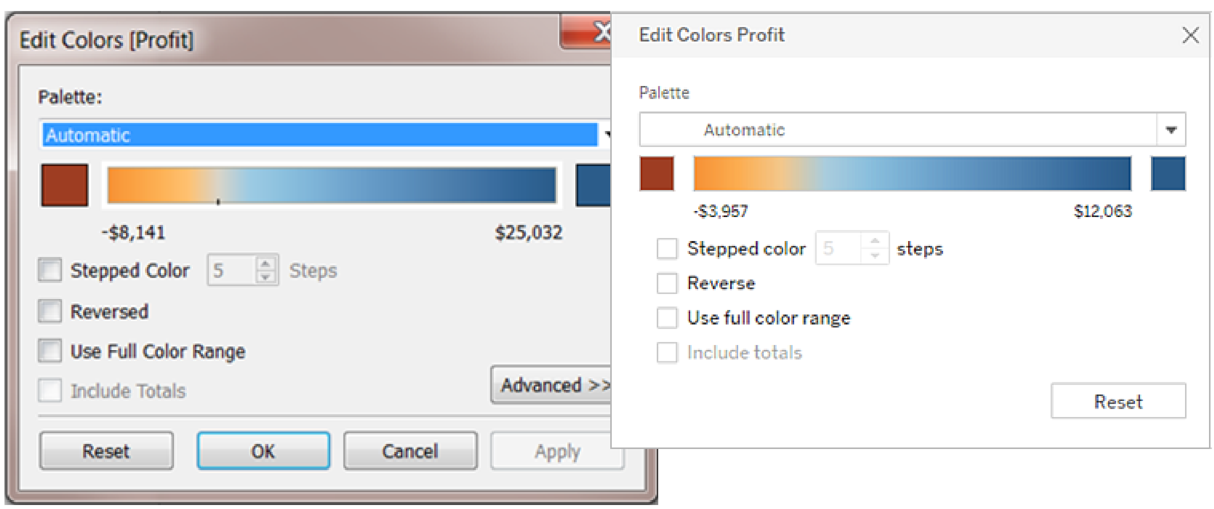
When you drop a field with continuous values on the **Marks** card (typically a measure), Tableau displays a quantitative legend with a continuous range of colors.



You can change the colors used in the range, the distribution of color, and other properties. To edit colors, click in the upper right of the color legend. In Tableau Desktop, select **Edit Colors** from the context menu. In Tableau Server or Tableau Online, the Edit Colors dialog opens automatically.

When there are both negative and positive values for the field, the default range of values will use two color ranges and the Edit Colors dialog box for the field has a square color box on either end of the range. This is known as a diverging palette.

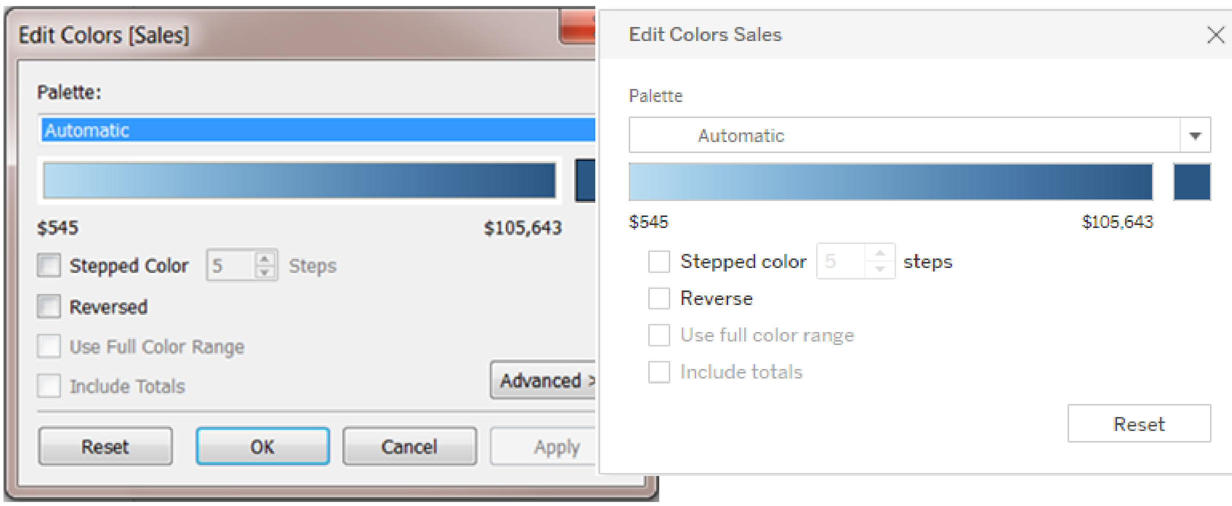
Tableau Desktop version **Web version**



Edit Colors dialog box for a diverging palette

When all values are either positive or negative, the default range of values will use a single color range and the Edit Colors dialog box for the field has a square color box only at the right end of the range. This is known as a sequential palette.

Tableau Desktop version **Web version**



Edit Colors dialog box for a sequential palette

You can specify whether Tableau uses a diverging or a sequential palette for a continuous field on **Color**, and also configure the range of colors for the field's values.

The **Palette** drop-down list provides a range of color palettes from which you can choose. There are two types of quantitative palettes available for continuous fields:

- All palettes with *Diverging* in the name are diverging quantitative palettes—for example, *Orange-Blue Diverging*. You can choose a diverging palette for any continuous field—it isn't necessary for the range of values to contain both positive and negative numbers.

  To change the colors for a diverging palette, click one of the square color boxes at either end of the palette spectrum. Depending on whether you are authoring in Tableau Desktop or on the web, do one of the following:

  - In Tableau Desktop, in the color configuration dialog box (which is part of your computer's operating system), select a color from the color picker or enter custom values.
  - In Tableau Server or Tableau Online, enter a custom Hex value in the **Custom Color** field. If the value isn't valid, no changes are made.

- All other palettes are sequential quantitative palettes. To change the colors for a sequential palette, click the square color box at the right end of the palette spectrum to either open the color configuration dialog box (Tableau Desktop), or enter a custom Hex value in the **Custom Color** field (Tableau Server or Tableau Online).

Options for quantitative palettes

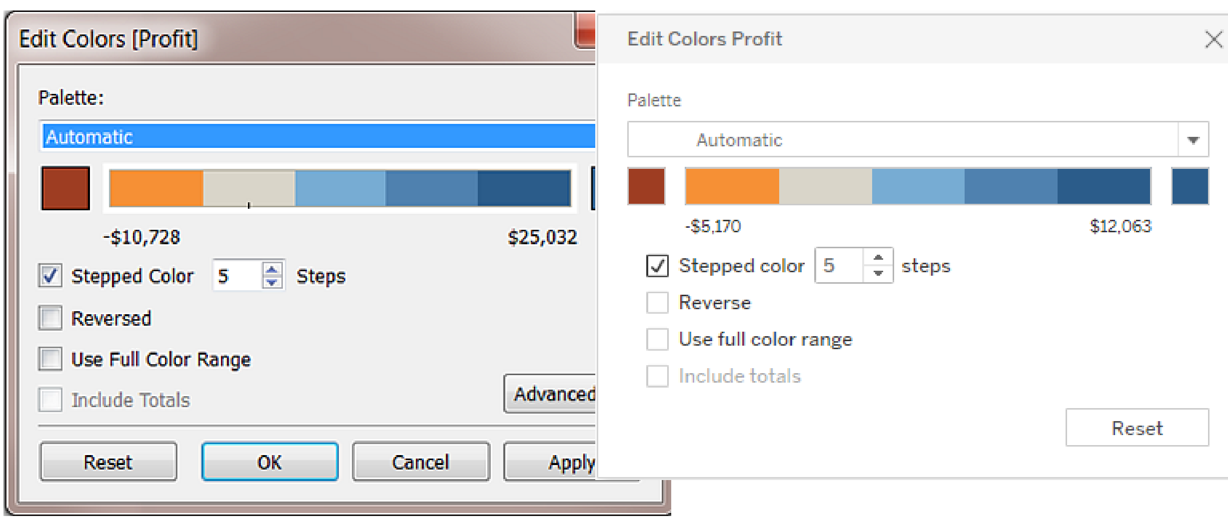
The following options are available in the Edit Colors dialog box for a continuous field.

> **Note**: Options differ where noted for Tableau Server and Tableau Online.

Stepped Color

Select **Stepped Color** to group values into uniform bins, where each bin is associated with a color. Use the spin control to specify how many steps (bins) to create. For example, for a range of values from 0 to 100 you could specify five steps to sort values into five bins (0-20, 20-40, etc.).

Tableau Desktop version | **Web version**



If a diverging color palette is selected, the point where the palette transitions between colors is shown on the color ramp with a small black tick mark. When the number of steps is odd, the mark is placed in the middle of the transitional step. When the number of steps is even, the mark is placed at the boundary between the steps where the color changes.

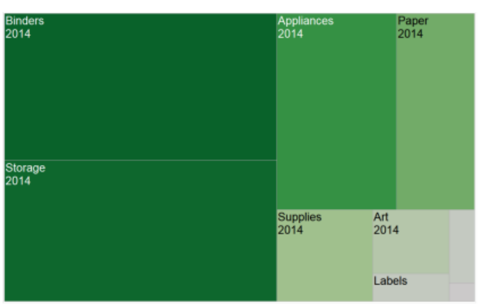
Reversed

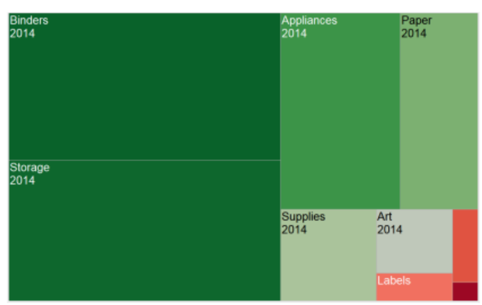
Select **Reversed** to invert the order of colors in the range. For example, if you want lower values to have a darker intensity in a sequential palette, reverse the palette. For a diverging palette, reversing the color palette means swapping the two colors in the palette, in addition to inverting the shades within each color range.

Use Full Color Range

With a diverging (two-color) palette, you can select to **Use Full Color Range**. Tableau assigns both the starting number and the ending number a full intensity for both color ranges. So if the range is from -10 to 100, the color representing negative numbers will be adjusted to change in shade much more quickly than the color representing positive numbers. If you don't select **Use Full Color Range**, Tableau assigns the color intensity as if the range of values was from -100 to 100, so that the change in shade is the same on both sides of zero. This means there will be much less change on the negative side, where actual values only range from -10 to 0, than on the positive side, where values range from 0 to 100.

The image on the left below shows a red-green diverging color palette for values from -858 to 72,986. Without using the full color range, -858 (associated with the small box at the lower right of the chart) shows as gray, because -858 is only about 1% as far to the negative side as 72,986 is to the positive side. When the full color range is used, as in the image on the right, -858 shows as a dark red, equal in intensity to the maximum positive value.





Include totals

Select **Include Totals** to include totals, sub-totals and grand totals in the color encoding. This option only applies when total values are included in the view.

Limit the color range

In Tableau Desktop, when you click **Advanced** in the Edit Colors dialog box, you can choose to specify the start, end, and center values for the range by selecting the check box and typing a new value into the field and the color ramp is adjusted accordingly.

The **Start** value is the lower limit in the range, the **End** value is the upper limit, and the **Center** value is where the neutral color is located on a diverging color palette.

> **Note**: This option is not currently available in Tableau Server or Tableau Online.

Reset the color range

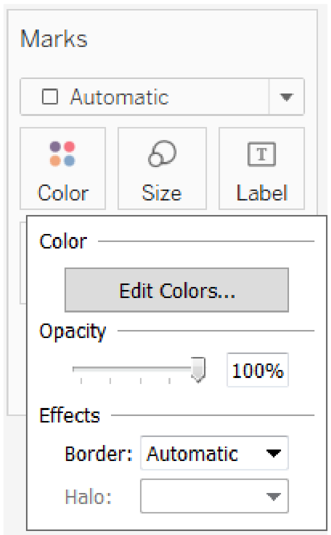
To return to the Automatic palette and the default color assignments, click **Reset** in the Edit Colors dialog box.

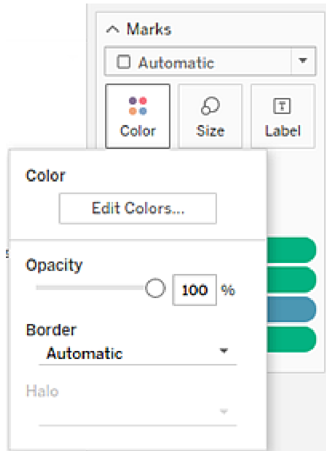
> **Note**: If you are in web authoring mode and click **Reset**, the color palette will return to the default settings. Any options that were set in the **Advanced** option will also be reset. To undo this action, you can click **Undo** in the top menu. If your changes have already been saved, you must change the Advanced options in Tableau Desktop and republish the view.

Configure Color Effects

Click the **Color** drop down on the **Marks** card to configure additional **Color** settings not related to the actual colors shown.

Tableau Desktop version	Web version





Opacity

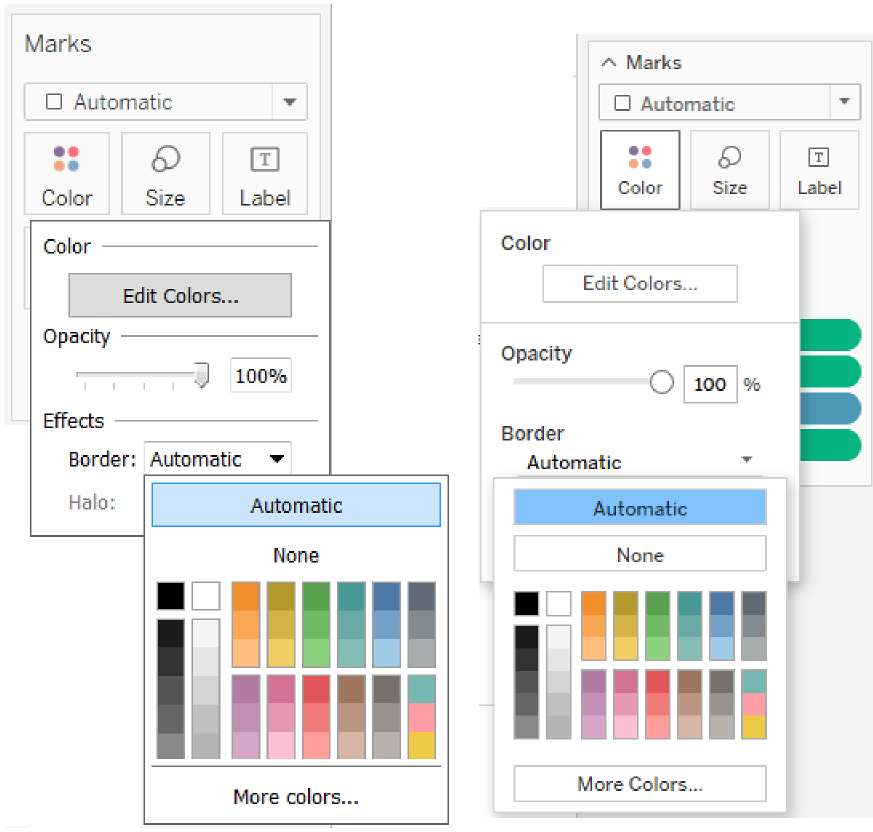
Modify the opacity of marks by moving the slider.

Adjusting opacity is especially useful in dense scatter plots or when you are looking at data overlaying a map or background image. As you slide the slider toward the left, marks become more transparent.

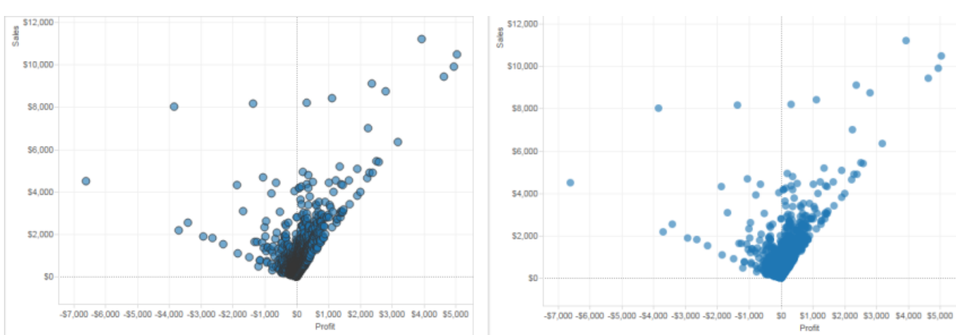
Mark borders

By default, Tableau displays all marks without a border. You can turn on mark borders for all mark types except text, line, and shape. On the **Color** drop-down control, select a mark border color.

Tableau Desktop version	**Web version**



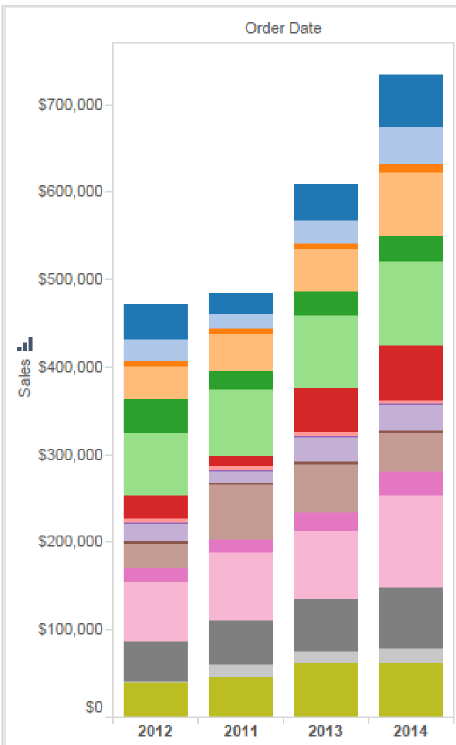
Borders can be useful for visually separating closely spaced marks. For example, the views below show a scatterplot with mark borders turned on (left) and turned off (right). When borders are turned on, marks are easier to distinguish in areas where they are tightly clustered.

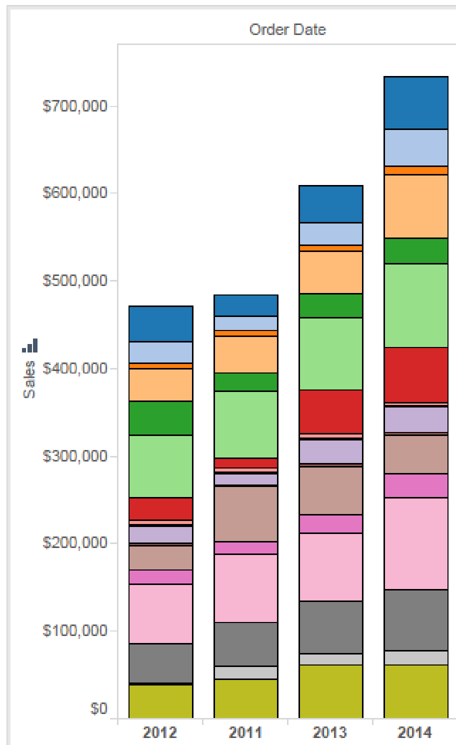


Note: You can also use the opacity setting to show the density of marks.

When you are viewing a large number of color-encoded small marks, it is usually better to leave mark borders off. Otherwise borders can dominate the view, making it difficult to see the color encoding.

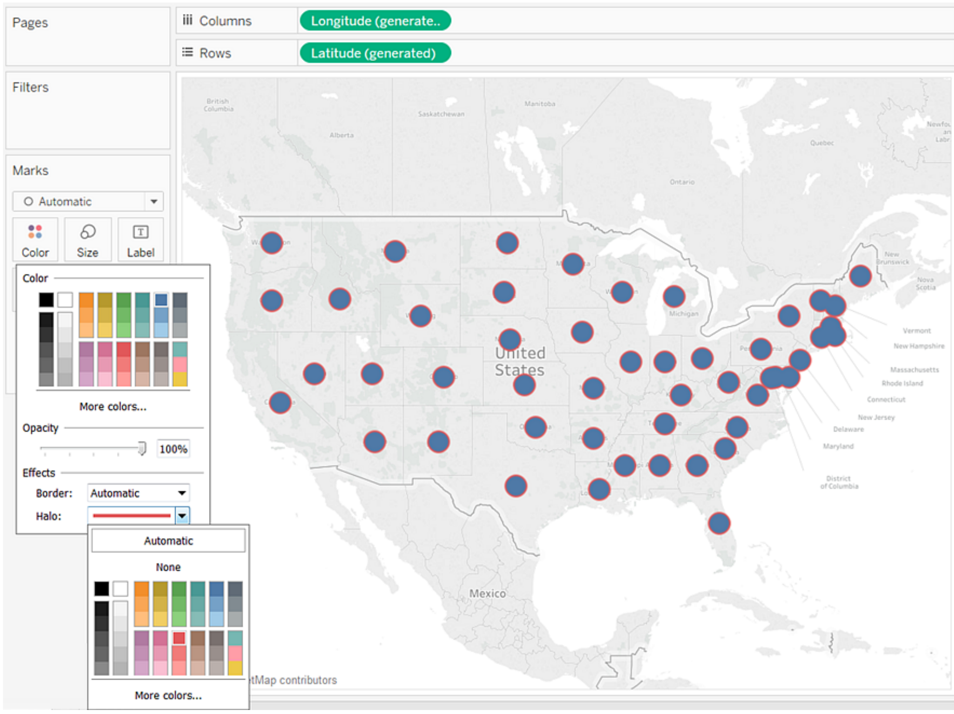
For example, the views below show bars that are segmented by a large number of color-encoded dimension members. With mark borders turned on (right), some of the narrower marks are difficult to identify by color. With borders turned off (left), the marks are easy to distinguish.





Mark halos

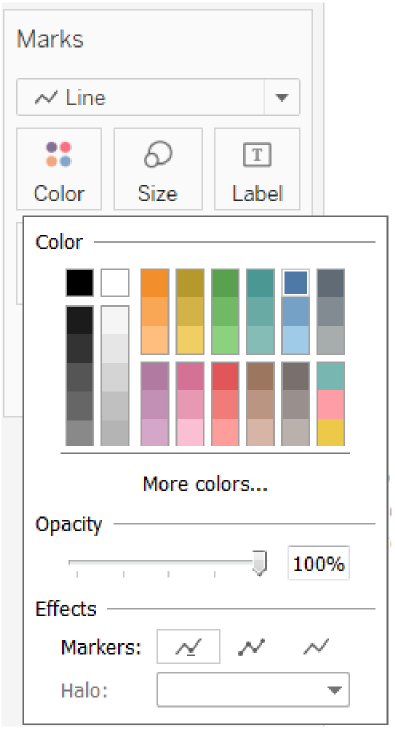
To make marks more visible against a background image or map, surround each mark with a solid contrasting color called a halo. Mark halos are available when you have a background image or a background map. On the **Color** drop-down control, select a mark halo color.



Markers

In Tableau Desktop, when you are using the Line mark type, you can add a marker effect to show or hide points along the line. You can show selected points, all points, or no points. On the **Color** drop-down control, select a marker in the **Effects** section.

Note: This option is not currently available in Tableau Server or Tableau Online.



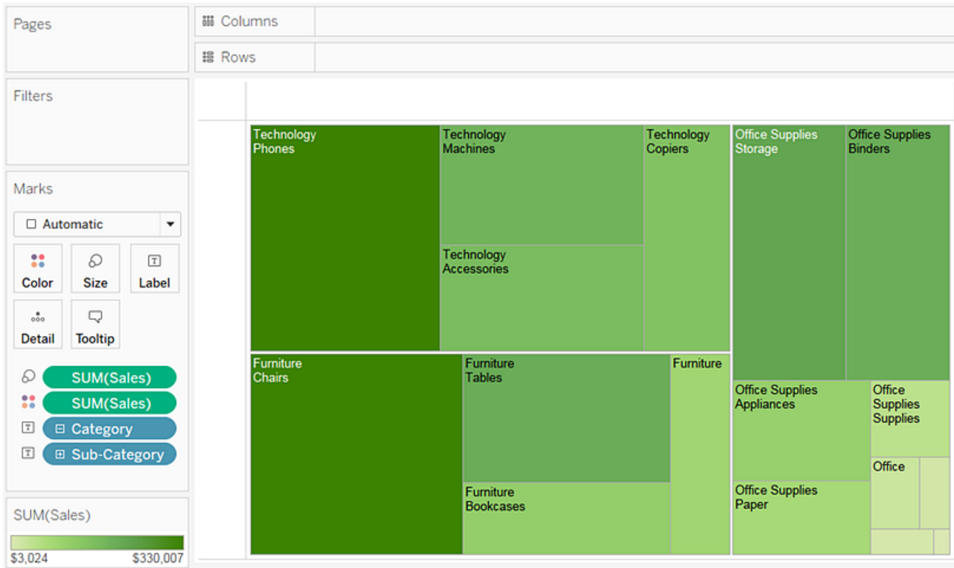
Example - Multiple Fields on Color

If you drop one field on Color and then drop a different field on Color, the second field replaces the first field. However, depending on the chart type, for example treemaps and bullet graphs, you can put multiple fields on color. You can use one field to set the hue, and the other to show gradations within that hue.

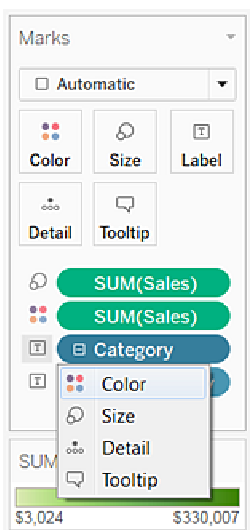
Follow these steps, using the Sample - Superstore data source, to build a treemap with two fields on Color.

1. Drag **Category** and **Sub-Category** to **Columns**.
2. Drag **Sales** to **Size** on the **Marks** card.
3. Click **Show Me** in the toolbar, then select the Treemap chart type.

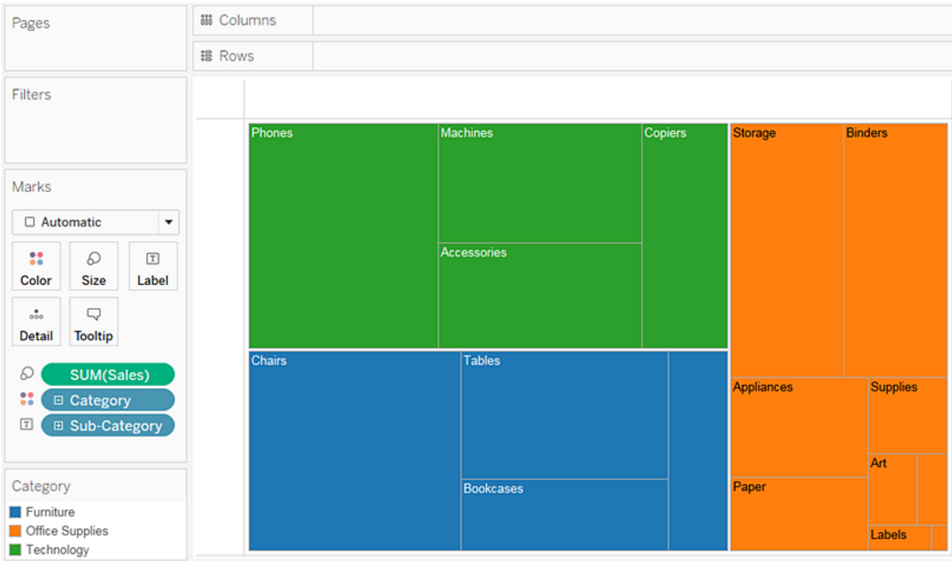
Tableau moves all fields to the **Marks** card, putting **SUM(Sales)** on both **Size** and **Color**, and **Category** and **Sub-Category** on **Label**:



4. Click the label icon to the left of **Category** on the **Marks** card and select **Color**:

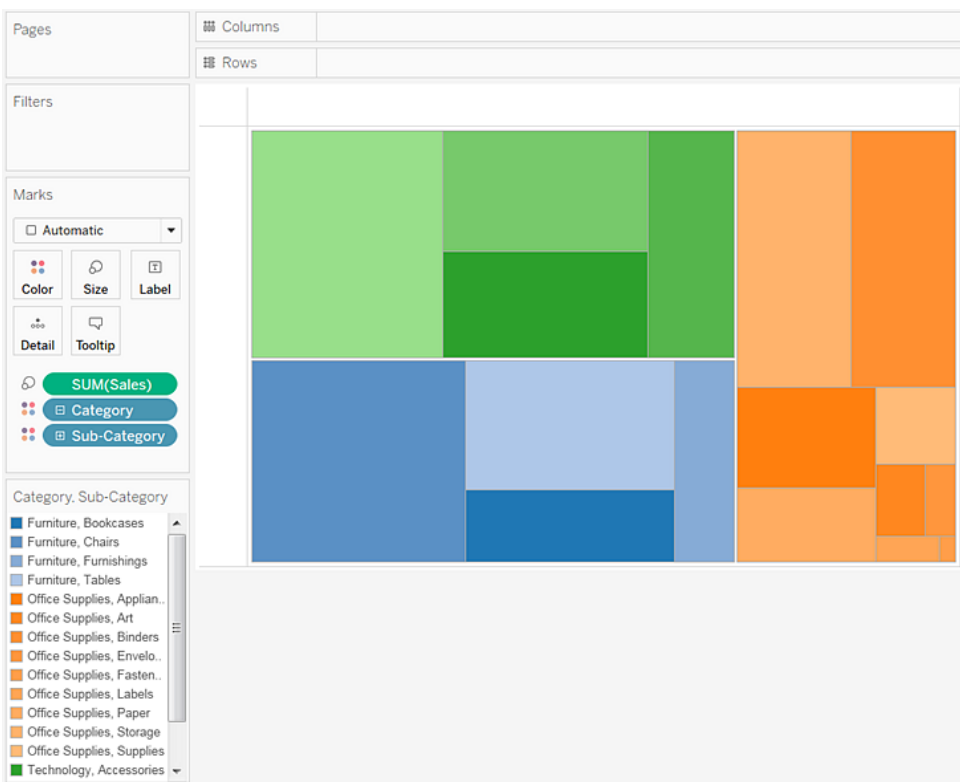


Category replaces **SUM(Sales)** on **Color**. The marks are still sized by the sum of **Sales**, but now they are colored by **Category**:



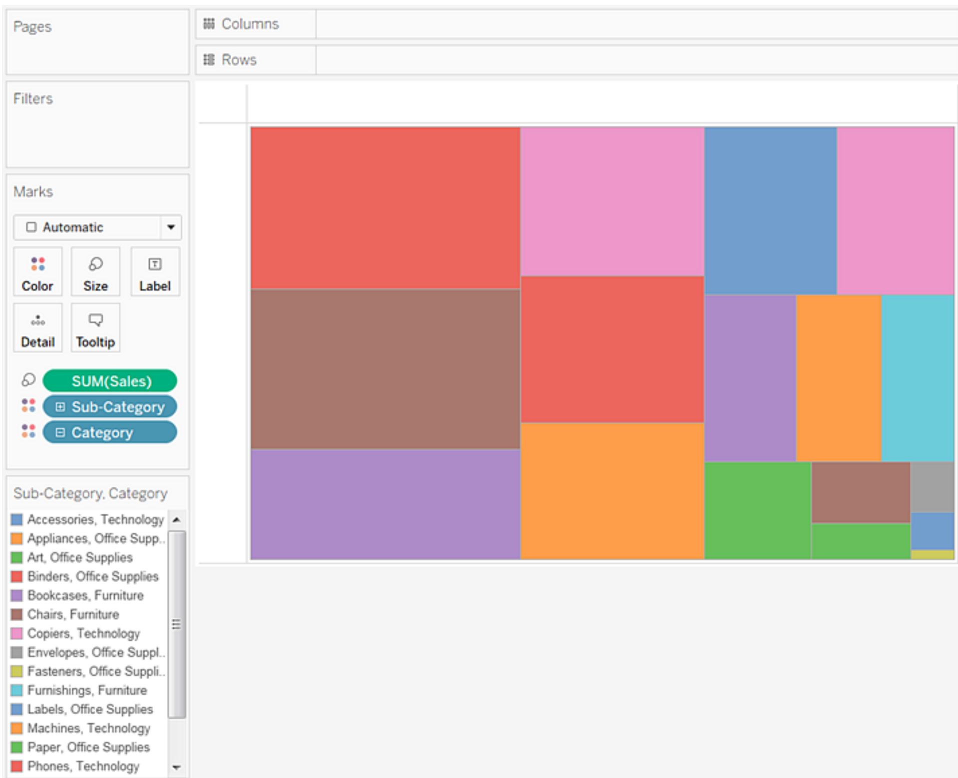
5. Click the label icon to the left of **Sub-Category** on the Marks card and select **Color**.

   Tableau uses distinct, categorical colors for the first field, **Category**, and a range of sequential shades to distinguish values for the second field, **Sub-Category**:



The size of the individual rectangles is still determined by **Sales**, per **Category** and **Sub-Category**.

The two fields on **Color** (**Category** and **Sub-Category**) are related within a hierarchy, so if you swap their positions on the **Marks** card, moving **Sub-Category** to be above **Category**, the effect is the same as if you had removed **Category** from the view altogether. The treemap changes to show a uniquely colored rectangle for each **Sub-Category**:

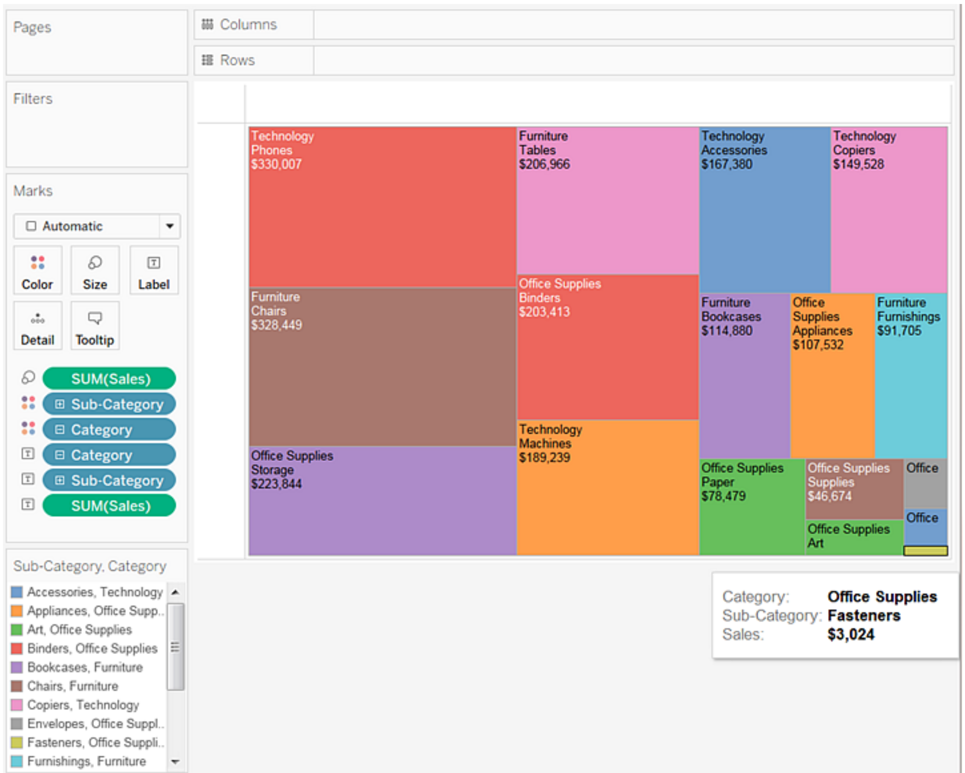


When the two fields on Color are not related within a hierarchy, you can switch the order of the fields on the Marks card so that the field that was used for categorical colors was used for sequential shades, and vice versa.

If you aren't satisfied with the colors that Tableau used, you can change them. To open the Edit Colors dialog, do one of the following:

- In Tableau Desktop, double-click the color legend.
- In Tableau Server or Tableau Online, click the drop-down arrow in the top right-hand corner of the legend.

6. Make the view more readable by adding **Category**, **Sub-Category**, and **Sales** to **Label**. Users can hover to see tooltips for any rectangle that is too small to show text by default.



Filter and Sort Data in the View

This section describes the various ways you can filter and sort data in Tableau visualizations. Read the following articles for information on how to filter data from your visualizations and across multiple worksheets and data sources, sort data in the visualization, create nested sorts, and more.

Watch a video: For a 3-minute introduction to filtering in Tableau, watch the Ways to Filter free training video. Use your tableau.com account to sign in.

For a 4-minute walk through of the various ways to sort in Tableau, watch the Sorting free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Filter Data from Your Views

Filtering is an essential part of analyzing data. This article describes the many ways you can filter data from your view. It also describes how you can display interactive filters in the view, and format filters in the view.

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Ways to Filter (2 minutes), Where Tableau Filters (4 minutes), Using the Filter Shelf (7 minutes), Interactive Filters (4 minutes), and Additional filtering topics (7 minutes). Use your tableau.com account to sign in.

Filtering Order of Operations

Before you begin filtering data in Tableau, it's important to understand the order in which Tableau executes filters in your workbook.

Tableau performs actions on your view in a very specific order; this is called the Order of Operations. Filters are executed in the following order:

1. Extract filters
2. Data source filters
3. Context filters
4. Filters on dimensions (whether on the Filters shelf or in filter cards in the view)
5. Filters on measures (whether on the Filters shelf or in filter cards in the view)

To learn more, see **Tableau's Order of Operations** on page 330.

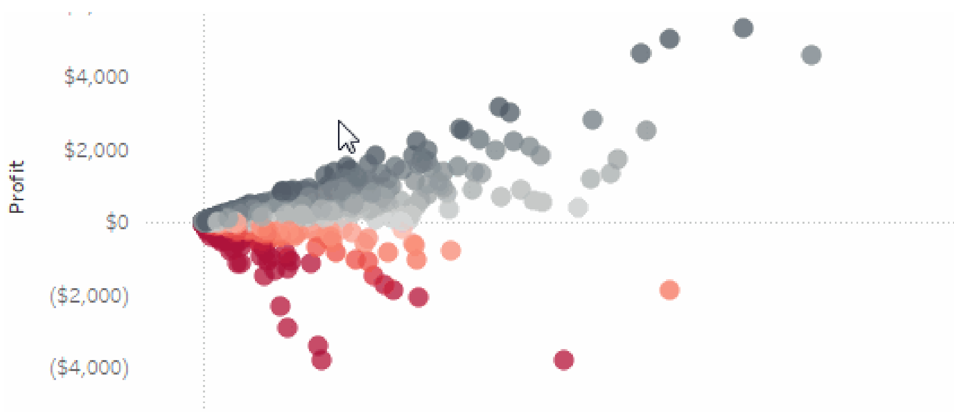
Note: When you drag a discrete dimension to the Filters shelf, the **Filter** dialog box offers four tabs for filtering: General, Wildcard, Condition, and Top. The settings on each of these tabs are additive starting with the General tab; what you set on each tab will affect the filter results on each tab to the right. For details on the Filter dialog box, see **Drag dimensions, measures, and date fields to the Filters shelf** on page 1413 and **Filter categorical data (dimensions)** on page 1414.

Select to keep or exclude data points in your view

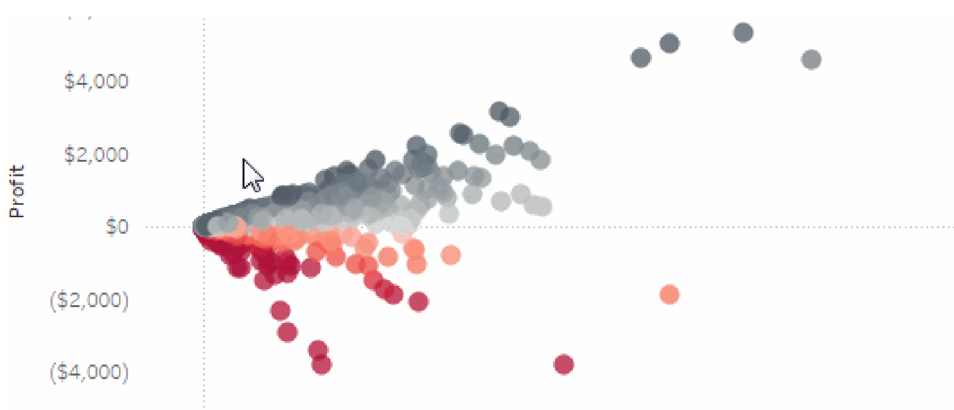
You can filter individual data points (marks), or a selection of data points from your view. For example, if you have a scatter plot with outliers, you can exclude them from the view so you can better focus on the rest of the data.

To filter marks from the view, select a single mark (data point) or click and drag in the view to select several marks. On the tooltip that appears, you can:

- Select **Keep Only** to keep only the selected marks in the view.



- Select **Exclude** to remove the selected marks from the view.



Note: These filtering options are not available if a Wildcard Match filter is already specified for the same field. See **Drag dimensions, measures, and date fields to the Filters shelf** on the facing page to learn more about Wildcard Match filters.

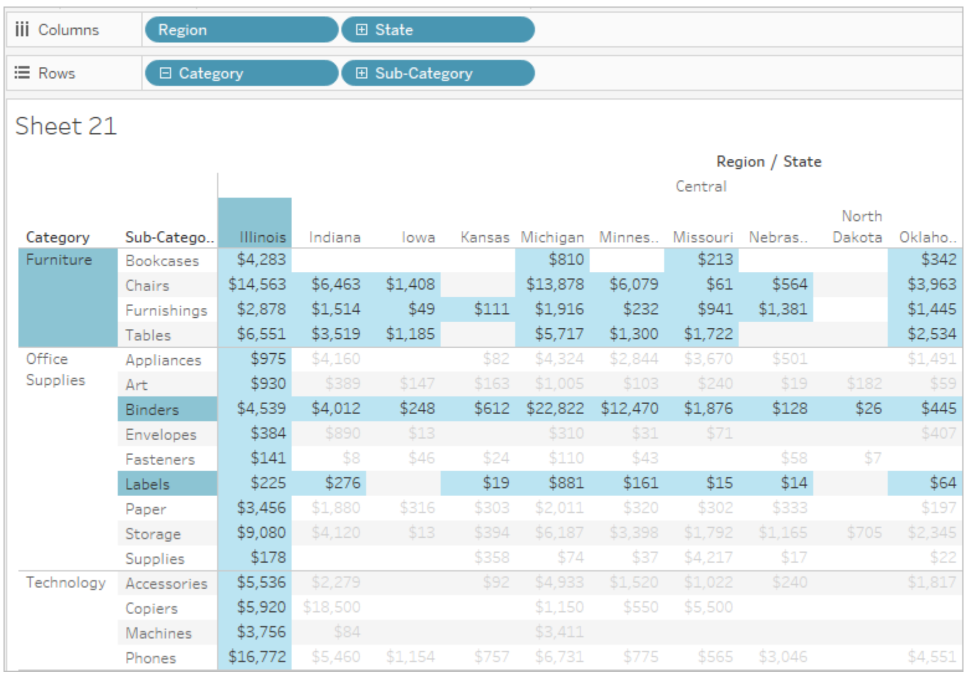
Select headers to filter data

You can also select headers to filter them from your view.

To filter entire rows or columns of data from your view, select the header in the view. On the tooltip that appears, select to **Exclude** or **Keep Only** the selected data.

When you select a table header that is part of a hierarchy, all of the next level headers are also selected. For example, the view shown below consists of two unrelated dimensions placed on the **Columns** shelf, and two levels of the same hierarchy placed on the Rows shelf.

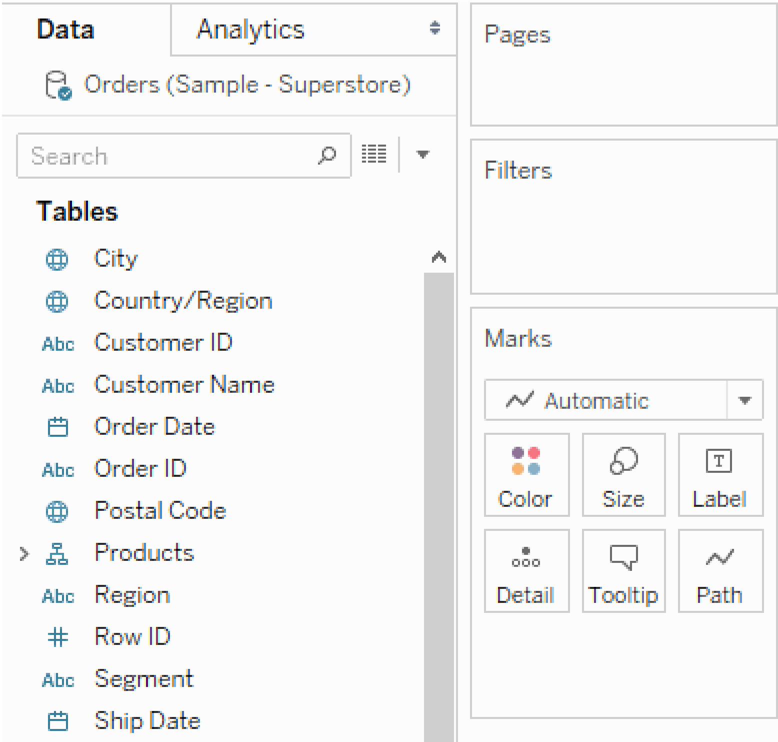
The selected row headers include the Furniture member of the **Category** dimension, and the Binders and Labels members of the **Sub-category** dimension. When Furniture is selected, all members from the next (inner) level in the hierarchy are automatically selected. In this case, that means the Bookcases, Chairs, Furnishings, and Tables members.



Category	Sub-Catego..	Illinois	Indiana	Iowa	Kansas	Michigan	Minnes..	Missouri	Nebras..	North Dakota	Oklaho..
Furniture	Bookcases	$4,283				$810		$213			$342
	Chairs	$14,563	$6,463	$1,408		$13,878	$6,079	$61	$564		$3,963
	Furnishings	$2,878	$1,514	$49	$111	$1,916	$232	$941	$1,381		$1,445
	Tables	$6,551	$3,519	$1,185		$5,717	$1,300	$1,722			$2,534
Office Supplies	Appliances	$975	$4,160		$82	$4,324	$2,844	$3,670	$501		$1,491
	Art	$930	$389	$147	$163	$1,005	$103	$240	$19	$182	$59
	Binders	$4,539	$4,012	$248	$612	$22,822	$12,470	$1,876	$128	$26	$445
	Envelopes	$384	$890	$13		$310	$31	$71			$407
	Fasteners	$141	$8	$46	$24	$110	$43		$58	$7	
	Labels	$225	$276		$19	$881	$161	$15	$14		$64
	Paper	$3,456	$1,880	$316	$303	$2,011	$320	$302	$333		$197
	Storage	$9,080	$4,120	$13	$394	$6,187	$3,398	$1,792	$1,165	$705	$2,345
	Supplies	$178			$358	$74	$37	$4,217	$17		$22
Technology	Accessories	$5,536	$2,279		$92	$4,933	$1,520	$1,022	$240		$1,817
	Copiers	$5,920	$18,500			$1,150	$550	$5,500			
	Machines	$3,756	$84			$3,411					
	Phones	$16,772	$5,460	$1,154	$757	$6,731	$775	$565	$3,046		$4,551

Drag dimensions, measures, and date fields to the Filters shelf

Another way to create a filter is to drag a field directly from the Data pane to the Filters shelf.

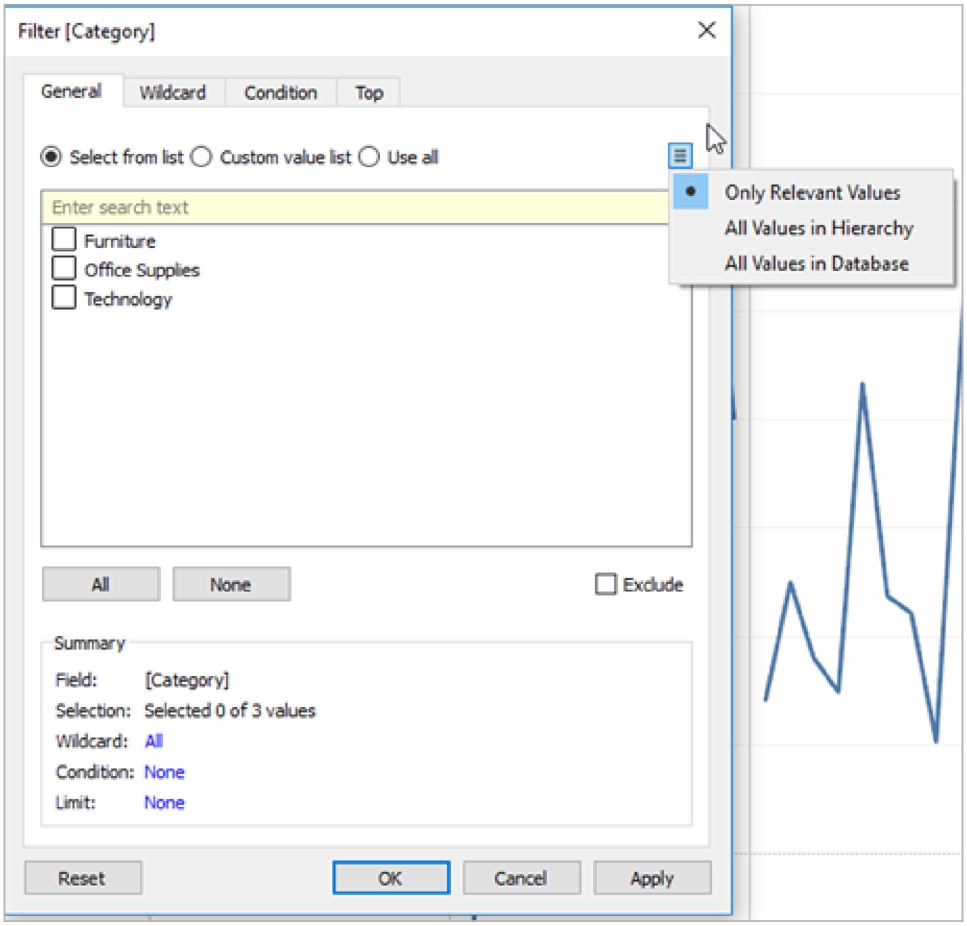


When you add a field to the Filters shelf, the Filter dialog box opens so you can define the filter. The Filter dialog box differs depending on whether you are filtering categorical data (dimensions), quantitative data (measures), or date fields.

Filter categorical data (dimensions)

Dimensions contain discrete categorical data, so filtering this type of field generally involves selecting the values to include or exclude.

When you drag a dimension from the Data pane to the Filters shelf in Tableau Desktop, the following Filter dialog box appears:



In Tableau Desktop, there are four tabs in the dialog box, and one tab in Tableau Online and Tableau Server.

- **General**: Use the General tab to select the values you want to include or exclude.
- **Wildcard** (Tableau Desktop only): Use the Wildcard tab to define a pattern to filter on. For example, when filtering on email addresses you might want to only include emails from a specific domain. You can define a wildcard filter that ends with "@gmail.com" to only include Google email addresses.
- **Condition**: Use the Condition tab in the Filter dialog box to define rules to filter by. For example, in a view showing the average Unit Price for a collection of products, you may want to only show the Products that have an average unit price that is greater than or equal to $25. You can use the built-in controls to write a condition or you can write a custom formula.
- **Top**: Use the Top tab in the Filter dialog box to define a formula that computes the data

that will be included in the view. For example, in a view that shows the average Time to Ship for a collection of products, you can decide to only show the top 15 (or bottom) products by Sales. Rather than having to define a specific range for Sales (e.g., greater than $100,000), you can define a limit (top 15) that is relative to the other members in the field (products).

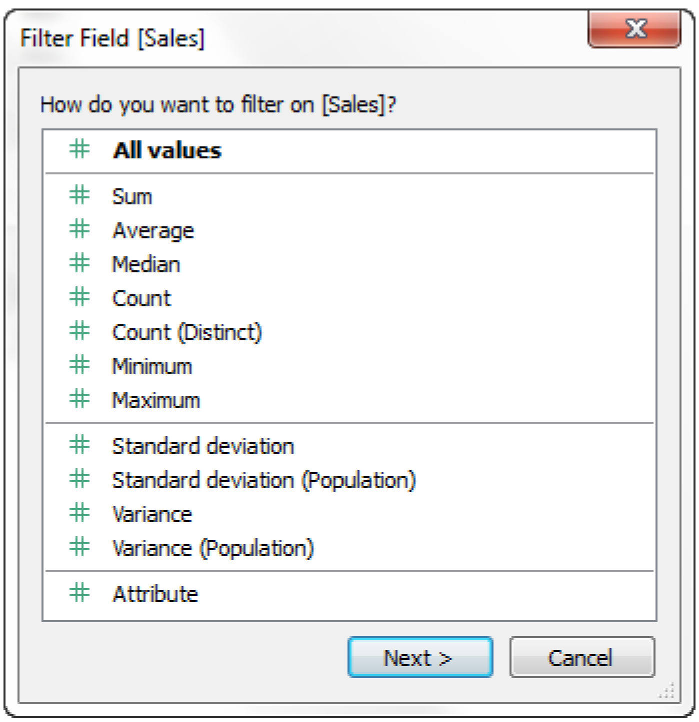
Important Note: Each tab adds additional definitions to your filter. For example, you can select to exclude values under the General tab, and also add limits under the Top tab. Selections and configurations from both tabs are applied to your filter.

At any time, you can see the definitions of your filter under Summary on the General tab.

Filter quantitative data (measures)

Measures contain quantitative data, so filtering this type of field generally involves selecting a range of values that you want to include.

When you drag a measure from the Data pane to the Filters shelf in Tableau Desktop, the following dialog box appears:



Select how you want to aggregate the field, and then click **Next**.

In the subsequent dialog box, you're given the option to create four types of quantitative filters:

Range of Values: Select the Range of Values option to specify the minimum and maximum values of the range to include in the view. The values you specify are included in the range.

At Least: Select the At Least option to include all values that are greater than or equal to a specified minimum value. This type of filter is useful when the data changes often so specifying an upper limit may not be possible.

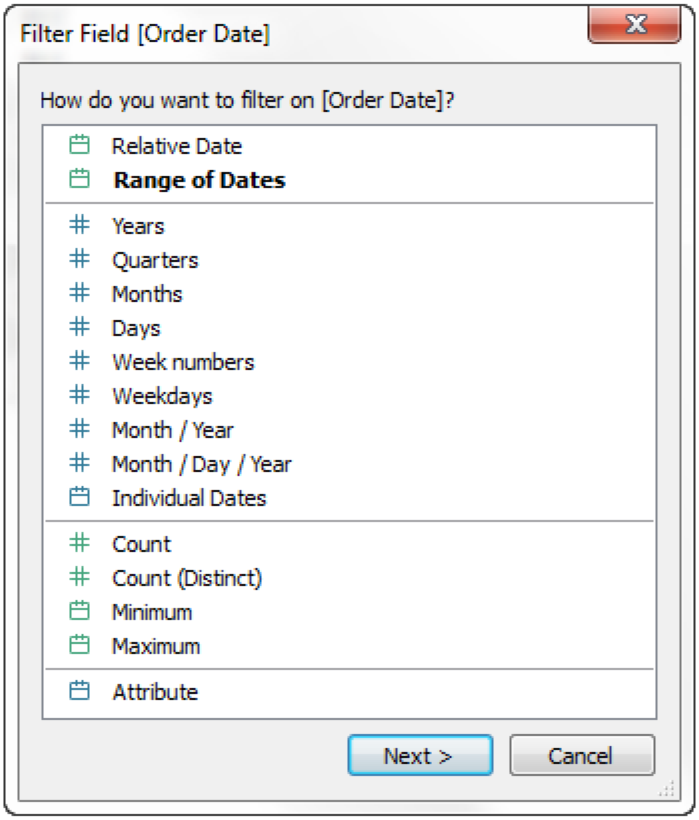
At Most: Select the At Most option to include all values that are less than or equal to a specified maximum value. This type of filter is useful when the data changes often so specifying a lower limit may not be possible.

Special: Select the Special option to filter on Null values. Include only Null values, Non-null values, or All Values.

> **Note:** If you have a large data source, filtering measures can lead to a significant degradation in performance. It is sometimes much more efficient to filter by creating a set containing the measure and then apply a filter to the set. For more information about creating sets, see **Create Sets** on page 1186.

Filter dates

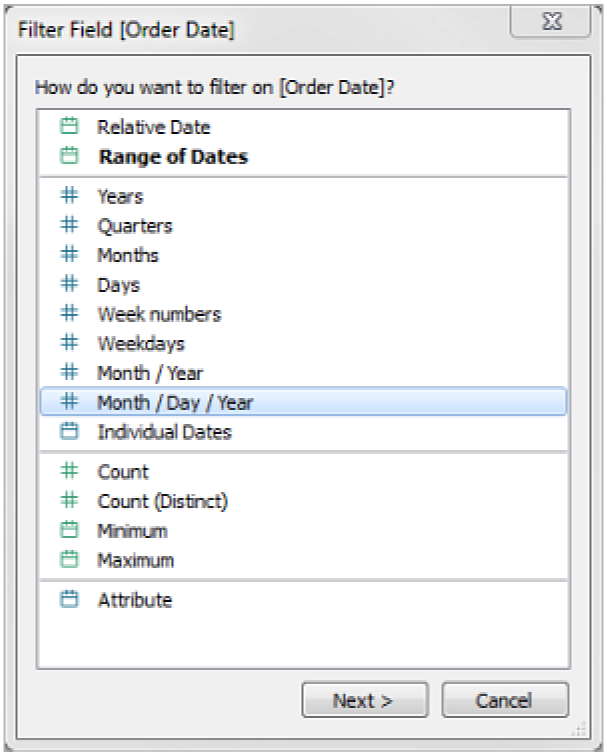
When you drag a date field from the Data pane to the Filters shelf in Tableau Desktop, the following Filter Field dialog box appears:

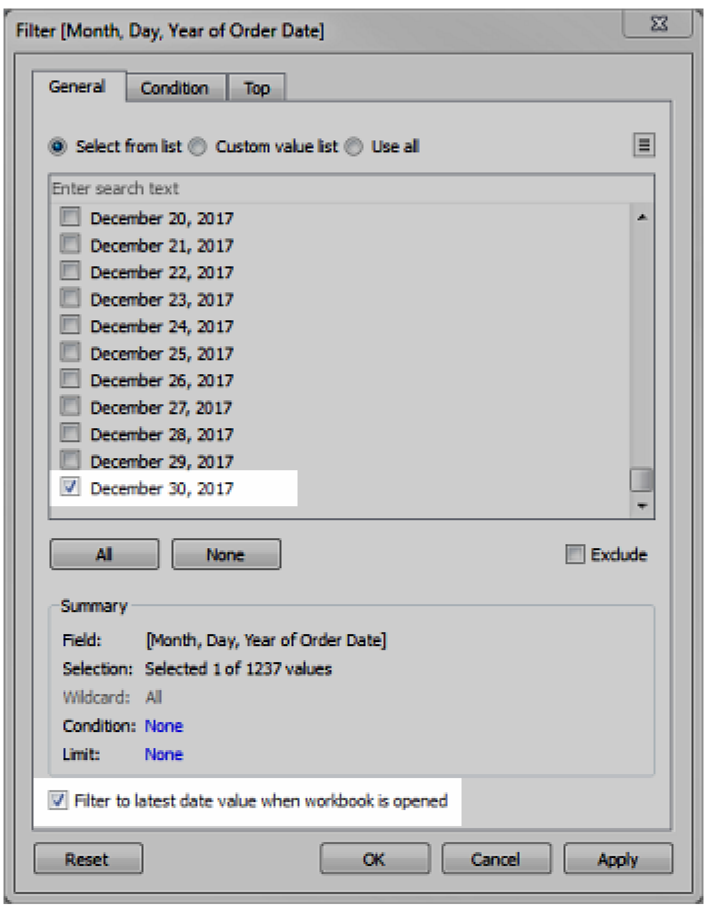


You can select whether you want to filter on a relative date; filter between a range of dates; or select discrete dates or individual dates to filter from the view.

- **Filter relative dates**: Click Relative dates to define a range of dates that updates based on the date and time you open the view. For example, you may want to see Year to Date sales, all records from the past 30 days, or bugs closed last week. Relative date filters can also be relative to a specific anchor date rather than today.
- **Filter a range of dates**: Select Range of dates to define a fixed range of dates to filter. For example, you may want to see all orders placed between March 1, 2009 and June 12, 2009.
- **Filter discrete dates**: Select a discrete date value in the dialog box if you want to include entire date levels. For example, if you select Quarters, you can choose to filter specific quarters (e.g. Q1, Q2, Q3, Q4) from your view, regardless of the year.

  Latest date preset: If you want to ensure that only the most recent date in a data source is selected in the filter when the workbook is shared or opened, select a discrete date such as Month/Day/Year or Individual Dates and then, on the General tab, select **Filter to latest date value when workbook is opened**.


Filter Field [Order Date]
How do you want to filter on [Order Date]?
Relative Date
Range of Dates
Years
Quarters
Months
Days
Week numbers
Weekdays
Month / Year
Month / Day / Year
Individual Dates
Count
Count (Distinct)
Minimum
Maximum
Attribute
Next >
Cancel


Filter [Month, Day, Year of Order Date]
General
Condition
Top
Select from list
Custom value list
Use all
Enter search text
December 20, 2017
December 21, 2017
December 22, 2017
December 23, 2017
December 24, 2017
December 25, 2017
December 26, 2017
December 27, 2017
December 28, 2017
December 29, 2017
December 30, 2017
All
None
Exclude
Summary
Field: [Month, Day, Year of Order Date]
Selection: Selected 1 of 1237 values
Wildcard: All
Condition: None
Limit: None
Filter to latest date value when workbook is opened
Reset
OK
Cancel
Apply

Notes: When you filter to the latest date value, this setting applies only to data source filters in a workbook.

In the order of operations, the latest date filter is global to the workbook, while context filters apply per worksheet. The latest date is determined just after the workbook opens for first use, after data source filters, but before context filters. At that point the date is set, and the latest date preset is used as a dimension filter.

If you are using additional filters in views (including relative date filters and context filters), the latest date value setting may result in an empty view with no data when those additional filters do not select data from the latest date in the database.

On Tableau Server and Tableau Online, presets are applied when the view first loads in the browser, but not when the browser or data is refreshed.

- **Filter individual dates**: Select Individual dates to filter specific dates from your view.
- **Additional date filter options**: When you select Relative dates or Range of dates, the Filter dialog box opens. In that dialog box, you can define a Starting date or Ending date. You can also select Special to include null dates, non-null dates, or all dates.

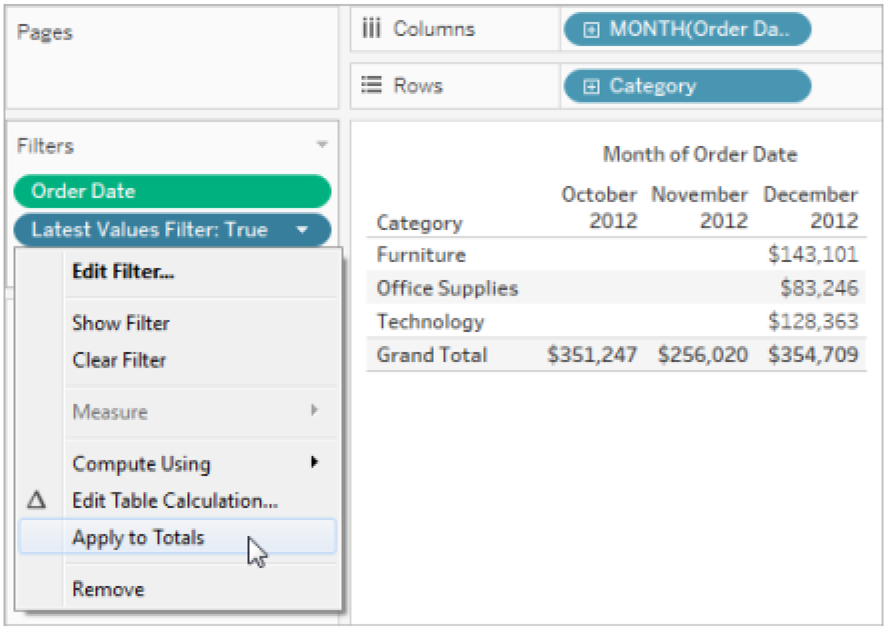
Filter table calculations

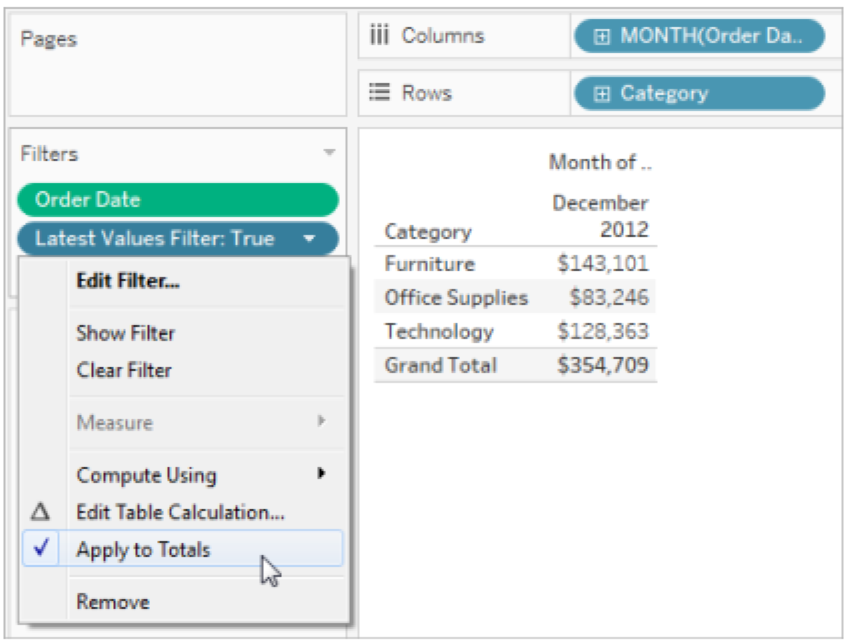
To create a table calculation filter, create a calculated field, and then place that field on the Filters shelf.

Filters based on table calculations do not filter out underlying data in the data set, because table calculation filters are applied last in the order of operations. This means Tableau evaluates any table calculations in the view first, and then applies table calculation filters on the results in the current view.

Apply table calculation filter to totals

When you show totals in a view and you want a table calculation filter to apply to the totals, you can select **Apply to totals** in the drop-down menu for that filter (on the Filters shelf). This option lets you decide when a table calculation filter should be applied to totals.





This options is available when you show totals in the view, and you add a table calculation filter to the view. Select **Apply to totals** to apply the table calculation filter to all of the results in the table, including the totals.

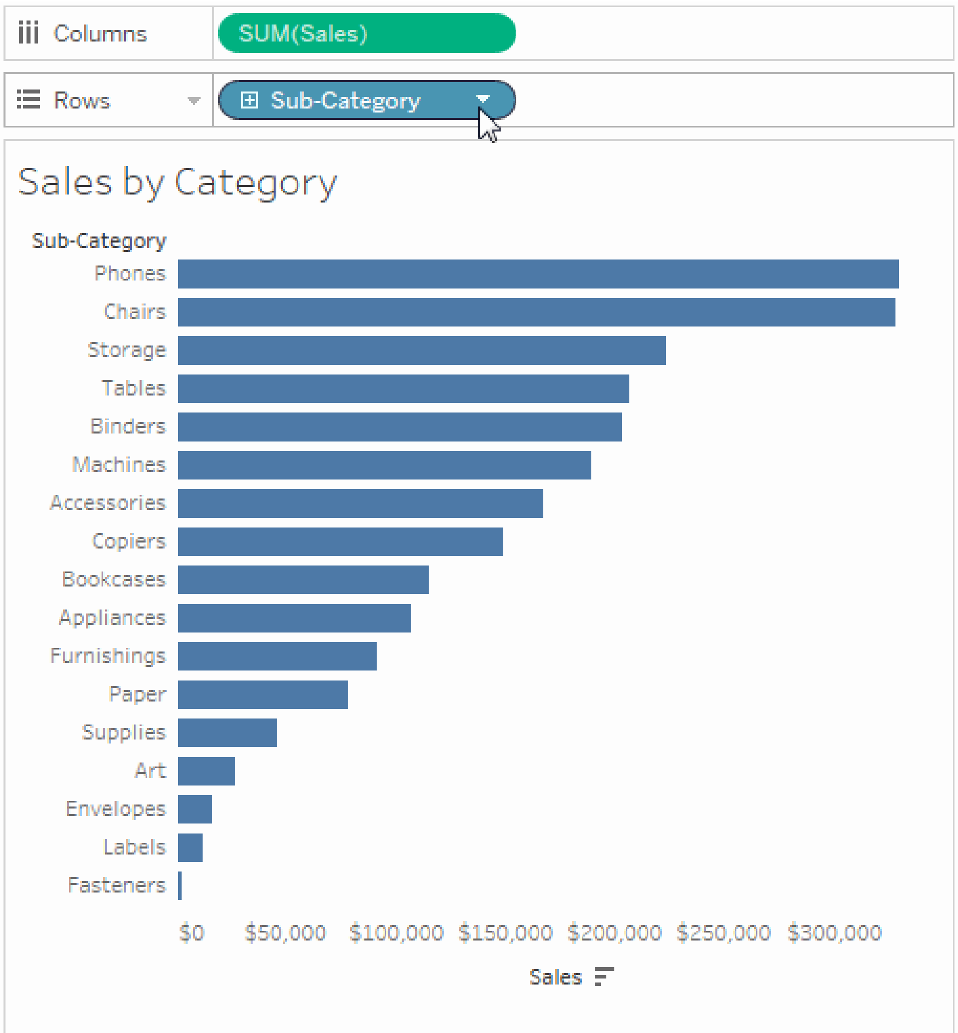
Display interactive filters in the view

When an interactive filter is shown, you can quickly include or exclude data in the view.

Note: In web authoring, interactive filters are automatically added to the view when you drag a field to the Filters shelf.

To show a filter in the view:

1. In the view, click the field drop-down menu and select **Show Filter**.

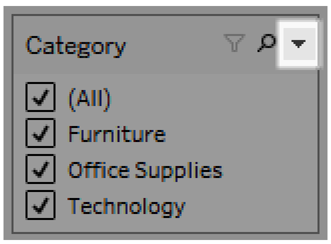


The field is automatically added to the Filters shelf (if it is not already being filtered), and a filter card appears in the view. Interact with the card to filter your data.

Note: In Tableau Desktop, you can add an interactive filter to the view for a field that is not currently used in the view. To do so, In the Data pane, click the field drop-down menu, and then select **Show Filter**.

Set options for filter card interaction and appearance

After you show a filter, there are many different options that let you control how the filter works and appears. You can access these options by clicking the drop-down menu in the upper right corner of the filter card in the view.



Some options are available for all types of filters, and others depend on whether you're filtering a categorical field (dimension) or a quantitative field (measure).

You can customize how filters appear in the view, in dashboards, or when published to Tableau Server or Tableau Online.

Here are some of the general filter card options:

- **Edit Filter** - This option opens the main Filter dialog box so you can further refine the filter by adding conditions and limits.
- **Remove Filter** - Removes the filter from the Filters shelf and removes the filter card from the view.
- **Apply to worksheets** - Allows you to specify whether the filter should apply to only the current worksheet or be shared across multiple worksheets. For more information, see **Apply Filters to Multiple Worksheets** on page 1444.
- **Format Filters** (Tableau Desktop only) - Customize the font and colors of all your filter cards in the view.
- **Only relevant values** - Specifies which values to show in the filter. When you select this option other filters are considered and only values that pass these filters are shown. For example, a filter on State will only show the Eastern states when a filter on Region is set.

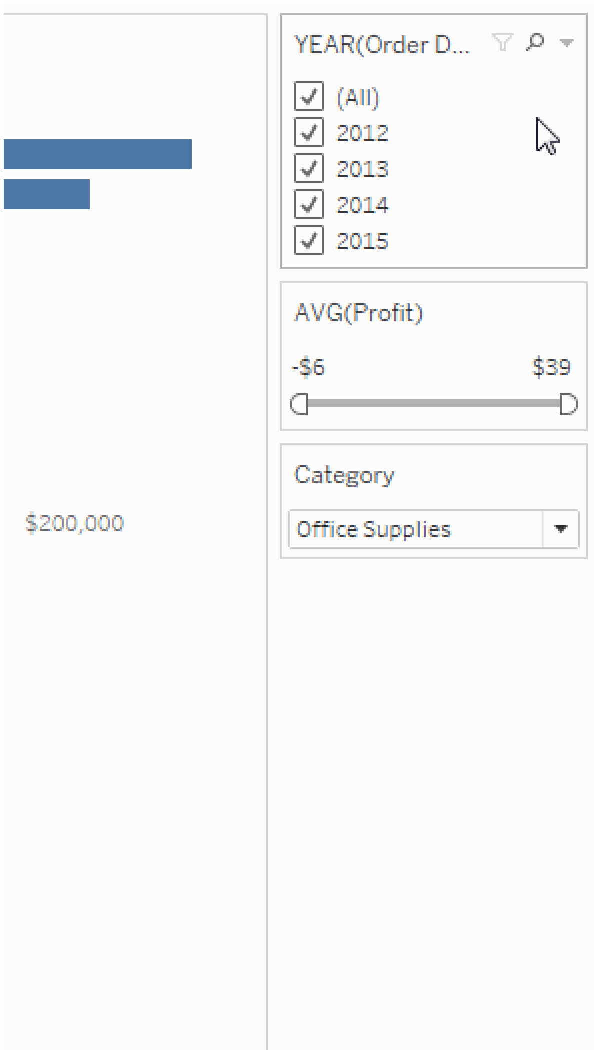
You can use the toggle at the top of the filter card to switch between this option and the All Values in Database option.

- **All values in hierarchy** - Specifies which values to show in the filter. When you create a filter from a hierarchical field, this option is selected by default. Filter values are displayed based on relevance of the parent/child relationships in the hierarchy.
- **All values in database** - Specifies which values to show in the filter. When you select this option all values in the database are shown regardless of the other filters on the view.
- **All values in context** (Tableau Desktop only) - When one of the filters in the view is a context filter, select this option on a different filter to only display values that pass through the context filter. For more information, see **Improve View Performance with Context Filters** on page 1439.
- **Include values** - When this option is selected, the selections in the filter card are included in the view.
- **Exclude values** - When this option is selected, the selections in the filter card are excluded from the view.
- **Hide Card** (Tableau Desktop only) - Hides the filter card but does not remove the filter from the Filters shelf.

Filter card modes

You can control the appearance and interaction of your filter card in the view by selecting a filter card mode.

To select a filter card mode, in the view, click the drop-down menu on the filter card and then select a mode from the list.



The types of filter card modes you see in the list of options depend on whether your filter is on a dimension or a measure. Below you can find brief descriptions of the types of filter card modes available for dimensions and measures.

For dimensions, you can choose from the following filter modes:

- **Single Value (List)**: Displays the values of the filter as a list of radio buttons where only a single value can be selected at a time.
- **Single Value (Dropdown)**: Displays the values of the filter in a drop-down list where only a single value can be selected at a time.
- **Single Value (Slider)**: Displays the values of the filter along the range of a slider. Only a single value can be selected at a time. This option is useful for dimensions that have an

implicit order such as dates.

- **Multiple Values (List)**:Displays the values in the filter as a list of check boxes where multiple values can be selected.
- **Multiple Values(Dropdown)**: Displays the values of the filter in a drop-down list where multiple values can be selected.
- **Multiple Values (Custom List)**: Displays a text box where you can type a few characters and search for the value. Alternatively, you can type or paste a list of values into the text box to create a custom list of values to include.
- **Wildcard Match**: Displays a text box where you can type a few characters. All values that match those characters are automatically selected. You can use the asterisk character as a wildcard character. For example, you can type “tab*” to select all values that begin with the letters “tab”. Pattern Match is not case sensitive. If you are using a multidimensional data source, this option is only available when filtering single level hierarchies and attributes.

For measures, you can choose from the following filter modes:

- **Range of Values/Dates**: shows the filtered values as a pair of sliders that you can adjust to include or exclude more values. Click on the upper and lower limit readouts to enter the values manually.

  The darker area inside the slider range is called the data bar. It indicates the range in which data points actually lie in the view. Use this indicator to determine a filter that makes sense for the data in your data source. For example, you may filter the Sales field to only include values between $200,000 and $500,000 but your view only contains values between $250,000 and $320,000. The range of data you can see in the view is indicated by the data bar while the sliders show you the range of the filter.

  > **Note:** Data bars only show in filters where the filtered field is also used in the view (e.g., on Columns, Rows, or on the Marks card, and son on) and are at the same aggregation level as the field on the Filters shelf. For example, a filter on SUM (Sales) will only display data bars if the SUM(Sales) field is used in the view. It won't show if AVG(Sales) is used in the view. Even though in both scenarios, the filtered field, Sales is used in the view; in the latter case the aggregation is

different than the aggregation of the filter.

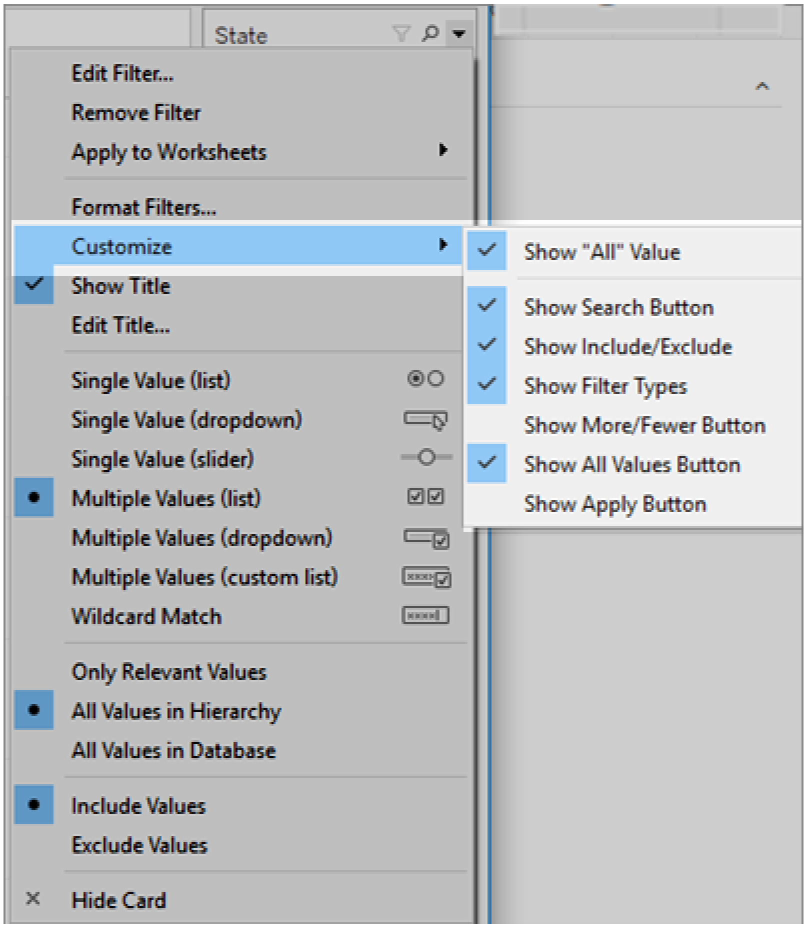
- **At Least/Starting Date**: shows a single slider with a fixed minimum value. Use this option to create a filter using an open ended range.
- **At Most/Ending Date**: shows a slider with a fixed maximum value. Use this option to create a filter using an open ended range.
- **Relative to Now**: this option shows a control where you can define a dynamic date range that updates based on when you open the view. The option is only available for filters on continuous date fields.
- **Browse periods**: shows common date ranges such as past day, week, month, three months, one year, and five years. This option is only available for filters on continuous date fields.

Note: When you expose a filter for Measure Values or Measure Names as a single value list, selecting All will automatically convert the filter to a multiple values list. For information on Measure Values and Measure Names, see **Measure Values and Measure Names** on page 1310.

Customize filter cards

In addition to the general filter options and the filter modes, you control how your filter appears in the worksheet, on dashboards, or when published to the web even further in Tableau Desktop.

To customize filters, click the filter card drop-down menu and select **Customize**.

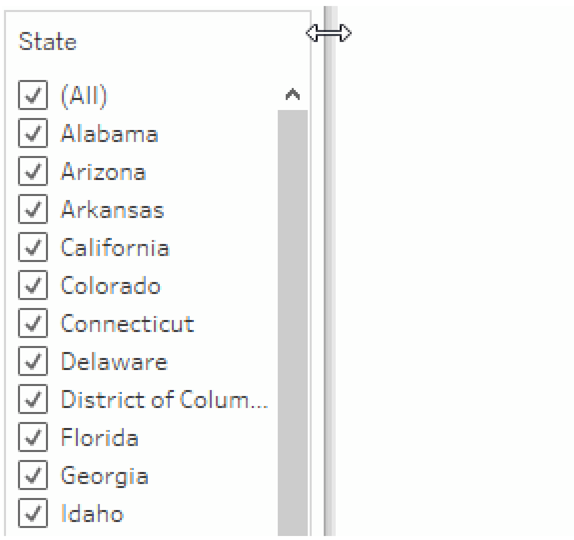


You can select from the following options:

- **Show “All” Value** - toggles whether to show the “All” option that displays by default in multiple values and single value lists.
- **Show Search Button** - toggles whether to show the search button at the top of the filter.
- **Show Include/Exclude** - toggles whether to show the Include Values and Exclude Values commands on the filter card menu. When shown, users can switch the filter between include and exclude modes.
- **Show Filter Types** - toggles whether to let users change the type of quick filter shown. For example, when shown, a user can change a multiple values list to a compact list.
- **Show More/Fewer button** - toggles whether to show the More/Fewer button at the top of the filter.

- **Show All Values button** - toggles whether to display the Show All Values button on the filter card.

  Whenever data is excluded in the filter, the small red "x" appears on the Show All Values button. When all values are showing, the red "x" disappears.

- **Show Apply Button** - toggles whether to show the Apply button at the bottom of the filter. When shown, changes to the filter are only applied after you click the button. Pending changes are indicated with a green color. This option is only available in multiple values lists and dropdowns. This options is available in web authoring.

- **Show Readouts** - controls whether the minimum and maximum values are displayed as text above a range of values. The readouts can be used to manually type a new value instead of using the sliders.

- **Show Slider** - controls whether the slider displays. When this option is cleared, the filter only displays the readouts.

- **Show Null Controls** - shows a drop-down list that lets you control how the filter handles null values. You can select from values in a range; values in a range and null values; null values only, non-null values, or all values.

  > **Note:** Not all of the above options are available for views published to Tableau Server or Tableau Online.

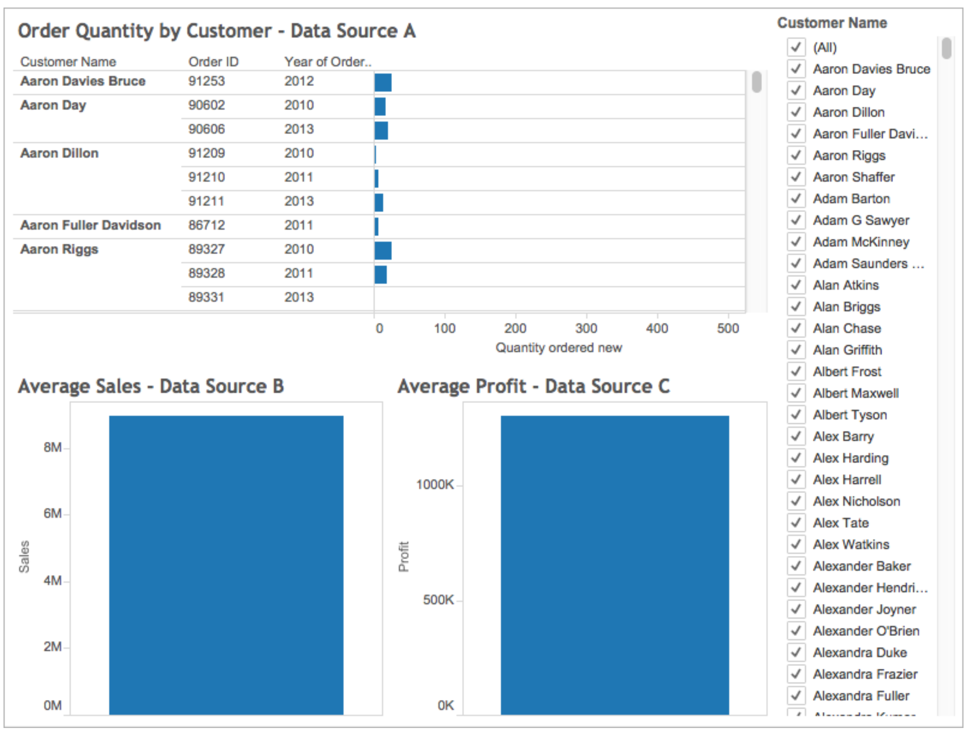
See Also

- **Filter Data from Data Sources** on page 1006
- **Restrict Access at the Data Row Level** on page 3088

Filter Data Across Multiple Data Sources

While working with multiple data sources in a workbook, you might want to compare the data between them using a field they have in common. To do so, you can apply a filter across multiple primary data sources.

For example, the following dashboard shows the order quantity, average sales, and average profit for customers. It has three views. Each of the views uses a different data source as its primary data source, and those data sources all have a field in common: Customer Name. There is also a filter card in the view for Customer Name.



This is an interesting dashboard with a lot of great information, but you might want to update all of the views in the dashboard at the same time by the customer you're analyzing. For example, maybe you want to see the average sales, profit, and number of orders you've received from one of your customers, Aaron Riggs.

To do so, you can filter all three data sources on the Customer Name field.

Follow the steps below to learn how to filter data across multiple data sources.

Before you start, you might want to know the following:

Starting with version 10.0, you can filter data across multiple primary data sources. You cannot filter data across secondary data sources. Filtering data across a worksheet's secondary data source is not currently supported in Tableau Desktop.

If you want to filter data across secondary data sources, consider the following alternatives:

- Join tables that are in the same data source, instead of blending them. For more information, see **Join Your Data** on page 883.
- Create a cross-database join if your tables are in different data sources. For more information, see **Join Your Data** on page 883.
- Make the primary data source the secondary data source, and the secondary data source the primary. **Note:** This might not always be feasible given the level of detail you want in the final view.

> **Note**: To ensure the data strings with mixed capitalization are treated as case-insensitive in the filter, create a calculated field using the UPPER() string function, and then create the filter relationship using that calculated field.

For more FAQs about cross data-source filters, see the Cross data-source filtering FAQs forum post in the Tableau Community.

Step 1: Define relationships between your data sources

Before you can create relationships between data sources, you must ensure that there is a common field between the data sources you're mapping. The fields do not need to be named the same in each data source, but they should have some data in common.

After you've identified the common fields, you must create relationships between them, or *map the fields* to one another.

If the fields happen to have the same name, Tableau might recognize them as related automatically. You can modify an existing relationship that was created automatically by Tableau, or create a new relationship between two fields in different data sources, by following the procedure below.

To define a relationship between your two data sources:

1. Select **Data** > **Edit Relationships**.
2. In the Edit Relationships dialog box, do the following:
   - Select a data source for **Primary data source**, and select a data source for **Secondary data source**.
   - Select **Custom**.
   - To define a relationship between fields that have different names, click **Add**.

     To edit an existing relationship, select the fields on the right, and then click **Edit**.
3. In the Add/Edit Field Mapping dialog box, do the following, and then click **OK**:
   - Under **Primary data source field**, select a field.
   - Under **Secondary data source field**, select a field with similar data to the field you selected for the primary data source.

     For more information about editing relationships, see **Blend Your Data** on page 910.

     Note: After you define your relationships, you do not need to enable blends (that is, you do not have to click the link icon in the **Data** pane) to filter across your data sources.

     For more information about the difference between primary and secondary data sources, see **Blend Your Data** on page 910.

Step 2: Add a filter to the view

After you have defined relationships between your data sources, go to one of your worksheets and drag a dimension to the **Filters** shelf. Then select to include or exclude data from the view. For more information about filtering your data, see **Filter Data from Your Views** on page 1411.

Optional Step: Display a filter card in the view. Right-click the field on the **Filters** shelf and select **Show filter**. A filter card for that field appears in the view.

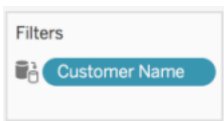
For more information about filter cards (previously known as quick filters), see **Display interactive filters in the view** on page 1421.

Step 3: Apply the filter to your worksheets

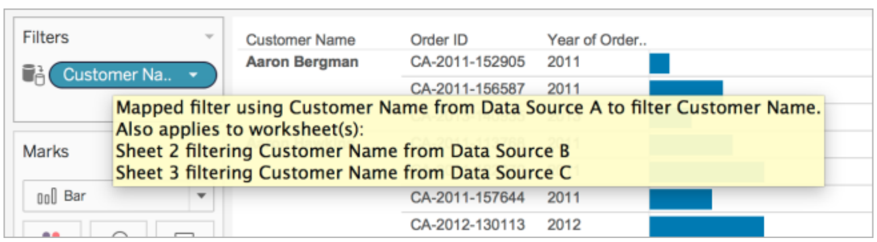
After you set up your filter, you can apply that filter to all the worksheets in your workbook that use related data sources, or you can apply it to selected worksheets.

- To apply the filter to all worksheets that use related data sources, right-click the field on the **Filters** shelf and select **Apply to worksheets** > **All using related data sources**.

  The Customer Name field is added to the **Filters** shelf on every worksheet that uses a related data source. An icon is added next to the field on the **Filters** shelf, indicating that the filter is being applied to multiple data sources.

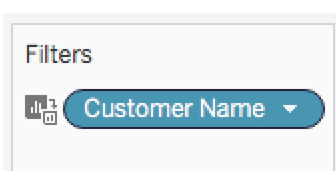
  

  If you hover over the field on the Filters shelf in any worksheet, you can find details about the type of filter, the source field for the filter, and the other sheets the filter applies to.

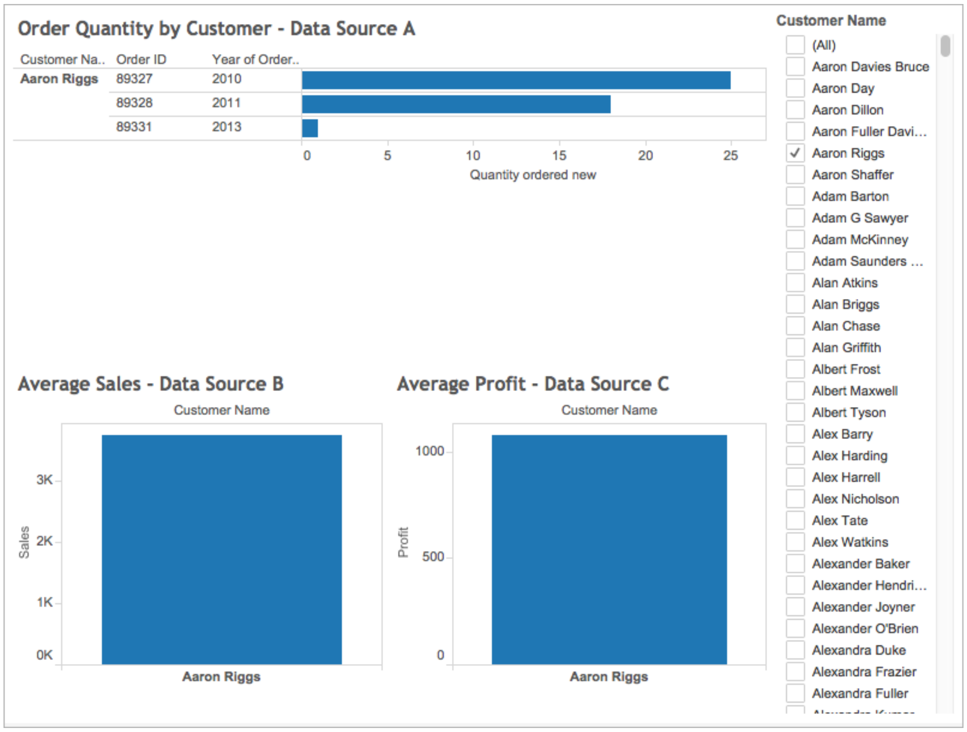
  

- To apply the filter to selected worksheets, right-click the field on the **Filters** shelf and select **Apply to worksheets** > **Selected worksheets**.

  The Customer Name field is added to the **Filters** shelf on the worksheets you selected. An icon is added next to the field on the **Filters** shelf to indicate that the filter is being applied to select worksheets.

Now, on the dashboard, when you filter the view down to Aaron Riggs, all three views update and you can see that Aaron made orders in 2010, 2011, and 2013, and spent an average of 3,700 USD. The average profit for these orders was approximately 1,000 USD.



Source fields and target fields

When you apply a filter to multiple data sources, you create a source field and one or more target fields. Both source fields and target fields appear on the **Filter** shelf in their respective worksheets.

The source field is the field you're filtering with.

The target field on any given worksheet is a field from another data source that is related to the source field. This is the field that is actually being filtered on that worksheet.

The source field determines the data that is included or excluded from the target fields. For example, let's say you have three worksheets that use three separate data sources (A, B, and C) as their primary data source. Each of the data sources has a field in common (Fruit), and the data is as follows:

Fruit from data source A	Fruit from data source B	Fruit from data source C
Apples	Apples	Apples
Oranges	Grapefruit	Grapefruit
Bananas	Oranges	Oranges
	Pears	Tomatoes
		Bananas

If the Fruit field from data source A is the source field for the cross data source filter, then the data that appears for the target fields is as follows:

Fruit from data source A - Source field	Fruit from data source B - Target field	Fruit from data source C - Target field
Apples	Apples	Apples
Oranges	Oranges	Oranges
Bananas		Bananas

Any data that does not match the data in the source field is excluded from the target fields, and will not appear in your worksheets or in your filter cards.

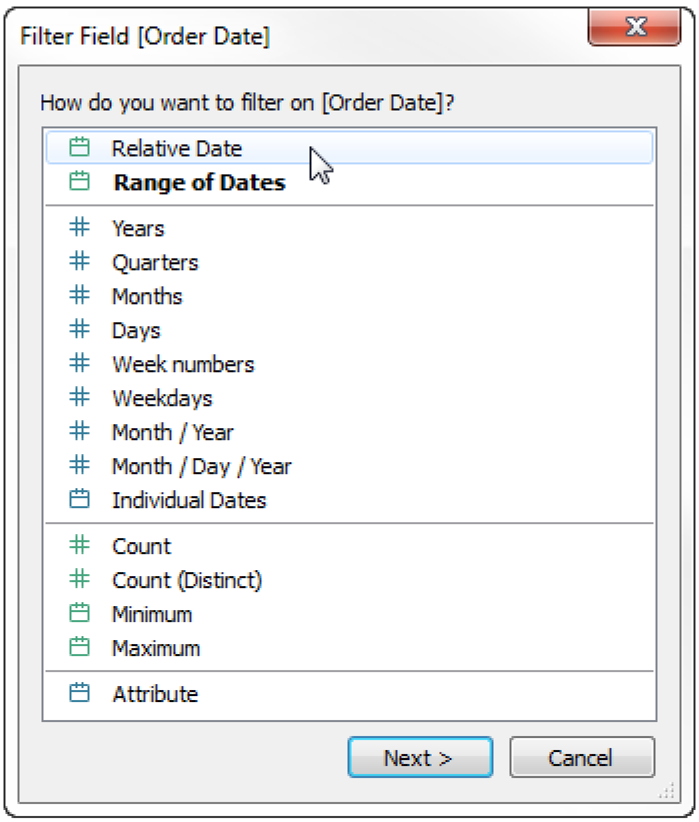
On the Filters shelf, source fields are indicated with either a icon when the source field applies to all worksheets with a related data source, or a icon when it applies to select worksheets. Target fields are indicated with a icon on the field on the Filters shelf. They are also indicated with a or icon next to the field on the Filters shelf.

Create Relative Date Filters

Relative date filters dynamically update to show a time period relative to when you open the view, such as the current week, the year to date, or the past 10 days. Relative date filters make it easy to create views that always show the most recent data.

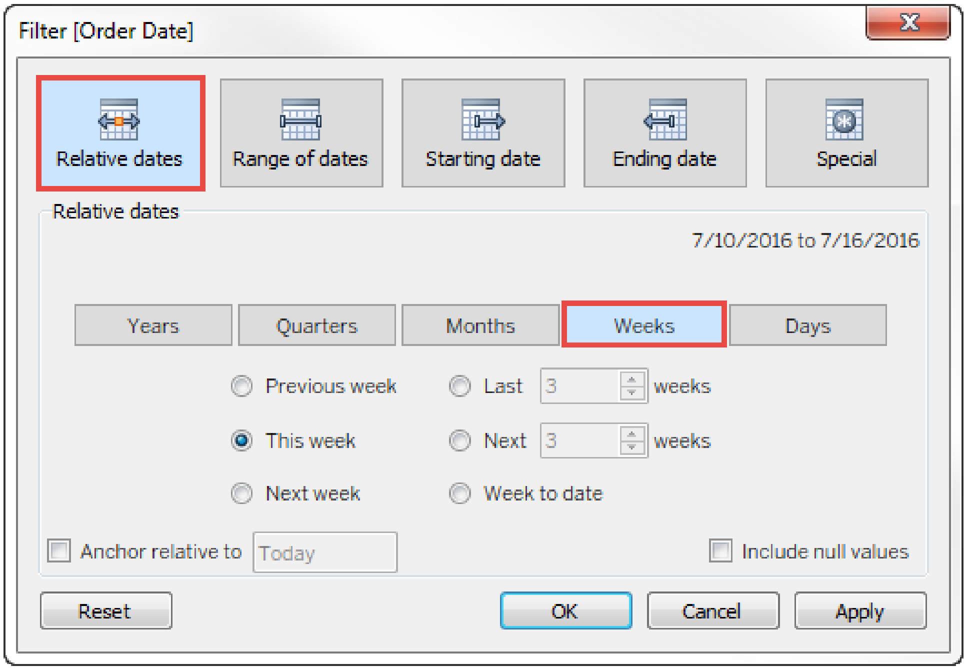
Step 1: Drag a date field to the filter shelf

Drag a date field from the **Data** pane to the Filters shelf. In the Filter Field dialog box, click **Relative Date**, and then click **Next**.



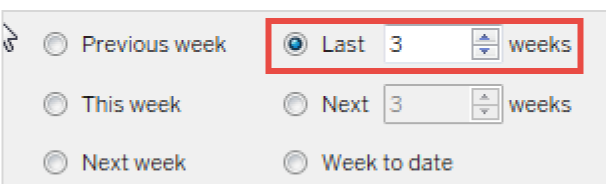
Step 2: Select a time unit

In the Filter dialog box, click **Relative dates**, and then select the unit of time for the filter. For example, to show only the last three weeks, select **Weeks**.

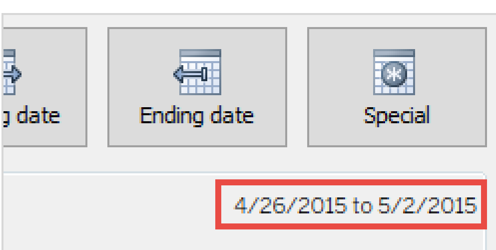


Step 3: Define the date period

Use the options in the lower part of the Filter dialog box to specify which date period to include in the view. For example, to show the last three weeks, click **Last**, and then enter the number **3**.



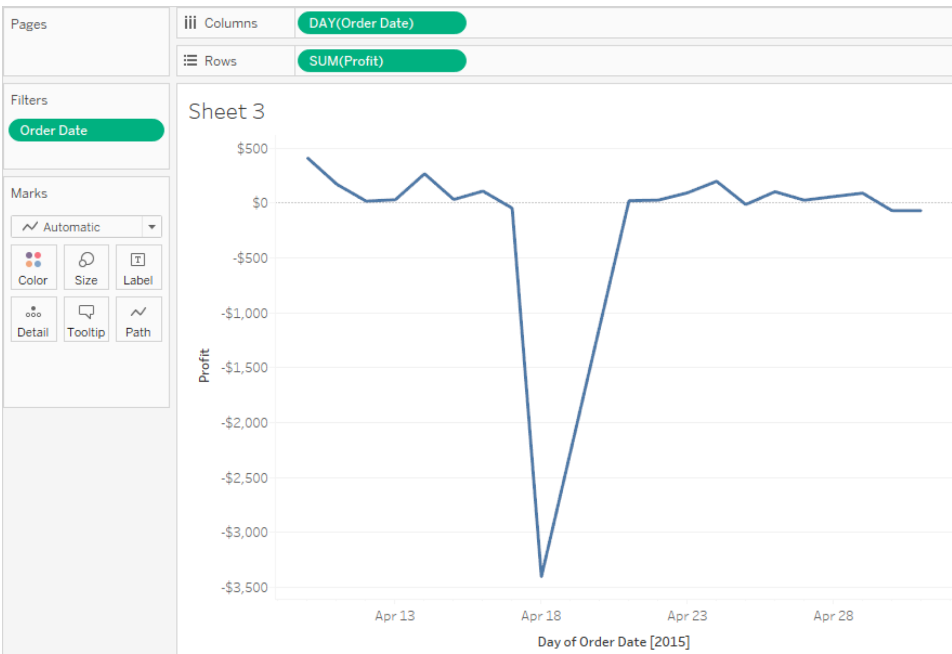
The specific range of time that you have selected is displayed in the upper right of the Filter dialog box.



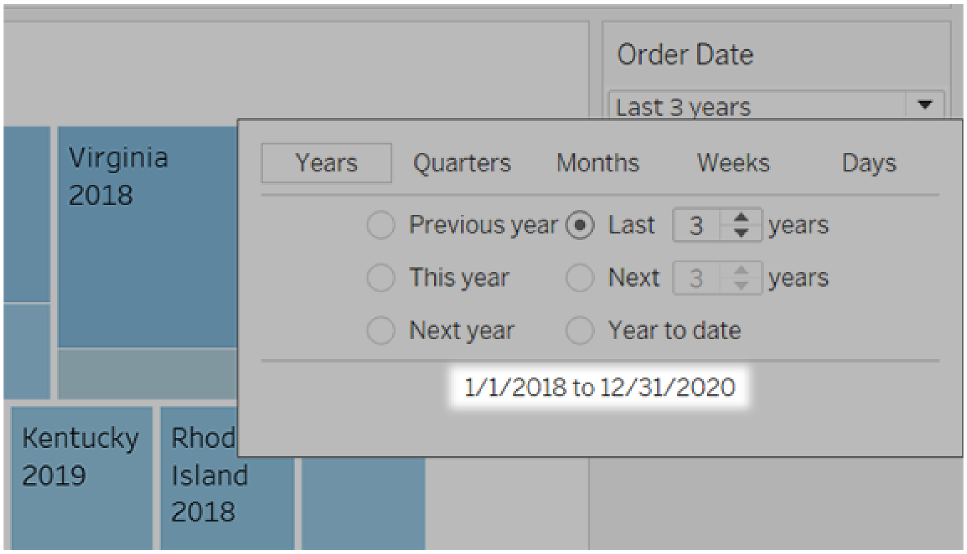
Note: “Last” date periods include the complete current unit of time, even if some dates haven't occurred yet. For example, if you select the last month and the current date is January 7th, Tableau will display dates for January 1st through January 31st.

Step 4: Watch the view update

The view will now update to always show the relative date range you've chosen.



Once created, you can show relative date filters in the view as cards. See **Display interactive filters in the view** on page 1421. To confirm the specific date range, users can click the menu on the filter card:



Improve View Performance with Context Filters

By default, all filters that you set in Tableau are computed independently. That is, each filter accesses all rows in your data source without regard to other filters. However, you can set one or more categorical filters as context filters for the view. You can think of a context filter as being an independent filter. Any other filters that you set are defined as dependent filters because they process only the data that passes through the context filter.

You may create a context filter to:

- Improve performance – If you set a lot of filters or have a large data source, the queries can be slow. You can set one or more context filters to improve performance.
- Create a dependent numerical or top N filter – You can set a context filter to include only the data of interest, and then set a numerical or a top N filter.

For example, suppose you're in charge of breakfast products for a large grocery chain. Your task is to find the top 10 breakfast products by profitability for all stores. If the data source is very large, you can set a context filter to include only breakfast products. Then you can create a top 10 filter by profit as a dependent filter, which would process only the data that passes through the context filter.

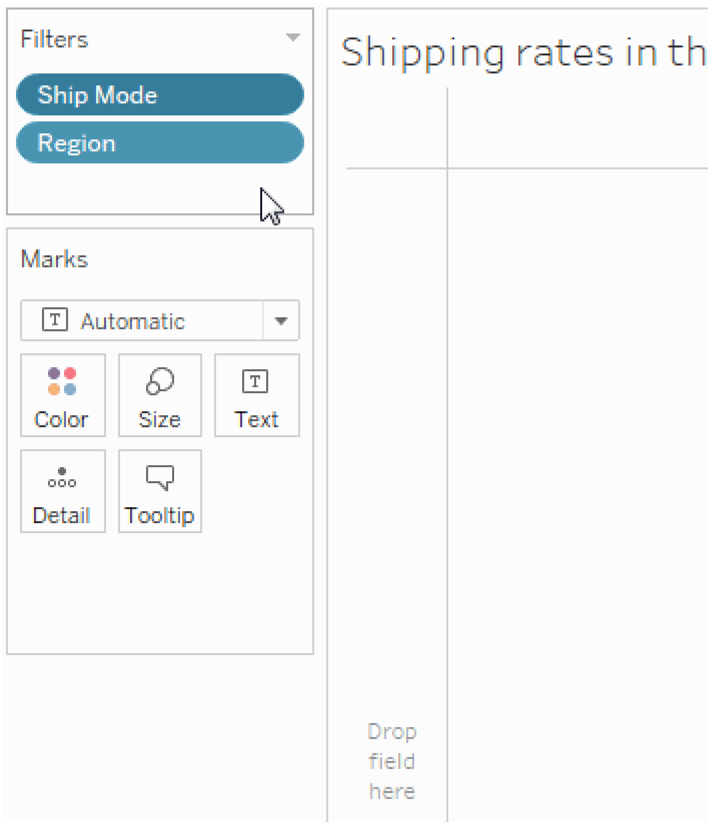
Note: As of Tableau 9.0, context filters no longer create temporary tables, except for generic ODBC data sources and customized data sources.

Create Context Filters

To create a context filter, select **Add to Context** from the context menu of an existing categorical filter. The context is computed once to generate the view. All other filters are then computed relative to the context. Context filters:

- Appear at the top of the Filters shelf.
- Are identified by a gray color on the Filters shelf.
- Cannot be rearranged on the shelf.

As shown below, the **Ship Mode** dimension is set to be the context for a view. The **Region** filter is computed using only the data that passes through **Ship Mode**.



You can modify a context filter by:

- Removing the field from the Filters shelf – If other context filters remain on the shelf, a new context is computed.
- Editing the filter – A new context is computed each time you edit a context filter.

- Selecting **Remove from Context** – The filter remains on the shelf as a standard filter. If other context filters remain on the shelf, a new context is computed.

Speed up Context Filters

To improve performance of context filters, especially on large data sources, follow these general rules.

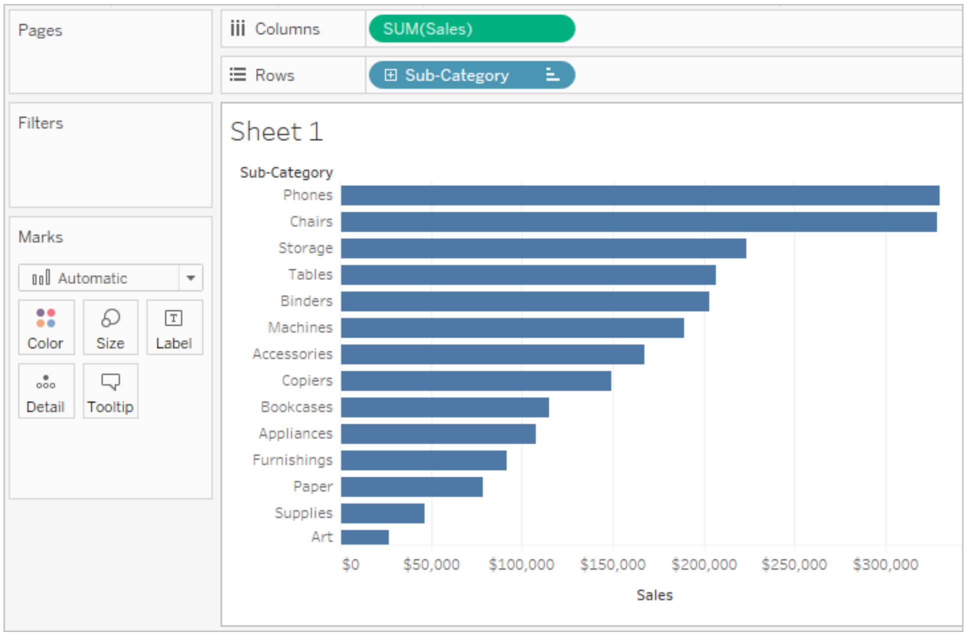
- Using a single context filter that significantly reduces the size of the data set is much better than applying many context filters. In fact, if a filter does not reduce the size of the data set by one-tenth or more, it is actually worse to add it to the context because of the performance cost of computing the context.
- Complete all of your data modeling before creating a context. Changes in the data model, such as converting dimensions to measures, require recomputing the context.
- Set the necessary filters for the context and create the context before adding fields to other shelves. Doing this work first makes the queries that are run when you drop fields on other shelves much faster.
- If you want to set a context filter on a date you can use a continuous date. However, using date bins like YEAR(date) or context filters on discrete dates are very effective.

> **Note**: If your data set is heavily indexed, context filters may not provide performance improvement and may actually cause slower query performance.
>
> Context filters can adversely affect any query performance improvements when you use the **Assume Referential Integrity** option in the Data menu of the data source. For more information, see **Assuming Referential Integrity for Joins** on page 899.

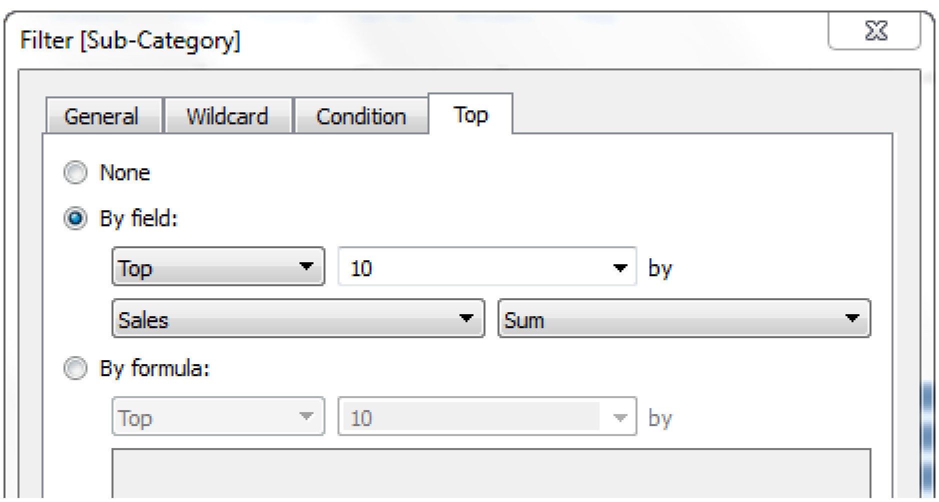
Example: Create Context Filters

This example walks you through how to create a context filter. First you'll filter a view to show the top 10 products by sales. Then you'll create a context filter on product category so you can see the top 10 furniture products.

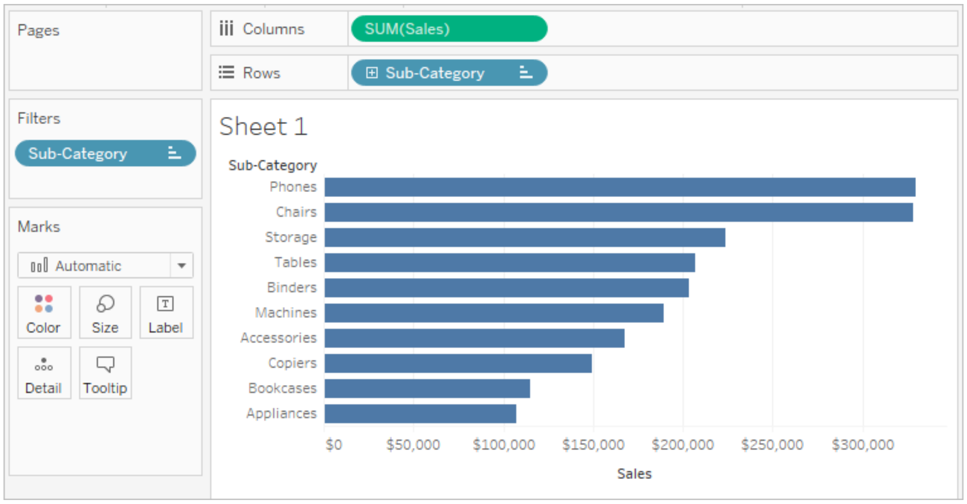
1. Use the **Sample - Superstore** data source to create the initial view shown below. The view shows the sales for all sub-categories, sorted with the highest sale at the top.



2. Now create a Top 10 filter to just show the top selling products. You can create this filter by dragging the **Sub-Category** field to the Filters shelf. In the Filter dialog box, switch to the **Top** tab and define a filter that is Top 10 by Sum of Sales. See **Filter Data from Your Views** on page 1411 to learn more about defining a Top N filter.

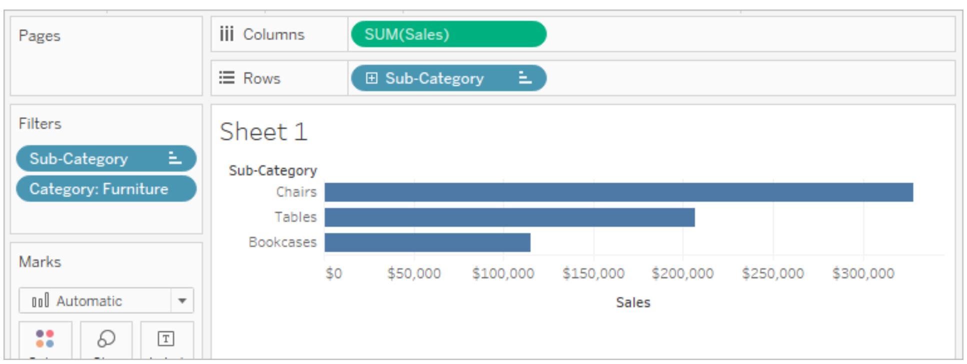


3. When you click **OK**, you'll see that the view is filtered to show the top 10 product sub-categories in terms of sales.

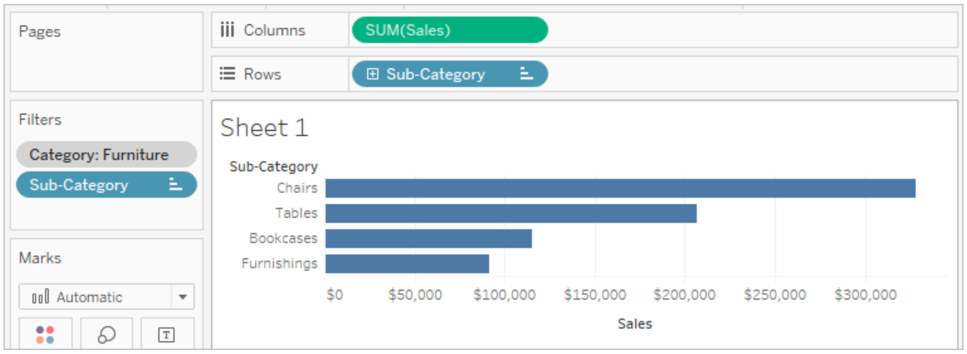


4. Now, let's add another filter to show only furniture products. Drag the **Category** field to the Filters shelf and select only **Furniture**. When finished, click **OK**.

   The view is filtered but instead of 10 products, it now shows 3. This is because by default all filters are evaluated separately and the view shows the intersection of the results. So this view shows that three of the top 10 overall products are furniture products.



5. To find out what the top 10 furniture products are we need to make the Category filter a context filter. Right-click the field on the Filters shelf and select **Add to Context**.
6. The filter is marked as a context filter and the view updates to show the top four furniture products. Why not 10? Because only four of the sub-categories contain furniture. But we now know that the Top 10 filter is being evaluated on the results of that context.



Apply Filters to Multiple Worksheets

When you add a filter to a worksheet, by default it applies to the current worksheet. Sometimes, however, you might want to apply the filter to other worksheets in the workbook.

You can select specific worksheets to apply the filter to or apply it globally to all worksheets that use the same data source or related data sources. For example, you might have a filter that only includes a specific region or product of interest. Rather than adding this filter every time you create a new worksheet, you can simply create the filter once and then apply it to multiple worksheets.

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Additional filtering topics (7 minutes), Ways to Filter (2 minutes), Where Tableau Filters (4 minutes). Use your tableau.com account to sign in.

For more examples of global filters, also see The Global Filters Tab on Ryan Sleeper's website, and Dashboard Design: Adding Interactivity on the Interworks blog.

Apply filters to all worksheets that use a related primary data source

This option applies the filter to all worksheets that use related data sources as their primary data source.

Note: To apply a filter to worksheets that use a related primary data source in web authoring, relationships between the data sources must be set up in Tableau Desktop, and then published to the web.

For more information, see **Filter Data Across Multiple Data Sources** on page 1430.

To apply a filter to all worksheets using a related primary data source:

- On the Filters shelf, right-click the field and select **Apply to Worksheets** > **All Using Related Data Sources**.

Filters that use this option are global across the workbook.

Filters that apply to all related data sources are marked with an icon. The filter is automatically created on any existing worksheets, and on any new worksheets you create that use a related data source.

Any changes you make to the filter affects all of those worksheets.

Apply filters to all worksheets that use the current primary data source

This option applies the filter to all worksheets that use the current worksheet's primary data source as their primary data source.

To apply a filter to all worksheets using the current primary data source:

- On the Filters shelf, right-click the field and select **Apply to Worksheets** > **All Using This Data Source**.

Filters that use this option are global across the workbook.

Filters that apply to all worksheets are marked with a data source icon . The filter is automatically created on any new worksheets you create after you drag a field to the view.

Any changes you make to the filter affects all of those worksheets.

Note: If you are blending multiple data sources in a view, **All Using This Data Source** adds the filter to all sheets that use the same primary data source in the current sheet. The sheets that the filter gets applied to is not based on the filter field’s data source.

Apply filters to select worksheets

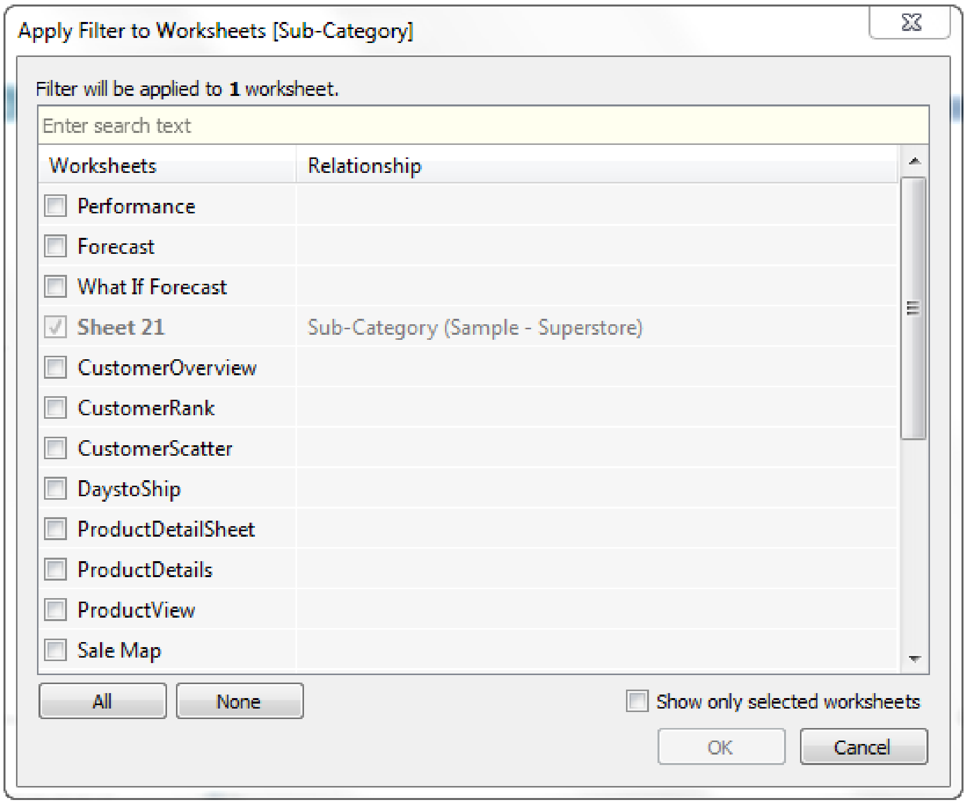
This option opens a dialog box where you can select from a list of worksheets that use the same data source or related data sources.

To apply a filter to select worksheets:

- On the Filters shelf, right-click the field and select **Apply to Worksheets** > **Selected Worksheets**.

- In the Apply Filter to Worksheets dialog box, select the worksheets that you want to apply the filter. If any of the sheets already contain a filter on the same field, the dialog box will provide details about the filter.

  If you select the sheet, the current filter will override any existing filter selections.



Filters that apply to a selection of worksheets are marked with the worksheet icon . Any changes you make to the filter affect all of the selected worksheets.

Apply filters to the current worksheet only

This option only applies to the current worksheet. This option is selected by default when you create new filters. Filters that are local to the current worksheet are shown without any additional icons.

To apply a filter to the current worksheet only:

- On the Filters shelf, right-click the field and select **Apply to Worksheets** > **Only This Worksheet**.

If you apply a filter to all worksheets or selected worksheets and then change the setting to apply the filter to just the current worksheet, the filters are not removed from all other worksheets. Rather, the filters are disconnected and they are all made local to their respective worksheets. You can go to each worksheet and remove the filter or modify selections.

Filter all worksheets on a dashboard

This option applies the filter to all worksheets in the dashboard that use the same data source or related data sources as their primary data source.

To filter all worksheets on a dashboard:

- In a dashboard, click the drop-down menu on a filter card and select **Apply to Worksheets** > **Selected worksheets**.
- In the Apply Filter to Worksheets dialog box, click **All on dashboard**, and then click **OK**.

> **Note:** In previous versions of Tableau Desktop, the **All Using This Data Source** option was called Make Global and the **Only This Worksheet** option was Make Local.

Sort Data in a Visualization

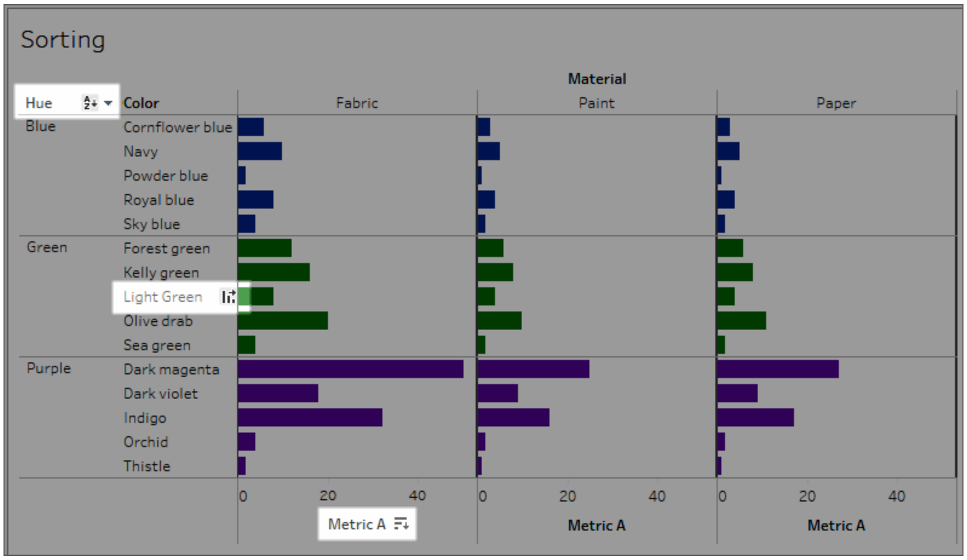
There are many ways to sort data in Tableau. When viewing a visualization, data can be sorted using single click options from an axis, header, or field label. In the authoring environment, additional sorting options include sorting manually in headers and legends, using the toolbar sort icons, or sorting from the sort menu.

Tip: Download the workbook used in these examples to explore on your own (requires Tableau Desktop).

Quickly sort from an axis, header, or field label

There are multiple ways to sort a visualization with single click sort buttons.

In all cases, **one** click sorts ascending, **two** clicks sorts descending, and **three** clicks clear the sort.

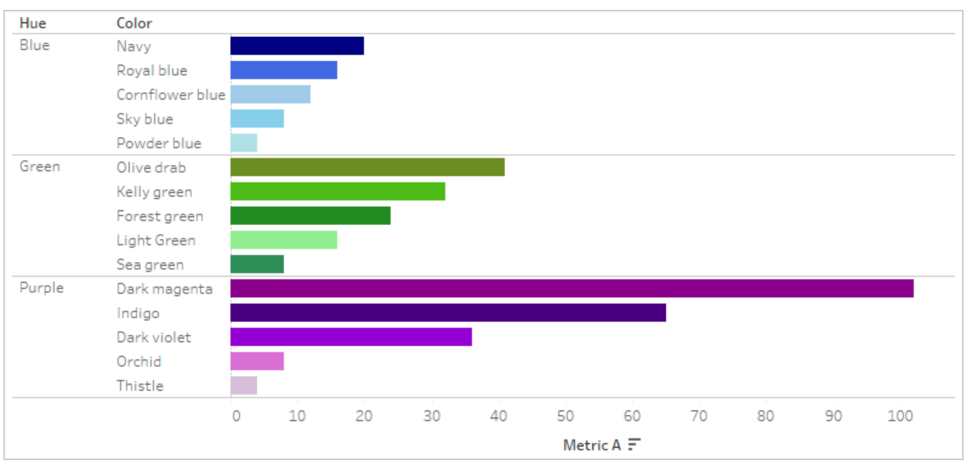


Sort icons may appear on an axis (Metric A), header (Light Green) or field label (Hue)

The sort will update correctly if the underlying data changes.

Sort from an axis

1. Hover over a numerical axis to bring up the sort icon.
2. Click the icon to sort.

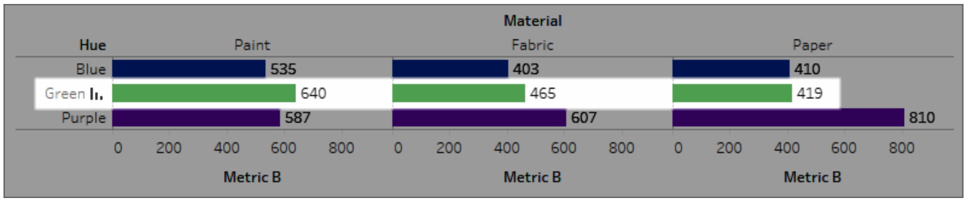


Sort: Color is sorted within each Hue in decreasing order of Metric A

In this example, the sort is applied to **Color** (sorting the rows) based on the values for **Metric A**. If there are hierarchical dimensions like above, the sort is applied to the innermost dimension. Here, that means that **Color** will sort inside **Hue**. Dark magenta cannot sort to the top of the viz because it must stay inside the pane for the Purple hue.

Sort from a header

1. Hover over a header to bring up the sort icon.
2. Click the icon to sort.



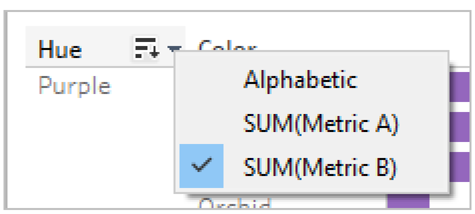
Sort: Materials are arranged in decreasing order of Metric B for Green.

In this example, the sort is applied to **Material** (sorting the order of the columns—Paint, Fabric, and Paper) based on the values for Green, since that header was used for the sort.

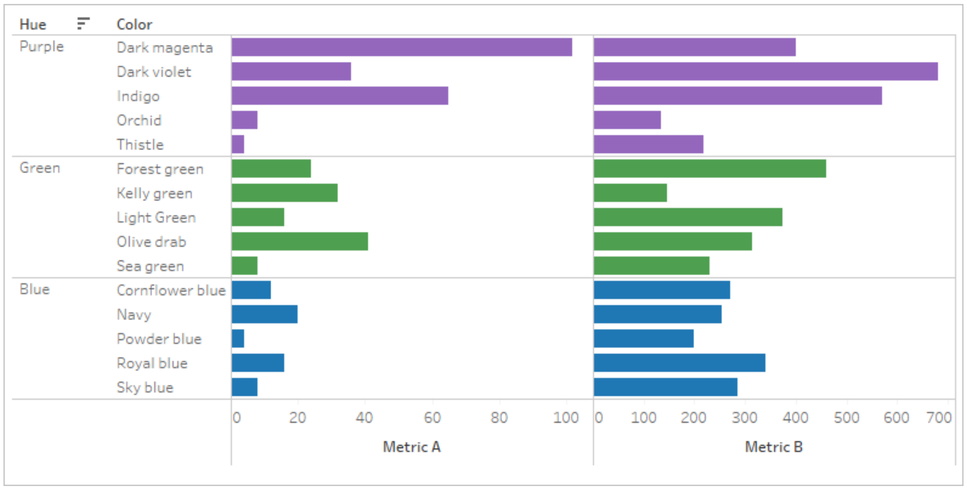
Sort from a field label

1. Hover over a field label to bring up the sort icon.

   The sort icon for a field label is slightly different from a header or axis. The default option is alphabetical sorting, but there is also a menu where you can choose to sort by a field in the view.

2. Click the A-Z icon to sort alphabetically, or open the menu to see a list of fields it is possible to sort by then chose a field. The icon switches to the bar icon and you can click to sort.



Sort: Hues are arranged in descending order by Metric B.

In this example, the sort is applied to the outermost dimension (Hue) based on total Metric B. (Metric B is aggregated for all the colors within each hue and Hue is sorted. Therefore, Purple is first, then Green, then Blue.)

Missing sort icons

If the sort icons do not appear, this functionality may have been turned off or it might not be possible to sort the view. For example, scatterplots cannot be sorted from a numerical axis because the position of the marks are fully determined by the data. No sort icon will appear on the axes in scatterplots.

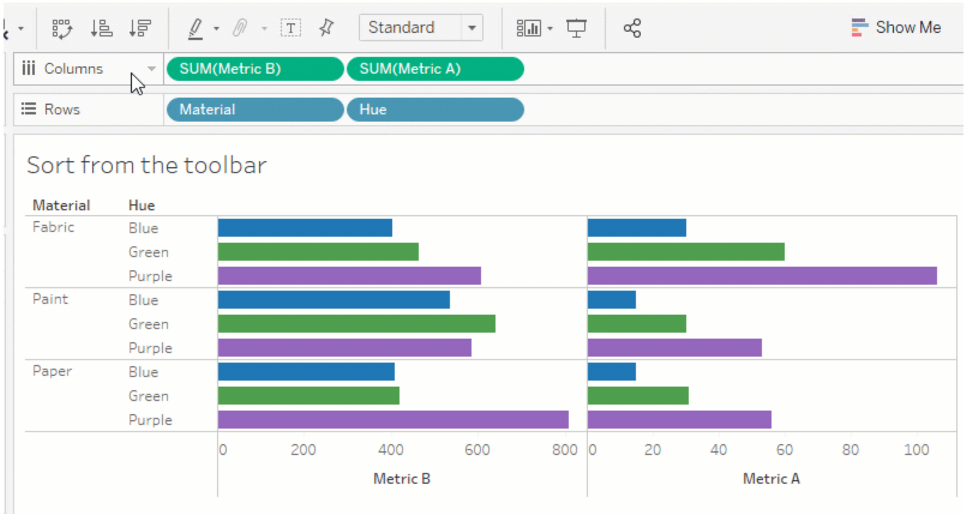
Sort options while authoring

In an authoring environment, there are additional sorting options to those available on published content.

Sort from the toolbar

1. Select the dimension you wish to sort.

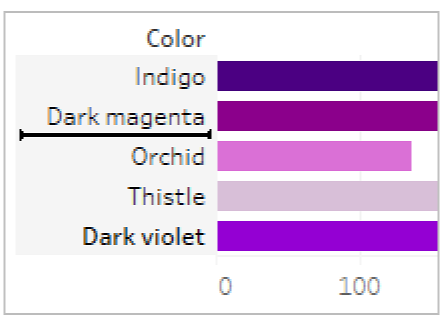
   If you do not select a field before sorting, the default behavior is to sort the deepest dimension.
2. Choose the appropriate sort button (ascending or descending) in the toolbar.

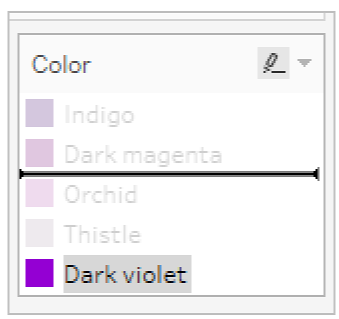


For example, in the gif above, the sort is applied to **Hue** unless the **Material** field is selected prior to sorting. The toolbar sort also applies to the leftmost measure, in this case Metric B. To sort by Metric A, it would be necessary to reverse their order on the Columns shelf or use another method of sorting. (To see the effect of sorting by Material, Hue is removed from the view. This makes it easier to see how the sort was computed.)

Sort by drag and drop

To manually sort, select a header in a viz or on a legend and drag it to the correct location—a heavy black line indicates where it will go.





Note: Sorting on a legend also changes the order of the marks, not simply how the legend is displayed. Whatever is bottommost in the legend becomes the bottommost mark in the viz. This can either mean closest to the axis or header, or actually underneath in the case of scatterplots and other viz types that may have overlapping marks.

Sort specific fields in the viz

Customize sorting by using the Sort menu for specific fields. (The Sort menu is not available for continuous measures.)

1. Right-click (Windows) or control-click (Mac) the field you want to sort, and select **Sort**.
2. Select a **Sort By** option, and configure its behavior:

   - **Data Source Order** sorts based on how the data is sorted in the data source. Generally for relational data sources, this tends to be natural sort order. Natural sort order is an alphabetic sort where multi-digit numbers are treated as a single character. For example, natural sorting puts 2 before 19 because 2 is less than 19, whereas alphabetic sorting puts 19 before 2 because 1 is less than 2.

     If you are using a cube, data source order is the defined hierarchal order of the members within a dimension.

   - **Alphabetic** sorts the data alphabetically. This sort is case sensitive, sorts [A-Z] before [a-z], and treats digits individually (that is, 19 comes before 2).

     To create a non-case sensitive sort, create a calculated field using the UPPER() or LOWER() functions to remove variation in capitalization. For more information, see **String Functions** on page 2160.

   - **Field** lets you select the field whose value will be used to determine the sort order. The field does not need to be used in the visualization. You can also select an aggregation for the sorting field. The aggregation options available depend on the field type.

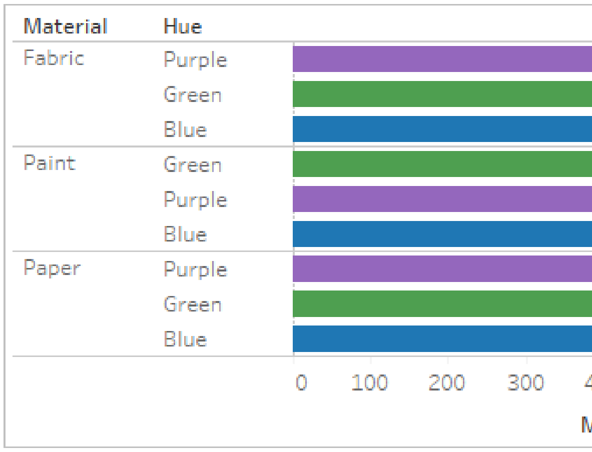
   - **Manual** lets you select a value and move it to the desired position, either by dragging it in the list or using the arrows to the right.

   - **Nested** lets you select the field whose value will be used to determine the sort order. The field does not need to be used in the visualization. You can also select an aggregation for the sorting field. The aggregation options available depend on the field type.

Nested sorts

When the values to be sorted occur in multiple panes, there are two ways to sort: nested and non-nested (or comparative). In the image below, all three values for **Hue** occur for all three **Materials**. Nested sorting considers values independently per pane rather than in aggregate across panes.

Nested



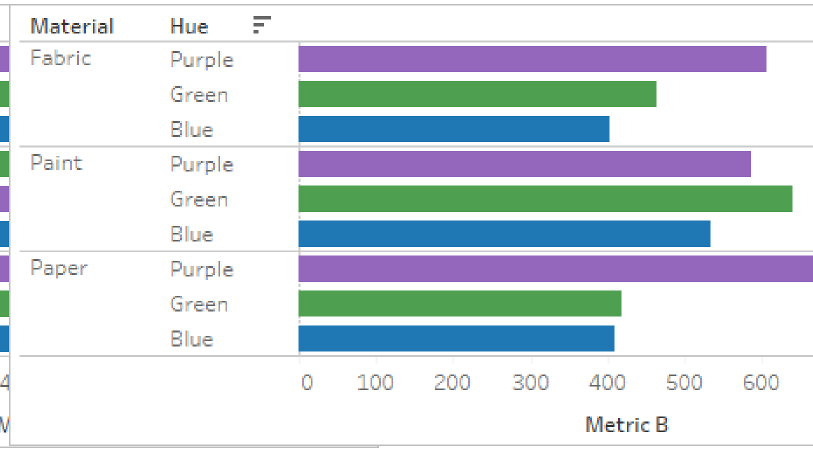
A nested sort considers each pane independently and sorts the rows per pane.

Purple is sorted above Green for Fabric and Paper, but below Green for Paint.

Nested sorts look correct within the context of the pane, but don’t convey the aggregated information about how the values compare overall.

Sorting from an axis gives a nested sort by default.

Non-nested



A non-nested sort considers the value across panes and will have the same order of values per pane.

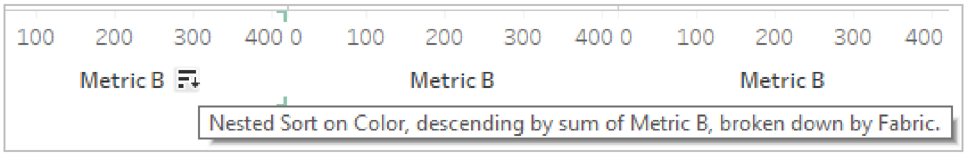
Purple is above Green for all materials because in aggregate, Purple is higher than Green.

Non-nested sorts may look incorrect in a single pane, but consistently convey how the aggregated values compare overall.

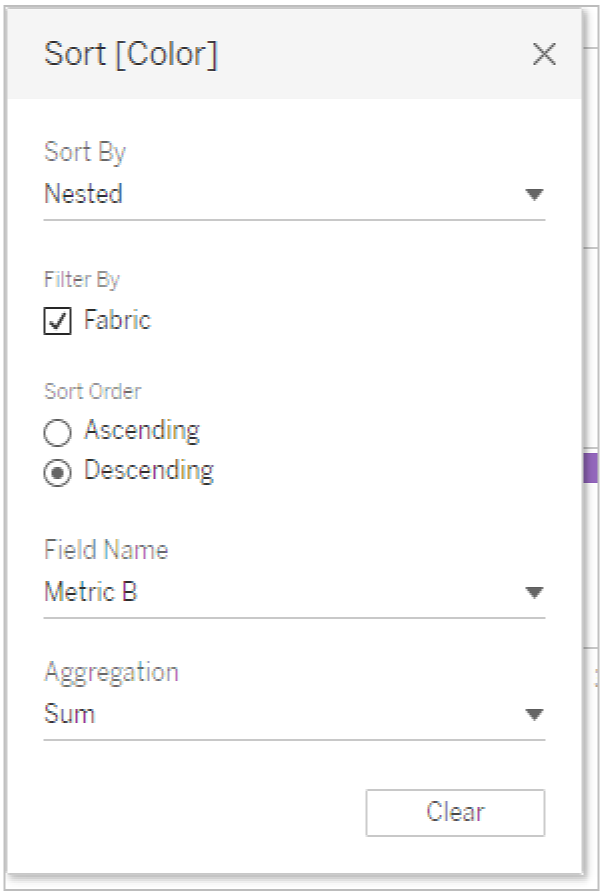
Sorting from a field label gives a non-nested sort by default.

Note: When creating a nested sort, the sort is inherited when you drill down through the dimensions. For example, a nested sort on Hue will apply to Color.

If a dimension has been placed on the same shelf as a measure, that measure will have an axis in the view for each value of the dimension. Using the axis sort icon will generate a nested sort specifically for that value.



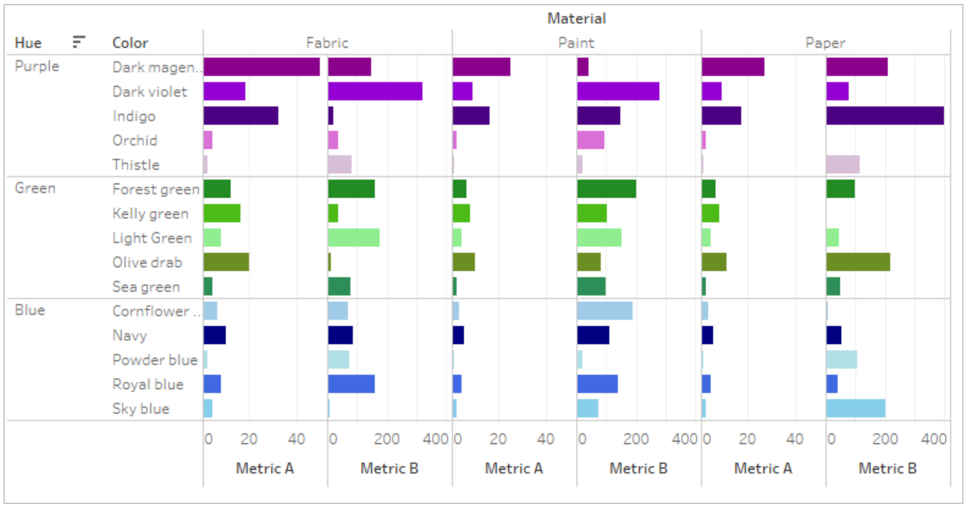
In this scenario, an additional field is added to the sort menu dialog for the sorted dimension. To stop breaking down the sort by this value, uncheck the box for Filter By.



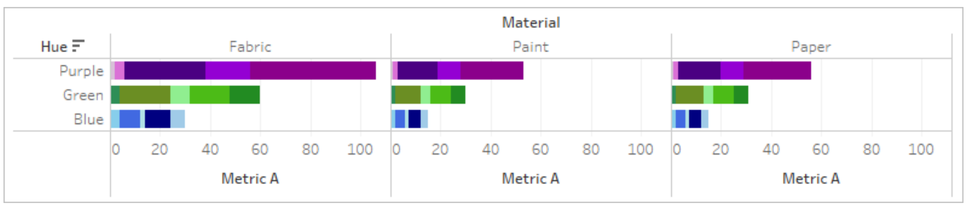
Understanding “incorrect” sorts

It is easy to think a view is sorted incorrectly when in fact the sort logic is simply not clear. A view with a nested sort when a non-nested sort is expected (or vice versa) can be perceived as incorrect. Sorting when there are multiple panes can also give results that seem unsorted.

For example, this view may look like it has not been sorted when in fact it is sorted on Hue by Metric A.



Simplifying the view makes the sort more obvious.



Hovering over the sort icons to see the description of the sort or temporarily simplifying the view can help clarify what is going on.

Clear sorts

Sorting can be done from a variety of places in the authoring environment. Although there are icons in the view and on the fields that indicate where a sort is applied, it can sometimes be useful to clear all sorts at once.

To remove the sort on a specific field, right click to open the menu and select **Clear Sort**.

To remove all sorts in the view, open the menu next to the clear sheet icon in the toolbar and select Clear Sorts. Alternatively, go to Worksheet > Clear > Sorts.

Remove the ability to sort

As an author, you can disable the sort icons in published content. This can be useful to preserve the visualization’s sort as you built it. Navigate to the Worksheet menu and uncheck the option to **Show Sort Controls**. This will prevent the sort icons from appearing when a consumer of the view hovers over an axis, header, or field label.

Hiding field labels, headers, or the axis will also remove the sort icons.

Showcase Insights

This section describes the various ways you spotlight data insights. Read the following articles for information on how to show totals, add annotations, and embed visualizations in tooltips when you hover over a data point. For text elements such as titles and captions, see **Format Titles, Captions, Tooltips, and Legends** on page 2937.

Watch a video: For a 6-minute walk through on how to embed visualizations within a tooltip, watch the Viz in Tooltip free training video. Use your tableau.com account to sign in.

For a 7-minute walk through on how to create tooltips in Tableau, watch the Basic Tooltips free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Show Totals in a Visualization

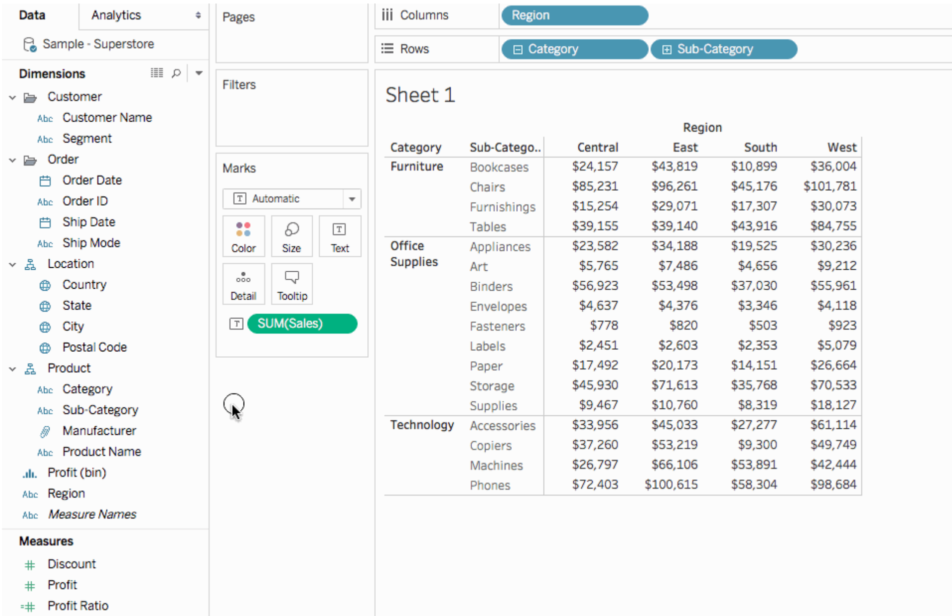
This article demonstrates how to show grand totals and subtotals in a visualization, as well as how to customize how those totals are calculated and where they appear in the visualization.

Show grand totals

To show grand totals in a visualization:

1. Click the **Analytics** pane.
2. In the **Analytics** pane, under Summarize, drag **Totals** into the Add Totals dialog, and

drop it over either the **Row Grand Totals** or **Column Grand Totals** option.

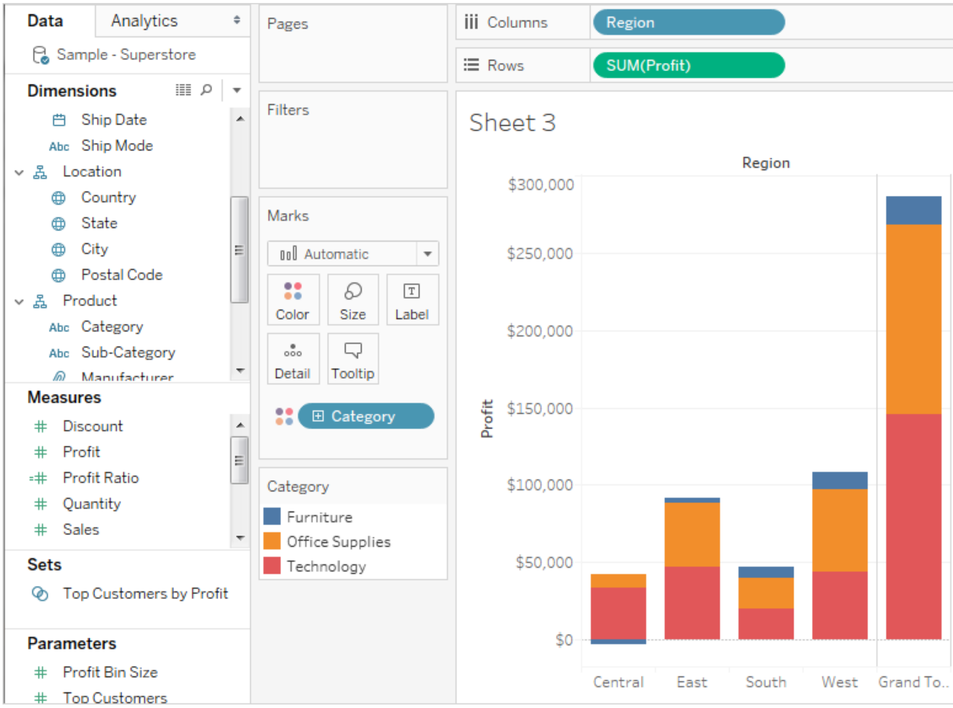


Row grand totals appear automatically on the right-side of the visualization. Column grand totals appear automatically at the bottom of the visualization. For information how to move where totals appear, see **Move totals** on page 1464.

In order to turn on grand totals:

- The view must have at least one header – Headers are displayed whenever you place a dimension on the **Columns** shelf or the **Rows** shelf. If column headers are displayed, you can calculate grand totals for columns. If row headers are displayed, you can calculate grand totals for rows.
- Measures must be aggregated – The aggregation determines the values displayed for the totals. See **Grand totals and aggregations** on page 1460 for more information.
- Grand totals cannot be applied to continuous dimensions.

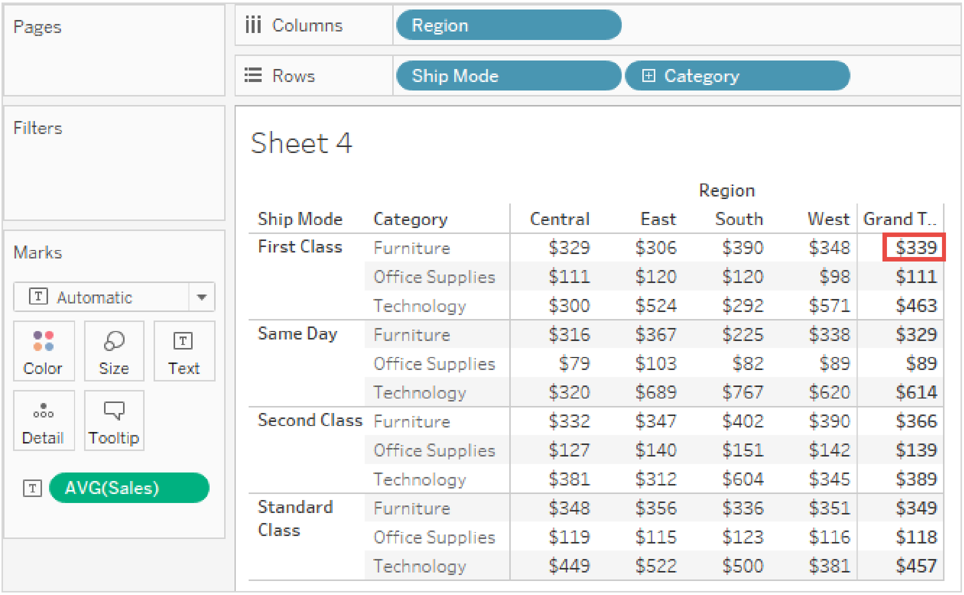
You can also display totals for graphical views of data. In the figure below, only column totals are calculated because the table contains only column headers.



> **Note:** By default, totals are computed on the server if you are connected to a Microsoft Analysis Services data source, and locally if you are connected to an Essbase data source, using the aggregation specified in the cube. Refer to **Configure total aggregation** on page 1464 to learn more.

Options for calculating grand totals

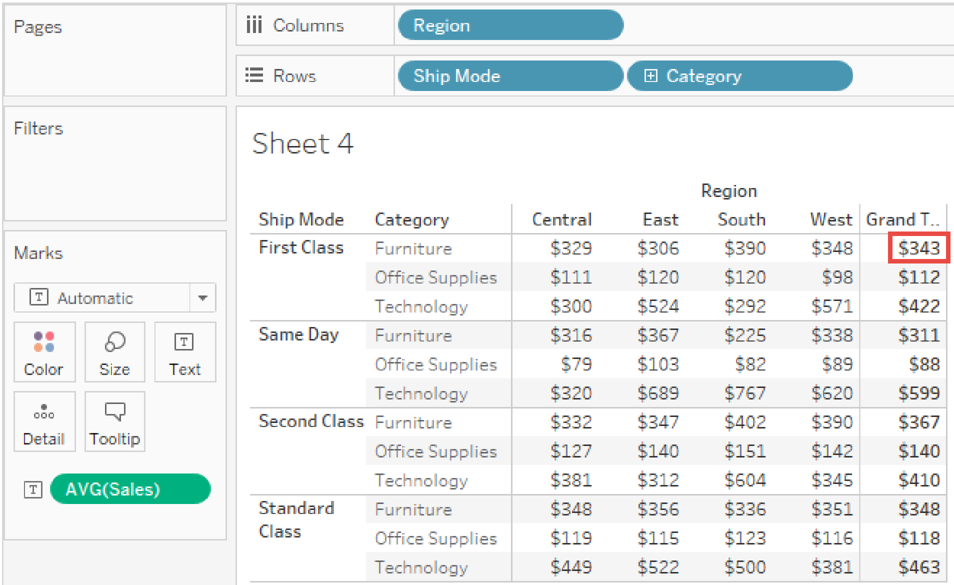
When you first turn on grand totals, the totals are computed using disaggregated data in the underlying data source. Consider the following view:



		Region				
Ship Mode	Category	Central	East	South	West	Grand T..
First Class	Furniture	$329	$306	$390	$348	$339
	Office Supplies	$111	$120	$120	$98	$111
	Technology	$300	$524	$292	$571	$463
Same Day	Furniture	$316	$367	$225	$338	$329
	Office Supplies	$79	$103	$82	$89	$89
	Technology	$320	$689	$767	$620	$614
Second Class	Furniture	$332	$347	$402	$390	$366
	Office Supplies	$127	$140	$151	$142	$139
	Technology	$381	$312	$604	$345	$389
Standard Class	Furniture	$348	$356	$336	$351	$349
	Office Supplies	$119	$115	$123	$116	$118
	Technology	$449	$522	$500	$381	$457

The average that you see at the right of the first row under Grand Total is $339. But if you compute the average for the four values you see in the row ($329, $306, $390, $348), the result comes to $343.25, not $339. The discrepancy is due to the fact that Tableau is averaging the data in the underlying data source. In this case there are more than four numbers to average, perhaps many more. The result is derived by averaging all values that have the properties Ship Mode = First Class, Category = Furniture, without regard to region.

To see totals that correspond to the numbers you see in your view requires two averaging operations in Tableau: first, values for the individual marks (or cells) in the view must be derived—for example, by averaging all values that have the properties Ship Mode = First Class, Category = Furniture, and Region = Central. Then the results for each region must be derived, by averaging the results for the individual marks. Fortunately, you do not need to perform two operations. To display a result of this type, from the **Analysis** menu choose **Totals** > **Total All Using** > **Average** . Now the average is performed on the values you see, and not on the disaggregated date in the data source:



		Region				
Ship Mode	Category	Central	East	South	West	Grand T..
First Class	Furniture	$329	$306	$390	$348	$343
	Office Supplies	$111	$120	$120	$98	$112
	Technology	$300	$524	$292	$571	$422
Same Day	Furniture	$316	$367	$225	$338	$311
	Office Supplies	$79	$103	$82	$89	$88
	Technology	$320	$689	$767	$620	$599
Second Class	Furniture	$332	$347	$402	$390	$367
	Office Supplies	$127	$140	$151	$142	$140
	Technology	$381	$312	$604	$345	$410
Standard Class	Furniture	$348	$356	$336	$351	$348
	Office Supplies	$119	$115	$123	$116	$118
	Technology	$449	$522	$500	$381	$463

This kind of total is sometimes referred to as a two-pass total, because the average you see in the grand total column is aggregated twice--once to derive the column or row value, and then again across column or rows to derive the grand total.

Grand totals and aggregations

When you turn on grand totals, the initial values are computing using the current aggregation for the fields in the view. In this case, totals are based on the underlying data rather than the data in the view.

For example, if you are totaling the **SUM(Profit)** for several products, the grand total will be the sum of the sums of profit. For aggregations such as SUM, you can easily verify the grand total because a summation of a group of sums is still a summation. However, be aware that your results may be unexpected when using other aggregations, especially custom aggregations. For details, See **Configure total aggregation** on page 1464. You can verify any calculation such as an aggregation or a grand total by viewing the underlying disaggregated data.

The following table summarizes the standard aggregations and the grand totals that are calculated by default when, from the Analysis menu, **Totals >Total All Using** is set to the default value **Automatic**.

> **Note**: Only Automatic totals are available for table calculations and fields from a secondary data source. Total aggregations cannot be applied to table calculations or

fields from a secondary data source.

For more information, see Showing Grand Totals with Blended Data and Grand Totals and Subtotals Do Not Show Expected Numbers With Table Calculations in the Tableau Knowledge Base.

Aggregation	Calculation Description
Sum	Shows the sum of the values shown in the row or column.
Average	Shows the average of the values shown in the row or column.
Median	Shows the median for the values shown in the row or column.
Count; Count Distinct	Shows how many values or distinct values are displayed in the rows

Aggregation	Calculation Description
	and columns in the view.
Minimum	Shows the minimum value shown in the row or column.
Maximum	Shows the maximum value shown in the row or column.
Percentile	Shows the average percentile for all values shown in the row or column.
Standard Deviation	The grand total using standard deviation is the standard deviation of the values shown in the row or

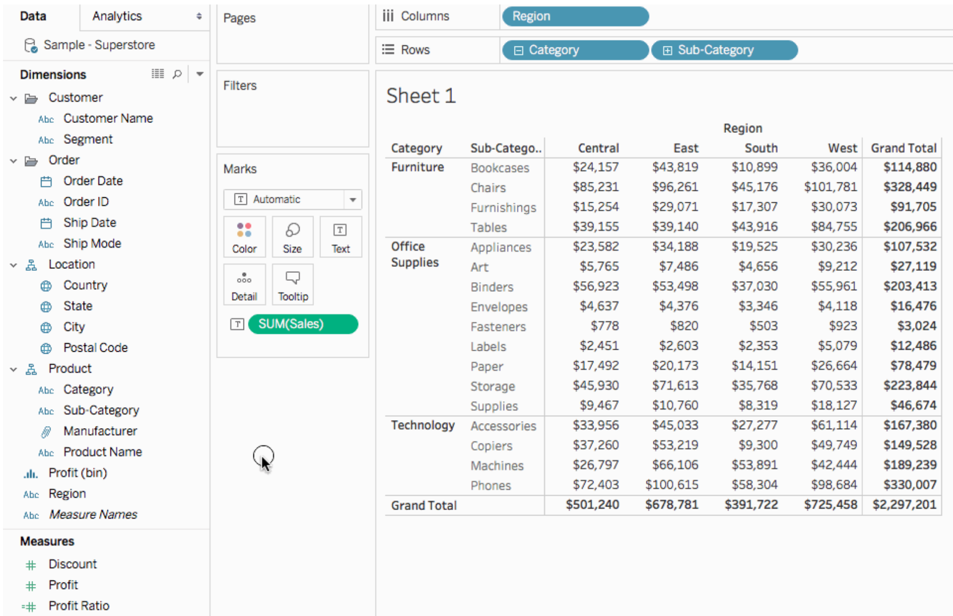
Aggregation	Calculation Description
	column.
Variance	The grand total using variance is not the variance of the rows and columns in which they reside but rather of the underlying data behind the row or column.

Show subtotals

To show subtotals in a visualization:

1. Click the **Analytics** pane.
2. In the **Analytics** pane, under Summarize, drag **Totals** into the Add Totals dialog, and

drop it over **Subtotals**.



Move totals

By default, row grand totals and subtotals appear on the right of the view, and column grand totals and subtotals appear at the bottom of the view.

In Tableau Desktop, you can also select to display totals on the left or top of the view.

To move row totals to the left of the view:

- Select **Analysis** > **Totals**, and then select **Row Totals to Left**.

To move column totals to the top of the view:

- Select **Analysis** > **Totals**, and then select **Column Totals to Top**.

Configure total aggregation

When totals are turned on in the visualization (either grand totals, subtotals, or both), you can specify how totals should be computed. For example, you can choose to calculate totals using a sum, average, minimum, or maximum.

To configure all totals:

- Select **Analysis** > **Totals** > **Total All Using**, and then select an aggregation from the list.

To configure totals for a specific field:

- Right-click (Control-click on a Mac) a field in the view, select **Total using**, and then select an aggregation from the list.

When you choose **Automatic**, totals are based on the underlying data, which is disaggregated, and not on the data in the view. See **Options for calculating grand totals** on page 1458. For details on how Tableau computes totals using the current aggregation, see **Grand totals and aggregations** on page 1460.

When you choose any of the other values (**Sum**, **Average**, **Minimum**, or **Maximum**), all totals are computed using the selected aggregation. The computations are performed on the aggregated data you see in the view.

An additional value, **Server**, may be available. Server computation is not always available and sometimes the totals will be blank for specific members in the view. When using server computation keep in mind the following information:

- Server computation is only available for ASO cubes.
- Server computation is not available for dynamic hierarchies. This means that if the members in the view are part of a dimension or hierarchy that is tagged as dynamic, they cannot be included in the set of values you are using to compute the totals and will show up as blank in the view.
- If you are computing totals for a calculated field whose formula makes assumptions about other calculated members at different levels in the hierarchy, the totals will display as blank in the view.

See Also

Apply Advanced Analysis to a View (Analytics Pane) on page 213

Add Annotations

You can add annotations to a visualization to call out a specific mark, a specific point, such as a location on a map, or an area, such as a cluster of scatter marks.

Note: In web authoring, you can create and delete annotations by right-clicking the view (in a worksheet or dashboard), and then selecting point or mark annotation. Right-click the annotation again to edit or remove the annotation. Formatting and moving annotations are not currently supported on the web.

To add and format other text elements on a visualization, see **Format Titles, Captions, Tooltips, and Legends** on page 2937.

Add an annotation

To add an annotation to your viz:

1. In a worksheet, right-click (control-click on Mac) a data point or a spot on the viz where you want to add an annotation and select **Annotate**, and then select the type of annotation you want to add.

   There are three types of annotations in Tableau:

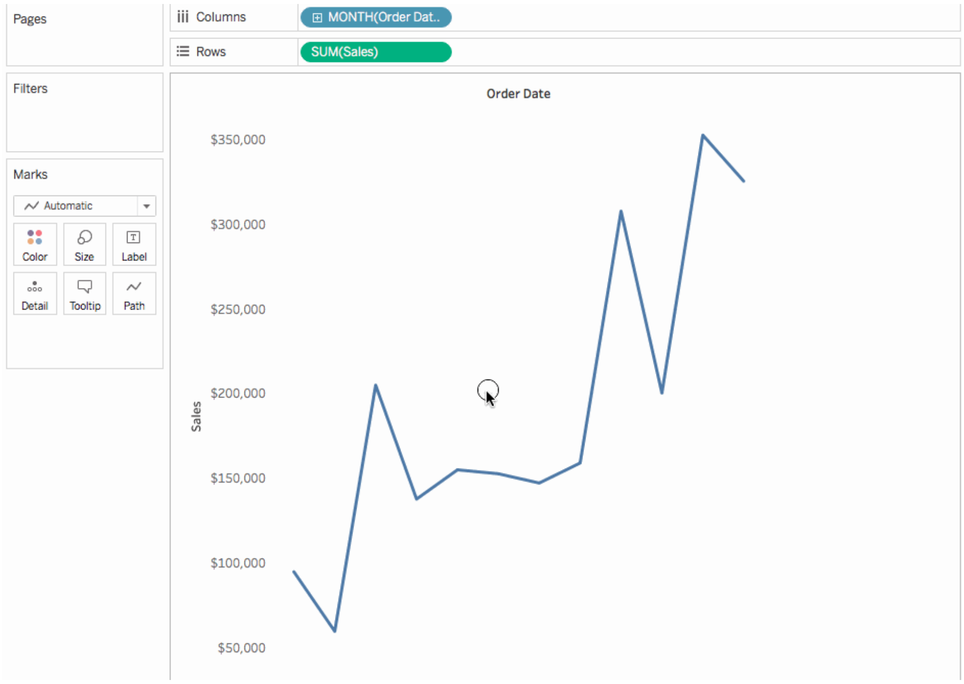
   - **Mark** - select this option to add an annotation that is associated with the selected mark. This option is only available if a data point (mark) is selected.
   - **Point** - select this option to annotate a specific point in the view.
   - **Area** - select this option to annotate an area in the viz, such as a cluster of outliers or a targeted region.

2. In the Edit Annotation dialog box that opens, type the text you want to show in the annotation.

   Use the **Insert** menu to insert dynamic variables into the annotation text. For example, the annotation can display data values that update as the underlying data changes. The dynamic variables that are available are dependent on whether you are annotating a mark, point, or area.

3. When finished, click **OK**.

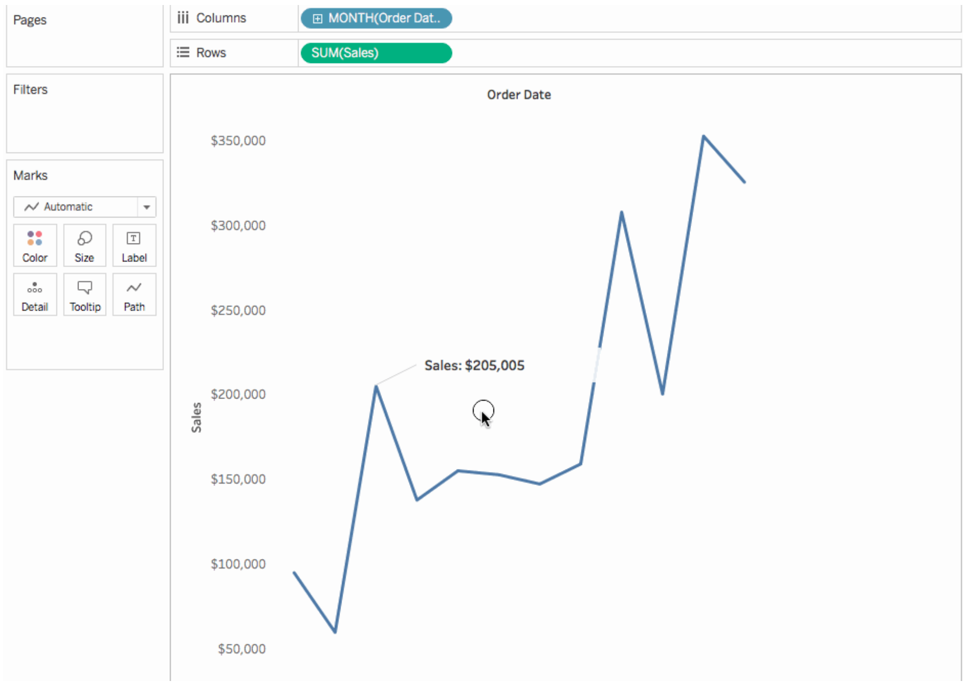
The viz updates with your annotation.



Edit an annotation

To edit an annotation:

1. Right-click (control-click on a Mac) the annotation in the viz and select **Edit**.
2. In the Edit Annotation dialog box that opens, edit the annotation text and then click **OK**.

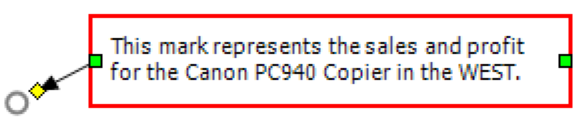


Rearrange an annotation

After you add an annotation, you can move it around, resize it, adjust the line, and move the text. Each type of annotation can be rearranged and modified in different ways. This section discusses how to rearrange, resize, and modify each type of annotation.

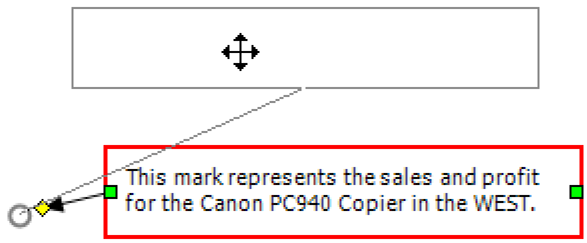
Mark annotations

When you select a mark annotation the body and line are selected and several resize handles display. Use these handles to resize the body and line.



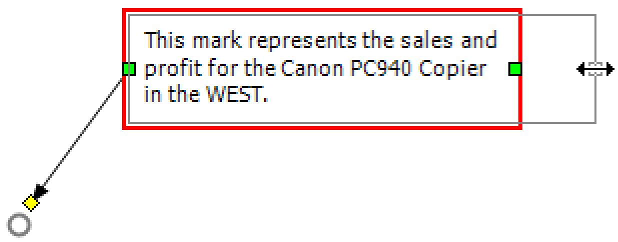
To reposition the body

- Click and drag the body of the selected annotation to a new position.



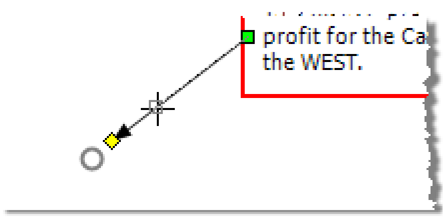
To resize the body

- Click and drag the body resize handle left and right. The text and height are automatically adjusted to fit the width of the body.



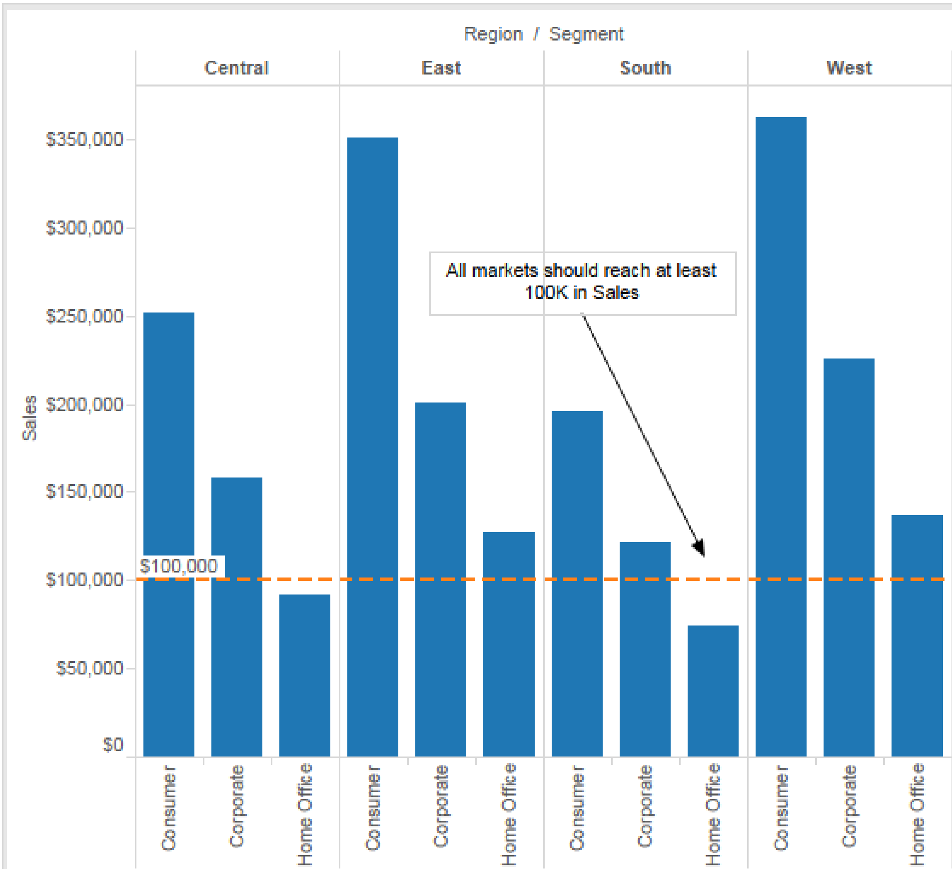
To resize the line

- Click and drag the line resize handle ◇



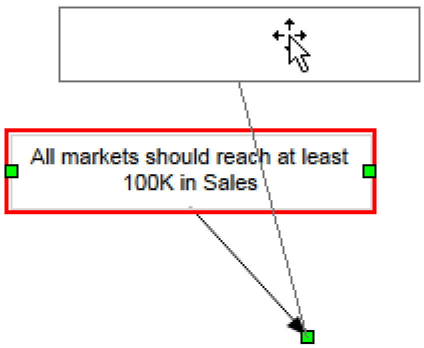
Point Annotations

A point annotation marks a specific point in the view, such as a reference line or a value on an axis. Point annotations display as text with a line. When you select a point annotation, several resize handles display. Use these handles to reposition and resize the body and line.



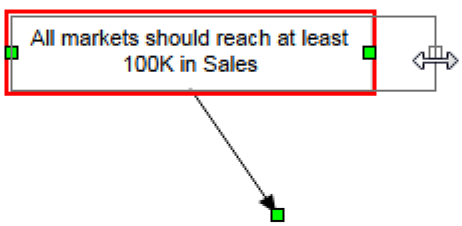
To reposition the body:

- Click and drag the body of the selected annotation to a new position. As you move the body, the line is automatically resized so that it continues to point at the specific point you selected.



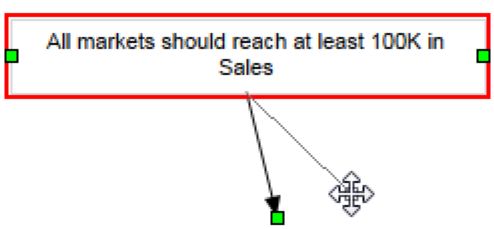
To resize the body:

- Click and drag the side resize handles left and right. The text and height are automatically adjusted to fit the width of the body.



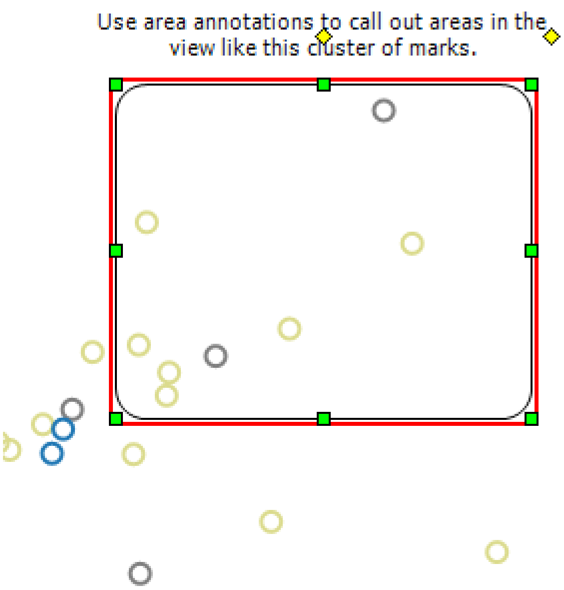
To move the line end point:

- Click and drag the end point of the line so that it points at a new location.



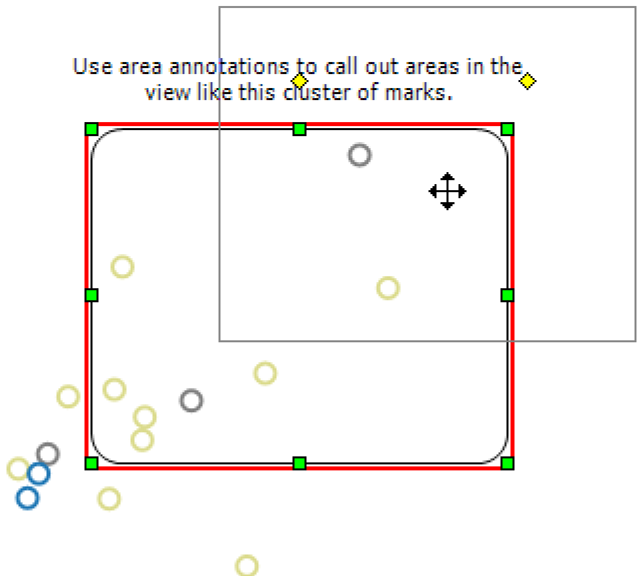
Area Annotations

An area annotation is a way to highlight or call out an area in the view. Area annotations are not associated with any particular mark, in fact, these annotations are commonly used to call out several marks. When you select an area annotation, several resize handles and two text handles display. Use these handles to reposition and resize the box and text.



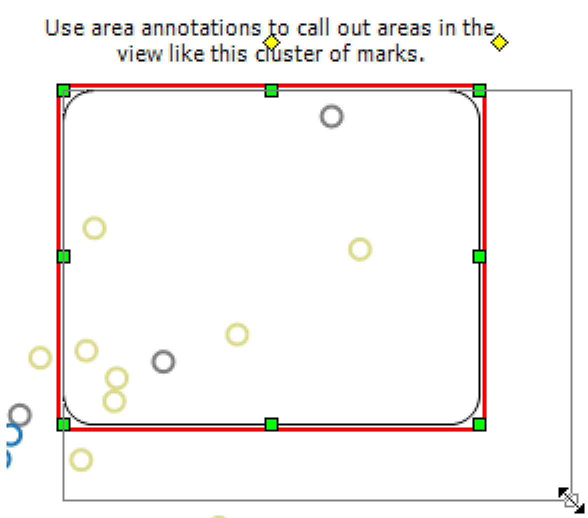
To reposition the box

- Click and drag the box of the selected annotation to a new position.



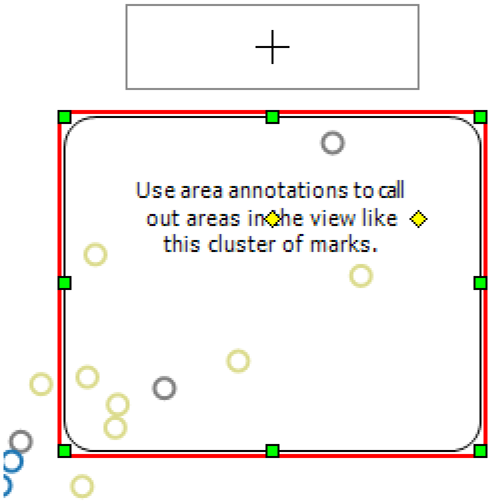
To resize the box

- Click and drag one of the box resize handles



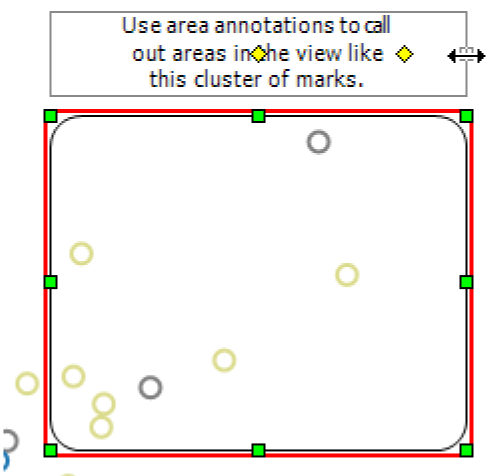
To reposition the text

- Click and drag the center text handle to a new position.



To resize the text width

- Click and drag the right text handle ◆ left and right. The text height is automatically adjusted to fit the width.



Format annotations

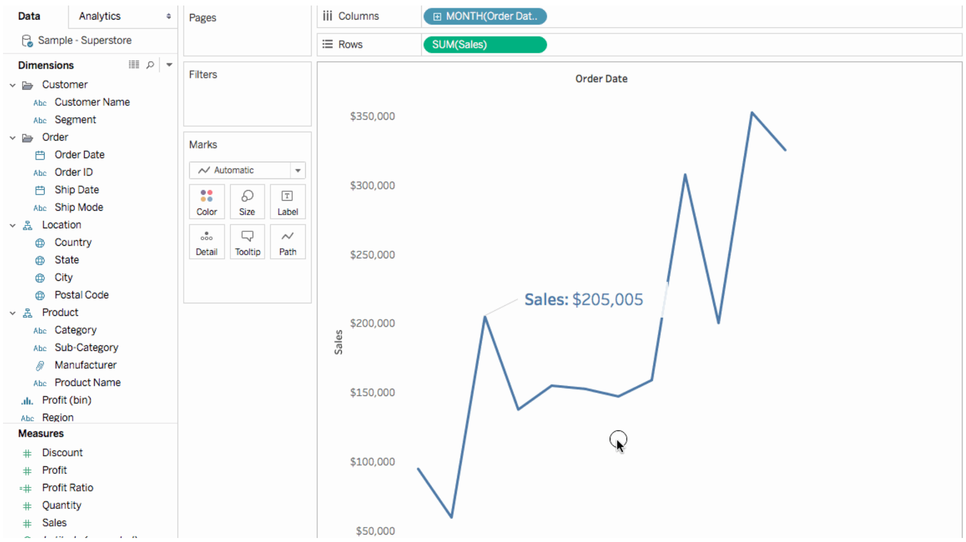
You can modify the text, body, and line of an annotation. For example, you can specify whether the body should be a box, a single edge, or not shown at all. Additionally, you can specify whether the lines on mark and point annotations end with an arrow, dot, or a simple line.

To format annotations:

1. Select one or more annotations, right-click (control-click on Mac) one of the selected annotations, and then select **Format**.

   The **Format** pane opens on the left-side of the workspace, over the **Data** pane.

2. In the **Format** pane, use the drop-downs to specify font properties, text alignment, line style, and shading.



Remove an annotation

To remove annotations from the viz:

1. Select one or more annotations in the viz.

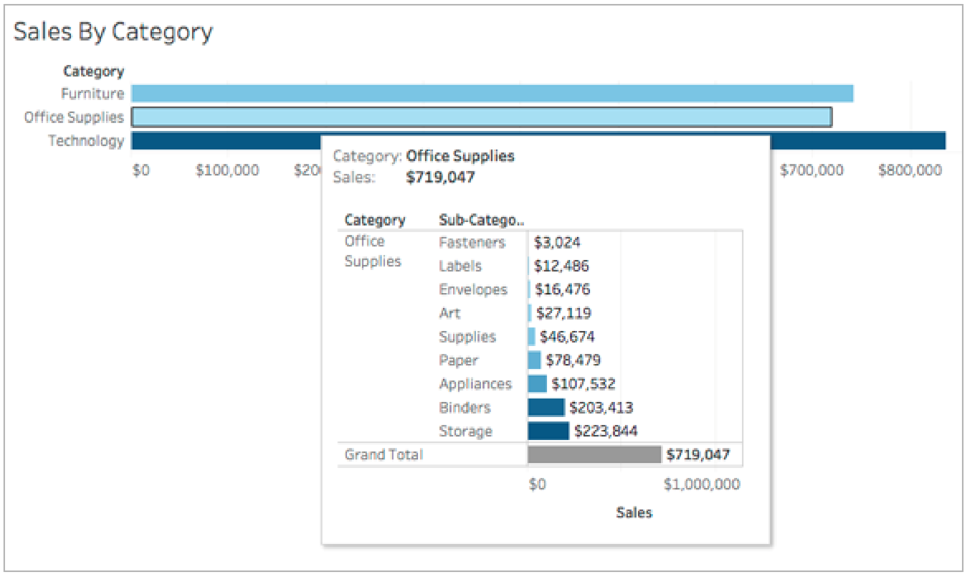
2. Right-click (control-click on Mac) one of the selected annotations and select **Remove**.

See Also

Show, Hide, and Format Mark Labels on page 1378

Create Views in Tooltips (Viz in Tooltip)

As you craft views and look for ways to reveal more details about data to your audience, you can embed visualizations within tooltips—also known as "Viz in Tooltip."



When a user hovers over a mark, the tooltip displays relevant data and details from another visualization filtered to that mark.

You can show related vizzes in tooltips to help your audience engage with the data at a different or deeper level, while keeping them in the current context and maximizing the space available for the current view.

Watch a video: To see related concepts demonstrated in Tableau, watch Viz in Tooltip, a 6-minute free training video. Use your tableau.com account to sign in. For more advanced tips, see Next Level Viz in Tooltip, a 55-minute free video presentation.

For information on how to edit regular tooltips, see **Format tooltips** on page 2941. To see related concepts demonstrated in Tableau, watch Basic Tooltips, a 7-minute free training video. Use your tableau.com account to sign in.

General steps to create a Viz in Tooltip

1. Create a visualization in the *source* worksheet in Tableau.
2. Create a visualization in a *target* worksheet view to serve as the Viz in Tooltip. Give the worksheet a name that helps you identify it as a Viz in Tooltip.
3. In the source worksheet, click Tooltip in the Marks card. In the Tooltip Editor, insert a reference to the Viz in Tooltip target worksheet. For details, see **Configure a Viz in Tooltip** on the facing page.
4. **Optional**: Hide the target sheet for the Viz in Tooltip. For details, see **Create Views in**

Tooltips (Viz in Tooltip) on the previous page.

5. Test the resulting Viz in Tooltip by hovering over different marks in the source worksheet view. If the Viz in Tooltip is too large for the tooltip window, adjust the height and width of the target worksheet visualization. You might also consider simplifying the structure and detail in the target visualization. For details, see **Change the size of the Viz in Tooltip** on page 1481.

> **Note**: By default, Viz in Tooltip is filtered on *All Fields*. Change the level of detail for Viz in Tooltip by defining a filter on *Selected Fields*. For details, see **Change the filter for the Viz in Tooltip** on page 1482.

For information on how to edit regular tooltips, see **Format tooltips** on page 2941. To see related concepts demonstrated in Tableau, watch Basic Tooltips, a 7-minute free training video. Use your tableau.com account to sign in.

Configure a Viz in Tooltip

You will need a *source* worksheet visualization and a *target* worksheet visualization to create a Viz in Tooltip.

These steps use the example of a source view that is a map showing sales profits by state, and a target sheet with a chart that shows profits by product sub-category.

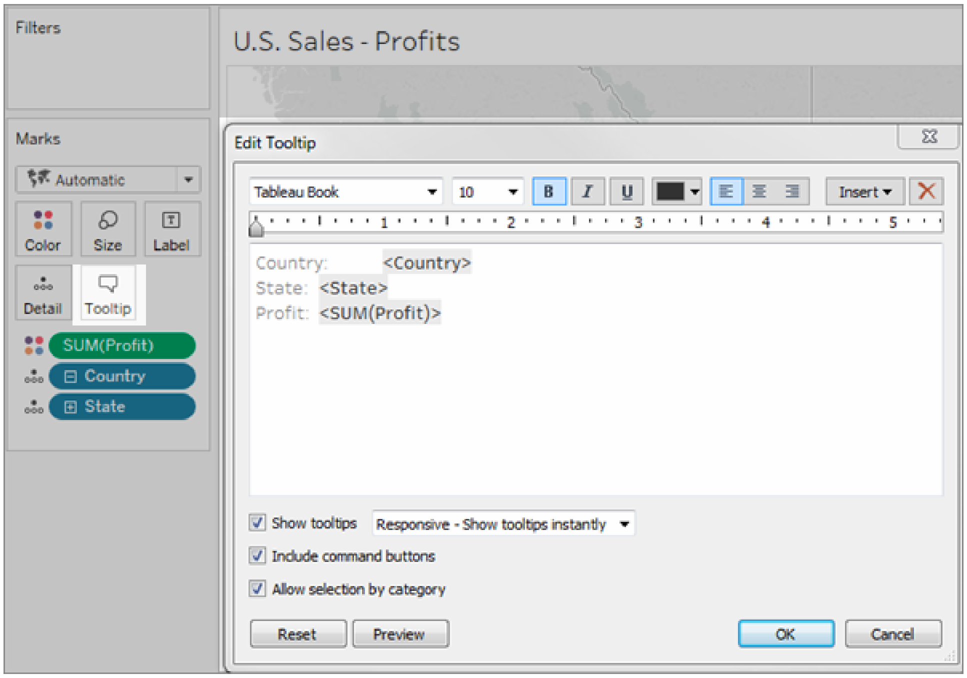
Create the source and target visualizations in a worksheet

1. In Tableau, figure out the worksheet that you want to use as your source visualization. Or build a new visualization on a new worksheet. This will be your source view.
2. Create a new worksheet to serve as the target visualization.

   Keep the number of filters used in the target view to a minimum.
3. Name the target view so that you will be able to identify it in a list of other sheets.

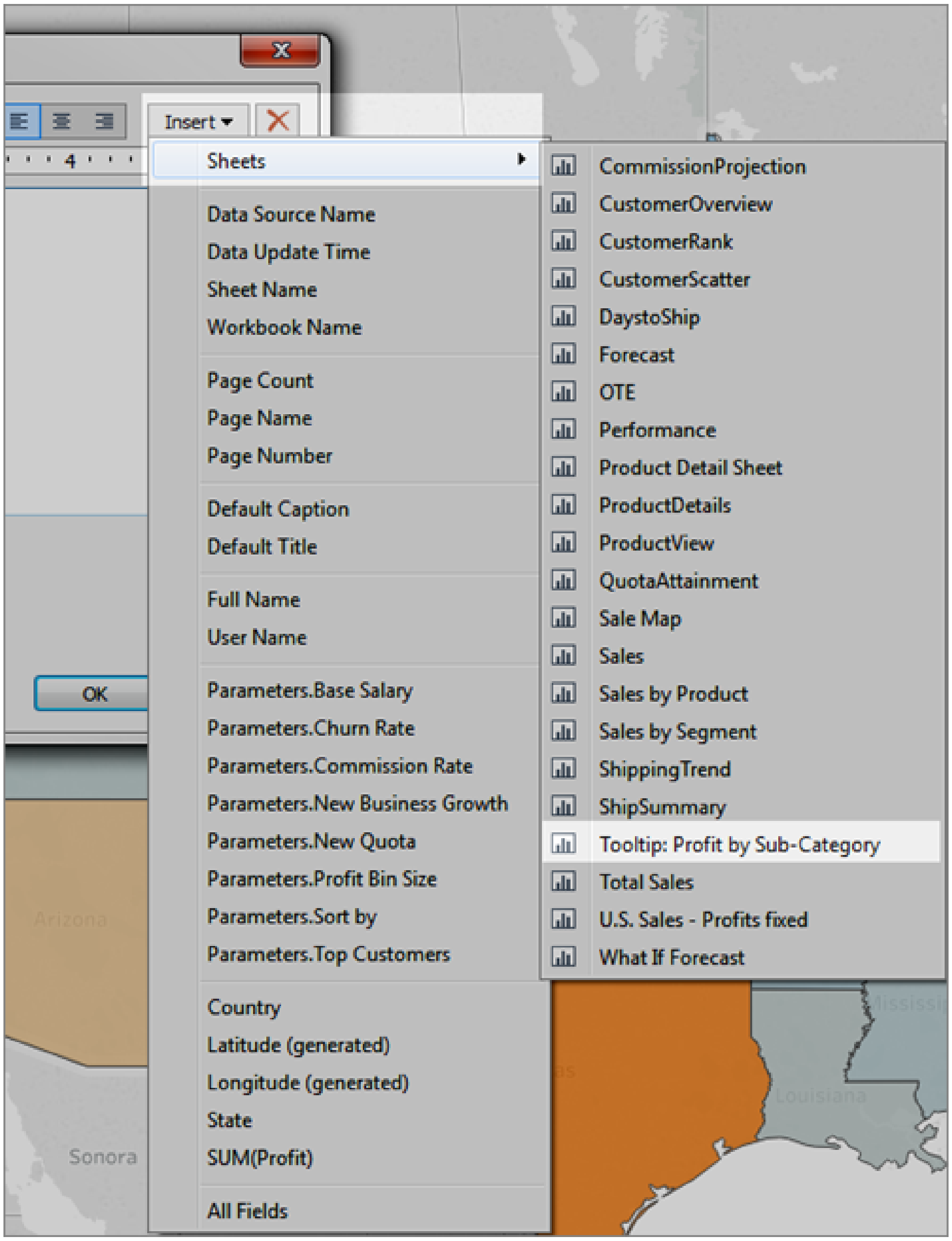
Insert a reference to the target worksheet in the source worksheet tooltip

1. In the source sheet, click the Tooltip button in the Marks card to open the Tooltip Editor.

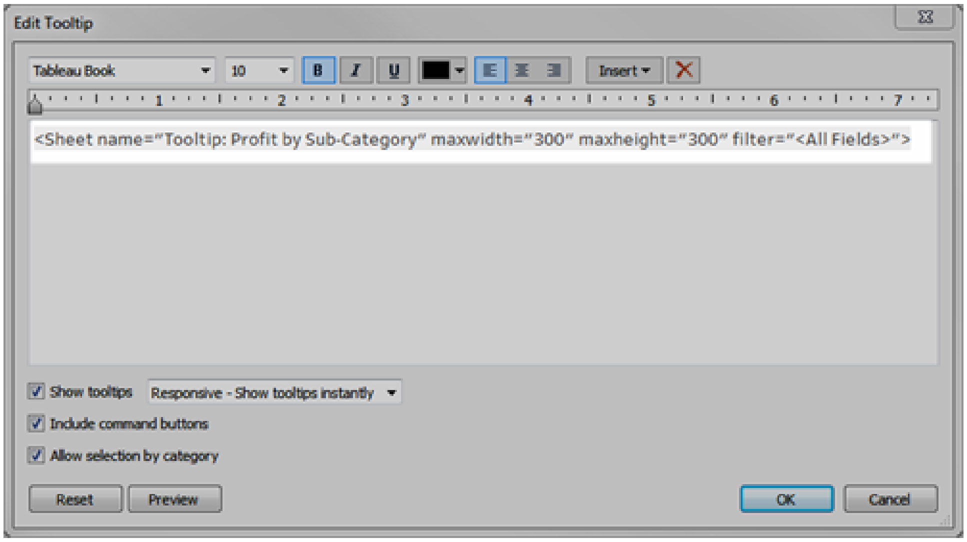


2. Click the **Insert** menu in the Tooltip Editor. In the **Insert** menu, select **Sheets**, and then select a target sheet.

   For example:



The markup for the Viz in Tooltip is automatically added. (In this example, the original markup fields have been removed.)



Example of markup automatically generated for Viz in Tooltip

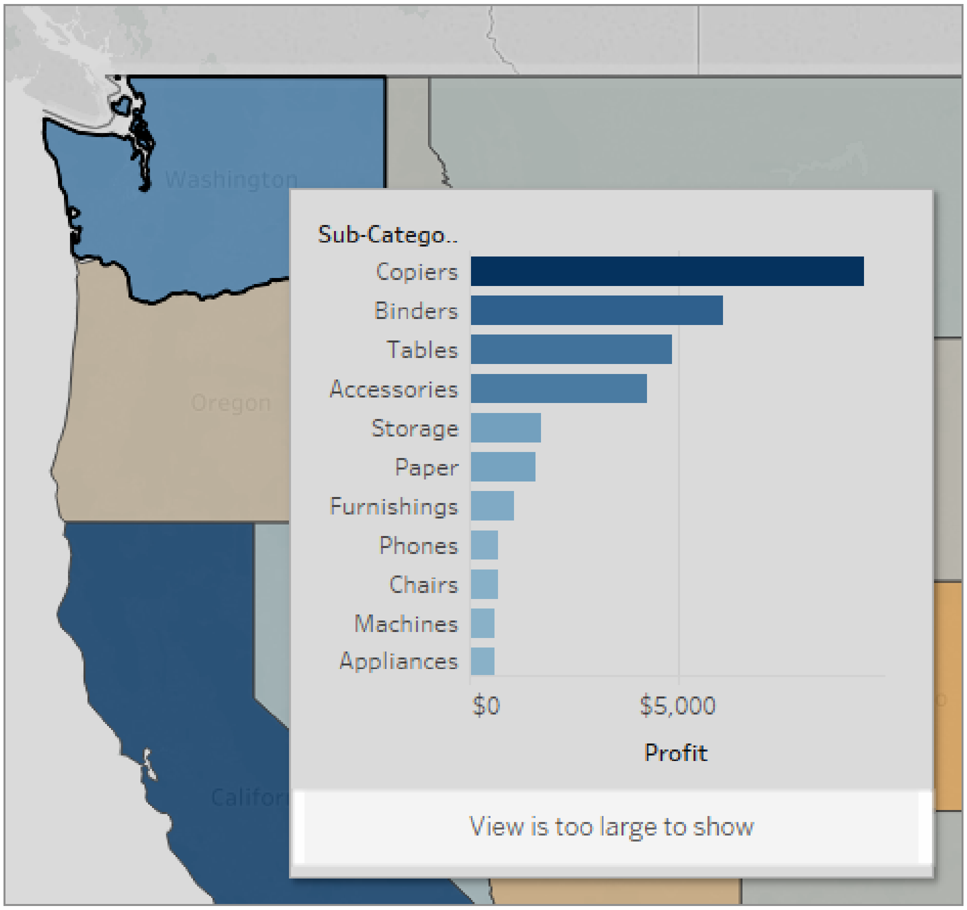
This is the resulting markup for this example:

```
<Sheet name="Tooltip: Profit by Sub-Category" maxwidth="300"
maxheight="300" filter="<All Fields>">
```

By default, the Viz in Tooltip is filtered on All Fields (filtered on all fields possible, and on the most specific level of detail).

Click **OK**.

3. Go back to the source sheet and test the Viz in Tooltip. Hover over different marks to see the resulting Viz in Tooltip. Make adjustments to the target view as necessary to improve the Viz in Tooltip.



The Viz in Tooltip in this example shows a message that indicates that some of the data in the target view is not being shown. If you see this message, you can adjust the height and width settings in the parameters for the Viz in Tooltip to make the view size larger.

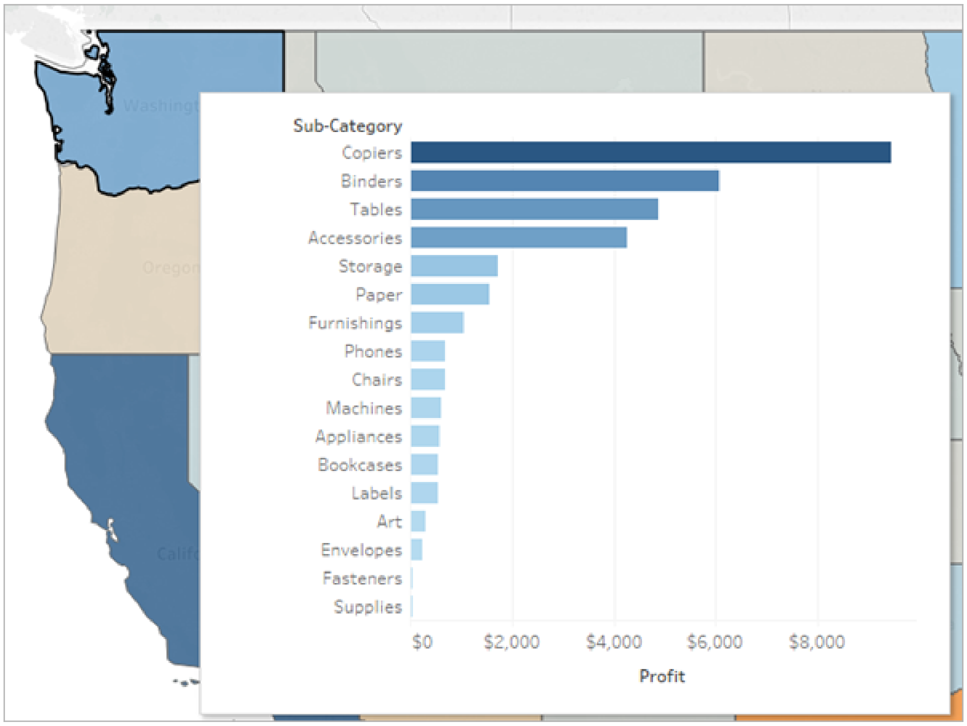
Change the size of the Viz in Tooltip

You can manually change the `maxwidth` and `maxheight` values to resize the Viz in Tooltip. The default size is 300 by 300 pixels. To change the size, manually replace `"300"` with another value. If you need to set the value greater than 600 pixels, you might want to reconsider whether the target view is a good candidate for Viz in Tooltip.

1. In the source sheet, click the Tooltip button in the Marks card to open the Tooltip Editor.
2. Select the number value for `maxwidth` and `maxheight` and type a different value to replace it. For example:

```
<Sheet name="Tooltip: Profit by Sub-Category" maxwidth="500"
maxheight="500" filter="<All Fields>">
```

3. Click **OK**



Example of a Viz in Tooltip set to 500 pixels width and height. The author also created more space for the header text in the in the target sheet view.

Change the filter for the Viz in Tooltip

By default, Viz in Tooltip is filtered on *All Fields*. This means the view is filtered on all dimensions in the current view (not including fields on the Filters shelf), at the most specific level of detail.

You can change the level of detail for Viz in Tooltip by defining a filter on selected fields, similar to filtering on **Selected Fields** in Filter Actions.

1. In the source sheet, click the Tooltip button in the Marks card to open the Tooltip Editor.
2. Place your cursor within the filter value (`filter="<place cursor here>"`), and then click the **Insert** menu to select an available field. Or, manually replace the <*All Fields*> value with the name of a field in the view. For example:

```
<Sheet name="Tooltip: Profit by Sub-Category" maxwidth="300"
maxheight="500" filter="<State>">
```

You can also filter more than one selected field by separating the field names with a comma. For example:

```
<Sheet name="Tooltip: Profit by Sub-Category" maxwidth="300"
maxheight="500" filter="<Country>,<State>">
```

To filter on date level, you will need to include the string that specifies the date level with the field name. For example:

```
<Sheet name="Tooltip: Profit by Month" maxwidth="300"
maxheight="300" filter="<MONTH(Order Date)>">
```

> **Note**: The date filter in the Viz in Tooltip script must match exactly the date level of the field in the view.

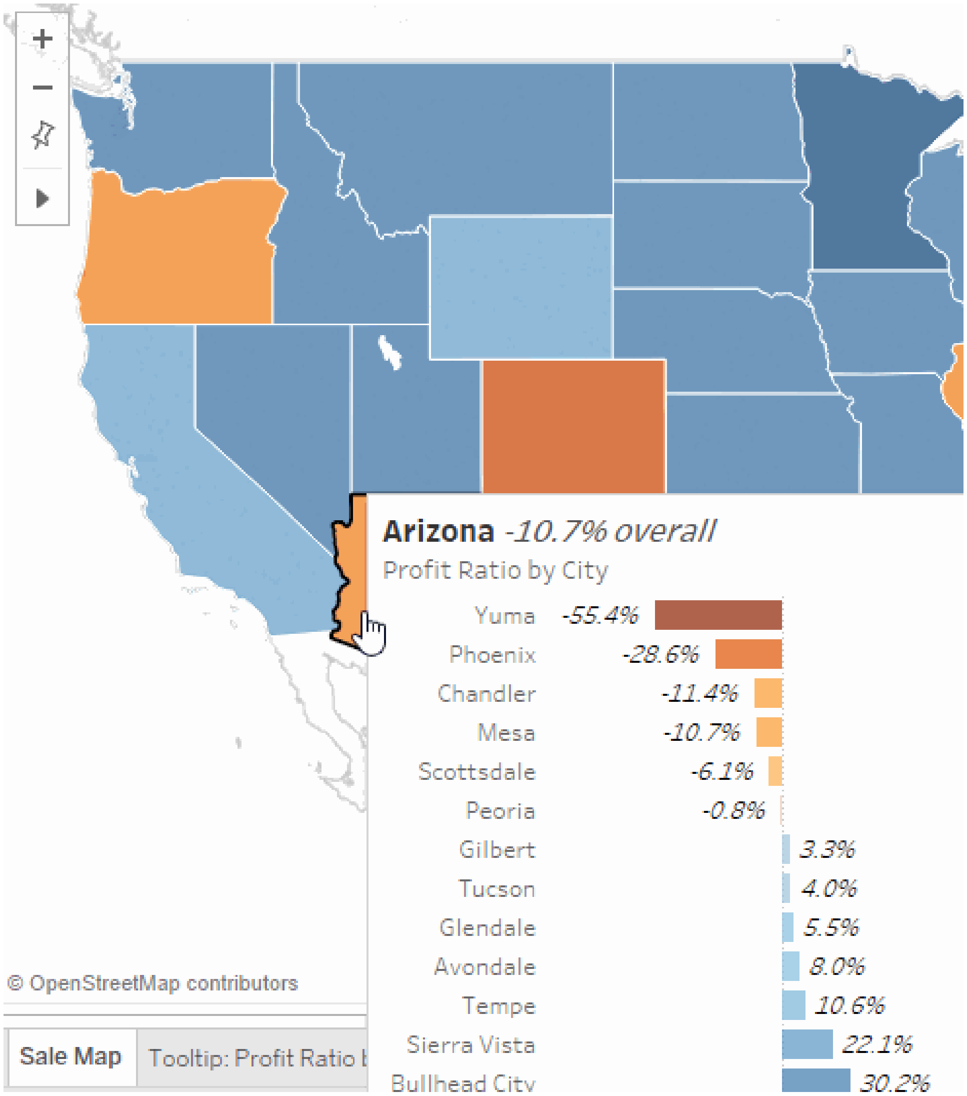
Hide or show a Viz in Tooltip worksheet

You can hide and show a worksheet used for Viz in Tooltip with options similar to those for dashboards and stories.

Hide a Viz in Tooltip worksheet

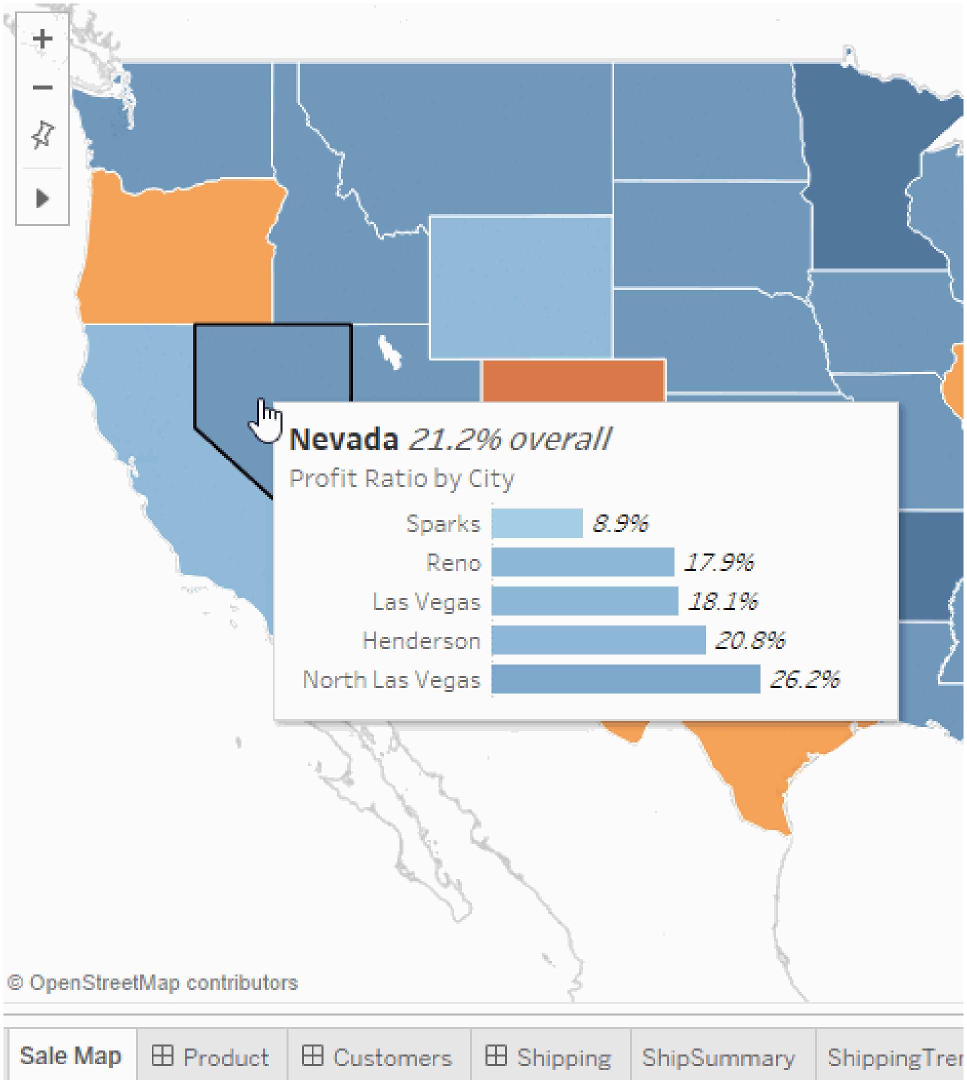
> **Note**: If the source sheet is hidden for a dashboard, you will first need to first unhide that worksheet from its dashboard to access it. For details, see **Manage Sheets in Dashboards and Stories** on page 2834.

To hide a Viz in Tooltip worksheet, from the tab for the target worksheet that is the Viz in Tooltip, select **Hide**.



Show a Viz in Tooltip worksheet

To show a Viz in Tooltip worksheet, from the tab for the source worksheet, select **Unhide All Sheets**.



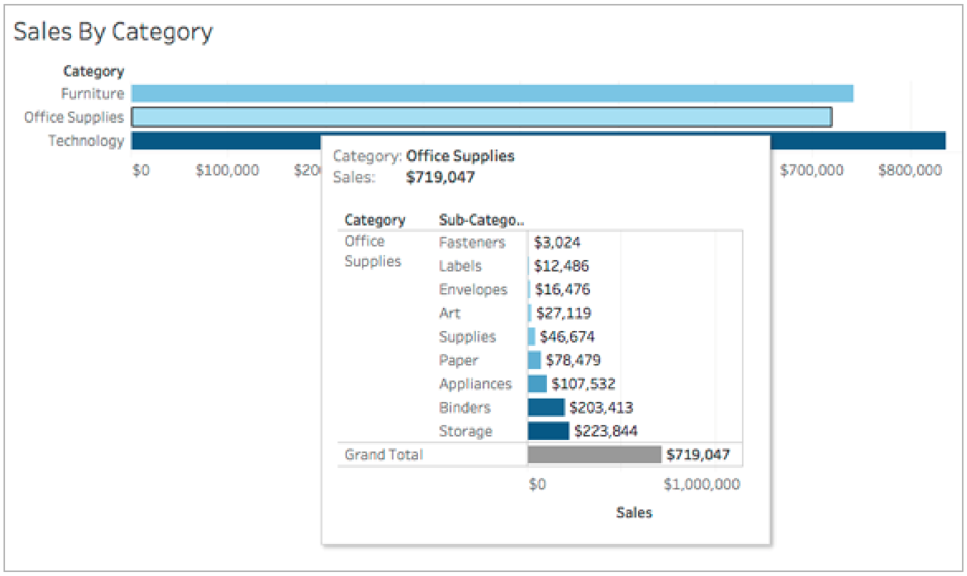
Examples of Viz in Tooltip

When you create a Viz in Tooltip, users can hover over a mark to examine details-on-demand, within the context of the original view. A Viz in Tooltip is a static image of data from another view that is relevant to a mark in the current view. Hovering over or selecting the mark reveals data from another sheet—filtered for that mark—in its tooltip.

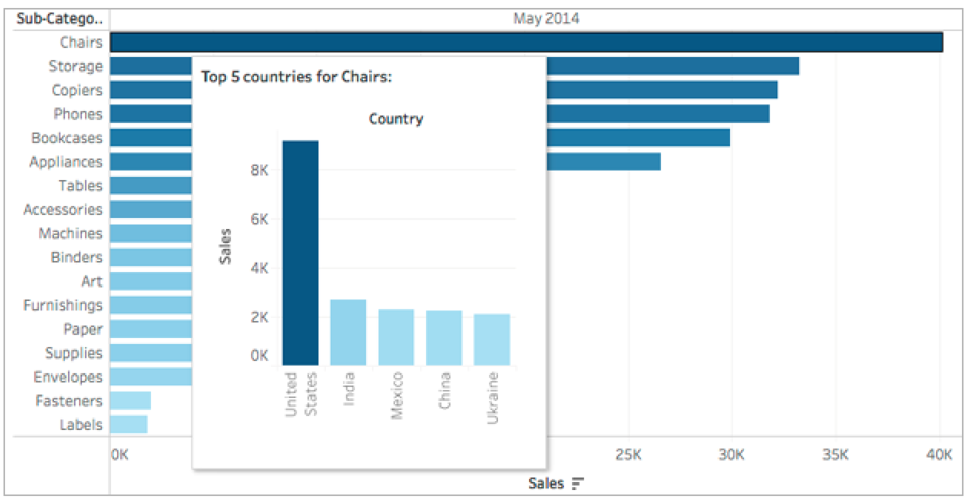
Watch a video: To see more examples and related concepts, see Next Level Viz in Tooltip, a 55-minute free video presentation. Use your tableau.com account to sign in.

Use Viz in Tooltip to show:

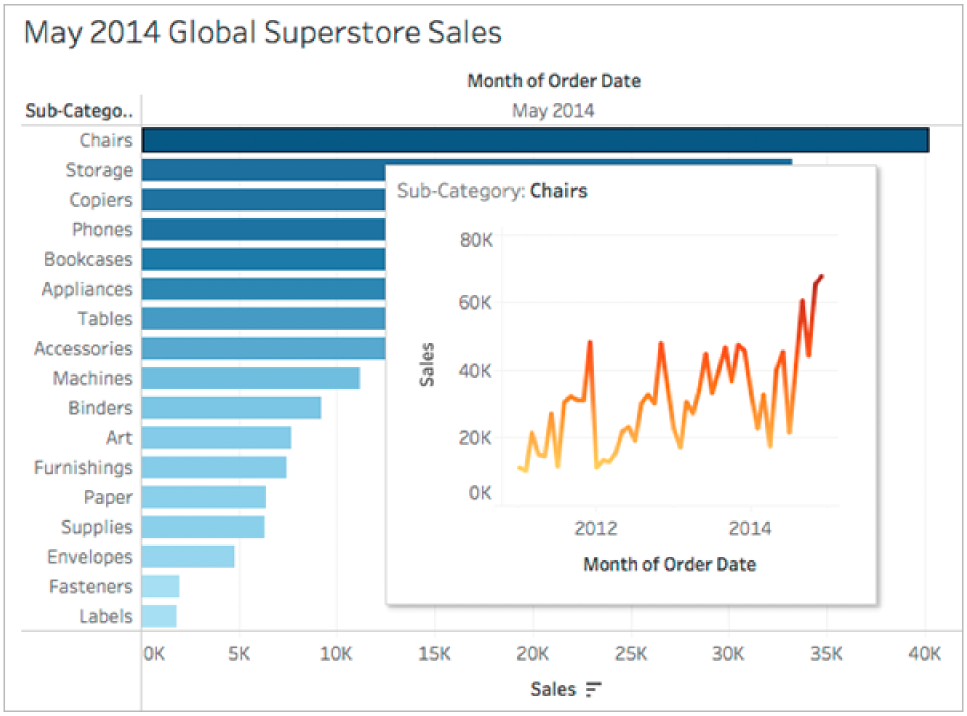
- Data at another level of detail



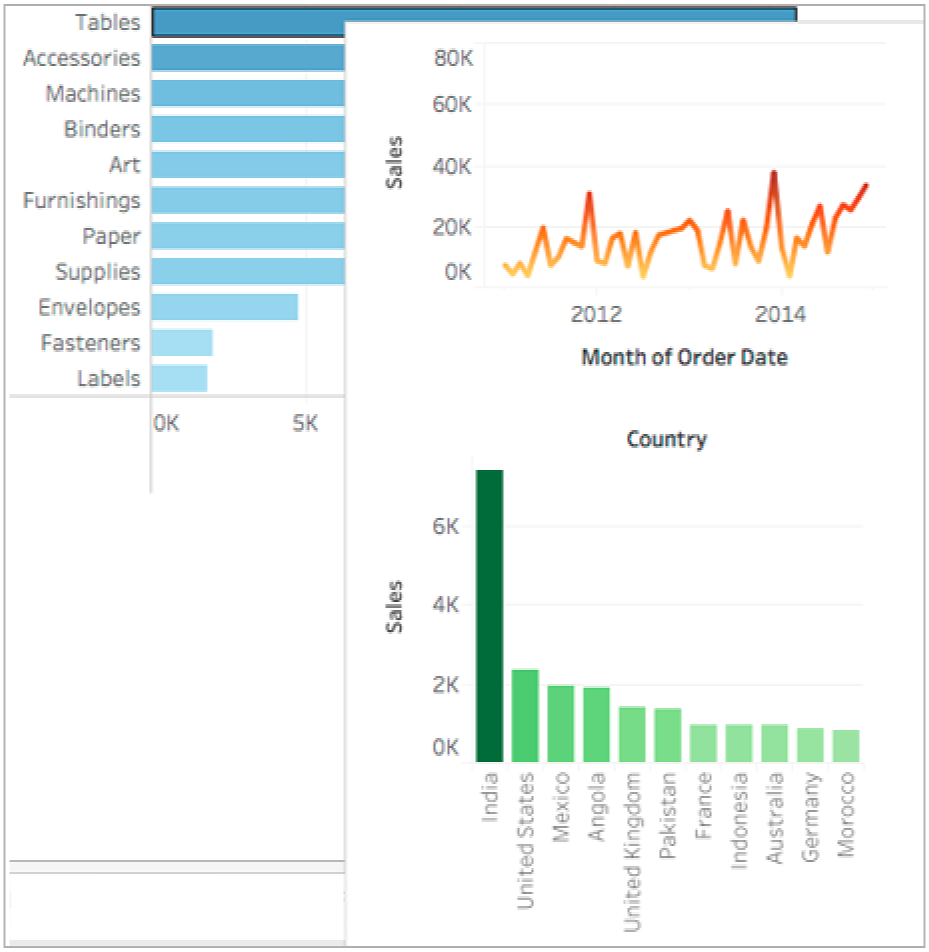
- Different, but relevant data



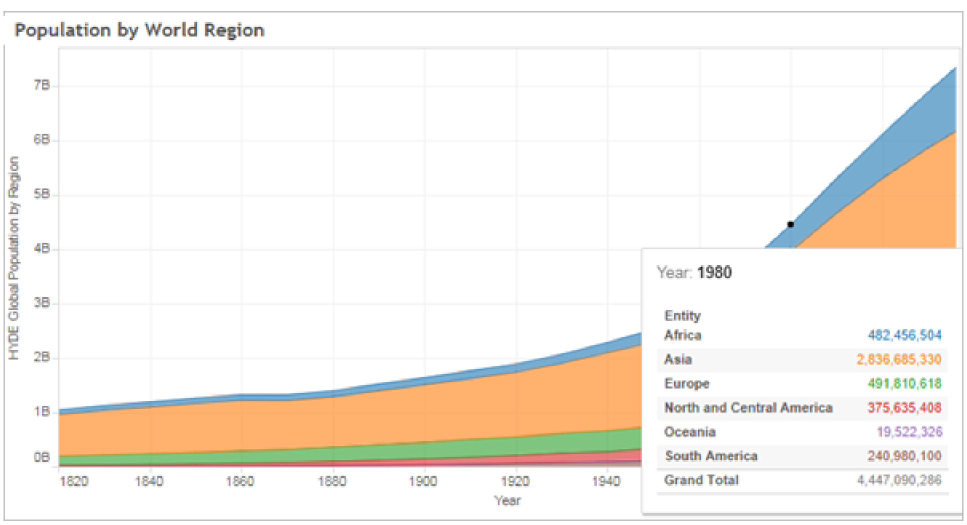
- How the value of a mark changes over time



- Multiple visualizations in one tooltip



- A legend for the main view



When you show related views in tooltips, you can help your audience engage with the data at a deeper level, while maximizing the space available for the current view.

Tips and notes on using and configuring Viz in Tooltip

- You can use worksheets to create a Viz in Tooltip, but you cannot use dashboards or stories to create a Viz in Tooltip.
- A Viz in Tooltip is a static image of a target view, not an interactive sheet. A Viz in Tooltip cannot have its own Viz in Tooltip.
- You will need a *source* worksheet visualization and a *target* worksheet visualization to create a Viz in Tooltip. You will need to create a target view to make it available in the Tooltip Editor.
- Use a standard naming scheme for the target sheets you plan to show in tooltips, such as Tooltip: *Name of View*. Using a standard naming scheme will help you keep track of views that you are using in tooltips.
- For the target view, consider the size of the view data and how the view will look in the tooltip. You will be able to specify the size for the Viz in Tooltip when you configure it, but you will need to check to see how it is displaying and possibly make adjustments to the view before you publish it.

  Remember that the target view is displayed in the context of the source view. Keeping the target visualization simple can help with performance and reduce cognitive load.
- If you click Show Me in the source sheet and it changes the view structure, all tooltip edits including Viz in Tooltip references will be reset. You will need to reconfigure the Viz in Tooltip.
- One target sheet can be referenced by one Viz in Tooltip source sheet at a time, because filters are applied directly to the referenced sheet. When a sheet is already being used as a target sheet in a tooltip, it becomes unavailable for selection in the Tooltip Editor.
- By default, Viz in Tooltip is filtered on *All Fields*, which considers all fields in the view (at the most specific level of detail) when identifying matching records. You can change the level of detail for Viz in Tooltip by defining a filter on *Selected Fields*, similar to filtering on **Selected Fields** in Filter Actions. For related details, see **Create Advanced Highlight Actions** on page 1507.

  Filtering on *Selected Fields* does not work across different data sources. If the source and target views are using different data sources, filtering on *All Fields* will automatically detect the fields in common (if they share the same alias), and filter on them. Filtering on *Selected Fields*, however, will not work.

Actions

Add context and interactivity to your data using actions. Users interact with your visualizations by selecting marks, or hovering, or clicking a menu, and the actions you set up can respond with navigation and changes in the view.

For example, in a dashboard showing home sales by neighborhood, you could use actions to display relevant information for a selected neighborhood. Selecting a neighborhood in one view can trigger an action that highlights the related houses in a map view, filters a list of the houses sold, then opens an external web page showing census data for the neighborhood. For related information and examples, see A Rough Guide to Dashboard Actions on the Tableau Public blog.

> **Watch a Video**: To see related concepts demonstrated in Tableau, watch the free training video Dashboard Interactivity Using Actions (5:43 minutes). Use your tableau.com account to sign in.

Here's how you use the different types of actions:

- **Filter**. Use the data from one view to filter data in another to help guide analysis.
- **Highlight**. Call attention to marks of interest by coloring specific marks and dimming all others.
- **Go to URL**. Create hyperlinks to external resources, such as a web page, file, or another Tableau worksheet.
- **Go to Sheet**. Simplify navigation to other worksheets, dashboards, or stories.
- **Change Parameter**. Let users change parameter values by directly interacting with marks on a viz.
- **Change Set Values**. Let users change the values in a set by directly interacting with marks on a viz.

Filter Actions

Filter actions send information between worksheets. Typically, a filter action sends information from a selected mark to another sheet showing related information. Behind the scenes, filter actions send data values from the relevant source fields as filters to the target sheet.

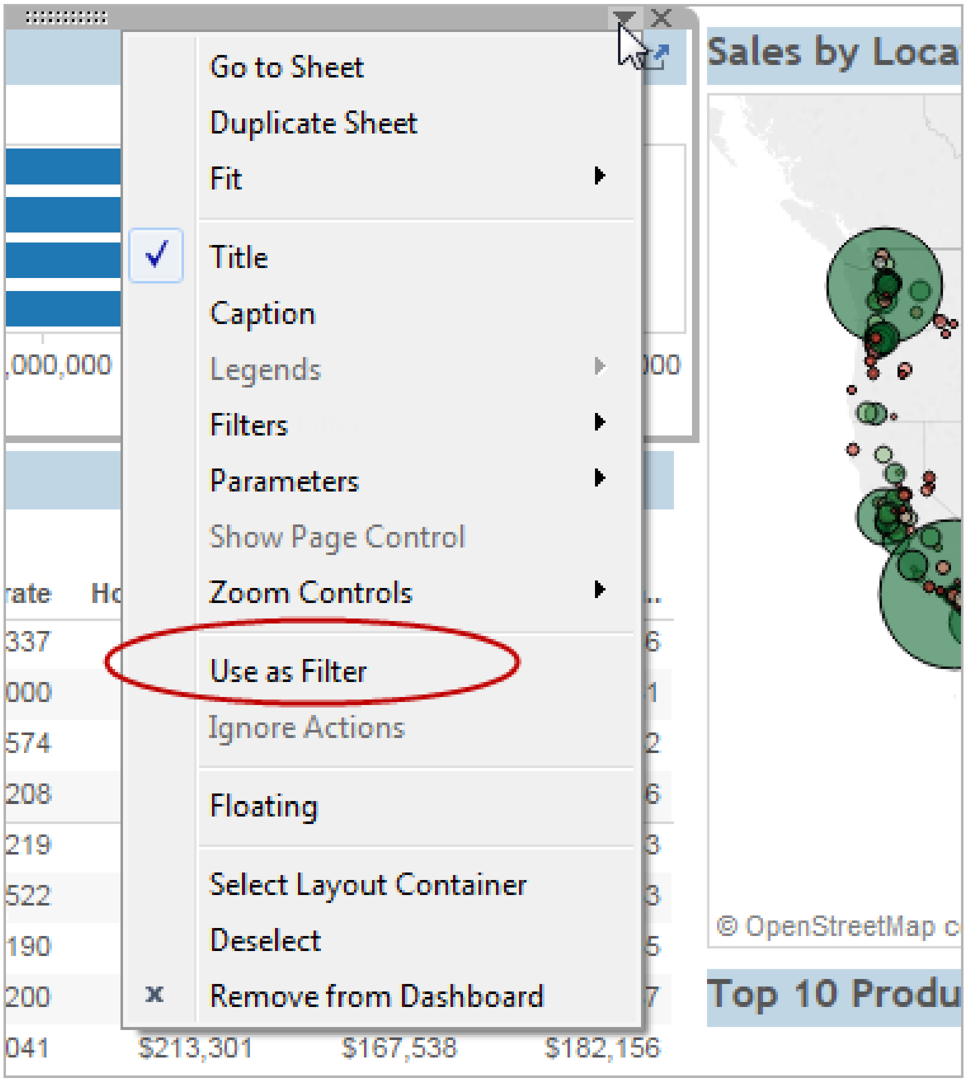
For example, in a view showing the sales price of houses, when you select a particular house, a filter action can show all comparable houses in a different view. The source fields for the filter might contain sales price and square footage.

Watch a video to see real-world examples of actions. Filter actions are covered at the 2:20 mark.

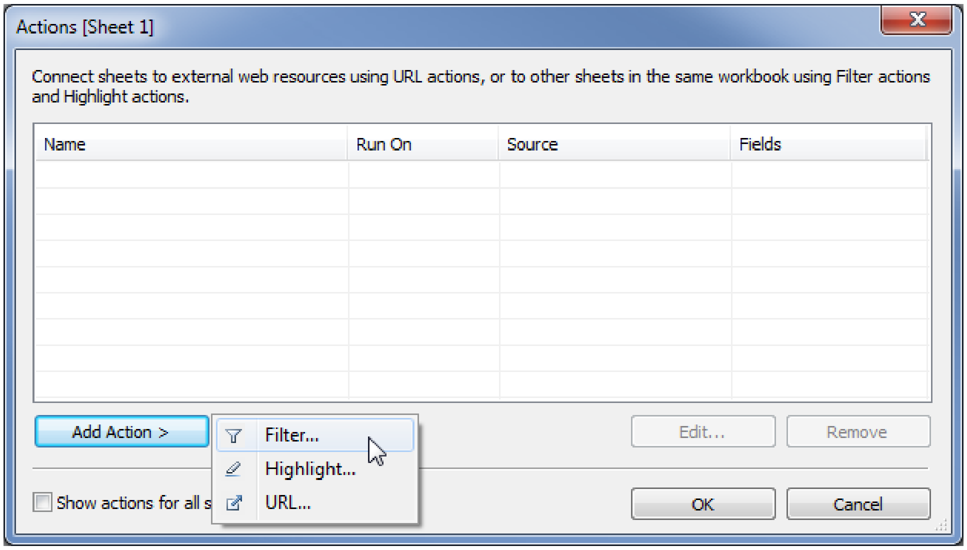
Create or edit a filter action

1. Do either of the following:
   - On a worksheet, select **Worksheet** > **Actions**.
   - On a dashboard, select **Dashboard** > **Actions**.

     From the drop-down menu of a dashboard sheet, you can also select **Use as Filter**. In the Actions dialog box, "generated" appears in the default names of actions created this way.



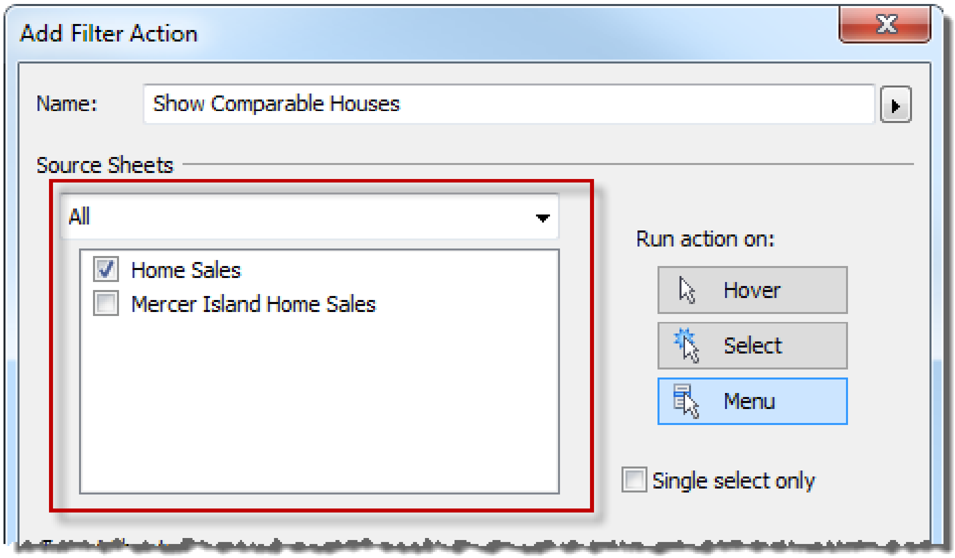
2. In the Actions dialog box, click **Add Action**, and then select **Filter**. Or select an existing action, and choose **Edit**.



3. Specify a name for the action.

> **Tip:** If you plan to launch an action from a tooltip menu, use a descriptive name so users will understand the action's purpose. From the menu to the right of the Name box, you can insert variables that are drawn from the values of selected fields.

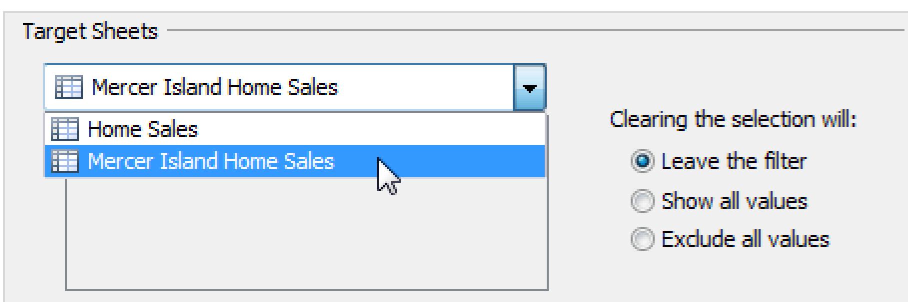
4. Select a source sheet or data source. If you select a data source or dashboard, you can select related sheets you want to launch the action from.



5. Specify how the action runs:

   - **Hover** - Runs when you mouse over marks in the view.
   - **Select** - Runs when you click marks in the view. If you want to avoid running the action when multiple marks are selected, also select **Single-select only**.
   - **Menu** - Runs when you right-click (Windows) or Control-click (macOS) a mark in the view, and then click an option in the tooltip menu.

6. Select a target sheet. When you select a dashboard, you can select one or more sheets within it.

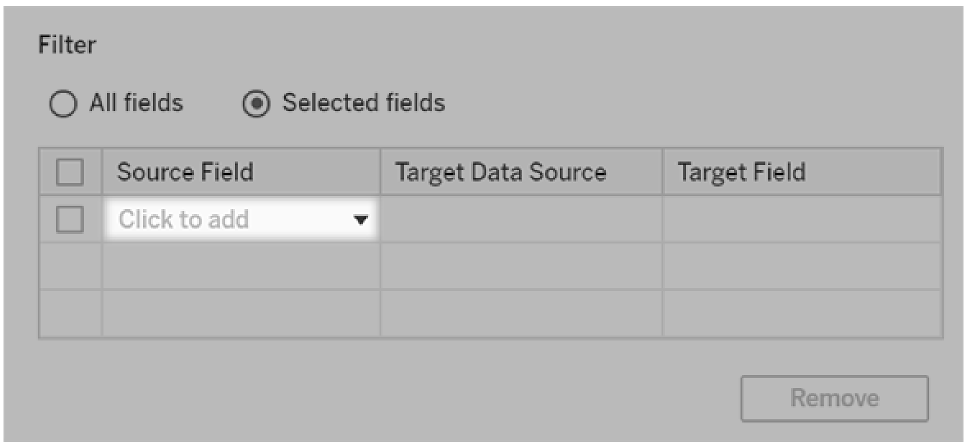


7. Specify what happens when the selection is cleared in the view:

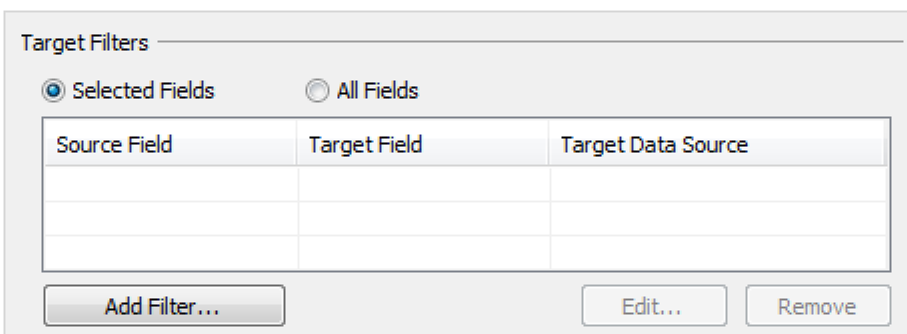
   - **Leave the filter** - Continues to show filtered results on the target sheets. (In web-authoring mode, this is labeled **Keep filtered values**.)
   - **Show all values** - Changes the filter to include all values.

- **Exclude all values** - Changes the filter to exclude all values. This option is useful when you're building dashboards that only show some sheets if a value in another sheet is selected.

8. Specify the data that you want to show on the target sheets. You can filter on **All Fields** or **Selected Fields**.
9. If you chose **Selected Fields**, do either of the following:
   - In Tableau Online or Tableau Server, click a drop-down menu in the **Source Field** column, and select a field. Then select a target data source and field.



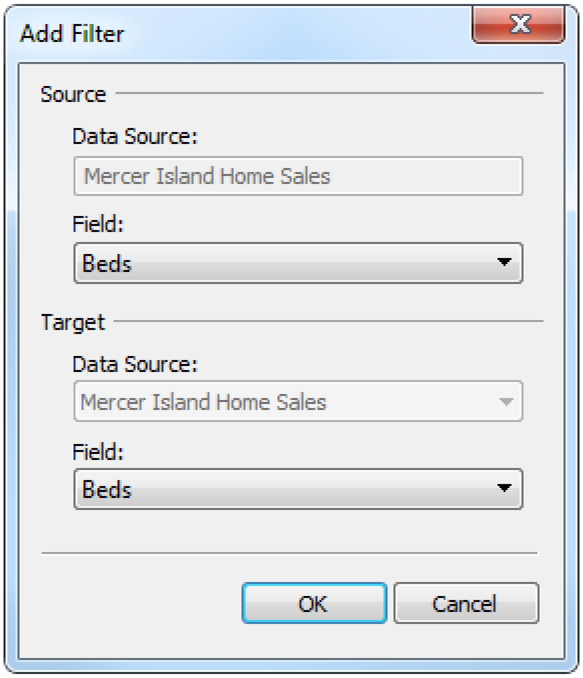
   - In Tableau Desktop, click **Add Filter**.



In the Add Filter dialog box, select source and target data sources and fields.

When you run the action from a specific mark on the source sheet, a filter is added to the target sheet that only includes values for the target field that match the source field. In the comparable houses example, if the Source Field and Target

Field are both set to Beds, when you click a sheet link for a house with three bedrooms, the target worksheet shows only three-bedroom houses.



Note: Filter actions that depend on a user function, such as USERNAME(), will not work because row-level security restricts access to the data.

Understanding available target fields

In the Add Filter dialog box, the fields available in the Target Field drop-down list are limited to the data type of the Source Field. For example, if you select a text field for the source, only text fields are available as targets.

If you are connected to a relational data source, you can add sheet links across data sources even if the field names don't match. For example, if one data source has a Latitude field while another has a Lat field, you can associate the fields using the drop down lists in the Add Filter dialog box. When using a multidimensional data source, the destination sheet must use the same data source as the source sheet, and the source and target field names must match. (In Tableau, multidimensional data sources are supported only in Windows.)

Highlight Actions

Highlight actions allow you to call attention to marks of interest by coloring specific marks and dimming all others. You can highlight marks in the view using a variety of tools. For example, you can manually select the marks you want to highlight, use the legend to select related marks, use the highlighter to search for marks in context or create an advanced highlight action.

The following table describes the different methods you can use to highlight marks in a view, dashboard, or story.

Highlight method	Benefits	When you might use this
Select marks on page 2722	• Manually select a group of marks to highlight in a view. • Your selection is saved with the workbook.	• When you want to manually highlight a selection of marks and dim all others. • Works well with small domains or views with a small amount of data.
Legends	• Supports one-way and two-way highlighting. • Highlight on color, size, or shape. • You can disable or enable the highlighting action for the workbook or sheets from the toolbar. • Your selection is saved with the workbook and can be included in dashboards and stories and when publishing.	• When you want to focus on select members in a view and dim all others. • When you want to highlight using onlythe legend or the legend and the view. • Works well with small domains or views with a small amount of data.
Highlighter	• Search for data points in a view using keywords or select from a drop-down list. • Highlight marks while maintaining the context of the other data	• When you want to highlight a mark or group of marks for a discrete field that is included in the view.

	points. • Values automatically update when the underlying data is updated. • Highlighters added to worksheets also appear on dashboards and stories.	• When you want to do ad hoc comparisons with instant highlighting. • Works well with large domains and large amounts of data.
Actions (Tableau Desktop only)	• Highlight data based on criteria that you define. • Specify the source and target sheets to apply the highlight action to. • Specify the fields to use for highlighting. • You can specify different types of actions to run on the same click (for example, filter and highlight).	• When you want to build interactive exploration into a dashboard. • When you want to highlight data points in a dashboard using specific fields.

Legend Highlighting

You can use legend highlighting to focus your users' attention on specific marks in a view. When legend highlighting is on, marks associated with the legend item are highlighted and the other marks are dimmed.

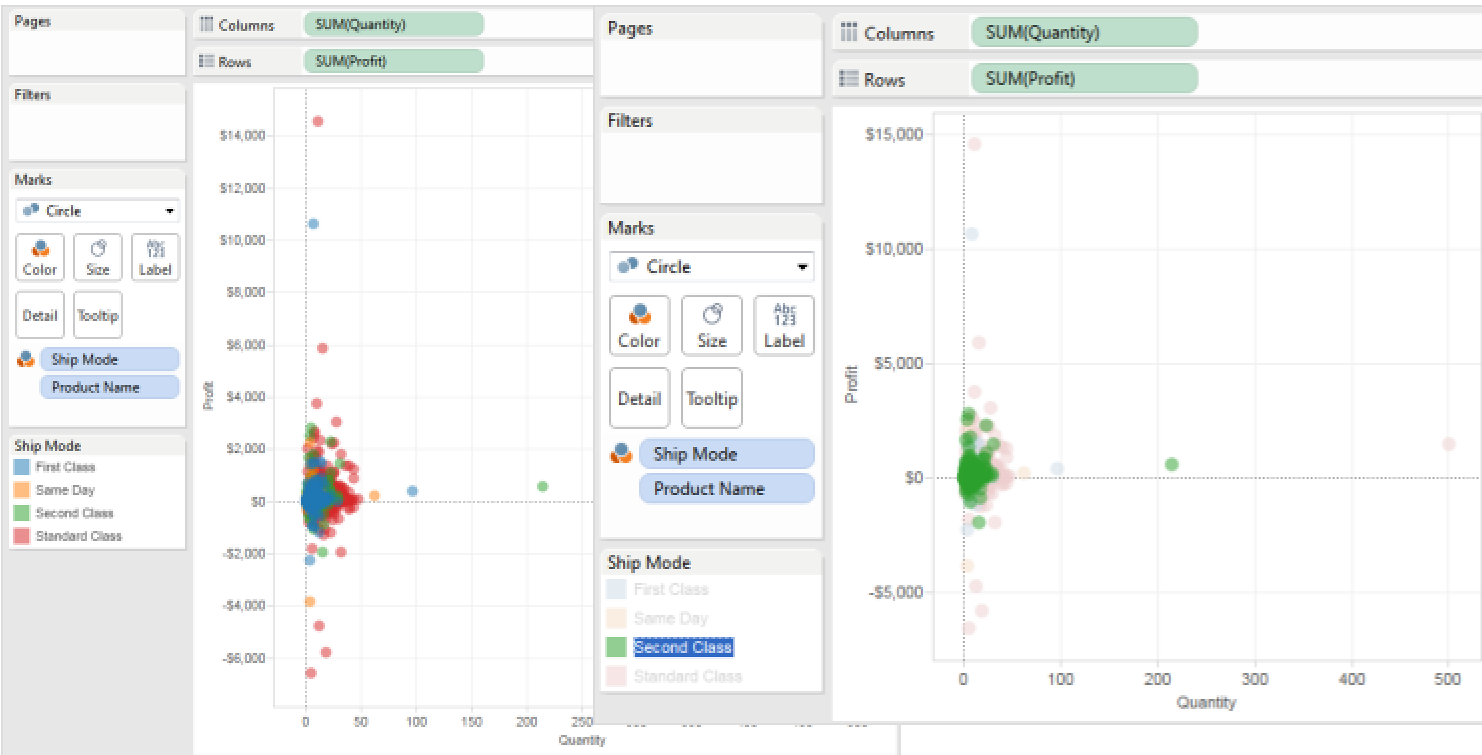
You can enable either one-way or two-way highlighting to highlight marks in the view. The icon at the top of the legend indicates which mode you're using.

- One-way highlighting enables you to highlight marks using the values in the legend.
- Two-way highlighting enables you to highlight marks using either the legend or the view. This is the default. When you highlight marks in the view, the matching member in the legend is also highlighted.

You can also turn off highlighting for the workbook or the sheet from the toolbar. This action hides the highlight icon on the legend. For more information about using the toolbar highlighting option, see **Highlight Toolbar Button** on page 1505.

In this example, the views below show the relationship between order quantity and profit for several products. The view on the left uses the standard functionality for the color legend, where all marks are colored based on their shipping mode. The view on the right uses legend highlighting to call out the products that were delivered via Second Class shipping.

Normal Color Legend **Color Legend Highlighting Enabled**



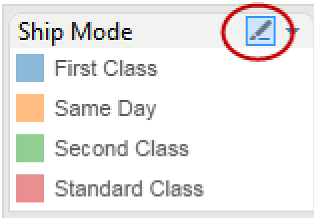
You can easily switch between legend highlighting and normal modes using the legend card menu. For color legends, if you like how a view is highlighted, you can assign the highlight colors to the color palette. The old colors are replaced with the highlight colors.

Turn on legend highlighting

1. Click the **Highlight** button at the top of the legend.

   If you're using Tableau Desktop you can also turn on legend highlighting by selecting **Highlight Selected Items** on the legend card menu.

This example shows highlighting using the color legend. Color legend highlighting is turned on by default.



2. Select an item in the color legend.

   After legend highlighting is turned on, you can quickly focus on specific data in the view by selecting different items in the color legend. When you turn on legend highlighting, a highlight action is created that you can modify in the Actions dialog box.

Turn off legend highlighting

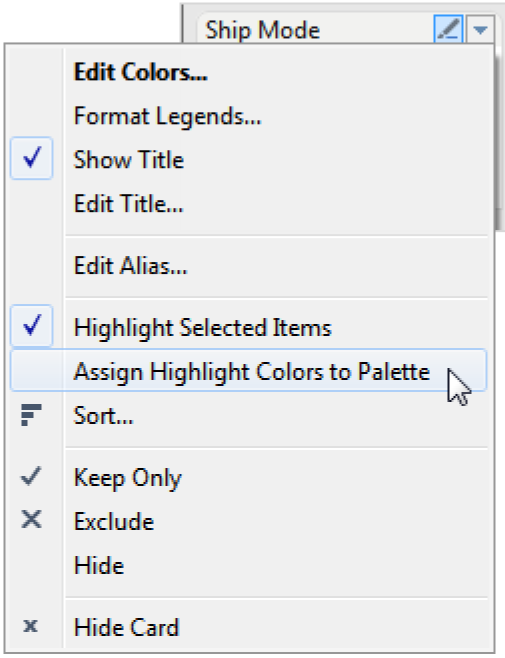
Click the **Highlight** button at the top of the legend. This changes highlighting to one-way and you can use the legend to highlight matching marks in the view.

Legend highlighting in Tableau Desktop

If you're using Tableau Desktop you can also turn off legend highlighting by selecting **Highlight Selected Items** on the legend card menu. When you turn legend highlighting off the action is removed from the Actions dialog box.

If you like how the view is highlighted and want to keep a specific member highlighted even when you turn off legend highlight mode, you can assign the highlight colors to the existing color palette. The original color legend is discarded and the highlight colors become the new color palette for the legend.

To assign the highlight colors to the color palette, select **Assign Highlight Colors to Palette** on the color legend card menu.



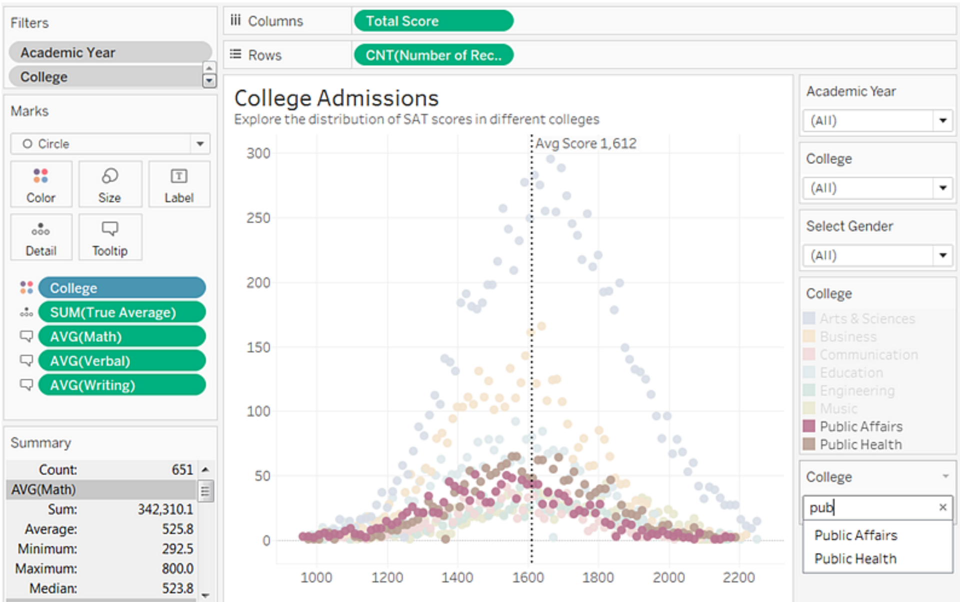
Highlight Data Points in Context

When you have a view with a large amount of data you might want to explore your data interactively and highlight a specific mark or group of marks while still maintaining the context of where those marks show in your view.

To do this you can turn on the Highlighter for one or more discrete fields that are included in your view and that affect the level of detail (see **How dimensions affect the level of detail in the view** on page 319 for more information).

You can use keywords to search for matching data points. The highlighter immediately highlights the marks that match or partially match your keyword search. If you update the underlying data source for your view the data shown in the highlighter is automatically updated too.

In the example below, the Highlighter is turned on for the **College** field. Entering a partial keyword search for **Public** returns two possible matches. In the view, Tableau highlights the group of marks that match the partial search: **Public Affairs** and **Public Health**.



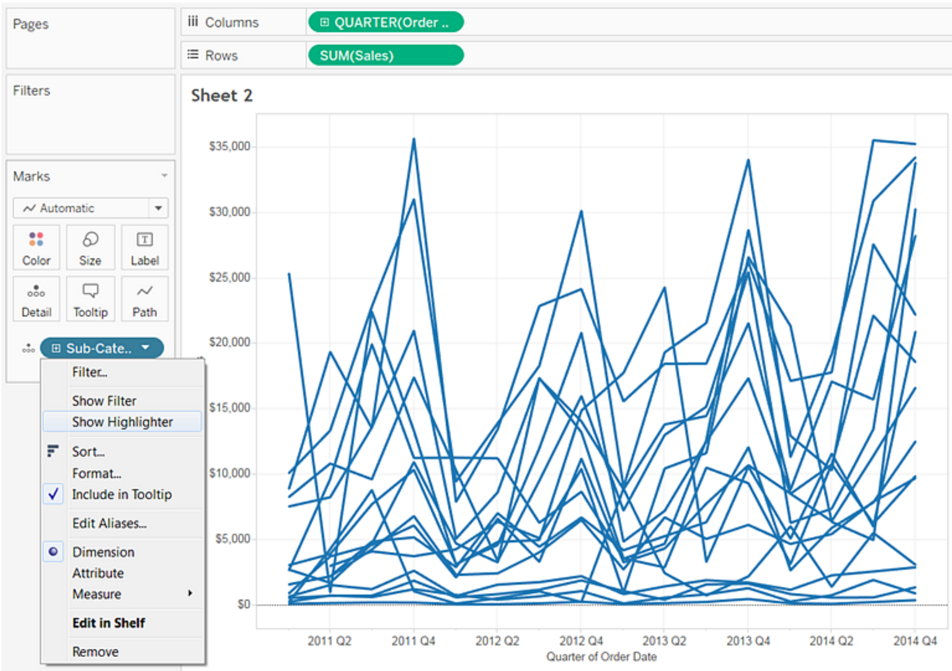
You can turn on highlighters for as many discrete fields in our view as you need. However, you can only highlight using one value set at a time. If you want to include the highlight option on dashboards and stories, turn them on in the worksheet before you add them to the dashboard, or select the worksheet on the dashboard, and then select **Analysis** > **Highlighters** from the top menu and select from the fields in the list.

Tableau Desktop only: If you want to change the format for the highlighter card after you turn it on, select **Format** > **Highlighter**.

> Highlighters work across worksheets on a dashboard if the same field is included across those worksheets. If the same field is not found, then no match is found for the highlighted value, and the values in those worksheets are dimmed. If you have multiple highlighters showing on a dashboard, only one highlighter can be active at a time. In this situation, the highlight shows for the highlighter that was used last.

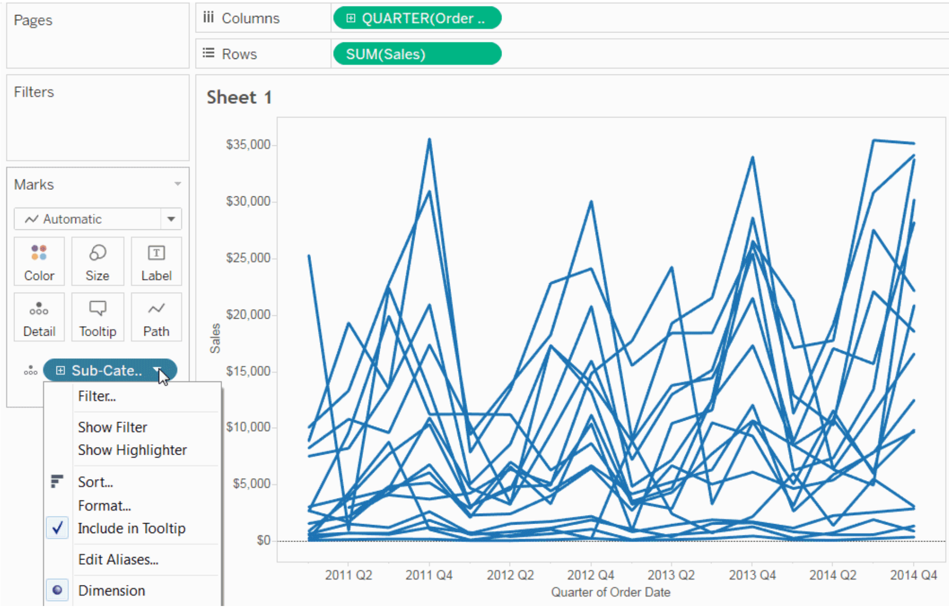
Turn on highlighting

1. Right-click on a discrete field (dimension) that is included in the view and impacts the level of detail in the view. Then select **Show Highlighter** from the context menu.



2. Click in the highlighter dialog box and do one of the following:

    - Enter a keyword to search for matching marks to highlight.
    - Enter a partial key word to search for all relevant matches that contain your search text.
    - Select an item in the drop-down list. You can select one item at a time.
    - Hover over items in the drop-down list to highlight marks in the view for an ad hoc analysis experience.



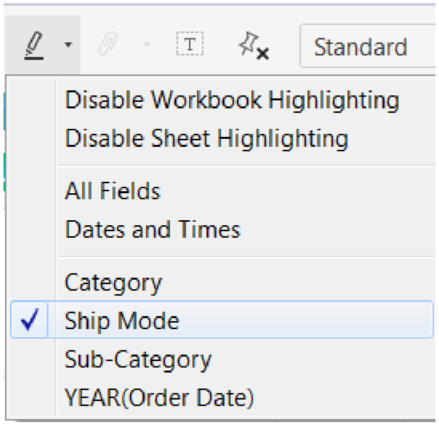
Click the image above to replay the animation.

3. Repeat steps 1 and 2 to add additional highlighters.

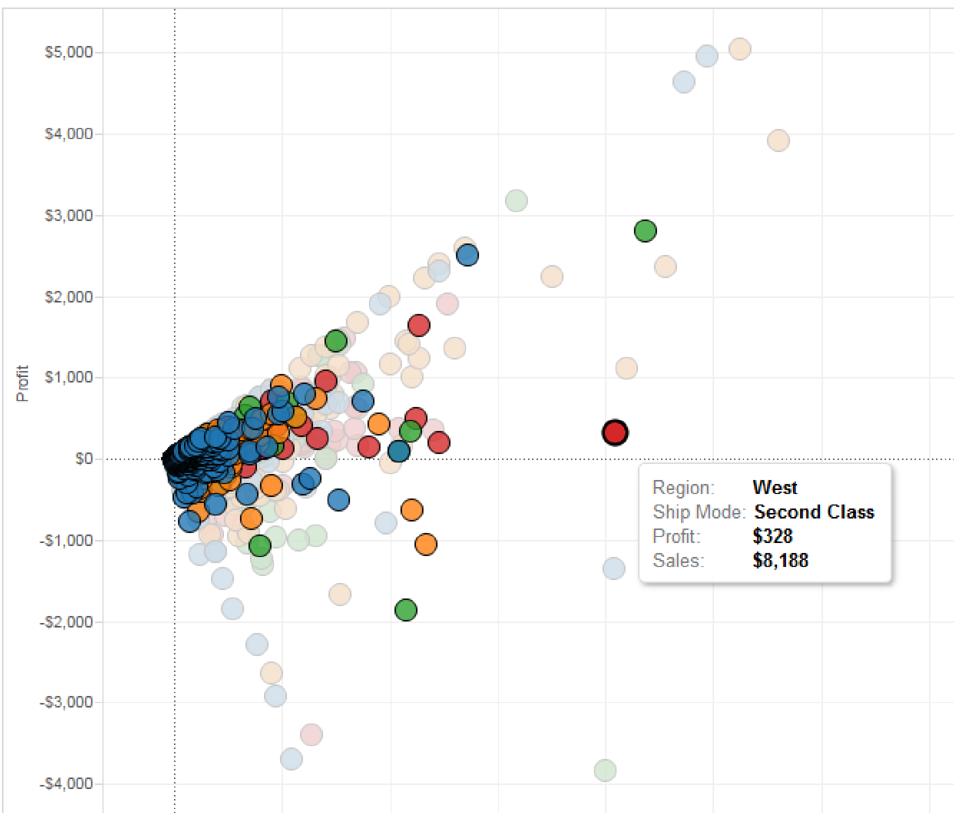
   You can also show mark labels on highlighted marks. To turn on mark labels, on the Marks card, click **Label**, select the **Show mark labels** check box and select **Highlighted** in the Marks to Label section.

Highlight Toolbar Button

Another way to add a highlight action is to use the highlight button in the toolbar. Similar to legend highlighting, the toolbar button lets you highlight a collection of related marks in the view and functions the same way that two-way highlighting does. To turn on highlighting, select the fields you want to use for highlighting on the toolbar menu. The drop-down list shows the active fields in the view. Then select a mark in the view to see the related data.



For example, the view below shows **Sales** vs. **Profit** by **Region**. If you turn on highlighting for **Ship Mode**, when a mark is selected, all other marks that were shipped via that mark's ship mode are highlighted. In this example you can quickly see all products from across the United States that were shipped via Second Class.

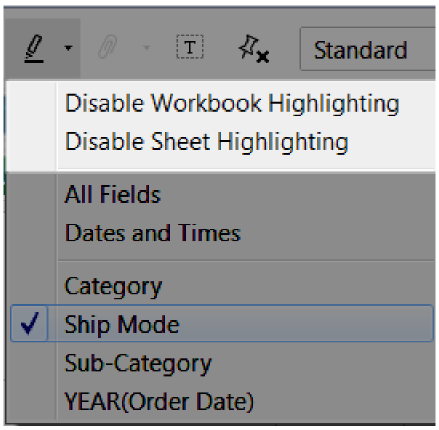


The toolbar menu also lets you highlight on **All Fields** or **Dates and Times**. **All Fields** considers all fields in the view when identifying matching records; **Dates and Times** considers all date and time fields in the view.

When you use the Highlight toolbar button in Tableau Desktop, an action is created in the Actions dialog box. You can modify the action to create more advanced highlighting behavior. For more information about editing actions, see **Create Advanced Highlight Actions** below.

Finally, you can use the toolbar button to disable highlighting across the entire workbook or for just the active sheet. These options disable highlighting for legends only. They do not turn off the ability to manually highlight marks or use the Highlighter control.

When you turn off highlighting from the toolbar button, the highlight icon is hidden on the legend, and the **Highlight Selected items** menu option is grayed out on the context menu for the legend.

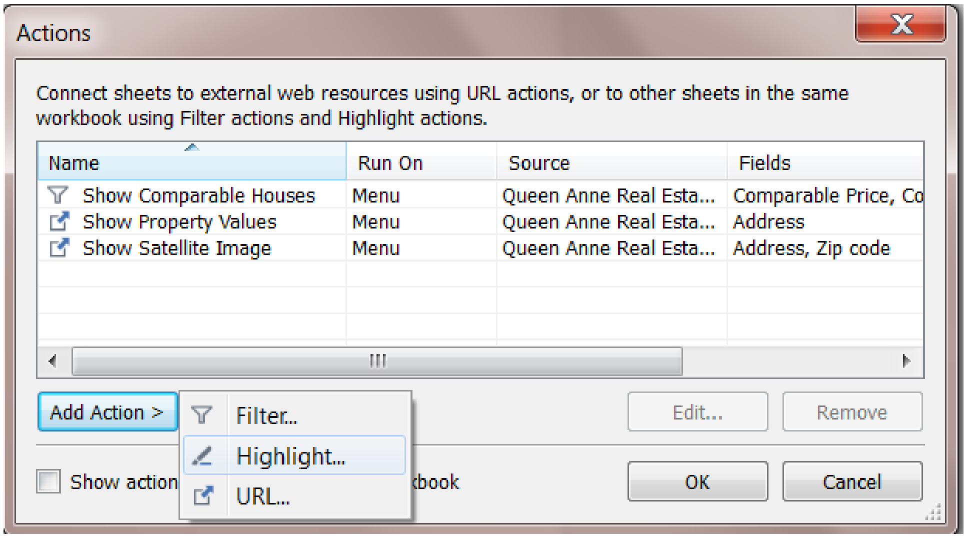


Create Advanced Highlight Actions

You can define more advanced highlight actions using the Actions dialog box. There you can specify source and target sheets and the fields you want to use for highlighting. Follow the steps below to create a highlight action.

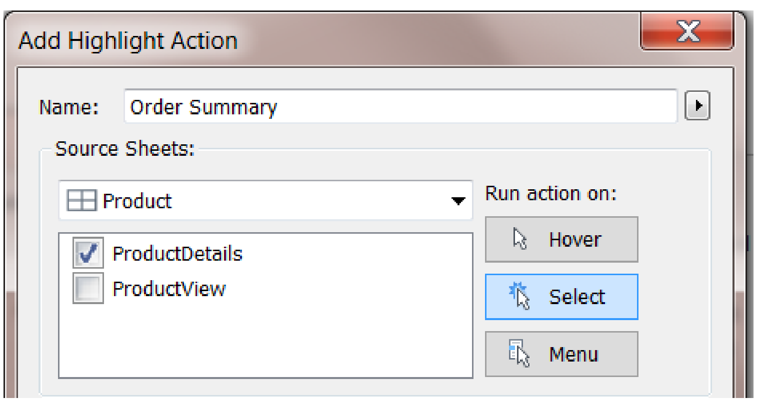
1. On a worksheet select **Worksheet > Actions**. From a dashboard, select **Dashboard > Actions**.

2. In the Actions dialog box click the **Add Action** button and then select **Highlight**.



3. Name the action to identify it in the Actions dialog box. Try to make the name descriptive, such as, *Highlight Products Shipped by Delivery Truck*. You can select variables from a drop-down list and use them in the name. Then they are filled in based on the values of the selected field.

4. Use the drop-down list to select the source sheet or data source. If you select a data source or a dashboard sheet, you can further select individual sheets within them.

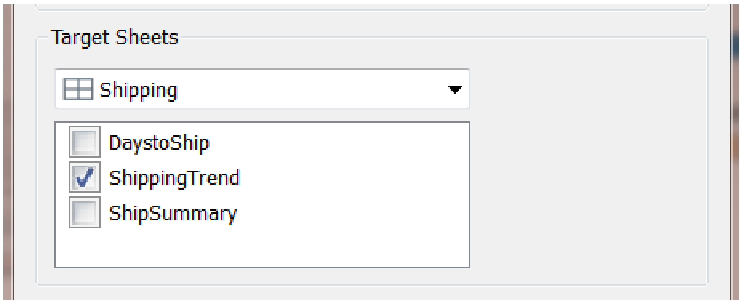


5. Select how you want to trigger the action. You can select from the following options:

   - **Hover** - Rest the pointer over a mark in the view to run the action. This option works well for highlight and filter actions within a dashboard.
   - **Select** - Click a mark in the view to run the action. This option works well for all types of actions.
   - **Menu** - Right-click (control-click on Mac) a selected mark in the view and then

select an option on the context menu. This option works well for filter and URL actions.

6. Select a target sheet. If you select a dashboard, you can further select individual sheets within the dashboard.



7. Select the fields you want to use for highlighting. Select from the following options:
   - **All Fields** - Marks in the target sheet are highlighted when they match the marks selected in the source sheet. All fields are considered when determining a match.
   - **Dates and Times** - Marks in the target sheet are highlighted when their date and time match those of the marks selected in the source sheet. The source and target worksheets can only have one date field each, however the date fields can have different names.
   - **Selected Fields** - Marks in the target sheet are highlighted based on select fields. For example, highlighting using the Ship Mode field will result in an action that highlights all marks in the target sheet that have the same ship mode as the selected mark in the source sheet.
8. When finished, click **OK** twice to close the dialog boxes and return to the view.

See Also

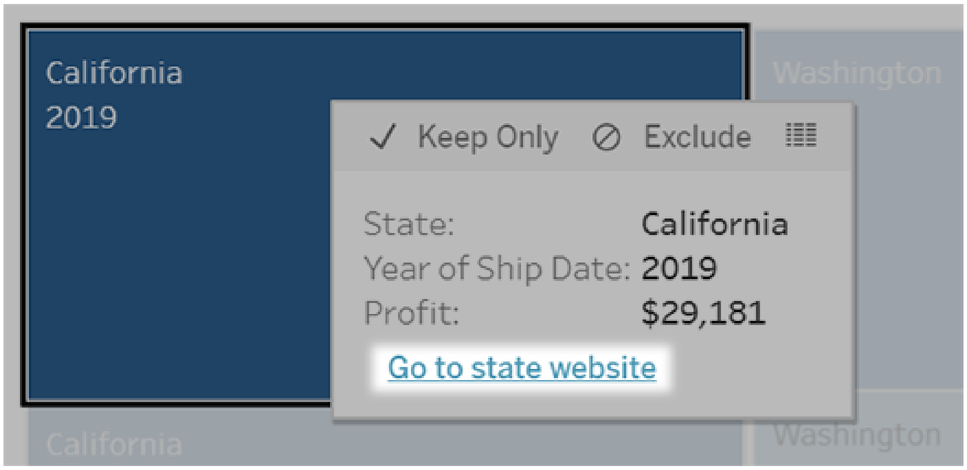
Highlight Actions on page 1498

Highlight Data Points in Context on page 1502

URL Actions

A URL action is a hyperlink that points to a web page, file, or other web-based resource outside of Tableau. You can use URL actions to create an email or link to additional information about your data. To customize links based on your data, you can automatically enter field values as parameters in URLs.

Tip: URL actions can also point to a web page object in a dashboard. See Actions and Dashboards to learn more.



A URL action run from a tooltip menu. The link reflects the action name, not the target URL.

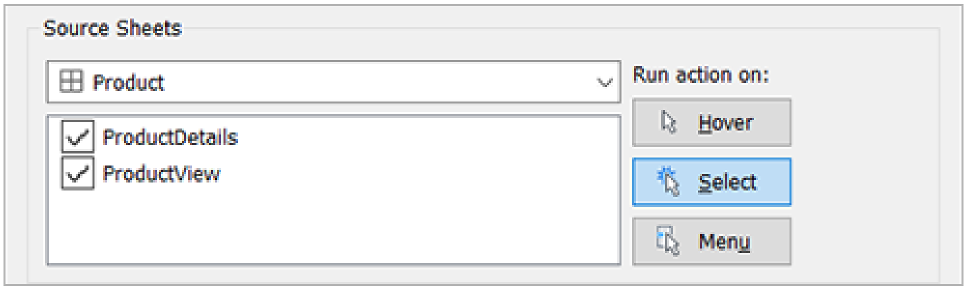
Open a web page with a URL action

Note: Steps on this page refer to the interface of Tableau Online and Tableau Server. The sequence and naming of options differs slightly in Tableau Desktop.

1. On a worksheet, select **Worksheet** > **Actions**. From a dashboard, select **Dashboard** > **Actions**.
2. In the Actions dialog box, click **Add Action** and then select **Go to URL**.
3. In the next dialog box, enter a name for the action. To enter field variables in the name, click the **Insert** menu to the right of the **Name** box.

   Note: Give the action a descriptive name, because in tooltip menus the link reflects that name, not the URL. For example, when linking to more product details, a good name could be “Show More Details”.

4. Use the drop-down list to select a source sheet or data source. If you select a data source or dashboard you can select individual sheets within it.



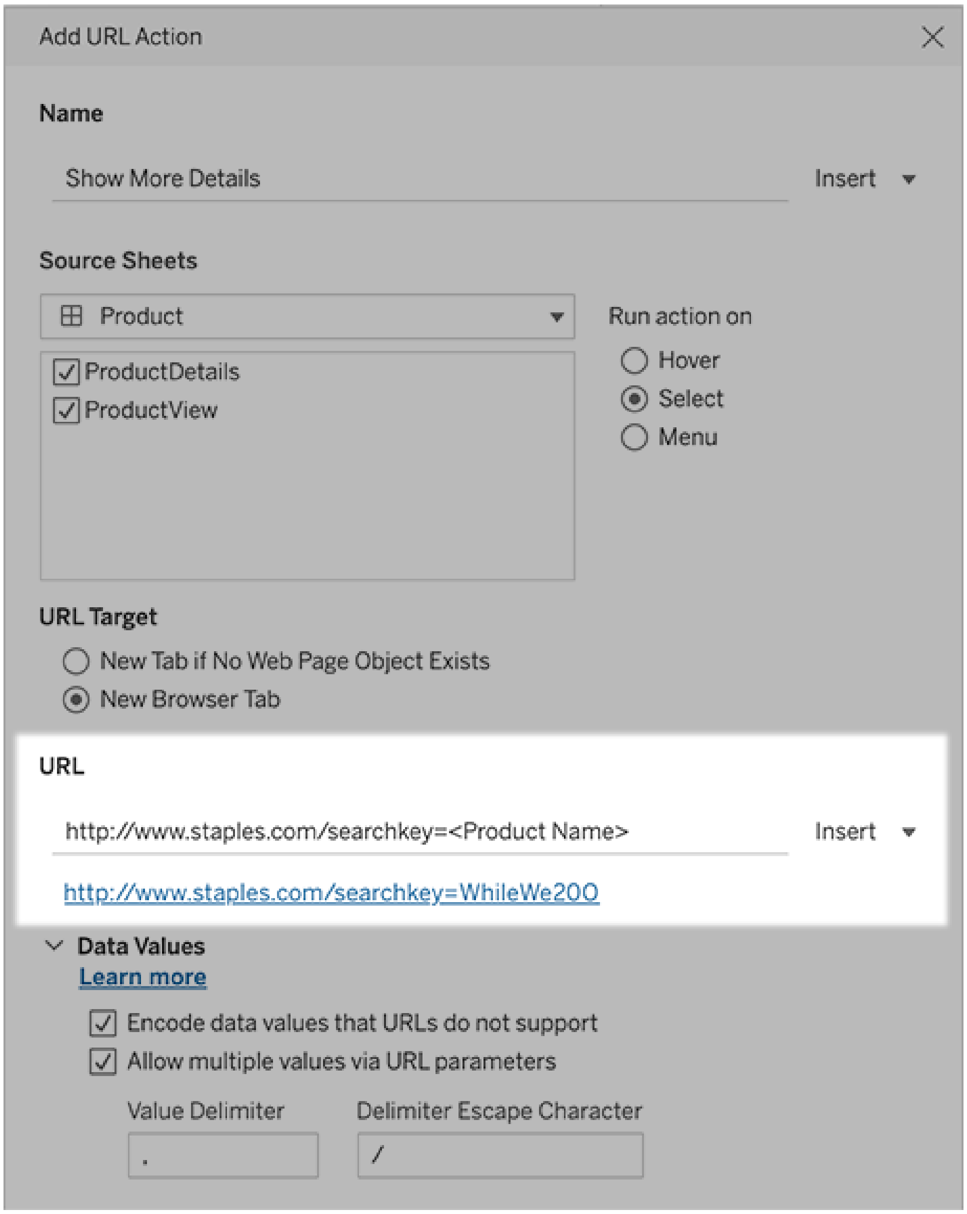
5. Select how users will run the action.

If you choose this option...	The action is run when the user...
Hover	Mouses over a mark in the view. This option works best for highlight and filter actions within a dashboard.
Select	Clicks a mark in the view. This option works well for all types of actions.
Menu	Right-clicks (control-clicks on Mac) a selected mark in the view, then clicks an option in a tooltip menu. This option works particularly well for URL actions.

6. For URL Target, specify where the link will open:
   - **New Tab if No Web Page Object Exists** — Ensures that the URL opens in a browser on sheets that lack web page objects. This is a good choice when Source Sheets is set to All or a data source.
   - **New Browser Tab** — Opens in the default browser.
   - **Web Page Object** — (Available only for dashboards with Web Page objects) Opens in the web page object you select.
7. Enter a URL with one of the following prefixes:
   - `http`, `https`, `ftp`, `mailto`, `news`, `gopher`, `tsc`, `tsl`, `sms`, `tel`, `file`
   - Tableau Desktop also supports local paths like `C:\Example folder\example.txt`

   To enter field and filter values as parameters in the URL, click the **Insert** menu to the right of the URL. Be aware that any referenced fields must be present in the view, not just a related data source. For details, see **Using field and filter values in URLs** on page 1515.

Note: On a dashboard, you can specify an ftp address only if the dashboard doesn't contain a web object. If a web object exists, the ftp address won't load.



Below the URL you enter is a hyperlinked example you can click for testing.

8. (Optional) In the Data Values section, select any of the following options:

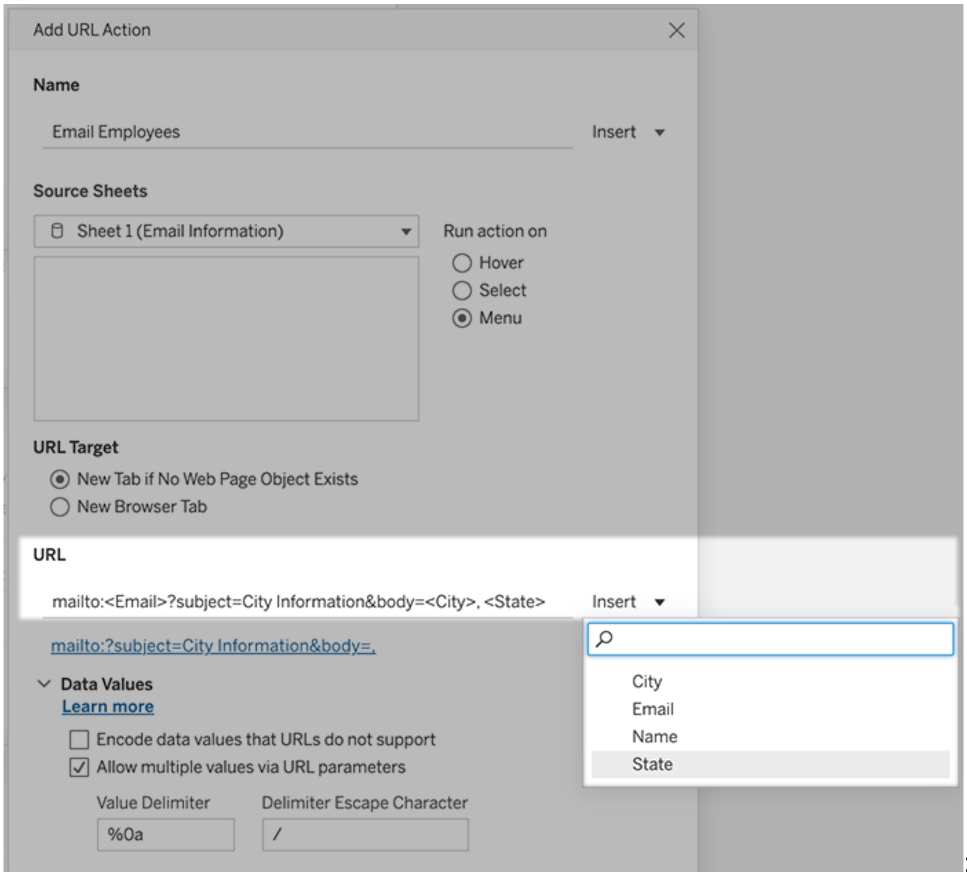
- **Encode Data Values that URLs Do Not Support** — Select this option if your data contains values with characters that browsers don't allow in URLs. For example, if one of your data values contains an ampersand, such as “Sales & Finance,” the ampersand must be translated into characters that your browser understands.
- **Allow Multiple Values via URL Parameters** — Select this option if you are linking to a web page that can receive lists of values via parameters in the URL. For example, say you select several products in a view and you want to see each product’s details hosted on a webpage. If the server can load multiple product details based on a list of identifiers (product ID or product name), you could use multi-select to send the list of identifiers as parameters.

  When you allow multiple values, you must also define the delimiter escape character, which is the character that separates each item in the list (for example, a comma). You must also define the Delimiter Escape, which is used if the delimiter character is used in a data value.

Create an email with a URL action

1. On a worksheet, select **Worksheet** > **Actions**. From a dashboard, select **Dashboard** > **Actions**.
2. In the Actions dialog box, click **Add Action**, and select **Go to URL**.
3. In the Source Sheets drop-down list, select the sheet that contains the field with the email addresses you want to send to.
4. In the URL box, do the following:
   - Type **mailto:**, and click the **Insert** menu at right to select the data field that contains email addresses.
   - Type **?subject=**, and enter text for the Subject line.
   - Type **&body=**, and click the **Insert** menu at right to select the fields of information that you want to include in the body of the email.

   In the example below, the “Email” field contains the email addresses, the subject is “City Information”, and the body text of the email consists of the city and state data that is associated with the email address.



:

5. (Optional) Display data from your workbook in the body of your email as a vertical list instead of the default horizontal list. For example, suppose you have a horizontal list of cities, such as Chicago, Paris, Barcelona, which you would rather display vertically, like this:

   Chicago
   Paris
   Barcelona

   To make the list vertical, in the Data Values section, do the following:

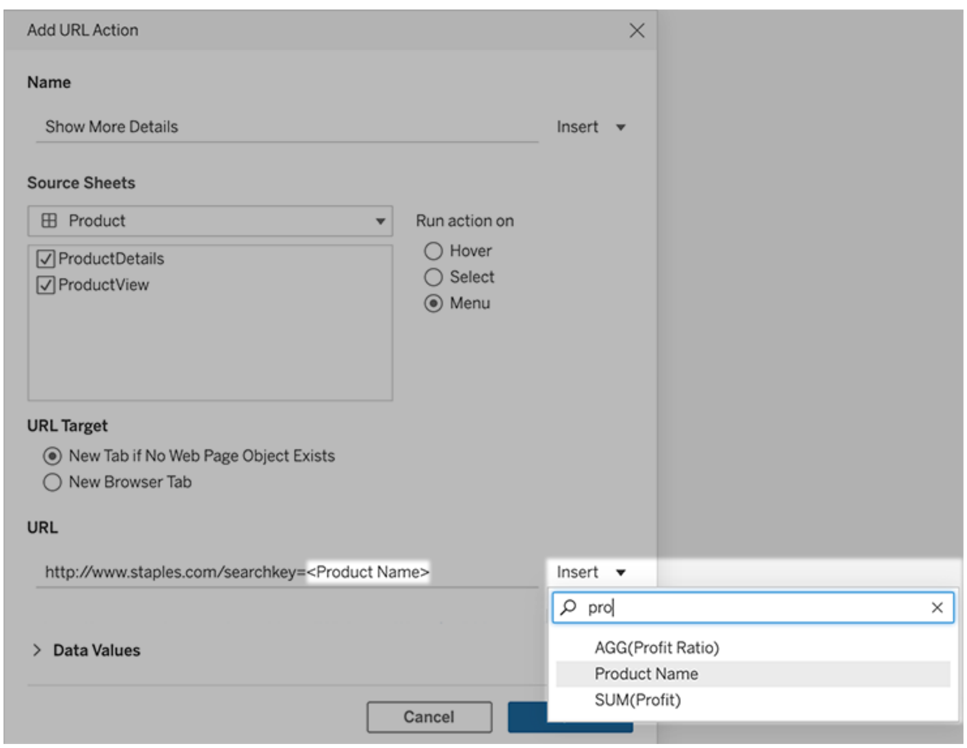
   - Deselect **Encode Data Values that URLs Do Not Support**
   - Select **Allow Multiple Values via URL Parameters**.
   - Type **%0a** in the **Value Delimiter** text box to add line breaks between each item in the list. (These are the URL-encoded characters for a line break.)

Using field and filter values in URLs

When users trigger URL actions from selected marks, Tableau can send field, filter, and parameter values as variables in the URL. For example, if a URL action links to a mapping website, you could insert the address field to automatically open the currently selected address on the website.

1. In the Edit URL Action dialog box, begin typing the URL for the link.
2. Place the cursor where you want to insert a field, parameter, or filter value.
3. Click the **Insert** menu to the right of the text box and select the field, parameter, or filter you want to insert. The variable appears within angle brackets. You can continue adding as many variables as you need.

   Note: Any referenced fields must be present in the view, not just a related data source. Otherwise, the link won't display in the viz, even if it functions when you click Test Link.



Including aggregated fields

The list of available fields includes only non-aggregated fields. To use aggregated field values as link parameters, first create a related calculated field, and add that field to the view. (If you don't need the calculated field in the visualization, drag it to Detail on the Marks card.)

Inserting parameter values

When inserting parameter values, URL actions send the Display As value by default. To instead send the actual value, add the characters `~na` after the parameter name.

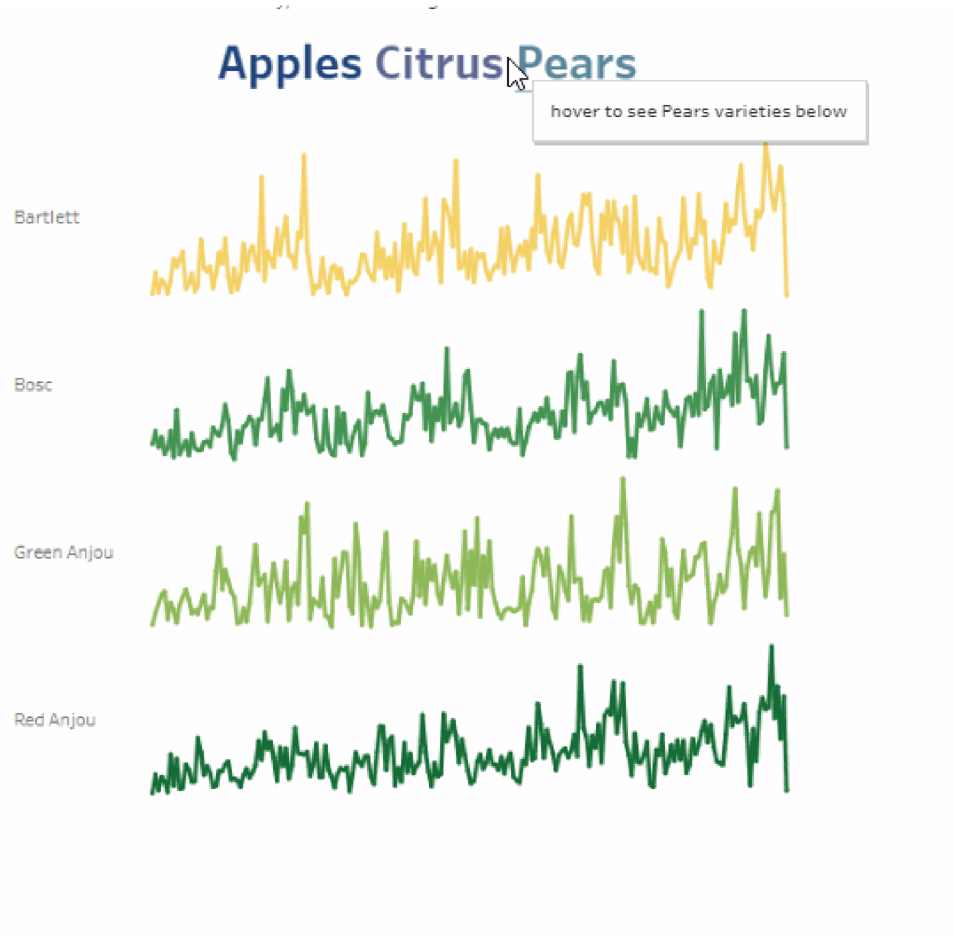
For example, say you have a parameter that includes IP addresses, with Actual Value strings such as 10.1.1.195 and Display As strings with more friendly values such as `Computer A (10.1.1.195)`. To send the actual value, you'd revise the parameter in the URL to look like this: `http://<IPAddress~na>/page.htm`.

Parameter Actions

Use parameter actions to let your audience change a parameter value through direct interaction with a viz, such as clicking or selecting a mark. You can use parameter actions with reference lines, calculations, filters, and SQL queries, and to customize how you display data in your visualizations.

Parameter actions open up new possibilities for creating summary values and statistics without using calculations. You can configure parameter actions to let users select multiple marks that are automatically aggregated into a single parameter value. For example, a parameter action could show the AVG(Sales) or COUNTD(Orders) for currently selected marks.

For examples of how parameter actions can be used, see **Examples of parameter actions on page 1521**.



You can create parameter actions in Tableau Desktop, Tableau Online, and Tableau Server.

When you define a parameter action, it must include:

- the source sheet or sheets it applies to.
- the user behavior that runs the action (hover, select, or menu).
- the target parameter, the source field, and the respective aggregation (if applicable).

The parameter referenced in the action must also be referenced in some part of the visualization. You can do this in different ways, such as referencing the parameter in a calculated field or reference line that is used in the viz. You can also build a view that uses a source field that is referred to in the parameter action.

Watch a video: To see related concepts demonstrated in Tableau, watch Parameter Actions (7 minutes), a free training video. Use your tableau.com account to sign in.

General steps for creating parameter actions

1. Create one or more parameters. For more information, see **Create Parameters** on page 1218.
2. **Optional**: Depending on the behavior you want to make available to users for their analysis, you might want to create a calculated field that uses the parameter. For details on creating calculated fields, see **Create a calculated field** on page 2097.
3. Build a visualization that will make use of parameter actions.

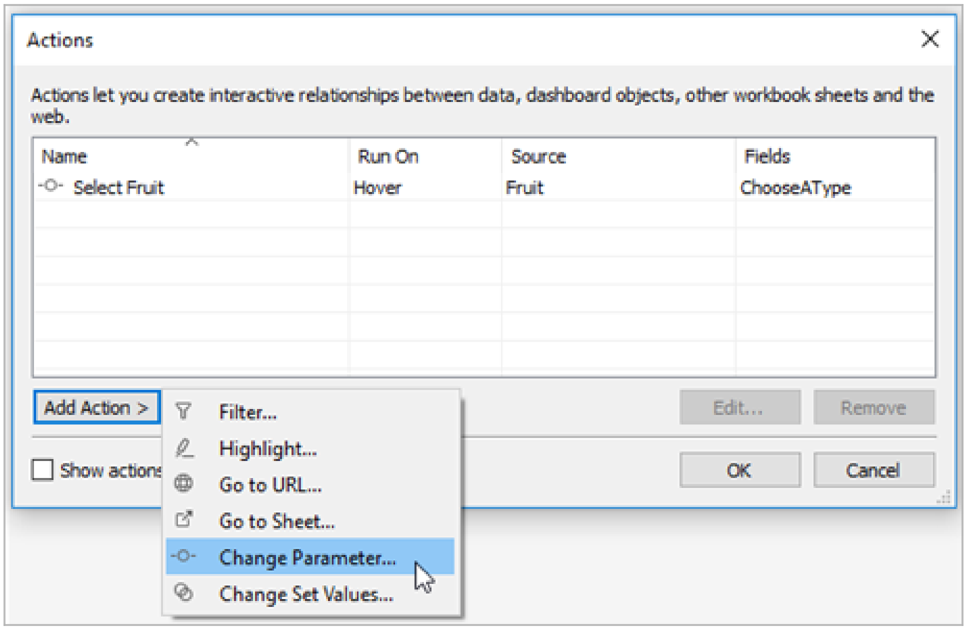
   For example, if you created a calculated field that uses the parameter, build the view using that calculated field. Or, drag a field that you plan to tie to the parameter into the view.
4. Create a parameter action that uses a parameter. You can create multiple parameter actions for different purposes. The parameter action must refer to the sheet name of the visualization, the parameter name, and the source field that the parameter will be associated with.
5. Test the parameter action and adjust its settings or other related elements in your visualization as needed to get the behavior you want your audience to experience.

Note: It is helpful to understand how you can use parameters in Tableau before you start using them in parameter actions. For more information:

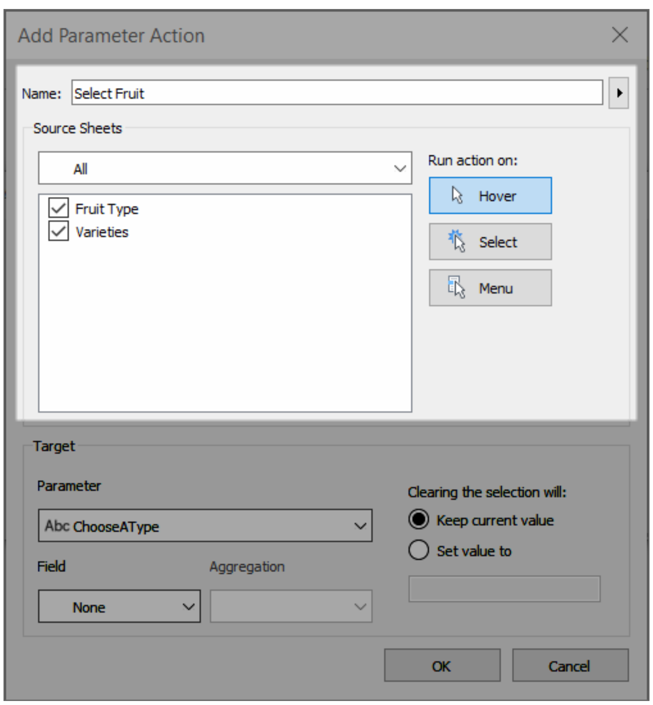
- Read the **Create Parameters** on page 1218 help topic and related parameter topics.
- See Use Parameters to Add Multiple Views to Your Viz on the Tableau blog and multiple practical topics on how to use parameters on the Data School website, such as Four Steps to Creating a Parameter.
- Watch the free training video Parameters (5 minutes). Use your tableau.com account to sign in. For a more in-depth presentation, see Parameters | Oh, the places you'll go! (45 minutes).

Create a parameter action

1. In a worksheet, select **Worksheet >Actions**. In a dashboard, select **Dashboard >Actions**.
2. In the Actions dialog box, click **Add Action** and then select **Change Parameter**.



3. In the **Actions** dialog box, specify a meaningful name for the action.
4. Select a source sheet or data source. The current sheet is selected by default. If you select a data source or dashboard you can select individual sheets within it.



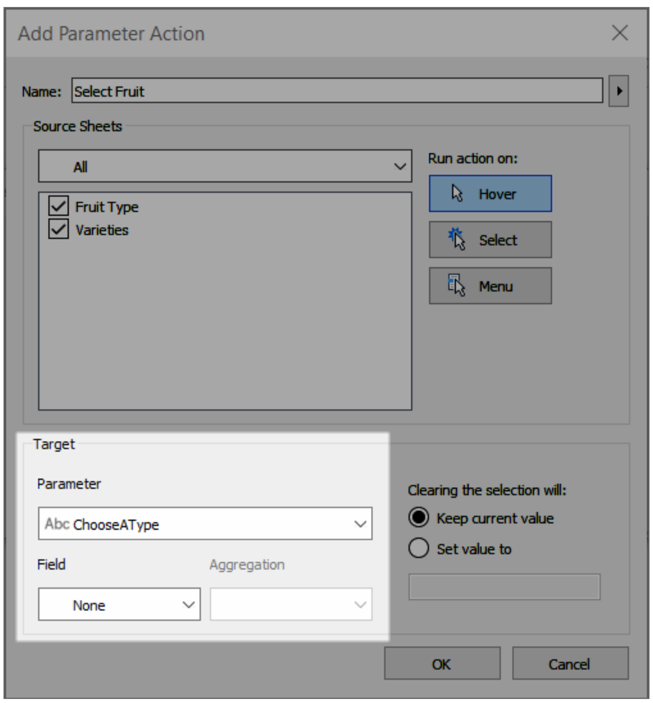
5. Select how users will run the action.

- **Hover** - Runs when a user hovers the mouse cursor over a mark in the view. Also exposes an option for the action in the mark's tooltip.
- **Select** - Runs when a user clicks a mark or selects multiple marks in the view.
- **Menu** - Runs when a user right-clicks (macOS: control-clicks) a selected mark in the view, then selects an option on the context menu. Also exposes an option for the action in the mark's tooltip.

6. Under **Target**, specify the target parameter and a source field. All parameters in the workbook are available in the **Parameter** list.

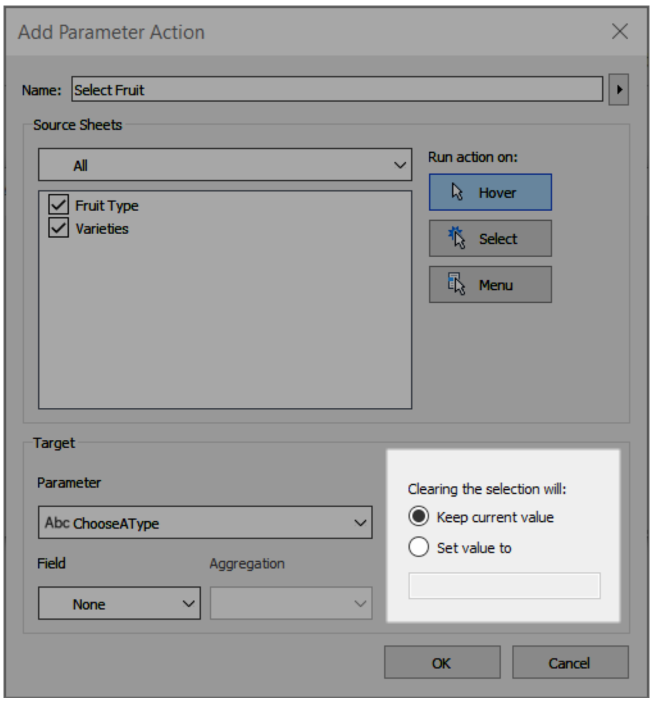
   To let your users select multiple marks that get aggregated into a single parameter value, select an aggregation type.

   > **Note**: If you want your users to be able to select multiple marks, you will need to select an aggregation. If **Aggregation** is set to **None**, and multiple marks that contain different values for the source field are selected, the parameter action will not run.



The aggregation that you choose for the parameter action can be different than the aggregation used for that field in the view. For example, you could create a view that uses SUM(Sales), but set the Sales source field referenced in the parameter action to use Average for the aggregation. Note that if you then change the aggregation of the Sales field in the view to AVG(Sales), you will need to update the parameter action to reference the Sales field again.

7. Specify what will happen when the selection is cleared.
   - **Keep current value** - Current parameter value will remain in the view.
   - **Set value to** - Parameter value will revert to the indicated value.



8. Click **OK** to save your changes and return to the view.
9. Test the parameter action by interacting with the visualization. Tweak some of the settings for the action to adjust the behavior as needed.

Examples of parameter actions

The following examples demonstrate how you can use parameter actions, but these are just the beginning of what is possible. To follow along, it helps to be familiar with creating and using parameters and calculated fields. For more information, see **Create Parameters** on page 1218 and **Create a calculated field** on page 2097.

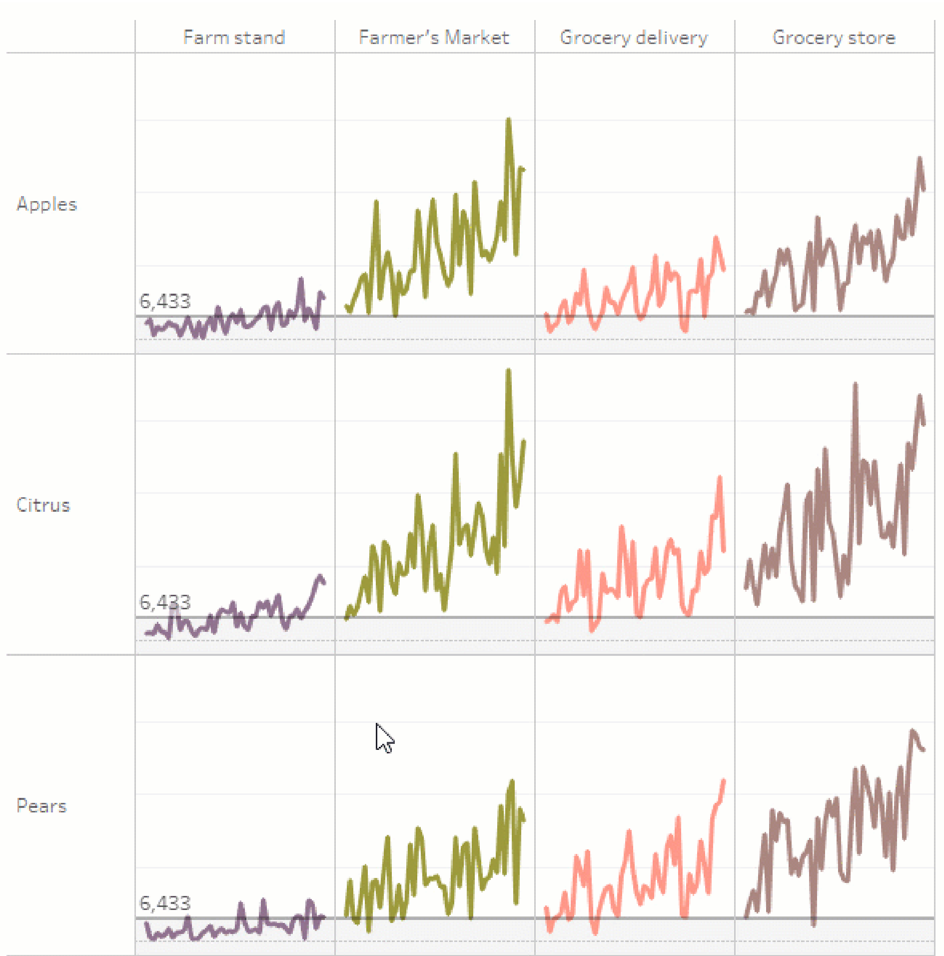
For additional examples of how you can use parameter actions, see:

- How and when to use parameter actions
- Parameter Actions: KPI Navigator
- Parameter Actions: Year over Year Comparison
- Parameter Actions: Layered Analysis
- Parameter Actions: Concatenation
- 3 Creative Ways To Use Tableau Parameter Actions from Tableau Zen Master Ryan Sleeper on the PlayfairData website.
- Tableau Parameter Actions Sneak Peek from Tableau Zen Master Joshua Milligan on the VizPainter website.

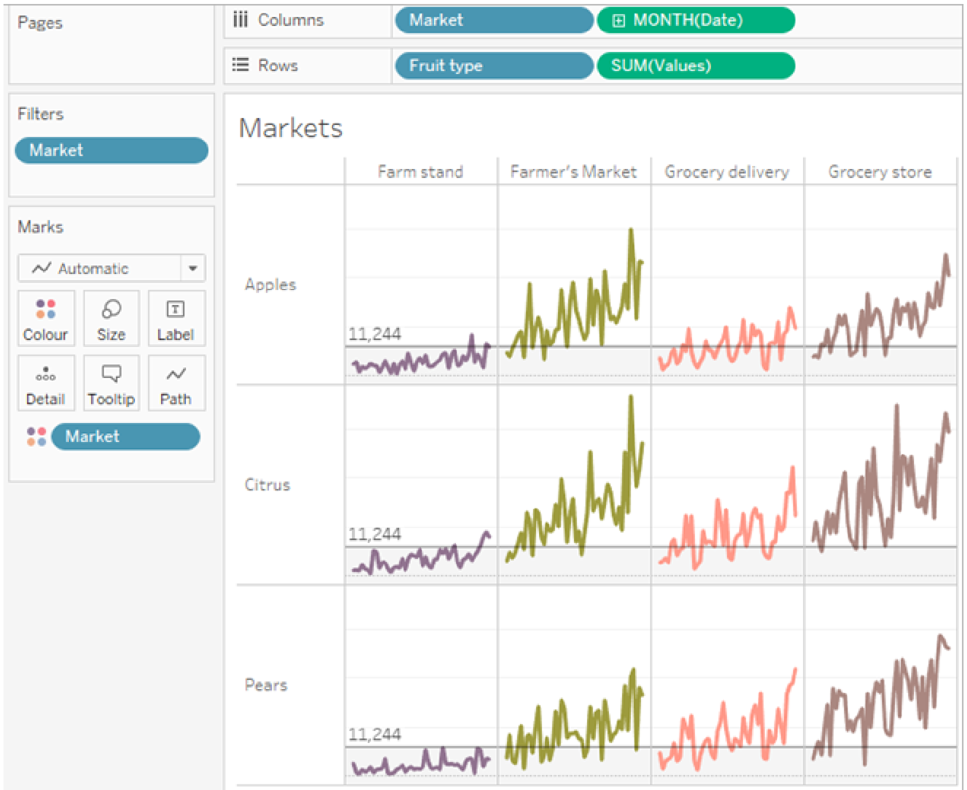
Make reference lines dynamic

This example demonstrates how you could use a parameter action to interactively update a reference line, without using a calculation.

Reference lines provide a convenient way to compare values in a visualization. You can then use parameter actions to make reference lines dynamic. You can associate the reference line with a parameter, and then use a parameter action to make the parameter interactive. As your users interact with the view, the reference lines automatically update to provide more context to the data.

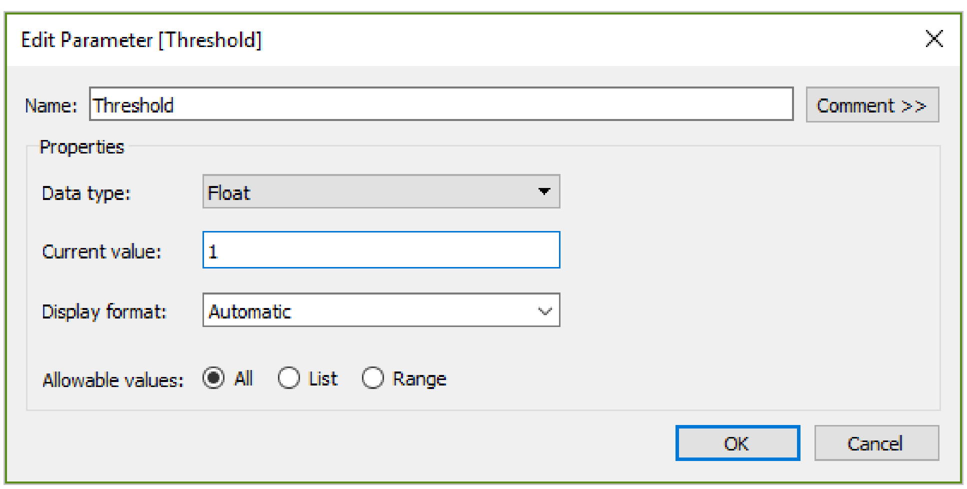


The following fields are used in this view.

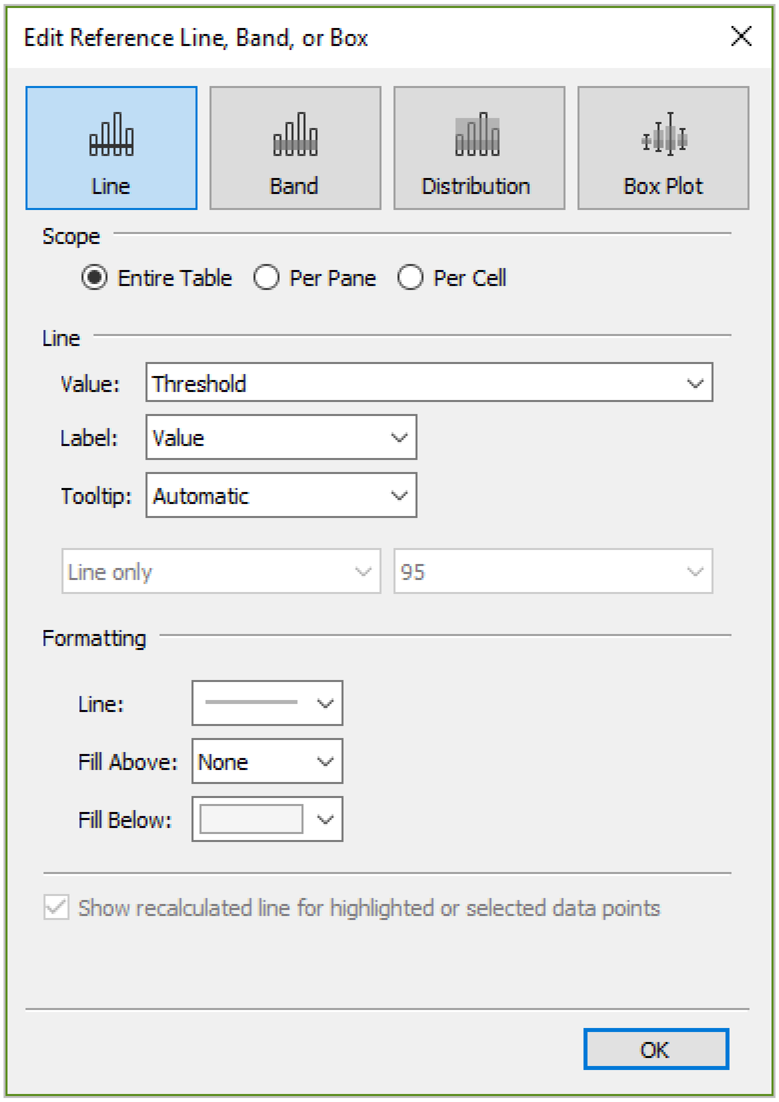


Dynamic reference lines example details

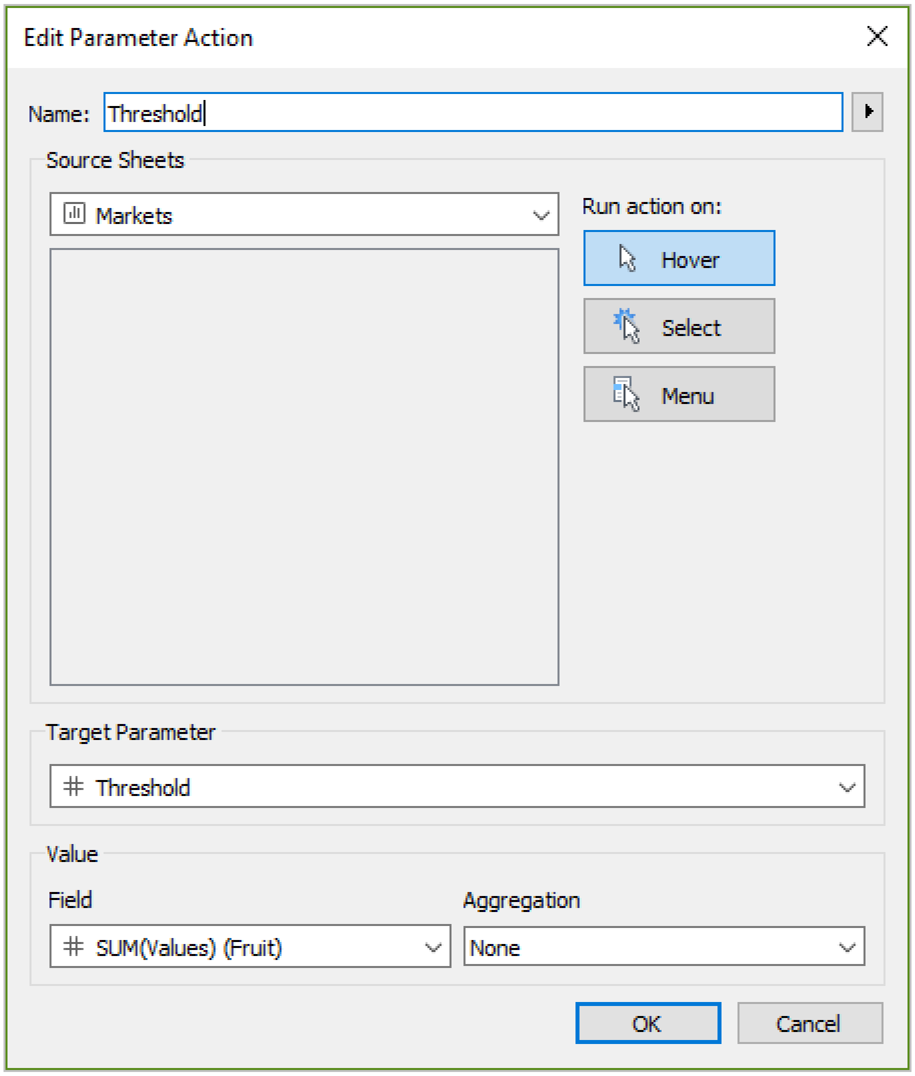
1. Create the parameter. The parameter is named *Threshold*. The data type is set to Float.



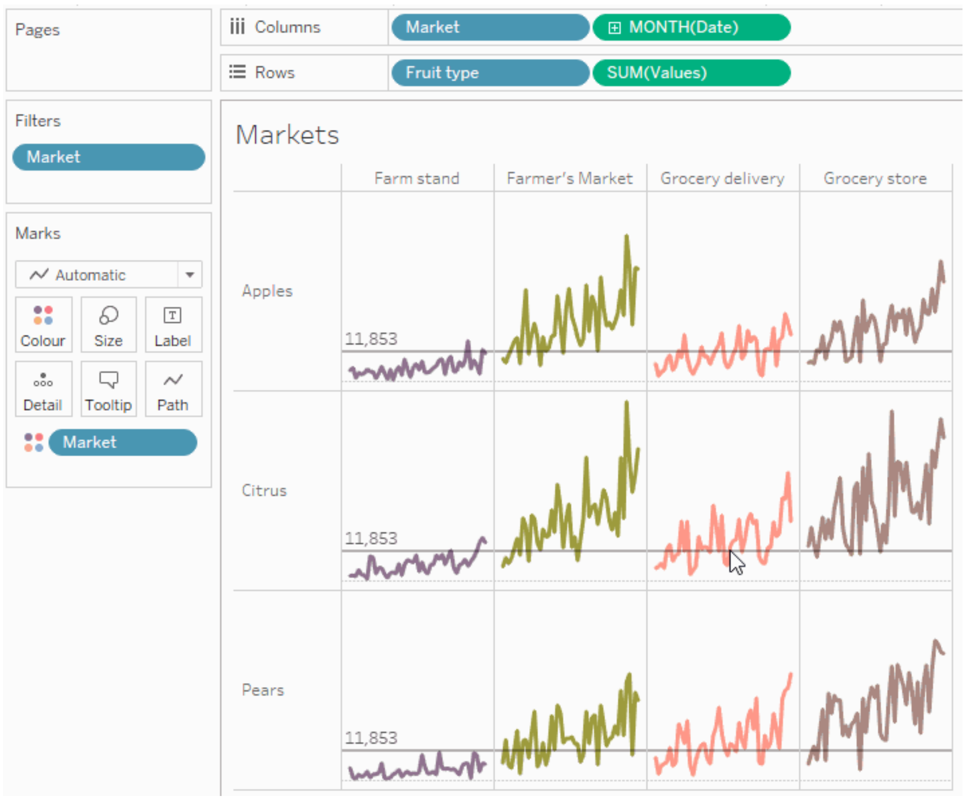
2. Create a reference line that uses the Threshold parameter. The label is set to Value. The Fill Below option is set to a light gray color.



3. Create a parameter action. This action references the Threshold parameter and the Values field. The action runs on Hover.

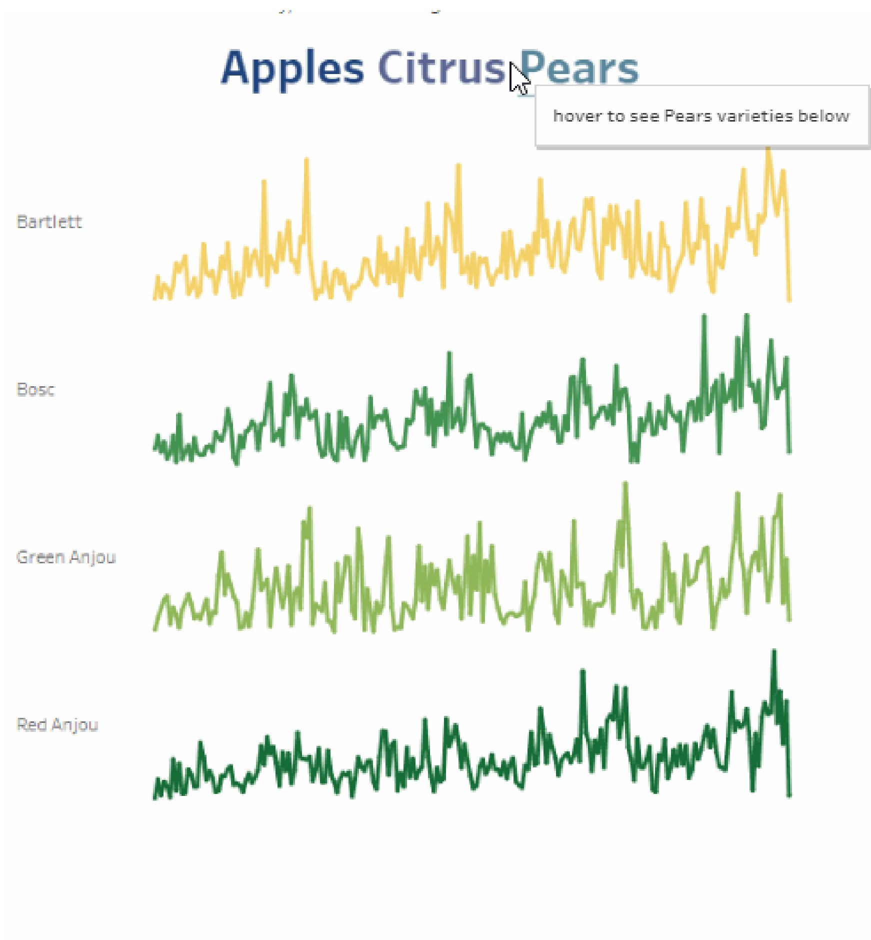


4. Test the parameter action in the sheet. Hovering over a mark in the view updates the reference line to see how that value appears in context with the other categories of data.

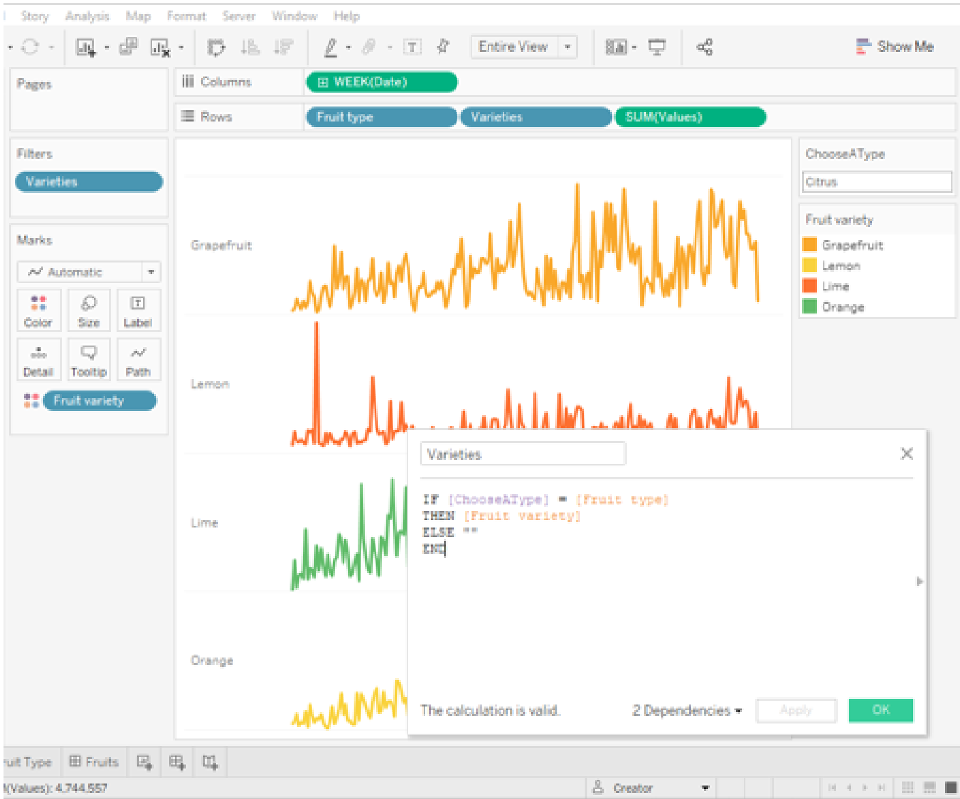


Display a selective hierarchy

This example demonstrates how you could use a parameter action that is referenced by a calculated field to let your audience control what information is being displayed in a view.



One common use for parameters is using a parameter to change the input value in a calculation. This example uses a parameter *ChooseAType* for the type of fruit and a calculated field *Varieties* that returns the varieties for the type of fruit. The calculated field has been added to the view instead of the original field *Fruit variety*. Users can select a fruit type using a parameter control to change the fruit type and varieties being displayed.



Using a parameter with a calculation and a parameter control to interactively change a parameter value

This is one way to let your users explore the data in the view. Parameter actions make this same scenario possible, but with added functionality and benefits. When you create a parameter action that updates the value of the parameter based on a user interacting with the marks in the view:

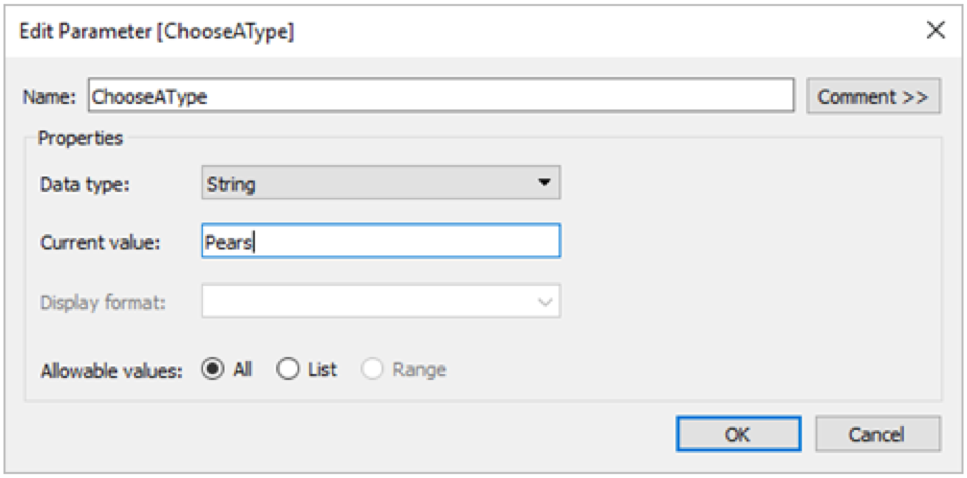
- User interaction with the viz is more natural. Your users can stay in the flow of exploring the data. Rather than needing to interact with a parameter control, your users can hover or click in the viz to update the parameter value.
- If the data source is updated at some time in the future, and more fruit type and fruit variety values are added, those values will automatically be displayed in the viz and are automatically available to the parameter.

To incorporate parameter actions, this example makes use of a second sheet (*Fruit Type*) that consists of the names of the fruit types. The two sheets (*Fruit Type* and *Varieties*) are combined into a dashboard (*Fruits*).

Selective hierarchy example details

1. The workbook shown in the image above uses the *ChooseAType* parameter and the *Varieties* calculated field.

   The *ChooseAType* parameter is set to allow *All Values*.

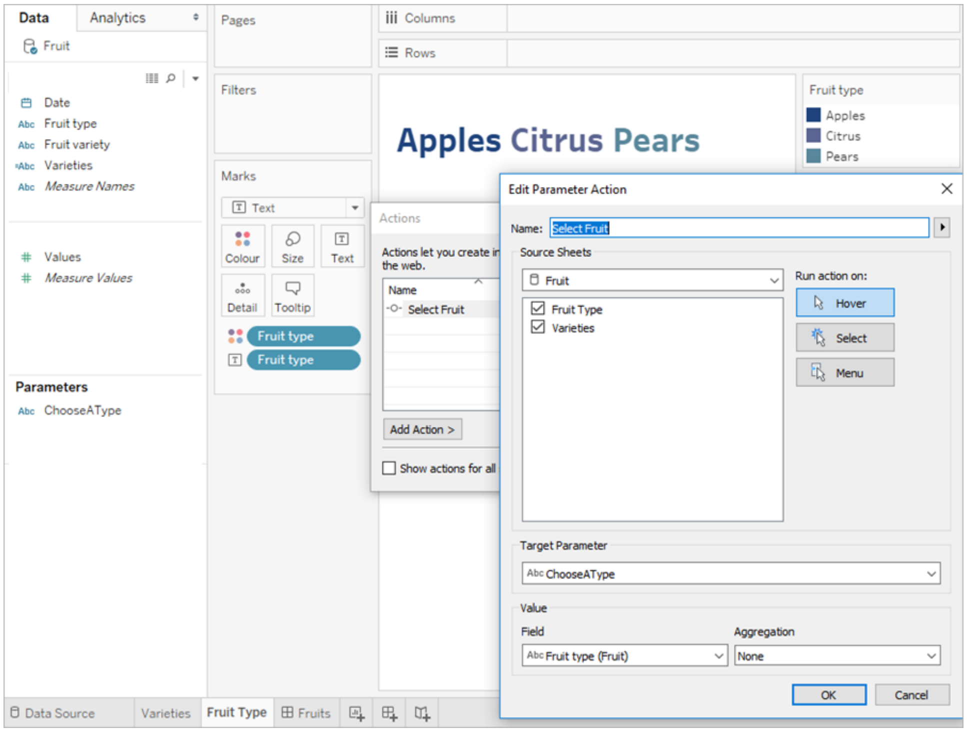
   

   The *Varieties* calculated field references the *ChooseAType* parameter to show the values from the *Fruit variety* field based on the current value selected for *Fruit type*.

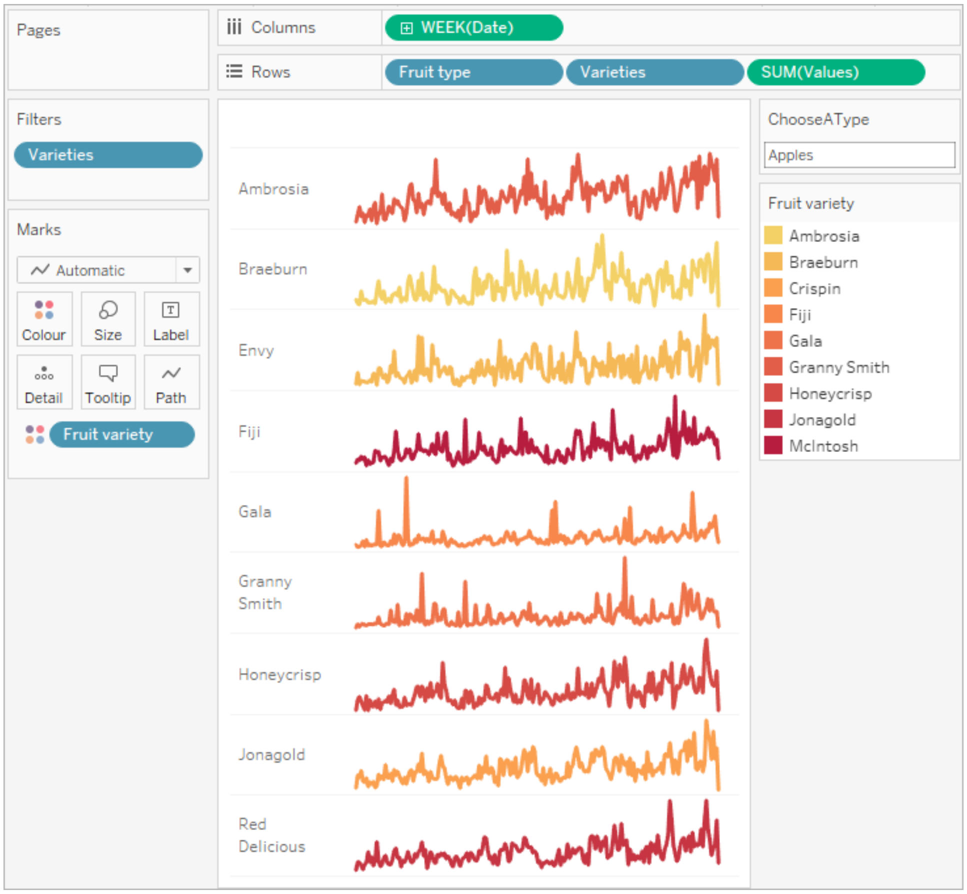
2. The workbook uses a parameter action named *Select Fruit*. To create the parameter action, the author clicked a sheet, and then selected **Worksheet** > **Actions** > **Add Action** > **Change Parameter**.

The action uses the Fruit data source. Because the parameter and calculation both need to be aware of the *Fruit type* field to function, both the *Fruit Type* and *Varieties* sheets need to be selected.

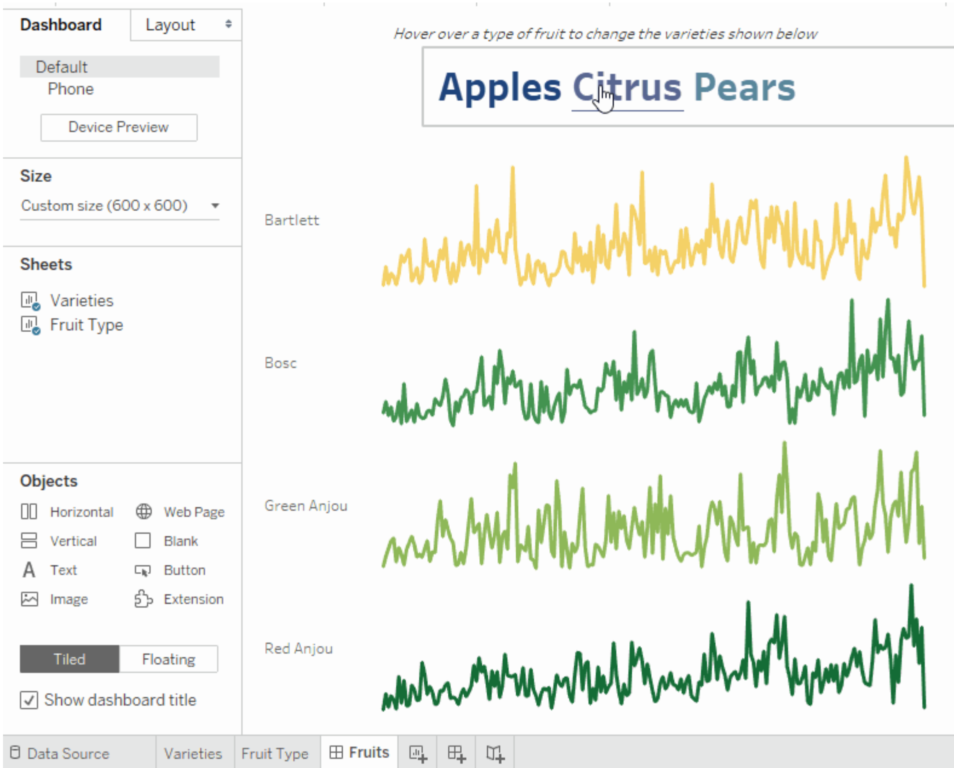


The parameter *ChooseAType* is associated with the *Fruit type* field. The parameter action will run whenever a user hovers over the fruit type text in the *Fruits* dashboard, or in the *Fruit Type* sheet.

In the *Varieties* sheet, the *Fruit type* field is on the Rows shelf, but the Show Header option is unchecked for that field to remove the fruit type label from this view.



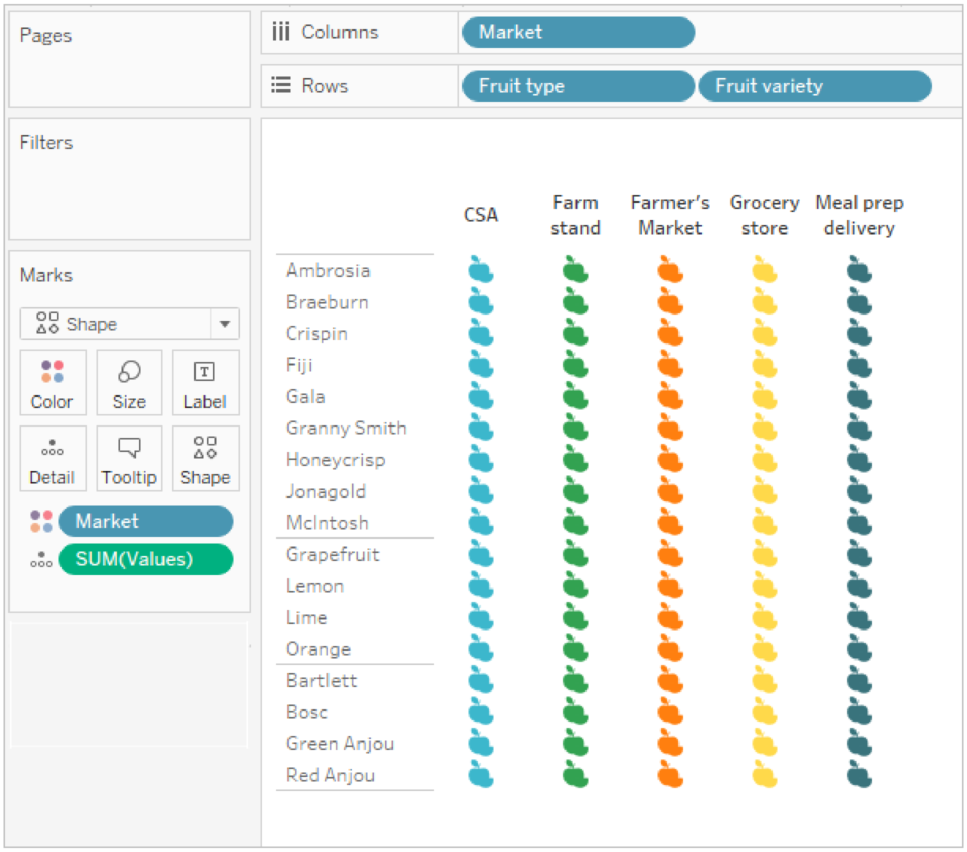
3. Test the parameter action in the Fruits dashboard. Hovering over the fruit type updates the view to show the varieties of that fruit.



Display summary data for selected marks

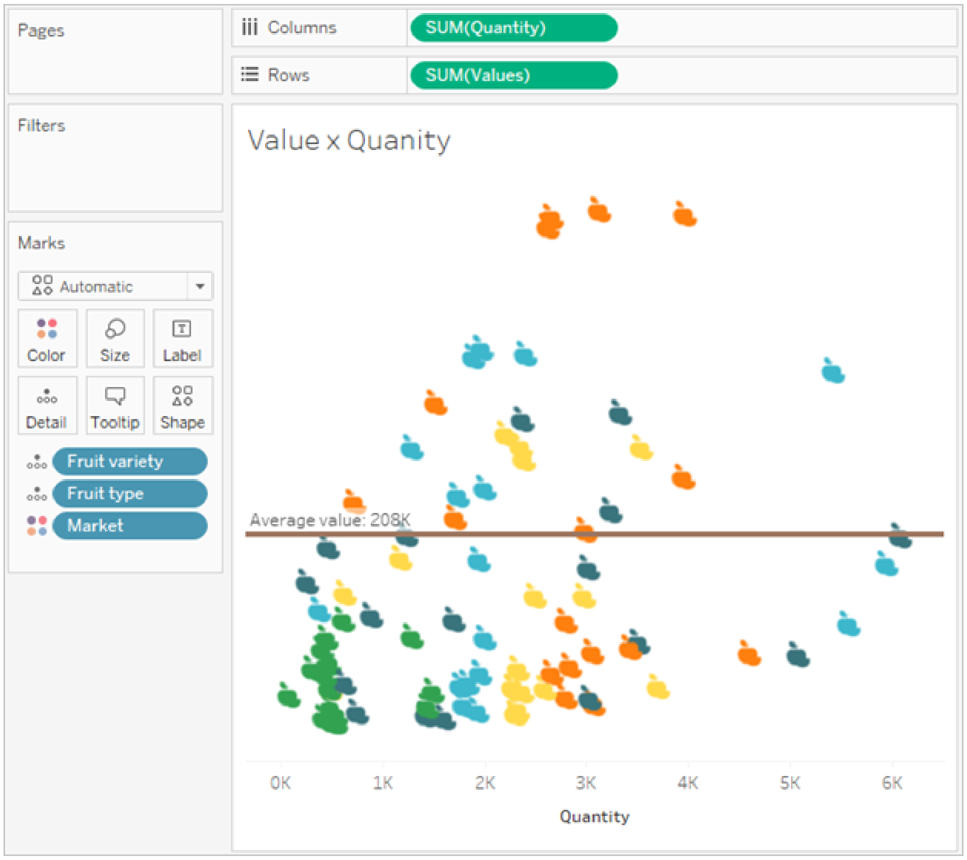
Getting summary statistics for a subset of data is a valuable analytical tool. This example shows how you can use aggregated parameter actions for selected marks to automatically generate summary data.

This view of a sheet named Options shows fruit varietals and their intended markets.

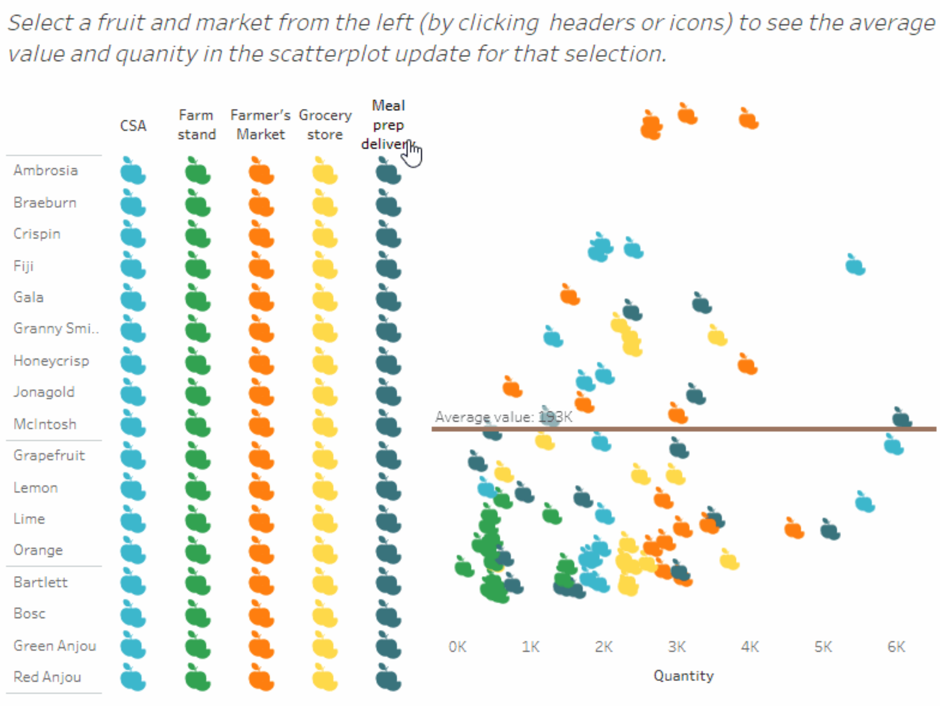


The SUM(Values) field needs to be present on both sheets for the parameter to work. SUM (Values) is placed on **Detail** in the Marks card in this view.

The previous Options view will function as a selector for this second view, a scatter plot that plots the same marks against their total value and quantity. SUM(Values) is placed on Rows in this view.

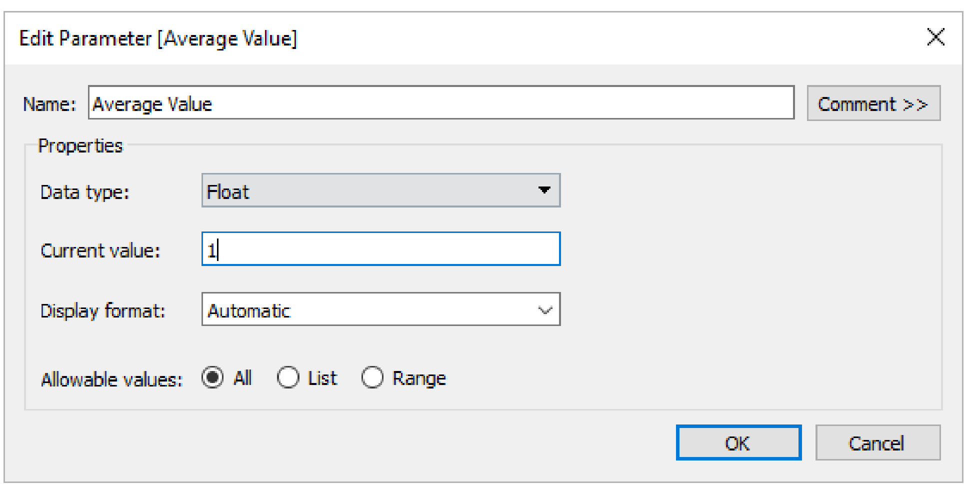


A parameter action will be used to make the two views work together in a dashboard.

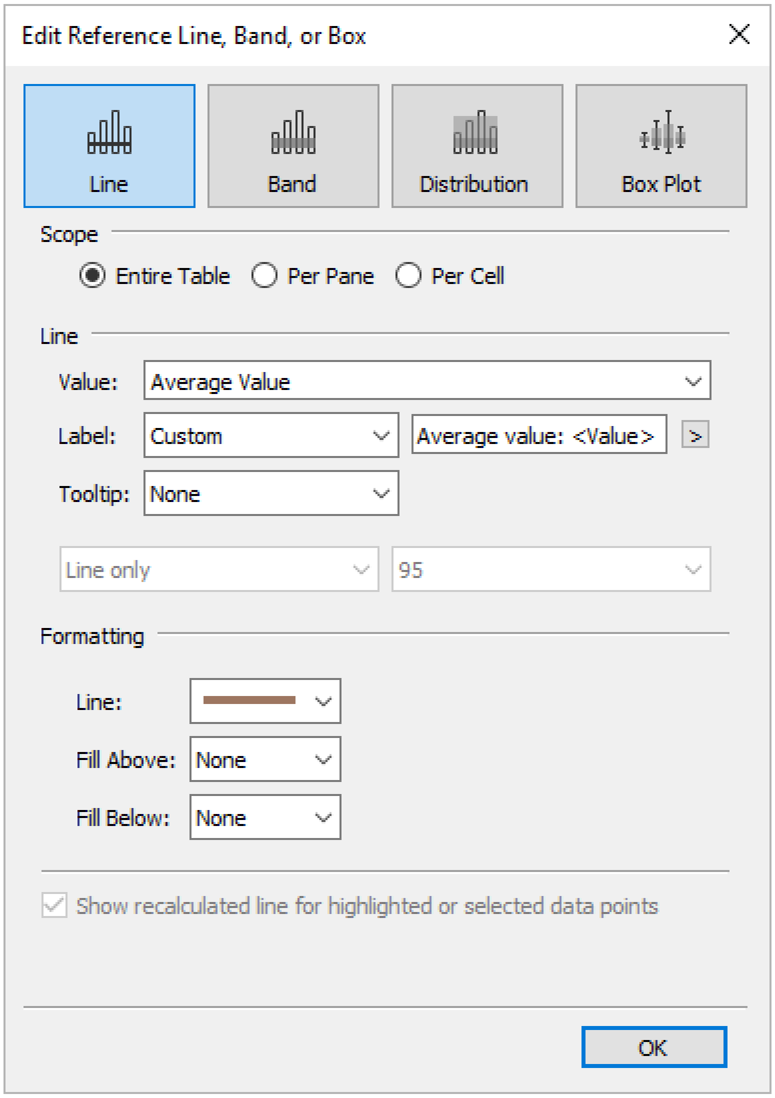


Summary data example details

1. Create the parameter. This parameter is named *Average Value*. The data type is set to Float.



2. Create a reference line that uses the Average Value parameter. The label is set to Custom with the text "Average value: <Value>". The Line option is set to a brown color.

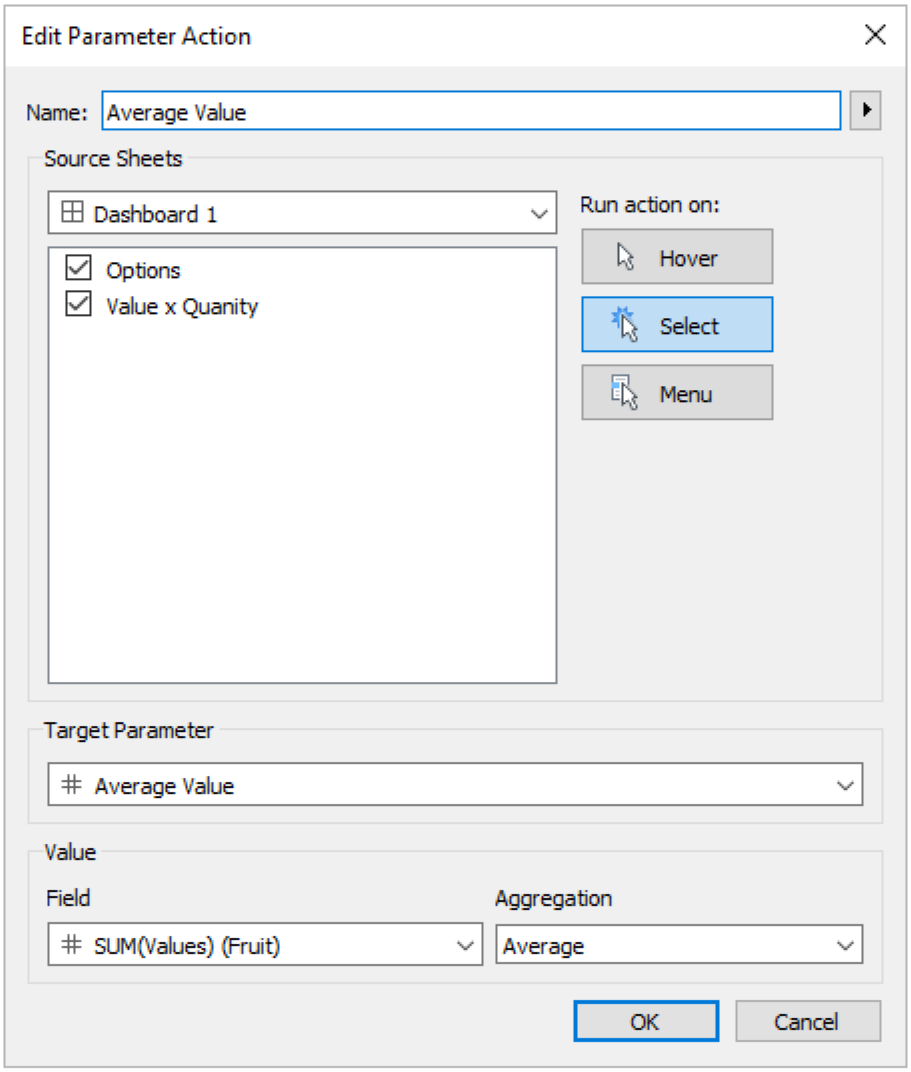


3. The two sheets are combined in a dashboard. To create the parameter action, click the dashboard tab, and then select **Dashboard** > **Actions** > **Add Action** > **Change Parameter**.

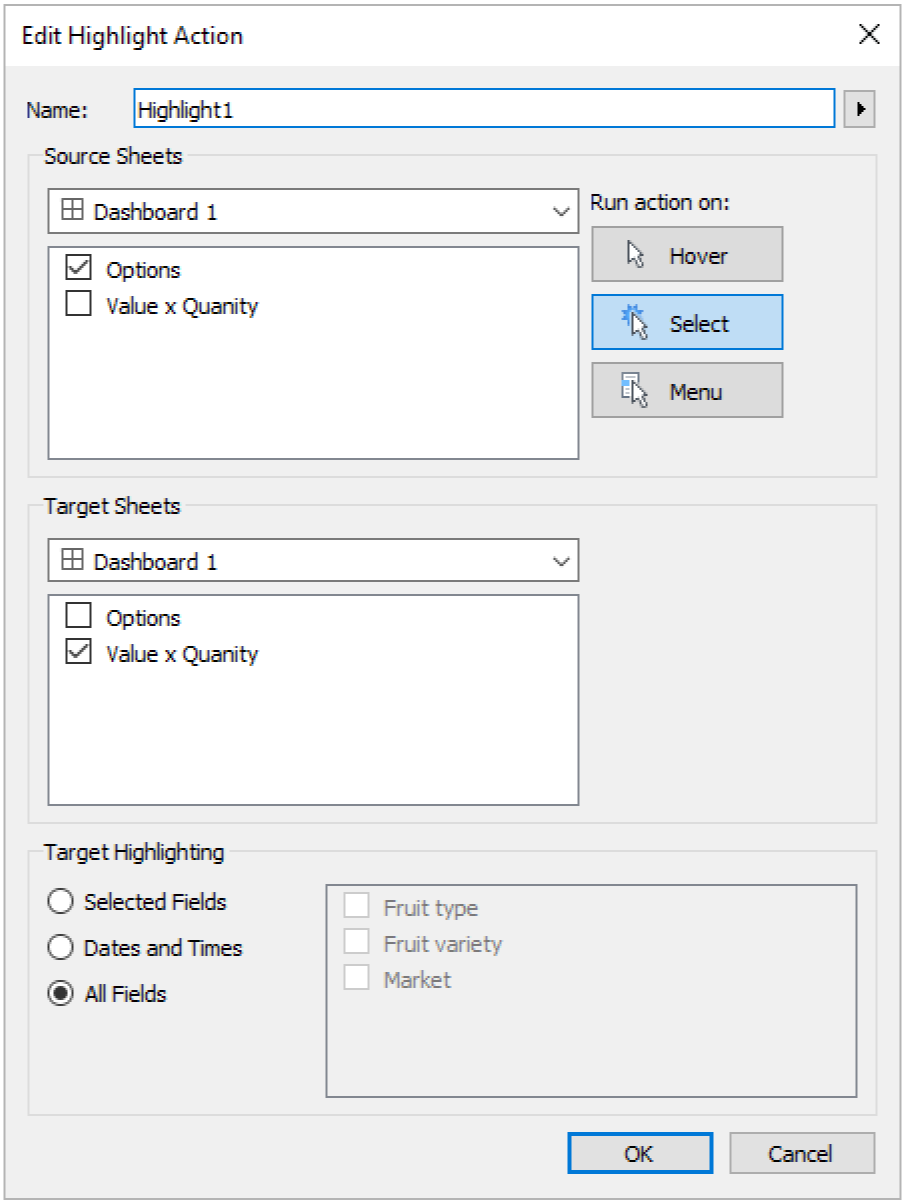
   The action is named *Average Value*. The target parameter references the Average Value parameter. The source field is set to SUM(Values) (Fruit).

The setting for aggregation is where the summary statistic information comes into play. By choosing Average, Median, Count, etc., you set the parameter to update with the statistic of your choosing.

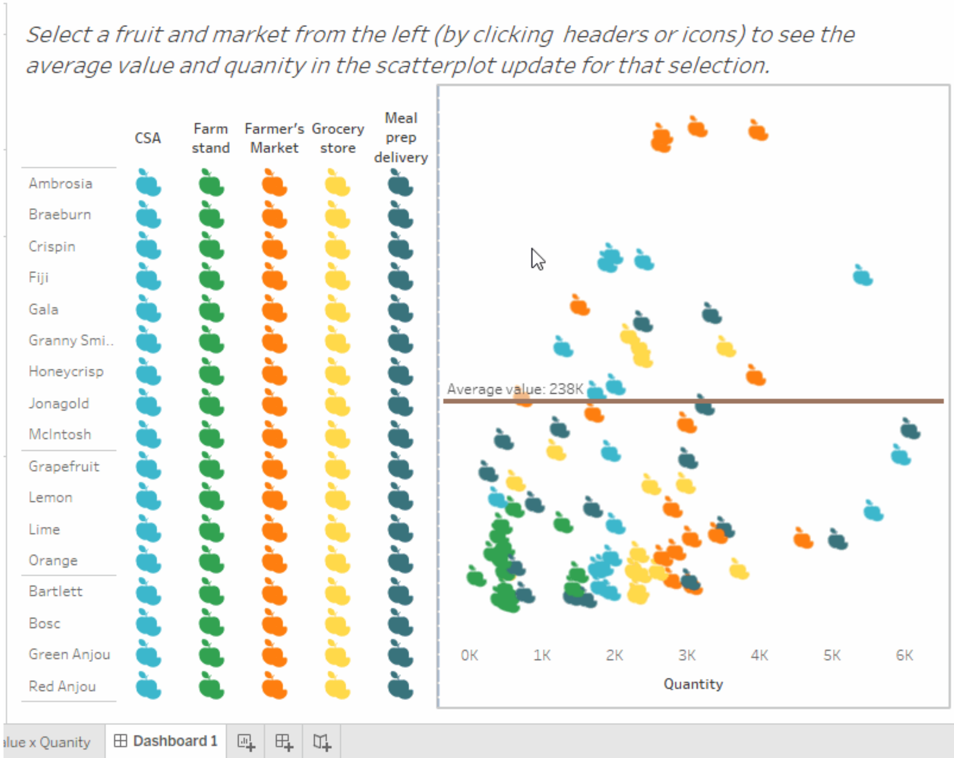
The aggregation is set to Average. Also note that the aggregation for the source field in the view can be different from the aggregation that you use for the parameter action.



Optionally, you could also create a highlight action for marks selected in the Options view.



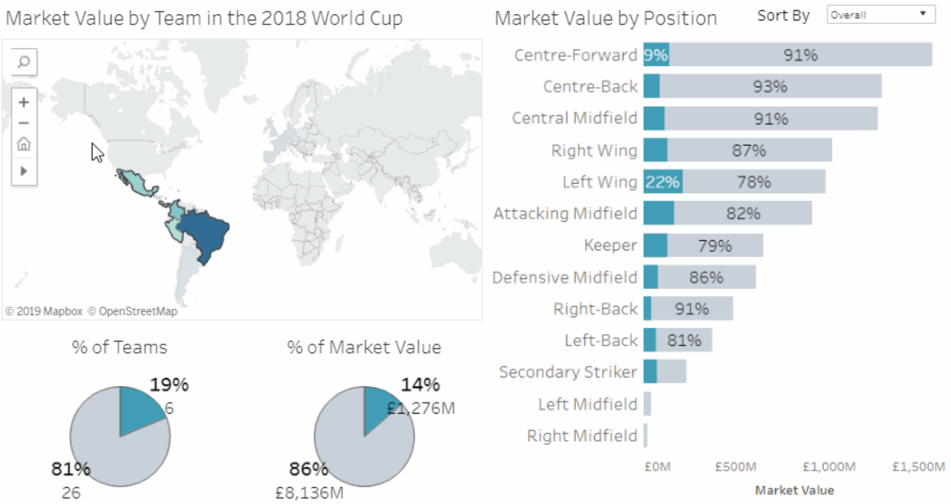
4. Test the parameter action in the dashboard. As you select headers and marks, the reference line updates dynamically and displays the average of the values.



Set Actions

Use set actions to let your audience interact directly with a viz or dashboard to control aspects of their analysis. When someone selects marks in the view, set actions can change the values in a set. For examples on how set actions can be used, see **Examples of set actions** on page 1545.

Watch a video: To see related concepts demonstrated in Tableau, watch Set Actions, a 7-minute free training video. Use your tableau.com account to sign in.



Set actions take an existing set and update the values contained in that set based on user actions in a viz. You can define the set action to include:

- the source sheet or sheets it applies to.
- the user behavior that runs the action (hover, select, or menu).
- the target set (the data source and set to be used).
- what happens when the selection is cleared.

To change or affect a visualization, the set referenced in the action must be used somehow in the visualization. You can do this in different ways, such as using the set in a calculated field that you then use to build the viz, or by placing the set in the view or on a Marks card property.

You can create set actions in Tableau Desktop, Tableau Online, and Tableau Server.

General steps for set actions

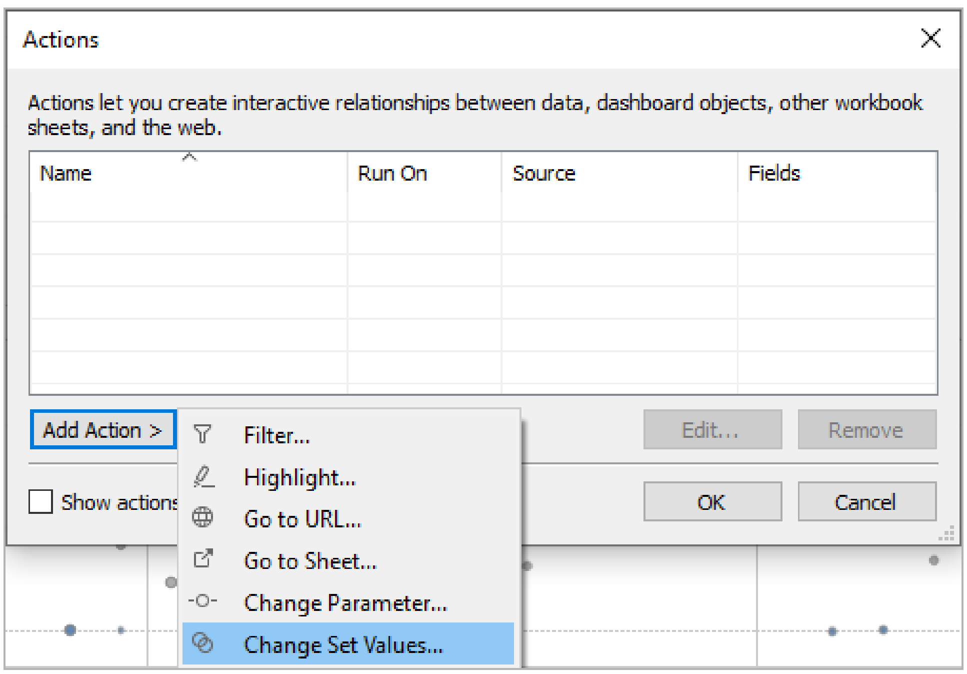
1. Create one or more sets. The sets you create will be associated with the data source that is currently selected. For details on creating sets, see **Create Sets** on page 1186.
2. Create a set action that uses one of the sets you created. You can create multiple set actions for different purposes.
3. Depending on the behavior you want to make available to users for their analysis, you might want to create a calculated field that uses the set. For details on creating calculated fields, see **Create a calculated field** on page 2097.
4. Build a visualization that uses a set referenced by a set action. For example, if you create a calculated field that uses the set, build the view using that calculated field. Or, drag the set to Color in the Marks card.

5. Test the set action and adjust its settings as needed to get the behavior you want your audience to experience.

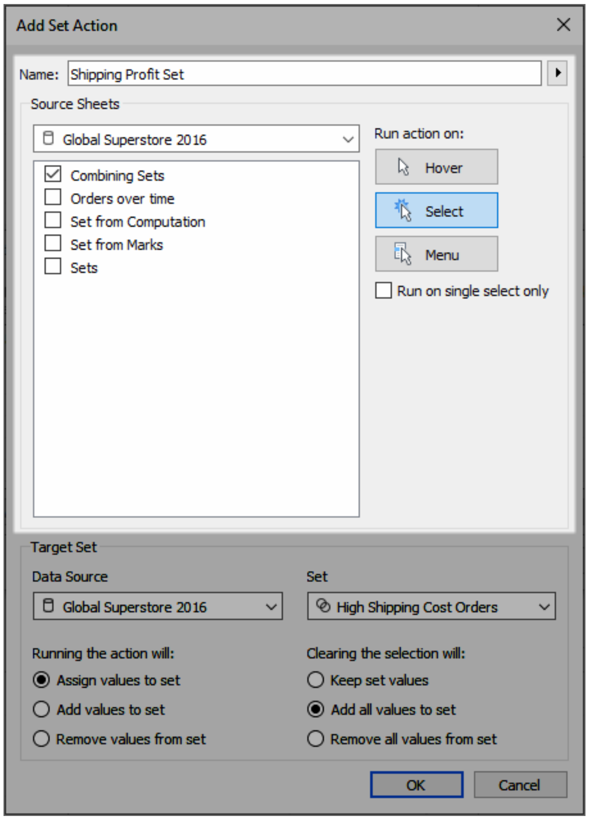
Getting Started: For an overview of what you need to know to create set actions, see On Your Marks, Get Set, Go: Everything You Need to Know to Get Started with Set Actions on the Data Viz Linds blog.

Create a set action

1. In a worksheet, select **Worksheet >Actions**. In a dashboard, select **Dashboard >Actions**.
2. In the Actions dialog box, click **Add Action** and then select **Change Set Values**.



3. In the **Add/Edit Set Action** dialog box, specify a meaningful name for the action.
4. Select a source sheet or data source. The current sheet is selected by default. If you select a data source or dashboard you can select individual sheets within it.

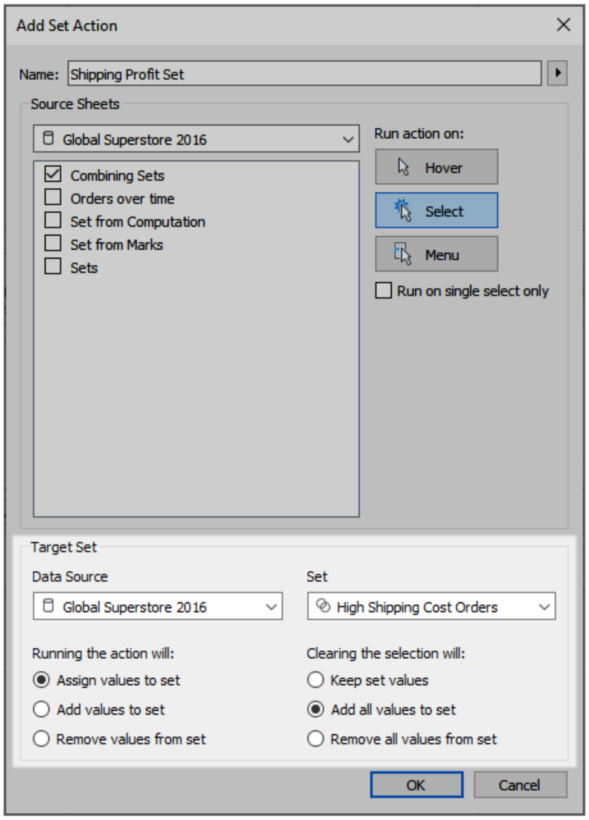


5. Select how users will run the action.

   - **Hover** - Runs when a user hovers the mouse cursor over a mark in the view.
   - **Select** - Runs when a user clicks a mark in the view. This option works well for set actions.
   - **Menu** - Runs when a user right-clicks (control-clicks on Mac) a selected mark in the view, then selects an option on the context menu.

6. Specify the target set. First, select the data source, and then select the set.

   The sets available in the Target Set lists are determined by the data source that you select and the sets that you have created in the workbook that are associated with that data source.



In this example, the target set is using *Global Superstore 2016* as the data source and *High Shipping Cost Orders* as the set.

7. Specify what happens when the action is run in the view:

   - **Assign values to set** - Replaces all values in the set with the values selected.
   - **Add values to set** - Adds individually selected values to the set.
   - **Remove values from set** - Removes individually selected values from the set.

   Note: Selecting **Add values to set** or **Remove values from set** via the Set Action dialog modifies the values selected in the General tab of the set's definition—just like **Assign values to set** does. This differs from the **Add to** and **Remove from** functionality available only in authoring mode via the set's context menu in the tooltip. The latter unions or removes values to and from the set after the condition specified has been applied.

8. Specify what happens when the selection is cleared in the view:

- **Keep set values** - Current values in the set remain in the set.
- **Add all values to set** - Adds all possible values to the set.
- **Remove all values from set** - Removes previously selected values from the set.

9. Click **OK** to save your changes and return to the view.
10. Test the set action by interacting with the visualization. Tweak some of the settings for the action to adjust the selection behavior as needed.

Examples of set actions

The following examples show just a few of the ways you can use set actions. To follow along, you will need to be familiar with creating and using sets and calculated fields. For details, see **Create Sets** on page 1186 and **Create a calculated field** on page 2097.

Learn more: For additional in-depth examples of how you can use set actions demonstrated in Tableau, see:

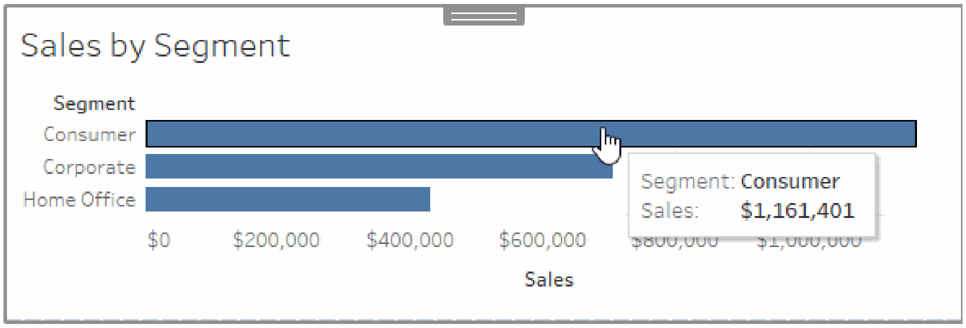
- 8 ways to bring powerful new comparisons to viz audiences with set actions in the Tableau blog
- 8 analytic concepts to express with Tableau set actions (Tableau blog)
- Designing advanced analytics applications with Tableau set actions (Tableau blog)
- Topics on set actions on the Data Viz Linds blog
- Set action examples in the Art of the Vizable website by TabLabs.

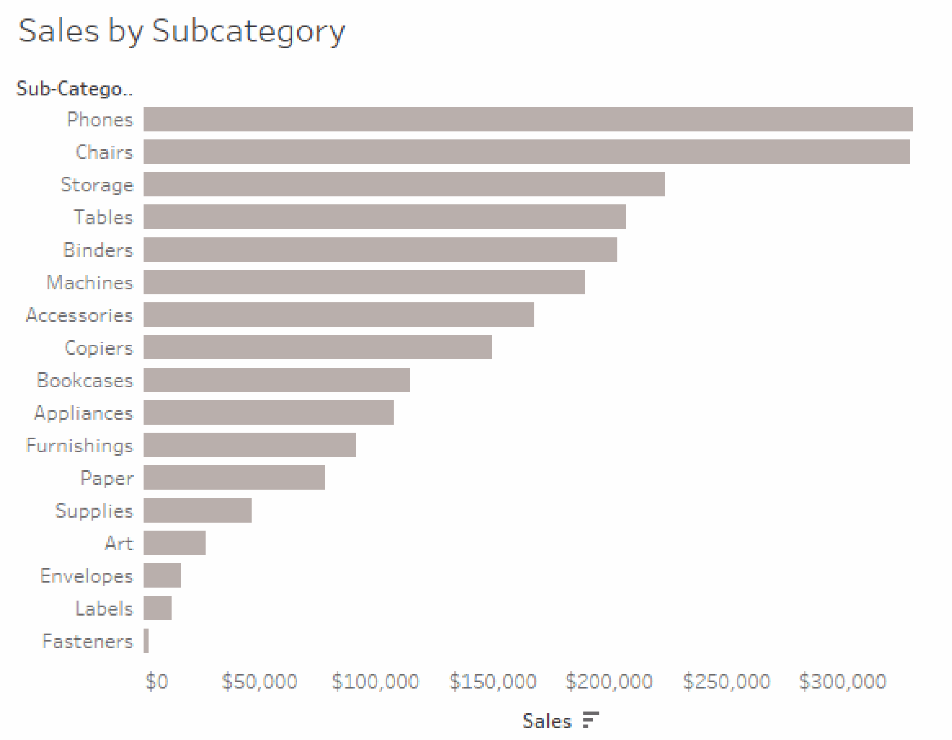
Watch a video:

- Ready, set, action! (1 hour)
- Sets Appeal (1 hour)
- Beyond set actions | Interactive statistics (1 hour)

Proportional brushing

A classic use case for set actions is proportional brushing, described in How to do proportional highlighting with set actions on the Tableau blog and Tableau Set Actions from The Information Lab blog (along with other examples).

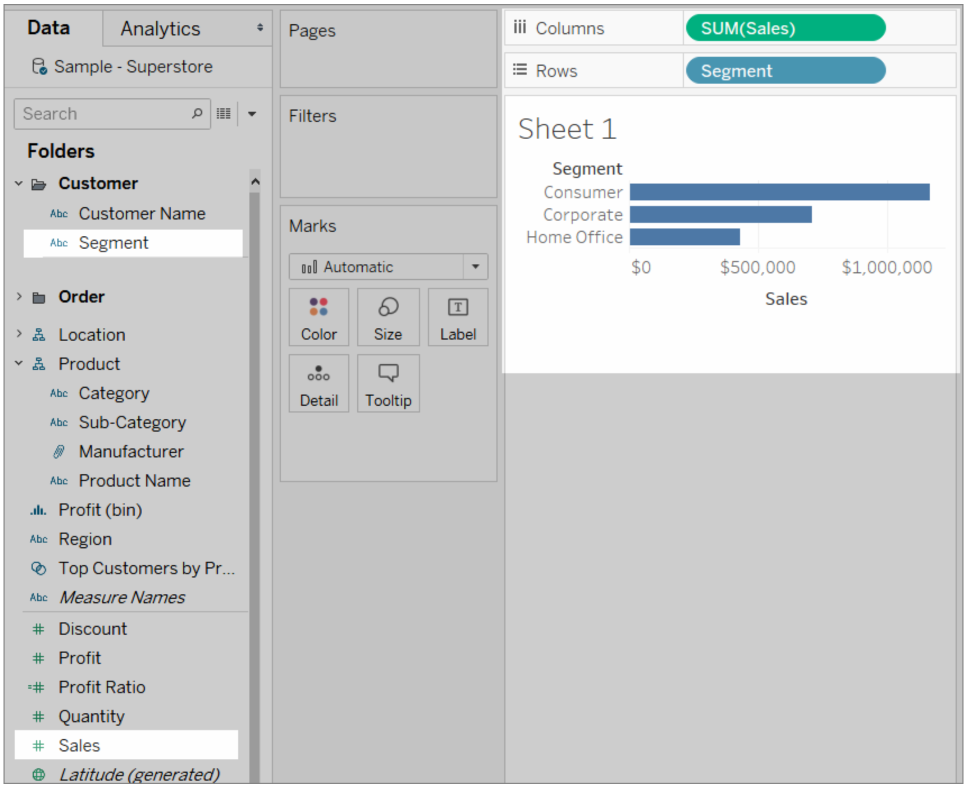




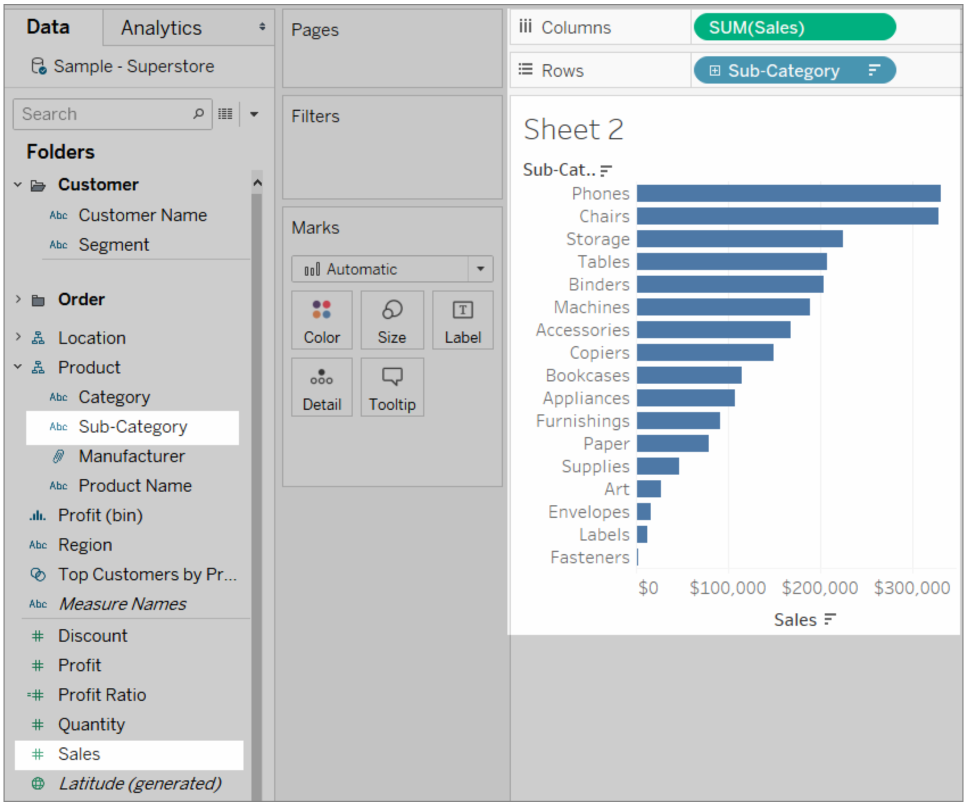
Proportional brushing example details

This example uses the **Sample - Superstore** data source. This set action changes the color of existing marks in the view based on the proportion of the mark that is in or out of the set.

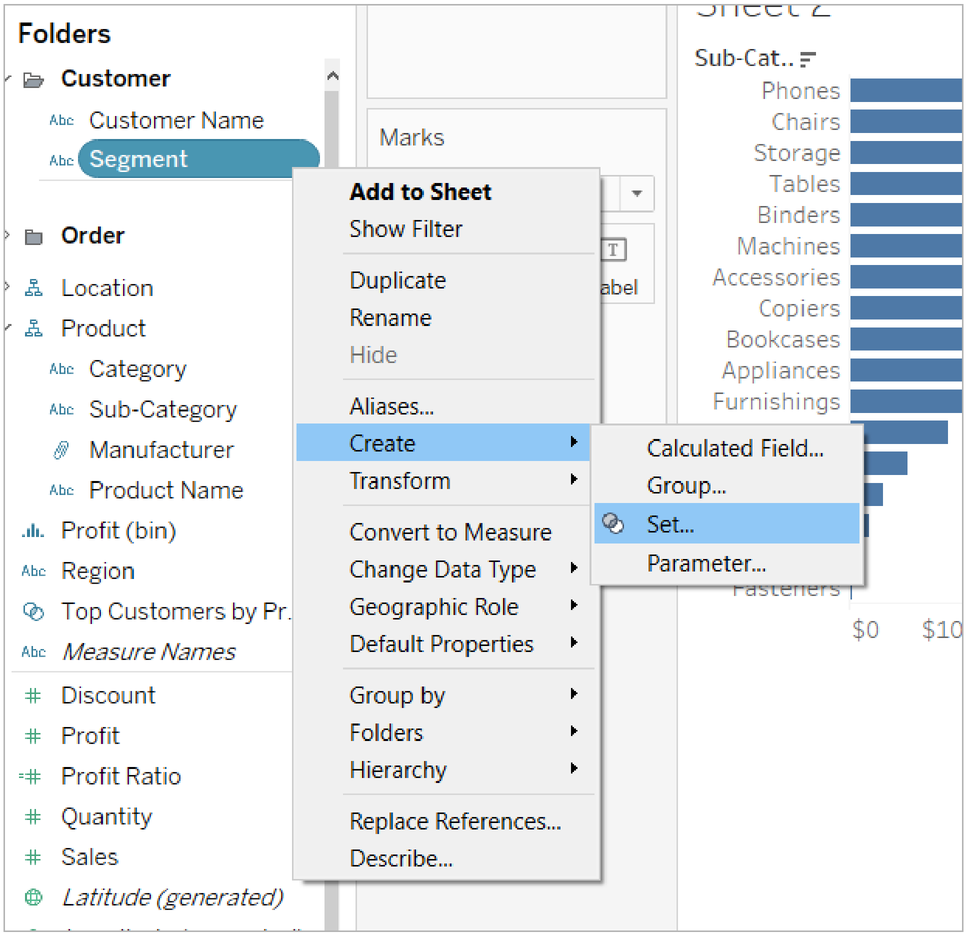
1. Connect to **Sample - Superstore** data in Tableau Desktop.
2. In a new sheet, drag the **Sales** measure to Columns, and drag the **Segment** dimension to Rows.



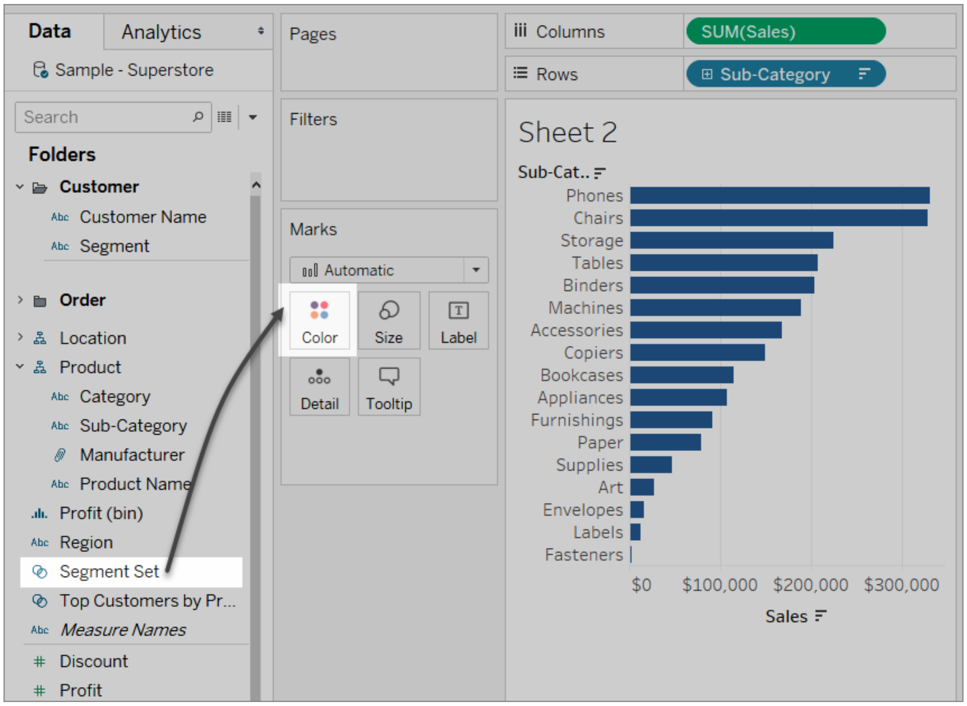
3. In another blank sheet, drag the **Sales** measure to Columns, and drag the **Sub-Category** dimension to Rows.



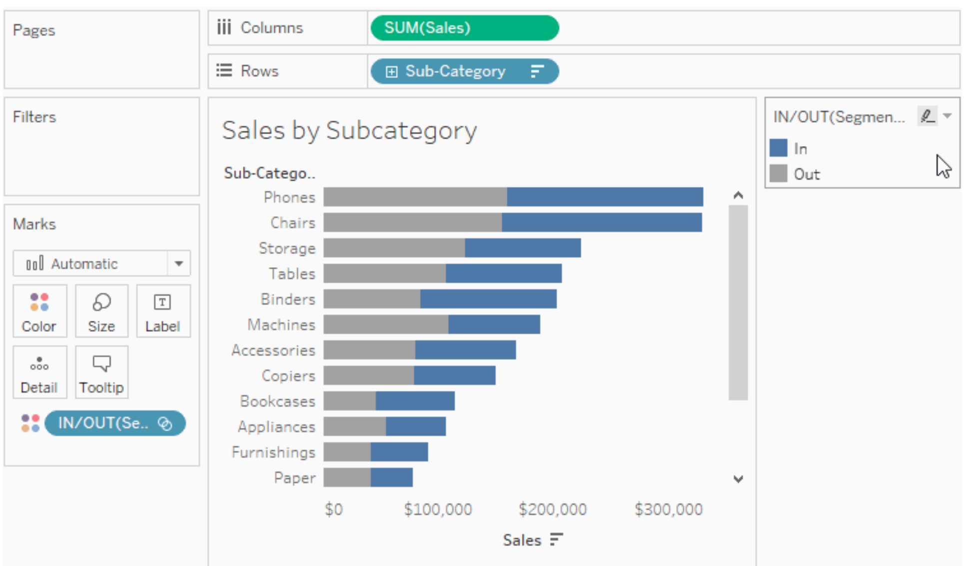
4. Create a set for the **Segment** dimension named *Segment Set*.



5. In the sheet that shows Sales by Sub-Category, drag **Segment Set** onto **Color** in the Marks card.

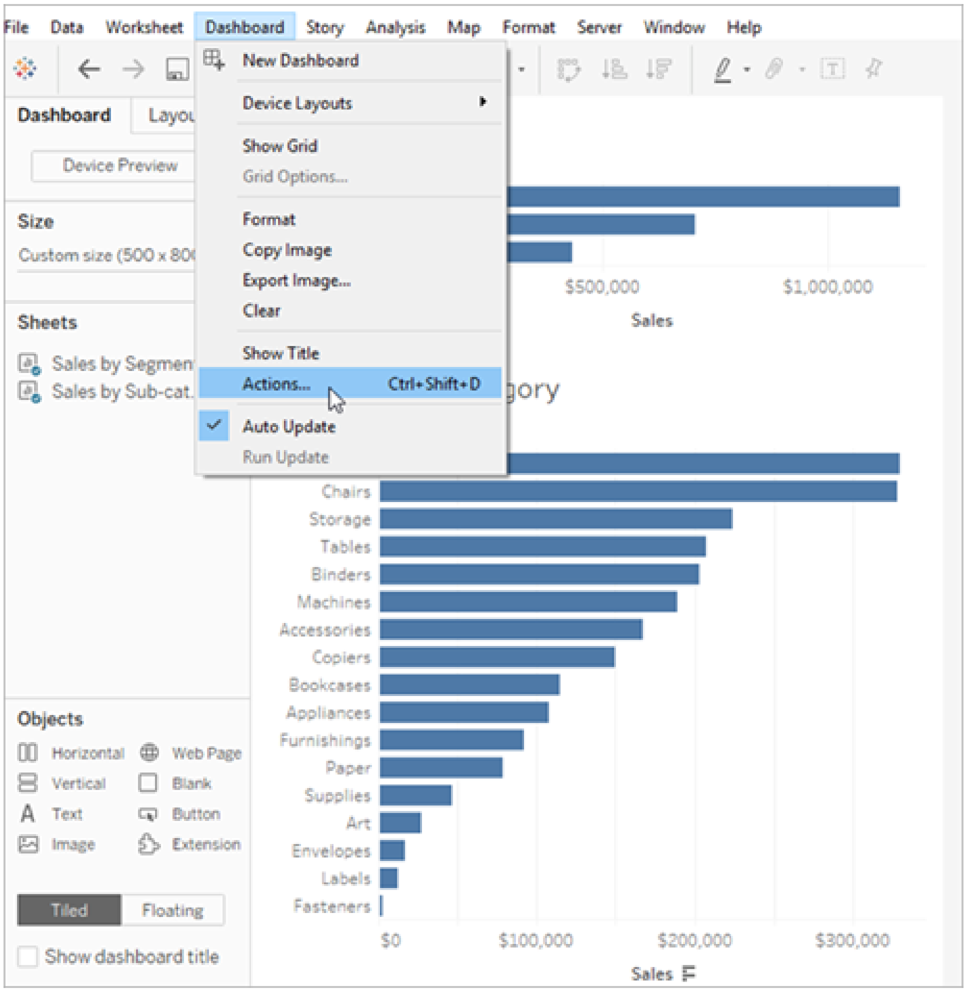


Swap the colors used to show marks IN or OUT of the set.

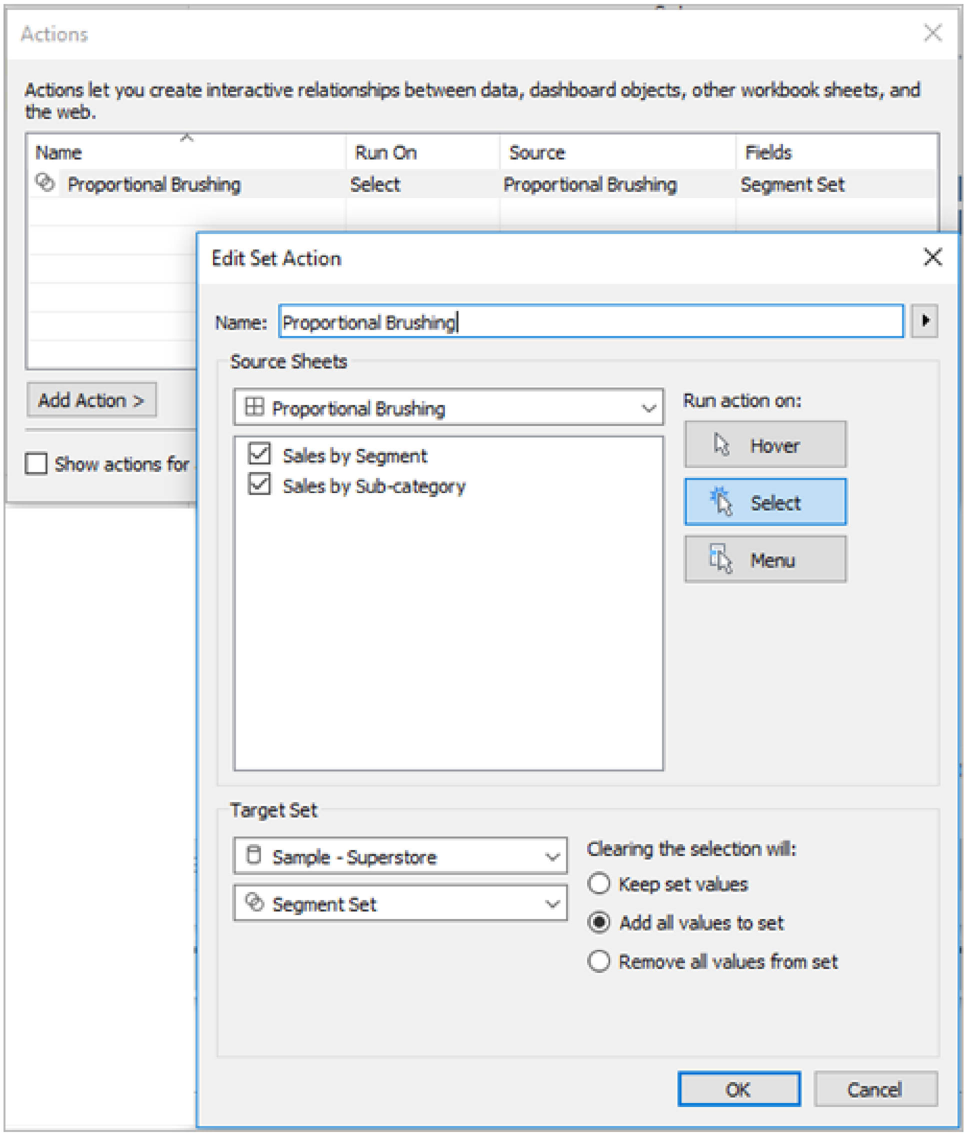


6. Create a new dashboard. Drag both sheets into the dashboard.

   On the **Dashboard** menu, select **Actions**. Click **Add Action**, and then select **Change Set Values**.



7. Configure the action using the same settings as shown in the image below.

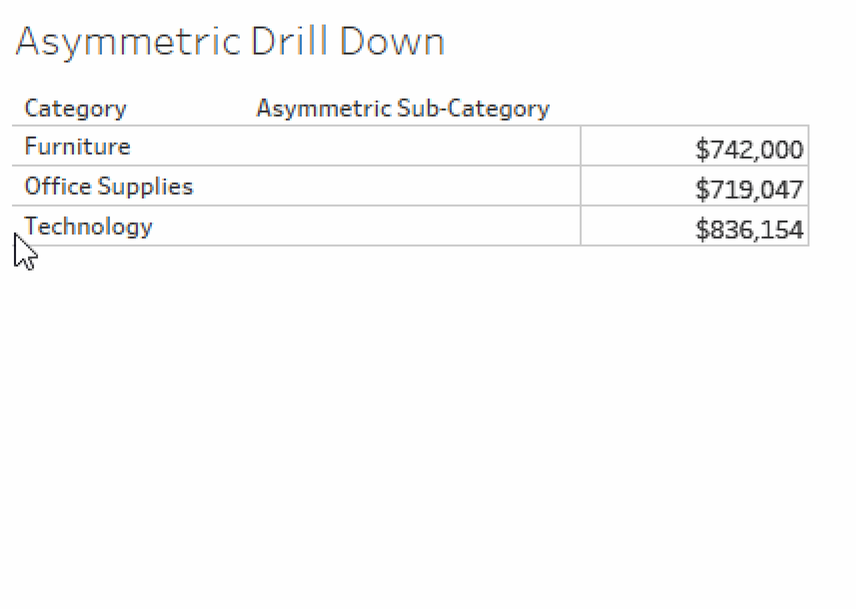


8. Click **OK** to save your changes and return to the view.

9. Test the set action by clicking the marks for each segment.

Asymmetric drill down

Set actions can also be used across multiple sets for more complex interactions. If a data set has hierarchical information, such as Category, Sub-Category, and Manufacturer, normally drilling down the hierarchy expands all values at a given level. However, you can use set actions for asymmetric drill down, opening only the next hierarchy level for the selected value.



Asymmetric drill down example details

This example uses the **Sample - Superstore** data source.

Set actions are used here in conjunction with two calculated fields to create an asymmetric drill down experience.

1. Create a set for the Category field named *Category set*.

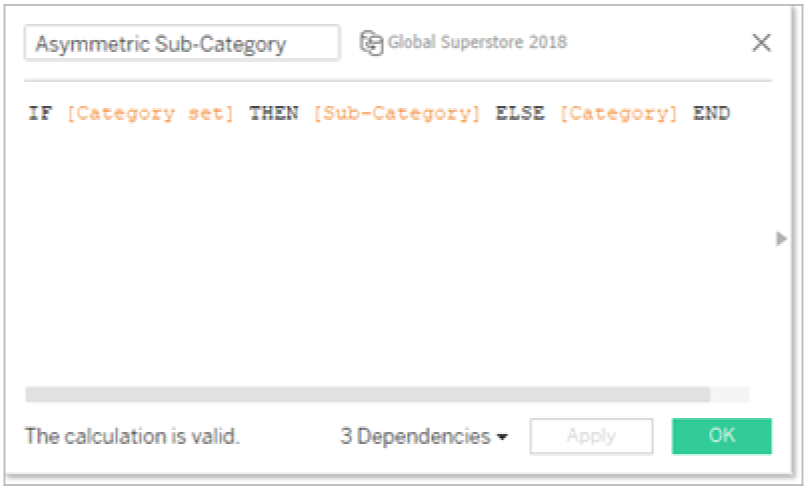
   Choose any category to be a member of that set. The membership is temporary and will be overwritten by the set action.

2. Create a calculated field named *Asymmetric Sub-Category*.

   Right-click the **Sub-Category** field in the Data pane, and then select **Create Calculated Field**.

   Type the following calculation:

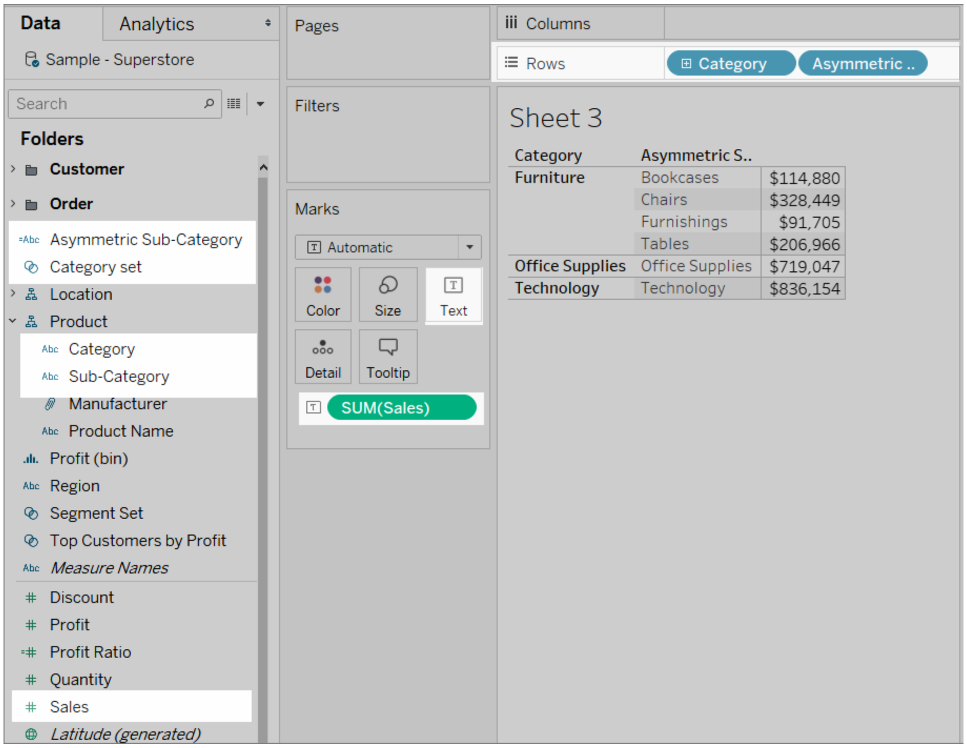
   ```
   IF [Category set] THEN [Sub-Category] ELSE [Category] END
   ```



If the category of the mark or marks in the view is in the Category Set, the calculation will return the Sub-categories for that Category. Otherwise, it will return the Category again.

The set isn't in use in the view yet and has no action associated with it, so it is static for the moment.

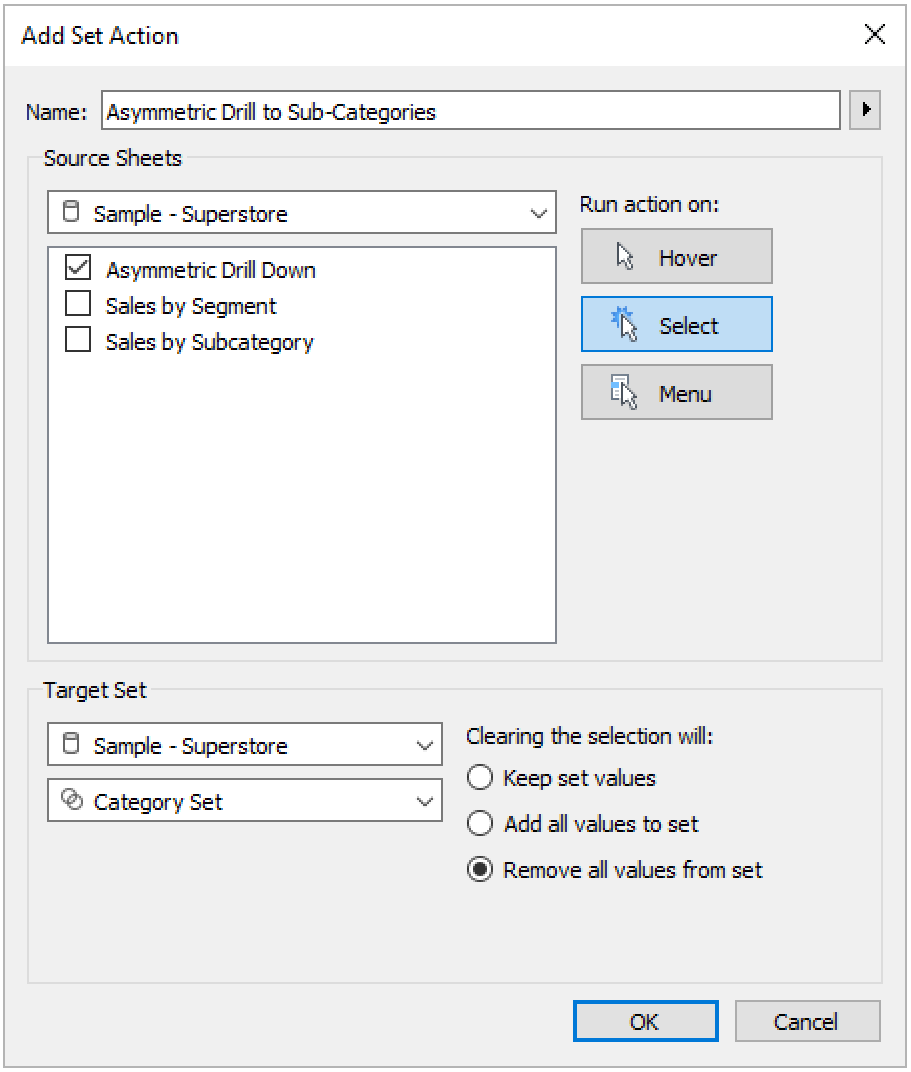
3. Build the view by dragging the **Category** and **Asymmetric Sub-Category** fields to Rows (in that order). Then Drag **Sales** to Text in the Marks card.



4. Create a set action.

   On the Worksheet menu, select **Actions**. Click **Add Action**, and then select **Change Set Values**.

   Name the action *Asymmetric Drill to Sub-Category*.



Select the following options:

- Source Sheets: Make sure the only the current sheet is selected.
- Run action on: Choose **Select**. This will make the action apply when the user selects a mark or marks in the view.
- Target Set: Select the current data source from the first drop-down list and the Category set from the second drop-down list.
- Clearing the selection: Select **Remove all values from set**.

Click **OK** to save your changes.

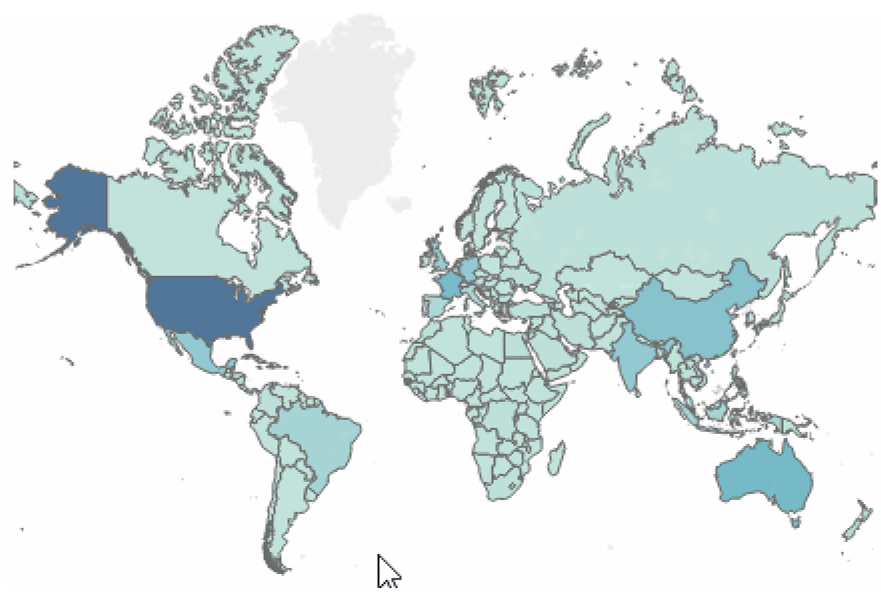
5. Test the set action.

When you click on a header or mark in the view, it will expand out to the next level. Clicking a category bar will expand its sub-categories.

Color scaling

Color values in a viz often can be distorted by outliers. When one country has a significantly higher value than other countries, the nuance of differences between those non-extreme values can be lost.

A set action can update the values in a Country set based on the selection in the viz. That set is then used in a calculated field that computes the value of the relevant measure for only countries in that set, and that calculated field is placed on color. Now, when countries are selected in the map, the color palette is based only on those countries, showing greater visual differences between those values.



Color scaling example details

This example uses the Global Superstore data source.

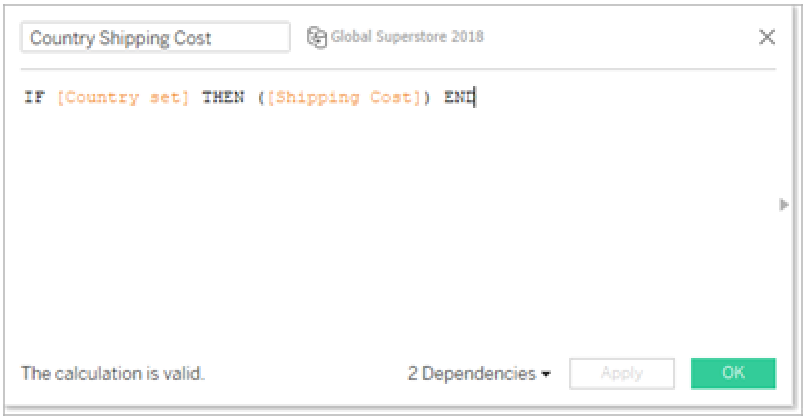
1. Create a set on the **Country** field named *Country set*. Chose any country to be a member of the set. The membership is temporary and will be overwritten by the set action.

2. Create a calculated field named *Country Shipping Cost*.

Right-click in the Data pane, and then select **Create Calculated Field**.

Type the following calculation:

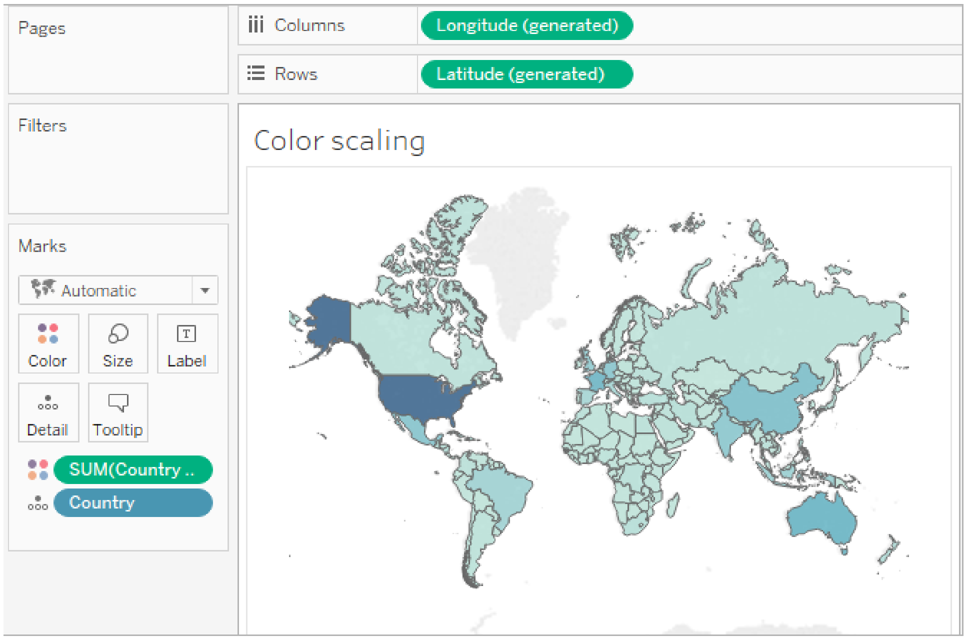
```
IF [Country set] THEN ([Shipping Cost]) END
```



If the country of the mark(s) in the view is in the Country set, the calculation returns the Shipping Cost. Otherwise, it does nothing.

The calculated field ties the set into the view.

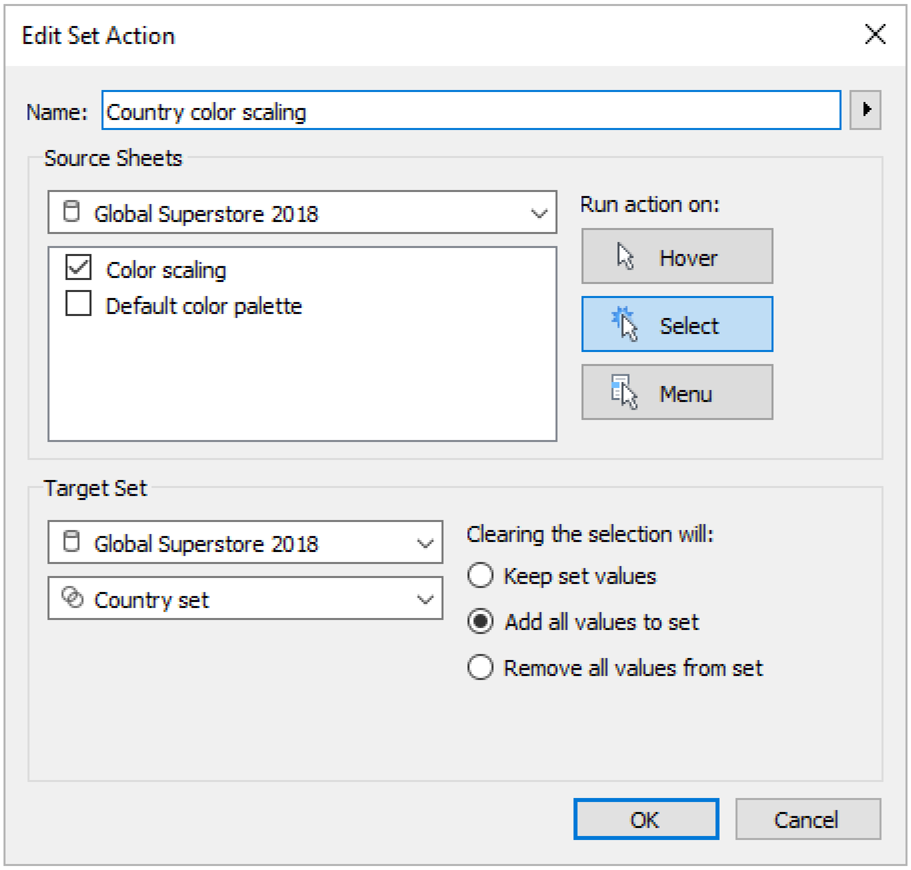
3. Double-click the **Country** dimension to automatically create a map.
4. Drag **Country Shipping Cost** to Color in the Marks card. The map should automatically update to a filled map.



5. Create the set action: *Country color scaling*.

   On the Worksheet menu, select **Actions**. Click **Add Action**, and then select **Change Set Values**.

   Name the action *Country color scaling*.



Select the following options:

- Source Sheets: Select only the current sheet.
- Run action on: Choose **Select**. This will make the action apply when the user selects a mark or marks in the view.
- Target Set: Select the current data source from the first drop-down list and the Category set from the second drop-down list.
- Clearing the selection: Select **Add all values to set**.

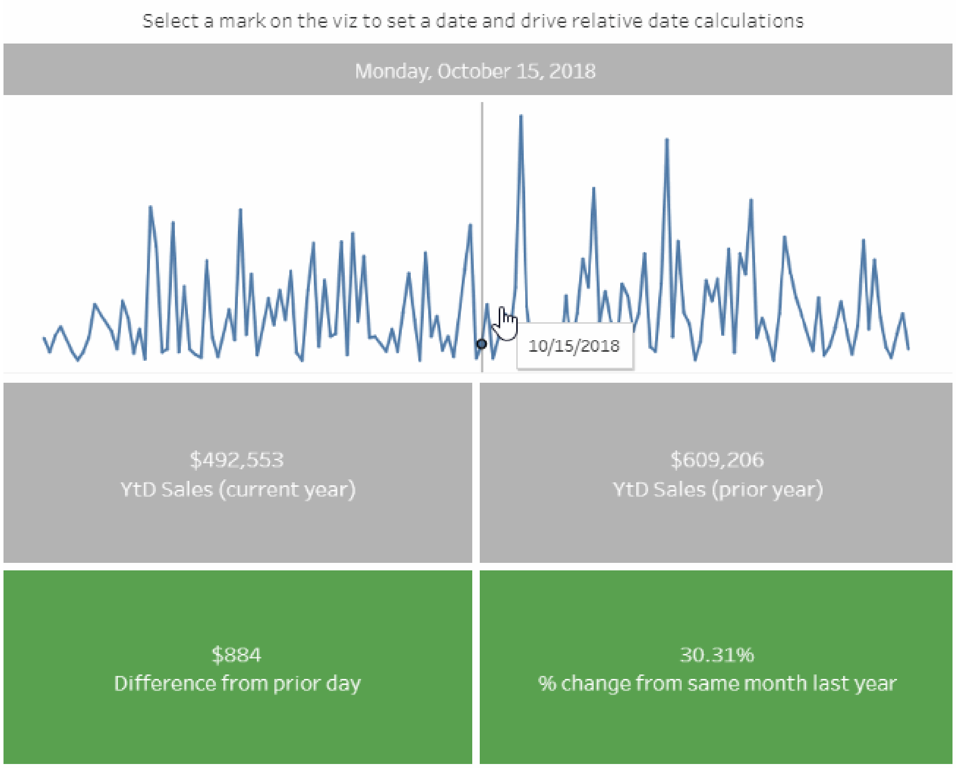
6. Click **OK** to save your changes.
7. Test the set action.

   When you select a mark or group of marks in the view, the color palette responds by representing the values for those marks, not the entire data set.

Relative dates

This dashboard example contains multiple relative date calculations. Based on a target date, the dashboard displays the difference from the previous day, the percent change from the same month in the previous year, the Year to Date sales for the current year, and Year to Date sales for the previous year.

The set action in this dashboard updates the target date and relevant date calculations in response to a user clicking a mark on a timeline.



Relative date example details

This example uses the Global Superstore data source.

The set action in this example is applied to the Timeline sheet in the dashboard, and uses that sheet to drive the behavior of the dashboard. When a user selects a mark in the Timeline view, the set action updates the Selected Date set with a single date value.

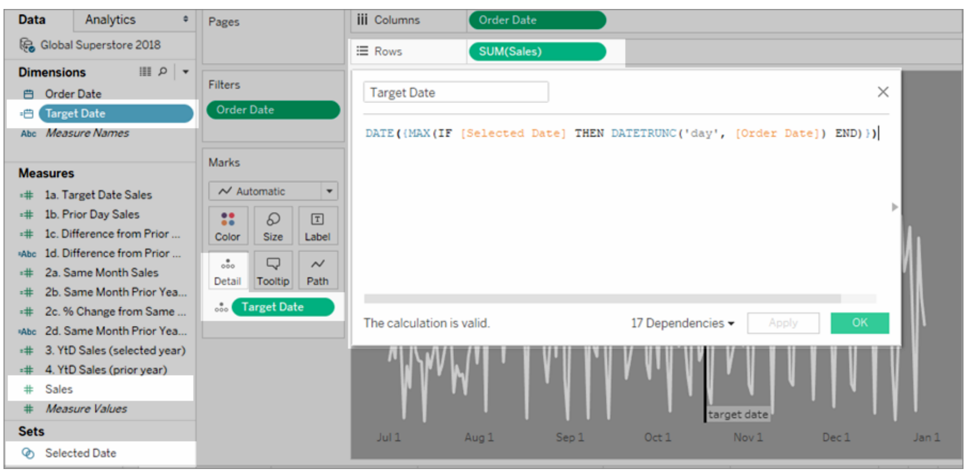
In this example:

- A Selected Date set is built on the Order Date field.
- An Update Target Date set action updates the Selected Date set so the date value for the mark selected in the view is the only value in the set.
- A Target Date calculated field captures the date value from the Selected Date set, and that calculated field is referenced by other relative date calculations in the viz.
- Each sheet in the dashboard uses calculated fields to update its displayed values.

Note: Because this example is more complex, the description below gives a summary of the major steps and settings involved rather than specific details.

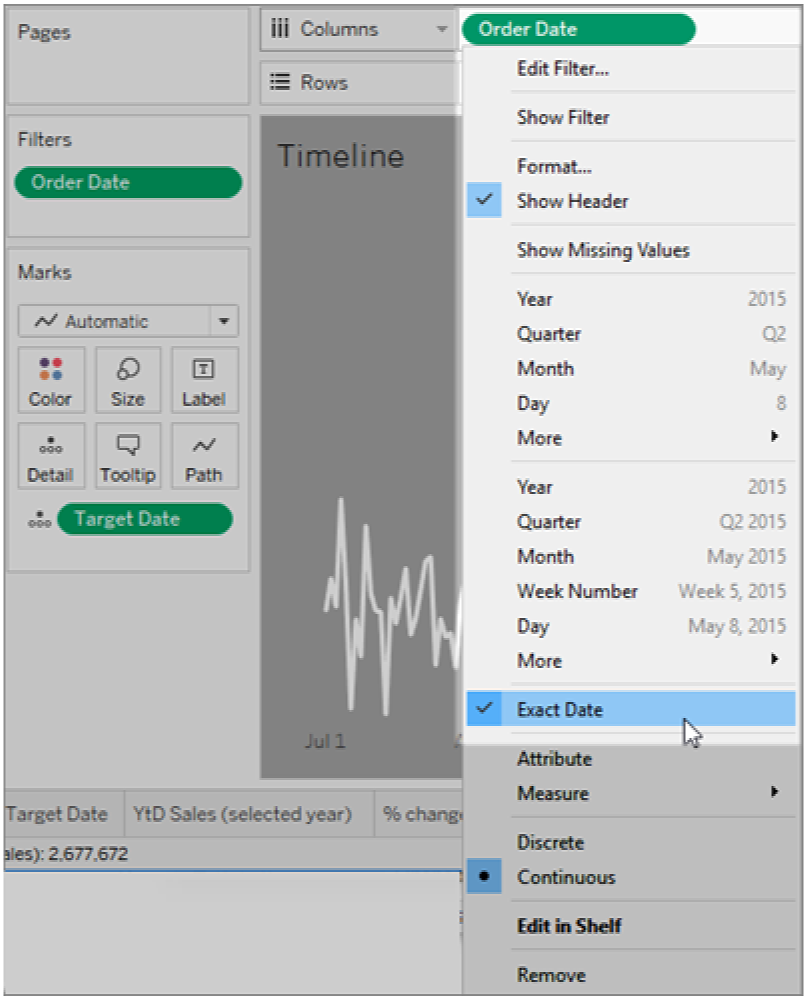
Timeline sheet settings

Sales is on Rows. A calculated field named **Target Date** is placed on Detail to make it available to the reference line.

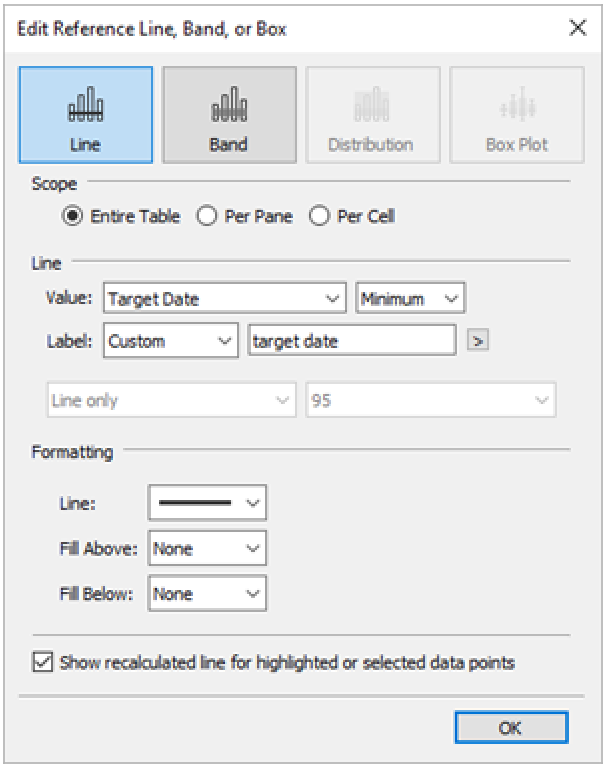


Starting in Tableau 2020.2: The above image doesn't reflect the latest user interface. For each table or folder in the Data pane, a gray line separates dimensions (above line) and measures (below line). The Dimensions and Measures labels are no longer used in the Data pane.

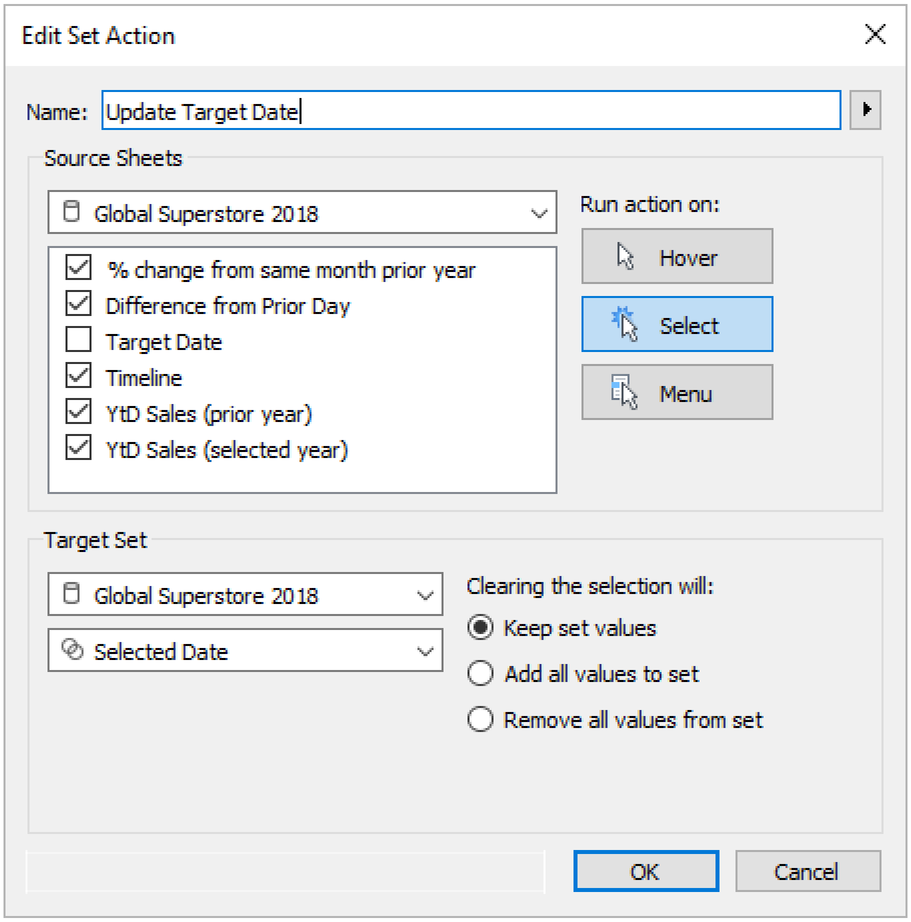
Order Date is on Columns as Exact Date.



The reference line is applied to the Order Date field and is set to be the Minimum of Target Date.



A set action is applied to this sheet that updates the Selected Date set based on what mark is selected in the view.

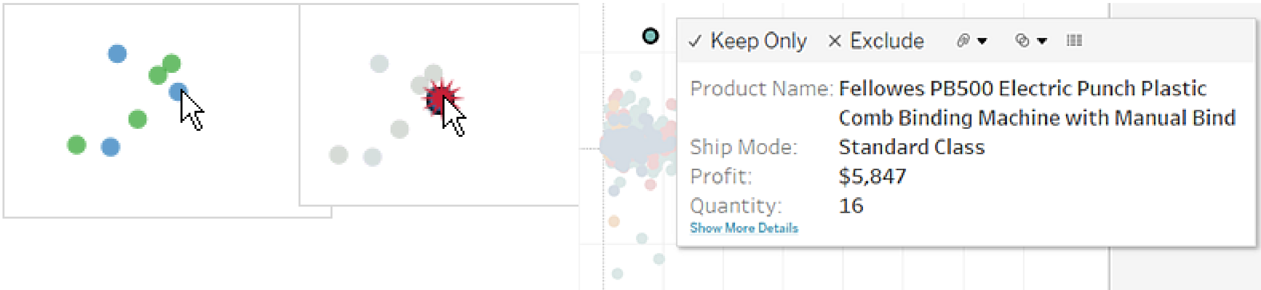


Running Actions

Depending on how the action is created, you can run an action using one of the following three methods:

- **Hover** - Rest the pointer over a mark in the view to run the action. This option works well for highlight and filter actions within a dashboard.
- **Select** - Click on a mark in the view to run the action. This option works well for all types of actions.
- **Menu** - On Tableau Desktop, right-click (control-click on Mac) a selected mark in the view and then select an option on the context menu. On Tableau Server or Online, left-click (click on Mac) a selected mark in the view to see the context menu. This option works well for filter and URL actions.

Hover **Select** **Menu**



Links are not always visible for every worksheet and mark. Because links are mapped to specific fields in the data source, links will only be available for the worksheets that use the mapped fields. For example, if you add a hyperlink that uses both Latitude and Longitude as parameters in the link, the link will only be available to worksheets that use Latitude and Longitude in the view. Additionally, the link is only available on marks and headers that contain relevant values.

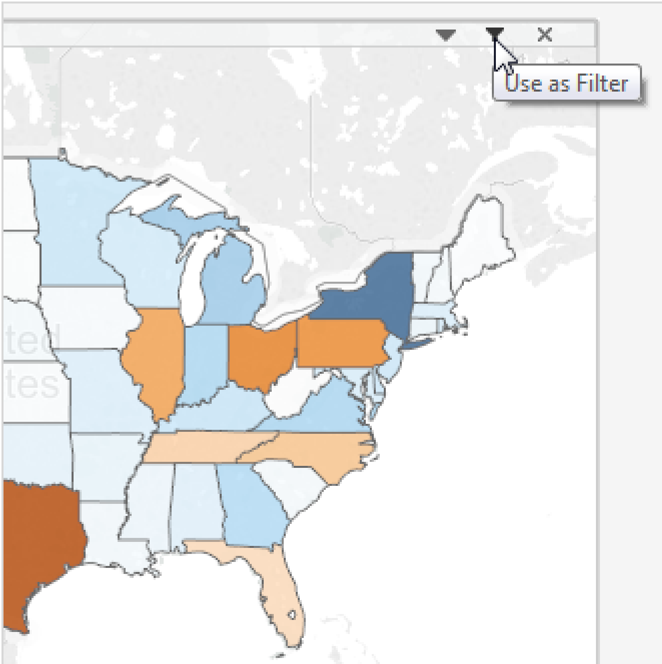
Actions and Dashboards

Actions often have unique behavior when the source or destination is a dashboard. Because a dashboard can contain multiple views, a single filter or highlight action can have broad impact. Dashboards can also contain web page objects, which you can target with interactive URL actions.

Use a single view to filter other views in a dashboard

Imagine you have a dashboard that contains three views about profitability: a map, a bar chart, and a table of customer names. You can use a filter action to make one of the views in your dashboard, such as the map, the controlling view for the filter action. When your users select a region in the map, the data in the other views is filtered so that it relates to just that region.

1. On the dashboard, select the view you want to use as a filter.
2. On the view's shortcut menu, choose **Use as Filter**. You can perform the same action by clicking the Use as Filter icon .



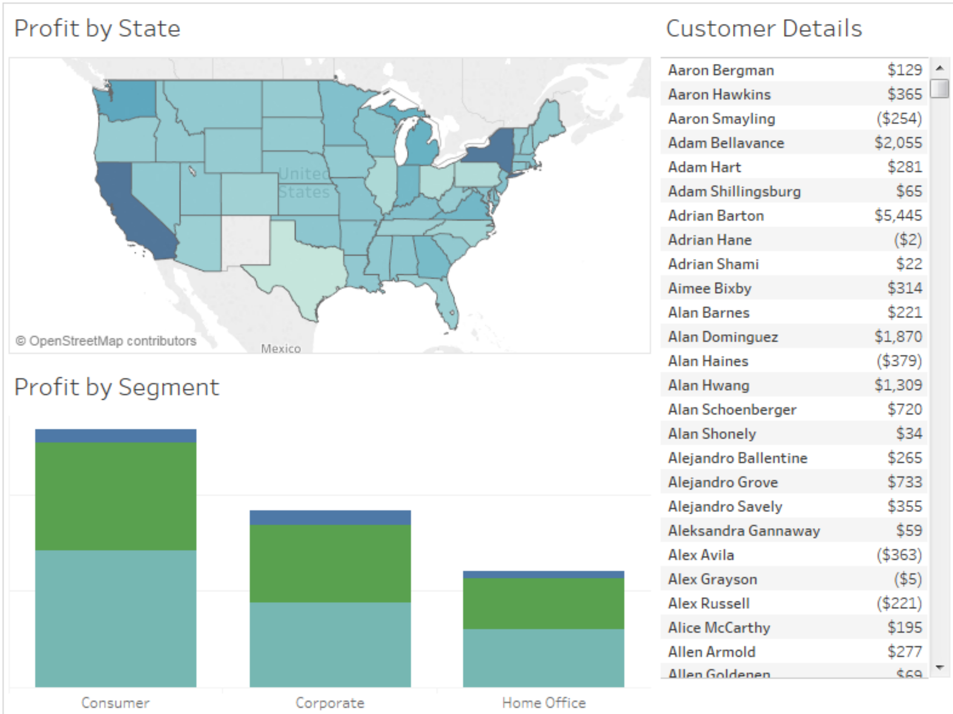
You can also use filter actions to filter the data on a dashboard when the data comes from multiple data sources. For more information, see Filtering Across Multiple Data Sources Using a Parameter in the Tableau Knowledge Base.

For more information about filter actions, see **Filter Actions** on page 1491.

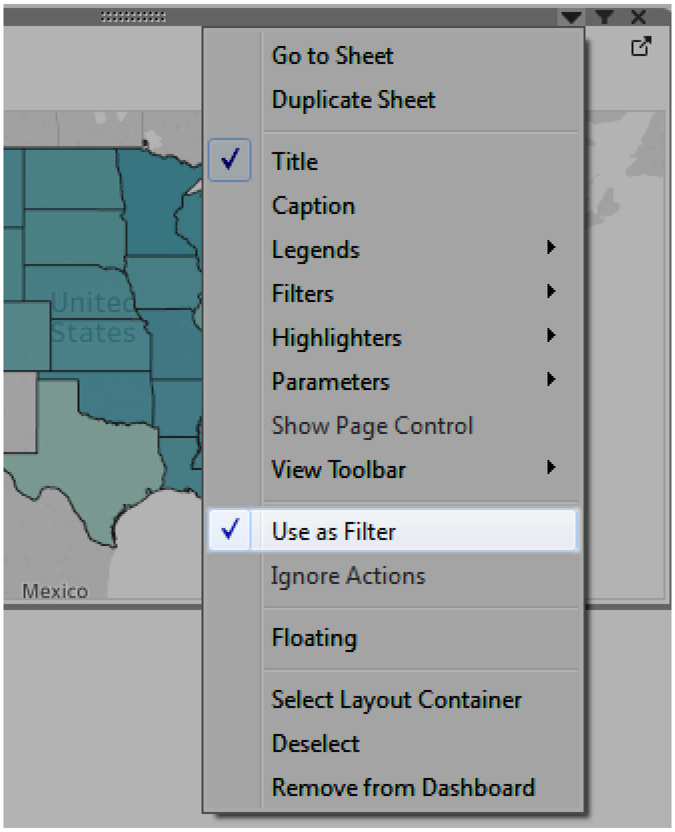
Use multiple views to filter other views in a dashboard

Similar to how you can use a single view to filter other views in a dashboard, you can also use multiple views as a filter. The trick is to not only use these views as filters, but to also disable their ability to be filtered themselves.

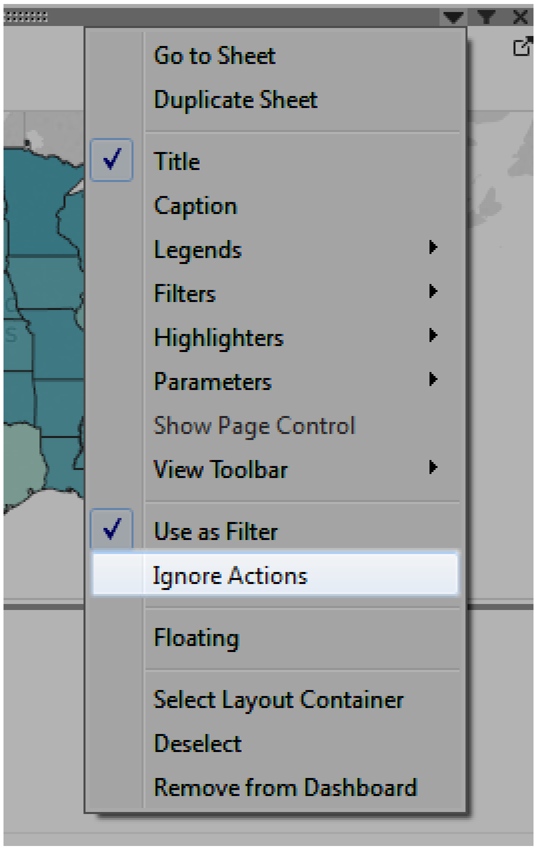
1. Create or open a dashboard that has at least three views.



2. Select the first view that you want to use as a filter (such as a map), and from its shortcut menu, select **Use as Filter**.

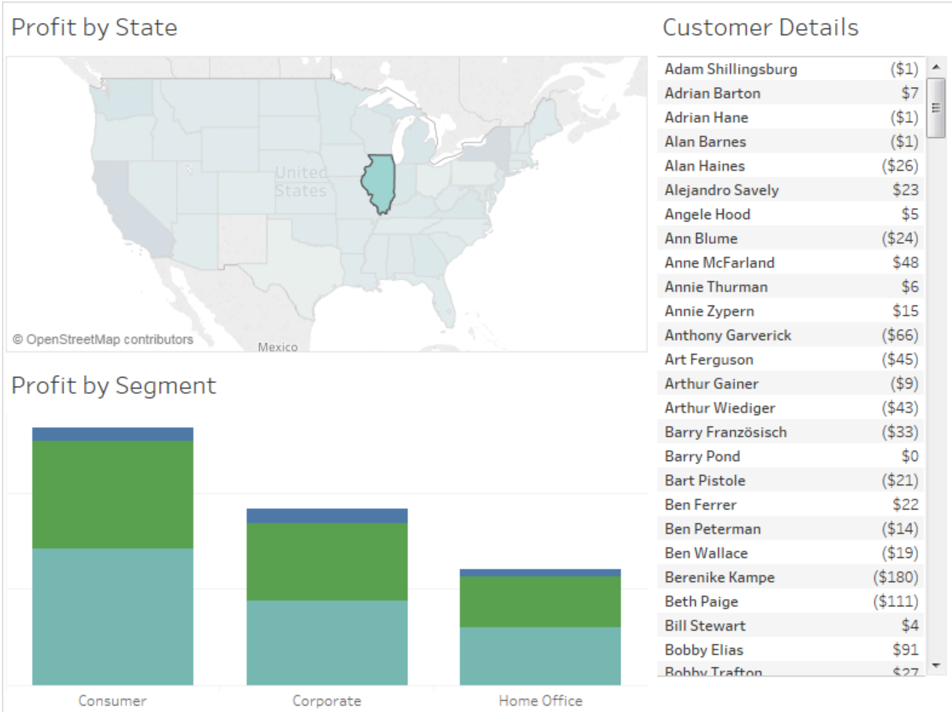


3. Open the same view's shortcut menu again and select **Ignore Actions**. This ensures that other filter actions, including the one you'll create next, will not affect this view.



4. Repeat the steps 2 and 3 for any other views you want to use as a filter.

   Now, selecting marks in these controlling views filters data in one or more detail views—all without affecting any other controlling views.

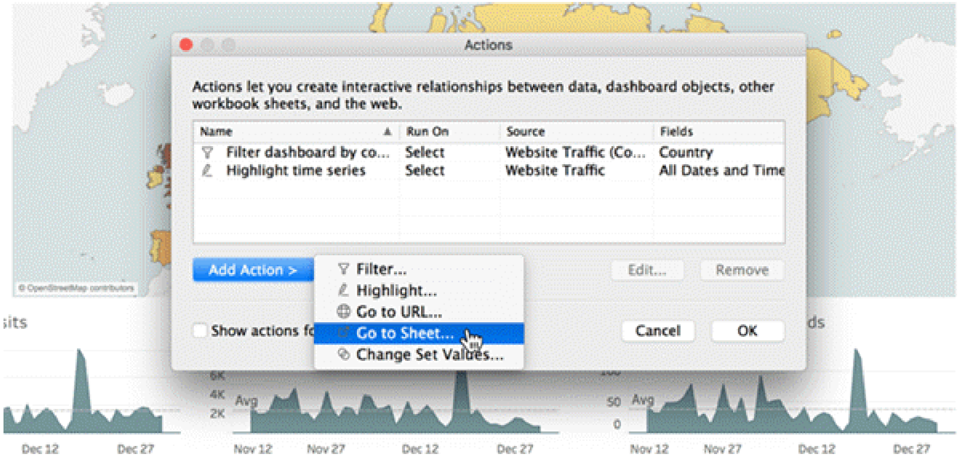


Navigate from one view to another view, dashboard, or story

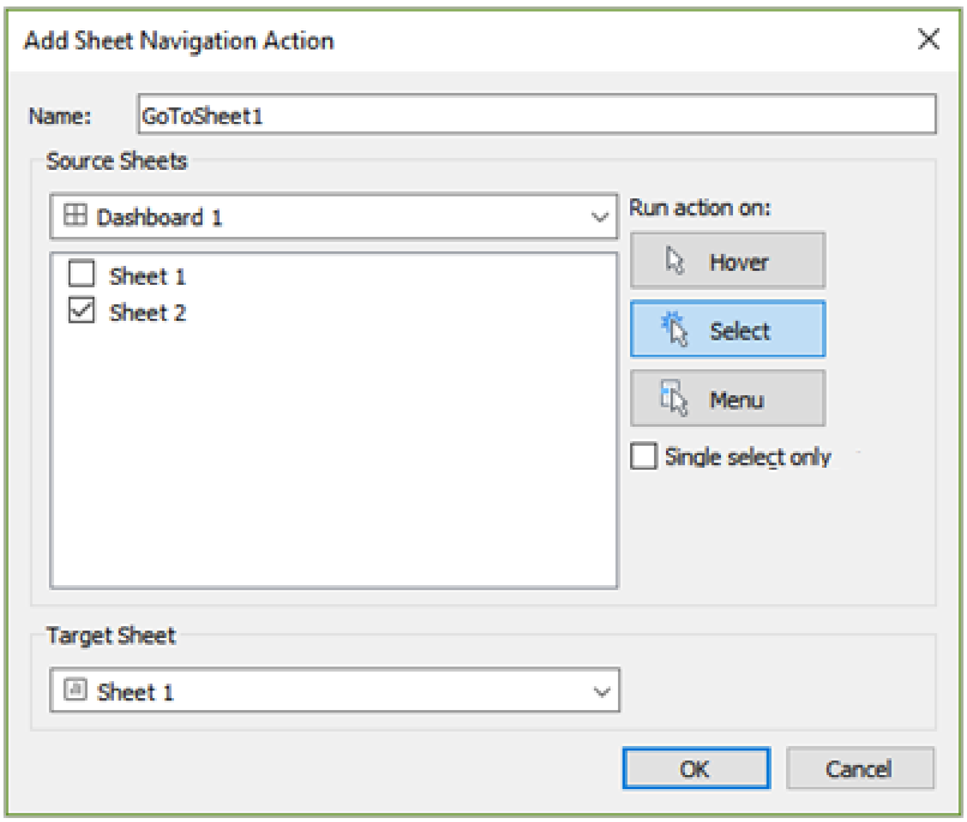
Use the Go to Sheet action to let users quickly navigate to a related visualization—a dashboard, sheet, or story—when they click on a mark or a tooltip menu item in the original view.

> **Note:** To simply navigate from one dashboard to another, without requiring users to interact with data, consider using the Navigation object.

1. From your dashboard, select **Dashboard** > **Actions**.
2. In the Actions dialog box, click **Add Action** and then select **Go to Sheet**.



3. Specify a name for the action. (If you choose to run the action using a tooltip menu, the name you specify here is what the tooltip displays.)



4. Select a source sheet that will initiate the action.

5. Specify how people viewing your dashboard will run the action. **Select** or **Menu** are the

best choices for a navigation action.

If you choose **Select**, consider selecting **Single-select only** so users won't navigate away from the view when multiple marks are selected.

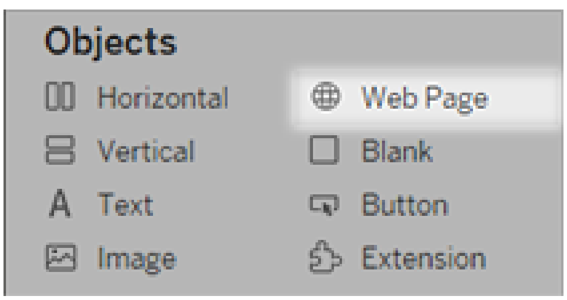
6. For the target sheet, select the navigation destination that appears when users click marks or tooltip menu items in the source sheet. Then click **OK**.

Interactively display a web page in a dashboard (Create in Tableau Desktop)

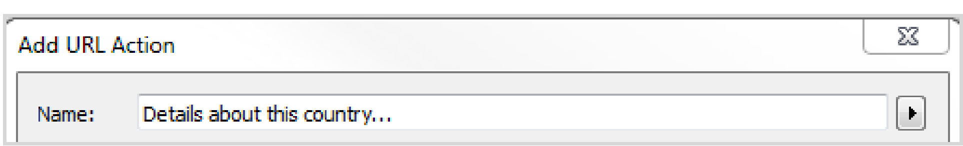
To interactively display information from the web inside a dashboard, you can use a URL action with a web page object. For example, you might have a dashboard that shows profits by country. In addition to showing the profit data in your dashboard, you also want to display supplemental information about the countries from a web site.

Tip: To easily organize and target multiple web page objects in a dashboard, rename them.

1. Drag a **Web Page** object onto your dashboard, and enter a URL.



2. From your dashboard, select **Dashboard** > **Actions**.
3. In the Actions dialog box, click **Add Action** and then select **Go to URL**.
4. Specify a name for the link. If you choose to run the action using a menu, such as a menu option on a tooltip, the name you specify here is what's displayed.



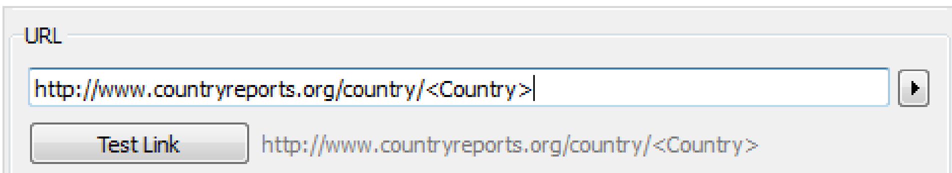
5. Under Source Sheets, select the view or data source that will initiate the action. For example, if you want the action to be initiated when a user clicks a link on a map's tooltip,

select the map view.

6. Specify whether people viewing your dashboard will run the action on hover, select, or menu. For details, see **Running Actions** on page 1565.

7. Enter the URL, starting with the http:// or https:// prefix, such as `http://www.example.com`.

   You can use field values as parameters in your URL. For example, if Country is a field used by a view in your dashboard, you can use `<Country>` as a parameter in your URL. For details, see **URL Actions** on page 1509.

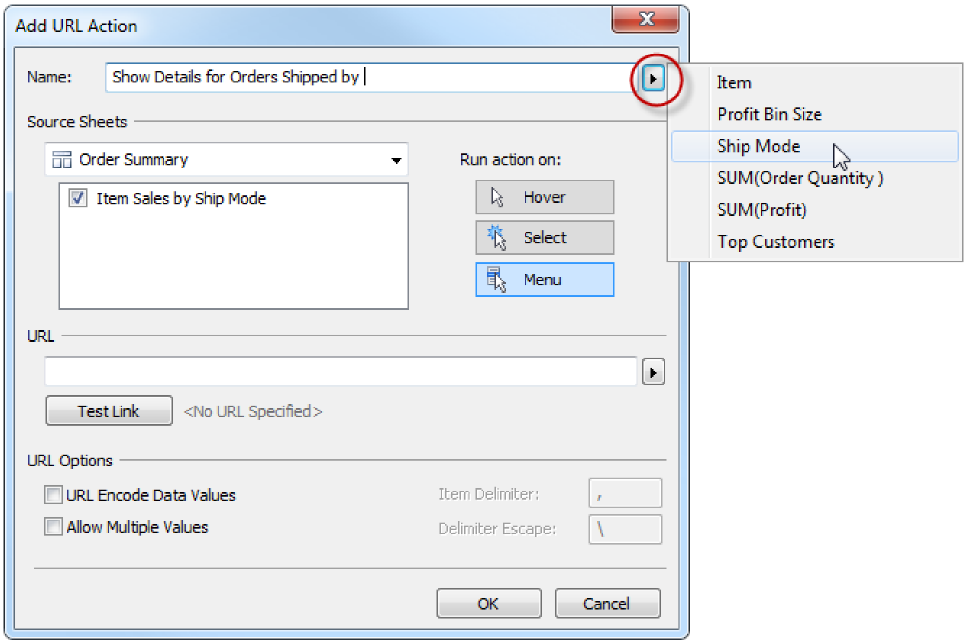
8. For URL Target, select **Web Page Object**, and select the object you created in step 1.

   When you launch the action, a web page automatically loads within the dashboard rather than opening a separate browser window.

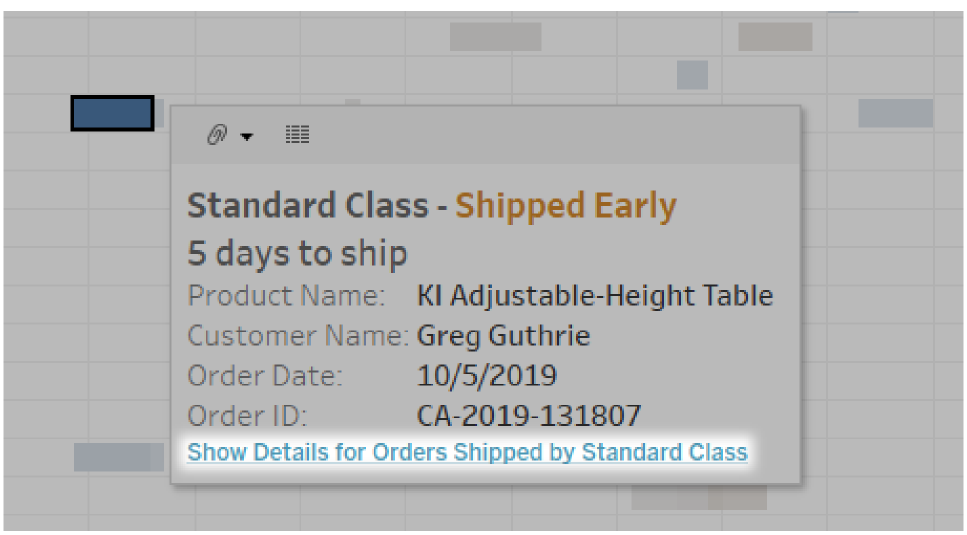
Using Field and Filter Values in Action Names

In addition to using field, parameter, and filter values in URLs, you can use this information as variables in the action names. The name of the action displays on the context menu when an action is launched using the menu. Using field and filter variables in the name is useful in making the action specific to the selected mark. In a view showing real estate information, you could name a URL action that points at satellite images from an online mapping service, “Show satellite image of <Address>.” When you right-click (control-click on Mac) a specific mark, the <Address> tag is replaced with the location value associated with that mark.

1. In the Add Action dialog box, begin typing the name for the action.
2. Place the cursor where you want to insert the field or filter value.
3. Click the arrow to the right of the text box and select the field or filter you want to add as a variable. The field or filter name is added between angle brackets.



Here's a finished example for an action run from a tooltip menu:



Build Common Chart Types in Data Views

This section includes detailed exercises that guide you through the steps involved in building some common chart types in data views. All exercises use the **Sample - Superstore** data source, which is included with Tableau Desktop. This collection of topics is just a sample of the many types of data views that you can create in Tableau. For details on options for building views from scratch, see **Build Data Views from Scratch** on page 1284 and **Build a Basic View to Explore Your Data** on page 169.

> For a 6-minute walkthrough on building data views, see the Getting Started with Visual Analytics free training video. Use your tableau.com account to sign in.
>
> For an in-depth, 25-minute walkthrough of the Tableau environment, see the Getting Started free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Create an Area Chart

An area chart is a line chart where the area between the line and the axis are shaded with a color. These charts are typically used to represent accumulated totals over time and are the conventional way to display stacked lines. Follow the steps below to create an area chart.

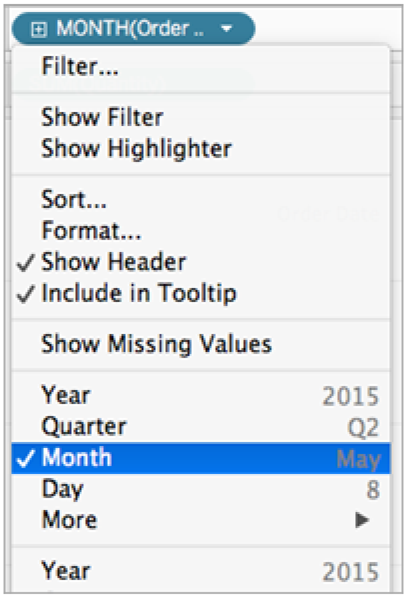
The basic building blocks for an area chart are as follows:

Mark type:	Area
Columns shelf:	Dimension
Rows shelf:	Measure
Color:	Dimension

To create an area chart, follow the steps below:

1. Open Tableau Desktop and connect to the **Sample - Superstore** data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, drag **Order Date** to the **Columns** shelf.

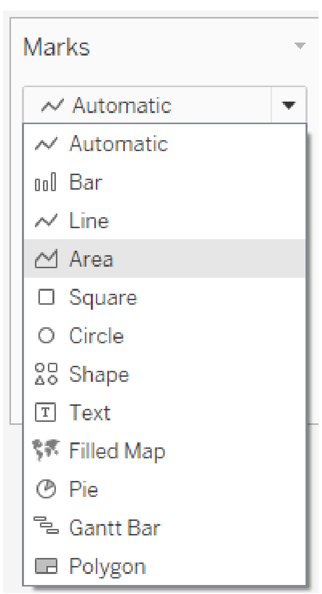
4. On the Columns shelf, right-click **YEAR(Order Date)** and select **Month**.



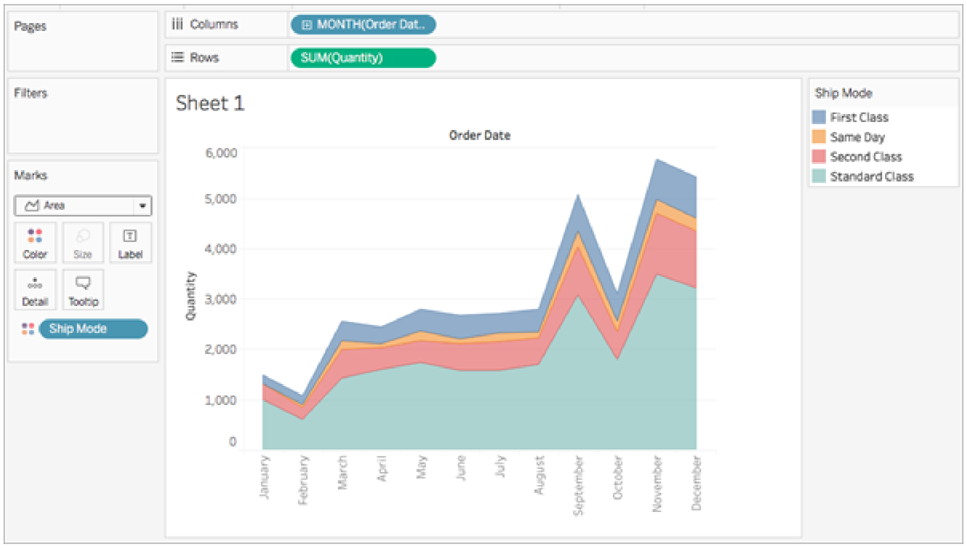
5. From the **Data** pane, drag **Quantity** to the **Rows** shelf.

6. From the **Date** pane, drag **Ship Mode** to **Color** on the Marks card.

7. On the Marks card, click the Mark Type drop-down and select **Area**.



The visualization updates to the following:



You can add formatting to an area chart. For example, you can edit the color legend and turn on mark labels and borders. For more information, see **Format at the Worksheet Level** on page 2892.

You can also use highlight actions with area charts. For example, selecting a color in the legend or turning on the highlighter will highlight the entire area instead of just the line. For more information, see **Highlight Actions** on page 1498.

Build a Bar Chart

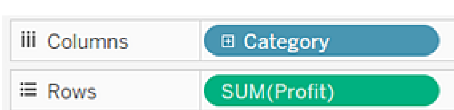
Use bar charts to compare data across categories. You create a bar chart by placing a dimension on the **Rows** shelf and a measure on the **Columns** shelf, or vice versa.

A bar chart uses the **Bar** mark type. Tableau selects this mark type when the data view matches one of the two field arrangements shown below. You can add additional fields to these shelves.

For more information about the **Bar** mark type, see **Bar mark** on page 1337.

> **Note**: At the end of the procedure is an extra step you can take to display totals at the tops of the bars.

Creates Vertical Bars



Creates Horizontal Bars

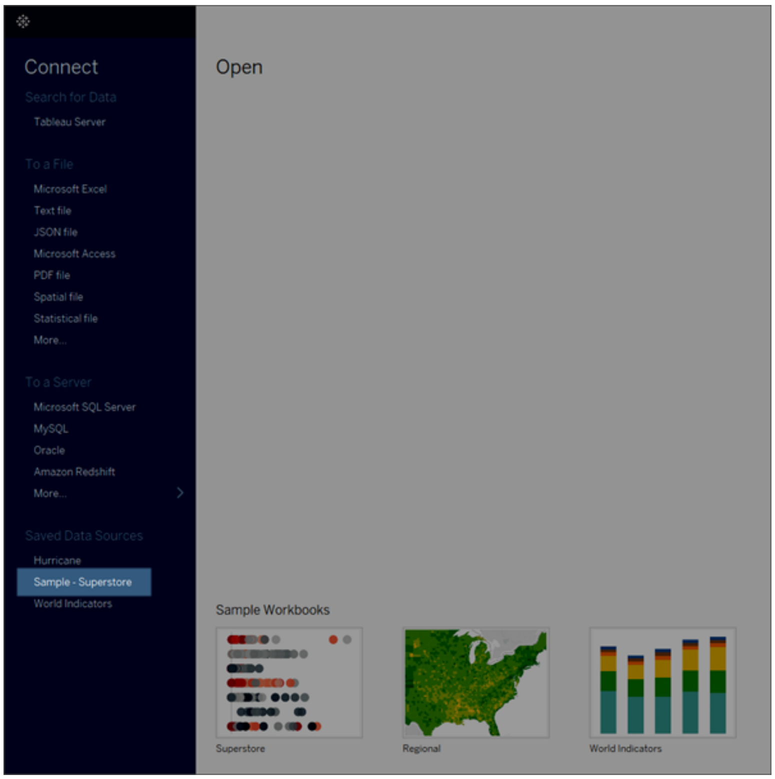
To create a bar chart that displays total sales over a four-year period, follow these steps:

How to Build a Bar Chart

1. Connect to the **Sample - Superstore** data source.

Step 1:

Connect to the **Sample - Superstore** data source.



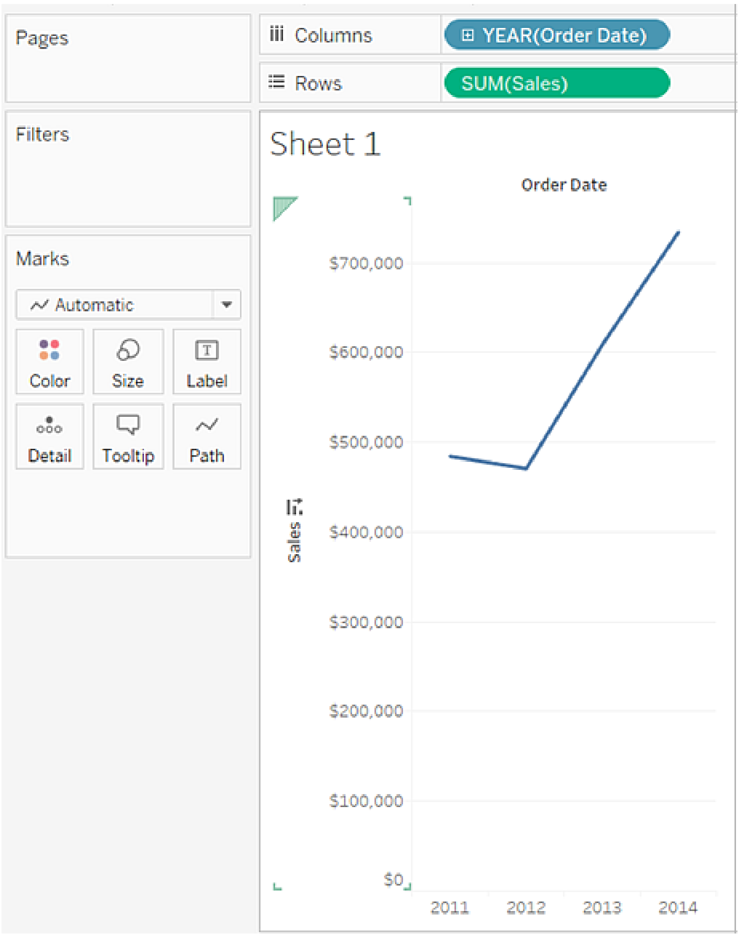
Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

2. Drag **Order Date** to **Columns** and **Sales** to **Rows**.

Step 2:

Drag the **Order Date** dimension to **Columns** and drag the **Sales** measure to **Rows**.

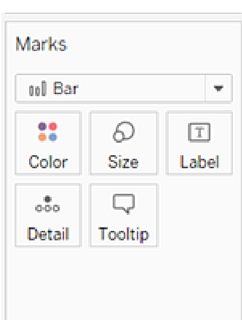
Notice that the data is aggregated by year and column headers appear. The Sales measure is aggregated as a sum and an axis is created, while the column headers move to the bottom of the view. Tableau uses **Line** as the mark type because you added the date dimension.



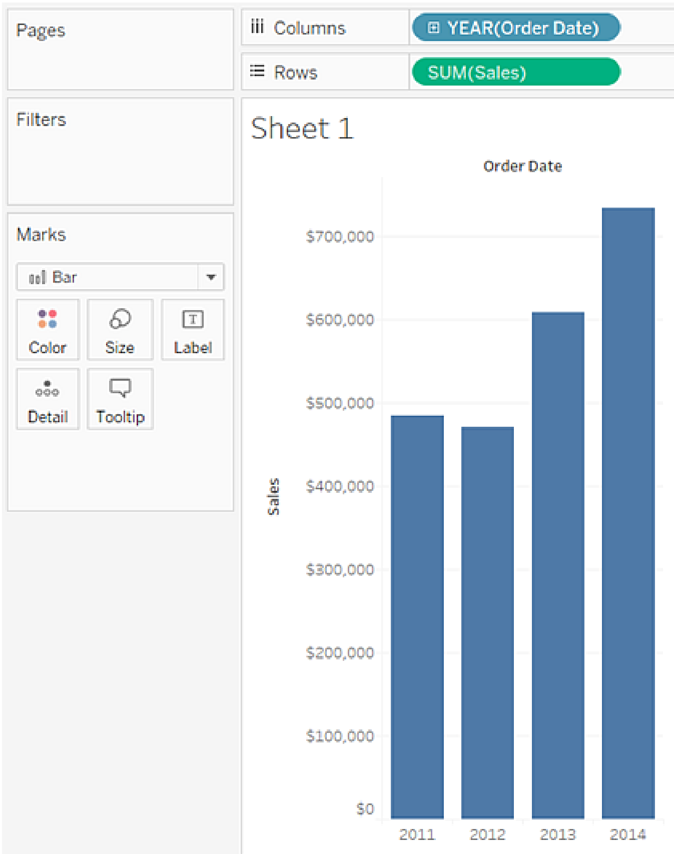
3. On the **Marks** card, select **Bar** from the drop-down list.

Step 3:

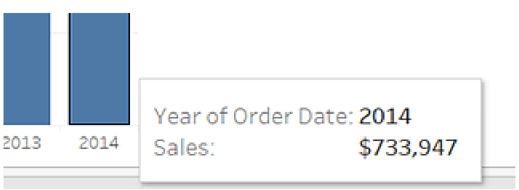
On the **Marks** card, select **Bar** from the drop-down list.



The view changes to a bar chart.



The marks (which are bars in this case) are vertical because the axis is vertical. The length of each mark represents the sum of the sales for that year. The actual numbers you see here might not match the numbers you see—the sample data changes from time to time.

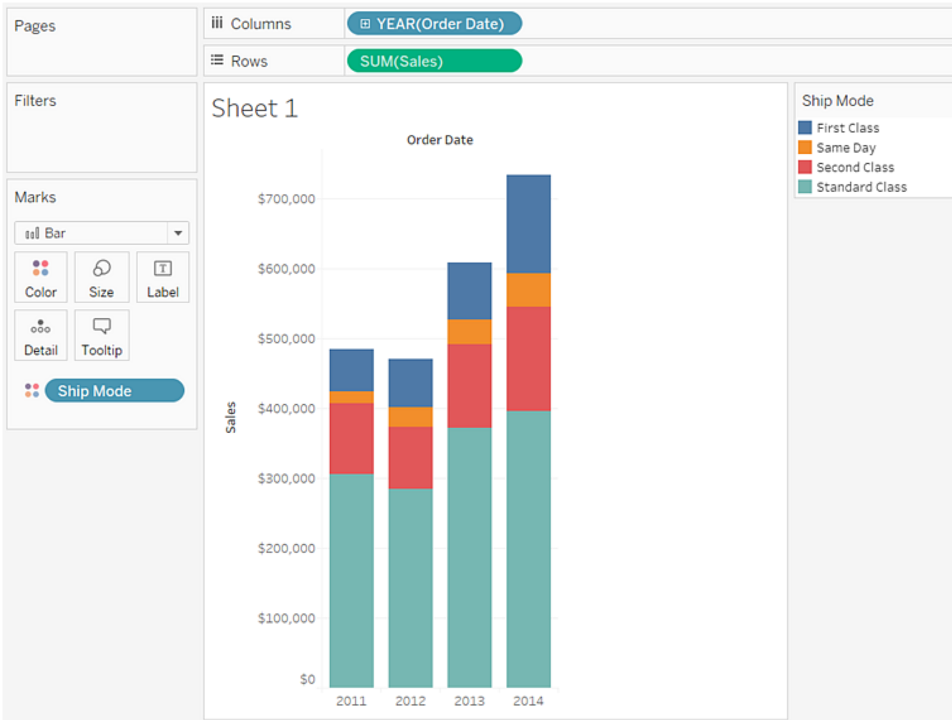


4. Drag the **Ship Mode** dimension to **Color** on the **Marks** card.

Step 4:

Drag the **Ship Mode** dimension to **Color** on the **Marks** card.

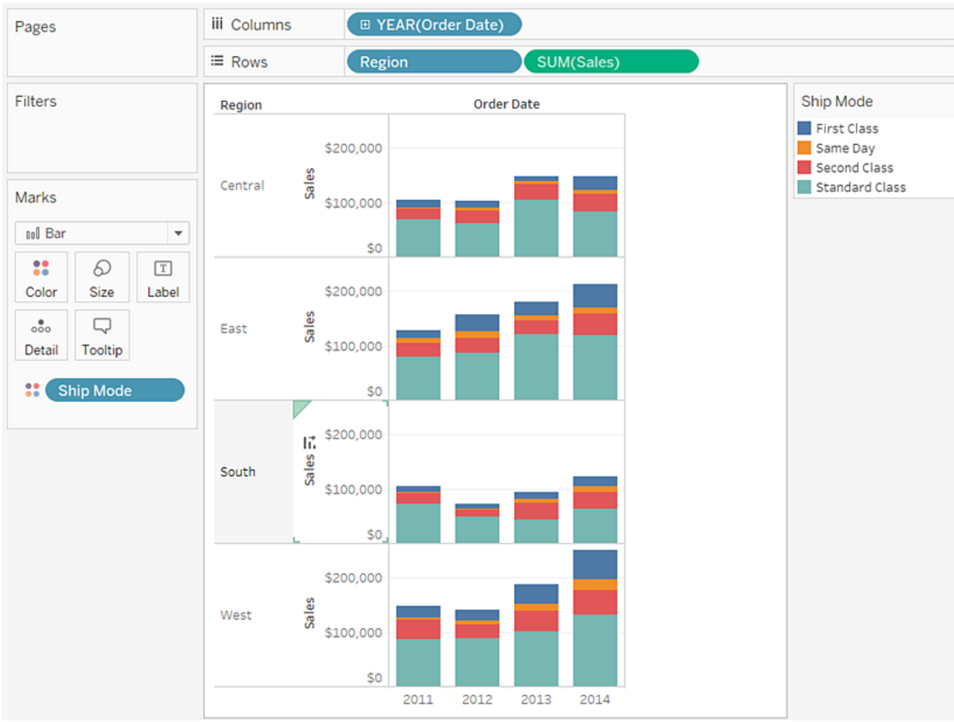
The view shows how different shipping modes have contributed to total sales over time. The ratios look consistent from year to year.



5. Drag the **Region** dimension to **Rows**, and drop it to the left of **Sales** to produce multiple axes for sales by region.

Step 5:

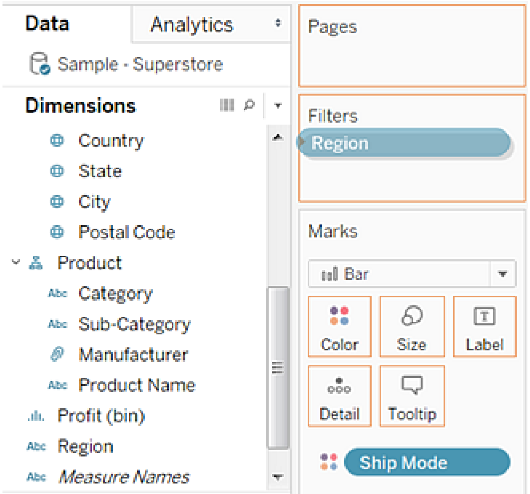
Drag the **Region** dimension to **Rows**, and drop it to the left of **Sales** to produce multiple axes for sales by region.



6. To view data in the West region only, you can filter out the other regions. To do this, drag the **Region** dimension again, this time from the **Data** pane to the **Filters** shelf.

Step 6:

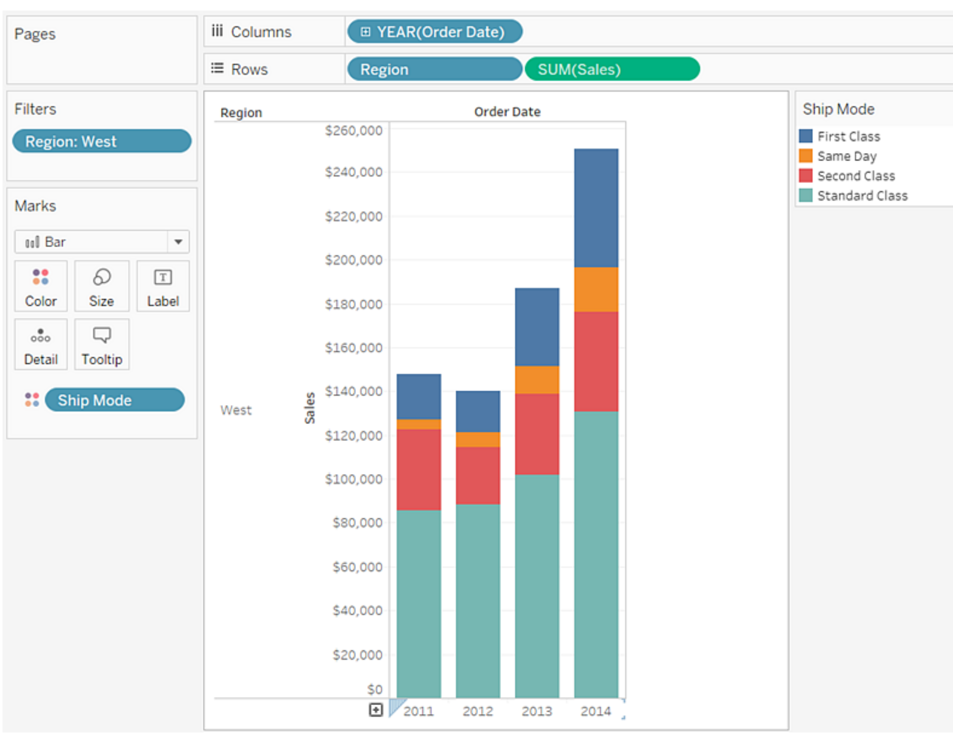
To view data in the West region only, you can filter out the other regions. To do this, drag the **Region** dimension again, this time from the **Data** pane to the **Filters** shelf.



7. In the Filter [Region] dialog box, clear the **Central**, **East**, and **South** check boxes, and then click **OK**.

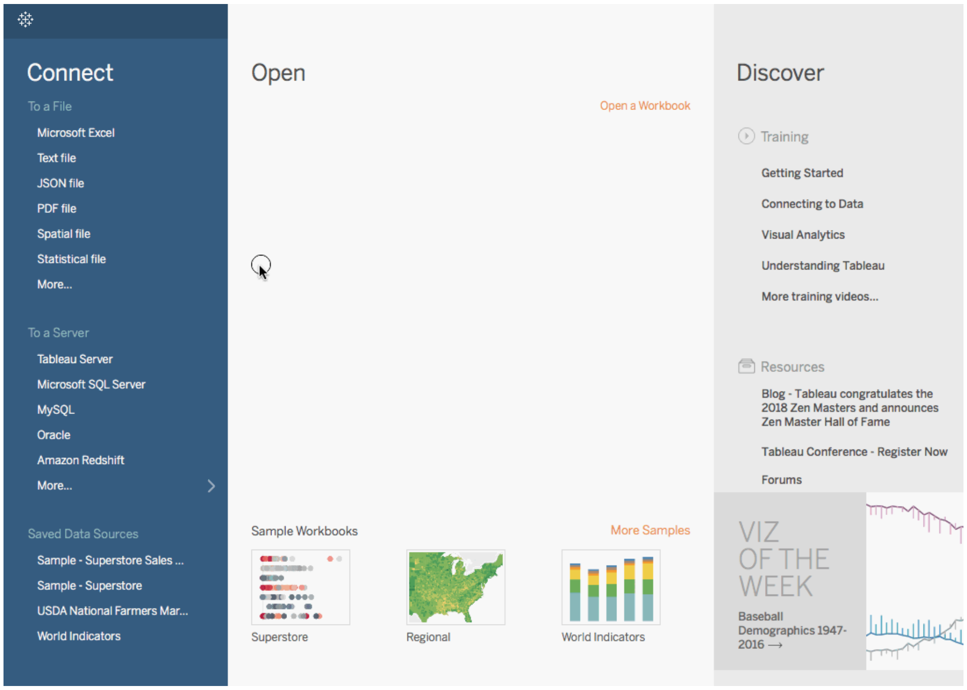
Step 7:

In the Filter [Region] dialog box, clear the **Central**, **East**, and **South** check boxes, and then click **OK**.



This view gives you insight into your data—for example, how the ship mode changed in the West over the four-year period.

Check your work! Watch steps 1-10 below.

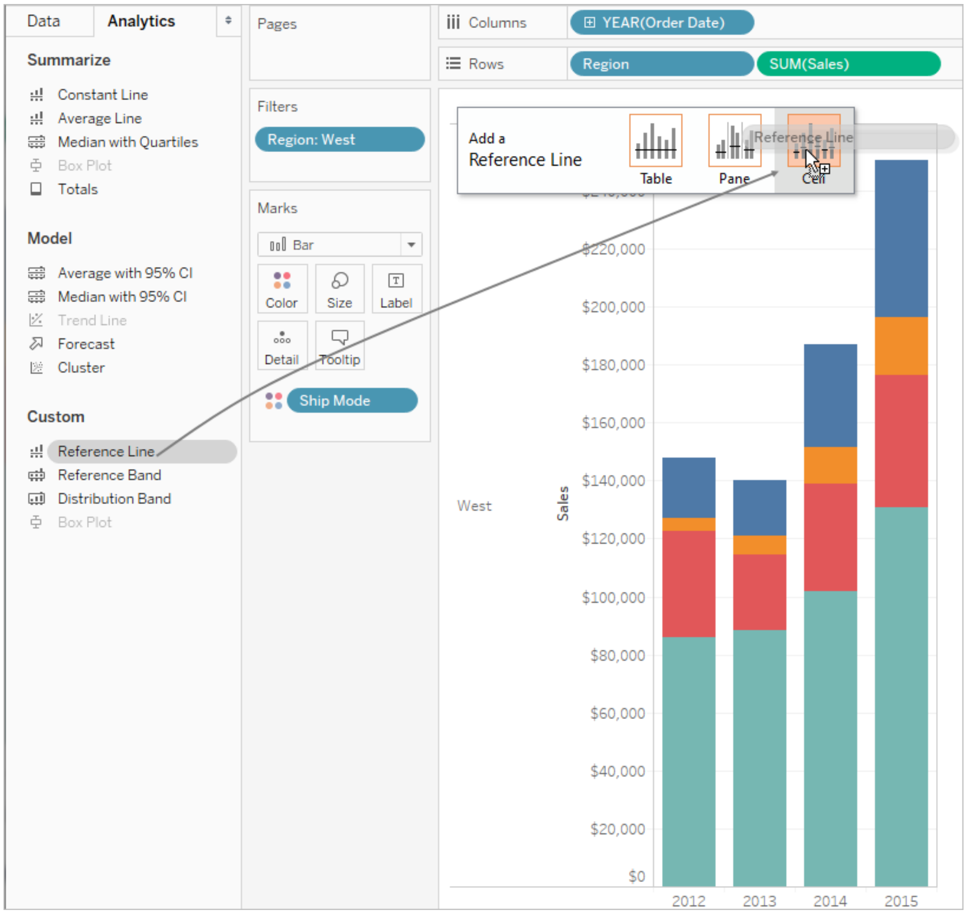


Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

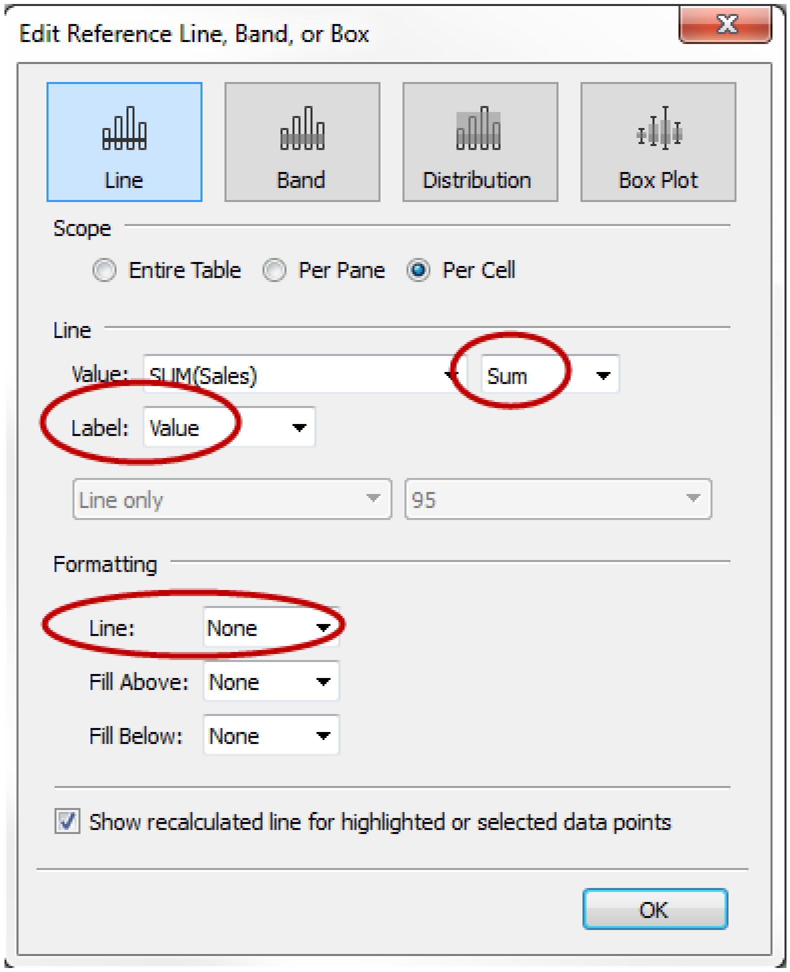
One Step Further: Add Totals To Stacked Bars

Adding totals to the tops of bars in a chart is sometimes as simple as clicking the **Show Mark Labels** icon in the toolbar. But when the bars are broken down by color or size, each individual segment would labeled, rather than the total for the bar. With a few steps, you can add a total label at the top of every bar even when the bars are subdivided as in the view you just created. In the following procedure you will technically be adding a reference line. But by configuring that "line" in a certain way, you end up with the labels you want.

1. From the **Analytics** pane, drag a **Reference Line** into the view and drop it on **Cell**.

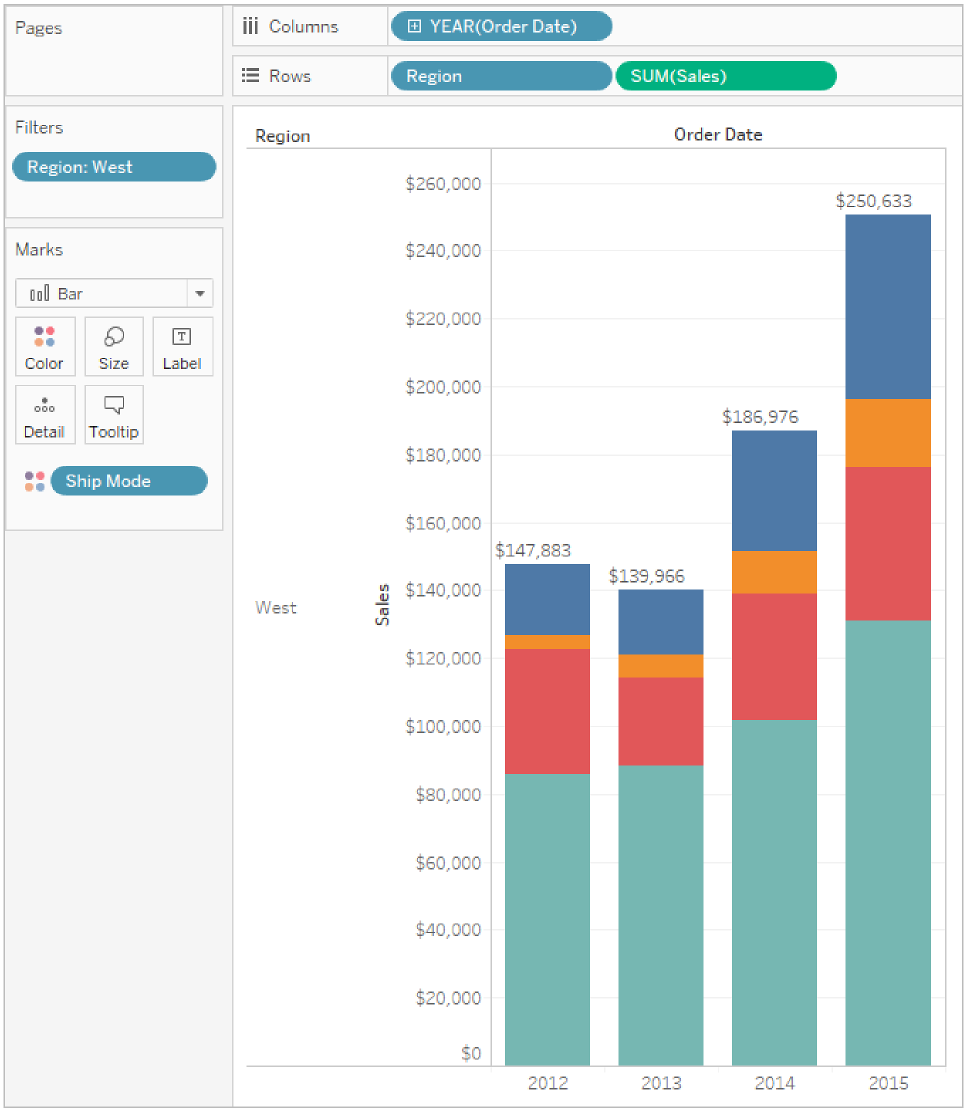


2. In the Edit Line, Band, or Box dialog box, set the aggregation for **SUM(Sales)** to **Sum**, set **Label** to **Value**, and set **Line** under Formatting to **None**:



Then click **OK** to close the Edit Reference Line, Band, or Box dialog box.

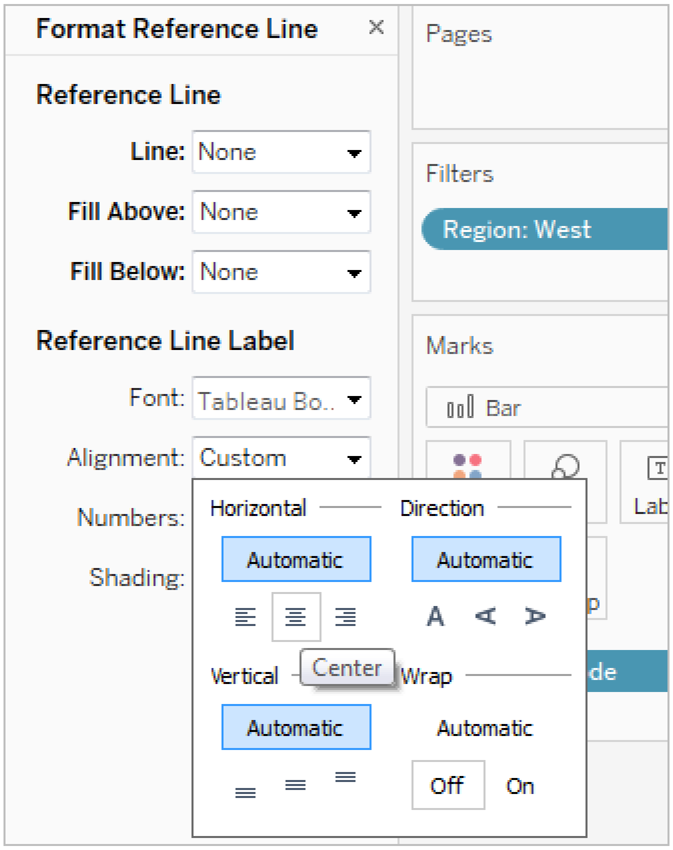
Your view now has currency totals at the top of each bar:



You may need to adjust the view to make it look just right. If the bars are too narrow, the numbers are truncated; to fix this, press Ctrl + Right on the keyboard to make the bars wider. Or if you want to center the totals over the bars—by default, they are left-aligned. Do the following:

1. Right-click any of the totals on the bar chart and select **Format**.
2. In the Format window, in the **Reference Line Label** area, open the **Alignment**

control and select the Center option for Horizontal alignment:



Other resources

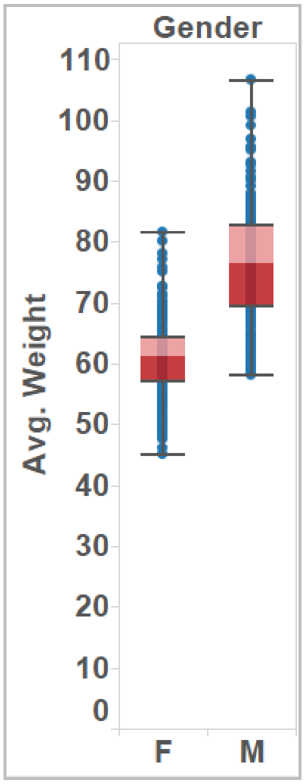
Creation of a Grouped Bar Chart A knowledge base article with embedded videos that describes how to create grouped bar charts, also known as side-by-side bar charts.

Build a Box Plot

Use box plots, also known as box-and-whisker plots, to show the distribution of values along an axis.

Boxes indicate the middle 50 percent of the data (that is, the middle two quartiles of the data's distribution).

You can configure lines, called *whiskers*, to display all points within 1.5 times the interquartile range (in other words, all points within 1.5 times the width of the adjoining box), or all points at the maximum extent of the data, as shown in the following image:



The basic building blocks for a box and whisker plot are as follows:

Mark type:	Circle
Columns shelf:	Dimension
Rows shelf:	Measure
Detail:	Dimension
Reference Line:	Box Plot For information on how to add a reference line, see **Reference Lines, Bands, Distributions, and Boxes** on page 2502.

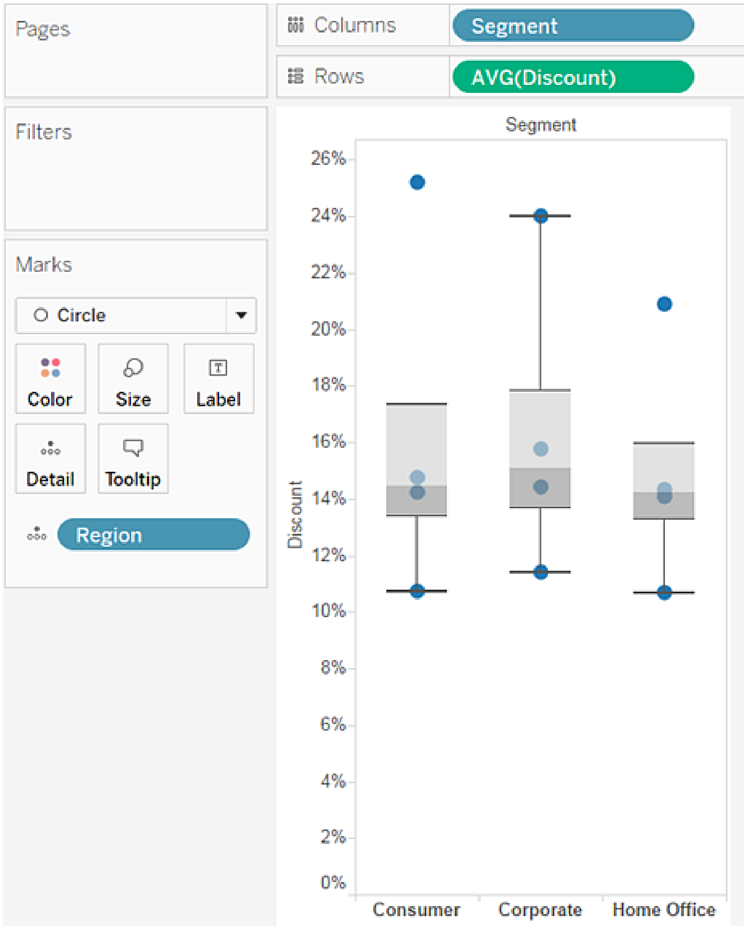
To create a box plot that shows discounts by region and customer segment, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Segment** dimension to **Columns**.
3. Drag the **Discount** measure to **Rows**.

   Tableau creates a vertical axis and displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.
4. Drag the **Region** dimension to **Columns**, and drop it to the right of **Segment**.

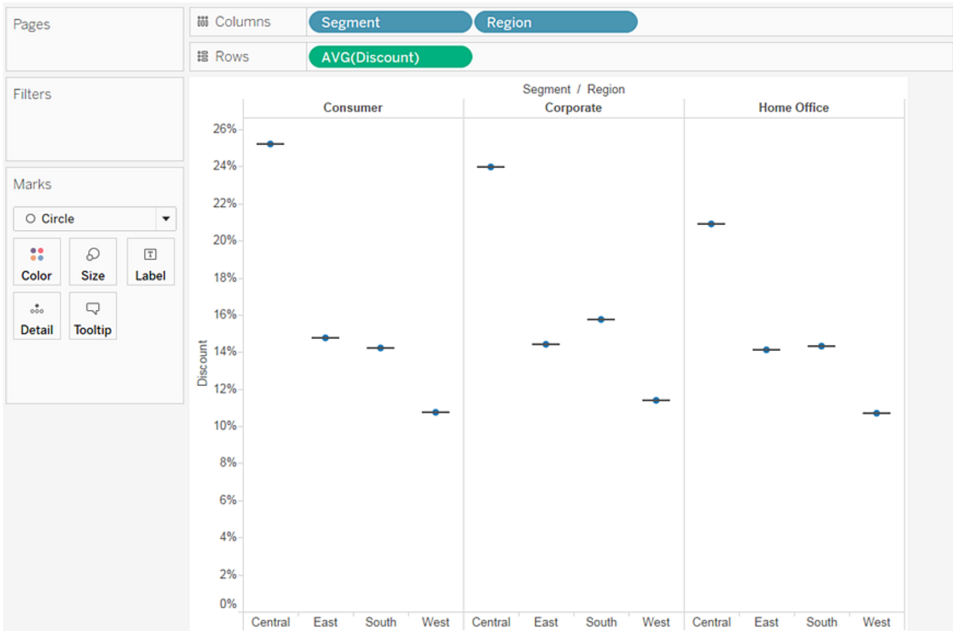
   Now you have a two-level hierarchy of dimensions from left to right in the view, with regions (listed along the bottom) nested within segments (listed across the top).
5. Click **Show Me** in the toolbar, then select the box-and-whisker plot chart type.

   Tableau displays the a box plot:



Notice that there are only a few marks in each box plot. Also, Tableau reassigned **Region** from the **Columns** shelf to the **Marks** card. When you changed the chart type to a box plot, Tableau determined what the individual marks in the plot should represent. It determined that the marks should represent regions. We'll change that.

6. Drag **Region** from the **Marks** card back to **Columns**, to the right of **Segment**.



The horizontal lines are flattened box plots, which is what happens when box plots are based on a single mark.

Box plots are intended to show a distribution of data, and that can be difficult when data is aggregated, as in the current view.

7. To disaggregate data, select **Analysis** > **Aggregate Measures**.

   This command turns aggregation on or off, and because data is aggregated by default in Tableau, the first time you select this command, it disaggregates the data.

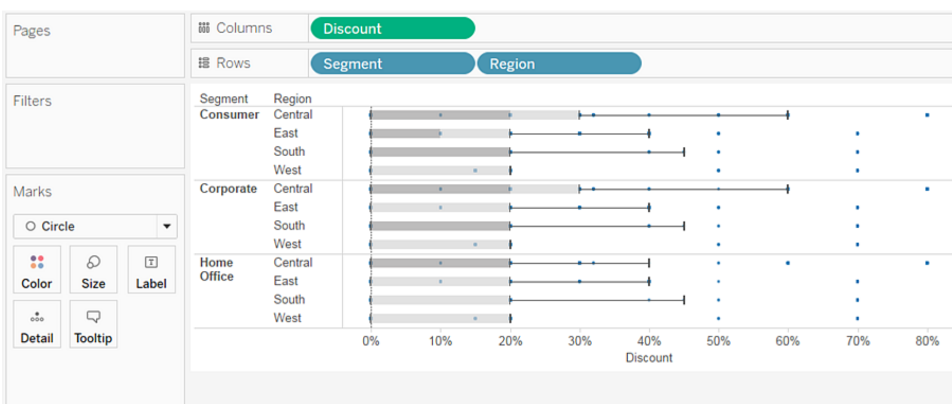
   For more information, see **How to Disaggregate Data** on page 350.

   Now, instead of a single mark for each column in the view, you see a range of marks, one for each row in your data source.

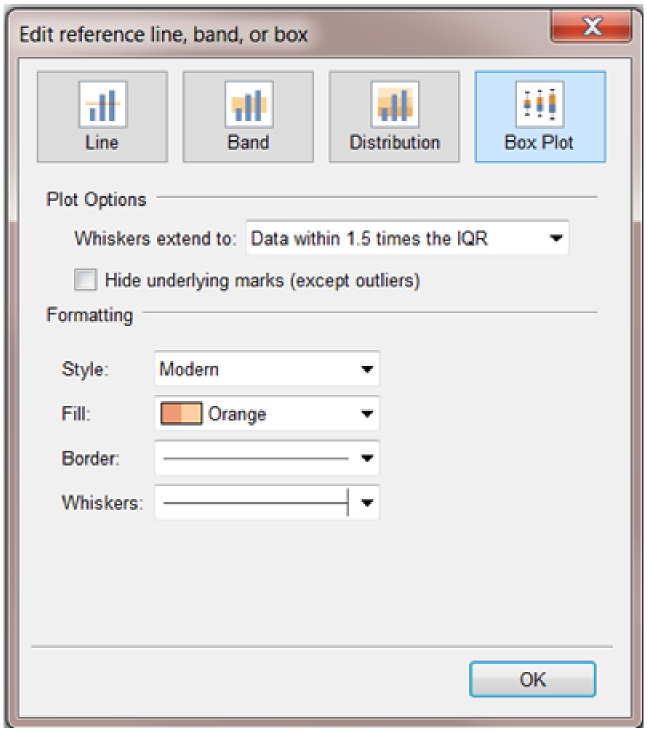
The view now shows the information we want to see. The remaining steps make the view more readable and appealing.

8. Click the **Swap**  button to swap the axes:

   The box plots now flow from left-to-right:

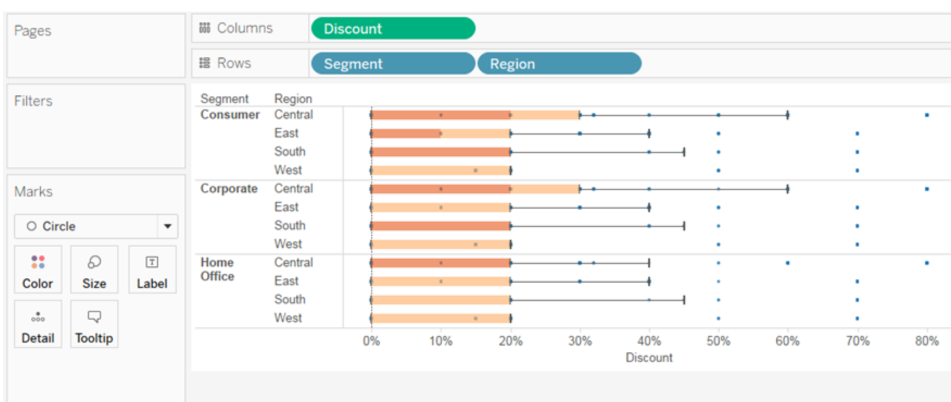


9. Right-click (control-click on Mac) the bottom axis and select **Edit Reference Line**.
10. In Edit Reference Line, Band, or Box dialog box, in the **Fill** drop-down list, select an interesting color scheme.



For more on these options, see **Add a Box Plot** on page 2523 in the **Reference Lines, Bands, Distributions, and Boxes** article.

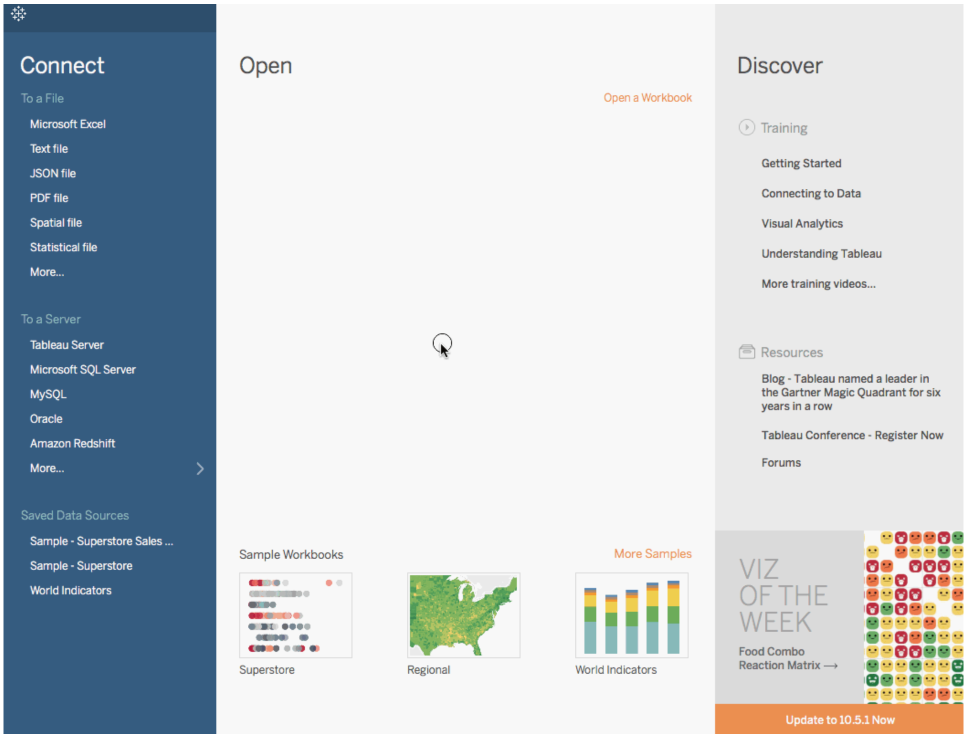
Now your view is complete:



You can see that the discount was the same for all segments in the West. You can also see that the interquartile range (from the 25th percentile to the 75th percentile) for discount was greatest in the Central region for the consumer and corporate segments.

For more information about box plots, see **Reference Lines, Bands, Distributions, and Boxes** on page 2502.

Check your work! Watch steps 1-10 below:



Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Build a Bullet Graph

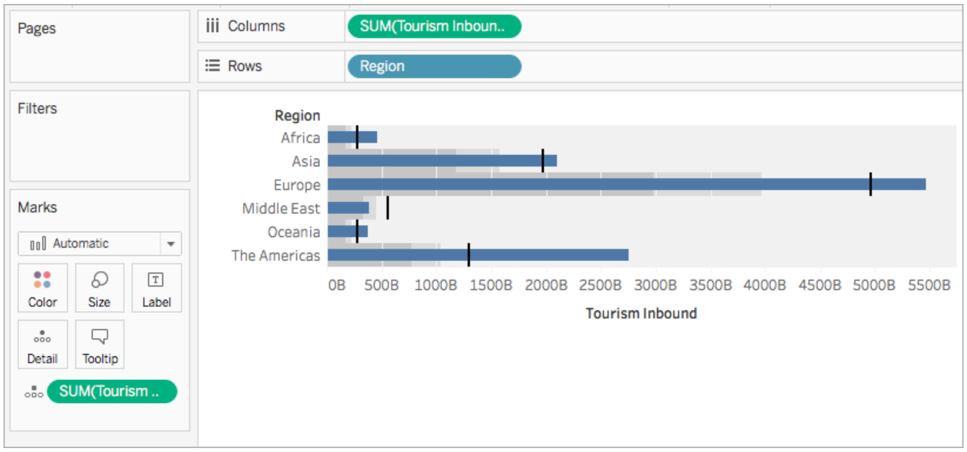
A bullet graph is a variation of a bar graph developed to replace dashboard gauges and meters. A bullet graph is useful for comparing the performance of a primary measure to one or more other measures. Below is a single bullet graph showing how actual sales compared to estimated sales.

Follow the steps below to learn how to create a bullet graph.

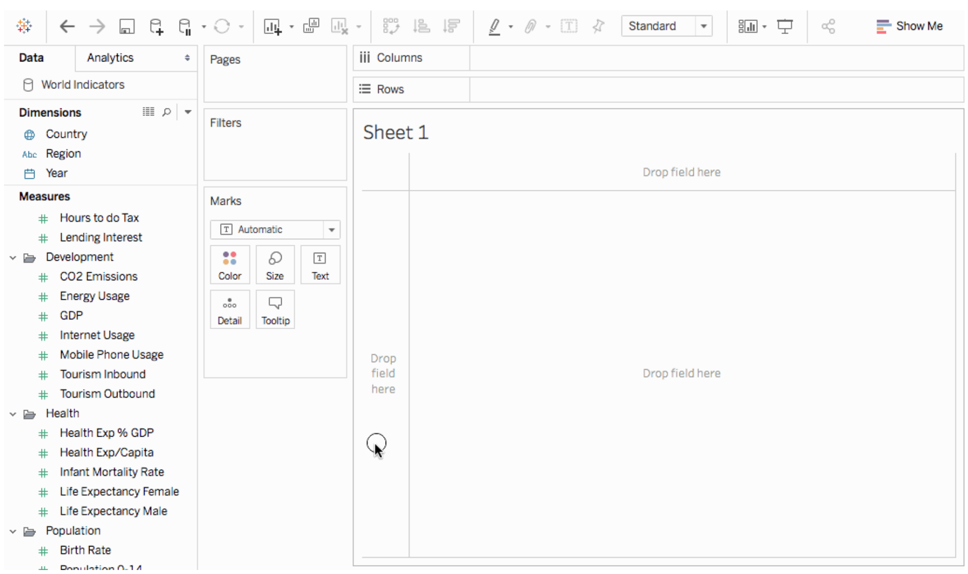
1. Open Tableau Desktop and connect to the **World Indicators** data source.
2. Navigate to a new worksheet.
3. Hold down Shift on your keyboard and then, on the **Data** pane, under **Development** select **Tourism Inbound** and **Tourism Outbound**.
4. In the upper-right corner of the application, click **Show Me**.
5. In Show Me, select the **Bullet Graph** image.

6. Click **Show Me** again to close it.
7. From the **Data** pane, drag **Region** to the **Rows** shelf.

   The graph updates to look like the following:



Check your work! Watch steps 3 - 7 below:

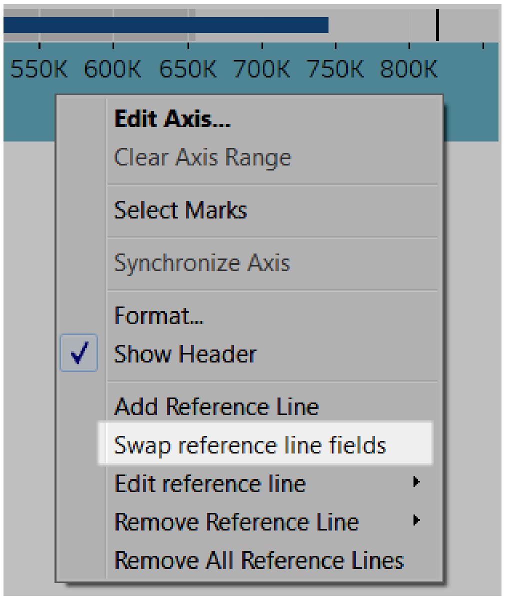


Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Swap reference line fields

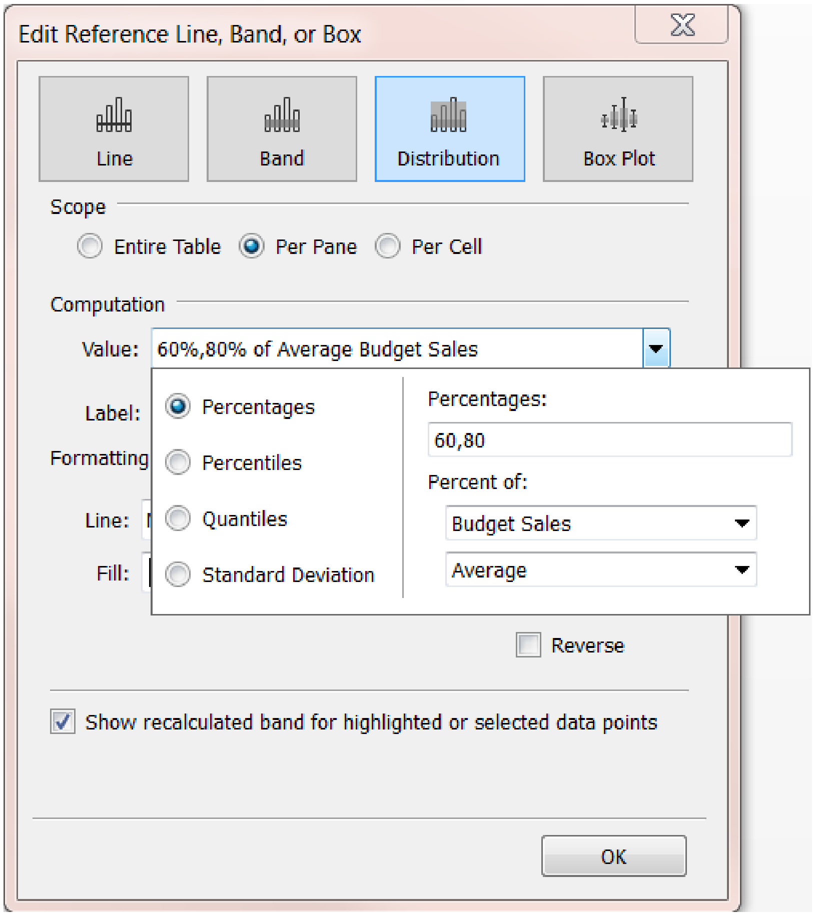
Sometimes you might want to swap the reference lines fields. For example, the actual sales is shown as a reference distribution instead of a bar.

To swap the two measures, right-click (control-click on the Mac) the axis and select **Swap Reference Line Fields**.



Edit the distribution

Right-click (control-click on the Mac) the axis in the view and select **Edit Reference Line**, and then select one of the reference lines to modify.



Build with Density Marks (Heatmap)

Use density chart to visualize patterns or trends in dense data with many overlapping marks. Tableau does this by grouping overlaying marks, and color-coding them based on the number of marks in the group.

Density maps help you identify locations with greater or fewer numbers of data points.

In Tableau, you can create a chart using the density mark by placing at least one continuous measure on the Columns shelf, and at least one dimension or measure on the Rows shelf (or vice versa), and then adding a field to the Marks card.

> **Note**: Density charts work best when used with data sources containing many data points.

The basic building blocks for a density chart are as follows:

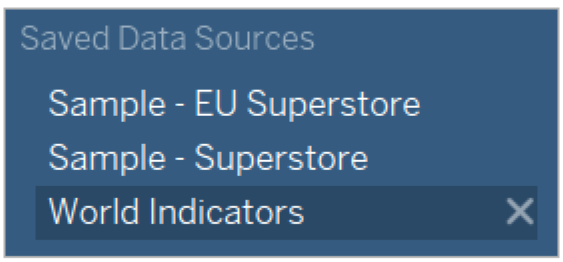
Mark type:	Density
Rows and Columns:	At least one continuous measure, and at least one measure or dimension
Marks card:	At least one continuous measure

Density charts use the **Density** mark type. By default, Tableau will use the automatic mark type.

To show how density charts can help make sense of overlapping marks in Tableau, we're going to start with a scatter plot with a large number of marks and re-create it as a density chart.

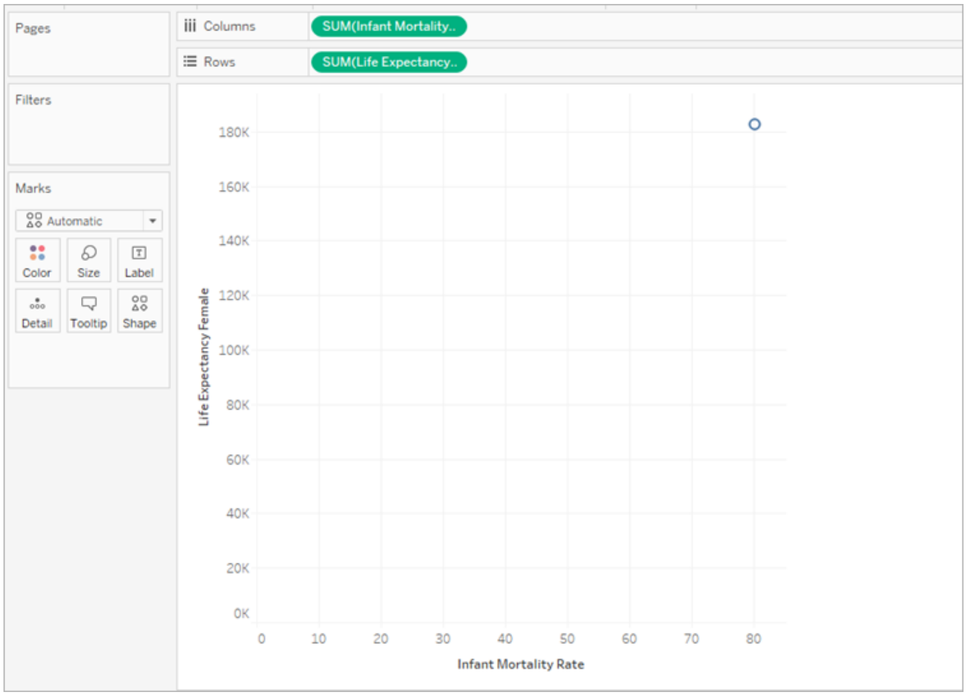
To use a density chart to see orders by date, follow these steps:

1. Open the **World Indicators** data source from the **Saved Data Sources** section of the Start screen.

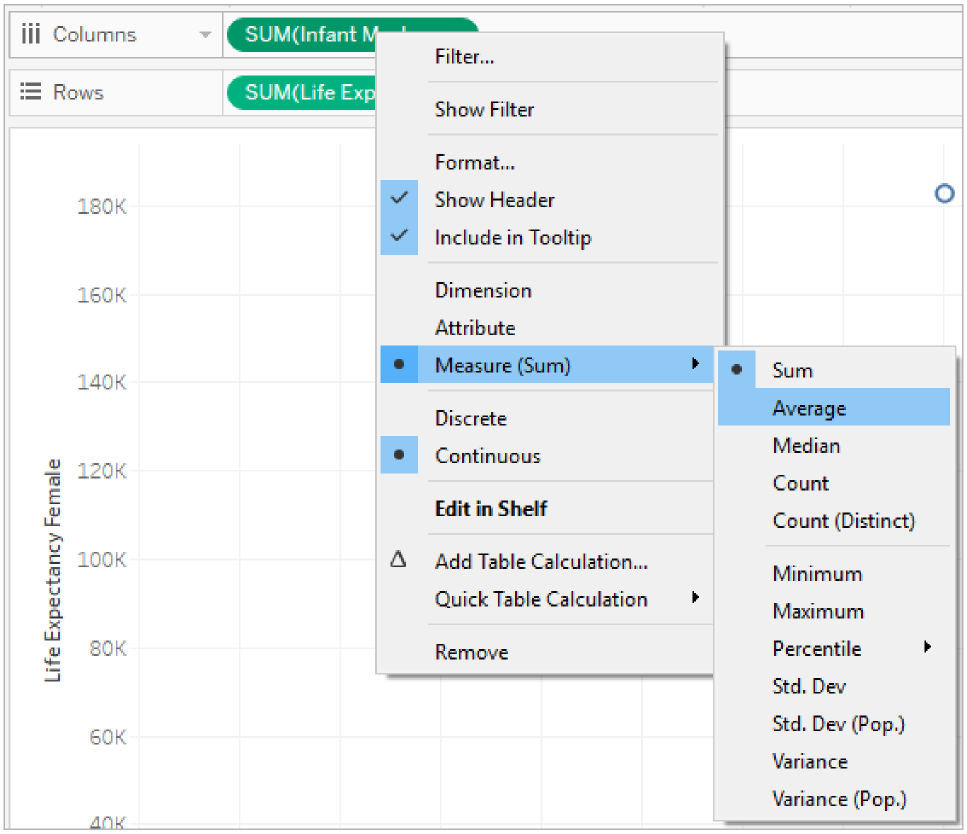
   

2. From the Health folder, drag **Infant Mortality** to the Columns shelf. Tableau aggregates the measure as a sum and creates a horizontal axis.
3. Drag the **Life Expectancy Female** to the **Rows** shelf.

   Now you have a one-mark scatter plot.

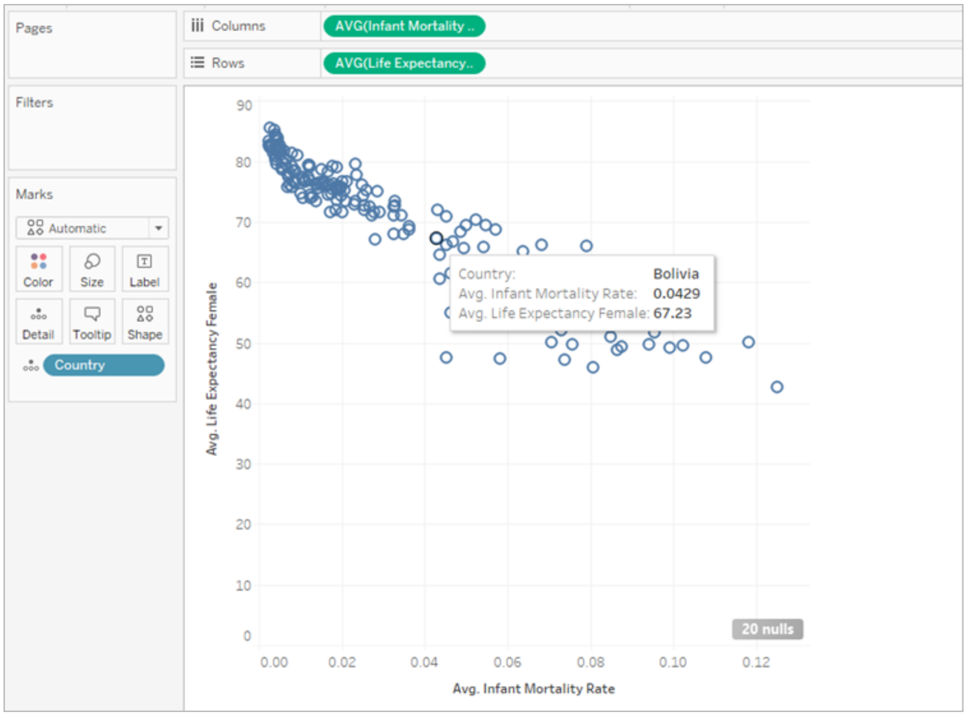


4. Both Infant Mortality and Life Expectancy are listed as a **Sum**, rather than average. Right click on both of these measures and to change Measure(Sum) to **Average**.



5. Drag the **Country** dimension to **Details** on the Marks card.

   Now there are many more marks in your view. The number of marks in your view is now equal to the number of distinct countries in this data set. If you hover over a mark, you can see the country name, female life expectancy, and infant mortality rate.



We've created a basic scatter plot, but there are lots of overlapping marks in the view and it's hard to see where the marks are most dense.

6. On the **Marks** card, select **Density** from the menu to change this scatter plot into a density chart.

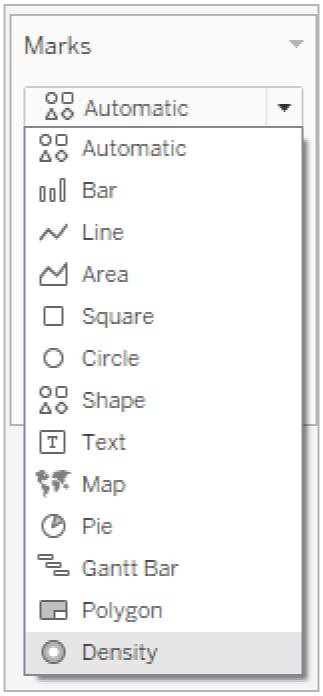
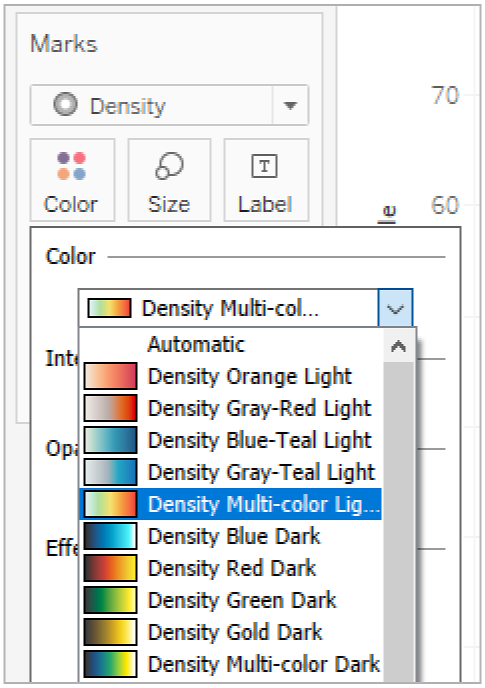


Tableau created a density chart by overlaying marks, called kernels, and color-coding where those kernels overlap. The more overlapping data points, the more intense the color is.

Tableau selected a blue color palette by default, but you can choose from ten density color palettes or any of the existing color palettes.

7. Select **Color** from the **Marks** card and select **Density Multi-color Light** from the menu.

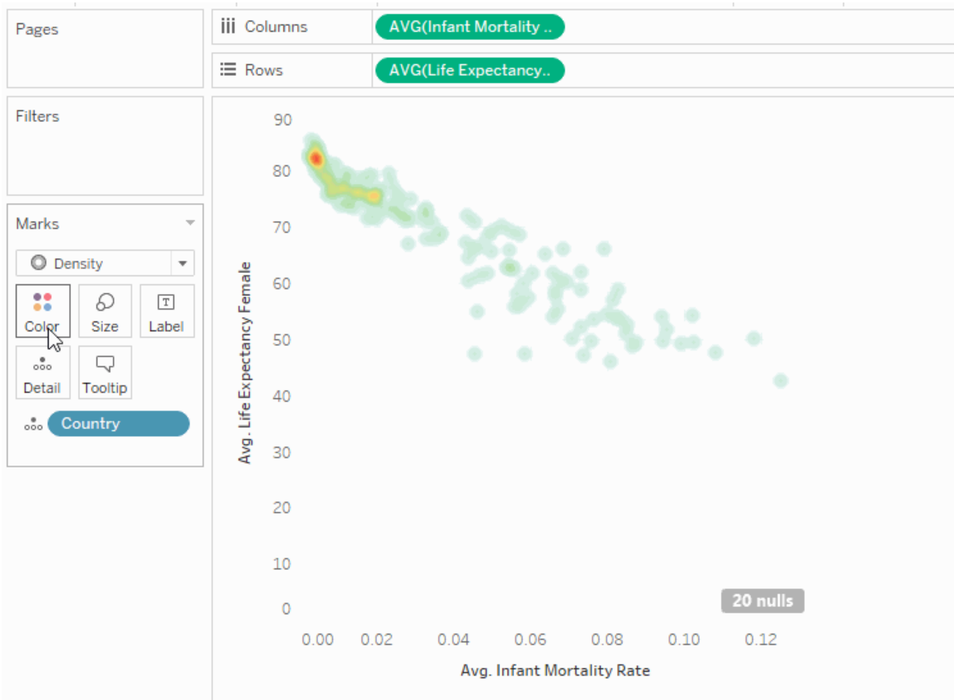


The names of the color palettes indicate whether they're designed for use on charts with dark or light backgrounds. Since our chart has a light background, we picked a "Light" palette.

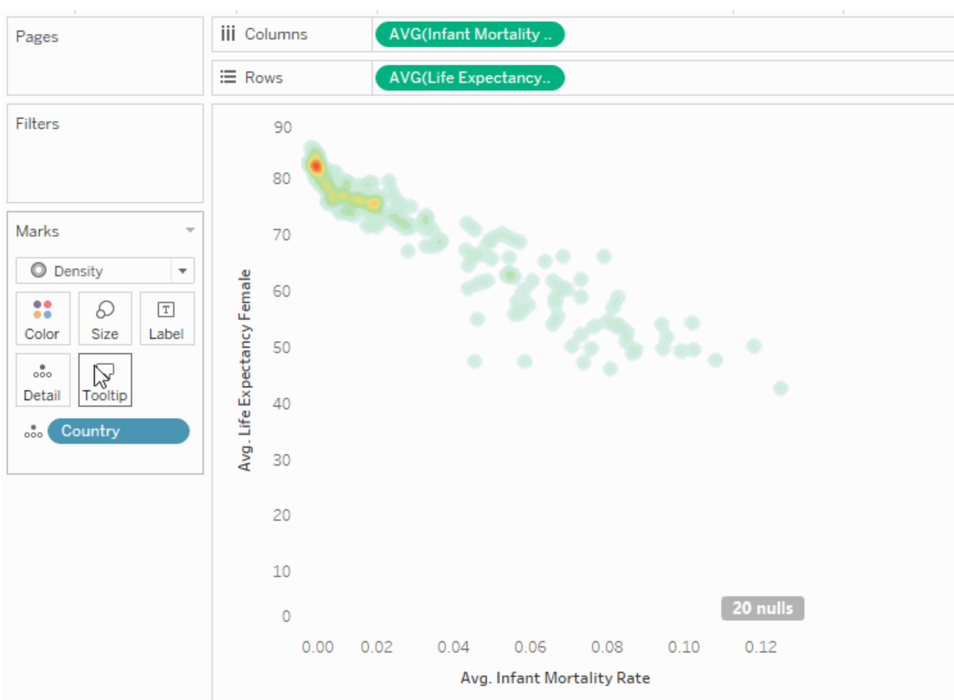
This changes the color palette on your chart. More concentrated areas will appear red, while areas without overlapping marks will appear green.

Note: Color legends are not available for density marks.

8. In the Color menu, use the Intensity slider to increase or decrease the vividness of the density marks. For example, increasing intensity, or vividness, lowers the "max heat" spots in your data, so that more appear.



9. Select **Size** from the Marks card to adjust the size of the density's kernel.



For more information, see **Change the Type of Mark in the View** on page 1332 and **Create Heatmaps that Show Trends or Density in Tableau** on page 1910.

Build a Gantt Chart

Use Gantt charts to show the duration of events or activities.

In a Gantt chart, each separate mark (usually a bar) shows a duration. For example, you might use a Gantt chart to display average delivery time for a range of products.

The basic building blocks for a gantt chart are as follows:

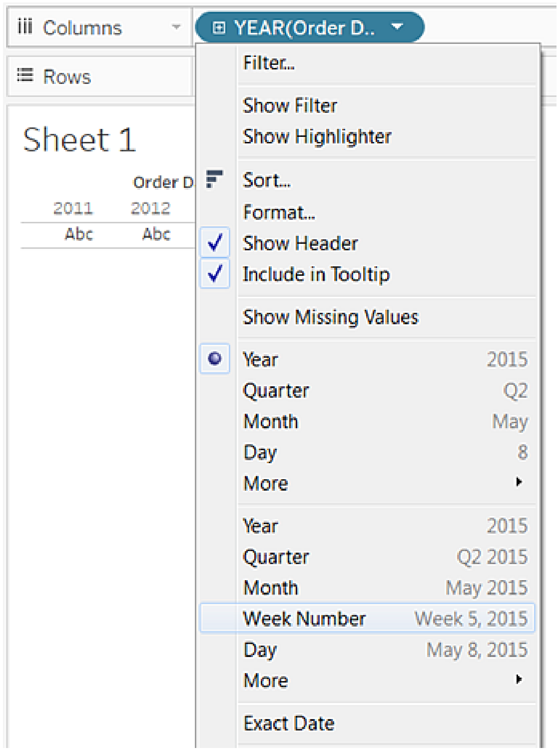
Mark type:	Automatic or Gantt Bar
Columns shelf:	Date or Time field (continuous measure)
Rows shelf:	Dimension(s)
Size:	Continuous measure

For more information about the Gantt bar mark type, see **Gantt Bar mark** on page 1351.

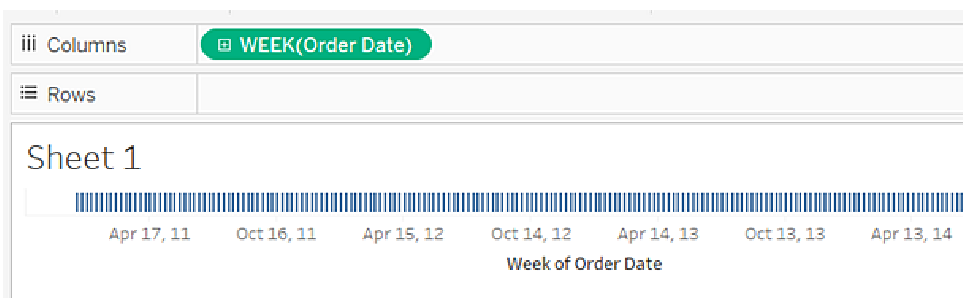
To create a Gantt chart that shows how many days elapse on average between order date and ship date, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Order Date** dimension to **Columns**.

   Tableau aggregates the dates by year and creates column headers with labels for the years.
3. On the **Columns** shelf, click the **Year (Order Date)** drop-down arrow, and then select **Week Number**.



The column headers change. Individual weeks are indicated by tick marks because there are 208 weeks in a four-year span—too many to show as labels in the view.

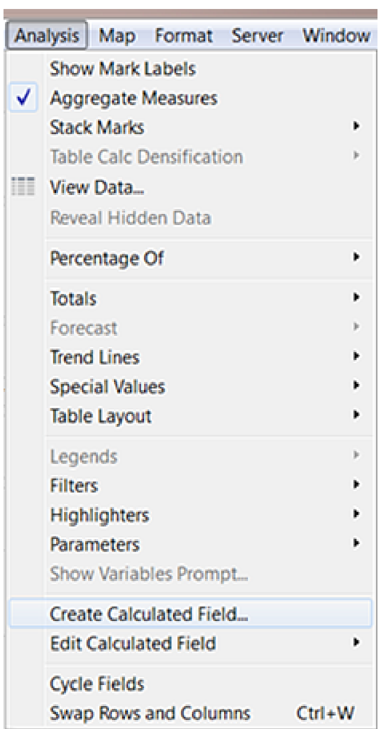


4. Drag the **Sub-Category** and **Ship Mode** dimensions to the **Rows** shelf. Drop **Ship Mode** to the right of **Sub-Category**.

   This builds a two-level nested hierarchy of dimensions along the left axis.

Next, we'll size the marks according to the length of the interval between the order date and the ship date. To do this, create a calculated field to capture that interval.

5. In the toolbar menu, click **Analysis > Create Calculated Field**. You can also right-click (Control-click on Mac) any field in the **Data** pane and select **Create** > **Calculated Field**.



6. In the calculation dialog box, name your calculated field **OrderUntilShip**.

7. Clear any content that's in the **Formula** box by default.

8. In the **Formula** box, enter the following formula and then click **OK**:

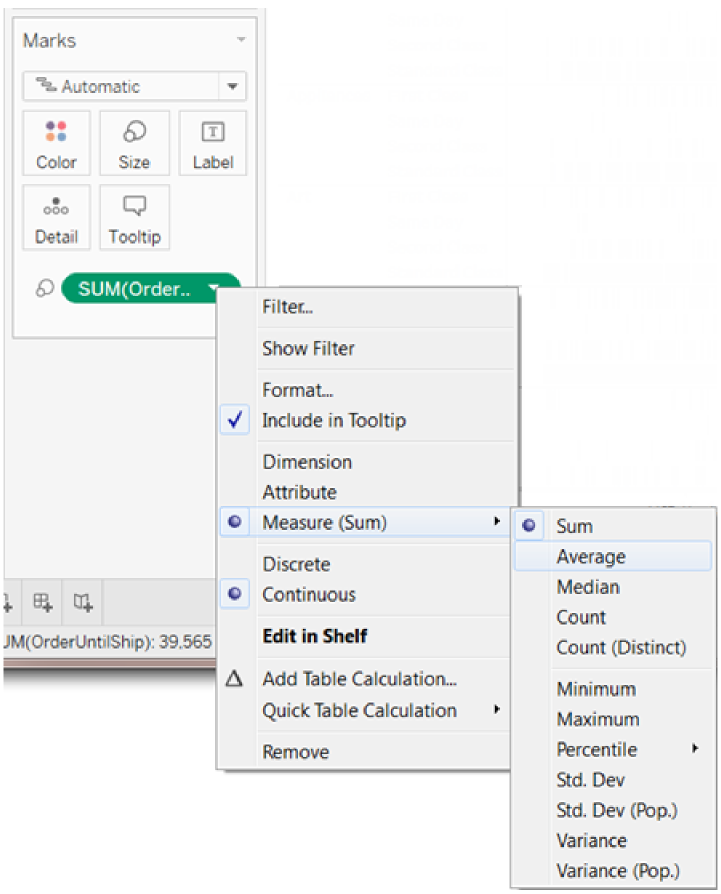
   ```
   DATEDIFF('day',[Order Date],[Ship Date])
   ```

   The formula creates a custom measure that captures the difference between the **Order Date** and **Ship Date** values, in days.

9. Drag the **OrderUntilShip** measure to **Size** on the **Marks** card.

   The default aggregation for **OrderUntilShip** is **Sum**, but in this case it makes more sense to average the values.

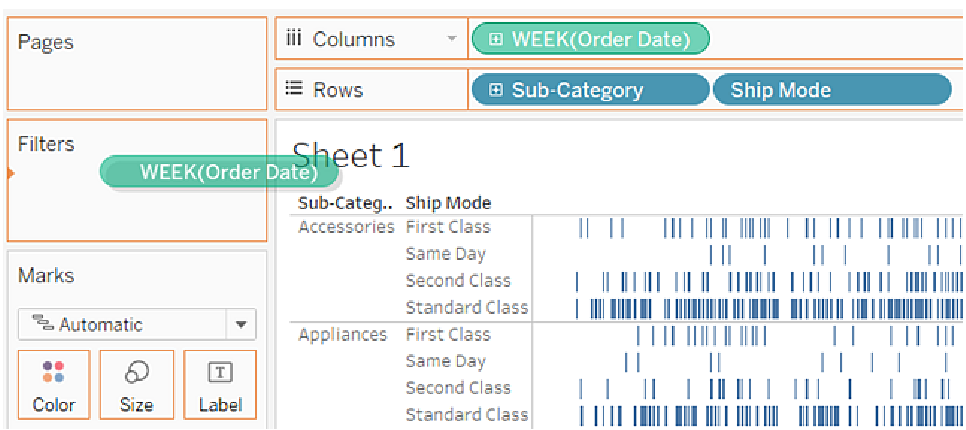
10. Right-click (Control-click on Mac) the **SUM(OrderUntilShip)** field on the **Marks** card, and then select **Measure (Sum)** > **Average**.



The view is coming along. But there are too many marks squeezed into the view.

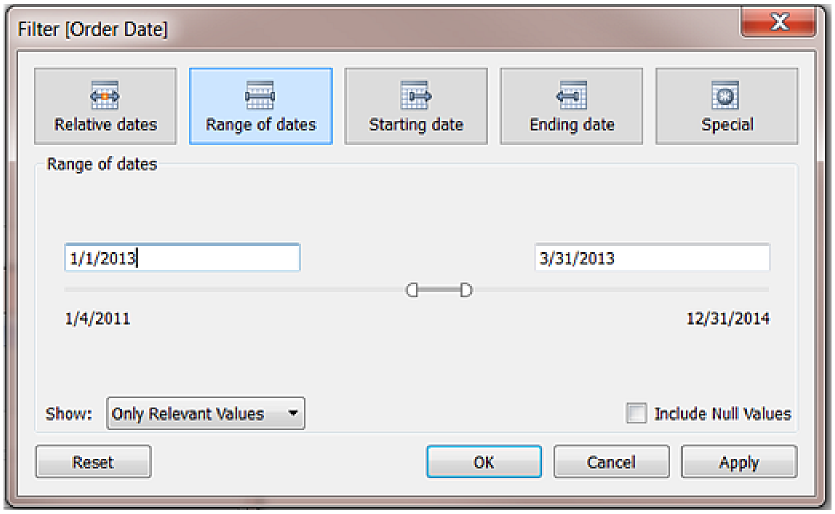
We can make our data more readable by filtering down to a smaller time window.

11. Hold down the Ctrl key (Option key on the Mac) and drag the **Week(Order Date)** field from the **Columns** shelf to the **Filter** shelf.



By holding down the Ctrl key (or the Option key), you tell Tableau that you want to copy the field to the new location, with whatever customizations you have added, without removing it from the old location.

12. In the Filter Field dialog box, select **Range of Dates** and then click **Next**.

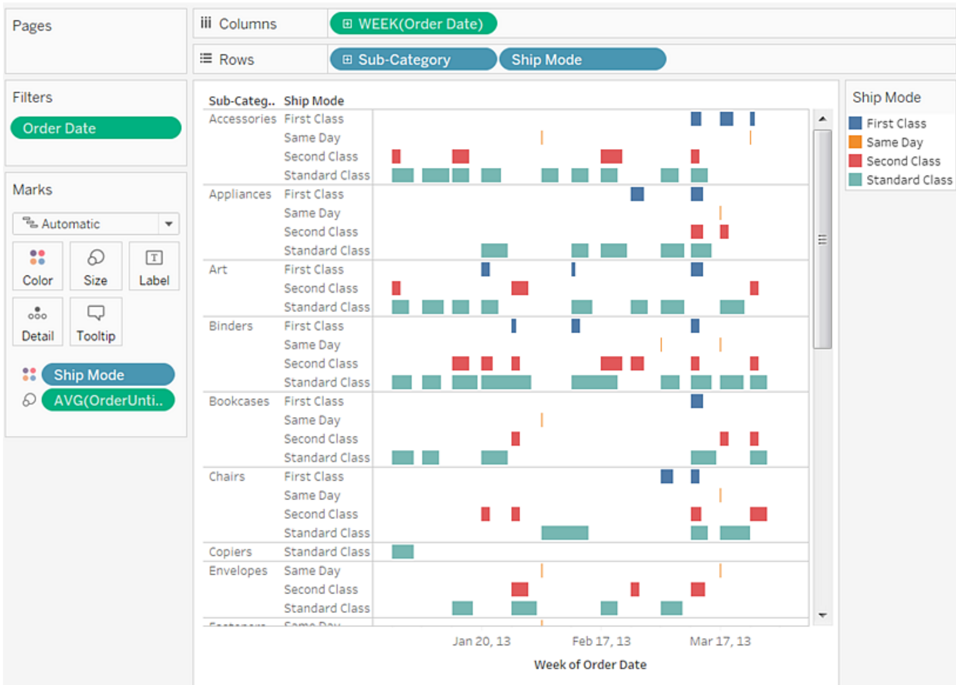


13. Set the range to a three-month time interval, such as 1/1/2013 to 3/31/2013, and then click **OK**.

    It can be difficult to get the exact date using the sliders—it's easier just to enter the numbers you want directly into the date boxes or use the calendar to select the dates.

14. Drag the **Ship Mode** dimension to **Color** on the **Marks** card.

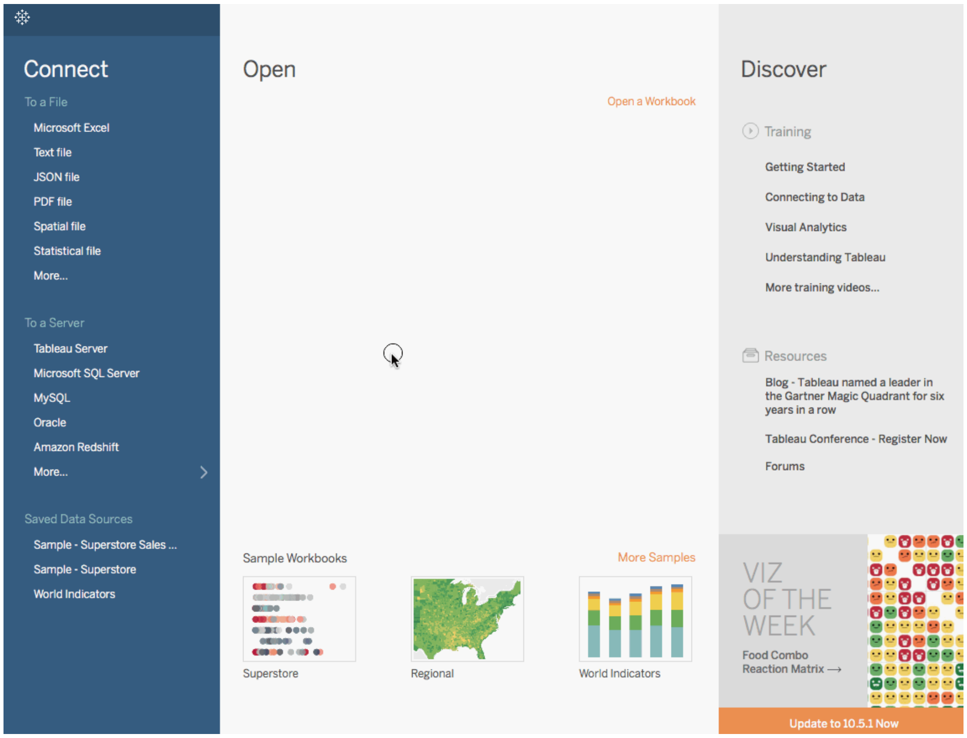
    Now your view shows you all sorts of information about the lag between order times and ship times.



For example, you can see which ship modes are more prone to longer lag times, whether lag times vary by category, and whether lag times are consistent over time.

Note: If you publish this view to Tableau Server, you can include filters that let users interact with the view by varying the time window, or filtering out various sub-categories or ship modes. For more information, see **Publish Data Sources and Workbooks** on page 3051.

Check your work! Watch steps 1-14 below:



Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Build a Highlight Table or Heat Map

Use highlight tables to compare categorical data using color.

In Tableau, you create a highlight table by placing one or more dimensions on the **Columns** shelf and one or more dimensions on the **Rows** shelf. You then select **Square** as the mark type and place a measure of interest on the **Color** shelf.

You can enhance this basic highlight table by setting the size and shape of the table cells to create a heat map.

To create a highlight table to explore how profit varies across regions, product sub-categories, and customer segments, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Segment** dimension to **Columns**.

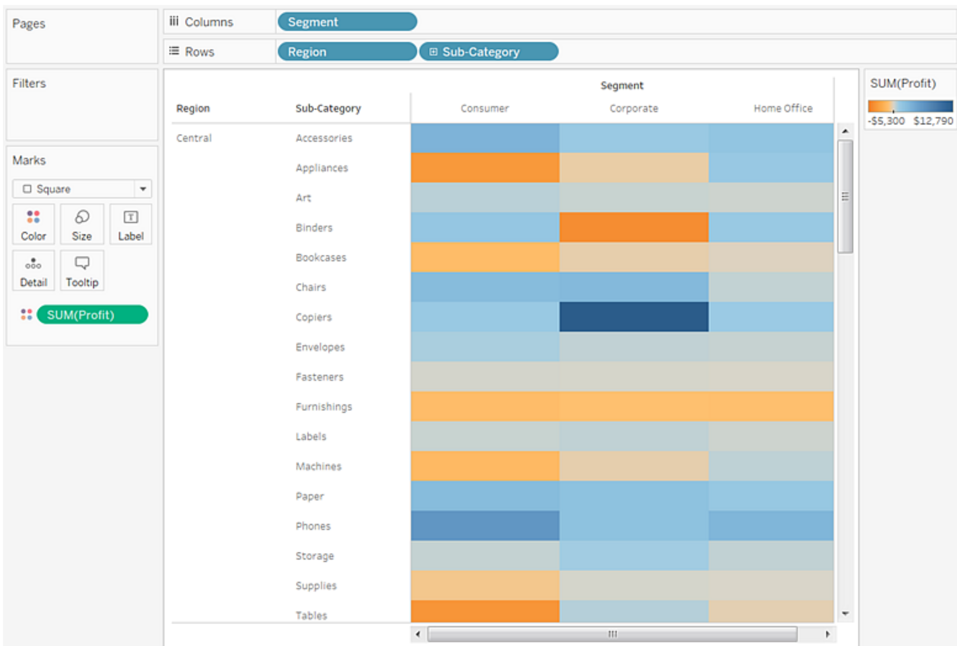
   Tableau creates headers with labels derived from the dimension member names.

3. Drag the **Region** and **Sub-Category** dimensions to **Rows**, dropping **Sub-Category** to the right of **Region**.

   Now you have a nested table of categorical data (that is, the **Sub-Category** dimension is nested within the **Region** dimension).

4. Drag the **Profit** measure to **Color** on the **Marks** card.

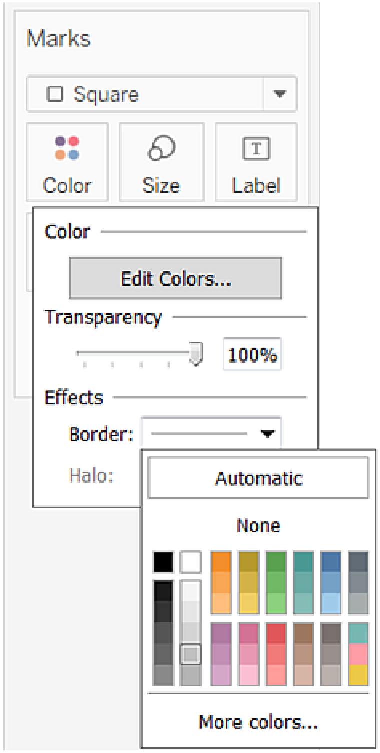
   Tableau aggregates the measure as a sum. The color legend reflects the continuous data range.

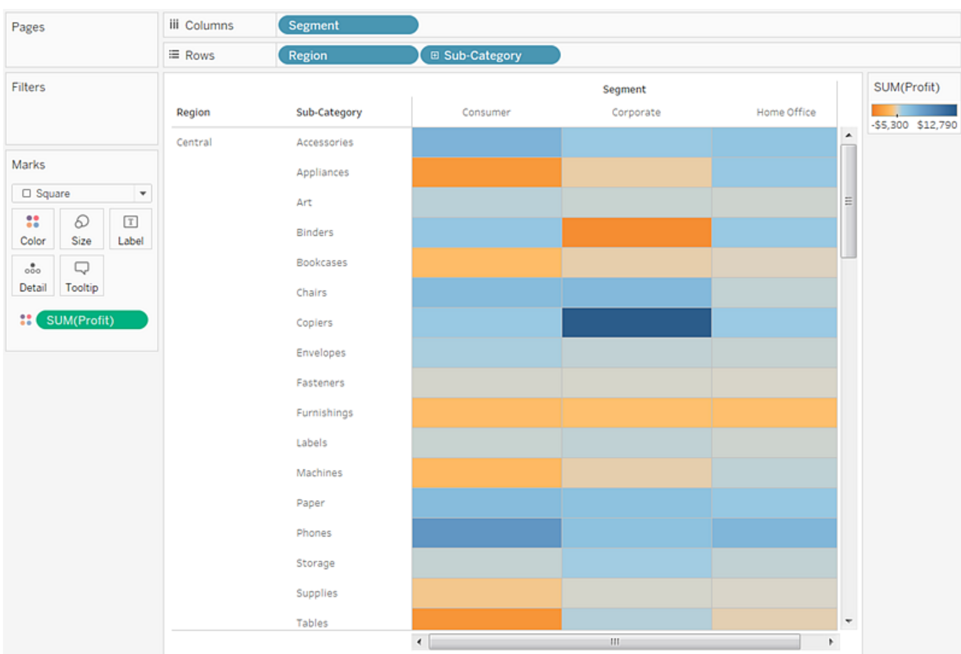
   In this view, you can see data for only the Central region. Scroll down to see data for other regions.

   In the Central region, copiers are shown to be the most profitable sub-category, and binders and appliances the least profitable.

5. Click **Color** on the **Marks** card to display configuration options. In the **Border** drop-down list, select a medium gray color for cell borders, as in the following image:



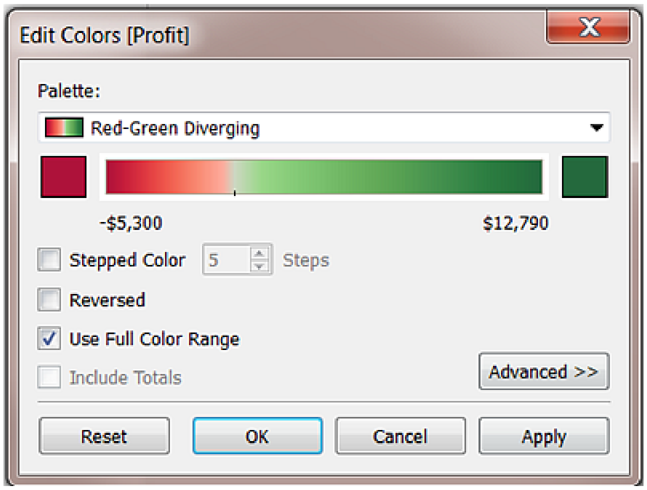
Now it's easier to see the individual cells in the view:



6. The default color palette is Orange-Blue Diverging. A Red-Green Diverging palette

might be more appropriate for profit. To change the color palette and to make the colors more distinct, do the following:

- Hover over the **SUM(Profit)** color legend, then click the drop-down arrow that appears and select **Edit Colors**.
- In the **Edit Colors** dialog box, in the **Palette** field, select **Red-Green Diverging** from the drop-down list.
- Select the **Use Full Color Range** check box and click **Apply** and then click **OK**.



When you select this option, Tableau assigns the starting number a full intensity and the ending number a full intensity. If the range is from -10 to 100, the color representing negative numbers changes in shade much more quickly than the color representing positive numbers.

When you do not select **Use Full Color Range**, Tableau assigns the color intensity as if the range was from -100 to 100, so that the change in shade is the same on both sides of zero. The effect is to make the color contrasts in your view much more distinct.

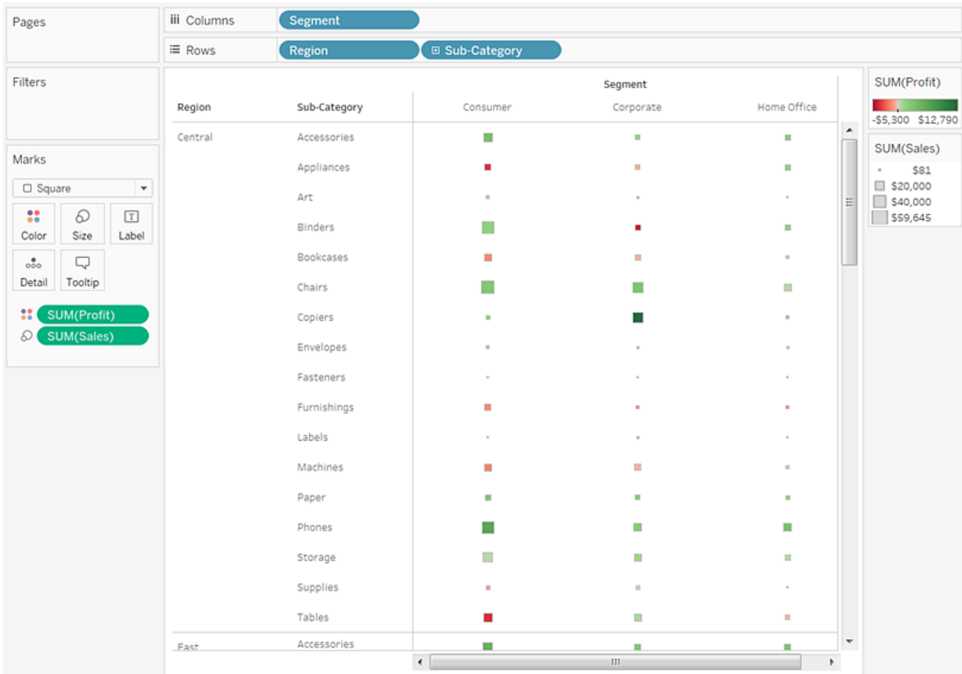
For more information about color options, see **Color Palettes and Effects** on

page 1394.

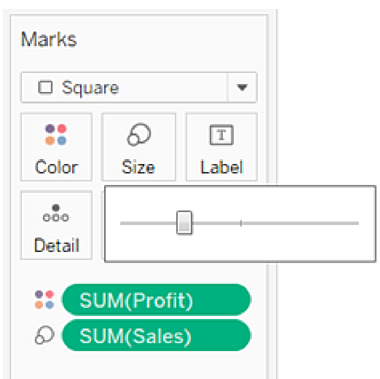
Modify the size to create a heat map

7. Drag the **Sales** measure to **Size** on the **Marks** card to control the size of the boxes by the Sales measure. You can compare absolute sales numbers (by size of the boxes) and profit (by color).

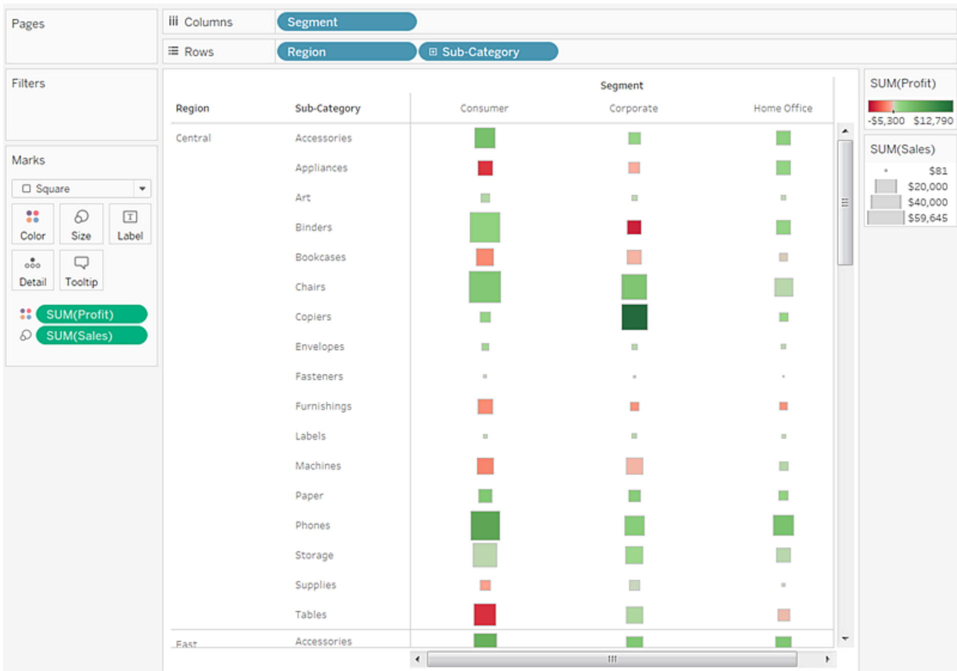
   Initially, the marks look like this:



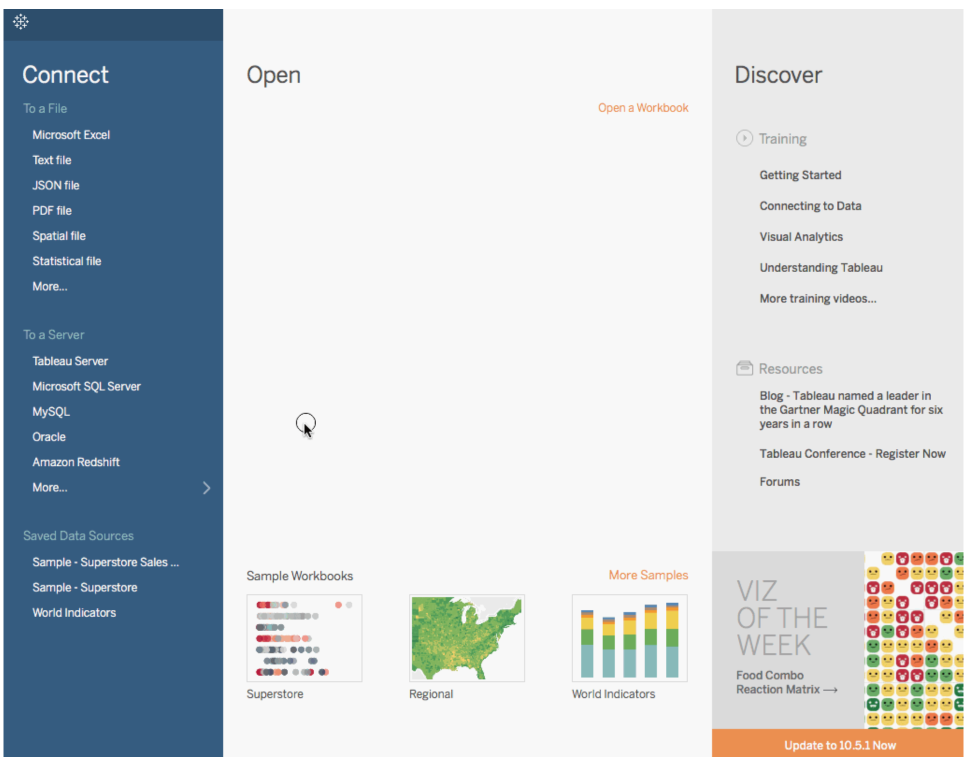
8. To enlarge the marks, click **Size** on the **Marks** card to display a size slider:



9. Drag the slider to the right until the boxes in the view are the optimal size. Now your view is complete:



Check your work! Watch steps 1-9 below:



Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Build a Histogram

A histogram is a chart that displays the shape of a distribution. A histogram looks like a bar chart but groups values for a continuous measure into ranges, or bins.

The basic building blocks for a histogram are as follows:

Mark type:	Automatic
Rows shelf:	Continuous measure (aggregated by Count or Count Distinct)
Columns shelf:	Bin (continuous or discrete). ***Note**: This bin should be created from the continuous measure on the Rows shelf. For more information on how to create a bin from a continuous measure, see* ***Create Bins from a Continuous Measure** on page 1705*.

In Tableau you can create a histogram using **Show Me**.

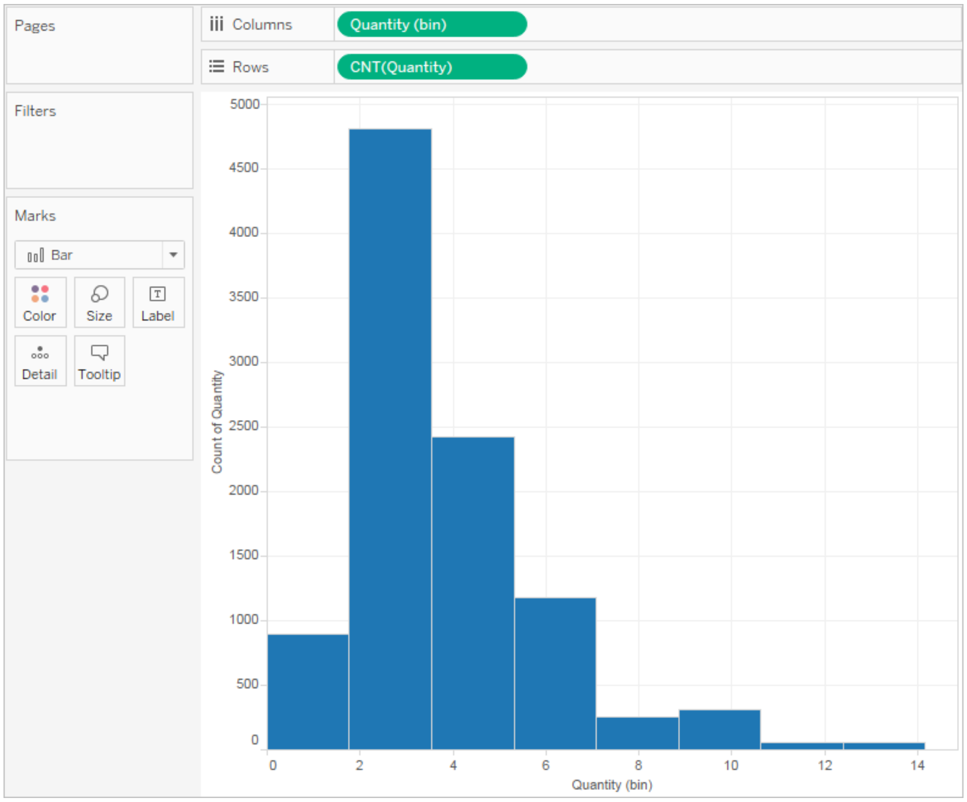
1. Connect to the **Sample - Superstore** data source.
2. Drag **Quantity** to **Columns**.
3. Click **Show Me** on the toolbar, then select the histogram chart type.

The histogram chart type is available in **Show Me** when the view contains a single measure and no dimensions.

Three things happen after you click the histogram icon in **Show Me**:

- The view changes to show vertical bars, with a continuous x-axis (1 – 14) and a continuous y-axis (0 – 5,000).
- The **Quantity** measure you placed on the **Columns** shelf, which had been aggregated as SUM, is replaced by a continuous **Quantity (bin)** dimension. (The green color of the field on the **Columns** shelf indicates that the field is continuous.)

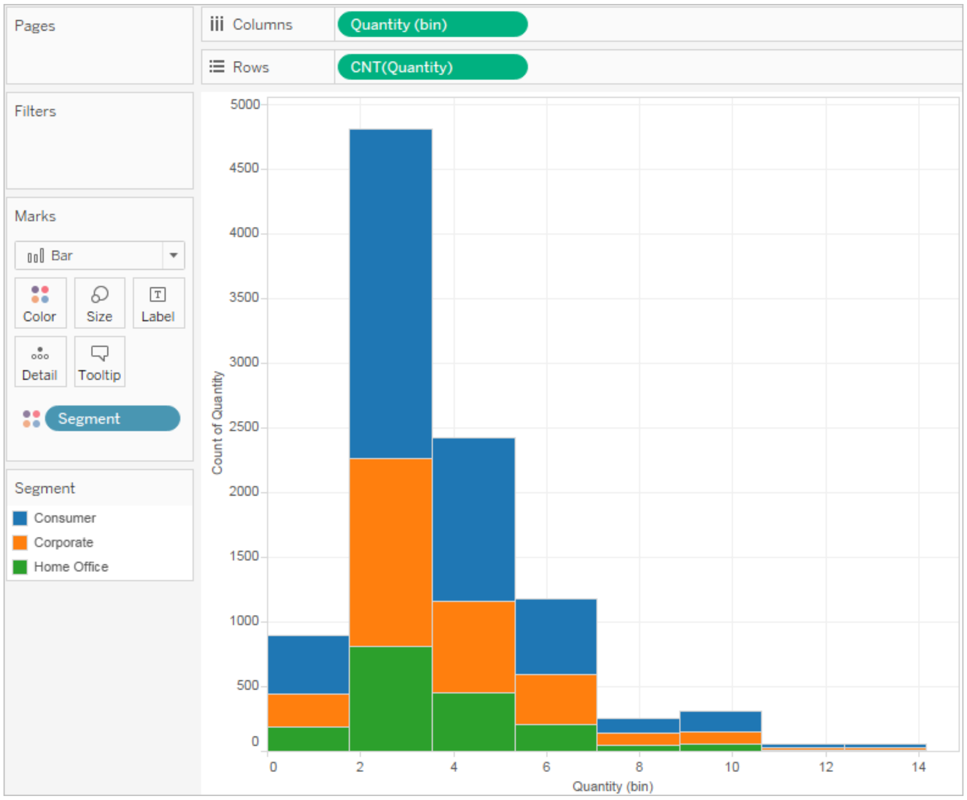
  To edit this bin: In the Data pane, right-click the bin and select **Edit**.
- The **Quantity** measure moves to the **Rows** shelf and the aggregation changes from SUM to CNT (Count).



The **Quantity** measure captures the number of items in a particular order. The histogram shows that about 4,800 orders contained two items (the second bar), about 2,400 orders contained 4 items (the third bar), and so on.

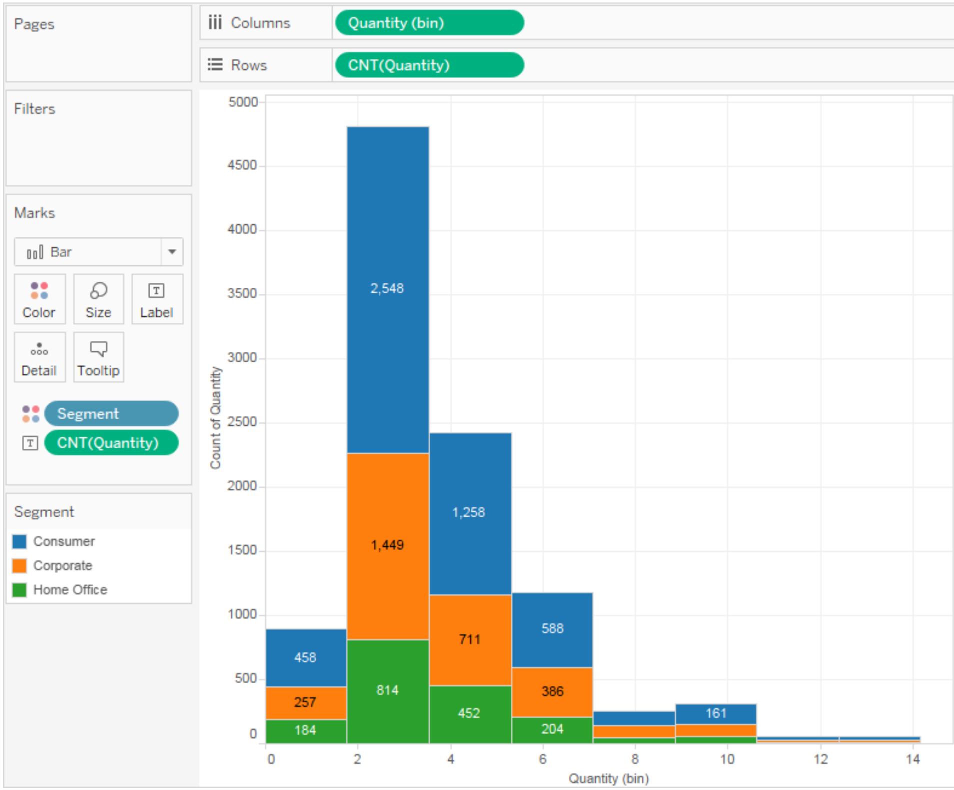
Let's take this view one step further and add **Segment** to **Color** to see if we can detect a relationship between the customer segment (consumer, corporate, or home office) and the quantity of items per order.

4. Drag **Segment** to **Color**.



The colors don't show a clear trend. Let's show the percentage of each bar that belongs to each segment.

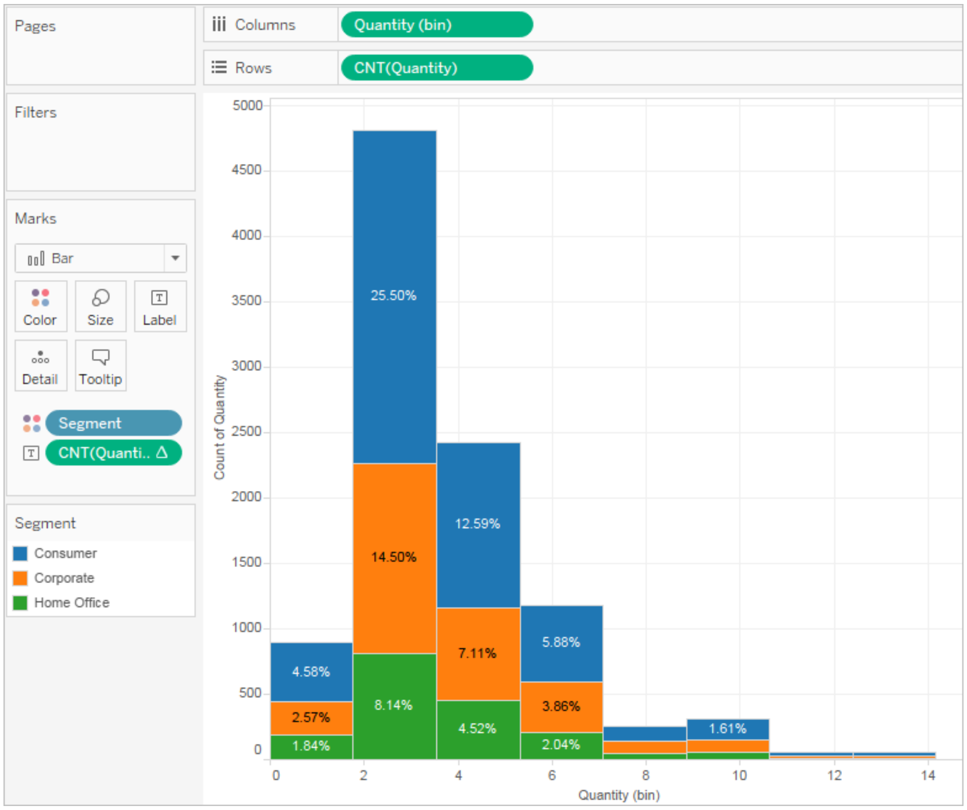
5. Hold down the Ctrl key and drag the **CNT(Quantity)** field from the **Rows** shelf to **Label**.



Holding down the Ctrl key copies the field to the new location without removing it from the original location.

6. Right-click (Control-click on a Mac) the **CNT(Quantity)** field on the **Marks** card and select **Quick Table Calculation** > **Percent of Total**.

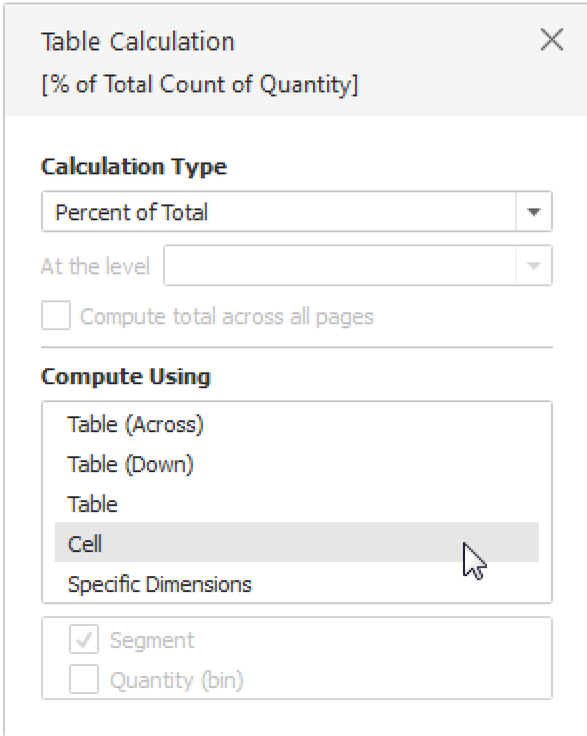
   Now each colored section of each bar shows its respective percentage of the total quantity:



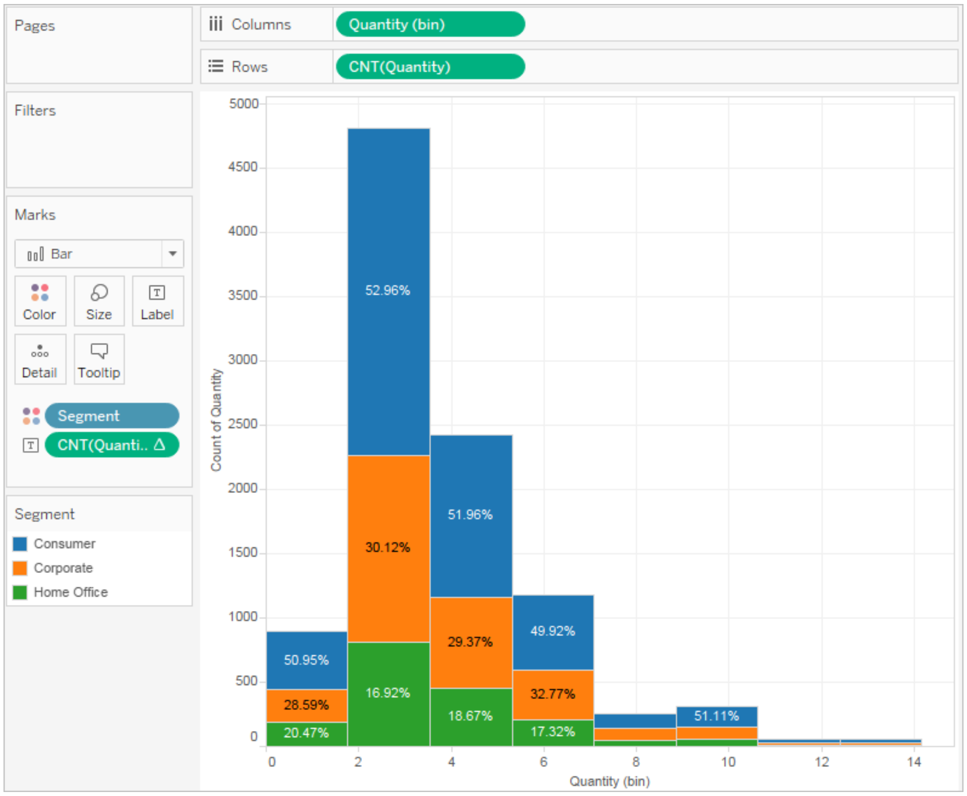
But we want the percentages to be on a per-bar basis.

7. Right-click the **CNT(Quantity)** field on the **Marks** card again and select **Edit Table Calculation**.

8. In the Table Calculation dialog box, change the value of the **Compute Using** field to **Cell**.

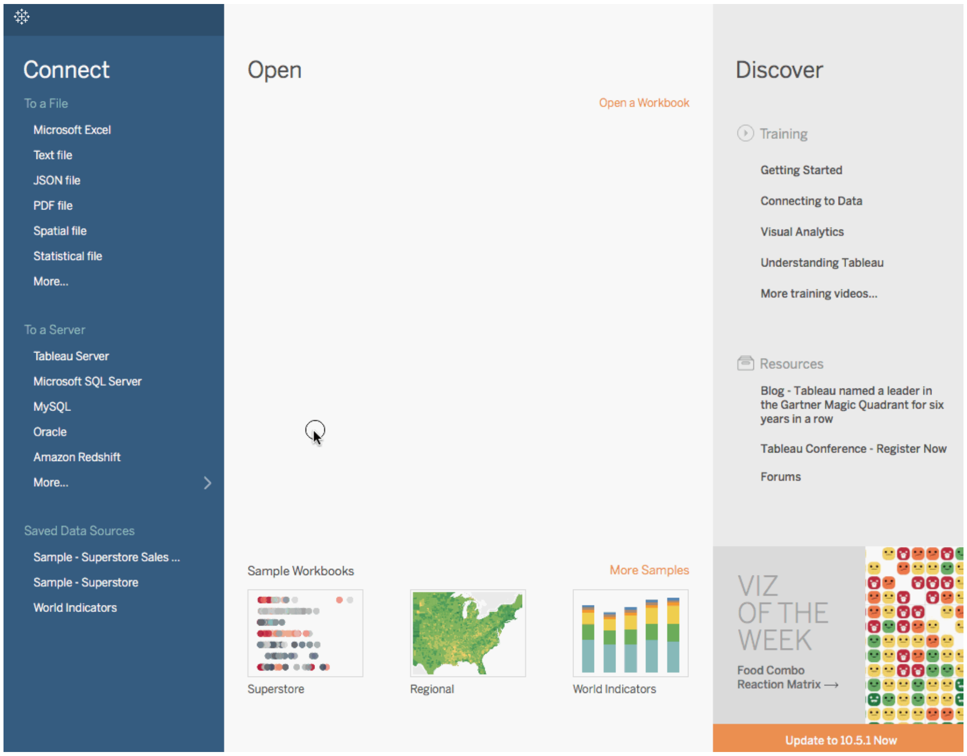


Now we have the view that we want:



There is still no evidence that the percentages by customer segment show any trend as the number of items in an order increases.

Check your work! Watch steps 1-8 below:



Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Building Line Charts

Line charts connect individual data points in a view. They provide a simple way to visualize a sequence of values and are useful when you want to see trends over time, or to forecast future values. For more information about the line mark type, see **Line mark** on page 1338.

Note: In views that use the line mark type, you can use the **Path** property in the Marks card to change the type of line mark (linear, step, or jump), or to encode data by connecting marks using a particular drawing order. For details, see Path properties in the **Control the Appearance of Marks in the View** on page 1356

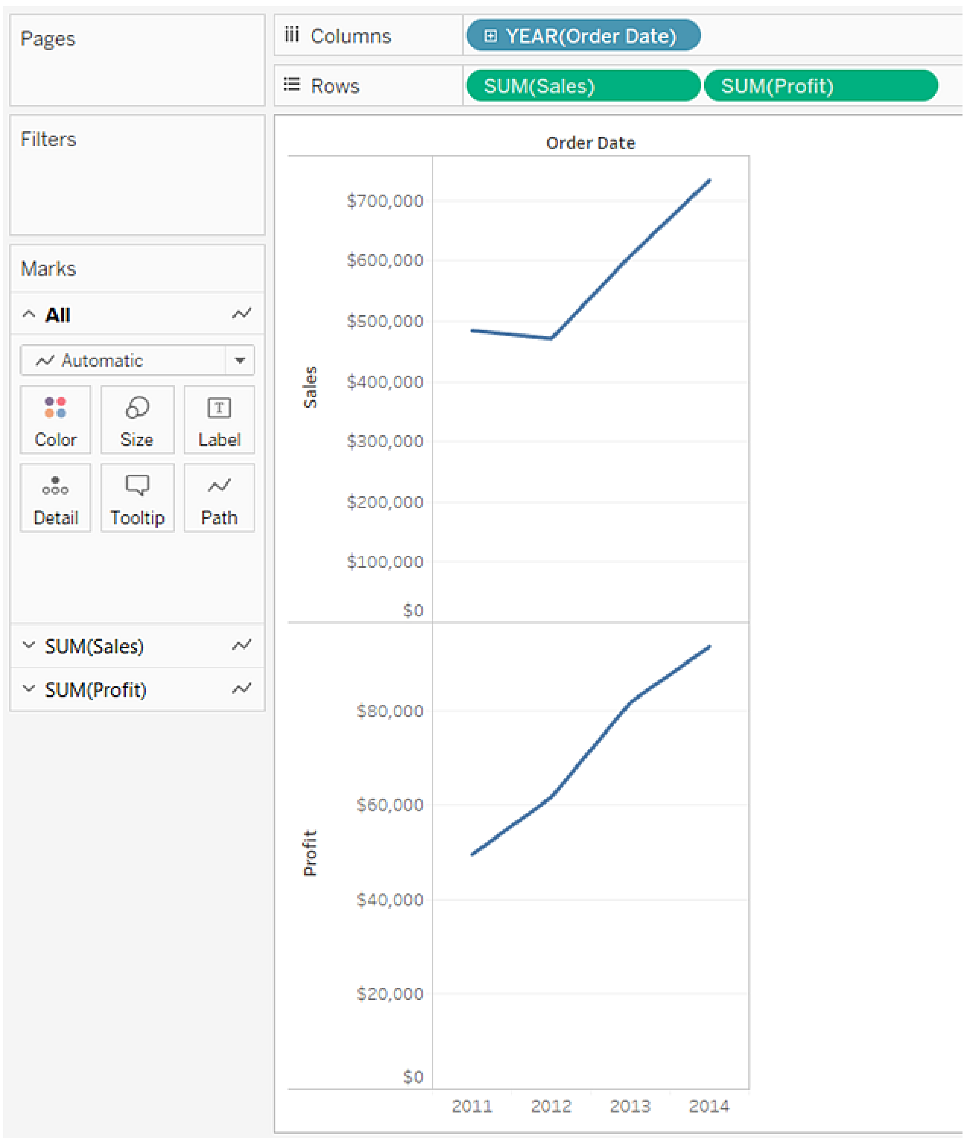
To create a view that displays the sum of sales and the sum of profit for all years, and then uses forecasting to determine a trend, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Order Date** dimension to **Columns**.

   Tableau aggregates the date by year, and creates column headers.
3. Drag the **Sales** measure to **Rows**.

   Tableau aggregates **Sales** as SUM and displays a simple line chart.
4. Drag the **Profit** measure to **Rows** and drop it to the right of the **Sales** measure.

   Tableau creates separate axes along the left margin for **Sales** and **Profit**.



Notice that the scale of the two axes is different—the **Sales** axis scales from $0 to $700,000, whereas the **Profit** axis scales from $0 to $100,000. This can make it hard to see that sales values are much greater than profit values.

When you are displaying multiple measures in a line chart, you can align or merge axes to make it easier for users to compare values.

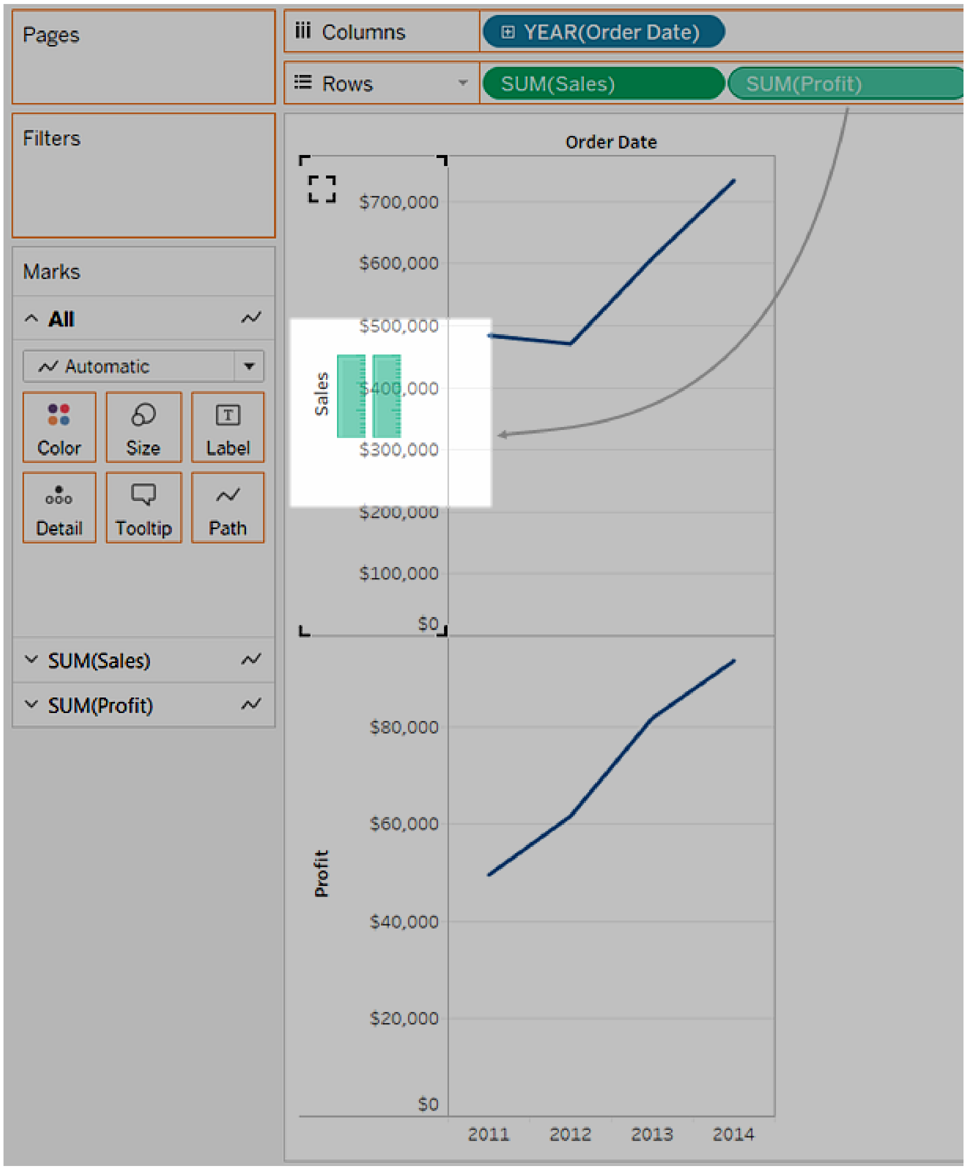
For more information about aligning the axes, see **Compare two measures using dual axes** on page 1302.

For more information about enforcing a single axis across multiple measures, see **Blend axes for multiple measures into a single axis** on page 1300.

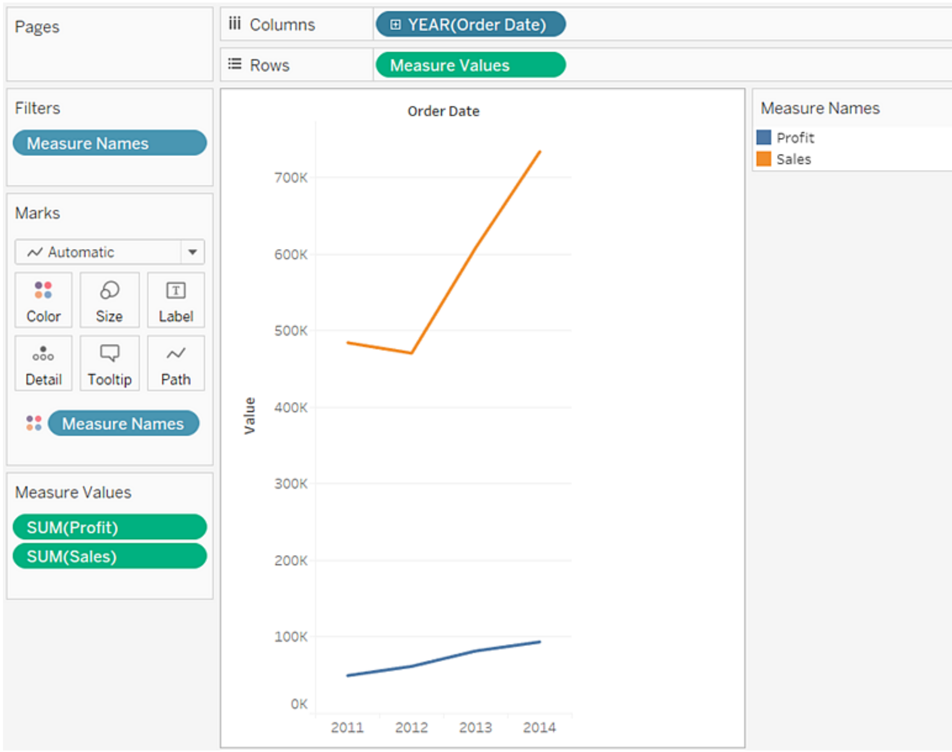
With either of these options, you can create a combination chart to change the mark type for one of your measures.

For more information, see **Create a combo chart (assign different mark types to measures)** on page 1306.

5. Drag the **SUM(Profit)** field from **Rows** to the **Sales** axis to create a blended axis. The two pale green parallel bars indicate that **Profit** and **Sales** will use a blended axis when you release the mouse button.

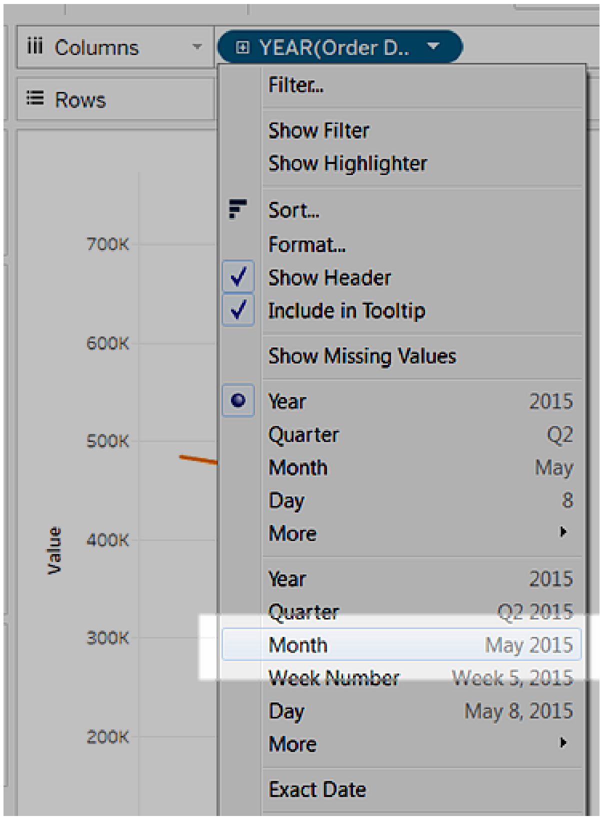


The view updates to look like this:

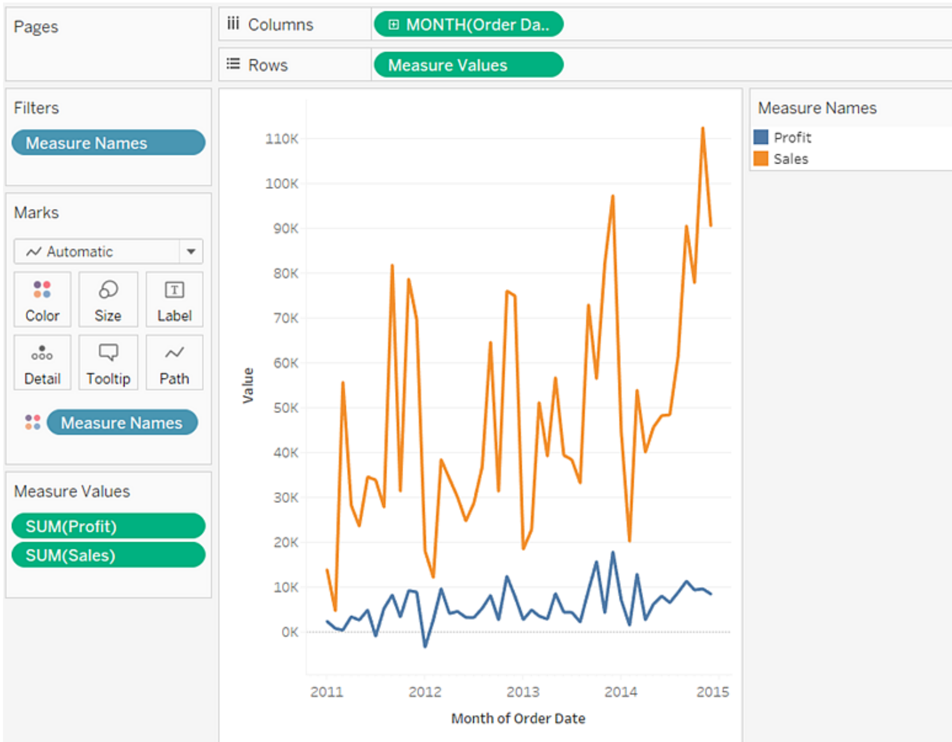


The view is rather sparse because we are looking at a summation of values on a per-year basis.

6. Click the drop-down arrow in the **Year(Order Date)** field on the **Columns** shelf and select **Month** in the lower part of the context menu to see a continuous range of values over the four-year period.

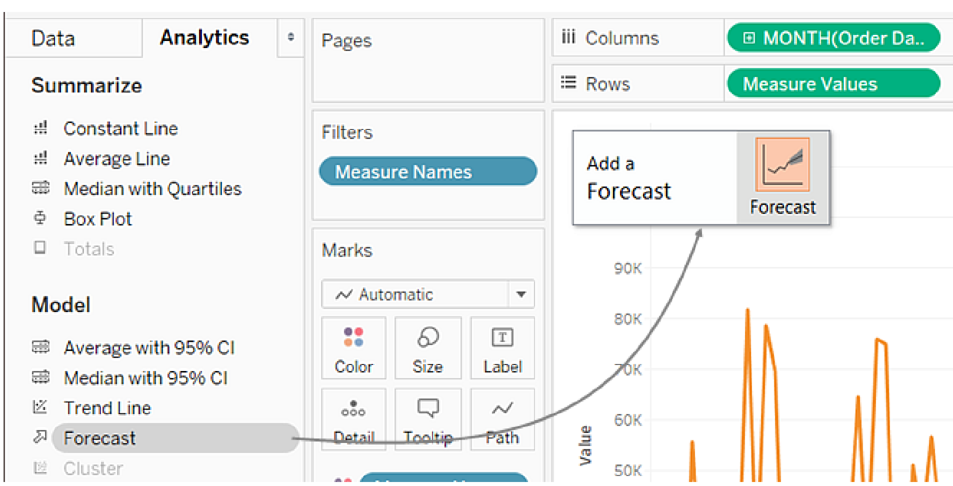


The resulting view is a lot more detailed than the original view:

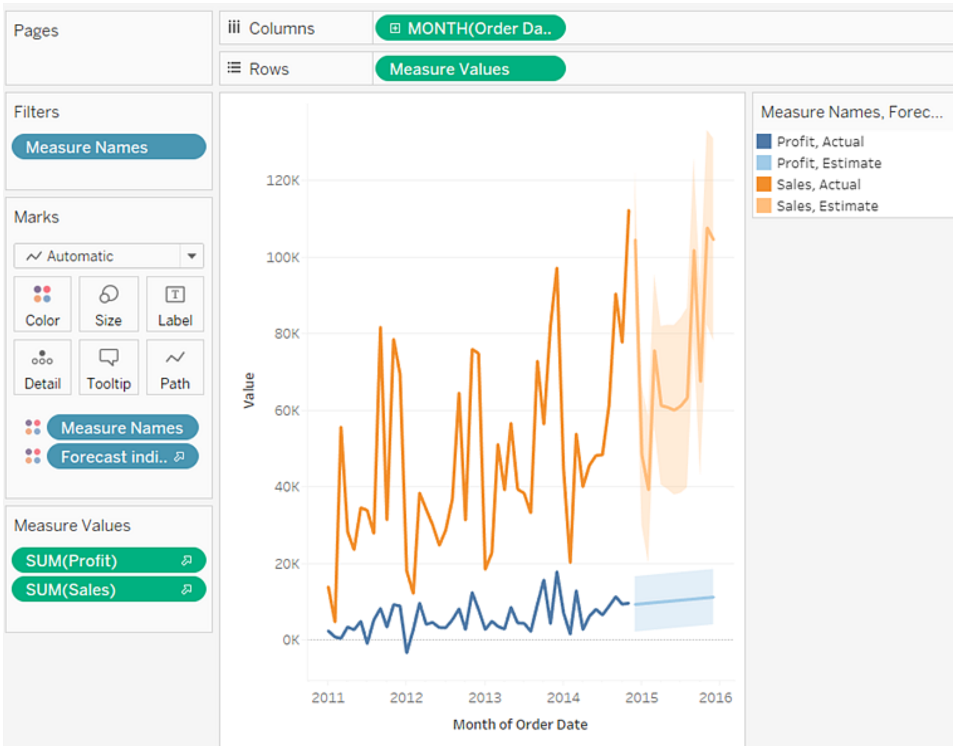


Notice that the values seem to go much higher just before the end of each year. A pattern like that is known as *seasonality*. If we turn on the forecasting feature in the view, we can see whether we should expect that the apparent seasonal trend will continue in the future.

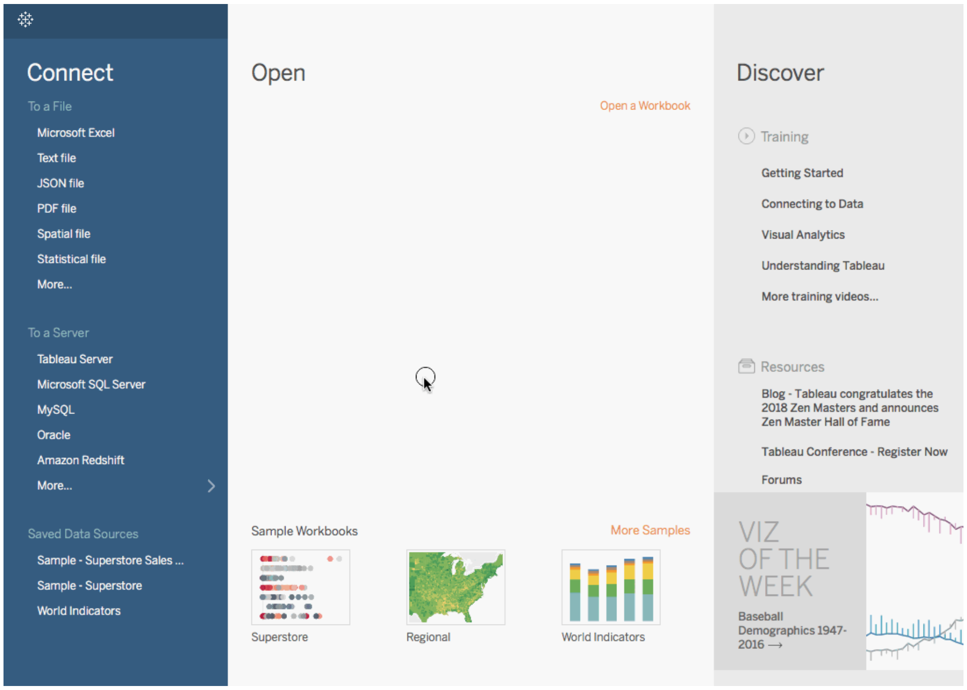
7. To add a forecast, in the **Analytics** pane, drag the **Forecast** model to the view, and then drop it on **Forecast**.



We then see that, according to Tableau forecasting, the seasonal trend does continue into the future:



Check your work! Watch steps 1-7 below:



Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Build a Packed Bubble Chart

Use packed bubble charts to display data in a cluster of circles. Dimensions define the individual bubbles, and measures define the size and color of the individual circles.

The basic building blocks for a packed bubble chart are as follows:

Mark type:	Circle
Detail:	Dimension
Size:	Measure
Color:	Dimension or Measure
Label (optional):	Dimension or Measure

To create a basic packed bubble chart that shows sales and profit information for different product categories, follow these steps:

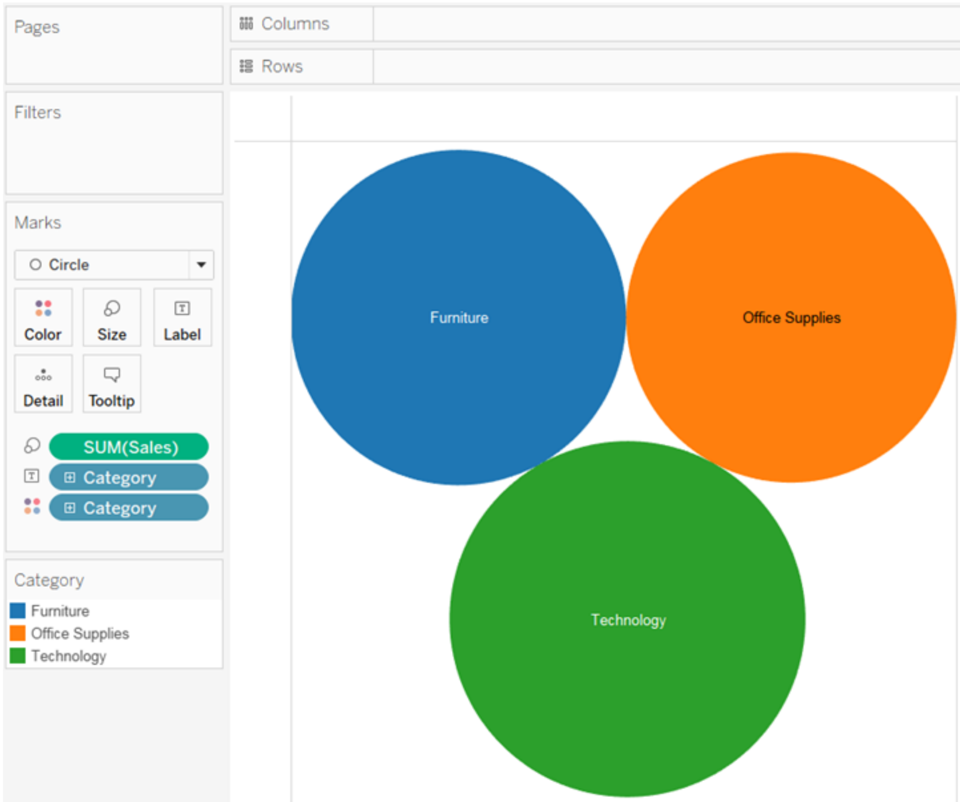
1. Connect to the **Sample - Superstore** data source.
2. Drag the **Category** dimension to **Columns**.

   A horizontal axis displays product categories.
3. Drag the **Sales** measure to **Rows**.

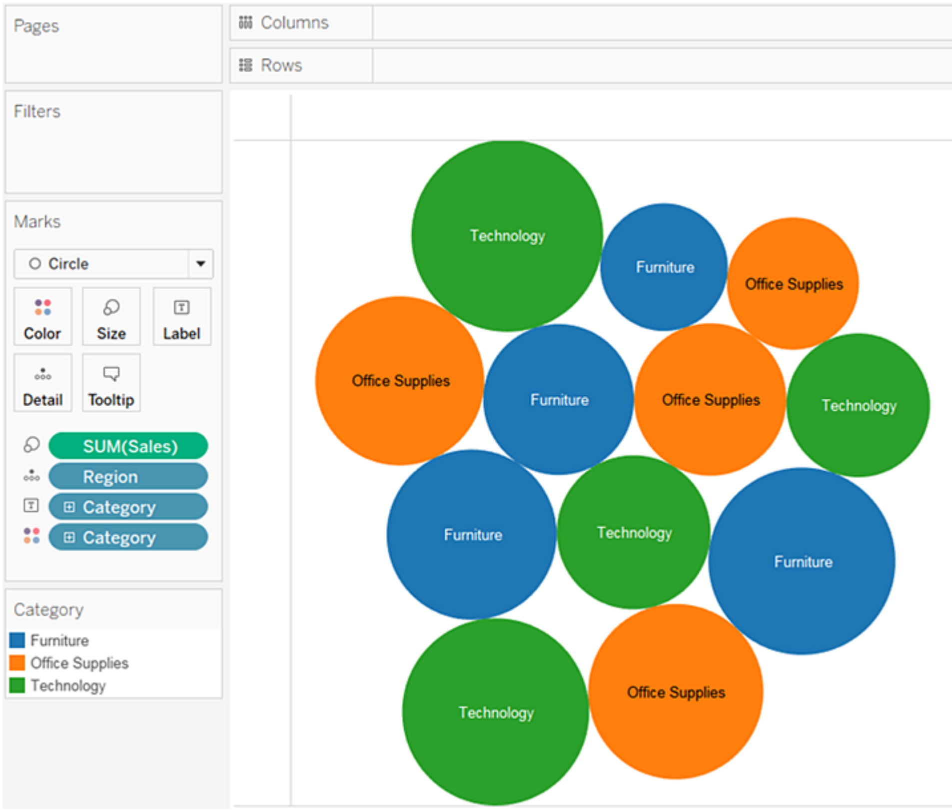
   The measure is aggregated as a sum and a vertical axis appears.

   Tableau displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.
4. Click **Show Me** on the toolbar, then select the packed bubbles chart type.

   Tableau displays the following packed bubble chart:

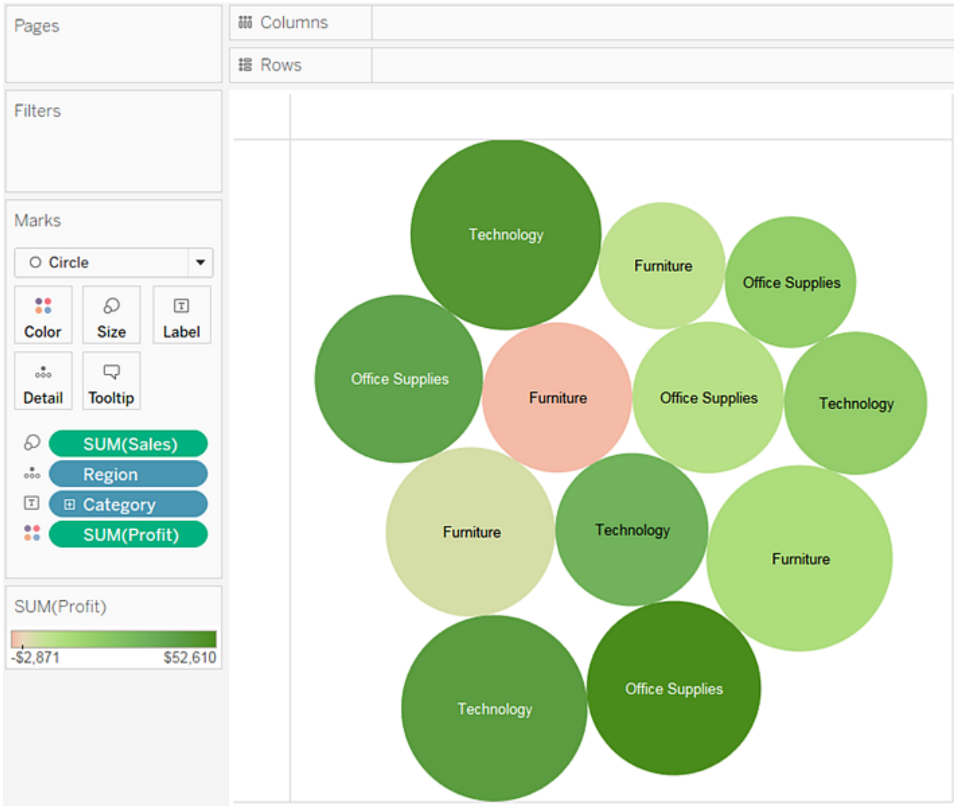


5. Drag **Region** to **Detail** on the **Marks** card to include more bubbles in the view.

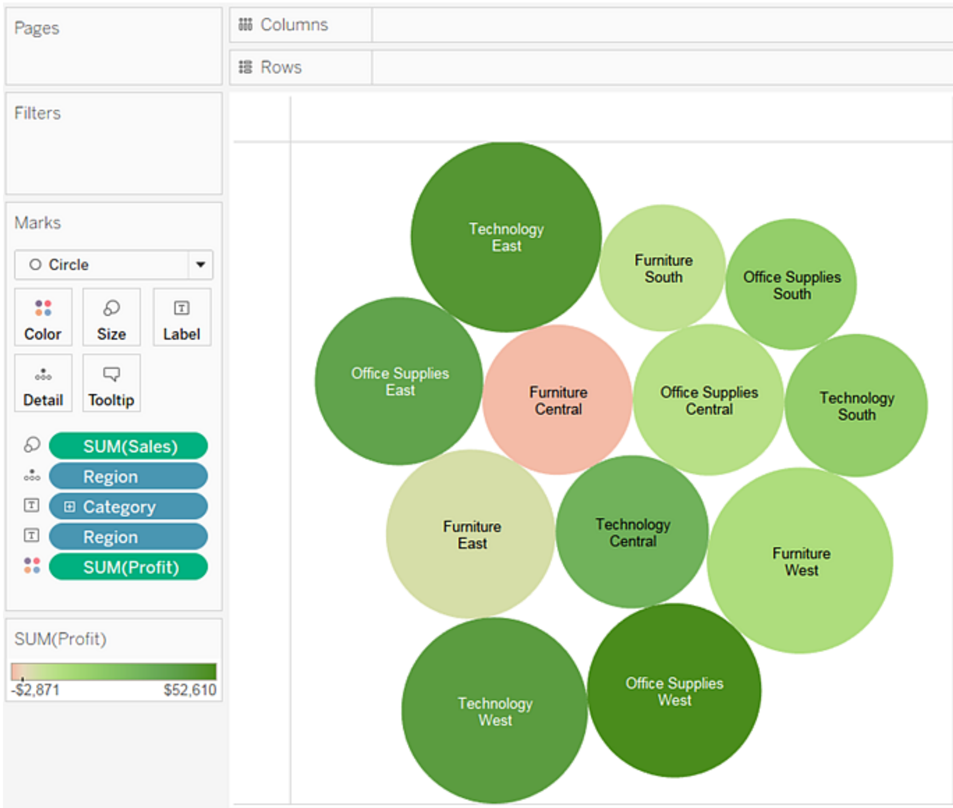


Next we'll add another layer of information to the view.

6. Drag **Profit** to **Color** on the **Marks** card:



7. Drag **Region** to **Label** on the **Marks** card to clarify what each bubble represents.

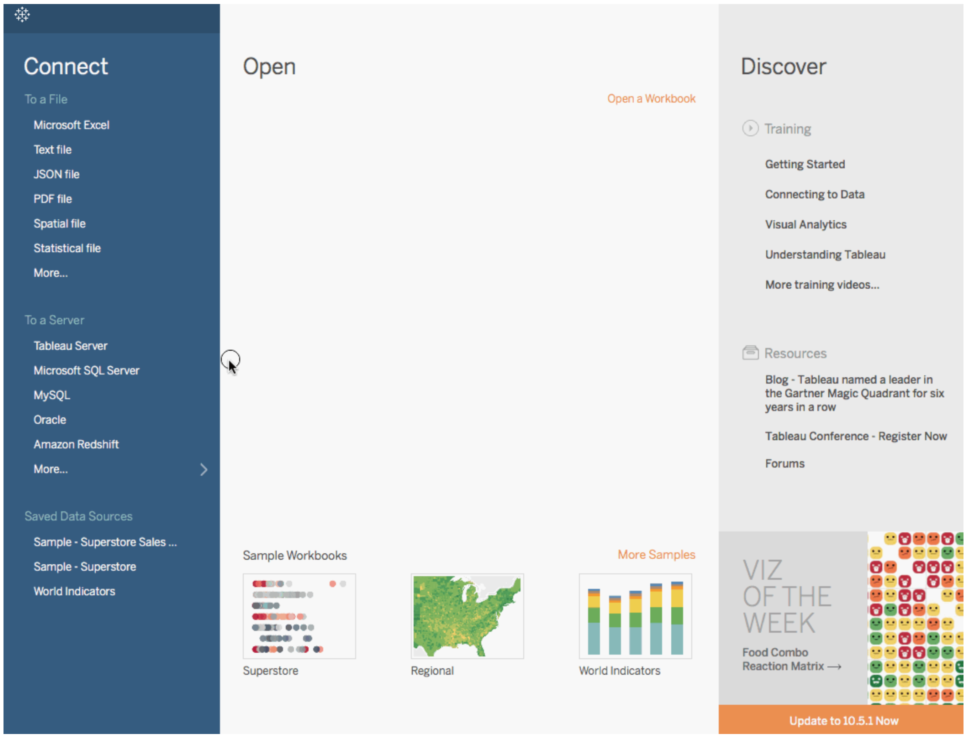


The size of the bubbles shows the sales for different combinations of region and category. The color of the bubbles shows the profit (the darker the green, the greater the profit).

For information about formatting mark labels, see **Show, Hide, and Format Mark Labels** on page 1378.

To further develop this view, you might edit the colors for **Profit** to show negative profit in a different color, or create a calculated field to shows profit divided by sales (that is, profit margin) and then drop that on **Color** instead of absolute profit. For more information, see **Color Palettes and Effects** on page 1394.

Check your work! Watch steps 1-7 below:



Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

Build a Pie Chart

Use pie charts to show proportions of a whole.

The basic building blocks for a pie chart are as follows:

Mark type:	Pie
Color:	Dimension
Angle:	Measure

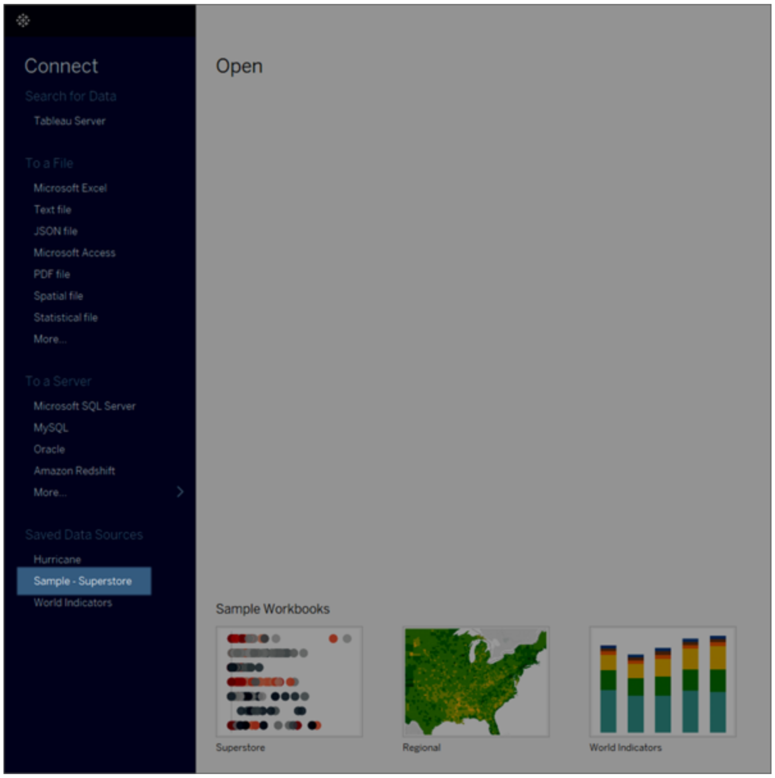
To create a pie chart view that shows how different product categories contribute to total sales, follow the steps below:

How to Build a Pie Chart

1. Connect to the **Sample - Superstore** data source.

Step 1:

Connect to the **Sample - Superstore** data source.



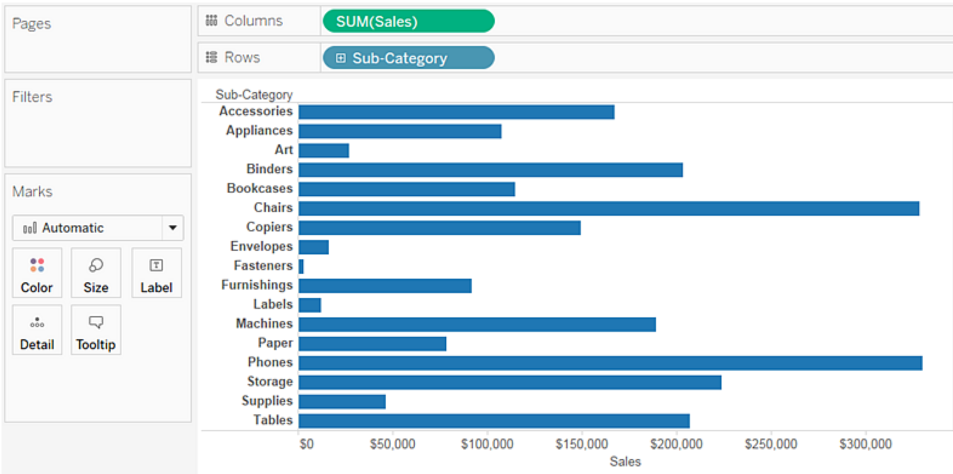
2. Drag **Sales** to **Columns** and **Sub-Category** to **Rows**.

Step 2:

Drag the **Sales** measure to **Columns** and drag the **Sub-Category** dimension to **Rows**.

Tableau aggregates the **Sales** measure as a sum.

Also note that the default chart type is a bar chart.



3. Click **Show Me** on the toolbar, then select the pie chart type.

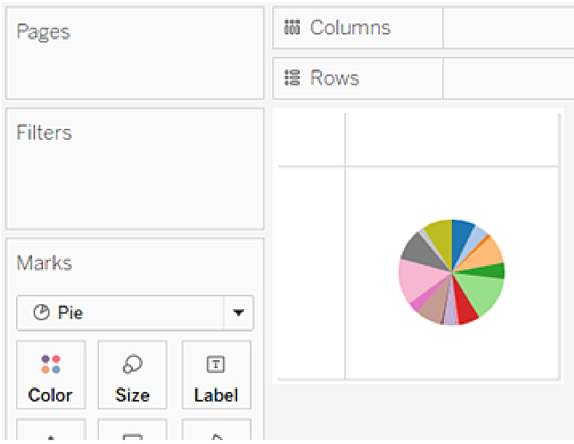
Step 3:

Click **Show Me** on the toolbar, then select the pie chart type.

4. Re-size the pie chart.

Step 4.

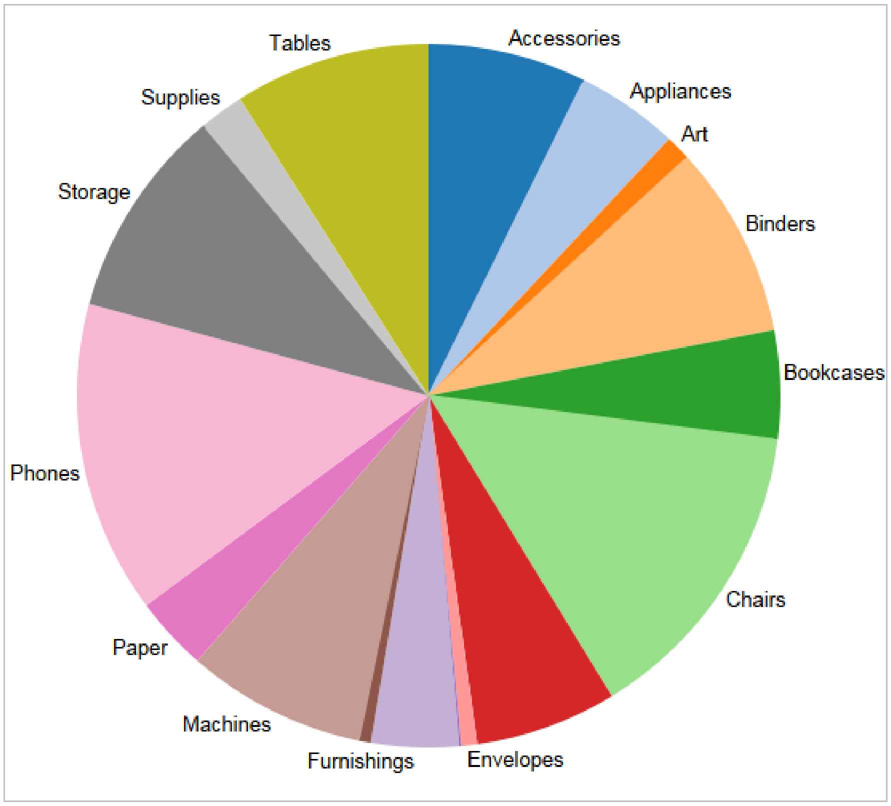
The result is a rather small pie. To make the chart bigger, hold down Ctrl + Shift (hold down ⇧ + ⌘ on a Mac) and press B several times.



5. To add labels, drag the **Sub-Category** dimension from the **Data** pane to **Label** on the **Marks** card.

Step 5:

To add labels, drag the **Sub-Category** dimension from the **Data** pane to **Label** on the **Marks** card.



6. If you don't see labels, press Ctrl + Shift + B (press ⇧ + ⌘ + B on a Mac) to make sure most of the individual labels are visible.

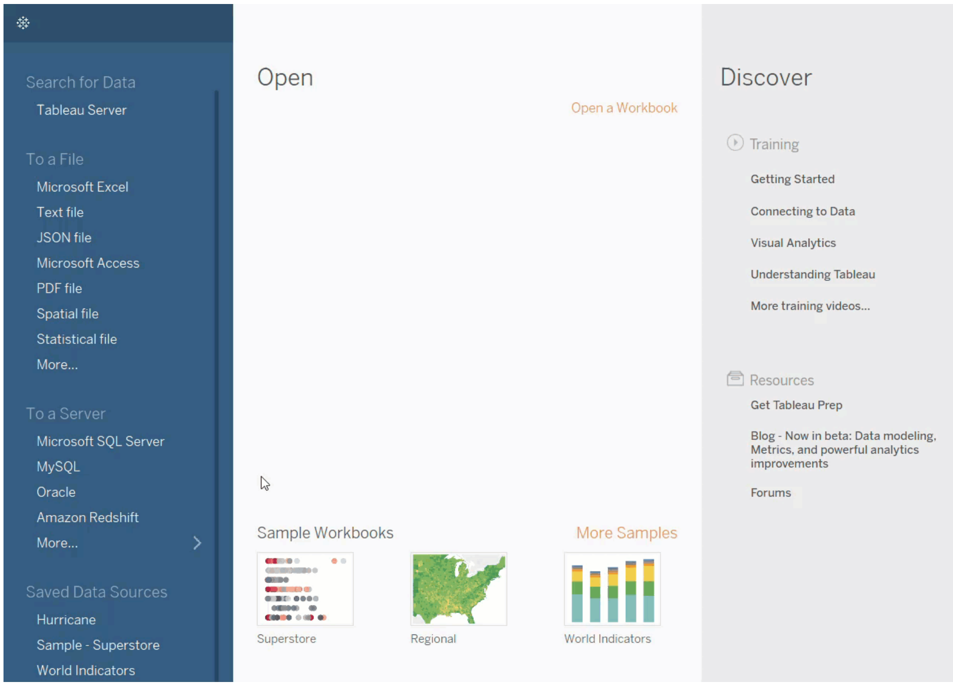
Step 6:

If you don't see labels, press Ctrl + Shift + B (press ⇧ + ⌘ + B on a Mac) to make sure most of the individual labels are visible.

You can make a pie chart interactive in a dashboard. For more information, see **Actions and Dashboards** on page 1566.

Note: Pie charts can also be used as a mark type in a visualization. For more information, see **Pie mark** on page 1350.

Check your work! Watch steps 1-7 below:

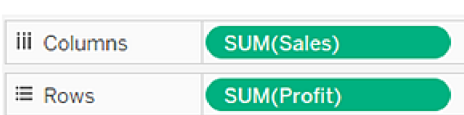


Build a Scatter Plot

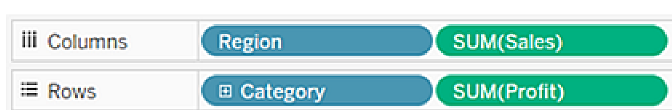
Use scatter plots to visualize relationships between numerical variables.

In Tableau, you create a scatter plot by placing at least one measure on the **Columns** shelf and at least one measure on the **Rows** shelf. If these shelves contain both dimensions and measures, Tableau places the measures as the innermost fields, which means that measures are always to the right of any dimensions that you have also placed on these shelves. The word "innermost" in this case refers to the table structure.

Creates Simple Scatter Plot



Creates Matrix of Scatter Plots



A scatter plot can use several mark types. By default, Tableau uses the shape mark type. Depending on your data, you might want to use another mark type, such as a circle or a square. For more information, see **Change the Type of Mark in the View** on page 1332.

To use scatter plots and trend lines to compare sales to profit, follow these steps:

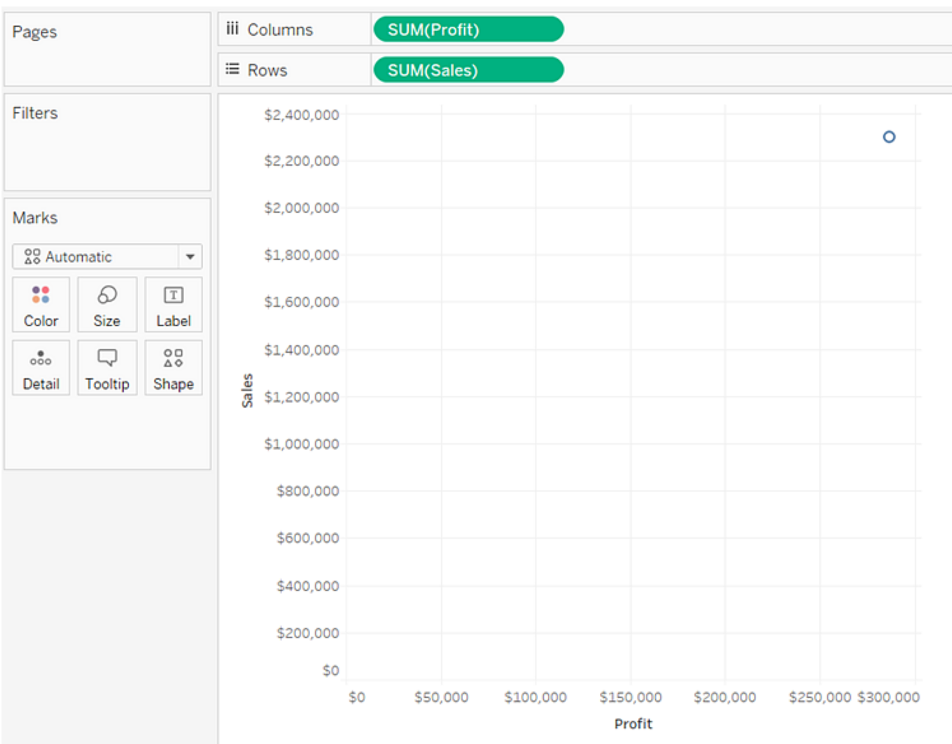
1. Open the **Sample - Superstore** data source.
2. Drag the **Profit** measure to **Columns**.

   Tableau aggregates the measure as a sum and creates a horizontal axis.
3. Drag the **Sales** measure to **Rows**.

   Tableau aggregates the measure as a sum and creates a vertical axis.

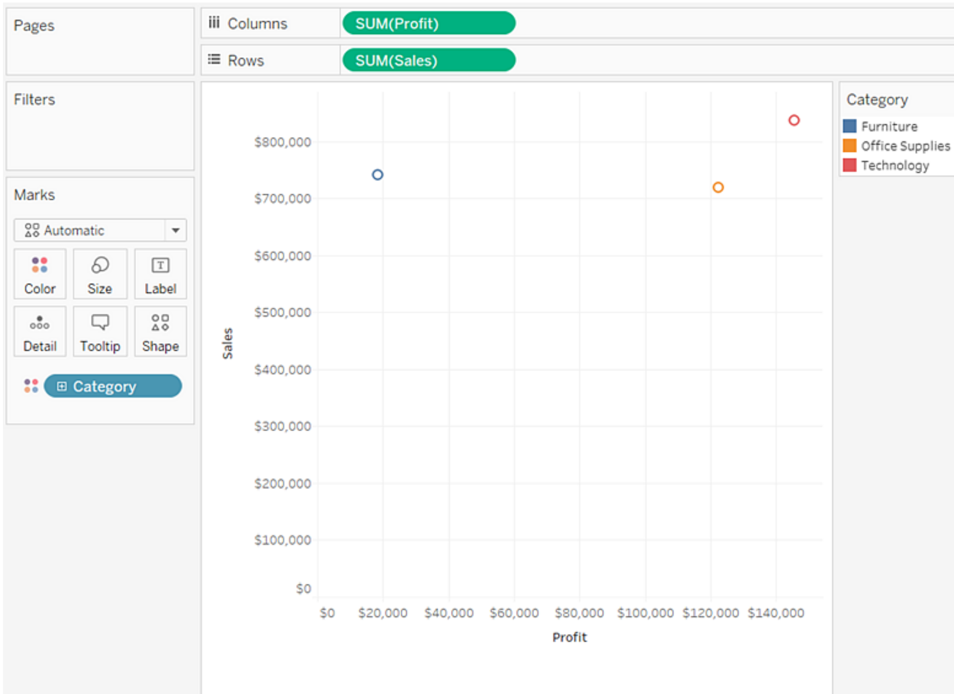
   Measures can consist of continuous numerical data. When you plot one number against another, you are comparing two numbers; the resulting chart is analogous to a Cartesian chart, with x and y coordinates.

   Now you have a one-mark scatter plot:



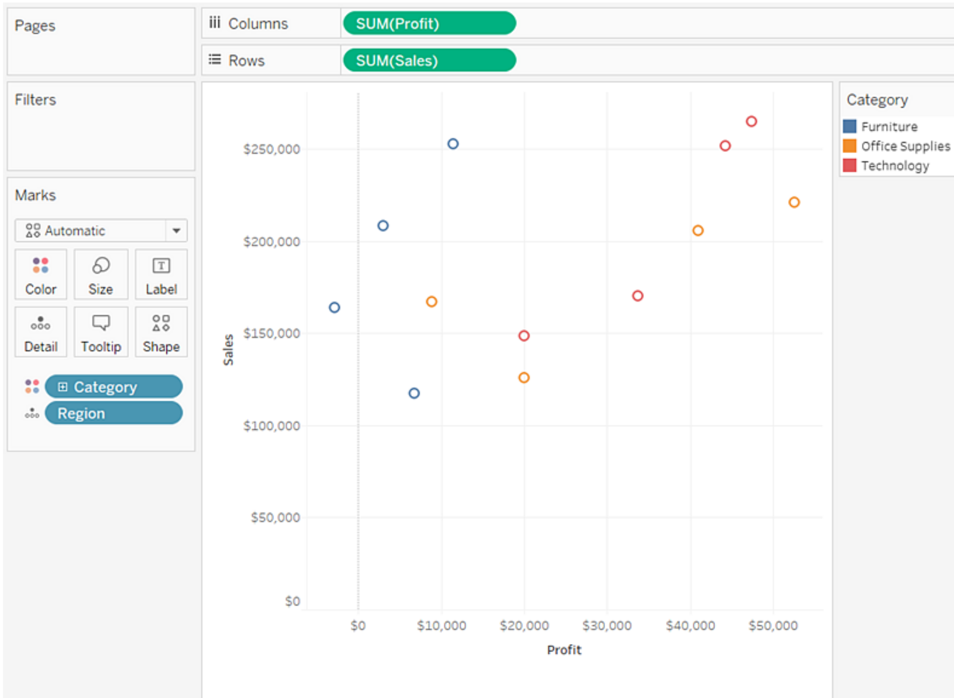
4. Drag the **Category** dimension to **Color** on the Marks card.

   This separates the data into three marks—one for each dimension member—and encodes the marks using color.

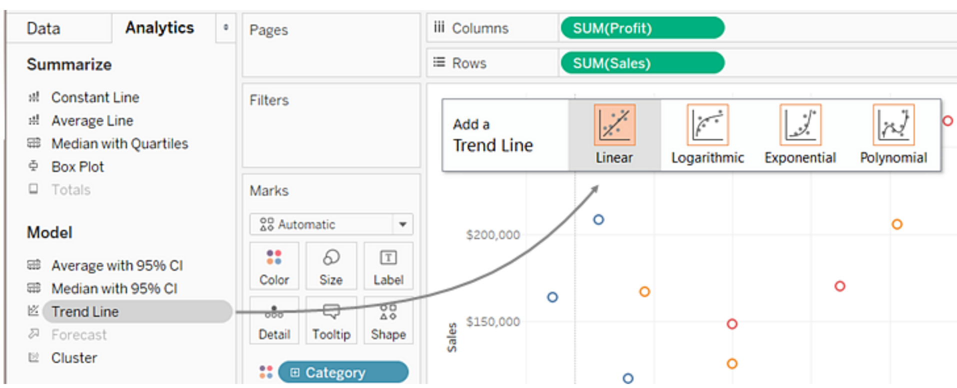


5. Drag the **Region** dimension to **Detail** on the **Marks** card.

   Now there are many more marks in the view. The number of marks is equal to the number of distinct regions in the data source multiplied by the number of departments. (If you're curious, use the **Undo** button on the toolbar to see what would have happened if you'd dropped the **Region** dimension on **Shape** instead of **Detail**.)

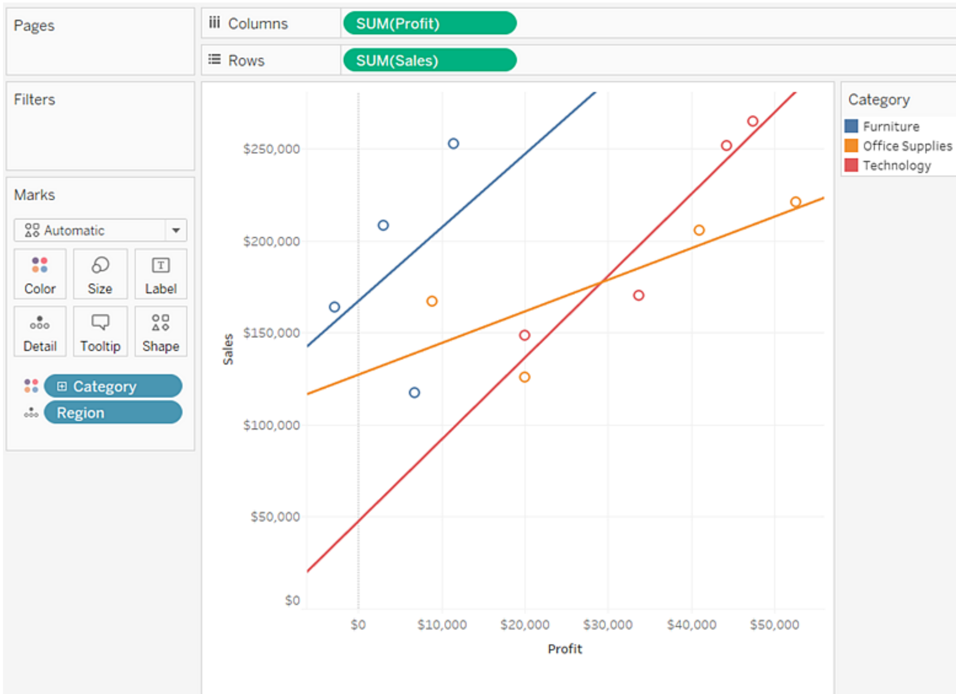


6. To add trend lines, from the **Analytics** pane, drag the **Trend Line** model to the view, and then drop it on the model type.

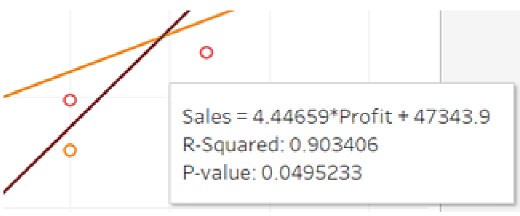


A trend line can provide a statistical definition of the relationship between two numerical values. To add trend lines to a view, both axes must contain a field that can be interpreted as a number—by definition, that is always the case with a scatter plot.

Tableau adds three linear trend lines—one for each color that you are using to distinguish the three categories.

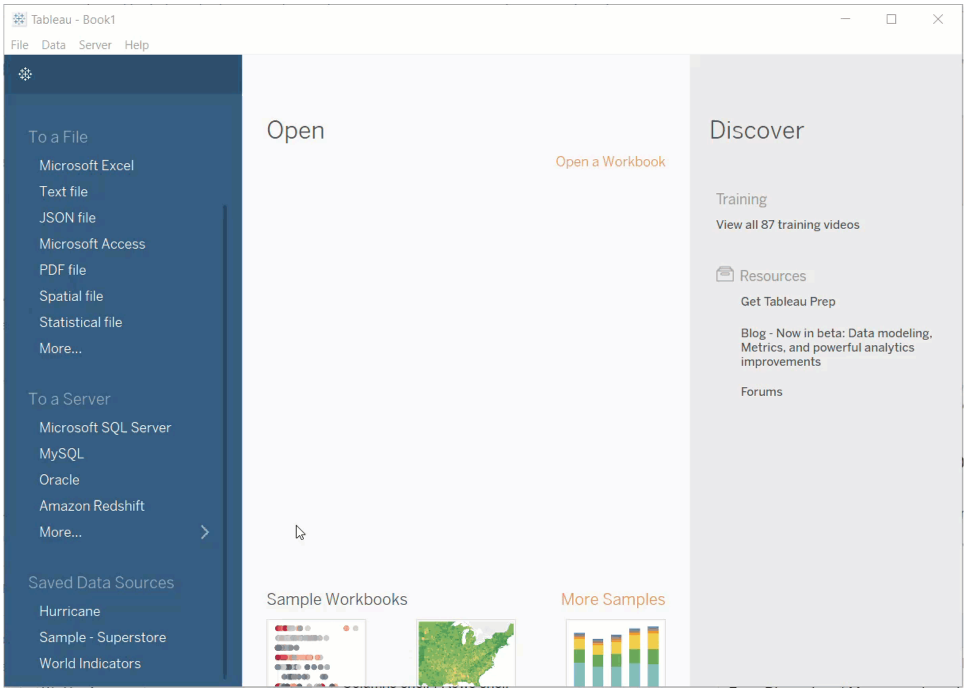


7. Hover the cursor over the trend lines to see statistical information about the model that was used to create the line:



For more information, see **Assess Trend Line Significance** on page 2540. You can also customize the trend line to use a different model type or to include confidence bands. For more information, see **Add Trend Lines to a Visualization** on page 2528.

Check your work! See steps 1-7 below:



See Also

Example: Scatter Plots, Aggregation, and Granularity on page 351

Build a Text Table

In Tableau, you typically create text tables (also called cross-tabs or pivot tables) by placing one dimension on the **Rows** shelf and another dimension on the **Columns** shelf. You then complete the view by dragging one or more measures to **Text** on the **Marks** card.

A text table uses the text mark type. Tableau uses this mark type automatically if the view is constructed using only dimensions (assuming the mark type is set to **Automatic**). For more information about the text mark type, see **Text mark** on page 1345.

To create a text table that shows sales totals by year and category, follow these steps:

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Order Date** dimension to **Columns**.

   Tableau aggregates the date by year and creates column headers.
3. Drag the **Sub-Category** dimension to **Rows**.

Tableau creates row headers. Columns with headers plus rows with headers means that a valid table structure now exists. For more information about changing the layout for row and column headers, see **Define Table Structure** on page 2966.

Now you can add a measure to the view to see actual data.

4. Drag the **Sales** measure to **Text** on the **Marks** card.

   Tableau aggregates the measure as a sum.

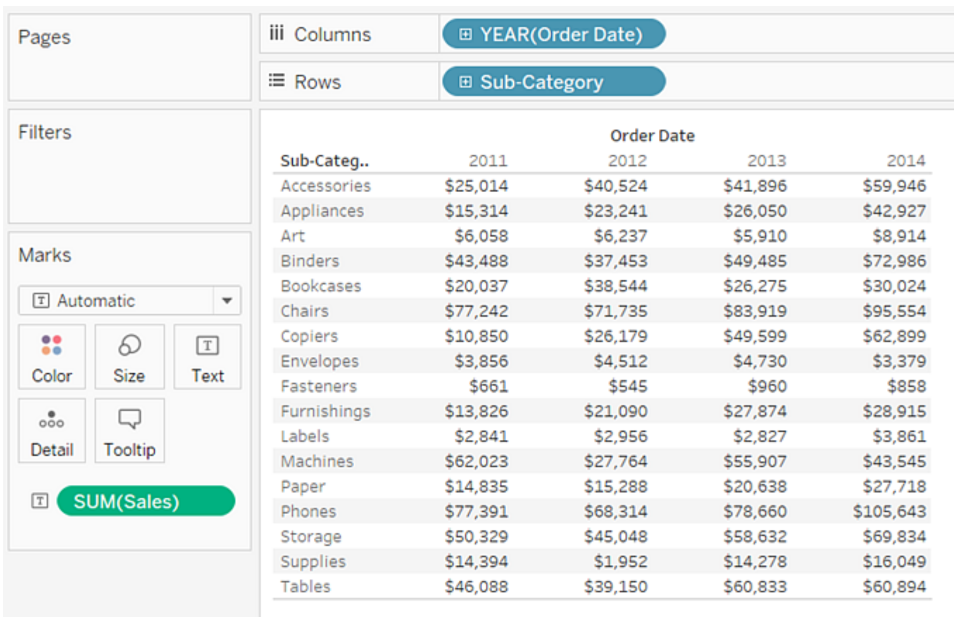
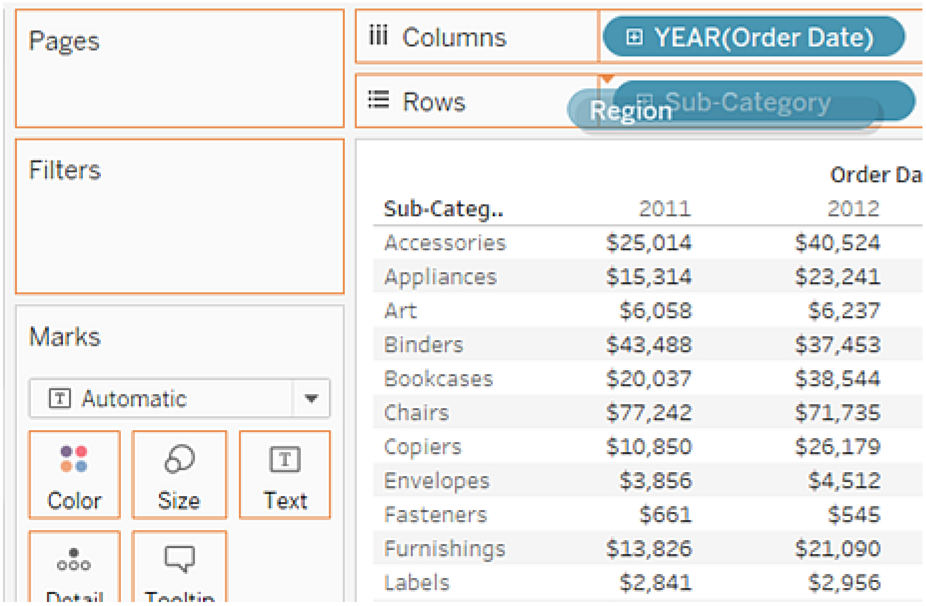


   Tableau uses text as the mark type. Each cell in the table displays the sum of sales for a particular year and sub-category.

   We can see that the chairs and phones sub-categories had the highest sales in every year.

5. Drag the **Region** dimension to **Rows** and drop it to the left of **Sub-Category**. A small triangle will appear to indicate that the new field will be inserted to the left of the existing field.

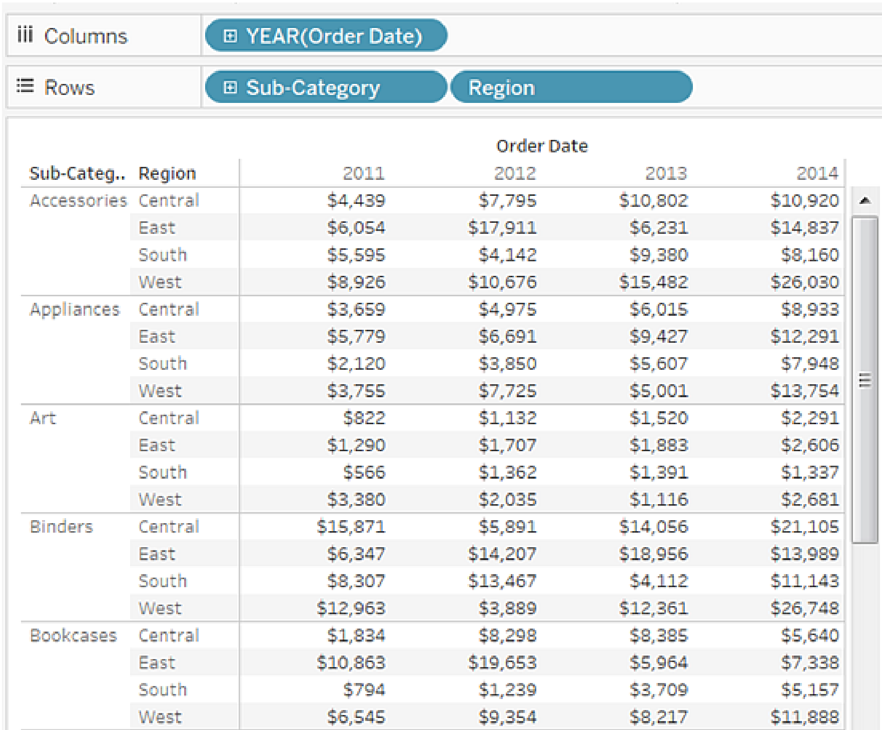


Sub-Categ..	Order Da 2011	2012
Accessories	$25,014	$40,524
Appliances	$15,314	$23,241
Art	$6,058	$6,237
Binders	$43,488	$37,453
Bookcases	$20,037	$38,544
Chairs	$77,242	$71,735
Copiers	$10,850	$26,179
Envelopes	$3,856	$4,512
Fasteners	$661	$545
Furnishings	$13,826	$21,090
Labels	$2,841	$2,956

The view now breaks out sales by region, in addition to year and sub-category.

Region	Sub-Categ..	Order Date 2011	2012	2013	2014
Central	Accessories	$4,439	$7,795	$10,802	$10,920
	Appliances	$3,659	$4,975	$6,015	$8,933
	Art	$822	$1,132	$1,520	$2,291
	Binders	$15,871	$5,891	$14,056	$21,105
	Bookcases	$1,834	$8,298	$8,385	$5,640
	Chairs	$20,754	$17,909	$23,350	$23,218
	Copiers	$3,270	$12,810	$17,500	$3,680
	Envelopes	$1,599	$871	$971	$1,197
	Fasteners	$122	$89	$247	$320
	Furnishings	$2,536	$2,529	$5,116	$5,074
	Labels	$1,048	$305	$511	$587
	Machines	$16,292	$1,852	$2,659	$5,995
	Paper	$2,347	$3,544	$5,366	$6,235
	Phones	$9,926	$19,364	$19,902	$23,211
	Storage	$11,093	$8,331	$12,812	$13,694
	Supplies	$440	$324	$4,295	$4,408
	Tables	$7,785	$6,857	$13,923	$10,589
East	Accessories	$6,054	$17,911	$6,231	$14,837
	Appliances	$5,779	$6,691	$9,427	$12,291

Regions are listed alphabetically. You can drag **Region** to the right of **Sub-Category** to organize the view first by sub-category, and then by region.



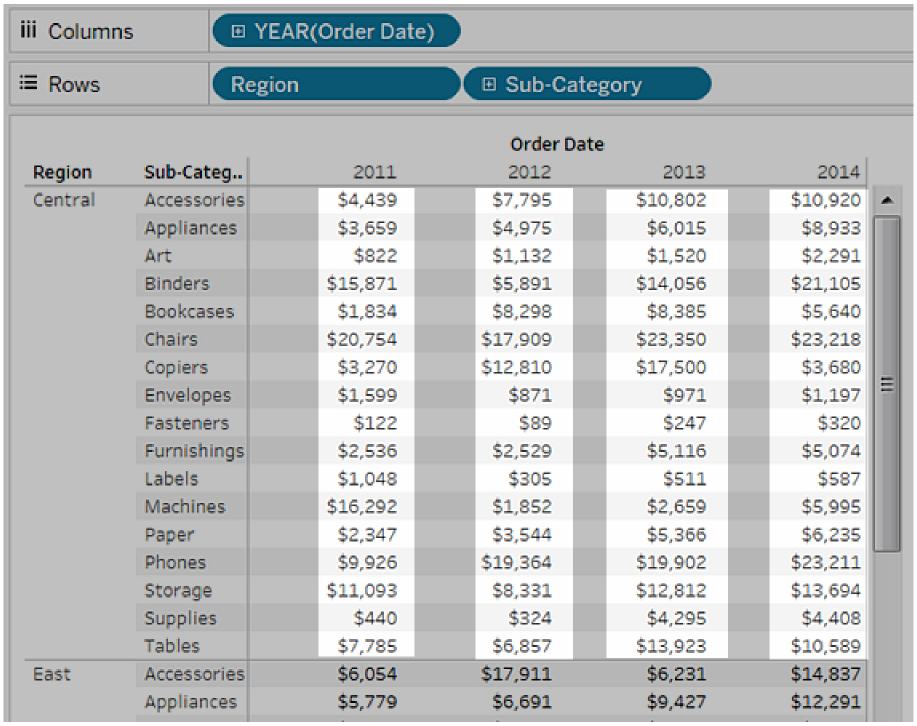
		Order Date			
Sub-Categ..	Region	2011	2012	2013	2014
Accessories	Central	$4,439	$7,795	$10,802	$10,920
	East	$6,054	$17,911	$6,231	$14,837
	South	$5,595	$4,142	$9,380	$8,160
	West	$8,926	$10,676	$15,482	$26,030
Appliances	Central	$3,659	$4,975	$6,015	$8,933
	East	$5,779	$6,691	$9,427	$12,291
	South	$2,120	$3,850	$5,607	$7,948
	West	$3,755	$7,725	$5,001	$13,754
Art	Central	$822	$1,132	$1,520	$2,291
	East	$1,290	$1,707	$1,883	$2,606
	South	$566	$1,362	$1,391	$1,337
	West	$3,380	$2,035	$1,116	$2,681
Binders	Central	$15,871	$5,891	$14,056	$21,105
	East	$6,347	$14,207	$18,956	$13,989
	South	$8,307	$13,467	$4,112	$11,143
	West	$12,963	$3,889	$12,361	$26,748
Bookcases	Central	$1,834	$8,298	$8,385	$5,640
	East	$10,863	$19,653	$5,964	$7,338
	South	$794	$1,239	$3,709	$5,157
	West	$6,545	$9,354	$8,217	$11,888

You can use a table calculation to show percentages of total instead of raw dollar values. First, you must determine how to frame the calculation.

In this case, there are three dimensions in the view: **Order Date**, **Sub-Category**, and **Region**.

You could show percentages of total for a single dimension, but that can be unwieldy. For example, if you show percentages just by region, the percentages would be calculated across the two remaining dimensions: **Sub-Category** (there are 17 sub-categories) and **Year(Order Date)** (there are 4 years). So you would be dividing the total 17 x 4 = 68 ways. That would make for some tiny percentages.

Instead, show percentages using two dimensions: **Year(Order Date)** and **Region**. Then the percentages are calculated on the remaining dimension, **Sub-Category**, that is, you calculate percent of total within each highlighted area shown below.

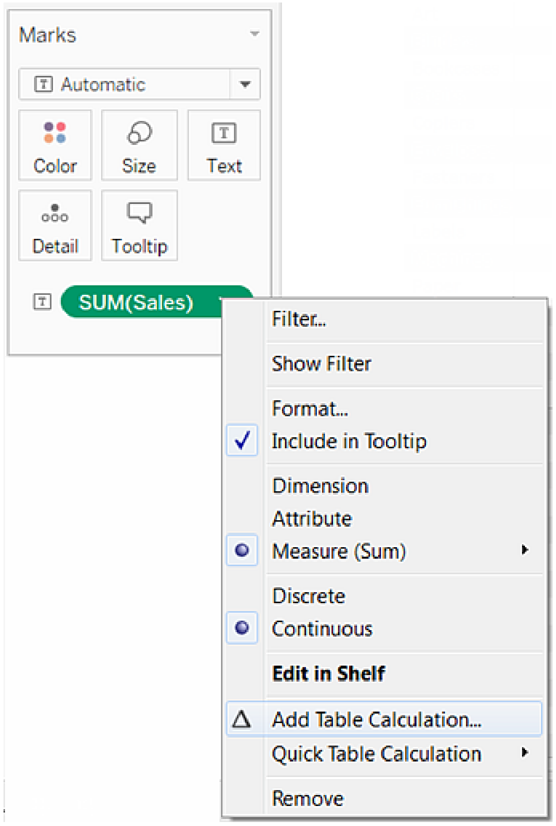


		Order Date			
Region	Sub-Categ..	2011	2012	2013	2014
Central	Accessories	$4,439	$7,795	$10,802	$10,920
	Appliances	$3,659	$4,975	$6,015	$8,933
	Art	$822	$1,132	$1,520	$2,291
	Binders	$15,871	$5,891	$14,056	$21,105
	Bookcases	$1,834	$8,298	$8,385	$5,640
	Chairs	$20,754	$17,909	$23,350	$23,218
	Copiers	$3,270	$12,810	$17,500	$3,680
	Envelopes	$1,599	$871	$971	$1,197
	Fasteners	$122	$89	$247	$320
	Furnishings	$2,536	$2,529	$5,116	$5,074
	Labels	$1,048	$305	$511	$587
	Machines	$16,292	$1,852	$2,659	$5,995
	Paper	$2,347	$3,544	$5,366	$6,235
	Phones	$9,926	$19,364	$19,902	$23,211
	Storage	$11,093	$8,331	$12,812	$13,694
	Supplies	$440	$324	$4,295	$4,408
	Tables	$7,785	$6,857	$13,923	$10,589
East	Accessories	$6,054	$17,911	$6,231	$14,837
	Appliances	$5,779	$6,691	$9,427	$12,291

The dimensions that you use to frame your calculation are called the *addressing fields*, and the fields in which you run your calculation are the *partition fields*.

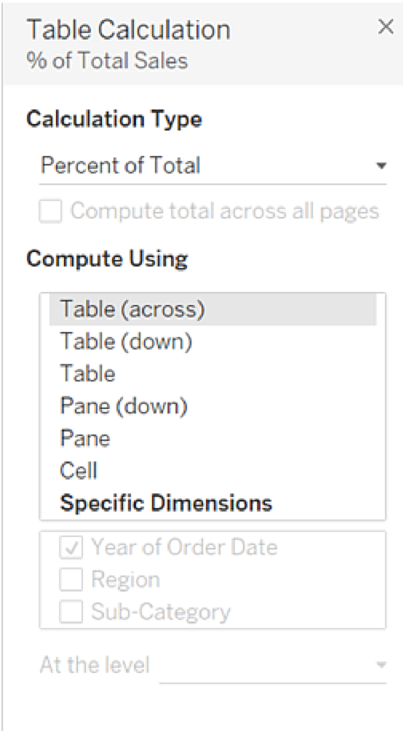
For more information about these concepts, see **The basics: addressing and partitioning** on page 2408.

6. To create a table calculation to show percentages, right-click (control-click on Mac) the **SUM(Sales)** field on the **Marks** card, and then select **Add Table Calculation**.



7. In the Table Calculation dialog box, set **Calculation Type** to **Percent of Total**.

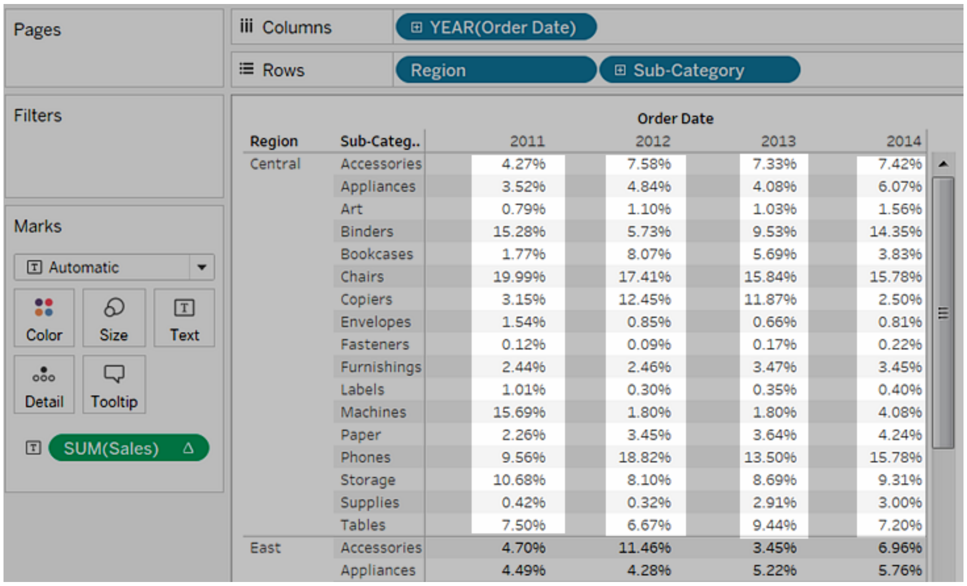
   The options in the dialog box change depending on the type of calculation you choose.



For more information about using table calculations, see **Transform Values with Table Calculations** on page 2406.

8. For the Calculation definition, select **Pane (Down)**, and then close the Table Calculation dialog box.

   Now we see percentages calculated within each sub-category, duplicated for each year within each region. The numbers within each highlighted area add up to 100%.

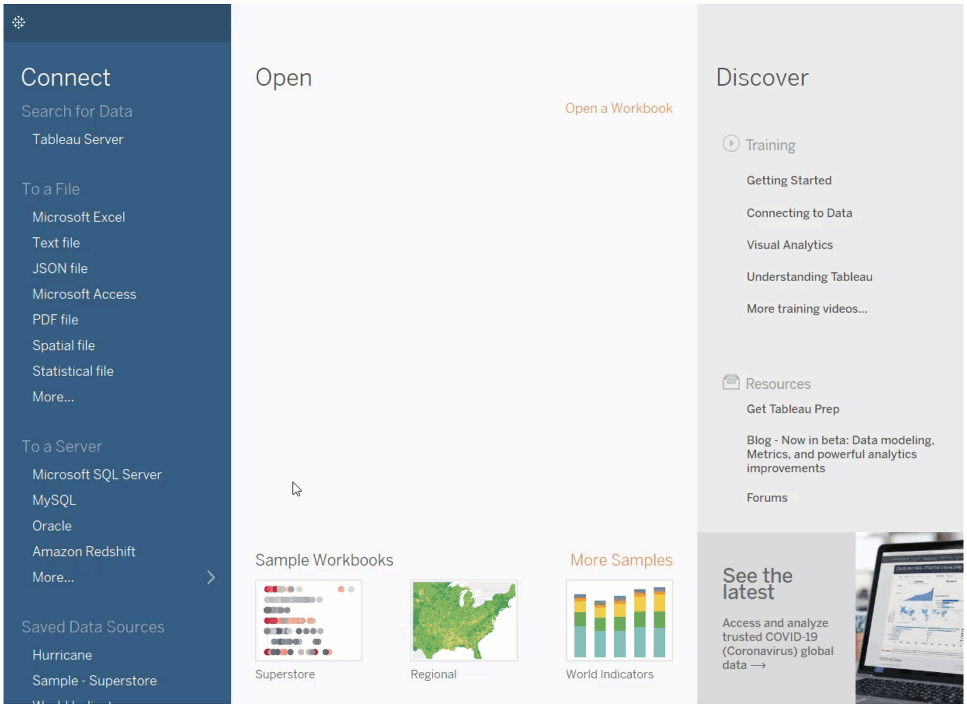


		Order Date			
Region	Sub-Categ..	2011	2012	2013	2014
Central	Accessories	4.27%	7.58%	7.33%	7.42%
	Appliances	3.52%	4.84%	4.08%	6.07%
	Art	0.79%	1.10%	1.03%	1.56%
	Binders	15.28%	5.73%	9.53%	14.35%
	Bookcases	1.77%	8.07%	5.69%	3.83%
	Chairs	19.99%	17.41%	15.84%	15.78%
	Copiers	3.15%	12.45%	11.87%	2.50%
	Envelopes	1.54%	0.85%	0.66%	0.81%
	Fasteners	0.12%	0.09%	0.17%	0.22%
	Furnishings	2.44%	2.46%	3.47%	3.45%
	Labels	1.01%	0.30%	0.35%	0.40%
	Machines	15.69%	1.80%	1.80%	4.08%
	Paper	2.26%	3.45%	3.64%	4.24%
	Phones	9.56%	18.82%	13.50%	15.78%
	Storage	10.68%	8.10%	8.69%	9.31%
	Supplies	0.42%	0.32%	2.91%	3.00%
	Tables	7.50%	6.67%	9.44%	7.20%
East	Accessories	4.70%	11.46%	3.45%	6.96%
	Appliances	4.49%	4.28%	5.22%	5.76%

Pane (Down) is the appropriate choice because it specifies that the calculation should be performed from top to bottom within each pane of the table. The table has two vertical dimensions, so **Table (Down)** would have calculated the percent of total from top to bottom for the entire table, ignoring the **Region** dimension.

The pane is always the finest level of detail for the relevant direction (across or down). If you had three dimensions on the vertical axis, you might have had to use field names to define the calculation, because only the dimension furthest to the left on the **Rows** shelf (defined as Table) and the dimension furthest to the right (defined as Pane) could be captured with the structural options.

Check your work! Watch steps 1-8 below:



Build a Treemap

Use treemaps to display data in nested rectangles. You use dimensions to define the structure of the treemap, and measures to define the size or color of the individual rectangles. Treemaps are a relatively simple data visualization that can provide insight in a visually attractive format.

The basic building blocks for a treemap are as follows:

Mark type:	Automatic or Square
Color:	Dimension or Measure
Size	Measure
Label or **Detail**:	Dimension(s)

To create a treemap that shows aggregated sales totals across a range of product categories, follow the steps below.

1. Connect to the **Sample - Superstore** data source.
2. Drag the **Sub-Category** dimension to **Columns**.

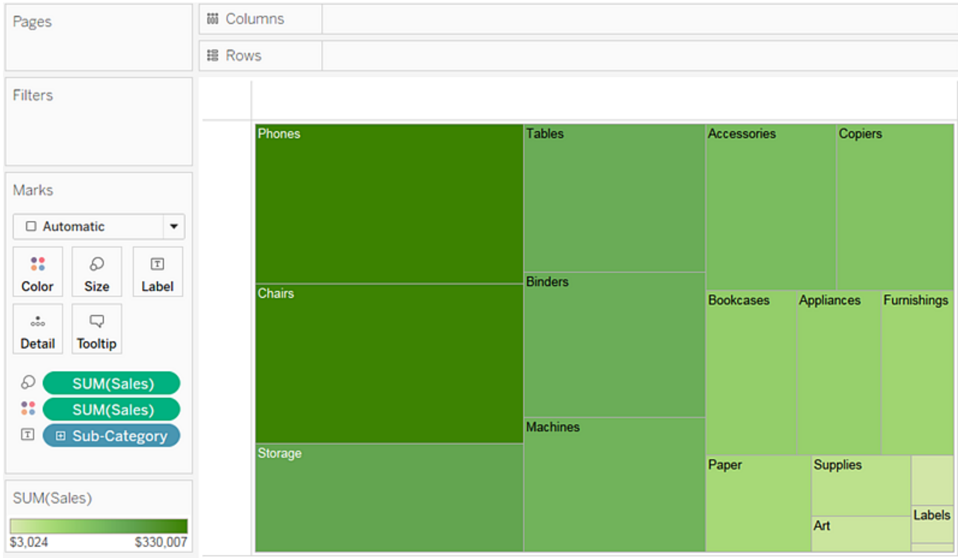
   A horizontal axis appears, which shows product categories.
3. Drag the **Sales** measure to **Rows**.

   Tableau aggregates the measure as a sum and creates a vertical axis.

   Tableau displays a bar chart—the default chart type when there is a dimension on the **Columns** shelf and a measure on the **Rows** shelf.
4. Click **Show Me** on the toolbar, then select the treemap chart type.

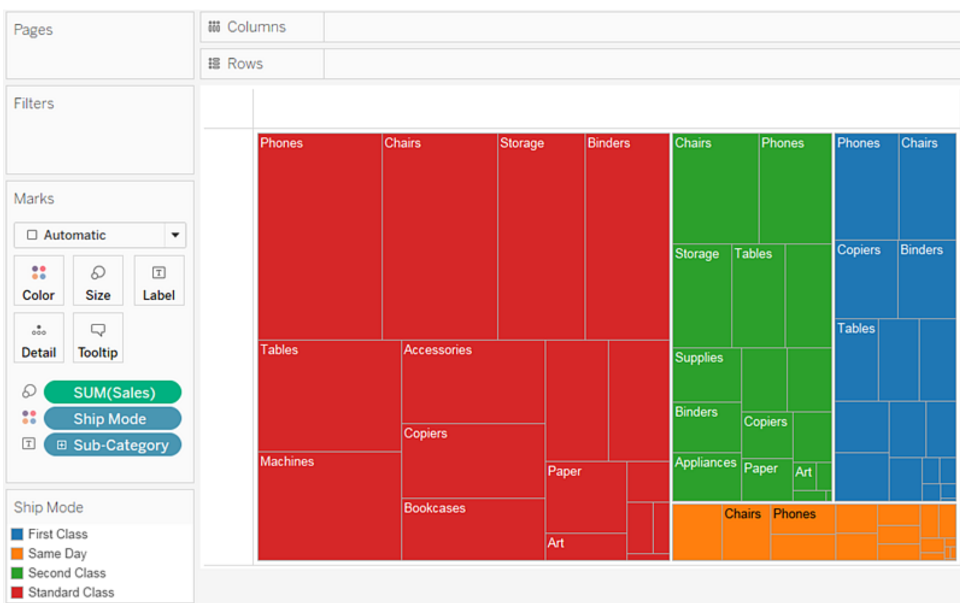
   

   Tableau displays the following treemap:



In this treemap, both the size of the rectangles and their color are determined by the value of **Sales**—the greater the sum of sales for each category, the darker and larger its box.

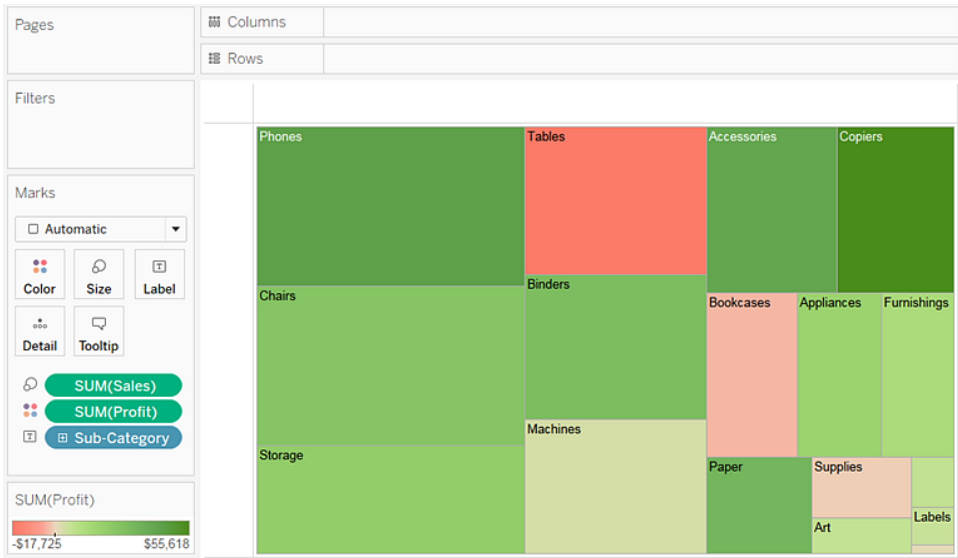
5. Drag the **Ship Mode** dimension to **Color** on the **Marks** card. In the resulting view, **Ship Mode** determines the color of the rectangles—and sorts them into four separate areas accordingly. **Sales** determines the size of the rectangles:



6. Try another option to modify the treemap: click the **Undo** button to remove **Ship Mode**

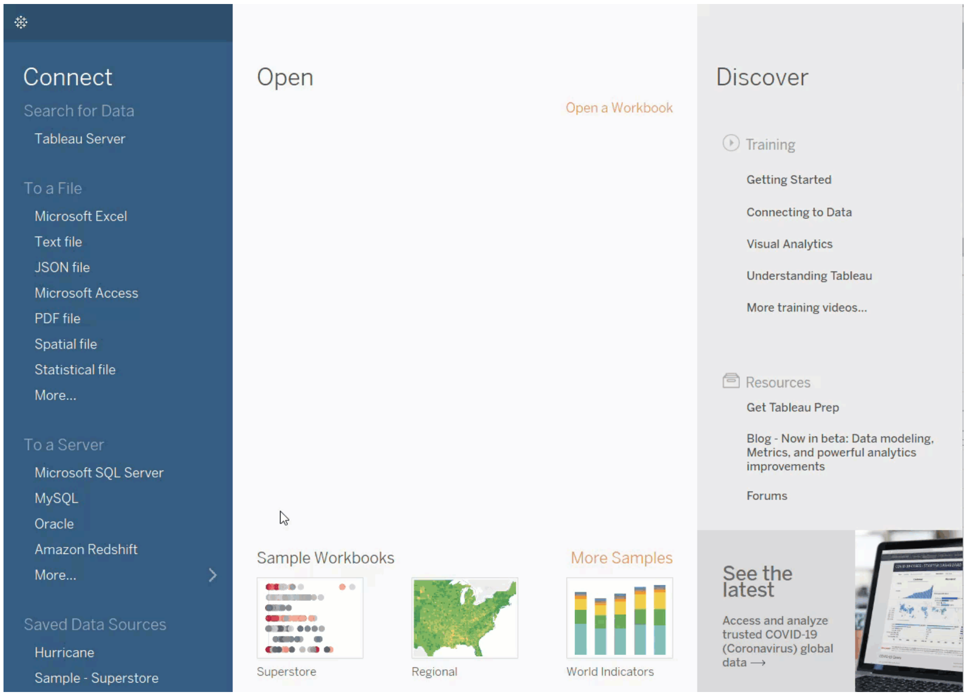
from view.

7. Drag the **Profit** measure to **Color** on the **Marks** card. Now **Profit** determines the color of the rectangles, and **Sales** determines their size:



With treemaps, **Size** and **Color** are the crucial elements. You can place measures on **Size** and **Color**, but placing a measure anywhere else has no effect. Treemaps can accommodate any number of dimensions, including one or even two on **Color**. But beyond that, adding dimensions only breaks the map into an ever greater number of smaller rectangles. For information on how to place two dimensions on **Color** in a treemap, see **Example – Multiple Fields on Color** on page 1405.

Check your work! Watch steps 1-7 below:

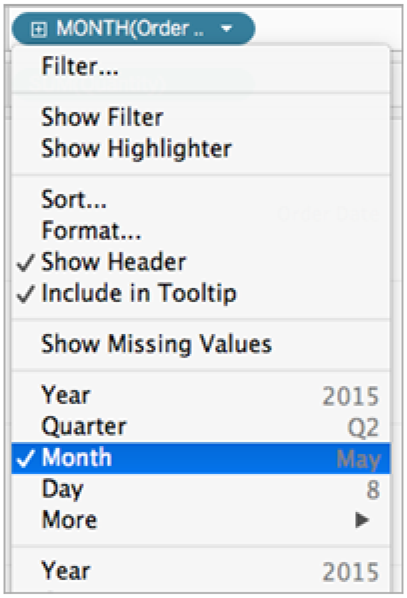


Build a Combination Chart

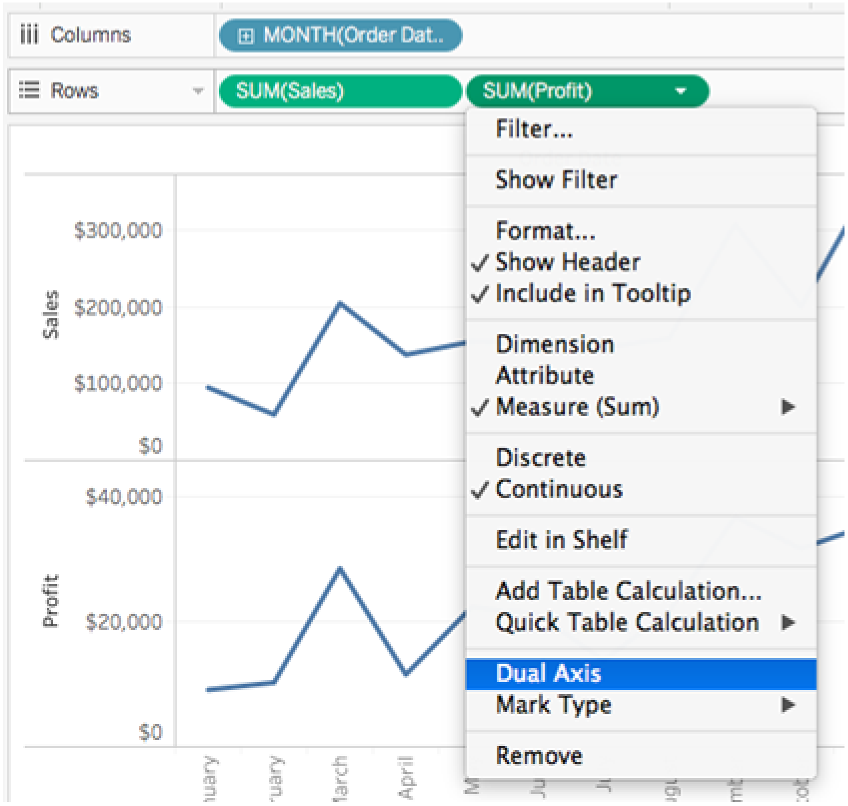
Combination charts are views that use multiple mark types in the same visualization. For example, you may show sum of profit as bars with a line across the bars showing sum of sales. You can also use combination charts to show multiple levels of detail in the same view. For example, you can have a line chart with individual lines showing average sales over time for each customer segment, then you can have another line that shows the combined average across all customer segments.

To create a combination chart, follow the steps below:

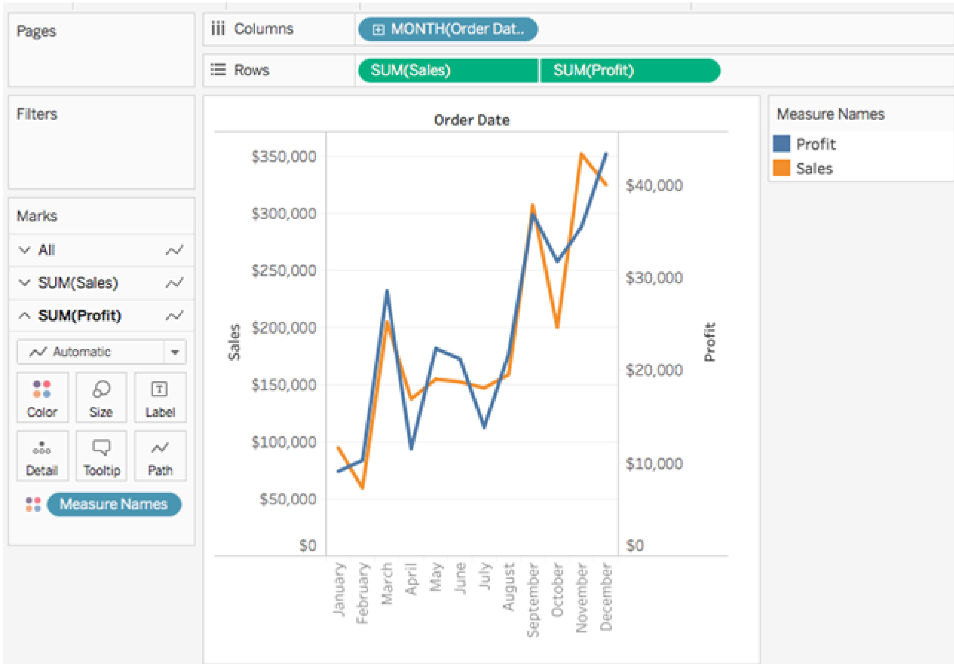
1. Open Tableau Desktop and connect to the **Sample - Superstore** data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, drag **Order Date** to the **Columns** shelf.
4. On the Columns shelf, right-click **YEAR(Order Date)** and select **Month**.



5. From the **Data** pane, drag **Sales** to the **Rows** shelf.
6. From the **Data** pane, drag **Profit** to the **Rows** shelf and place it to the right of SUM (Sales).
7. On the Rows shelf, right-click **SUM(Profit)** and select **Dual-Axis**.

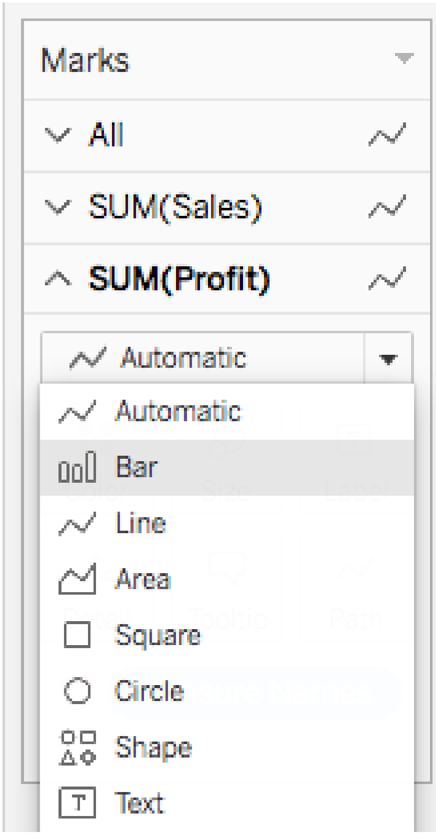


The view updates. Measure Names is added to Color on the Marks card to differentiate the lines.

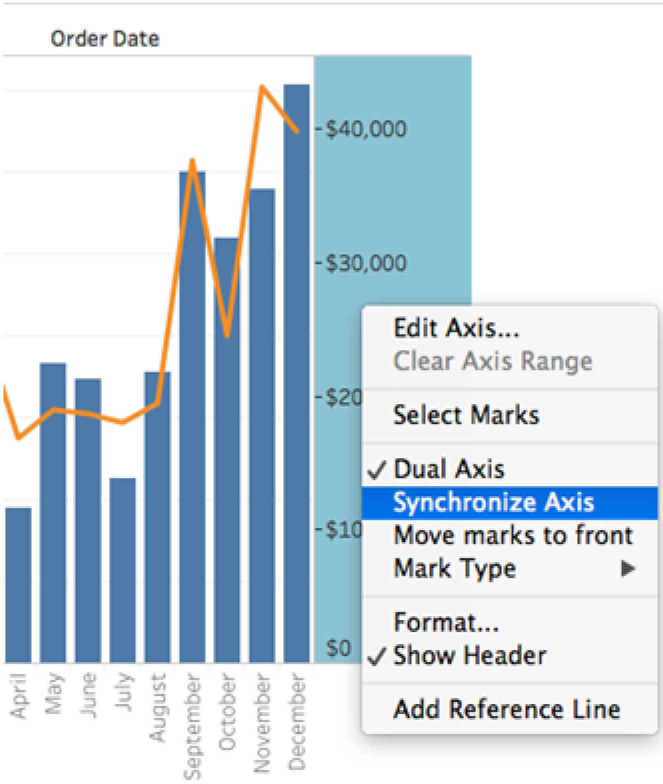


> **Note**: Some marks can be hidden behind others. To move the marks forward or backward, right-click one of the axes in the visualization and select Move Marks to Back or Move Marks to Front.

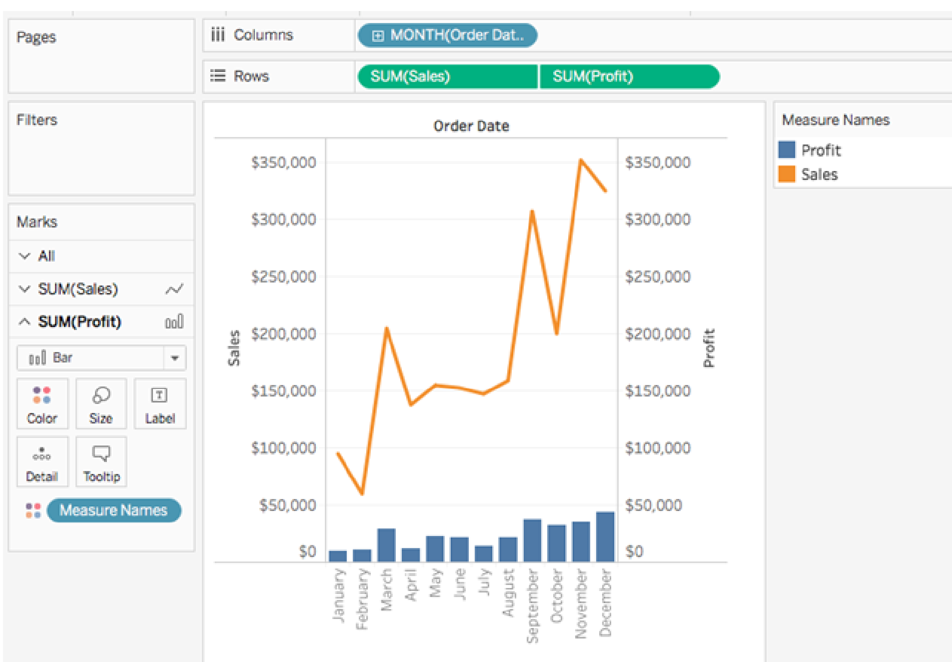
8. On the SUM(Profit) Marks card, click the Mark Type drop-down and select **Bar**.



9. In the visualization, right-click the **Profit** axis and select **Synchronize Axis**.



The view updates to look like this:



Build Advanced Chart Types

The topics in this section illustrate how to create views that include industry-standard analytical benchmarks and visualization types.

> **Note**: Most of these examples can only be built in Tableau Desktop—not as you edit views on the web. Before attempting to follow the steps in one of these topics, check the Applies To line at the top of the topic to determine whether the procedure is valid for editing on the web. If Tableau Server and Tableau Online are not listed, then the procedure will not work on the web.

Add a Calculated Column to a View

Sometimes your data may not have all the pieces you need for your analysis. A common way to handle this is to create a calculated column (also referred to as a calculated field or just a calculation) that modifies the existing data. Calculations can be created by opening the **Analysis** menu and selecting **Create Calculated Field...**

It's important to note that there are often several different ways to achieve the same results for a calculated field. Additionally, the correct value of a calculation may depend on how the data is shaped and how the visualization is set up. Make sure you understand your data structure and how the viz will be built when determining how to write a calculation.

Examples of when a calculation might be useful

Combine first and last name

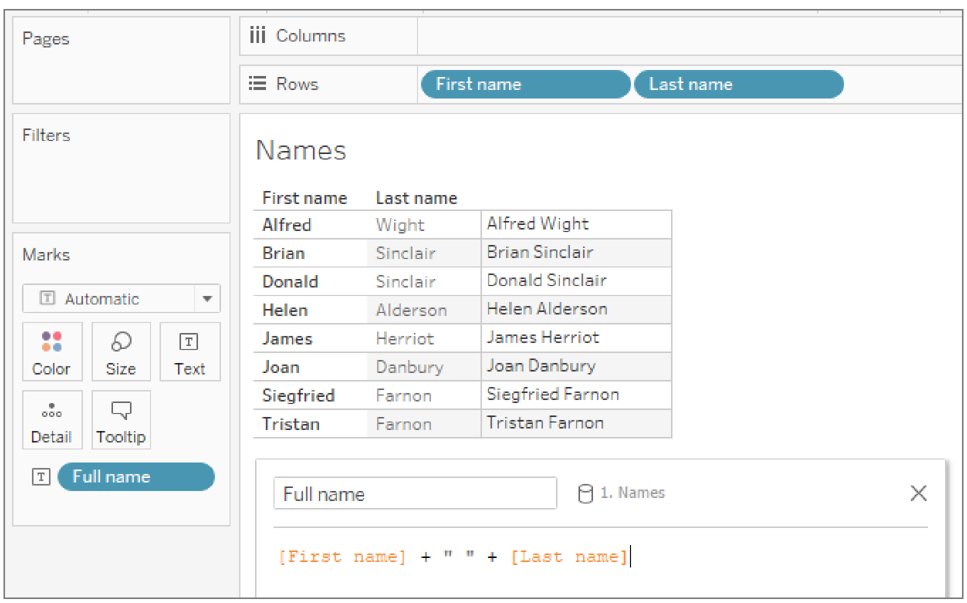
You may have First name and Last name in two different columns and want a single field for name.

Abc Names First name	Abc Names Last name	Abc Names Real person
James	Herriot	no
Helen	Alderson	no
Siegfried	Farnon	no
Tristan	Farnon	no
Donald	Sinclair	yes
Alfred	Wight	yes
Joan	Danbury	yes
Brian	Sinclair	yes

The calculation:

```
[First name] + " " + [Last name]
```

A finished viz might look like this



Find a sequence inside a string

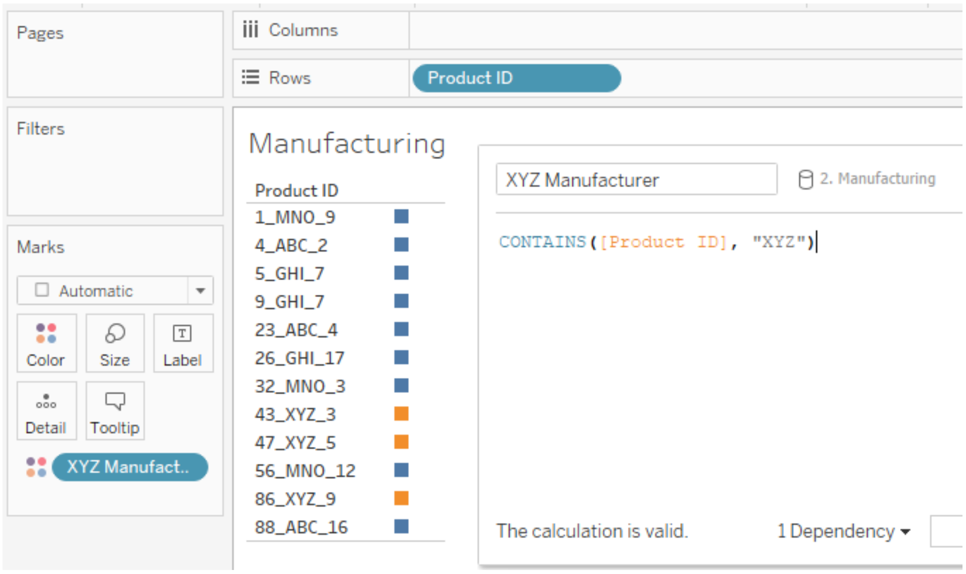
You may want to flag every record that has a certain manufacturing code XYZ in the Product ID field.

Abc Manufacturing Product ID	Abc Manufacturing Unit Cost	# Manufacturing Number on hand
4_ABC_2	$44	277
47_XYZ_5	$20	193
9_GHI_7	$30	41
56_MNO_12	$34	103
88_ABC_16	$11	67
86_XYZ_9	$43	216
26_GHI_17	$11	198
1_MNO_9	$19	213
23_ABC_4	$14	39
43_XYZ_3	$16	185
5_GHI_7	$4	252
32_MNO_3	$1	273

The calculation will return "true" if the code is present and "false" if not.

```
CONTAINS([Product ID}, "XYZ")
```

A finished viz might look like this, with the calculation on Color.



Assign categories for value ranges

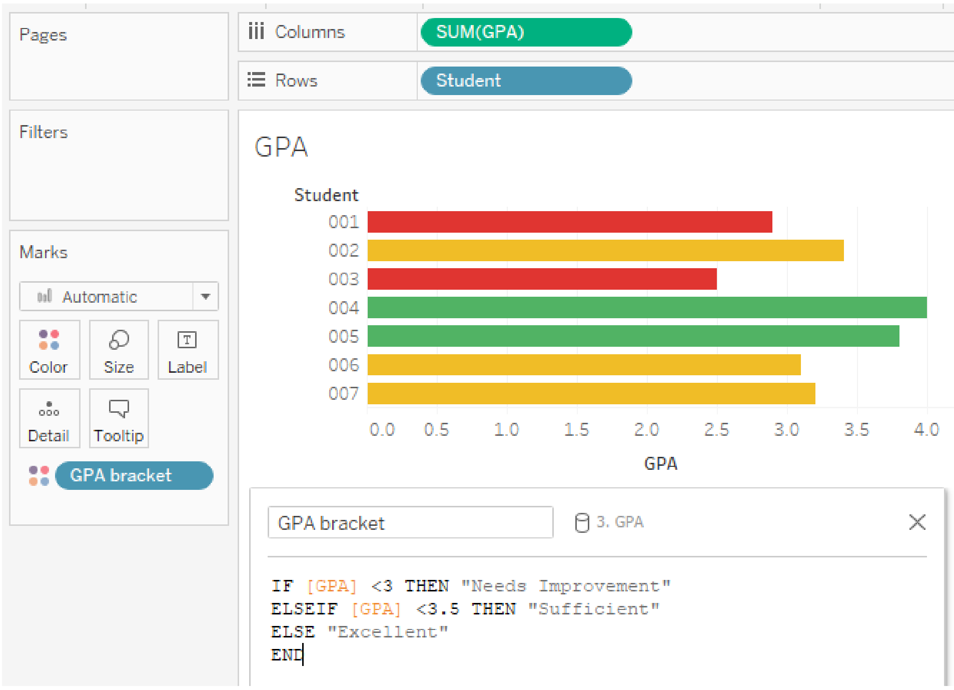
You want to assign categories based on several thresholds.

Abc Grades Student	# Grades GPA
001	2.90000
002	3.40000
003	2.50000
004	4.00000
005	3.80000
006	3.10000
007	3.20000

The calculation will evaluate each student's GPA against the cutoff value and assign a label.

```
IF [GPA] <3 THEN "Needs Improvement" ELSEIF [GPA] <3.5 THEN
"Sufficient" ELSE "Excellent" END
```

A finished viz might look like this, with the calculation on Color.



Calculate percent change

You want to display the year-over-year (YOY) percentage change in the weight of children from age 1 to age 2.

Abc Babies Baby	# Babies Age	# Babies Weight
A	1	19.3600
A	2	26.4000
B	1	22.6600
B	2	30.1400
C	1	17.3800
C	2	24.1000
D	1	21.3400
D	2	24.2000

This calculation is for the *difference/original* for each child.

```
(SUM((IF [Age] = 2 THEN [Weight] ELSE 0 END)) - SUM((IF [Age] = 1
THEN [Weight] ELSE 0 END)))/SUM(IF [Age] = 1 THEN [Weight] ELSE 0
END)
```

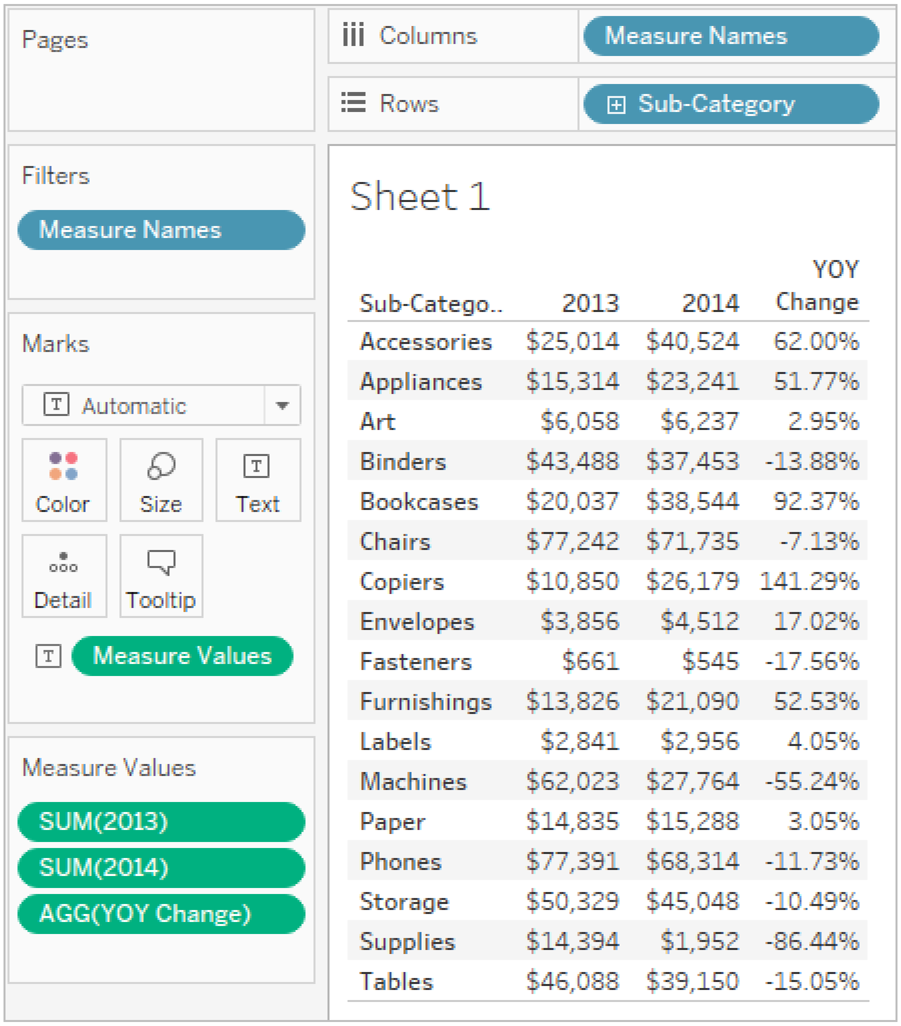
A finished viz might look like this.

This example is one where the calculation could change quite a lot based on the structure of the data. If there was a field for weight at age 1 and weight at age 2, for example, the calculation would only need to be ([Weight at age 2] - [Weight at age 1]) / [Weight at age 1]. However, the data structure detailed above is more standard and more likely to be encountered.

Follow along: create a YOY calculation

Objective

To create a view like the following, that shows sales results for two years in the first two columns, and then the year-over-year change, as a percentage, in the third column. The scenario uses the **Sample - Superstore** data source provided with Tableau Desktop to show how to build the viz.



Sub-Catego..	2013	2014	YOY Change
Accessories	$25,014	$40,524	62.00%
Appliances	$15,314	$23,241	51.77%
Art	$6,058	$6,237	2.95%
Binders	$43,488	$37,453	-13.88%
Bookcases	$20,037	$38,544	92.37%
Chairs	$77,242	$71,735	-7.13%
Copiers	$10,850	$26,179	141.29%
Envelopes	$3,856	$4,512	17.02%
Fasteners	$661	$545	-17.56%
Furnishings	$13,826	$21,090	52.53%
Labels	$2,841	$2,956	4.05%
Machines	$62,023	$27,764	-55.24%
Paper	$14,835	$15,288	3.05%
Phones	$77,391	$68,314	-11.73%
Storage	$50,329	$45,048	-10.49%
Supplies	$14,394	$1,952	-86.44%
Tables	$46,088	$39,150	-15.05%

Create the necessary calculated fields

1. Connect to the **Sample - Superstore** data source.

2. Choose **Analysis** > **Create Calculated Field** to open the calculation editor. Name the calculation **2013** and type or paste the following in the formula area:

   ```
   IF YEAR([Order Date]) = 2013 THEN [Sales] ELSE 0 END
   ```

3. Create a second calculated field and name it **2014**; the formula is the same except with 2014 instead of 2013:

   ```
   IF YEAR([Order Date]) = 2014 THEN [Sales] ELSE 0 END
   ```

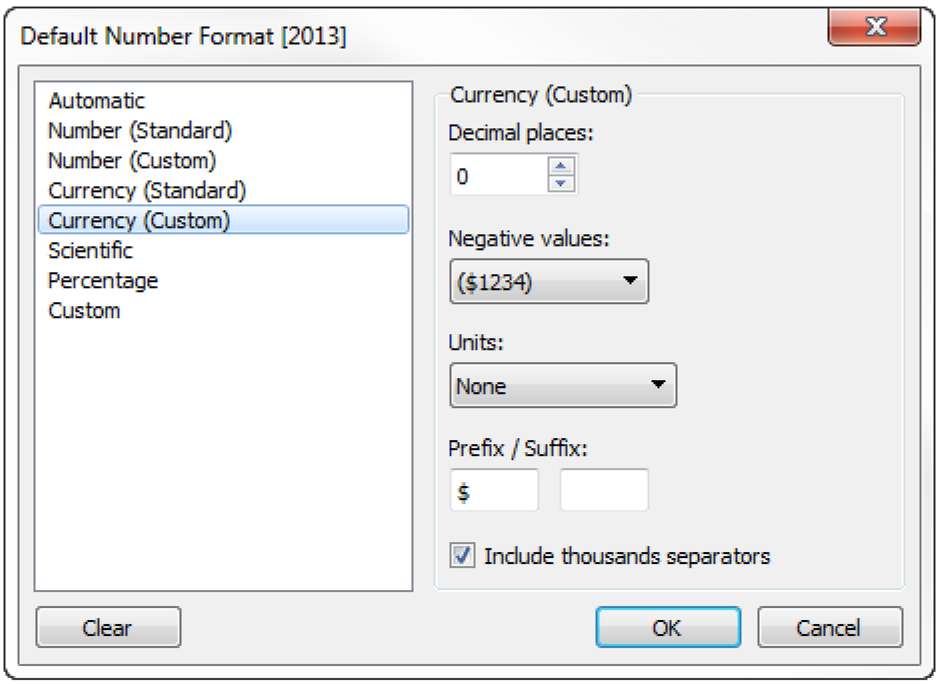
4. Create a third calculated field, **YOY Pct. Change**, to calculate the change from 2013 to

2014:

```
(SUM([2014]) - SUM([2013])) / SUM([2013])
```

Format the calculated fields

1. Click **2013** in the Measures area of the **Data** pane and choose **Default Properties** > **Number Format**.
2. In the Default Number Format dialog box, set the format to **Currency (Custom)** and the **Decimal places** to **0**.



3. Format the **2014** measure the exact same way.
4. Format the **YOY Pct. Change** field as **Percentage**, with **2** decimal places.

Build the view

1. Drag **2013** to Text on the Marks card.
2. Double-click **2014** and then **YOY Pct. Change**.
3. Drag **Measure Names** from **Rows** to **Columns**.

4. Drag **Sub-Category** to **Rows**.

   Your view should now look like the one at the beginning of this section.

Calculate Z-scores

In statistics, the z-score (or standard score) of an observation is the number of standard deviations that it is above or below the population mean.

To calculate a z-score you must know the population mean and the population standard deviation. In cases where it is impossible to measure every observation of a population, you can estimate the standard deviation using a random sample.

Create a z-score visualization to answer questions like the following:

- What percentage of values fall below a specific value?
- What values can be considered exceptional? For example, in an IQ test, what scores represent the top five percent?
- What is the relative score of one distribution versus another? For example, Michael is taller than the average male and Emily is taller than the average female, but who is relatively taller within their gender?

As a general rule, z-scores lower than -1.96 or higher than 1.96 are considered unusual and interesting. That is, they are statistically significant outliers.

This article demonstrates how to calculate a z-score in Tableau.

1. Connect to the **Sample - Superstore** data source provided with Tableau Desktop.
2. Create a calculated field to calculate average sales.

   Choose **Analysis** > **Create Calculated Field** to open the calculation editor. Name the calculation **Average Sales** and type or paste the following in the formula area:

   ```
   WINDOW_AVG(SUM([Sales]))
   ```

3. Create another calculated field to calculate the standard deviation. Name the calculation **STDEVP Sales** and type or paste the following in the formula area:

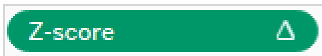
   ```
   WINDOW_STDEVP(SUM([Sales]))
   ```

4. Create one more calculated field, this one to calculate the z-score. Name the calculation **Z-score** and type or paste the following in the formula area:

   ```
   (SUM([Sales]) - [Average Sales]) / [STDEVP Sales]
   ```

5. Drag **Z-Score** from the **Data** pane to **Columns** and **State** to **Rows**.

   Notice that the **Z-score** field on Columns has a table calculation icon on the right side (that is, a small triangle):

   The **STDEVP Sales** function is based on the **WINDOW_STDEVP** function, which is a table calculation function. The **Z-Score** function, in turn, is a table calculation function because it includes **STDEVP Sales** in its definition. When you use a calculated field that includes a table calculation function in a view, it's the same as adding a table calculation to a field manually. You can edit the field as a table calculation. In fact, that's what you do next.

6. Click the **Z-score** field on Columns and choose **Compute Using** > **State**.

   This causes the z-scores to be computed on a per-state basis.

7. Click the **Sort Descending** icon on the toolbar:

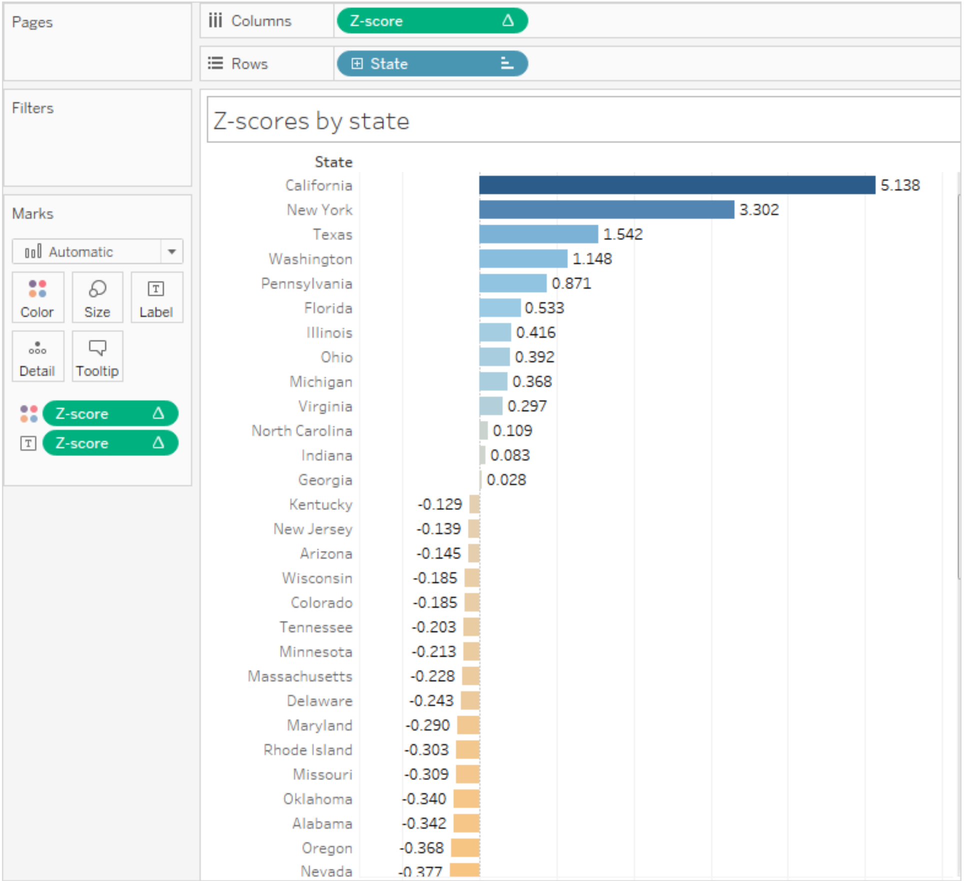
   

8. Hold down the Ctrl key and drag the **Z-score** field from **Columns** to **Color**.

   Ctrl + Drag copies a field as currently configured to an additional location.

9. Ctrl + Drag **Z-score** from Columns once again. This time drop it on **Label**.

You now have a distribution of z-scores broken out by state. California and New York both have z-scores greater than 1.96. You could conclude from this that California and New York have significantly higher average sales than other states.



Visualize Key Progress Indicators

This article shows how to create a view that shows Key Progress Indicators (KPIs). A Key Performance Indicator is a measurable value that shows how effectively a company is achieving key business objectives. At a high level, the procedure requires you to do the following:

1. Create a view that includes the field or fields (measures) you want to assess.
2. Create a calculated field that establishes the threshold that demarcates success from failure.
3. Update the view to use KPI-specific shape marks to show which values are above the threshold and which are below.

> **Note**: When connected to Microsoft Analysis Services, any KPI calculations defined in the cube are not available in Tableau. But as the procedure below demonstrates, you

> can write your own KPI calculations directly inside Tableau, and then use Tableau parameters to create highly flexible what-if KPI analysis. For more on special considerations for cube data sources, see **Cube Data Sources** on page 1085.

The scenario uses the **Sample - Superstore** data source provided with Tableau Desktop to show how to build a KPI view that displays a green check mark for any sales figure over $25,000, and a red X for any sales figure under $25,000.

Create a view that includes the field you want to assess

In this case, that field is **Sales**.

1. Connect to the **Sample - Superstore** data source.
2. From the Data pane, drag **Sub-Category** to **Rows** and **Region** to **Columns**.
3. From the Data pane, drag **Sales** to **Text** on the Marks card.

Create a calculated field that establishes the threshold that demarcates success from failure

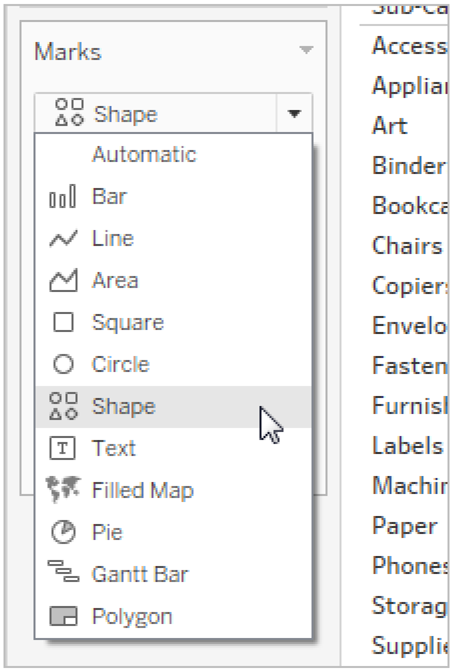
1. In the **Analysis** menu, select **Create Calculated Field** to open the calculation editor. Name the calculation KPI and type or paste the following in the formula area

   ```
   IF SUM ([Sales]) > 25000 THEN "Above Benchmark" ELSE "Below
   Benchmark" END
   ```

2. Click **OK**.

Update the view to use KPI-specific shape marks

1. On the Marks card, select Shape from the drop-down list of views:



2. Drag the **KPI** field from the **Measures** area of the **Data** pane to **Shape** on the Marks card.

3. Click **Shape** on the Marks card to open the **Edit Shape** dialog box.

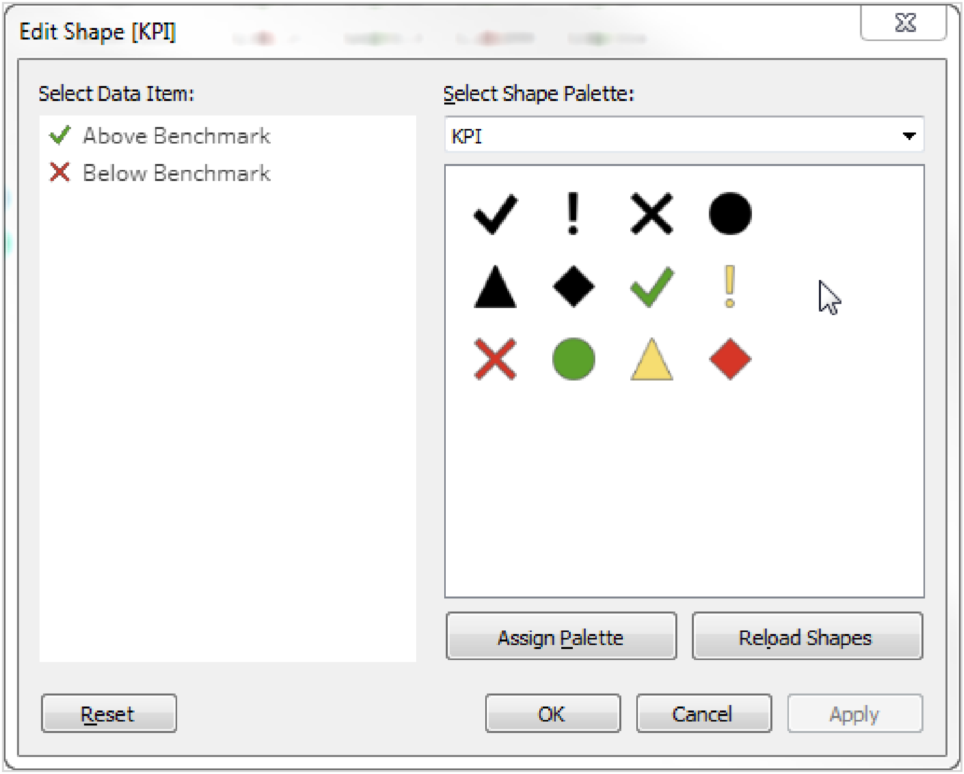
4. From the **Select Shape Palette** drop down list, choose **KPI**.

   Now you are ready to associate specific values for the KPI field with specific shapes.

5. Click **Above Benchmark** under **Select Data Item** and then click the green check mark in the palette.

6. Click **Below Benchmark** under **Select Data Item** and then click the red X in the palette.

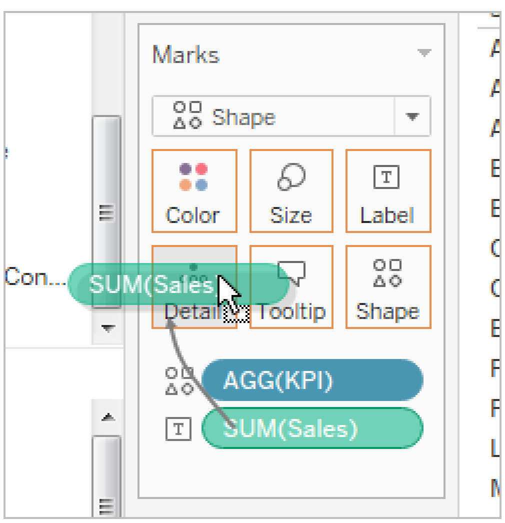
   The Edit Shape dialog box should now look like this:



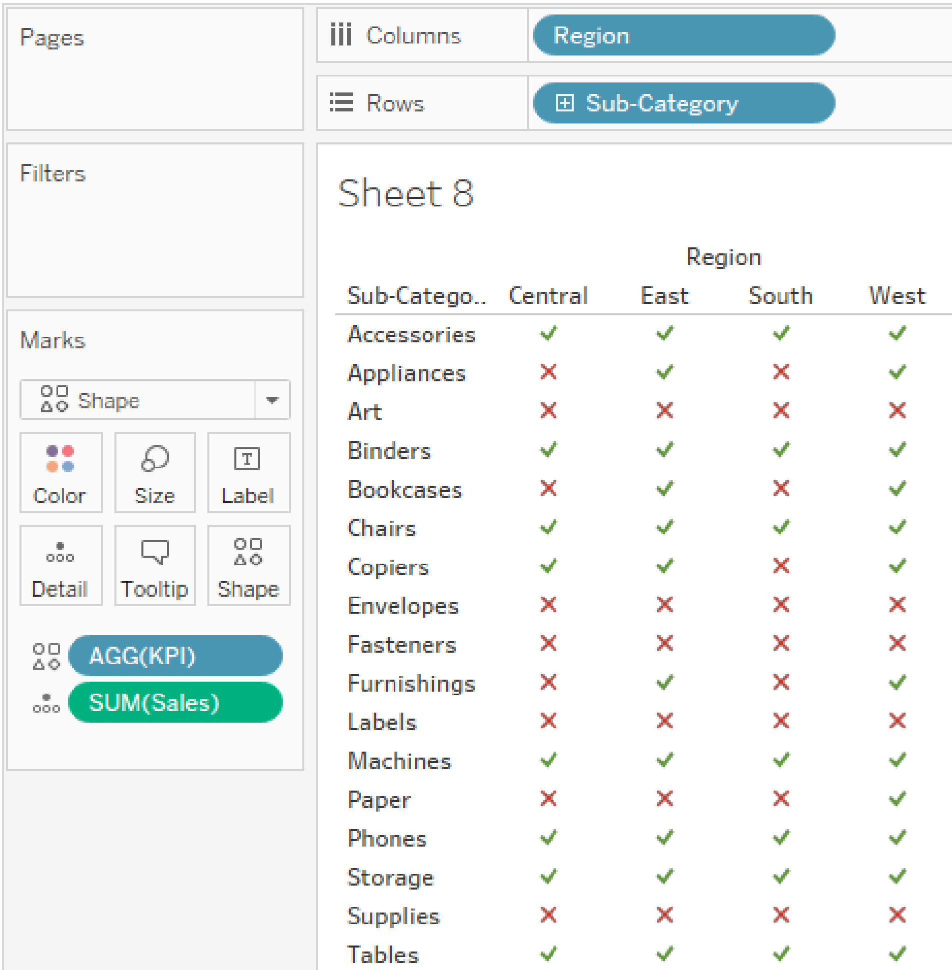
7. Click **OK** to close the Edit Shape dialog box.

   The shapes in the view show the correct indicators. Now you just need to hide the sales numbers.

8. Drag **SUM(Sales)** on the Marks card to **Detail**.



You now have a completed view that show how individual products (sub-categories) are performing across all four regions:



The view may not be terribly exciting on its own, but it would make a nice addition to a dashboard that showed other performance metrics.

Create a Pareto Chart

A Pareto chart is a type of chart that contains both bars and a line graph, where individual values are represented in descending order by bars, and the ascending cumulative total is represented by the line. It is named for Vilfredo Pareto, an Italian engineer, sociologist, economist, political scientist, and philosopher, who formulated what has become know as the Pareto principal. Pareto made the observation that 80% of land was typically owned by 20% of the population. Pareto extended his principle by observing that 20% of the peapods in his garden contained 80% of the peas. Eventually, the principal was further extrapolated by others to propose that for many events, roughly 80% of the effects come from 20% of the causes. In business, for example, 80% of profits not infrequently derive from 20% of the available products.

In Tableau, you can apply a table calculation to sales data to create a chart that shows the percentage of total sales that come from the top products, and thus identify the key segments of your customer base that are most important for your business's success.

The procedure uses the **Sample - Superstore** data source provided with Tableau Desktop.

Preparing for the analysis

Before starting your analysis, decide what questions you want answered. These questions determine the category (dimension) and number (measure) on which to base the analysis. In the example to follow, the question is which products (captured by the **Sub-Category** dimension) account for the most total sales.

At a high level, the process requires you to do the following:

1. Create a bar chart that shows **Sales** by **Sub-Category**, in descending order.
2. Add a line chart that also shows **Sales** by **Sub-Category**.
3. Add a table calculation to the line chart to show sales by Sub-Category as a **Running Total**, and as a **Percent of Total**.

The scenario uses the **Sample - Superstore** data source provided with Tableau Desktop

Create a bar chart that shows Sales by Sub-Category in descending order

1. Connect to the **Sample - Superstore** data source.
2. From the **Data** pane, drag **Sub-Category** to **Columns**, and then drag **Sales** to **Rows**.
3. Click **Sub-Category** on **Columns** and choose **Sort**.

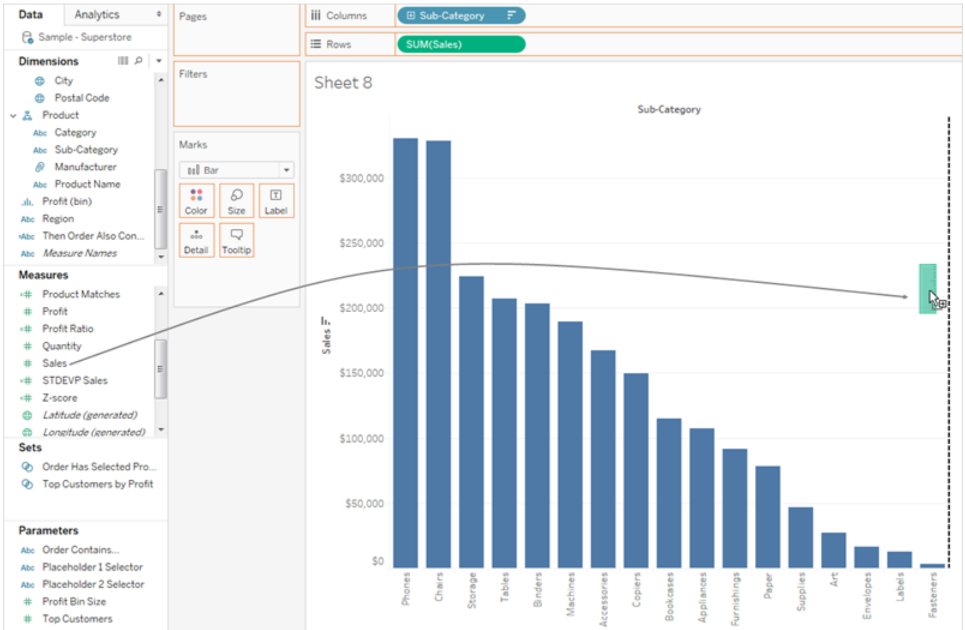
   In the Sort dialog box, do the following:

   a. Under **Sort order**, choose **Descending**.
   b. Under **Sort by**, choose **Field**.
   c. Leave all other values unchanged, including **Sales** as the selected field and **Sum** as the selected aggregation.
   d. Click **OK** to exit the Sort dialog box.

Products are now sorted from highest sales to lowest.

Add a line chart that also shows Sales by Sub-Category

1. From the Data pane, drag **Sales** to the far right of the view, until a dotted line appears.

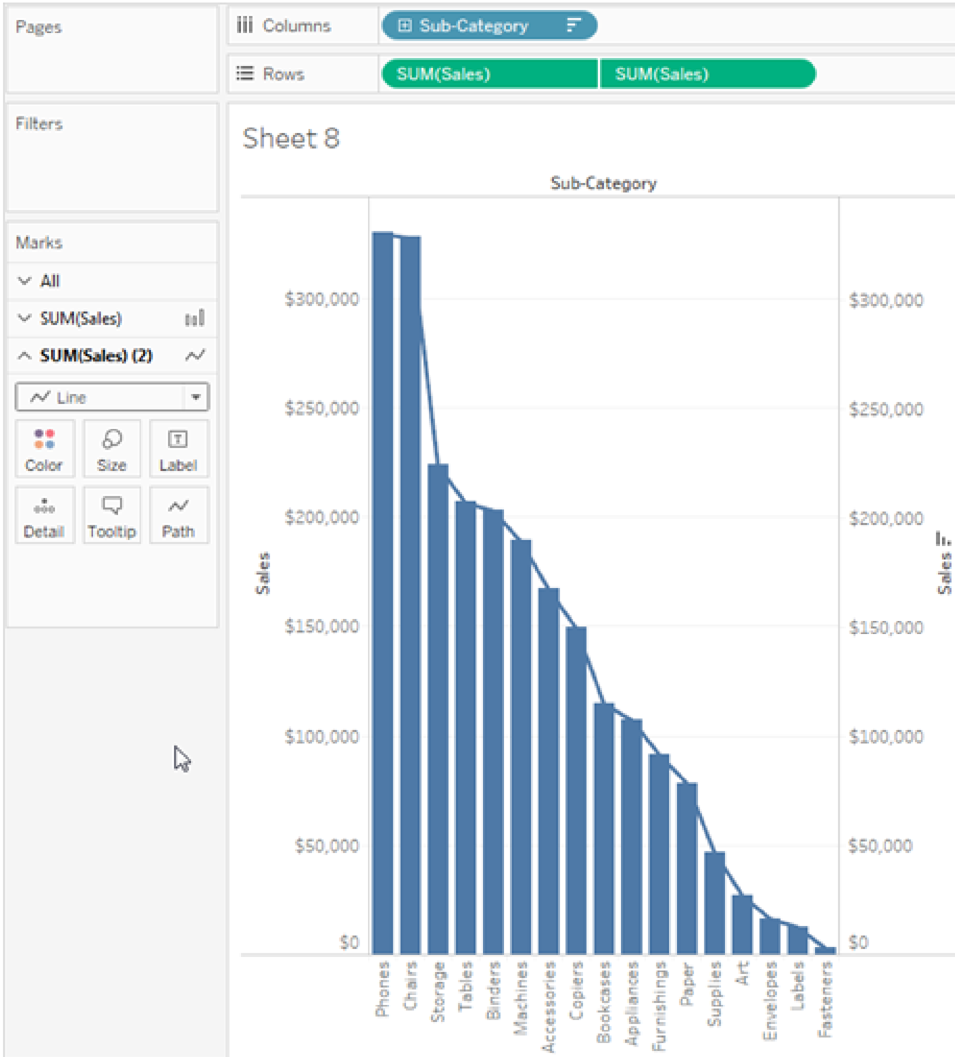


Note: In Tableau 2020.2 and later, the Data pane no longer shows Dimensions and Measures as labels. Fields are listed by table or folder.

2. Drop Sales, to create a dual-axis view. It's a bit hard to see that there are two instances of the Sales bars at this point, because they are configured identically.

3. Select **SUM(Sales) (2)** on the Marks card, and change the mark type to **Line**.

This is what the view should look like at this point:



Add a table calculation to the line chart to show sales by Sub-Category as a running total, and as a percent of total

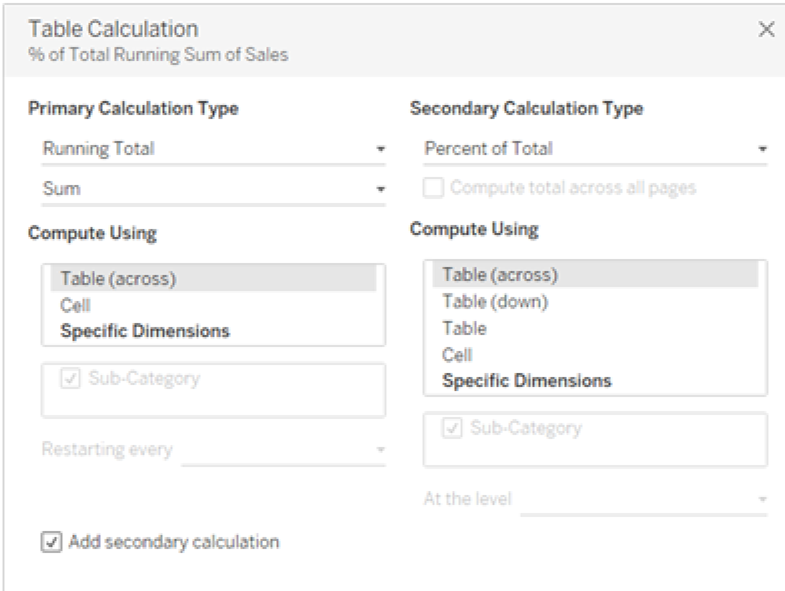
1. Click the second copy of **SUM(Sales)** on **Rows** and choose **Add Table Calculation**.
2. Add a primary table calculation to **SUM(Sales)** to present sales as a running total.

   Choose **Running Total** as the **Calculation Type**.

   Do not close the Table Calculation dialog box.
3. Add a secondary table calculation to present the data as a percent of total.

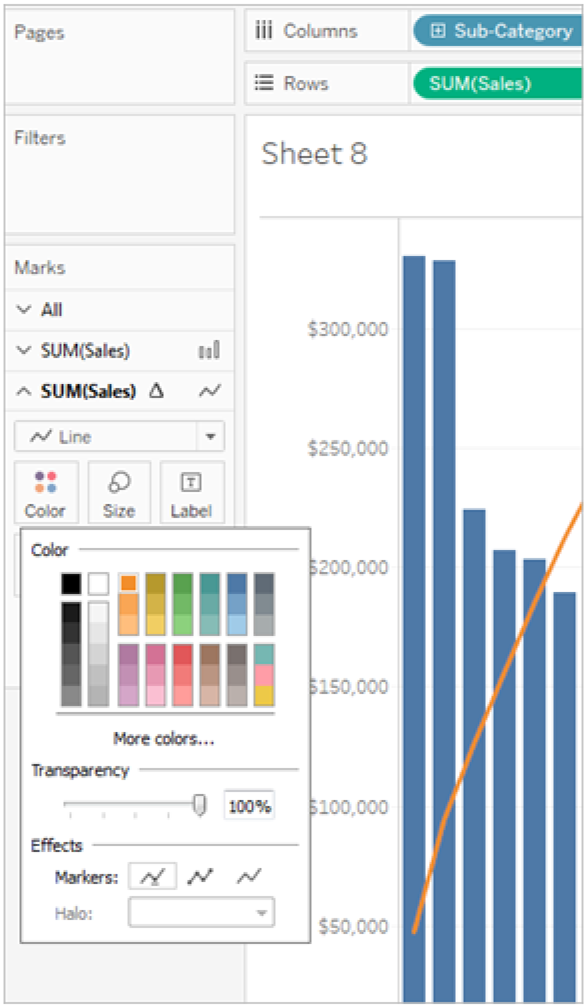
Click **Add Secondary Calculation** and choose **Percent of Total** as the **Secondary Calculation Type**.

This is what the Table Calculation dialog box should look like at this point:

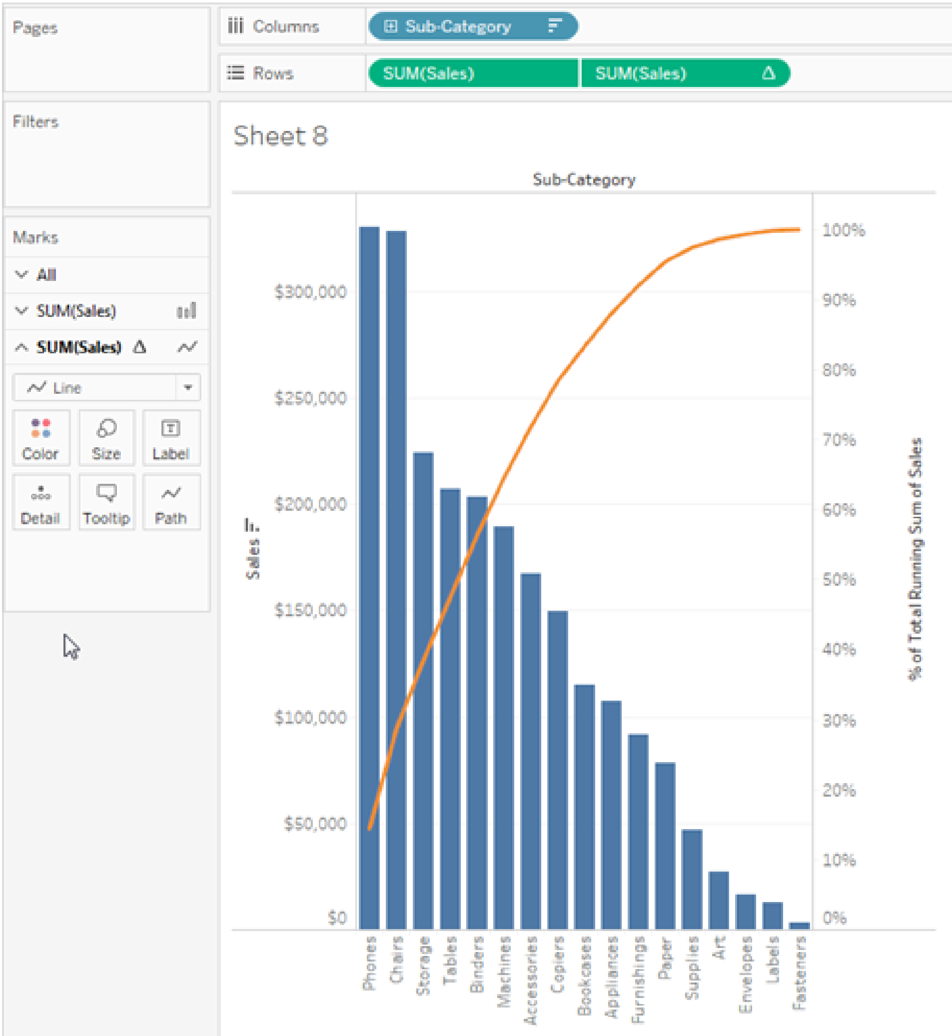


4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

5. Click Color in the Marks card to change the color of the line.



The result is now a Pareto chart:



Additional information

For additional tips on how you would compare the percentage of sales with the percentage of products, or draw reference lines that help make the 80-20 principles more apparent, see the Tableau On-Demand Training video titled Pareto.

Create a Population Pyramid

A population pyramid, also known as an age structure diagram, shows the distribution of various age groups in a population.

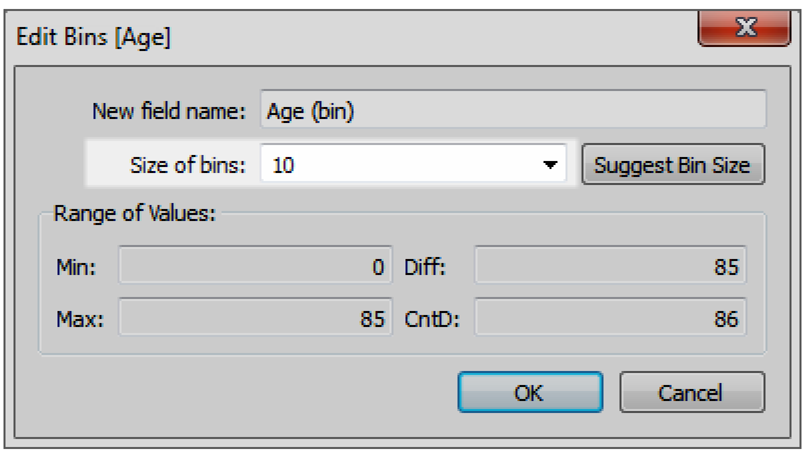
A common distribution often used with this type of visualization is female and male populations by age. To create a population pyramid using Tableau, first separate the population (measure) into two groups, females and males, and then create "bins" for the age cohorts you want to represent in the population pyramid.

For example, suppose you are working with a table from the United States Census Bureau that contains population, gender, and age data.

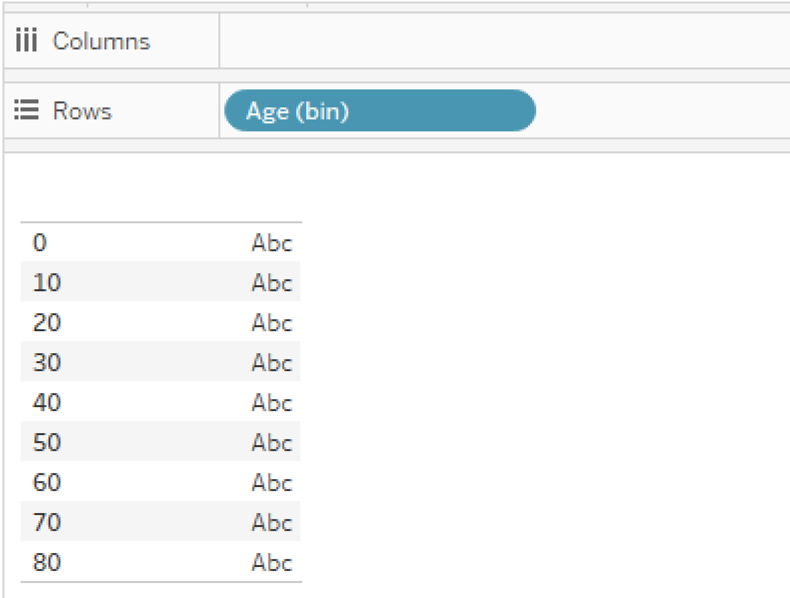
To create a bin and divide a measure into two groups.

1. Download and open the following workbook from Tableau Public.
    1. Go to https://public.tableau.com/profile/tableau.docs.team#!/vizhome/CreateaPopulationPyramid/Sheet1.
    2. Click **Download Workbook** in the upper-right corner and then open the workbook.
2. Select **Worksheet** > **Clear** > **Sheet**.
3. In the Data pane, right-click the **Age** field and select **Create** > **Bins**.
4. In the Create Bins dialog box, enter a bin size based on the age groups you're interested in focusing on, and then click **OK**.

   In this example, the bin size is 10. This means that the age cohorts are defined by increments of 10 years.



5. Drag the bin you just created to the Rows shelf.



6. Select **Analysis** > **Create a Calculated Field**, and then do the following:

   - Enter a name for the calculation. For this example, enter **Male Population**.
   - Enter a formula similar to the following to isolate the male constituent from population:

     ```
     IF [Gender] = 1 THEN [ESTBASE2010] END
     ```

     In this case, the census data has defined the Gender value for male as "1." The field "ESTBASE2010" contains estimated population values.

7. Similar to the step 5, select **Analysis** > **Create a Calculated Field**, and then do the following:

   - Enter a name for the calculation. For this example, enter **Female Population**.
   - Enter a formula similar to the following to isolate the female constituent from population:

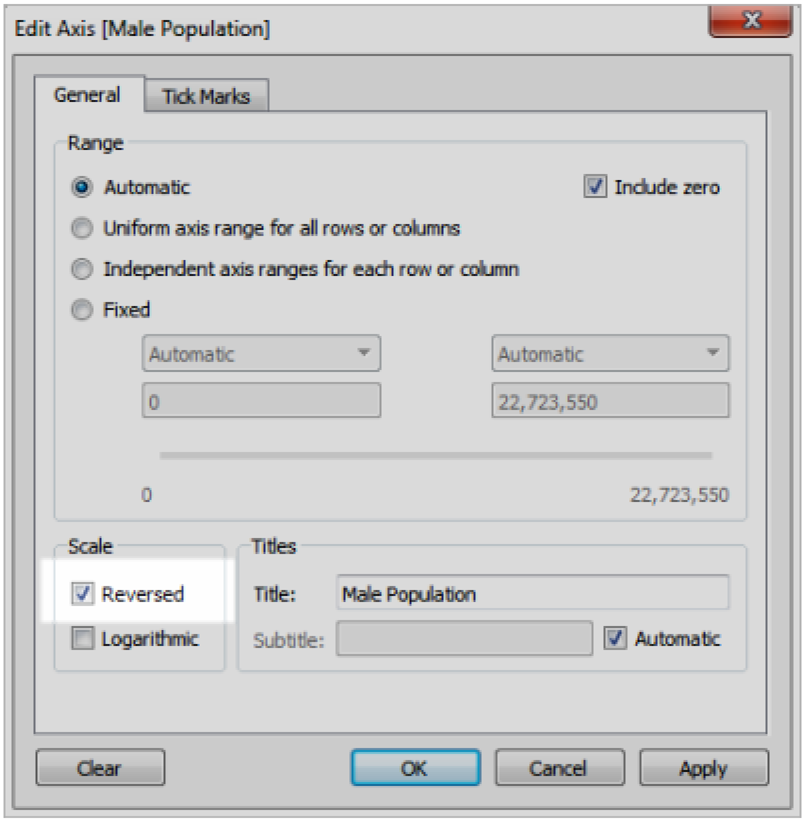
     ```
     IF [Gender] = 2 THEN [ESTBASE2010] END
     ```

     In this case, the census data has defined the Gender value for female as "2." The field "ESTBASE2010" contains estimated population values.

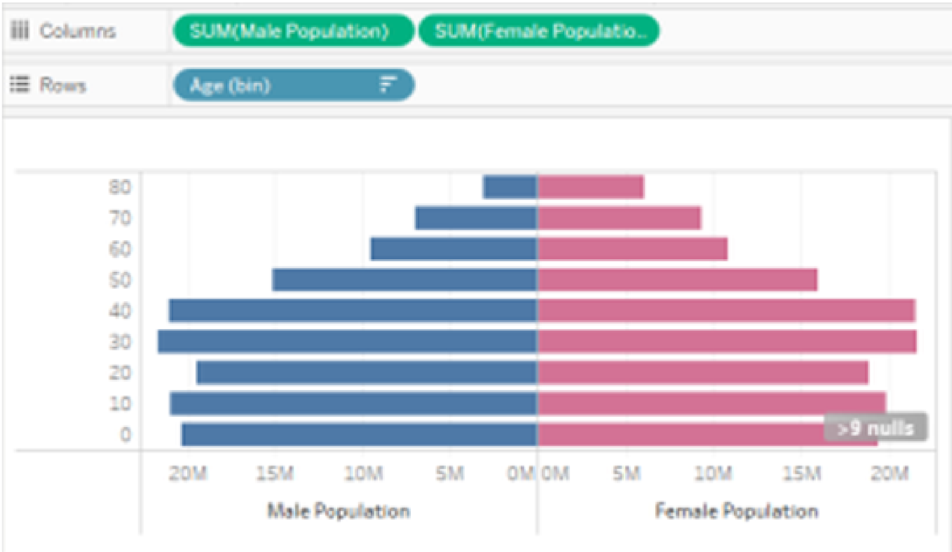
8. Drag the calculated fields you created to the Columns shelf and the **Gender** field to the Color shelf.

Note: To change the color assignments, click the Color shelf and then click **Edit Colors**.

9. Right-click the axis for Male Population, select **Edit Axis**, and then select the check box for **Reversed** to reverse the order the values display on the axis, and then click **OK**.



After changing the sort order of the bin, the population pyramid looks like the following:



Create a Co-Occurrence Visualization Using a Parameter and a Set

You can create co-occurrence visualizations in Tableau. With a co-occurrence visualization, users can pick one field value and then see which other fields values it co-occurs with, and how often. One practical application of such analysis is a market basket analysis, which you can use to discover and understand customer purchasing behavior. You can use a market basket analysis to answer questions like the following:

- How many people bought both Product A and Product B?
- What other products do people who purchased Product A generally buy?
- Which other courses do students who enrolled in Course A frequently enroll in?

Follow the steps in this article to create a co-occurrence visualization (in this case, a market basket analysis view) using a parameter, calculated fields, and a set. At a high level, the steps involved are:

1. Create a parameter that you will use to dynamically modify the view based on the item you select.
2. Create calculated fields that you will use to return which items are also ordered when a particular item is ordered.
3. Create a set to determine whether an order has the item that was selected in the

parameter control.

4. Build the view to display which items are also contained in an order with the selected item.

The scenario uses the **Sample - Superstore** data source provided with Tableau Desktop. The details are covered in the following sections.

Create a Parameter

Follow these steps to create a parameter and then display a parameter control that users can use to pick a **Sub-Category** value.

1. Connect to the **Sample - Superstore** data source.
2. Right-click (control-click on a Mac) in the **Data** pane and select **Create** > **Parameter**.
3. In the Create Parameter dialog box, do the following:
    a. Name the parameter **Order Contains**.
    b. For the **Data Type**, choose **String**.
    c. For **Allowable Values**, choose **List**.
    d. In the **List of values** section click **Add from Field** > **Sub-Category**.
    e. Click **OK**.
4. Right-click the **Order Contains** parameter in the Data pane and choose **Show Parameter Control**.

Create Calculated Fields

Follow these steps to create the calculated fields that you will use to show which items are also ordered when the item selected by the user (via the parameter control) is ordered.

1. Create a calculated field to identify products that the order also contains (in addition to the one that the user selects).

    Choose **Analysis** > **Create Calculated Field** to open the calculation editor. Name the calculated field **Then Order Also Contains** and type or paste the following in the formula area:

    ```
    IF [Sub-Category] <> [Order Contains] THEN [Sub-Category] END
    ```

    You may have to replace `<>` with `<>` after you paste.

2. Create another calculated field to identify matching products.

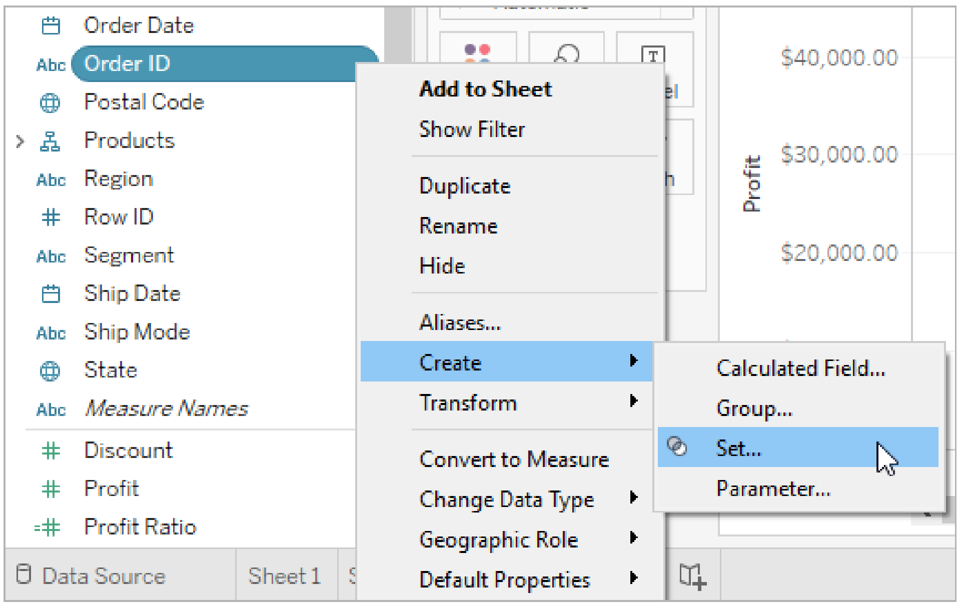
Name the field **Product Matches** and type or paste the following in the formula area:

```
IF [Sub-Category] = [Order Contains] THEN 1 END
```

Create a Set

1. Now create a set to determine whether an order has the item that was selected in the parameter control.

2. Hover the cursor over the **Order ID** dimension in the Data pane, click the down arrow at the right end of the field, and choose **Create** > **Set**.



3. In the Create Set dialog box, type **Order Has Selected Product** in the **Name** text box.

4. Go to the Condition tab, select **By field**, and in the drop-down lists make the following selections and entries to build the condition:
   a. In the first drop-down list, select **Product Matches**.
   b. In the second drop-down list, select **Sum**.
   c. In the next drop-down list, select **>=**.
   d. In the last text box, type **1**.
   e. Click **OK**.

5. Click **OK**.

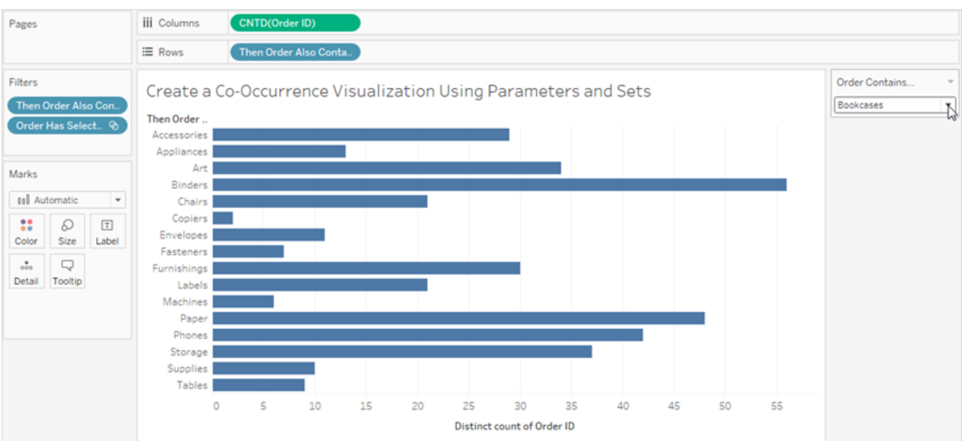
Build the View

Finally, build the view to display which items are also contained in an order with the selected item.

1. Drag **Then Order Also Contains** to **Columns**.
2. Drag **Order ID** to Rows.

   In the warning dialog box, click **Add all members**.
3. Click the **Order ID** field on Rows and choose **Measure** > **Count (Distinct)** to change the aggregation.
4. Right-click (control-click on a Mac) the Null bar on the x-axis and choose **Exclude**.
5. Drag the **Order Has Selected Product** set to the **Filters** shelf.
6. Press Ctrl+W to swap the fields on Rows and Columns.

You (or your users) can now use the **Order Contains** parameter control to select an item in an order, and then see a bar chart showing which other items are also included in orders with the selected item.



Visualize Benford's Law

Benford's Law is a mathematical law that states that the leading, or left-most, digit in many real-life data sources is distributed in a very specific manner. Specifically, the number 1 occurs as the leading digit about 30% of the time, and as numbers get larger they occur less frequently, with the number 9 occurring less than 5% of the time. When fraudsters are fabricating data, they may not know to create fake data that conforms to Benford's law, and in some cases this makes it possible to detect fake data or at least to create doubt about its veracity.

This article describes how to apply Benford’s Law to Sales data, using the **Sample - Superstore** data source provided with Tableau Desktop.

The process requires you to do the following:

1. Create calculated fields to use in your view.
2. Set up the view.

The following sections break these procedures down into specific instructions.

Create calculated fields to use in your view

1. In the **Analysis** menu, select **Create Calculated Field** to open the calculation editor. Name the calculation **Leftmost Integer** and type or paste the following in the formula area:

   ```
   LEFT(STR([Sales]),1)
   ```

2. Create a second calculated field and name it **Benfords Law**. Type or paste the following in the formula area:

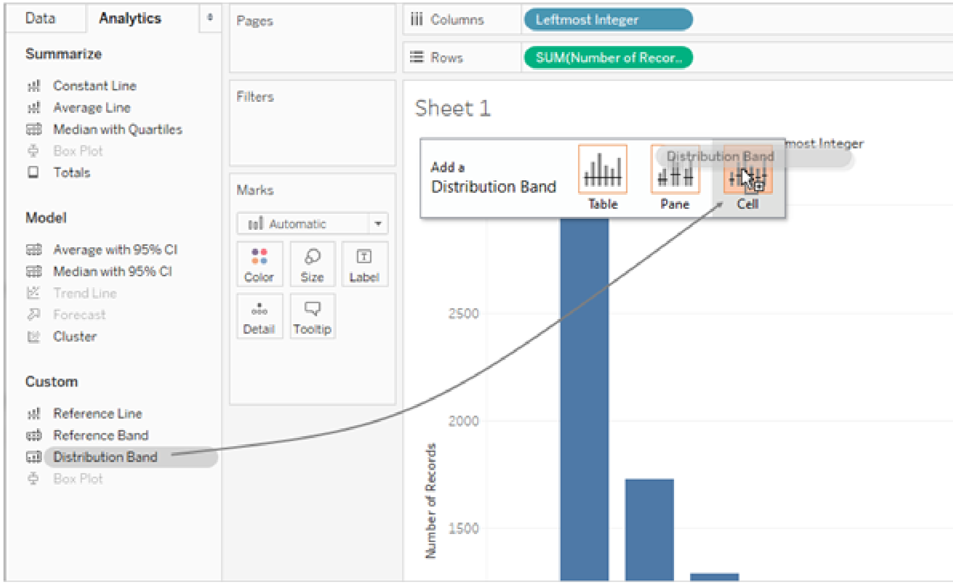
   ```
   LOG(INT([Leftmost Integer])+1)-LOG(INT([Leftmost Integer]))
   ```

Set up the view

1. From the **Data** pane, drag **Leftmost Integer** to **Columns**, and then drag **Orders (Count)** to **Rows**.
2. Click **CNT(Orders)** on Rows and choose **Quick Table Calculation** > **Percent of Total**.

   Your view now shows the distribution of first digits, and the size of the bars (decreasing from left to right) suggests that the data in this case conforms to Benford's law. But we can do more to frame the data by adding reference distributions.

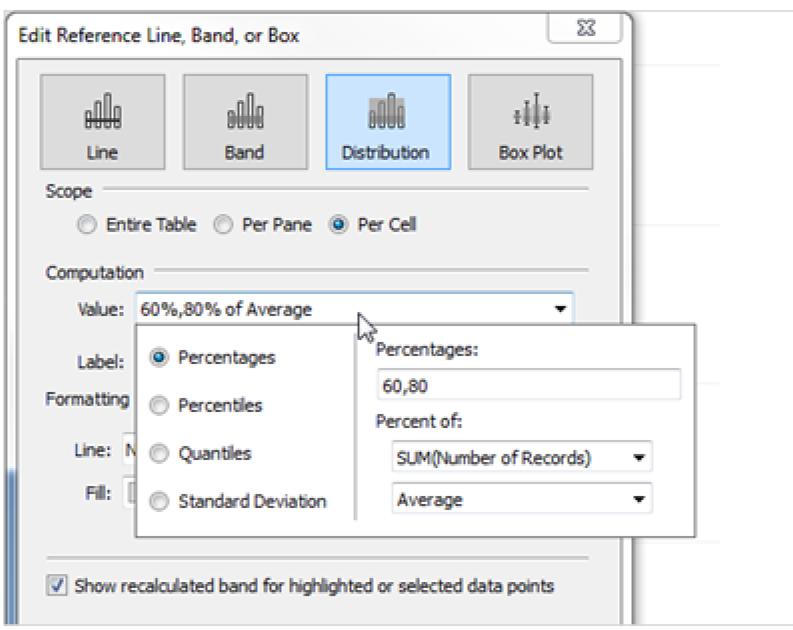
3. From the **Data** pane, drag **Benfords Law** to Detail on the Marks card. Click **Benfords Law** on the Marks card and select **Measure** > **Minimum**.
4. Switch from the **Data** pane to the **Analytics** pane and drag **Distribution Band** into the view. Drop it on **Cell**.



> **Note**: Distribution Bands are supported on web platforms starting with Tableau 10.2.

5. In the Edit Reference Line, Band, or Box dialog box, do the following:

   a. Click in the Value field to view an additional set of options:



   b. In the Percentages area, type `80,100,120`.

This specifies that you want bands spanning from 80 to 100 percent, and from 100 to 120 percent. Next you will specify what value the percentages are referencing.

c. In the **Percent of** field, choose **MIN(Benfords Law)**.

The Value field should now read `80%,100%,120% of Average Min. Benfords Law`.

The remaining steps configure the appearance of the reference bands.

d. Set **Label** to **None**.

e. Set **Line** to the thinnest available line.

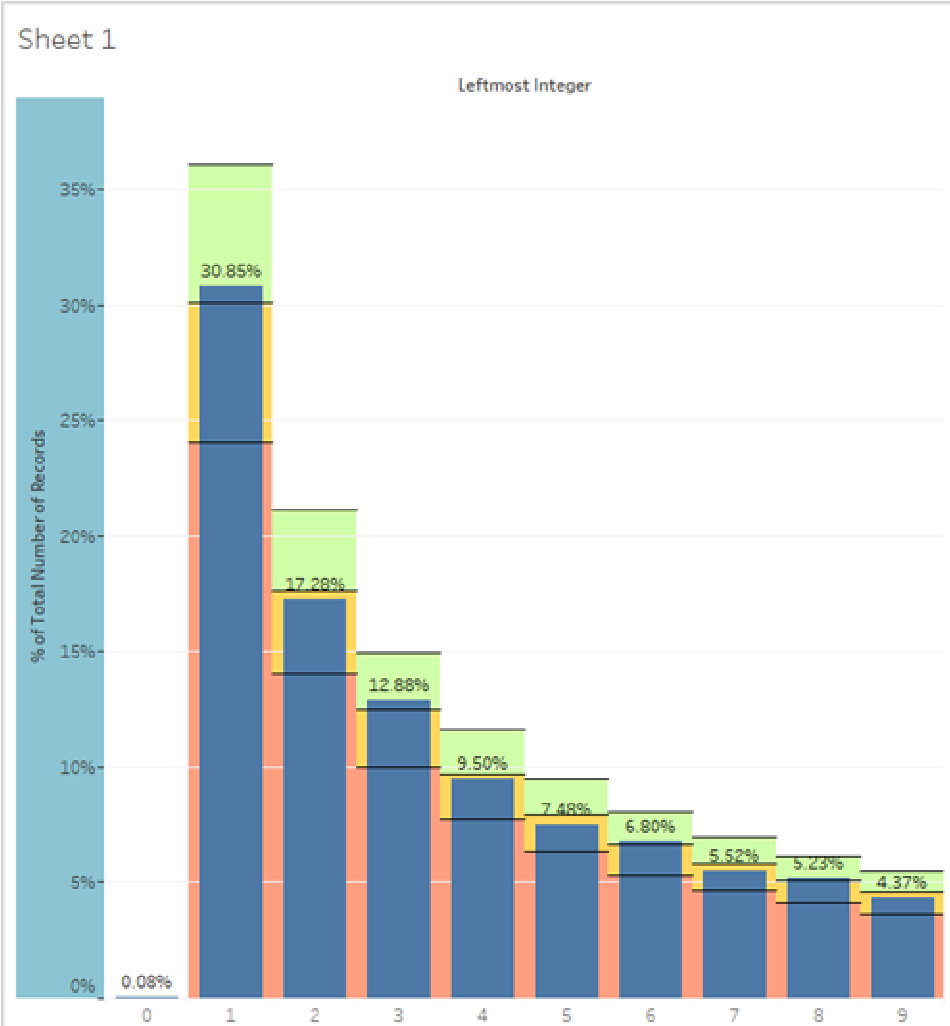
f. Choose **Fill Below**.

g. From **Fill**, select **Stoplight**.

h. Click **OK** to exit the Edit Reference Line, Band, or Box dialog box.

6. Click the toolbar button to display mark labels:

The finished view should look like this:



Even though Superstore is demo data, it's realistic as far as conforming to Benford's law. The blue bars that indicate actual percentages of initial digits align very well with the 100% value (that is, the line that separates the green zone from the yellow zone in the distribution bands) that shows expected Benford values in the view.

Create Bins from a Continuous Measure

Sometimes it's useful to convert a continuous measure (or a numeric dimension) into bins.

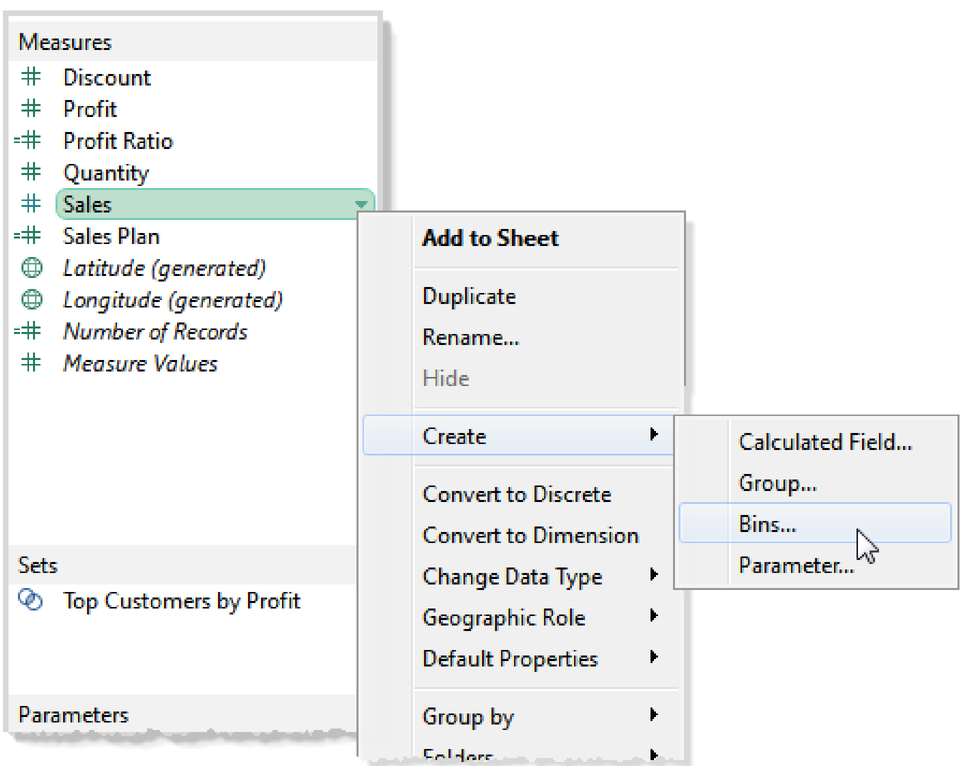
Any discrete field in Tableau can be considered as a set of bins. For example, suppose you create a view with **Profit** on **Rows** and **State** on **Columns**. You could consider the **State** field as a set of bins—each profit value is sorted into a bin corresponding to the state from which the value was recorded. But if you want to see values for **Profit** assigned to bins without reference to a dimension, you can create a numeric bin, with each individual bin corresponding to a range of values.

> **Note:** You can bin data only for relational data sources and binned fields cannot be used in calculations. However, it is possible to create a calculated field that will replicate a bin of a specific size. For example: `(FLOOR([Sales]/1000)*1000)` will create bins with a size of 1000. By dragging this calculation to the dimensions pane, you can use these bins with cube data sources and calculated fields.

When you create bins from a measure you create a new dimension. That's because you are creating a field with a limited and discrete set of possible values out of a field with an unlimited, continuous range of values. However, once the dimension is created, you can convert it to a continuous dimension. This can be useful, for example, if you want to create a histogram. See **Create a Histogram from a Binned Dimension** on page 1708.

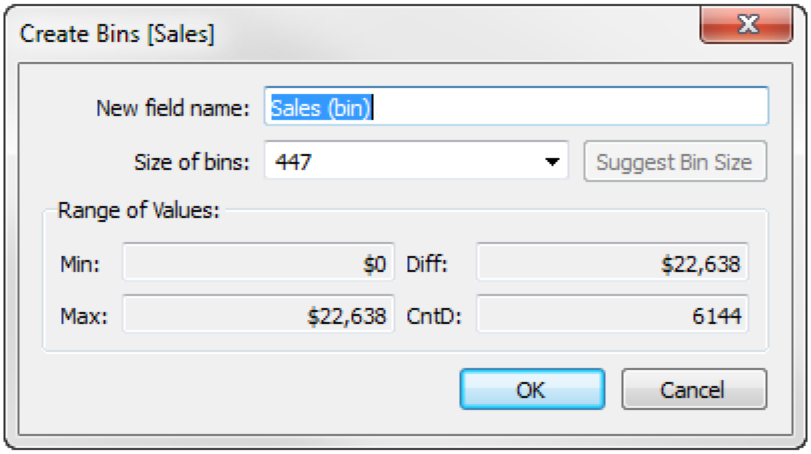
Create a Binned Dimension:

1. In the **Data** pane, right-click (control-click on Mac) a measure and select **Create** > **Bins**.



2. In the Create Bins dialog box, accept the proposed New field name or specify a different

name for the new field.



On the web, the dialog box is named Edit Bins and has a slightly different appearance, but the options are the same.

3. Either enter a value in the **Size of bins** field or have Tableau calculate a value for you.

   If Tableau can perform the optimizing calculation quickly enough, the value you see initially in **Size of bins** is Tableau's estimate of the optimal bin size.

   If Tableau cannot perform the optimizing calculation quickly, the **Size of bins** field defaults to 10. In this case you can click **Suggest Bin Size** to have Tableau perform the optimizing calculation.

   The formula that Tableau uses to calculate an optimal bin size is

   $\texttt{Number of Bins = 3 + log}_2\texttt{(n) * log(n)}$

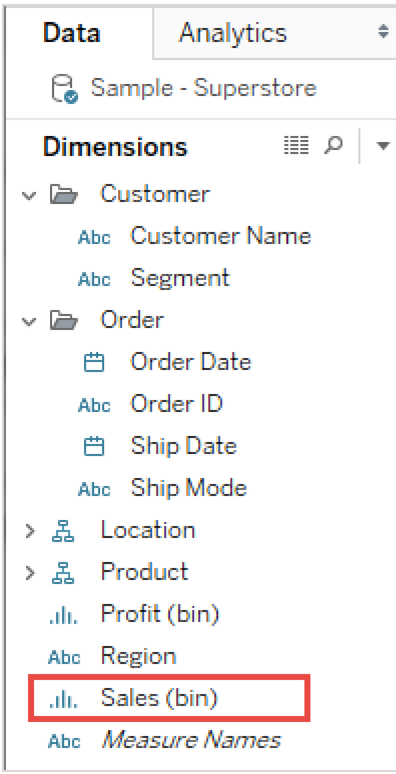
   In the formula, *n* is the number of distinct rows in the table. The size of each bin is determined by dividing the difference between the smallest and the largest values by the number of bins.

The four read-only fields in the lower part of the Create Bins dialog box show you the data that Tableau uses to suggest a bin size. You can also consider these values if you want to set a bin size manually. The values are:

Min	The field's minimum value.
Max	The field's maximum value.
Diff	The difference between the field's minimum and maximum values.

CntD	The number of distinct values (rows) in the data.

After you click **OK** to dismiss the Create Bins dialog box, a new binned field appears in the **Dimensions** area of the **Data** pane.



When you add a binned dimension to the view, each bin acts as an equal-sized container that summarizes data for a specific range of values. Column or row headers are created, where each bin label designates the lower limit of the range of numbers that is assigned to the bin. Note that the lower limit is inclusive.

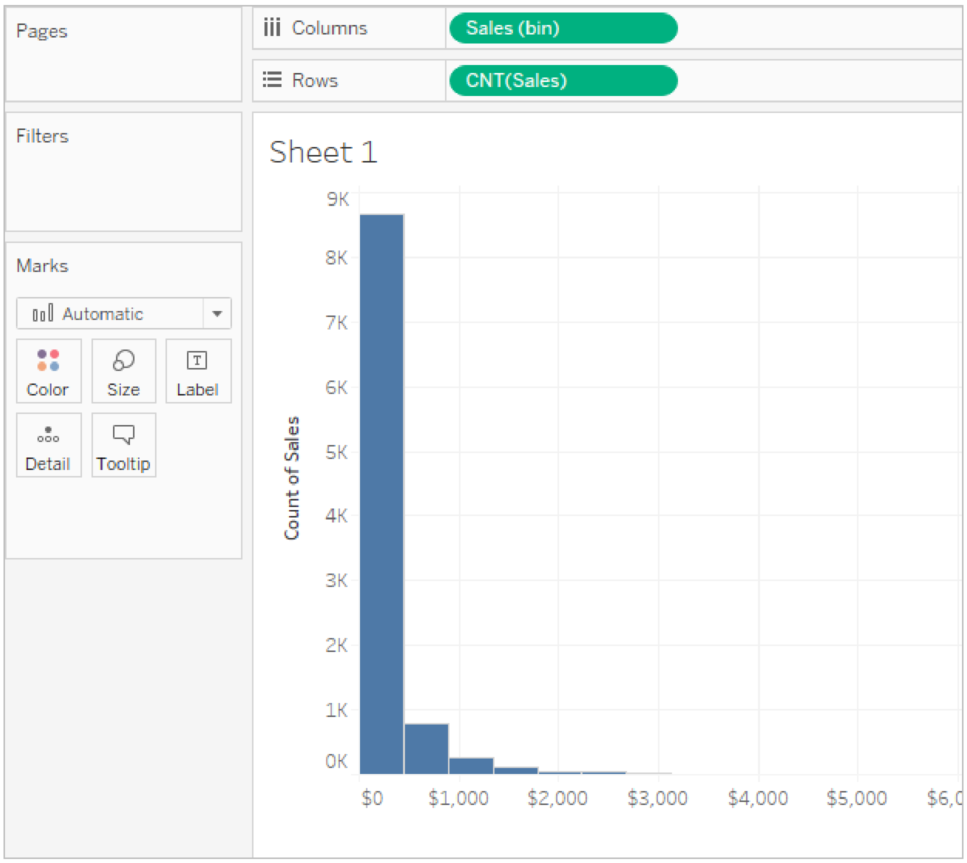
Create a Histogram from a Binned Dimension

If you create a binned dimension, you can use it as the starting point for creating a histogram. Using the **Sales (bin)** dimension created according to the instructions above, use the following steps to create a histogram.

> **Note**: A quicker way to create a histogram is using Show Me. See **Build a Histogram** on page 1623 in the Build-It-Yourself Exercises section for information on creating a histogram using Show Me.

1. Click the **Sales (bin)** dimension in the Data pane and choose **Convert to continuous**.
2. Drag the **Sales (bin)** dimension from the Data pane and drop it on the **Columns** shelf.
3. Drag the original **Sales** field from the Measures area of the Data pane and drop it on the **Rows** shelf.
4. Click **SUM(Sales)** on **Rows** and change the aggregation from Sum to Count.

The result is a histogram:



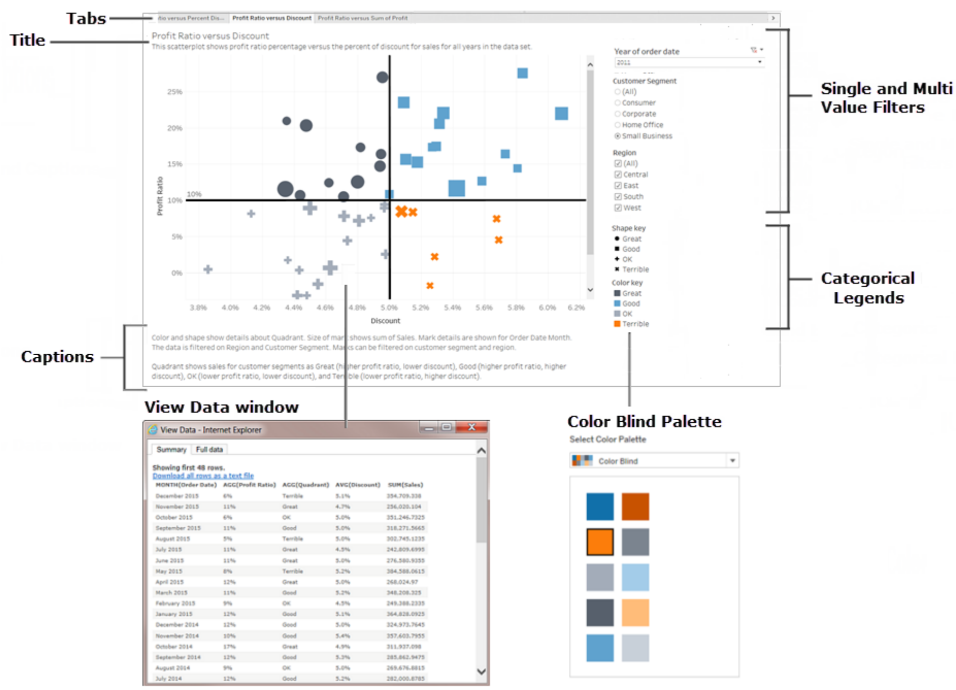
Build Data Views for Accessibility

If you want to make your views accessible to as many people as possible, or if you work in an environment that is subject to US Section 508 requirements, other accessibility-related laws and regulations, you can use Tableau to build data views that conform to the Web Content Accessibility guidelines (WCAG 2.0 AA). This includes building views that are accessible to users who use screen readers, braille keyboards, keyboard-only navigation, and so on.

View controls that support accessibility

To create an accessible view, use Tableau Desktop to create a view that includes WCAG-conformant elements, and then publish and embed that view in a web page that conforms to Web Content Accessibility guidelines (WCAG 2.0 AA).

The example below shows a view with a scatter plot chart and includes various elements that you can use to be WCAG conformant. For more details, see **Best Practices for Designing Accessible Views** on the facing page and **Author Views for Accessibility** on page 1719.



Note: The color blind palette can help you select colors that can be recognized by users with visual impairments. When you assign colors to different dimension values make sure that they provide enough contrast and aren't too close to each other on the light-dark spectrum.

Additional support for accessible views

In addition to the controls shown in the illustration, Tableau also supports the following features to help you create accessible views:

- Keyboard navigation
- Programmatic context for assistive technologies (using ARIA roles)
- A text equivalent for charts and visualizations
- Compliance with contrast standards
- Authentication when signing in to Tableau Server for embedded views

Note: If you use a data source that requires authentication to access it, that authentication page does not conform to the WCAG 2.0 AA guidelines. To prevent the authentication page from showing, you can embed the credentials for the data source when you publish. For information about how to embed credentials for the data source, see **Set Credentials for Accessing Your Published Data** on page 3079, "Set the authentication type", Embedded password section.

Additional resources

The following external resources can help you when designing your views for accessibility.

- Web Content Accessibility Guidelines (WCAG 2.0)
- Color Contrast Tips and Tools
- Color Contrast Analyzer

Disclaimer: Although we make every effort to ensure these links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

For information, guidelines and examples about how to create accessible views, refer to the following topics in this section.

Best Practices for Designing Accessible Views

You created a great view, and you want to make sure that all of your users can see and understand the data that you've put together- but some users have visual or physical impairments. So what can you do?

Tableau supports several controls to enable you to build a view that complies with US Section 508 requirements when consuming a view in Tableau Online or Tableau Server that meets the Web Content Accessibility guidelines (WCAG 2.0 AA). Ensure that content shared from Tableau Online or Tableau Server follows the steps in **Author Views for Accessibility** on page 1719 and conforms to the following WCAG 2.0 AA principles:

- **Perceivable** - Information and user interface components must be presented to users in a way that they can perceive. Consider including text alternatives and alternate ways to present the content.
- **Operable** - The user interface components and navigation must be accessible to users from the different devices or methods that they use to interact with the view.
- **Understandable** -The information presented in the view must be understandable to your users. For example, using clearly distinguishable names and labels for different elements shown in your view.

Follow the best practices described in this article and incorporate the steps described in **Author Views for Accessibility** on page 1719, to build views that are accessible to all of your users when published on Tableau Server or Tableau Online.

Keep it simple

WCAG 2.0 AA principle: Understandable

You might have a lot of information that you want to communicate with your view. However, dense views can be difficult to understand or navigate using a screen reader or keyboard. Use the following guidelines to help you communicate everything that you want to without overwhelming your users with an over-packed view.

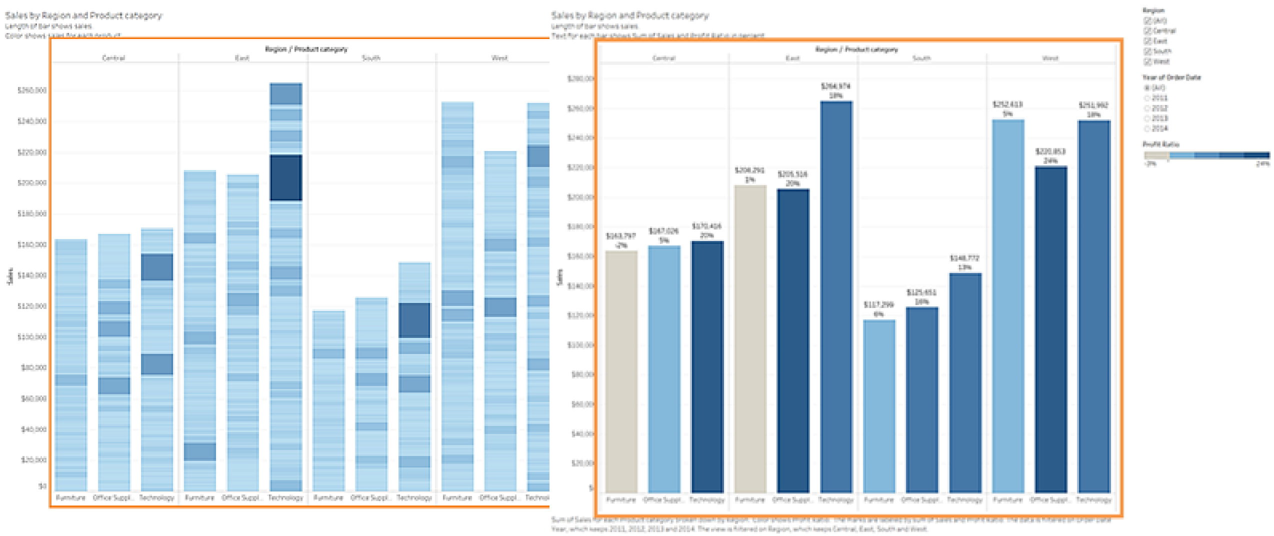
- Aggregate your data whenever possible to help reduce the number of marks you are showing. Also, showing more than 1000 marks in the view can cause the view to be rendered by the server instead of the browser and server-rendered views are not yet supported for WCAG conformance.

  Users can also access the **View Data** page (enabled by default) to review the underlying data for the marks or can download the data from that page to an accessible application to view it that way.

  Example: This example shows two different bar chart views to illustrate the difference between a detailed and aggregated view.

Not easily accessible - Too many	More accessible - Aggregated view

marks



- This much detail makes the view harder to understand.
- The view is showing over 5000 marks and does not include enough text to indicate what the different marks represent.
- It is too difficult to use for users who need screen readers.

- This example shows the same view with the data aggregated at a higher level.
- Key data points are still included, but they are now easier for users to read and understand.
- The number of marks is reduced from over 5000 to about 20.
- Users can still read the underlying details for the marks by placing focus on the view and then pressing **Enter** to open the **View Data** page.

- Consider using simple graphic elements like bar charts or line charts that allow you to use text, color, and shapes to add additional context to the view.
- Limit the number of marks to only those that emphasize the most important data points.

For information and examples about how to build this type of view, see **Keep it simple** on page 1719 in **Author Views for Accessibility** on page 1719.

Titles and captions

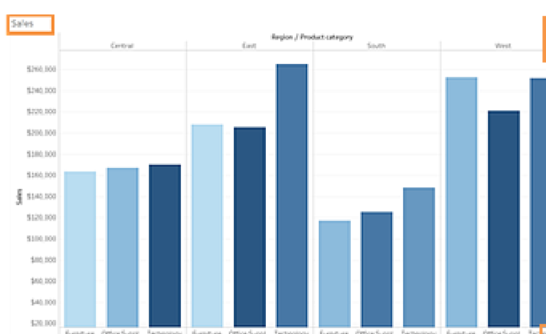
WCAG 2.0 AA principle: Perceivable, Understandable

Providing good descriptive text in titles and captions provides context to users who are using assistive technology, and can help them understand the data in your view. Use the following guidelines to adapt the visual nature of Tableau to meet the needs of all of your users.

- Think about your views as a supplement to the text you use to describe it.
- Use text in titles and captions to describe your visualizations and what you are showing.
- Use simple, easy-to-understand language. Avoid jargon, acronyms, or abbreviations.
- Don't include words like "image of" or "picture of" in your text descriptions, because screen readers sometimes already include this information.
- Avoid using all capital letters (for example in headings or titles), because they can be difficult to read.

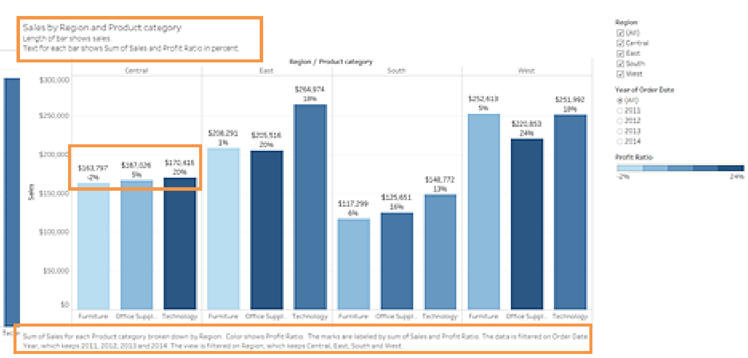
  Example: This example shows two different bar charts. One using very little text and the other using titles and captions to add context

Not easily accessible - Too little text



- A one-word title is not descriptive enough.
- The marks are differentiated by color and size. But without additional text, the context for

More accessible - Adding descriptive text to provide context



- This example shows the same view, but includes additional explanatory text in both the title and the caption.
- The same contrast color scale and

these marks can be difficult to understand. • There is no caption or other explanatory text to help explain this view.	size differentiators are used, but labels are added to the bar marks to give additional context.

For information and examples about how to build a view that includes text for context, see **Show more text and make it helpful** on page 1731 in **Author Views for Accessibility** on page 1719.

Additional text

WCAG 2.0 AA principle: Perceivable, Understandable

Using text beyond just titles and captions throughout your view can help users understand the context of the different elements that you are showing, as well as help describe the relationship between the different controls (such as legends and filters) and your data.

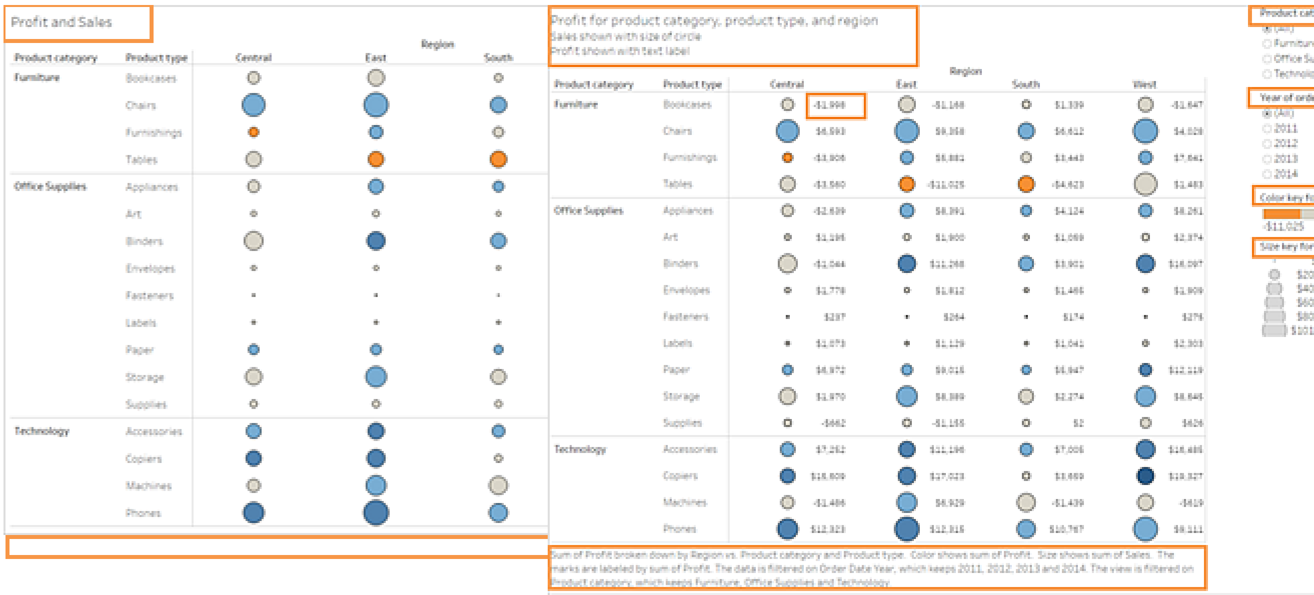
Use the following guidelines when adding additional text:

- Use text in headings on legends or filters to describe the control and what it does. You can also use text zones on a dashboard to add additional context for your visualizations to further describe what you are showing.
- Refer to controls by label whenever possible. For example changing the label for a legend from **Subcategory** to **Color key for product type** can help users understand the relationship between the controls and the data.
- If you include link text in your view, use text that describes where the link will take the user. For example, use link text like "Global Warming statistics for 1990-2000". Avoid wording like "Click here", "More", or "More information". These link text examples are too generic and can be confusing to users.
- Consider using Natural Language Generation (NLG) tools to help produce data-driven, textual narratives for visualizations. For example Wordsmith from Automated Insights, Narratives for Tableau from Narrative Science, or the Natural Language Insights extension from ARRIA.

Example: This example shows two bubble charts. One that uses only the text that shows by default when creating a view and one that adds context-specific text throughout the view to help convey meaning in the view.

Not easily accessible - Too little text

More accessible - More descriptive text



Not easily accessible - Too little text

- This view uses the default text only for the sheet title and the default labels for the filters and legend.
- The marks are differentiated by size and color only.
- There is no caption or other explanatory text included in the view to help provide context.

More accessible - More descriptive text

- This example shows the same view, but includes additional explanatory text.
- Additional text was added to the title and caption to explain the relationship of the marks and provide additional context about what the view is showing.
- Mark labels are added to show profit numbers so that users don't have to rely on the color only to understand this information.

For information and examples about how to build a view that includes additional text for context, see **Show more text and make it helpful** on page 1731 in **Author Views for Accessibility** on page 1719.

Color and contrast

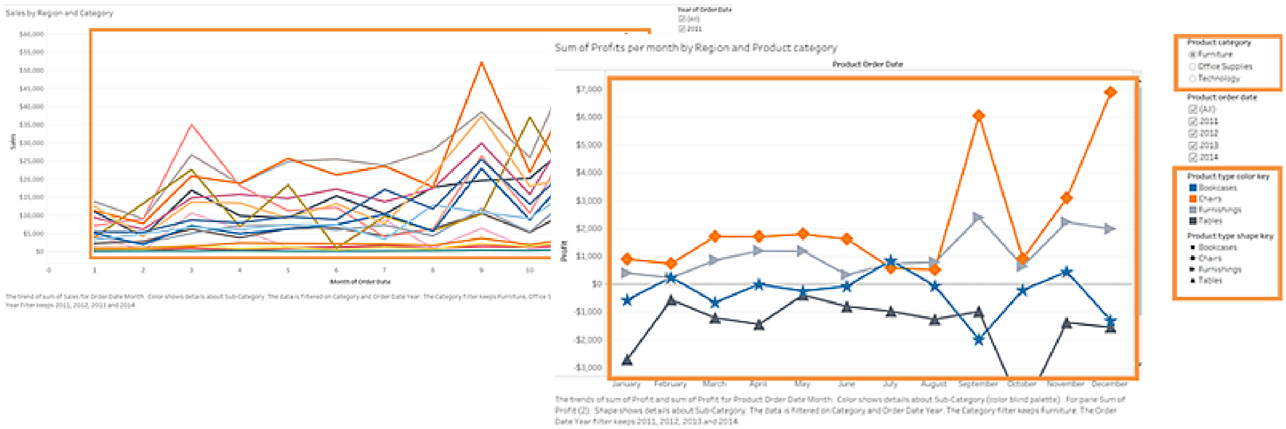
WCAG 2.0 AA principle: Perceivable, Understandable

You can use color to help distinguish marks in your view. However, for users with visual impairments, using color alone doesn't always provide enough of a distinction, especially when there are many marks in a view. Use the following guidelines when using color in your views:

- Tableau provides a color-blind palette that you can use to help you select appropriate colors for your view. For any color palette that you use, try to provide enough contrast and assign colors that differ from each other on the light-dark spectrum.
- For line marks, use additional options like shapes, size and labels to help distinguish them.
- Use contrast analyzer tools to help select the best text colors and backgrounds with sufficient contrast ratios. Make sure that color contrast for text is strong and meets the contrast ratio standards of 4.5:1 (3:1 for large text).

  Example: This example shows two line charts. One that uses color only to differentiate the lines and the other that uses more appropriate colors plus shapes to differentiate marks

Not easily accessible - Using only color to differentiate marks	**More accessible - Using color and shape to differentiate marks**



- This view uses only color to distinguish the lines from one another.
- The marks in the view are not using a color- palette that is fully accessible to users with visual impairments.

- This example shows the same view, but uses both color and shape to identify the marks.
- A color legend and a shape legend with clear titles identify what the colors and shapes represent in the view.

For information and examples about how to build a view that uses color to help distinguish marks, see **Use color thoughtfully and provide contrast** on page 1739 in **Author Views for Accessibility** on the facing page.

Publishing your view

WCAG 2.0 AA principle: Perceivable, Operable, Understandable

To make your views available to your users, publish your view to Tableau Server or Tableau Online, in the toolbar menu, click **Share**/ Users can interact with the view and toolbar buttons using a screen reader or a keyboard. For more information, see **Keyboard Accessibility for Tableau on the Web** on page 3164.

Note: Tableau does not support WCAG conformance for web editing, so Web Edit permissions must be turned off in the published workbook.

For information about how to publish and embed workbooks, find the embed code to copy into your web pages, and turn off the toolbar, see **Publish and share the view** on page 1751 in **Author Views for Accessibility** on the facing page.

Author Views for Accessibility

This article provides more detail on the design guidelines discussed in **Best Practices for Designing Accessible Views** on page 1711, and steps on how to create these views in Tableau Desktop (version 10.2 and later).

General steps

As a content author, you can create WCAG-conformant views following these general steps:

Step 1: Author views in Tableau Desktop (version 10.2 and later) following the best practices.

Step 2: Publish the views to Tableau Server or Tableau Online (version 10.2 and later).

Step 3: Share the link for users to access the content in Tableau Online or Tableau Server, or embed the views in a web page for users to access the content.

Most types of views that you create following these best practice guidelines in Tableau Desktop will be WCAG-conformant; they are treated by screen readers as images with a title and a description. Users will be able to use keyboard navigation to open the View Data window and access the underlying data for the view. However, map views in Tableau are not currently supported for WCAG conformance.

Building the views

Many of the principles that apply to creating effective views in Tableau also apply to creating WCAG-conformant views. The main difference in creating conformant views is that you should always provide other types of mark encoding when you use color in the view. You will also want to use text more liberally in the view to provide context for the view and its marks.

Keep it simple

When creating accessible views, keep it simple. Doing so helps people more easily see the relationships and intersections in your data.

How can you make a view more simple? Emphasize only the most relevant and important data. This means reducing the marks in the view to only what is needed. Limiting the level of detail and granularity in the view also helps to prevent cognitive overload.

When you design your views, keep in mind that your users will be accessing the underlying data in the view. The data you include in the view is what screen readers will read when users access the underlying data in the Summary tab of the View Data window.

Guidelines that support simplicity:

- Limit the number of marks in the view
- Orient your views for legibility
- Limit the number of colors and shapes in a single view
- Use filters to reduce the number of marks in the view at a given time

Limit the number of marks in the view

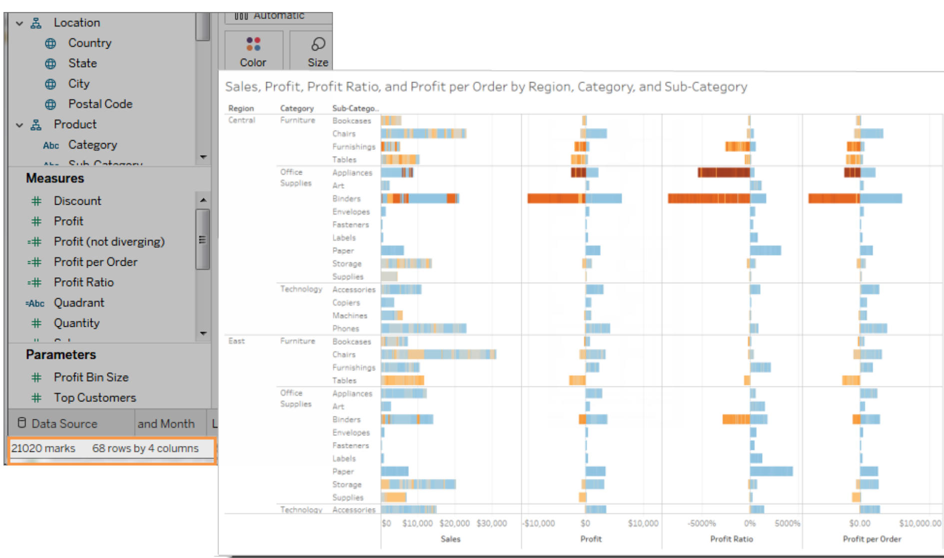
Keep the number of marks below 1000 to ensure the view will be rendered in the browser and not on the server.

> **Note**: Server-rendered views are not WCAG-conformant; only client-rendered views are WCAG-conformant. To ensure an embedded view is rendered in the browser, make sure the view has less than 1000 marks (including items in legends and filters).

You can see the number marks a view has when you are authoring the view in Tableau Desktop (bottom left corner of the window).

Example of a view with too many marks

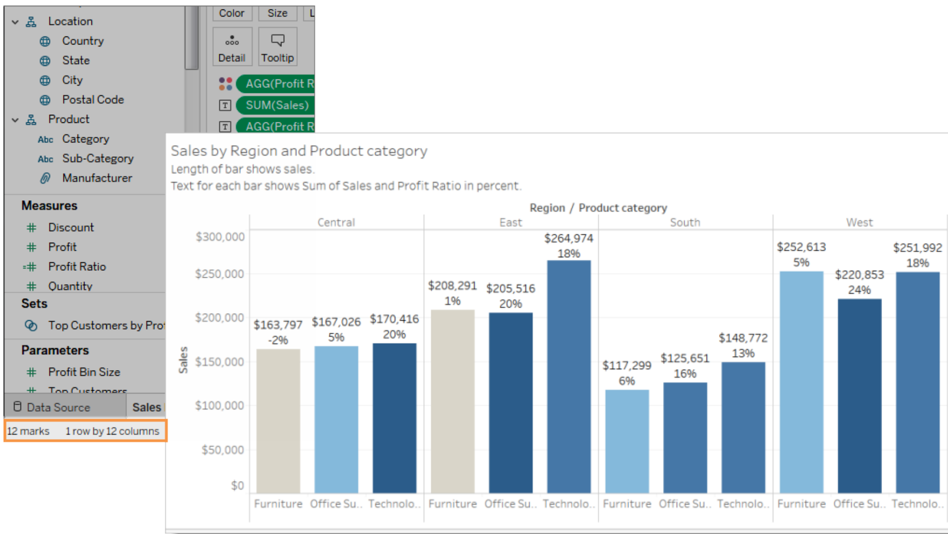
The following example shows a view with over 20,000 marks. The view shows a more granular level of detail in the data (versus aggregated). Multiple dimensions (Region, Category, Sub-Category) and measures (Sales, Profit, Profit per Order, and Profit Ratio) have been added to the view. Product Name is on Detail, which adds a mark for every product included in the data set.



Example of a view with fewer marks

The next example shows a more accessible view (fewer than 100 marks). The data is aggregated and shows a less granular level of detail in the data. Two dimensions (Region and Product Category) and one measure (Sales) are used in the view.

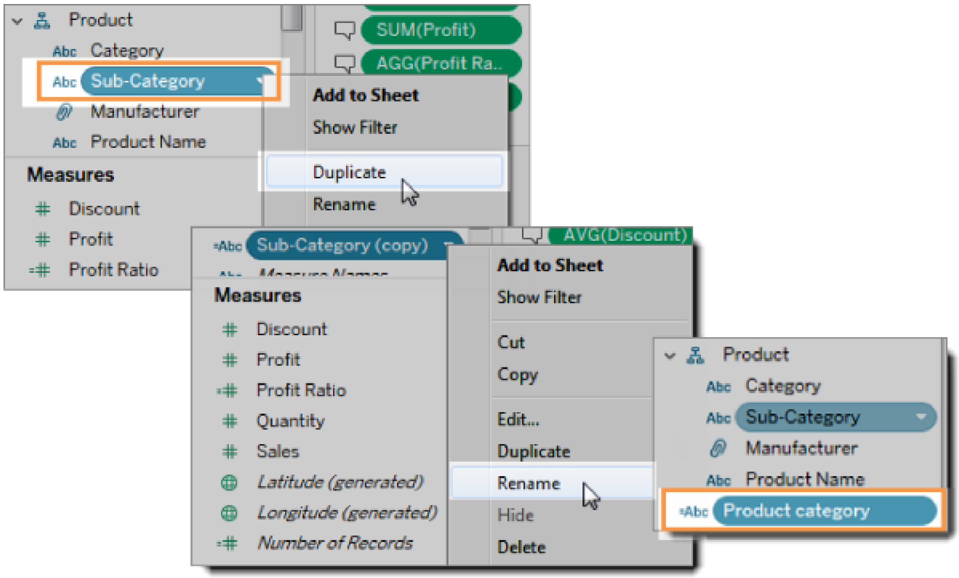
For more details about these concepts, see **Data Aggregation in Tableau** on page 341 and **How dimensions affect the level of detail in the view** on page 319.



There are several ways you can limit the number of marks:

- Create views of aggregated data rather than showing more granular levels of detail in the view. Don't place dimensions that contain thousands of values on **Detail** in the Marks card.
- If you want to use a dimension that is in a hierarchy, but not make the hierarchy available in the view, use a copy of that particular dimension. Copy the field you want to use from the hierarchy and build the view with the copied version of the dimension.

  Doing this prevents users from expanding the hierarchy in the embedded view, which might increase the number of marks in the view and result in the view being rendered on the server instead of the browser.



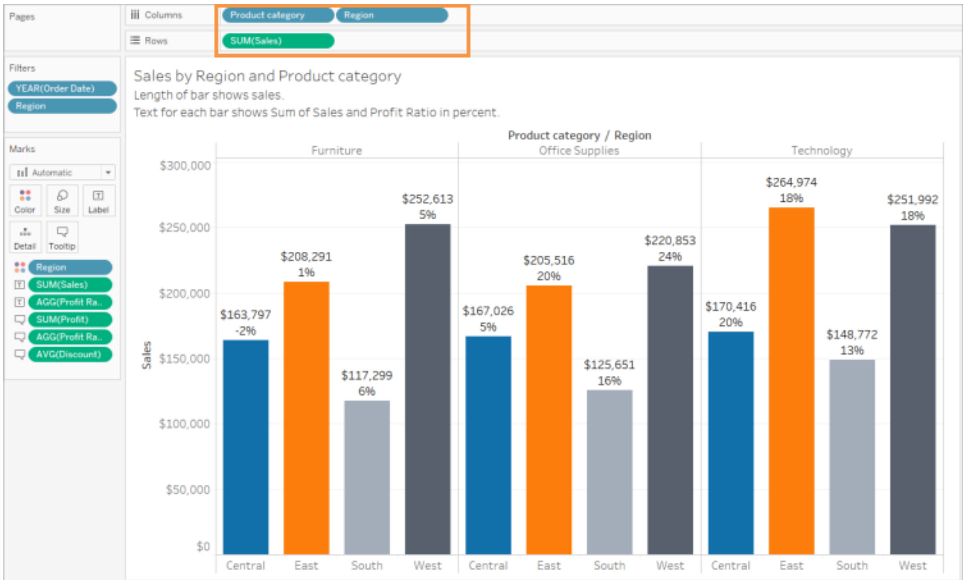
1. Right-click a dimension in the hierarchy, and then click **Duplicate**. A copy of the dimension will appear under Dimensions.

2. Right-click the copied dimension, and then click **Rename**. Give the copied dimension a unique, meaningful name. Now use that dimension to build the view.

- Limit the number of measures and dimensions used to create the view.

  Less accessible: The next example shows a view with four measures on Rows and three dimensions on Columns. More fields create a more complex view.

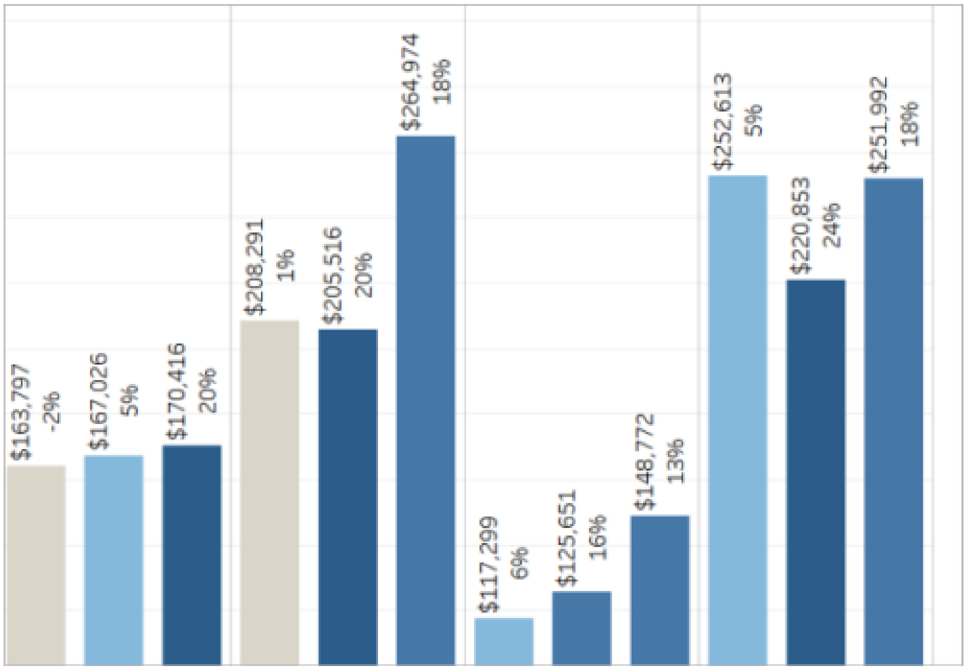
More accessible: This view has two dimensions on Columns and one measure on Rows. Fewer fields in the view structure reduces cognitive overload.



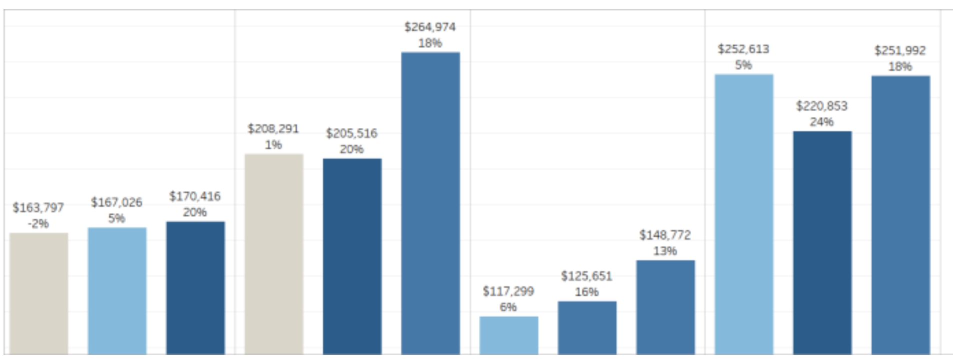
Orient your views for legibility

Horizontally-oriented labels and headers are easier to read than vertically-oriented labels.

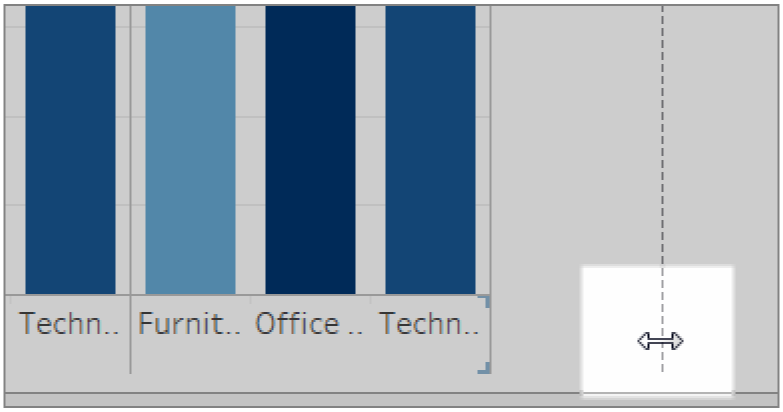
Less accessible: In this example, the text is being displayed vertically.



More accessible: In this example, space has been added to the view to allow the text to be displayed horizontally.



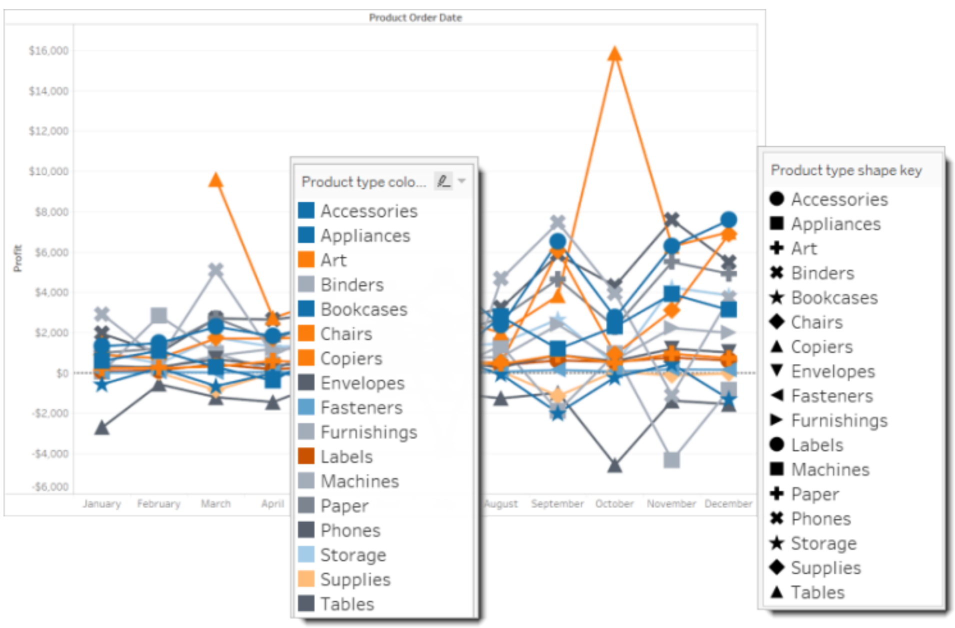
To change the orientation of the view and its headers and labels, you can click the **Swap** button on the toolbar, or you can resize the view to allow more room for horizontal labels. In this example, the table was resized. For details, see **Resize Tables and Cells** on page 2963.



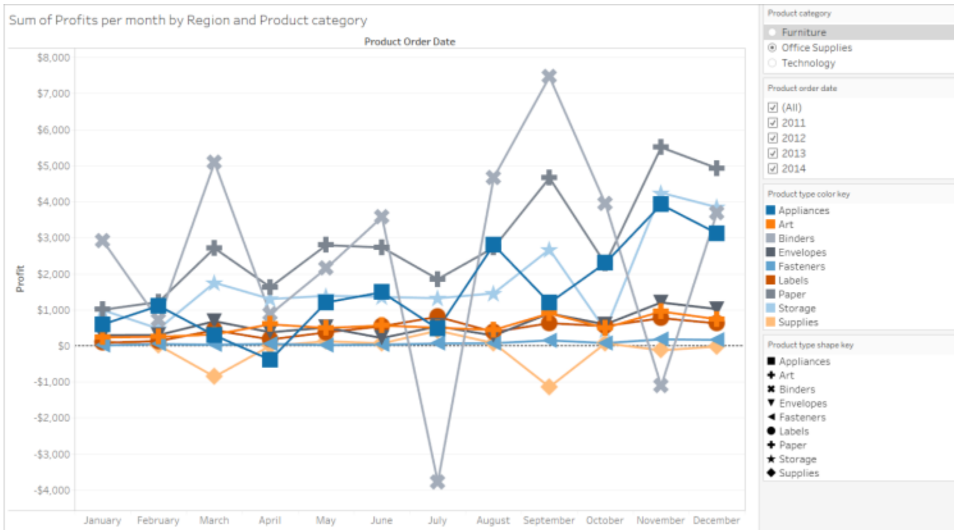
Limit the number of colors and shapes in a view

If you are going to show a dimension on Color or Shape, don't use more than 10 colors and shapes in one view so that users can distinguish each color and shape and see important patterns.

Don't do this: In the following example, the line colors have started repeating after the 10th mark applied to Product type.



Do this: In the next example, filters have been used to limit the number of marks that can be shown at a time. As a result, the number of colors and shapes in the view is never more than ten.

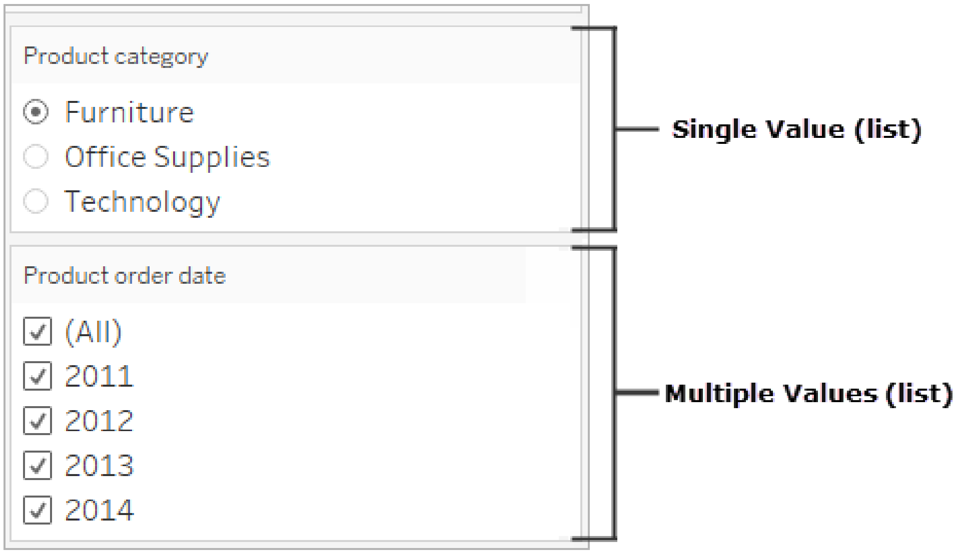


Use filters to reduce the number of marks in a view

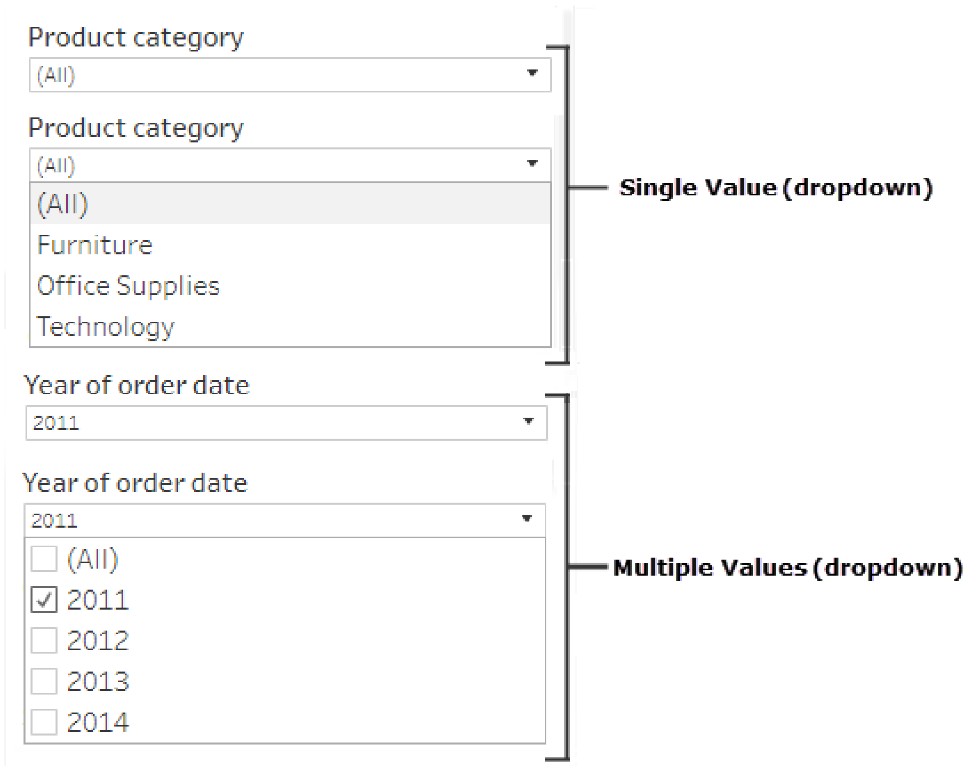
Using filters helps to focus the number of marks in the view to only what you want users to see. For more details about creating filters, see Filter Data from Your Views.

You can also show the following filter modes to allow users to control what data is in the view. The filter modes that currently are supported for WCAG conformance in Tableau are:

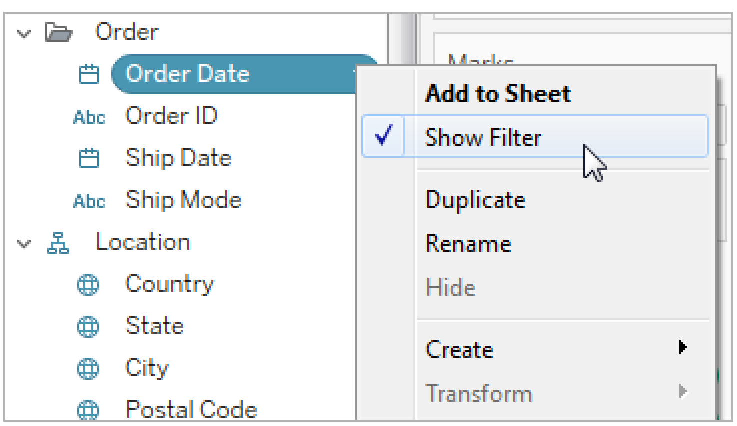
- **Single Value (list)**: a filter with radio buttons. Only one item can be selected at a time. Giving your users single-value filters is a great way to reduce the number of marks in the view.
- **Multiple Values (list)**: a filter with a list of items (with check boxes) that can be selected at the same time. Multiple-value filters will allow more marks to be shown at the same time.



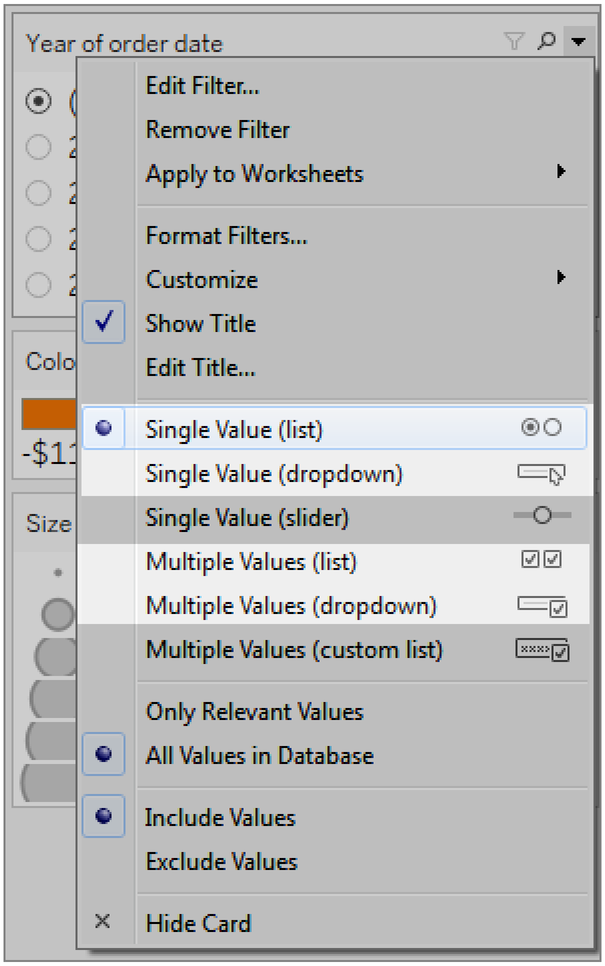
- **Single Value (dropdown)**: a filter with a dropdown list of items. Only one item can be selected at a time. Giving your users single-value filters is a great way to reduce the number of marks in the view.
- **Multiple Values (dropdown)**: a filter with a dropdown list of items that can be selected at the same time. Multiple-value filters will allow more marks to be shown at the same time.



To show a filter, right-click the field you want to use as a filter, and then select **Show Filter**.

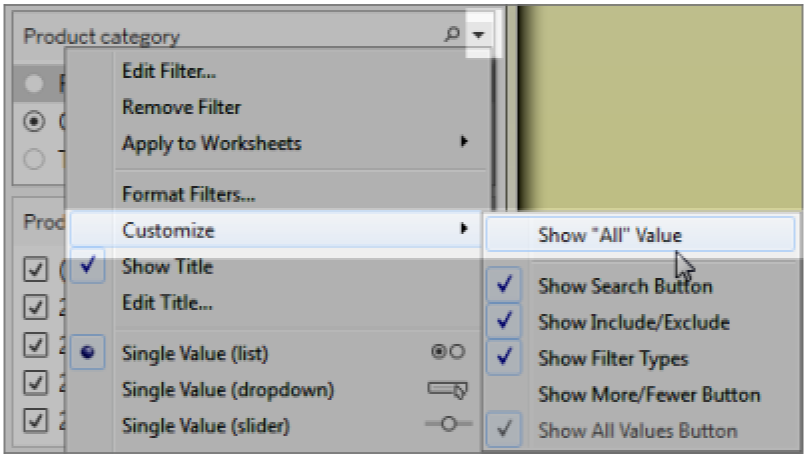


To select the filter mode, in the drop-down menu for a filter, select a **Single Value (list)**, **Single Value (dropdown)**, **Multiple Values (list)**, or **Multiple Values (dropdown)** filter.



For a Single Value (list) or Single Value (dropdown filter), you can remove the **All** option from the filter. In the drop-down menu for the filter, select **Customize** and then clear the **Show "All" Value** option.

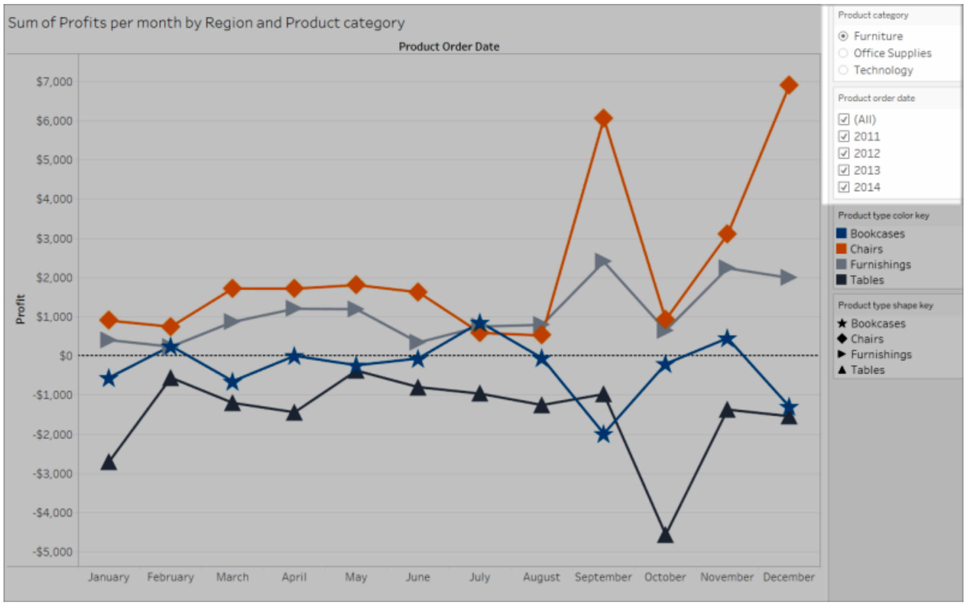
The following image shows the **Customize** option in a filter.



When you are using multiple filters in a worksheet, keep the following guidance in mind:

- When you are creating a view, test the browser zoom up to 200% to make sure all of the components in the view display as intended.
- Avoid positioning filters horizontally in a worksheet. In instances where the browser window is not wide enough, components in the view (such as filters) will overlap with the visualization.
- Don't reposition the filters after you add them to the view. If you reposition the filters or change their order, the tab order may change.

This example shows a view with a **Single Value (list)** filter for the Product category and a **Multiple Values (list)** filter for Product order date. The **Single Value (list)** filter only allows one product type to be shown at a time (**All** is hidden).



Show more text and make it helpful

In your views, make sure you show titles, captions, and mark labels. You can edit titles and captions to provide more context and details that help users understand the view.

- Be precise and consistent in how you name every object in the view.
- Provide text that explains how components of a visualization relate.
- Simplify text to remove redundancies.

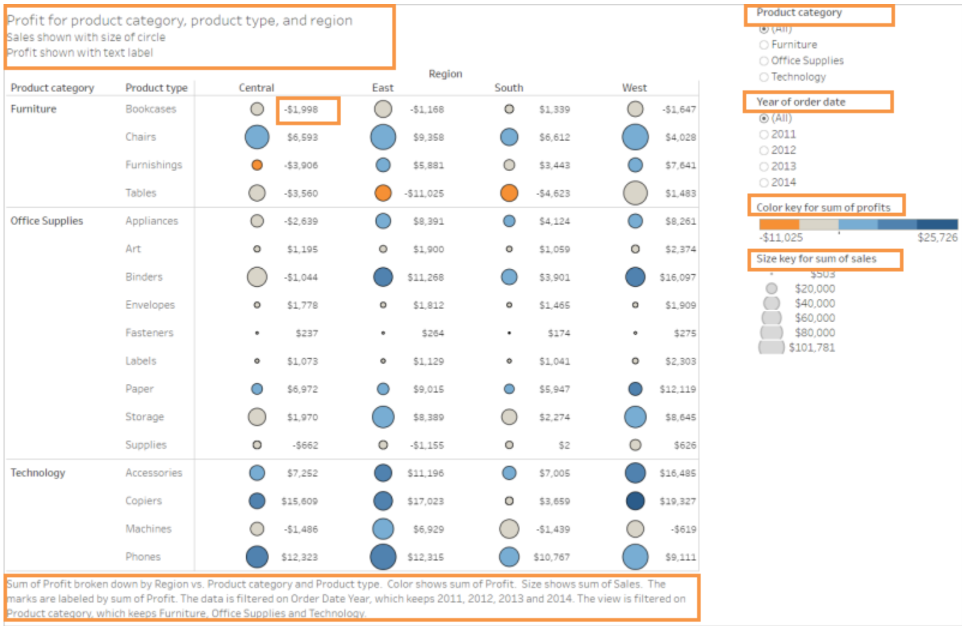
You can also rename measures and dimensions in the Data pane to make sure field names in the view are meaningful.

Here are the different elements in a view that you can show and edit:

- **Titles (worksheet, filters, and legends)**. Show titles for the worksheet, filters, and legends. The default title text is the sheet name or field name referenced by the filter or legend. You can edit any title to provide more context.
- **Captions (in worksheets)**. Show the caption in the worksheet. The default caption text gives a summary of what is shown the view. You can add to this text to provide more context.
- **Labels (Marks card, Label button)**. By default, labels are not shown in a view, so you must to choose the option to show them.

Less accessible: This example shows a view with limited text. Many text elements are not shown: the worksheet caption, mark labels, a descriptive title, and additional text about the view in the Title area. Also, the filter and legend titles have not been edited to provide more context.

More accessible: This example shows the same view with multiple text elements shown to provide more detail and context. Titles and captions are shown in the worksheet. The worksheet title was edited to include text that describes how data is indicated in the view. The worksheet caption includes the default summary text that is automatically generated by Tableau. The legend titles have been edited to make more sense in this view. Mark labels show Profit values so that users don't have to rely only on seeing the color.



You also can build views that show only text, such as this example of a Text Table view that shows similar data.

Sales by Region, Product category, and Product type

		Region			
Product category	Product type	Central	East	South	West
Furniture	Bookcases	$24,157	$43,819	$10,899	$36,004
	Chairs	$85,231	$96,261	$45,176	$101,781
	Furnishings	$15,254	$29,071	$17,307	$30,073
	Tables	$39,155	$39,140	$43,916	$84,755
Office Supplies	Appliances	$23,582	$34,188	$19,525	$30,236
	Art	$5,765	$7,486	$4,656	$9,212
	Binders	$56,923	$53,498	$37,030	$55,961
	Envelopes	$4,637	$4,376	$3,346	$4,118
	Fasteners	$778	$820	$503	$923
	Labels	$2,451	$2,603	$2,353	$5,079
	Paper	$17,492	$20,173	$14,151	$26,664
	Storage	$45,930	$71,613	$35,768	$70,533
	Supplies	$9,467	$10,760	$8,319	$18,127
Technology	Accessories	$33,956	$45,033	$27,277	$61,114
	Copiers	$37,260	$53,219	$9,300	$49,749
	Machines	$26,797	$66,106	$53,891	$42,444
	Phones	$72,403	$100,615	$58,304	$98,684

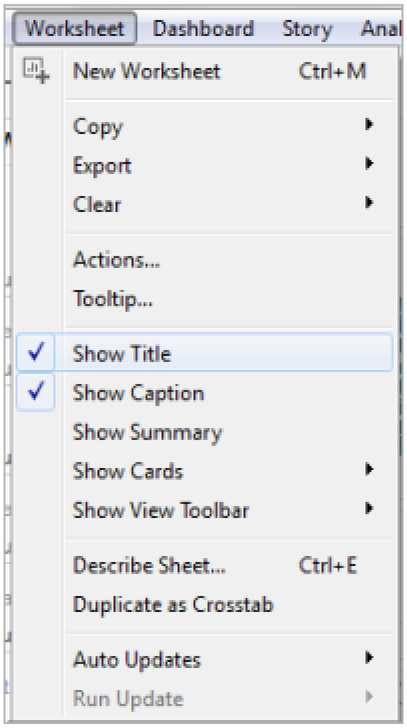
Region
☑ (All)
☑ Central
☑ East
☑ South
☑ West

Year of Order Date
◉ (All)
○ 2011
○ 2012
○ 2013
○ 2014

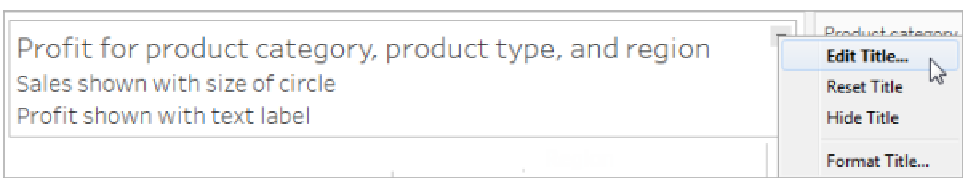
Sum of Sales broken down by Region vs. Product category and Product type. The data is filtered on Order Date Year, which keeps 2011, 2012, 2013 and 2014. The view is filtered on Region, which keeps Central, East, South and West.

To show and edit titles and captions for the worksheet

1. Select **Worksheet** > **Show Title**, and then select **Worksheet** > **Show Caption**.

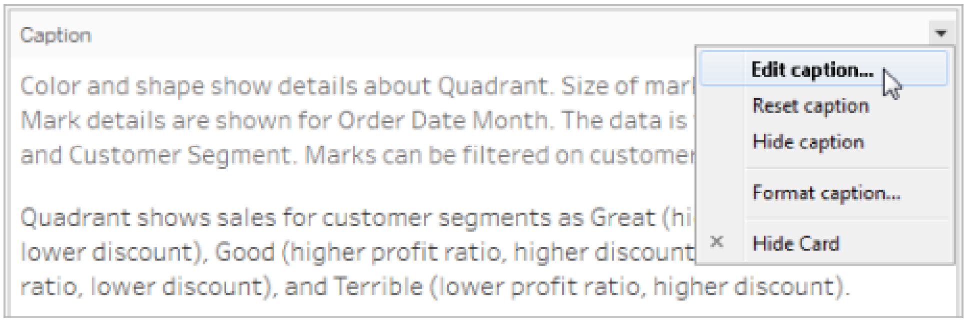


2. In the Title box at the top of the view, click the drop-down menu, and then click **Edit title**.



   Type text in the text area and format it, and then click **OK**. You can add text in addition to markup text, or replace the markup text.

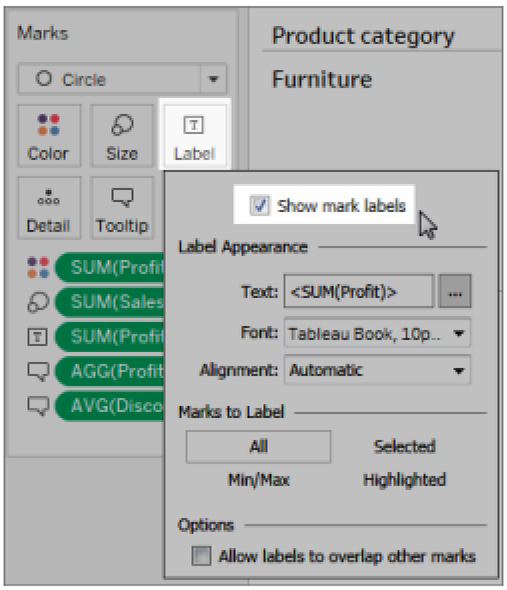
3. In the Caption box, click the drop-down menu, and then click **Edit caption**.



Type text in the text area and format the text, click the drop-down menu, and then click **OK**. You can add text in addition to the default text, or replace the default text.

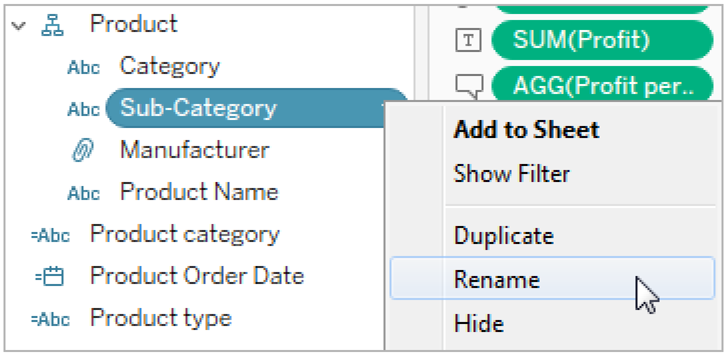
To show mark labels in the view

- Click **Label** in the Marks card, and then click **Show mark labels**.



To edit field names in the Data pane

1. Right-click the dimension or measure, and then click **Rename**.



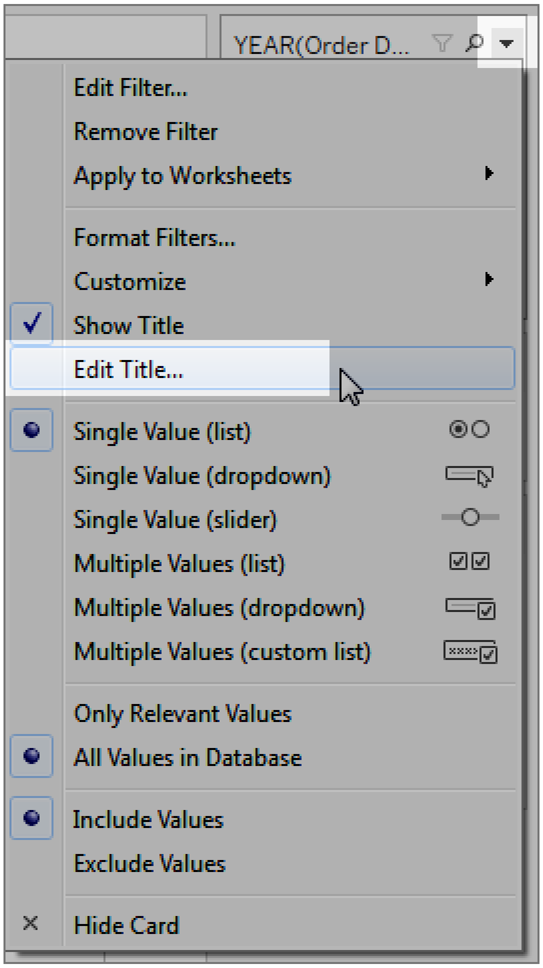
2. Edit the name, and the press **Enter**.

   For more details, see Rename fields.

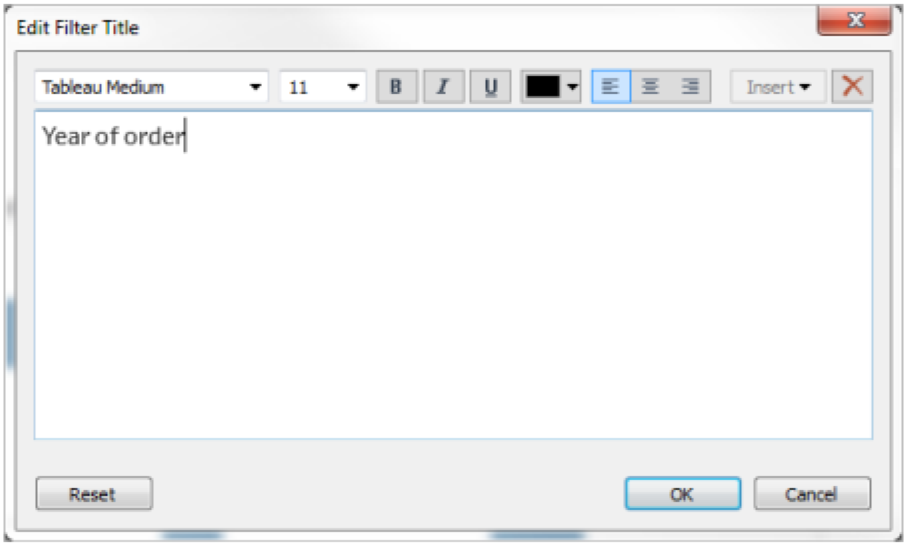
To edit filter or legend titles

The text associated with legends and filters should clearly describe their purpose. Edit titles for filters and legends to indicate function. Refer to filters by name in any instructions that you provide in the worksheet title or caption.

1. Click the drop-down menu for the filter, and then click **Edit Title**.

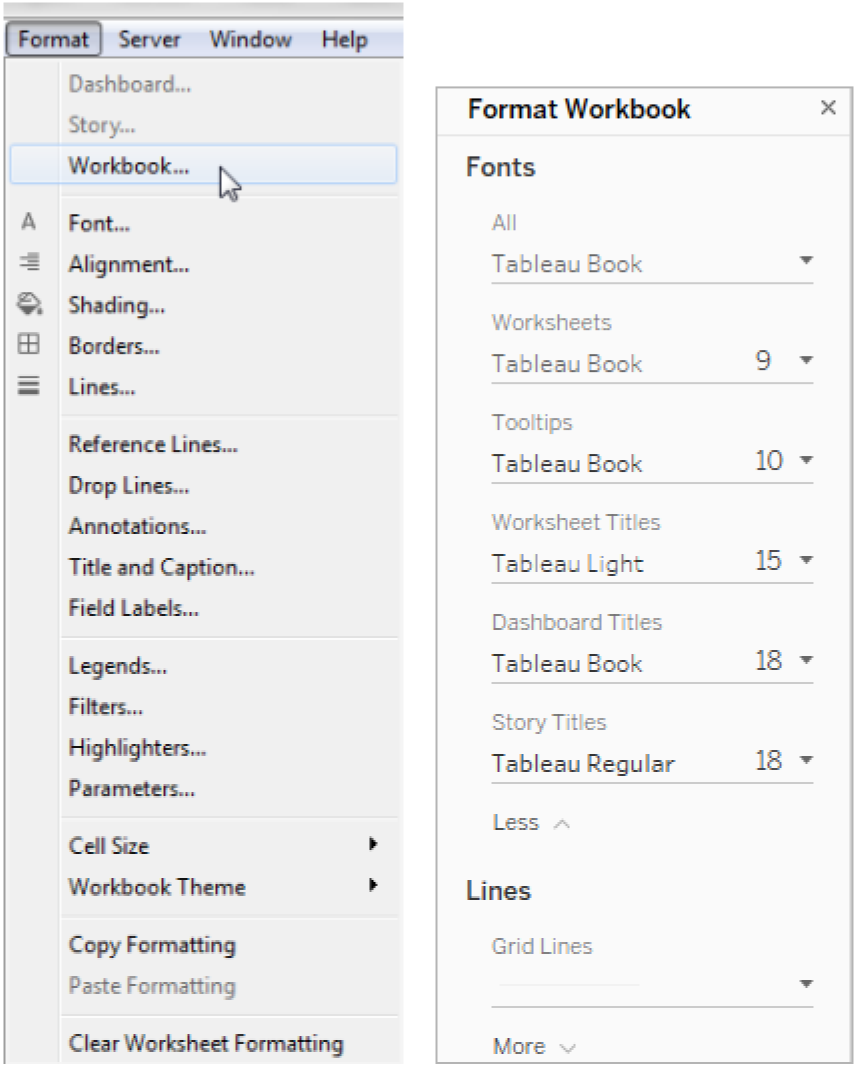


2. Edit the text, and then click **OK**.



Text Size

The default text sizes for elements in a Tableau workbooks are WCAG-conformant. However, if you want to change text size, you can select **Format** > **Workbook** to change the default text sizes for the workbook. For more details, see Format Your Work.



Use color thoughtfully and provide contrast

Different color palettes are available depending on the type of field that being placed on Color in the Marks card. For any color palette that you use, ensure that you are providing enough contrast and assign colors that differ from each other on the light-dark spectrum.

- For dimensions (discrete), a suggested practice is to use the **Color Blind** palette. You may need to adjust the colors assigned to different dimension values. Try to provide enough contrast and assign colors that differ from each other on the light-dark spectrum. For monochromacy, try using the **Seattle Grays** palette with **Stepped Color** set to 5.
- For measures (continuous), try using the **Blue** or **Orange-Blue Diverging** palette with

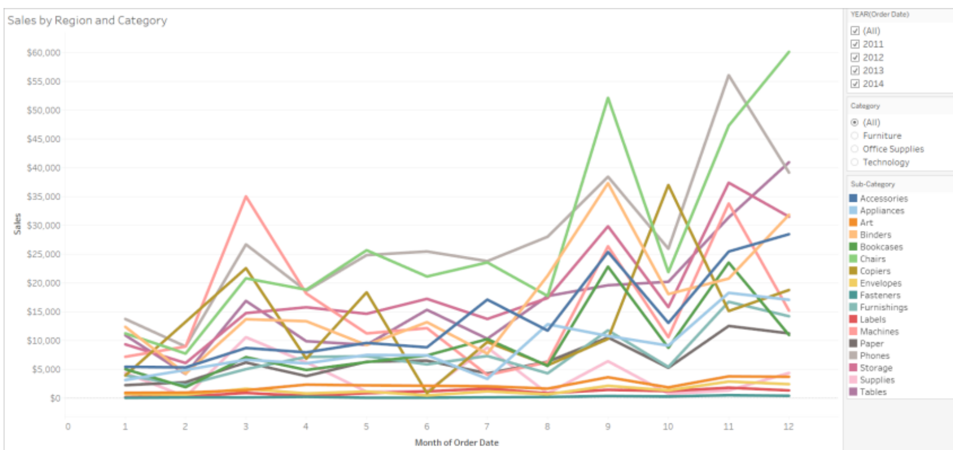
the **Stepped Color** option set to 5. You may need to adjust the number of steps used. For monochromacy, consider using the **Gray** palette with **Stepped Color** set to 5.

For text color, general guidelines are that the visual presentation of text and images of text has a contrast ratio of at least 4.5:1 (large text 3:1). Tableau's default color formatting follows these prescribed contrast ratios. You can use a contrast analyzer tool to test the contrast ratios of text colors and backgrounds.

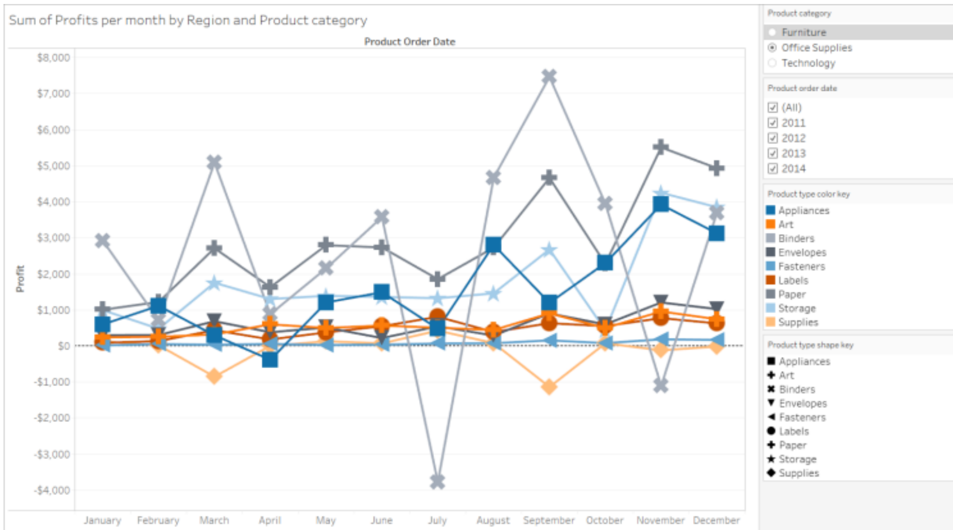
Important: When you use color in the view, you must also provide other types of encoding for marks, such as labels, size, shape, and position. Don't rely on color alone to communicate differences in marks.

Dimension on Color

Less accessible: This view shows an example of a discrete dimension being placed on Color, which is set to the Tableau 20 palette. The line path provides helpful position information, but the colors do not provide clear distinction between the line marks.

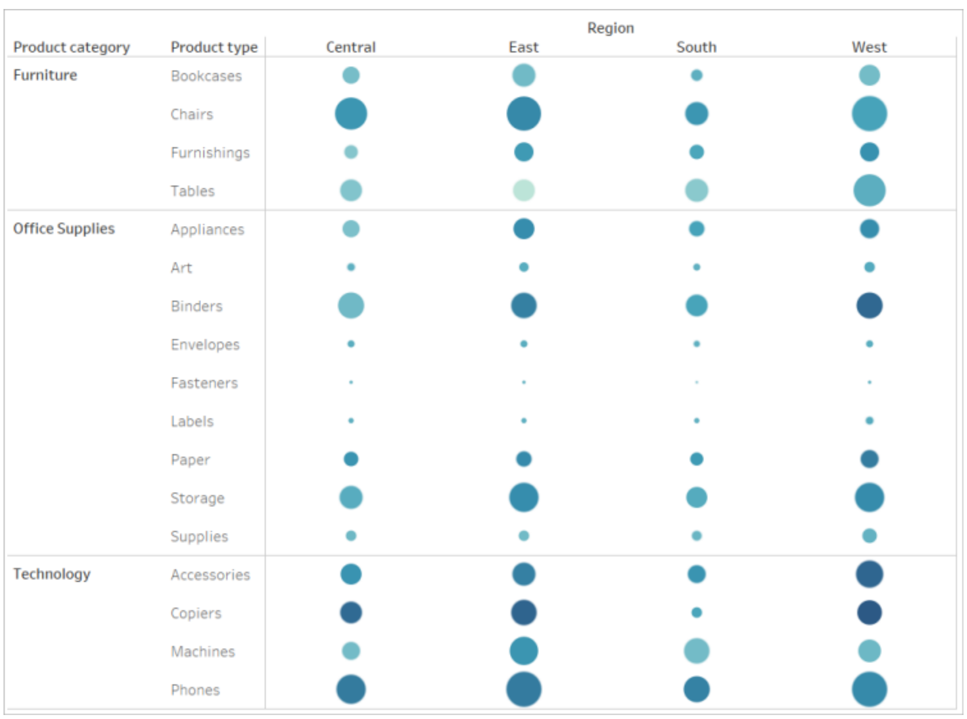


More accessible: In the next example, the Color Blind palette is being used. Shape is also being used to reinforce what is being shown in color.

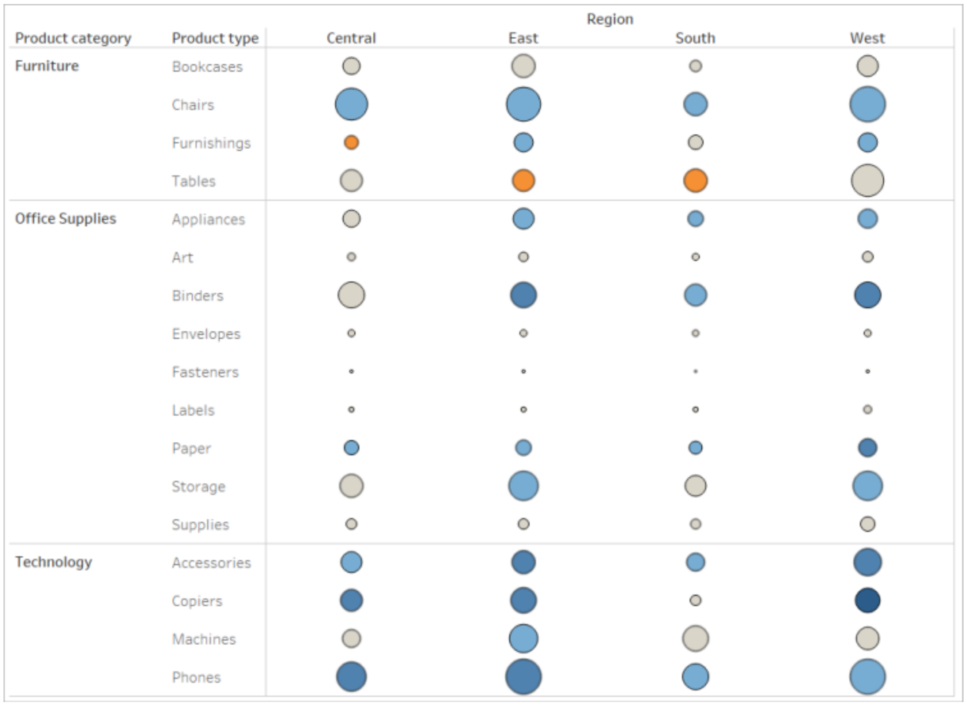


Measure on Color

Less accessible: This view shows an example of a continuous measure placed on Color, which is set to the Blue-Teal palette with gradient colors. The view does not provide enough contrast between colors and there is insufficient contrast on the light-dark spectrum. The use of size helps this view, but this color palette can make it hard for people with color vision deficiencies to see contrast differences between marks.



More accessible: This view uses the Orange-Blue Diverging palette with stepped colors. This palette provides improved contrast between marks.

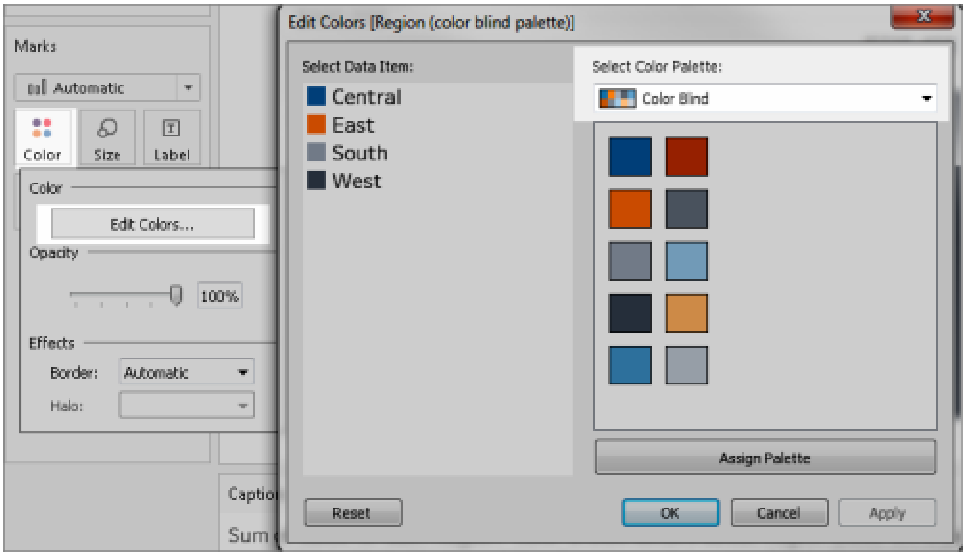


To select a palette

You must already have a measure or dimension on Color in the view for this option to be available.

1. Click **Color** in the Marks card, and then click **Edit Colors**.
2. For **Select Color Palette**, click the drop-down arrow and select the palette.

   This example shows selecting the Color Blind palette.

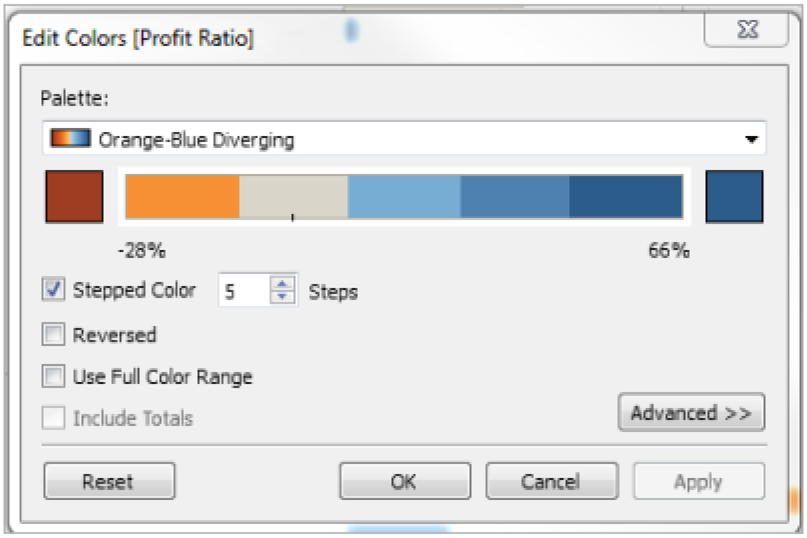


For the Color Blind palette, click **Assign Palette**, and then click **OK**. Otherwise, click **OK**.

To change the color of a value:

1. Click a value on the left, under **Select Data Item**.
2. Click a new color in the palette on the right. Hover over a swatch to identify the color. Repeat for as many values that you want to change.
3. Click **OK** to exit the Edit Colors dialog box.

The following example shows the Orange-Blue Diverging palette with Stepped Color set to 5 steps.



3. For the Color Blind palette, click **Assign Palette**, and then click **OK**. Otherwise, click **OK**.

   For details about colors and palettes, see Color Properties.

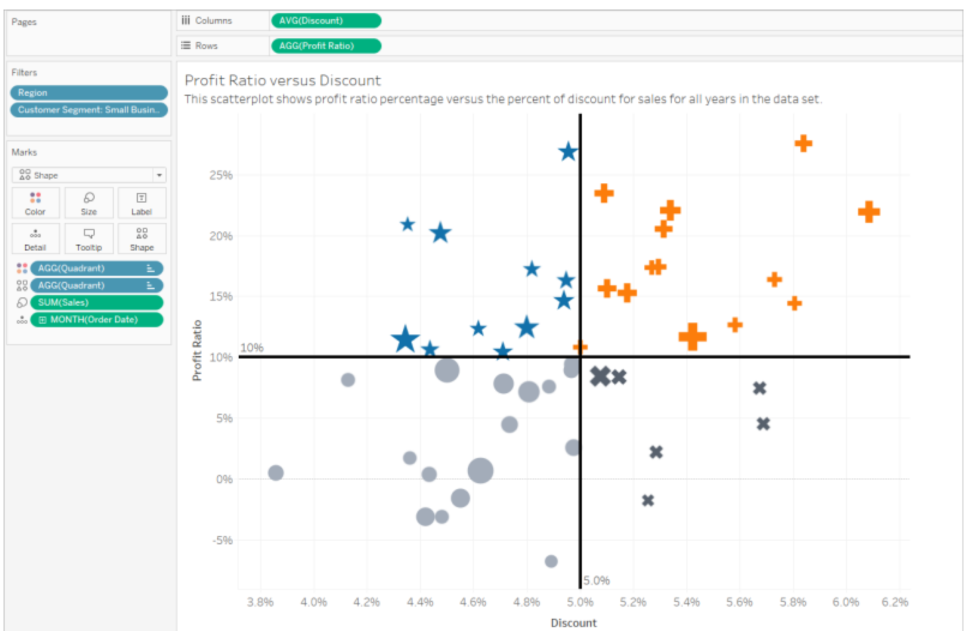
Provide visual cues beyond color: position, size, and shape

When you use color for field values, you must also provide other types of encoding for marks, such as labels, and position, size, and shape. Use position, size, or shape for the same field values to reinforce the differences between marks. These mechanisms all communicate information about the data to your users.

If you are using color to show measure or dimension values, make sure that you provide enough light-dark spectrum contrast for the colors that you assign to different marks.

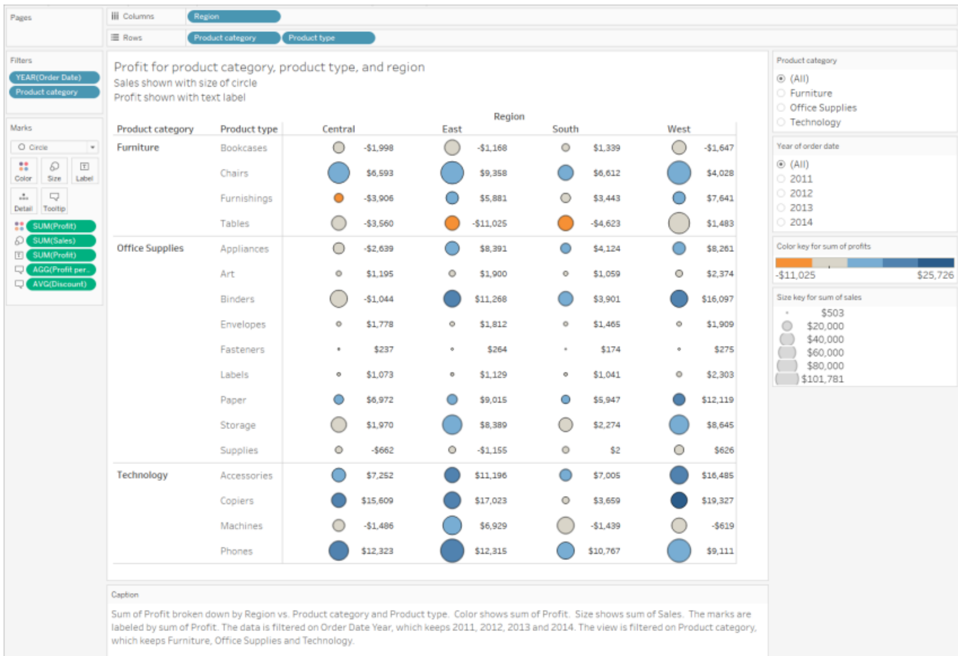
Less accessible: This example shows the same scatter plot with no use of color, shape, or size to create contrast. The position of the marks provides meaningful information, but the view could make use of more visual cues.

More accessible: This example shows a calculated field named Quadrant being used on color and shape. Position, color, and shape are used to distinguish each mark. The axes also clarify the mark position and meaning.



Putting it all together

This next example shows how text, shape, color, and size can all work together to create an accessible view. Sales are represented by the size of each circle (mark). Profit represented by color and text. The title and caption for the view are shown and have been edited to provide more context. The filter and legend titles have also been edited to provide more context.



Walkthrough: The journey from less accessible to more accessible

Let's convert a viz that isn't very accessible into one that is. The data in this starting view has a more granular level of detail and a structure that could contribute to cognitive overload with a screen reader.

This view is designed to show total sales for each sales region, along with the profit ratio, to determine which types of products seem to be selling well but are not actually profitable. Profit and Profit per Order are also shown.

We'll make changes to it based on the concepts and best practices mentioned previously in this article.

Step 1: Reduce the level of detail in the view.

Because the Product Name dimension is on Detail, every product in the data set is being represented with a mark in the view.

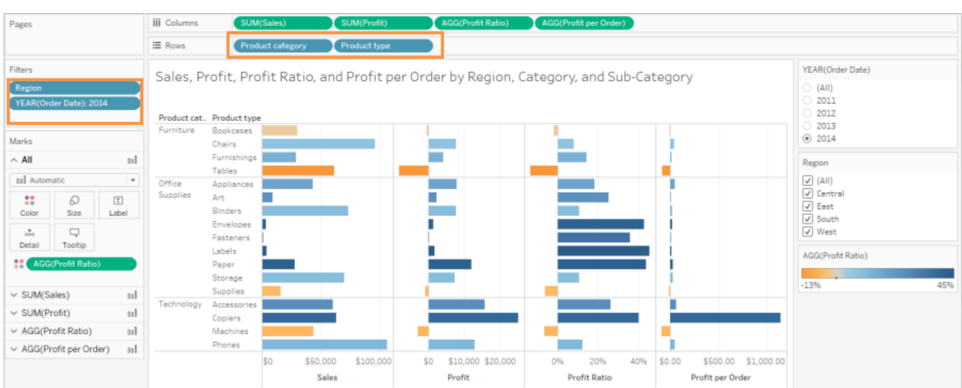
- Remove Product Name from Detail. This change reduces the number of marks in the view from over 20,000 to 272.

Reducing the number of marks in the view helps to ensure the embedded view will be rendered on the browser instead of the server.

Step 2: Create a more aggregated view of the data.

1. Reduce the number of measures on Columns and the number of dimensions on Rows.
2. Remove Region (we can use a filter for Region). The number of marks changes to 68.
3. Replace Category and Sub-Category (both dimensions that are part of the Product hierarchy) with copies (not part of a hierarchy) that have been renamed Product category and Product type. Remove the Sub-Category filter.

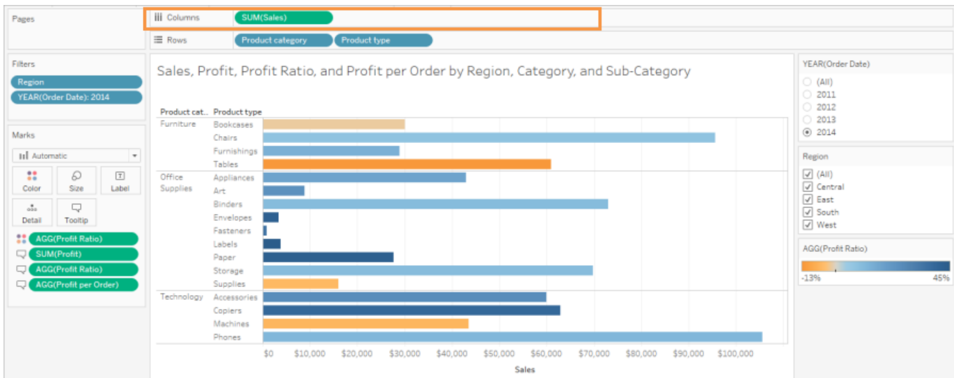
   Doing this prevents users from expanding the hierarchy in the embedded view, which might increase the number of marks in the view and result in the view being rendered on the browser instead of the server.



The view is already more aggregated and simplified, but we'll keep going.

- Move the Profit, Profit per Order, and Profit Ratio from the Columns shelf to **Tooltip**.

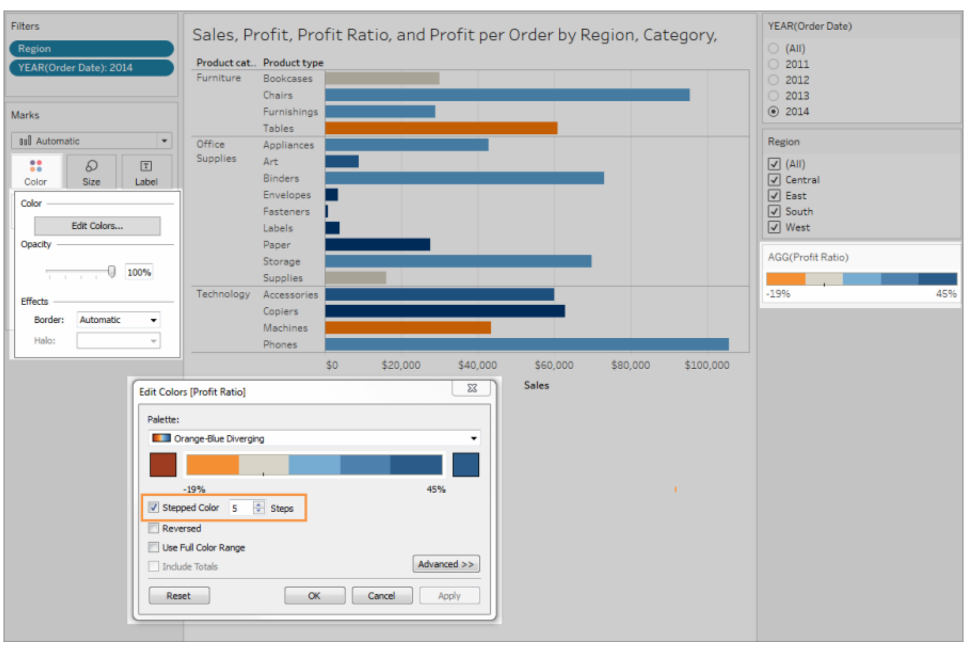
The number of marks is now 17. The underlying details in the data will still be available in the View Data window, but the reduced number of marks ensures the view will be WCAG-conformant because it will be rendered in the browser.



Step 3: Check the color palette settings.

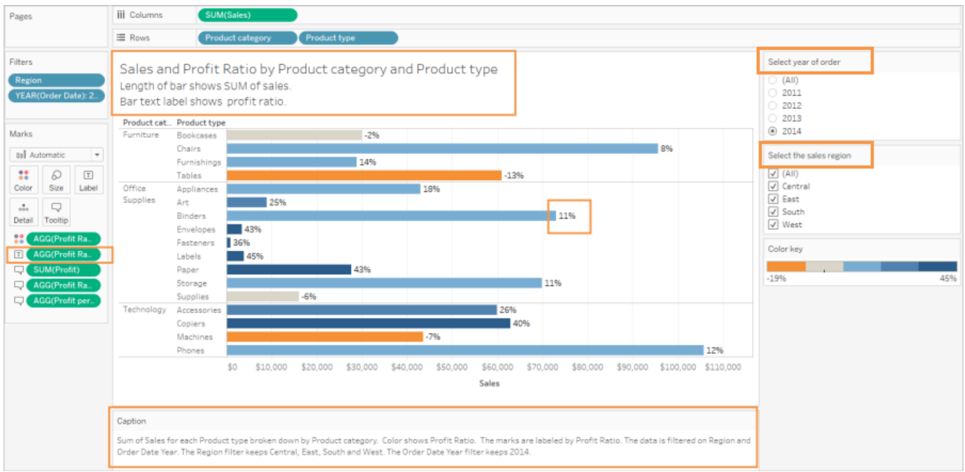
The palette is set to Orange-Blue Diverging.

- Set **Stepped Color** to 5.



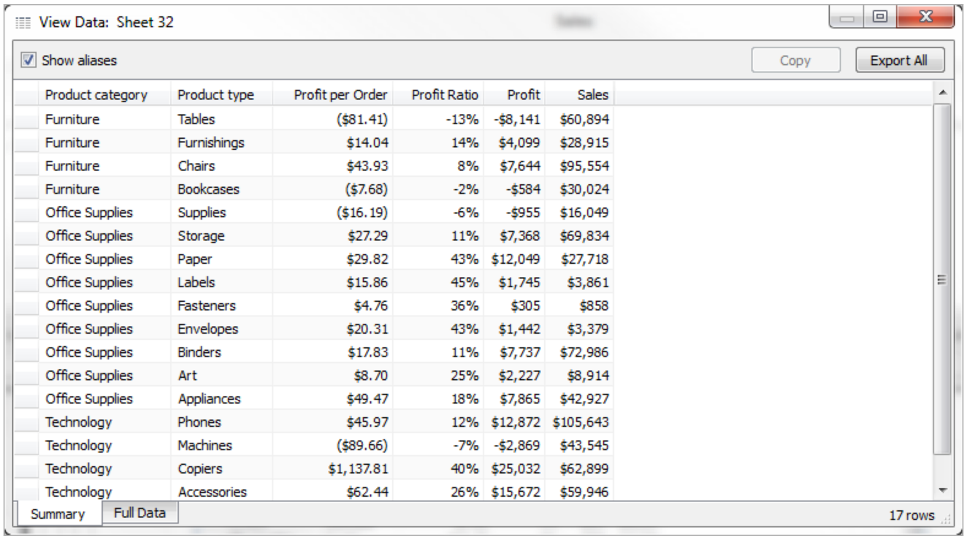
Step 4: Update and show text in the view.

1. Change the Title to make sense with the new view.
2. Show the worksheet Caption.
3. Show Mark labels.
4. Drag Profit Ratio to Label in the Marks card.
5. Edit the filter and legend titles.



Step 5: Open the View Data window to the Summary tab.

This is the text that screen readers will read for this view. The four measures are included in the underlying data for each product type.


View Data: Sheet 32

Show aliases | Copy | Export All

Product category	Product type	Profit per Order	Profit Ratio	Profit	Sales
Furniture	Tables	($81.41)	-13%	-$8,141	$60,894
Furniture	Furnishings	$14.04	14%	$4,099	$28,915
Furniture	Chairs	$43.93	8%	$7,644	$95,554
Furniture	Bookcases	($7.68)	-2%	-$584	$30,024
Office Supplies	Supplies	($16.19)	-6%	-$955	$16,049
Office Supplies	Storage	$27.29	11%	$7,368	$69,834
Office Supplies	Paper	$29.82	43%	$12,049	$27,718
Office Supplies	Labels	$15.86	45%	$1,745	$3,861
Office Supplies	Fasteners	$4.76	36%	$305	$858
Office Supplies	Envelopes	$20.31	43%	$1,442	$3,379
Office Supplies	Binders	$17.83	11%	$7,737	$72,986
Office Supplies	Art	$8.70	25%	$2,227	$8,914
Office Supplies	Appliances	$49.47	18%	$7,865	$42,927
Technology	Phones	$45.97	12%	$12,872	$105,643
Technology	Machines	($89.66)	-7%	-$2,869	$43,545
Technology	Copiers	$1,137.81	40%	$25,032	$62,899
Technology	Accessories	$62.44	26%	$15,672	$59,946

Summary | Full Data | 17 rows

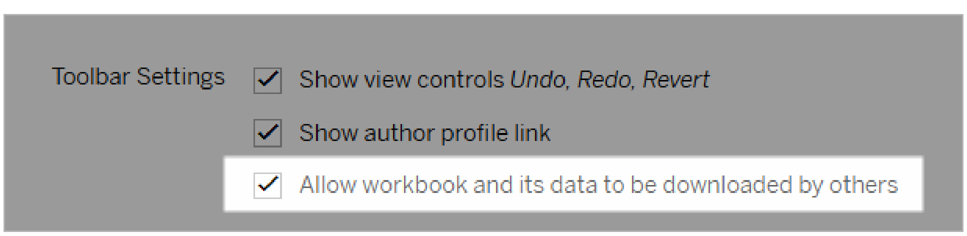
Publish and share the view

After you create your views, to make them WCAG-conformant, you can:

- Publish views to Tableau Server or Tableau Online and share the link, or embed the view in a WCAG-compliant web page. For information about how to publish a view, see **Comprehensive Steps to Publish a Workbook** on page 3058.
- Web editing is not supported in WCAG-conformant views, so Web Edit permissions must be turned off in the published workbook.
- Make sure users have permissions to access to the views and are able to view the underlying data in the View Data window.

Publishing to Tableau Public

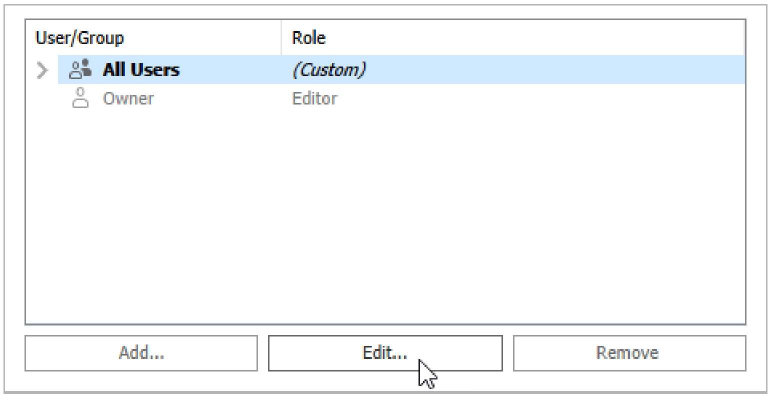
After creating your workbook and saving to Tableau Public, the "Allow workbook and its data to be downloaded by others" option must be enabled in order for the end user to be able to open the View Data window.



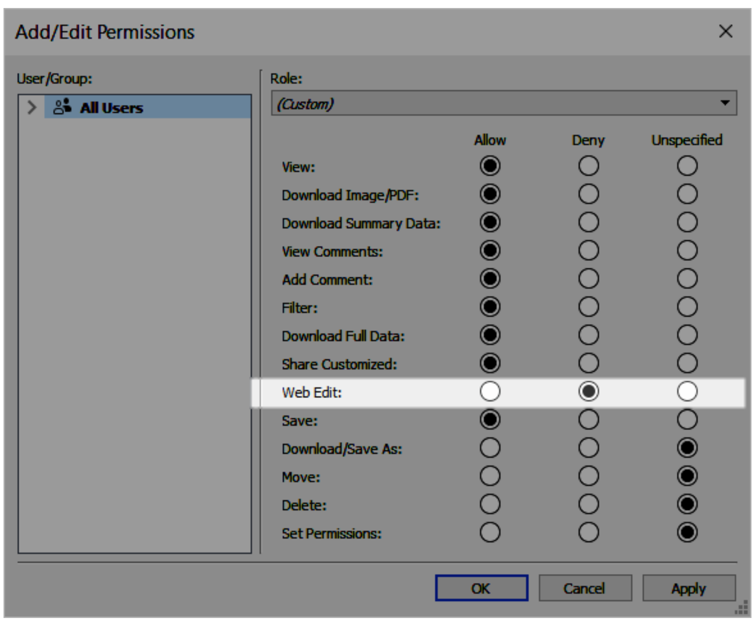
Turn off Web Edit permissions

To prevent your users from seeing the Edit option in the view toolbar, set the Web Edit permission to Deny when you publish your workbook. You should also let your Tableau Online administrator know when you publish workbooks that should not allow users to access web editing mode.

1. In Desktop, select Server > Publish Workbook.
2. Under Permissions, select Edit, select All Users, and then select Edit again.



3. For the Web Edit permissions, select Deny. Click Apply, then click Ok.



Share or embed the view

When you are ready to share, go to the view in Tableau Server or Tableau Online and click the **Share** button in the view toolbar to copy the link or share with a user on the site. The view toolbar and navigation panel are supported for conformance, and include a "Skip to Content" option for keyboard navigation as well as a focus indicator.

Alternately, views can be embedded in the HTML code of a WCAG-conformant web page. For more information about how to embed views, see **Embed Views into Webpages** on page 3263.

Sample embed code

```
<script type='text/javascript' src='http://Your_
Server/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width: 1256px; height:
818px;'>
  <object class='tableauViz' width='1256' height='818'
style='display:none;'>
    <param name='host_url' value='http%3A%2F%2FYour_Server%2F' />
    <param name='site_root' value='' />
    <param name='name' value='name of the visualization' />
    <param name='tabs' value='yes' />
    <param name='showShareOptions' value='true' />
  </object>
</div>
```

In a view URL:

```
https://myserver/t/Sales/views/MyCoSales/SalesScoreCard?embed=y
```

Embedded views must be rendered on the browser (not the server)

Server-rendered views are not WCAG-conformant; only client-rendered views are WCAG-conformant. To ensure an embedded view is rendered in the browser, make sure the view has less than 1000 marks, or set *render* to *true* in the URL for the embedded view.

For example:

```
http://localhost/views/Supplies/MyView?:render=true
```

Client-side rendering is enabled (*true*) by default.

About permissions

Users who need access to the underlying data will need the proper site role and permissions to view and interact with your embedded views. This includes having access to the projects and workbooks that contain the views. At the very minimum, your users will need View and Download Summary Data permissions. The ability to view the underlying data is necessary for users who use screen readers.

Users viewing the embedded view should have:

- Site role of Viewer, Interactor, or Guest.

  Note: The Viewer site role is more limited and will not be able to use filters or view the Full Data tab.

- **View** permission for the project that the workbook is in.
- **View** permissions for the workbook and its views.
- **Download Summary Data** permission to read the underlying data in the view on the **Summary** tab of the View Data window.
- **Download Full Data** permission if the user needs to be able to read all of the available data in the data source on the **Full Data** tab of the View Data window.

For information about how to set permissions for workbooks and views, see Set Content Permissions in the online help.

Test the view

Test your embedded view using keyboard shortcuts for embedded views. Provide your users with this information in your user documentation.

Maps and Geographic Data Analysis in Tableau

When it comes to plotting your data on a map, Tableau has got you covered. This section provides all the documentation you need to get started with creating maps in Tableau.

Start with these topics

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Getting Started with Mapping (3 minutes) and Maps in Tableau (4 minutes). Use your tableau.com account to sign in.

Get Started Mapping with Tableau

This tutorial walks you through some of the most common tasks you might perform when creating maps in Tableau.

You'll learn how to connect to and join geographic data; format that data in Tableau; create location hierarchies; build and present a basic map view; and apply key mapping features along the way.

If you're new to building maps in Tableau, this a great place to start.

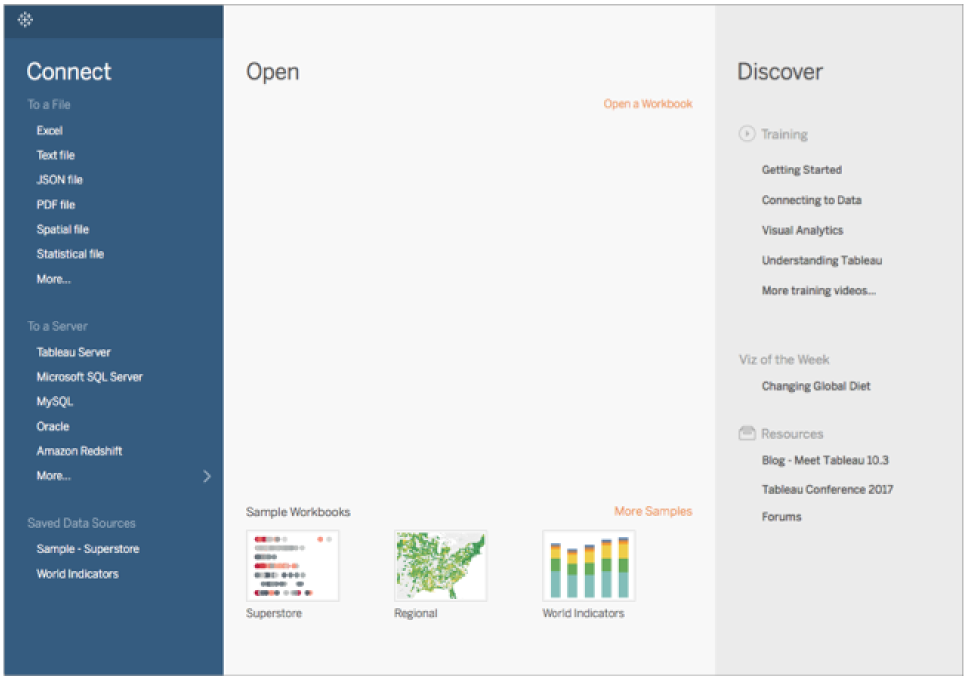
Step 1: Connect to your geographic data

Geographic data comes in many shapes and formats. When you open Tableau Desktop, the start page shows you the connectors available in the left **Connect** pane. These are how you will connect to your data.

You can work with geographic data by connecting to spatial files, or you can connect to location data stored in spreadsheets, text files, or on a server.

Spatial files, such as a shapefile or geoJSON file, contain actual geometries (points, lines, or polygons), whereas text files or spreadsheets contain point locations in latitude and longitude coordinates, or named locations that, when brought into Tableau, connect to the Tableau geocoding (stored geometries that your data references).

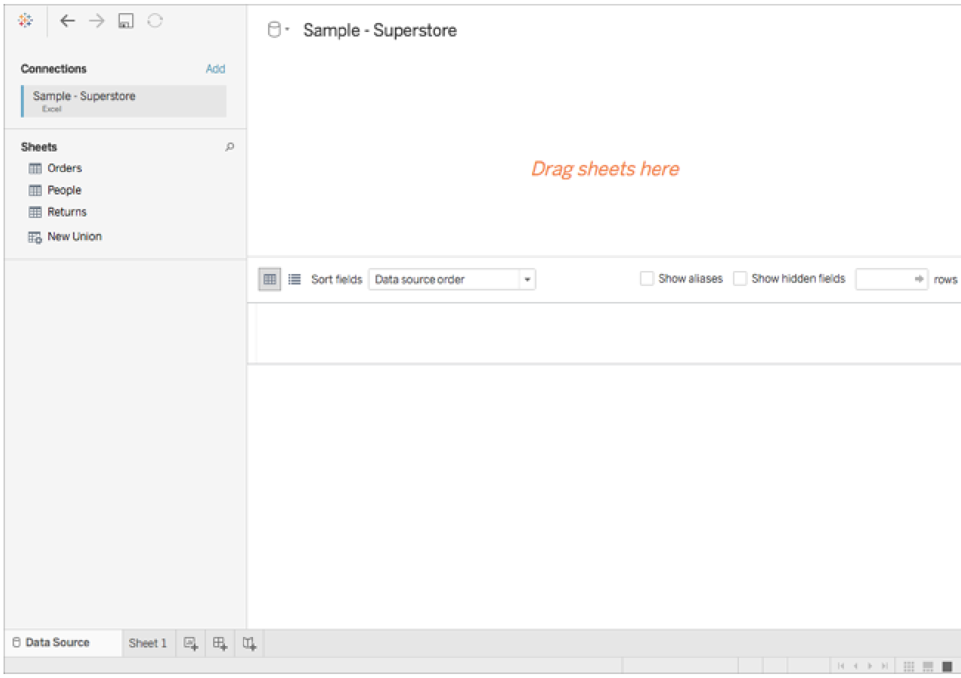
For a complete list of connections Tableau supports, see the list of Data Connections on the Tableau website.



For this tutorial, you are going to connect to an Excel file that comes with Tableau Desktop. It contains location names that Tableau can geocode. When you build a map view, the location names reference the geometries stored in the Tableau Map Service based on the geographic role you assign to the field. You'll learn more about geographic roles later in this tutorial.

1. Open Tableau Desktop.
2. In the Connect pane, click **Excel**.
3. Navigate to **Documents** > **My Tableau Repository** > **Data Sources**, and then open the **Sample - Superstore.xls** file.

Once you connect to the data source, your screen will look like this:



This is called the Data Source page, and it is where you can prepare your location data for use in Tableau.

Some of the tasks you can perform on the Data Source page include the following, but you don't have to do all these things to create a map view:

- Adding additional connections and joining your data
- Adding multiple sheets to your data source
- Assigning or changing geographic roles to your fields
- Changing the data type of your columns (from numbers to strings, for example)
- Renaming columns
- Splitting columns, such as splitting a full address into multiple columns for street, city, state, and postal code

For more information about the Data Source page and some of the tasks you can perform while on it, see the topics in the **Set Up Data Sources** on page 741 section.

Step 2: Join your data

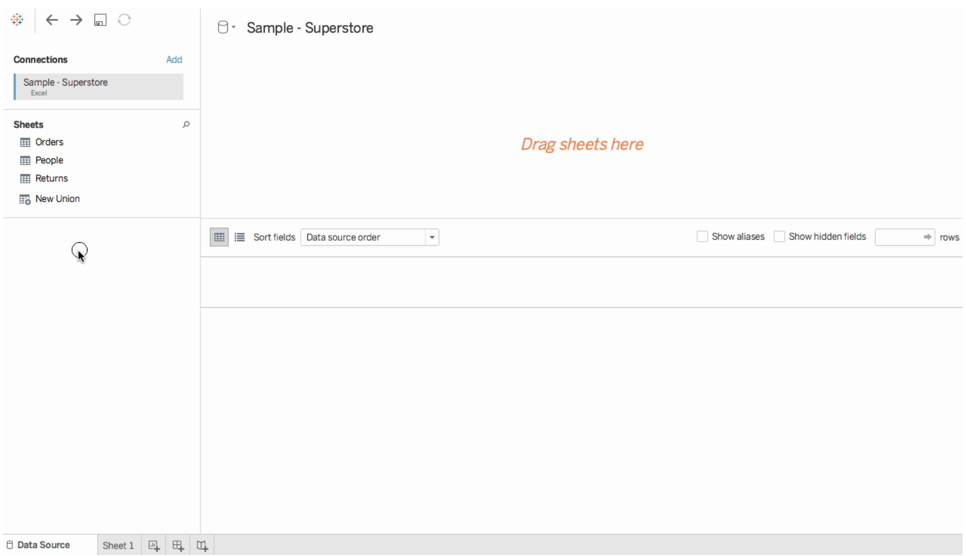
Your data is often held in multiple data sources or sheets. As long as those data sources or sheets have a column in common, you can join them in Tableau. Joining is a method for combining the related data on those common fields. The result of combining data using a join is a virtual table that is typically extended horizontally by adding columns of data.

Joining is often necessary with geographic data, particularly spatial data. For example, you can join a KML file that contains custom geographies for school districts in Oregon, U.S. with an Excel spreadsheet that contains demographic information about those school districts.

For this example, you will join two sheets in the Sample-Superstore data source.

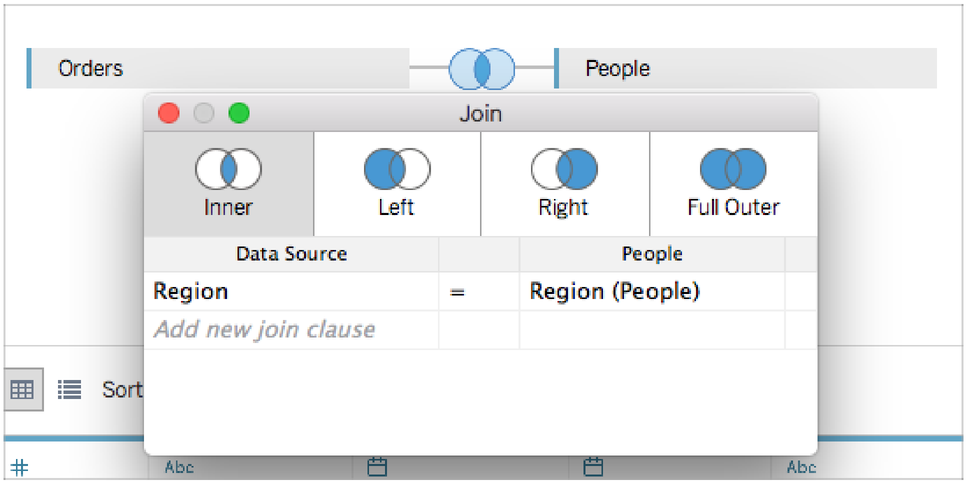
1. On the left side of the Data Source page, under Sheets, double-click **Orders**.
2. Under Sheets, double-click **People**.

   Tableau creates an inner-join between the two spreadsheets, using the Region column from both spreadsheets as the joining field. Now there is a sales person assigned to every location in your data source, as well as to regions.



   To edit this join, click the join icon (the two circles). You can edit the join in the Join dialog box that opens. For more information about joining data in Tableau, see **Join Your**

Data on page 883.



Step 3: Format your geographic data in Tableau

After you set up your data source, you might need to prepare your geographic data for use in Tableau. Not all of these procedures will always be necessary to create a map view, but it's important information to know when it comes to preparing geographic data for use in Tableau.

Depending on the type of map you want to create, you must assign certain data types, data roles, and geographic roles to your fields (or columns).

For example, in most cases, your latitude and longitude fields should have a *data type* of **number (decimal)**, a *data role* of **measure**, and be assigned the **Latitude** and **Longitude** *geographic roles*. All other geographic fields should have a *data type* of **string**, a *data role* of **dimension**, and be assigned the appropriate geographic roles.

Note: If you are connecting to a spatial file, a Geometry field is created. It should have a data role of measure.

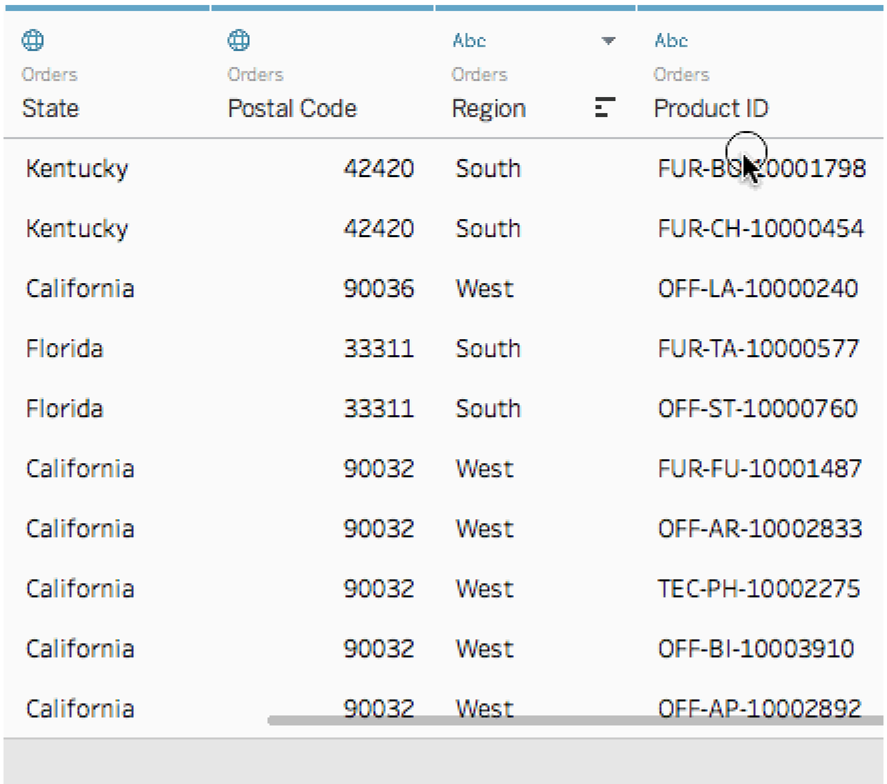
This step demonstrates how to format your geographic data to meet this criteria.

Change the data type of a column

When you first connect to geographic data, Tableau assigns data types to all of your columns. These data types include Number (decimal), Number (whole), Date and Time, Date, String, and Boolean. Sometimes Tableau does not get these data types right, and you must edit them. For example, Tableau might assign a Postal Code column a data type of Number (whole). To create map views, your Postal Code data must have a data type of String.

To change the data type of a column:

1. On the Data Source page, click the data type icon (the globe) for Postal Code and select **String**.



For more information about data types, see **Data Types** on page 325.

Assign geographic roles to your geographic data

In Tableau, a *geographic role* associates each value in a field with a latitude and longitude value. When you assign the correct geographic role to a field, Tableau assigns latitude and longitude values to each location in that field by finding a match that is already built in to the installed geocoding database. This is how Tableau knows where to plot your locations on the map.

When you assign a geographic role to a field, such as State, Tableau creates a Latitude (generated) field and a Longitude (generated) field.

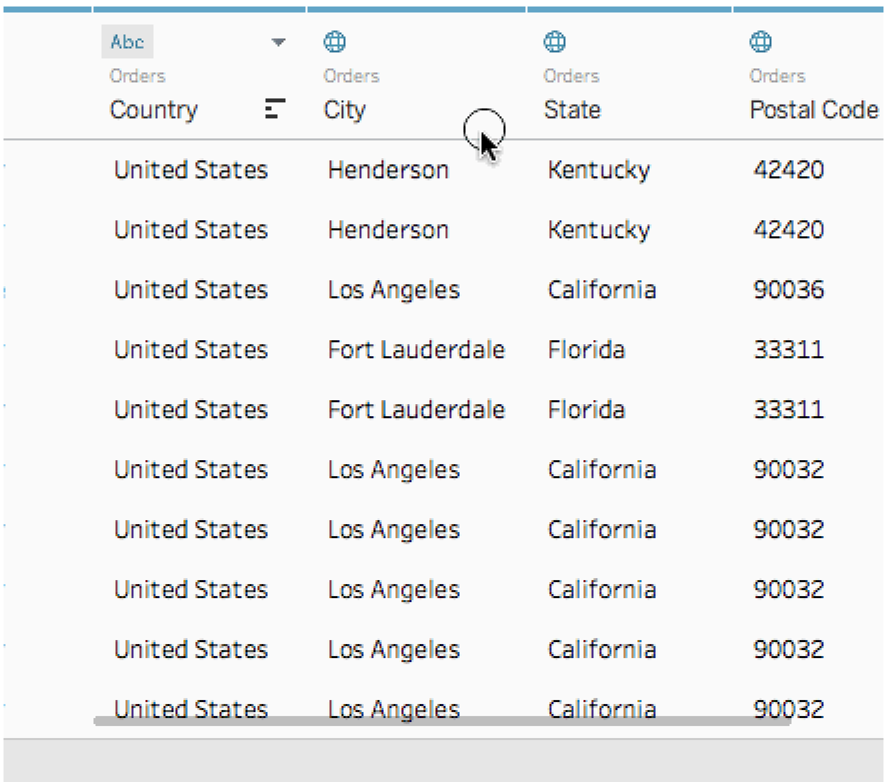
Geographic roles are sometimes automatically assigned to your data, such as in this example. You can tell a geographic role has been assigned to your data because the column includes a globe icon.

If a geographic role is not automatically assigned, you can manually assign one to your field. You don't need to do so for this example, but it's important to know how so you can do it for your own data.

To assign or edit a geographic role:

1. On the Data Source page, click the globe icon.
2. Select **Geographic Role**, and then select a role that best matches your data.

   For example, in this case, the Country column does not have a geographic role assigned to it, so the Country/Region geographic role is assigned.



Abc Orders Country	Orders City	Orders State	Orders Postal Code
United States	Henderson	Kentucky	42420
United States	Henderson	Kentucky	42420
United States	Los Angeles	California	90036
United States	Fort Lauderdale	Florida	33311
United States	Fort Lauderdale	Florida	33311
United States	Los Angeles	California	90032
United States	Los Angeles	California	90032
United States	Los Angeles	California	90032
United States	Los Angeles	California	90032
United States	Los Angeles	California	90032

Note: If you have difficulties assigning geographic roles to your data, or have data that is not built in to the Tableau map server, there are a few things you can do to get that data into Tableau. See **Assign Geographic Roles** on page 1834.

Change from dimensions to measures

When you connect to geographic data, Tableau also assigns data roles to all of your columns. A column can be a *dimension* or *measure*. In most cases, your latitude and longitude columns should be measures. For special cases, such as if you want to plot every location in your data source on a map without the ability to drill up or down a level of detail (such as from City to State), they can be dimensions. A great example of this is a point distribution map.

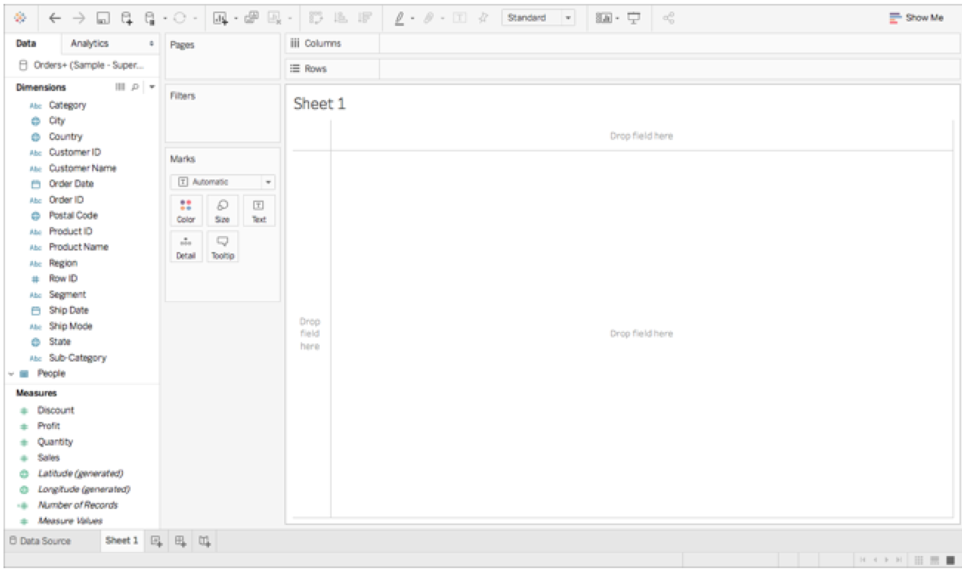
The rest of your geographic data should be dimensions.

You don't need to change the data role of a column for this example, but it's important to know how so you can do it for your own data. Feel free to practice here. You can always undo any changes you make.

To change the data role of a column:

1. On the Data Source page, click **Sheet 1**.

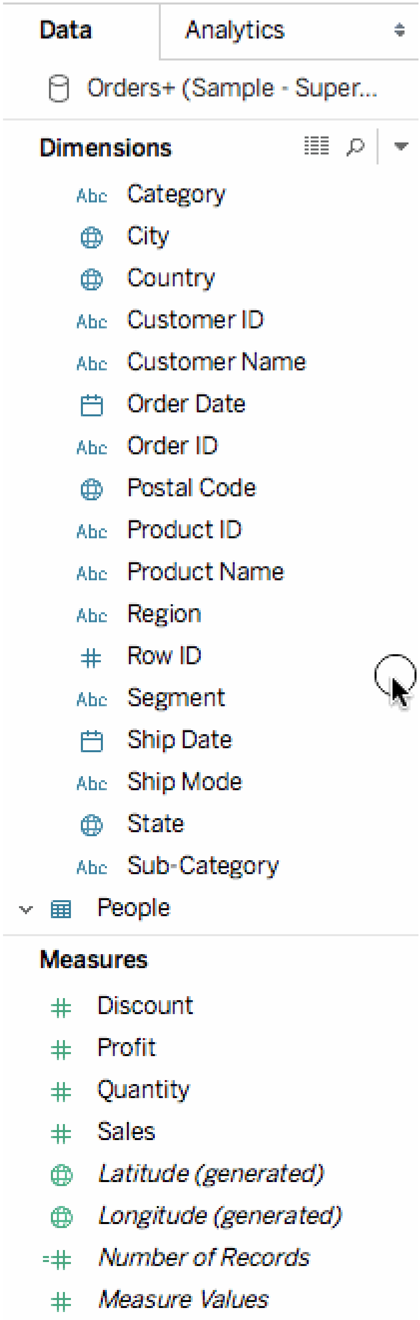
   Your workspace updates to look like this:

   This is called a worksheet, and it is where you will build your map. On the left-side of the screen is the **Data** pane. All of the columns in your data source are listed as fields in this pane. For example, Country and State. These fields contain all the raw data in your columns. Note that Tableau has generated a Latitude and Longitude field (*Latitude (generated)* and *Longitude (generated)*). This is because you assigned geographic roles to your data.

The fields in the data pane are divided into measures and dimensions. The fields placed in the Dimensions section of the Data pane are often categorical data, such as Date and Customer ID, while the fields placed in the Measures section of the Data pane are often quantitative data, such as Sales and Quantity.

2. In the **Data** pane, under Dimensions, select a field, such as Row ID, and drag it down to the Measures section.



The field is added to the Measures section and changes from blue to green. You just converted a Dimension to a Measure. To convert a field from a measure to a dimension, drag the field from the Measures section up to the Dimensions section.

For more information, see **Dimensions and Measures, Blue and Green** on page 314.

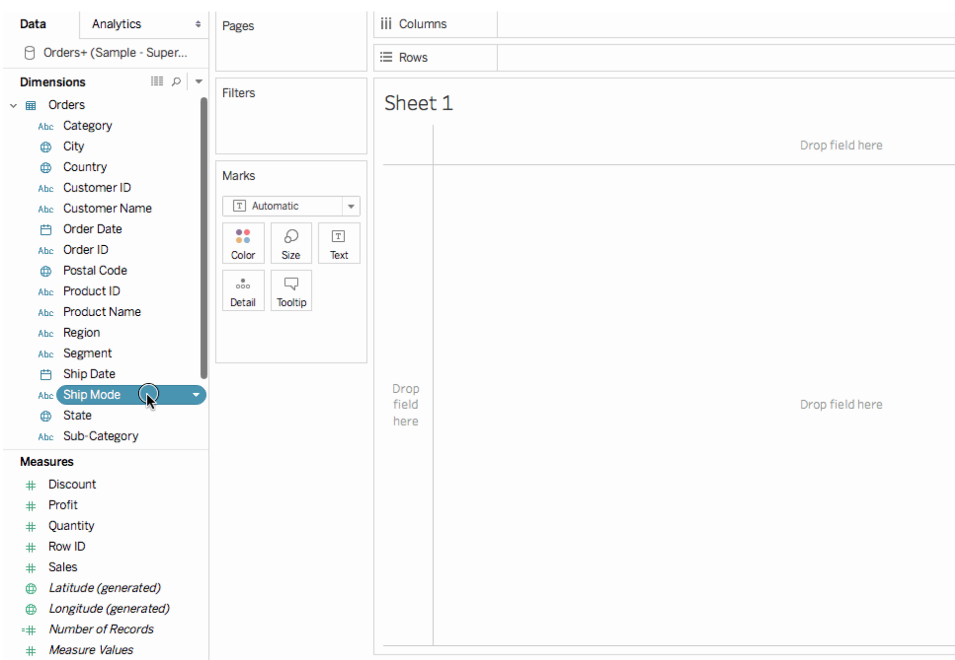
Step 4: Create a geographic hierarchy

Now that you are in the worksheet space, you can create geographic hierarchies. This is not required to create a map view, but creating a geographic hierarchy will allow you to quickly drill into the levels of geographic detail your data contains, in the order you specify.

To create a geographic hierarchy:

1. In the Data pane, right-click the geographic field, **Country**, and then select **Hierarchy** > **Create Hierarchy**.
2. In the Create Hierarchy dialog box that opens, give the hierarchy a name, such as Mapping Items, and then click **OK**.

   At the bottom of the Dimensions section, the Mapping Items hierarchy is created with the Country field.
3. In the Data pane, drag the State field to the hierarchy and place it below the Country field.
4. Repeat step 3 for the City and Postal Code fields.



When you are finished, your hierarchy should be in the following order:

- Country
- State

- City
- Postal Code

Step 5: Build a basic map

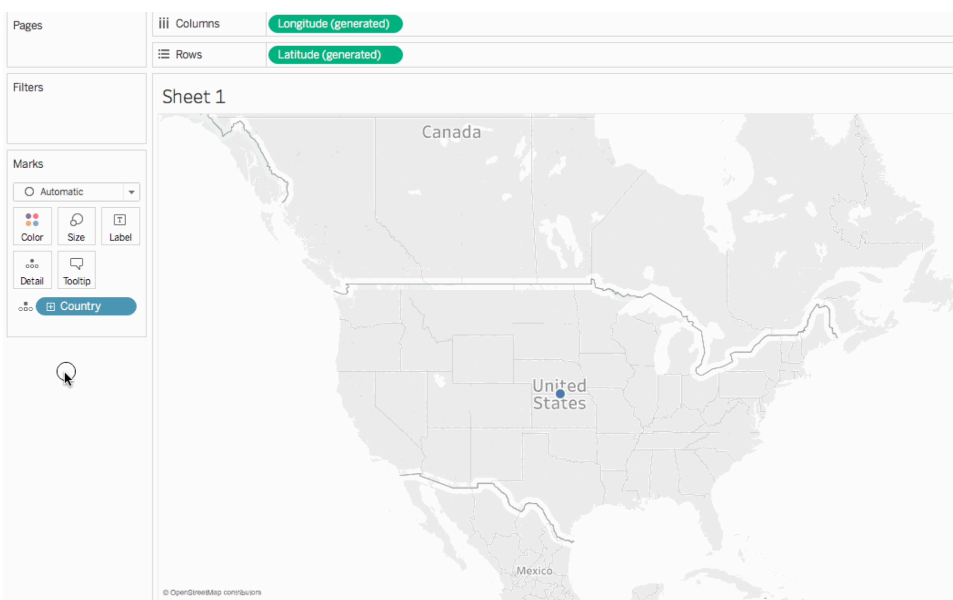
Now that you have connected to and joined your data, formatted your data, and built a geographic hierarchy, you are now ready to start building your map. You will start by building a basic map view.

1. In the Data pane, double-click **Country**.

   The Country field is added to Detail on the Marks card, and Latitude (generated) and Longitude (generated) are added to the Columns and Rows shelves. A map view with one data point is created. Since a geographic role is assigned to Country, Tableau creates a map view. If you double-click any other field, such as a dimension or measure, Tableau adds that field to the Rows or Columns shelf, or the Marks card, depending on what you already have in the view. Geographic fields are always placed on Detail on the Marks card, however.

   Since this data source only contains one country, (United States), that is the only data point shown. You will need to add more levels of detail to see additional data points. Since you created a geographic hierarchy, this is easy.

2. On the Marks card, click the **+** icon on the **Country** field.



The State field is added to Detail on the Marks card and the map updates to include a data point for every state in the data source.

If you did not create a hierarchy, the + icon on the Country field will not be available. In this case, to add State as another level of detail, manually drag **State** from the **Data** pane to **Detail** on the Marks card.

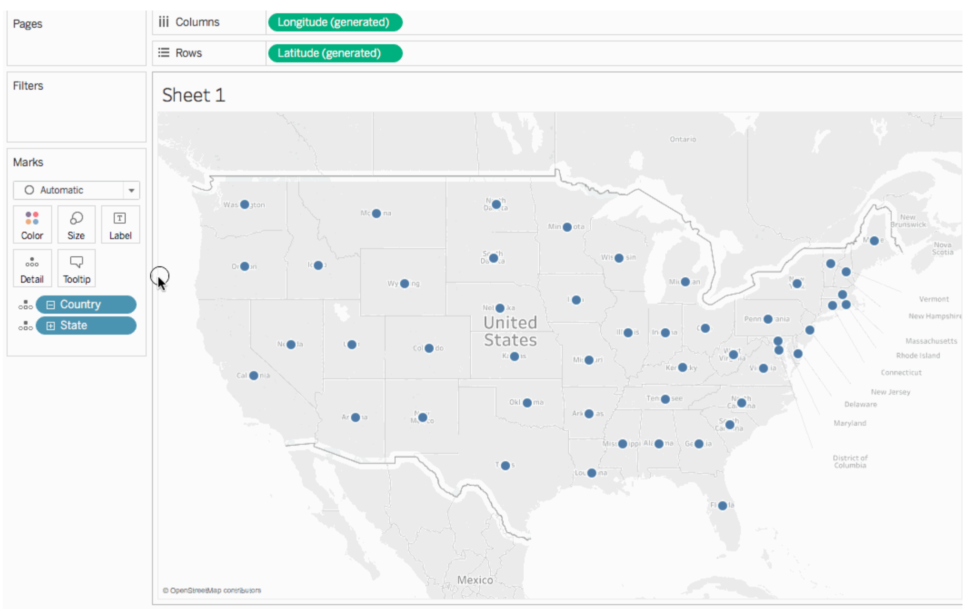
Congratulations! You now have a basic map view that you can customize and build upon in the next steps.

Step 6: Change from points to polygons

The default map type in Tableau is often a point map. When you have geographic roles assigned to your geographic data, however, it's easy to change those data points to polygons.

> **Note**: Filled maps are not available for cities or airports.

1. On the Marks card, click the Mark Type drop-down and select **Filled Map**.



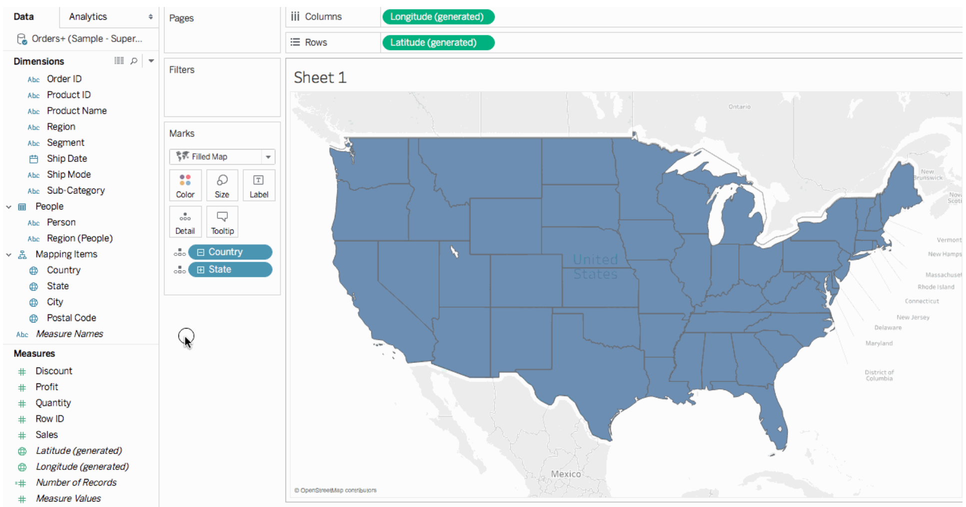
The map updates to a polygon map.

Step 7: Add visual detail

You can add measures and dimensions to the Marks card to add visual detail to your view. In this example, you will add color and labels to the view.

Add color

- From Measures, drag **Sales** to **Color** on the Marks card.



Each state is colored by sum of sales. Since Sales is a measure, a qualitative color palette is used. If you place a dimension on color, then a categorical color palette is used.

Add labels

1. From Measures, drag **Sales** to **Label** on the Marks card.

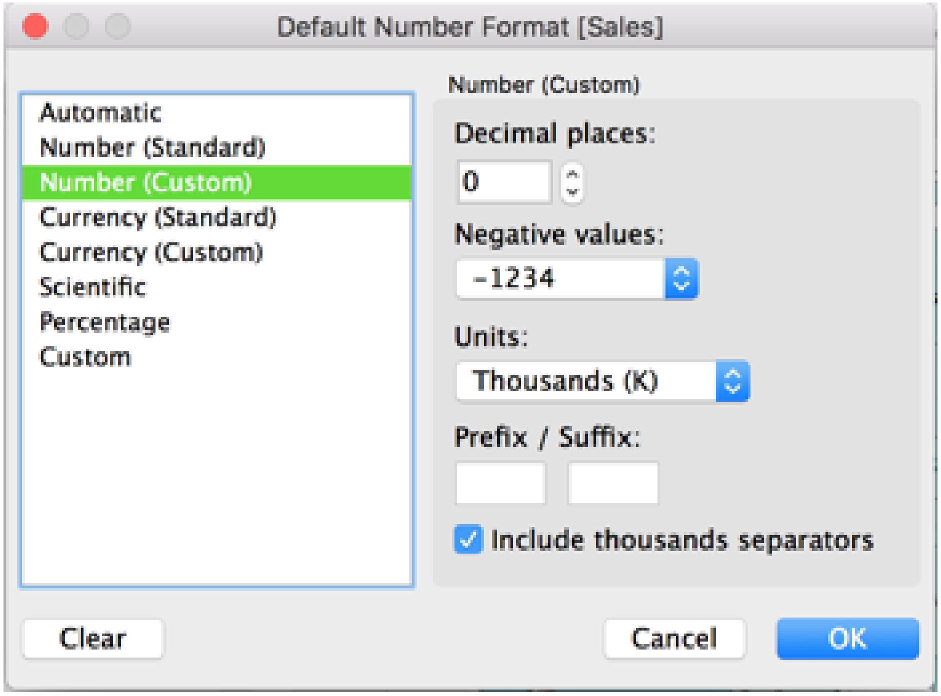
   Each state is labeled with sum of sales. The numbers need a little bit of formatting, however.

2. In the Data pane, right-click **Sales** and select **Default Properties** > **Number Format**.

3. In the Default Number Format dialog box that opens, select **Number (Custom)**, and then do the following:

   - For **Decimal Places**, enter **0**.
   - For **Units**, select **Thousands (K)**.
   - Click **OK**.

The labels and the color legend update with the specified format.



Step 8: Customize your background map

The background map is everything behind your marks (borders, oceans, location names, etc.) You can customize the style of this background map, as well as add map layers and data layers. In addition to customizing the background maps, you can also connect to your own WMS server or Mapbox map. For more information, see **Use Web Map Service (WMS) Servers** on page 1996 and **Use Mapbox Maps** on page 1993.

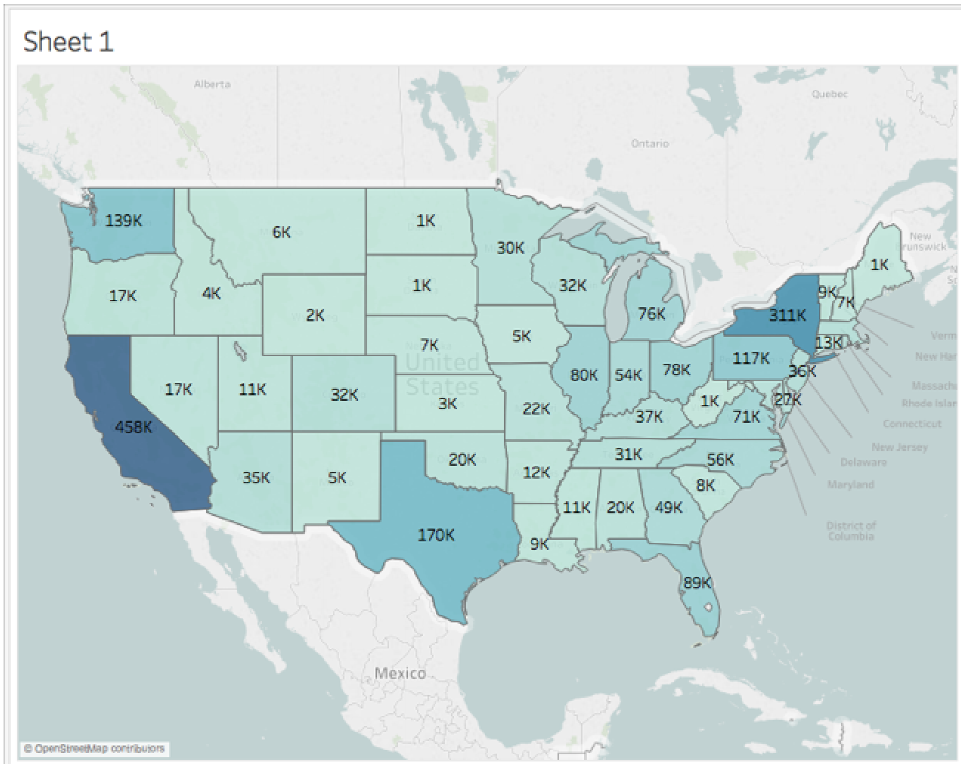
To customize your background map:

1. Select **Map** > **Map Layers**.

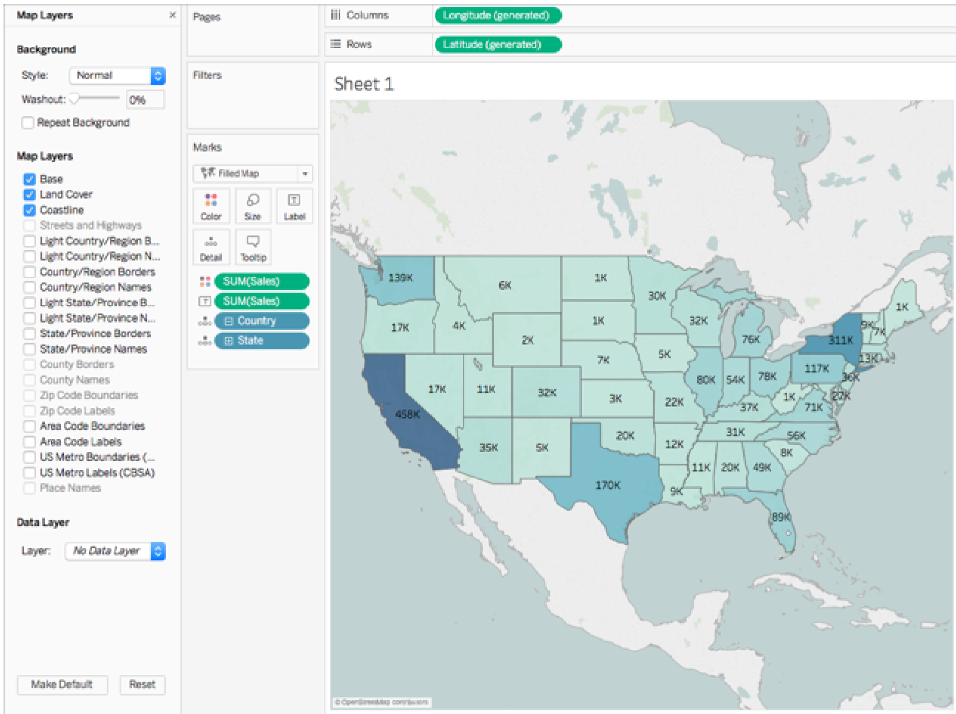
   The Map Layers pane appears on the left side of the workspace. This is where all background map customization happens.

2. In the Map Layers pane, click the **Style** drop-down and select **Normal**.

   The background map updates to look like this:



3. In the Map Layers pane, under Map Layers, select **Coastlines**, and then clear **Country/Region Borders**, **Country/Region Names**, **State/Province Borders**, and **State/Province Names**.



4. At the top of the Map Layers pane, click the **X** to return to the **Data** pane.

   The background map is now simplified to draw attention to your data.

Step 9: Create custom territories

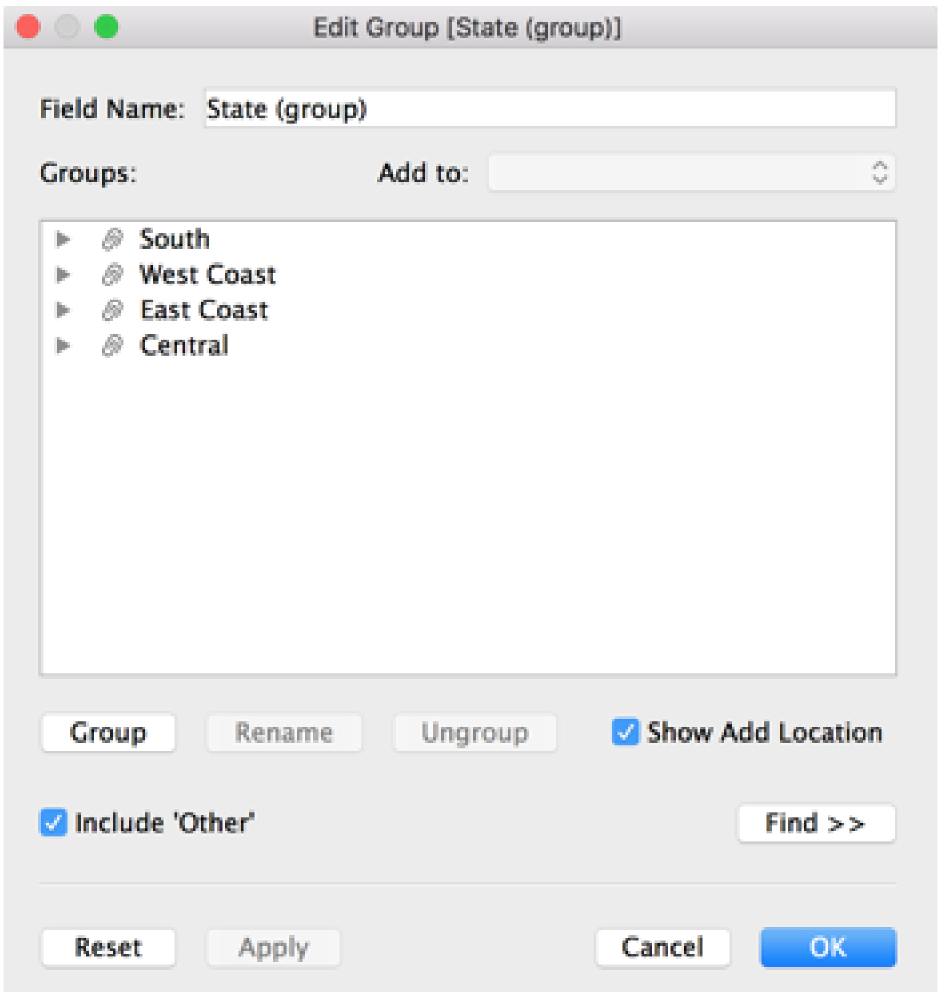
As you build your map view, you might want to group existing locations together to create your own territories or regions, such as sales territories for your organization.

1. In the Data pane, right-click **State** and select **Create** > **Group**.
2. In the Create Group dialog box that opens, select **California**, **Oregon**, and **Washington** , and then click **Group**. Each group you create represents a territory.

   Note: To multi-select, hold down Ctrl (Command on Mac) as your select states.
3. Right-click the new group you just created and select **Rename**.
4. Rename the group, **West Coast**.
5. For the next territory, select **Alabama**, **Florida**, **Georgia**, **Louisiana**, **Mississippi**, **South Carolina**, and **Texas**, and then click **Group**.
6. Rename this group, **South**.
7. For the third territory, select **Connecticut**, **Delaware**, **District of Columbia**, **Main**,

Maryland, **Massachusetts**, **New Hampshire**, **New Jersey**, **New York**, **Pennsylvania**, **Rhode Island**, **Vermont**, and finally, **West Virginia**, and then click **Group**.

8. Rename this group, **East Coast**.
9. Select **Include Other** to group the remaining states.
10. Rename the **Other** group, **Central**.

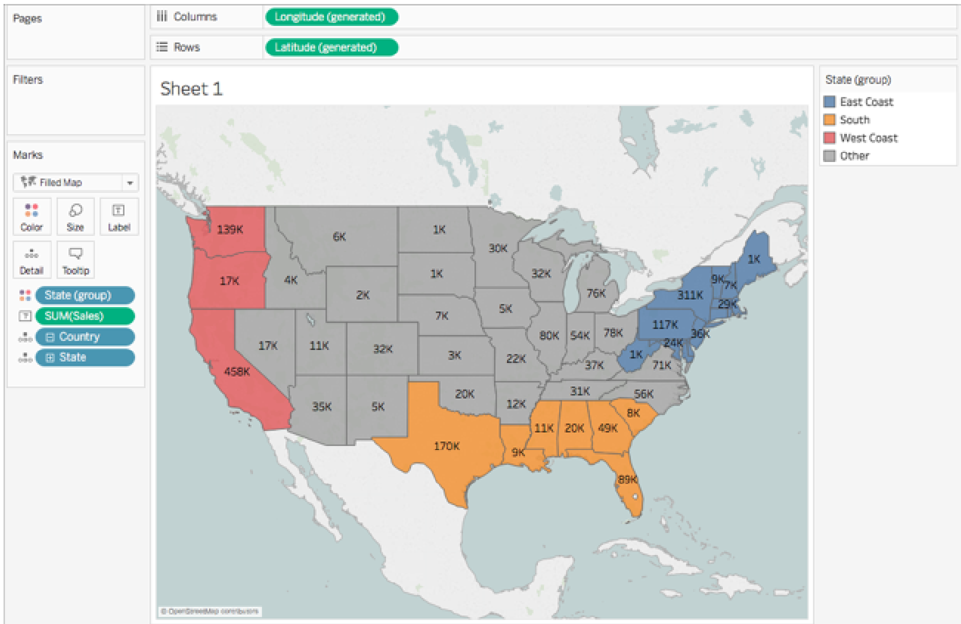


11. Click **OK**.

    A State (group) field appears in the **Data** pane beneath your other mapping items.

12. From the Data pane, drag **State (group)** to **Color** on the Marks card.

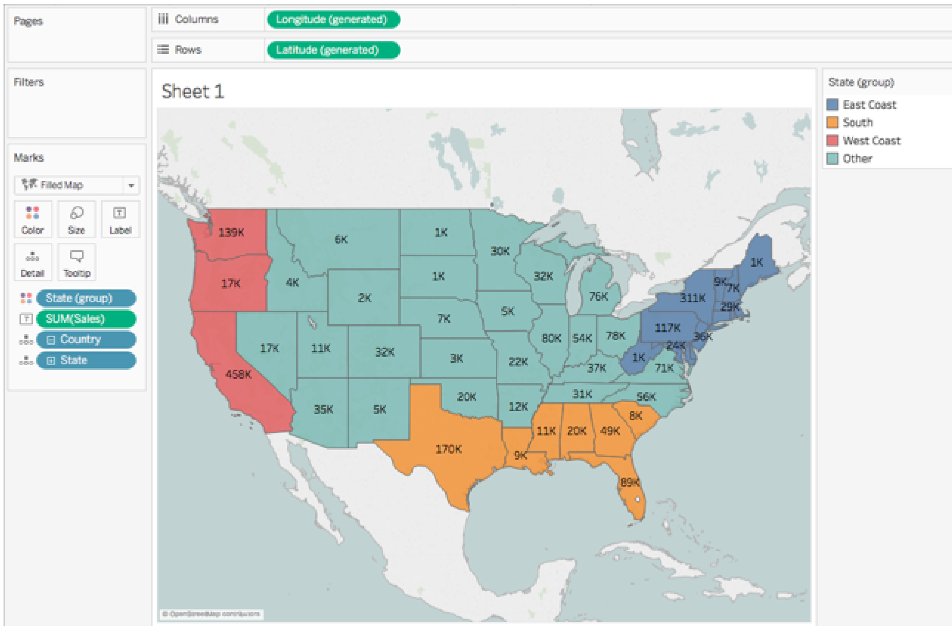
The view updates to look like this:



Notice that each group has a different color.

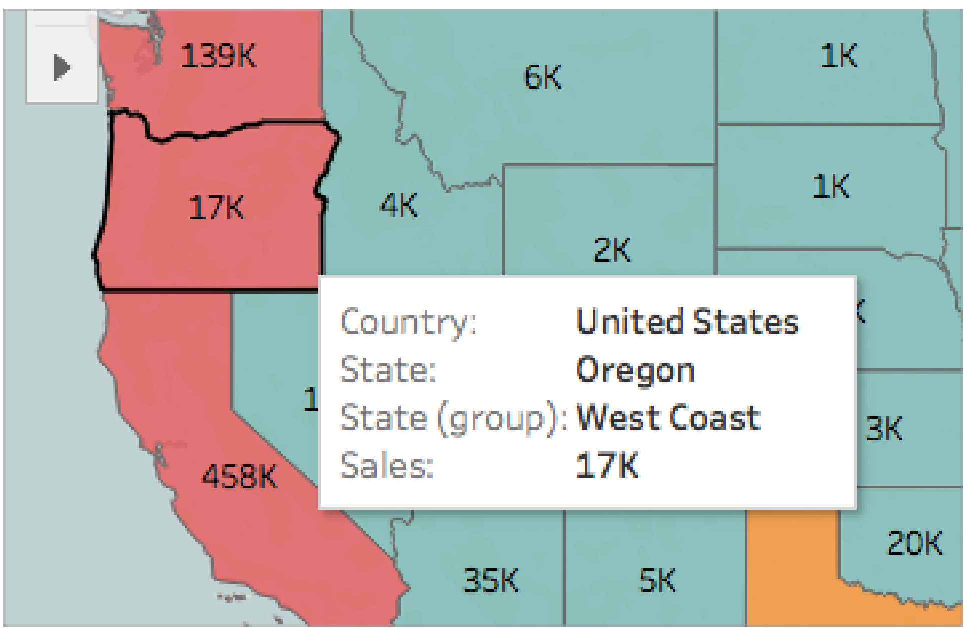
13. On the Marks card, click the **Color** icon and select **Edit Colors**.
14. In the Edit Colors dialog box that appears, select **Assign Palette**, and then click **OK**.

    The marks update with new colors.



15. From Measures, drag **Sales** to **Tooltip** on the Marks card.

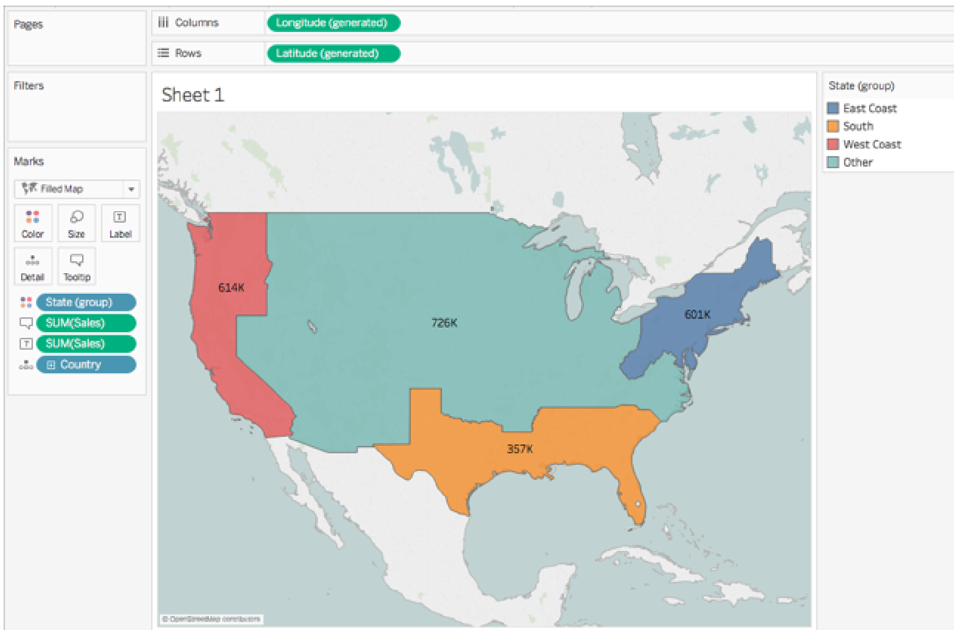
    When you hover over a state, a tooltip appears with the sales for that state, among other information. You'll learn how to edit this tooltip later.



16. On the Marks card, click the minus (-) icon on the **Country** field to remove State from the level of detail.

If you did not create a hierarchy, you can drag **State** from the view to remove it. You can remove any field by dragging it from the view.

The states no longer appear on the map. Notice how the sum of sales has updated for the labels and in the tooltip? This is because custom territories aggregate at the level of the group, rather than separately for each location within the group. So the sum of sales your are seeing in the West Coast group, for example, are the total sales for California, Oregon, and Washington combined.



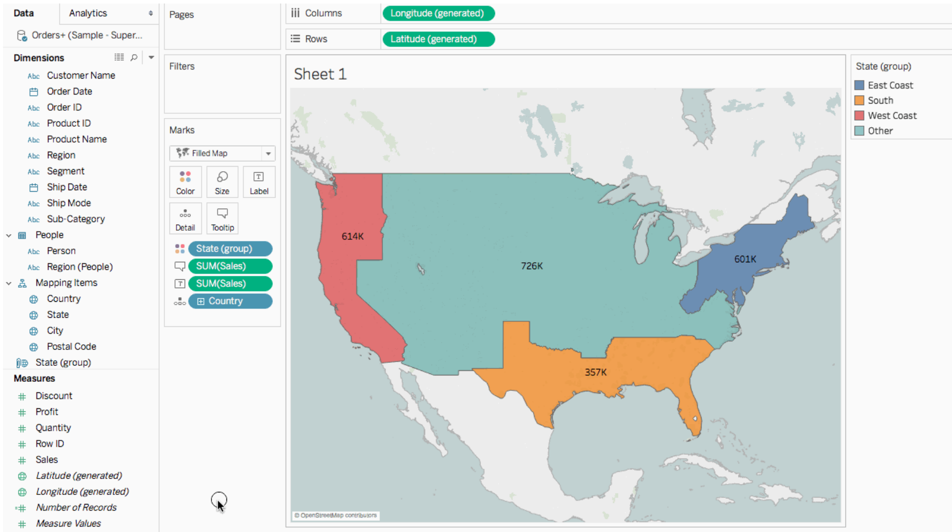
Step 10: Create a dual axis map

So far you have created two map views: one that shows the sales per state, and one that shows the sales per region. Could you layer these maps on top of one another? Yes! In Tableau, you can create a map with two layers of marks. This is called a dual axis map in Tableau, and is often used to layer points over polygons. In this example, you will layer two polygons maps.

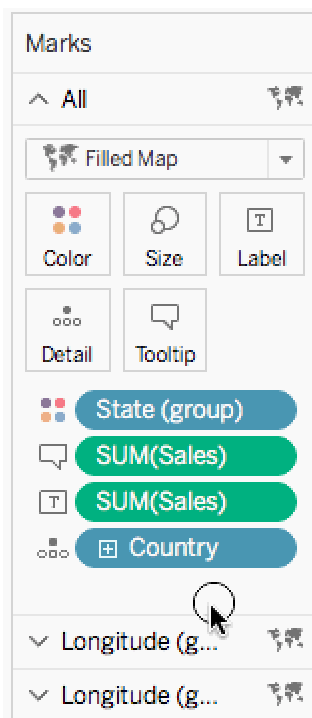
To create a dual axis map:

1. From the Data pane, drag **Longitude (generated)** to the **Columns** shelf and place it to the right of the first Longitude field.

   The view updates with two identical maps.



There are now three tabs on the Marks card: one for each map view, and one for both views (All). You can use these to control the visual detail of the map views. The top Longitude tab corresponds to the map on the left of the view, and the bottom Longitude tab corresponds to the map on the right of the view.



2. On the Marks card, select the top **Longitude (generated)** tab.

3. From Measures, drag **Sales** to **Color**on the top Longitude (generated) Marks card.

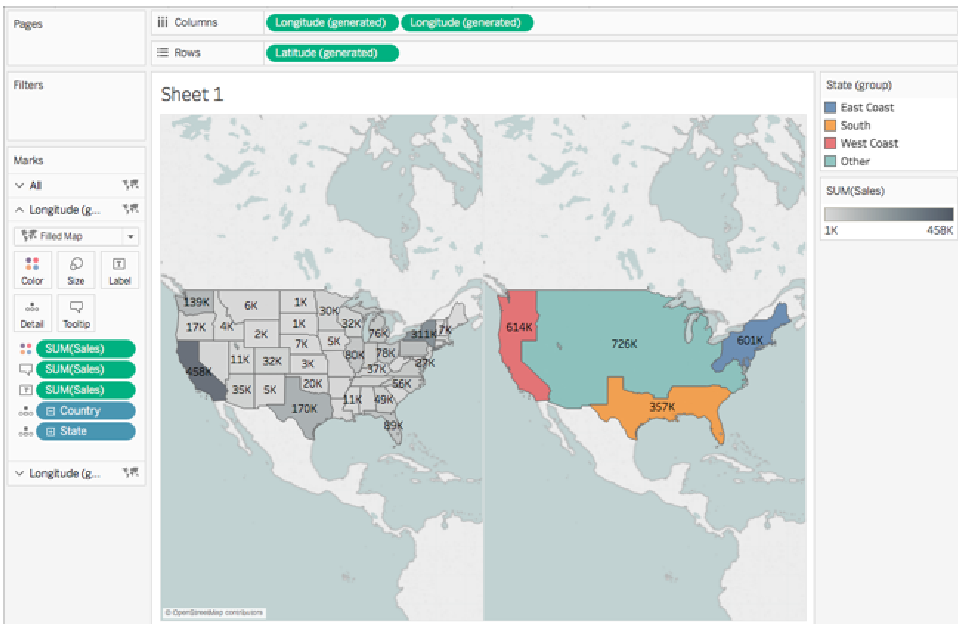
   The map on the left updates.

4. On the top Longitude (generated) Marks card, click the **+** icon on the **Country** field to drill back down to the State level of detail.

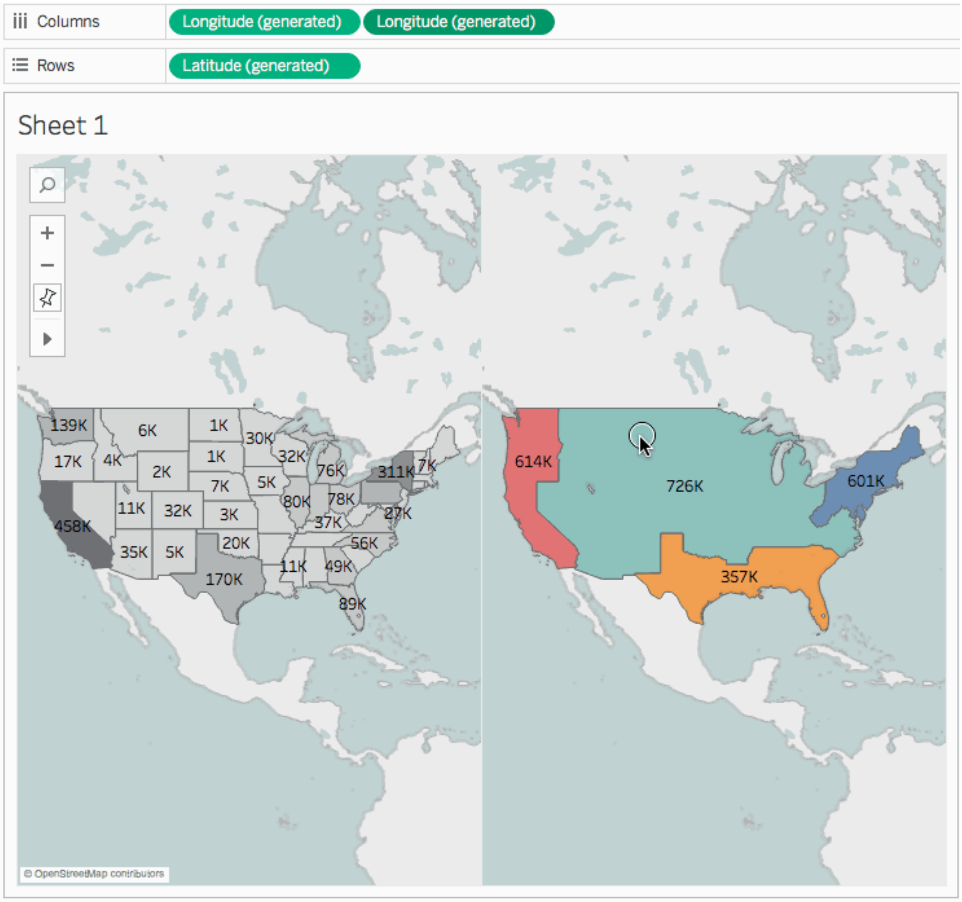
5. On the Marks card, click **Color**, and then select **Edit Colors**.

6. In the Edit Colors dialog box that opens, click the Palette drop-down, select **Gray**, and then click **OK**.

   At this point, your maps look like this:



7. On the Columns shelf, right-click the **Longitude (generated)** field on the right and select **Dual Axis**.



8. On the Marks card, select the bottom **Longitude (generated)** tab.

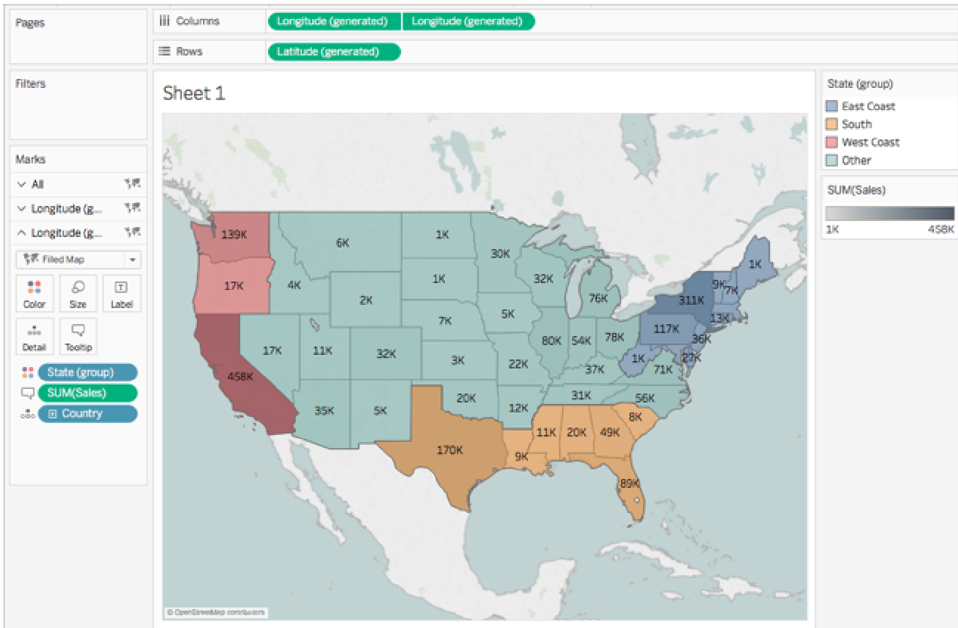
9. On the bottom **Longitude (generated)** Marks card, drag both **SUM(Sales)** fields from the view to remove them.

   The labels for each map no longer overlap.

10. On the bottom **Longitude (generated)** Marks card, click **Color**, and then, for **Opacity**, enter **50%**.

    This is a crucial step if you want to be able to see the map on the bottom layer.

    The map view updates to look like this:



You can now see how each state performed within each group.

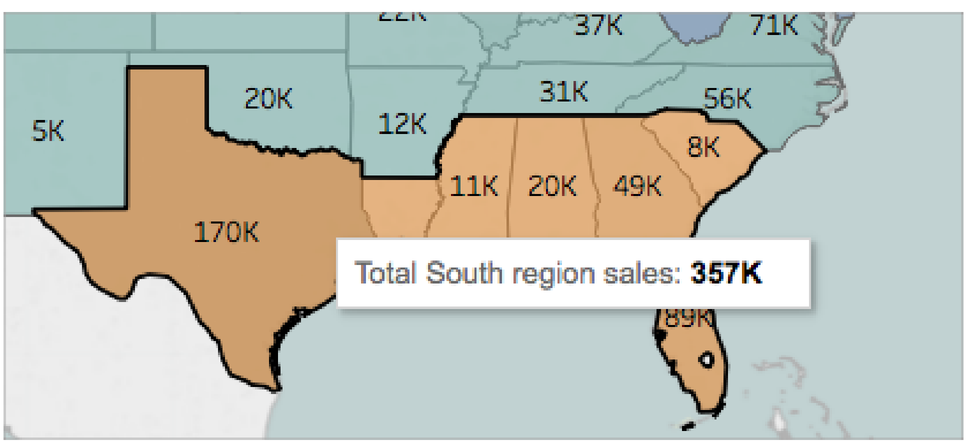
11. On the bottom **Longitude (generated)** Marks card, click **Tooltip**.

    An Edit Tooltip dialog box opens.

12. Copy the following text and paste it into the Edit Tooltip dialog box, and then click **OK**:

    Total <State (group)> region sales: **<SUM(Sales)>**

    The tooltip looks similar to this:



Congrats! You've created a dual axis map! There's only one thing left to do.

For more information about dual axis maps, see Dual Axis Mapping - Many Ways on the Tableau Community.

Step 11: Customize how others can interact with your map

Now that you have created your map view, you can customize how people will interact with it. For example, you might now want anyone to be able to zoom in or out of your map, or pan. Or perhaps you want to display a map scale? You can customize these two options and more in the Map Options dialog box.

To customize how others can interact with your map:

1. Select **Map** > **Map Options**.
2. In the Map Options dialog box that appears, do the following:
   - Select **Show Map Scale**.
   - Clear **Show Map Search**.
   - Clear **Show View Toolbar**.

   A scale appears in the bottom-right corner of the map, and the map search icon and the toolbar in the top left corner of the map disappear. You can still pan and zoom using keyboard shortcuts. For more information, see **Zoom and Pan Views, and Select Marks** on page 2719.

   And that's a wrap! Your map view is now ready to be presented or combined with another view in a dashboard. For more information about dashboards, see **Create a Dashboard** on page 2741.

Want to check your work? Download the example workbook from Tableau Public.

What's Next?

Learn some mapping visualization basics.

Learn about the types of maps you can build in Tableau and how to create them.

Looking for more advanced content? See the mapping series on the Tableau Community.

Explore other Help content about mapping.

Watch online video tutorials.

Got a specific question or problem? Check out the Mapping Forum on the Tableau Community or search the Knowledge Base.

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Getting Started with Mapping (3 minutes) and Maps in Tableau (4 minutes). Use your tableau.com account to sign in.

Mapping Concepts in Tableau

If you want to analyze your data geographically, you can plot your data on a map in Tableau. This topic explains why and when you should put your data on a map visualization. It also describes some of the types of maps you can create in Tableau, with links to topics that demonstrate how to create each one.

If you're new to maps in Tableau, this is a great place to start learning.

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Getting Started with Mapping (3 minutes) and Maps in Tableau (4 minutes). Use your tableau.com account to sign in.

Why put your data on a map?

There are many reasons to put your data on a map. Perhaps you have some location data in your data source? Or maybe you think a map could really make your data pop? Both of those are good enough reasons to create a map visualization, but it’s important to keep in mind that maps, like any other type of visualization, serve a particular purpose: they answer spatial questions.

You make a map in Tableau because you have a spatial question, and you need to use a map to understand the trends or patterns in your data.

But what is a spatial question? Some examples might be:

- Which state has the most farmers markets?
- Where are the regions in the U.S. with the high obesity rates?
- Which metro station is the busiest for each metro line in my city?

- Where did the storms move over time?
- Where are people checking out and returning bikes from their local bike share program?

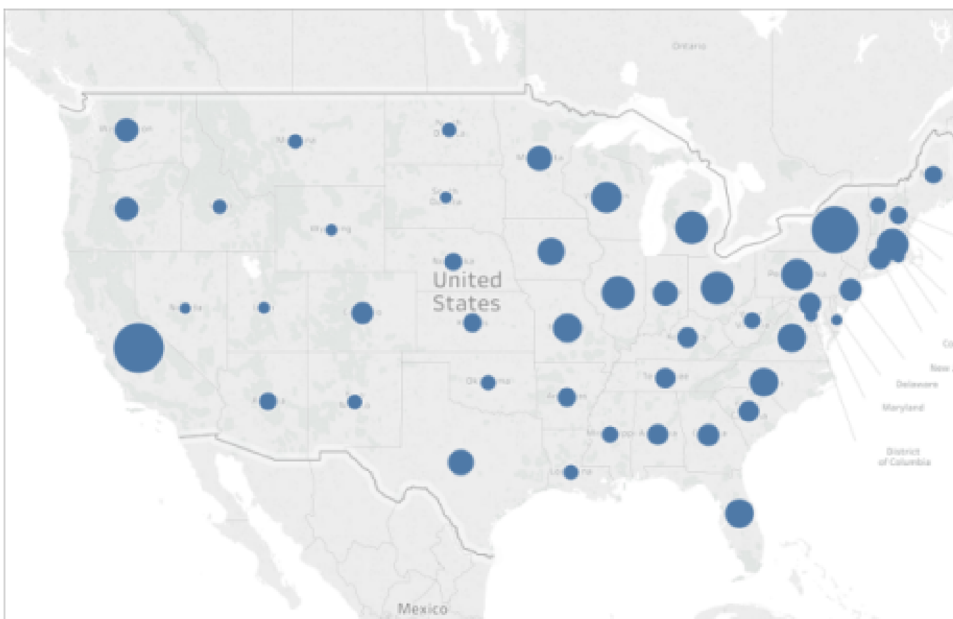
All of these are spatial questions. However, is a map the best way to answer them?

When should you use a map to represent your data?

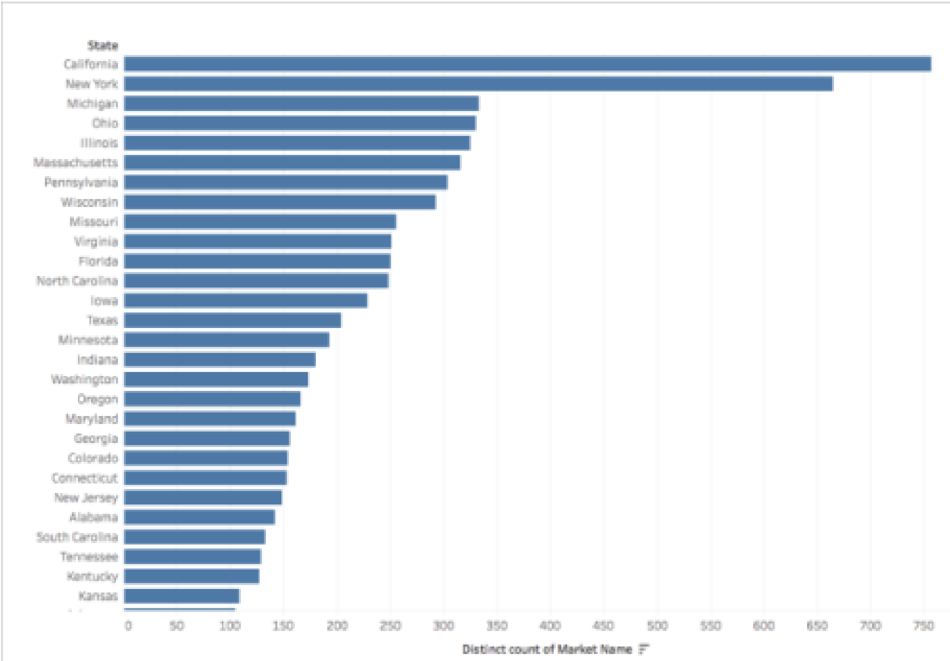
If you have a spatial question, a map view might be a great way to answer it. However, that might not always be the case.

Take for example, the first question from the list above: Which state has the most farmers markets?

If you had a data source with a list of farmers markets per state, you might create a map view like the one below. Can you easily tell the difference between New York and California? Which one has more farmers markets?



What if you create a bar chart instead? Now is it easy to spot the state with the most farmers markets?



The above example is one of many where a different type of visualization would be better to answer a spatial question than a map.

So when do you know if you should use a map view?

One rule of thumb is to ask yourself whether or not you could answer your question faster, or easier with another visualization. If the answer is yes, then perhaps a map view is not the best visualization for the data you're using. If the answer is no, then take the following into account:

Maps that answer questions well have both appropriate data representation, and attractive data representation. In other words: the data is not misleading, and the map is appealing.

If your map is beautiful, but the data is misleading, or not very insightful, you run the risk of people misinterpreting your data. That's why it's important to create maps that represent your data accurately, as well as attractively.

What types of maps can you build in Tableau?

With Tableau, you can create the following common map types:

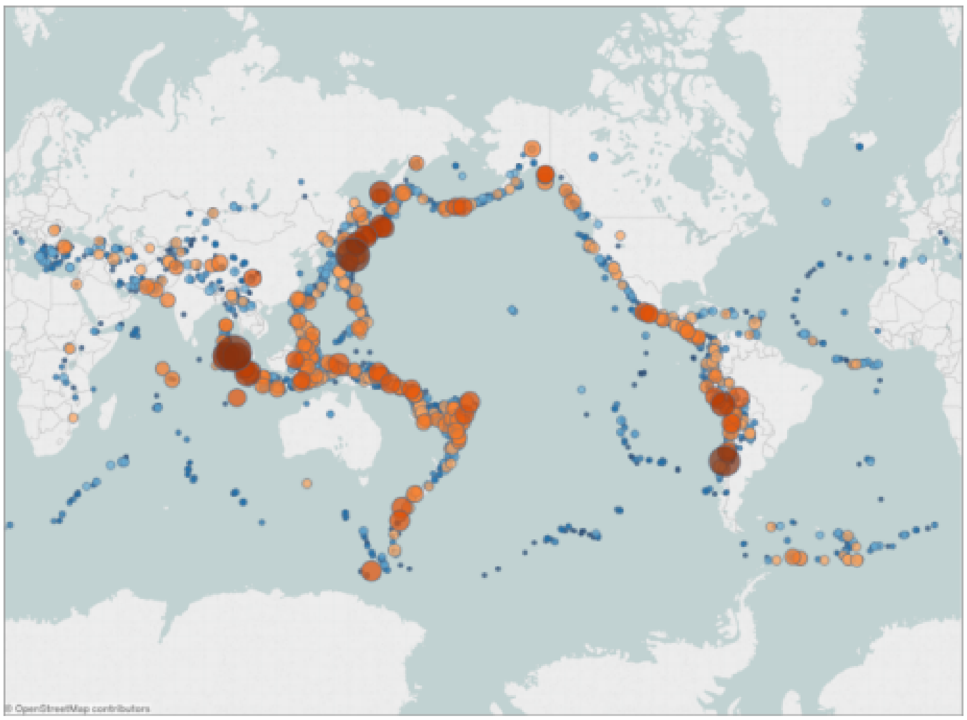
- **Proportional symbol maps** on the next page
- **Choropleth maps (filled maps)** on the next page
- **Point distribution maps** on page 1785
- **Heatmaps (density maps)** on page 1786

- **Flow maps (path maps)** on page 1786
- **Spider maps (origin-destination maps)** on page 1787

Proportional symbol maps

Proportional symbol maps are great for showing quantitative data for individual locations. For example, you can plot earthquakes around the world and size them by magnitude.

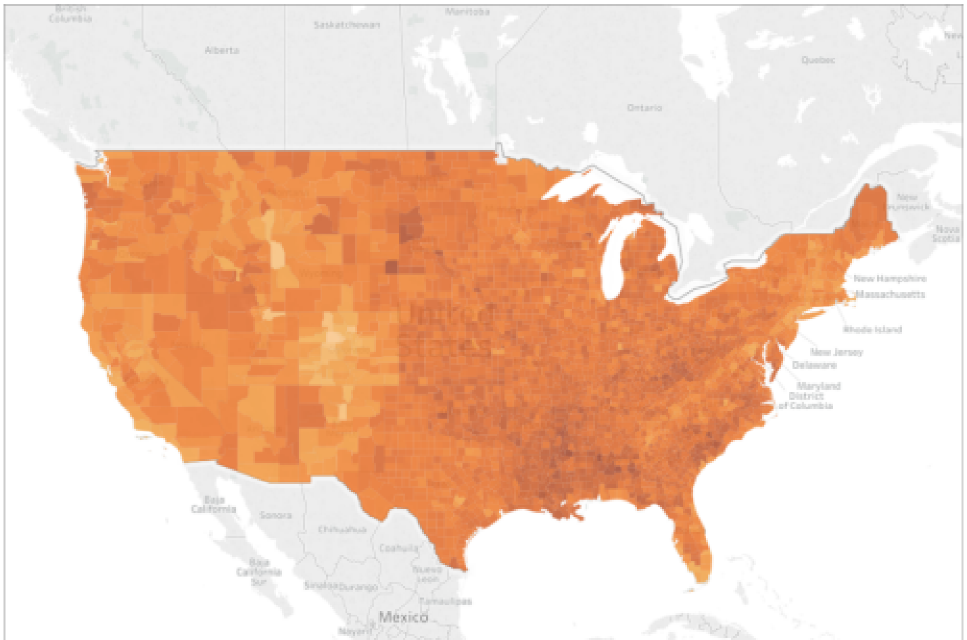
For more information about proportional symbol maps, and to learn how to create them in Tableau, see **Create Maps that Show Quantitative Values in Tableau** on page 1883.



Choropleth maps (filled maps)

Also known as filled maps in Tableau, Choropleth maps are great for showing ratio data. For example, if you want to see obesity rates for every county across the United States, you might consider creating a choropleth map to see if you can spot any spatial trends.

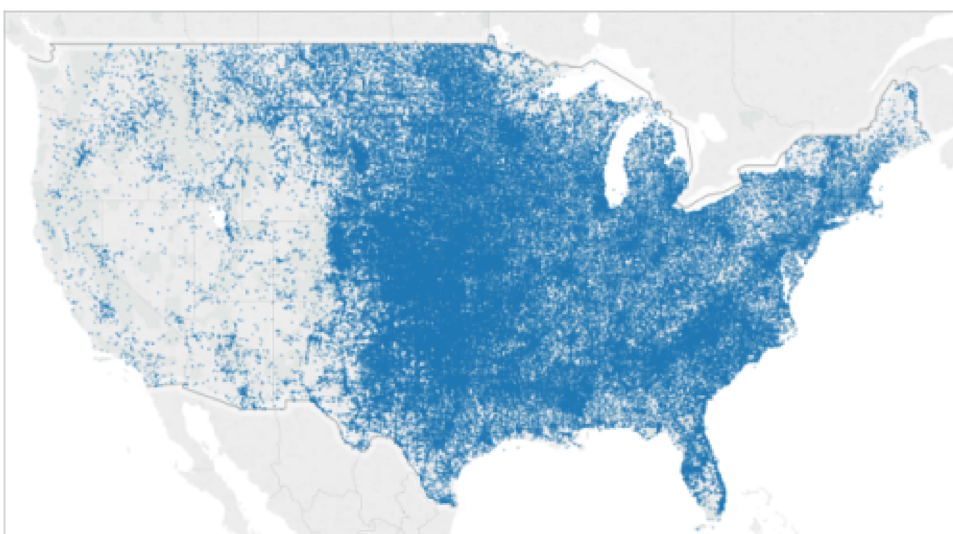
For more information about Choropleth maps, and to learn how to create them in Tableau, see **Create Maps that Show Ratio or Aggregated Data in Tableau** on page 1895.



Point distribution maps

Point distribution maps can be used when you want to show approximate locations and are looking for visual clusters of data. For example, if you want to see where all the hailstorms were in the U.S. last year, you can create a point distribution map to see if you can spot any clusters.

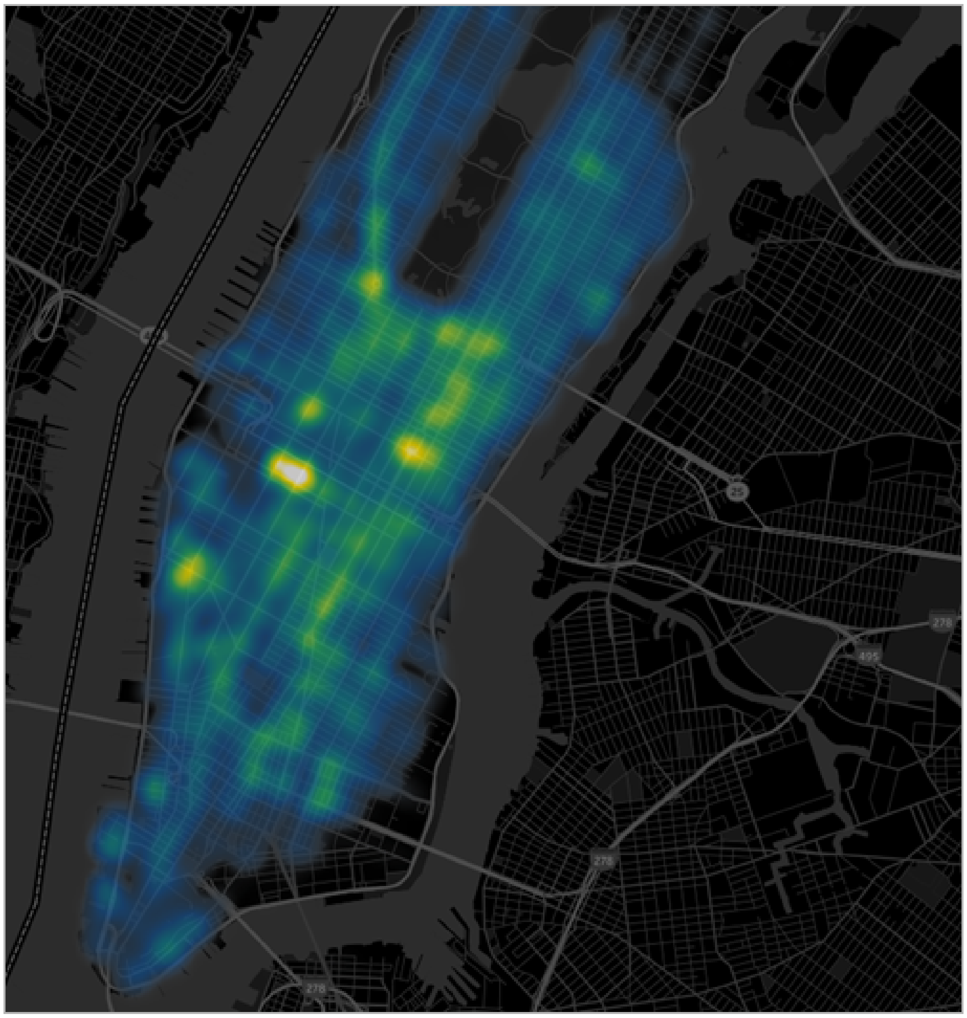
For more information about point distribution maps, and to learn how to create them in Tableau, see **Create Maps that Highlight Visual Clusters of Data in Tableau** on page 1889.



Heatmaps (density maps)

Heatmaps, or density maps, can be used when you want to show a trend for visual clusters of data. For example, if you want to find out which areas of Manhattan have the most taxi pickups, you can create a density map to see which areas are most popular.

For more information about density maps, and to learn how to create them in Tableau, see **Create Heatmaps that Show Trends or Density in Tableau** on page 1910.

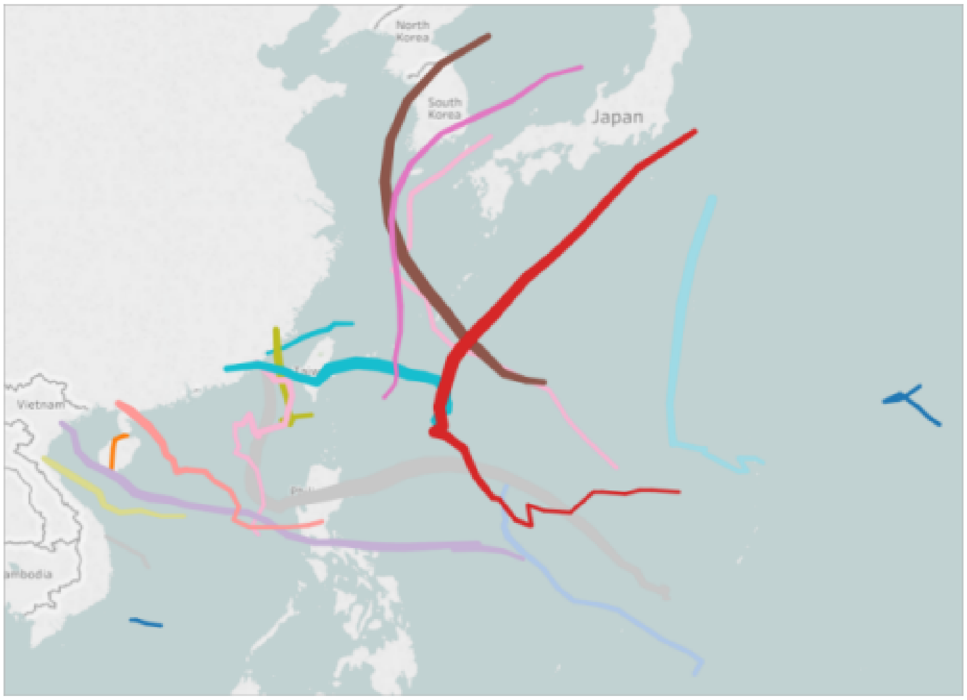


Back to top

Flow maps (path maps)

You can use flow maps to connect paths across a map and to see where something went over time. For example, you can track the paths of major storms across the world over a period of time.

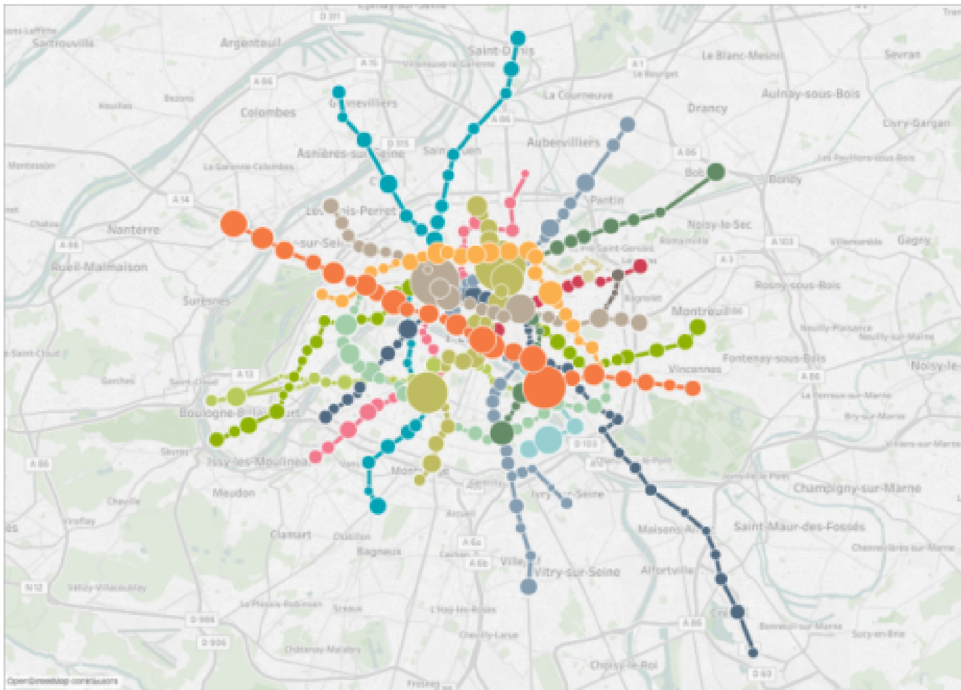
For more information about flow maps, and to learn how to create them in Tableau, see **Create Maps that Show a Path Over Time in Tableau** on page 1903.

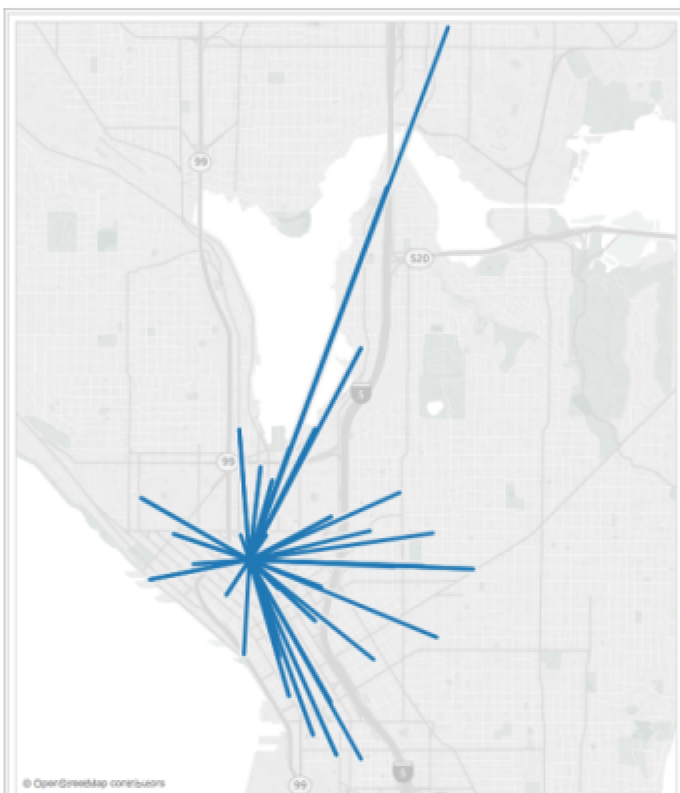


Spider maps (origin-destination maps)

You can use a spider map to show how an origin location and one or more destination locations interact. For example, you can connect paths between metro stations to plot them on a map, or you can track bike share rides from an origin to one or more destinations.

For more information about spider maps, and to learn how to create them in Tableau, see **Create Maps that Show Paths Between Origins and Destinations in Tableau** on page 1917.


Sartrouville
Argenteuil
Saint-Denis
Le Blanc-Mesnil
Sevran
Colombes
La Courneuve
Aulnay-sous-Bois
Drancy
Aubervilliers
Pantin
Bondy
Nanterre
Noisy-le-Sec
Gagny
Rueil-Malmaison
Rosny-sous-Bois
Suresnes
Montreuil
Fontenay-sous-Bois
Vincennes
Meudon
Champigny-sur-Marne
Clamart
Saint-Maur-des-Fossés
Bagneux
Vitry-sur-Seine
Alfortville
Choisy-le-Roi



Resources to help you get started

Before you get started with building map views in Tableau, review some of the following resources.

Get your geographic data into Tableau

- **Create Tableau Maps from Spatial Files** on page 1855: If you have Esri Shapefiles, Mapinfo tables, or KML files, start here.
- **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map** on page 1818: If you have data that Tableau does not recognize, start here.
- **Blend Geographic Data** on page 1827: If you have geographic data you want to blend with another data source, start here.

Format your geographic fields

- **Assign Geographic Roles** on page 1834: Once you bring your geographic data into Tableau, you'll need to format that data for use in Tableau.

Watch a Video: To see related concepts demonstrated in Tableau, watch these free training videos: Getting Started with Mapping (3 minutes) and Maps in Tableau (4 minutes). Use your tableau.com account to sign in.

See also

Build-It-Yourself: Build a map view

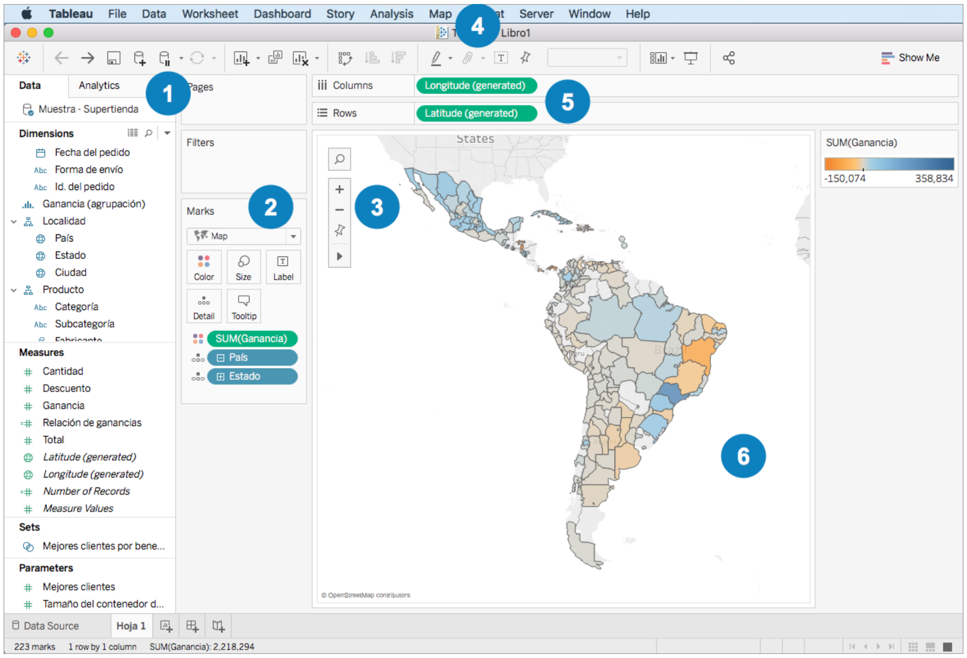
Customize How Your Map Looks on page 1965

Use Mapbox Maps on page 1993

Use Web Map Service (WMS) Servers on page 1996

The Mapping Workspace in Tableau

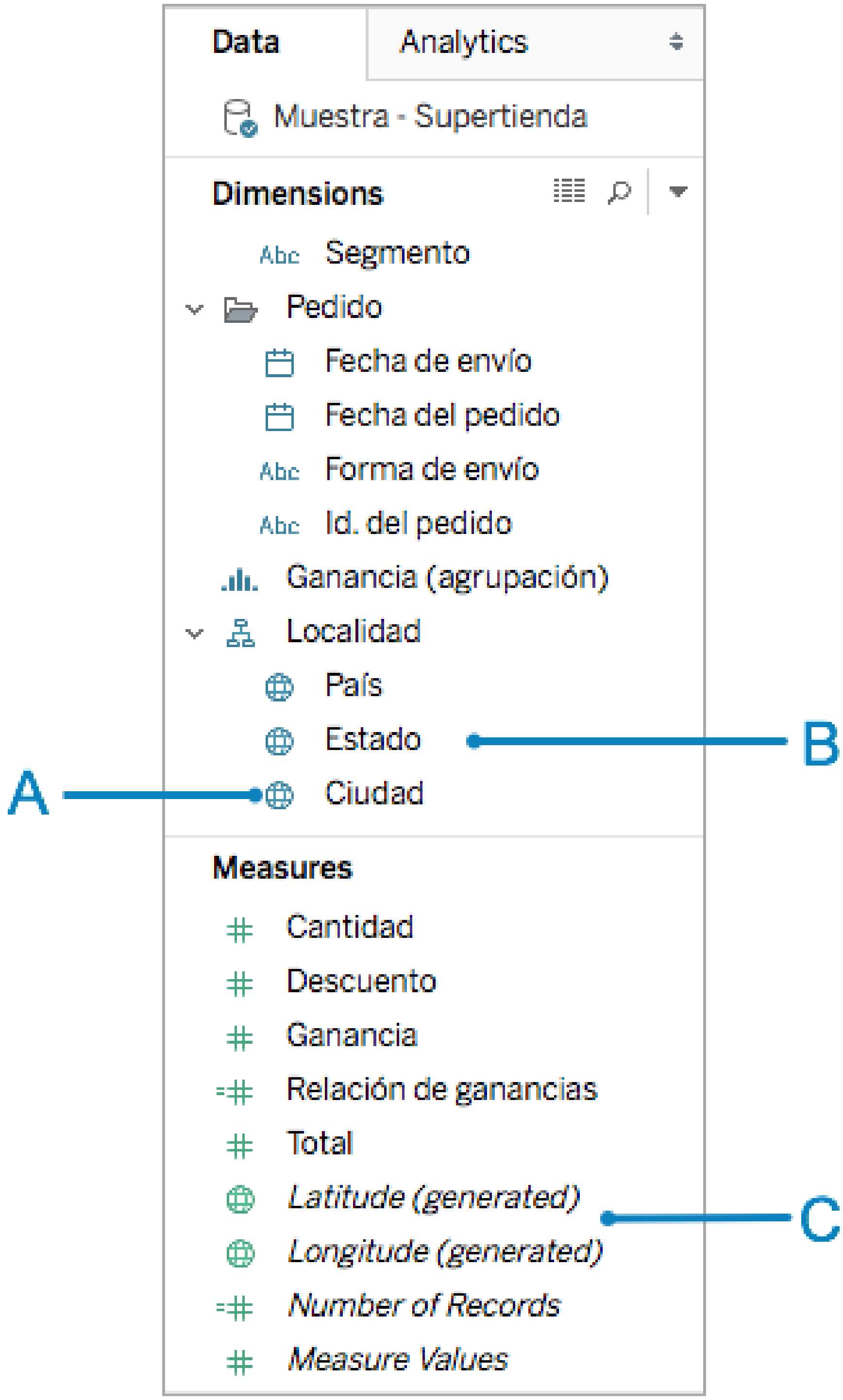
The mapping workspace in Tableau consists of the following elements.



The Data Pane

The Data pane organizes your location data into dimensions and measures. Typically, your location fields (country, state, and city names, for example) should be dimensions, and your latitude and longitude fields should be measures. However, there are some scenarios where latitude and longitude fields can be dimensions.

In the Data pane, a globe icon next to a location field means that a geographic role is assigned to that field. For more information, see **Assign Geographic Roles** on page 1834.



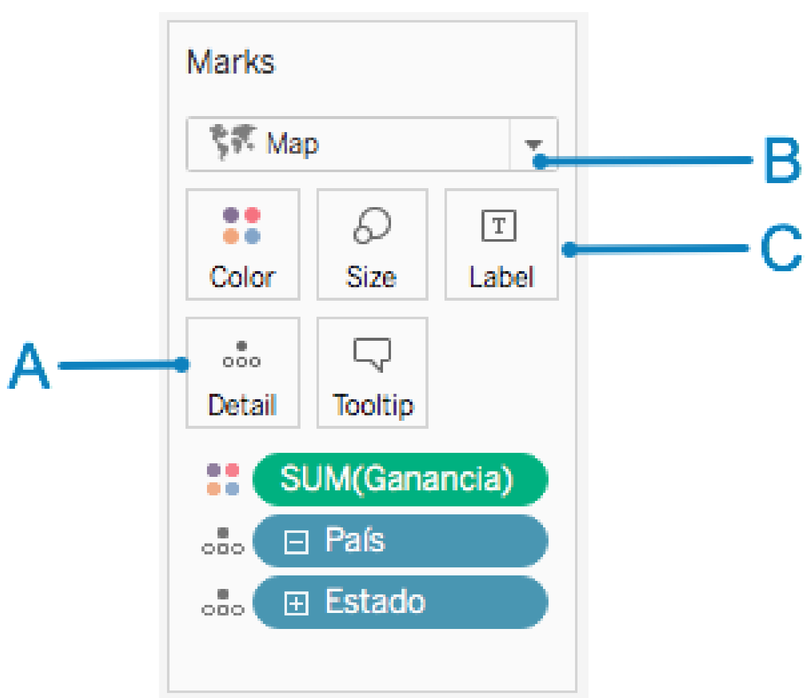
A: Globe icon - indicates that a field is a geographic field and can be automatically mapped by Tableau.

B: Location fields - These categorical fields, often location names, can be found under Dimensions. They are often recognized by Tableau.

C: Latitude and Longitude generated fields - When you have geographic fields in your data source, Tableau often generates Latitude and Longitude fields. You can use these to build your map, or you can include your own latitude and longitude fields.

2 The Marks Card

The Marks card is where you control the granularity and visual detail of your map view. In the Marks card, you can drag locations to Detail to add more granularity, drag fields to Color, Size, or Label to add visual detail, or change the map type from a point map to a polygon map.



A: Add location fields to Detail to create a map and add granularity to the view.

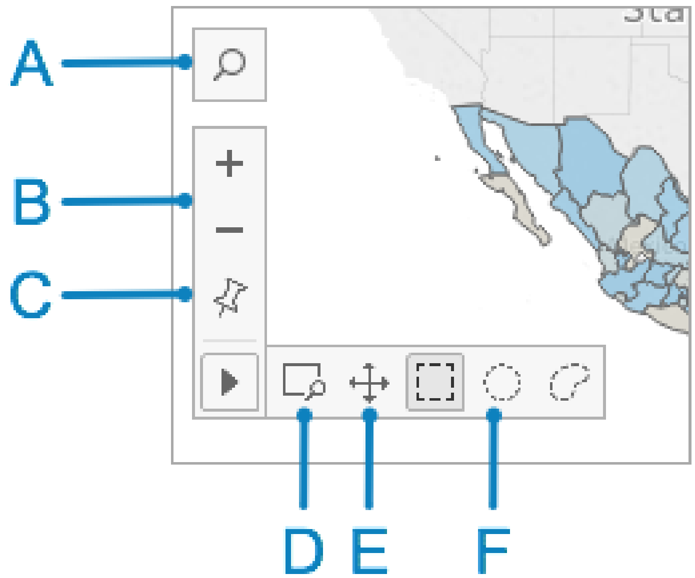
B: Select the Automatic mark type for a point map and the Map mark type for a polygon (filled) map.

C: Add fields to Color, Size, and Label to change the visual detail of the marks (data points) in the view.

3 The View Toolbar

The view toolbar holds many of the tools you might need to explore your map data.

Note: Many of the tools on this toolbar have keyboard shortcuts. For more information, see **Keyboard Shortcuts** on page 3344.



A: Use the map search icon to search for locations in your map.

B: Use the plus and minus icons to zoom in and out of the map.

C: Use the pin icon to fix the map at a specific zoom level.

D: Use the zoom area icon to zoom in to a specific area in the map.

E: Use the pan icon to pan around the map.

F: Use the selection tools to select data points within a rectangle, circle, or drawn area. You can also measure distances with the circle selection tool. For more information, see **Measure Distances Between Data Points and Locations in a Map** on page 2007.

4 The Map Menu

The Map menu holds several maps-specific options:

- Select **Map** > **Background Maps** to add a map service (such as Mapbox or a WMS server) or to switch between the background maps you have available in your workbook. For more information, see **Import your own background map** on page 1967.
- Select **Map** > **Background Images** to add a static image instead of a background map to your view.
- Select **Map** > **Geocoding** to import locations that Tableau does not recognize into your workbook. For more information, see **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map** on page 1818.
- Select **Map** > **Edit Locations** to edit your location data to match the Tableau map data. For more information, see **Edit Unknown or Ambiguous Locations** on page 1839.
- Select **Map** > **Map Layers** to customize your background map style, add or remove map layers, or add U.S. data layers. For more information see **Customize How Your Map Looks** on page 1965.
- Select **Map** > **Map Options** to control if people can pan and zoom, search for locations, or use the view toolbar. For more information, see **Customize How People Interact with your Map** on page 1989

5 The Columns and Rows Shelves

The Columns and Rows shelves are where you put your latitude and longitude fields. Longitude is placed on the Columns shelf, and Latitude is placed on the Rows shelf, regardless of whether they are your latitude and longitude fields, or the generated fields Tableau creates.

6 The Map Visualization

This is your map. As you makes changes to your map, the viz updates. This visualization is dynamic and can be interacted with. Hover over the marks to learn more information about each location. You can edit the information that displays in the tooltip that appears using the Tooltip button on the Marks card. For more information, see **Add tooltips to marks** on page 1366 and **Create Views in Tooltips (Viz in Tooltip)** on page 1475.

In addition to the options available in the mapping workspace, Tableau provides many other options for analyzing your data. You can create custom territories, filter data from your view, set up parameters, create sets, and much more.

See Also

Get Started Mapping with Tableau on page 1755

Location Data that Tableau Supports for Building Map Views

Tableau supports worldwide airport codes, cities, countries, regions, territories, states, provinces, and some postcodes and second-level administrative districts (county-equivalents). U.S. area codes, Core Based Statistical Areas (CBSA), Metropolitan Statistical Areas (MSA), Congressional districts, and Zip codes are also supported by Tableau. Additionally, any latitude and longitude coordinates are supported, as long as they are in decimal degrees.

> **Note:** See About Tableau Maps on the Tableau website for a complete list of location data by country that Tableau supports.

What to do if Tableau doesn't recognize your location data

- **If Tableau recognizes your location data and automatically assigns geographic roles to your fields** (you'll know this has happened because there will be a globe icon next to them in the **Data** pane), then you're ready to build a map view. Just double-click one of those geographic fields and you've got a map.
- **If Tableau does not immediately recognize your location data**, and you can't build a map view, you'll need to assign geographic roles to your fields. See the **Assign a geographic role to a field** on page 1835 section to learn how.

  Note: This procedure only works if your location data is supported by Tableau. If your data isn't supported by Tableau, there are a couple of things you can try in the meantime to map that data in Tableau:

  - Edit location names in your data source to match Tableau location names
  - Custom geocode your data
  - Blend your geographic data
  - Use spatial data to build a map view

  > **Note**:Connecting to spatial data is only supported on Tableau Desktop version 10.2 and later.

See Also

About Tableau Maps Data

About Tableau Maps Partners

Spatial File

This topic describes how to connect Tableau to Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, TopoJSON files, GeoJSON files, and Esri File Geodatabases.

> **Note:** Connecting to spatial data is supported in Tableau Desktop version 10.2 and later.

Before you connect

Before you can connect to spatial files, make sure to include all of the following files in the same directory:

- **For Esri shapefiles**: The folder must contain `.shp`, `.shx`, `.dbf`, and `.prj` files as well as `.zip` files of the Esri shapefile.

  For Esri File Geodatabases: The folder must contain the File Geodatabase's `.gdb` or the `.zip` of the File Geodatabases's `.gdb.`
- **For MapInfo tables** (Tableau Desktop only): The folder must contain `.TAB`, `.DAT`, `.MAP`, and `.ID` or `.MID` and `.MIF` files.
- **For KML files**: The folder must contain the `.kml` file. (No other files are required.)
- **For GeoJSON files**: The folder must contain the `.geojson` file.(No other files are required.)
- **For TopoJSON files**: The folder must contain the `.json` or `.topojson` file. (No other files are required.)

> **Note**: Flat files will pull in the contents of the entire folder. For performance reasons, remove unneeded files and reduce the amount of data in the files.
>
> You can connect only to point geometries, linear geometries, and polygons in current versions of Tableau. You cannot connect to mixed geometry types.
>
> If your data does not display diacritics (accent marks on characters) properly, check to make sure the file is **UTF-8** encoded.

Make the connection and set up the data source

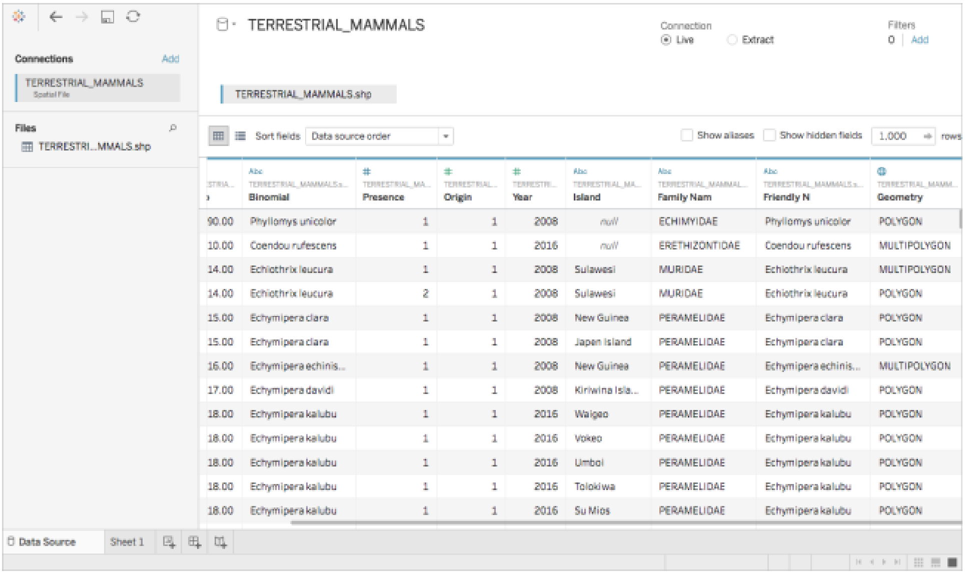
In Tableau Desktop: click the New Data Source icon and select **Spatial file**.
In Tableau Online or Tableau Server (Creator role): Select **Create** > **Workbook**. Select the **Files** tab.

Then do the following:

1. Navigate to the folder that contains your spatial data and select the spatial file you want to connect to.
2. Select **Open**.

Spatial file data source example

Here is an example of a spatial file data source using Tableau Desktop on a Mac computer:



Work with the Geometry column

There are many tasks that you can optionally perform on your data before you start your analysis, such as hiding or renaming fields. Note, however, the following restrictions apply when working with the Geometry column:

- You can't hide the Geometry column.
- You can't split the Geometry column.
- On the data source page, you can't create a calculated field using the Geometry column.

About .ttde and .hhyper files in Tableau Desktop

You might notice .ttde or .hhyper files when navigating your computer's directory. When you create a Tableau data source that connects to your data, Tableau creates a .ttde or .hhyper file. This file, also known as a shadow extract, is used to help improve the speed your data source loads in Tableau Desktop. Although a shadow extract contains underlying data and other information similar to the standard Tableau extract, a shadow extract is saved in a different format and can't be used to recover your data.

In certain situations, you might need to delete a shadow extract from your computer. For more information, see Low Disk Space because of TTDE Files in the Tableau Knowledge Base.

See also

Create Tableau Maps from Spatial Files on page 1855

Connect to Spatial Data in a Database

You can use spatial columns stored in your database to build map visualizations in Tableau.

Tableau supports direct connections to spatial data using these connectors:

- Microsoft SQL Server
- PostgreSQL + PostGIS
- Pivotal Greenplum + PostGIS
- Oracle

For information on using these connectors, see the **Microsoft SQL Server** on page 563, **PostgreSQL** on page 595, **Pivotal Greenplum Database** on page 593, or **Oracle** on page 574 connector examples.

Supported spatial data and SRIDs

For **Microsoft SQL Server**, only Geography-type spatial data is supported.

PostgreSQL + PostGIS, **Oracle**, and **Pivotal Greenplum + PostGIS** support Geography- and Geometry- type fields. Geometry fields must specify the geo-type (e.g., POINT, LINESTRING) and SRID in order to be recognized as a spatial field. Geography fields must specify the geo-type in order to be recognized.

> **Note**: In PostgreSQL + PostGIS, you can connect to tables with both mixed geometry types (points, lines, and polygons), but Tableau will not render these simultaneously.

The following **SRIDs** are supported by these connectors:

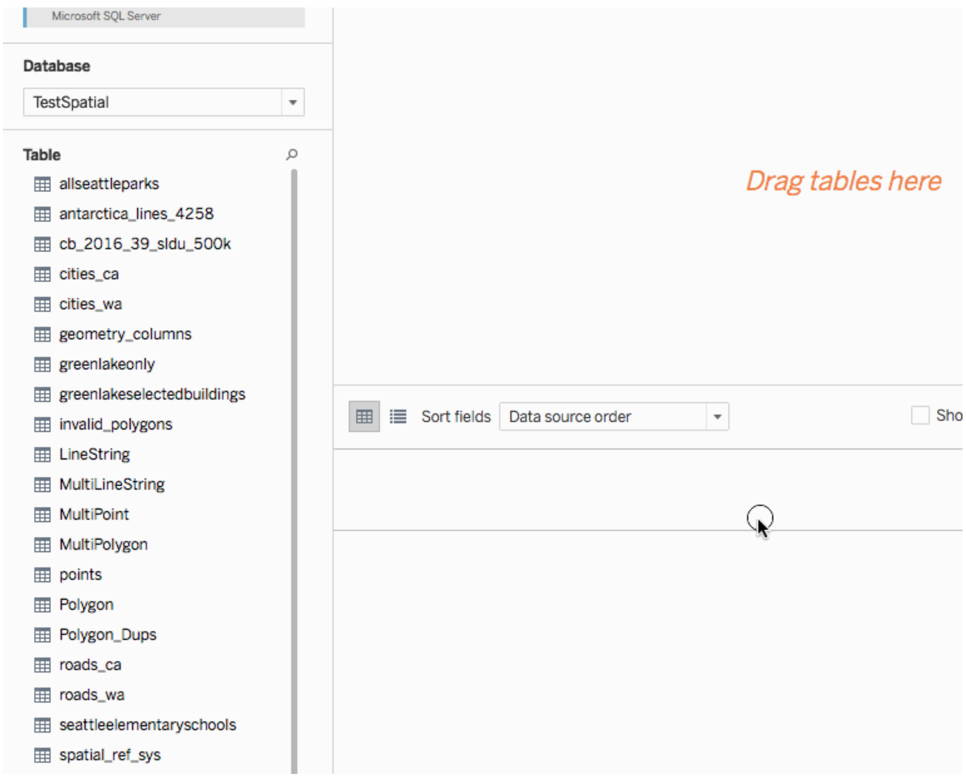
- NAD83 (EPSG:4269)
- ETRS89 (EPSG:4258)
- WGS84 (EPSG:4326)

In PostgreSQL + PostGIS, SRID always defaults to WGS84 (EPSG:4326).

Tableau supports both live and extract connections using these connectors.

Connect to spatial columns

1. Open Tableau and connect to your data. For more information on how to connect to data using supported connectors, see the **Microsoft SQL Server** on page 563, **PostgreSQL** on page 595, **Oracle** on page 574, or **Pivotal Greenplum Database** on page 593 connector examples.
2. On the Data Source page, in the left pane under Table, drag a table that contains spatial data onto the canvas.



Spatial columns will display the same name they have in your database.

You are now ready to build a map in Tableau using spatial data. For more information on how to create maps from spatial files, see **Build a map view from spatial data** on page 1857.

Use Custom SQL and RAWSQL to perform advanced spatial analysis

Connect to a Custom SQL query

1. Open Tableau and connect to your database.
2. On the Data Source page, in the left pane, drag **New Custom SQL** onto the canvas.
3. Type or paste your query into the Edit Custom SQL dialog box that appears. For examples of Custom SQL you can use with spatial data, see **Custom SQL query examples** below.
4. When finished, click **OK**.

For more information, see **Connect to a Custom SQL Query** on page 959

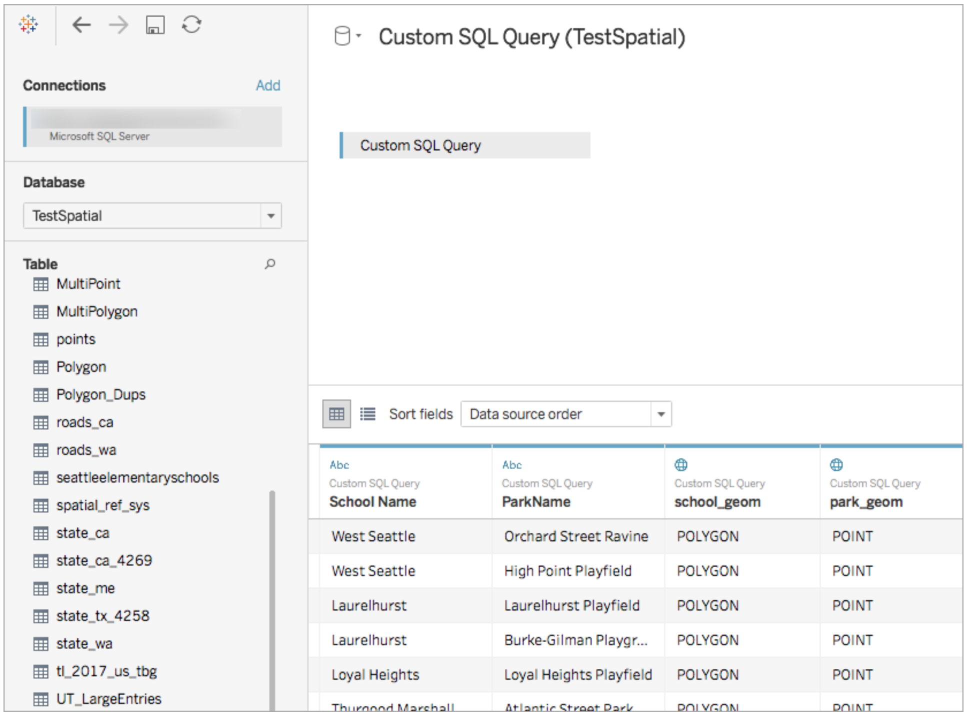
Custom SQL query examples

The following queries are examples. Note that Custom SQL queries will vary from data source to data source.

If your data has two sets of points, such as schools and parks, and you want to show where they intersect:

```
SELECT S.[common nam] as SchoolName, P.[common nam] as
ParkName, S.geom.STBuffer(<Parameters.Radius>) as school_
geom, P.geom as park_geom FROM
TestSpatial.dbo.seattleelementaryschools S LEFT JOIN
TestSpatial.dbo.allseattleparks P on S.geom.STBuffer
(<Parameters.Radius>).STIntersects(P.geom) = 1
```

This example query will produce a row for each case where a park is within Radius distance (meters) from a school. The result of the query looks like this in the data grid:



Note: This example references a parameter named "Radius". It measures distance (meters) and has the following settings:

- Data type: Float
- Current Value: 600
- Allowable values: Range
- Minimum: 100
- Maximum 2,000
- Step size: 50

For more information, see **Use parameters in a custom SQL query** on page 968.

If you only want data around a certain point

```
SELECT * FROM [Test_spatial_<username>].[dbo].[SDOT_
Collisions] C WHERE C.geom.STIntersects
(geography::STGeomFromText('POINT(-122.344706 47.650388)',
4326).STBuffer(1000))=1
```

This example query limits data to 1000 meters around the POINT. The radius can be a parameter, like the parameter used in the first example.

Or, rather than a point and radius, a rectangle can be used. For example:

```
SELECT * FROM [Test_spatial_<username>].[dbo].[SDOT_
Collisions] C
WHERE C.geom.STIntersects(geography::STGeomFromText('POLYGON
((-122.3625 47.6642,-122.3625 47.6493,-122.3427 47.6493,-
122.3422 47.6642,-122.3625 47.6642))', 4326))=1
```

Note: the winding order of the polygon points (clockwise vs. counter-clockwise) determines which set of points you get. You can reverse the winding order with the SQL Server function `ReorientObject()`.

If you get an error about mixed data types

```
SELECT *, F.geom.STGeometryType() as geomtype FROM [Test_
spatial_<username>].[dbo].[us_historic_fire_perimeters_dd83]
F
WHERE F.geom.STGeometryType() = 'MultiPolygon' OR
F.geom.STGeometryType() = 'Polygon'
```

This example query filters out the geometry-type spatial data, since Tableau can not render geometry-type spatial data for Microsoft SQL Server connections. You can select other spatial types as well.

If your data renders too slowly in Tableau

```
SELECT [Id], [OBJECTID], [fire_num], [year_], [acres], [fire_
name], [unit_id], [Shape_Leng], [Shape_Area], [geom].Reduce
(500) as SimpleGeom
FROM [Test_spatial_<username>].[dbo].[us_historic_fire_
perimeters_dd83]
WHERE [geom].Reduce(500).STGeometryType() = 'MultiPolygon' OR
[geom].Reduce(500).STGeometryType() = 'Polygon'
```

This example query may greatly reduce data size. (Some objects are reduced to Points, and these are discarded.)

Use RAWSQL

To use RAWSQL with spatial data, you can create a calculated field using either the RAWSQL_SPATIAL or RAWSQLAGG_SPATIAL functions. For example:

```
RAWSQL_SPATIAL("Select %1.STIntersection(%2.STBuffer(200))",
[school_geom], [park_geom])
```

This formula returns unique spatial data based on the intersection of two values.

```
RAWSQLAGG_SPATIAL("ST_ConvexHull(ST_Collect(%1))", [Geog])
```

```
RAWSQLAGG_SPATIAL("ST_ConcaveHull(ST_Collect(%1), %2,
false)", [Geog], [ParameterValue])
```

Each of these will return a bounding area around the supplied data points. ConcaveHull is a minimum bounding area based on the precision, which in this function is the parameter value.

For more information about using RAWSQL with spatial data, see **Pass-Through Functions (RAWSQL)** on page 2202. For more information about how to create calculated fields, see **Create a calculated field** on page 2097.

Note: RAWSQL spatial functions are not available when connected to **Oracle**.

> **Note**: It's possible to cause errors when using RAWSQL specific to PostGIS functions when using PostgreSQL + PostGIS.

Troubleshooting spatial connections

Performance issues

When working with joined spatial columns with a large number of records, duplicate marks can occur. This can result in slow performance.

To improve performance, extract the joined data source. For more information, see **Extract Your Data** on page 1014.

Microsoft SQL Server error messages and solutions

When the spatial table is using an unsupported spatial reference

Error message:

An error occurred while communicating with the Microsoft SQL Server data source '<data source name>'.
Bad Connection: Tableau could not connect to the data source.
[Microsoft][ODBC Driver 13 for SQL Server][SQL Server]Operand type clash: geometry is incompatible with geography
[Microsoft][ODBC Driver 13 for SQL Server][SQL Server]Statement(s) could not be prepared.

Solution:

- Export the spatial data from Microsoft SQL Server to a Shapefile and connect to it in Tableau. Spatial file connections go through a data transformation process that supports thousands of projections.
- Export the data from Microsoft SQL Server and transform the data using a GIS tool like QGIS or ArcGIS. Then reload the table back into Microsoft SQL Server.

When the spatial table includes multiple types of spatial objects

Error message:

Unable to complete action
The spatial operation resulted in a MixedGeometry or MixedGeography, which Tableau does not support yet.
The spatial operation resulted in a MixedGeometry or MixedGeography, which Tableau does not support yet.

Solution:

Use Custom SQL to filter the data. For more information, see **If you get an error about mixed data types** on page 1803 Custom SQL example.

When measures are being aggregated in the visualization

Error message:

An error occurred while communicating with the Microsoft SQL Server data source '<data source name>'
Bad Connection: Tableau could not connect to the data source.
Cannot perform an aggregate function on an expression containing an aggregate or a subquery.

Solution:

Disaggregate measures in the visualization: Select **Analysis**, and then clear **Aggregate Measures**.

PostgreSQL + PostGIS, Oracle, and Pivotal Greenplum + PostGIS error messages and solutions

When using mixed SRIDs and one of the SRIDs is unsupported

Error message:

An error occurred while communicating with the PostgreSQL data source '<data source name>'.
Bad Connection: Tableau could not connect to the data source.
Error: Operation on mixed SRID geometries;
Error while executing the query
<query> from <datasource>
HAVING (COUNT(1) > 0)

Solution:

Disaggregate measures in the visualization: Select **Analysis**, and then clear **Aggregate Measures**. This will eliminate the error message and the data where the SRID is unsupported.

When the spatial table is using an unsupported spatial

reference

In PostgreSQL + PostGIS, you can connect to a table with unsupported SRIDs, but the Geometry column will display as "Unknown."

Solution:

- Export the spatial data from PostgreSQL + PostGIS to a Shapefile and connect to it in Tableau. Spatial file connections go through a data transformation process that supports thousands of projections.
- Export the data from PostgreSQL + PostGIS and transform the data using a GIS tool like QGIS or ArcGIS. Then reload the table back into PostgreSQL + PostGIS.

When a query result exceeds the limit of 1GB

Error message:

ERROR: array size exceeds the maximum allowed (1073741823); Error while executing the query"
SELECT ST_Collect(geom) FROM <data source>

Solution:

Rewrite the calculation to be more efficient. See **Best Practices for Creating Calculations in Tableau** on page 2128 or **Create Efficient Calculations** on page 3003.

Example - Build a map from Microsoft SQL Server spatial data

The following example demonstrates how to create the map below, which shows schools within 600 meters of parks in Seattle, Washington.

This example uses a Microsoft SQL Server connection to a database called, TestSpatial. The following tables from this database are used:

- seattleelementaryschools
- allseattleparks

Step 1: Connect

1. Open Tableau Desktop and connect to Microsoft SQL Server.
2. On the Data Source page, in the left pane, drag **New Custom SQL** onto the canvas.
3. Type or paste a query into the Edit Custom SQL dialog box that appears.

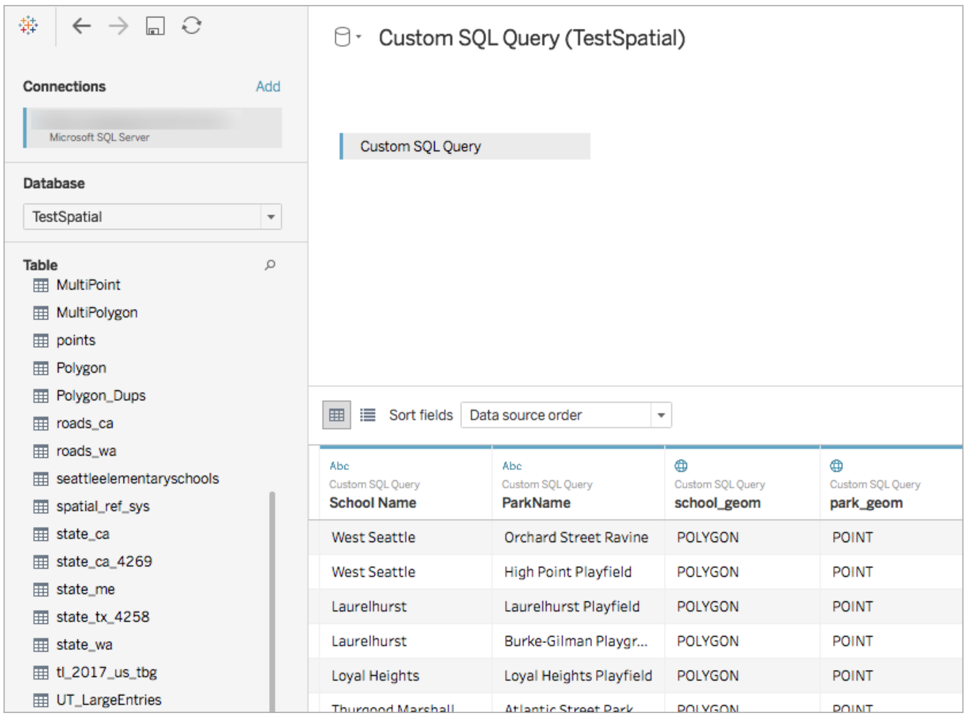
   For this example, the following query is used:

```
SELECT S.[common nam] as SchoolName, P.[common nam] as
ParkName, S.geom.STBuffer(<Parameters.Radius>) as
school_geom, P.geom as park_geom FROM
TestSpatial.dbo.seattleelementaryschools S LEFT JOIN
```

```
TestSpatial.dbo.allseattleparks P on S.geom.STBuffer
(<Parameters.Radius>).STIntersects(P.geom) = 1
```

Note: This example references a parameter named "Radius". For more information, see **Use parameters in a custom SQL query** on page 968.

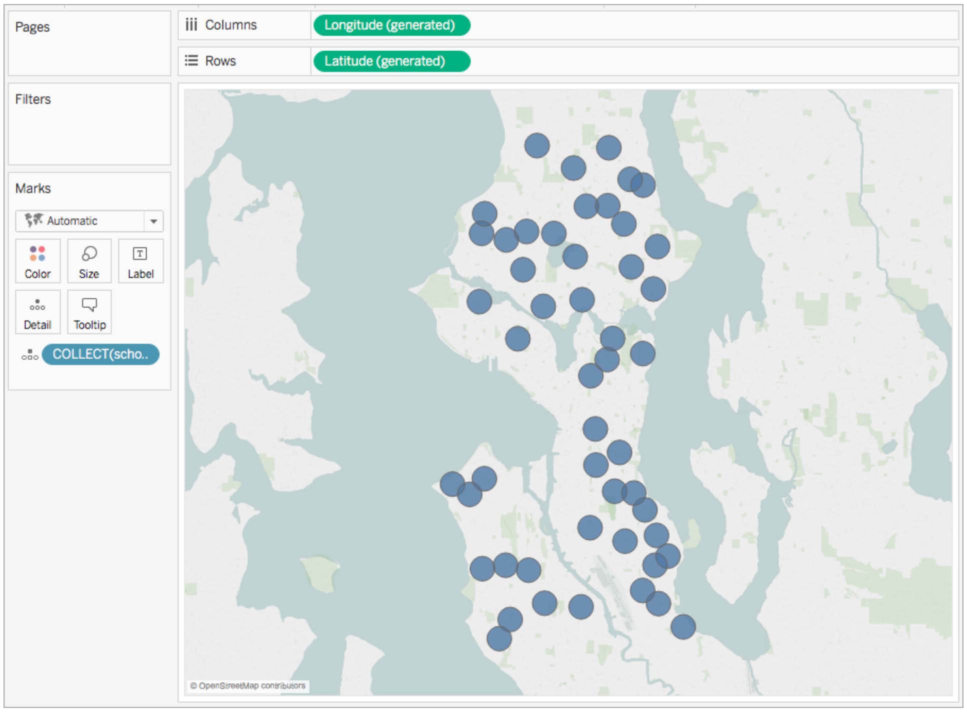
The result of this query looks like the following:



Step 2: Build the map

1. Go to a new worksheet.
2. In the **Data** pane, double-click a spatial field.

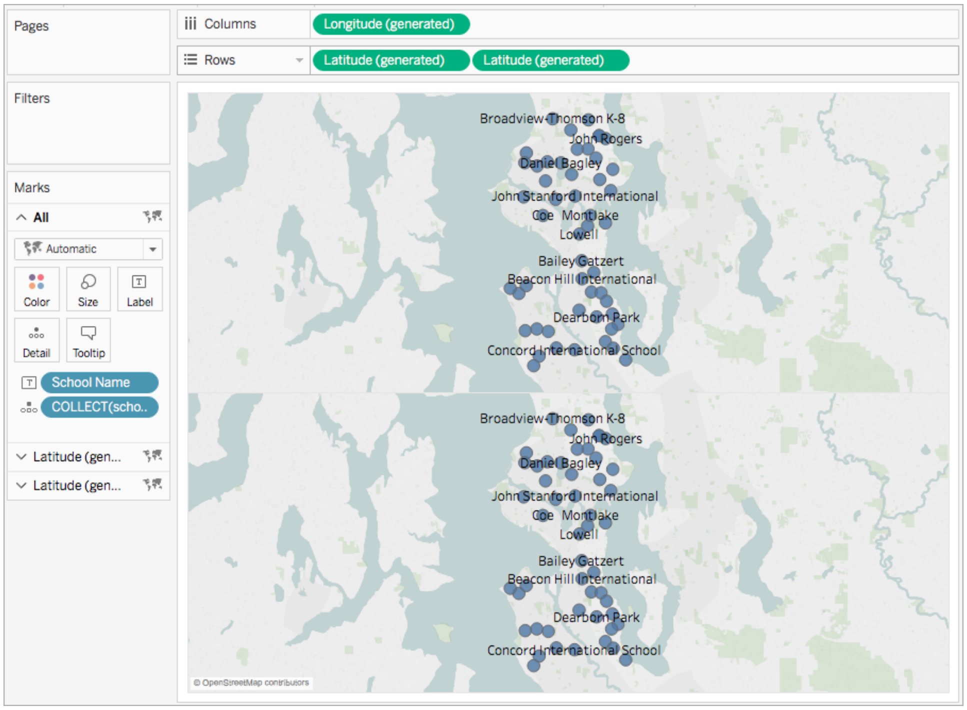
   In this example, **school_geom** is used. When double-clicked, it is automatically added to Detail on the Marks card and a map view is created.



3. From the **Data** pane, drag a dimension to **Label**on the Marks card. In this example, **School Name** is used.

4. On the Rows shelf, control-click (Command-click on a Mac) and drag **Latitude (generated)** to the right. This copies the field.

   A duplicate map appears below the first map and the Marks card updates with two Latitude(generated) tabs. The top tab is for the map on the top and the bottom tab is for the map on the bottom.



5. On the Marks card, click the bottom **Latitude(generated)** tab, and remove the fields by dragging them back to the Data pane.

   The bottom map updates to a blank map.

6. Select **Analysis** > **Create Calculated field**.

7. In the Calculation editor that opens, do the following:

   - Name the calculated field. In this example, the calculated field is named, Intersection.

   - Enter a RAWSQL formula. This example uses the following formula:

     ```
     RAWSQL_SPATIAL("Select %1.STIntersection(%2.STBuffer
     (200))",[school_geom], [park_geom])
     ```

     This formula returns unique spatial data based on the intersection of two values.

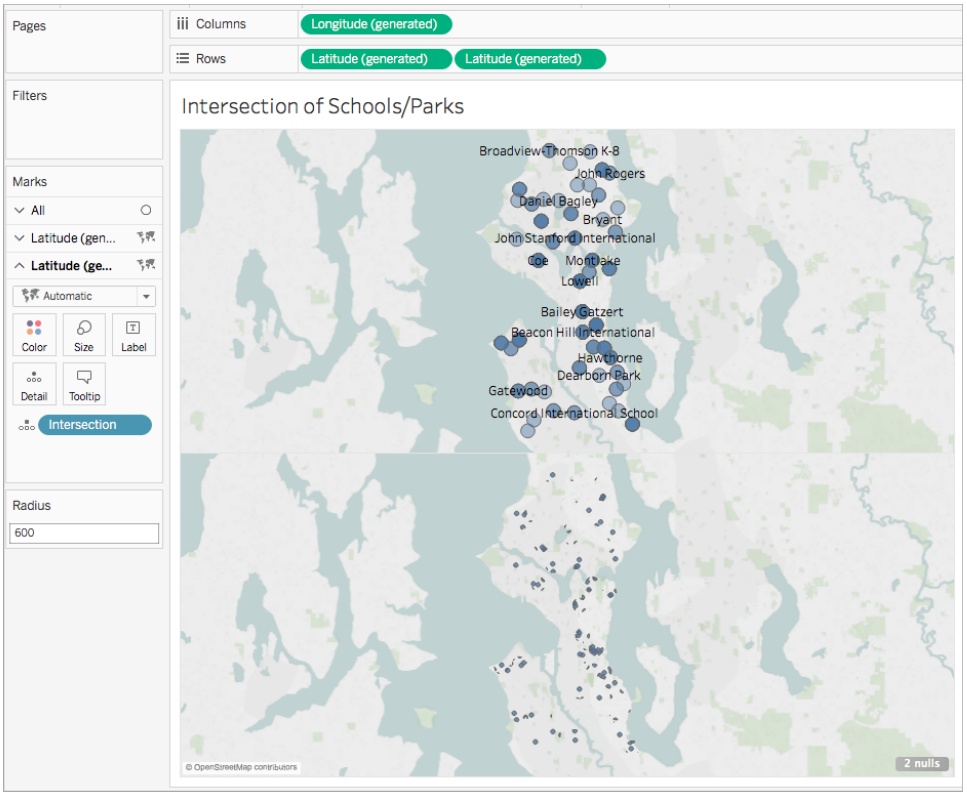
   - When finished, click **OK**.

8. Select **Analysis**, and then clear **Aggregate Measures**.

9. From the **Data** pane, drag the new calculated field (in this case, **Intersection**) to **Detail**

on the Marks card.

The bottom map updates with new marks. Notice that the marks are unique spatial shapes due to the RAWSQL formula.

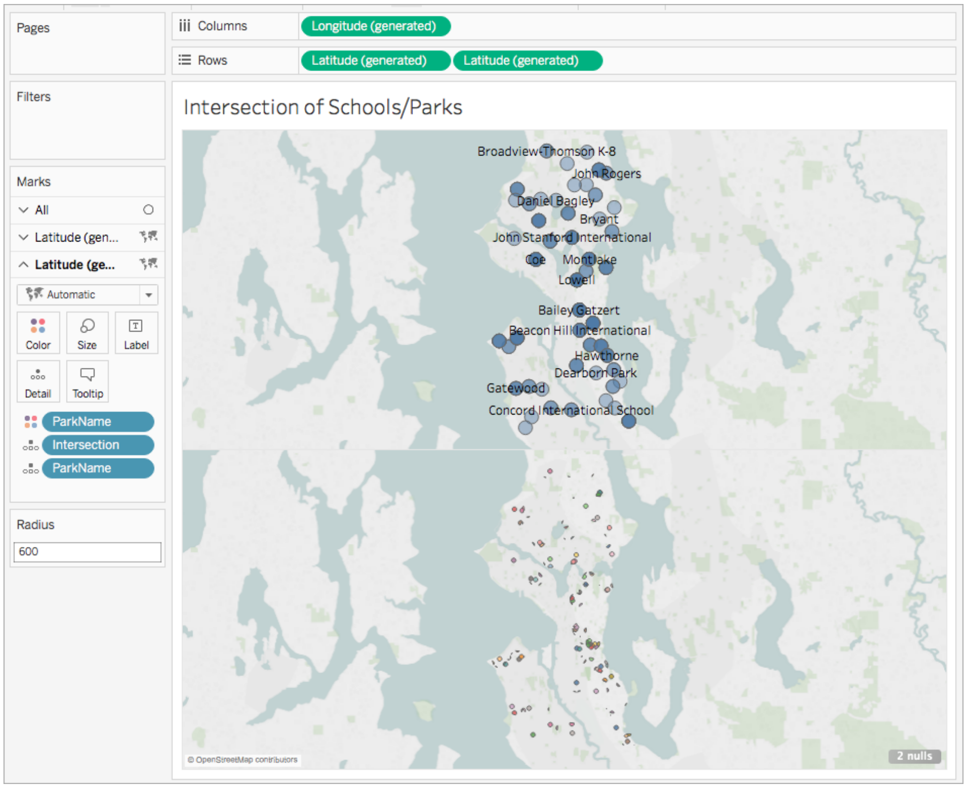


10. From the **Data** pane, drag a dimension to **Detail** on the Marks card. This disaggregates the marks.

    In this example, the dimension, ParkName, is used.

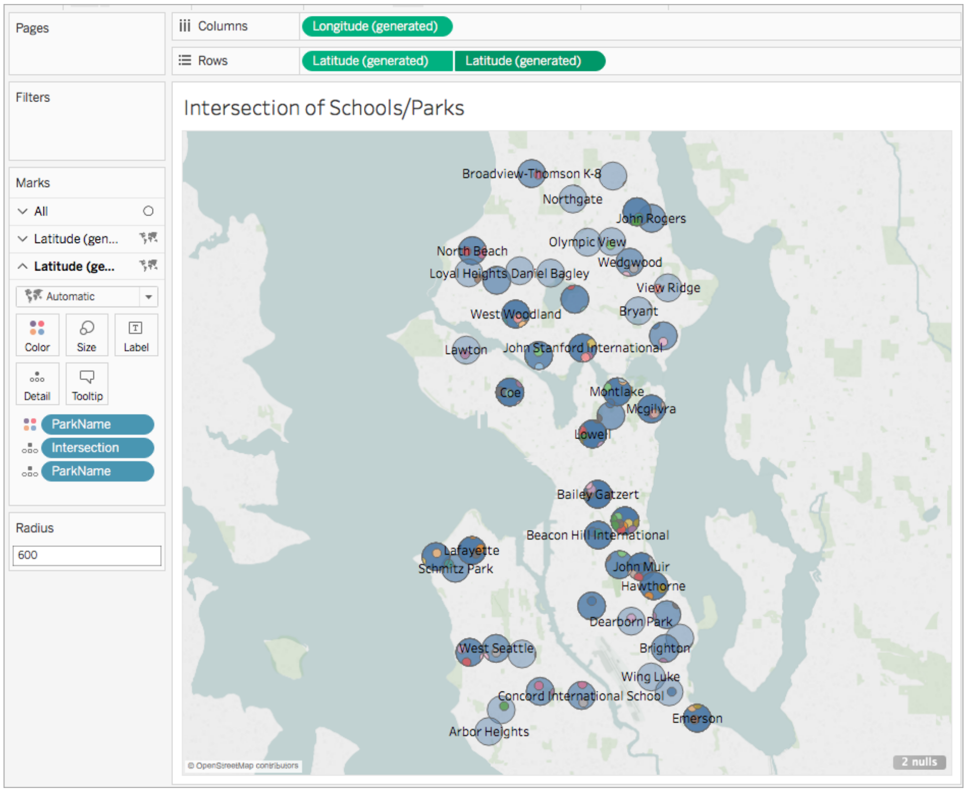
11. From the **Data** pane, drag the same dimension to **Color** on the Marks card.

    In this example, each park is now assigned a color.



12. On the Rows shelf, right-click the **Latitude (generated)** field on the right and select **Dual Axis**.

    The maps are now combined and the data is layered. For more information, see **Create Dual-Axis (Layered) Maps in Tableau** on page 1935.

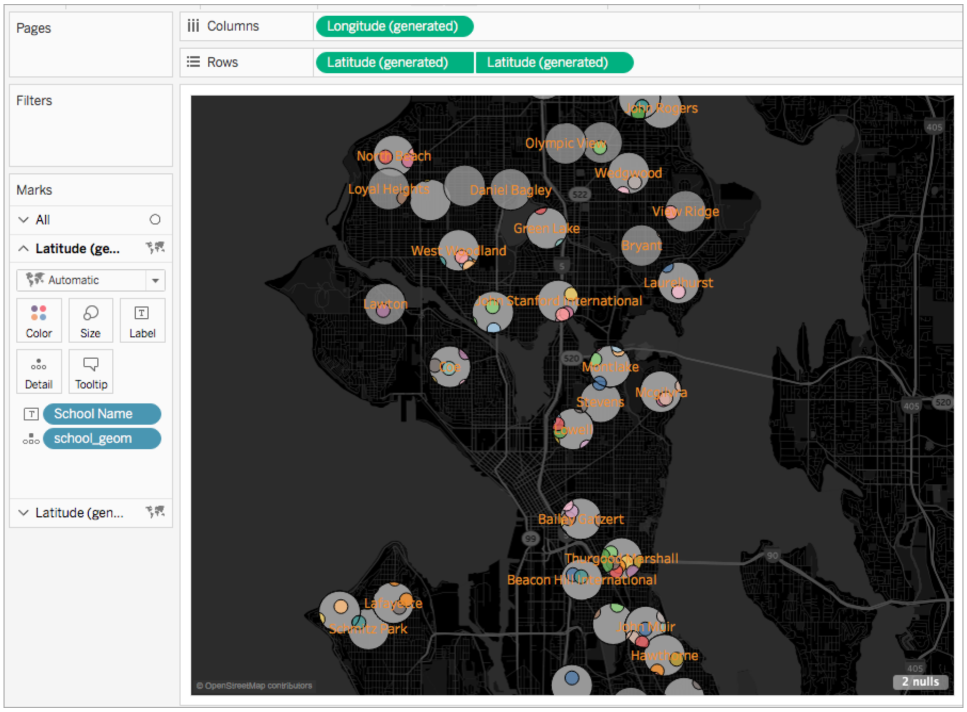


13. Format the map. For more information, see **Customize How Your Map Looks** on page 1965

    In this example, the following formatting is applied:

    - The School Name marks are colored gray.
    - The School Name mark labels are colored orange .
    - The map background style is set to Dark.
    - The Streets and Highways layer is shown on the background map.

    The map is now finished.



See Also

Create Tableau Maps from Spatial Files on page 1855

Connect to a Custom SQL Query on page 959

Pass-Through Functions (RAWSQL) on page 2202

Get Started with Calculations in Tableau on page 2095

Join Spatial Files in Tableau

This article demonstrates how to join spatial files in Tableau using an example.

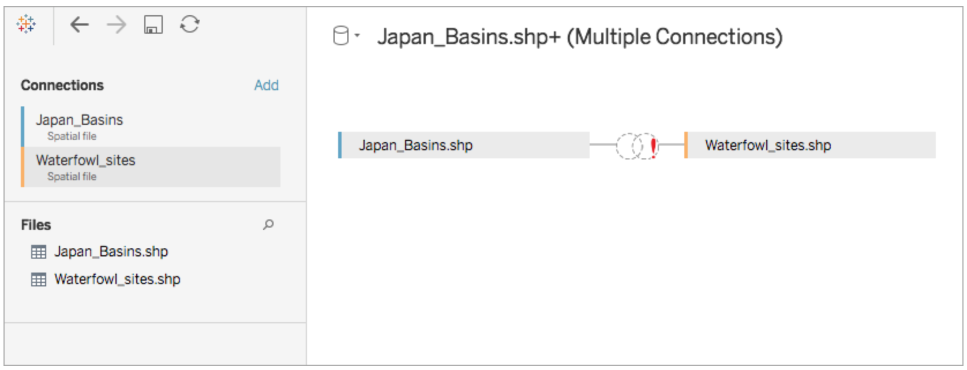
You can connect to the following spatial data sources: Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, and GeoJSON files. You can join two spatial files together, or use a spatial calculation to join a spatial file with non-spatial data that includes latitude and longitude fields. For more information, see **Spatial Functions** on page 2245.

> **Important note:** Tableau supports joining two spatial data sources using their spatial features (geography or geometry). You can only create spatial joins between **points** and **polygons**.

Join spatial files

1. Open Tableau and connect to the first spatial data source. For more information on the types of spatial files you can connect to in Tableau, as well as how to connect to them, see the **Spatial File** on page 1796 connector example.
2. On the upper-left side of the Data Source page, under Connections, click **Add**.
3. In the Add a Data Source menu that appears, connect to your second data source.

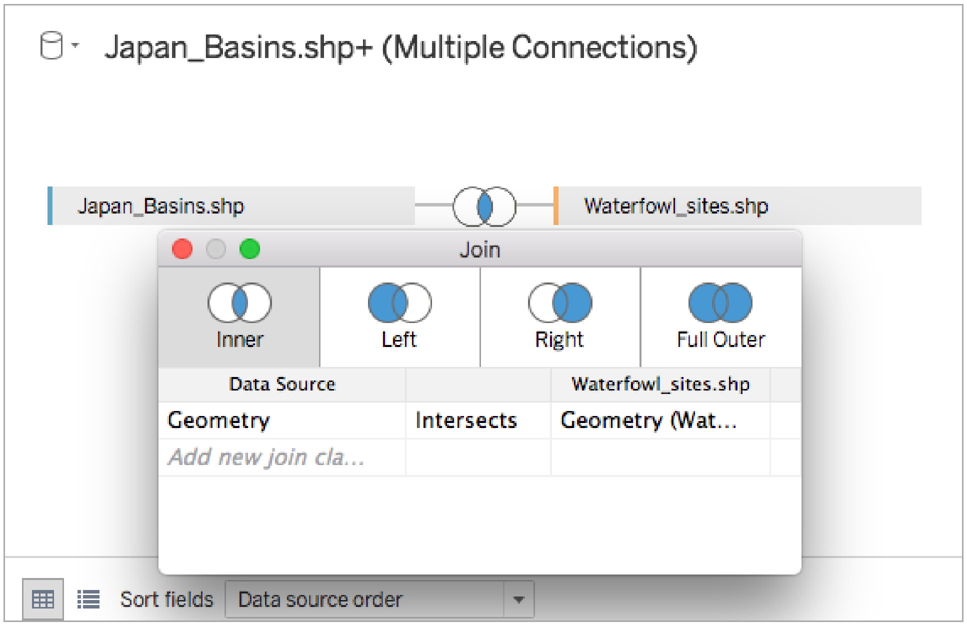
   The two data sources are added to the canvas.



4. Click the Join icon.
5. In the Join dialog box that appears, do the following:
   - Select a join type.

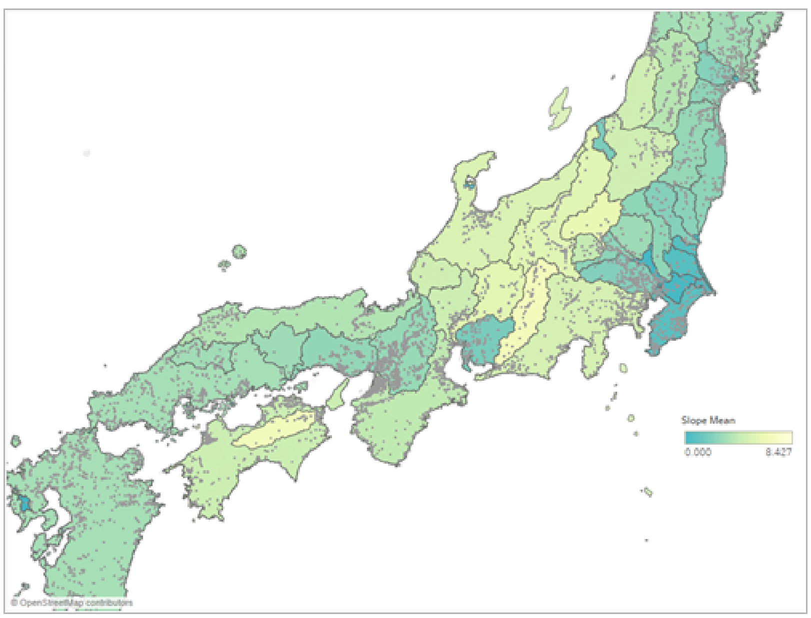
     For more information about each of these types, see **Join Your Data** on page 883.
   - Under Data Source, select a spatial field to join by. (Note, Geometry is the default field name for spatial file sources except in SQL Server, where users create field names). Spatial fields have a globe icon next to them.
   - For the second data source, select another spatial field. If your second data source is not a spatial file and contains latitude and longitude fields, select **Create Join Calculation** as the join clause so that your data can be used in a spatial join. For more information, see **Spatial Functions** on page 2245.
   - Click the **=** sign and then select **Intersects** from the drop-down menu. You can

only intersect two spatial fields.



6. When finished, close the Join dialog box.

   You are now ready to start analyzing your spatial data.



Spatial data can be used to create maps or other chart types in Tableau. For more information on building different chart types, see **Build Charts and Analyze Data** on page 1109.

Troubleshoot spatial joins

SQL Server error: Geometry is incompatible with geography

Although SQL Server supports both a geography and a geometry data type, Tableau only supports geography data type from SQL Server from the following geographies: EPSG:WGS84 = 4326, EPSG:NAD83 = 4269, EPSG:ETRS89=4258. If you try to add a different geography, or a geometry field from SQL Server to your analysis, you'll receive an error.

Vertex ordering

Different spatial systems may order their vertices differently. If two spatial files contain two different vertex orders, it may cause problems with your analysis. Tableau interprets ordering specification according to the order of the spatial data source.

See Also:

Spatial File on page 1796

Join Your Data on page 883

Create Tableau Maps from Spatial Files on page 1855

Create Dual-Axis (Layered) Maps in Tableau on page 1935

Geocode Locations Tableau Does Not Recognize and Plot Them on a Map

If you have locations that Tableau can't map, such as street addresses, you can custom geocode those locations. Custom geocoding means that you assign latitude and longitude coordinates to your locations so Tableau can plot them accurately. It also allows you to create custom geographic roles that you can use as you create map views in Tableau.

For example, if your data contains country, state/province, and street address data, Tableau Desktop will geocode your data to the country and state/province level, but will not recognize the street address data. In this case, you can create a custom geographic role for the street address data so that you can plot it on a map view in Tableau.

Follow the steps below to learn how to custom geocode your data.

> **Before you start**: Custom geocoding your data can be tricky. We recommend that you check out the **Assign Geographic Roles** on page 1834 topic to learn more about geographic roles and the types of data Tableau recognizes before you get started.

Step 1: Create a CSV file with your location data

The first step to custom geocoding your data is creating a CSV file to import into Tableau. The contents of this file differ depending on whether you are extending an existing geographic hierarchy, adding a new geographic role, or adding a new hierarchy to Tableau.

However, the following applies to all custom geocoding files:

- The .csv file must contain Latitude and Longitude columns.
- The latitude and longitude values you add must be real numbers. Make sure to include at least one decimal place when specifying these values.

Extend an Existing Role

The built-in geographic roles in Tableau contain hierarchies that can be extended to include locations relevant to your data. For example, the existing hierarchy of Country > State/Province may not contain all of the states or provinces in your data. You can extend this level to include missing states or provinces.

The import file for this type of geocoding should contain every level of the hierarchy above the level you are extending. For example, if you are extending (adding data to) the State/Province geographic role, the existing State/Province hierarchy has a column for Country and State/Province, along with Latitude and Longitude.

	A	B	C	D
1	**Country (Name)**	**State/Province**	**Latitude**	**Longitude**
2	England	United Kingdom	51.5000	0.1167
3	Scotland	United Kingdom	55.9500	3.1833
4	Wales	United Kingdom	51.4833	3.1833
5	Northern Ireland	United Kingdom	54.6000	5.9167

In the import file, the names of the columns define the geographic roles. When extending an existing role, the column names must match the existing geographic roles in the hierarchy that you are extending. This will ensure that the new locations are added to the proper roles and hierarchies.

See the table below for information on how to organize hierarchies in your import file.

Built-In Hierarchy	Columns to include in the .csv file (in order; left to right)
Country (Name)	Country (Name) Latitude Longitude
Country (Name), State/Province	Country (Name) State/Province Latitude Longitude
Country (Name), State/Province, City	Country (Name) State/Province City Latitude Longitude
Country (Name), State/Province, County	Country (Name) State/Province County Latitude Longitude
Country (Name), ZIP Code/Post-code	Country (Name) ZIP Code/Postcode Latitude Longitude
Country (Name), Area Code	Country (Name)

	Area Code Latitude Longitude
Country (Name), CBSA	Country (Name) CBSA Latitude Longitude

In addition to the Country (Name) column, you can optionally include the following columns: Country 2 char (ISO 3166-1), Country 3 char (ISO 3166-1), and Country (FIPS 10). If you include these columns, they should be just to the right of the Country (Name) column in any order.

Note: The highest level in the hierarchy is Country and cannot be extended to include higher levels such as Continent, etc.

Add New Roles

To add new geographic roles to the existing geographic hierarchy in Tableau, format your import file to include the new roles along with their parent roles.

When adding new roles to an existing hierarchy, the import file for those roles needs to contain the columns for each level in the existing hierarchy. For more information on the columns to include in each geographic hierarchy, see the built in hierarchy table in the Extend An Existing Role section.

Below is an example of an import file containing the locations of crater impacts in North America. Importing the file below would add the geographic role Crater Name to the existing Country (Name), State hierarchy. Notice that the column name for country matches the existing Country (Name) geographic role.

Crater Name	State	Country (Name)	latitude	longitude
Ames	Oklahoma	U.S.A.	35.007752	-97.092877
Avak	Alaska	U.S.A.	64.200841	-149.493673
Barringer	Arizona	U.S.A.	34.048928	-111.093731
Beaverhead	Montana	U.S.A.	46.879682	-110.362566
Brent	Ontario	Canada	51.253775	-85.323214
Calvin	Michigan	USA	44.314844	-85.602364
Carswell	Saskatchewan	Canada	52.939916	-106.450864
Charlevoix	Quebec	Canada	46.813878	-71.207981
Chesapeake Bay	Virginia	U.S.A.	37.431573	-78.656894
Chicxulub	Yucatan	Mexico	20.709879	-89.094338

Once you have created your import file, save the file as a Comma Delimited (.csv) file (Windows Comma Separated if on a Mac) in a folder on your computer.

Step 2 (Optional): Create a schema.ini file

Sometimes when you attempt to import custom geographic information in Tableau using a .csv file, you may see a "`.csv could not be used because it does not contain a unique column`" error message.

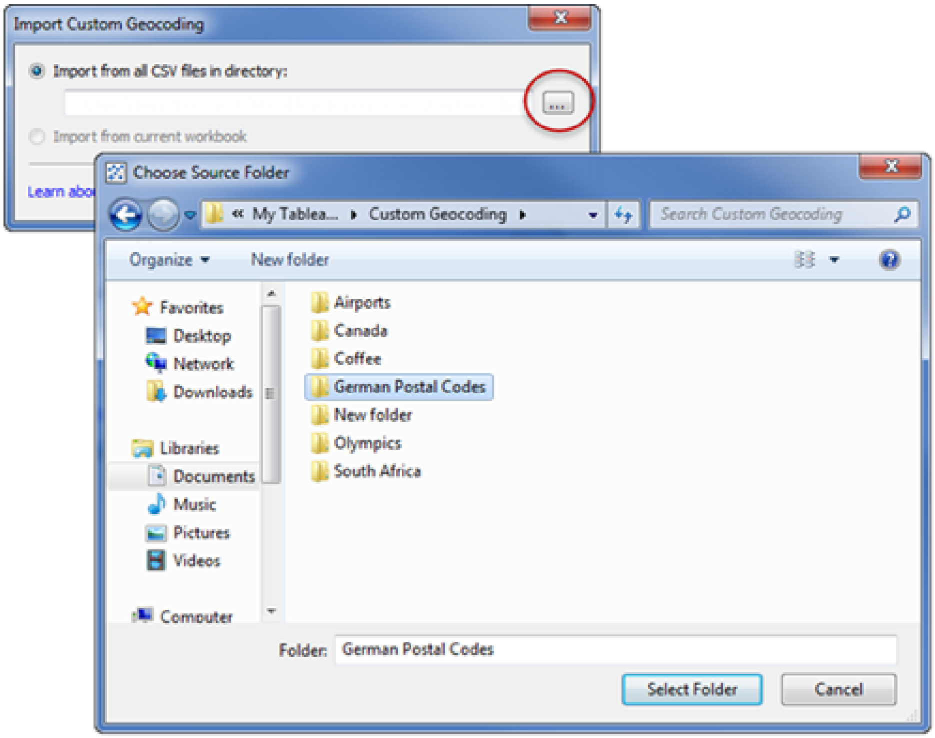
This is because your .csv file might contain numerical data, such as numeric postcodes. Tableau will only accept text fields for new geographic roles. However, you can create a schema.ini file to tell Tableau that the numeric field you want to import should be treated as a text field.

To learn how to create a schema.ini file, see **Create a schema.ini File** on page 1825.

Step 3: Import custom geocoding file(s) into Tableau Desktop

Once you have created a .csv file with custom geocoding you can import that file into Tableau.

1. Open Tableau Desktop and navigate to a new or existing worksheet.
2. Select **Map** > **Geocoding** > **Import Custom Geocoding**.
3. In the Import Custom Geocoding dialog box, click the button to the right of the text field to browse to the folder your import file is saved in. All files in the folder will be imported into Tableau.



4. Click **Import**.

The custom geocoding data is imported into the workbook and the new geographic roles become available.

Where does Tableau store my custom geocoding data?

When you import custom geocoding, the data is stored in the Local Data folder in your Tableau Repository. To import custom geocoding, your Tableau Repository must be on a local hard drive. The custom geocoding is then available for all workbooks.

Save custom geocoding files in a separate folder on your local computer. If you have several different sets of custom geocoding files, save each of them under their own folder names.

You can remove the custom geocoding stored in your Tableau Repository by clicking **Map** > **Geocoding** > **Remove Custom Geocoding**. This will not remove the geocoding from a packaged workbook, but it will remove it from the Local Data folder in your My Tableau Repository.

What happens to my custom geocoding data when I save my workbook?

When you save your workbook as a packaged workbook, the custom geocoding data is packaged with the workbook. When you open a packaged workbook you can import the custom geocoding from that workbook into your own Tableau Repository.

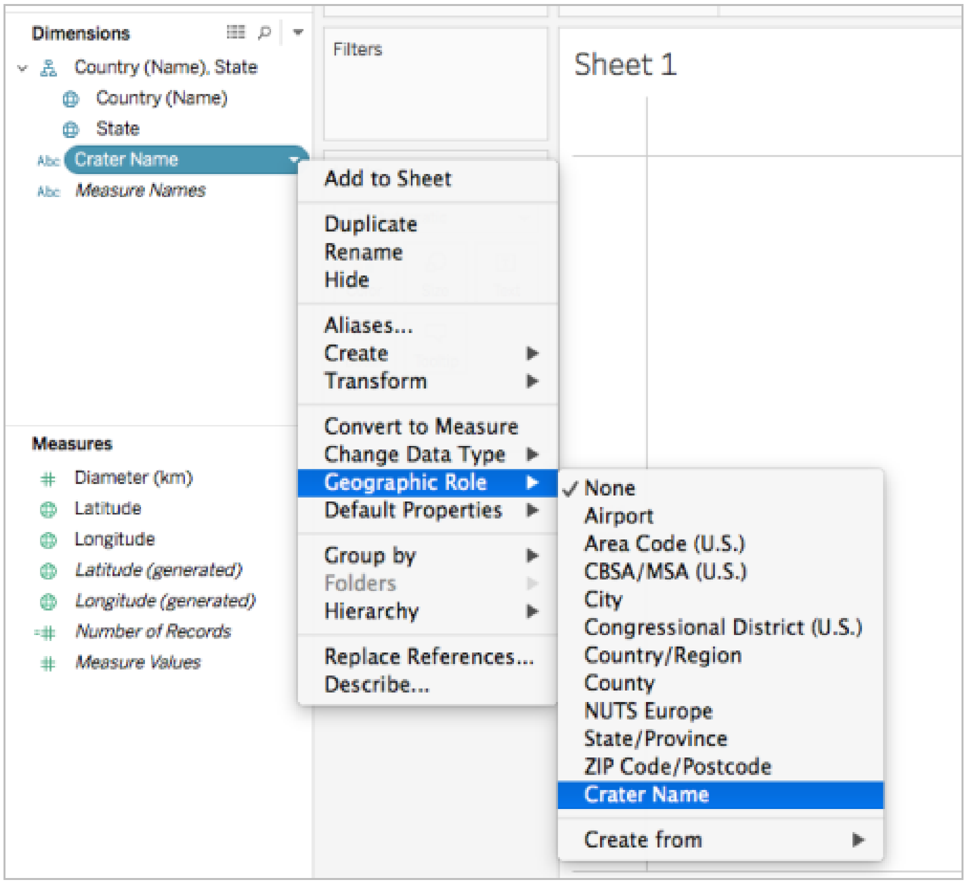
> **Note:** Importing a new custom geocoding file will replace any custom geographic roles previously imported into Tableau.

Step 4: Assign geographic roles to your fields

Once you import custom geocoding into your workbook, the custom geographic roles become available. The next step is to assign those geographic roles to fields in your data source.

For example, if you decided to add a new geographic role that lists the latitude and longitude coordinates of crater impacts in North America, you can assign that geographic role to a field in your data source that lists the names of craters so when you create a map view with that field, the crater impacts are plotted in the correct locations on the map view.

To assign a custom geographic role to a field, in the Data pane, click the data type icon next to the field and select **Geographic Role**. Next, select a custom geographic role from the list.



You're now ready to start building a map view with your custom geographic roles. For more information on how to build a map view, see **Mapping Concepts in Tableau** on page 1781.

See also

Data Blending vs. Custom Geocoding on page 1832

Blend Geographic Data on page 1827

Create a schema.ini File

Sometimes when you attempt to import custom geographic information in Tableau using a .csv file, you may see a ".csv could not be used because it does not contain a unique column" error message.

This is because your .csv file might contain numerical data, such as numeric postcodes. Tableau will only accept text fields for new geographic roles. However, you can create a schema.ini file to tell Tableau that the numeric field you want to import should be treated as a text field.

A schema.ini file is a Microsoft configuration file associated with the Microsoft JET engine. It tells JET how to interpret the contents of text files, including data structure, date formats, and other settings. For example, creating a schema.ini file that specifies that the numeric postcodes in your import file should be treated as text will allow you to plot the postcodes on a map.

To create a schema.ini file

1. Open a text file.
2. In the text file, using the following syntax, specify the names and data types for each column in your import file:

   ```
   [YOURCSVFILENAME.csv]
   ColNameHeader=True
   Format=CSVDelimited
   Col1="Name of 1st Column Header in your .csv file"
   Dataype
   Col2="Name of 2nd Column Header in your .csv file"
   Dataype
   Col3="Name of 3rd Column Header in your .csv file"
   Dataype
   Col4="Name of 4th Column Header in your .csv file"
   Dataype
   ```

   For example, if your import (.csv) file contained the columns **German Post Code**, **Latitude**, and **Longitude**, and was named **German Postal Codes.csv**, your schema.ini file would look like the following:

   ```
    [German Postal Codes.csv]
   ColNameHeader=True
   Format=CSVDelimited
   Col1="German Post Code" Text
   Col2="Latitude" Double
   Col3="Longitude" Double
   ```

3. Save the file as **schema.ini** in the same folder as your.csv file.

You're now ready to import your .csv and schema.ini files into Tableau.

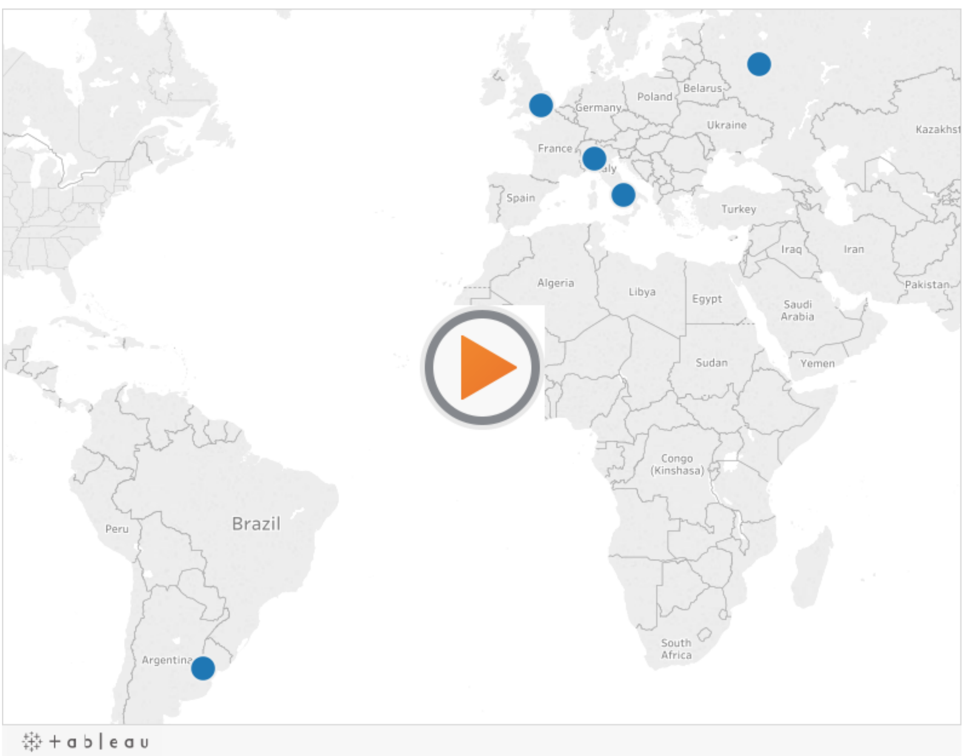
See also

Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1818

Blend Geographic Data

This example demonstrates how to blend geographic data in Tableau using two small sample data sources. Follow the steps in this topic to learn how to create a file that defines your geographic data, blend two data sources, and build the map view below using the two data sources.

The embedded view below shows famous theaters around the world, and was created by blending geographic data with another data source. Hover over the marks in the view to see information about each theater. If you would like to follow along with the steps in this topic, click **Download** in the view below, and then select **Tableau Workbook**. This will open the view in Tableau Desktop.



Step 1: Create a file that defines your geographic data

When you have a data source that contains geographic information that is not automatically geocoded in Tableau, the first step is to create a second data source that defines the latitudinal and longitudinal values for that information. You will later connect to this file in Tableau Desktop and blend it with your original data source.

Create a second data source with the following information:

- **Geographic name column**: This column includes any geographic information you want to plot on a map view and typically matches a geographic column name in your original data. For example, if your original data source has a column called Street Address, your second data source should also have a column called Street Address. The data in this column would then be all of the street addresses you want to plot on a map view.

  If the column does not match a geographic column name in your original data source, you may need to edit the relationships between your two data sources in Tableau Desktop. For more information, see Step 3.

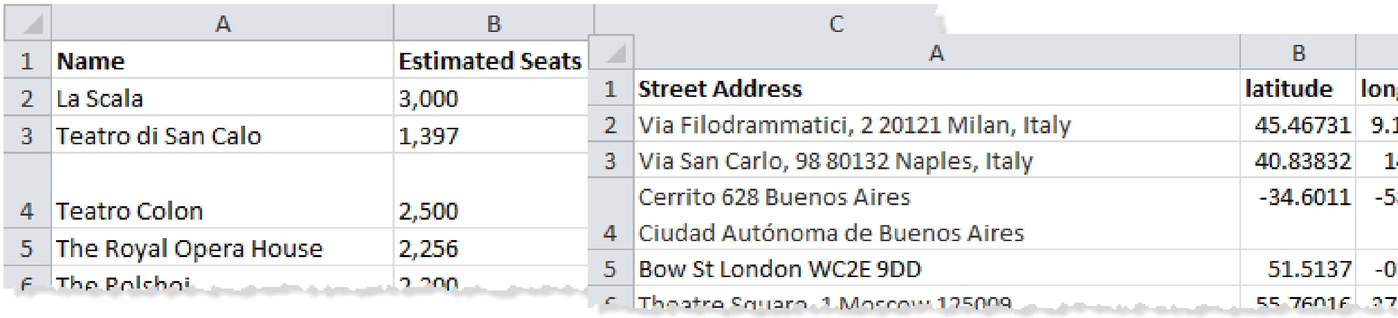
  Finally, this column should not match the names of any geographic roles already in Tableau, such as **County**, **Area Code**, or **CBSA/MSA**.

  > **Note:** The data in these columns can be strings or numbers. However, if your locations are numbers, especially numbers with leading zeros (for example, 00501), make sure the data type of the field is set to **String** in Tableau. For more information, see **Data Types** on page 325.

- **Latitude and Longitude columns**: The values in these columns need to be in decimal degrees (for example, 47.651808, -122.356881).

Below is an example of an original data source with geographic locations, and a second data source that defines those geographic locations.

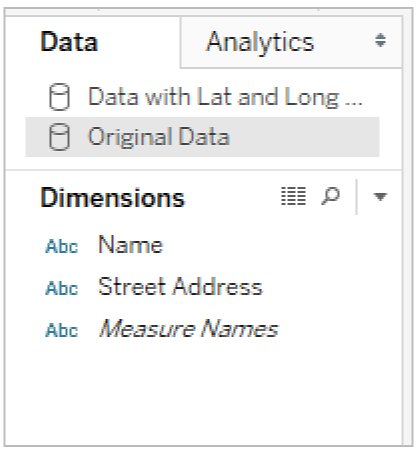
Original data source **Second data source**



Step 2: Connect to the data sources

1. Connect to the original data source.

2. Connect to the data source that defines your geographic data.



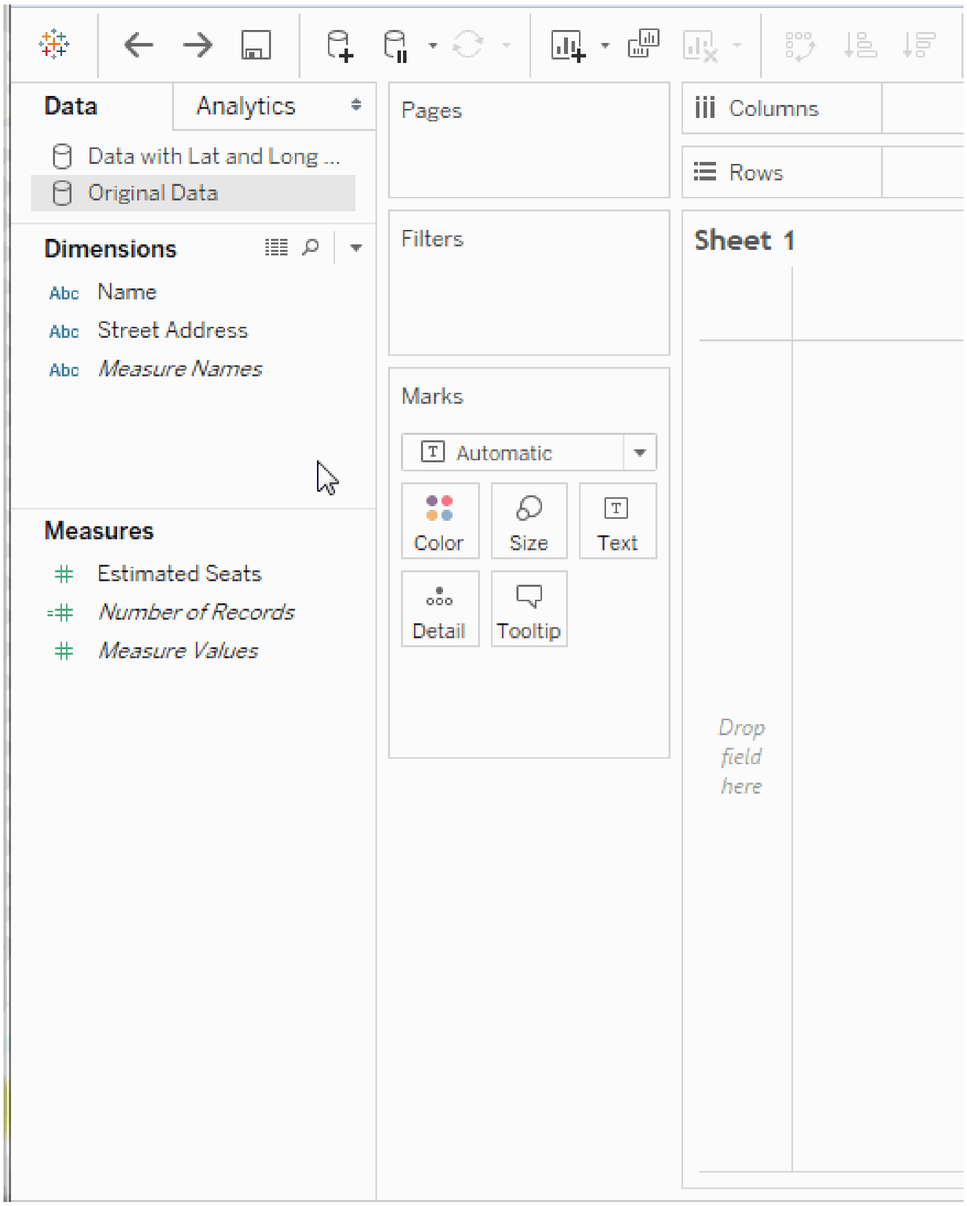
Step 3: Edit Relationships

(Optional) If the two data sources share the same column names, your data should blend automatically when you add fields to the view. However, if the file you created to define your geographic data does not share the same column name as the geographic data in your original data source, you will need to create a relationship between those two fields using the **Edit Relationships** option. For information about how to edit relationships, see **Blend Your Data** on page 910.

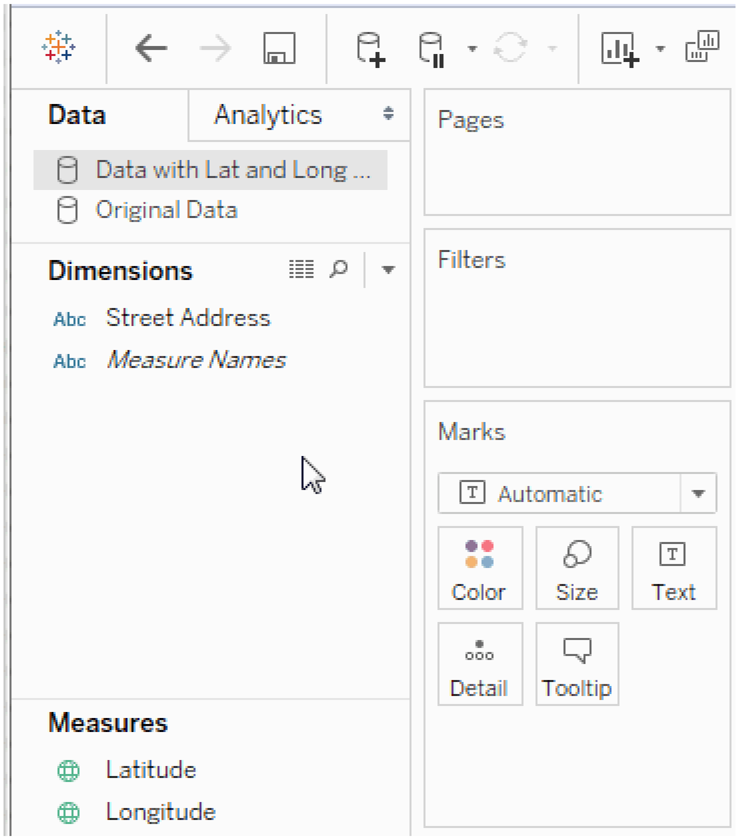
Step 4: Plot the data on a map view

1. In the **Data** pane, select the second data source that defines your geographic data. Under **Measures**, assign the **Latitude** geographic role to the **Latitude** field and the

Longitude geographic role to the **Longitude** field. For more information, see **Assign Geographic Roles** on page 1834.



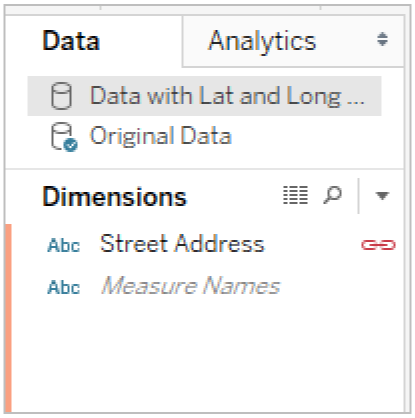
2. In the **Data** pane, select the original data source, and then drag the field you want to plot on a map to **Detail** on the **Marks** card.



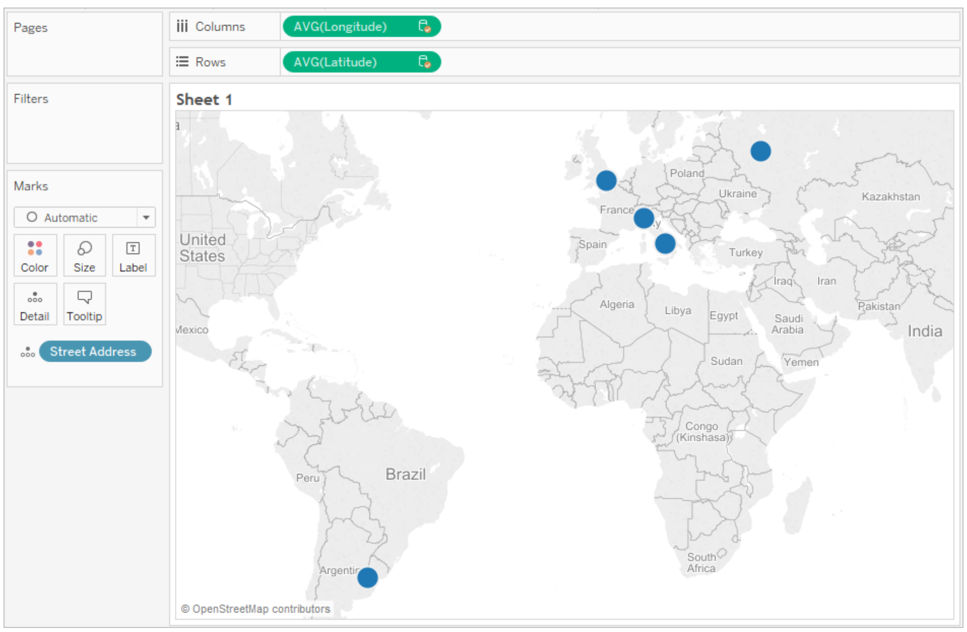
> **Note:** This field should have a data type of String, and should be a dimension. For more information on how to change the data type of a field, see **Data Types** on page 325.

3. In the **Data** pane, select the second data source , and then double-click **Latitude** and **Longitude** to add them to the **Columns** and **Rows** shelves.

   The link icon indicates that the **Street Address** field is blended with the **Street Address** field in the original data source. A broken link icon indicates that the **Street Address** field is not blended with the **Street Address** field in the original data source.



The view is now complete. Each mark represents an address from the original data source.



See also

Data Blending vs. Custom Geocoding below

Data Blending vs. Custom Geocoding

If you have geographic locations in your data that are not automatically geocoded in Tableau, there are two ways to plot them on a map view–data blending and custom geocoding. Both data blending and custom geocoding allow you to plot your own locations on a map. Data blending is easier to set up and you can work with data from any data source. Custom geocoding lets you add to existing roles, and create hierarchies. Custom geocoding can be easier to use once it is set up and imported.

This topic explains the difference between blending geographic information with another data source and importing custom geocoding data into Tableau.

The following is a quick comparison of data blending and custom geocoding.

Capability	Data Blending	Custom Geocoding
Plot your own locations on a map view	Yes	Yes
Use any data	Yes	No, text files only
Add new geographic roles	No	Yes
Add to an existing geographic role	No	Yes
Create new geographic hierarchies	No	Yes
Can be reused for other workbooks	No	Yes

Data Blending

Data blending is the easiest way to plot your data on a map view. Data blending works great if you are adding a single level of geographic information with a latitude and longitude. You can use any data source, unlike Custom geocoding where you can only use text files. You can share the geographic data source with other workbooks on Tableau Server. You can also speed up your map performance by creating an extract that contains only your geocoding data.

The following is a high-level procedure for blending geographic data. For a more detailed example, see **Blend Geographic Data** on page 1827.

To blend geographic data

1. Create a data source that defines your geographic data.
2. In Tableau Desktop, connect to the original data you want to map, and then connect to the data source that defines your geographic data.
3. Blend the two data sources together. For more information, see **Blend Your Data** on page 910.
4. Plot the data on a map view.

Custom Geocoding

Custom geocoding is a more flexible way to plot your data on a map. Custom geocoding is available for all workbooks on a computer once the custom geocoding data is imported. The custom geocoding data will be copied into any packaged workbook (.twbx) or published workbook that uses a custom geographic role. This will make the workbooks about 40 MB larger.

If you upgrade Tableau Desktop, you may need to refresh your custom geocoding to take advantage of any fixes made to the geocoding data in Tableau.

In custom geocoding, you can use additional columns to define larger geographic locations. For instance, if you are creating a set of US census tracts, you may need to define which US county they fall in. You can include additional columns to define larger geographic locations in the import file.

You can also use multiple files for multiple geographic roles that have a matching relationship, meaning they share larger geographic data, such as country or state/province. Once the custom geocoding data is imported, you will see additional geographic roles that can be assigned to your geographic data.

Finally, custom geocoding lets you add additional places to an existing geographic role, such as adding new cities to the city role. It also allows you to define a hierarchy of geographic roles that extends the built-in hierarchies (e.g. census tracts in counties) or defines a new hierarchy (e.g. sub-territories in territories).

For more information about custom geocoding, see **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map** on page 1818. Additionally, you can learn how to custom geocode data by watching the Custom Geocoding training video on the Tableau website.

See also

Assign Geographic Roles below

Create a schema.ini File on page 1825

Assign Geographic Roles

This article describes how to assign a geographic role to a field in Tableau so you can use it to create a map view.

A *geographic role* associates each value in a field with a latitude and longitude value. When you assign a geographic role to a field, Tableau assigns latitude and longitude values to each location in your data based on data that is already built in to the Tableau map server.

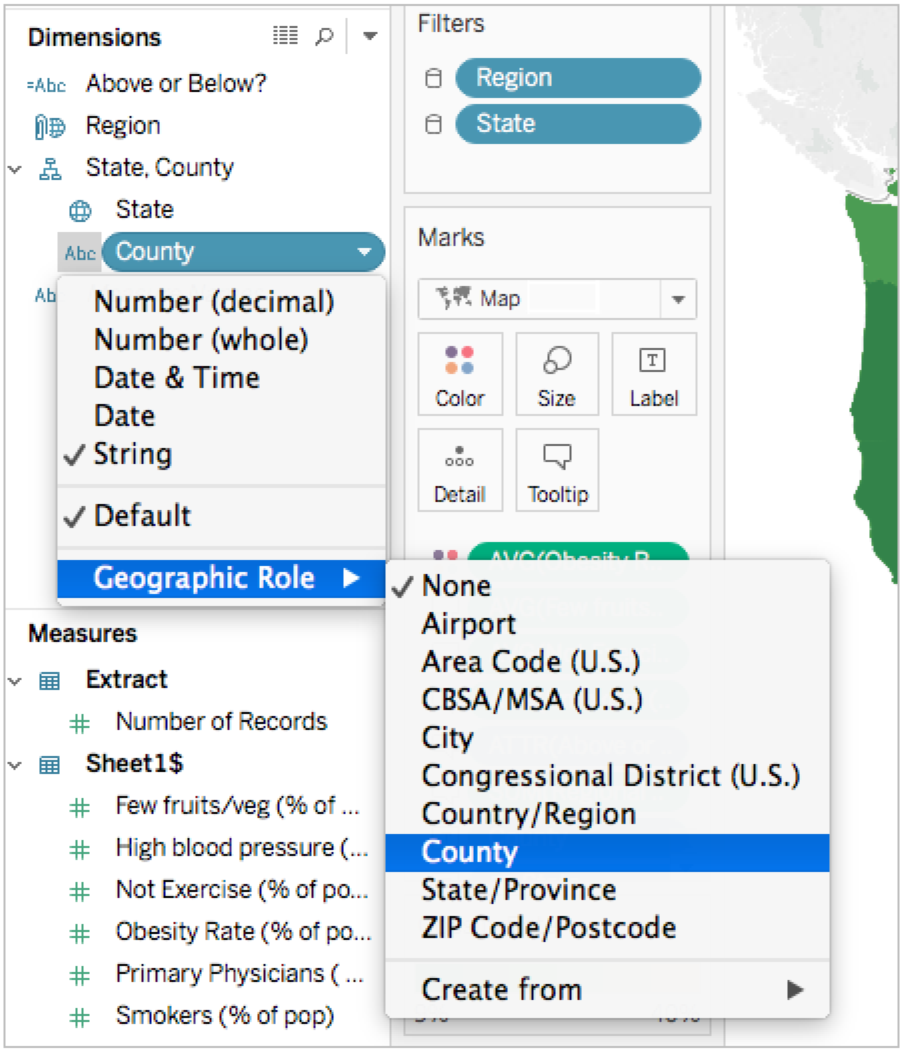
Assign a geographic role to a field

Assigning a geographic role based on the type of location (such as state versus postcode) helps insure that your data is plotted correctly on your map view. For example, you can assign the City geographic role to a field that contains a list of city names.

When a field is assigned a geographic role, Tableau creates a map view when you add the field to **Detail** on the **Marks** card. In other words, Tableau *geocodes* the information in that field.

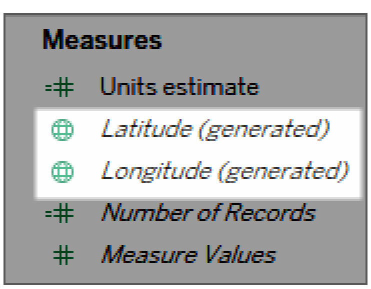
To assign a geographic role to a field:

- In the Data pane, click the data type icon next to the field, select **Geographic Role**, and then select the geographic role you want to assign to the field.



When you assign a geographic role to a field, Tableau adds two fields to the **Measures** area of the **Data** pane: Latitude (generated) and Longitude (generated).

These fields contain latitude and longitude values and are assigned the Latitude and Longitude geographic roles. If you double-click each of these fields, Tableau adds them to the Columns and Rows shelves and creates a map view using the Tableau background map.



Types of geographic roles in Tableau

The following table describes the geographic roles available in Tableau. Many of the roles are international, but some are limited to the U.S. only.

You can assign geographic roles to your fields based on the type of geographic data they contain. For example, you can assign the Airport geographic role to a field that contains International Air Transport Association (IATA) codes.

If your location data does not fit into one of these roles, you may have to import custom geocoding to plot the data on a map. For more information, see **Geocode Locations Tableau Does Not Recognize and Plot Them on a Map** on page 1818.

Geographic Role	Assign this role to a field if it contains:
Airport	International Air Transport Association (IATA) or International Civil Aviation Organization (ICAO) airport codes.
Area Code (U.S.)	U.S. telephone area codes; numbers only.
CBSA/MSA (U.S.)	U.S. Core Based Statistical Areas (CBSA), which includes Metropolitan Statistical Areas (MSA), as defined by the U.S. Office of Management and Budget. CBSA/MSA Codes and Names are recognized.
City	Worldwide cities with population of 15,000 or more. Names are in English (UK or US), French, German, Spanish, Brazilian-Portuguese, Japanese, Korean, and Chinese (Simplified and Traditional).
Congressional District (U.S.)	U.S. congressional districts.
Country/Region	Worldwide countries, regions, and territories. Names are in English (UK or US), French, German, Spanish, Brazilian-Portuguese, Japanese, Korean, and Chinese (Simplified and Traditional). Tableau also recognizes, FIPS 10, ISO 3166-1 alpha 2, and ISO 3166-1 alpha 3. Names are included in various forms, including long, short, and various abbreviations.

County	Second-level administrative divisions for select countries. For example, U.S. counties, French départements, German kriese, etc. **Note:** Second-level administrative division definitions vary by country. In Tableau, all second-level administrative divisions are geocoded with the County geographic role. For more information, see the **Location Data that Tableau Supports for Building Map Views** on page 1795 section.
NUTS Europe	NUTS (Nomenclature of Territorial Units for Statistics) levels 1 - 3 codes. Codes and names, including synonyms, are supported.
Latitude	Latitude in decimal degrees. Only available for numeric fields.
Longitude	Longitude in decimal degrees. Only available for numeric fields.
State/Province	Worldwide state, province, and other first-level administrative divisions. Names are in English (UK or US), French, German, Spanish, Brazilian-Portuguese, Japanese, Korean, and Chinese (Simplified and Traditional). **Note:** Some names are available only in their local form.
ZIP Code/Postcode	ZIP codes and postcodes for select countries. For example, U.S. five-digit zip codes, Australian four-digit postcodes, German five-digit postcodes, etc. For more information, see the **Location Data that Tableau Supports for Building Map Views** on page 1795 section.

See also:

Location Data that Tableau Supports for Building Map Views on page 1795

Geocode Locations Tableau Does Not Recognize and Plot Them on a Map on page 1818

Blend Geographic Data on page 1827

Create Tableau Maps from Spatial Files on page 1855

Edit Unknown or Ambiguous Locations on the facing page

Edit Unknown or Ambiguous Locations

Sometimes Tableau will not recognize one or more of the location names in your data. When this happens those values are marked as **unknown** in the lower right corner of the map view. This may happen if there is a location or abbreviation that Tableau does not recognize, or if the location is ambiguous and could exist in multiple places.

When this happens, you can add additional information to the view to define locations, or can edit the unknown location names to map to known locations.

Add more fields to view

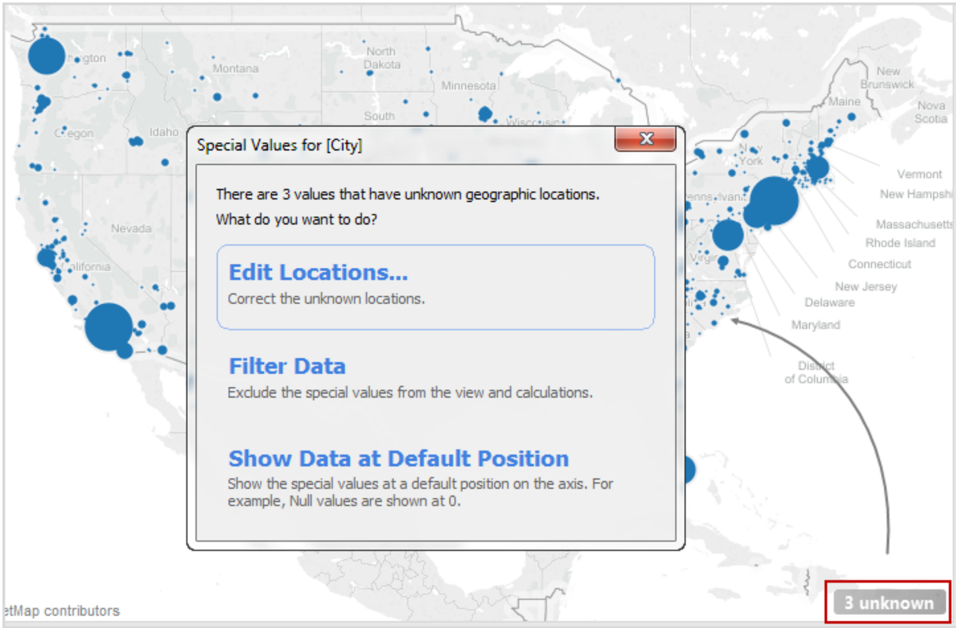
If your data set includes ambiguous locations (for example, "Aberdeen," which could appear in multiple states or countries), adding another geographic field, like State or Country, to the view defines the correct location for that data. If there is a hierarchy in the data pane, Tableau will automatically use the appropriate levels of the hierarchy to solve location ambiguities.

Edit locations in the Special Values menu

Sometimes, unknown locations can be prompted because there's a misspelling or another issue. When this happens, you can correct the unknown or ambiguous locations in your data using the **Special Values** menu. To do this:

1. In the bottom right corner of the view, click the special values indicator that lists the number of unknown locations in your map. This opens the Special Values menu.

2. In the Special Values dialog box, select **Edit Locations**.



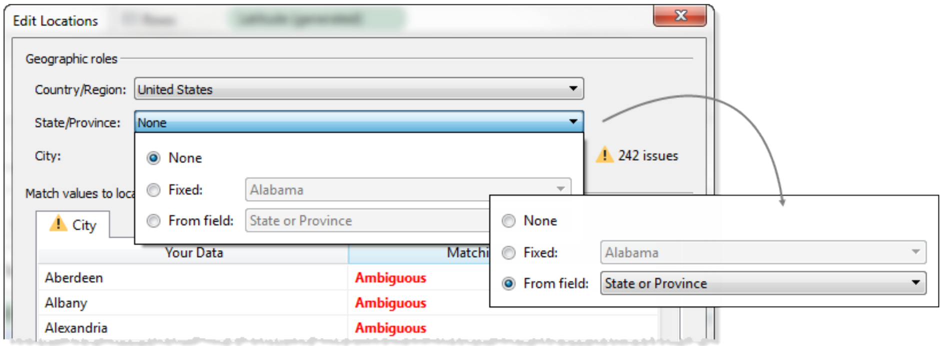
> **Note:** If the special values indicator is not visible, you can also select **Map** > **Edit Locations**

The Edit Locations dialog lets you clarify unrecognized and ambiguous values to locations in Tableau.

Edit ambiguous locations

Both unrecognized and ambiguous locations can be listed in the Edit Locations dialog box.

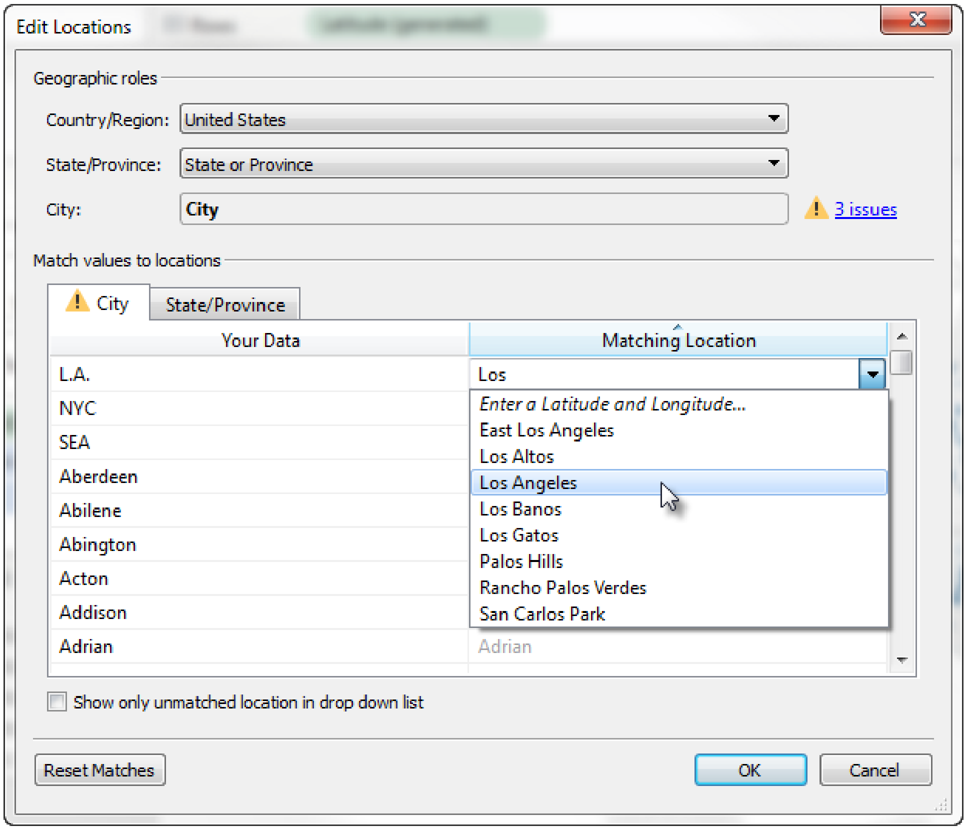
You can fix some ambiguous locations by specifying the **Country/Region** and/or **State/Province** for each city from a specific field in your data, either by hard-coding a value if your data only spans one state or country, or by telling Tableau which fields to refer to for that information. This option is in the Geographic Roles section of the Edit Locations dialog box. For example, if you have several cities that are ambiguous, you can specify a State/Province to fix them.



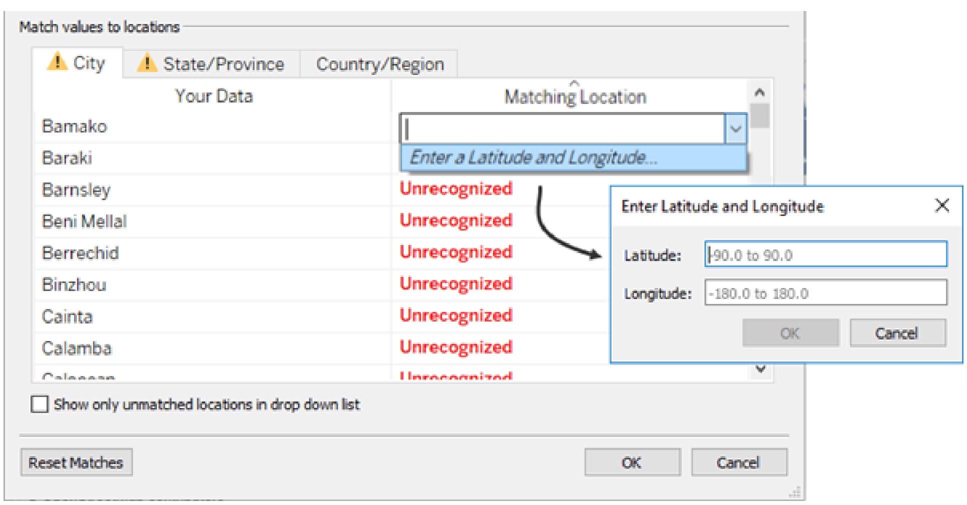
If there are ambiguous or unknown locations in the Country/Region or State/Province tabs in the Edit Locations menu, fixing those first may resolve some ambiguous locations in smaller geographic roles, such as cities.

Edit unknown locations

In the Edit Locations dialog box, click on one of the **Unrecognized** cells to match a known location to your unknown data. When you click on an unrecognized cell, a search box appears. As you begin typing in the search box, Tableau generates a list of possible locations. Select a location from the list.



Alternatively, you can enter latitude and longitude to manually map a value to a point location on the map. To do this, begin typing into the unrecognized cell and select **Enter Latitude and Longitude** from the drop-down menu.



> **Note:** When you type a latitude and longitude, enter the values in decimal format. For example, Addis Ababa is Latitude: 9.033140, Longitude: 38.750080.

Restrict Locations in the List

By default, all possible locations are listed in the drop-down menu. To restrict the list to locations that have not yet been matched to your data, select the **Show only unmatched locations in drop-down list** option in the bottom left corner of the Edit Locations dialog box.

See also:

Assign Geographic Roles on page 1834

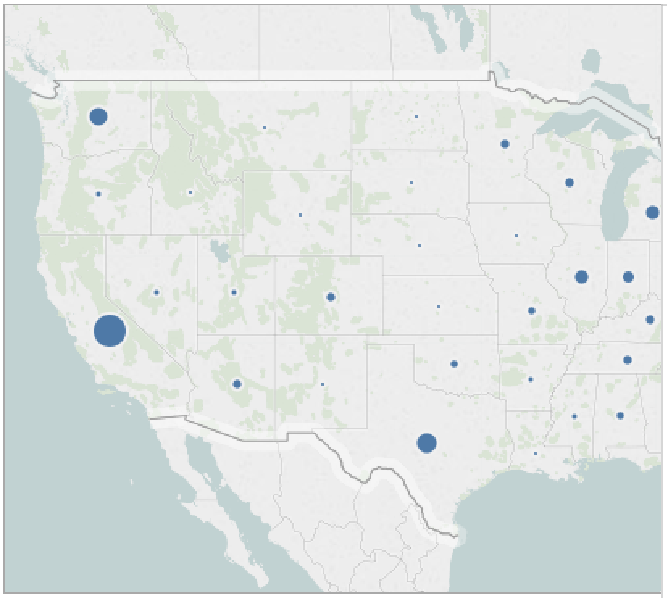
Location Data that Tableau Supports for Building Map Views on page 1795

Build Maps in Tableau

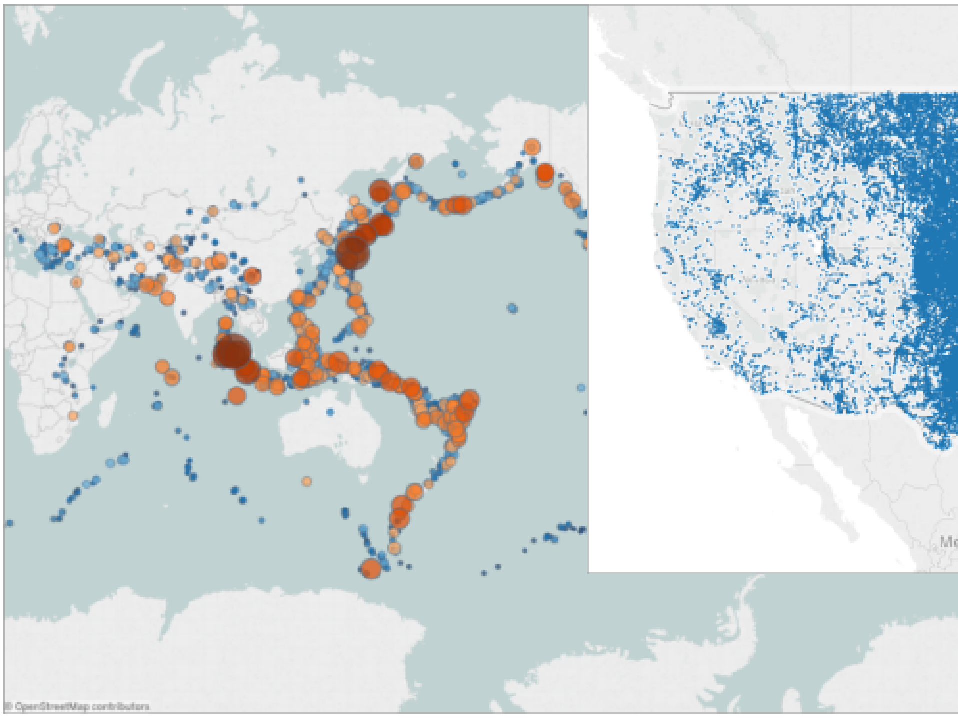
What type of map do you want to create?

Simple Map

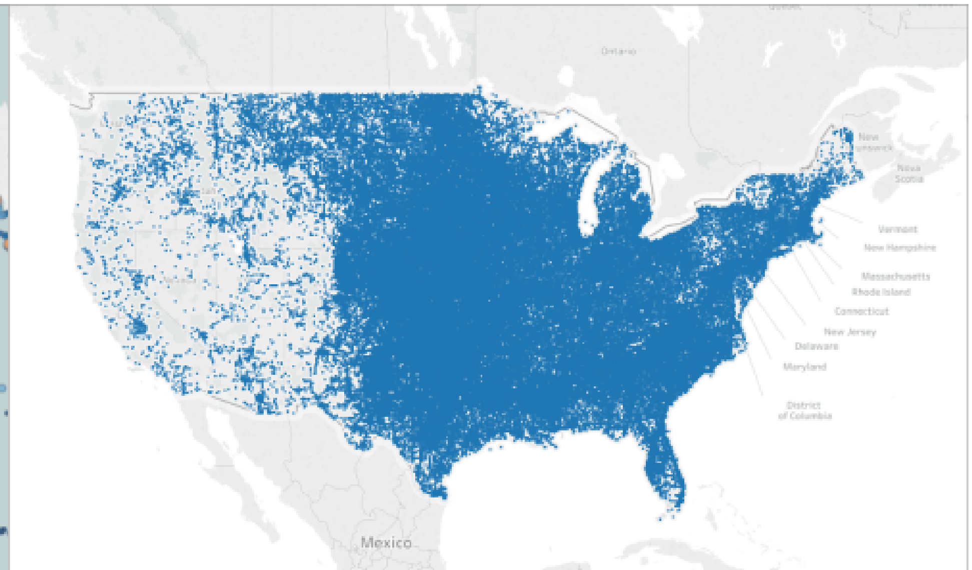
Map from Spatial Files



Proportional Symbol Map

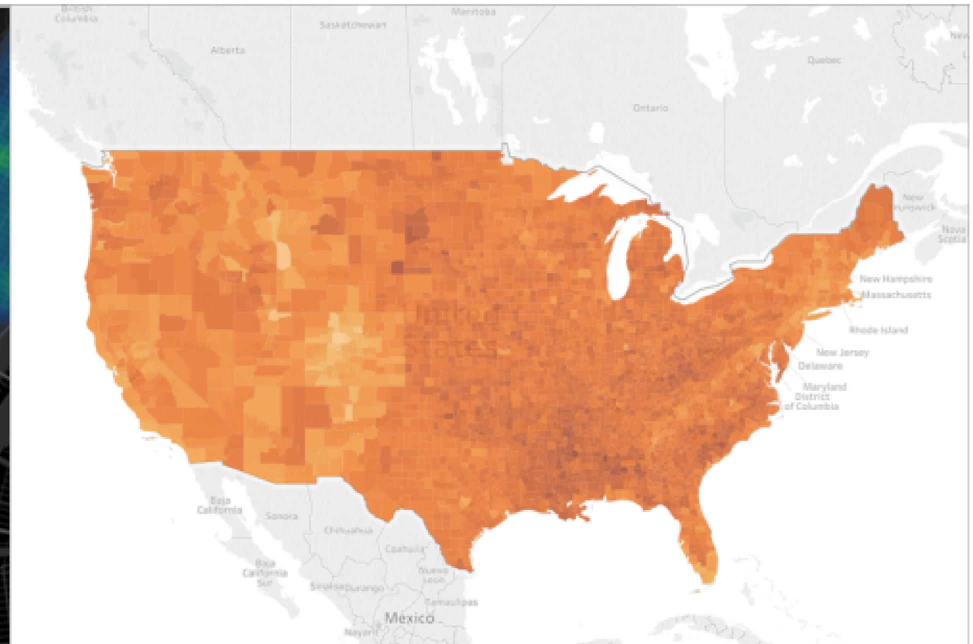


Point Distribution Map

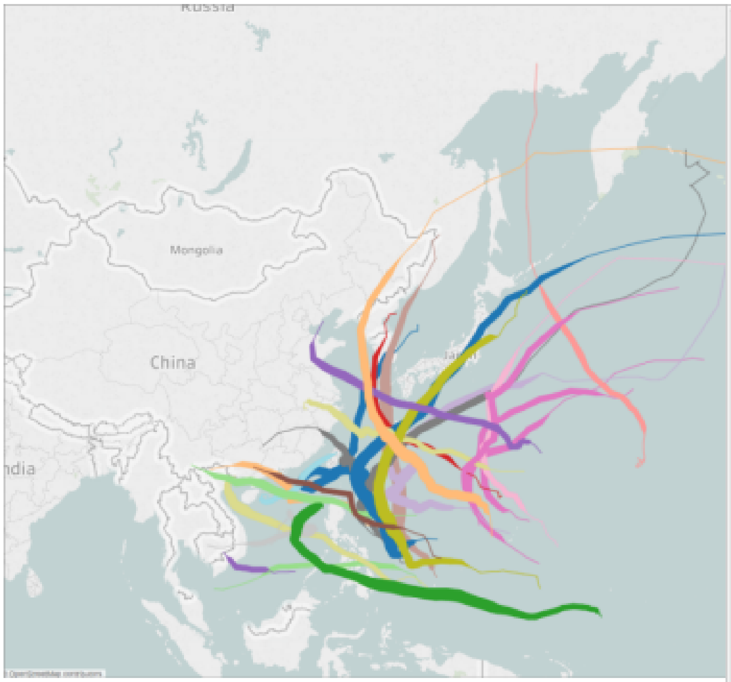


Heatmap (Density Map)

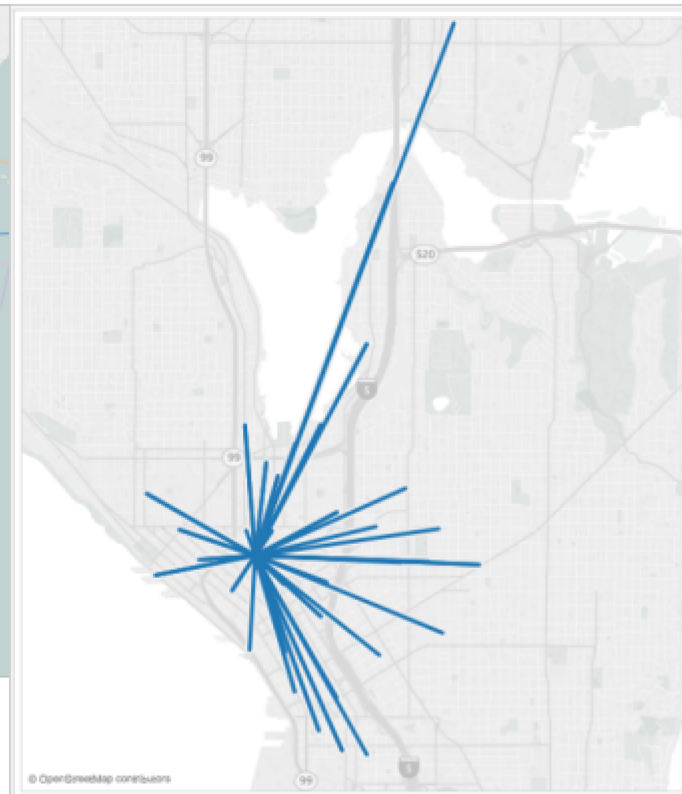
Filled (Choropleth) Map



Flow (Path) Map

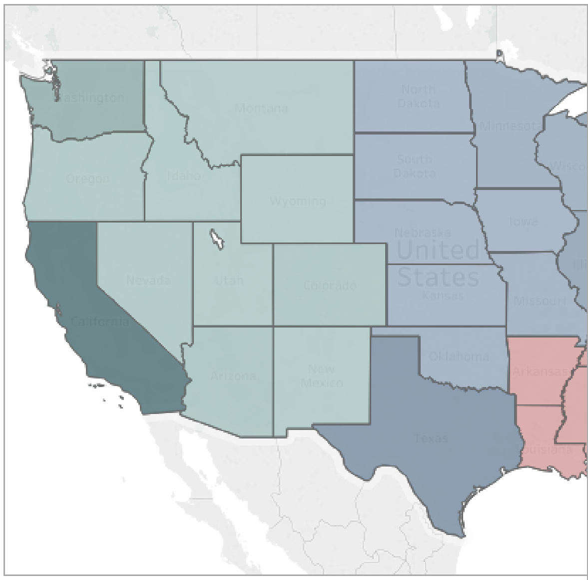


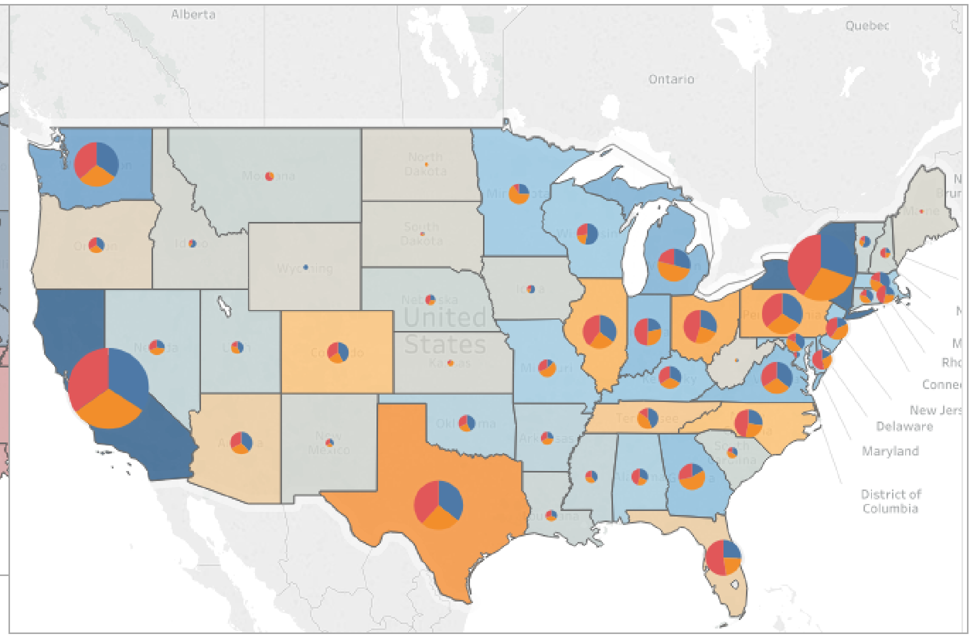
Origin-Destination Map



Dual Axis (Layered) Map

Filled Map with Pie Charts

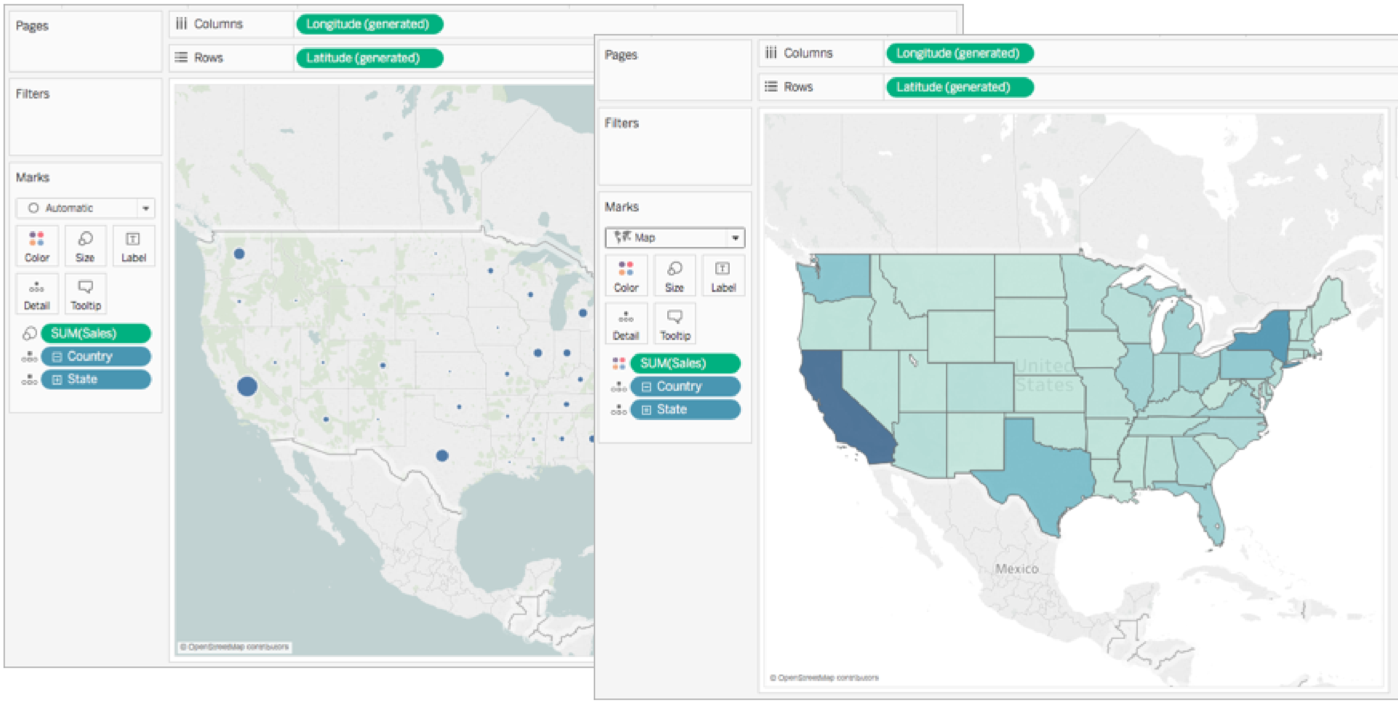




Build a Simple Map

You can build several different types of maps for your geographic analysis in Tableau. If you're new to maps, or simply want to take advantage of the built in mapping capabilities that Tableau provides, you can create a simple point or filled (polygon) map similar to the examples below.

Prerequisites: To build a simple map, your data source must contain location data (location names, or latitude and longitude coordinates). If your data source does not contain location data, see the Map Data section for ways you can connect to location data.

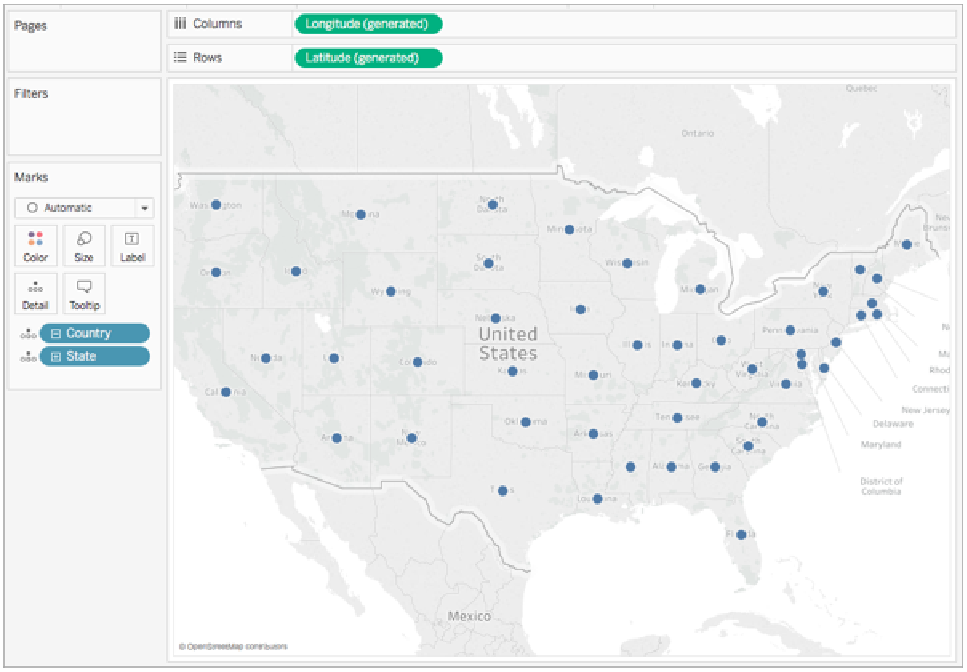


This topic illustrates how to create a simple map using an example. To follow along with the example below, open Tableau Desktop and connect to the **Sample-Superstore** data source, which comes with Tableau.

Build a simple point map

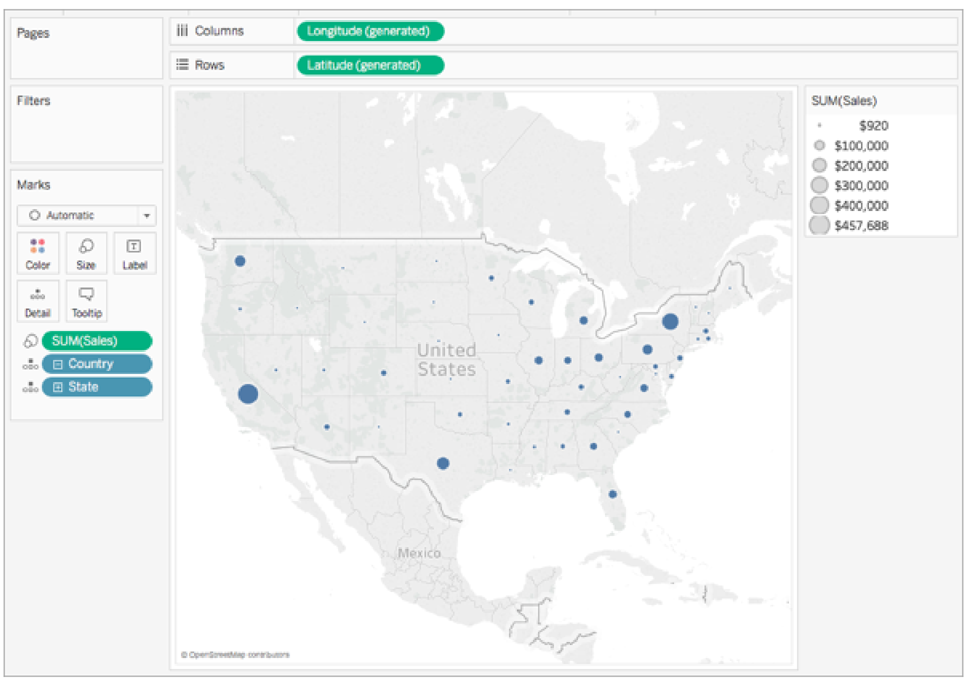
1. Navigate to a worksheet.
2. In the **Data** pane, under Dimensions, double-click **State**.

   A map view is automatically created because the State field is a geographic field. To learn more about geographic fields and how to create them, see **Assign a geographic role to a field** on page 1835.



3. From Measures, drag **Sales** to **Size** on the Marks card.

   The data points on the map update to show the amount of sales proportionally.

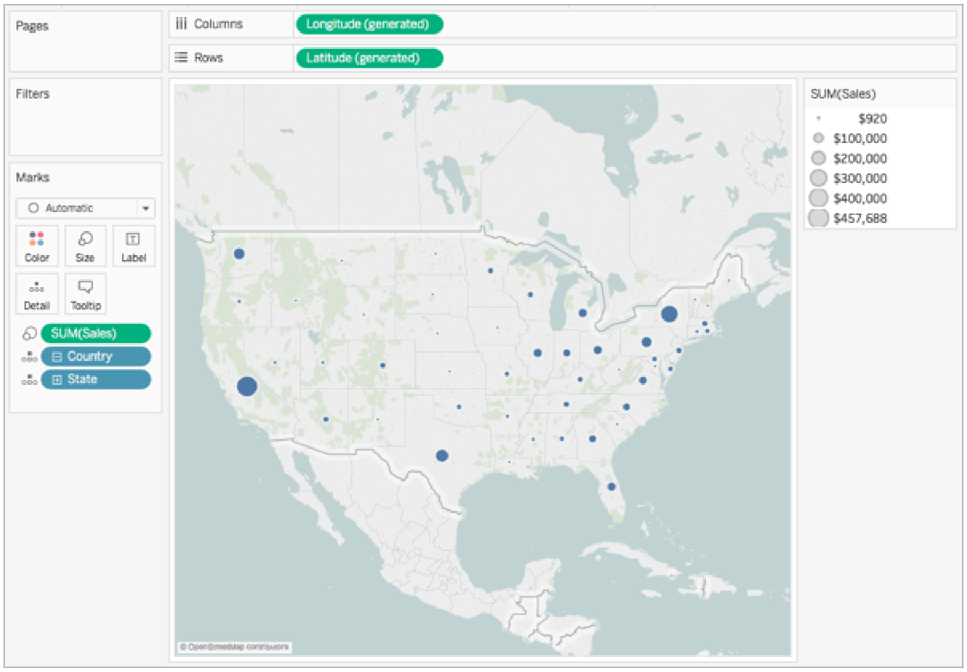


4. Select **Maps** > **Map Layers**.

5. In the Map Layers pane, do the following:

    - Click the Style drop-down and select **Normal**.
    - Under Map Layers, clear **Country/Region Names**.

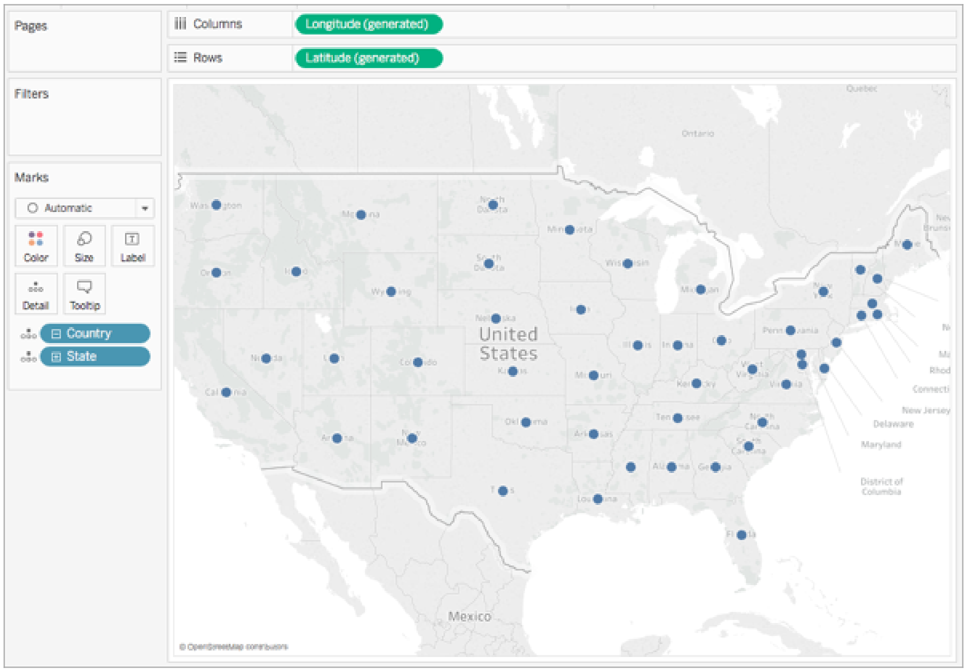
    The background map updates with the new settings.



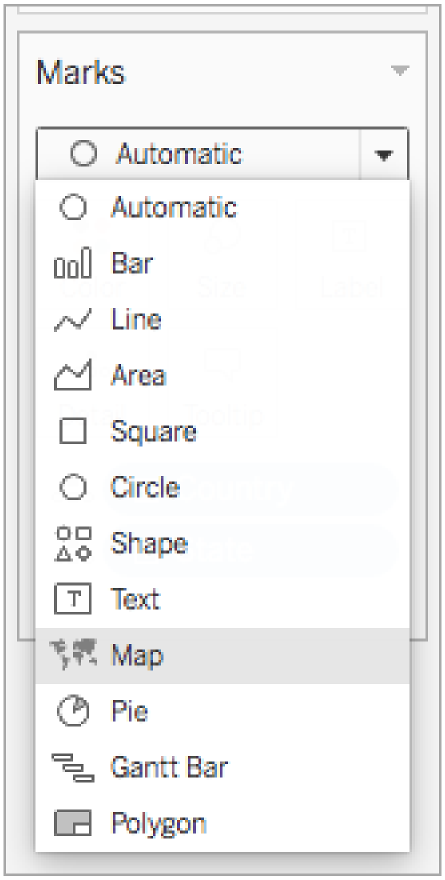
Build a simple filled (polygon) map

1. Navigate to a new worksheet.
2. In the **Data** pane, under Dimensions, double-click **State**.

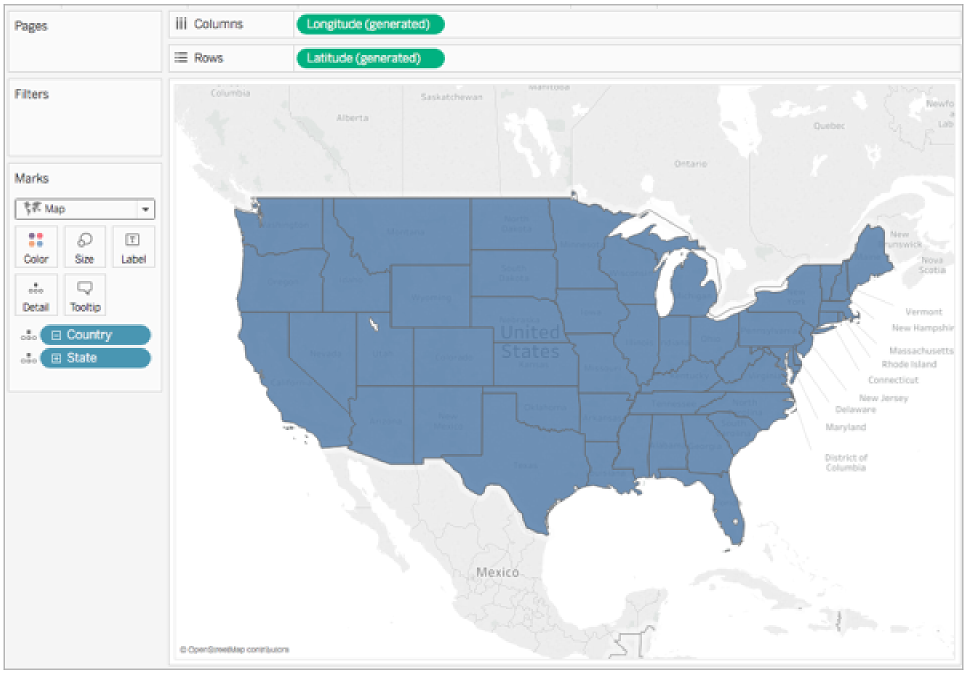
    A map view is automatically created.



3. On the Marks card, click the Mark Type drop-down and select **Map**.

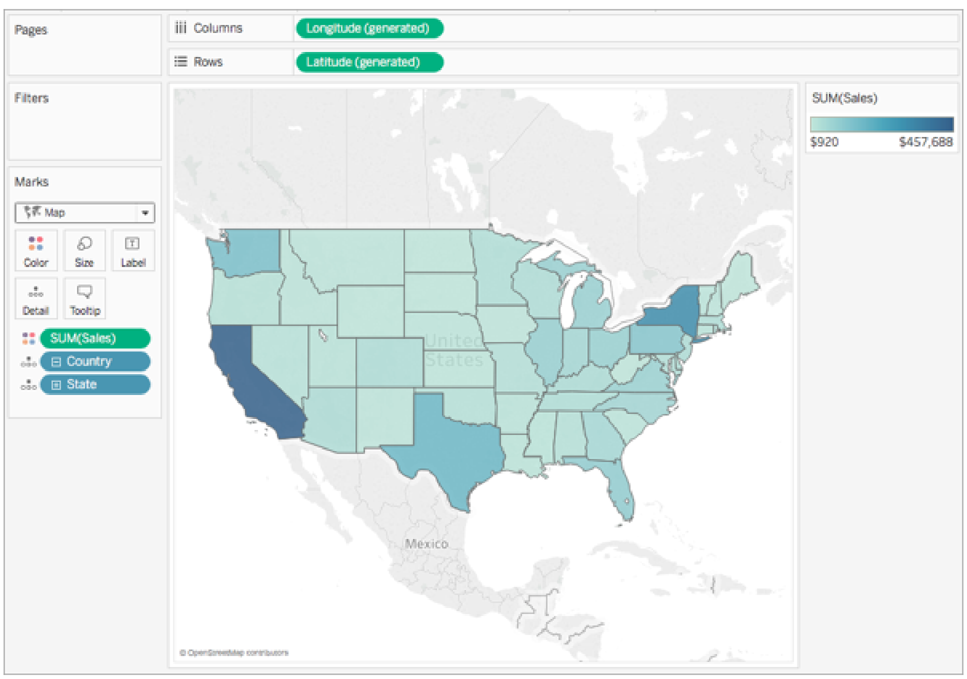


The map view updates to a filled (polygon) map.



4. From Dimensions, drag **Sales** to **Color** on the Marks card.

   The polygons on the map update to show the amount of sales using color.



See also:

Mapping Concepts in Tableau on page 1781

Build Maps in Tableau on page 1843

Customize How Your Map Looks on page 1965

Create Tableau Maps from Spatial Files

In Tableau Desktop, you can connect to the following spatial file types: Shapefiles, MapInfo tables, KML (Keyhole Markup Language) files, GeoJSON files, TopoJSON files, and Esri File Geodatabases. You can then create point, line, or polygon maps using the data in those files.

With a Creator license in Tableau Online or Tableau Server, you can upload spatial file formats that only require one file (KML, GeoJSON, TopoJSON, Esri shapefiles packaged in a `.zip`, and Esri File Geodatabases with the extension `.gdb.zip`) in the Files tab when you create a new workbook and connect to data.

> **Note:**In current versions of Tableau, you can only connect to point geometries, linear geometries, or polygons. You cannot connect to mixed geometry types.

Where to find spatial files

If you do not already have spatial files, you can find them at many open data portals. You can also find them on websites for your city or for a particular organization, if they provide them.

Here are some examples:

- LONDON DATASTORE
- EGIS South Africa
- U.S. Energy Information Administration
- USGS Water Resources
- Geospatial Information Authority of Japan
- Data.gov
- Census.gov

Connect to spatial files

1. In Tableau Desktop: click the New Data Source icon and select **Spatial file**.
   In Tableau Online or Tableau Server (Creator role): Select **Create** > **Workbook**. Select the **Files** tab.

2. Navigate to the folder that contains your spatial data, select the spatial file you want to connect to, and then click **Open**.

Note: To connect to spatial files, you must include all of the following files in the same directory:

- **For Esri shapefiles**: The folder must contain `.shp`, `.shx`, `.dbf`, and `.prj` files as well as `.zip` files. When connecting to Esri shapefiles in Tableau Online or Tableau Server, the file must be packaged in a `.zip`.

  For Esri File Geodatabases: The folder must contain the File Geodatabase's `.gdb` or the `.zip` of the File Geodatabases's `.gdb.`. When connecting to Esri shapefiles in Tableau Online or Tableau Server, the Esri File Geodatabase must be packaged with the extension `.gdb.zip`.
- **For MapInfo tables** (Tableau Desktop only): The folder must contain `.TAB`, `.DAT`, `.MAP`, and `.ID` or `.MID` and `.MIF` files.
- **For KML files**: The folder must contain the `.kml` file. (No other files are required.)
- **For GeoJSON files**: The folder must contain the `.geojson` file.(No other files are required.)
- **For TopoJSON files**: The folder must contain the `.json` or `.topojson` file. (No other files are required.)

How Tableau interprets your spatial data

As soon as you connect to your spatial data, Tableau reads the spatial reference information of the data set and transforms the data into latitude and longitude coordinates. All data, regardless of the spatial reference system, is transformed to **WGS84 (EPSG:4326)**.

> **Note**: If your data does not display diacritics (accent marks on characters) properly, check to make sure the file is **UTF-8** encoded.

The Geometry field

When you connect to spatial data, Tableau creates a Geometry field for your point geometries or your polygons. You use the Geometry field to create a map with your spatial data.

The values in this field display the geometry primitive—for example, **Point** for point geometries, **Linestring** or **Multilinestring** for linear geometries, and **Polygon** or **Multipolygon** for polygons. But this is simply an alias for the underlying spatial data.

For more information on how to build a map with the Geometry field, see the **Build a map view from spatial data** below section.

Join spatial data

Sometimes your spatial data includes only the geometry information, and does not include any demographic or other information. In this case, you can join a spatial data source with another data source type, or even another spatial file that includes the additional data you need for your analysis, as long as the files have a column (or field) in common.

For example, you can join a KML file that contains custom geographies for school districts in Oregon, U.S. with an Excel spreadsheet that contains demographic information about those school districts.

For more information on the different ways to join spatial data, see **Join Spatial Files in Tableau** on page 1815.

Build a map view from spatial data

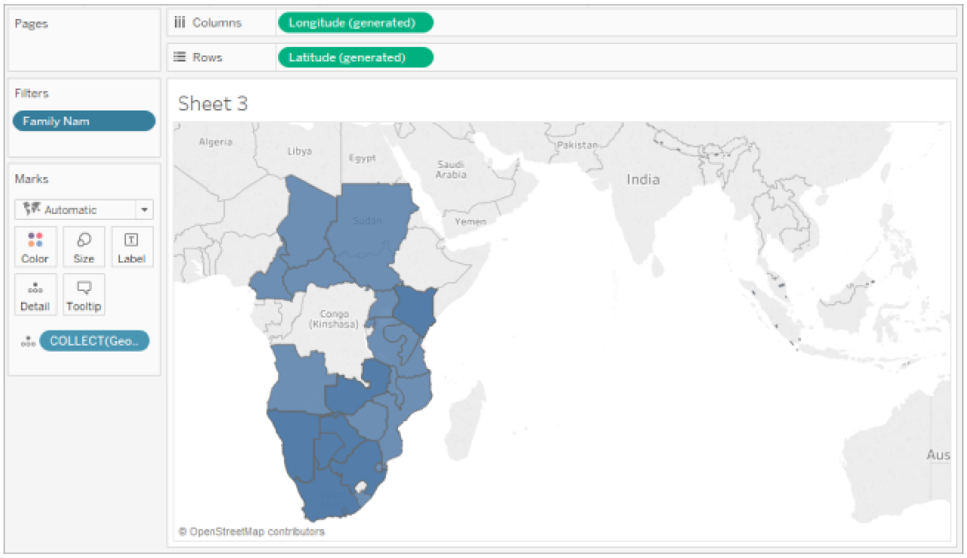
1. Open a new worksheet.
2. In the Data pane, under Measures, double-click the **Geometry** field.

   The Geometry field is added to Detail on the Marks card, and the Latitude (generated) and Longitude (generated) fields are added to the Columns and Rows shelves. A map view is created.

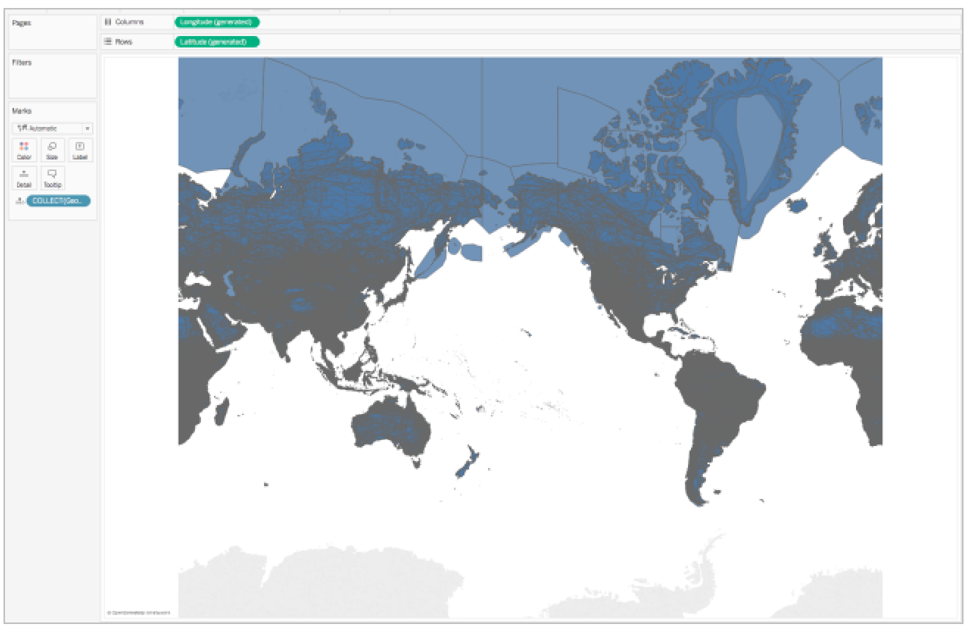
Filter geometries to improve view performance

When working with large spatial data sets, your view might take a long time to render. If this is the case, you can filter the number of geometries being added to the view using another dimension in your data source. For more information about filtering data, see **Filter Data from Your Views** on page 1411.

For example, in the image below, the view has been filtered down to a small subset of polygons using a dimension (Family Nam). The data source, from the IUCN List of Threatened Species, contains data on endangered mammals around the world. Therefore, the dimension Family Nam contains a list of mammal family names. This view has been narrowed down to one family name: rhinoceroses. Polygons for only rhinoceroses are shown in the view.



Without the filter, the polygons for every mammal in the data source are shown around the world, and the view takes a long time to render every time you perform an action, such as select a mark in the view.



Add levels of detail to the view

The Geometry field is a measure, and by default, is aggregated into a single mark using the COLLECT aggregation when it is added to the view. All of your polygons or marks will be in the view, but they will operate as a single mark. You therefore need to :

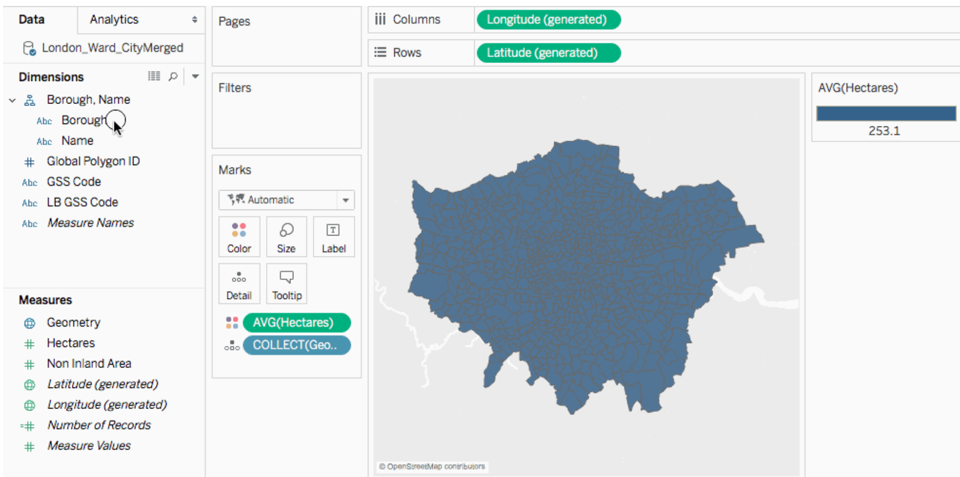
- Add additional levels of detail to the view to break it up into separate marks (based on the level of detail you specify)

  or

- Dissaggregate the data all together so every single mark (polygon or data point) is separate.

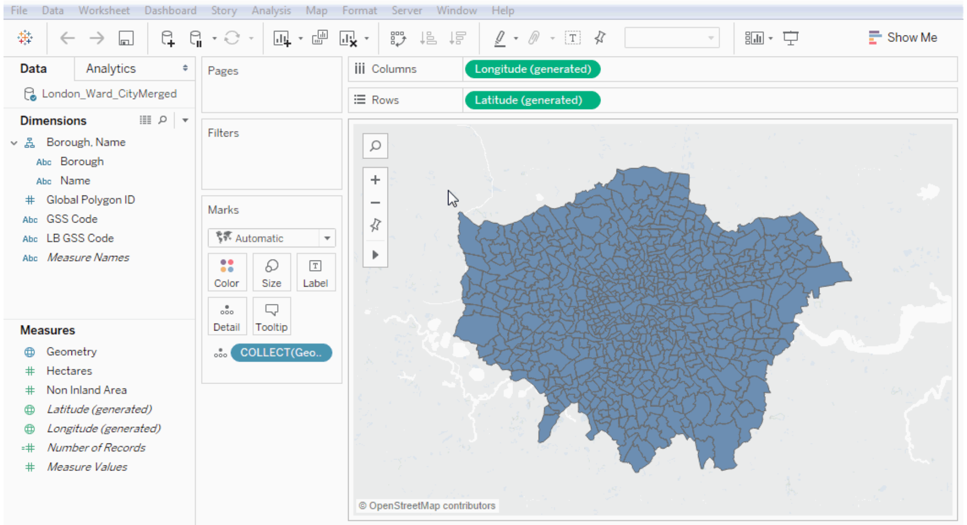
To add additional levels of detail to the view:

- From Dimensions, drag one or more fields to **Detail** on the Marks card.



To disaggregate the data:

- Click **Analysis**, and then clear **Aggregate Measures**.



Customize the appearance of geometries

You can customize the appearance of points, polygons, and lines by adding color, hiding polygon lines, specifying which polygons or data points appear on top, and adjusting the size of your data points.

Add color

To add color to your data points or polygons, drag a dimension or measure to **Color** on the Marks card.

For example, in the images above, the dimension (Presence), is placed on Color to represent the presence of an animal in a particular area.

Hide polygon lines

By default, polygon lines are shown when you create a polygon map from spatial data. If you want a cleaner view, you can remove them.

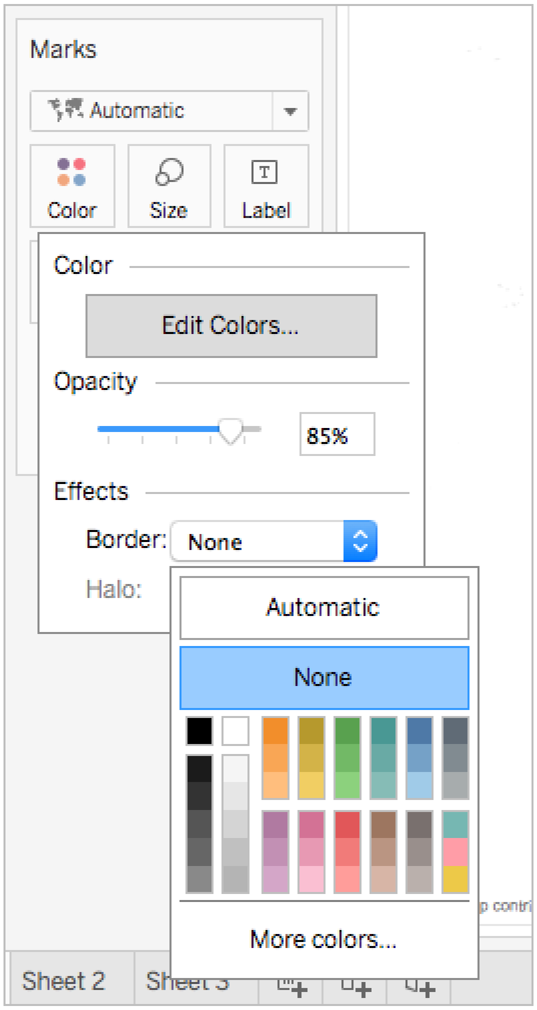
Take the following images, for example. The first image shows polygon lines. The second image does not show polygon lines.

Polygon lines shown by default.

Polygon lines removed.

To hide polygon lines:

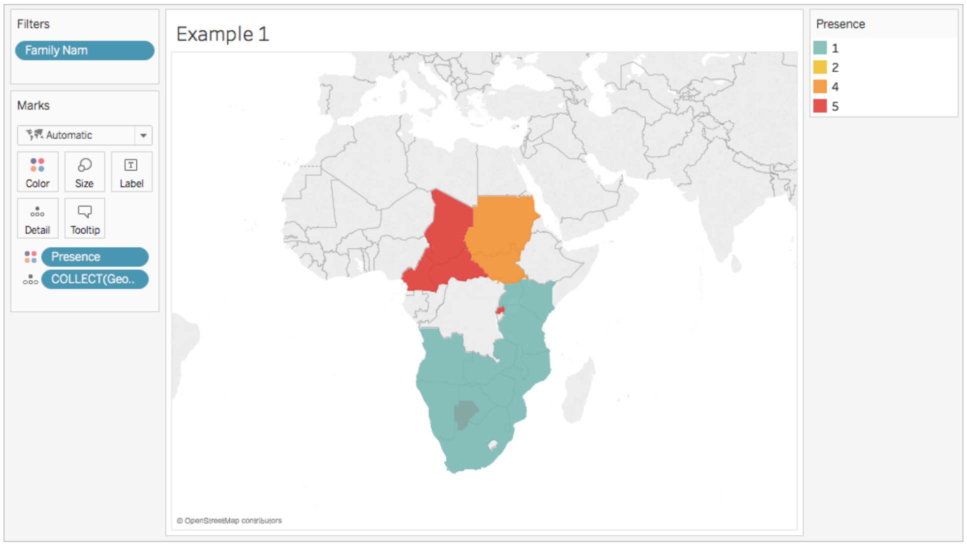
1. On the Marks card, click **Color**.
2. Under Effects, select the **Border** drop-down, and then click **None**.



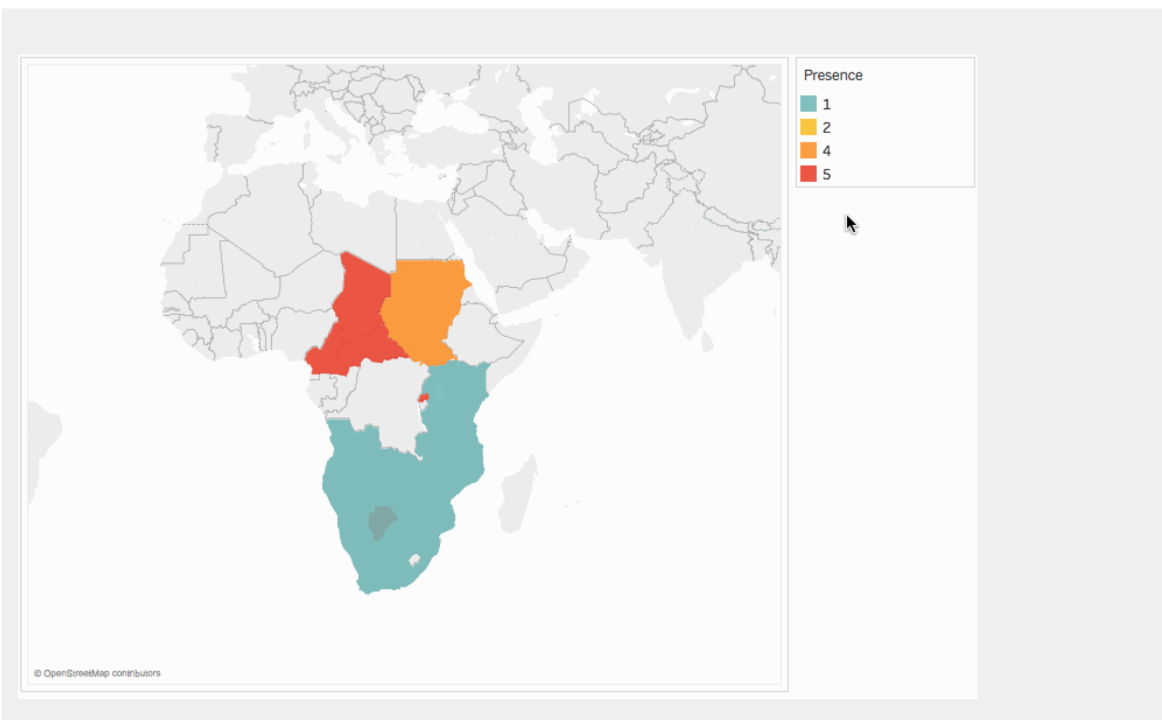
Specify which polygons or data points appear on top

Your polygons or data points might overlap or cover each other. You can specify which polygons or data points appear on top if you have a color or size legend in the view.

For example, in the image below, notice that there is a smaller polygon hidden behind the larger teal polygon in southern Africa.



You can rearrange the items in your legend to control which data points or polygons appear on top. To do so, in the legend, select the item you want to be on top, and then drag it to the top of the list.



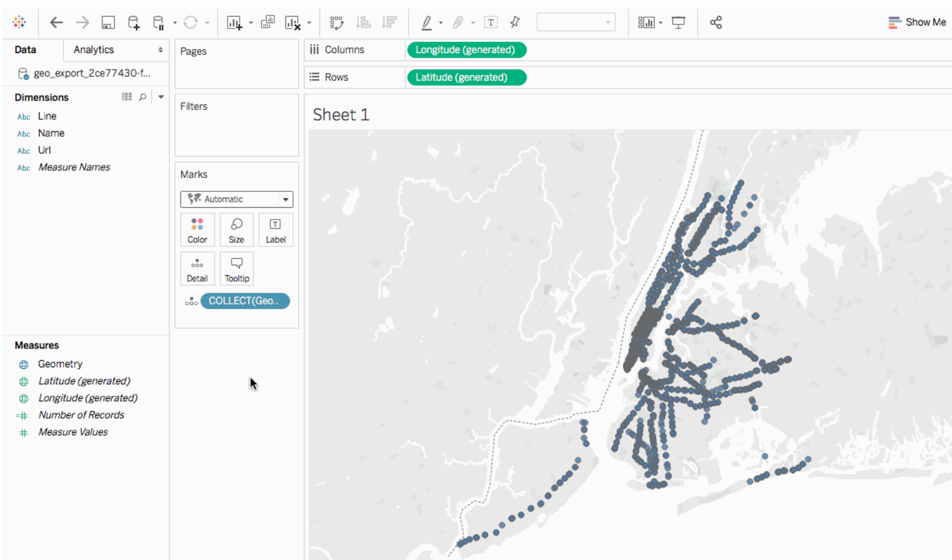
Adjust the size of data points

If you're using point geometries, you can adjust the size of the points on the map view. This is useful if you want to proportion your data points by quantitative values, such as by average sales, or profit.

To adjust the size of your data points:

1. From the **Data** pane, drag a measure to **Size** on the Marks card.
2. On the Marks card, click the Mark Type drop-down and select **Circle**.
3. Optional: From the **Data** pane, drag one or more dimensions to **Detail** on the Marks card to add more data points to your view.

   > **Note:** The level of detail determines which data points are sized. Add additional dimensions to Detail on the Marks card to add levels of detail (more data points), otherwise you might end up with one large data point.

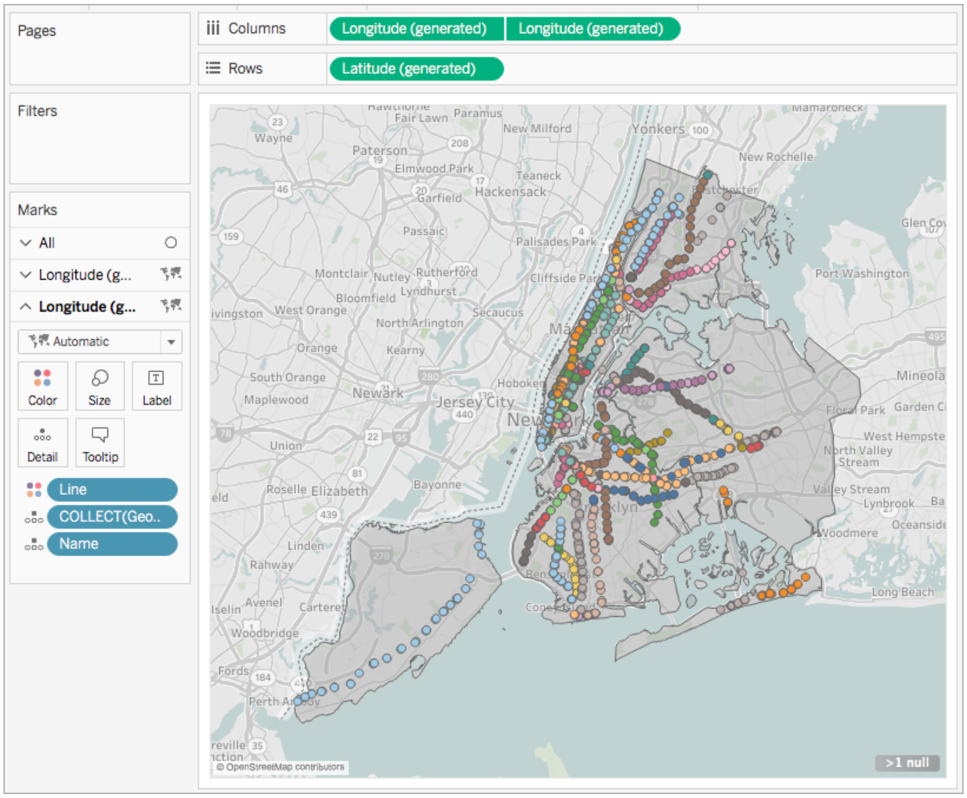


For more information about how to add levels of detail to the view, see the **Add levels of detail to the view** on page 1858 section.

Build a dual-axis map from spatial data

If you join a spatial file either with another spatial file, or a different file type, you can create a dual-axis map using the geographic data from those files. This enables you to create more than one layer of your data on a map.

For example, the following is a dual-axis map view that was created using two spatial files. It contains two maps; one map shows the boroughs of New York City as polygons, and the other shows data points for subway entrances around the city. The subway entrance data is layered over top of the city boroughs polygons.



1. In Tableau Desktop, open a new worksheet.
2. Connect to your data sources.
3. Create the first map view.

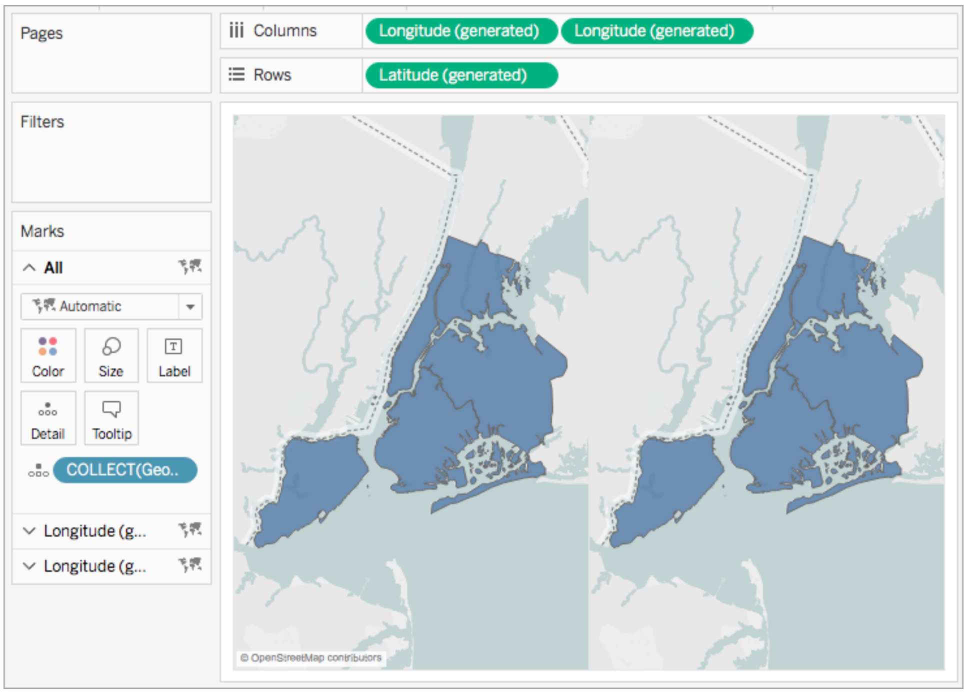
   See **Build a map view from spatial data** on page 1857 above for how to build a map view from spatial files.

4. On the Columns shelf, control-drag (command-drag on a Mac) the **Longitude** field to copy it, and place it to the right of the first **Longitude** field.

   > **Important**: This example uses the Latitude (generated) and Longitude (generated) fields that Tableau creates when you connect to spatial data. If your data source contains its own Latitude and Longitude fields, you can use them instead of the Tableau generated fields, or in combination with the Tableau

generated fields. For more information, see **Create Dual-Axis (Layered) Maps in Tableau** on page 1935.

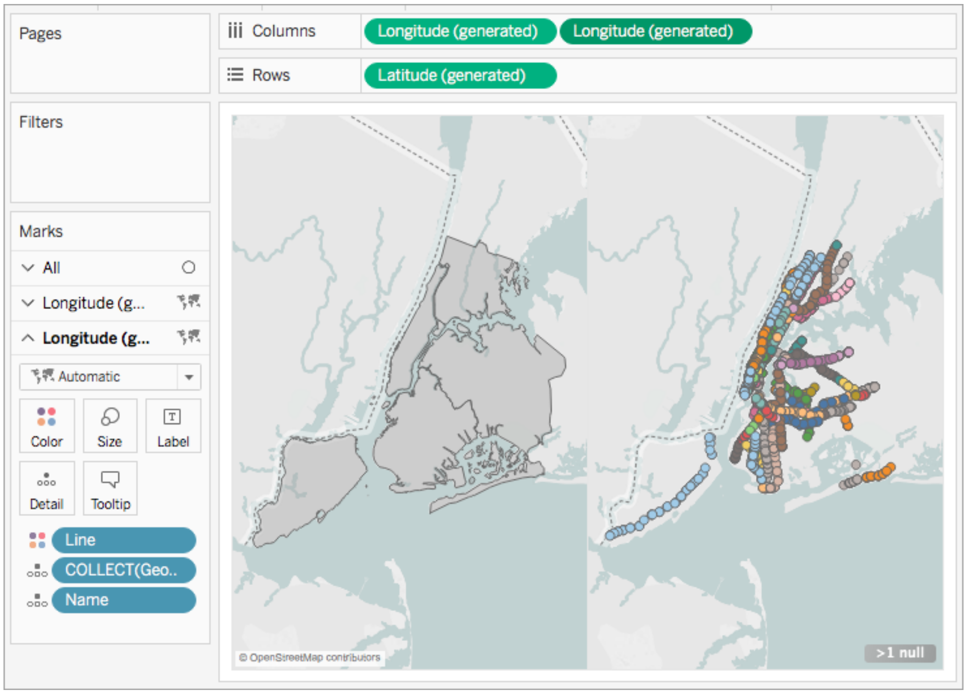
You now have two identical map views. There are now three tabs on the Marks card: one for each map view, and one for both views (All). You can use these to control the visual detail of the map views. The top Longitude tab corresponds to the map on the left of the view, and the bottom Longitude tab corresponds to the map on the right of the view.



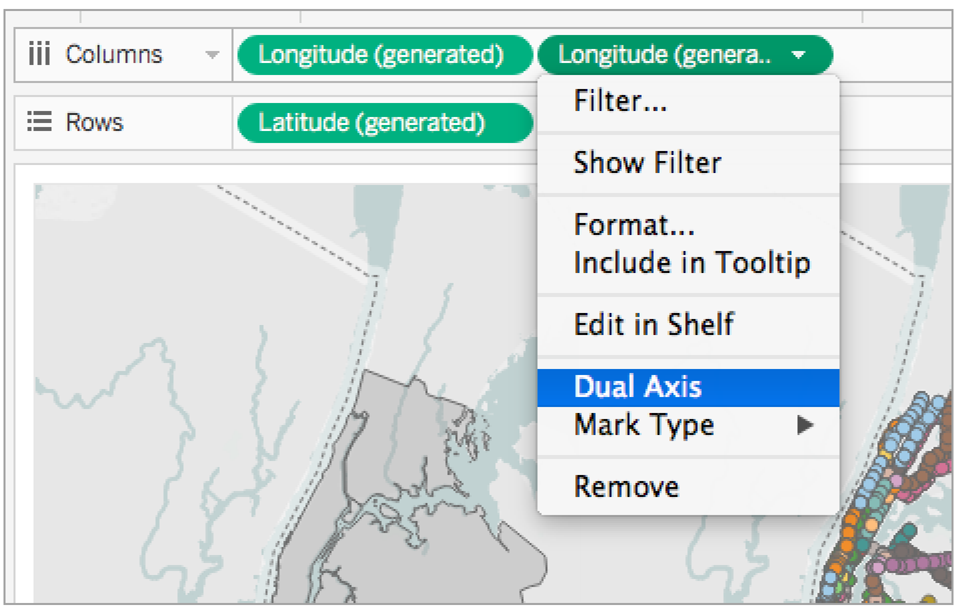
5. On the Marks card, click one of the **Longitude** tabs, and then remove all fields on that tab.

   One of your map views is now blank.

6. Create the second map view by dragging the appropriate fields from the Data pane to the blank **Longitude** tab on the Marks card.



7. When your two map views are finished, on the Columns shelf, right-click the **Longitude** field on the right and select **Dual Axis**.



Your map data is now layered on one map view.

To change which data appears on top, on the Columns shelf, drag the **Longitude** field on the right, and place it in front of the **Longitude** field on the left.

See also

Spatial File on page 1796

Tackle your geospatial analysis with ease in Tableau 10.2 (Tableau blog post)

Join Spatial Files in Tableau on page 1815

Mapping Concepts in Tableau on page 1781

Create Dual-Axis (Layered) Maps in Tableau on page 1935

Create Geographic Layers for Maps

When you need to add multiple layers of geographic data to a map, use marks layers in the Marks card. Marks layers act like transparent overlays. Each layer functions independently and can have its own mark type, captions, and color.

Note:

- Marks layers can be used only with geographic data.
- All geographic data must come from the data source. Layers from a secondary data source are not supported.
- Fields can be shared between marks layers, but any filters that you apply to a field will apply to all marks layers

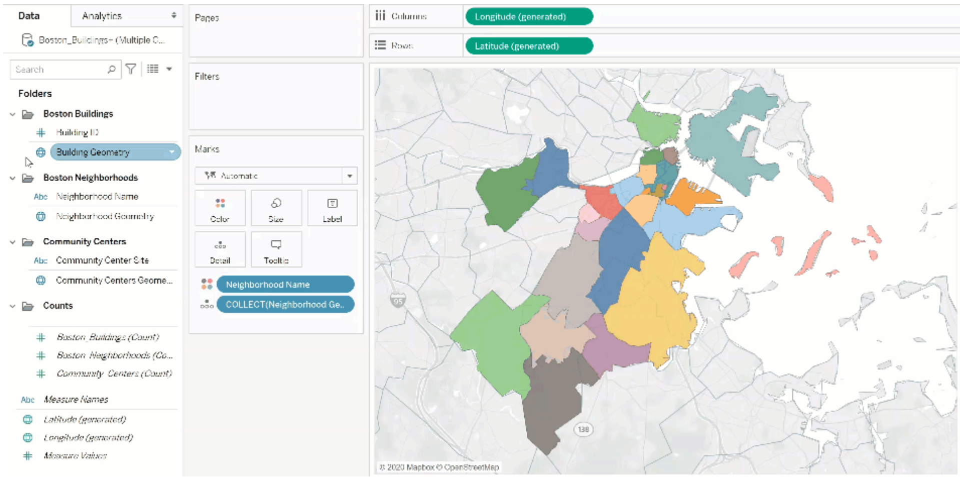
Build a view using marks layers

As you build a map, options for creating marks layers become available when you add more geographic fields to the view.

Add a marks layer

1. Build your first map in the view. Think of this as the base layer.
2. Drag a geographic field into the view. Notice the Add a Marks Layer control becomes available in the top left corner of the view.
3. Drop the geographic field on the Add a Marks Layer control. A new marks layer is added

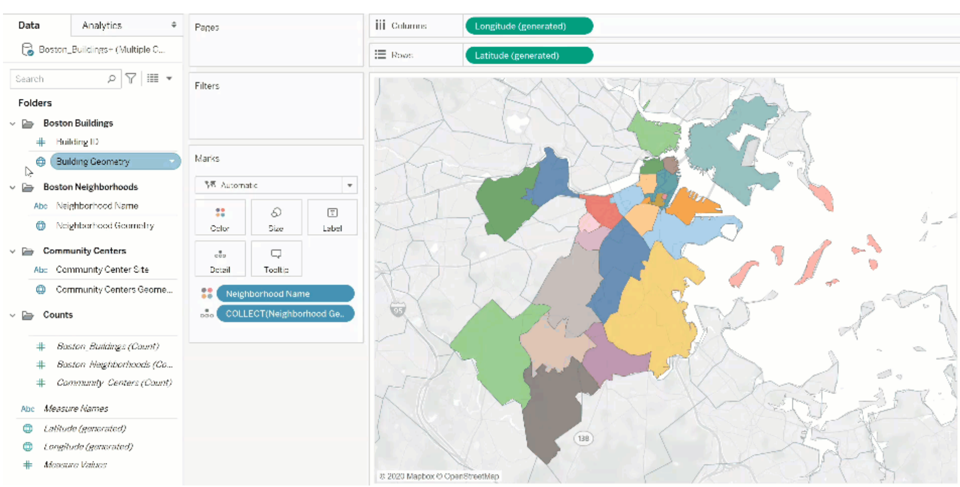
to the Marks card and the layer is displayed in the view.



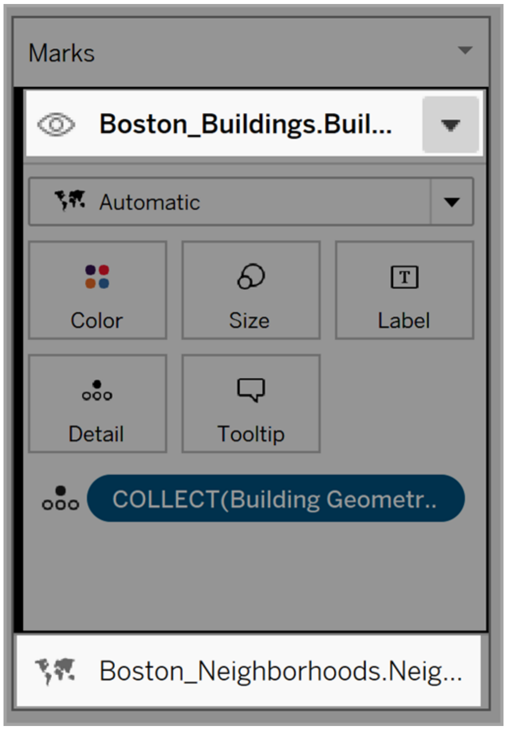
Example: Working with mark layers

The following walk through uses the sample workbook Marks Layers Example.

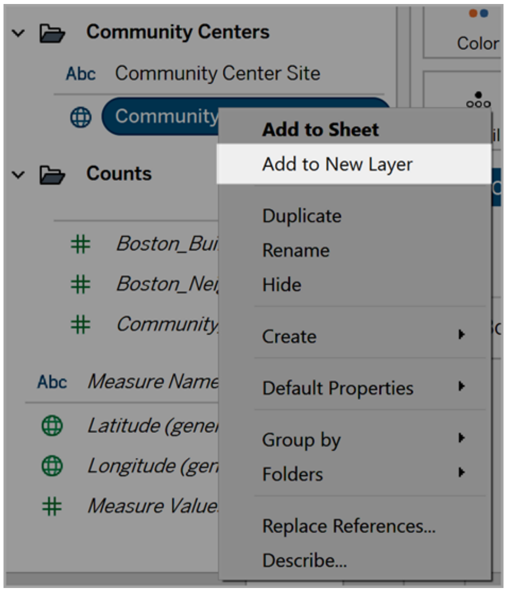
1. Open the Marks Layers Example workbook. It uses a connection to a geographic data source.
2. Drag the Neighborhood Geometry field to the view. This creates a map in the view.
3. Drag the Neighborhood Names field to Color on Marks card.
4. Drag the Building Geometry field to the view. The Add a Marks Layer drop target displays in the top left corner of the view.



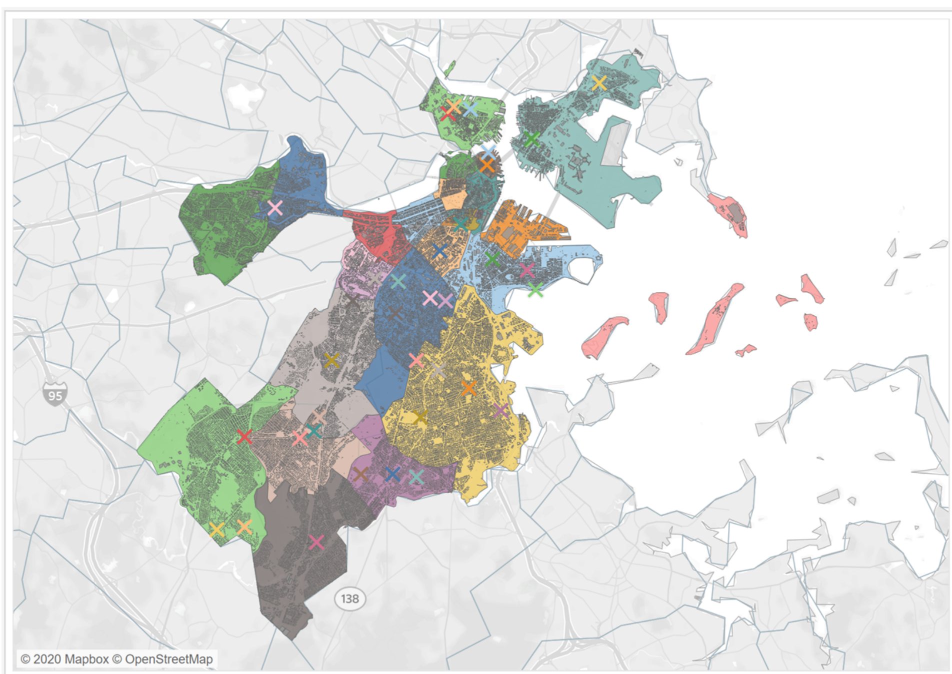
Notice that the Marks card now includes two layers labeled Boston_Buildings.Building Geometry and Boston_Neighborhoods.Neighborhood Geometry.



5. Add another layer of data to the view. Drag Community Centers Geometry to the view. Note: As a shortcut, you can right click a geographic field and click Add to New Layer in the drop-down menu.



6. In the sample workbook, click the new layer to open its Marks options, and then drag the Community Centers Name field to Color.
7. To change the mark type, click the mark type drop-down arrow, and then select Shape. Click Shape field on the Marks card for this layer, and then select the heavy 'X'.



Each layer's mark type, color, and formatting can be configured independently. To change the look of any layer, click that layer to expand its Marks card, and then drag a geographic field to one of the Marks card properties such as Color.

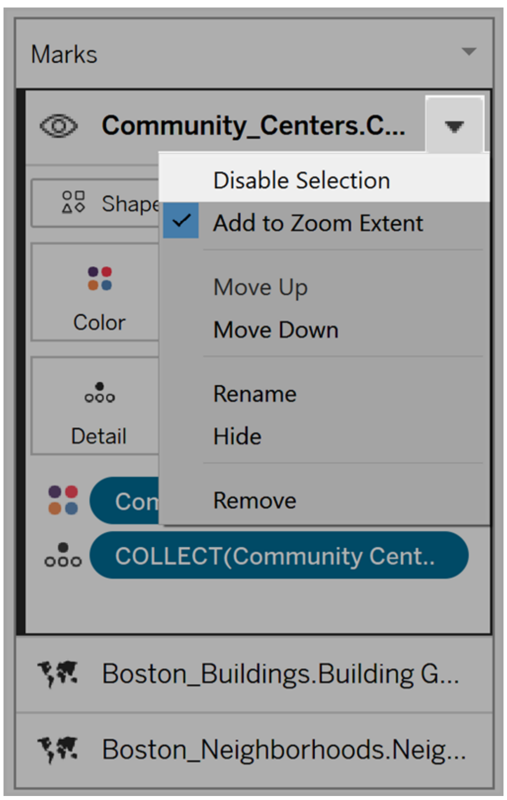
Control mark selection in layers

After you add layers to the view, you may want to control how the user interacts with the layers. With one layer, it is easy to select a single mark to get extra information, such as a caption or text label. However, with multiple layers, it might not be the best user experience to allow your audience to select all marks on all layers. You can control how your users interact with layers in the marks layers settings.

There are two ways to control how a user selects layers in the view.

Disable mark selection

In the Marks card, click the drop-down arrow for a layer, and then select Disable Selection. This prevents users from selecting marks from that layer in the view.



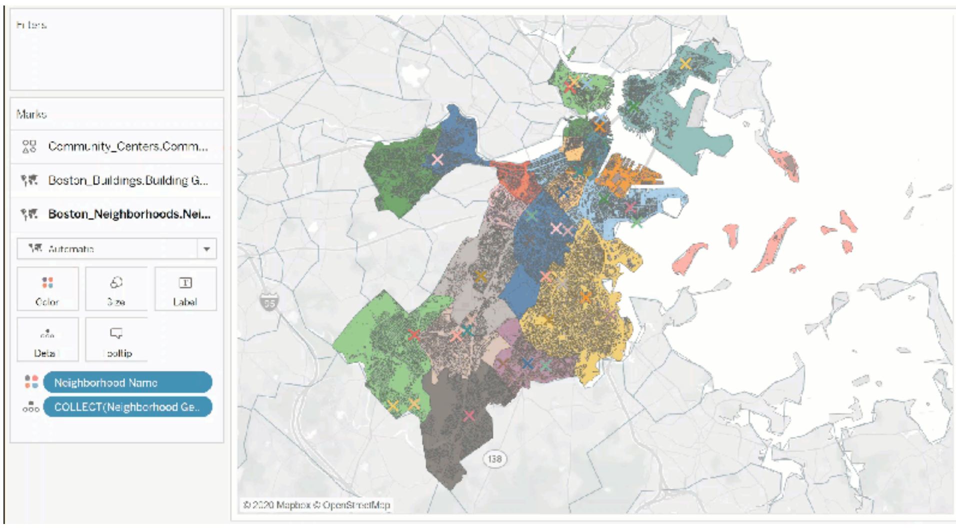
Hide a layer

To hide all marks on a layer

1. In the Marks card, click the Control Layer Visibility icon to the left of the layer name.

   The icon becomes visible when you select the layer in the Marks card, and stays visible when layer visibility is disabled.

2. To show a layer, click the icon again.



Change layer settings

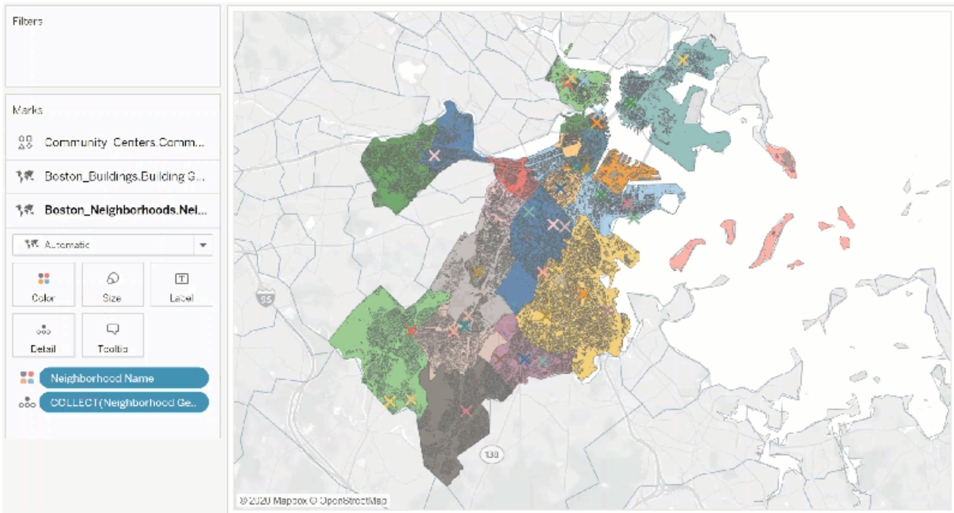
After you create a map with multiple layers, you can change layer order, rename layers, remove layers, or hide layers.

Change layer order

Tableau renders the layers based on their order in the Marks card. Marks in layers at the top of the list will be displayed over marks in layers at that are lower in the list. To change the layer order

1. In the Marks card, click a layer name and drag it until the orange indicator arrow appears. Orange indicates where the layer can be moved to.

2. Drop the layer in the new order location. Note: To cancel the move while you are dragging the layer, drop the layer in a different part of the view (wherever the orange indicator arrow isn't displayed).

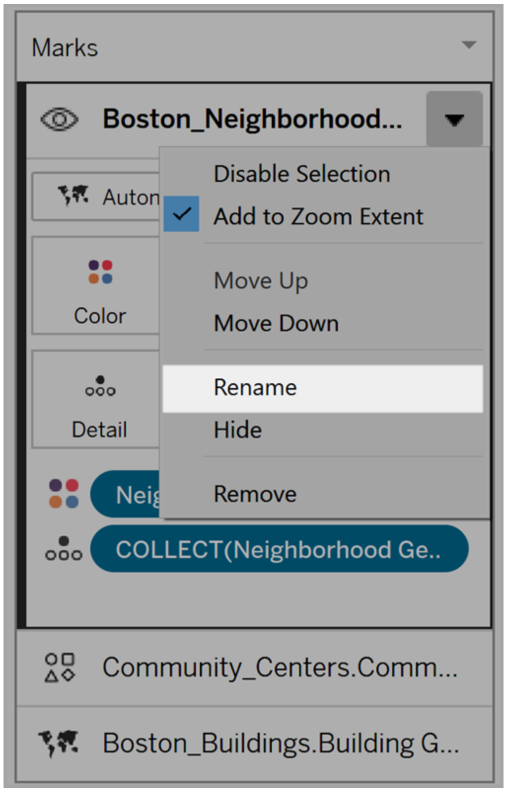


Rename layers

When a new layer is created, Tableau assigns a name automatically based on the table name and field name. With large data sources, it can become difficult to manage many different layers. You can make this easier by renaming the layers and giving each layer a distinct name.
To rename a layer

1. In the Marks card, click a layer, and then click the drop-down arrow to the right of the layer name.

2. Select Rename, type the new name, and then press Enter.

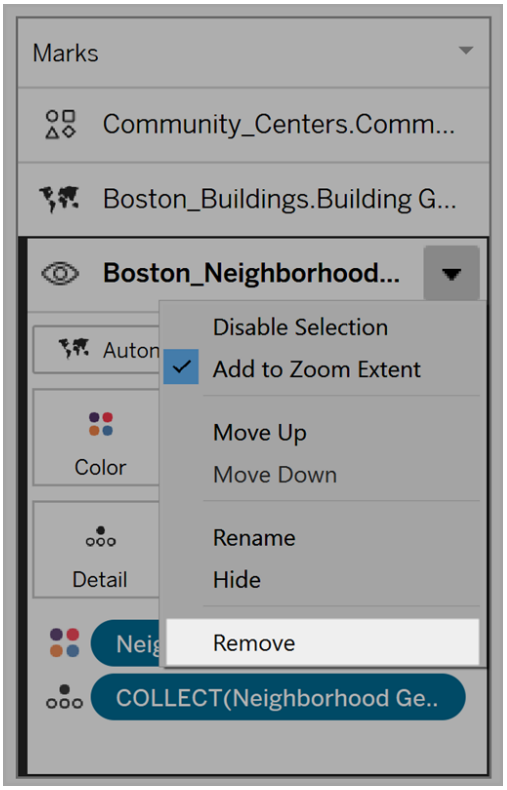


Remove layers

If a layer is not useful anymore, you can remove it.

To remove a layer:

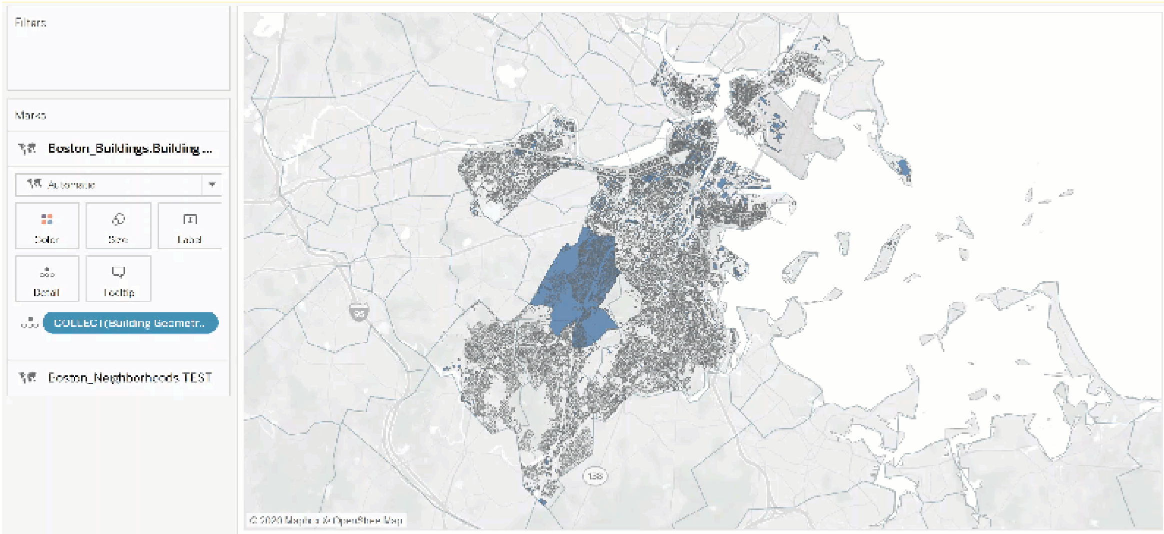
1. In the Marks card, click a layer.
2. Click the drop-down arrow to the right of the layer name, and then select Remove.



Add to Zoom Extent

By default, when you add a new layer to a map, Tableau will include all marks in the view. If there is a large difference between the scale of marks—for example, one layer showing all streetlights in a neighborhood and the other layers showing the whole country's electrical grid—the desired detail can be lost.

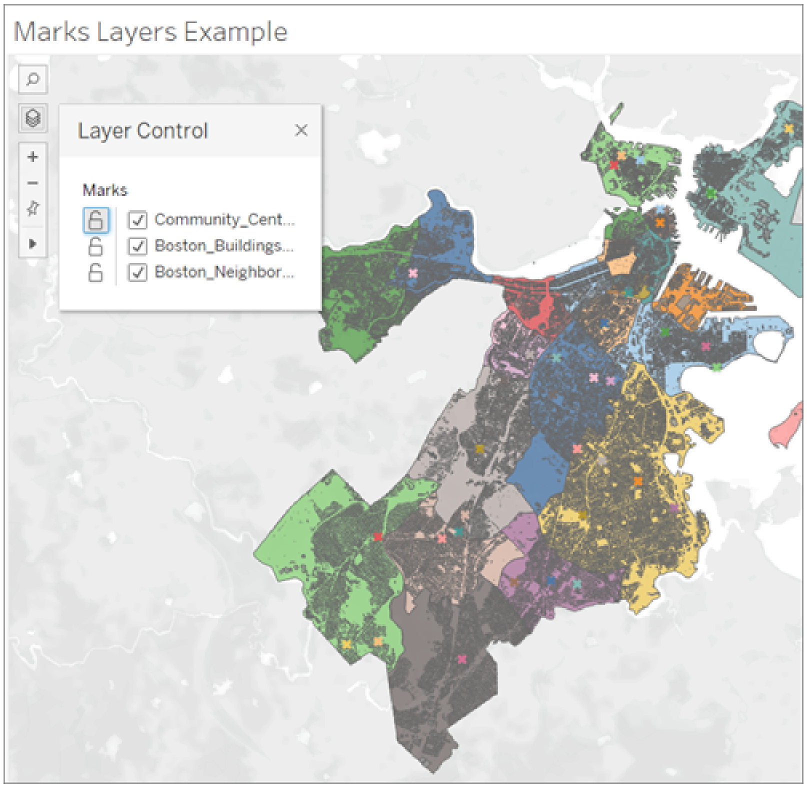
To resolve this, we can disable this feature on a per-layer basis by selecting the drop-down arrow on the layer name then deselect Add to Zoom Extent.



Let users customize views with the Layer Control

With the advent of Geographic Layers for Maps, you have the ability to create dense and information rich views. However, these views may become difficult to navigate as the number of layers and marks increases.

The Layer Control is a panel that expands in the view when selected, displaying all the geographic data layers, and giving you quick access to each marks layer.



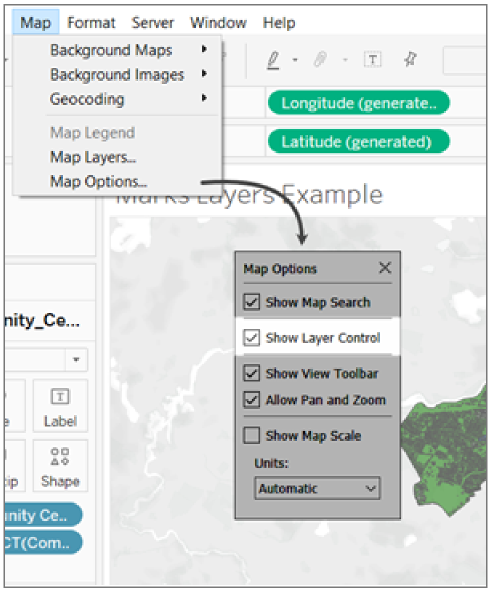
By using the Layer Control, you can hide or disable interactions on each layer, as well as save that selection as a custom view if you are working with a published workbook.

Enable the Layer Control

As the author, you can enable the Layer Control prior to publishing the workbook.

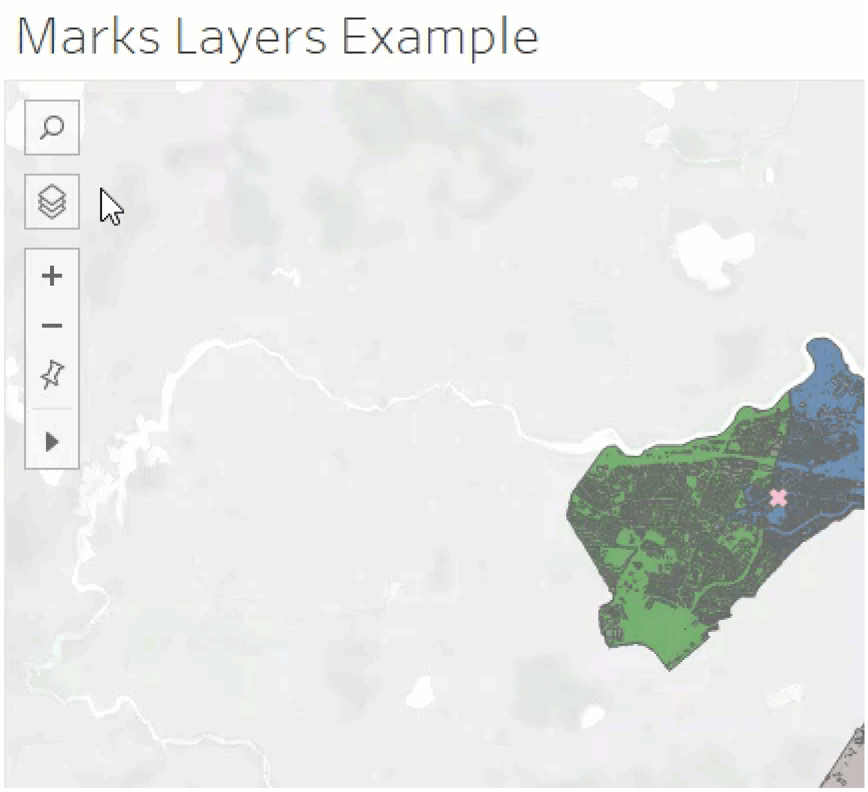
- Select **Maps** > **Map Options** > Check or Uncheck **"Layer Control"**

By default, Layer Control is checked, so when a workbook is published, users will be able to access the panel and hide or show mark layers.



Using the Layer Control

If the Layer Control is enabled, the Layer Control icon will appear in the Map Controls on the upper left hand side of the view. Clicking on the icon will expand a panel where we can see all of the mark layers in the view.

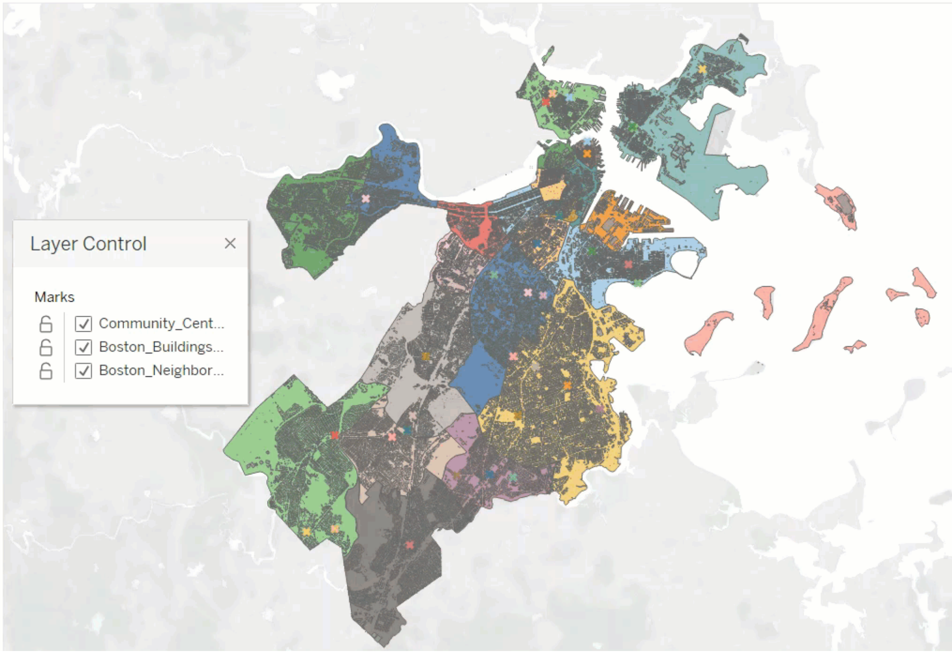


The Layer Control can be dragged around the view freely, but the position will reset each time the view is reloaded.

Each layer in the panel has a pair of icons next to it: the Interaction icon and the Visibility toggle. Any changes you make using the Control will be reflected in the Marks Layers. Hiding a layer using the Control functions exactly the same as hiding a layer in the Marks Layers card.

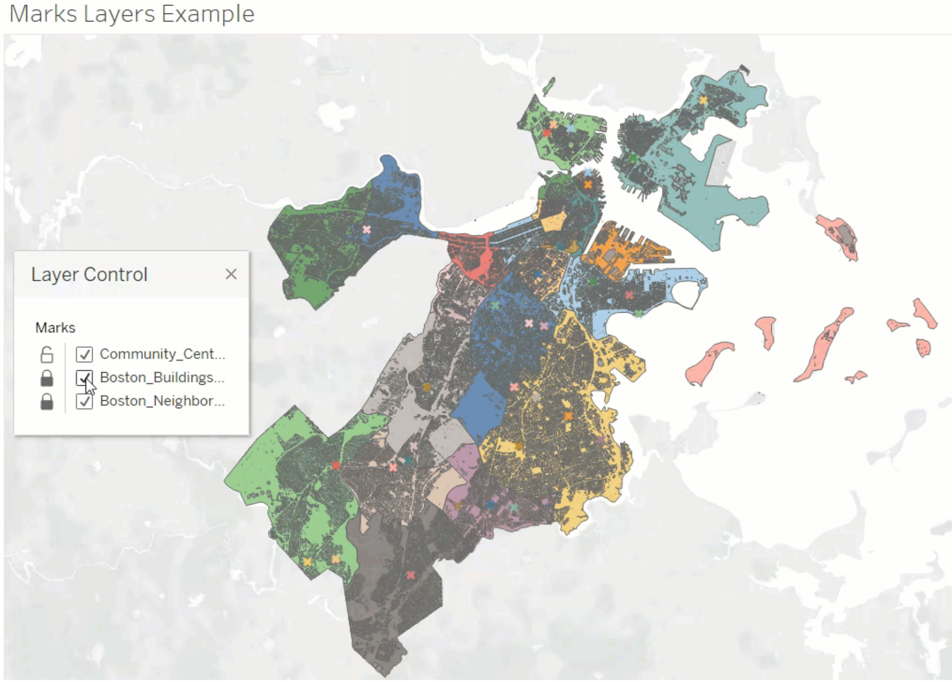
Selecting the Interaction icon will turn off the ability to select or interact with marks on that layer. This can be very useful for keeping background elements from being selected when doing a lasso selection, for example.

Marks Layers Example



Toggling Visibility off will hide all the marks on a layer, while toggling it on will show all the marks in the layer. Note that marks that are hidden are also removed from any selections.

Marks Layers Example



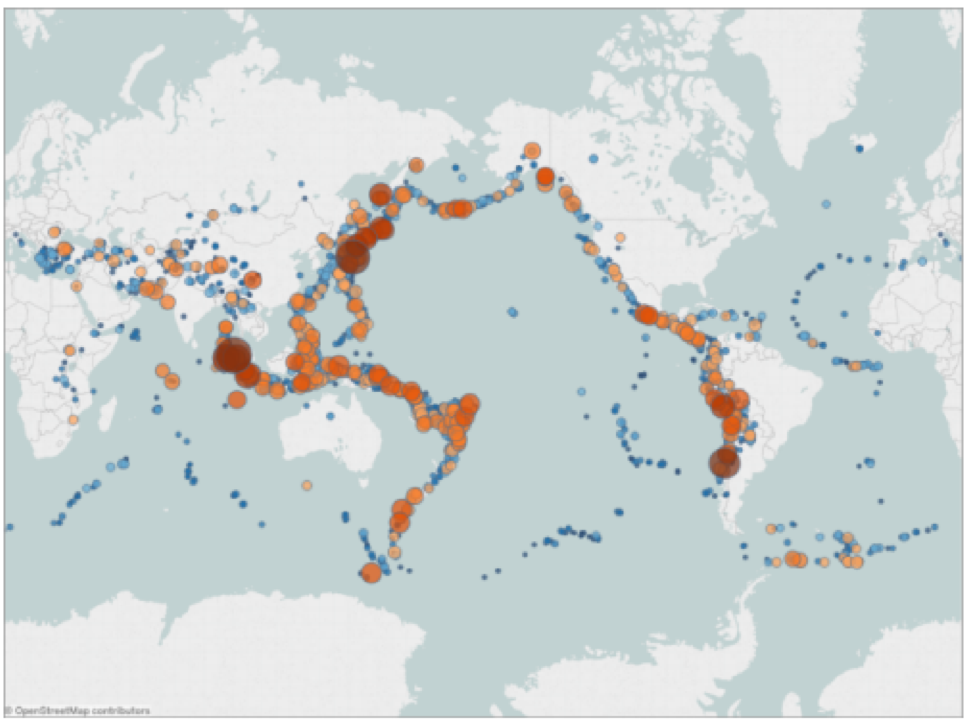
Custom views with the Layer Control

When you publish a view with the Layer Control enabled, any changes a user makes to the layers with the Layer Control won't be retained. However, the user can **Use Custom Views** on page 3224 to keep their changes intact.

Create Maps that Show Quantitative Values in Tableau

You can create maps in Tableau Desktop that show quantitative values, similar to the example below. These types of maps are called proportional symbol maps.

Proportional symbol maps are great for showing quantitative values for individual locations. They can show one or two quantitative values per location (one value encoded with size, and, if necessary, another encoded with color). For example, you can plot earthquakes recorded between 1981 to 2014 around the world, and size them by magnitude. You can also color the data points by magnitude for additional visual detail.



This topic illustrates how to create a proportional symbol map using an example. Follow the example below to learn how to set up your data source, and build the view for a proportional symbol map.

Your data source

To create a proportional symbol map, your data source should include the following types of information:

- Quantitative values
- Latitude and Longitude coordinates or location names (if recognized by Tableau)

It's also recommended that your data contain a large variation of values, otherwise your symbols will appear approximately the same size in the view.

The following table is a snippet of the Earthquake data source, which is included in the Create Proportional Symbol Maps in Tableau Example Workbook on Tableau Public. It contains columns for earthquake magnitude and magnitude to the power of ten, and columns for latitude and longitude vales. It also contains a column for the date and earthquake ID for added clarity and organization.

Earthquake Date Time	ID	Magnitude	Magnitude^10	Latitude	Longitude
1/1/73	centennial19730101114235	6.00000	17,488,747.04	-35.570	-15.427
1/2/73	pde19730102005320300_66	5.50000	25,329,516.21	-9.854	117.427
1/3/73	pde19730103022942800_33	4.80000	6,492,506.21	1.548	126.305
1/4/73	pde19730104003142000_33	4.50000	3,405,062.89	41.305	-29.272
1/5/73	pde19730105003948200_36	4.70000	5,259,913.22	0.683	-80.018
1/6/73	pde19730106061852300_83	4.90000	7,979,226.63	-22.354	-69.310

Basic map building blocks:

Columns shelf:	*Longitude* (continuous measure, longitude geographic role assigned)

Rows shelf:	*Latitude* (continuous measure, latitude geographic role assigned)
Detail:	One or more *dimensions*
Size:	*Measure* (aggregated)
Mark type:	*Automatic*

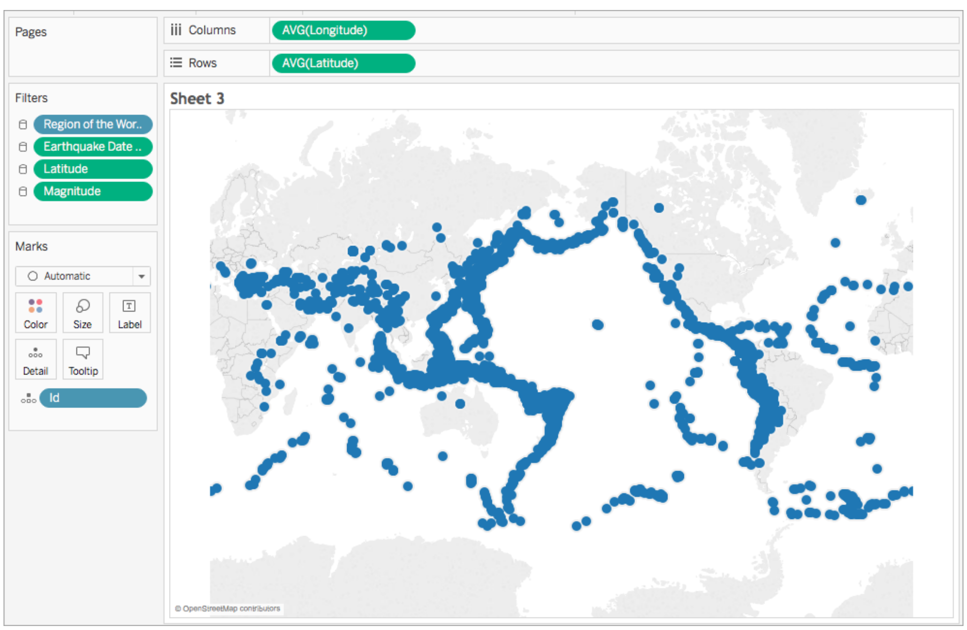
Build the map view

To follow along with this example, download the Create Proportional Symbol Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

1. Open a new worksheet.
2. In the Data pane, under Measures, double-click **Latitude** and **Longitude**.

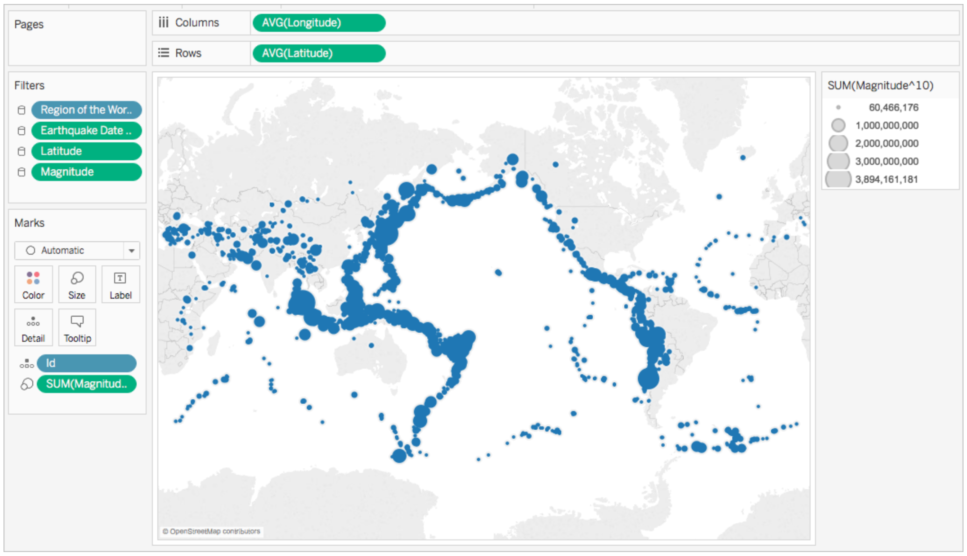
   Latitude is added to the Rows shelf, and Longitude is added to the Columns shelf. A map view with one data point is created.
3. From Dimensions, drag **ID** to **Detail** on the Marks card. If a warning dialog appears, click **Add all members**.

   A lower level of detail is added to the view.

4. From Measures, drag **Magnitude^10** to **Size** on the Marks card.

Note that the Magnitude^10 field is used to encode size, instead of the Magnitude field. This is because Magnitude^10 contains a wider range of values, so the differences between values can be seen visually.



You now have a proportional symbol map. The larger data points represent earthquakes with larger magnitudes, and the smaller data points represent earthquakes with smaller magnitudes.

In most cases, this is as far as you need to go to show quantitative values for single locations. However, in this case, since there are so many data points in the view, more visual detail is needed to help you differentiate between the earthquake magnitudes, and to help you spot any trends.

5. From Measures, drag **Magnitude** to **Color** on the Marks card.
6. On the Marks card, click **Color** > **Edit Colors**.
7. In the Edit Colors dialog box, do the following:
    - Click the color drop-down and select the **Orange-Blue Diverging** palette from the list.
    - Select **Stepped Color**, and then enter **8**.

      This creates eight colors: four shades of orange, and four shades of blue.
    - Select **Reversed**.

This reverses the palette so that orange represents a higher magnitude than blue.

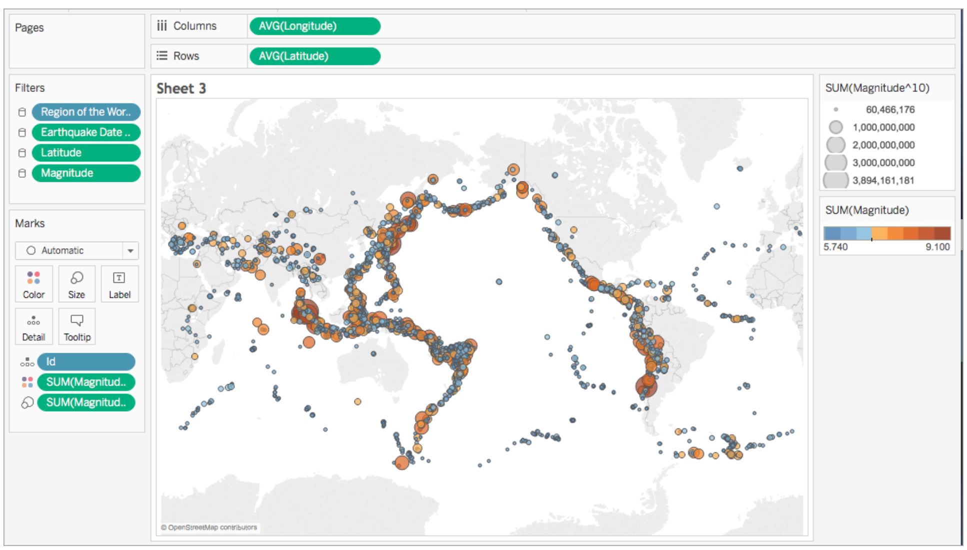
- Click **Advanced**, select **Center**, and then enter **7**.

  This shifts the color palette and ensures that any earthquake over 7.0 magnitude will appear orange in color, and any earthquake under 7.0 magnitude will appear blue in color.

- Click **OK**.

8. On the Marks card, click **Color** again, and then do the following:
   - For **Opacity**, enter **70%**.
   - Under Effects, click the **Border** drop-down menu and select a dark blue border color.

   The map view updates with new colors. The dark orange data points represent earthquakes with higher magnitudes, while the dark blue data points represent earthquakes with lower magnitudes. The opacity of the marks is at 70% so you can see where the data points overlap.



9. On the Marks card, right-click the **ID** field and select **Sort**.

10. In the Sort dialog box, do the following:

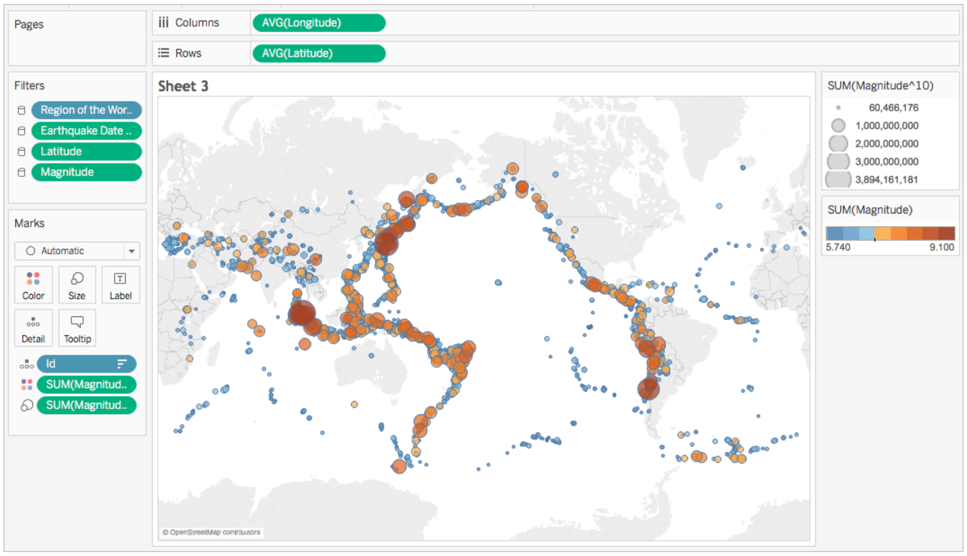
    - For Sort Order, select **Descending**.

    - For Sort By, select **Field**, and then click the drop-down and select **Magnitude**.

    - Click **OK**.

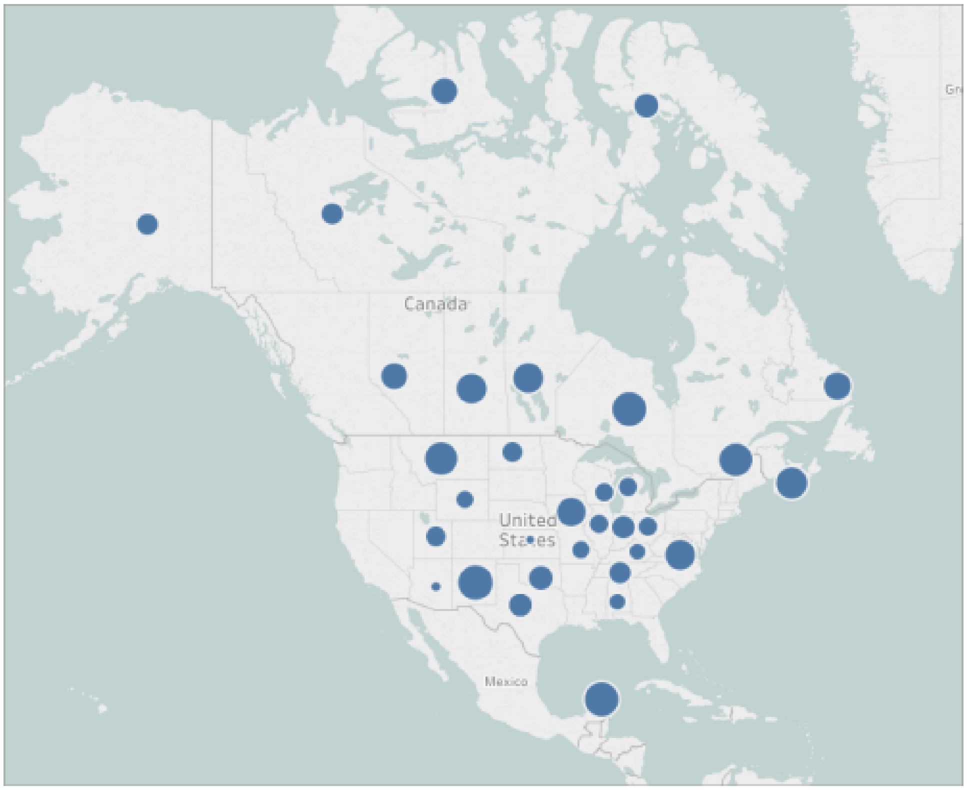
This sorts the data points in the view so that the larger magnitudes appear on top.

Your proportional symbol map is now complete.



Point location and attribute meaning

It's important to note that symbols on a map can sometimes be misinterpreted as representing actual ground area. For example, if you have a map view that plots earth impact craters across North America, and sizes each symbol by the diameter (in kilometers) of the impact area, you might get a map view that looks like this:



In this particular case, it could be very easy to interpret the size of these data points as representing the actual ground area of the craters. Your audience might assume that most of Montana, U.S. was destroyed by a crater, which is not accurate. In reality, the crater in Montana was simply one of the larger craters in the data source and has been sized accordingly.

In cases such as these, it might be helpful to include annotations, or explanations of what the size actually represents to avoid misinterpretations. Even if it seems obvious.

See Also:

Mapping Concepts in Tableau on page 1781

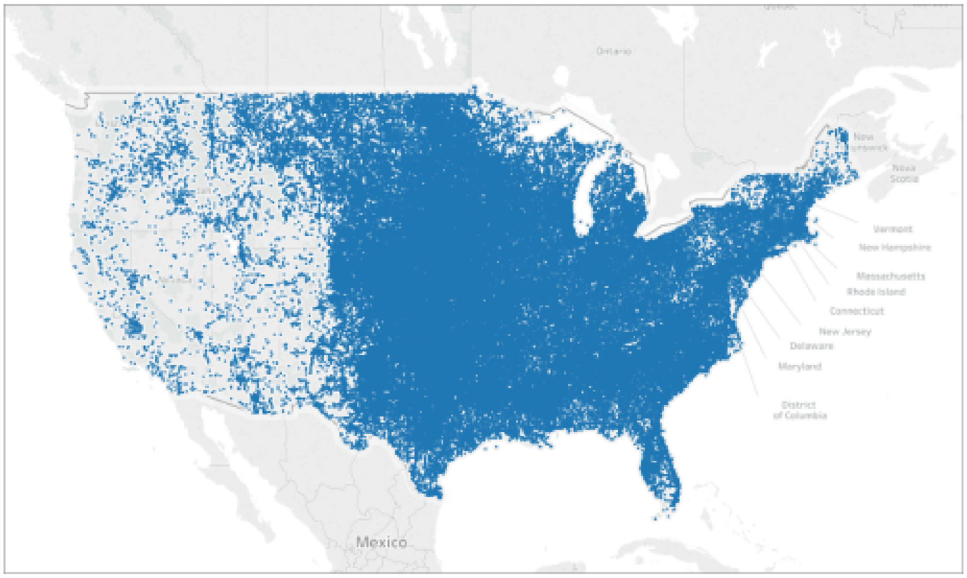
Assign Geographic Roles on page 1834

Create Maps that Highlight Visual Clusters of Data in Tableau

You can create maps in Tableau Desktop that help you spot visual clusters, similar to the example below. These types of maps are called point distribution maps.

Point distribution maps are great for showing the how locations of your data points are distributed.

This topic illustrates how to create a point distribution map using an example. Follow the example in this topic to learn how to set up your data source, and build the view for a point distribution map.



Your data source

To create a point distribution map, your data source should include the following types of information:

- Latitude and longitude coordinates for all of your locations

For example, the following table is a snippet of the Hail data source, which is included in the Create Point Distribution Maps in Tableau Example Workbook on Tableau Public. It contains columns for the latitude and longitude coordinates of hail storms across the United States from 1955 to the 2013.

Latitude	**Longitude**
32.2000	-101.5000
38.5800	-92.5800
41.2000	-89.6800
39.2800	-87.4000

41.7800	-87.7800
39.5000	-90.0800

Basic map building blocks

Columns shelf:	*Longitude* (continuous dimension, longitude geographic role assigned)
Rows shelf:	*Latitude* (continuous dimension, latitude geographic role assigned)
Mark type:	*Automatic*

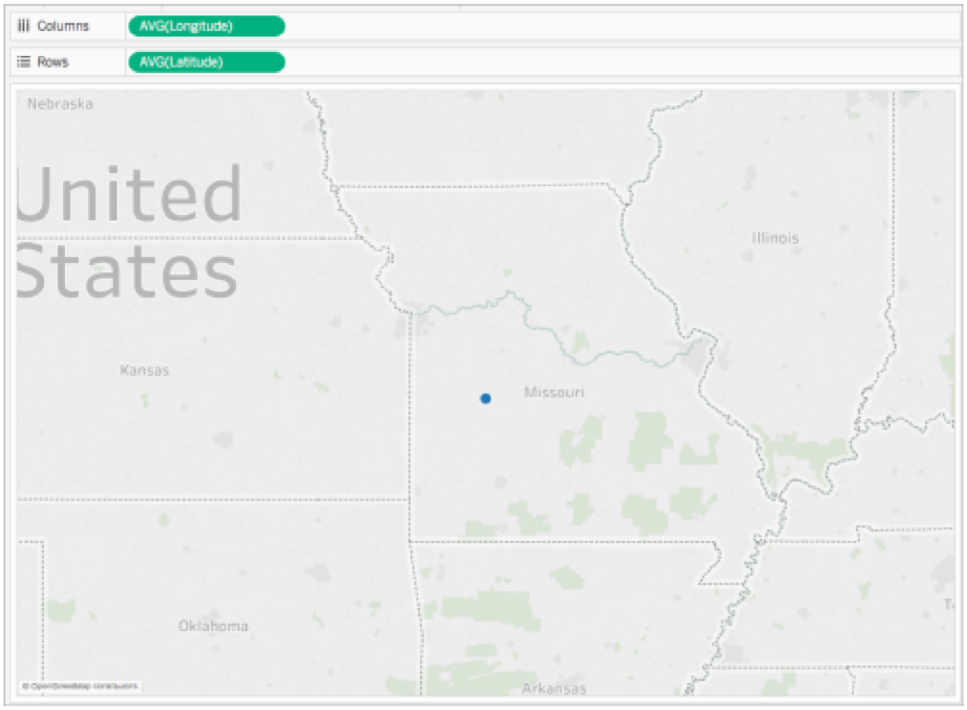
Build the map view

To follow along with this example, download the Create Point Distribution Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

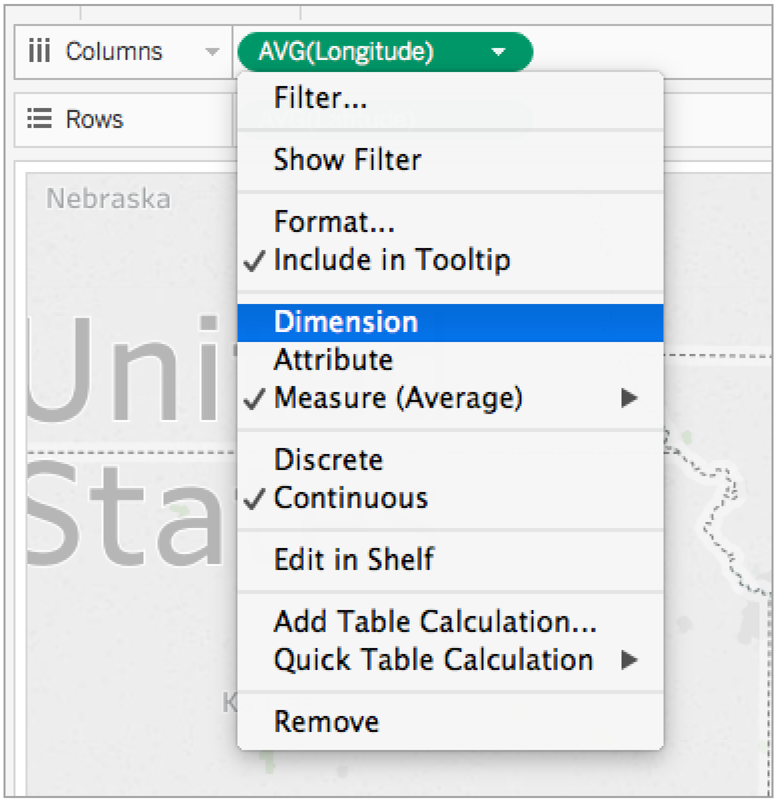
1. Open a new worksheet.
2. Ensure that the **Latitude** geographic role is assigned to your latitude field, and the **Longitude** geographic role is assigned to your longitude field.

   For more information, see **Assign a geographic role to a field** on page 1835.
3. In the Data pane, double-click **Latitude** and **Longitude** to add them to the view.

   The Latitude and Longitude fields are added to the Columns and Rows shelves, and a map view with one data point is created.



4. On the Columns shelf, right-click **Longitude** and select **Dimension**.

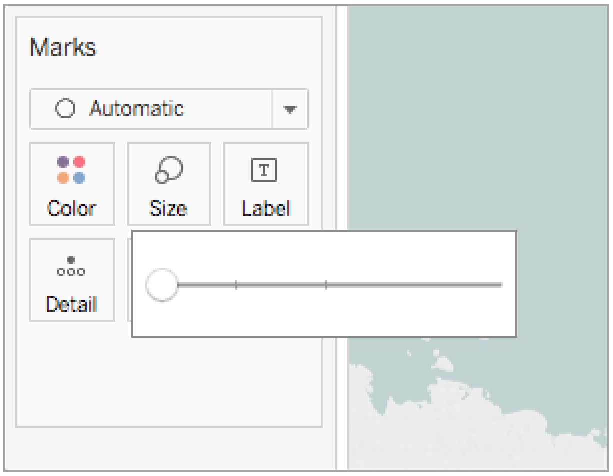


5. On the Rows shelf, right-click **Latitude** and select **Dimension**.

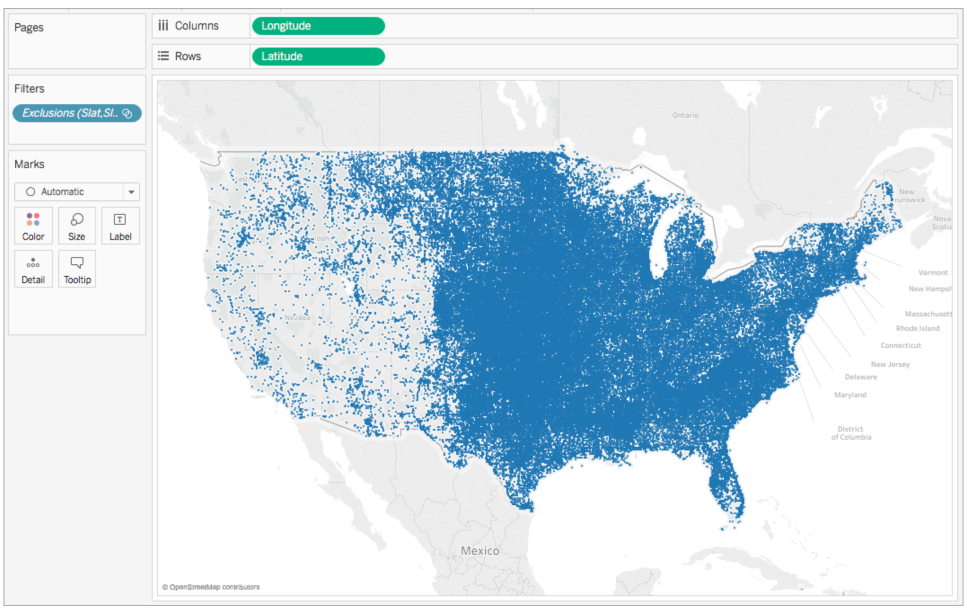
   The map view updates with every location in your data source.

   > **Note**: You might need to filter some data points from the view.

6. On the Marks card, click **Size**, and then move the slider to the left.



Your point distribution map is now complete. A data point for every location in your data source is now plotted on the map (minus any you filtered from the view). You can now see that a majority of hailstorms occur in the eastern half of the United States.



See also:

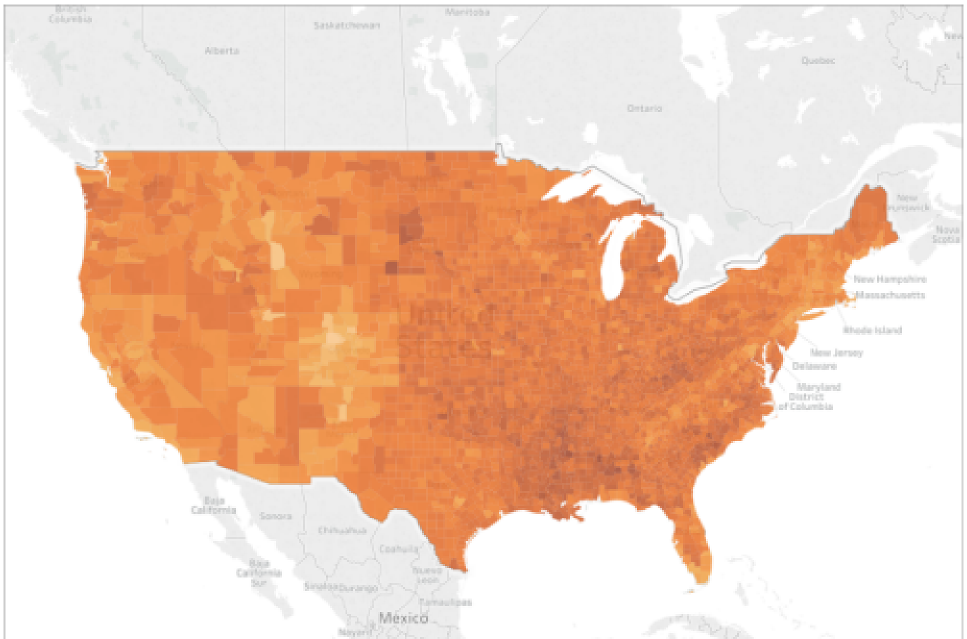
Mapping Concepts in Tableau on page 1781

Create Maps that Show Quantitative Values in Tableau on page 1883

Create Maps that Show Ratio or Aggregated Data in Tableau

You can create maps in Tableau Desktop that show ratio or aggregated data, similar to the example below. These types of maps are called choropleth maps, or filled maps.

Choropleth maps are best for showing ratio or aggregated data for polygons. These polygons can be counties, regions, states, or any area or region that can be geocoded in Tableau. They can even be custom territories created in Tableau. For more information, see **Create Territories on a Map** on page 1981.



This topic illustrates how to create a choropleth map using an example. Follow the example below to learn how to set up your data source, and build the view for a choropleth map.

Your data source

To create a choropleth map, your data source should include the following types of information:

- Quantitative or qualitative values.
- Location names (if recognized by Tableau), or custom polygons. For more information, see **Create Tableau Maps from Spatial Files** on page 1855.

The following table is a snippet of the countyObesity + (Obesity_State_County) data source, which is included in the Create Choropleth Maps in Tableau Example Workbook on Tableau Public. It contains columns for State, County, and Obesity Percent - 2012.

State	County	Obesity Percent - 2012
Alabama	Autauga County	30.9000
Alabama	Baldwin County	26.7000
Alabama	Barbour County	40.8000
Alabama	Bibb County	40.1000
Alabama	Blount County	32.4000
Alabama	Bullock County	44.5000

Basic map building blocks:

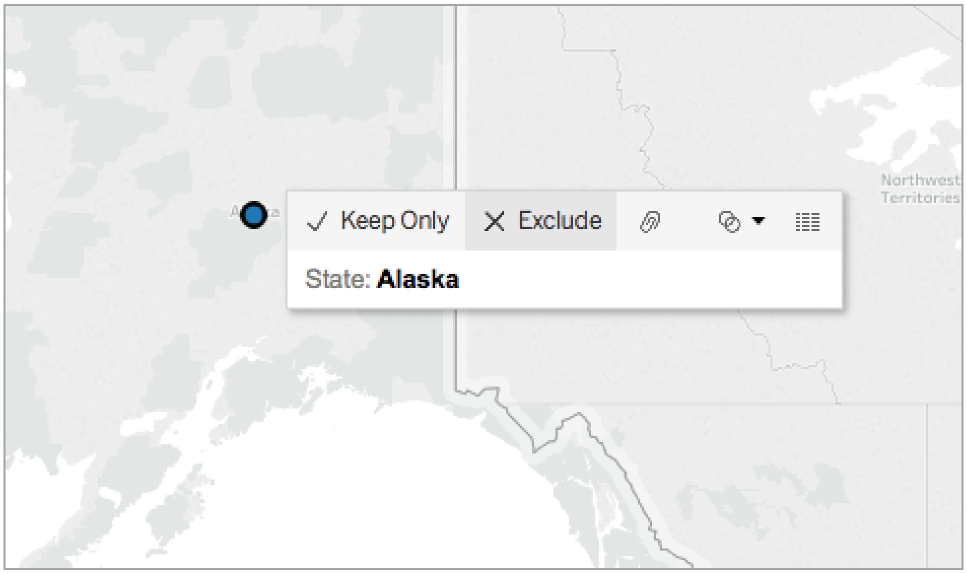
Columns shelf:	*Longitude* (continuous measure, longitude geographic role assigned)
Rows shelf:	*Latitude* (continuous measure, latitude geographic role assigned)
Detail:	One or more *geographic units* (dimensions with geographic roles assigned)
Color:	*Measure* or *dimension*
Mark type:	*Automatic* or *Map*

Build the map view

To follow along with this example, download the Create Choropleth Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

1. Open a new worksheet.
2. In the **Data** pane, under Dimensions, state_Obesity, double-click **State**.

   State is added to Detail on the Marks card and Longitude and Latitude are added to the Columns and Rows shelves. A map view is created with a data point for each state in your data source.
3. In the map view, select the **Alaska** and **Hawaii** data points, and then click **Exclude** on the tooltip that appears.

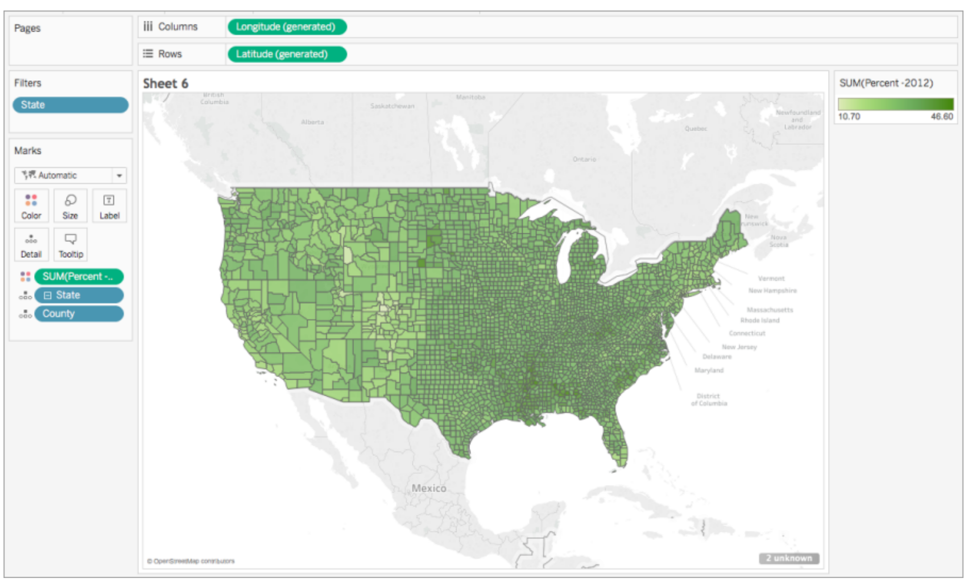


4. On the Marks card, click the plus icon on the **State** field to drill down to the County level of detail.

   There is now a data point for every county in your data source.

5. From Measures, under countyObesity, drag **Percent-2012** to **Color** on the Marks card.

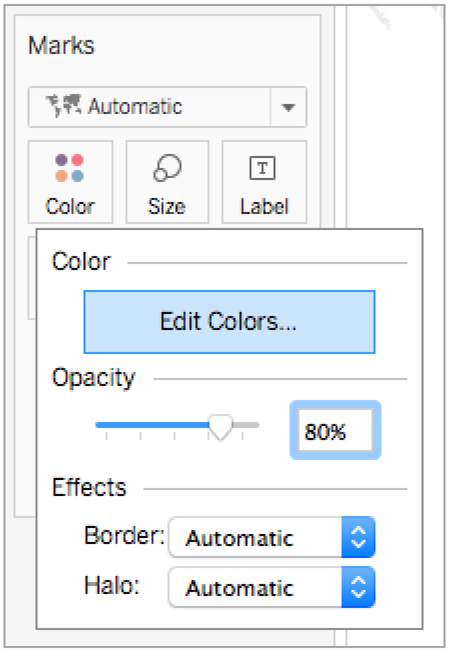
   The map view changes to a filled map mark type and the polygons are colored green. Note that the default aggregation for the Percent-2012 measure is SUM by default.



6. On the Marks card, right-click the **Percent-2012** field and select **Attribute**.

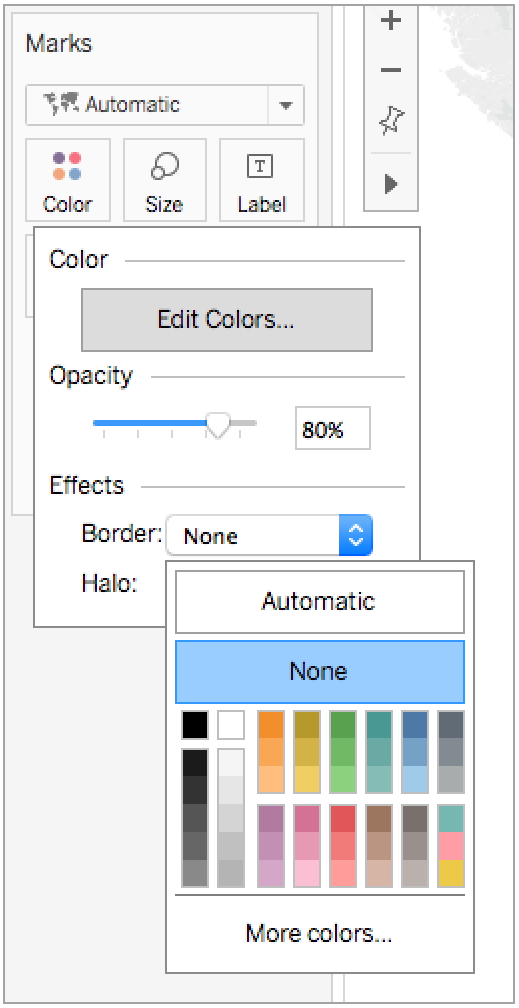
Since the data in this field is already a percentage, it does not make sense to aggregate it as a sum.

7. On the Marks card, click **Color** and select **Edit Colors**.

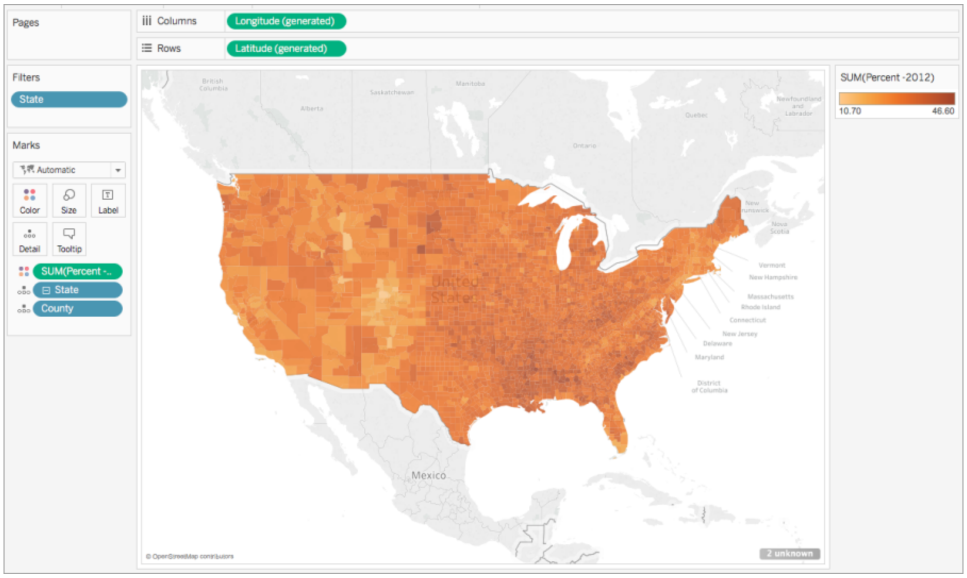


8. In the Edit Color dialog box, click the color drop-down, select **Orange**, and then click **OK**.

9. On the Marks card, click **Color** and then, under Effects, click the **Border** drop-down and select **None**.

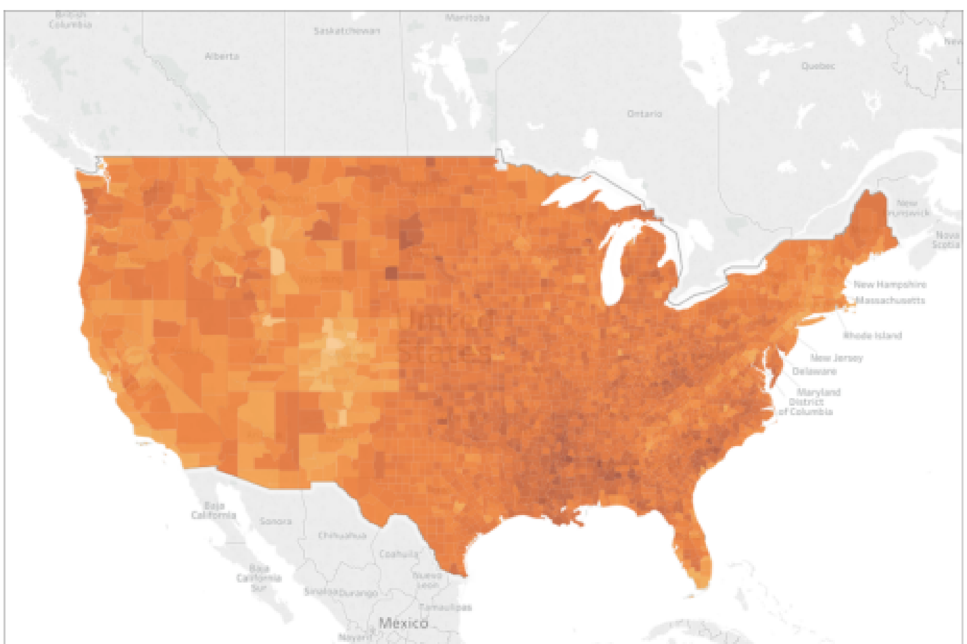


You now choropleth map is now complete. There is an obesity percentage for each polygon (county) on the map view.

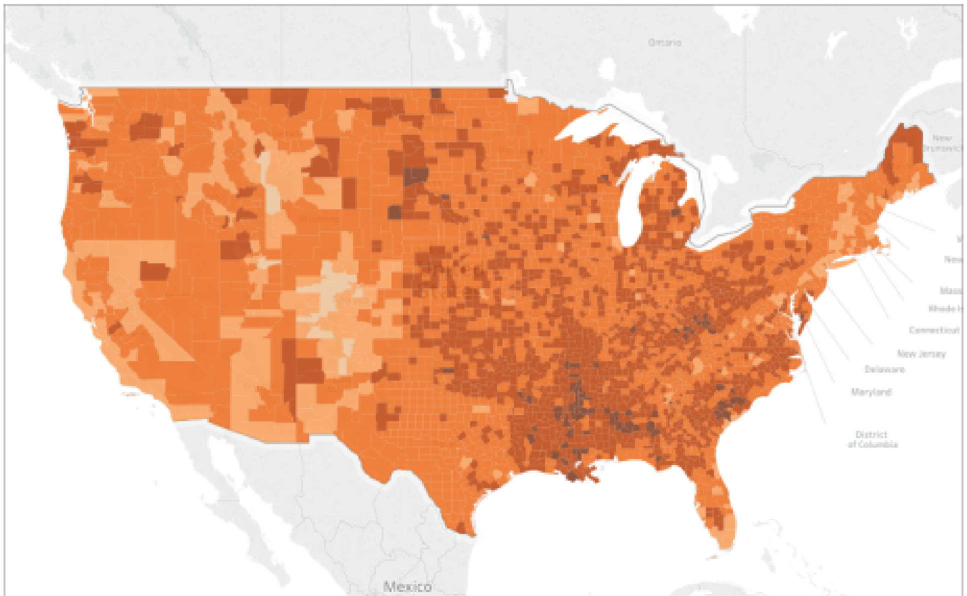


Note about color distribution

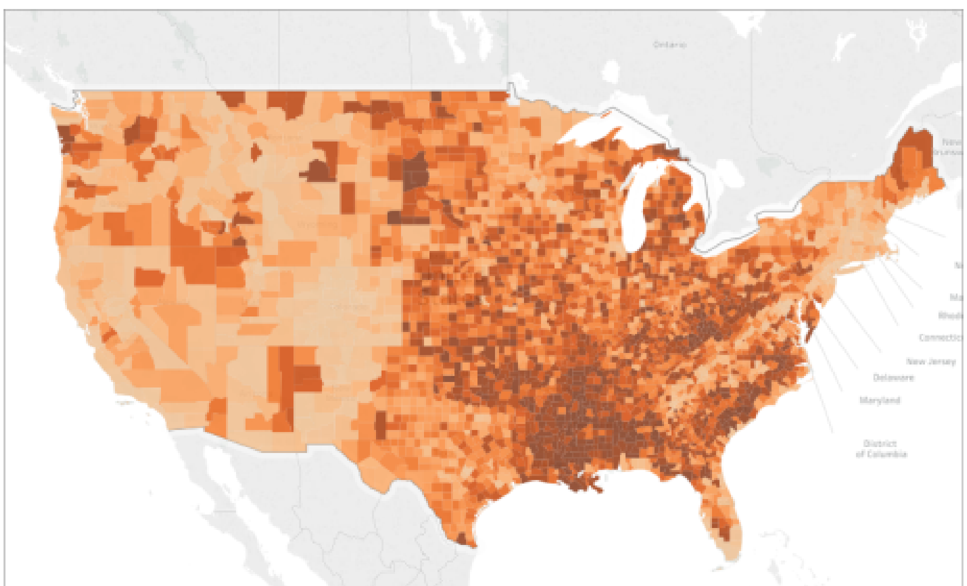
The distribution you specify for the color of the polygons highly affects how people interpret your data. For example, the map you created in the **Build the map view** on page 1896 section used the Tableau default color distribution. Color was distributed evenly across all values. The map view looked like this:



However, if you decide to distribute the values across five colors (stepped color, in Tableau), the map view might look like this:



Or maybe you want to see quartiles. Well, then your map might look like this:



Notice how the high obesity rates in the South are more visible when you use quartiles?

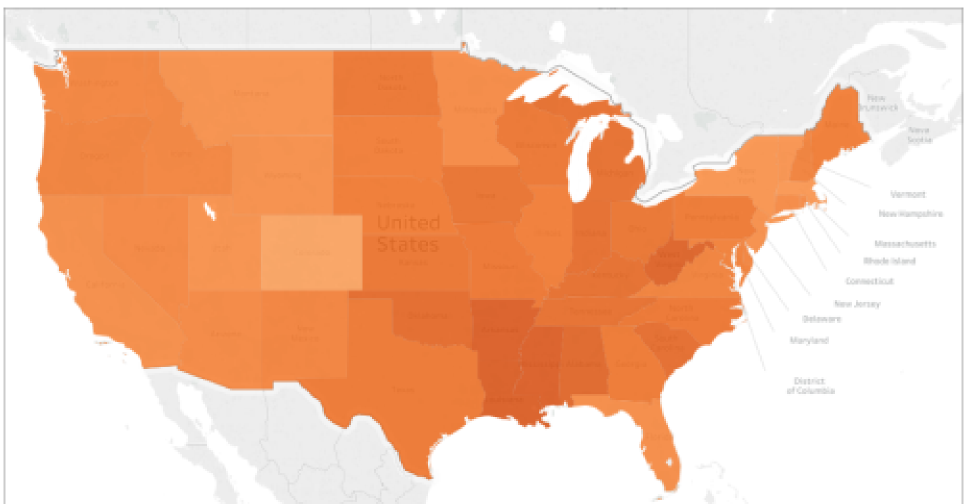
The color distribution you choose depends on the information you're trying to show.

For example, if you wanted to show that obesity rates are higher in the South, you might choose the quartile example. However, if you want to show that obesity rates are high in the United States in general, you might choose the first example, which distributed color evenly. Neither map is right or wrong; they just tell different stories.

Note about level of detail

The level of detail you specify in a choropleth map determines the patterns you want to see in the data. If you want to see a simplified pattern, consider aggregating up to a higher level of detail. If you want to dig in to find smaller trends, consider aggregating down to a lower level of detail.

For example, the map view you created in the **Build the map view** on page 1896 section was aggregated at the County level of detail. The county level of detail was the lowest level of detail in the data source. At a higher level of detail, such as the State level, the map view might look similar to the following. Notice that the patterns in the data are simplified at this level of detail.



See also:

Mapping Concepts in Tableau on page 1781

Assign Geographic Roles on page 1834

Create Maps that Show Quantitative Values in Tableau on page 1883

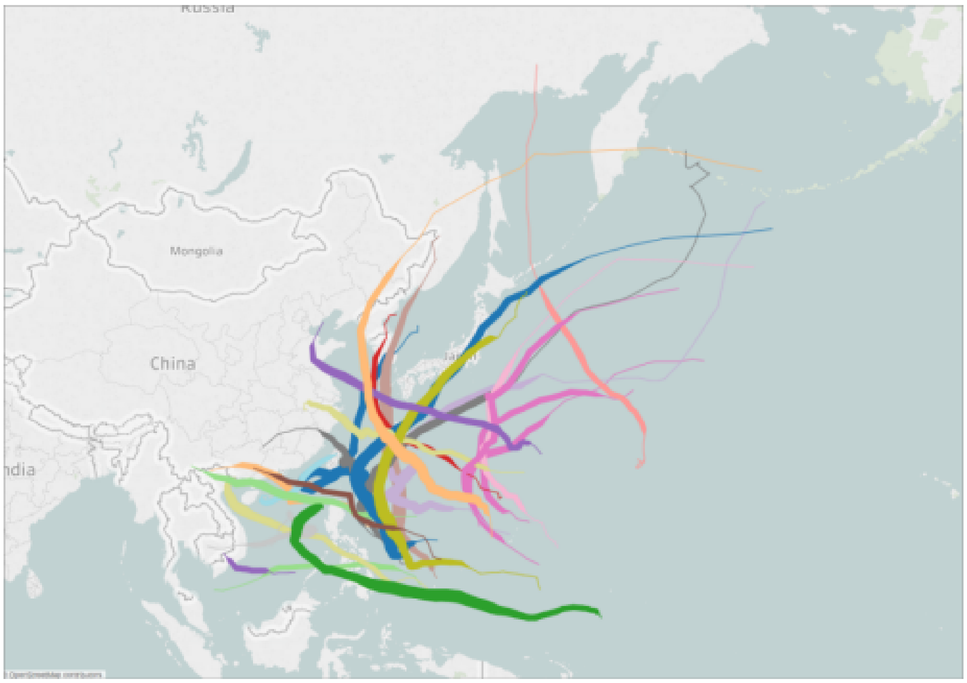
Create Territories on a Map on page 1981

Create Maps that Show a Path Over Time in Tableau

You can create maps in Tableau Desktop that show a path over time, similar to the example below. These types of maps are called flow maps, or path maps.

Flow maps are great for when you want to show where something went over time, such as the path of a storm.

This topic illustrates how to create a flow map using an example. Follow the example below to learn how to set up your data source, and build the view for a flow map.



Your data source

> **Note**: Starting in Tableau version 10.4, you can connect to spatial files that contain linear geometries. If you have spatial data with linear geometries, you may not need to perform the steps below. To learn how to create a map using spatial data with linear geometries, see **Create Tableau Maps from Spatial Files** on page 1855

To create a flow map, your data source should include the following types of information:

- Latitude and longitude coordinates for each data point in a path
- A column to define the order to connect the points (this can be date information, or manually applied numbers, such as 1, 2, 3, 4, 5)
- A unique ID for each path
- Enough data points to shape each path into a line

For example, the following table is a snippet of the Storm data source, which is included in the Create Flow Maps in Tableau Example Workbook on Tableau Public. It contains data on the paths of storms, and has columns for **Latitude** and **Longitude**, **Date**, and **Storm Name**. In this example, the Date column is used as an order to connect the data points, and the Storm Name column is used as a unique ID for each path.

Though only a few data points for the storm PAKHAR are shown in this example, the actual data source has enough entries to provide a detailed path for every storm recorded in 2012.

Note that the table also includes two optional columns: Basin and Wind Speed. These fields can be used to quickly filter and add visual detail to the view. You'll see how in the section, **Build the map view** on the facing page.

Storm Name	Date	Latitude	Longitude	Basin	Wind Speed (kt)
PAKHAR	3/26/12 12:00:00 AM	9.5000	115.700	West Pacific	0
PAKHAR	3/26/12 6:00:00 AM	9.5000	115.400	West Pacific	0
PAKHAR	3/26/12 12:00:00 PM	9.5000	115.100	West Pacific	0
PAKHAR	3/26/12 6:00:00 PM	9.4000	114.800	West Pacific	0
PAKHAR	3/27/12 12:00:00 AM	9.4000	114.500	West Pacific	0
PAKHAR	3/27/12 6:00:00 AM	9.4000	114.300	West Pacific	35

Basic map building blocks:

Columns shelf:	*Longitude* (continuous measure, longitude geographic role assigned)
Rows shelf:	*Latitude* (continuous measure, latitude geographic role assigned)
Detail:	*Dimension* (unique ID for each path)
Path:	*Date field* or *order field* to define order to connect the data points
Mark type:	*Line*

Build the map view

To follow along with this example, download the Create Flow Maps in Tableau Example Workbook from Tableau Public, and open it in Tableau Desktop.

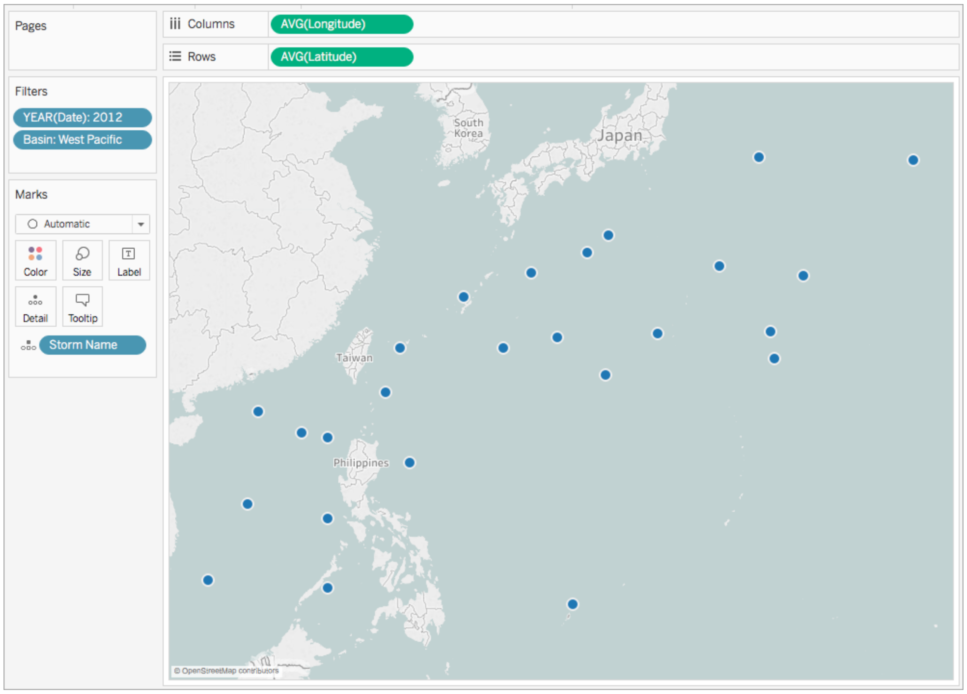
1. Open a new worksheet.
2. In the **Data** pane, under Measures, double-click **Latitude** and **Longitude**.

   The Latitude and Longitude fields are added to the Columns and Rows shelves, and a map view with one data point is created.
3. From Dimensions, drag **Storm Name** to **Detail** on the Marks card.

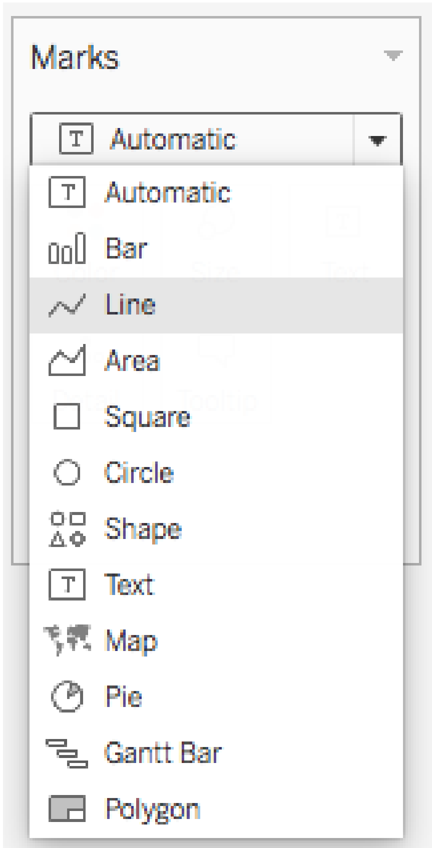
   The map view updates with a data point for every storm in the data source. In the next steps, you will narrow the storms down to only those that occurred in the western Pacific Ocean in 2012.
4. From Dimensions, drag **Date** to the **Filters** shelf.
5. In the Filter Field [Date] dialog box that appears, select **Years**, and then click **Next**.
6. In the Filter [Year of Date] dialog box that appears, click **2012**, and then click **OK**.

   The map view updates to show only the storms that occurred in 2012.
7. From Dimensions, drag **Basin** to the **Filters** shelf.
8. In the Filter Field [Basin] dialog box that appears, select **West Pacific**, and then click **OK**.

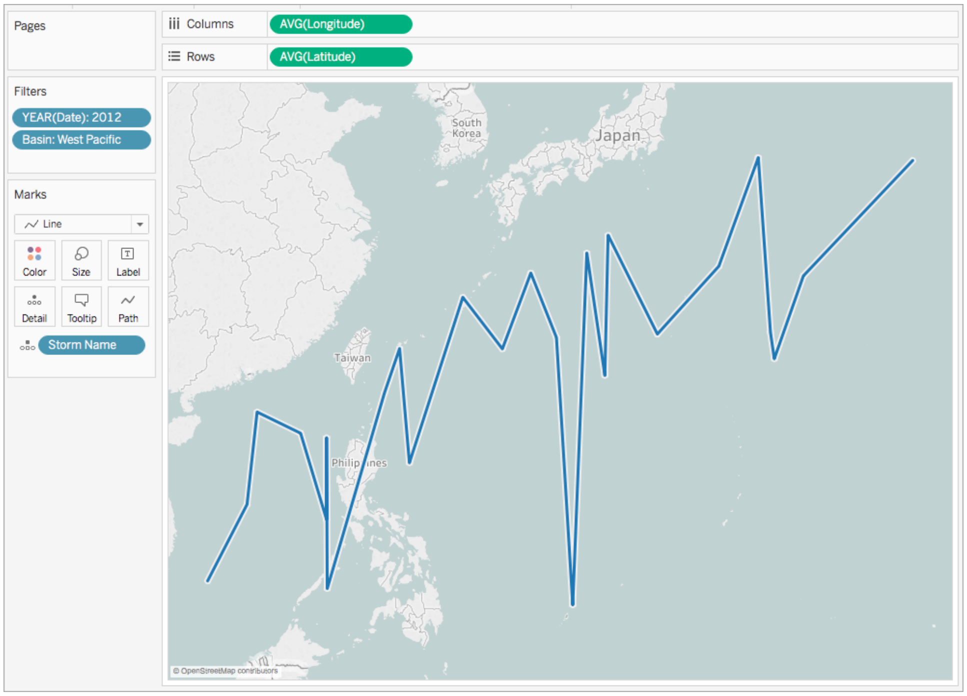
   The map view updates to show only storms that occurred in the western part of the Pacific Ocean.



9. On the Marks card, click the mark-type drop-down and select **Line**.



A Path button appears on the Marks card, and the map view updates with a line connecting every data point.

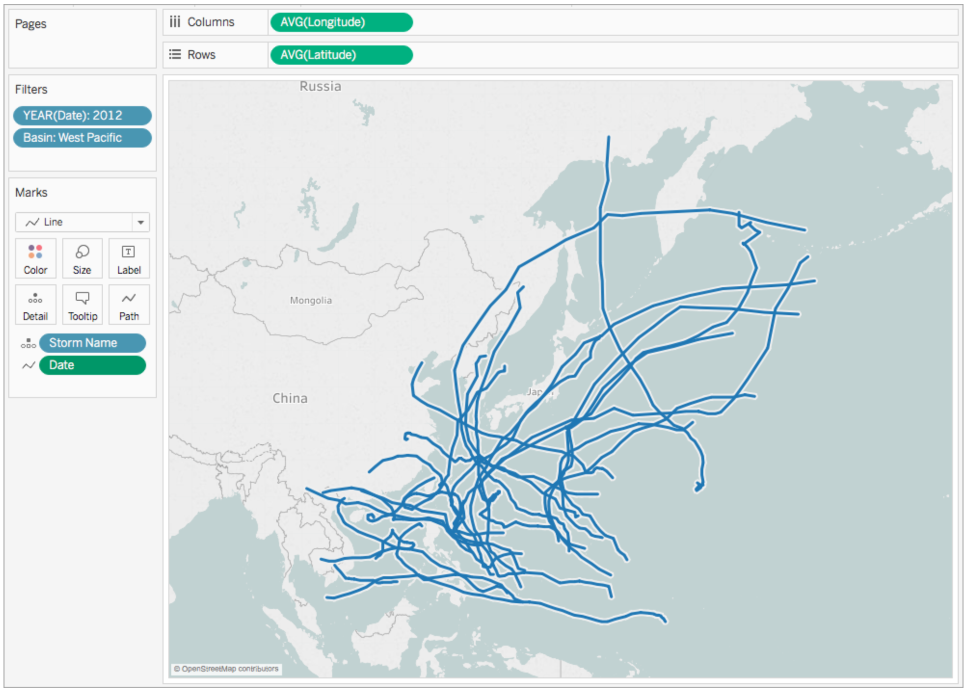


10. From Dimensions, drag **Date** to **Path** on the Marks card.

    The line disappears. This is because the Date field is set to discrete years. Since the date field in the Storm data source includes day, month, year, and time, this is not the correct level of detail for this field.

11. On the Marks card, right-click the **YEAR(Date)** field and select **Exact Date**.

    Now the map view updates with a data point for every recorded date and time. You can now see the individual paths of each storm.



12. From Measures, drag **Wind Speed** to **Size** on the Marks card.

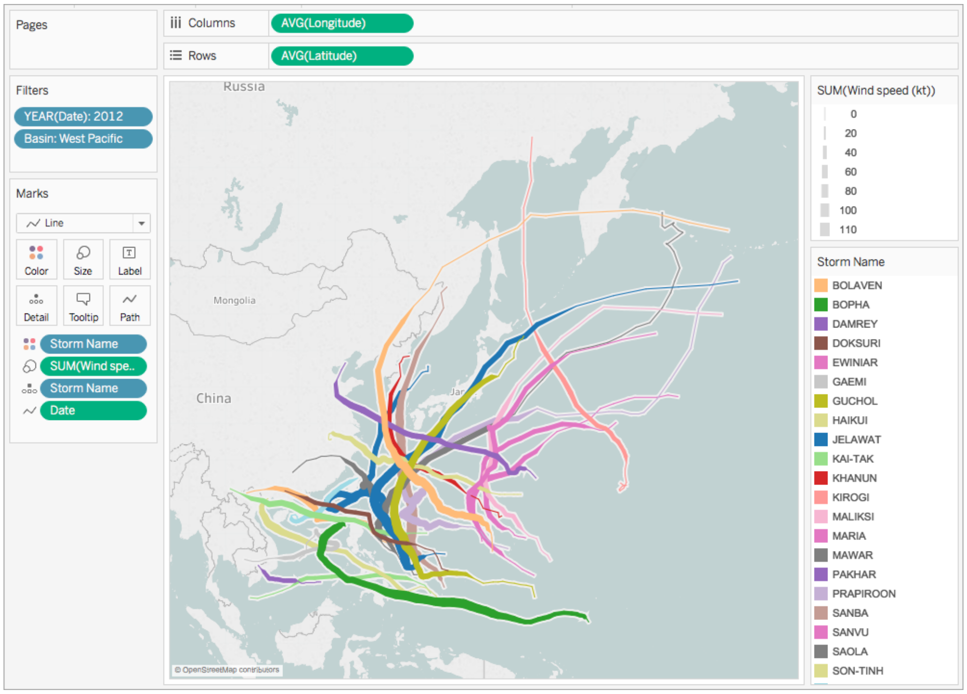
    The map view updates to show the varying wind speeds along each storm path.

13. On the Marks card, right-click the **SUM(Wind Speed)** field and select **Measure** > **Average**.

14. From Dimensions, drag **Storm Name** to **Color** on the Marks card.

    Each storm path is assigned a color and the flow map is now complete.

You can now see the paths for every recorded storm that occurred in the West Pacific basin in 2012. You can also see at what point in their path their wind speeds were strongest.



See also:

Mapping Concepts in Tableau on page 1781

Create Maps that Show Paths Between Origins and Destinations in Tableau on page 1917

Tableau Community post: Origin-Destination Maps (or Flow Maps)

Create Heatmaps that Show Trends or Density in Tableau

You can create maps in Tableau to reveal the patterns or relative concentrations that might otherwise be hidden due to an overlapping mark on a map. These types of maps are called heatmaps, or density map. Tableau creates a heatmap by grouping overlaying marks, and color-coding them based on the number of marks in the group.

Density maps help you identify locations withhow greater or fewer numbers of data points. Density maps are most effective when working with a data set containing many data points in a small geographic area.

Your data source

To create a density map, your data source should:

- Latitude and Longitude coordinates or location names (if recognized by Tableau)

Tableau can recognize location names and create a density map using the point locations assigned to Tableau geocoding locations, but density maps are most effective when the location data is very precise, such as location coordinates in a limited space. Density marks work best where the specific locations change continuously and smoothly across space, rather than values constrained to discrete locations like borough or neighborhood.

Basic map building blocks:

Columns shelf:	*Longitude* (continuous dimension, longitude geographic role assigned)
Rows shelf:	*Latitude* (continuous dimension, latitude geographic role assigned)
Detail:	One or more fields with many underlying data points
Mark type:	*Density*

Build the map view

You can choose “Density” from the mark type drop-down and Tableau will compute a density surface on your view. The density surface that appears recomputes as you zoom or filter data on the remaining marks. When using Pages or a small multiples view, the Density is computed across the full domain of the data for comparative analysis.

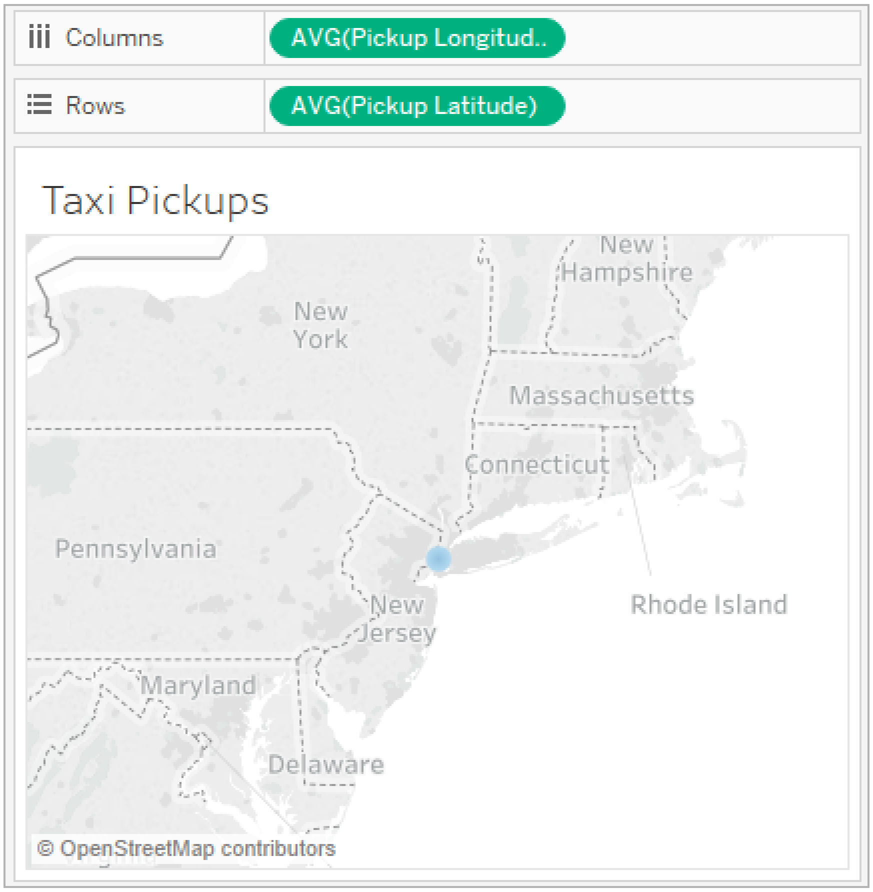
To follow along with this example, download the Create Heatmaps in Tableau Example Workbook (click Download in the upper right hand corner), and open it in Tableau Desktop.

1. Open a new worksheet. In this data source, these fields are named Pickup Latitude and Pickup Longitude. Ensure that the Pickup Latitude geographic role is assigned to your latitude field, an the Pickup Longitude geographic role is assigned to your longitude field.

   For more information, see Assign a geographic role to a field.

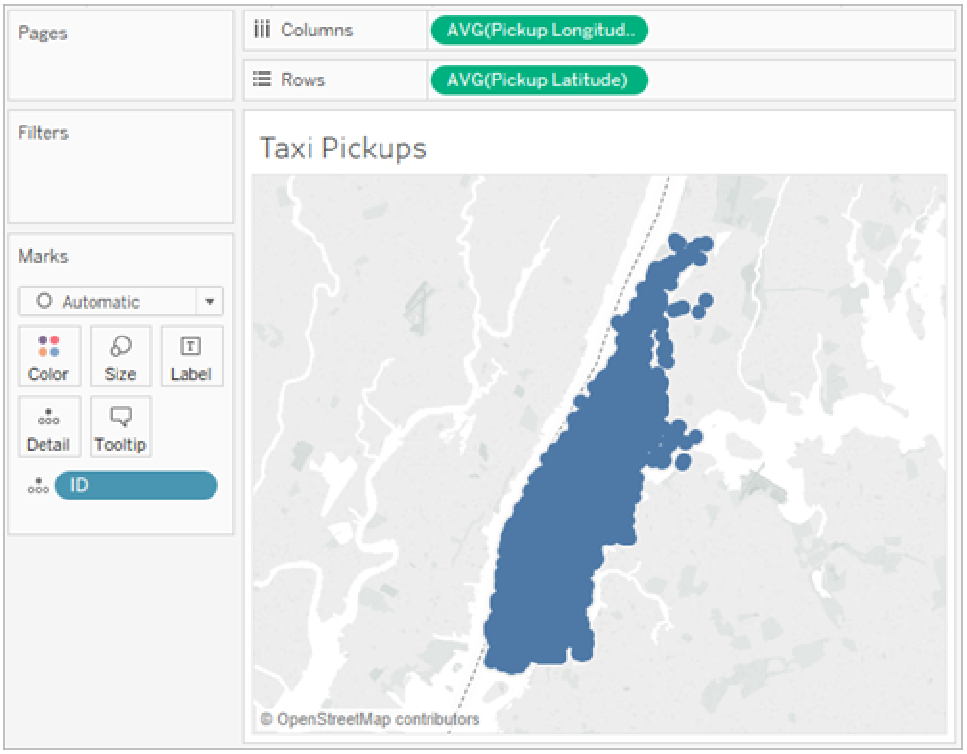
2. In the Data pane, double-click Pickup Latitude and Pickup Longitude to bring them to the view.

   The Latitude and Longitude fields are added to the Columns and Rows shelves, and a map view with one data point is created.



3. Now, let's add distinct marks to the view. Select ID and place it on the Details mark. Because each pickup has its own ID, this breaks up the mark and distinguishes one pickup from another on our map.

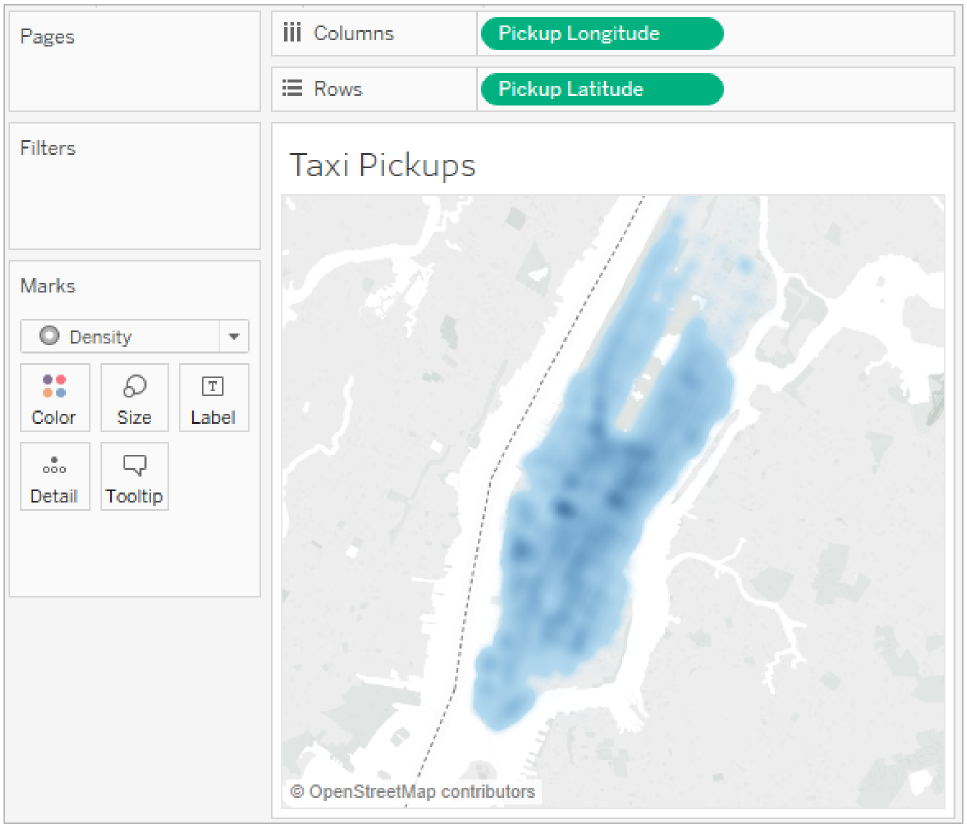
   There will be a warning letting you know that the field added may contain more than the recommended maximum of 1000. Select Add all members.



The map view updates to show marks for every pickup location in your data source. Since every location is within Manhattan, the map will zoom to focus on Manhattan in New York City.

> **Note**: You might need to filter some data points from your view to create the level of zoom desired.

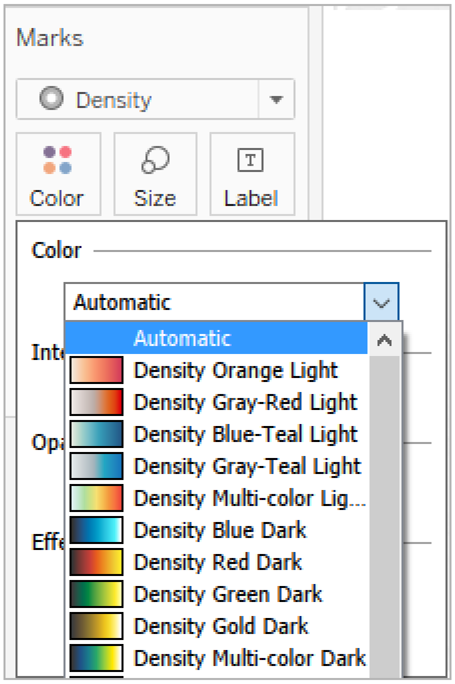
4. On the Marks card, change the mark type to density by selecting the drop-down menu to the right of Automatic and select **Density**.



5. Your basic density map is created. You can see that midtown is the most popular area for taxi pickups in October, although you can adjust the focus further by using the zoom tool. Density will recompute as you zoom in or out.

Adjust the appearance

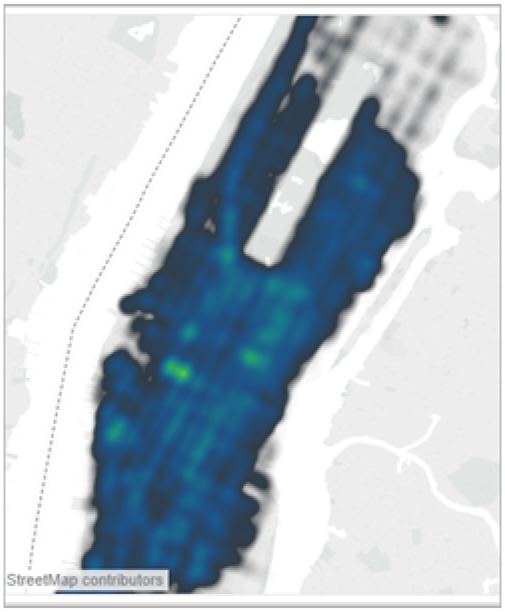
Color: Adjusting the colors of your density map by selecting **Color** from the Marks card. Choose from ten density color palettes, or any of the existing color palettes.

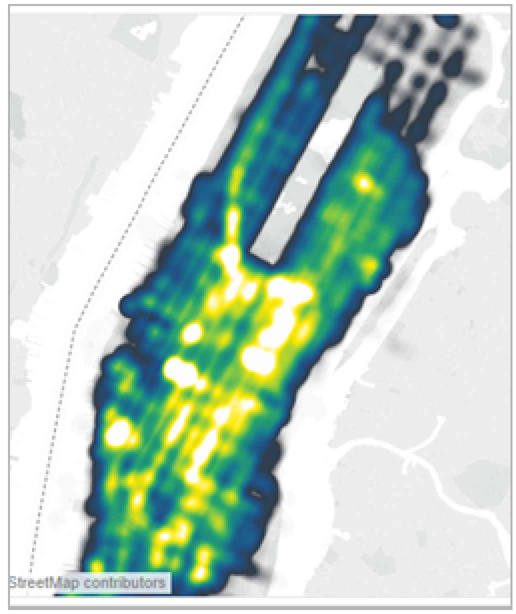


> **Note:** If your data source contains them, negative values will also appear when a measure field is added to **Color**. Use a diverging color palette to make a clear distinction between positive and negative values.

Intensity: In the Color menu, use the **Intensity** slider to increase or decrease the vividness of your map. For example, increasing density lowers the "max heat" spots in your data, so that more appear.

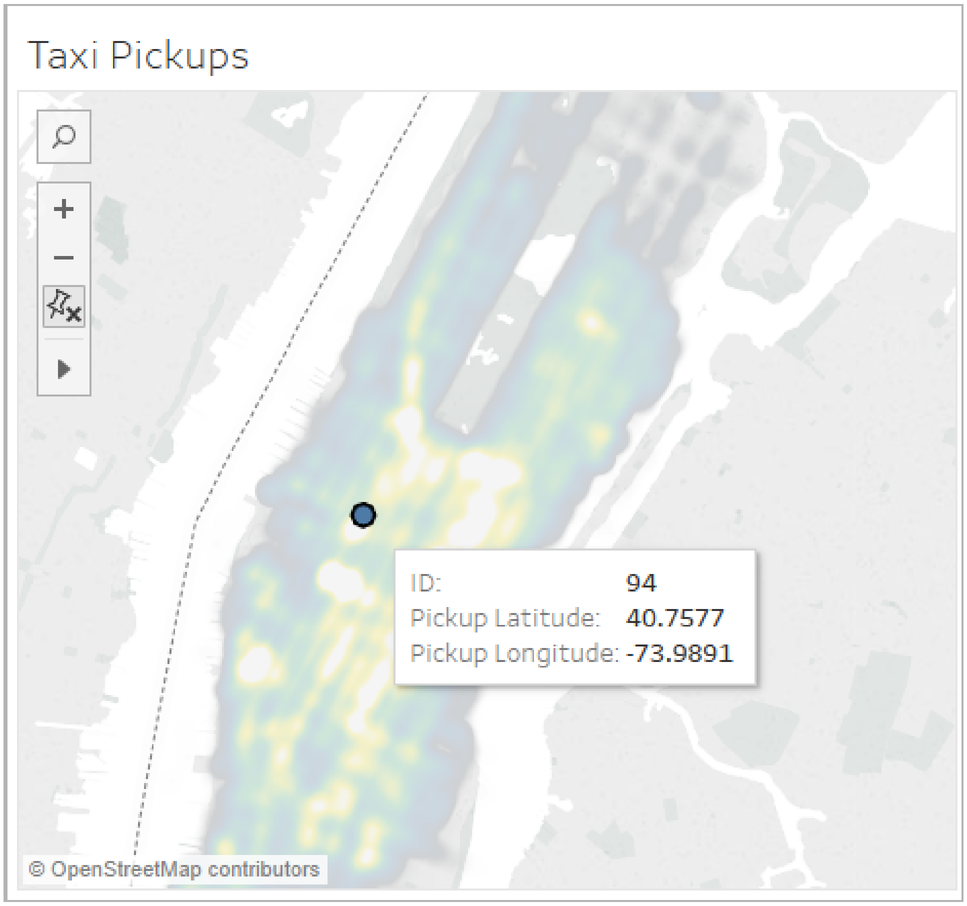
Low Intensity (50%)	High Intensity (80%)





Size: You can use the **Size** shelf to adjust the size of the density marks. Clicking on the Size mark will show the size slider. Adjusting the slider will apply a size increase or decrease to the size of the marks group that makes your heatmap.

Underlying marks): You can select an individual data point from anywhere in your density map. These marks have size (10 pixels) and color (blue) applied by default. Size and color are not adjustable for underlying marks.



Zoom: Selection, tooltips, labels, and hovering all work based on the marks in the zoom of the view. Density maps have no fixed or constant display and will always recompute as you zoom.

Create Maps that Show Paths Between Origins and Destinations in Tableau

You can create maps in Tableau Desktop that show paths between origins and destinations, similar to the examples below. These types of maps are called spider maps, or origin-destination maps.

Spider maps are great for when you're working with hubs that connect to many surrounding points. They are an excellent way to show the path between an origin and one or more destination locations.

There are several ways to create spider maps in Tableau. This topic illustrates how to create a spider map using two examples. Follow the examples below to learn how to set up your data source, and build the view for two different spider maps.

For other examples that might fit closer to your data, see the following workbooks on Tableau Public:

- Puget Sound Radio Tower Bandwidth
- Recruitment Strategies in the English Premier League
- 2014 U.S. Flight Departure Delays

Example 1: Metro station traffic in Paris, France

Learn how to build it!

Your data source

> **Note**: Starting in Tableau version 10.4, you can connect to spatial files that contain linear geometries. If you have spatial data with linear geometries, you may not need to perform the steps below. To learn how to create a flow map using spatial data with linear geometries, see **Create Tableau Maps from Spatial Files** on page 1855

To create this type of spider map in Tableau, your data source should include the following information:

- A path ID for every unique path. See **The Line Group (Path ID) column** on the facing page below for more information.
- Numbers to define the drawing order of each data point (location). See **The Order of Points column** on the facing page below for more information.
- Latitude and longitude coordinates for every location. See the example table below.

It's also recommended that your data source contain a column with your location names, but it is not required.

The following table is a snippet of the Transports data source, which is included in the Create Spider Maps in Tableau Example 1 Workbook on Tableau Public. It contains metro traffic data for all metro lines in Paris, France. The first three metro stations for two metro lines are included in this example, and it contains the required columns **Line Group (Path ID)**, **Order**

of Points, **Latitude**, and **Longitude**. It also contains the additional columns **Line**, **Station**, and **Traffic** for added clarity and organization, but they're not required to build the map view.

Line	Line Group (Path ID)	Order of Points	Station	Latitude	Longitude	Traffic
1	1	1	La Défense (Grande Arche)	48.891934	2.237883	14,275,382
1	1	2	Esplanade de la Défense	48.887843	2.250442	9,843,051
1	1	3	Pont de Neuilly	48.884509	2.259892	6,902,931
10 BOUCLE	10	1	Boulogne-Jean-Jaurès	48.842222	2.238836	3,847,782
10 BOUCLE	10	2	Porte d'Auteuil	48.848074	2.258648	687,237
10 BOUCLE	10	3	Michel-Ange-Auteuil	48.847740	2.264297	2,222,709

The Line Group (Path ID) column

For each path, there is a unique key or string, which is added to every location in that path.

In this example, the Line Group (Path ID) column is used to identify each unique path. You will use this column to create your spider map.

For example, in the table above, there are two metro lines (1 and 10 Boucle), and each of those metro lines have a unique path ID listed in the **Line Group (Path ID)** column. For metro line 1, the Line Group is 1. For metro line 10 Boucle, the Line Group is 10. Every location in line one contains the line group 1, and every location in line 10 Boucle contains the line group 10.

The Order of Points column

This column defines the point order and the direction in which the line is drawn from the first data point to the last data point on your map, which, in this example is the start and end of the metro line. This field is very important if you have more than two locations in a path, because it draws your paths in the desired order (you can think of it like connecting the dots).

In the example table above, there are three stations on line 1. They are listed in order from La Défense (Grande Arche) to Pont de Neuilly using the numbers **1** through **3** in the **Order of Points** column. The same is true for line 10 Boucle.

Basic map building blocks:

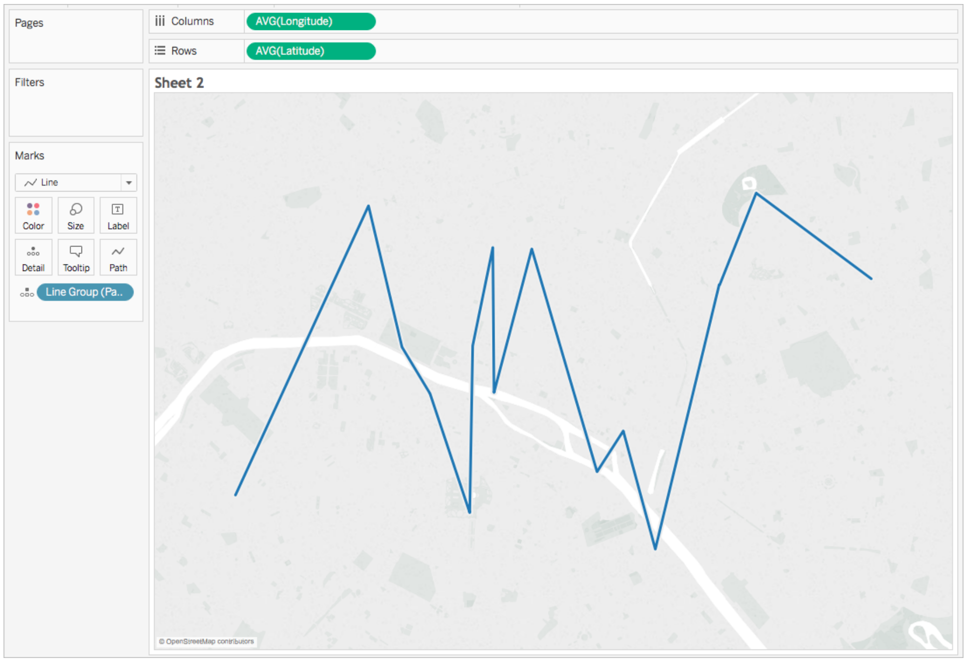
Columns shelf:	*Longitude* (continuous measure, longitude geographic role assigned)
Rows shelf:	*Latitude* (continuous measure, latitude geographic role assigned)
Detail:	*Path ID field* (discrete dimension)
Path:	*Order field* (continuous dimension)
Mark type:	*Line*

Build the map view

To follow along with this example, download the Create Spider Maps in Tableau Example 1 Workbook from Tableau Public, and open it in Tableau Desktop.

1. In the **Create Spider Maps in Tableau Example Workbook**, click the New Worksheet icon.
2. In the new worksheet, from Measures, drag **Longitude** to the **Columns** shelf, and **Latitude** to the **Rows** shelf.
3. From Dimensions, drag **Line Group (Path ID)** to **Detail** on the Marks card.
4. On the Marks card, click the Mark Type drop-down and select **Line**.

   The map view updates with a line connecting all of the points. The Path button should appear on the Marks card.

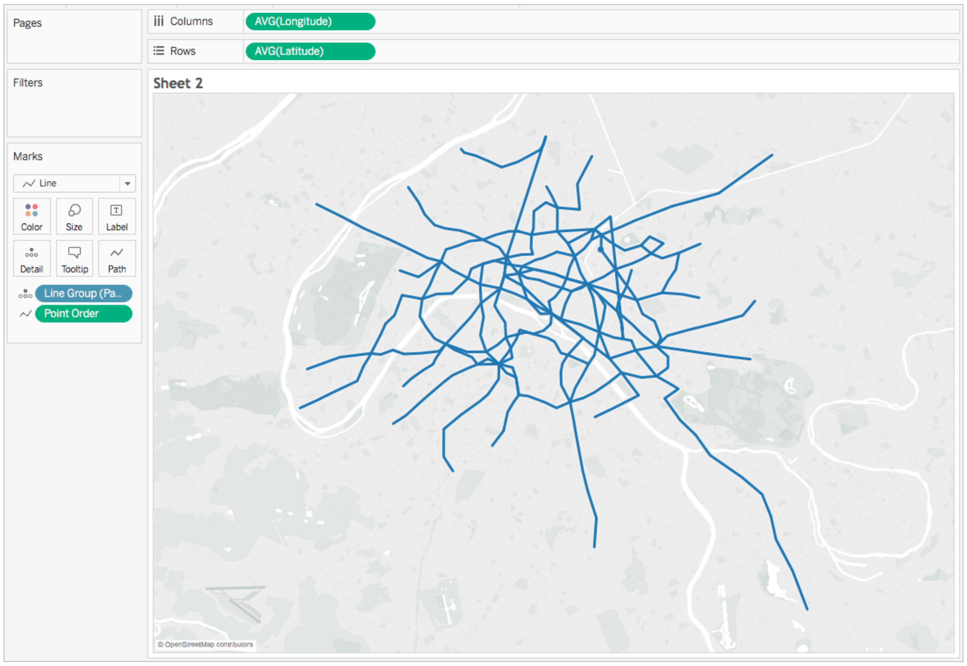


5. From Measures, drag **Point Order** to **Path** on the Marks card.

   Point Order is aggregated as a sum.

6. On the Marks card, right-click the **SUM(Point Order)** field and select **Dimension**.

   The map view updates with a line for every metro line.

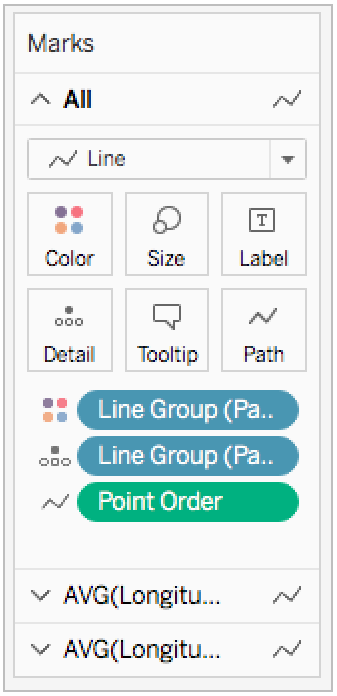


7. From Dimensions, drag **Line Group (Path ID)** to **Color** on the Marks card.

   Each line now has its own color associated with it, and a color legend is added to the view.

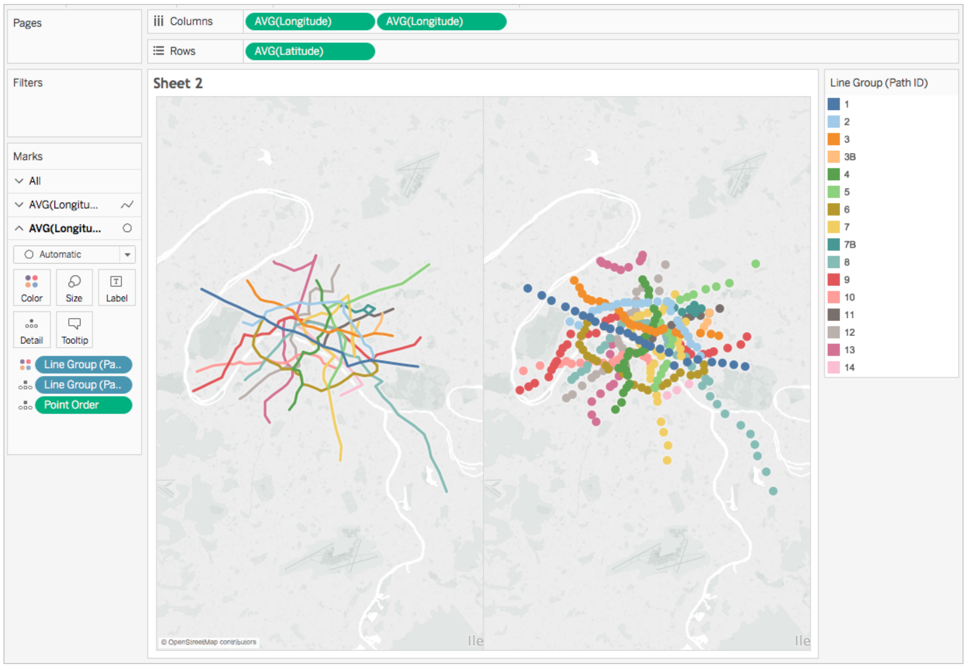
8. From Measures, drag **Longitude** to the **Columns** shelf and place it to the right of the first Longitude field.

   There are now two Longitude fields on the columns shelf. The view updates with two identical maps. The Marks card updates with two tabs: one for the map on the left, and one for the map on the right. You can customize each of these tabs to change the visual detail of each map view. There is an All tab to control the visual detail in both maps at the same time.

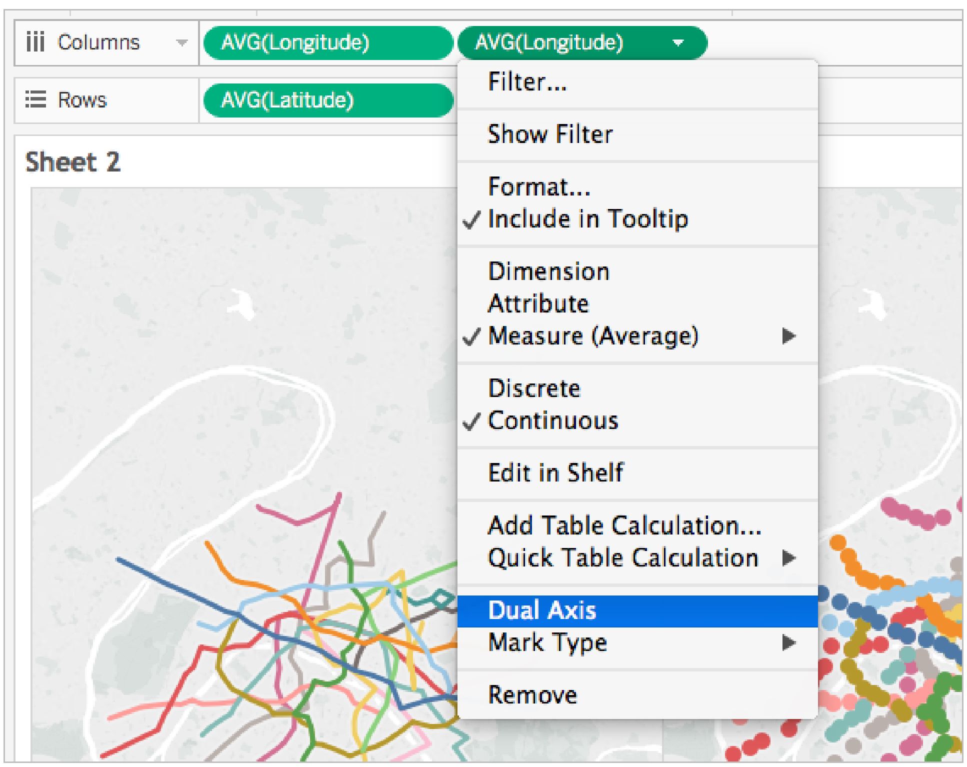


9. On the Marks card, click the bottom **AVG(Longitude)** tab.
10. On the Marks card, under the bottom AVG (Longitude) tab, click the Mark type drop-down and select **Automatic**.

    The map view on the right updates to be a point map.



11. On the Columns shelf, right-click the second **AVG (Longitude)** field (on the right), and select **Dual Axis**.



Your map views are now layered on top of one another.

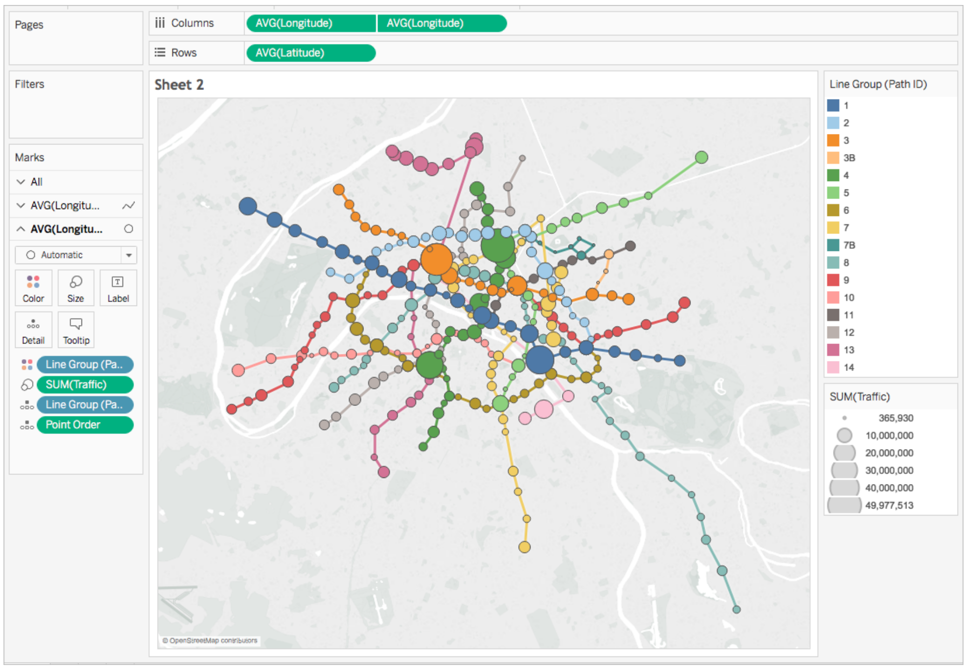
12. From Measures, drag **Traffic** to **Size**, on the bottom AVG (Longitude) Marks card.

    The size of the data points update to show the amount of traffic per station.

13. On the Marks card, click **Size** and move the slider to the right.

14. On the Marks card, click **Color**, and then, under Effects, click the **Border** drop-down and select a color.

    The view is now complete. You can quickly find the stations on each metro line with the most traffic.

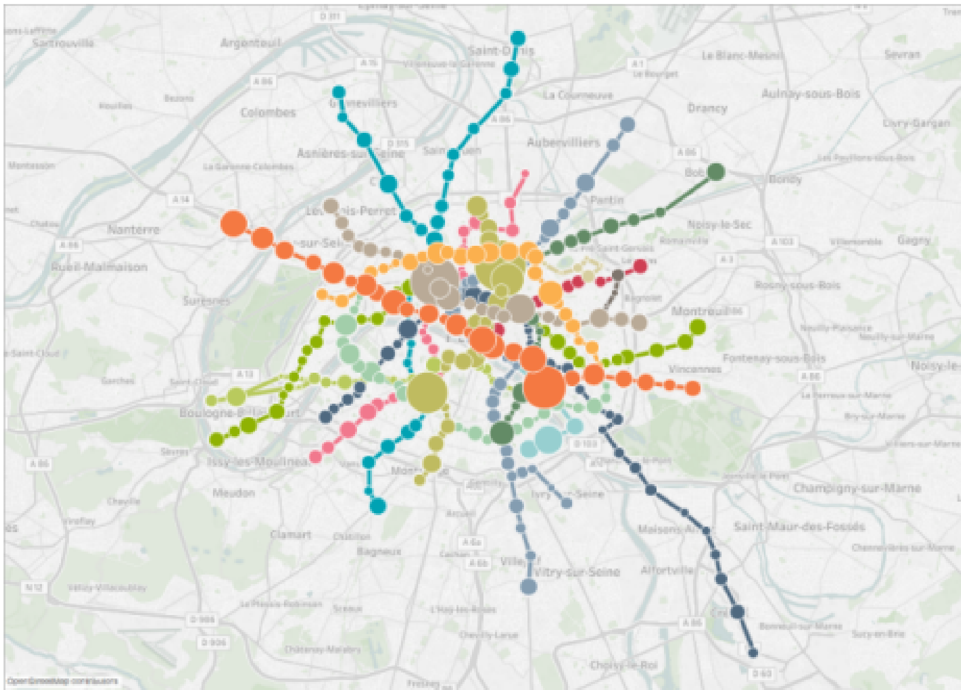


Filter the amount of information in the view

If you want to filter the amount of lines you see in the view:

- From Dimensions, drag **Line Group** to the Filters shelf.

  You can also right-click the **Line Group** field and select **Show Filter**, to show a filter card in the view.



Example 2: Bike share data in Seattle, Washington

Learn how to build it!

Your data source

> **Note**: Starting in Tableau version 10.4, you can connect to spatial files that contain linear geometries. If you have spatial data with linear geometries, you may not need to perform the steps below. To learn how to create a flow map using spatial data with linear geometries, see **Create Tableau Maps from Spatial Files** on page 1855

Similar to the first example, for this type of spider map, your data source should contain the following information:

- A Path ID for every unique path. See **The Path ID column** on page 1928 below for more information.
- Latitude and longitude coordinates for every location. See the example table below.

For this type of spider map, there should also be two rows in your data source for each path. That means you should have one row for your origin location data, and one row for your destination location data, repeated for every path. This is a crucial step that enables Tableau to draw your paths correctly. See **Origin-destination rows** below below for more information.

It's also recommended that your data source contain a column with your location names, but it is not required.

This example is a snippet of the Seattle bike share data source, which is included in the Create Spider Maps in Tableau Example 2 Workbook. It has a column for **Origin-Destination**, **Station**, **Path ID**, **Latitude**, and **Longitude**. Only the last three columns in this data source are required to create a spider map, but the Origin-Destination and Location Name columns provide further clarity and organization.

Origin-Destination	Station	Path ID	Latitude	Longitude
Origin	BT-01	BT-01_BT-01	47.61841	-122.35101
Destination	BT-01	BT-01_BT-01	47.61841	-122.35101
Origin	BT-01	BT-01_BT-03	47.61841	-122.35101
Destination	BT-03	BT-01_BT-03	47.61576	-122.34843
Origin	BT-01	BT-01_BT-04	47.61841	-122.35101
Destination	BT-04	BT-01_BT-04	47.61613	-122.34108
Origin	BT-01	BT-01_BT-05	47.61841	-122.35101
Destination	BT-05	BT-01_BT-05	47.61303	-122.34410

Origin-destination rows

For each unique path you want to create, you need a row for your origin location, and a row for you destination location in your data source. This means your origin location will be paired with every destination location.

For example, when showing the path between an origin bike share location and several destination locations in a city, you need a row for the origin location, and a row for the destination location for *every single path*.

In the example above, the origin station BT-01 is paired with several different destination locations (BT-01, BT-03, BT-04, BT-05) to show that bikes were checked out of the BT-01 location and returned either to the same location or to a different location. Each origin-destination pair is color-coded to show that they make up one path. There is a column for Origin-Destination to further illustrate this concept, but this column is not required.

The Path ID column

The Path ID column is used to identify each unique origin-to-destination path. You will use this column to create your spider map.

For each origin and destination location, there is a unique key or string that identifies them as a pair.

In the example below, for the first origin-destination path, the Path ID is BT-01_BT-01. For the second origin-destination path, the Path ID is BT-01_BT-03. Each path ID is listed twice, once for the origin location, and once for the destination location. Again, each pair is color-coded to indicate that they make up one path.

Origin-Destination	Location Name	Path ID	Latitude	Longitude
Origin	BT-01	BT-01_BT-01	47.61841	-122.35101
Destination	BT-01	BT-01_BT-01	47.61841	-122.35101
Origin	BT-01	BT-01_BT-03	47.61841	-122.35101
Destination	BT-03	BT-01_BT-03	47.61576	-122.34843

Note: Your path ID can be anything you want. However, if you want to create complex calculated fields to help filter the locations later on, it helps if your path IDs are consistent across all paths. A great way to do this is to create Path IDs that are a combination of your origin and destination location names, separated by a delimiter. For example, the Path ID for the origin location BT-01 and the destination location BT-03 is BT-01_BT-03. For an example of why you might want to do this, see the **Option 2: Create a dynamic filter:** on page 1932 section.

Basic map building blocks:

Columns shelf:	*Longitude* (continuous dimension, longitude geographic role assigned)
Rows shelf:	*Latitude* (continuous dimension, latitude geographic role assigned)

Detail:	*Path ID field* (discrete dimension)
Mark type:	*Line*

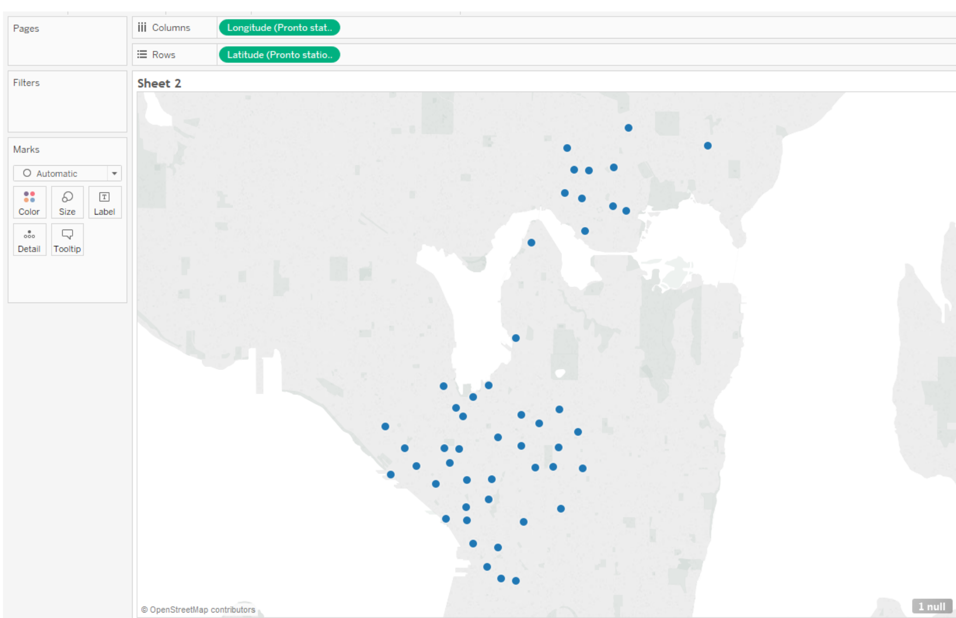
Build the map view

After you set up your data source, you can connect to it in Tableau Desktop and build a spider map. To follow along with this example, download the Create Spider Maps in Tableau Example 2 Workbook from Tableau Public.

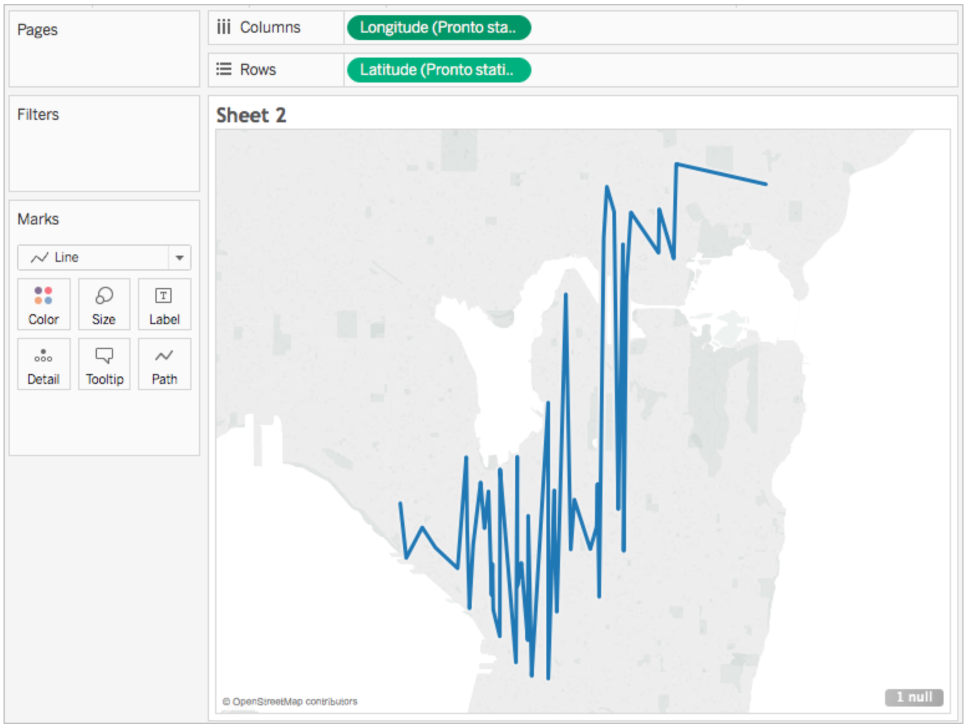
1. From Measures, drag **Longitude** to the **Columns** shelf, and **Latitude** to the **Rows** shelf.
2. On the Columns shelf, right-click the **Longitude** field and select **Dimension**. Do the same for the **Latitude** field on the **Rows** shelf.

   This will ensure Tableau does not aggregate your origin and destination locations.

   You should see all the locations in the data source as data points on your map. In this example, there is a data point for every bike share location in the data source.

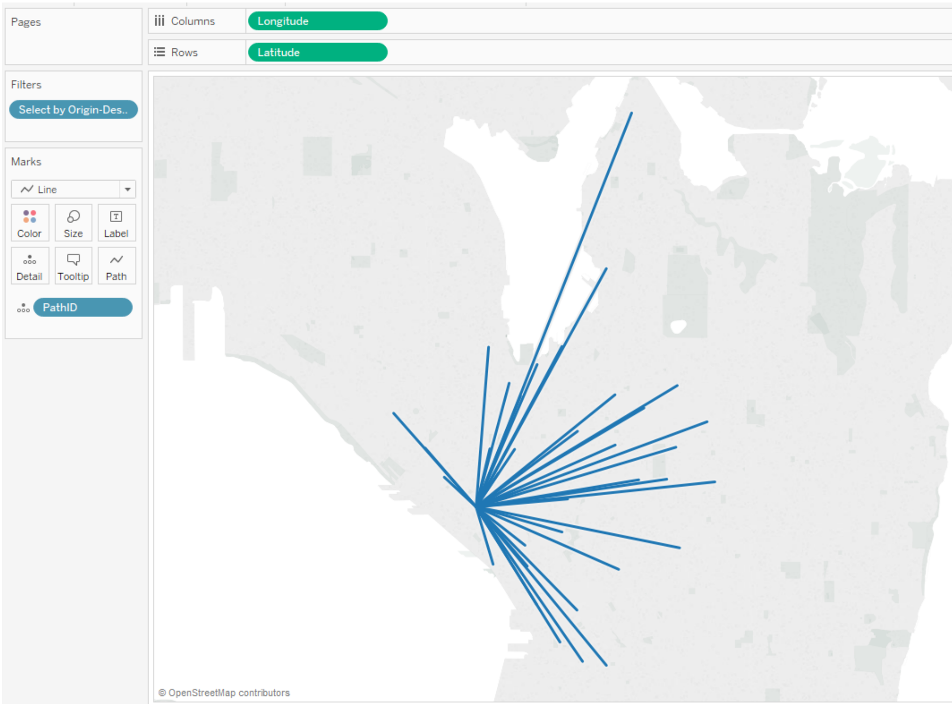
   

3. On the Marks card, click the Marks Type drop-down, and select **Line**. The view updates to show a line connecting each data point, and the Marks card updates with a Path button.

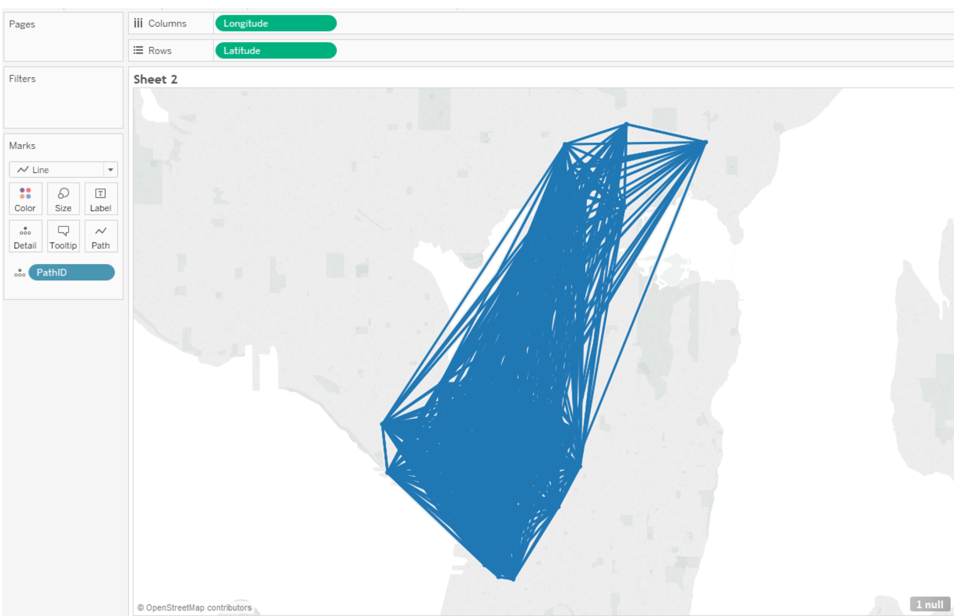


4. From Dimensions, drag **Path ID** to **Detail** on the Marks card.

   If you have only a few origin-destination pairs, your view might look something like the following:



However, if you have many origin-destination pairs, your view might look more like this:



This is very common, and can be fixed by filtering a large portion of your paths from the view. Continue to the following section to learn a couple of ways to do so.

Filter the amount of information in the view

If your data source contains a lot of origin-destination pairs, you can filter most of them from the view.

Option 1: Create a simple filter:

1. From Dimensions, drag **Path ID** to the Filters shelf.
2. In the Filter dialog box that opens, do the following:
    - Under the **General** tab, select **None**.
    - Click the **Wildcard** tab.
    - Under the **Wildcard** tab, for **Match Value**, enter **BT-01**, and then click **Starts With**.
    - Click **OK**.

      This filters the view down to only the paths that start with BT-01.

Option 2: Create a dynamic filter:

You can also create a calculated field and combine it with a parameter so you can switch between the paths you want to see right in the view. Follow the steps below to learn how.

Step 1: Create the parameter

1. On the Data pane, to the right of Dimensions, click the Data pane drop-down and select **Create Parameter**.
2. In the Create Parameter dialog box, do the following:
    - Name the parameter **StationSelected**.
    - For **Data type**, select **String**
    - For **Allowable values**, click **List**, select **Add from field**, and then select the **Location Names** field.
    - Click **OK**.

Step 2: Create the calculated field

1. Select **Analysis** > **Create Calculated Field**.
2. In the calculation editor, name the calculated field, **Select by Origin-Destination**, and then enter the following formula:

```
IF

 LEFT([Path ID], FIND([Path ID], "_") -1) =
[StationSelected] THEN "Origin"

ELSEIF

 RIGHT([Path ID], LEN([Path ID]) - FIND([Path ID], "_"))
= [StationSelected] THEN "Destination"

ELSE

 "Unselected stations"
END
```

In the table above, each Path ID (for example BT-01_BT-03) contains an underscore (_) as a delimiter to separate the origin location name (BT-01) from the destination location name (BT-03). This delimiter is used in the formula to tell Tableau which locations (selected in the parameter you created in step 1 of this procedure) are origin locations, and which are destination locations. The parameter you created above is also used in the formula (StationSelected).

3. Check that the calculation is valid, and then click **OK**.

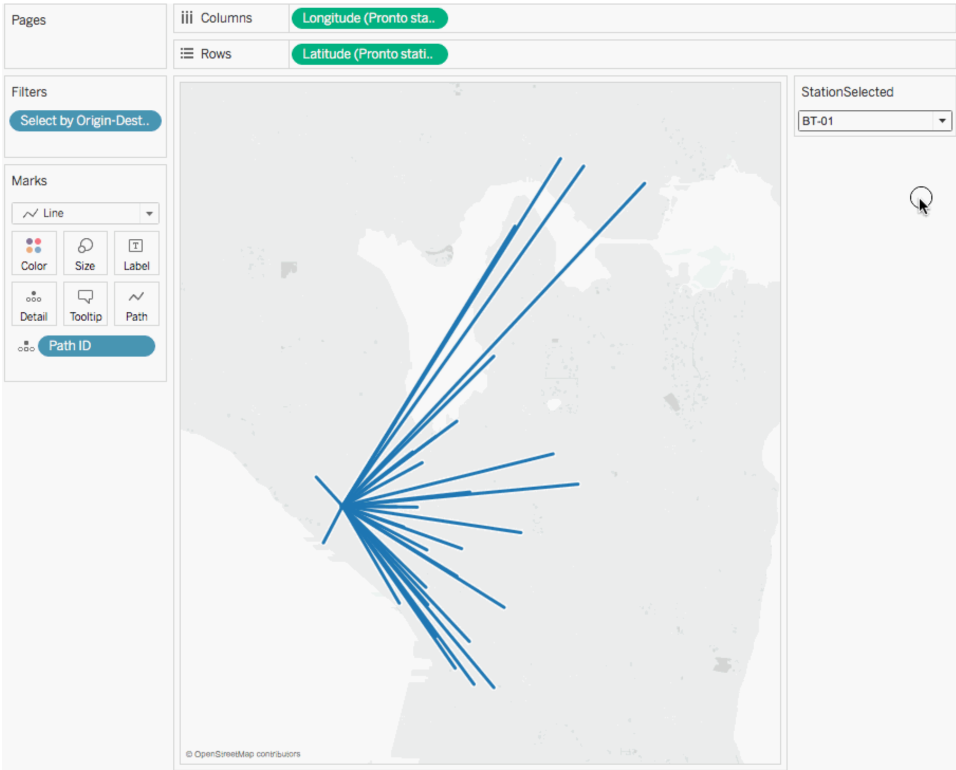
Step 3: Add the calculated field to the Filters shelf

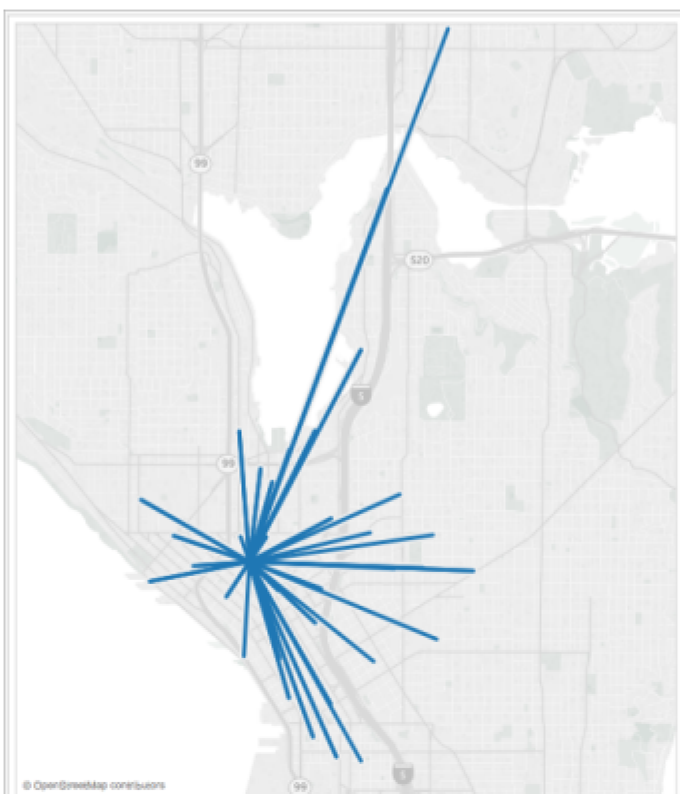
1. From Dimensions, drag the calculated field to the **Filters** shelf.
2. In the Filter dialog box, select **Origin**, and then click **OK**.

Step 4: Show the parameter control

In the Data pane, under Parameters, right-click the parameter you created and select **Show Parameter Control**.

You can now select an origin from the parameter control in the view, and the marks on your map view will update.


Pages
Columns
Longitude (Pronto sta..
Rows
Latitude (Pronto stati..
Filters
Select by Origin-Dest..
StationSelected
BT-01
Marks
Line
Color
Size
Label
Detail
Tooltip
Path
Path ID
© OpenStreetMap contributors


© OpenStreetMap contributors

See also:

Tableau Community post: Origin-Destination Maps (or Flow Maps)

Create Maps that Show a Path Over Time in Tableau on page 1903

Mapping Concepts in Tableau on page 1781

Create Parameters on page 1218

Filter Data from Your Views on page 1411

Create Dual-Axis (Layered) Maps in Tableau

This article demonstrates how to create dual axis (layered) maps in Tableau using several examples. A dual-axis map is a map with two sets of geographic data overlaid on top of one another. For example, a filled map of U.S. states with data points for each city layered on top.

There are three ways to create a dual-axis map in Tableau:

- By using Tableau Latitude(generated) and Longitude(generated) fields
- By using custom latitude and longitude fields
- By using a combination of Tableau Latitude(generated) and Longitude (generated) fields, and custom latitude and longitude fields

Follow the steps below to learn how to create a dual-axis map using each of these methods.

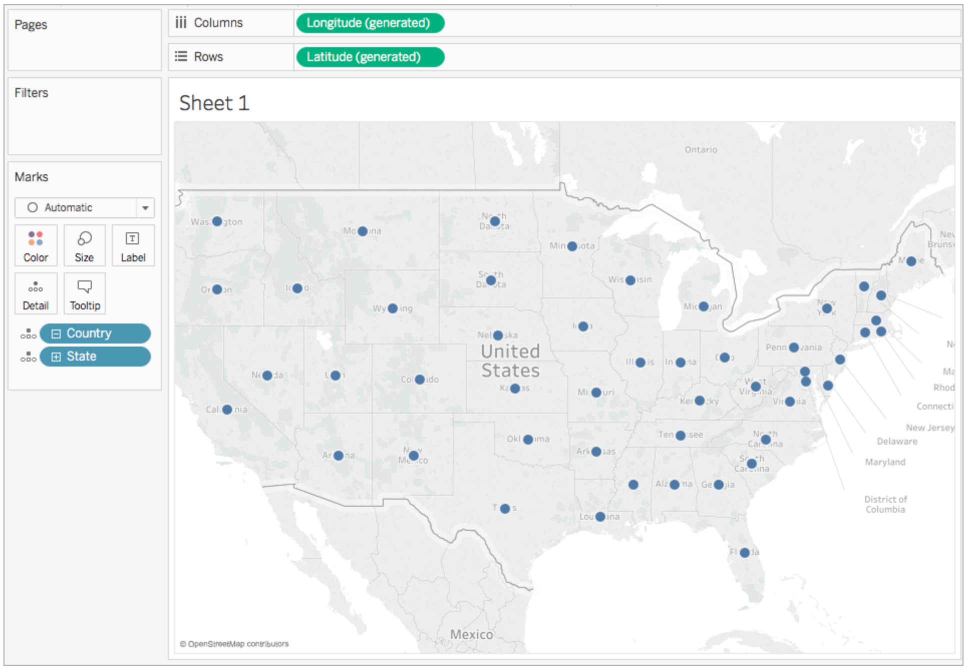
Create a dual-axis map from Tableau Latitude (generated) and Longitude (generated) fields

1. Open Tableau Desktop.

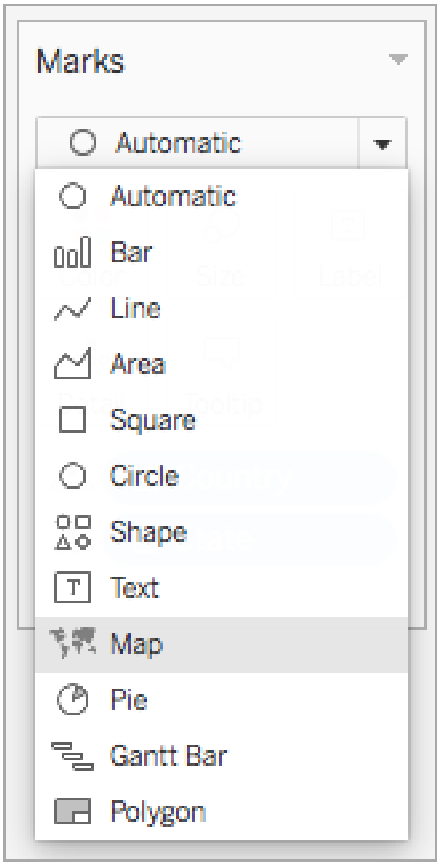
2. In the Connect pane, under Saved Data Sources, connect to the **Sample-Superstore** data source.

3. In the **Data** pane, under Dimensions, double-click **State**.

   A map view is created.

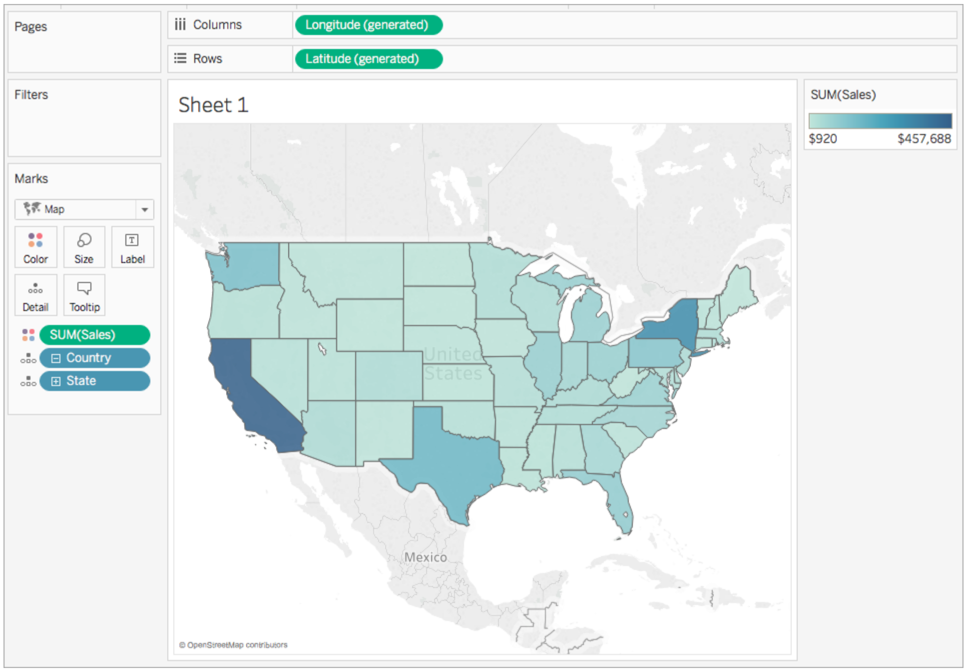


4. On the Marks card, click the Mark Type drop-down and select **Map**.

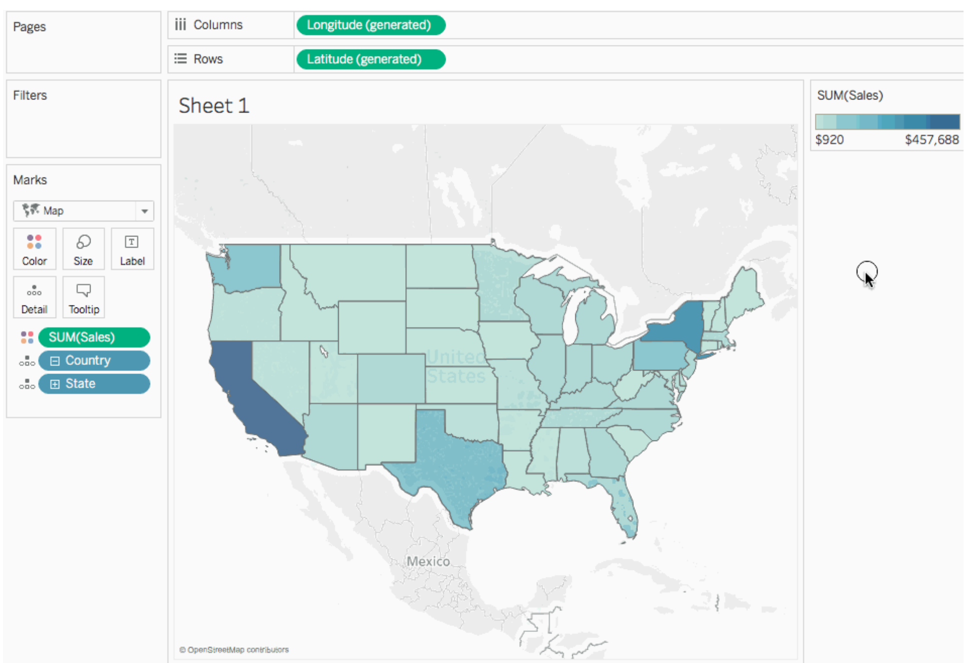


5. From the **Data** pane, under Measures, drag **Sales** to **Color** on the Marks card.

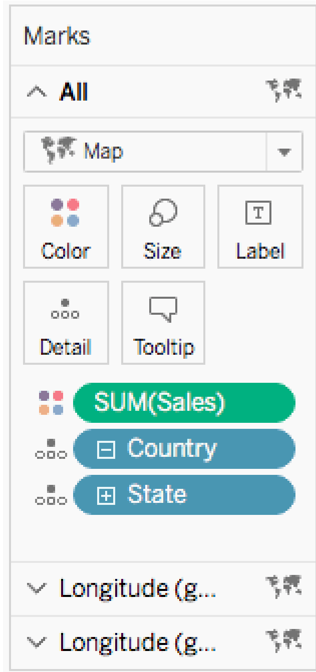
   The map updates to show the sum of sales by state. The states with higher sales are darker blue, and the states with fewer sales are lighter blue.



6. On the **Columns** shelf, control-drag (command-drag on a Mac) the **Longitude (generated)** field to copy it, and place it to the right of the first Longitude field.

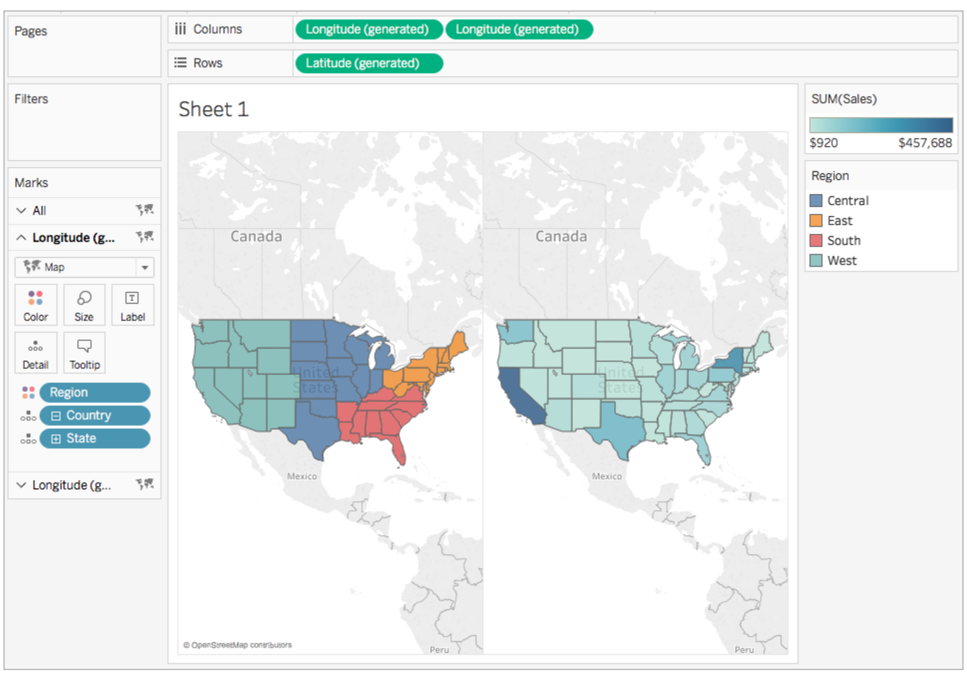


7. On the Marks card, select the top **Longitude (generated)** tab.



8. From the **Data** pane, under Dimensions, drag **Region** to **Color** on the Marks card.

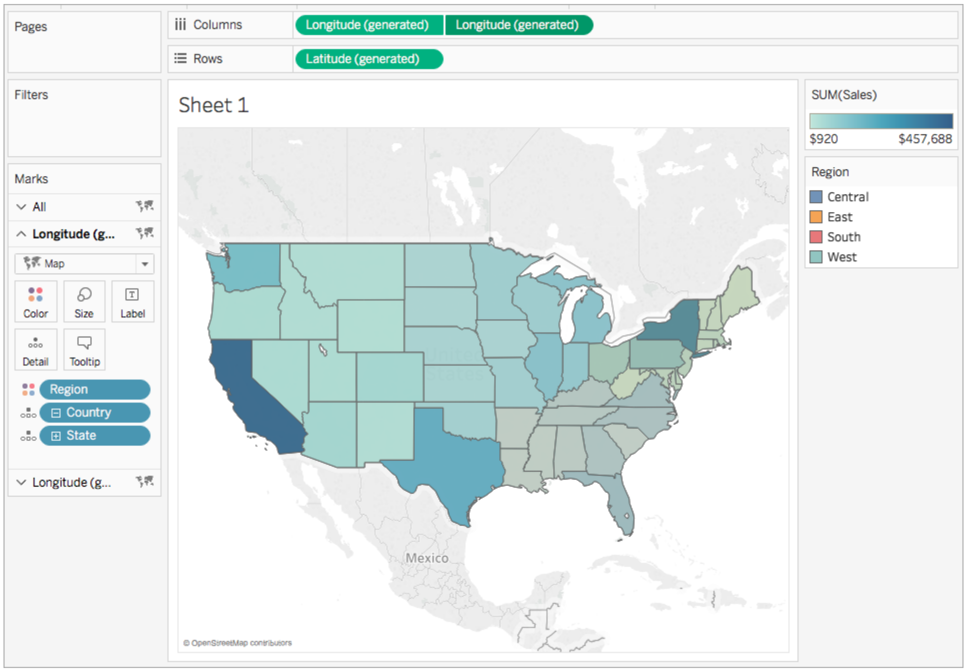
   The map view on the left updates.



9. On the Columns shelf, right-click the **Longitude (generated)** field on the right and select

Dual Axis.

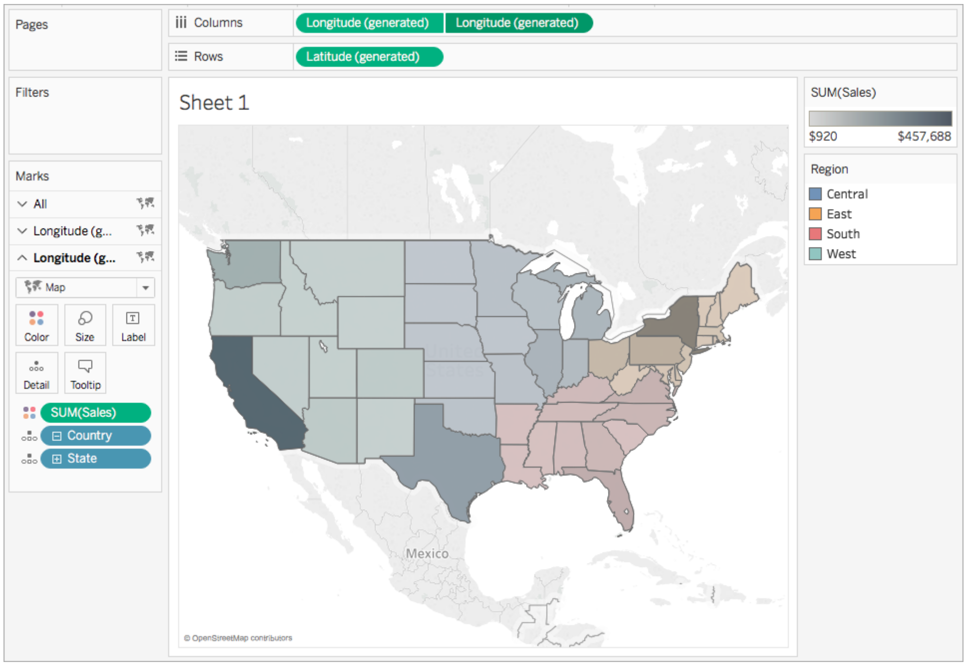
The map views are now overlapping each other. You might not be able to see the map on the bottom layer.



10. On the Marks card, ensure that the bottom Longitude (generated) tab is selected, and then click **Color** > **Edit Colors**.

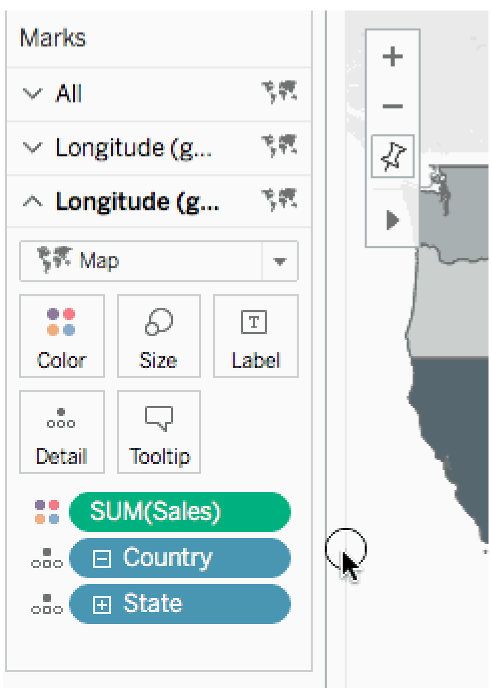
11. In the Edit Colors dialog box that opens, click the Palette drop-down, select **Gray**, and then click **OK**.

    Notice that the colors of the map update.



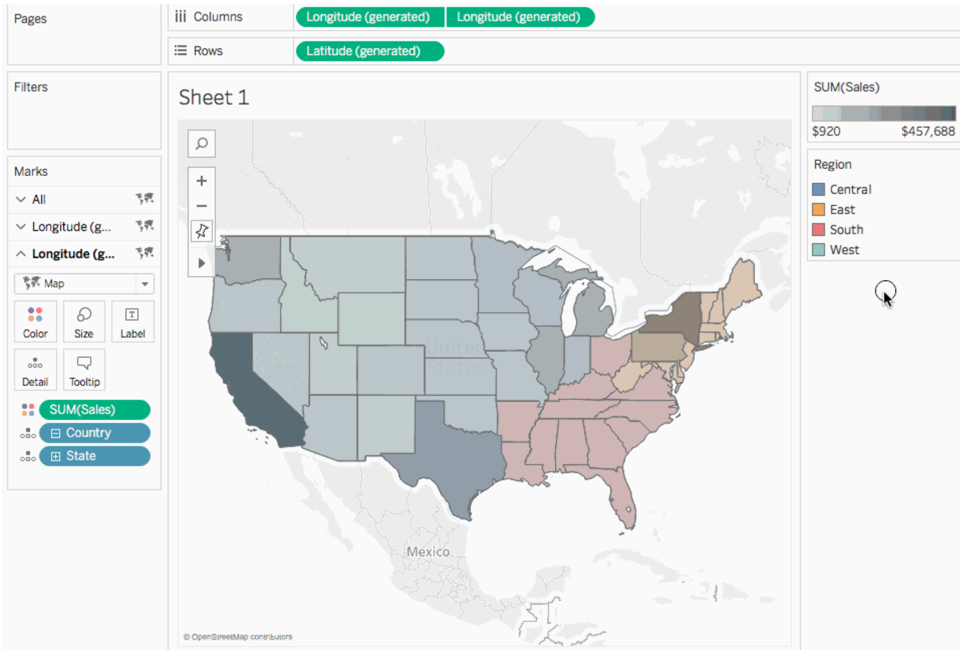
12. On the Marks card, click **Color** again.

13. In the Color pop-up dialog box, under Opacity, move the slider to approximately 75%.



The dual axis (layered) map is now complete. For each region, you can now see which states has the most sales.

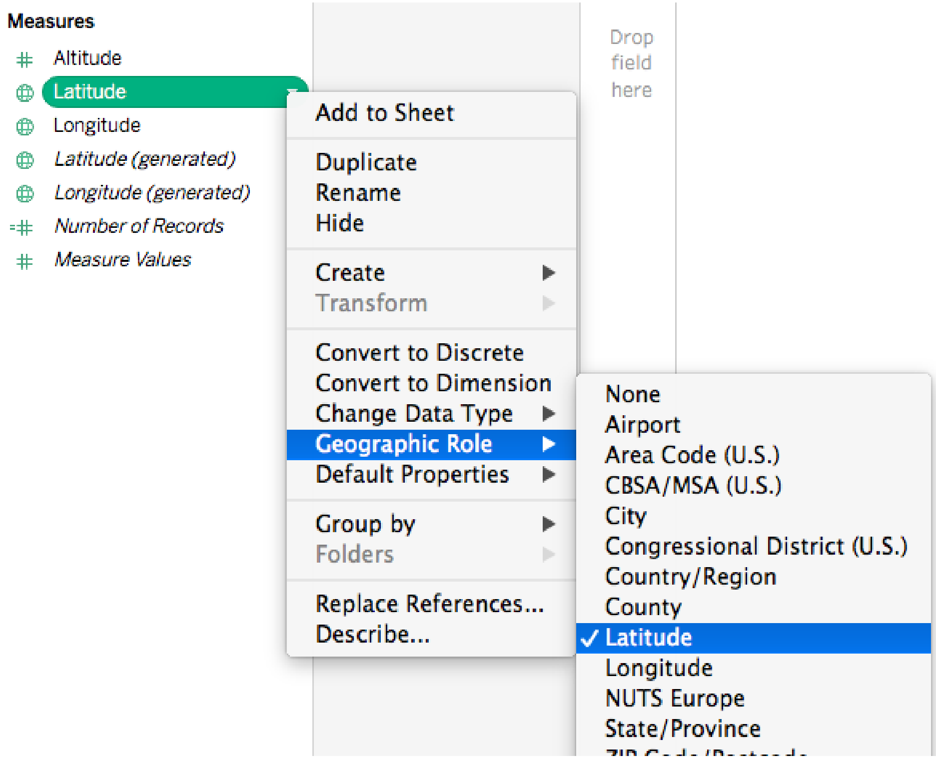
Tip: To change which map is on top, rearrange the Longitude (generated) fields on the Columns shelf.



Create a dual-axis map from custom latitude and longitude fields

If your data source contains custom latitude and longitude fields, you can use them instead of the Tableau Latitude (generated) and Longitude (generated) fields to create a dual-axis map. Follow the steps below to learn how.

1. Open Tableau and connect to a data source with custom latitude and longitude values.
2. Navigate to a new worksheet.
3. In the **Data** pane, right-click the custom latitude field and select **Geographic Role** > **Latitude**.



Note that the Latitude geographic role may already be assigned to the field.

4. In the **Data** pane, right-click the custom longitude field and select **Geographic Role** > **Longitude**.

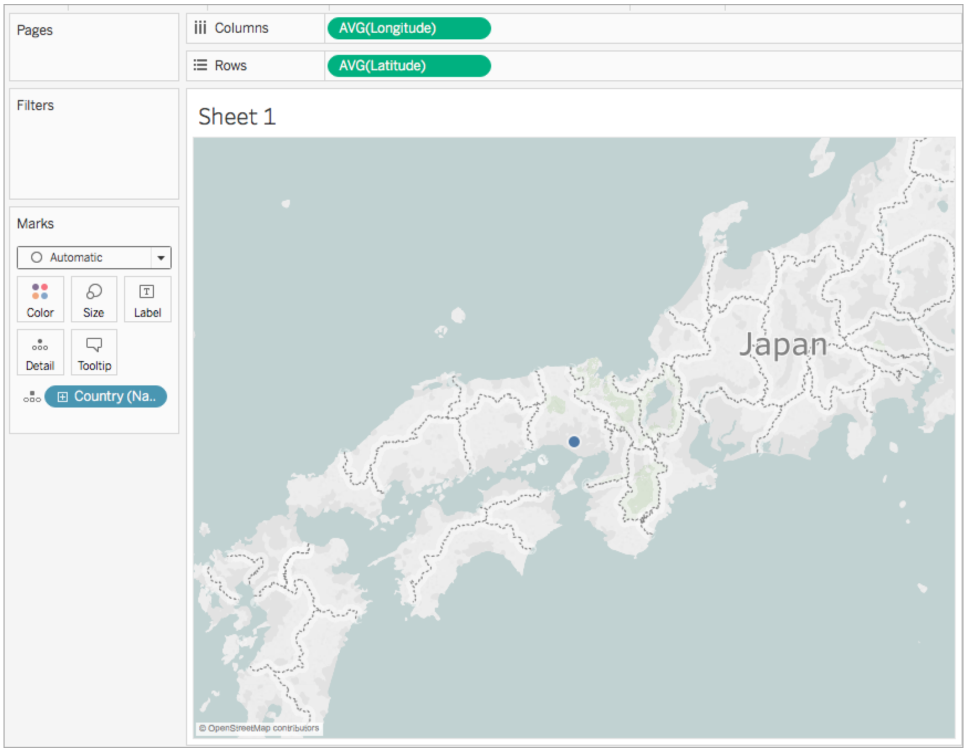
   Note that the Longitude geographic role may already be assigned to the field.

5. From the **Data** pane, drag the custom latitude field to the **Rows** shelf.

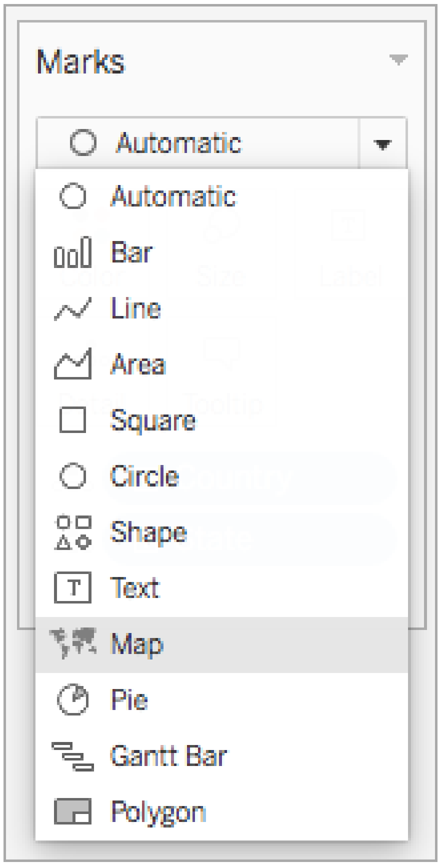
6. From the **Data** pane, drag the custom longitude field to the **Columns** shelf.

7. From the **Data** pane, under Dimensions, drag a geographic field to **Detail** on the Marks card.

   In this example, the geographic field, Country (Name), is used.



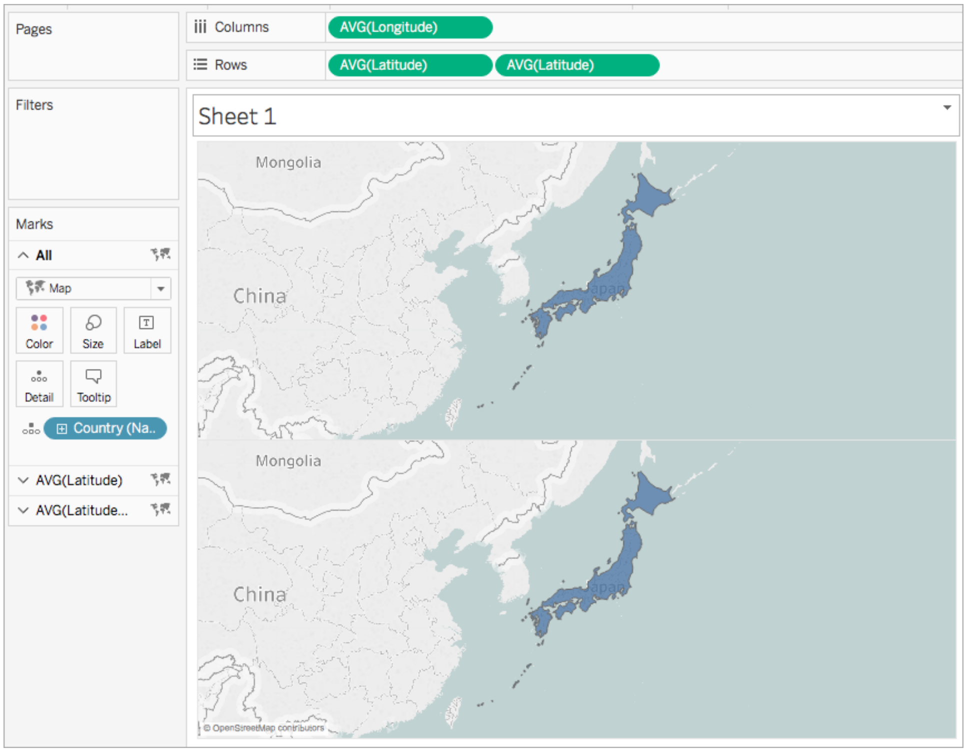
8. On the Marks card, click the Mark Type drop-down and select **Map**.



The map updates to a filled map.

9. On the Rows shelf, Ctrl-click (Command-click on a Mac) and drag the custom latitude field to the right. This copies the field.

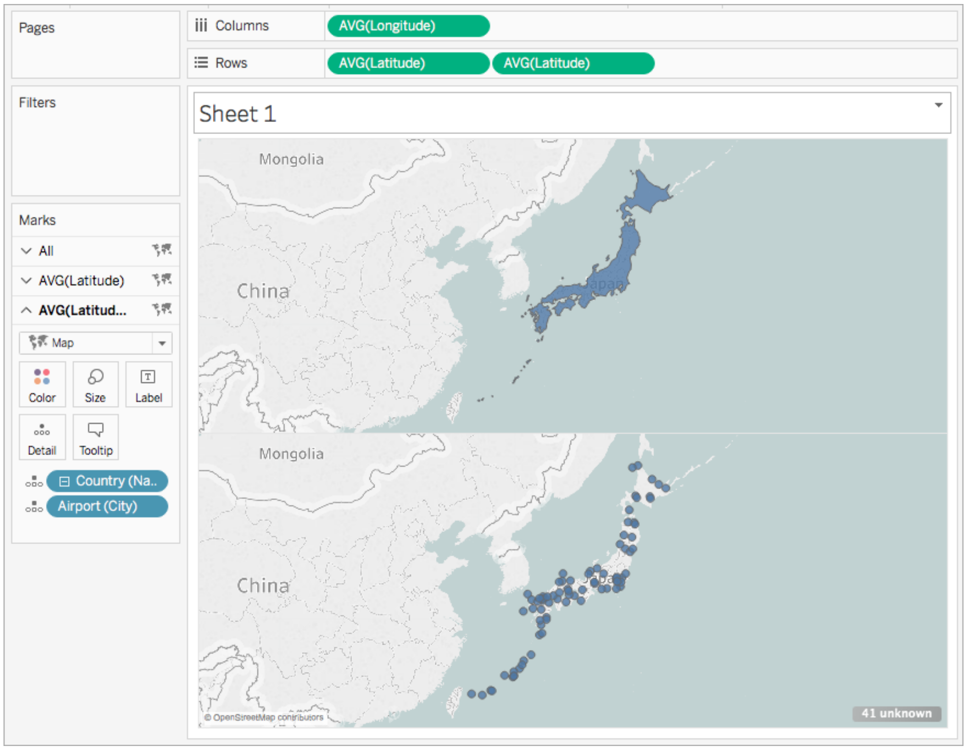
   A second, identical map is created and the Marks card updates to include three tabs. The middle tab is for the top map, and the bottom tab is for the bottom map.



10. On the Marks card, click the bottom tab and remove the geographic field by dragging it off.

11. From the **Data** pane, drag a new geographic field to **Detail** on the Marks card.

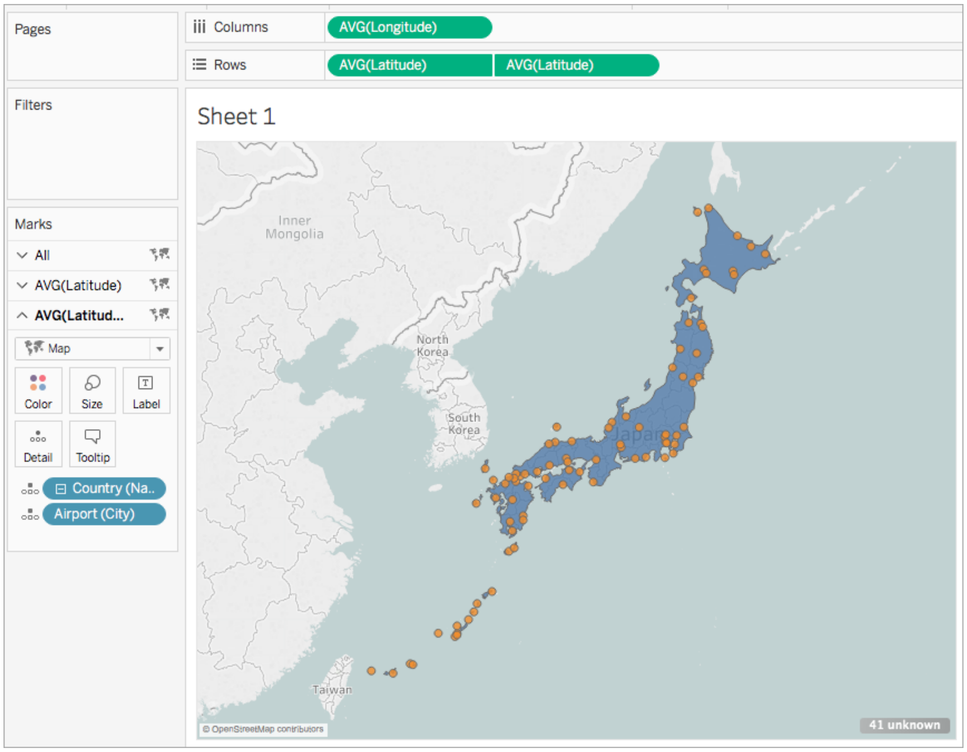
    In this example, Airport (City) is used.



12. On the Marks card, click **Color** and select a color for the marks. In this example, the color orange is used.

13. On the Rows shelf, right-click the custom latitude field on the right and select **Dual Axis**.

    The two maps are now combined.



Create a dual-axis map from a combination of generated and custom latitude and longitude fields

Follow the steps in each scenario to learn how to create a dual-axis map from generated and custom latitude and longitude fields.

Best practices for creating dual-axis maps with two sets of latitude and longitude fields:

- The Latitude(generated) and Longitude(generated) fields must be placed on the Columns and Rows shelves.
- The custom latitude and longitude fields must be placed on Detail on the Marks card, and then converted to dimensions. For more information, see steps 11 and 12 in Scenario 1.
- The custom latitude and longitude fields must be assigned the Latitude and Longitude geographic roles. For more information, see **Assign Geographic Roles** on page 1834.

- When joining data sources, use a Full Outer join. For more information, see **Join Your Data** on page 883.

Scenario 1: Use generated and custom latitude and longitude fields from a single data source

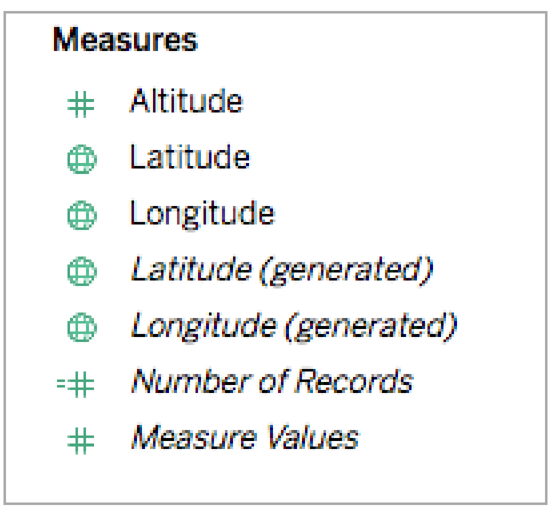
1. Open Tableau Desktop and connect to a data source.

   In this example, an excel spreadsheet that contains country and city names, as well as custom latitude and longitude columns for airport locations around the world, is used.

Abc Airports Airport (IATA)	Abc Airports Airport (Name)	Airports Airport (City)	Airports Country (Name)	Airports Latitude	Airports Longitude
GKA	GOROKA	GOROKA	PAPUA NEW GUINEA	-6.0817	145.392
LAE	*null*	LAE	PAPUA NEW GUINEA	0.0000	0.000
MAG	MADANG	MADANG	PAPUA NEW GUINEA	-5.2069	145.789
HGU	MOUNT HAGEN	MOUNT HAGEN	PAPUA NEW GUINEA	-5.8261	144.296
LAE	NADZAB	NADZAB	PAPUA NEW GUINEA	-6.5697	146.726
POM	PORT MORESBY JAC...	PORT MORESBY	PAPUA NEW GUINEA	-9.4433	147.220
RAB	*null*	RABAUL	PAPUA NEW GUINEA	0.0000	0.000

2. Navigate to a new worksheet.

   Notice that, in the Data pane, under Measures, there are two sets of latitude and longitude fields: the custom latitude and longitude fields from the data source, and the generated latitude and longitude fields that Tableau creates from your geographic fields.

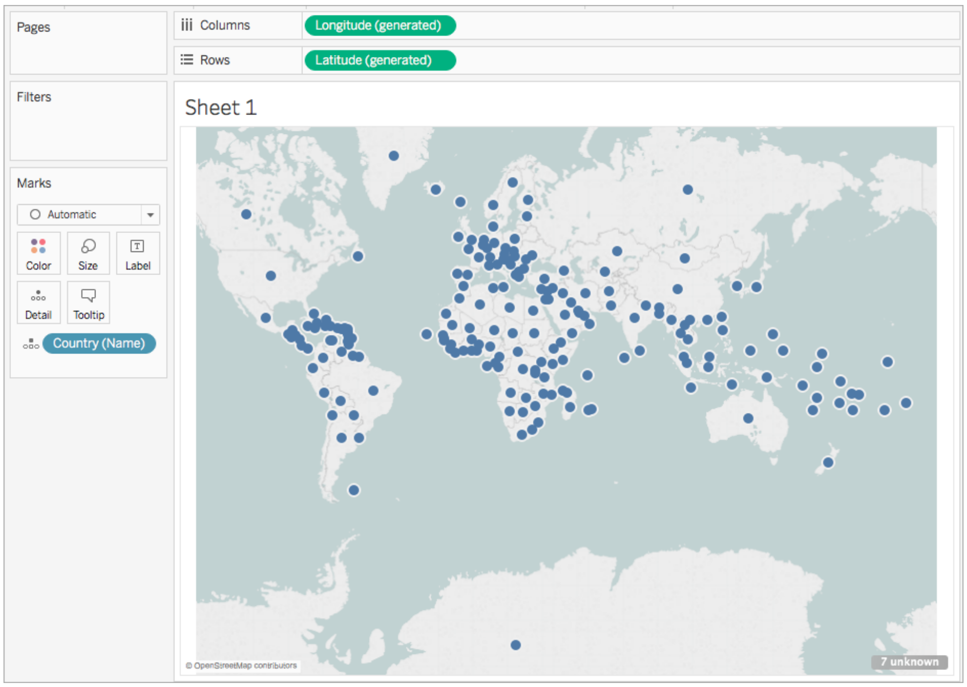


3. From the **Data** pane, under Measures, drag **Longitude(generated)** to the **Columns** shelf.

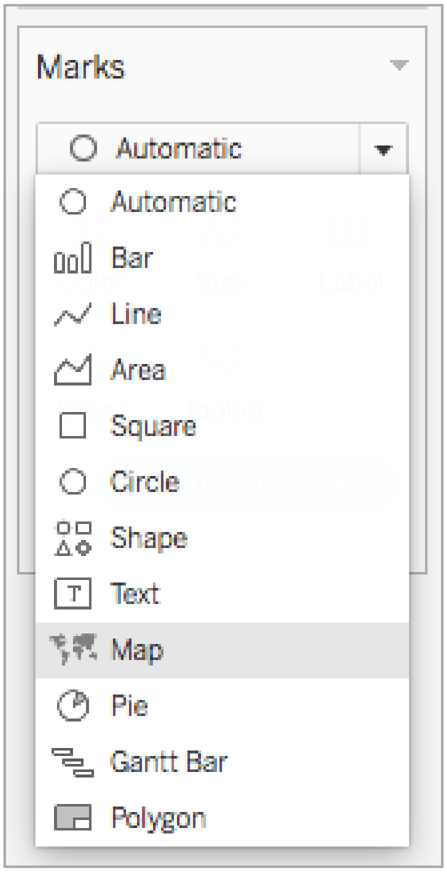
4. From the **Data** pane, under Measures, drag **Latitude(generated)** to the **Rows** shelf.

5. From the **Data** pane, under Dimensions, drag a geographic location to **Detail** on the Marks card.

   In this example, **Country (Name)** is used. A map view with a data point for every country in the data source is created.

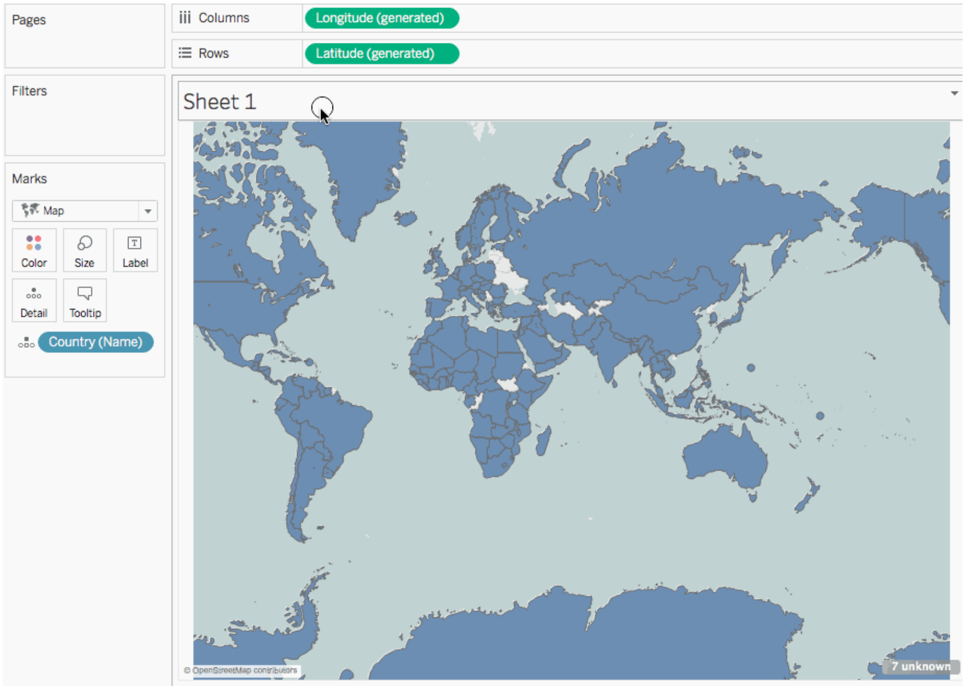


6. On the Marks card, click the Mark type drop-down and select **Map**.

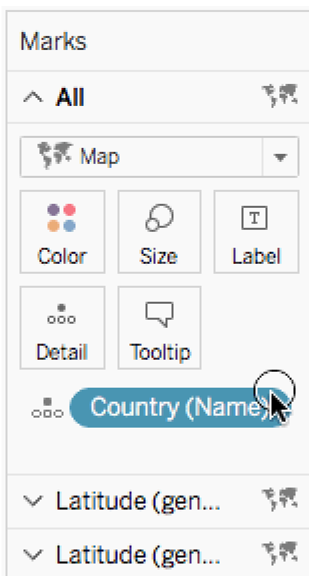


7. On the Rows shelf, Ctrl-click (Command-click on Mac) and drag the **Latitude (generated)** field to the right. This copies the field.

   A second, identical map is created and the Marks card updates to include two tabs labeled Latitude(generated). The top tab is for the top map, and the bottom tab is for the bottom map.



8. On the Marks card, click the bottom **Latitude(generated) tab** and remove **Country (name)**.



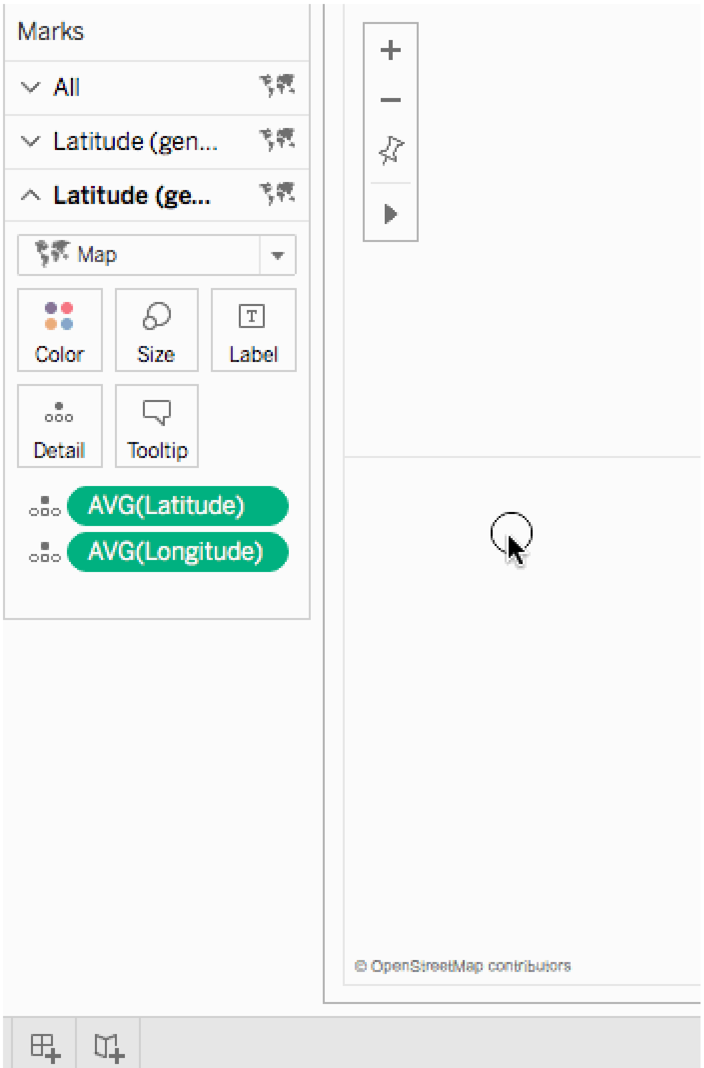
9. From the Data pane, under Measures, drag the custom **Latitude** field to **Detail** on the Marks card.

10. From the Data pane, under Measures, drag the custom **Longitude** field to **Detail** on

the Marks card.

11. On the Marks card, right-click the custom **Latitude** field and select **Dimension**.

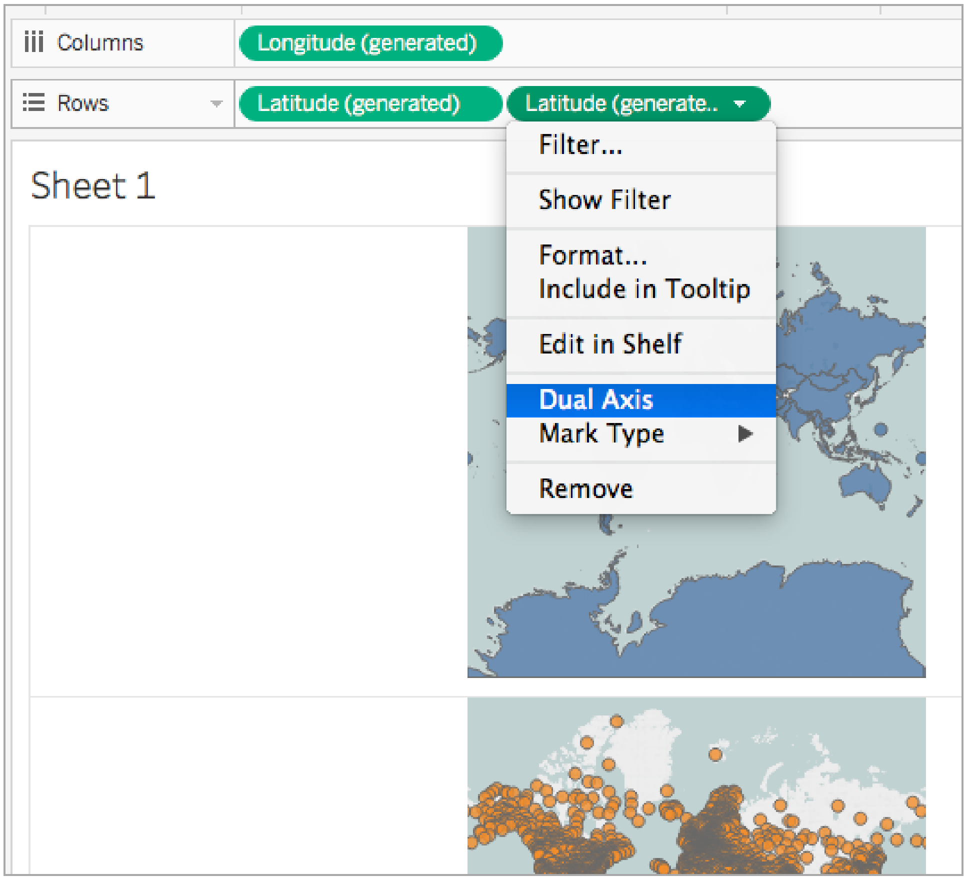
12. Repeat step 11 for the custom **Longitude** field.



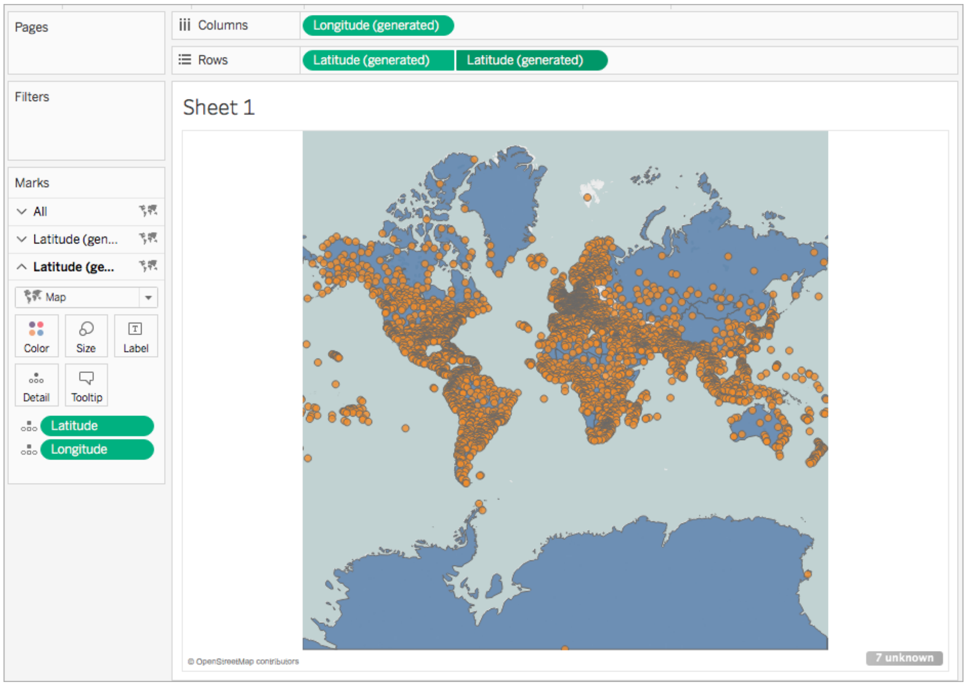
13. On the Marks card, click **Color**, and then select a new color.

    In this example, the color orange is used.

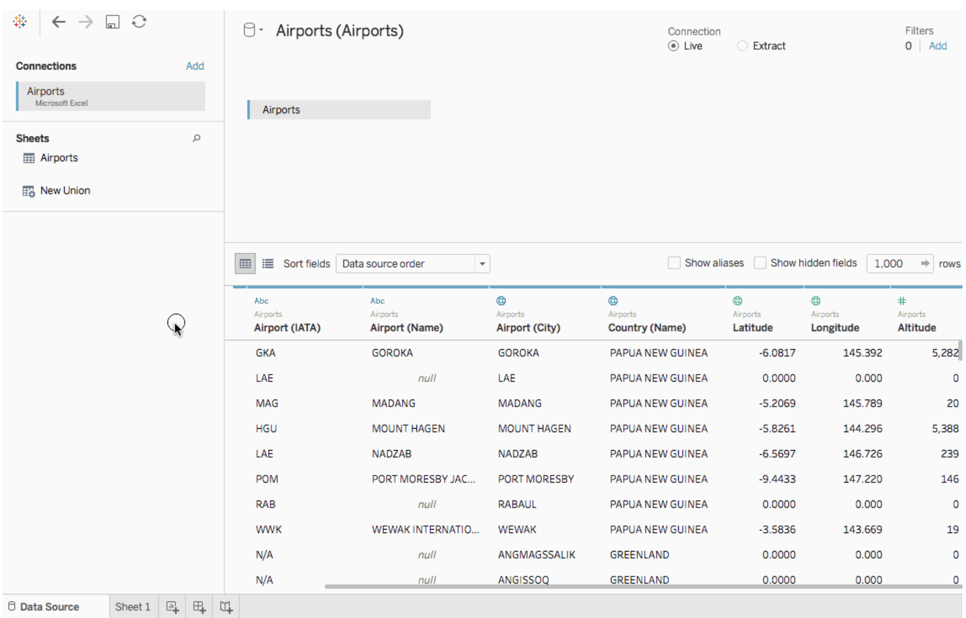
14. On the Rows shelf, right-click the **Latitude(generated)** field on the right and select **Dual Axis**.



The two maps are now combined. You can filter the view as needed or zoom in to a particular area.



Check your work! Watch steps 2 - 14 below:



Scenario 2: Join spatial data with an independent data source and plot the data from both on the same map

1. Open Tableau Desktop and connect to spatial data. For more information, see **Connect to spatial files** on page 1855.

   In this example, a shape file with polygon data for U.S. states is used.

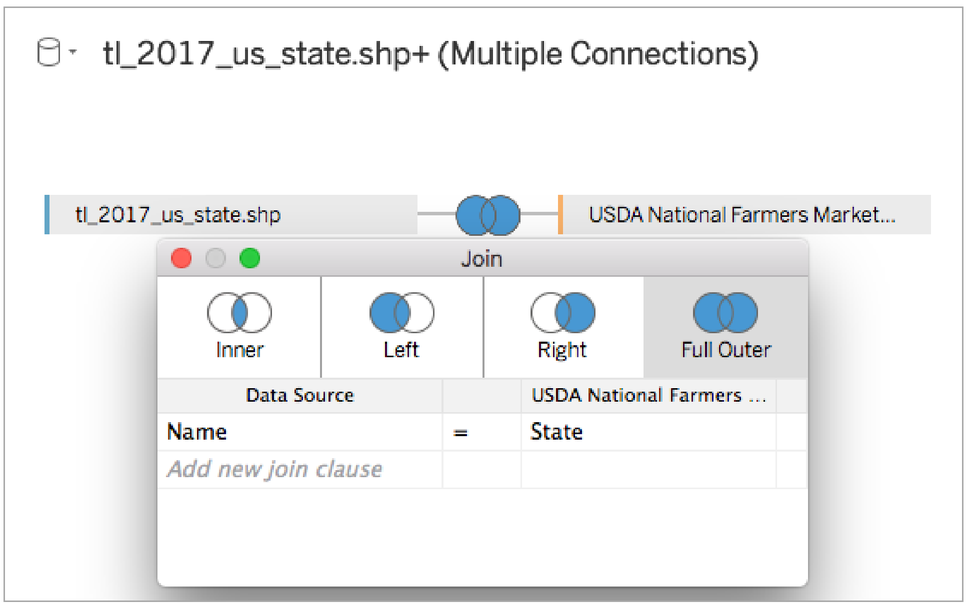
2. On the Data Source page, in the Connections pane, click **Add**.

3. Connect to another data source.

   In this example, an excel spreadsheet with custom latitude and longitude columns is used.

4. Join the new data source to the spatial data using a Full Outer join. For more information on how to join data, see **Join Your Data** on page 883.

   In this example, the excel spreadsheet is joined to the shape file using columns that contain state names from each data source.



5. Navigate to a new worksheet.

   Notice that, in the Data pane, there are two sets of Dimensions and two sets of Measures: one for each of your data sources.

6. From the **Data** pane, under Measures, drag **Latitude(generated)** to the **Rows** shelf.

7. From the **Data** pane, under Measures, drag **Longitude(generated)** to the **Columns**

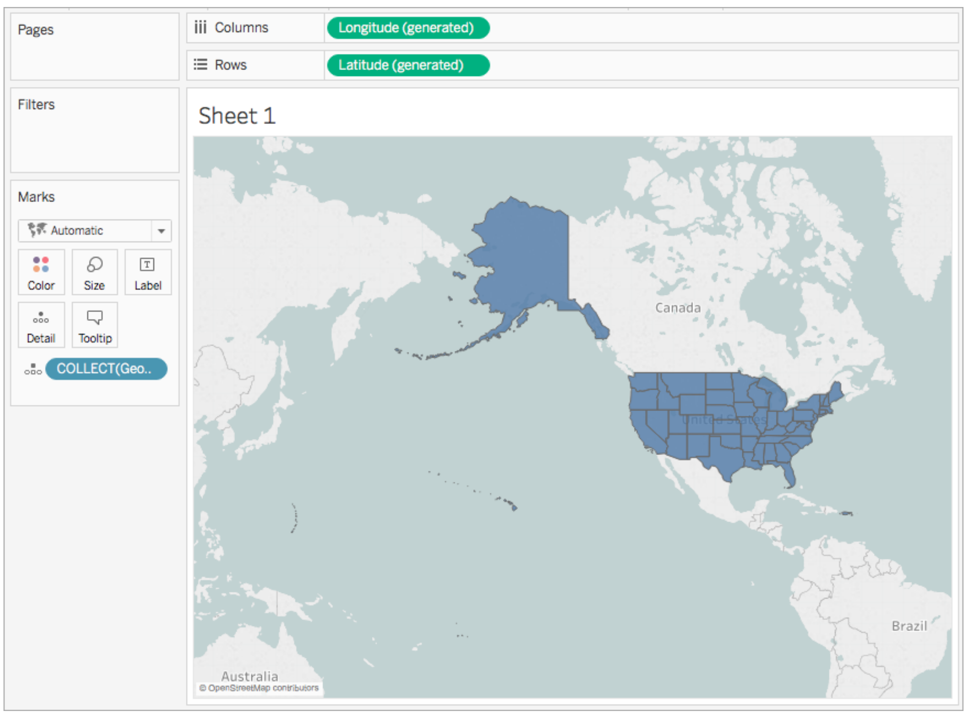
shelf.

A blank map is created.

8. From the **Data** pane, drag either the Geometry field from your spatial data source, or a geographic field from your excel data source, to **Detail** on the Marks card.

   The field you choose is for your bottom layer of data. For example, if you have polygon data in your spatial file, use the Geometry field so that the bottom layer will be filled polygons. If you have linear geometries in your spatial file, use a geographic field from your excel data source and format the map to be a filled or point map.

   In this example, the **Geometry** field from the shape file is used. The map updates to a filled map of the United States.



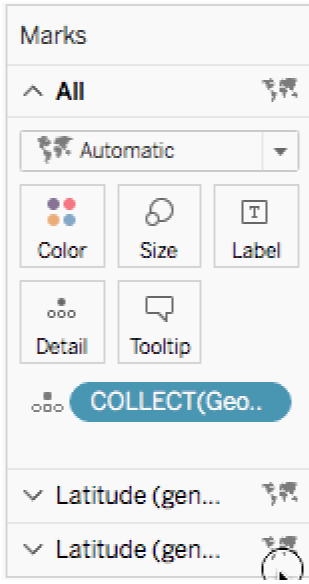
9. On the Rows shelf, Ctrl-click (Command-click on a Mac) and drag the **Latitude (generated)** field to the right. This copies the field.

   A second, identical map is created and the Marks card updates to include two tabs labeled Latitude(generated). The top tab is for the top map, and the bottom tab is for the bottom map.

10. On the Marks card, click the bottom **Latitude(generated) tab** and remove the

geographic field.

In this example, COLLECT(Geometry) is removed.



11. In the **Data** pane, under Measures, right-click the custom latitude field and select **Geographic Role** > **Latitude**. For more information, see **Assign Geographic Roles** on page 1834.

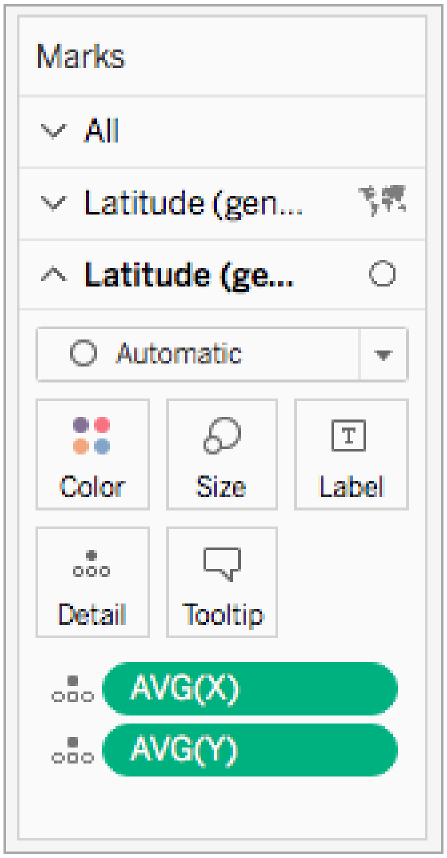
12. In the **Data** pane, under Measures, right-click the custom longitude field and select **Geographic Role** > **Longitude**.

13. From the **Data** pane, under Measures, drag the custom latitude field to **Detail** on the Marks card.

    In this example, the custom latitude field is named Y.

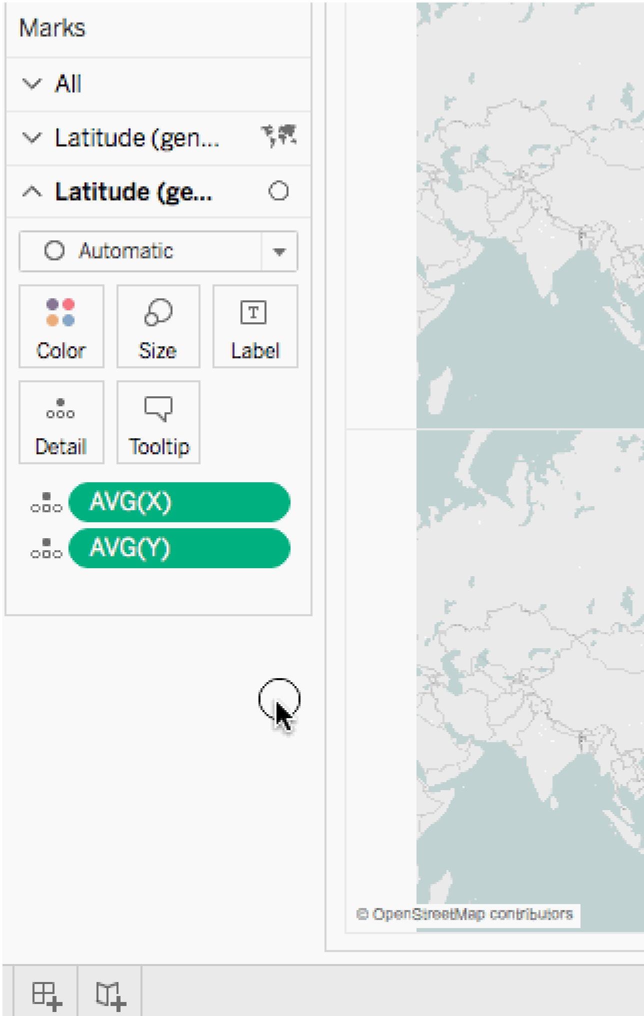
14. From the **Data** pane, under Measures, drag the custom longitude field to **Detail** on the Marks card.

    In this example, the custom longitude field is named X.



15. On the Marks card, right-click the custom latitude field and select **Dimension**.

16. Repeat step 13 for the custom longitude field.

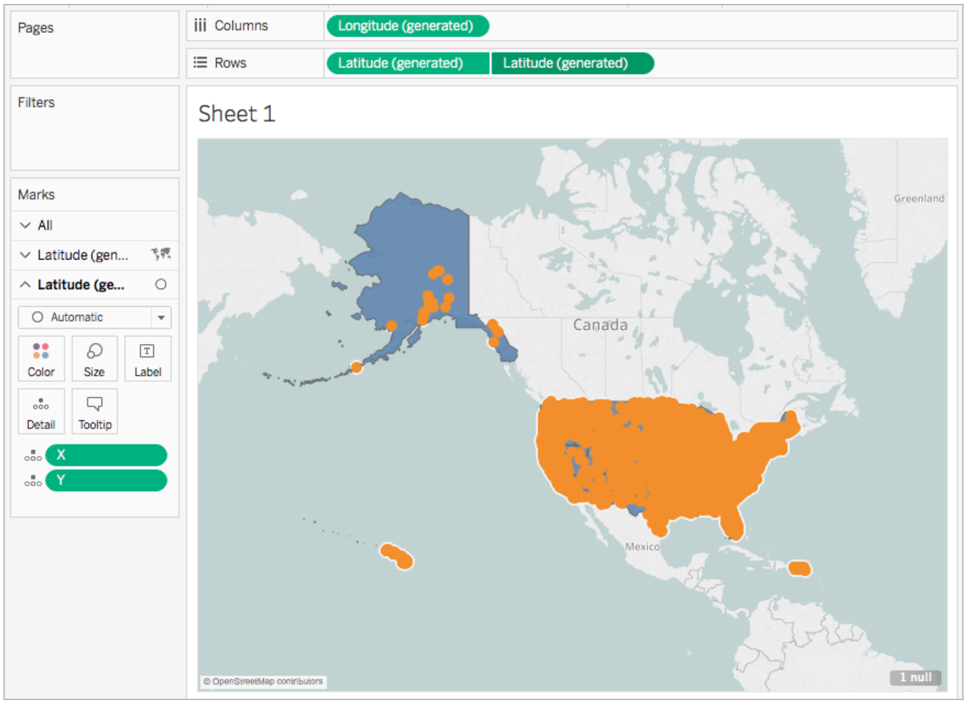


17. On the Marks card, click **Color**, and then select a new color.

    In this example, the color orange is used.

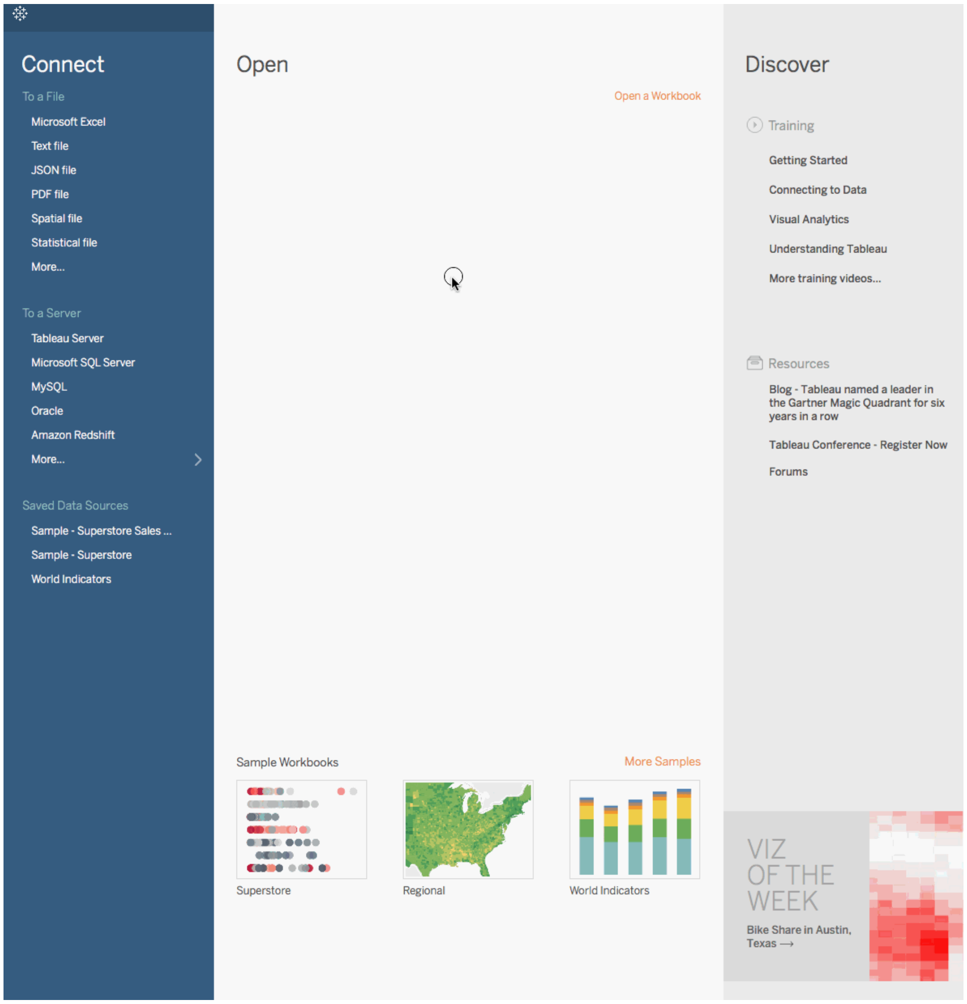
18. On the Rows shelf, right-click the **Latitude(generated)** field on the right and select **Dual Axis**.

    The two maps are now combined. You can filter the view as needed or zoom in to a particular area.



Pages
Columns
Longitude (generated)
Rows
Latitude (generated)
Latitude (generated)
Filters
Sheet 1
Marks
All
Latitude (gen...
Latitude (ge...
Automatic
Color
Size
Label
Detail
Tooltip
X
Y
Canada
Greenland
Mexico
1 null

Check your work! Watch steps 1- 18 below:



See Also

Build a dual-axis map from spatial data on page 1865

Join Your Data on page 883

Spatial File on page 1796

Assign Geographic Roles on page 1834

Create Filled Maps with Pie Charts in Tableau

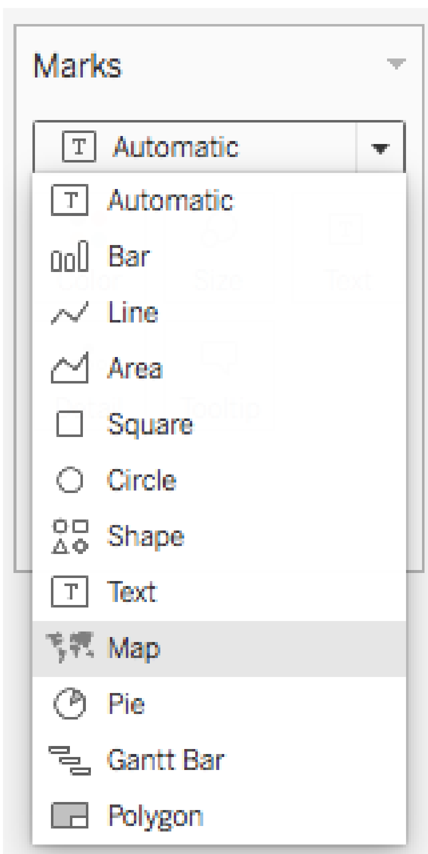
The pie mark type can be useful to show simple proportions to a relative whole. For example, pie marks might be effective when you want to show the percentage of profit for a product by geographic location.

In this example, the pie mark type is used with the filled maps mark type to show the percentage of profit for office supplies, furniture, and technology, relative to the total profits by state. It uses the Sample-Superstore data source that comes with Tableau Desktop.

Build the map view

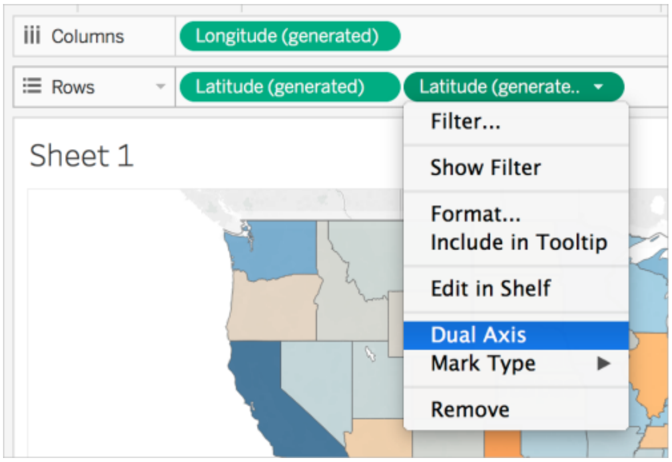
1. In Tableau, open a new workbook and connect to the Sample-Superstore data source.
2. On the Data Source page, click **Sheet 1** to go to a new worksheet.
3. In the Data pane, under Dimensions, double-click **State**.

   Tableau creates a symbol map, with a data point for each state in the Sample-Superstore data source.
4. On the Marks card, click the Mark-type drop-down and select the **Map** mark type.



5. From Measures, drag **Profit** to **Color** on the Marks card.
6. From Measures, drag **Latitude (generated)** to the **Rows** shelf, and place it to the right

of the other Latitude field.



You should now have two identical map views.

7. On the Rows shelf, right-click the second **Latitude** field and select **Dual Axis**.

   The second map is now layered on top of the first map. There are now three drop-downs on the Marks card: one for each map view, and one for both views (all). These are three separate marks cards that you can use to control the visual detail for each of the map views.

8. On the Marks card, click one of the **Latitude (generated)** tabs, and then click the Mark type drop-down and select the **Pie** mark type.

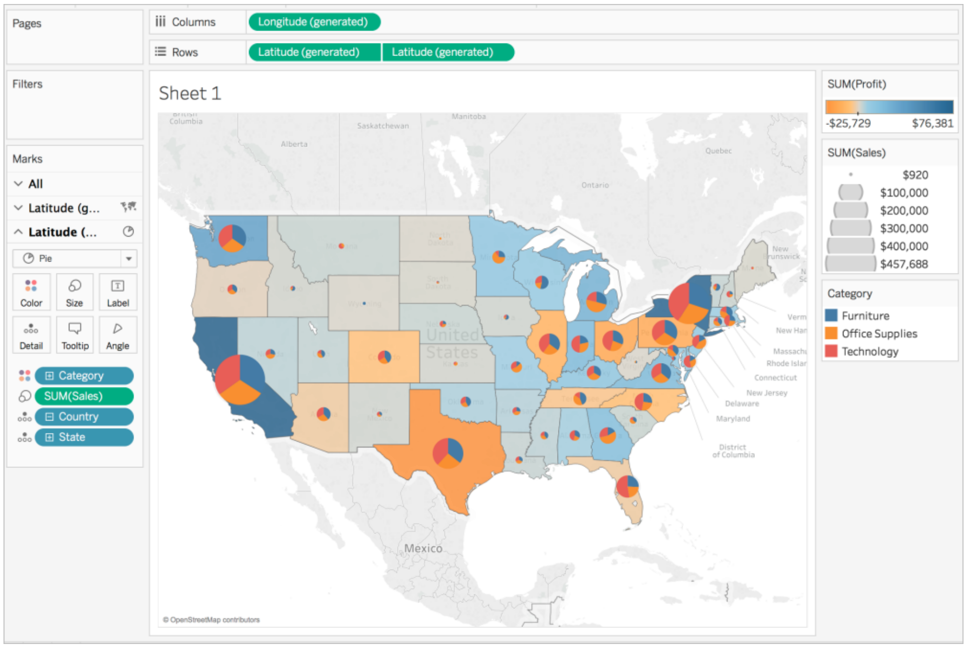
9. From Measures, drag **Sales** to **Size** on the Latitude (generated) Marks card you selected.

   The Sum of sales for each state is shown as text.

10. From Dimensions, drag **Category** to **Color** on the same Marks card.

    If the size of the pie charts is too small, click Size on the Marks card to adjust the size.

    The map view now shows the sum of profit, as well as the sum of sales for each category, for each state.



See also:

Mapping Concepts in Tableau on page 1781

Get Started Mapping with Tableau on page 1755

Customize Maps

Customize How Your Map Looks

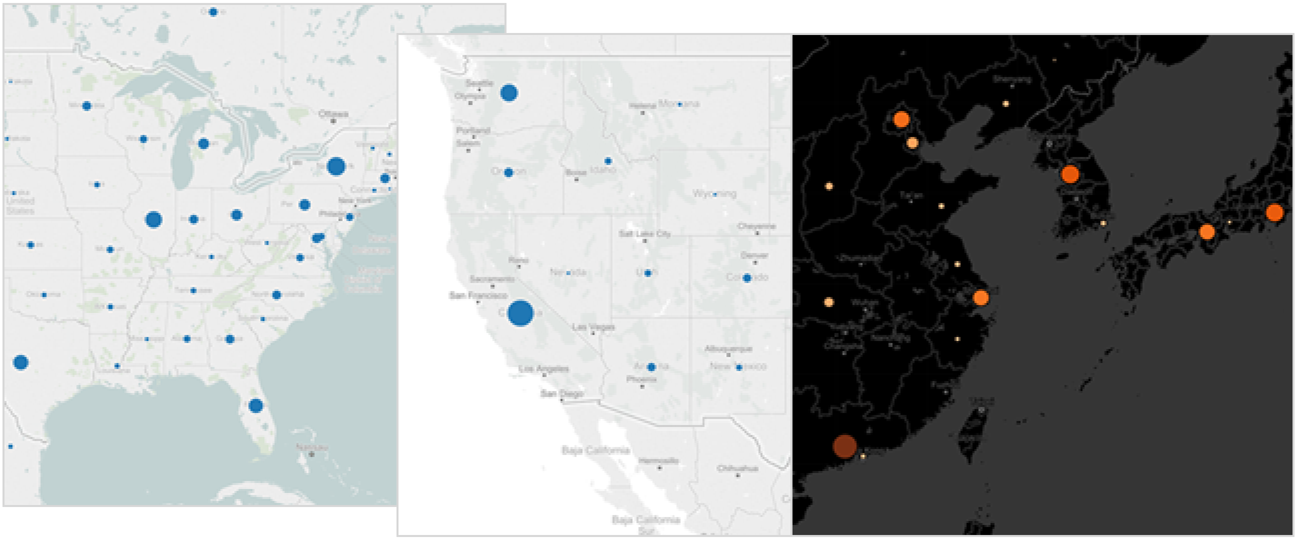
This article describes how you can customize the appearance of a map view in Tableau.

Note: Many of the tasks in this article make use of the **Map Layers** pane. To open this pane, select **Map** > **Map Layers**. In older versions of Tableau, select **Map** > **Map Options**.

Select a Tableau background map style

When you are connected to the Tableau background map, you can choose between six built-in background map styles: Normal, Light, Dark, Streets, Outdoors, and Satellite. You can see these styles below:

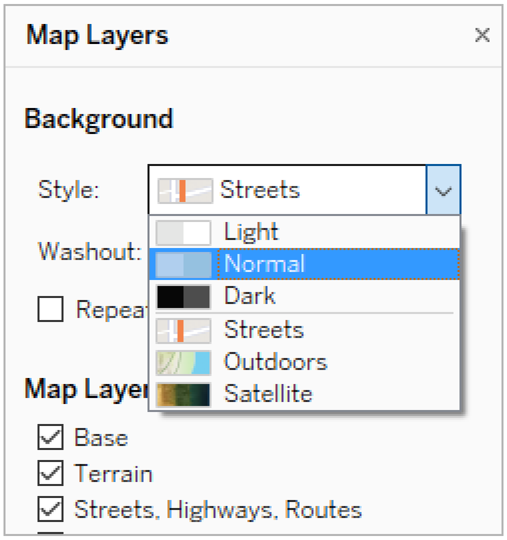
Normal	**Light**	**Dark**



Streets **Outdoors** **Satellite**

To select a Tableau background map style:

1. Select **Map** > **Map Layers**.
2. In the Map Layers pane on the left-hand side of the workspace, under Background, click the **Style** drop-down menu, and then select a background map style.



For more information about the built-in Tableau background maps, see **Select Background Maps** on page 1991.

Import your own background map

If the built-in Tableau background map styles don't meet your needs, you can import your own background map from a Web Map Service (WMS) server or a Mapbox map.

To import your own background map:

1. Select **Map** > **Background Maps** > **Map Services**.
2. In the Map Services dialog box, click **Add**, and then select to add WMS servers or Mapbox maps.
3. Follow the prompts to connect Tableau to the map service of your choice.

   For more information, see **Use Web Map Service (WMS) Servers** on page 1996 or **Use Mapbox Maps** on page 1993.
4. When finished, click **Close**.
5. Select **Map** > **Background Maps**, and then select the background map you want to use.

   The map updates to include the background map you choose.

Add a static background image

In addition to importing your own background map, you can add a static background image to your workbook and plot your data on it. For example, you can take a Google Map image and plot your data on it.

For more information, see the following topics:

- **Use Background Images in Your Views** on page 2012
- **Add Images of Google Maps and OpenStreetMap as Background Images in Tableau** on page 2022
- **Find Background Image Coordinates** on page 2024

Show or hide map layers

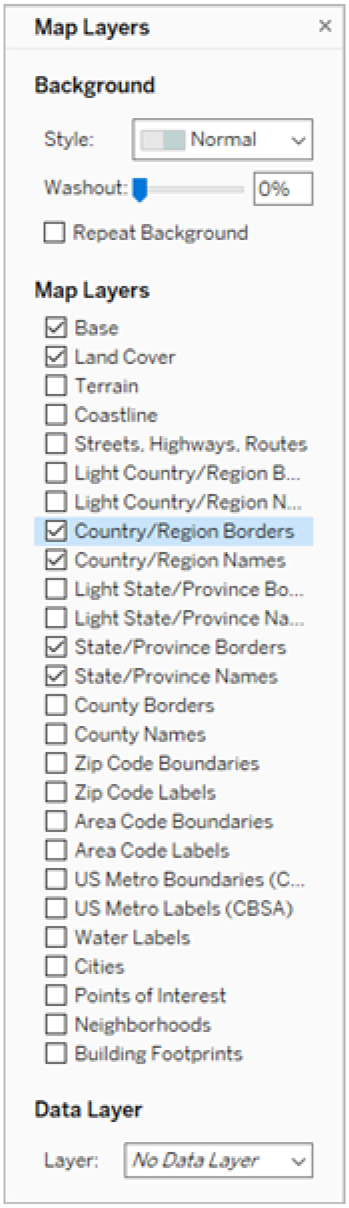
If you are using the Tableau background map, or a WMS map or Mapbox map that contains custom layers, you can show or hide layers on your background map to customize the background elements that frame your geospatial data. For example, you can overlay streets and highways or county boundaries on the map to give your data context.

To add or subtract map layers:

1. Select **Map** > **Map Layers**.
2. In the Map Layers pane, under the Map Layers section, select one or multiple map layers.

   Note: Some map layers are only visible at specific zoom levels. If a map layer is unavailable at your current level of zoom, it will appear grayed out.

To use layers that are not available, zoom further in to the view.



Some map layers are built to work with certain styles. What follows is a description of the different map layers available. Some of these layers are only available when using certain map styles.

Layer Name	Description	present in map style
Base	Shows the base map including water and land areas.	All
Land Cover	Shades wilderness areas and parks to give the map more depth.	All except Satellite
Terrain	Displays a relief of vertical elevation changes.	All except Satellite
Coastlines	Displays coastlines.	Light, Dark, Normal
Streets, High-ways, Routes	Marks freeways and highways as well as small city streets. This layer includes the highway and street names as well.	All
Light Coun-try/Region Borders	Shows a light gray outline of country/region borders and names.	Light, Dark, Normal
Light Coun-try/Region Names	Shows country and region names in a light gray.	Light, Dark, Normal
Country/Region Borders	Highlights country and region borders in a darker gray.	All
Country/Region Names	Highlights country and region names in a darker gray.	All
Light State/Province Borders	Shows a light gray outline of state borders and names.	Light, Dark, Normal
Light State/Province Names	Shows state and province names in a light gray.	Light, Dark, Normal
State/Province	Highlights state and province borders in a darker gray.	All

Layer Name	Description	present in map style
Borders		
State/Province Names	Highlights state and province names in a darker gray.	All
County Borders	Highlights U.S. county borders or second-level administration borders.	Light, Dark, Normal
County Names	Highlights U.S. county names or second-level administration names.	Light, Dark, Normal
Zip Code Boundaries	Marks U.S. zip code boundaries. You must zoom in to see this layer.	Light, Dark, Normal
Zip Code Labels	Shows labels for U.S. zip codes. You must zoom in to see this layer.	Light, Dark, Normal
Area Code Boundaries	Marks the U.S. area code boundaries. You must zoom in to see this layer.	Light, Dark, Normal
Area Code Labels	Shows labels for the U.S. area codes. You must zoom in to see this layer.	Light, Dark, Normal
US Metro Boundaries (CBSA)	Marks U.S. Metropolitan Statistical Areas and Micropolitan area boundaries.	Light, Dark, Normal
US Metro Labels (CBSA)	Shows labels for the U.S. Metropolitan Statistical Areas and Micropolitan areas.	Light, Dark, Normal
Water Labels	Shows labels for bodies of water.	All

Layer Name	Description	present in map style
Cities	Shows labels for cities.	All
Points of Interest	Displays labels for points of interest such as schools, parks, cemeteries, businesses, and important buildings. This layer is dependent on the zoom level.	All
Neighborhoods	Shows labels for neighborhoods in cities. You must zoom in to see this layer.	All
Subway and Train Stations	Displays the names of subway and train stations. You must zoom in to see this layer.	All except Satellite
Building Foot-prints	Shows the outline of buildings, where available. You must zoom in to see this layer.	Satellite, Streets, Outdoors
House Num-bers	Displays the house numbers of buildings, with and without building footprints. You must zoom in to see this layer.	Streets, Outdoors
Contour Lines	Displays lines indicating vertical elevation change in meters. This layer is dependent on zoom level.	Outdoors

Add layers for U.S. demographic data

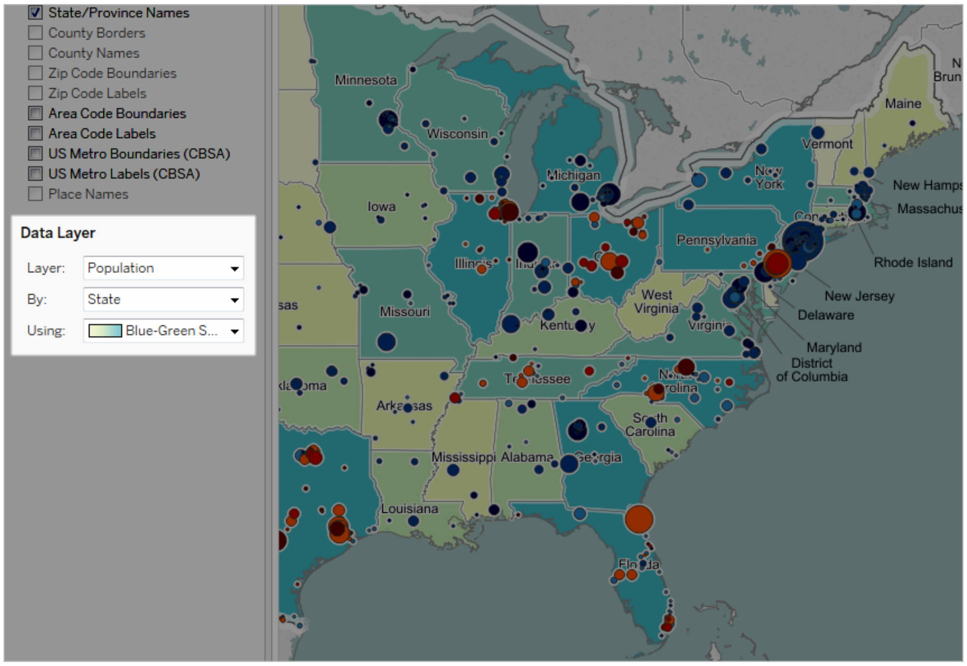
If you are using the Tableau background map, you can turn on a variety of predefined data layers that show U.S. census information. The data in these layers is for demographic data collected from the U.S. ACS Census.

> **Note:** Map data layers are only available for locations in the U.S.
>
> To add data layers for locations outside the U.S. you can connect to a WMS server that contains the desired demographic information. For more information, see **Use Web Map Service (WMS) Servers** on page 1996.

To add a U.S. data layer to your map view:

1. Select **Map** > **Map Layers**
2. In the **Map Layers** pane, under **Data Layer**, do the following:

- Click the **Layer** drop-down menu and select a data layer.
- Click the **By** drop-down menu, and then select to layer the data by **State**, **County**, **Zip Code** or **Block Group**.
- Click the **Using** drop-down menu to select a color scheme.



Once you select a data layer, it is added as shading to the map and a legend is shown to explain the colors of the layers. To hide or show this legend at any time, select **Map** > **Map Legend**.

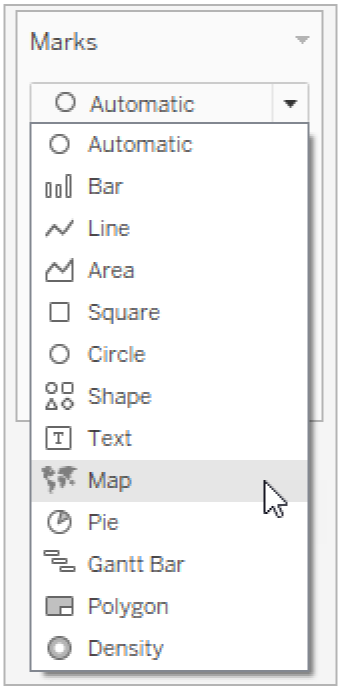
Change the mark type

By default, when you add a geographic field to the view, Tableau creates a point map. You can change this to a polygon (filled) map, a line map, or a density map (heatmap).

> **Note**: Filled maps are not available at the city or postcode level.

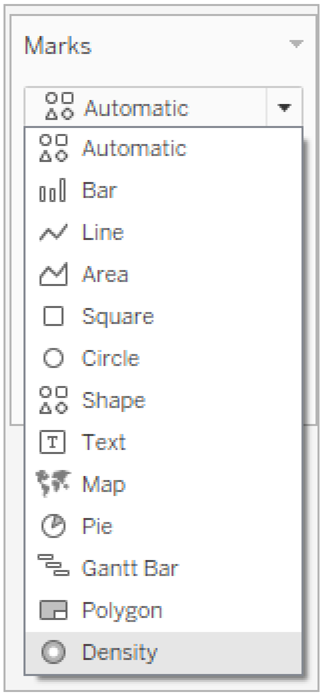
To change a point map to a filled or line map:

- On the Marks card, click the Mark Type drop-down and select **Map**.



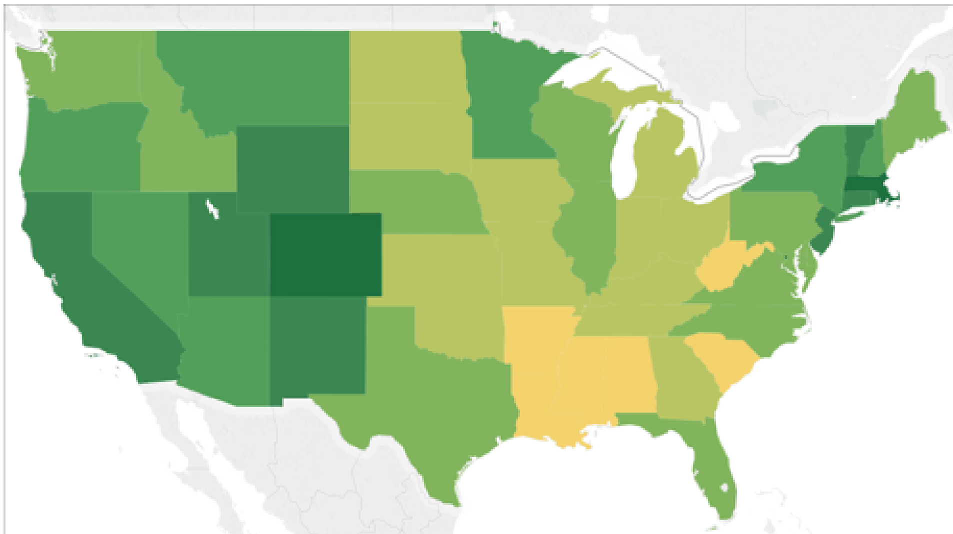
To change a point map to a heatmap (density map):

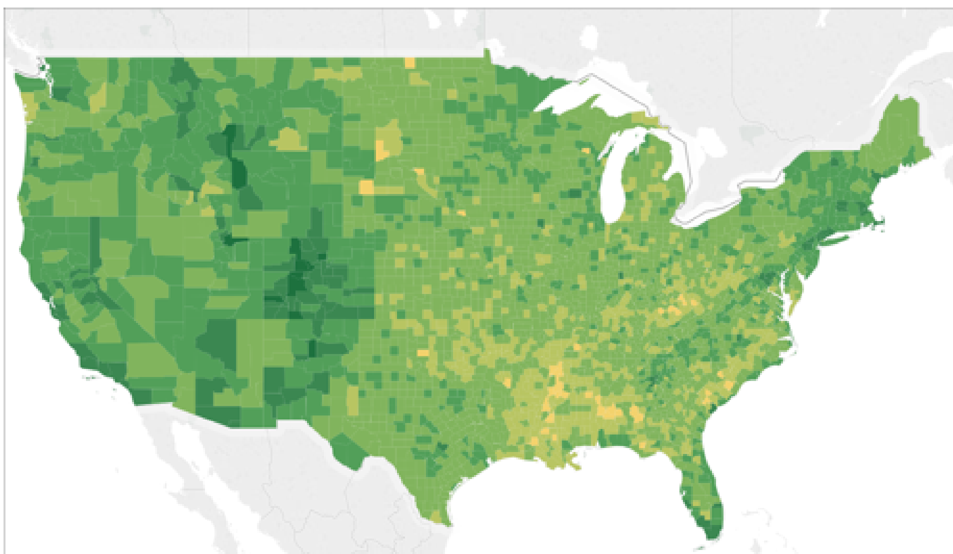
- On the Marks card, click the Mark Type drop-down and select **Density**.



Add levels of detail

With maps, for each level of detail you add, the more granular your data becomes. For example, you might look at obesity rates at the state level, or you could drill down into the county level, like the examples below. Adding or subtracting levels of detail changes the make up of your map.





To add levels of detail to the view:

- From Dimensions, drag a geographic field to **Detail** on the Marks card.

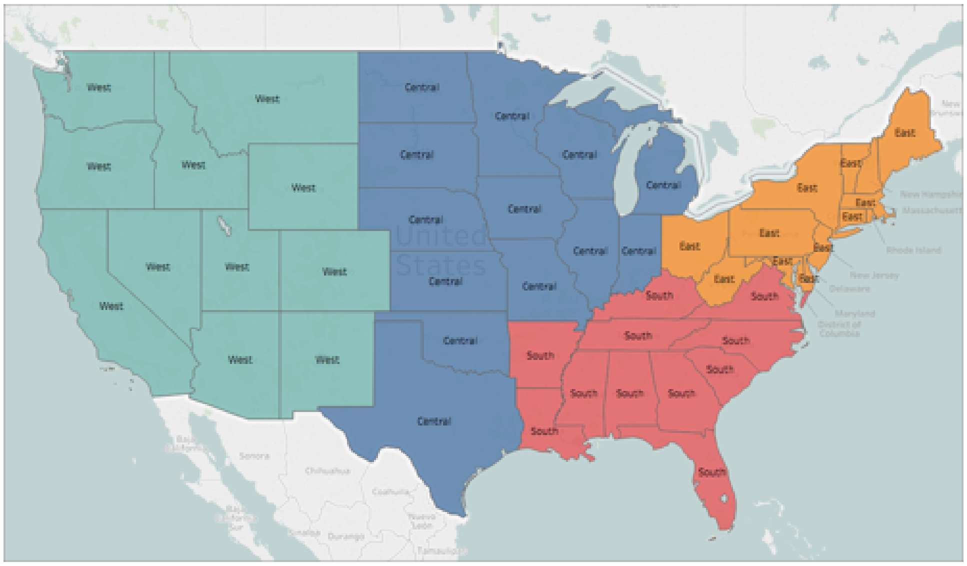
Add color

There are two ways you can add color to your map view: You can color locations categorically, or you can color locations quantitatively.

To color locations on your map categorically:

- From the **Data** pane, drag a dimension to **Color** on the Marks card.

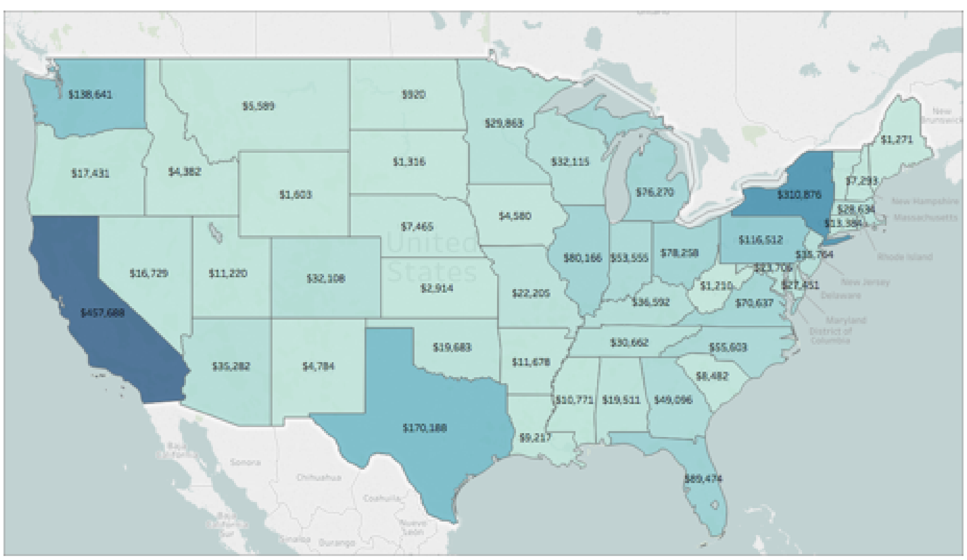
  The image below shows each state in the U.S. colored by region: West, Central, South, and East. The dimension, Region, is on Color on the Marks card.



To color each location on your map quantitatively:

- From the **Data** pane, drag a measure to **Color** on the Marks card.

  The image below shows each state in the U.S. colored by the amount of sales they achieved. The measure, Sales, is on Color on the Marks card.



For more information about color, see **Color Palettes and Effects** on page 1394.

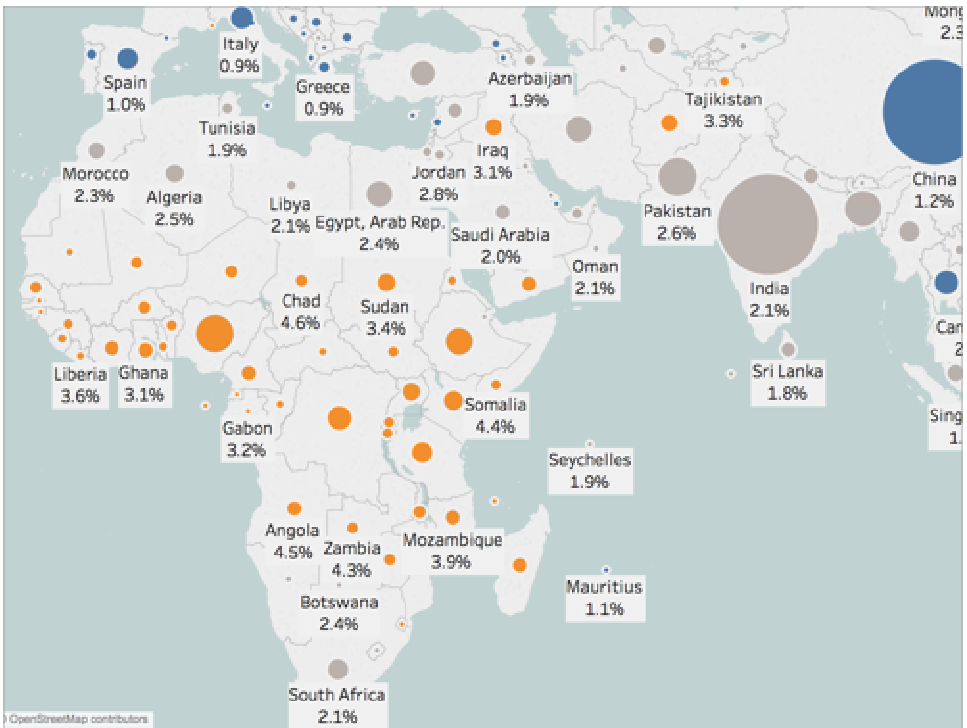
Add labels

You can add labels to your locations to provide extra context. For example, you can add labels for location name and sales.

To add labels to your data, from the **Data** pane, drag a dimension or measure to **Label** on the Marks card.

A label appears in the center of your location (if a polygon), or to the side of your location (if a data point).

You can add multiple labels.

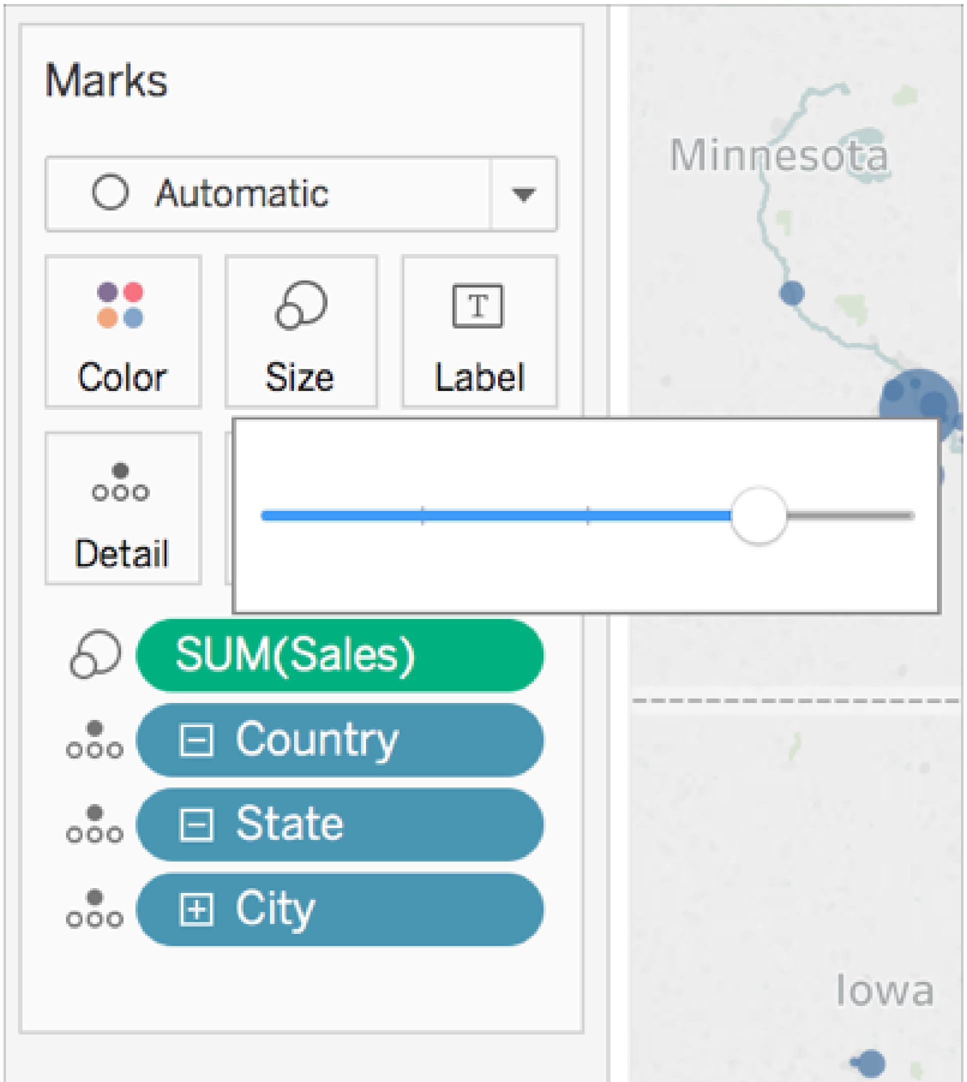


Adjust the size of your data points

You can adjust the size of your data points to compare and contrast them, or make smaller data points easier to see.

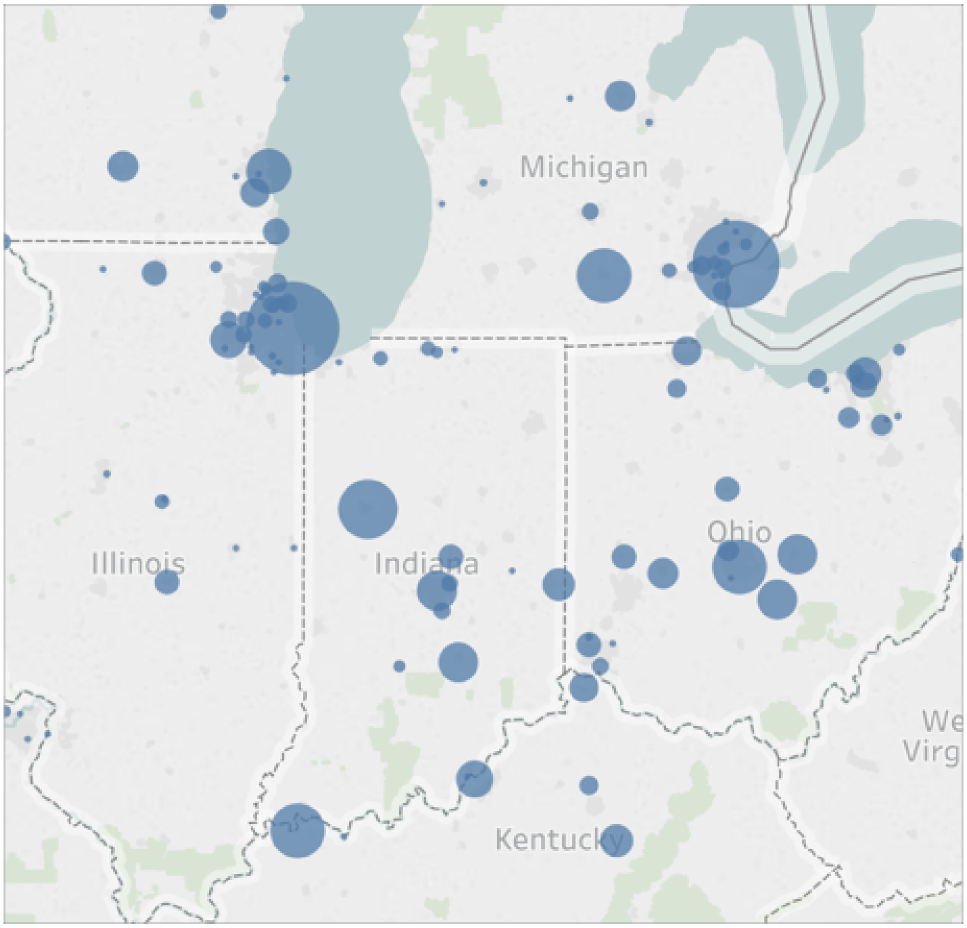
To uniformly adjust the size of your data points:

- On the Marks card, click **Size**, and then adjust the slider to the left or right.



To size your data points quantitatively:

- From Measures, drag a field to **Size** on the Marks card.



Create custom tooltips

You can create custom tooltips to show additional information about your locations when your audience hovers or clicks on them. You can type in your own information to appear for all marks, or add a field that will update with information specific to each mark.

To add a field to a tooltip:

- From the Data pane, drag the field to **Tooltip** on the Marks card.

To edit a tooltip:

1. On the Marks card, click **Tooltip**.
2. In the Edit Tooltip dialog box, format the tooltip how you would like it to appear.

For more information about customizing tooltips, see **Format tooltips** on page 2941 and **Add tooltips to marks** on page 1366.

See also:

Mapping Concepts in Tableau on page 1781

Use Web Map Service (WMS) Servers on page 1996

Use Mapbox Maps on page 1993

Select Background Maps on page 1991

Create Territories on a Map

When you're analyzing data on a map view, you might want to group *existing* locations together to create your own territories or regions, such as sales territories for your organization.

In Tableau, there are several ways to create territories:

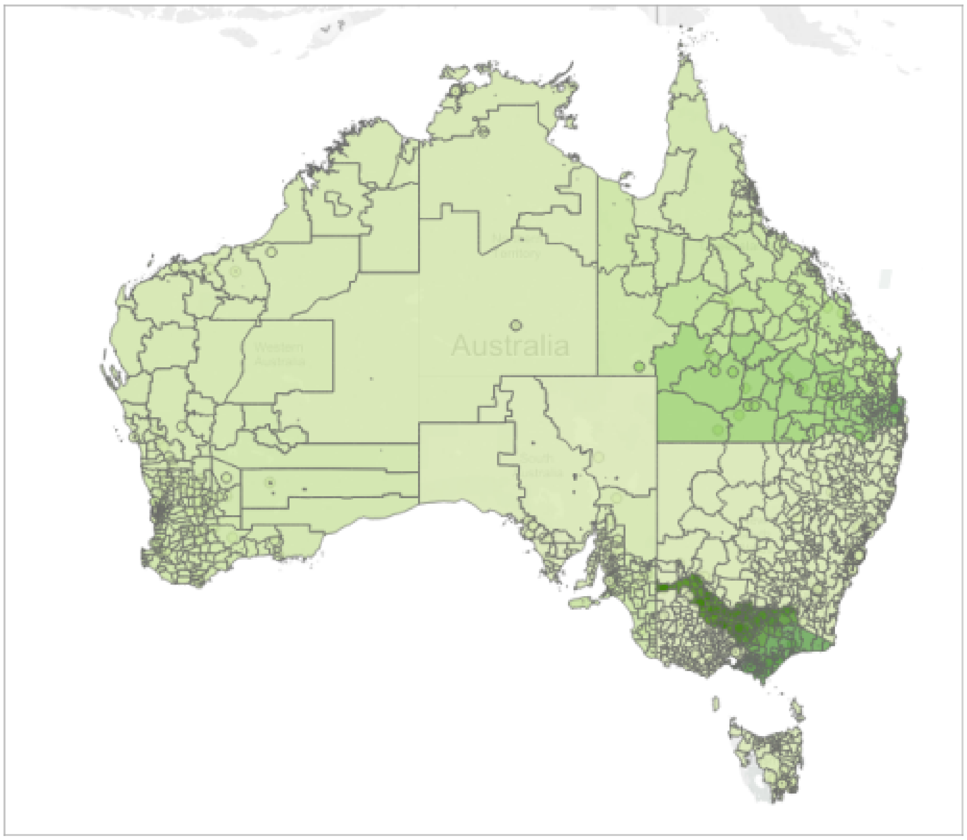
- **Option 1: Select and group locations on a map** on the next page
- **Option 2: Create a territory from a geographic field** on page 1985

- **Option 3: Geocode a territory field using another geographic field** on page 1985

Option 1: Select and group locations on a map

If you have created a map view, you can select and group locations (marks) on the map to create your territories.

For example, the following map view shows the total sales by postcode for a company with sales across Australia.



The first thing you might be thinking is: That's a lot of postcodes! In fact, there is a mark in this map view for every single 4-digit postcode in Australia, and each of those marks has an aggregation for SUM of Sales. If you're looking for sales information for every single postcode in Australia, than this is the view for you.

However, you might not want to analyze every individual location in your map view. Instead you might have your own areas, regions, or territories that you want to analyze as a whole. For example, maybe your organization splits Australia into five different sales territories, and you want to analyze the total sales for each of them.

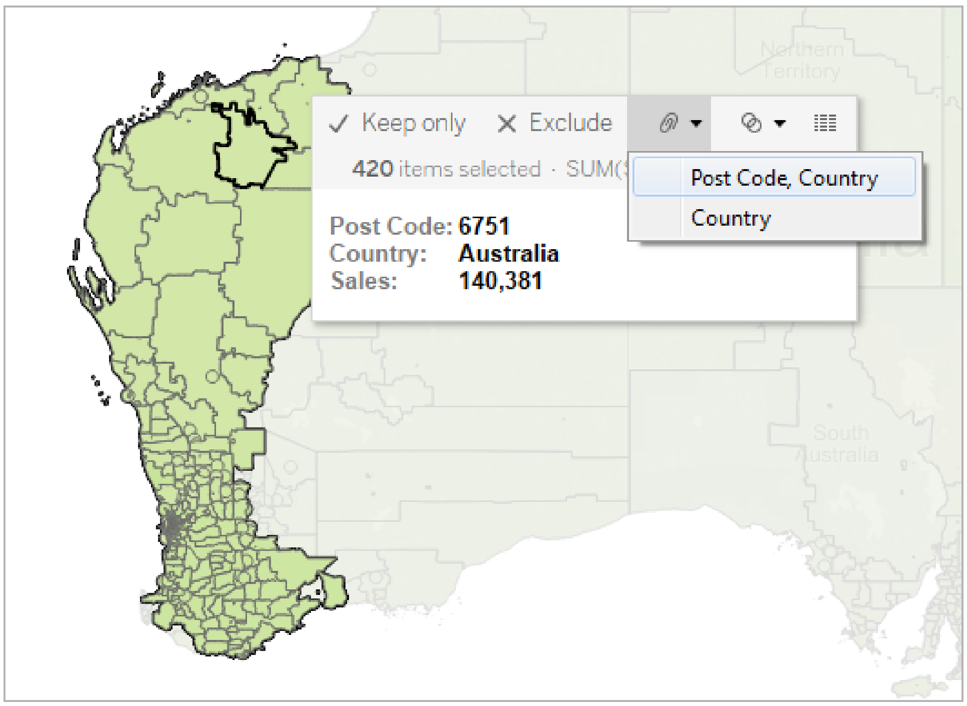
In this case, because you've already created a map view, you can select locations directly on the map and group them to create your own territories. Follow the steps below to learn how.

Step 1: Select locations on the map to group into a territory

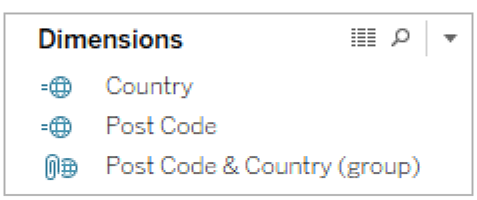
On the map view, select the locations (marks) that you want to group into your own territory. For more information about how to select marks, see **Select marks** on page 2722.

Step 2: Create the first group (your first territory)

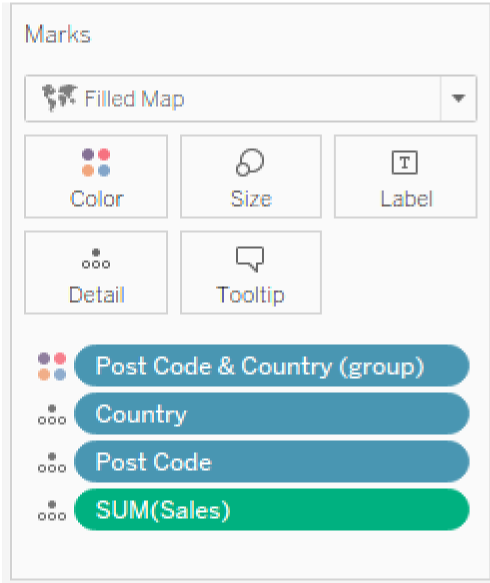
When you select marks in the view, or hover over a selection, a tooltip appears. On that tooltip, click the Group icon, and then select the level of detail you want to group the locations by.



A new group field is added in the **Data** pane with a group icon next to it. To change the locations within the group at any time, right-click the group field in the **Data** pane and select **Edit group**.



The new field is also added to **Color** on the **Marks** card.



Notice that the marks you selected are now all the same color.

Step 3: Continue grouping data to add territories

Repeat steps 1 and 2 to create additional territories. Each territory is added as a group within the group field you created in step 2. You can create as many territories as you want.

Step 4: Add the group field to the view

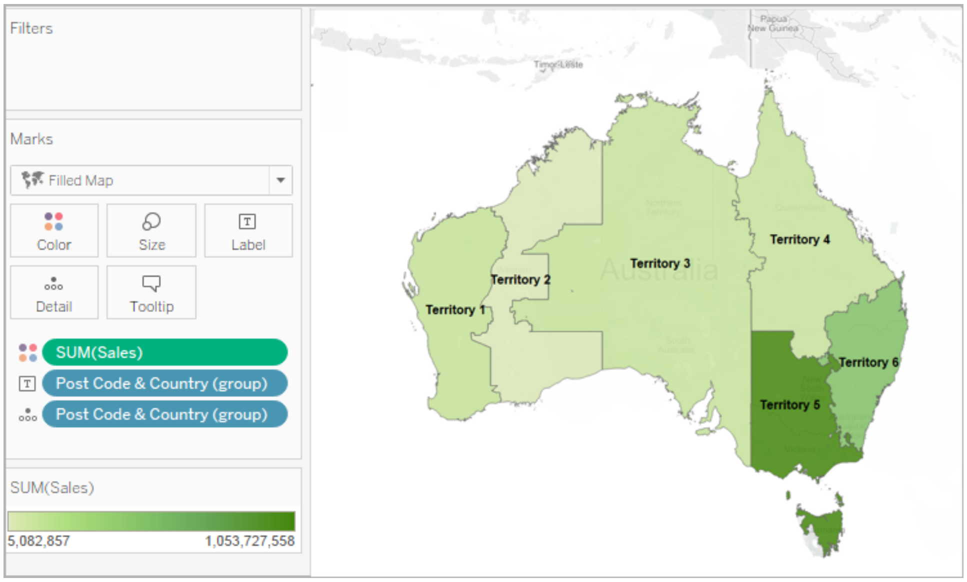
From the **Data** pane, drag the newly created group field to **Detail** on the **Marks** card. Next, to make each territory appear as a single mark, you must also remove the related geographic field(s) from the view.

For example, if you grouped postcodes at the **Post Code, Country** level, you must remove the **Post Code** and **Country** fields from the view.

In the image below, notice that the **Post Code** and **Country** fields are no longer on **Detail** on the **Marks** card because they have been replaced by the **Post Code & Country (group)** field.

In this example, there are five groups within the Post Code & Country (group) field, so five territories appear on the map. Aggregations, such as SUM (Sales), are computed at the level of the territory, rather than separately for each location within the territory.

You can now see the SUM (Sales) for each territory by color. Territory 5 has the most sales.



Option 2: Create a territory from a geographic field

You can also create territories by creating groups in the **Data** pane.

1. In the **Data** pane, right-click a geographic field (such as City or State) and select **Create** > **Group**.
2. In the Create Group dialog box, select the locations you want in your first territory and click **Group**. Each group you create represents a territory.
3. Repeat step 2 until you've created all of your territories, and then click **OK**.

   The new group field is added to the **Data** pane.
4. From the **Data** pane, drag the newly created group field to **Detail** on the **Marks** card.

   You can also drag the field to **Color** or **Label** on the **Marks** card to help differentiate between each territory.

Change locations in your territory groups

If you want to change the locations in your territory groups at any time, right-click the group field in the **Data** pane and select **Edit group**.

Option 3: Geocode a territory field using another geographic field

If your data source already contains a custom territory field , you can quickly geocode it to make a map in Tableau using an existing geographic field in your data source.

For example, the following data source contains a custom territory field for geographic areas called **Zone geographique**. It contains the values, **North**, **Central** and **South**.

If you were to bring this field into Tableau by itself, Tableau would not be able to recognize it as geographic data.

However, this data source also contains three geographic fields that Tableau *does* recognize: **Country**, **Region**, and **City**.

Each row in this data source contains a value for country, region, city and geographic zone. Therefore, each location is assigned a value from the custom territory field.

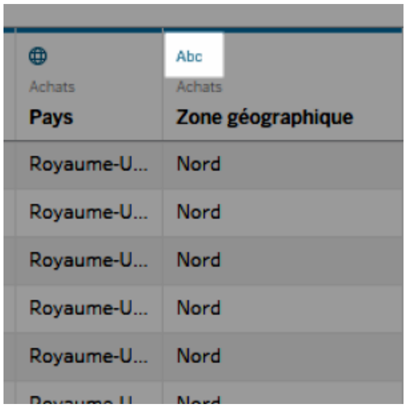
Achats **Ville**	Achats **Région**	Achats **Pays**	Abc Achats **Zone géographique**
Leeds	Angleterre	Royaume-U...	Nord
Leeds	Angleterre	Royaume-U...	Nord
West Bromwich	Angleterre	Royaume-U...	Nord
West Bromwich	Angleterre	Royaume-U...	Nord
West Bromwich	Angleterre	Royaume-U...	Nord
Le Bouscat	Aquitaine	France	Centre
Le Bouscat	Aquitaine	France	Centre
Le Bouscat	Aquitaine	France	Centre

Because the Zone geographique field is included in the workbook with other geographic fields that contain locations that Tableau recognizes, you can geocode it at the level of detail of one of those fields.

Follow the steps below to geocode a territory field using another geographic field:

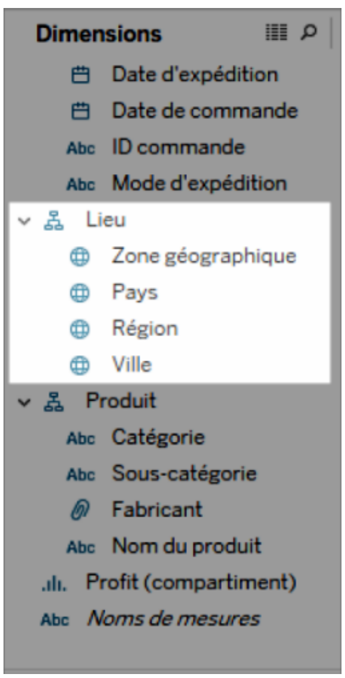
Step 1: Assign a geographic role to the territory field

On the data source page, click the data type icon for the custom territory field and select **Geographic Role** > **Create from**, and then select a level of detail to aggregate the field.



The options you see are based on the available geographic fields in your data source. Choose the level of aggregation you want in the view. If you're unsure how they will affect your view, try several different levels of detail. You can always undo and try again.

After you assign a geographic role to the territory field, it's added to its respective place in the location hierarchy. You can see it in the **Data** pane when you go to any worksheet.



Note: You can also click the data type icon next to the field in the **Data** pane to geocode a territory field using the above procedure.

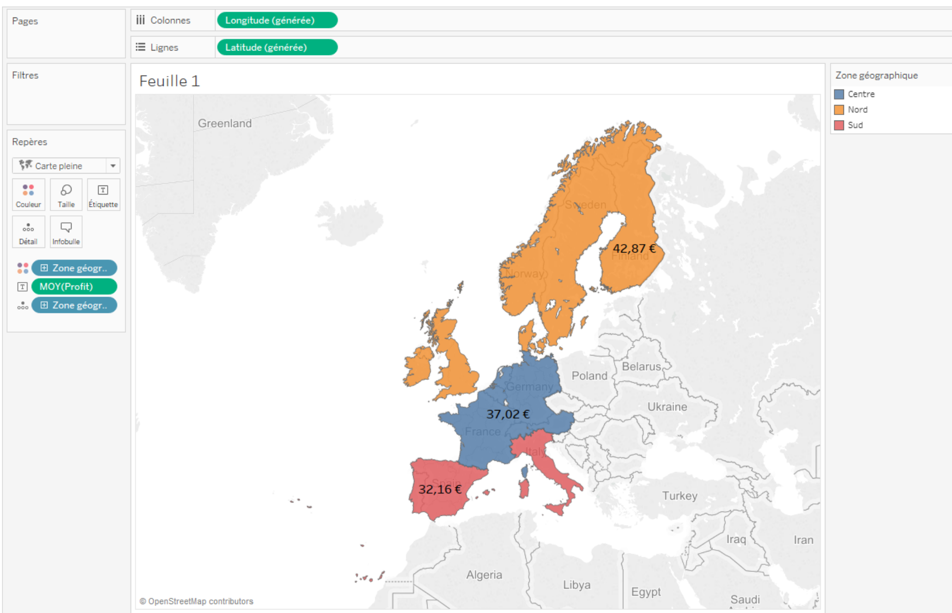
Step 2: Add the territory field to the view

Create a map view and add the newly geocoded territory field to the **Marks** card. You can place it on **Detail, Color, Label**, or **Tooltip**. Note that fields on **Detail** determine the level of aggregation in the view.

The custom territory field functions similar to your other geographic fields.

In the example below, the custom territory field (Zone geographique) is on **Detail** and **Color** on the **Marks** card. The average profit is shown for each territory (since the lowest level of detail is the territory field).

If you add another geographic field to **Detail** on the **Marks** card, such as the country field (Pays), the aggregation will update to show information for that level of detail.



Blend custom territory data

If your data source does not contain a territory field, you can blend a territory field with another geographic field in your data source. For more information, see **Blend Geographic Data** on page 1827. After you have blended your custom territory field with another field, follow the procedure in Step 1: Assign a geographic role to the custom territory field to geocode your territory field.

Customize How People Interact with your Map

When you create a map view, there are several default ways you can explore and interact with the view. You can zoom in and out of the view, pan, select marks, and even search for locations worldwide with map search.

However, sometimes you want to limit some of the ways your audience can interact with your map.

You can customize how your audience interacts with your view in the following ways using the **Map Options** dialog box.

Show a map scale

You can display a scale on your map so that your audience can understand distances between your data points. This scale appears in the bottom-right corner of the view, and updates as you zoom in and out of the map.

> **Note**: A map scale will not display on non-Web Mercator projections, such as a WMS using Plate Carrée.

To show a map scale on your map:

1. Select **Map** > **Map Options**.
2. In the Map Options dialog box that appears in the view, select **Show Map Scale**.
3. Under Units, click the drop-down and select the units of measurement you want.
    - If you want the scale to show measurements in meters and kilometers, select **Metric**.
    - If you want the scale to show measurements in feet and miles, select **U.S.**
    - If you want the units of measurement to be determined by your workbook locale, select **Automatic** .

    > **Note**: This setting also determines the units that the Radial tool uses to measure distances. For more information, see **Change the units of measurement** on page 2011 in the Measure Distances between Data Points and Locations on a Map View topic.

Hide map search

You can hide the map search icon so your audience cannot search for locations in your map view.

To hide the map search icon:

1. Select **Map** > **Map Options**.
2. In the Map Options dialog box that appears in the view, clear **Show Map Search**.

> **Note:** When you hide the map search icon, you also hide the locate me button for views published on the web. For more information about the locate me button, see **Explore Data in Maps** on page 2003.

Hide the view toolbar

You can hide the view toolbar in a map view so your audience cannot lock the map in place or zoom the map to all of your data.

To hide the view toolbar:

1. Select **Map** > **Map Options**.
2. In the Map Options dialog box that appears in the view, clear **Show View Toolbar**.

> **Note:** When you hide the view toolbar, you can still use keyboard shortcuts to zoom in and out of the view, pan, and select marks. For more information, see **Shortcuts for selecting and navigating marks** on page 3350.

Turn off pan and zoom

You can turn off pan and zoom in your map view, as well as in background images, so your audience cannot pan, or zoom in or out of the view.

To turn off pan and zoom:

1. Select **Map** > **Map Options**.
2. In the Map Options dialog box that appears in the view, clear **Allow Pan and Zoom**.

When you turn off pan and zoom, the pan tool and all zoom controls are removed from the view toolbar, and the rectangular tool becomes the default tool. Keyboard shortcuts for zooming in and out of the view, or panning, no longer work. If the map search icon is not hidden, you can still navigate the view by searching for locations with the map search box.

See also:

Search for Locations in Your Map on page 2006

Zoom and Pan Views, and Select Marks on page 2719

Select Background Maps

Tableau comes with a set of online and offline background maps that you can access to create map views.

By default, Tableau connects to an online map provider.

The three background maps that come with Tableau are described below. They can be found in the **Maps** > **Background Maps** menu. If the workbook author has added a WMS server or Mapbox map to the workbook, they will be listed using the name the author gave them in the **Maps** > **Background Maps** menu as well.

None	Displays data between latitude and longitude axes.
Offline	This background map stores the images that make up the map in a cache on your machine for improved performance and offline access. For more information, see the offline maps section below.
Tableau	Connects to the Tableau background map. By default, all map views connect to this background map, unless you specify otherwise.

To select a new background map:

- In Tableau Desktop, Tableau Server, or Tableau Online, select **Map** > **Background Maps** and then select the background map you want to use.

> **Note**: In Tableau Server, your administrator can restrict you to using only Offline maps. The specific server commands to configure this behavior can be found here.

Set a default background map in Tableau Desktop

By default, all map views connect to the Tableau background map. In Tableau Desktop, if you do not want maps views to automatically be created with the Tableau background map, you can specify a different background map as the default.

To specify a default background map:

1. In Tableau Desktop, select **Map** > **Background Maps**, and then select the background map that you want to make the default.
2. Select **Map** > **Background Maps** > **Set as Default** to set the selected background map to the default.

   The background map is automatically saved as a Tableau Map Source (.tms) and placed in the Mapsources folder of your My Tableau Repository. It is now the default background map for all new worksheets.

Use the Offline background map in Tableau Desktop, Online, or Server

You can create and inspect data in a map view offline using the offline background map that comes with Tableau Desktop.

To use the offline background map:

- In Tableau, select **Map** > **Background Maps** > **Offline**

Note: The offline background map uses map images stored on your machine. You can find these images in the following locations:

- On Windows: `C:\Program Files\Tableau\<Tableau Version>\Local\Maps`
- On Mac: `//Applications/<Tableau Version>.app/Contents/install/local/maps`

There are several actions, however, that require Tableau to retrieve a map image that may not be stored. If the new map image is not stored on your machine, you won’t be able to load the map until you reconnect to the online map that comes with Tableau.

You may need to reconnect to the online map if you would like to do one or more of the following:

- **Turn layers on or off** - if you decide to turn on a layer that isn’t stored in the cache, Tableau will need to connect to retrieve the necessary information.
- **Zoom in or out** - zooming in or out on a map requires different map images. If the images at the specified zoom level don’t exist in the cache, Tableau will need to retrieve the updated maps.

- **Pan** - panning sometimes requires new map images. If you are working offline and don’t have the necessary map images and legends stored in the cache, the new images and legends will not load.

To reconnect to the Tableau online map:

- On Tableau Desktop, select **Map** > **Background Maps** > **Tableau**

> **Note**: If you set a map to Offline and then publish the workbook, the published workbook will still use the Offline, stored maps, with all of the functionality and limitations of Offline maps noted above.

About the Tableau background map

The Tableau background map includes updated map and demographic data, as well as stylistic improvements. It has taken the place of the Tableau classic background map. Any map view created with an earlier version of Tableau Desktop, and with a connection to the Tableau classic background map, will automatically connect to the Tableau background map when opened in Tableau Desktop version 9.0 or later.

> **Note:** If you create a map view with data layers in Tableau Desktop version 8.3 or earlier using the Tableau classic background map, and then open that view in Tableau Desktop 9.0 or later, the color of the data layers may appear differently due to data layering and color ramp (color palette) improvements made to the Tableau background map.

Use Mapbox Maps

If you have access to Mapbox maps, you can add them to your workbooks or use them to create map views in Tableau Desktop. Fora list of country-specific data available, see Supported Map Data.

When you publish a view that uses Mapbox maps to Tableau Server, Tableau Online, or Tableau Public, your audience can view your data and your Mapbox map without having a Mapbox account.

Starting in Tableau Desktop 2019.2, Mapbox styles Streets, Outdoors, and Satellite are available by default in Tableau in the Map Layers pane. For more information on map styles and map layers, see **Customize How Your Map Looks** on page 1965.

Add a Mapbox map to your workbook

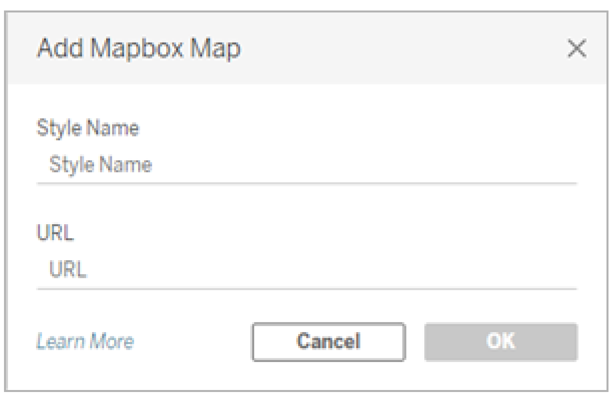
In Tableau Desktop, you can add a Mapbox map to your workbook and use it as a background map.

After you add a Mapbox map to your workbook, the map is saved with the workbook and available to anyone with whom you share the workbook. You can also save a Mapbox map as a Tableau Map Source (.tms) file that you can share with others so they can quickly connect to it and use it in their own workbooks. For more information, see **Save a Map Source** on page 2001. To add a Mapbox map:

1. In Tableau Desktop, select **Map** > **Background Maps** >**Mapbox Map**.
2. In the Add Mapbox Map dialog box, add:
    - **Style name**: enter a name for the Mapbox map. This name can be anything you want, and will appear in the Background maps menu after you add the map.
    - **Url**: enter a style URL for the Mapbox map you want to add.

      This URL contains the style ID for your Mapbox map, your access token, and your username. It might look similar to the following:

      ```
      https://api.mapbox.com/styles/v1/<username>/<stylei
      d>?access_token=<access token>
      ```

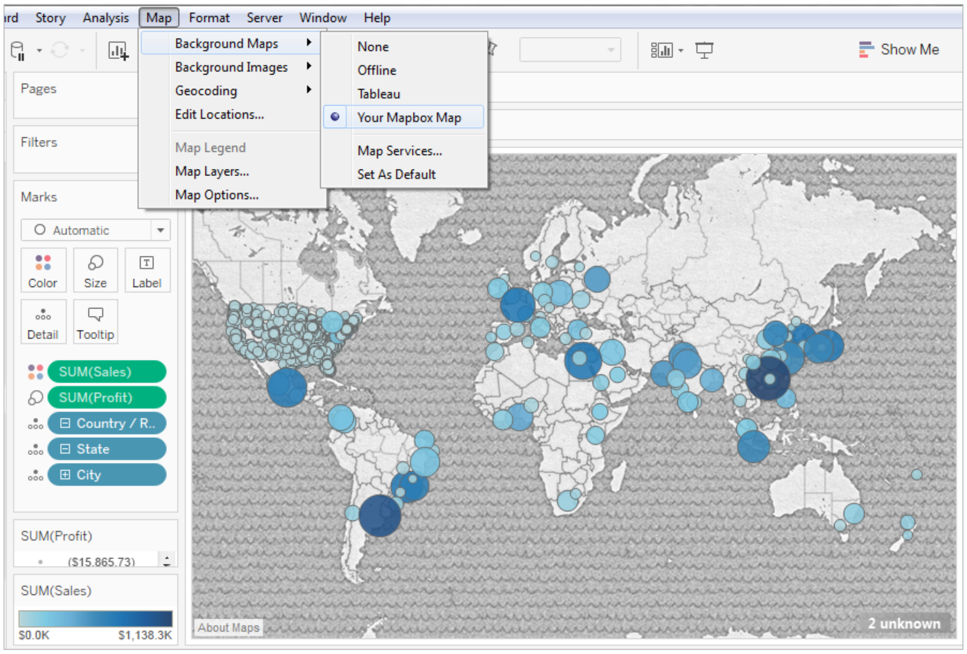
      

For more information about any of the above items, see the Access Tokens, Maps, and Styles sections of the Mapbox API Help.

3. When finished, click **OK** to exit the Add Mapbox Map dialog box, and then click **Close** to return to the view.

Create a map view using a Mapbox map

To create a map view using a Mapbox map, select **Map** > **Background Maps**, and then select the Mapbox map you want to use. Next, build the map view. For information, see **Mapping Concepts in Tableau** on page 1781.



Add or subtract map layers from the view

When you add a Mapbox Studio map to Tableau, all the map's layers are added, too. Layer groups and single layers in Mapbox Studio will display as individual layers in the Map Layers Pane in Tableau, and will retain the layer configuration set in Mapbox Studio.

If a layer group in Mapbox Studio includes sublayers that are turned off, those sublayers will also be turned off when the map is added to Tableau, and the layer group will appear as a single layer in the Map Layers pane. If the layer group is turned off in Tableau, then back on, all of that layer group's sublayers will be turned on.

Map layers are different from data layers in Tableau. In Tableau, Data Layers are pre-built tiles that include demographic information by various levels, like State and County. Data layers and Mapbox are independent of one another in Tableau. For more information about how data layers work, see **Customize How Your Map Looks** on page 1965.

Note: When you add custom map from Mapbox Studio into Tableau, the data for all layers are imported, even if those layers are turned off. Your map's performance could be affected if it includes many layers and sublayers. To improve performance, add or remove layers and sublayers in Mapbox Studio before adding your map to Tableau.

See also:

Use Web Map Service (WMS) Servers below

Save a Map Source on page 2001

Import a Map Source on page 2002

Select Background Maps on page 1991

Customize How Your Map Looks on page 1965

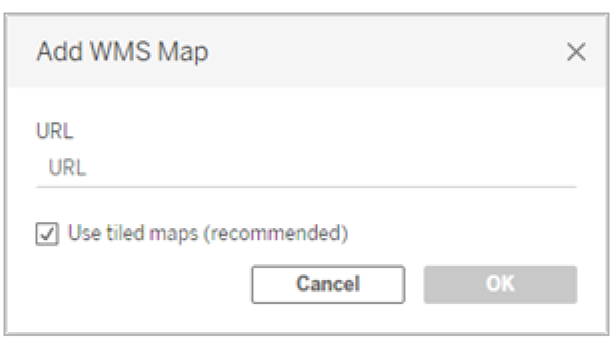
Use Web Map Service (WMS) Servers

In Tableau Desktop, you can connect to map servers with the Web Map Service (WMS) protocol. WMS is a standard protocol for requesting and receiving geographically referenced imagery.

You can connect to any WMS server that supports the WMS 1.0.0, 1.1.0, or 1.1.1 standards.

Connect to a WMS server

1. In Tableau Desktop, select **Map** > **Background Maps** >**WMS Servers**.
2. In the Add WMS Servers dialog box, type the URL for the server you want to connect to in Tableau, and then click **OK**.



You can add as many map servers as you want to a workbook. Each WMS server you add appears as a background map in the **Background Maps** menu.

Save a WMS server as a Tableau map source

After you add a WMS server to your workbook, it is saved with the workbook and available to anyone you share the workbook with. You can also save a WMS server as a Tableau Map Source (.tms) file, which you can share with others so they can quickly connect to it and use it in their own workbooks.

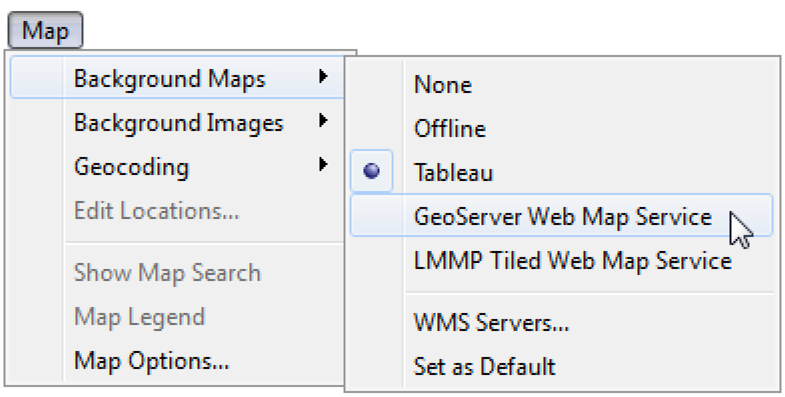
For more information about how to save a map as a Tableau map source, see **Save a Map Source** on page 2001.

Use a WMS background map

After you connect to a WMS server, you can create a map view using the WMS background map that Tableau creates.

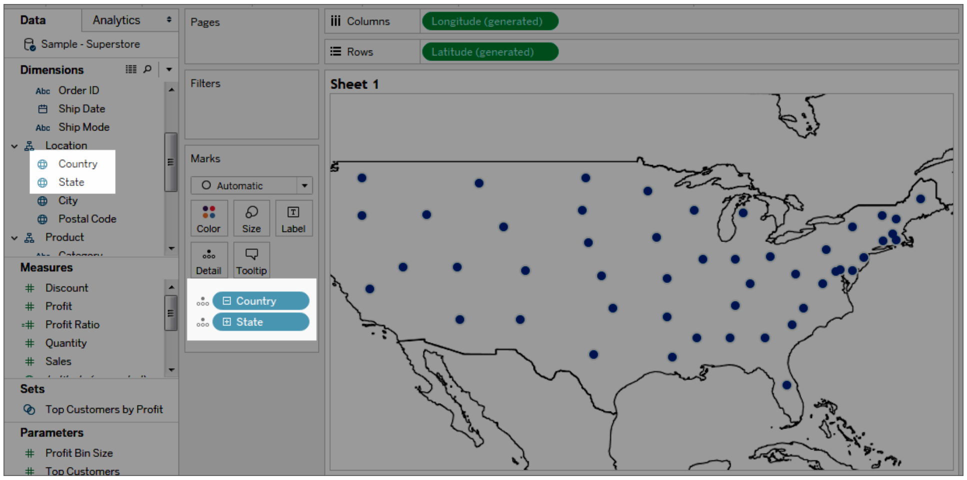
To create a map view using a WMS server:

1. In Tableau Desktop, select **Map** > **Background Maps**, and then select a WMS background map to use in the view.



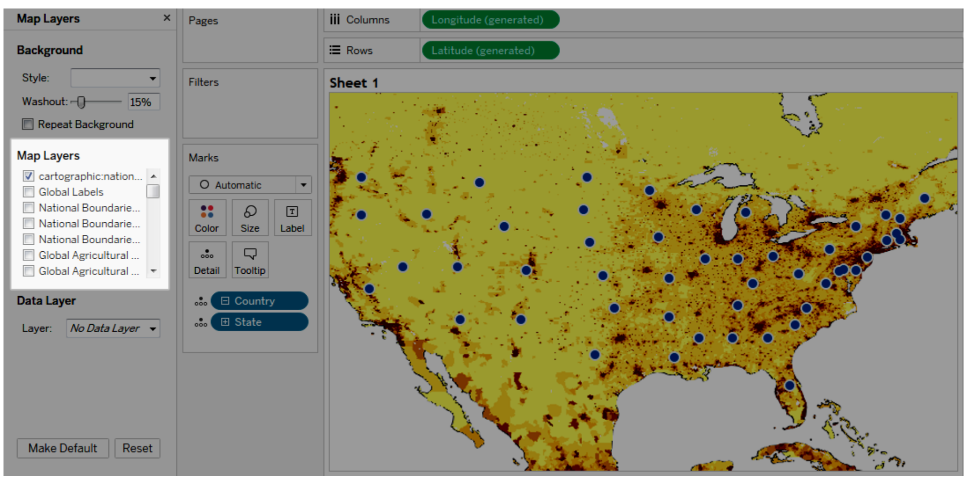
2. Add a geographic field to the view.

   For more information, see **Mapping Concepts in Tableau** on page 1781.



3. Select **Map** > **Map Layers**, and then select the map layers you want to show in the view.

   For more information, see **Customize How Your Map Looks** on page 1965.



Performance considerations

The content, speed, and performance of a WMS server is reliant on the network and WMS provider. If your Internet connection is interrupted or if you're working behind a restrictive firewall, you can use the Offline background map that comes installed with Tableau Desktop to avoid any performance issues while you build your map view. You can switch back to your WMS server at any time.

Note: With the offline background map, you might only be able to access up to four levels of zoom. For more information, see **Select Background Maps** on page 1991.

Supported Spatial Reference Systems

Tableau automatically supports the following list of Spatial Reference Systems (SRS) / European Petroleum Survey Group (EPSG) codes. WMS servers must support at least one of the following spatial reference systems to be compatible with Tableau mapping features.

Supported SRS/EPSG Codes in Tableau:

3857 - WGS 84 / Pseudo-Mercator [DEFAULT]

4326 - WGS 84

4269 - NAD83

3824 - TWD97

3889 - IGRS

4019 - Unknown datum based on the GRS 80 ellipsoid

4023 - MOLDREF99

4030 - Unknown datum based upon the WGS 84 ellipsoid

4031 - Unknown datum based upon the GEM 10C ellipsoid

4046 - RGRDC 2005

4075 - SREF98

4081 - REGCAN95

4126 - LKS94 (ETRS89)

4130 - Moznet

4140 - NAD83(CSRS98)

4148 - Hartebeesthoek94

4151 - CHTRF95

4152 - NAD83(HARN)

4163 - Yemen NGN96

4166 - Korean 1995

4167 - NZGD2000

4170 - SIRGAS 1995

4171 - RGF93

4172 - POSGAR

4173 - IRENET95

4176 - Australian Antarctic

4180 - EST97

4189 - REGVEN

4190 - POSGAR 98

4258 - ETRS89

4283 - GDA94

4319 - KUDAMS

4612 - JGD2000

4617 - NAD83(CSRS)

4619 - SWEREF99

4627 - RGR92

4640 - RRAF 1991

4645 - RGNC 1991

4659 - ISN93

4661 - LKS92

4667 - IKBD-92

4669 - LKS94

4670 - IGM95

4674 - SIRGAS 2000

4686 - MAGNA-SIRGAS

4687 - RGPF

4693 - Nakhl-e Ghanem

4694 - POSGAR 94

4702 - Mauritania 1999

4737 - Korea 2000

4742 - GDM2000

4747 - GR96

4749 - RGNC91-93

4755 - DGN95

4756 - VN-2000

4757 - SVY21

4758 - JAD2001

4759 - NAD83(NSRS2007)

4761 - HTRS96

4762 - BDA2000

4763 - Pitcairn 2006

4764 - RSRGD2000

4765 - Slovenia 1996

102100 - WGS 84 Web Mercator (Auxiliary Sphere)

See also:

Use Mapbox Maps on page 1993

Save a Map Source below

Import a Map Source on the next page

Select Background Maps on page 1991

Customize How Your Map Looks on page 1965

Save a Map Source

After you add a Mapbox map or WMS server to your workbook, you can save it as a Tableau map source (.tms).

Saving a Mapbox map or WMS server as a Tableau Map Source allows you to share your map with others so they can quickly import it into their own workbooks and use it to create new map views. For more information about importing a Tableau Map Source into a workbook, see **Import a Map Source** below.

1. Select **Map** > **Background Maps** > **Map Services**. This opens the Map Services dialog box.
2. Select the map that you want to save as a Tableau map source, and then click **Export**. This opens the Export Connection dialog box.
3. Type a name for the file, choose a location, and then click **Save**.

The Tableau map source includes any default map layer settings you have specified in the workbook. For example, the map source will include any custom set of map layers you have specified to show by default. For more information, see **Customize How Your Map Looks** on page 1965 in the Set Map Layers topic.

> **Note:** If you change the default settings for the map layer options, you should export the map again to include the new settings in the Tableau map source file.

See also:

Select Background Maps on page 1991

Import a Map Source

You can import a Tableau Map Source (.tms) that someone has shared with you into a workbook, and then use it to create custom map views.

1. Select **Map** > **Background Maps** > **Map Services**. This opens the Map Services dialog box.
2. Click **Import**. This opens the Import Connection dialog box.
3. Navigate to the saved Tableau map source file (.tms) that you want to import, select it, and then click **Open**. This closes the Import Connection dialog box and adds the Tableau map source to the list of maps in the Map Services dialog box.

   If you want to edit the newly added Tableau map source, select it from the list, and then click **Edit**. Otherwise, click **Close** to return to the view.

You can import as many Tableau map sources as you want into a workbook. Each Tableau map source you add appears as a background map in the **Background Maps** menu and is automatically selected as the default background map until another Tableau map source is imported, until you connect to a WMS server or Mapbox map, or until you select to use a different background map.

> **Note:** When you create a new map view, the background map that is selected in the **Background Maps** menu is used to create the map for that view.

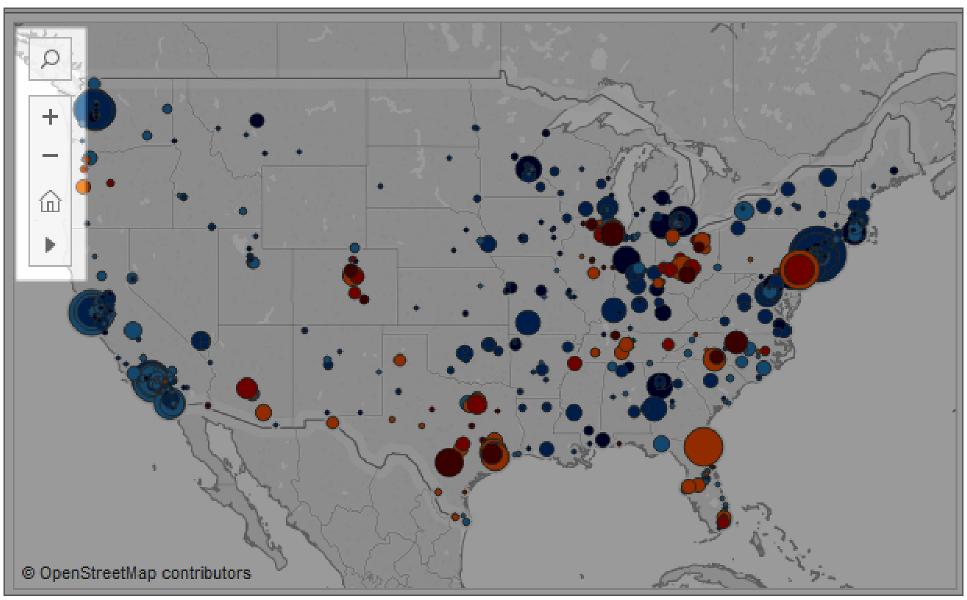
See also:

Save a Map Source on page 2001

Select Background Maps on page 1991

Explore Data in Maps

Tableau maps can help you quickly find locations and analyze data worldwide. There are many ways you can explore and interact with map views. You can zoom in and out, pan, and select marks with the view toolbar, and even search for locations worldwide with map search.



If you don't see the view toolbar or the map search icon, or if you can't pan or zoom, it's probably because the workbook author has chosen to customize how others can interact with the view.

The workbook author can choose to hide the view toolbar and the map search icon so they do not appear in the view. Similarly, the workbook author can choose to turn off pan and zoom to control how you interact with the view.

For more information, see Customize How People Interact with your Map.

Select marks, pan, and zoom

You can use the view toolbar in the upper left corner of the view to select marks, pan, and zoom in and out of a map view. For more information, see **Zoom and Pan Views, and Select Marks** on page 2719.

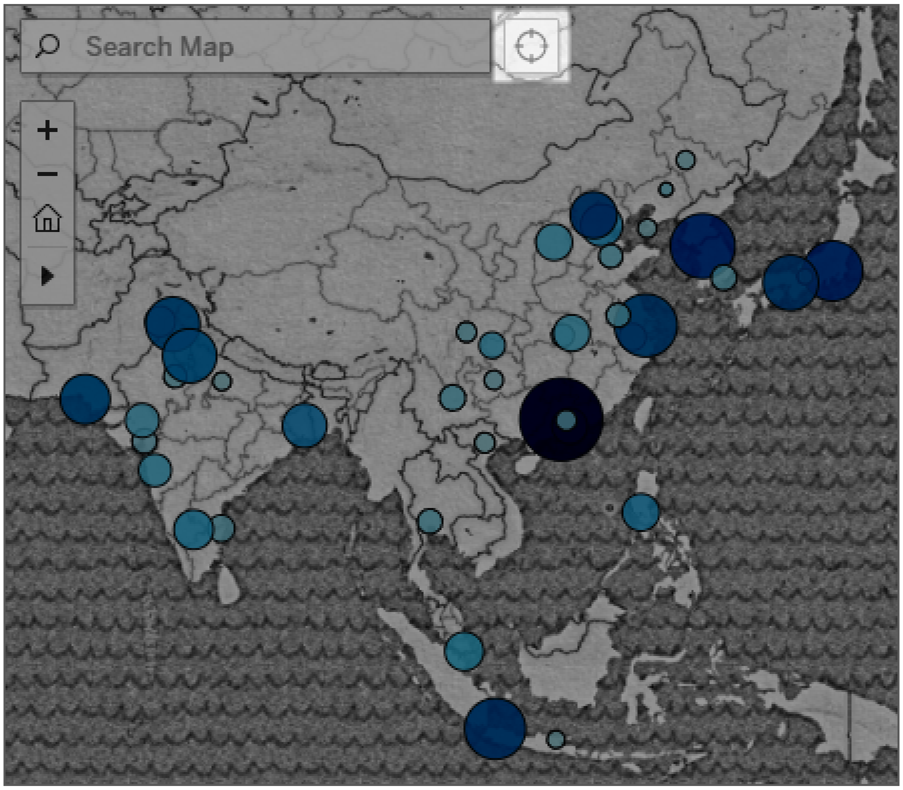
Search for locations

You can use the map search icon to search for locations in your map view. For more information, see Set Map Search Options.

Find your current location

If you're exploring a map view on Tableau Server or Tableau Online, you can use the locate me button to quickly pan and zoom to your current location. To do so, click the map search icon in the top-left corner of the view, and then, next to the search box that appears, click the locate me button.

Note: Finding your current location is not supported on Tableau Desktop.



Note: When you click the locate me button, you might be prompted by your web browser to allow Tableau access to find your current location. If you block access, Tableau will be unable to zoom to your current location.

Return to the initial view

After zooming in and out of the view, or panning to a new location, you can return to the initial view of your map. To do so:

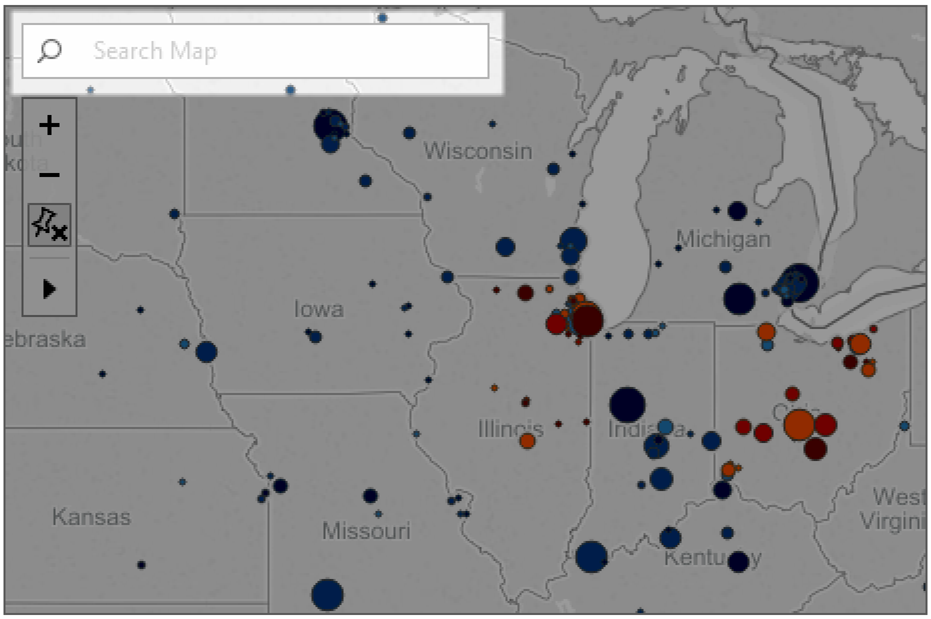
On Tableau Desktop, click the reset axes button on the view toolbar.

On Tableau Server or Tableau Online, click the zoom home button on the view toolbar.

Search for Locations in Your Map

Map search helps you find locations in a map view so you can quickly explore and inspect data.

The map search icon appears in the top left corner of the view. When you click the search  icon, a search box appears.

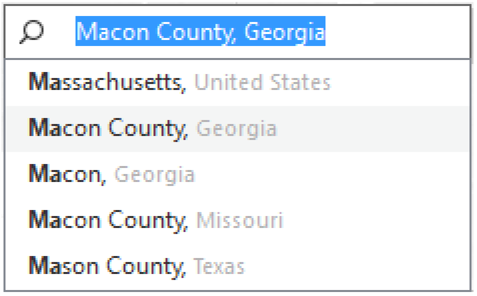


When you begin to type in the search box, map search suggests possible locations that are in your map view. The suggestions are based on location names and text in your data source.

You can search for the following location types:

- Continent
- Country
- State or province
- County
- City
- Postcode

Select a location from the list of suggestions to pan and zoom to that location on the map.



Hide Map Search

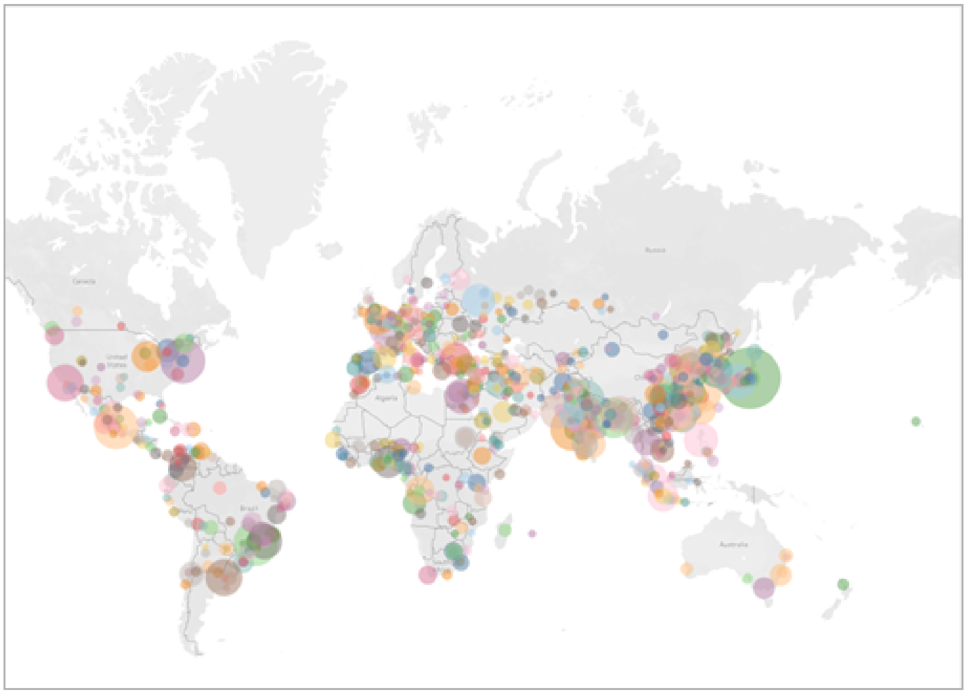
By default, the map search icon appears in the top left corner of the view. When you publish a view to Tableau Server or Tableau Online, or share the view through Tableau Reader, the map search icon remains in the view. If you do not want viewers to search for locations in your map view, you can hide the map search icon.

For more information about showing and hiding map search, see **Customize How People Interact with your Map** on page 1989.

Measure Distances Between Data Points and Locations in a Map

While exploring data in a map view, you might have questions about how that data relates to its surrounding geography, locations, or landmarks. To answer these types of questions, you can use the Radial tool to measure approximate distances in your map view.

The following view shows the number of earthquakes of magnitude 6.0 and higher that have occurred around the world between January, 1994 and February, 2014.



At a global scale, this view is very powerful, but you or your audience might want to zoom in and explore certain areas more closely. For example, maybe you want to see how many earthquakes have occurred in Indonesia in the past 10 years, specifically within 100 miles of its capital, Jakarta.

To do so, you can use the Radial tool to find all the earthquakes within approximately 100 miles of the capital city.

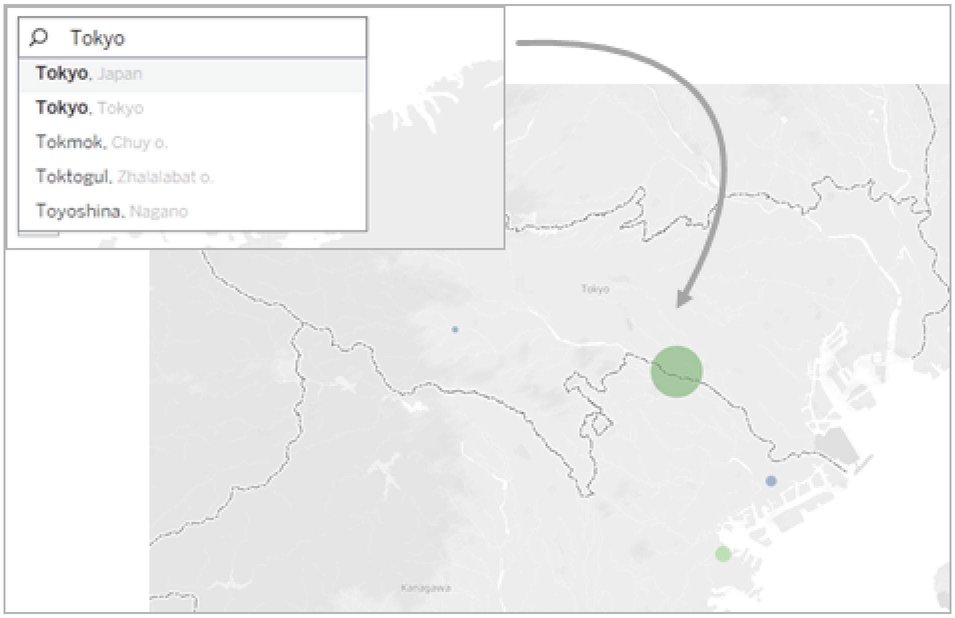
Follow the steps below to learn how to measure distance with the Radial tool.

Step 1: Zoom in to an area or location

The first step to measuring distance in maps with the Radial tool is to zoom in to an area or location in the map view. For more information about how to zoom in and out of the view, see **Zoom and Pan Views, and Select Marks** on page 2719.

You can also use map search to quickly navigate to a location in your map. For more information about how to use map search, see **Search for Locations in Your Map** on page 2006.

Note: You must zoom in to the map several times before you can measure distance with the Radial tool. The measured distance will not appear if you are zoomed too far out of the map. For more information, see the **Measurement accuracy** on page 2011 section.

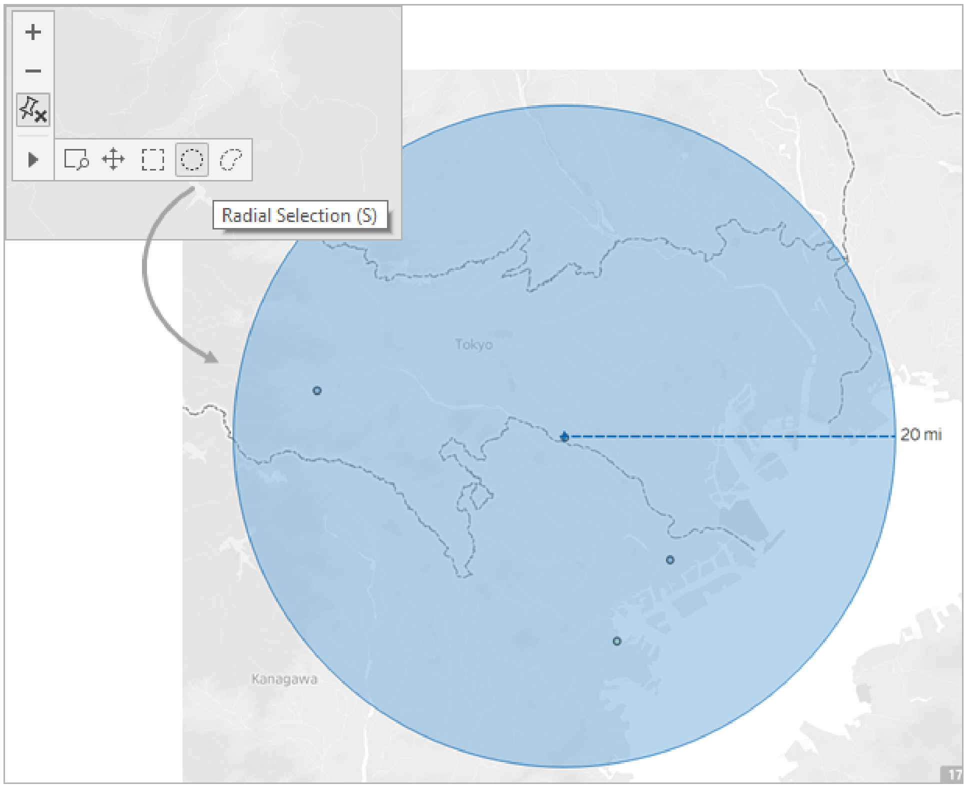


Step 2: Select the Radial tool

After you have zoomed in to a particular area or location in your view, select the Radial tool on the view toolbar, and then click and drag across the view. The measured distance appears to the right of the circle that appears when you drag across the view.

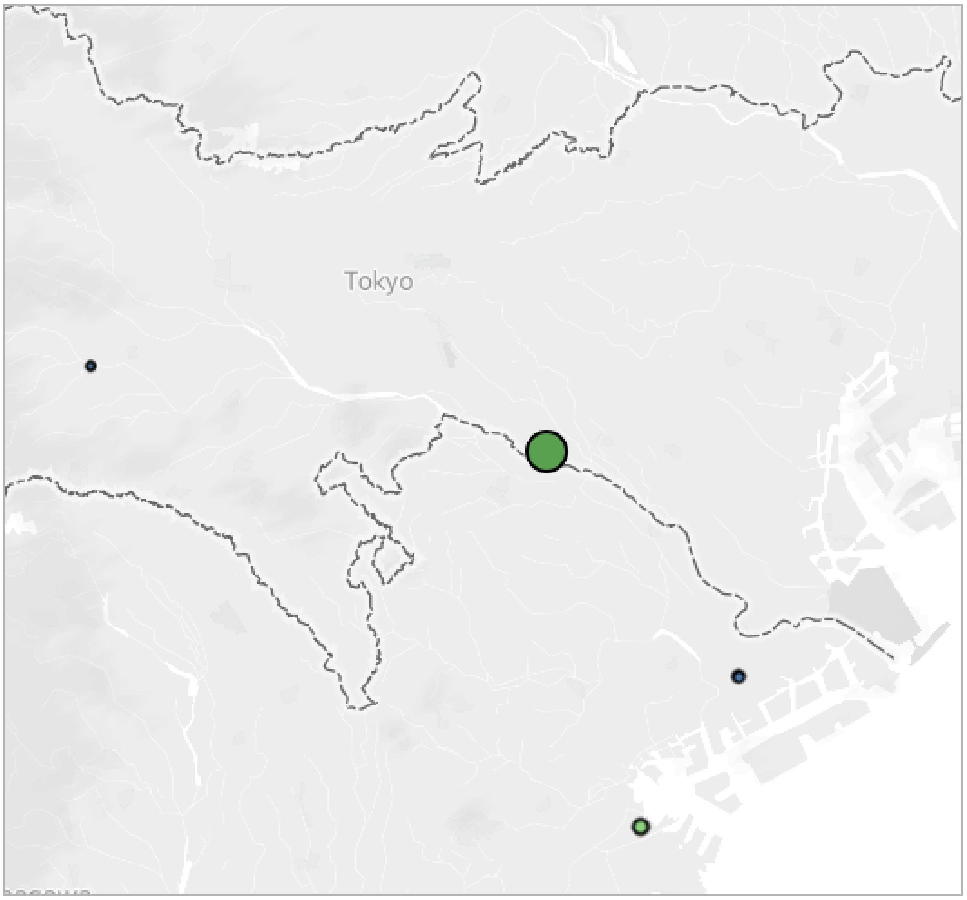
For more information about how to use the Radial tool, see **Select marks** on page 2722.

> **Note:** If you do not see a measured distance, you must zoom in further to a location or area in the view.



Note: If the view toolbar is hidden, press **S** on your keyboard to use the Radial tool.

As you drag, the Radial tool selects marks that are located within the radius of the circle. In this example, the radius is 100 miles, and it is centered over Jakarta. This means that, according to this data, all of the selected earthquakes (25) have occurred within approximately 100 miles of Jakarta in the past 10 years.



Measurement accuracy

By default, Radial tool measurements have a small margin of error because, in the map projection, distances become exaggerated and stretch as you move away from the equator. This means that the Radial tool can only measure approximate distances.

The Radial tool can measure distance more accurately the closer you are to the equator, and the further you zoom in to the view.

> **Note:** By design, the Radial tool does not display a measured distance when you are zoomed too far out of the view, because the measurement might be inaccurate.

Change the units of measurement

By default, your workbook locale determines which units the Radial tool uses to measure distance. If your workbook locale is set to a country that uses the Imperial system, the Radial tool measures distance in feet and miles. If your workbook locale is set to a country that uses the Metric system, the Radial tool measures distance in meters and kilometers.

You can change the units the Radial tool uses to measure distance for any map view in your workbook.

To do so, select **Map** > **Map Options**. Next, under **Units**, click the drop-down menu and choose from the following:

- If you want to measure distances in meters and kilometers, select **Metric**.
- If you want to measure distances in feet and miles, select **U.S.**
- If you want the units of measurement to be determined by your workbook locale, select **Automatic** .

The units you choose will be saved with the workbook and will remain if you publish the map view to Tableau Server, Tableau Online, or Tableau Public.

See also:

Show a map scale on page 1989

Map Storing

When you create map views using the online map provider, Tableau stores the images that make up the map in a cache. That way, as you continue your analysis you don't have to wait for the maps to be retrieved. In addition, by storing the maps, you can do a certain amount of work when you are offline. For more information, see **Select Background Maps** on page 1991.

The cache for the maps are stored with your temporary internet files and can be cleared at any time by deleting the temporary files from your browser.

Stored map images and legends remain valid for about thirty days. After that time, Tableau will not use the stored image; instead, it will require you to reconnect and fetch an updated map. This is to prevent the map images from becoming outdated.

Use Background Images in Your Views

Background images are images that you display underneath your data in order to add more context to the marks in the view. A common use of background images is adding custom map images that correspond to a coordinate system in your data.

For example, you might have data that corresponds to several floors in a building. You can use background images to overlay that data on the actual floor plan of the building to give more context. Other examples of using background images include showing a model of the sea floor, images of web pages for analyzing web logs, and even levels from video games to visualize player statistics.

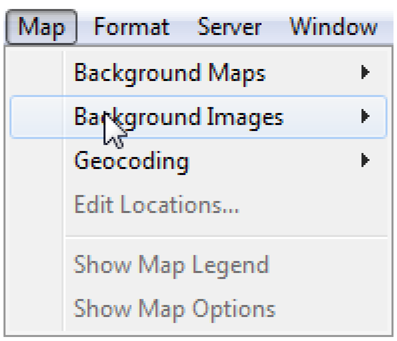
While Tableau allows you to load dynamic maps from the online and offline provider, background images allow you to use your own custom images whether they are special maps or any other image that corresponds to your data.

Add background images to your workbook

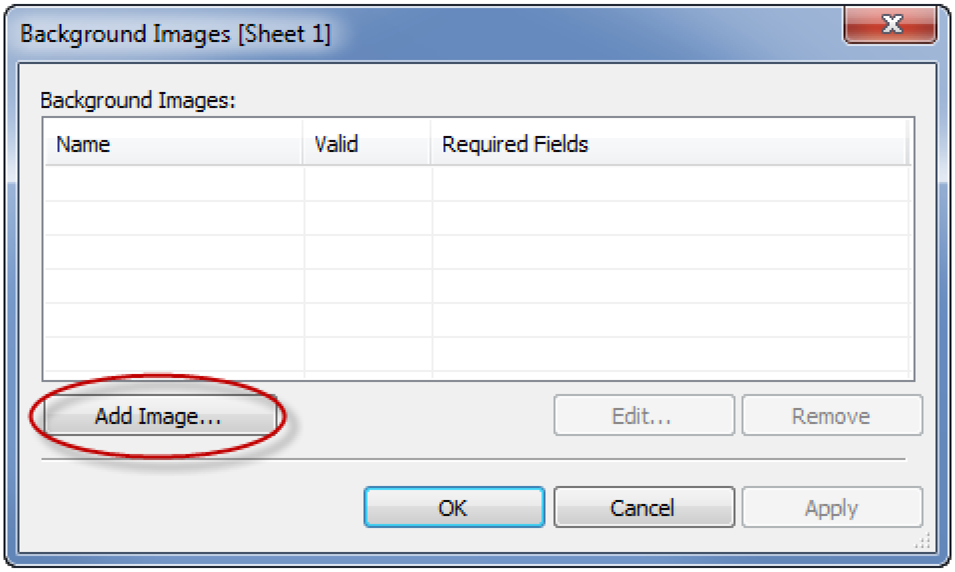
When you add a background image to the view, you need to specify a coordinate system by mapping both the X and Y axes to the values of fields in your database. If you are adding a map, the X and Y axes should be longitude and latitude expressed as a decimal. However, you can map the axes to any relevant fields based on your own coordinate system.

To add a background image:

1. Select **Map** > **Background Images** and then select a data source.



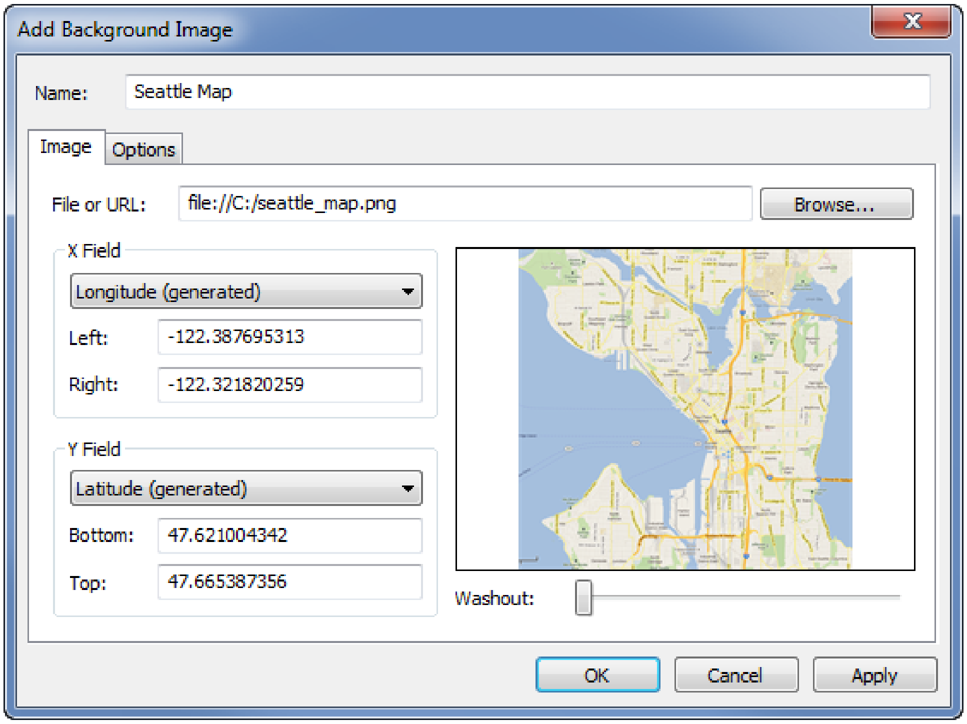
2. In the Background Images dialog box, click **Add Image**.



3. In the Add Background Image dialog box do the following:

   - Type a name for the image into the **Name** text box.
   - Click **Browse** to navigate to and select the image you want to add to the background. You can also type a URL to link to an image hosted online.
   - Select the field to map to the x-axis of the image and specify the left and right values. When adding a map, the longitude values should be mapped to the x-axis using decimal values (instead of degrees/minutes/seconds or N/S/E/W).
   - Select the field to map to the y-axis of the image and specify the top and bottom values. When adding a map, the latitude values should be mapped to the y-axis using decimal values (instead of degrees/minutes/seconds or N/S/E/W).
   - You can adjust the intensity of the image using the Washout slider. The farther the

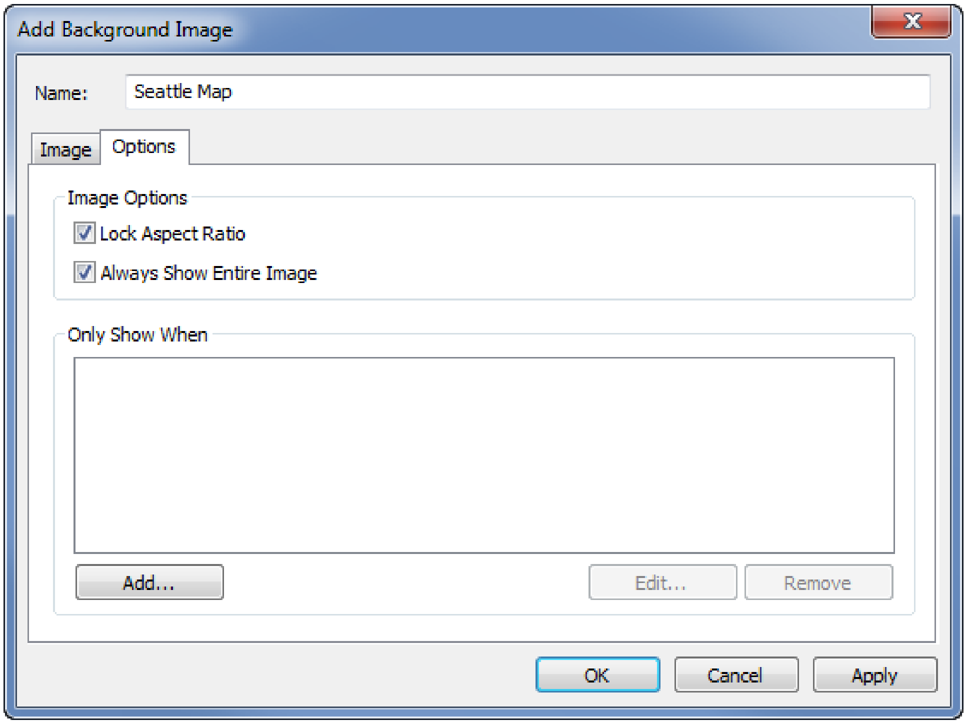
slider moves to the right, the more faded the image will appear behind your data.



4. You can specify the following options using the **Options** tab:

   - **Lock Aspect Ratio** - select this option to maintain the original dimensions of the image for any manipulations of the axes. Deselecting this option allows the image’s shape to be distorted.
   - **Always Show Entire Image** - select this option to avoid cropping the image when the data encompasses only a portion of the image. If you lock both the axis in a view, this option may be negated.
   - Add conditions for when to show the image. Refer to **Filter Data from Your**

Views on page 1411 to learn more about defining conditions.



5. Click **OK**.

When you add the x and y fields to the Rows and Columns shelf in the view, the background image displays behind the data. If the background image does not display, make sure that you are using the disaggregated measures for the x and y fields. To disaggregate all measures, select **Analysis > Aggregate Measures**. To change each measure individually, right-click the field on the shelf and select **Dimension**. Finally, if you've used the generated Latitude and Longitude fields for the x and y fields, you'll need to disable the built in maps before your background image will display. Select **Map > Background Maps > None** to disable the built-in maps.

> In order to make the marks in a view more visible when placed on top of a background image, each mark is surrounded by a solid contrasting color called a halo. You can turn mark halos off by selecting **Format** > **Show Mark Halos**.

Build a view with a background image

After you add a background image, you need to build the view in a way that matches the x and y mappings you specified for the image. That is, the fields you specified as x and y must be on the proper shelves. Follow the steps below to set up the view correctly:

1. Place the field mapped to the x-axis on the **Columns** shelf.

   If you are working with maps, the longitude field should be on the columns shelf. It may seem backward at first, however, the fields on the columns shelf determine the values distributed across the x-axis.

2. Place the field mapped to the y-axis on the **Rows** shelf.

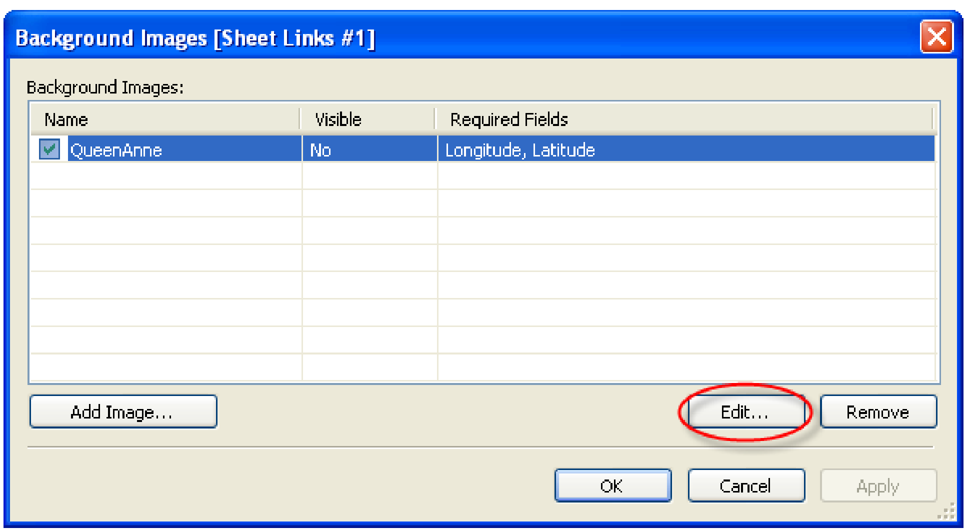
   If you are working with maps, the latitude field should be on the rows shelf. It may seem backward at first, however, the fields on the rows shelf determine the values distributed across the y-axis.

Edit a background image

After adding a background image, you can always go back and edit the x and y field mappings as well as any of the options on the Options tab.

To edit an image:

1. Select **Map** > **Background Images**.
2. In the Background Images dialog box, select the image you want to edit and click **Edit** (you can also just double-click the image name).



3. In the Edit Background Image dialog box, make the changes to the image and click **OK**.

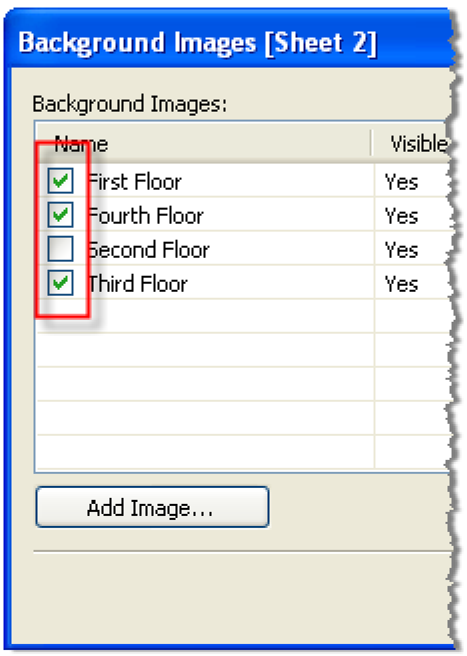
Enable or disable background images

Although you can add multiple images to a workbook, you may want to only use a subset of the images for a particular set of worksheets. For example, you may want to show a map of the entire United States of America on one view, and maps of individual states in other views.

Use the check boxes in the Background Images dialog box to enable and disable the images for the current worksheet. You can show several images by enabling multiple images on a single worksheet. For example, you may have several images that you want to tile in the background to make a larger background image.

To enable or disable a background image:

1. Select **Map** > **Background Images**.
2. In the Background Images dialog box, select the check boxes next to the images you want enabled.



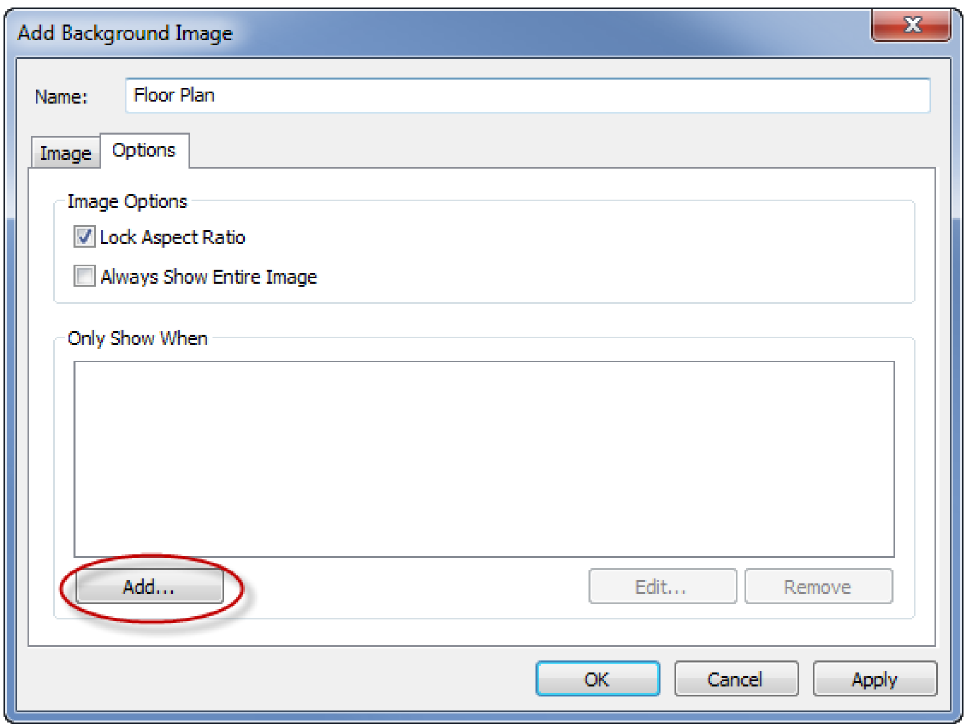
3. Click **OK**.

Add show and hide conditions to background images

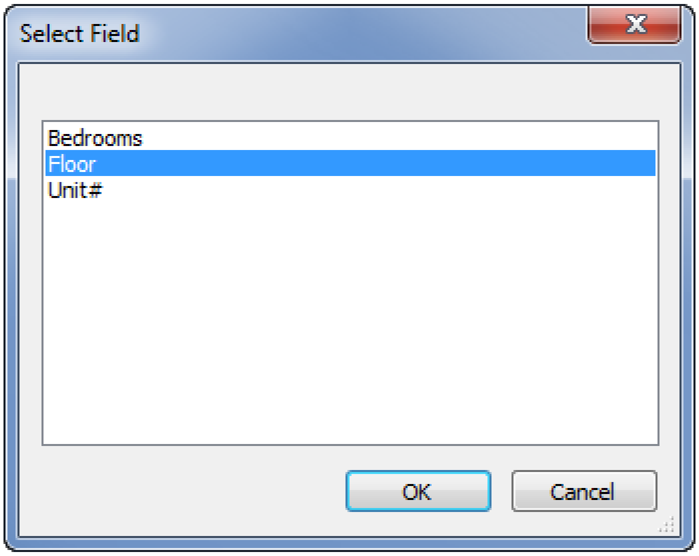
When you add a background image and enable it, the image will be shown automatically on any worksheet that has the required fields used in the view. To avoid showing an image on all the worksheets, you can specify show/hide conditions. Show/Hide conditions are conditional statements that you define to specify when to show the image. For example, you may have a floor plan image for a multi-story building. While each image is associated with the same coordinates (the corners of the building), you do not want to show the first floor map when looking at the third floor information. In this case, you can specify a condition to only show the first floor image when the Floor field is equal to one.

To specify show/hide conditions:

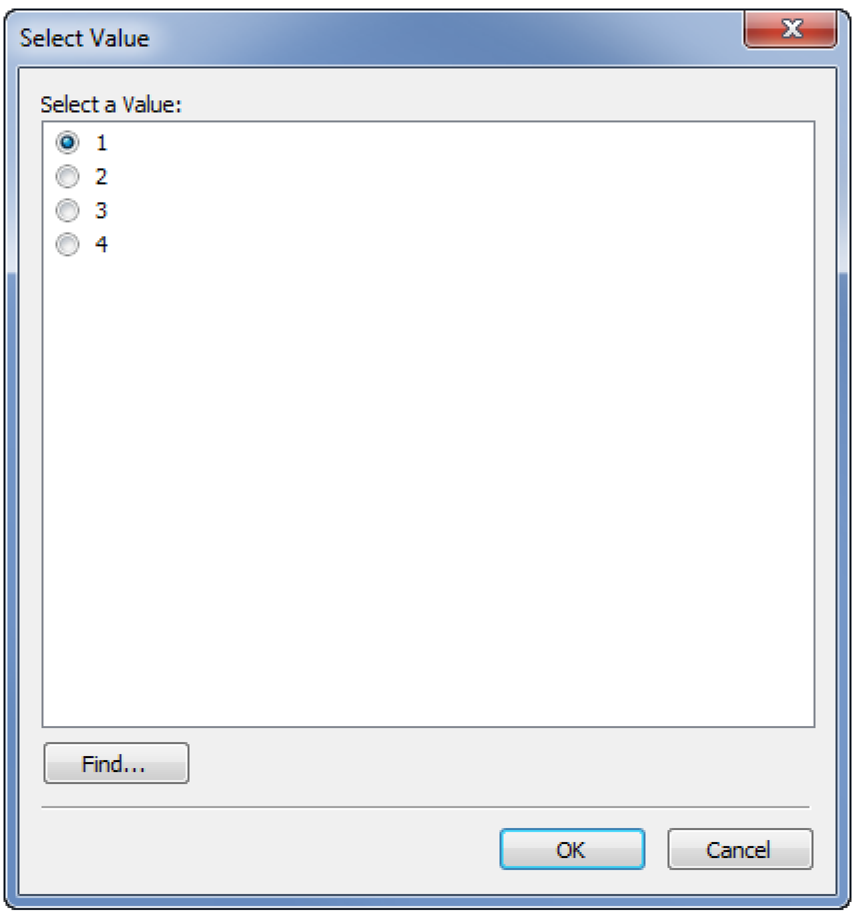
1. Select **Map > Background Images** and then select a data source.
2. In the Background Images dialog box, select the image you want to add a condition to and click **Edit**.
3. In the subsequent dialog box, select the **Options** tab.
4. Click the **Add** button at the bottom of the dialog box.



5. Select a field to base the condition on. In the example described above, the field is Floor.



6. Specify when you want to show the image by selecting a value of the field. For this example, one is selected.



7. Click **OK**.

   A condition statement is added to the image. In the building floor plan example, the condition statement is Only show the image when Floor is equal to One.

8. Click **OK** twice to close the Background Image dialog boxes and apply the changes.

When you add multiple conditions, the background image will only show when all conditions are met. For example, if a background image has two conditions on Property Name and Floor, it will only show when Property is Greenwood Estates and Floor is 3.

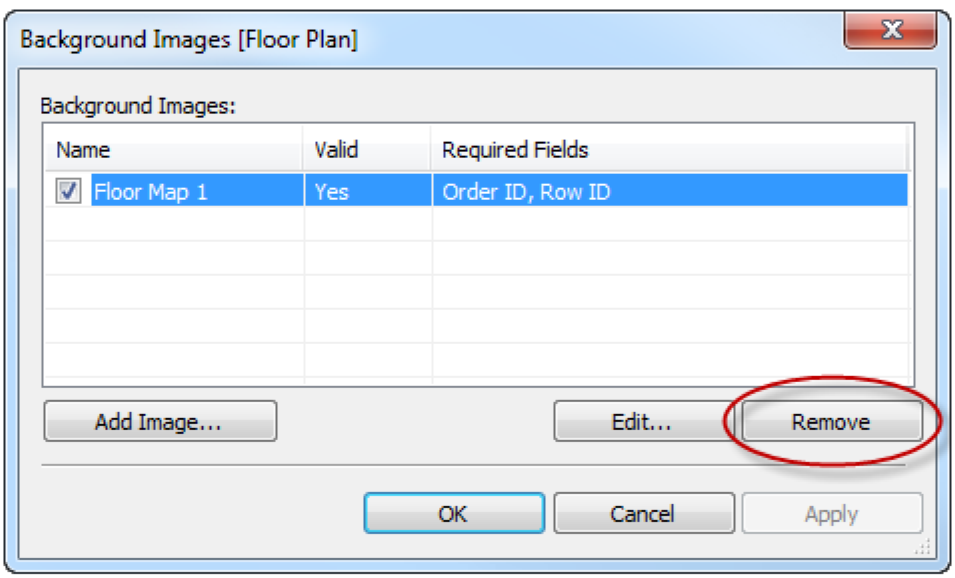
Remove a background image

When you no longer want to use a background image you can either disable it or remove it, making it unavailable to all worksheets.

To remove an image:

1. Select **Map** > **Background Images**.

2. In the Background Images dialog box, select the image you want to remove and click **Remove**.



3. Click **OK**.

See Also

- **Maps and Geographic Data Analysis in Tableau** on page 1755

Add Images of Google Maps and OpenStreetMap as Background Images in Tableau

You can add images of Google Maps or OpenStreetMap as static background images in Tableau, and then plot data points onto them. To plot data points onto a background image of GoogleMaps or OpenStreetMap, you need both latitude and longitude coordinates of your data.

This topic describes how to obtain latitude and longitude coordinates for Google Maps and OpenStreetMap, as well as how to plot them on a background image in Tableau.

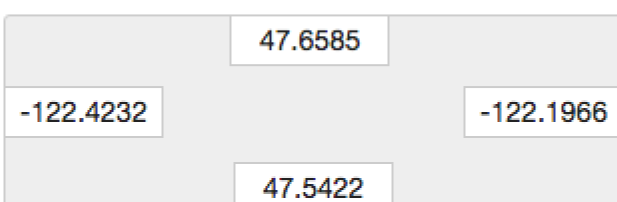
Step 1: Find your map coordinates

Follow the steps below to find map coordinates for Google Maps or OpenStreetMaps.

Option 1: Map data points onto Google Maps

1. Go to http://www.gorissen.info/Pierre/maps/googleMapLocation.php and locate the geographic area that you want to plot data points.
2. Once you have located the area you want, hover over the easternmost side of the area and write down the longitude coordinate (displayed in the bottom-right corner of the map). Do the same for the westernmost side of the area.
3. Hover over the northernmost part of the area and write down latitude coordinate. Do the same for the southernmost part of the area.

   You should now have two latitude coordinates (one top and one bottom) and two longitude coordinates (one east and one west).



You will use these coordinates to define the scale of your background image in Tableau, and to ensure that you can plot data points onto the image using latitude and longitude coordinates.

4. Take a screenshot of your map and save it to your computer.
5. Follow the steps in "**Step 2: Add your map image as a background image** below" to add the image as a background image in Tableau.

Option 2: Map data points onto a map in OpenStreetMap

1. Go to http://www.openstreetmap.org and zoom in on the geographic area you want to plot data points.
2. On the top menu bar, click **Export**.

   The Export pane provides the latitude and longitude coordinates of the area in the view. You can edit these coordinates to select a different area.
3. Write down the coordinates displayed in the box.
4. On the right side of the page, click the share icon, and then click the **Download** button.
5. Follow the steps listed in "**Step 2: Add your map image as a background image** below" to add the image as a background image in Tableau.

Step 2: Add your map image as a background image

1. Open Tableau Desktop and connect to the data you want to plot on the map image.
2. Select **Maps**> **Background Images**, and select the data source you want to use to plot data on your background image.
3. In the Background Images dialog box, click **Add Image**.
4. In the Add Background Image dialog box, click **Browse**, and then navigate to the location of the map image you created earlier in Step 1: Find your map coordinates.
5. In the Add Background Image dialog box, do the following:
   - For the **X Field** drop-down list, select **Longitude (generated)**.
   - In the **Left** box, type the easternmost longitude in your map image.
   - In the **Right** box, type the westernmost longitude in your map image.
   - For the **Y Field** drop-down list, select **Latitude (generated)**.

- In the **Bottom** box, type the southernmost latitude in your map image.
- In the **Top** box, type the northernmost latitude in your map image.

6. When finished, click **OK** in both the Add Background Image and Background Images dialog boxes.
7. Select **Maps**>**Background Maps**>**None**.

   The background image shows up only if there is no background map.
8. From the Measures pane, drag **Latitude (generated)** to the **Rows**shelf and **Longitude (generated)** to the **Columns**shelf.

   Your background image should appear in the view. If it does not, or if it appears at the wrong scale, complete the following steps to fix your axes.
9. In the view, double-click the Y axis.
10. In the Edit Axis dialog box, under Range, select **Fixed**.
11. In the **Fixed start** text box, enter the bottom most longitude in your map image.
12. In the **Fixed end** text box, enter the top most longitude in your map image.
13. Click **OK**.
14. Repeat steps 9 - 13 for the other axis.

To learn how to plot your data on the background image, see steps 1, 2, and 4 in **Find Background Image Coordinates** below.

See also:

Use Background Images in Your Views on page 2012

Find Background Image Coordinates below

Use Mapbox Maps on page 1993

Use Web Map Service (WMS) Servers on page 1996

Select Background Maps on page 1991

Find Background Image Coordinates

In many scenarios, you might want to map your data onto a background image instead of on a Tableau map. This article explains how to plot data points on a background image.

There are several steps to this process:

1. Create a table in your data source for X and Y coordinates.
2. Connect to your data source in Tableau Desktop and join the coordinate table with the rest of your data.
3. Import your background image and build the view.
4. Annotate points on the background image.
5. Add coordinates to the coordinates table in your data source.
6. Refresh your data source in Tableau.

Step 1: Create the coordinate table

Create a table in your data source that contains columns to identify a unique mark. In this example, there are three columns:

- The first column holds the unique identifier for each mark.
- The other two columns are for the X and Y coordinates.

After you create your columns, you join the new coordinate table to the original data source. The join is based on the unique identifier used in both the original data source and the coordinate table (the identifier that represents a single mark).

1. Select a scale that is appropriate for your background image. If the image is wider than it is tall, use X: 0-100 and Y: 0-50.
2. For this example, open the Sample Superstore Excel file from your Tableau repository in Excel and select a new sheet tab.
3. On the new tab, put a single row in a table that contains one of the unique identifiers and the X and Y end points you selected for your two scales.

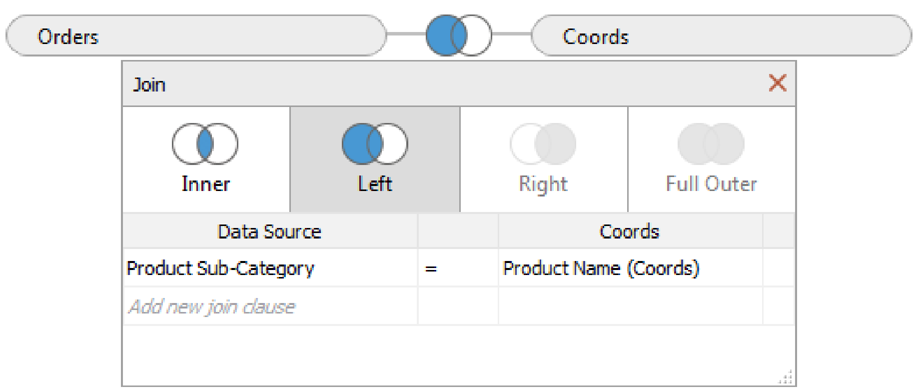
   In this case, **Product Sub-category** is a column in Sample Superstore, and one of its members is **Tables**. The **X** endpoint is **100**, and the **Y** endpoint is **50** because that is the scale for the background image.

	A	B	C	D
1	Product Sub-category	X	Y	
2	Tables	100	50	
3				
4				
5				

4. Change the tab name to **Coords** and save the file.

Step 2: Connect to the coordinate table

1. In a new Tableau workbook, select **Connect to Data**.
2. In the **Connect to Data** dialog box, select **Microsoft Excel** and then click **Next**.
3. On the data source page, do the following:
   - Drag the Orders sheet to the Join area.
   - Drag the Coords sheet to the Join area.
   - Click the join icon to modify the existing join.
   - In the Join dialog box that opens, select **Left**, and then select **Product Sub-Category** for the **Data Source**column, and **Product Name (Coords)** for the **Coords** column.



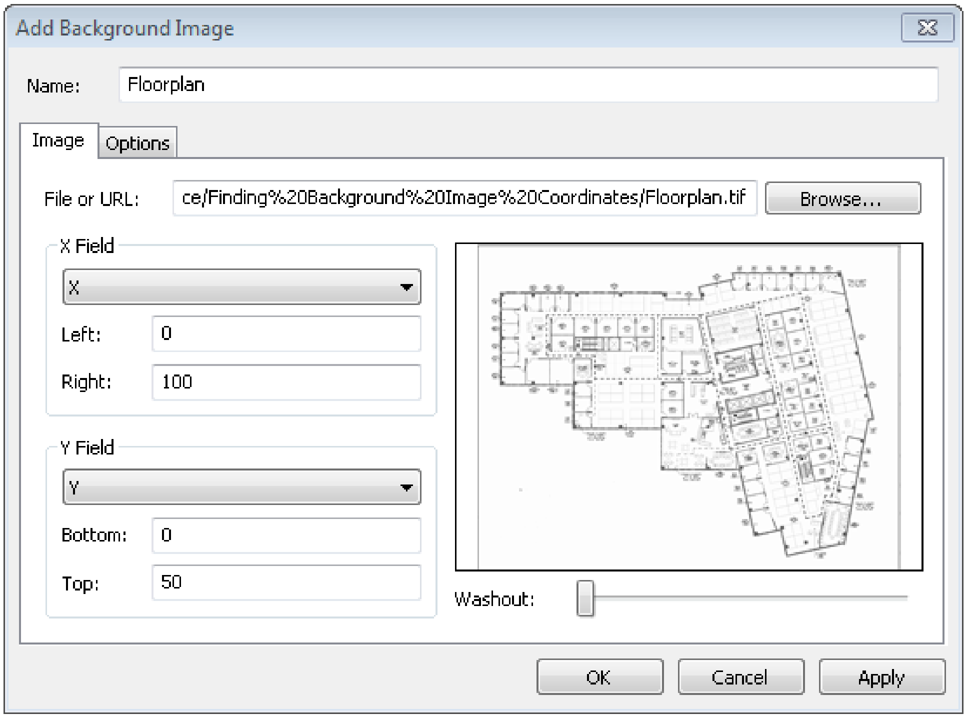
4. Close the Join dialog box, and then select the sheet tab to go to a worksheet.

Step 3: Import the background image

1. In the new worksheet, select **Maps** > **Background Images**, **Sample - Superstore**. This opens the Background Images dialog box.
2. In the **Background Images** dialog box, click **Add Image**.
3. In the **Add Background Image** dialog box, browse to and select the file.
4. For **X Field**, select **X** in the list.

   > **Note:** If you do not see the X field in the list, it means that you need to change the X field to be a continuous number.
   >
   > To change the field to be a number, right-click the **X** field in the Data pane, and select **Change Data Type** > **Number (Whole)**. To convert the field to be continuous, right-click the **X** field in the Data pane, and select **Convert to Continuous**.
   >
   > Repeat these steps for the **Y** field.

5. In the **Right** text box, type **100** (100 is the number you used in the coordinate table).
6. For **Y Field**, select **Y** in the list.
7. In the **Top** text box, type **50** (50 is the number you used in the coordinate table). Be sure to type in the correct text box.



8. Click **OK**, and then click **OK** in the **Background Images** dialog box. (Disregard the **Valid** setting of **No** in the **Background Images** dialog box.)

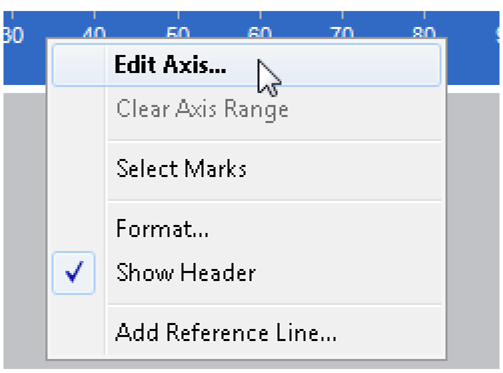
Step 4: Find the coordinates

1. From the Data pane, drag **X** to the **Columns** shelf and **Y** to the **Rows** shelf.
2. On the **Columns** shelf, right-click **X** and select **Dimension**.
3. On the **Rows** shelf, right-click **Y** and select **Dimension**.

> **Note:** This step is only necessary if the X and Y fields were brought in to Tableau as Measures.

The background image appears in the view.

4. Right-click the X axis and select **Edit Axis**.



5. In the **Edit Axis** dialog box, select **Fixed**, and then in the **Start** text box, type **0**, and in the **End** text box, type **100**.
6. Click **OK**.
7. Right-click the Y axis and select **Edit Axis**.

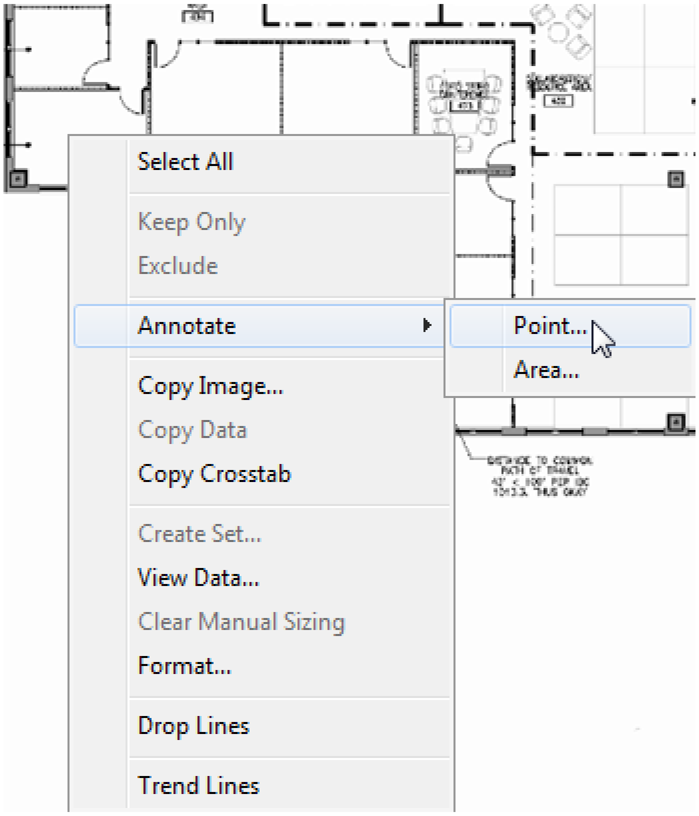
8. In the **Edit Axis** dialog box, select **Fixed**, and then in the **Start** text box, type **0**, and in the **End** text box, type **50**.

9. Click **OK**.

   Now the axes start at zero.

10. Right-click anywhere on the image where you want to find the coordinates, and select **Annotate** > **Point**.

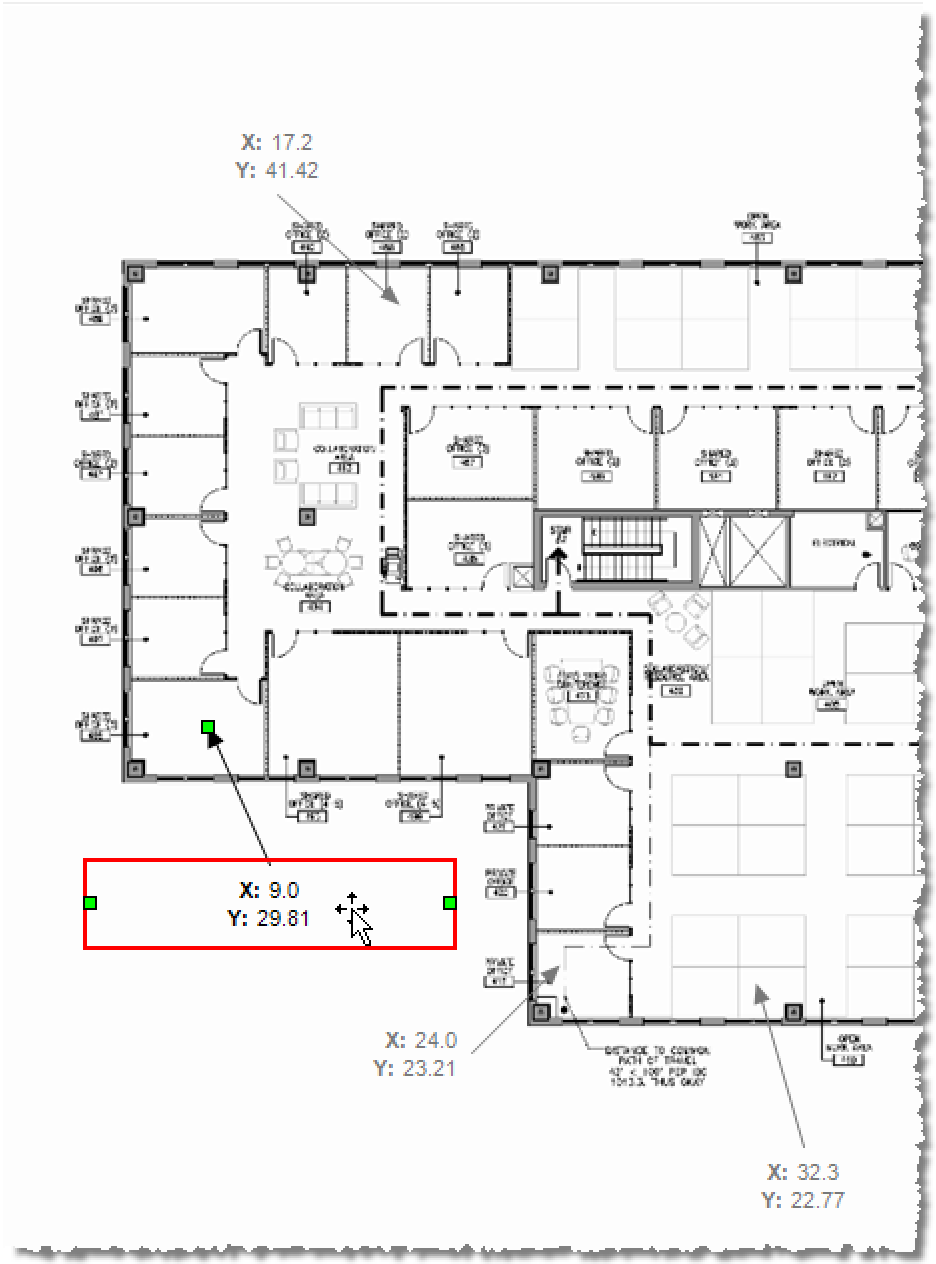
    For more information about annotating marks or points in the view, see the **Add Annotations** on page 1465 topic in the Tableau Help.



11. In the **Edit Annotation** dialog box, click **OK**.

12. Repeat steps 10 and 11 for each point you want to annotate.

    The coordinates appear on the background image as a callout. You can move a callout to a location where it shows up better; just click and drag it. You can also resize the callout by dragging one of the size handles. You can move a coordinate point to a new location by clicking and dragging the arrowhead; the coordinates in the callout change to reflect the new location.



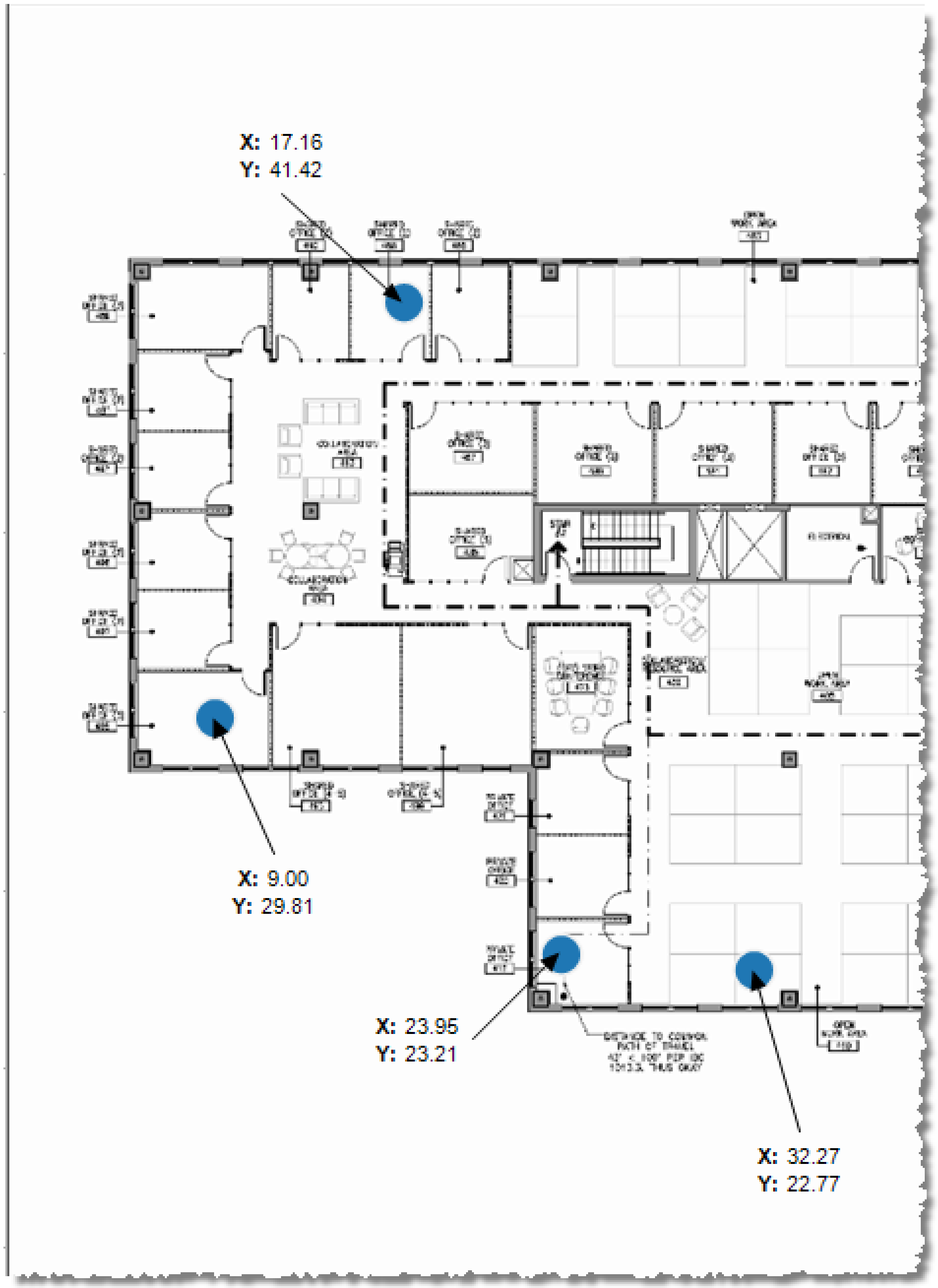
13. Add these coordinates to the Coords sheet in the Sample - Superstore Excel file and save the changes.

	A	B	C	D
1	Product Sub-category	X	Y	
2	Tables	100	50	
3	Labels	17.16	41.42	
4	Paper	9.00	29.81	
5	Bookcases	23.95	23.21	
6	Office machines	32.27	22.77	
7				
8				

14. In Tableau, select **Data** > **Sample - Superstore** > **Refresh**.

    The refreshed view displays marks at the coordinate locations.

    If you don't want to continue to display the callouts, you can select and remove them on the background image.



Other Ideas

Try placing another Measure on the Size shelf and put either a Dimension or Measure on Color on the Marks card. Add your own custom shapes. Develop powerful and unique analyses by creating Actions between this view and your other analyses.

If you want to hide the X and Y scales, right-click the respective field on the Columns or Rows shelf and clear **Show Header**.

See also

- **Use Background Images in Your Views** on page 2012
- **Maps and Geographic Data Analysis in Tableau** on page 1755

Analyze Data

This section describes the various ways you can perform more advanced analysis in Tableau. Read the following articles for information on how to create calculated fields, find clusters in data, calculate percentages, and use various tools to explore and inspect data.

Watch a video: For a 3-minute introduction to calculations, watch the Getting Started with Calculations free training video. Use your tableau.com account to sign in.

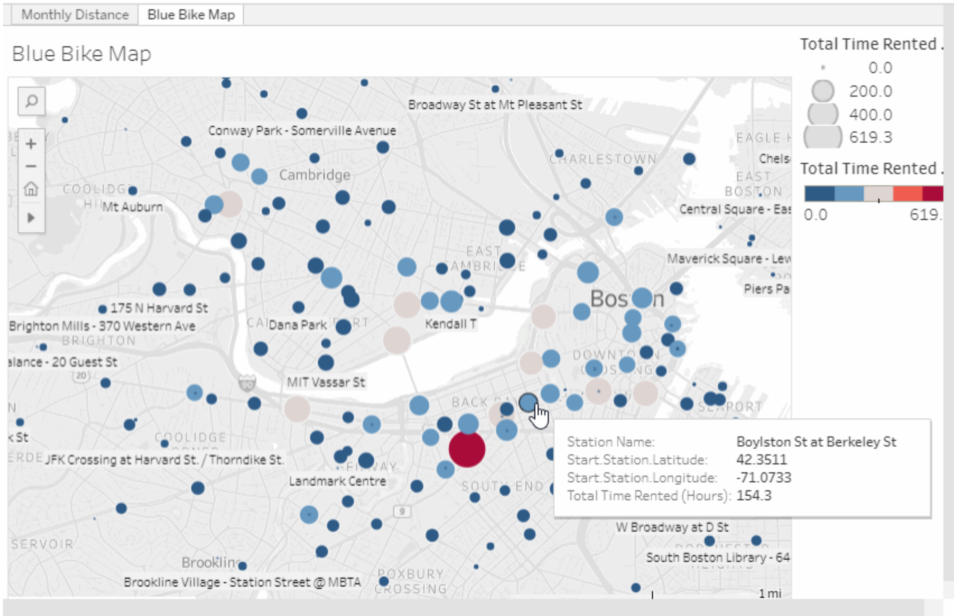
For a 7-minute walk through on how to create reference lines in Tableau, watch the Reference Lines free training video. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Discover Insights Faster with Explain Data

Explain Data gives you a new window into your data. Use it to inspect, uncover, and dig deeper into the marks in a viz as you build, explore, and analyze your data. When you select a mark while editing a view and run Explain Data, Tableau builds statistical models and proposes possible explanations for the selected mark, including potentially related data from the data source that isn't used in the current view.

For information on running Explain Data and exploring explanations, see **Get Started with Explain Data** on page 2037.

Note: This topic describes how Explain Data works in Tableau 2021.2 and later versions. If you have a previous version of Tableau, read this topic in version 2021.1 of Explain Data help.



Select a mark and run Explain Data from the tooltip menu. The Explain Data pane opens with possible explanations

As you build different views, use Explain Data as a jumping-off point to help you explore your data more deeply and ask better questions. For more information, see **How Explain Data helps to augment your analysis** below. For information on what characteristics make a data source more interesting for use with Explain Data, see **Requirements and Considerations for Using Explain Data** on page 2070.

Access to Explain Data for Authors vs. Viewers

Explain Data is always available to authors who can edit a workbook.

Authors can control whether Explain Data is available in viewing mode in published workbooks and which explanation types are displayed in the Explain Data pane.

When Explain Data is enabled by the author of a published workbook, all users can run Explain Data on a mark in viewing mode.

For information on controlling access to Explain Data, explanation types, and fields, see **Control Access to Explain Data** on page 2072.

How Explain Data helps to augment your analysis

Explain Data is a tool that uncovers and describes relationships in your data. It can't tell you what is causing the relationships or how to interpret the data. **You are the expert on your**

data. Your domain knowledge and intuition are key in helping you decide what characteristics might be interesting to explore further using different views. For related information, see **How Explain Data Works** on page 2083 and **Requirements and Considerations for Using Explain Data** on page 2070.

For more information on how Explain Data works and how to use Explain Data to augment your analysis, see these Tableau Conference presentations:

- From Analyst to Statistician: Explain Data in Practice (1 hour)
- Leveraging Explain Data (45 minutes)
- Explain Data Internals: Automated Bayesian Modeling (35 minutes)
- Machine Learning, Explainable AI, and Tableau (45 minutes), Session Materials
- Making Business More Bayesian (45 minutes)

Get Started with Explain Data

Use Explain Data in your flow of analysis as you are exploring the marks in a viz. The best way to get started with Explain Data is to select a mark, run Explain Data, and start exploring explanations.

Use Explain Data

- **Run Explain Data on a mark** below
- **Drill into explanations** on the next page
- **View analyzed fields** on page 2044
- **Why am I seeing a Reselect button?** on page 2042
- **Terms and concepts in explanations** on page 2045

- Explanation Types

Author Workbooks and Control Access

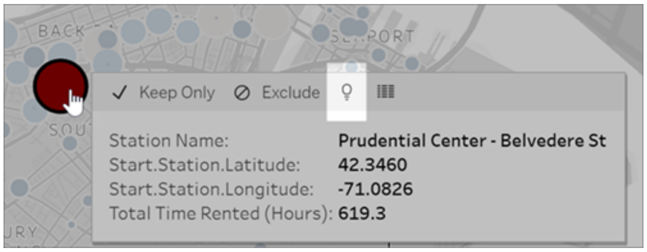
- Requirements and Considerations for Using Explain Data
- Change Explain Data Settings (Authors-only)

- **Control Access to Explain Data** on page 2072
- **How Explain Data Works** on page 2083

Run Explain Data on a mark

The basic steps to run Explain Data are:

1. Select a mark in a viz.
2. Hover the cursor over the mark, and then click **Explain Data** in the tooltip menu.



The Explain Data pane opens with possible explanations for the value of the explained mark. Click different explanation names to expand the details and start exploring.

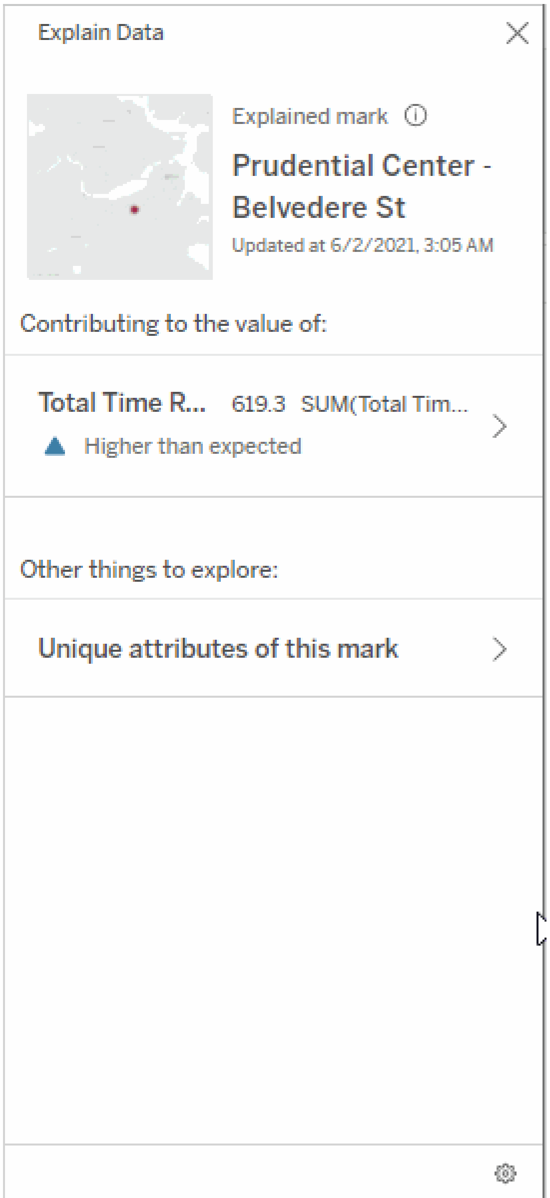
Tips for using Explain Data

- You must select a single mark—only one mark can be explained at a time.
- The view must contain marks that are aggregated using SUM, AVG, COUNT, COUNTD, or AGG (a calculated field).
- If Explain Data cannot analyze the type of mark selected, the Explain Data icon and context menu command will not be available. For more information, see Situations where Explain Data is not available.
- The data you analyze must be drawn from a single, primary data source. Explain Data does not work with blended or cube data sources.
- For information on what characteristics make a data source more interesting for use with Explain Data, see **Requirements and Considerations for Using Explain Data** on page 2070.

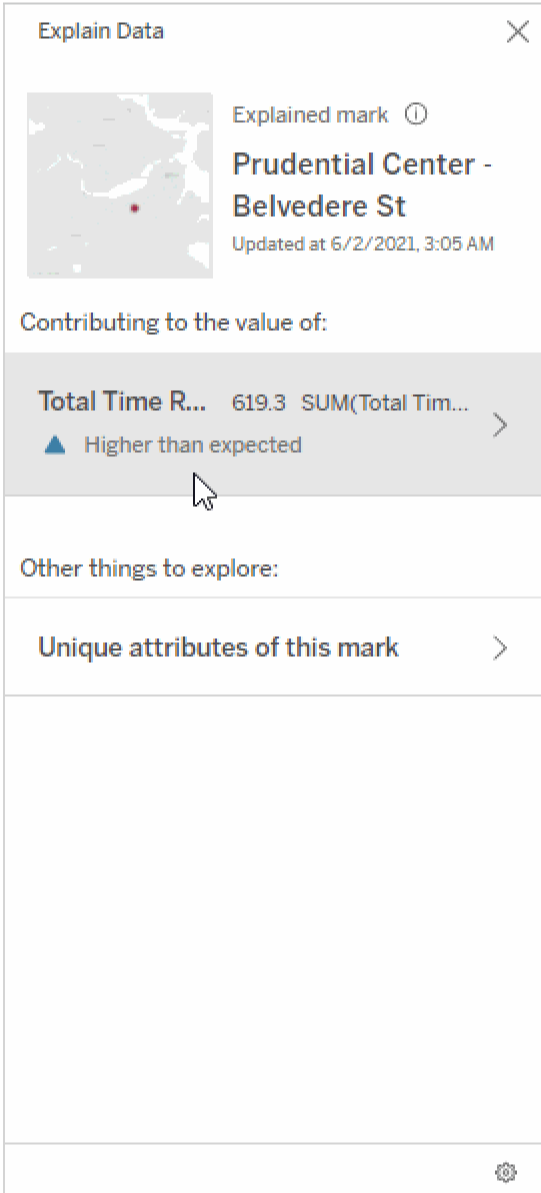
Drill into explanations

1. In the Explain Data pane, click an explanation name to see more details.

   Click the arrows to expand or contract explanations.

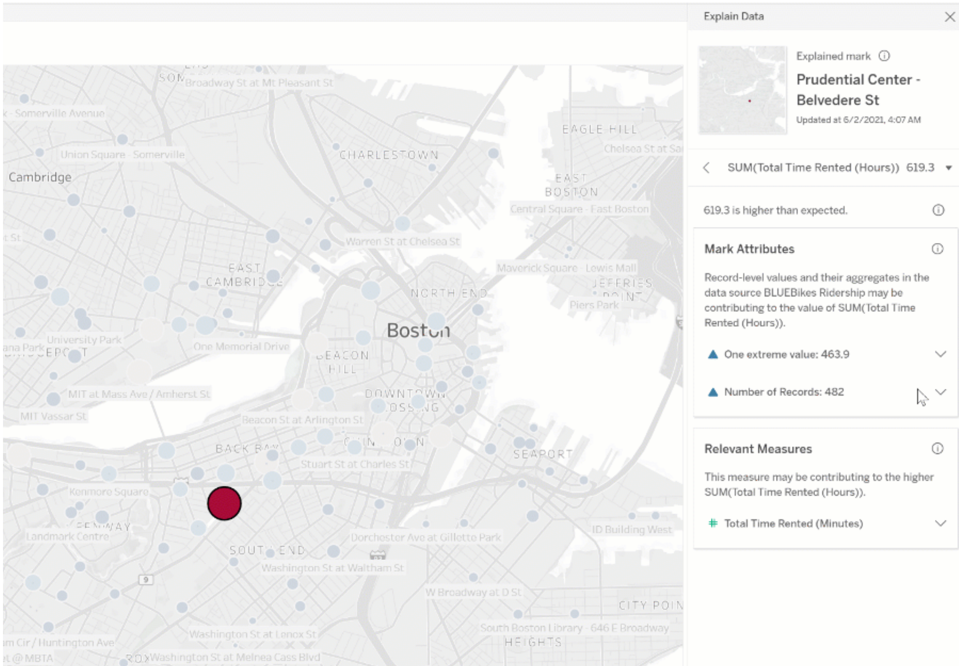


2. Scroll to see explanation details.



Some explanations offer different layers of details for further exploration

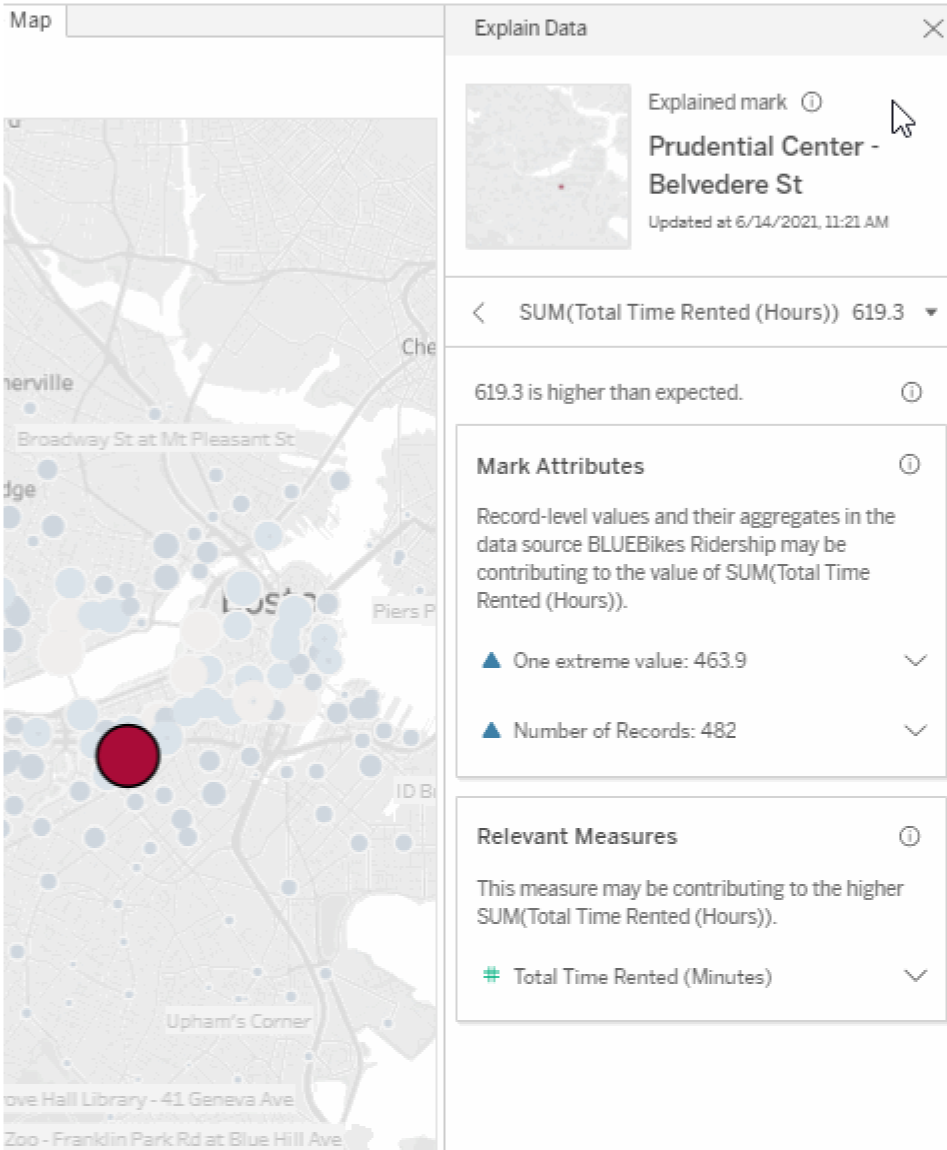
3. Hover over charts in the explanations to see more detail. Click the **Open** icon to see a larger version of the visualization.



Creators or Explorers who open the view for editing can click the **Open** icon to open the visualization as a new worksheet and explore the data further.

> **Note**: Creators and Explorers who have editing permissions can also control Explain Data Settings. For more information, see Control who can use Explain Data and what they can see.

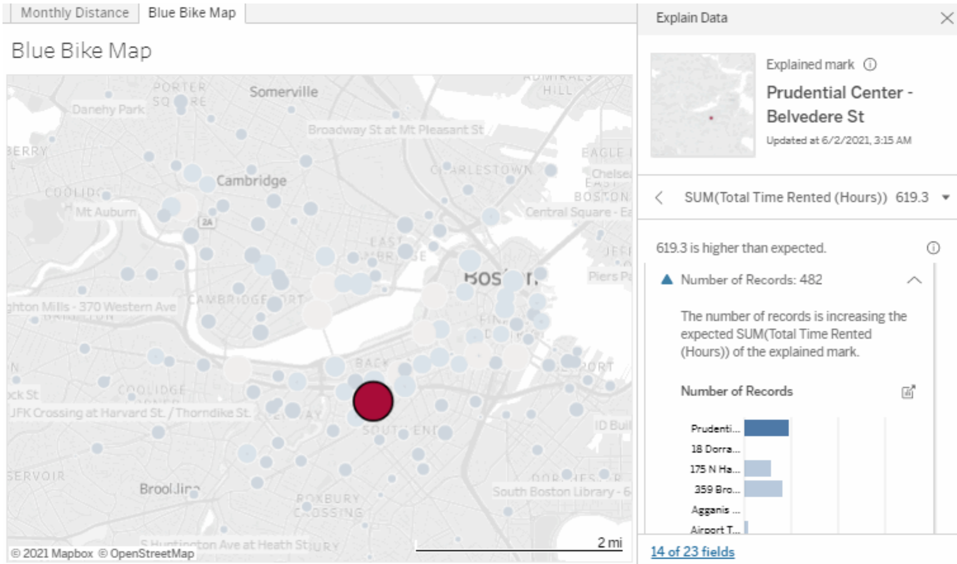
4. Hover over a Help icon ⓘ to see tooltip help for an explanation. Click the Help icon ⓘ to keep the tooltip open. Click a **Learn More** link to open the related help topic.



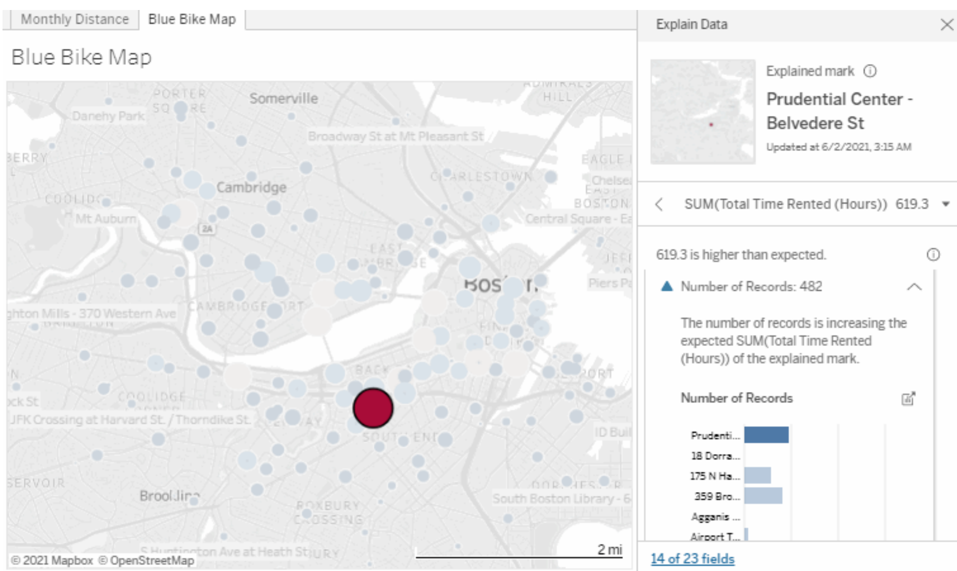
Why am I seeing a Reselect button?

When you run Explain Data, the explanations that are presented in the Explain Data pane specifically apply to the mark you selected. If you click a different mark, deselect the explained mark, or navigate to a different sheet in the workbook, you might see a message asking if you want to reselect the same mark.

If you click **Reselect**, Tableau returns you to the original view and worksheet, and reselects the explained mark.



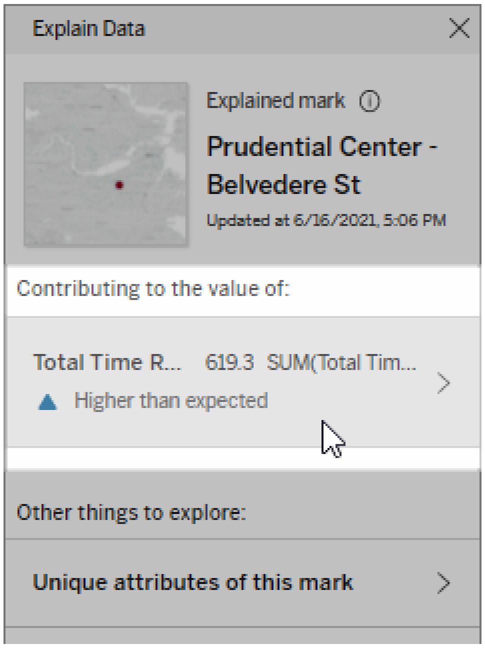
If you want to explore a new mark, ignore the Reselect message, click another mark, and then run Explain Data again.



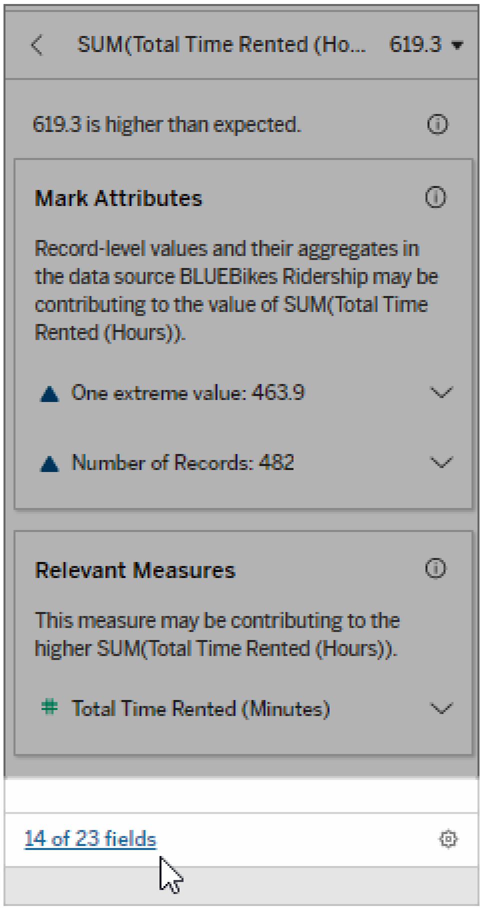
Explain Data will also display messages if the view has changed (such new fields or filters added or removed from the view), if the data source has changed, or if Explain Data settings have changed.

View analyzed fields

1. **Run Explain Data on a mark** on page 2037.
2. In the Explain Data pane, under **Contributing to the value of**, click a measure name.



3. Click the *number-of-fields* link at the bottom of the pane.



Authors have the option to open Explain Data Settings to control which fields are included in the analysis. For more information, see Change fields used for statistical analysis.

Terms and concepts in explanations

The following terms and concepts appear frequently in explanations. You may find it helpful to become acquainted with their meaning in the context of using Explain Data.

What is a mark?

A mark is a selectable data point that summarizes some underlying record values in your data. A mark can be made of a single record or multiple records aggregated together. Marks in Tableau can be displayed in many different ways such as lines, shapes, bars, and cell text.

Tableau gets the records that make up the mark based on the intersection of the fields in the view.

The **explained mark** refers to a mark that you selected in the view that was analyzed by Explain Data.

For more information on marks, see Marks.

What does expected mean?

The expected value for a mark is the median value in the expected range of values in the underlying data in your viz. The expected range is the range of values between the 15th and 85th percentile that the statistical model predicts for the explained mark. Tableau determines the expected range each time it runs a statistical analysis on a selected mark.

If an expected value summary says the mark is *lower than expected* or *higher than expected*, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is *slightly lower* or *slightly higher* than expected or *within the range of natural variation*, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher than the median.

For more information, see What is an expected range?

What are dimensions and measures?

Each column name in a database is a field. For example, Product Name and Sales are each fields. In Tableau, fields like Product Name that categorize data are called dimensions; fields with quantifiable data like Sales are called measures. Tableau aggregates measures by default when you drag them into a view.

Some explanations describe how the underlying record values and the aggregations of those values may be contributing to the value of the explained mark. Other explanations may mention the distribution of values across a dimension for the explained mark.

When you run Explain Data on mark, the analysis considers dimensions and measures in the data source that aren't represented in the view. These fields are referred to as unvisualized dimensions and unvisualized measures.

For more information on dimensions and measures, see Dimensions and Measures.

What is an aggregate or aggregation?

An aggregate is a value that is a summary or total. Tableau automatically applies aggregations such as SUM or AVG whenever you drag a measure onto Rows, Columns, a Marks card option, or the view. For example, measures are displayed as SUM(Sales) or AVG(Sales) to indicate how the measure is being aggregated.

To use Explain Data, your visualization must use a measure that is aggregated with SUM, AVG, COUNT, COUNTD, or AGG.

For more information about aggregation, see Data Aggregation in Tableau.

What is a record value?

A record is a row in a database table. A row contains values that correspond to each field. In this example, Category, Product Name, and Sales are fields (or columns). Furniture, Floor Lamp, and $96 are the values.

Category	Product Name	Sales
Furniture	Floor Lamp	$96.00

What is a distribution?

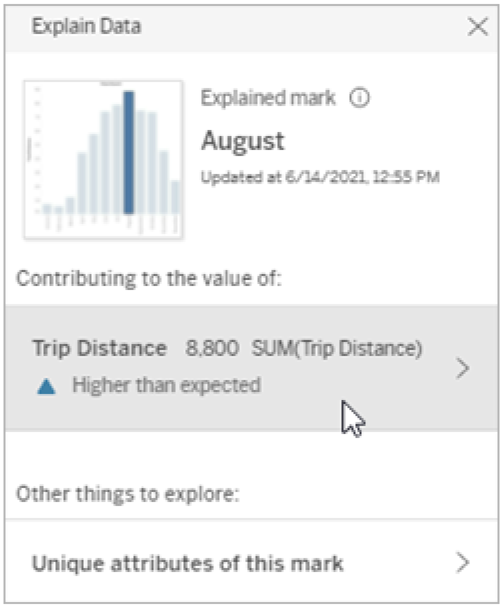
A distribution is a list of all the possible values (or intervals) of the data. It also indicates how often each value occurs (frequency of occurrence).

Explanation Types in Explain Data

Each time you select a new mark in a viz or dashboard and run Explain Data, Tableau runs a new statistical analysis considering that mark and the underlying data in the workbook. Possible explanations are displayed in expandable sections for the Explain Data pane. For information about how Explain Data analyzes and evaluates explanations, see **How Explain Data Works** on page 2083.

Contributing to the value of the mark

The **Contributing to the value of** section of the Explain Data pane lists explanations for each measure that can be explained (referred to as *target measures*). Each explanation listed here describes a relationship with the values of the target measure that are tested on the explained mark. Use your real-world, practical understanding of the data to determine if the relationships found by Explain Data are meaningful and worth exploring.



In this example, Trip Distance is the target measure

Mark Attributes

These explanations describe how underlying records of the marks in the view may be contributing to the aggregated value of the measure being explained. Mark attributes can include Extreme Values, Number of Records, or the Average Value of the mark.

> **Note**: For definitions of common terms used in explanations, see **Terms and concepts in explanations** on page 2045.

Extreme Values

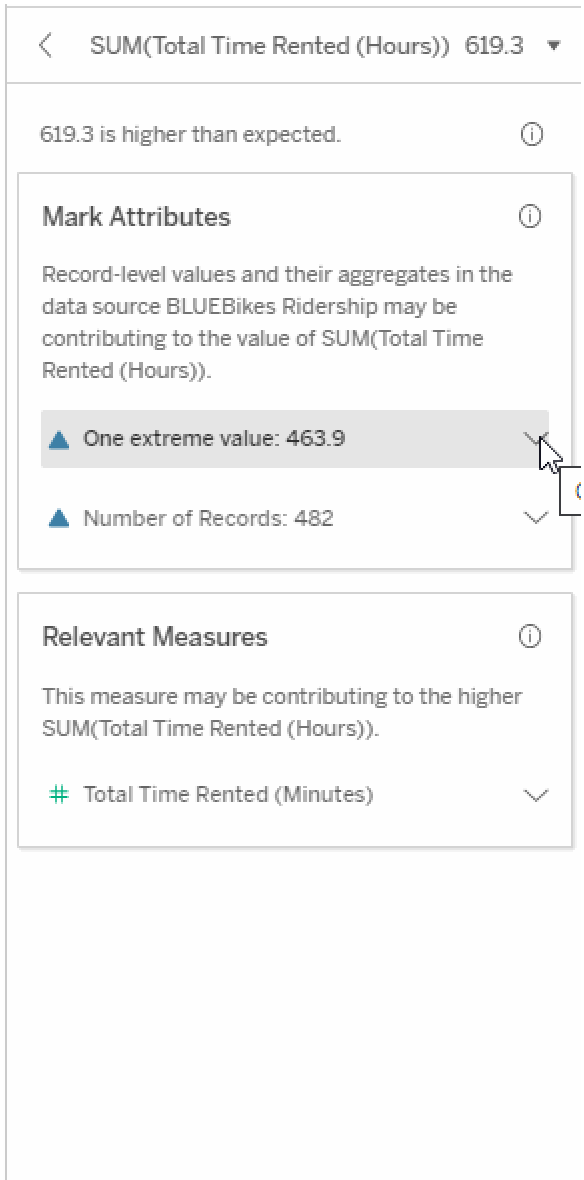
This explanation type indicates if one or more records have values that are significantly higher or lower than most records. If the explanation is supported by a model, it indicates the extreme value is affecting the target measure of the explained mark.

When a mark has extreme values, it doesn't automatically mean it has outliers or that you should exclude those records from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an extreme value in the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.

> **Note**: This explanation must be enabled by the author to be visible in viewing mode for a published workbook. For more information, see **Control Access to Explain Data** on page 2072.

This explanation shows:

- The number of underlying records in the explained mark.
- The extreme value or values contributing to the value of the target measure.
- The distribution of values in the mark.
- The record details that correspond to each distribution value.



Exploration options:

- Hover over a

In this example, a single extreme value of 463 hours rented is circle in the chart to see its corresponding value.

- Click the left or right arrow below the details list to scroll through record details.
- If available, click **View Full Data**, and then click the **Full Data** tab to see all records in a table.
- Click the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- If the number of records is low, examine these values

contributing to the higher than expected sum of Total Time Rented of 613 hours.

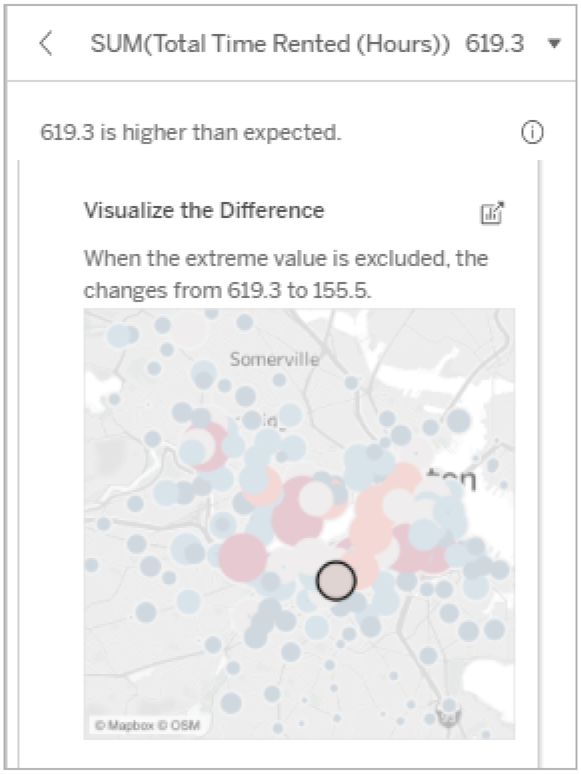
A likely reason for this high value could be that someone forgot to dock the bike when they returned it. In this case, the author might want to exclude this value for future analysis.

compared to the extreme value.

- If the extreme value is significantly higher or lower than the other record values, exclude it and consider how it changes the value of the explained mark.
- When considering the data with and without the extreme value, use this as an opportunity to apply your practical knowledge about the data.

This section shows:

- How the explained mark value changes when the extreme value is excluded.



Exploration options:

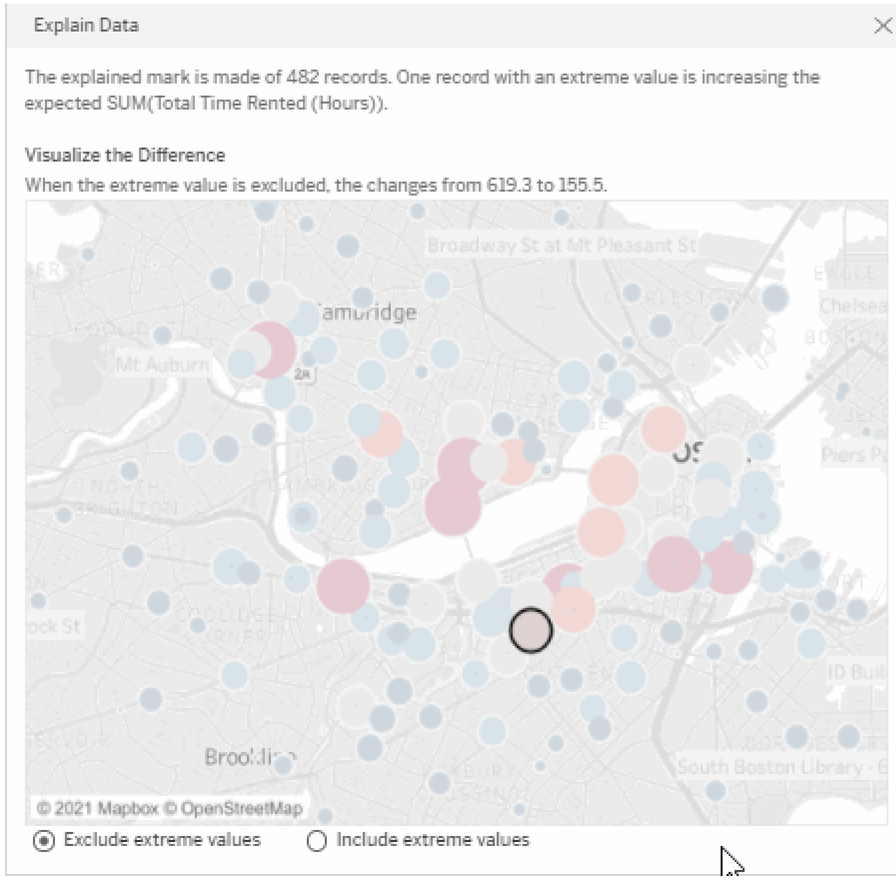
- Click the **Open** icon to see a larger version of the visualization.
- Explore the difference with and

without the extreme value (or values).

- Authors can open the view as a new sheet and apply a filter to exclude the extreme value.

Next steps for analysis:

- If the extreme value is significantly higher or lower than the other record values, exclude it and see how it changes the value of the explained mark.
- When considering the data with and without the extreme



In this example, when the extreme value of 483 is excluded, the explained mark is no longer high compared to other marks in the view. Other marks now stand out. The author might want to explore the other marks to consider why these other locations have higher hours for bike rentals.

value, use this as an opportunity to apply your practical knowledge about the data.

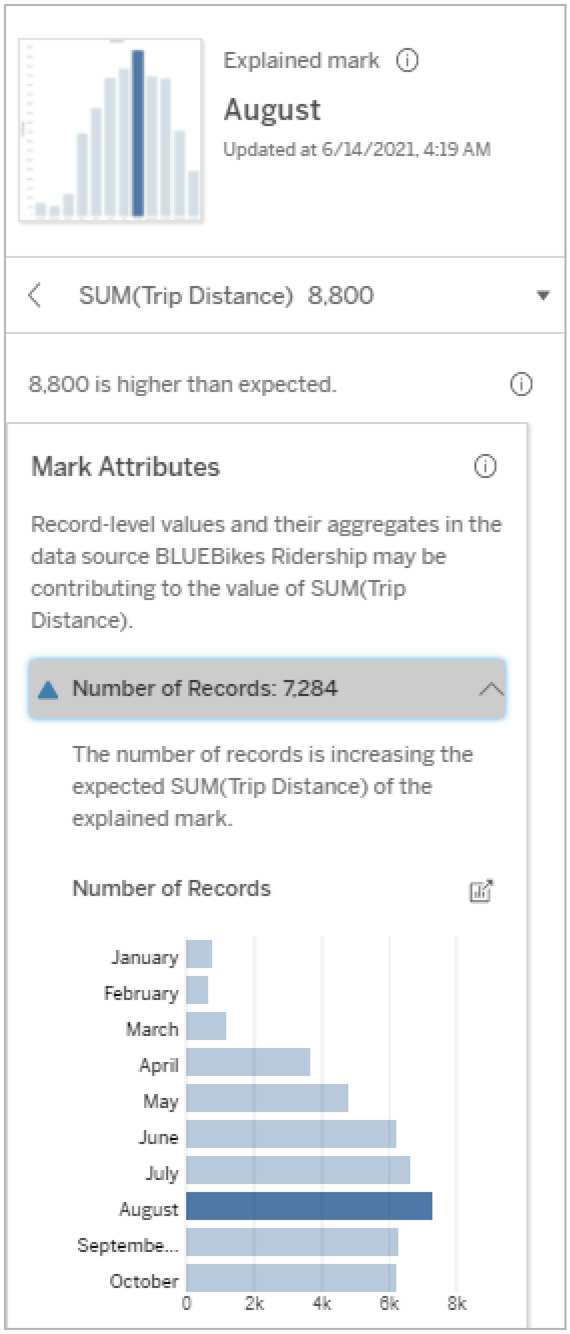
Number of Records

This explanation type describes when the count of the underlying records is correlated to the sum. The analysis found a relationship between the number of records that are being aggregated in a mark and the mark's actual value.

While this might seem obvious, this explanation type helps you explore whether the mark's value is being affected by the magnitude of the values in its records or simply because of the number of records in the explained mark.

This explanation shows:

- The number of records in the target measure for the explained mark (dark blue bar).
- The number of records in the target measure for other marks in the source visualization (light blue bar).



Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open** icon to see a larger version of the visualization.

In this example, the number of records for Trip Distance is listed for each value of Ride Month, which is a dimension in the original visualization. August has the highest total trip

Next steps for analysis: • Compare whether the individual values of records are low or high, or the number of records in the explained mark is low or high. • Authors, if you are surprised by a high number of records, you might need to normalize the data.	distance value. You might explore whether August has the highest value for trip distance because more rides occurred in August, or if it has the highest trip distance because some rides were longer.

Average Value of Mark

This explanation type describes when the average of a measure is correlated to the sum. Compare whether the average value is low or high, or the number of records is low or high.

This explanation shows:

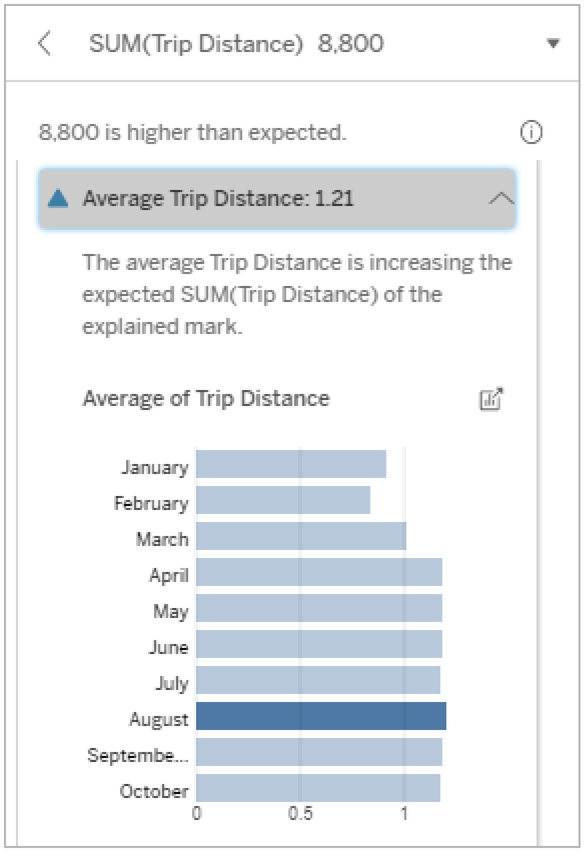
- The average of the target measure for each value of a dimension used in the source visualization.

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Compare whether the average value is low or high, or



In this example, the average trip distance for August is not significantly higher or lower than most months. This suggests that trip distance is higher for August because there were more rides in August, rather than from people taking longer rides.

the number of records is low or high. For example, are profits high because you sold a lot of items or because you sold expensive items?
- Try to figure out why the explained mark has a significantly higher or lower average value.

Relevant Single Value

Use this explanation to understand the composition of the record values that make up the explained mark.

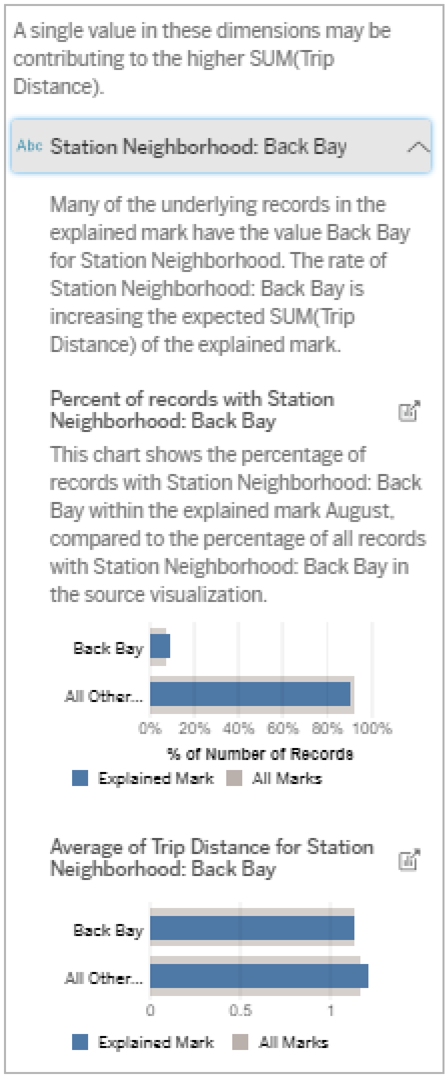
This explanation type identifies when a single value in an unvisualized dimension may be contributing to the aggregate value of the explained mark. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

This explanation indicates when every underlying record of a dimension has the same value, or when a dimension value stands out because either many or few of the records have the same single value for the explained mark.

Note: For definitions of common terms used in explanations, see **Terms and concepts in explanations** on page 2045.

This explanation shows:

- The percent of the number of records for a single value of a dimension for the explained mark (blue bar) versus all marks (gray bar) in the source visualization.
- The percent of the number of records for all other values of a dimension for the explained mark (blue bar) versus all marks (gray bar) in the source visualization.
- The average



In this example, the statistical analysis has exposed that many of the rides come from the station neighborhood of Back Bay. Note that Station Neighborhood is an unvisualized dimension that has some relationship to Trip Distance

of the target measure for the single value of a dimension in the explained mark (blue bar) versus all marks (gray bar). • The average of the target measure for all other values of a dimension for the explained mark (blue bar) versus all marks (gray bar) in the source visualization. **Exploration options:** • Hover over each bar to see its details. • Click the **Open** icon to see a lar-	in the underlying data for the source visualization.

ger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the explained mark.
- Authors might want to create a new visualization to explore any unvisualized dimension surfaced in this explanation.

Relevant Dimensions

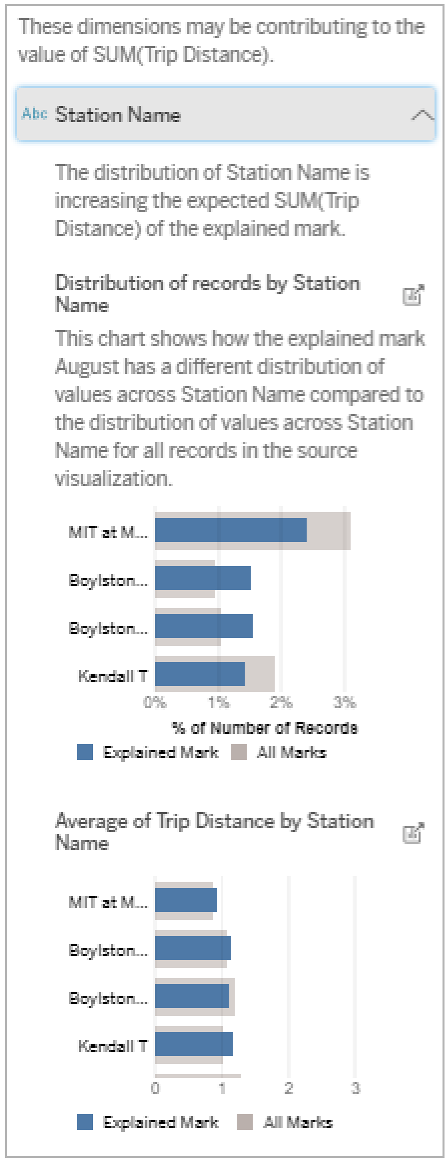
Use this explanation to understand the composition of the record values that make up the explained mark.

This explanation type shows that the distribution of an unvisualized dimension may be contributing to the aggregate value of the explained mark. This type of explanation is used for target measure sums, counts, and averages. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

> **Note**: For definitions of common terms used in explanations, see **Terms and concepts in explanations** on page 2045.

This explanation shows:

- The percent of the number of records for all values of a dimension for the explained mark (blue bar) versus all values of a dimension for all marks (gray bar) in the source visualization.
- The average of the target measure for all values of a dimension for the explained mark (blue bar) all values of a dimension for all marks



In this example, the statistical analysis has exposed that more (gray bar).

Exploration options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the explained mark.
- Authors might want to create a new visualization to

rides were taken from Boylston stations and fewer rides were taken from MIT and Kendall, compared to rides taken for marks overall.

Note that Station Name is an unvisualized dimension that has some relationship to Trip Distance in the underlying data for the source visualization.

explore any unvisualized dimensions surfaced in this explanation.

Relevant Measures

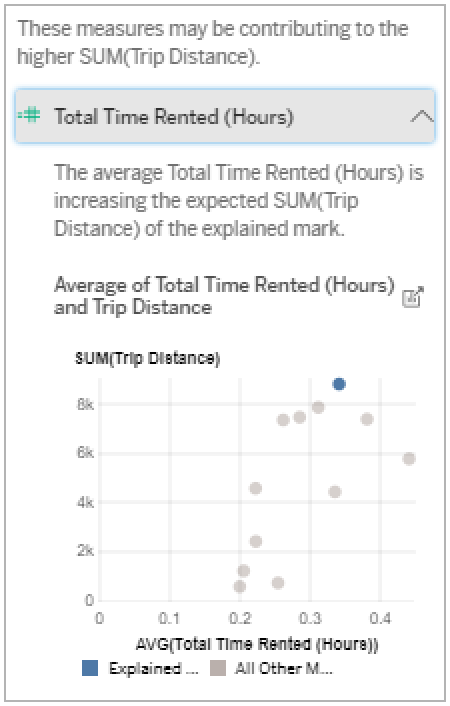
This explanation type shows that the average of an unvisualized measure may be contributing to the aggregate value of the explained mark. An unvisualized measure is a measure that exists in the data source, but isn't currently being used in the view.

This explanation can reveal a linear or quadratic relationship between the unvisualized measure and the target measure.

Note: For definitions of common terms used in explanations, see **Terms and concepts in explanations** on page 2045.

This explanation shows:

- The relationship between the sum of the target measure and the average of an unvisualized measure for the explained mark (blue circle) and all marks (gray circles) in the view.
- If the sum of the target measure is high or low because the average value of the unvisualized measure is high or low.



In this example, one possible reason why trip distance is high is because the average total time rented is also high.

Exploration options:

- Hover over each circle to see its details.

- Click the

  Open icon to see a larger version of the visualization.

Next steps for analysis:

- Authors might want to create a new visualization to explore any unvisualized measures surfaced in this explanation.

Other things to explore (Unique attributes of a mark)

This section of the Explain Data pane shows possible reasons why the explained mark is unique or unusual. These explanations:

- Do not explain why the value of this mark is what it is.
- Are not related in any way to the value of the measures in the source visualization.
- Do not take any target measures into account.

Relevant Single Value

The explanation type indicates when all records in the explained mark have the same single value in the unvisualized dimension, which is unusual compared to the distribution of values for all other marks in the view.

An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view

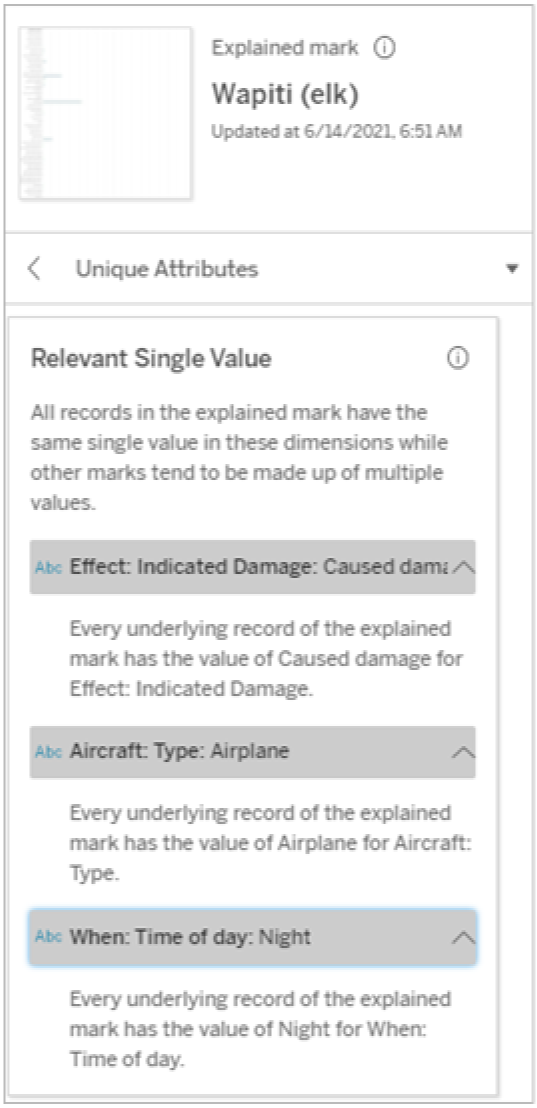
Note: For definitions of common terms used in explanations, see **Terms and concepts in explanations** on page 2045.

This explanation shows:

- When every underlying record has the same single value for a dimension.

Next steps for analysis:

- If all records in the explained mark have the same single value, you might want to check the number of records in that mark.
- You might check if the unvisualized dimension is a proxy for a dimension that is being used in the



This example shows an explanation for data about incidents related to birds and other wildlife colliding with aircraft. In the unfortunate case of a wapiti (elk), three underlying dimensions each had underlying records with single values:

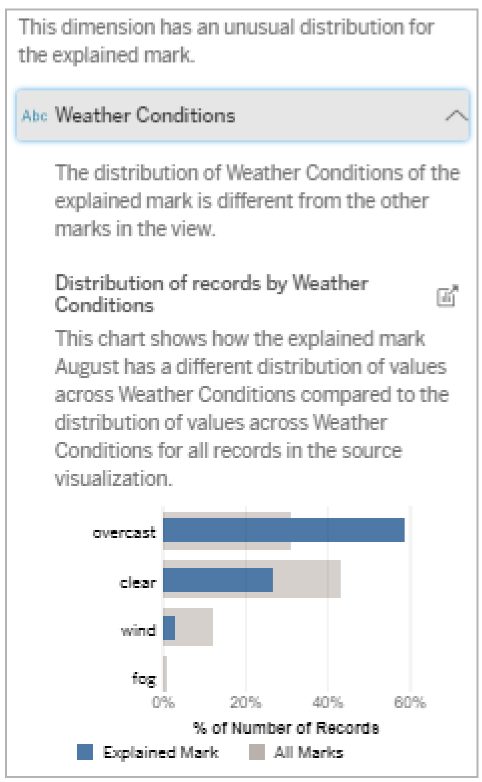
source visualization.	Aircraft, Indicated Damage, and Time of Day.

Relevant Dimensions

Use this explanation to understand the composition of the record values that make up the explained mark.

The distribution of an unvisualized dimension in the explained mark is unusual compared to the distribution of values for all other marks in the view. An unvisualized dimension is a dimension that exists in the data source, but isn't currently being used in the view.

> **Note**: For definitions of common terms used in explanations, see **Terms and concepts in explanations** on page 2045.

This explanation shows: • The percent of the number of records for all values of a dimension for the explained mark (blue bar) versus all values of a dimension for all marks (gray bar) in the source visualization.	
Exploration	In this example, a high percentage of

options:

- Hover over each bar to see its details.
- Scroll to see more of the chart.
- Click the **Open** icon to see a larger version of the visualization.

Next steps for analysis:

- Use this explanation to understand the composition of the record values that make up the explained mark.
- Authors might want to create a new visualization to explore any unvisualized dimensions

records are associated with overcast weather. Because the data is about bike rentals in Boston, and the explained mark is Trip Distance for August, we can assume that the weather is typically warm and humid. People might have rented bikes more often on overcast days to avoid the heat. It's also possible there were more overcast days in August.

surfaced in this explanation.

Requirements and Considerations for Using Explain Data

Explain Data is always available to authors in Tableau Desktop. In Tableau Online and Tableau Server, Creators and Explorers with editing permissions can run Explain Data when editing a workbook. Authors can allow any user to run Explain Data in viewing mode in published workbooks. For more information, see **Control Access to Explain Data** on page 2072.

What makes a viz a good candidate for Explain Data

Explain Data works best on visualizations that require deeper exploration and analysis, rather than infographic-style, descriptive vizzes that communicate summarized data.

- Row-level data is necessary for Explain Data to create models of your data and generate explanations. Vizzes with underlying, row-level data, where relationships might exist in unvisualized fields are good candidates for running Explain Data.
- Vizzes based on pre-aggregated data without access to row-level data are not ideal for the statistical analysis performed by Explain Data.

What data works best for Explain Data

When you are using Explain Data in a worksheet, remember that Explain Data works with:

- **Single marks only**—Explain Data must be run on a single mark. Multiple mark analysis is not supported.
- **Aggregated data**—The view must contain one or more measures that are aggregated using SUM, AVG, COUNT, or COUNTD. At least one dimension must also be present in the view.
- **Single data sources only**—The data must be drawn from a single, primary data source. Explain Data does not work with blended or cube data sources.

When preparing a data source for a workbook, keep the following considerations in mind if you plan to use Explain Data during analysis.

- Use a data source with underlying data that is sufficiently wide. An ideal data set has at least 10-20 columns in addition to one (or more) aggregated measures to be explained.
- Give columns (fields) names that are easy to understand.
- Eliminate redundant columns and data prep artifacts. For more information, see **Change fields used for statistical analysis** on page 2091.
- Don't discard unvisualized columns in the data source. Explain Data considers fields in the underlying data when is analyzes a mark.
- Low cardinality dimensions work better. The explanation of a categorical dimension is easier to interpret if its cardinality is not too high (< 20 categories). Dimensions with more than 500 unique values will not be considered for analysis.
- Don't pre-aggregate data as a general rule. But if the data source is massive, consider pre-aggregating the data to an appropriate level of detail.
- Use extracts over live data sources. Extracts run faster than live data sources. With live data sources, the process of creating explanations can create many queries (roughly one query per each candidate explanation), which can result in explanations taking longer to be generated.

Situations where Explain Data is not available

Sometimes Explain Data will not be available for a selected mark, depending on the characteristics of the data source or the view. If Explain Data cannot analyze the selected mark, the Explain Data icon and context menu command will not be available.

Explain Data can't be run in views that use:	• Map coordinate filters • Blended data sources • Data sources with parameters • Data sources that don't support COUNTD or COUNT(DISTINCT ...) syntax, such as Access. • Filters on aggregate measures • Disaggregated measures
Explain Data can't be run if you select:	• Multiple marks • Axis • Legend • Grand total • Trend line or reference line • A mark in a view that contains a very low number of marks

Explain Data can't be run when the measure to be used for an explanation:	• Isn't aggregated using SUM, AVG, COUNT, COUNTD • Is a table calculation • Is used in measure values
Explain Data can't offer explanations for a dimension when it is:	• A calculated field • A parameter • Used in Measure Names and Measure Values • A field with more than 500 unique values. Dimensions with more than 500 unique values will not be considered for analysis.

Control Access to Explain Data

Explain Data is always available to Creators and Explorers in authoring mode. Authors can control whether Explain Data is available in viewing mode in published workbooks and which explanation types are displayed in the Explain Data pane.

Be aware that Explain Data can surface values from dimensions and measures in the data source that aren't represented in the view. As an author, you should run Explain Data and test the resulting explanations to make sure that sensitive data isn't being exposed in your published workbooks.

Because Extreme Values explanations display record-level details, workbook authors must explicitly select an option to display Extreme Values explanations in published workbooks. For more information, see **Show or hide Extreme Values explanations in viewing mode** on page 2076.

Who can access Explain Data

Your access to Explain Data will vary depending on how you interact with Tableau workbooks, and whether you are viewing or editing a published workbook.

Mode	Who Can Access
Viewing mode	When an author makes Explain Data available in viewing mode in a published workbook, Tableau **Viewers**, **Explorers**, and **Creators** can run and explore Explain Data explanations in viewing mode.
Editing mode	Tableau **Creators** can run Explain Data when editing a view in

Mode	Who Can Access
	Tableau Desktop, Tableau Online, or Tableau Server. **Explorers** who have editing permissions can run Explain Data when editing a workbook in Tableau Online or Tableau Server. Creators and Explorers who have editing permissions can open new worksheets for further analysis. They also can use Explain Data Settings to control who can use Explain Data and what they can see.

Control who can use Explain Data and what they can see

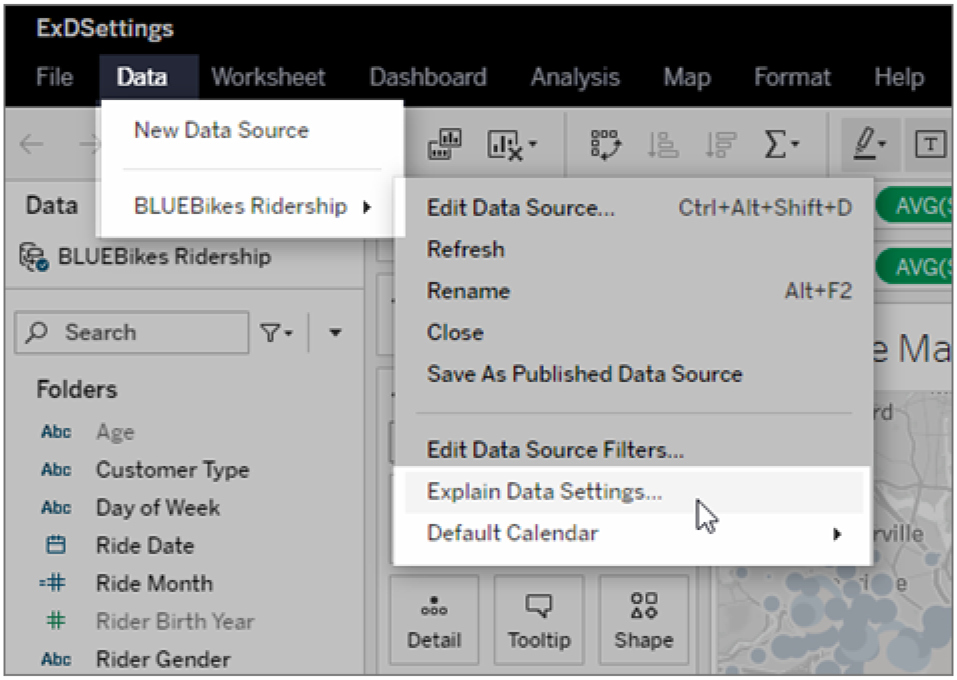
Creators and Explorers who have editing permissions can access **Explain Data Settings**, which provide options for controlling:

- Whether all users, including Viewers, are allowed to run Explain Data and see explanations while viewing published workbooks and dashboards. This option is not enabled by default, so authors need to explicitly allow it.
- Whether Extreme Values explanations and record-level details are displayed in viewing mode in published workbooks. This option is not enabled by default, so authors need to explicitly allow it.
- The explanation types that are displayed in the Explain Data pane.
- The fields that are included in, or excluded from statistical analysis.

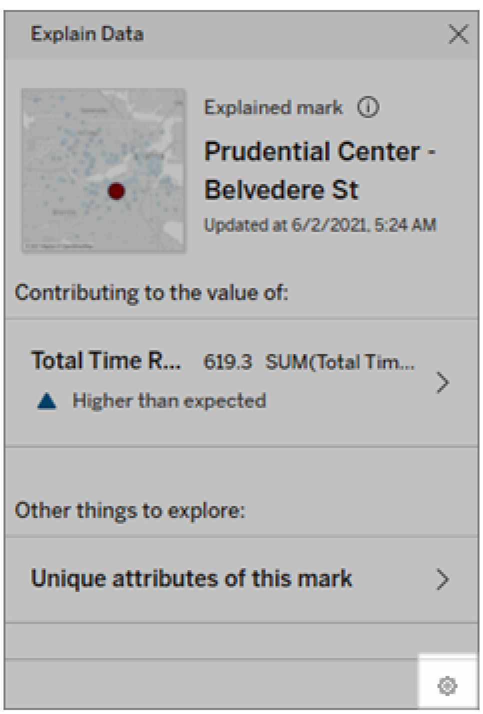
These options are set for the entire workbook and can only be set in the Explain Data Settings dialog box.

Open the Explain Data Settings dialog box

1. From the **Data** menu, select *data-source-name* > **Explain Data Settings**. For example:



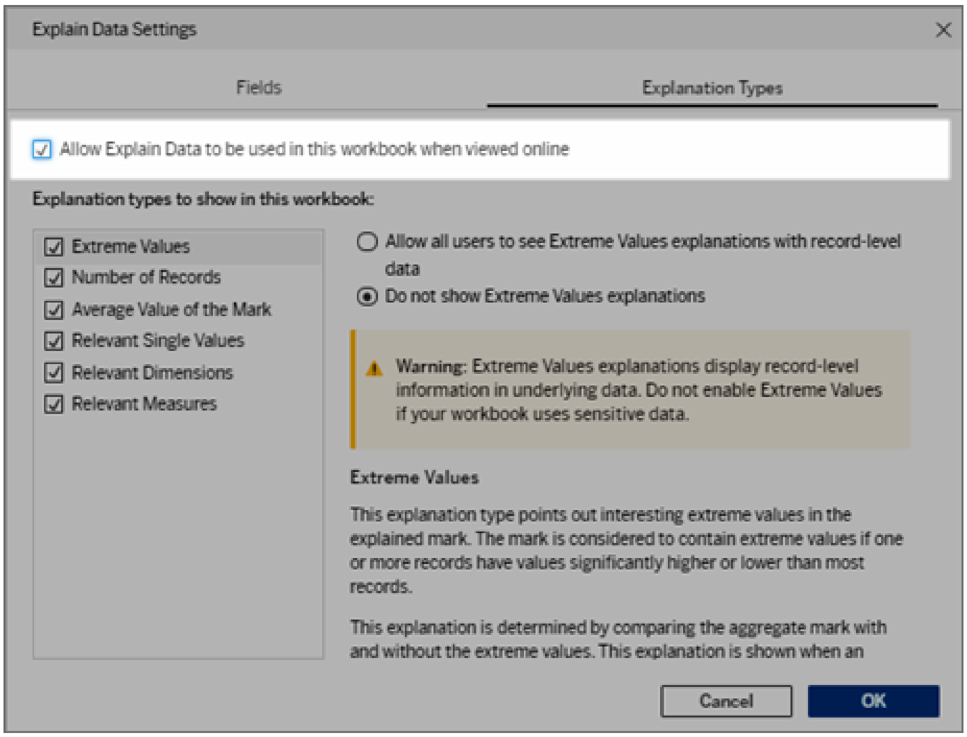
Or, in the Explain Data pane, click the settings icon (bottom right).



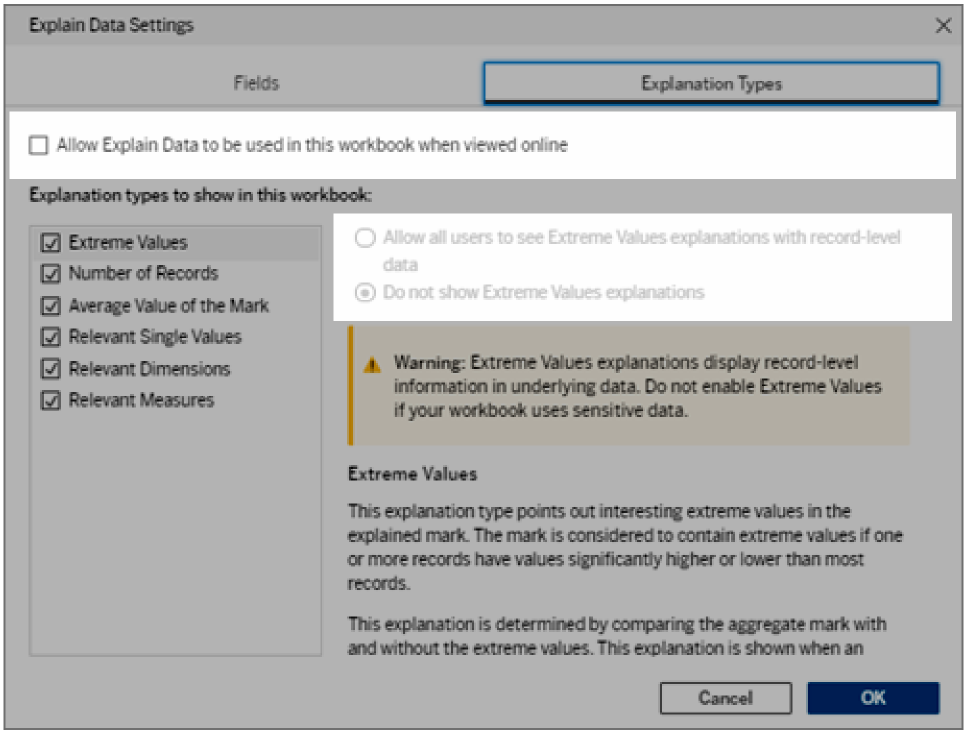
Enable Explain Data in viewing mode in published workbooks and dashboards

By default, Explain Data is not available in viewing mode. To turn Explain Data on for viewing mode, authors must do the following:

1. In the Explain Data Settings dialog box, click the **Explanation Types** tab.
2. Select **Allow Explain Data to be used in this workbook when viewed online**.



Alternatively, leave this option unchecked to return to the default setting and make Explain Data unavailable in viewing mode. Creators and Explorers who have editing permissions will still be able to run Explain Data when editing a workbook.



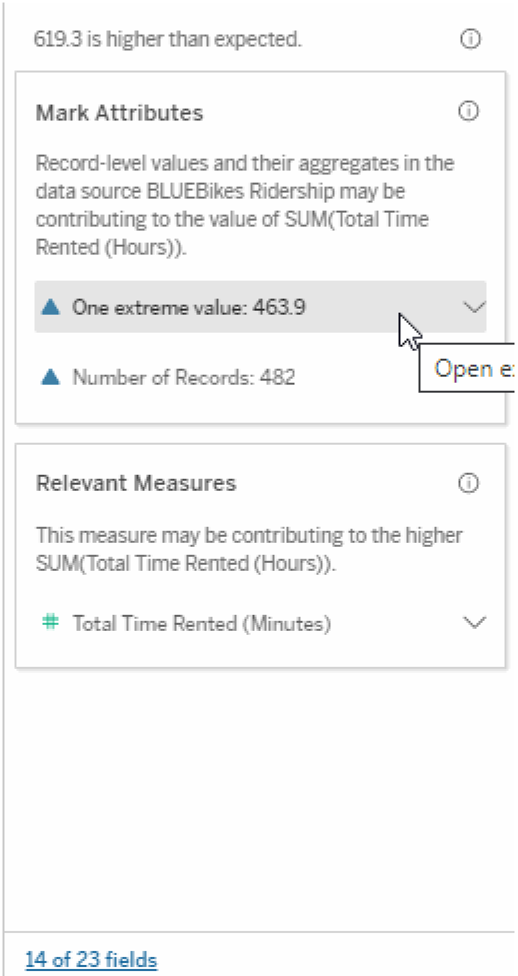
This image shows the default settings for the Explanation Types tab

3. Click **OK**.

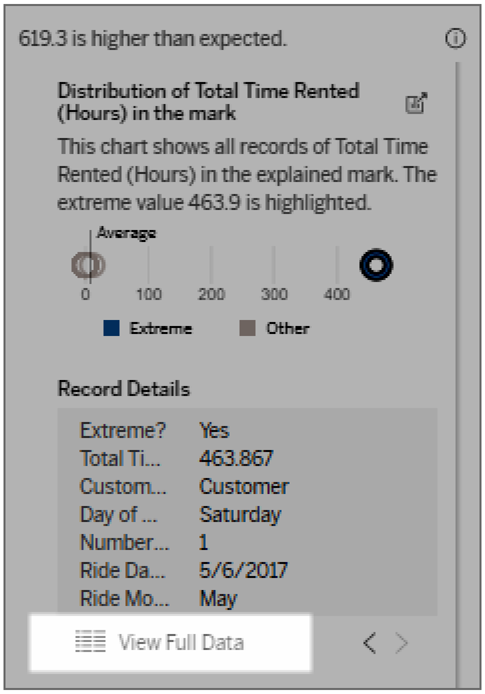
Test the setting by saving and closing the published workbook, and then open a view from the workbook in viewing mode. Select a mark, and then run Explain Data to check the explanation results.

Show or hide Extreme Values explanations in viewing mode

The Extreme Values explanation type includes record-level details. When editing a workbook, Creators and Explorers can always see Extreme Values explanations and record-level details. However, in viewing mode for published workbooks, Extreme Values explanations are hidden by default.



Note that only Creators and Explorers with Download Full Data permissions can see the View Full Data option for Extreme Values explanations.

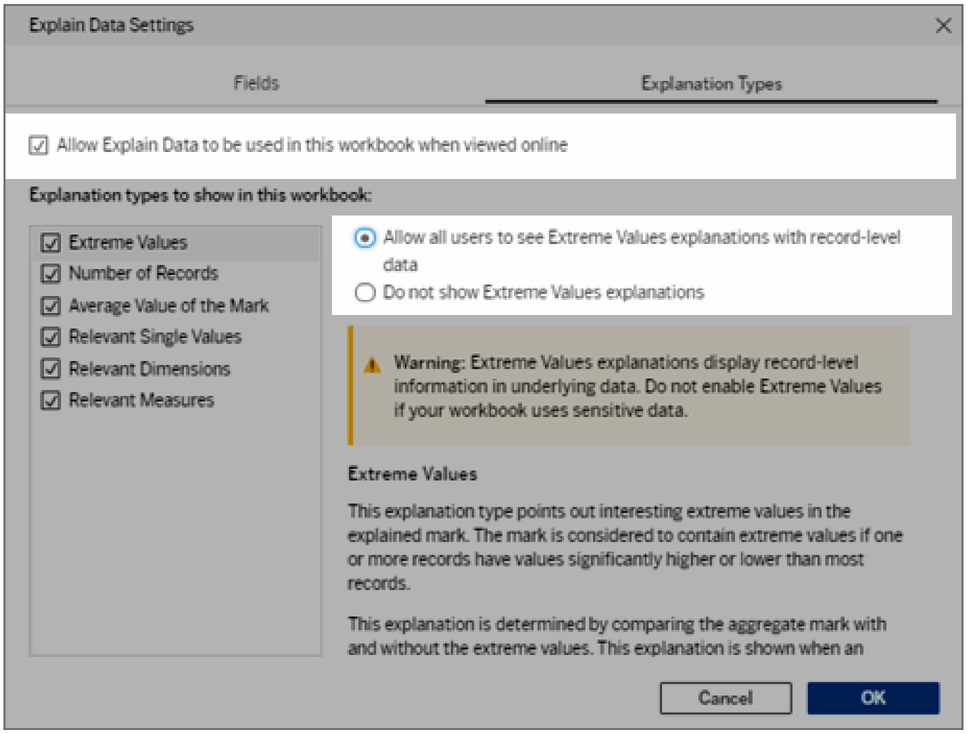


If you have sensitive data in your workbook, you can keep the Extreme Values explanation type hidden for viewing mode.

To show Extreme Values explanations in viewing mode in published workbooks

As an author, if you want Extreme Values explanations to be displayed in viewing mode, do the following:

1. In the Explain Data Settings dialog box, click the **Explanation Types** tab.
2. Select **Allow Explain Data to be used in this workbook when viewed online**.
3. Make sure that the Extreme Values explanation type is selected.
4. Select **Allow all users to see Extreme Values explanations with record-level data**.



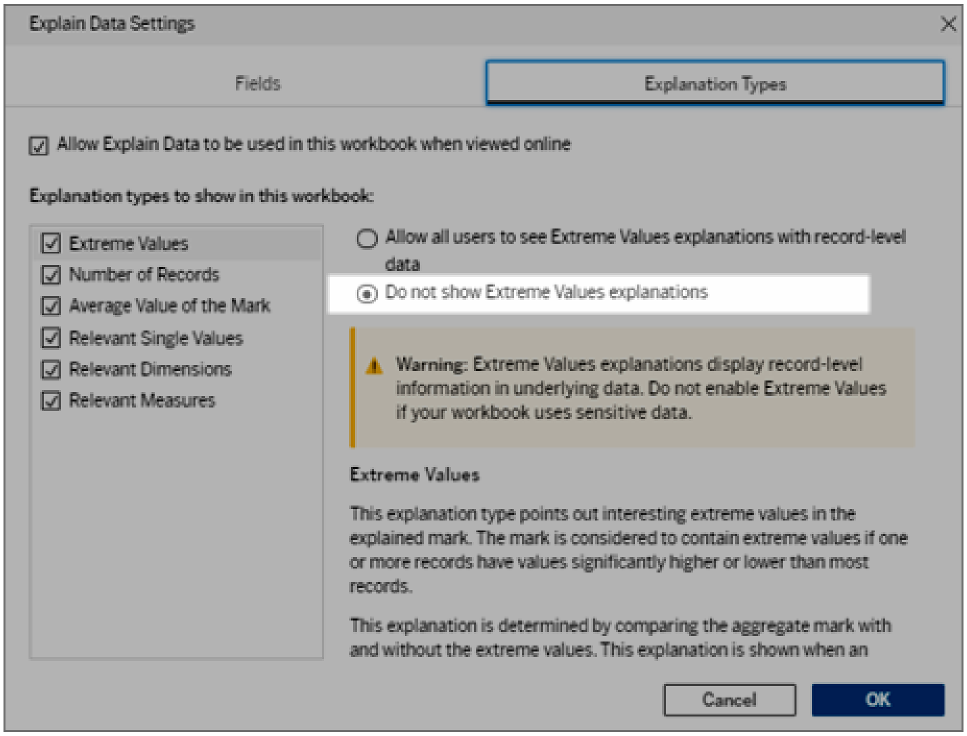
5. Click **OK**.

Test the setting by saving and closing the published workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

To hide Extreme Values explanations and record-level details in viewing mode (published workbooks)

By default, Extreme Values explanations and record-level details are not available in viewing mode. Use these steps to return to the default state.

1. In the Explain Data Settings dialog box, click the **Explanation Types** tab.
2. Make sure that the Extreme Values explanation type is selected (allows Creators and Explorers to see these explanations in editing mode).
3. Select the option **Do not show Extreme Values explanations** to hide Extreme Values explanations and record-level details from all users in viewing mode.



4. Click **OK**.

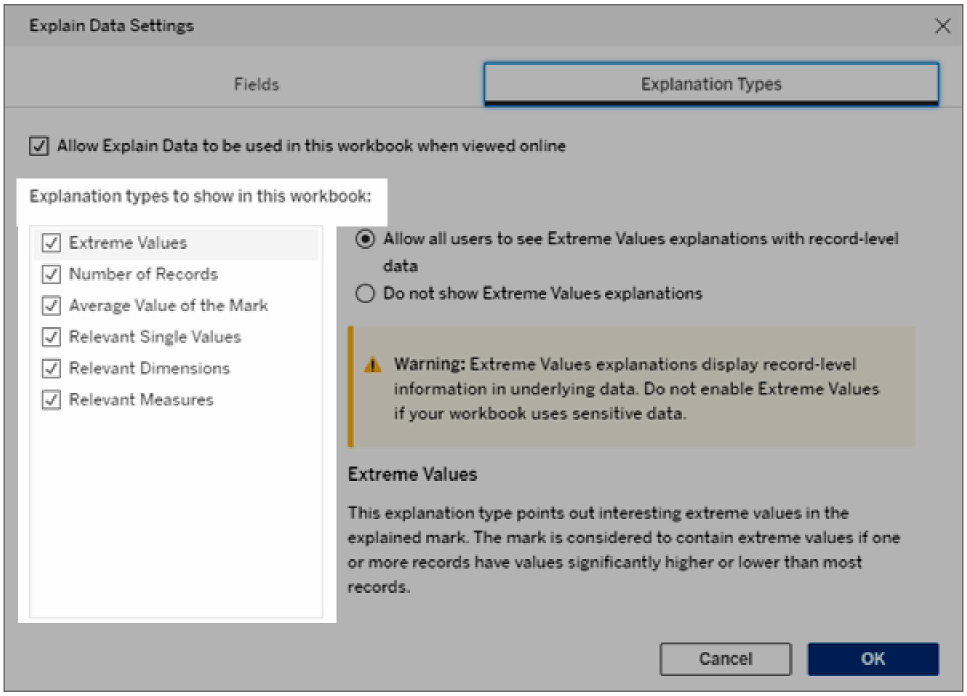
Test the setting by saving and closing the workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

Include or exclude explanation types displayed by Explain Data

Creators and Explorers who have editing permissions can choose to exclude (or include) explanation types displayed for all workbook users.

1. In the Explain Data Settings dialog box, click the **Explanation Types** tab.

2. In the list of explanation types, select or clear an explanation type.



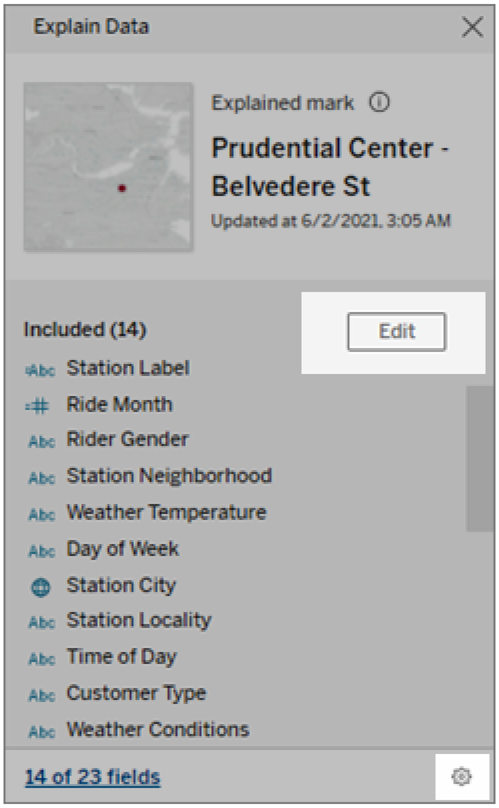
3. Click **OK**.

Test the setting by saving and closing the published workbook, and then opening a view from the workbook in viewing mode. Select a mark that typically has Extreme Value explanations, and then run Explain Data to check the explanation results.

Include or exclude fields used for statistical analysis

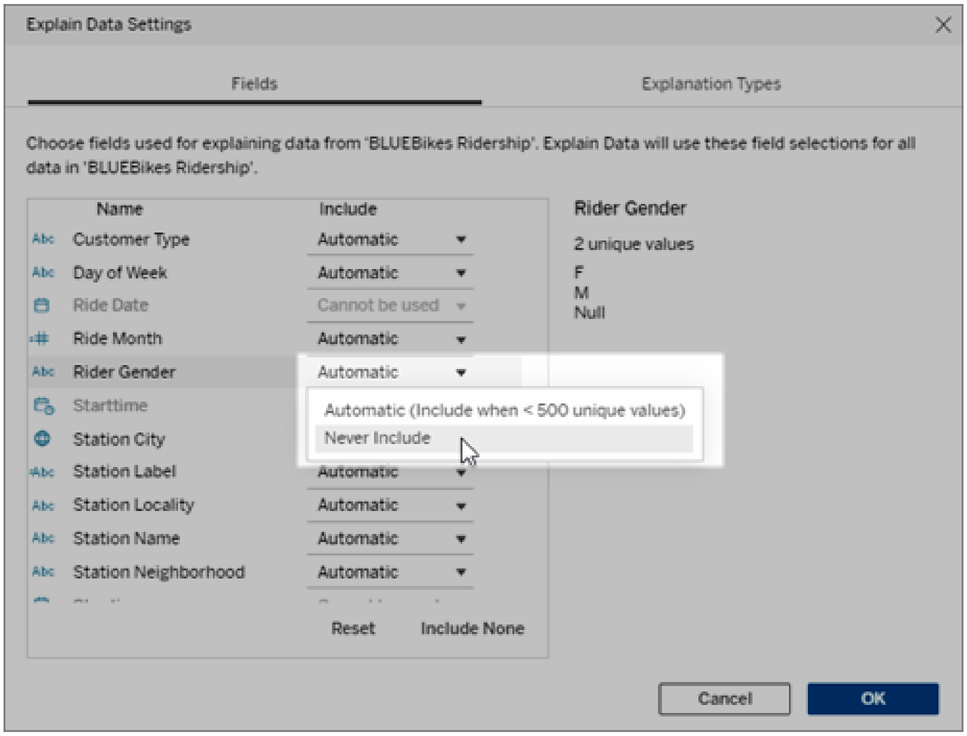
Creators or Explorers who have editing permissions can choose to exclude (or include) fields that are eligible for analysis.

1. In the Explain Data pane (bottom right), click the settings icon. Or, click the **Edit** button in the Analyzed Fields view.



2. In the Explain Data Settings dialog box, click the **Fields** tab.
3. In the list of fields under **Include**, click the drop-down arrow and select **Automatic** to include an eligible field every time Explain Data runs for that workbook.

   Note that fields must have less than 500 unique values to be included in the analysis.



Select **Never Include** to explicitly exclude the field.
Select **Include None** to run a statistical analysis on the data without considering fields.
Select **Reset** to return to the default settings.

4. Click **OK**.

Test the setting by saving the published workbook. Select a mark, and then run Explain Data to check the explanation results.

How Explain Data Works

Use Explain Data as an incremental, jumping-off point for further exploration of your data. The possible explanations that it generates help you to see the different values that make up or relate to a selected mark in a view. It can tell you about the characteristics of the data points in the data source, and how the data might be related (correlations) using statistical modeling. These explanations give you another tool for inspecting your data and finding interesting clues about what to explore next.

Note: Explain Data is a tool that uncovers and describes relationships in your data. It can't tell you what is causing the relationships or how to interpret the data. **You are the expert on your data.** Your domain knowledge and intuition is key in helping you decide what characteristics might be interesting to explore further using different views.

For related information on how Explain Data works, and how to use Explain Data to augment your analysis, see these Tableau Conference presentations:

- From Analyst to Statistician: Explain Data in Practice (1 hour)
- Leveraging Explain Data (45 minutes)
- Explain Data Internals: Automated Bayesian Modeling (35 minutes)
- Machine Learning, Explainable AI, and Tableau (45 minutes), Session Materials
- Making Business More Bayesian (45 minutes)

What Explain Data is (and isn’t)

Explain Data is:

- A tool and a workflow that leverages your domain expertise.
- A tool that surfaces relationships in your data and recommends where to look next.
- A tool and a workflow that helps expedite data analysis and make data analysis more accessible to a broader range of users.

Explain Data is not:

- A statistical testing tool.
- A tool to prove or disprove hypotheses.
- A tool that is giving you an answer or telling you anything about causality in your data.

When running Explain Data on marks, keep the following points in mind:

- **Consider the shape, size, and cardinality of your data**. While Explain Data can be used with smaller data sets, it requires data that is sufficiently wide and contains enough marks (granularity) to be able to create a model.

- **Don't assume causality**. Correlation is not causation. Explanations are based on models of the data, but are not causal explanations.

  A correlation means that a relationship exists between some data variables, say A and B. You can't tell just from seeing that relationship in the data that A is causing B, or B is causing A, or if something more complicated is actually going on. The data patterns are exactly the same in each of those cases and an algorithm can't tell the difference between each case. Just because two variables seem to change together doesn't necessarily mean that one causes the other to change. A third factor could be causing them both to change, or it may be a coincidence and there might not be any causal relationship at all.

However, you might have outside knowledge that is not in the data that helps you to identify what's going on. A common type of outside knowledge would be a situation where the data was gathered in an experiment. If you know that B was chosen by flipping a coin, any consistent pattern of difference in A (that isn't just random noise) must be caused by B. For a longer, more in-depth description of these concepts, see the article Causal inference in economics and marketing by Hal Varian.

How explanations are analyzed and evaluated

When you run Explain Data on a mark, a statistical analysis is run on the aggregated mark and then on possibly related data points from the data source that aren't represented in the current view.

Explain Data first predicts the value of a mark using only the data that is present in the visualization. Next, data that is in the data source (but not in the current view) is considered and added to the model. The model determines the range of predicted mark values, which is within one standard deviation of the predicted value.

What is an expected range?

The expected value for a mark is the median value in the expected range of values in the underlying data in your viz. The expected range is the range of values between the 15th and 85th percentile that the statistical model predicts for the explained mark. Tableau determines the expected range each time it runs a statistical analysis on a selected mark.

Possible explanations are evaluated on their explanatory power using statistical modeling. For each explanation, Tableau compares the expected value with the actual value.

value	Description
Higher than expected / Lower than expected	If an expected value summary says the mark is *lower than expected* or *higher than expected*, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is *slightly lower* or *slightly higher* than expected, or *within the range of natural variation*, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher than the median.

value	Description
Expected Value	If a mark has an expected value, it means its value falls within the expected range determined by Tableau when it runs a statistical analysis on the mark you selected.
Random Variation	When the explained mark has a low number of records, there may not be enough data available for Explain Data to form a statistically significant explanation. If the mark's value is outside the expected range, Explain Data can't determine whether this unexpected value is being caused by random variation or by a meaningful difference in the underlying records.
No Explanation	When the explained mark value is outside of the expected range and it does not fit a statistical model used for Explain Data, no explanations are generated.

Models used for analysis

Explain Data builds models of the data in a view to predict the value of a mark and then determines whether a mark is higher or lower than expected given the model. Next, it considers additional information, like adding additional columns from the data source to the view, or flagging record-level outliers, as potential explanations. For each potential explanation, Explain Data fits a new model, and evaluates how unexpected the mark is given the new information. Explanations are scored by trading off complexity (how much information is added from the data source) against the amount of variability that needs to be explained. Better explanations are simpler than the variation they explain.

Explanation type	Evaluation
Extreme values	Extreme values are aggregated marks that are outliers, based on a model of the visualized marks. The selected mark is considered to contain an extreme value if a record value is in the tails of the distribution of the expected values for the data. An extreme value is determined by comparing the aggregate mark with and without the extreme value. If the mark becomes less surprising by removing a value, then it receives a higher score.

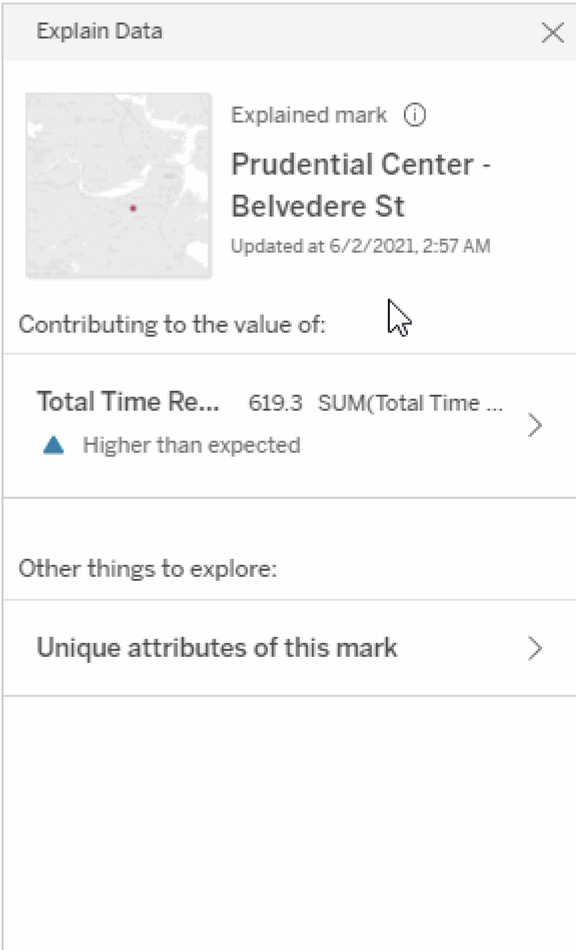
Explanation type	Evaluation
	When a mark has extreme values, it doesn't automatically mean it has outliers, or that you should exclude those records from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an interesting extreme value in the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.
Number of records	The number of records explanation models the aggregate sum in terms of the aggregate count; average value of records models it in terms of the aggregate average. The better the model explains the sum, the higher the score. This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.
Average value of the mark	This type of explanation is used for aggregate marks that are sums. It explains whether the mark is consistent with the other marks because in terms of its aggregate count or average, noting the relation SUM(X) = COUNT(X) * AVG(X). This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.
Relevant Dimensions	This explanation models the target measure of the explained mark in terms of the breakdown among categories of the unvisualized dimension. The analysis balances the complexity of the model with how well the mark is explained. An *unvisualized dimension* is a dimension that exists in the data source, but isn't currently being used in the view. This type of explanation is used for sums, counts and averages. The model for unvisualized dimensions is created by splitting out marks according to the categorical values of the explaining column, and then building a model with the value that includes all of the data points in the source visualization. For each row, the model attempts to recover each of the individual components that made each mark.

Explanation type	Evaluation
	The analysis indicates whether the model predicts the mark better when components corresponding to the unvisualized dimension are modeled and then added up, versus using a model where the values of the unvisualized dimension are not known. Aggregate dimension explanations explore how well mark values can be explained without any conditioning. Then, the model conditions on values for each column that is a potential explanation. Conditioning on the distribution of an explanatory column should result in a better prediction.
Relevant Measures	This explanation models the mark in terms of this unvisualized measure, aggregated to its mean across the visualized dimensions. An *unvisualized measure* is a measure that exists in the data source, but isn't currently being used in the view. The Relevant Measures explanation can reveal a linear or quadratic relationship between the unvisualized measure and the target measure.

Analyzed Fields in Explain Data

When you run Explain Data on a mark, a statistical analysis is run on the aggregated mark, and then on potentially related fields from the data source that aren't represented in the current view.

Explain Data might not include every column from the data source in the analysis. In many cases, certain types of fields will be automatically excluded from the analysis. For more information, see **Fields excluded by default** on page 2094.

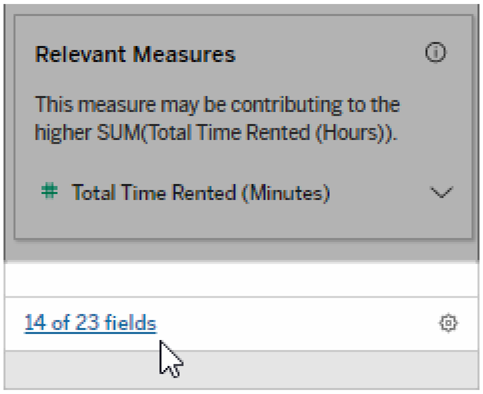


> **Note**: Dimensions with more than 500 unique values won't be considered for analysis (unless allowed by the author in Explain Data Settings).

All users can view information on which fields are included or excluded in the current analysis. Creators and Explorers who have editing permissions can edit the fields used by Explain Data for statistical analysis.

View fields analyzed by Explain Data

When you expand an explanation for a measure that is contributing to the value of the mark, a link that indicates the number of fields considered in the analysis is displayed at the bottom of the Explain Data pane.

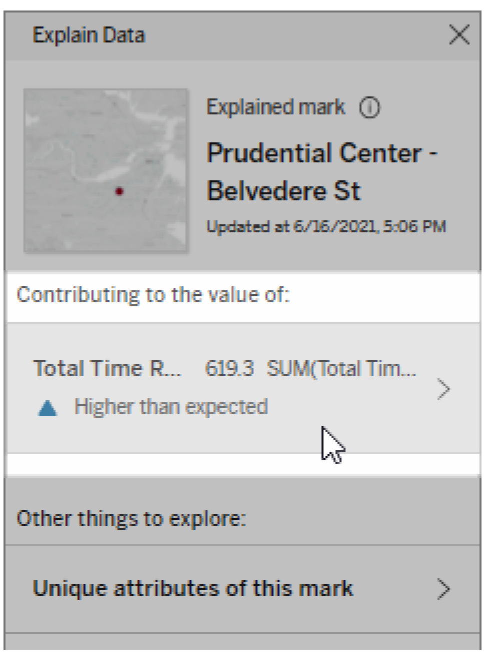


Click the link to see the list of fields included in or excluded from the current statistical analysis.

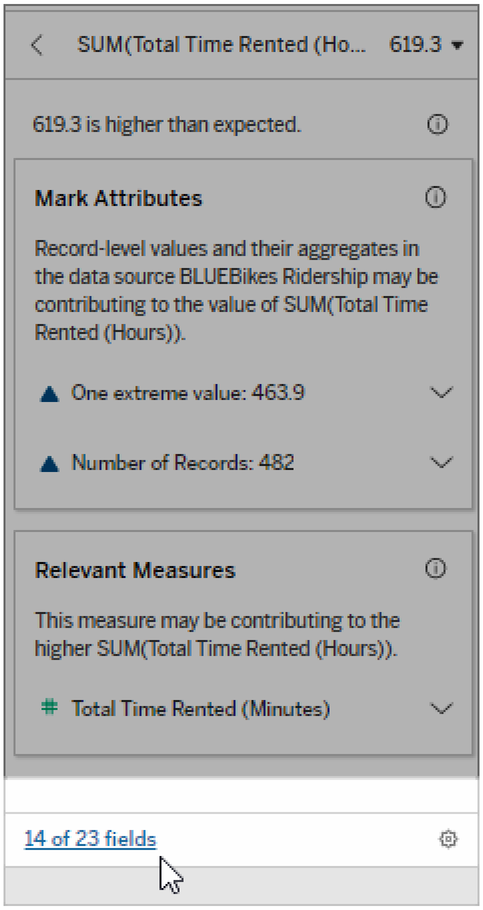
When a data source contains more than 1000 unvisualized dimensions or measures, you might see an alert asking if you want Explain Data to consider more fields. Click **Explain All** to run an analysis that includes more fields. The analysis may take longer to complete.

To view fields used by Explain Data for statistical analysis

1. Run Explain Data on a mark.
2. In the Explain Data pane, under **Contributing to the value of**, click a measure name.

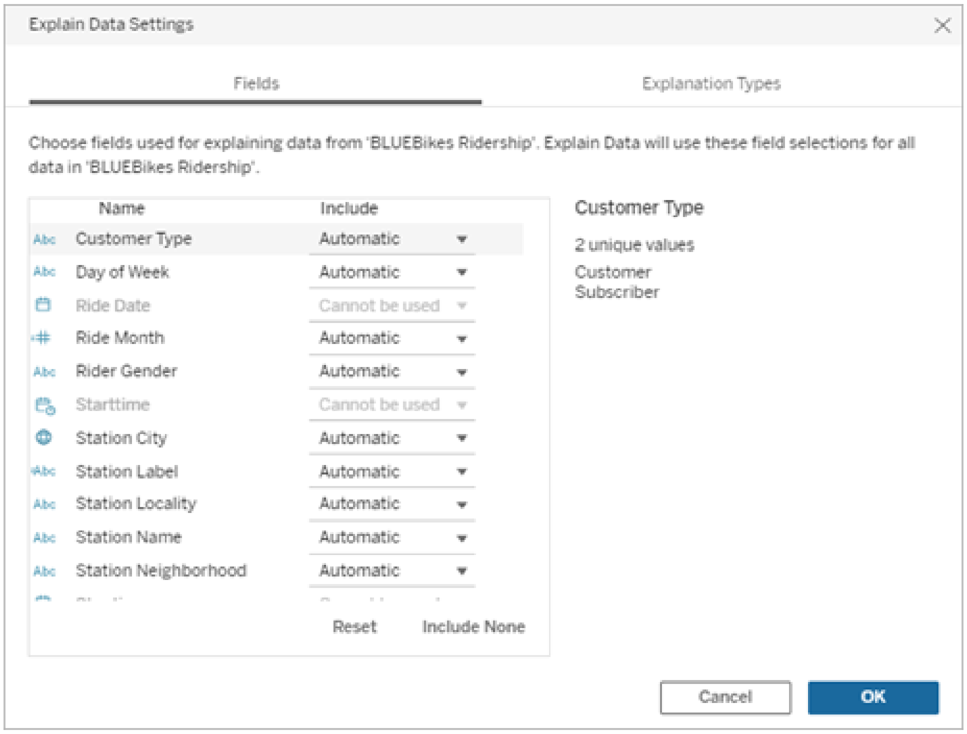


3. Click the *number-of-fields* link at the bottom of the pane.



Change fields used for statistical analysis

Creators and Explorers who have editing permissions can select fields to be included or excluded from the statistical analysis in the Fields tab of the Explain Data Settings dialog box.

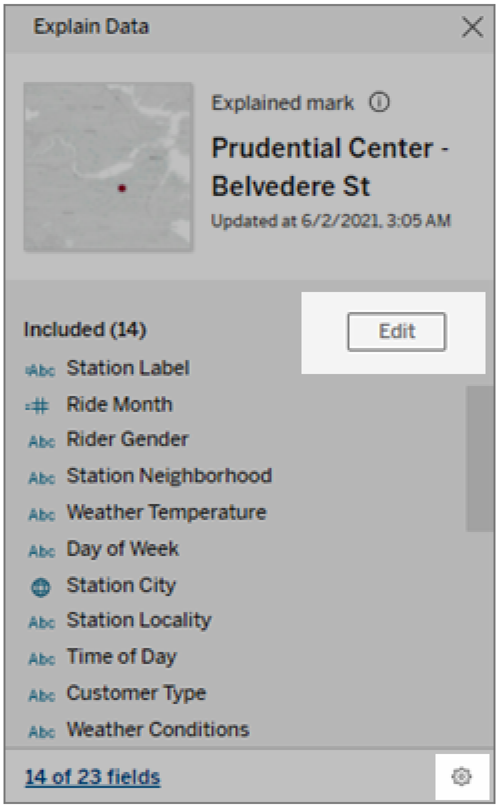


When a data source contains dimensions with a large number of unique values (up to 500), those fields won't be considered for analysis.

To edit the fields used by Explain Data for statistical analysis

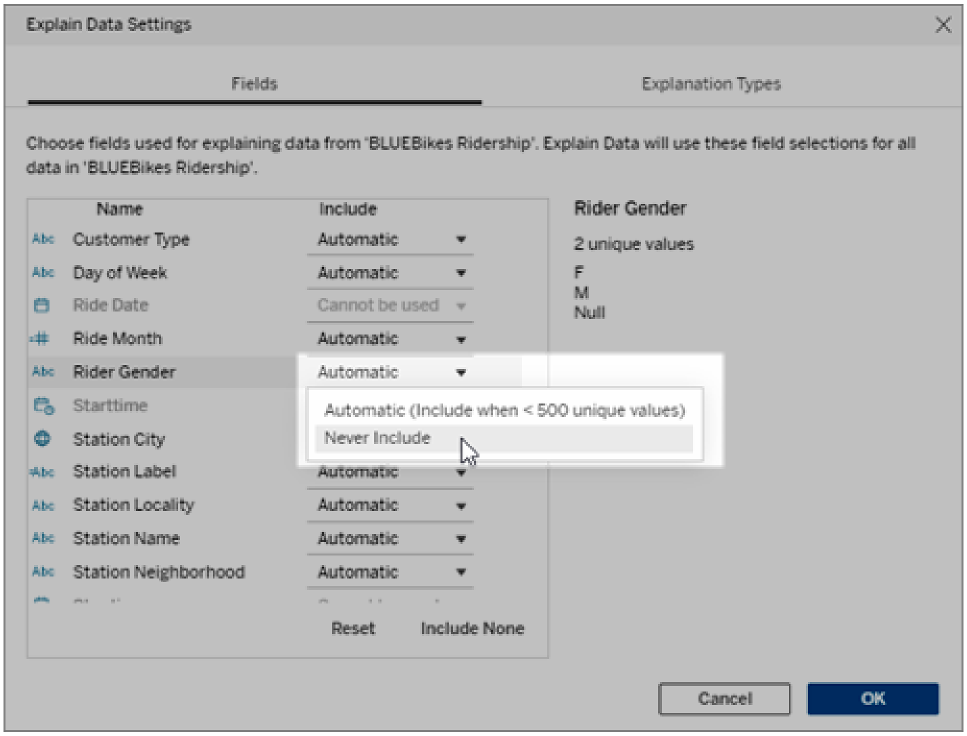
Settings for analyzed fields are applied at the data source level.

1. Run Explain Data on a mark when editing a view.
2. In the Explain Data pane, click the settings icon at the bottom of the Explain Data pane. Or, click the **Edit** button in the Analyzed Fields view (how to open analyzed fields).



3. In the Explain Data Settings dialog box, click the **Fields** tab.
4. Click a drop-down arrow next to a field name, select **Automatic** or **Never Include**, and then click **OK**.

   Note that fields must have less than 500 unique values to be included in the analysis.



Fields excluded by default

Fields excluded by default	Reasons for exclusion
All unvisualized measures when there are more than 1,000 measures in the data source. All unvisualized dimensions when there are more than 1,000 dimensions in the data source.	Computing explanations for more than 1000 unvisualized measures or dimensions can take longer to compute, sometimes several minutes. These fields are excluded by default for initial analysis, but you can choose to include them for further analysis. In this situation, you might see an alert asking if you want Explain Data to consider more fields. Click the alert link to get more information. Click **Explain All** to run an analysis that includes more fields.
Fields that use geometry, latitude, or longitude	Geometry, latitude, or longitude by themselves can never be explanations. It is highly likely that an explanation that calls out the latitude or the lon-

Fields excluded by default	Reasons for exclusion
	gitude as an explanation is due to a spurious correlation and not a probable explanation.
Dimensions with high cardinality (dimensions with > 500 members)	High cardinality dimensions take longer to compute. Dimensions with more than 500 unique values will not be considered for analysis.
Groups, bins, or sets	Not currently supported.
Table calculations	Table calculations cannot be analyzed when table calculations are at a different level of detail than the view.
Unvisualized measures that can't be averaged	Unvisualized measures that can't be averaged include measures that are calculated fields where the calculation expression includes aggregations (display as AGG() fields when added to the sheet).
Discrete measures and continuous dimensions	Not currently supported.
Hidden fields	Not available.
Calculated fields with errors	No values present to analyze.

Create Custom Fields with Calculations

If your underlying data doesn't include all of the fields you need to answer your questions, you can create new fields in Tableau using calculations and then save them as part of your data source. These fields are called calculated fields.

This section provides all the documentation you need to get started with calculations in Tableau.

Get Started with Calculations in Tableau

This article describes how to create and use calculated fields in Tableau using an example.

You'll learn Tableau calculation concepts, as well as how to create and edit a calculated field. You will also learn how to work with the calculation editor, and use a calculated field in the view.

If you're new to Tableau calculations or to creating calculated fields in Tableau, this is a good place to start.

Why Use Calculated Fields

Calculated fields allow you to create new data from data that already exists in your data source. When you create a calculated field, you are essentially creating a new field (or column) in your data source, the values or members of which are determined by a calculation that you control. This new calculated field is saved to your data source in Tableau, and can be used to create more robust visualizations. But don't worry: your original data remains untouched.

You can use calculated fields for many, many reasons. Some examples might include:

- To segment data
- To convert the data type of a field, such as converting a string to a date.
- To aggregate data
- To filter results
- To calculate ratios

Types of calculations

You create calculated fields using calculations. There are three main types of calculations you can use to create calculated fields in Tableau:

- **Basic calculations** - Basic calculations allow you to transform values or members at the data source level of detail (a row-level calculation) or at the visualization level of detail (an aggregate calculation).
- **Level of Detail (LOD) expressions** - Just like basic calculations, LOD calculations allow you to compute values at the data source level and the visualization level. However, LOD calculations give you even more control on the level of granularity you want to compute. They can be performed at a more granular level (INCLUDE), a less granular level (EXCLUDE), or an entirely independent level (FIXED) with respect to the granularity of the visualization.

  For more information, see **Create Level of Detail Expressions in Tableau** on page 2449.
- **Table calculations** - Table calculations allow you to transform values at the level of

detail of the visualization only. For more information, see **Transform Values with Table Calculations** on page 2406.

The type of calculation you choose depends on the needs of your analysis and the question you want to answer.

Create a calculated field

Once you have determined the type of calculation you want to use, it's time to create a calculated field. This example uses a basic calculation.

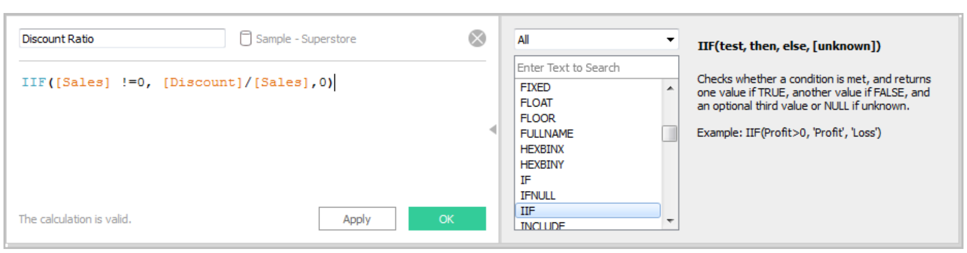
> **Note**: The example in this article uses the **Sample-Superstore** data source that comes with Tableau Desktop. To follow along with the steps in this article, connect to the **Sample-Superstore** saved data source and navigate to **Sheet 1**.

1. In Tableau, select **Analysis** > **Create Calculated Field**.
2. In the Calculation Editor that opens, do the following:
   - Enter a name for the calculated field. In this example, the field is called, **Discount Ratio**.
   - Enter a formula. This example uses the following formula:

     ```
     IIF([Sales] !=0, [Discount]/[Sales],0)
     ```

     This formula checks if sales is not equal to zero. If true, it returns the discount ratio (Discount/Sales); if false, it returns zero.

     Tip: To see a list of available functions, click the triangle icon on the right-side of the Calculation Editor.

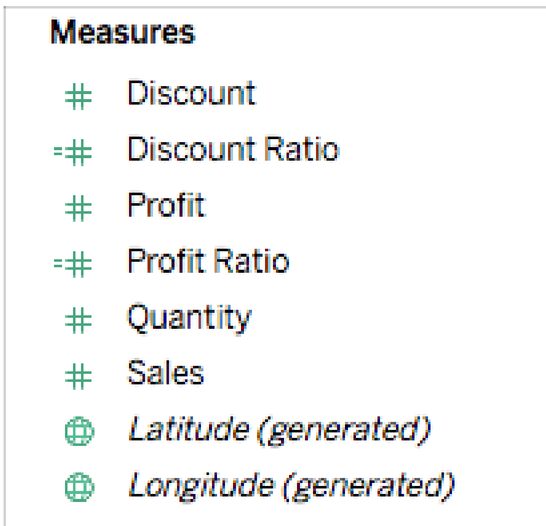
     Each function includes syntax, a description, and an example for your reference.

Double-click a function in the list to add it to the formula.

For more tips, see **Tips for Working with Calculated Fields in Tableau** on page 2494.

3. When finished, click **OK**.

   The new calculated field is added to Measures in the Data pane because it returns a number. An equal sign (=) appears next to the data type icon. All calculated fields have equal signs (=) next to them in the **Data** pane.

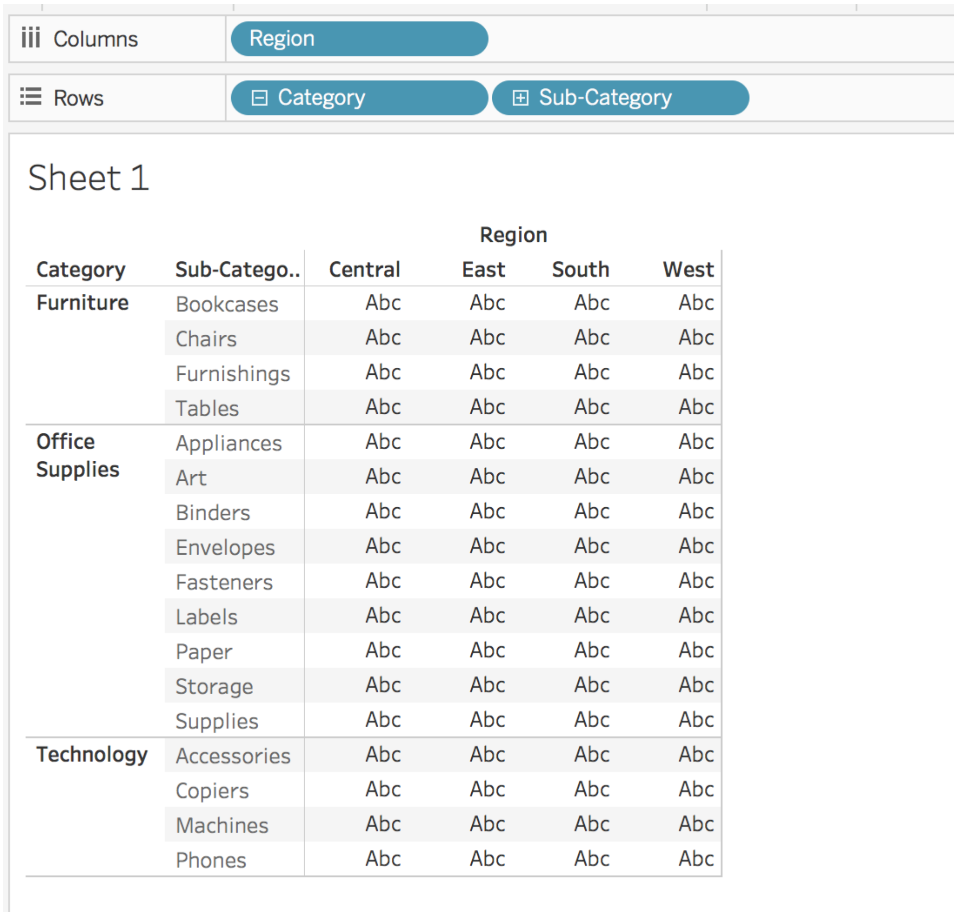


Use a calculated field in the view

Step 1: Build the view

1. From Dimensions, drag **Region** to the **Columns** shelf.
2. From Dimensions, drag **Category** to the **Rows** shelf.
3. On the **Rows** shelf, click the plus icon (**+**) on the **Category** field to drill-down to Subcategory.

The view updates to look like this:

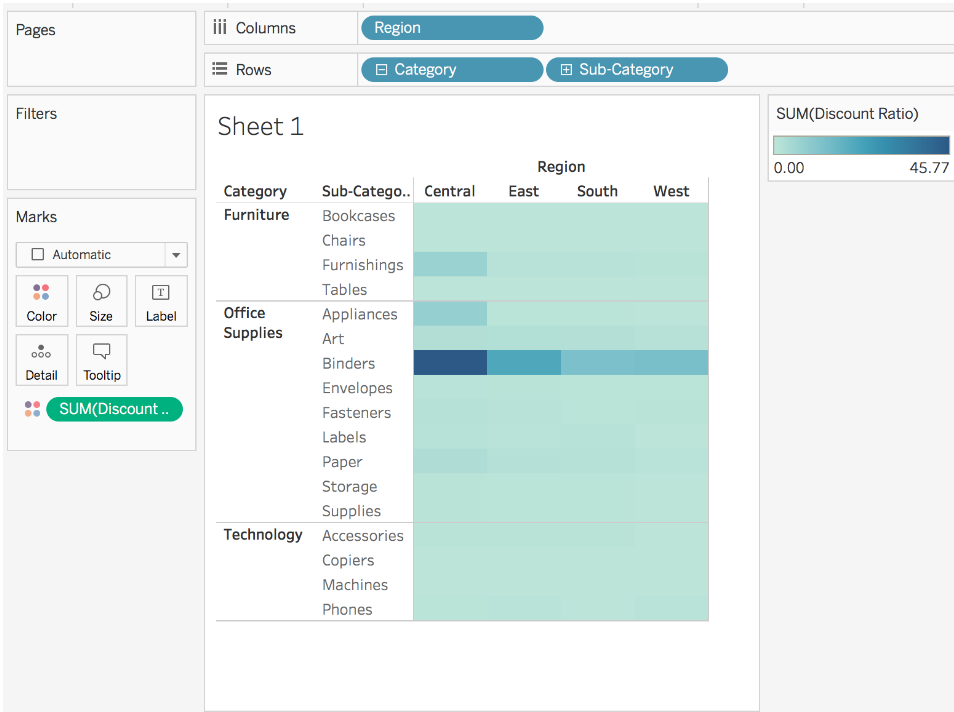


		Region			
Category	Sub-Catego..	Central	East	South	West
Furniture	Bookcases	Abc	Abc	Abc	Abc
	Chairs	Abc	Abc	Abc	Abc
	Furnishings	Abc	Abc	Abc	Abc
	Tables	Abc	Abc	Abc	Abc
Office Supplies	Appliances	Abc	Abc	Abc	Abc
	Art	Abc	Abc	Abc	Abc
	Binders	Abc	Abc	Abc	Abc
	Envelopes	Abc	Abc	Abc	Abc
	Fasteners	Abc	Abc	Abc	Abc
	Labels	Abc	Abc	Abc	Abc
	Paper	Abc	Abc	Abc	Abc
	Storage	Abc	Abc	Abc	Abc
	Supplies	Abc	Abc	Abc	Abc
Technology	Accessories	Abc	Abc	Abc	Abc
	Copiers	Abc	Abc	Abc	Abc
	Machines	Abc	Abc	Abc	Abc
	Phones	Abc	Abc	Abc	Abc

Step 2: Add the calculated field to the view

1. From Measures, drag **Discount Ratio** to **Color** on the Marks card.

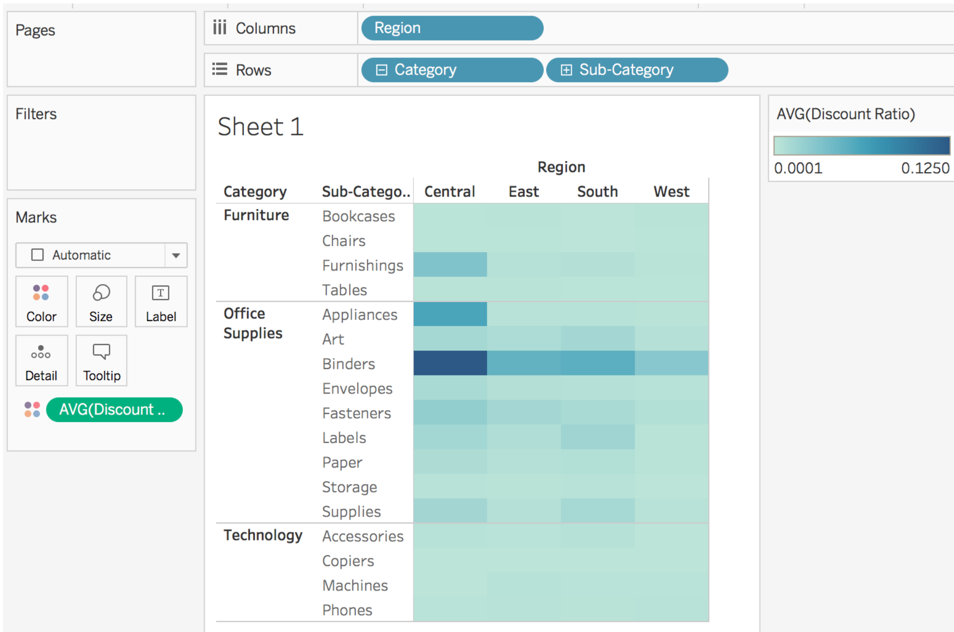
   The view updates to highlight table.



You can see that Binders are heavily discounted in the Central region. Notice that Discount Ratio is automatically aggregated as a sum.

2. On the Rows shelf, right-click **SUM(Discount Ratio)** and select **Measure (Sum)** > **Average**.

   The view updates with the average of discount ratio shown.



Edit a Calculated Field

If at any time you need to change a calculation, you can edit the calculated field and it will update across your entire workbook.

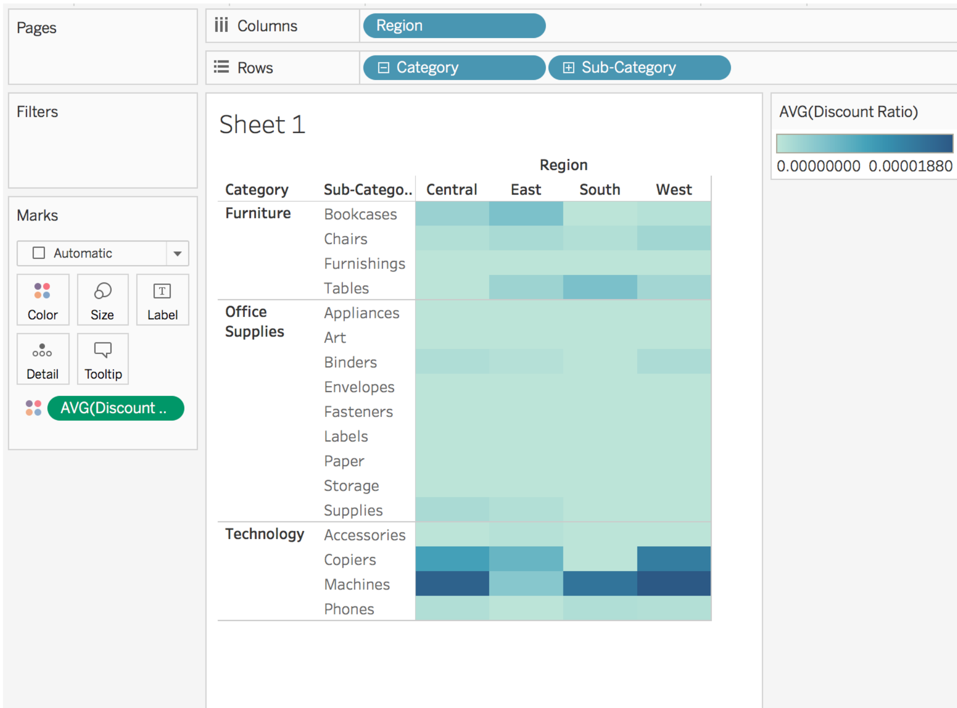
To edit a calculated field:

1. In the **Data** pane, right-click the calculated field and select **Edit**.
2. In the Calculation Editor that opens, you can do the following:
   - Edit the name of the calculated field.
   - Update the formula.

     For this example, the formula is changed to return a discount ratio for orders over 2000 USD in sales:

     ```
     IIF([Sales] > 2000, [Discount]/[Sales],0)
     ```
3. Click **OK**.

   The view updates to reflect the changes automatically. You *do not* need to re-add the updated calculated field to the view.



See Also

Understanding Calculations in Tableau on page 2104

Formatting Calculations in Tableau on page 2135

Functions in Tableau on page 2147

Create Level of Detail Expressions in Tableau on page 2449

Transform Values with Table Calculations on page 2406

Create a Simple Calculated Field

Sometimes your data source does not contain a field (or column) that you need for your analysis. For example, your data source might contain fields with values for Sales and Profit, but not for Profit Ratio. If this is the case, you can create a calculated field for Profit Ratio using data from the Sales and Profit fields.

This topic demonstrates how to create a simple calculated field using an example.

Step 1: Create the calculated field

1. In a worksheet in Tableau, select **Analysis** > **Create Calculated Field**.
2. In the Calculation Editor that opens, give the calculated field a name.

   In this example, the calculated field is called Profit Ratio.

Step 2: Enter a formula

1. In the Calculation Editor, enter a formula.

   This example uses the following formula:

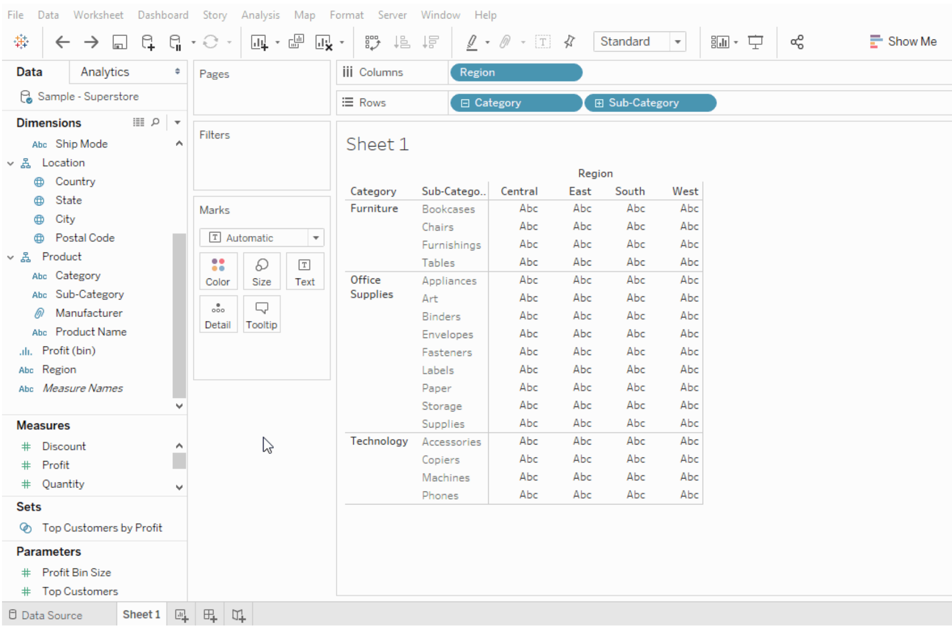
   ```
   SUM([Profit])/SUM([Sales])
   ```

   Formulas use a combination of functions, fields, and operators. To learn more about creating formulas in Tableau, see **Formatting Calculations in Tableau** on page 2135 and **Functions in Tableau** on page 2147.
2. When finished, click **OK**.

   The new calculated field is added to the Data pane. If the new field computes quantitative data, it is added to Measures. If it computes qualitative data, it is added to Dimensions.

   You are now ready to use the calculated field in the view.

Check your work! Watch how to create a simple calculated field in action:



See Also

Get Started with Calculations in Tableau on page 2095

Formatting Calculations in Tableau on page 2135

Functions in Tableau on page 2147

Create Level of Detail Expressions in Tableau on page 2449

Transform Values with Table Calculations on page 2406

Understanding Calculations in Tableau

This series introduces the basics of understanding calculations in Tableau. In this topic, you'll learn why and when to use calculations.

Why use calculations

Calculations allow you to create new data from data that already exists in your data source, as well as perform computations on your data. This allows you to perform complex analyzes and add fields to your data source on your own and on the fly.

When to use calculations

The first hurdle to learning calculations in Tableau is to recognize when you actually need to use one. You can use calculations for many, many reasons. Some examples might include:

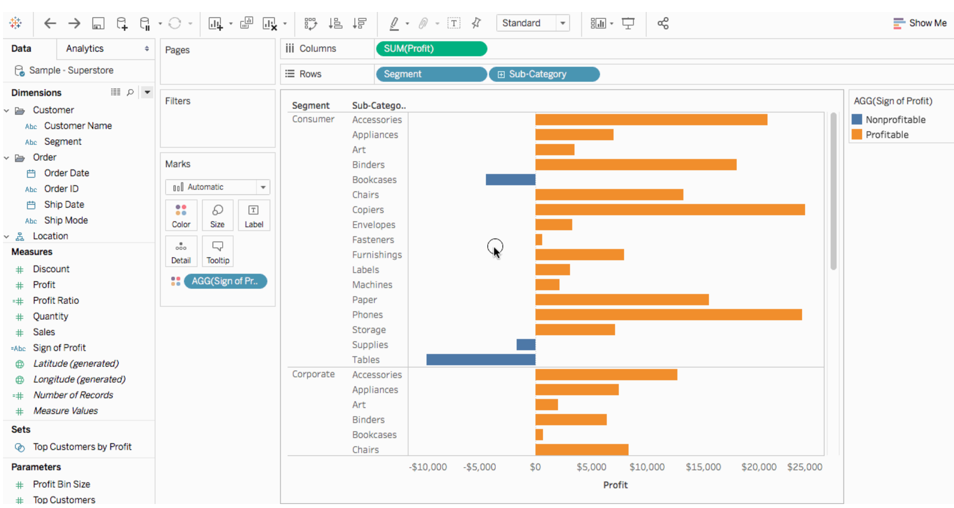
- To segment data
- To convert the data type of a field, such as converting a string to a date.
- To aggregate data
- To filter results
- To calculate ratios

Some common scenarios might include:

- **The data you need for your analysis is missing from your data source.**

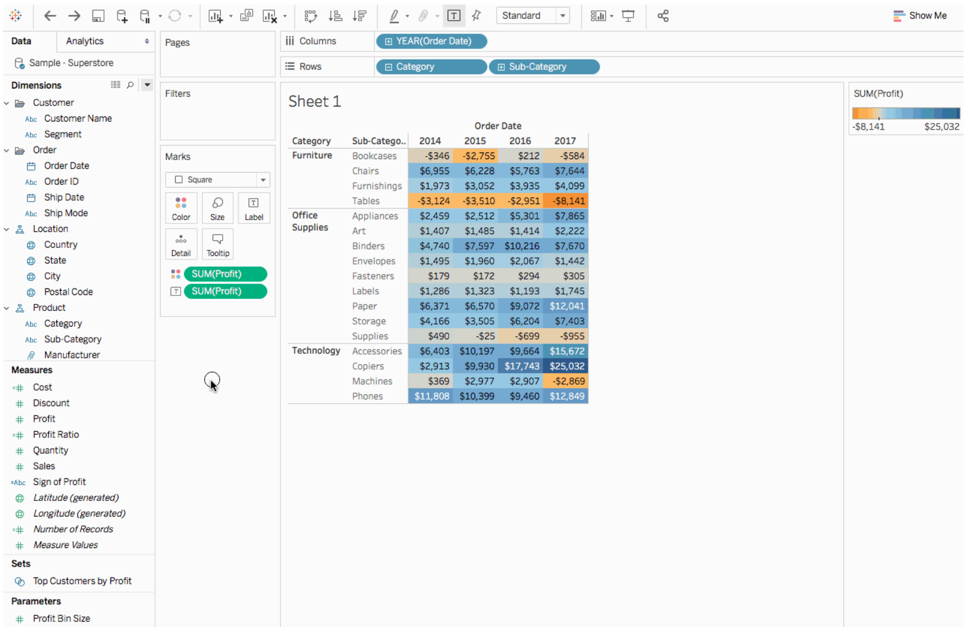
  For example, if you have a Sales and Profit field in your data source, but you want to calculate cost, you can *create* a Cost field using a formula similar to the following.

  ```
  [Sales] - [Profit]
  ```



- **You want to transform values in your visualization.**

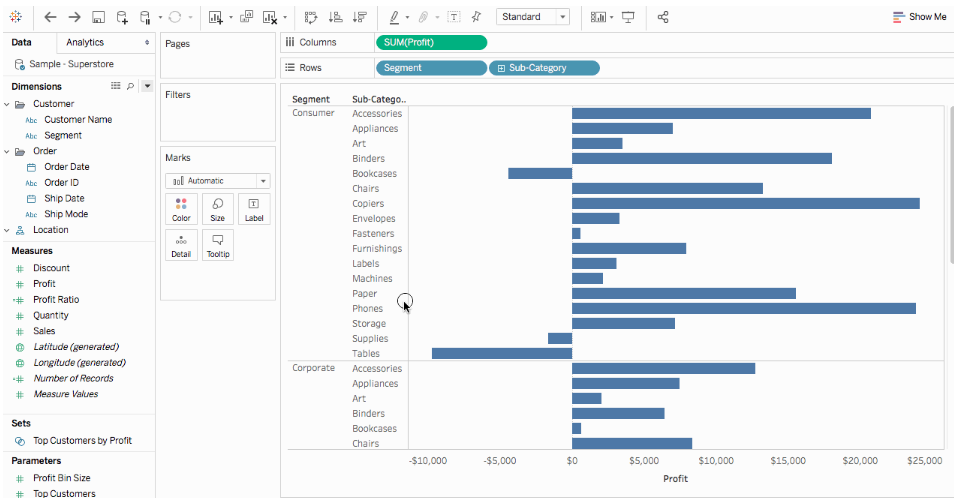
  For example, you might want to calculate the difference in profit from one year to the other. You can use a quick table calculation to show the difference in profit in the visualization.



- **You want to quickly categorize data.**

  For example, you might want to quickly color the data in your visualization as profitable or nonprofitable. You can create a calculated field using a calculation similar to the following and then add it to Color on the Marks card.

```
IF SUM([Profit]) > 0
THEN "Profitable"
ELSE "Nonprofitable"
END
```



Types of Calculations in Tableau

This article explains the types of calculations you can use in Tableau. You'll learn the difference between each calculation and how they are computed.

There are three main types of calculations you can use to create calculated fields in Tableau:

- Basic expressions
- Level of Detail (LOD) expressions
- **Table calculations** on page 2114

Basic expressions

Basic expressions allow you to transform values or members at the data source level of detail (a row-level calculation) or at the visualization level of detail (an aggregate calculation).

For example, consider the following sample table, which contains data on two fantasy authors and their books. Perhaps you want to create a column with only the author's last name and a column that displays how many books are in each series.

Book ID	Book Name	Series	Year Released	Author
1	The Lion, the Witch and the Wardrobe	The Chronicles of Narnia	1950	C.S. Lewis

2	Prince Caspian: The Return to Narnia	The Chronicles of Narnia	1951	C.S. Lewis
3	The Voyage of the Dawn Treader	The Chronicles of Narnia	1952	C.S. Lewis
4	The Silver Chair	The Chronicles of Narnia	1953	C.S. Lewis
5	The Horse and His Boy	The Chronicles of Narnia	1954	C.S. Lewis
6	The Magician's Nephew	The Chronicles of Narnia	1955	C.S. Lewis
7	The Last Battle	The Chronicles of Narnia	1956	C.S. Lewis
8	Daughter of the Forest	Sevenwaters	1999	Juliet Maril-lier
9	Son of the Shadows	Sevenwaters	2000	Juliet Maril-lier
10	Child of the Prophecy	Sevenwaters	2001	Juliet Maril-lier
11	Heir of Sevenwaters	Sevenwaters	2008	Juliet Maril-lier
12	Seer of Sevenwaters	Sevenwaters	2010	Juliet Maril-lier
13	Flame of Sevenwaters	Sevenwaters	2012	Juliet Maril-lier

Row-level calculations

To create a column that displays the author's last name for every row in the data source, you can use the following row-level calculation that splits on a space:

```
SPLIT([Author], '', 2 )
```

The result can be seen below. The new column, titled **Author Last Name** is shown on the far right. The colors demonstrate the level of detail the calculation is performed at. In this case, the calculation is performed at the row-level of the data source, so each row is colored separately.

Book ID	Book Name	Series	Year Released	Author	Author Last Name
1	The Lion, the Witch and the Wardrobe	The Chronicles of Narnia	1950	C.S. Lewis	Lewis
2	Prince Caspian: The Return to Narnia	The Chronicles of Narnia	1951	C.S. Lewis	Lewis
3	The Voyage of the Dawn Treader	The Chronicles of Narnia	1952	C.S. Lewis	Lewis
4	The Silver Chair	The Chronicles of Narnia	1953	C.S. Lewis	Lewis
5	The Horse and His Boy	The Chronicles of Narnia	1954	C.S. Lewis	Lewis
6	The Magician's Nephew	The Chronicles of Narnia	1955	C.S. Lewis	Lewis
7	The Last Battle	The Chronicles of Narnia	1956	C.S. Lewis	Lewis
8	Daughter of the Forest	Sevenwaters	1999	Juliet Marillier	Marillier
9	Son of the Shadows	Sevenwaters	2000	Juliet Marillier	Marillier
10	Child of the Prophecy	Sevenwaters	2001	Juliet Marillier	Marillier
11	Heir of Sevenwaters	Sevenwaters	2008	Juliet	Marillier

				Marillier	
12	Seer of Sevenwaters	Sevenwaters	2010	Juliet Marillier	Marillier
13	Flame of Sevenwaters	Sevenwaters	2012	Juliet Marillier	Marillier

Aggregate calculations

To create a column that displays how many books are in each series, you can use the following aggregate calculation:

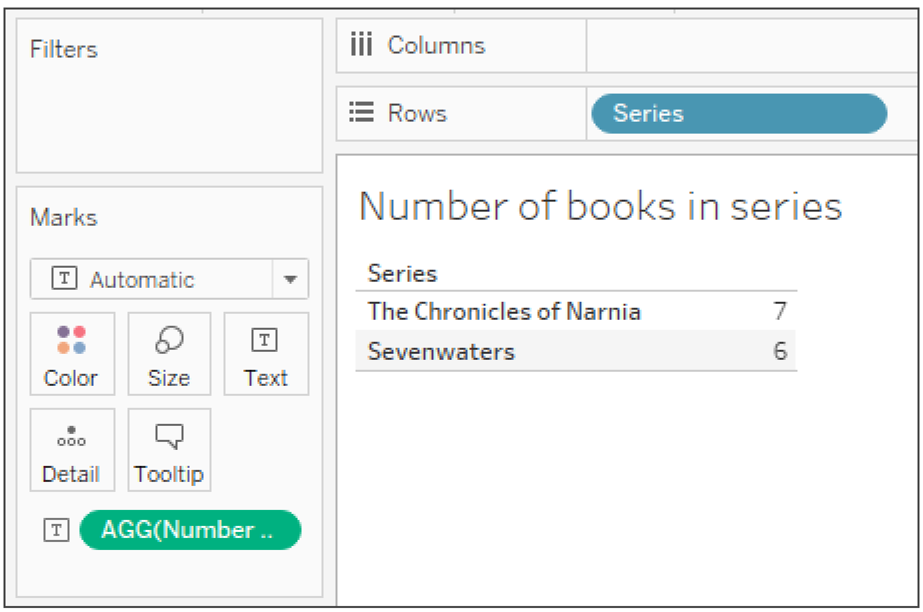
```
COUNT([Series])
```

The result can be seen below. The new column, titled **Number of Books in Series - at Series level of detail** shows how that calculation would be performed at the Series level of detail in the view. The colors help demonstrate the level of detail in which the calculation is being performed.

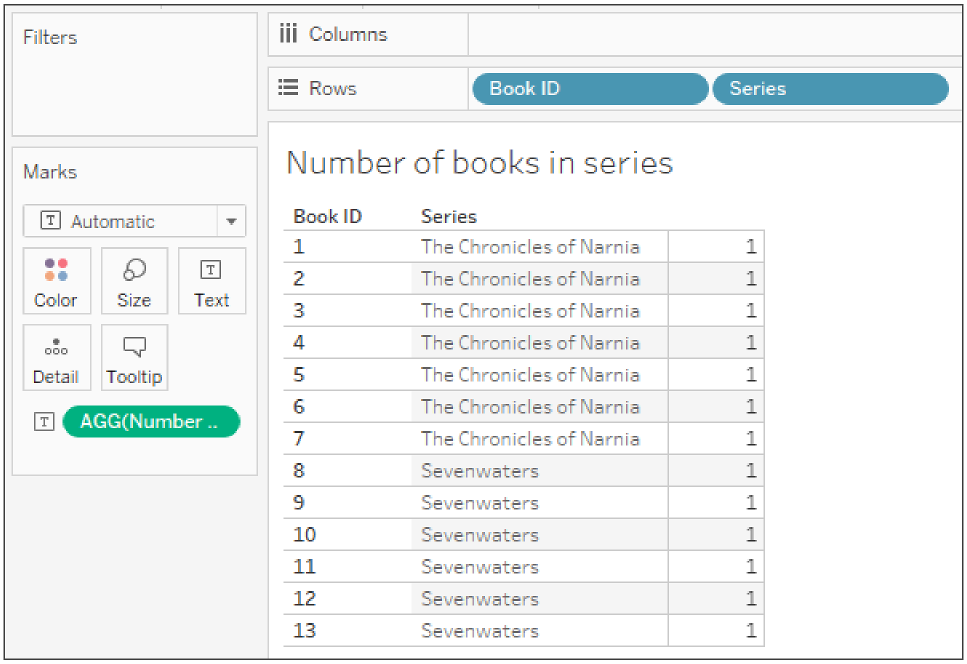
Series	Number of Books in Series - at Series level of detail
The Chronicles of Narnia	7
The Chronicles of Narnia	
The Chronicles of Narnia	
The Chronicles of Narnia	
The Chronicles of Narnia	
The Chronicles of Narnia	
The Chronicles of Narnia	

Sevenwaters	6
Sevenwaters	
Sevenwaters	
Sevenwaters	
Sevenwaters	
Sevenwaters	

In Tableau, the data looks like this:



But if you drag in **Book Id**, (which is a more granular field), the calculation updates based on that new granularity since aggregate calculations are performed at the visualization level of detail.



Level of Detail (LOD) expressions

Just like basic expressions , LOD expressions allow you to compute values at the data source level and the visualization level. However, LOD expressions give you even more control on the level of granularity you want to compute. They can be performed at a more granular level (INCLUDE), a less granular level (EXCLUDE), or an entirely independent level (FIXED).

For more information, see **Create Level of Detail Expressions in Tableau** on page 2449.

For example, consider the same sample table as above. If you wanted to compute when a book series was launched, you might use the following LOD expression:

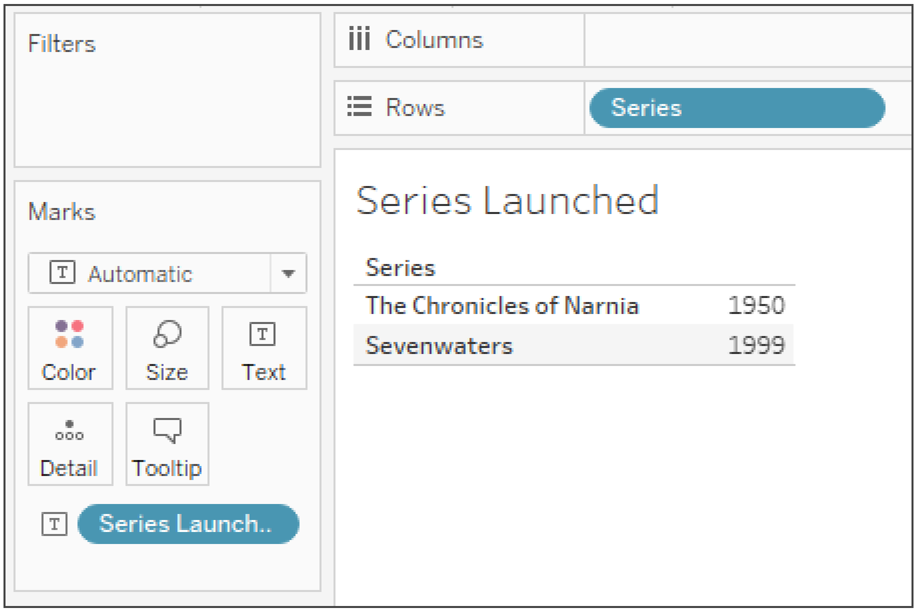
```
{ FIXED [Series]:(MIN([Year Released]))}
```

The result can be seen below. The new column, titled **Series Launched**, displays the minimum year for each series. The colors help demonstrate the level of detail in which the calculation is being applied.

Book ID	Book Name	Series	Year Released	Author	Series Launched
1	The Lion, the Witch and	The Chronicles	1950	C.S.	1950

	the Wardrobe	of Narnia		Lewis	
2	Prince Caspian: The Return to Narnia	The Chronicles of Narnia	1951	C.S. Lewis	1950
3	The Voyage of the Dawn Treader	The Chronicles of Narnia	1952	C.S. Lewis	1950
4	The Silver Chair	The Chronicles of Narnia	1953	C.S. Lewis	1950
5	The Horse and His Boy	The Chronicles of Narnia	1954	C.S. Lewis	1950
6	The Magician's Nephew	The Chronicles of Narnia	1955	C.S. Lewis	1950
7	The Last Battle	The Chronicles of Narnia	1956	C.S. Lewis	1950
8	Daughter of the Forest	Sevenwaters	1999	Juliet Marillier	1999
9	Son of the Shadows	Sevenwaters	2000	Juliet Marillier	1999
10	Child of the Prophecy	Sevenwaters	2001	Juliet Marillier	1999
11	Heir of Sevenwaters	Sevenwaters	2008	Juliet Marillier	1999
12	Seer of Sevenwaters	Sevenwaters	2010	Juliet Marillier	1999
13	Flame of Sevenwaters	Sevenwaters	2012	Juliet Marillier	1999

In Tableau, the calculation remains at the Series level of detail since it uses the FIXED function.



If you add another field to the view (which adds more granularity) the values for the calculation are not affected, unlike an aggregate calculation.

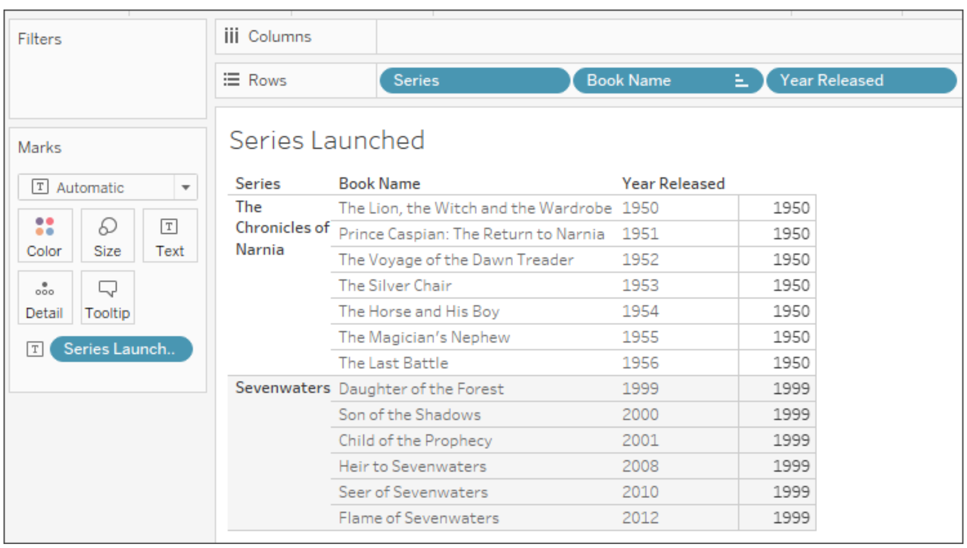


Table calculations

Table calculations allow you to transform values at the level of detail of the visualization only.

For more information, see **Transform Values with Table Calculations** on page 2406.

For example, consider the same sample table as above. If you wanted to compute the number of years since the author released their last book, you might use the following table calculation:

```
ATTR([Year Released]) - LOOKUP(ATTR([Year Released]), -1)
```

The result is shown below. The new column, titled **Years Since Previous Book**, displays the number of years between the book released in that row and the book released in the previous row (on the far right-side of the column) and demonstrates how the table calculation is being computed (on the left-side of the column).

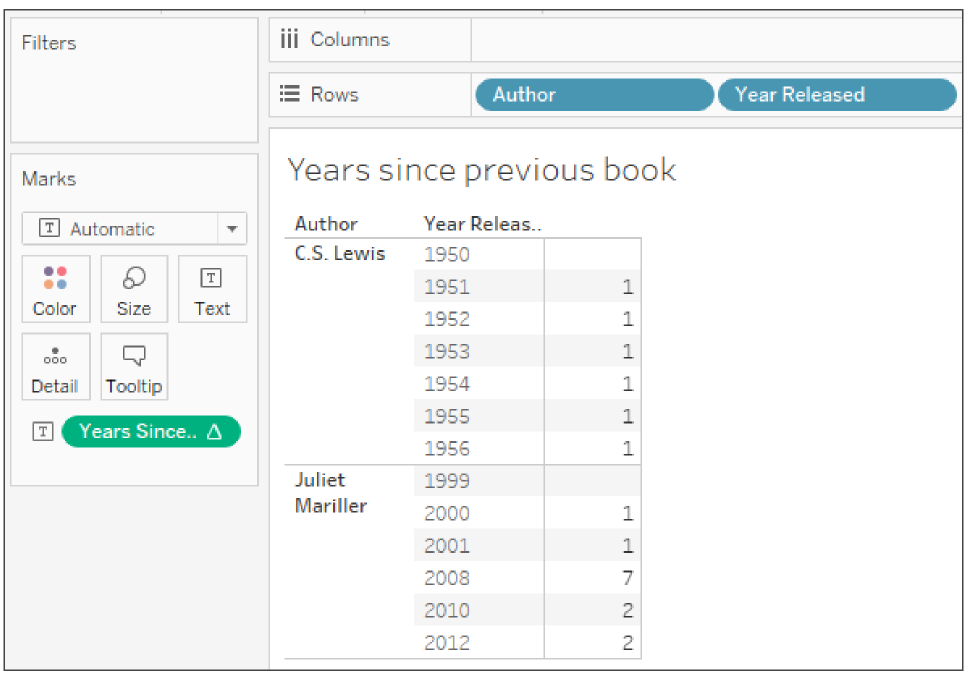
The colors help demonstrate how the table calculation is being computed. In this case, the table calculation is being computed down each pane.

> **Note**: Depending on the table calculation and how it is being computed across the table, the results may vary. For more information, see **Transform Values with Table Calculations** on page 2406.

Book ID	Book Name	Series	Year Released	Author	Years Since Previous Book		
1	The Lion, the Witch and the Wardrobe	The Chronicles of Narnia	1950	C.S. Lewis	NULL		
2	Prince Caspian: The Return to Narnia	The Chronicles of Narnia	1951	C.S. Lewis	1951-	1950	1
3	The Voyage of the Dawn Treader	The Chronicles of Narnia	1952	C.S. Lewis	1952-	1951	1
4	The Silver Chair	The Chronicles of Narnia	1953	C.S. Lewis	1953-	1952	1
5	The Horse and His Boy	The Chronicles of Narnia	1954	C.S. Lewis	1954-	1953	1
6	The Magician's Nephew	The Chronicles of Narnia	1955	C.S. Lewis	1955-	1954	1
7	The Last Battle	The Chron-	1956	C.S.	1956-	1955	1

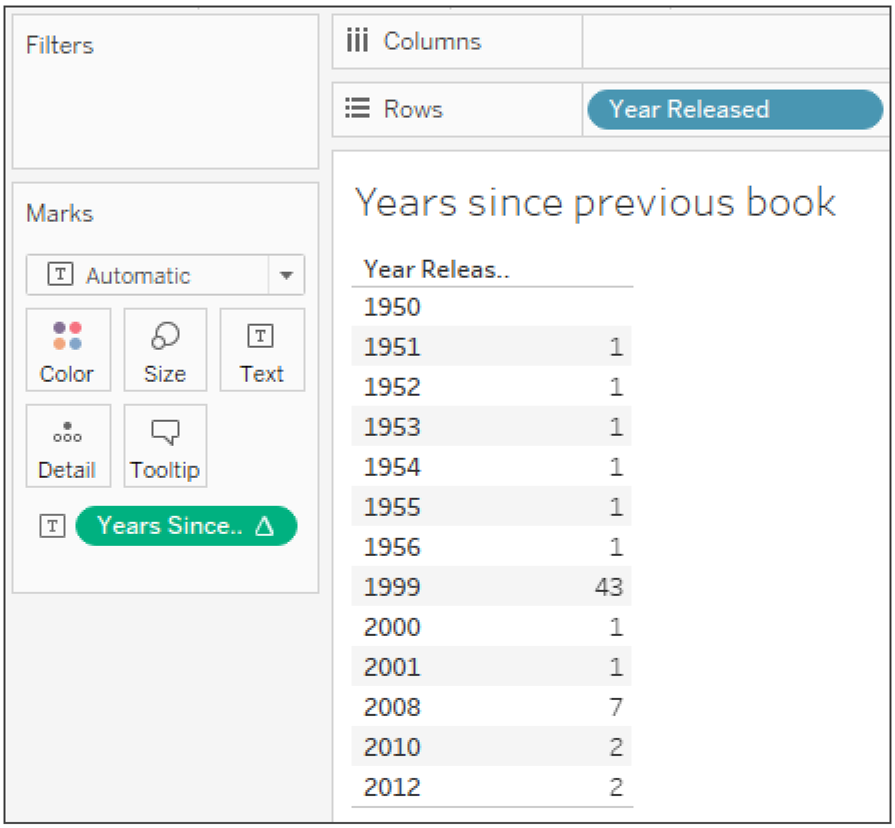
		icles of Narnia		Lewis			
8	Daughter of the Forest	Sevenwaters	1999	Juliet Marillier	NULL		
9	Son of the Shadows	Sevenwaters	2000	Juliet Marillier	2000-	1999	1
10	Child of the Prophecy	Sevenwaters	2001	Juliet Marillier	2001-	2000	1
11	Heir of Sevenwaters	Sevenwaters	2008	Juliet Marillier	2008-	2001	7
12	Seer of Sevenwaters	Sevenwaters	2010	Juliet Marillier	2010-	2008	2
13	Flame of Sev-enwaters	Sevenwaters	2012	Juliet Marillier	2012-	2010	2

In Tableau, the data looks like this:



However, if you change the visualization in a way that affects the layout, such as removing a dimension from the view, the calculation values change.

For example, in the image below, **Author** is removed from the viz. Since the table calculation is computed by pane, removing Author changes the granularity and layout of the viz (instead of two panes there is now only one). The table calculation therefore calculates the time between 1956 and 1999.



*Continue to **Choosing the Right Calculation Type** on the next page*

See Also

Understanding Calculations in Tableau

Tips for Learning How to Create Calculations

Choosing the Right Calculation Type

The type of calculation you choose depends on the needs of your analysis, the question you want to answer, and the layout of your visualization.

Which calculation is right for your analysis?

Choosing the type of calculation to use for your analysis is not always easy. When trying to decide, consider the following questions and examples.

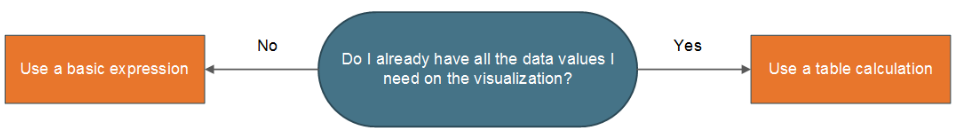
> **Note**: This content was originally published on the Tableau Blog. See A Handy Guide to Choosing the Right Calculation for Your Question to read it.

Basic expression or table calculation?

Question 1: Do you already have all the data values you need on the visualization?

If the answer is **yes**: You can use a table calculation.

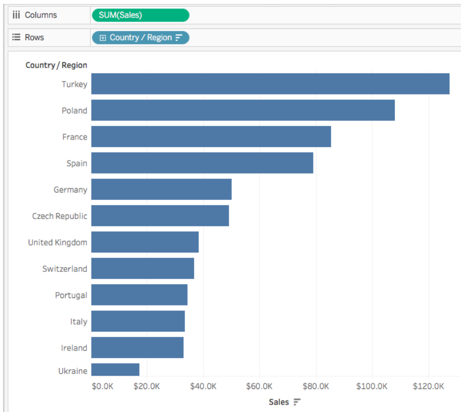
If the answer is **no**: Use a basic calculation.



Example:

Consider the following two visualizations. The visualization on the left is a bar chart that shows the total sales per country/region. The visualization on the right also shows sales per country/region, but sales has been disaggregated.

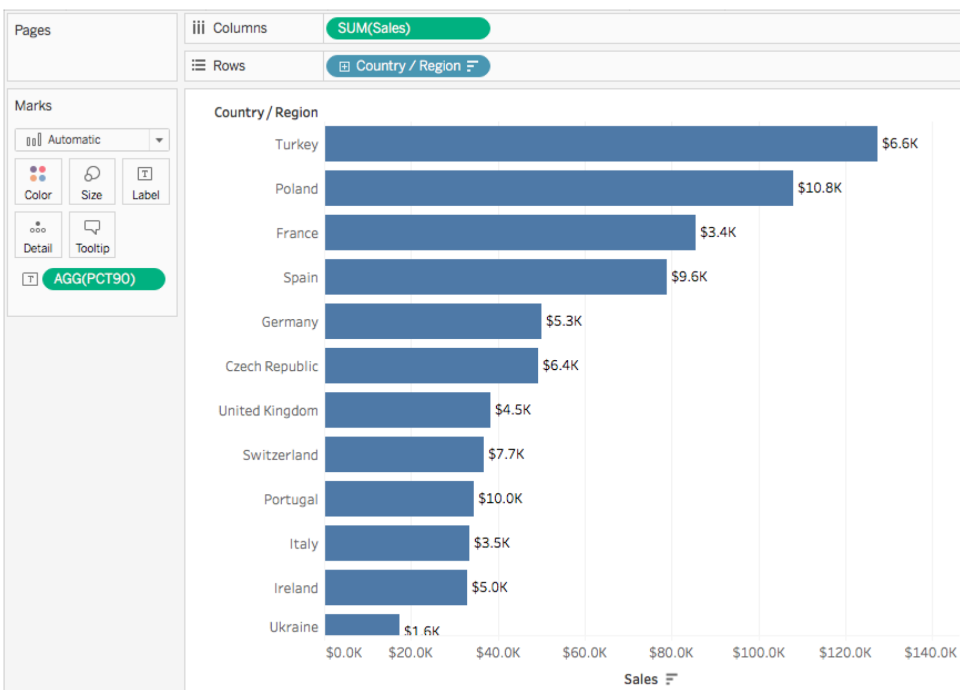
How could you calculate the 90th percentile of sales for each of these visualizations?



The bar chart on the left is aggregated by SUM. Therefore, there is not enough detail in this view to use a table calculation. You can use a basic aggregate expression to calculate the 90th percentile of sales for each country in this example using the following formula:

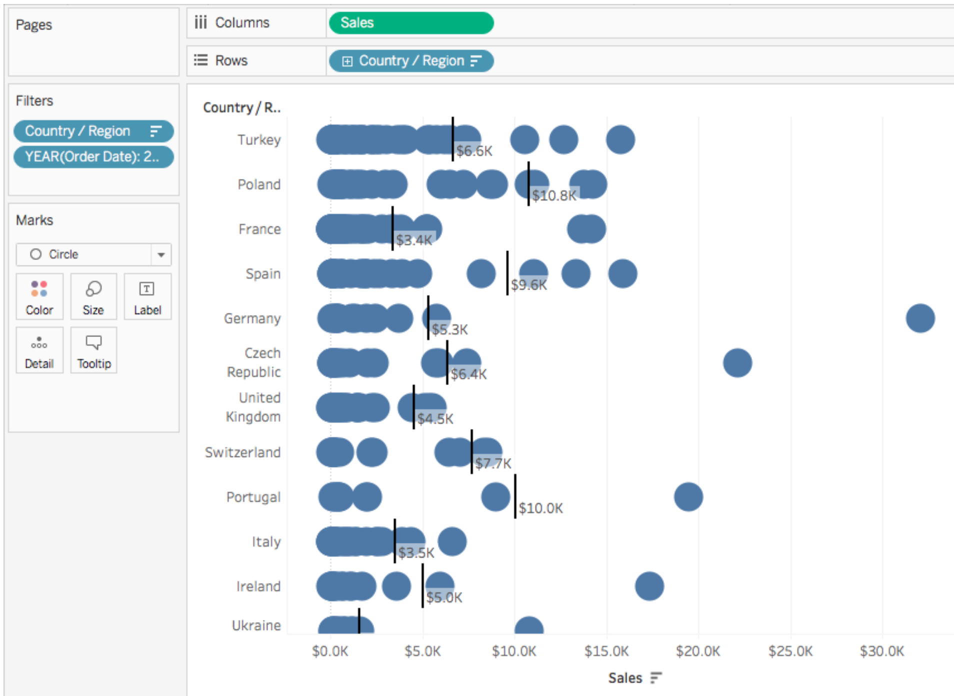
```
PERCENTILE([Sales], .90)
```

This results in a value for 90th percentile per country as a label for each bar.



However, the chart on the left includes a data value for every sales order. A larger distribution and outliers are shown. There is enough detail in the view to use a table calculation.

You can calculate the 90th percentile of sales for each country by using a distribution band (equivalent to a table calculation). There is more context in this visualization.



Both calculations achieve the same values, but the insights you gather from each differ based on the level of detail (the amount of data) in the visualization.

Basic expression or Level of Detail (LOD) expression?

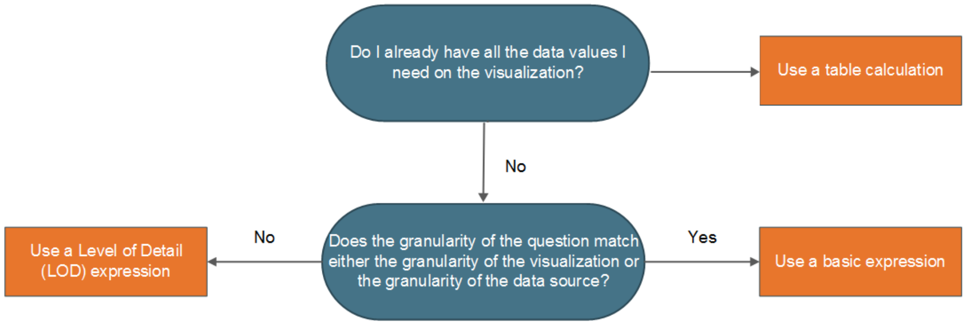
If you don't have all the data you need on the visualization, you need your calculation to be passed through to the data source. This means you must use a basic calculation or an LOD Expression.

If you answered **no** to question 1 ask yourself this:

Question 2: Does the granularity of your question match either the granularity of the visualization or the granularity of the data source?

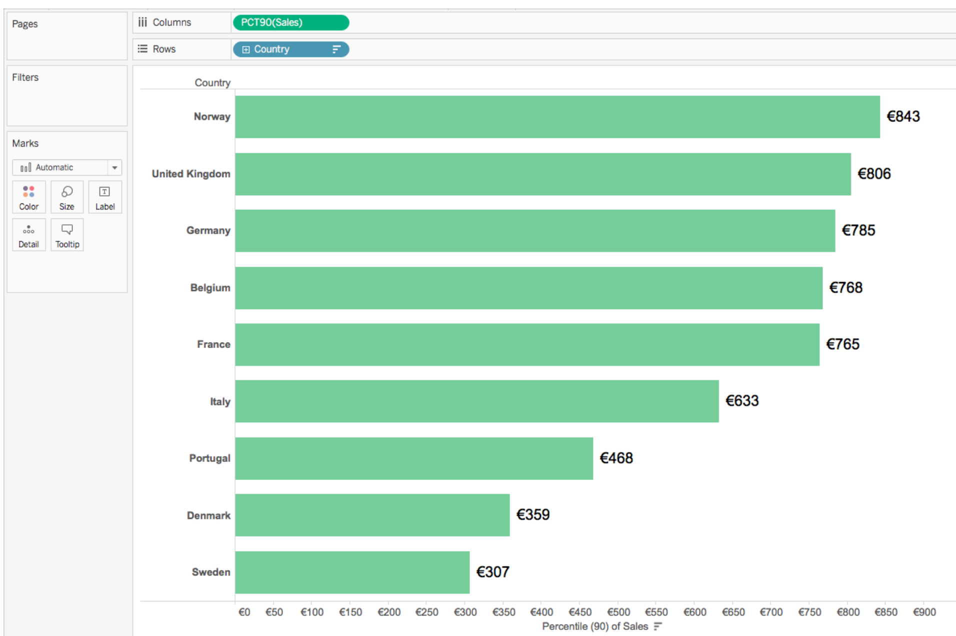
If the answer is **yes**: Use a basic expression.

If the answer is **no**: Use a Level of Detail (LOD) expression.



Example

Consider the following visualization. It shows the 90th percentile of sales for all orders in each country.



This example uses the Sample-Superstore data source that comes with Tableau. If you are familiar with the Sample-Superstore data source, you might know that there is one row of data per Order ID. Therefore, the granularity of the data source is Order ID. The granularity of the visualization, however, is Country.

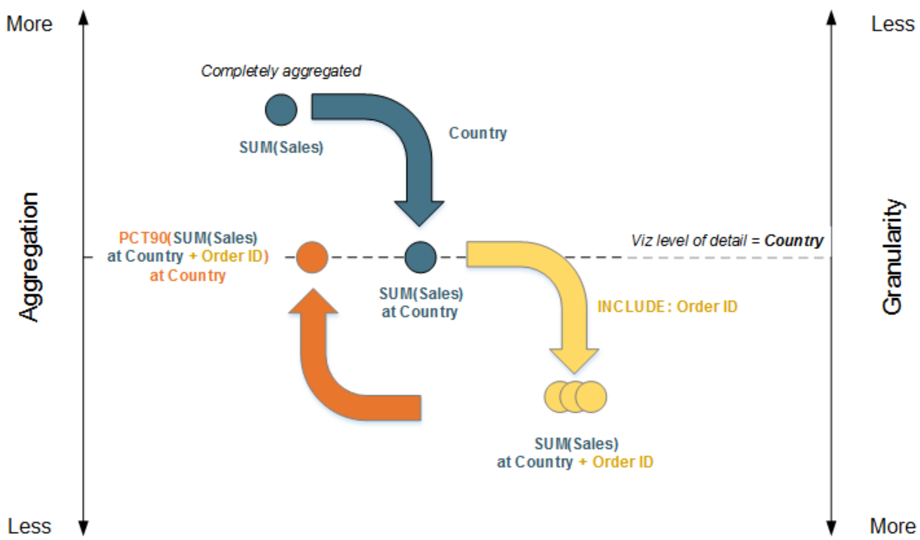
If you want to know what the 90th percentile value of sales is for orders in each country at the order ID level of granularity, you can use the following LOD expression:

```
{INCLUDE [Order ID] : SUM([Sales])}
```

You can then change the field to aggregate at the 90th percentile in the view.

To do so, click the field drop-down and select **Measure** > **Percentile** > **90**.

The following diagram demonstrates how the LOD Expression works in this case:



1. The data starts completely aggregated at SUM(Sales) and then moves down to the Country level of detail: SUM(Sales) at Country.
2. The LOD calculation is applied and the data gains more granularity: SUM(Sales) at Country + Order ID.
3. The LOD calculation is aggregated to the 90th percentile: PCT90(SUM(Sales) at Country + Order ID)

The result is as follows:

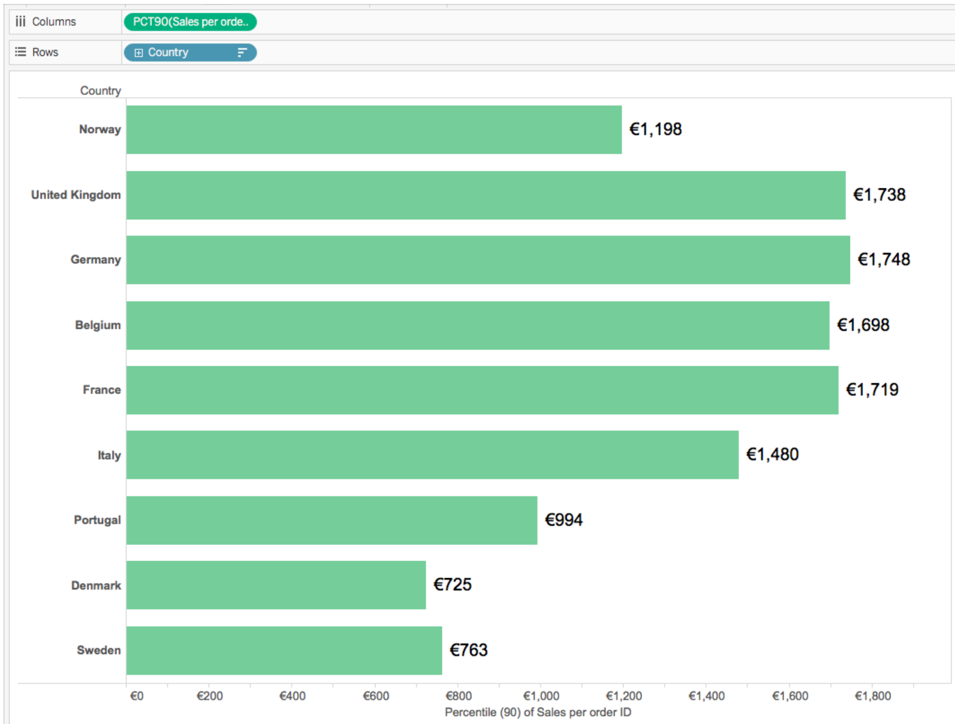


Table calculation or Level of Detail (LOD) expression?

When choosing between a table calculation or an LOD calculation, the process is very similar to choosing between a table calculation and a basic expression. Ask yourself the following questions:

Do you already have all the data values you need on the visualization?

- If the answer is **Yes**, then use a table calculation.
- If the answer is **No**, then ask yourself: *Does the granularity of the question match either the granularity of the visualization or the granularity of the data source?* If the answer is **No**, then use an LOD calculation.

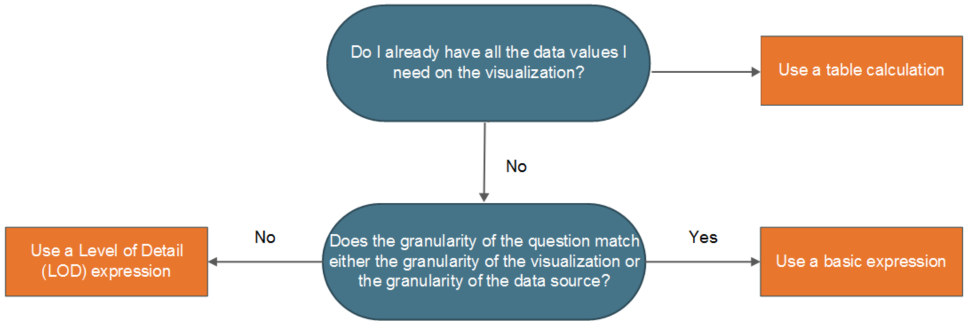
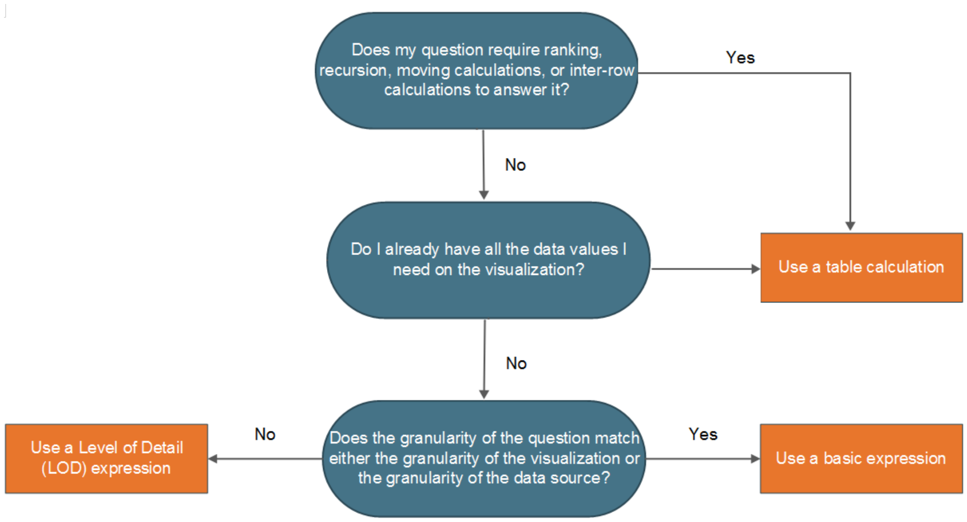


Table calculations only

There are some scenarios where only a table calculation will do. These include:

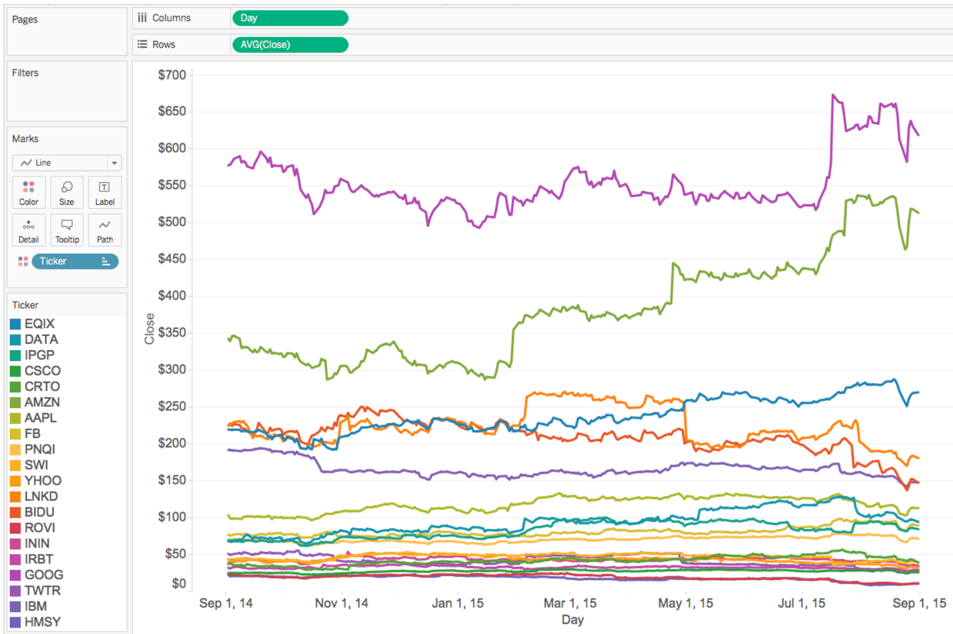
- Ranking
- Recursion (e.g. cumulative totals)
- Moving calculations (e.g. rolling averages)
- Inter-row calculations (e.g. period vs. period calculations)

If your analysis requires any of these scenarios, use a table calculation.



Example

Consider the following visualization. It shows the average closing price for several stocks between September 2014 and September 2015.



If you want to see the number of times the closing price exceeded its record close value to date, you must use a table calculation, specifically a recursive calculation.

Why? Because table calculations can output multiple values for each partition of data (cell, pane, table), while basic and LOD expressions can only output a single value for each partition or grouping of data.

To calculate the number of times the closing price exceeded its record closing price for each stock, there are a few steps you need to take.

1. You need to consider all the previous values before to tell if you have reached a new maximum close value. You can do this with a RUNNING_MAX function. For example, consider the following calculation computed using Day (across the table), titled **Record to Date**:

```
RUNNING_MAX(AVG([Close]))
```

2. Next, you can flag the days when the record was broken using the following calculation computed using Day (across the table), titled **Count Days Record Broken**:

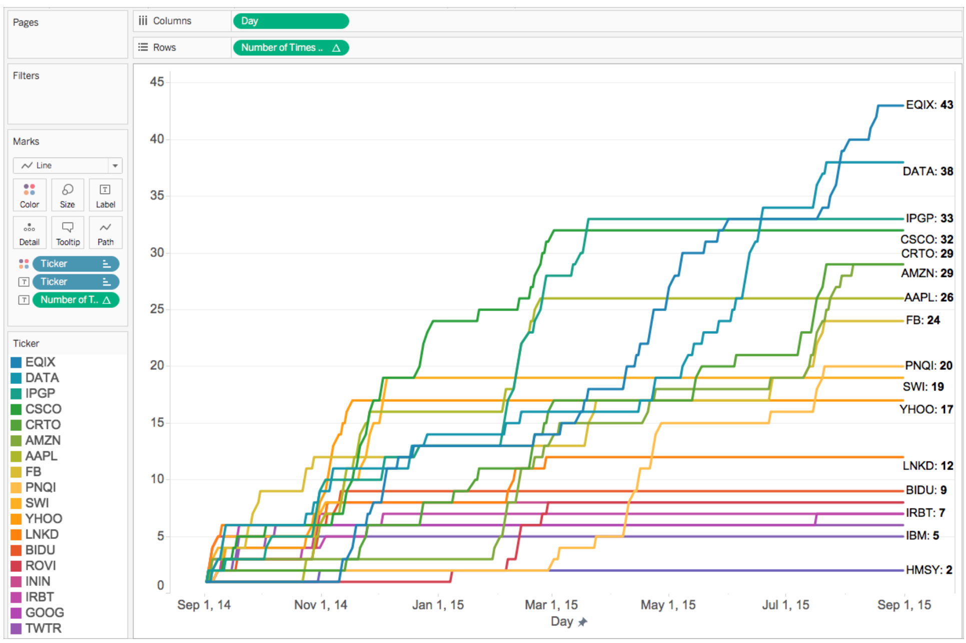
```
IF AVG([Close]) = [Record to Date]
THEN 1
```

```
ELSE 0
END
```

3. Finally, you can count these days using the following calculation computed using Day (across the table):

```
RUNNING_SUM([Count Days Record Broken])
```

When you add the final calculated field to the view in place of Avg(Close), you get something like this:



*Continue to **Tips for Learning How to Create Calculations** on the facing page*

Also in this series:

Understanding Calculations in Tableau

Types of Calculations in Tableau

Tips for Learning How to Create Calculations

Now that you know the difference between the types of calculations and when to use them in your analysis, how do you actually know how to create the formula for them? This is where things can get tricky.

There is no easy way to know exactly how to create the perfect formula; it takes practice and research. However, here are some ways you can start to learn:

- **Know your question or purpose**

  If you know the type of data you need, this can help you choose the correct function, as well as format your formula properly. For example, if you need to show profit for an order, you might create a formula similar to the following: SUM (Sales) - SUM (Cost).

- **Learn Tableau functions**

  There are many different functions available in Tableau. Each type serves a different purpose. To learn more, see **Functions in Tableau** on page 2147.

- **Learn how to format calculations**

  Once you are familiar with the different types of Tableau functions and their purpose, make sure to learn how to format calculations using the proper syntax. To learn more, see **Formatting Calculations in Tableau** on page 2135.

- **Learn from other examples**

  Learning by example is very effective. Try to collect a list of common calculations to use in your everyday analysis. There are several resources available to you:
  - Tableau Community Resource: Calculation Reference Library
  - Tableau Blog: Top 10 Tableau table calculations
  - Tableau Blog: Top 15 LOD Expressions
  - Tableau Blog: Save time and stay consistent with a template of calculated fields

Still having trouble? Ask for help on the Tableau Community.

Also in this series:

Understanding Calculations in Tableau

Types of Calculations in Tableau

Choosing the Right Calculation Type on page 2118

More calculations content:

Get Started with Calculations in Tableau on page 2095

Create a Simple Calculated Field on page 2102

Best Practices for Creating Calculations in Tableau

This article describes several tips and guidelines for creating efficient calculations in Tableau. These guidelines are meant to help you optimize your workbook performance. For more information about all the ways you can improve workbook performance, see the **Optimize Workbook Performance** on page 2993 series.

General Rule: Avoid using a calculated field multiple times in another calculation

Referencing the same calculated field multiple times within another calculation will result in performance issues. If you use a calculated field within a calculation (also known as creating a Nested Calculation), try to reference it only once in the calculation.

Note that referencing a field (terminal field) multiple times in a calculation shouldn't degrade performance.

Example

Let's say you create a calculated field that uses a complicated multiple line calculation to find mentions, or Twitter handles, in tweets. The calculated field is titled, Twitter Handle. Each handle that is returned starts with the '@' sign (for example: @user).

For your analysis, you want to remove the '@' symbol.

To do so, you can use the following calculation to remove the first character from the string:

```
RIGHT([Twitter Handle], LEN([Twitter Handle]) -1)
```

This calculation is quite simple. However, since it references the Twitter Handle calculation twice, it performs that calculation twice for each record in your data source: once for the RIGHT function and again for the LEN function.

In order to avoid calculating the same calculation more than once, you can rewrite the calculation to one that uses the Twitter Handle calculation only once. In this example, you can use MID to accomplish the same goal:

```
MID([Twitter Handle], 2)
```

Tip 1: Convert multiple equality comparisons to a CASE expression or a group

Let's say you have the following calculation, which uses the calculated field, Person (calc), multiple times and employs a series of OR functions. This calculation, though a simple logical expression, will cause query performance issues because it performs the Person (calc) calculation at least ten times.

```
IF [Person (calc)] = 'Henry Wilson'
OR [Person (calc)] = 'Jane Johnson'
OR [Person (calc)] = 'Michelle Kim'
OR [Person (calc)] = 'Fred Suzuki'
OR [Person (calc)] = 'Alan Wang'
THEN 'Lead'
ELSEIF [Person (calc)] = 'Susan Nguyen'
OR [Person (calc)] = 'Laura Rodriguez'
OR [Person (calc)] = 'Ashley Garcia'
OR [Person (calc)] = 'Andrew Smith'
OR [Person (calc)] = 'Adam Davis'
THEN 'IC'
END
```

Instead of using an equality comparison, try the following solutions.

Solution 1

Use a CASE expression. For example:

```
CASE [Person (calc)]
WHEN 'Henry Wilson' THEN 'Lead'
WHEN 'Jane Johnson' THEN 'Lead'
WHEN 'Michelle Kim' THEN 'Lead'
WHEN 'Fred Suzuki' THEN 'Lead'
WHEN 'Alan Wang' THEN 'Lead'

WHEN 'Susan Nguyen' THEN 'IC'
```

```
WHEN 'Laura Rodriguez' THEN 'IC'
WHEN 'Ashley Garcia' THEN 'IC'
WHEN 'Andrew Smith' THEN 'IC'
WHEN 'Adam Davis' THEN 'IC'
END
```

In this example, the calculated field, Person (calc), is only referenced once. Therefore, it is only performed once. CASE expressions are also further optimized in the query pipeline, so you gain an additional performance benefit.

Solution 2

Create a group instead of a calculated field. For more information, see **Group Your Data** on page 1173.

Tip 2: Convert multiple string calculations into a single REGEXP expression

> **Note**: REGEXP calculations are available only when using Tableau data extracts or when connected to Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources. For more information, see **Additional Functions** on page 2260.

Example 1: CONTAINS

Let's say you have the following calculation, which uses the calculated field, Category (calc), multiple times. This calculation, though also a simple logical expression, will cause query performance issues because it performs the Category (calc) calculation multiple times.

```
IF CONTAINS([Segment (calc)],'UNKNOWN')
OR CONTAINS([Segment (calc)],'LEADER')
OR CONTAINS([Segment (calc)],'ADVERTISING')
OR CONTAINS([Segment (calc)],'CLOSED')
OR CONTAINS([Segment (calc)],'COMPETITOR')
OR CONTAINS([Segment (calc)],'REPEAT')
THEN 'UNKNOWN'
ELSE [Segment (calc)] END
```

You can use a REGEXP expression to get the same results without as much repetition.

Solution

```
IF REGEXP_MATCH([Segment (calc)],
'UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT') THEN
'UNKNOWN'
ELSE [Segment (calc)] END
```

With string calculations that use a similar pattern, you can use the same REGEXP expression.

Example 2: STARTSWITH

```
IF STARTSWITH([Segment (calc)],'UNKNOWN')
OR STARTSWITH([Segment (calc)],'LEADER')
OR STARTSWITH([Segment (calc)],'ADVERTISING')
OR STARTSWITH([Segment (calc)],'CLOSED')
OR STARTSWITH([Segment (calc)],'COMPETITOR')
OR STARTSWITH([Segment (calc)],'REPEAT')
THEN 'UNKNOWN'
```

Solution

```
IF REGEXP_MATCH([Segment (calc)], '^
(UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT)') THEN
'UNKNOWN'
ELSE [Segment (calc)] END
```

Note that the '^' symbol is used in this solution.

Example 3: ENDSWITH

```
IF ENDSWITH([Segment (calc)],'UNKNOWN')
OR ENDSWITH([Segment (calc)],'LEADER')
OR ENDSWITH([Segment (calc)],'ADVERTISING')
OR ENDSWITH([Segment (calc)],'CLOSED')
OR ENDSWITH([Segment (calc)],'COMPETITOR')
OR ENDSWITH([Segment (calc)],'REPEAT')
THEN 'UNKNOWN'
ELSE [Segment (calc)] END
```

Solution

```
IF REGEXP_MATCH([Segment (calc)], '
(UNKNOWN|LEADER|ADVERTISING|CLOSED|COMPETITOR|REPEAT)$')
THEN 'UNKNOWN'
ELSE [Segment (calc)] END
```

Note that the '$' symbol is used in this solution.

Tip 3: Manipulate strings with REGEXP instead of LEFT, MID, RIGHT, FIND, LEN

Regular expressions can be a very powerful tool. When doing complex string manipulation, consider using regular expressions. In a lot of cases, using a regular expression will result in a shorter and more efficient calculation. For more information about regular expressions, see the Become a regex regular and wrangle imperfect data post on the Tableau blog.

Example 1

Let's say you have the following calculation, which removes protocols from URLs. For example: "https://www.tableau.com" becomes "www.tableau.com".

```
IF (STARTSWITH([Server], "http://")) THEN
MID([Server], Len("http://") + 1)
ELSEIF(STARTSWITH([Server], "https://")) THEN
MID([Server], Len("https://") + 1)
ELSEIF(STARTSWITH([Server], "tcp:")) THEN
MID([Server], Len("tcp:") + 1)
```

```
ELSEIF(STARTSWITH([Server], "\\")) THEN
MID([Server], Len("\\") + 1)
ELSE [Server]
END
```

Solution

You can simplify the calculation and improve performance by using a REGEXP_REPLACE function.

```
REGEXP_REPLACE([Server], "^(http://|https://|tcp:|\\\\)", "")
```

Example 2

Let's say you have the following calculation, which returns the second part of an IPv4 address. For example: "172.16.0.1" becomes "16".

```
IF (FINDNTH([Server], ".", 2) > 0) THEN
MID([Server],
FIND([Server], ".") + 1,
FINDNTH([Server], ".", 2) - FINDNTH([Server], ".", 1) - 1
)
END
```

Solution

You can simplify the calculation and improve performance by using a REGEXP_EXTRACT function.

```
REGEXP_EXTRACT([Server], "\.([^\.]*)\.")
```

Tip 4: Do not use sets in calculations

If you are using sets in a calculation, consider replacing them with an alternative, but equivalent calculation.

Example

Let's say you have the following calculation, which uses the set, Top Customers (set).

```
IF ISNULL([Customer Name]) OR [Top customers (set)] THEN
[Segment] ELSE [Customer Name] END
```

Solution 1

If the set is simple, you can create a calculated field that returns the same result as the set. For example:

```
CASE [Customer Name]
WHEN 'Henry Wilson' THEN True
WHEN 'Jane Johnson' THEN True
WHEN 'Michelle Kim' THEN True
WHEN 'Fred Suzuki' THEN True
WHEN 'Alan Wang' THEN True
ELSE False
END
```

Note: Using the pattern WHEN TRUE … ELSE is recommended in this situation to avoid performance issues due to the use of sets. It is not a recommended pattern in most scenarios.

Solution 2

If the set is more complex, consider creating a group that maps all the elements in the set to a given value or attribute, such as 'IN', and then modify the calculation to check for that value/attribute. For example:

```
IF ISNULL([Customer Name]) OR [Top Customers(group)]='IN'
THEN [Segment] ELSE [Customer Name] END
```

For more information, see **Group Your Data** on page 1173 and **Create Sets** on page 1186.

Tip 5: Do not use sets to group your data

Sets are meant to make comparisons on subsets of data. Groups are meant to combine related members in a field. Converting sets to groups, such as with the following example, is not recommended:

```
IF [Americas Set] THEN "Americas"
ELSEIF [Africa Set] THEN "Africa"
ELSEIF [Asia Set] THEN "Asia"
ELSEIF [Europe Set] THEN "Europe"
ELSEIF [Oceania Set] THEN "Oceania"
ELSE "Unknown"
END
```

This is not recommended for the following reasons:

- **Sets are not always exclusive**. Some members can appear in multiple sets. For example, Russia could be placed both in the Europe set and the Asia set.
- **Sets cannot always be translated to groups**. If the sets are defined by exclusion, conditions, or limits, it might be difficult or even impossible to create an equivalent group.

Solution

Group your data using the Group feature. For more information, see **Group Your Data** on page 1173.

See Also

Create Efficient Calculations on page 3003

Optimize Workbook Performance on page 2993

Types of Calculations in Tableau on page 2107

Functions in Tableau on page 2147

Formatting Calculations in Tableau

This article describes how to create and format calculations in Tableau. It lists the basic components of calculations, and explains the proper syntax for each.

Calculation building blocks

There are four basic components to calculations in Tableau:

- **Functions** - Statements used to transform the values or members in a field.
- **Fields** - Dimensions or measures (columns) from your data source.
- **Operators** - Symbols that denote an operation.
- **Literal expressions** - Constant values that are represented “as is”, such as "Profitable" and "Unprofitable".

  For example, consider the following calculation:

```
IF [Profit per Day] >  000 THEN "Highly Profitable"
ELSEIF [Profit per Day] <= 0 THEN "Unprofitable"
ELSE "Profitable"
END
```

  The components of this calculation can be broken down into the following:

  Functions: IF, THEN, ELSEIF, ELSE, and END

  Field(s): Profit per Day

  Operators: > and <=

  Literal expressions:

  - String literals: "Highly Profitable", "Unprofitable", and "Profitable"
  - Numeric literals: 2000 and 0

  Note that not all calculations need to contain all four components. For example, a calculation might not contain a literal expression, such as the calculation `SUM([Sales])/SUM([Cost])`, which only contains the function, SUM; the division operator (/); and the fields, Sales and Cost.

Additionally, calculations can contain:

- **Parameters** - Placeholder variables that can be inserted into calculations to replace constant values.
- **Comments** - Notes about a calculation or its parts, not included in the computation of the calculation.

For more information about how to use and format each of these components in a calculation, see the sections below.

At a glance: calculation syntax

Components	Syntax	Example
Functions	See **Tableau Functions (Alphabetical)** on page 2342 for examples of how to format all functions in Tableau.	`SUM (expression)`
Fields	A field in a calculation is often surrounded by brackets [ ]. See **Field syntax** on page 2139 for more information.	`[Sales]`
Operators	`+`, `-`, `*`, `/`, `%`, `==`, `=`, `>`, `<`, `>=`, `<=`, `!=`, `<>`, `^`, `AND`, `OR`, `NOT`, `( )`. See **Operator syntax** on page 2140 for information on the types of operators you can use in Tableau calculations, as well as the order they are performed in a formula.	`[Profit] - [Sales]`

Literal expressions	Numeric literals are written as numbers.
	String literals are written with quotation marks.
	Date literals are written with the # symbol.
	Boolean literals are written as either true or false.
	Null literals are written as null.
	See **Literal expression syntax** on page 2144 for more information.
		`1.3567` or `27`
		`"Unprofitable"`
		`#August 22, 2005#`
		`true` or `false`
		`Null`

Parameters	A parameter in a calculation is surrounded by brackets [ ]. See **Create Parameters** on page 1218 for more information.	`[Profit Bin Size]`
Comments	To enter a comment in a calculation, type two forward slashes // . See **Add comments to a calculation** on page 2145 for more information. Multi-line comments can be added by typing /* to start the comment and */ to end it.	`SUM ([Sales])/SUM ([Profit]) //John's calculation`

		`//To be used for profit ratio` `//Do not edit`

Calculation syntax in detail

See the following sections to learn more about the different components of Tableau calculations and how to format them to work in Tableau.

Function syntax

In Tableau, functions are the main components of a calculation and can be used for a variety of different purposes.

Functions are colored blue in Tableau calculations, with the exception of logical functions, which are colored black.

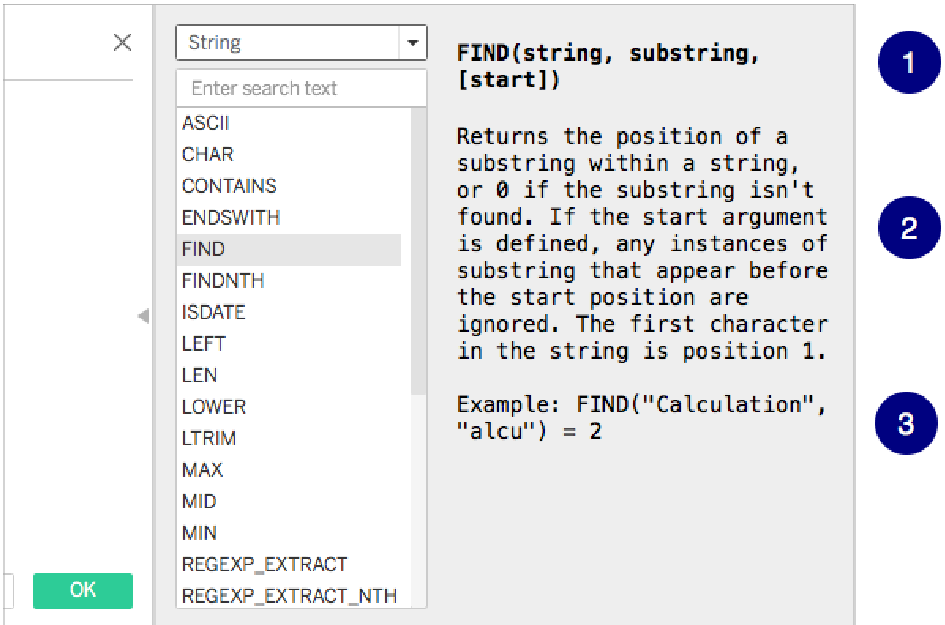
Every function in Tableau requires a particular syntax. For instance, the SUM function requires the following syntax: `SUM(expression)` (i.e. `SUM([Sales])`). The PERCENTILE function, on the other hand, requires the following syntax: `PERCENTILE(expression, number)` (i.e. `PERCENTILE([Sales],0.90)`).

At any time in Tableau, you can look up how to use and format a particular function.

To open the list of functions in Tableau:

1. Select **Analysis** > **Create Calculated Field**.
2. In the Calculation Editor that opens, click the triangle icon (located on the right edge of the editor).

   A list of functions appears for you to browse. When you select a function from the list, the section on the far right updates with information about that function's required syntax (**1**) and description (**2**), as well as with one or more examples (**3**).



Using multiple functions in a calculation

You can use more than one function in a calculation. For example:

```
ZN(SUM([Order Quantity])) - WINDOW_AVG(SUM([Order Quantity]))
```

There are three functions in the calculation: ZN, SUM, and WINDOW_AVG.

The ZN function and the WINDOW_AVG function are separated with the subtraction operator (-).

A function can also be part of another function (or, nested), as is the case with the `ZN(SUM([Order Quantity]))` portion of the example above. In this case, the SUM of Order Quantity is computed before the ZN function because it is inside parentheses. For more information on why, see **Parentheses** on page 2143.

Field syntax

Fields can be inserted into your calculations. Often, a function's syntax will indicate where a field should be inserted into the calculation. For example: `SUM(expression)`.

Field names should be encompassed by brackets [ ] in a calculation when the field name contains a space or is not unique. For example, [Sales Categories].

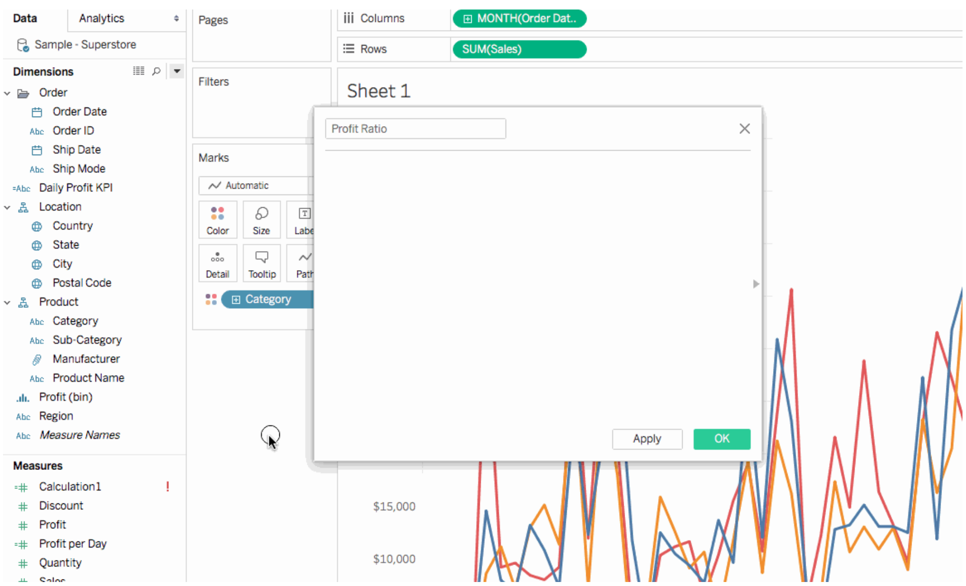
The type of function you use will determine the type of field you use. For example, with the SUM function, you can insert a numerical field, but you cannot insert a date field. For more information, see **Understanding data types in calculations** on page 2146.

The fields you choose to include in your calculations also depends on the purpose of calculation. For example, if you want to calculate profit ratio your calculation will use the Sales and Profit fields from your data source:

```
SUM([Sales])/SUM([Profit])
```

To add a field to a calculation, do one of the following:

- Drag it from the Data pane or the view and drop it in the calculation editor.
- In the Calculation Editor, type the field name. Note: The Calculation Editor attempts to auto-complete field names.



Fields are colored orange in Tableau calculations.

Operator syntax

To create calculations, you need to understand the operators supported by Tableau. This section discusses the basic operators that are available, as well as the order (precedence) they are performed.

Operators are colored black in Tableau calculations.

+ (addition)

The + operator means addition when applied to numbers and concatenation when applied to strings. When applied to dates, it can be used to add a number of days to a date. For example:

```
7 + 3
```

```
Profit + Sales
```

```
'abc' + 'def' = 'abcdef'
```

```
#April 15, 2004# + 15 = #April 30, 2004#
```

- (subtraction)

The - operator means subtraction when applied to numbers and negation if applied to an expression. When applied to dates, it can be used to subtract a number of days from a date. Hence it can also be used to calculate the difference in days between two dates. For example:

```
7 - 3
```

```
Profit - Sales
```

```
-(7+3) = -10
```

```
#April 16, 2004# - 15 = #April 1, 2004#
```

```
#April 15, 2004# - #April 8, 2004# = 7
```

* (multiplication)

The * operator means numeric multiplication. For example, `5 * 4 = 20`.

/ (division)

The / operator means numeric division. For example, `20 / 4 = 5`.

% (modulo)

The % operator returns the remainder of a division operation. For example, 9 % 2 returns 1 because 2 goes into 9 four times with a remainder of 1. Modulo can only operate on integers.

==, =, >, <, >=, <=, !=, <>(comparisons)

These are the basic comparison operators that can be used in expressions. Their meanings are as follows: == or = (equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to), != and <> (not equal to).

Each operator compares two numbers, dates, or strings and returns a either TRUE, FALSE, or NULL.

^ (power)

This symbol is equivalent to the POWER function. It raises a number to the specified power.

For example:

```
6^3 = 216
```

AND

This is a logical operator. An expression or a boolean must appear on either side of it. For example,

```
IIF(Profit =100 AND Sales =1000, "High", "Low")
```

If both expressions are `TRUE` (i.e., not `FALSE` and not `NULL`), then the result is `TRUE`. If either expression is `NULL`, then the result is `NULL`. In all other cases, the result is `FALSE`.

If you create a calculation in which the result of an AND comparison is displayed on a worksheet, Tableau displays TRUE and FALSE. If you would like to change this, use the Format area in the format dialog.

The `AND` operator employs "short circuit evaluation." This means that if the first expression is evaluated to be `TRUE`, then the second expression is not evaluated at all. This can be helpful if the second expression results in an error when the first expression is `TRUE`, because the second expression in this case is never evaluated.

OR

This is a logical operator. An expression or a boolean must appear on either side of it. For example,

```
IIF(Profit =100 OR Sales =1000, "High", "Low")
```

If either expression is `TRUE`, then the result is `TRUE`. If both expressions are `FALSE`, then the result is `FALSE`. If both expressions are `NULL`, then the result is `NULL`.

If you create a calculation in which the result of an `OR` comparison is displayed on a worksheet, Tableau displays TRUE and FALSE. If you would like to change this, use the Format area in the format dialog.

The `OR` operator employs "short circuit evaluation." This means that if the first expression is evaluated to be `TRUE`, then the second expression is not evaluated at all. This can be helpful if the second expression results in an error when the first expression is `TRUE`, because the second expression in this case is never evaluated.

NOT

This is a logical operator. It can be used to negate another boolean or an expression. For example,

```
IIF(NOT(Sales = Profit),"Not Equal","Equal")
```

Operator precedence

All operators in a calculation are evaluated in a specific order. For example, `2*1+2` is equal to 4 and not equal to 6, because multiplication is performed before addition (the * operator is always evaluated before the + operator).

If two operators have the same precedence (such as addition and subtraction (+ or -) they are evaluated from left to right in the calculation.

Parentheses can be used to change the order of precedence. See the **Parentheses** below section for more information.

Precedence	Operator
1	– (negate)
2	^ (power)
3	*, /, %
4	+, –
5	==, >, <, >=, <=, !=
6	NOT
7	AND
8	OR

Parentheses

Parentheses can be used as needed to force an order of precedence. Operators that appear within parentheses are evaluated before those outside of parentheses, starting from the innermost parentheses and moving outward.

For example,(1 + (2*2+1)*(3*6/3)) = 31 because the operators within the innermost parentheses are performed first. The calculation is calculated in the following order:

1. (2*2+1) = 5
2. (3*6/3) = 6
3. (1+ 5*6) = 31

Literal expression syntax

This section describes the proper syntax for using literal expressions in Tableau calculations. A literal expression signifies a constant value that is represented “as is.” When you are using functions you will sometimes want to use literal expressions to represent numbers, strings, dates, and more.

For example, you may have a function where your input is a date. Rather than just type “May 1, 2005”, which would be interpreted a string, you would type #May 1, 2005#. This is equivalent to using a date function to convert the argument from a string to a date (refer to **Date Functions** on page 2170).

You can use numeric, string, date, boolean, and null literals in Tableau calculations. Each type, and how to format them, are described below.

Literal expressions are colored black and gray in Tableau calculations.

Numeric Literals

A numeric literal is written as a number. For example, to input the number one as a numeric literal, enter `1`. If you want to input the number 0.25 as a numeric literal, enter `0.25`.

String Literals

A string literal can be written either using single quote or double quote. If your string has a single or double quote within it, simply type the symbol twice. For example, to input the string “cat” as a string literal, enter `‘cat’` or `“cat”`.

Additionally, if you want to type the string “She’s my friend.” as a string literal, type `‘She’s my friend.’` or `“She’s my friend.”`

Date Literals

Date literals are signified by the pound symbol (#). To input the date “August 22, 2005” as a literal date, enter the ISO formatted date, `#2005-08-22#`.

Boolean Literals

Boolean literals are written as either true or false. To input “true” as a boolean literal, enter `true`.

Null Literals

Null literals are written as Null. To input “Null” as a Null literal, enter `Null`.

Add parameters to a calculation

Parameters are placeholder variables that can be inserted into calculations to replace constant values. When you use a parameter in a calculation, you can then expose a parameter control in a view or dashboard to allow users to dynamically change the value.

For details, see **Use a parameter in a calculation** on page 1224.

Parameters are colored purple in Tableau calculations.

> **Note**: You cannot create parameters as you edit a view on the web.

Add comments to a calculation

You can add comments to a calculation to make notes about it or its parts. Note that comments are not included in the computation of the calculation.

To add a comment to a calculation, type two forward slash (//) characters.

For example:

```
SUM([Sales])/SUM([Profit]) //John’s calculation
```

In this example, //John’s calculation is a comment.

A comment starts at the two forward slashes (//) and goes to the end of the line. To continue with your calculation, you must start a new line.

A multiline comment can be written by starting the comment with a forward slash followed by an asterix(/*), and closed with an asterix followed by a forward slash (*/). For example:

```
SUM([Sales])/SUM([Profit]) /* John's calculation
To be used for profit ratio
Do not edit */
```

Comments are colored gray in Tableau calculations.

Understanding data types in calculations

Tableau supports string, date/datetime, number, and boolean data types. If you create calculated fields, you need to know how to use and combine the different data types in calculations. Many functions that are available to you when you define a calculation only work when they are applied to specific data types.

For example, the `DATEPART()` function can accept only a date/datetime data type as an argument. You can enter `DATEPART('year',#2004-04-15#)` and expect a valid result: 2004. You cannot enter `DATEPART('year',"Tom Sawyer")` and expect a valid result. In fact, this example returns an error because "`Tom Sawyer`" is a string, not a date/datetime.

> **Note**: Although Tableau will attempt to fully validate all calculations, some data type errors cannot be found until the query is run against the database. These issues appear as error dialogs at the time of the query rather than in the calculation dialog box.

The data types supported by Tableau are described below. Refer to **Type Conversion** on page 2182 to learn about converting from one data type to another.

STRING

A sequence of zero or more characters. For example, "`Wisconsin`", "`ID-44400`", and "`Tom Sawyer`" are all strings. Strings are recognized by single or double quotes. The quote character itself can be included in a string by repeating it. For example, `‘O''Hanrahan’`.

DATE/DATETIME

A date or a datetime. For example "`January 23, 1972`" or "`January 23, 1972 12:32:00 AM`". If you would like a date written in long-hand style to be interpreted as a a date/datetime, place the # sign on either side of it. For instance, “`January 23, 1972`” is treated as a string data type but `#January 23, 1972#` is treated as a date/datetime data type.

NUMBER

Numerical values in Tableau can be either *integers* or *floating-point numbers*.

With floating-point numbers, results of some aggregations may not always be exactly as expected. For example, you may find that the SUM function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential distraction by formatting the number to show fewer decimal places. For more information see, **Tableau Functions (by Category)** on page 2267 and select **Numbers**.

Operations that test floating point values for equality can behave unpredictably for the same reason. Such comparisons can occur when using level of detail expressions as dimensions, in categorical filtering, creating ad-hoc groups, creating IN/OUT sets, and with data blending.

> **Note**: The largest signed 64-bit integer is 9,223,372,036,854,775,807. When connecting to a new data source, any column with data type set to **Number (whole)** can accommodate values up to this limit; for larger values, because **Number (whole)** does not use floating-points, Tableau displays "Null". When the data type is set to **Number (decimal)**, larger values can be accommodated.

BOOLEAN

A field that contains the values `TRUE` or `FALSE`. An unknown value arises when the result of a comparison is unknown. For example, the expression `7 > Null` yields unknown. Unknown booleans are automatically converted to Null.

See Also

Understanding Calculations in Tableau on page 2104

Create a Simple Calculated Field on page 2102

Get Started with Calculations in Tableau on page 2095

Functions in Tableau

Tableau supports many functions for use in Tableau calculations. To learn more about the functions in Tableau, see the following articles and reference topics:

Number Functions

This article introduces number functions and their uses in Tableau. It also demonstrates how to create a number calculation using an example.

Why use number functions

Number functions allow you to perform computations on the data values in your fields. Number functions can only be used with fields that contain numerical values. For more information, see

Data Types on page 325.

For example, you might have a field that contains values for the variance in your budget, titled Budget Variance. One of those values might be -7. You can use the ABS function to return the absolute value of that number, and all the other numbers in that field.

The calculation might look something like this:

```
ABS[Budget Variance]
```

Therefore, ABS(-7) = 7.

Number functions available in Tableau

Function	Syntax	Description

ABS	`ABS(number)`	Returns the absolute value of the given number.
		Examples:
		`ABS(-7) = 7`
		`ABS([Budget Variance])`
		The second example returns the absolute value for all the numbers contained in the `Budget Variance` field.

ACOS	`ACOS(number)`	Returns the arc cosine of the given number. The result is in radians. Example: `ACOS(-1) = 3.14159265358979`
ASIN	`ASIN(number)`	Returns the arc sine of a given number. The result is in radians. Example: `ASIN(1) = 1.5707963267949`

ATAN	`ATAN(number)`	Returns the arc tangent of a given number. The result is in radians. Example: `ATAN(180) = 1.5652408283942`
ATAN2	`ATAN2(y number, x number)`	Returns the arc tangent of two given numbers (x and y). The result is in radians. Example: `ATAN2(2, 1) = 1.10714871779409`
CEILING	`CEILING (number)`	Rounds a number to the nearest integer of equal or greater value. Example: `CEILING(3.1415) = 4`

Availability by data source:

Data Source	Support
Microsoft Access	Not supported
Microsoft Excel	Supported
Text File	Supported
Statistical File	Supported
Tableau Server	Supported
Actian Vector	Not supported
Amazon Aurora for MySQL	Not supported

Amazon EMR Hadoop Hive	Supported
Amazon Redshift	Not supported
Aster Database	Not supported
Cloudera Hadoop	Supported
DataStax Enterprise	Supported
EXASOL	Not supported
Firebird	Not supported
Google Analytics	Supported
Google BigQuery	Supported
Google Cloud SQL	Not supported
Google Sheets	Not supported
Hortonworks Hadoop Hive	Supported
IBM BigInsights	Not supported
IBM DB2	Not supported
IBM PDA (Netezza)	Not supported
MapR Hadoop Hive	Supported
MarkLogic	Not supported
Microsoft Analysis Services	Not supported
Microsoft PowerPivot	Not supported
Microsoft SQL Server	Supported
MySQL	Not supported
Oracle	Not supported

		Oracle Essbase	Not supported
		Actian Matrix (ParAccel)	Not supported
		Pivotal Greenplum	Not supported
		PostgreSQL	Not supported
		Progress OpenEdge	Not supported
		Salesforce	Supported
		SAP HANA	Not supported
		SAP Sybase ASE	Not supported
		SAP Sybase IQ	Not supported
		Spark SQL	Supported
		Splunk	Not supported
		Teradata	Not supported
		Teradata OLAP Connector	Not supported
		Vertica	Not supported
COS	`COS(number)`	Returns the cosine of an angle. Specify the angle in radians. Example: `COS(PI( ) /4) = 0.707106781186548`	
COT	`COT(number)`	Returns the cotangent of an angle. Specify the angle in radians. Example: `COT(PI( ) /4) = 1`	

DEGREES	`DEGREES (number)`	Converts a given number in radians to degrees. Example: `DEGREES(PI( )/4) = 45.0`
DIV	DIV(integer1, integer2)	Returns the integer part of a division operation, in which integer1 is divided by integer2. Example: `DIV(11,2) = 5`

EXP	`EXP(number)`	Returns e raised to the power of the given number.
		Examples:
		`EXP(2) = 7.389`
		`EXP(-[Growth Rate]*[Time])`

FLOOR	`FLOOR (number)`	Rounds a number to the nearest integer of equal or lesser value. Example: `FLOOR(3.1415) = 3`

Availability by data source:

Data Source	Support
Microsoft Access	Not supported
Microsoft Excel	Supported
Text File	Supported
Statistical File	Supported

Tableau Server	Supported
Actian Vector	Not supported
Amazon Aurora for MySQL	Not supported
Amazon EMR Hadoop Hive	Supported
Amazon Redshift	Not supported
Aster Database	Not supported
Cloudera Hadoop	Supported
DataStax Enterprise	Supported
EXASOL	Not supported
Firebird	Not supported
Google Analytics	Supported
Google BigQuery	Supported
Google Cloud SQL	Not supported
Hortonworks Hadoop Hive	Supported
IBM BigInsights	Not supported
IBM DB2	Not supported
IBM Netezza	Not supported
MapR Hadoop Hive	Supported
MarkLogic	Not supported
Microsoft Analysis Services	Not supported
Microsoft PowerPivot	Not supported
Microsoft SQL Server	Supported

		MySQL	Not supported
		Oracle	Not supported
		Oracle Essbase	Not supported
		ParAccel	Not supported
		Pivotal Greenplum	Not supported
		PostgreSQL	Not supported
		Progress OpenEdge	Not supported
		Salesforce	Supported
		SAP HANA	Not supported
		SAP Sybase ASE	Not supported
		SAP Sybase IQ	Not supported
		Spark SQL	Supported
		Splunk	Not supported
		Teradata	Not supported
		Teradata OLAP Connector	Not supported
		Vertica	Not supported
HEXBINX	`HEXBINX (number, number)`	Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. HEXBINX and HEXBINY are binning and plotting functions for hexagonal bins. Hexagonal bins are an efficient and elegant option for visualizing data in an x/y plane such as a map. Because the bins are hexagonal, each bin closely approximates a circle and minimizes variation in the distance from the data point to the center of the bin. This makes the clustering both more accurate and informative.	

Example:

```
HEXBINX([Longitude], [Latitude])
```

HEXBINY	`HEXBINY (number, number)`	Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. Example: `HEXBINY([Longitude], [Latitude])`
LN	`LN(number)`	Returns the natural logarithm of a number. Returns `Null` if number is less than or equal to 0.
LOG	`LOG(number [, base])`	Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used.

MAX	`MAX(number, number)`	Returns the maximum of the two arguments, which must be of the same type. Returns `Null` if either argument is `Null`. `MAX` can also be applied to a single field in an aggregate calculation.
		
		Examples:
		
		`MAX(4,7)`
		`MAX(Sales,Profit)`
		`MAX([First Name],[Last Name])`

MIN	`MIN(number, number)`	Returns the minimum of the two arguments, which must be of the same type. Returns `Null` if either argument is `Null`. `MIN` can also be applied to a single field in an aggregate calculation.
		
		Examples:
		
		`MIN(4,7)`
		`MIN(Sales,Profit)`

```
MIN([First Name],[Last Name])
```

PI	`PI( )`	Returns the numeric constant pi: 3.14159.

POWER	`POWER (number, power)`	Raises the number to the specified power.
		Examples:
		`POWER(5,2) = 52 = 25`
		`POWER(Temperature, 2)`
		You can also use the ^ symbol:
		`5^2 = POWER(5,2) = 25`

RADIANS	`RADIANS (number)`	Converts the given number from degrees to radians. Example: `RADIANS(180) = 3.14159`
ROUND	`ROUND (number, [decimals])`	Rounds numbers to a specified number of digits. The `decimals` argument specifies how many decimal points of precision to include in the final result. If `decimals` is omitted, `number` is rounded to the nearest integer. Example: This example rounds every `Sales` value to an integer: `ROUND(Sales)` Some databases, such as SQL Server, allow specification of a negative `length`, where -1 rounds `number` to 10's, -2 rounds to 100's, and so on. This is not true of all databases. For example, it is not true of Excel or Access.

SIGN	`SIGN(number)`	Returns the sign of a number: The possible return values are -1 if the number is negative, 0 if the number is zero, or 1 if the number is positive.
		Example:
		If the average of the profit field is negative, then
		`SIGN(AVG(Profit)) = -1`

SIN	`SIN(number)`	Returns the sine of an angle. Specify the angle in radians.
		Examples:
		`SIN(0) = 1.0`
		`SIN(PI( )/4) = 0.707106781186548`

SQRT	`SQRT(number)`	Returns the square root of a number. Example: `SQRT(25) = 5`
SQUARE	`SQUARE (number)`	Returns the square of a number. Example: `SQUARE(5) = 25`
TAN	`TAN(number)`	Returns the tangent of an angle. Specify the angle in radians.. Example: `TAN(PI ( )/4) = 1.0`
ZN	`ZN (expression)`	Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null

values.

Example:

```
ZN([Profit]) = [Profit]
```

Create a number calculation

Follow along with the steps below to learn how to create a number calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet and select **Analysis** > **Create Calculated Field**.
3. In the calculation editor that opens, do the following:
   - Name the calculated field Minimum Sales transaction
   - Enter the following formula:

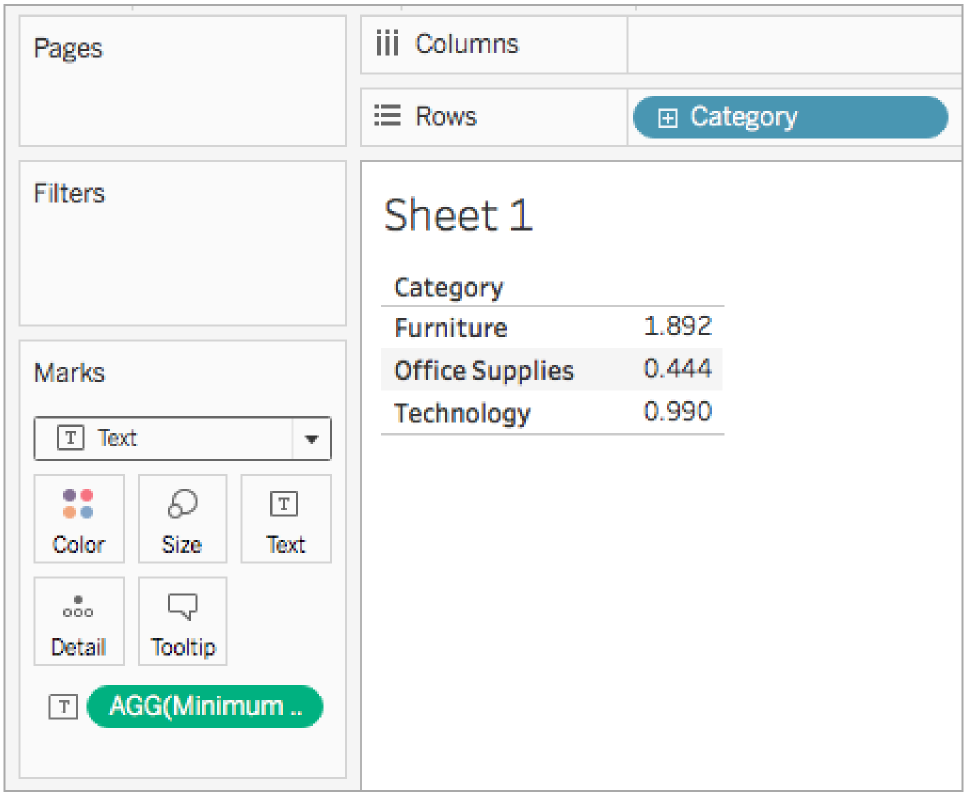
     ```
     MIN(Sales)
     ```

   - When finished, click **OK**.

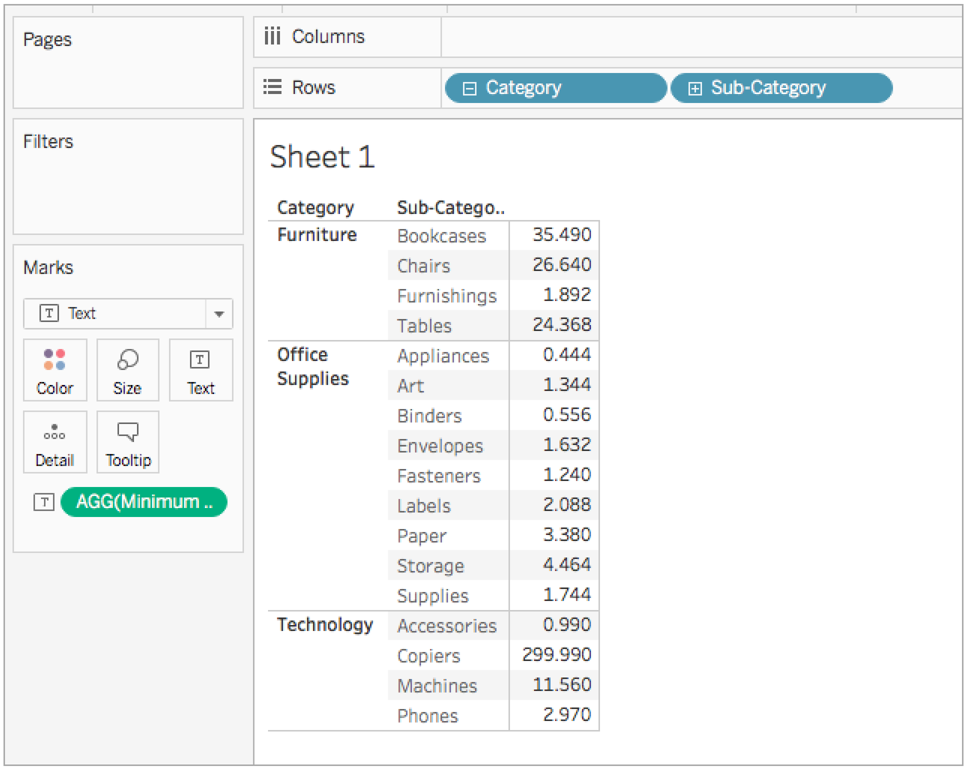
The new number calculation appears under **Measures** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

When Minimum Sales is placed on Text on the Marks card in the worksheet, its name is changed to AGG(Minimum Sales), which indicates that it cannot be aggregated any further, since it is already aggregated down to the lowest level of detail (the smallest sales value for all records).

This example shows the minimum sales per category.



When subcategory is brought into the view, the minimum sales for each subcategory are shown.



See Also

Functions in Tableau on page 2147

Tableau Functions (Alphabetical) on page 2342

Tableau Functions (by Category) on page 2267

Formatting Calculations in Tableau on page 2135

String Functions

This article introduces string functions and their uses in Tableau. It also demonstrates how to create a string calculation using an example.

Why use string functions

String functions allow you to manipulate string data (i.e. data made of text).

For example, you might have a field that contains all of your customers' first and last names. One member might be: Jane Johnson. You can pull the last names from all your customers into a new field using a string function.

The calculation might look something like this:

```
SPLIT([Customer Name], ' ', 2)
```

Therefore, `SPLIT('Jane Johnson' , ' ', 2) = 'Johnson'`.

String functions available in Tableau:

Function	Syntax	Definition
ASCII	`ASCII(string)`	Returns the ASCII code for the first character of `string`. Example: `ASCII('A') = 65`
CHAR	`CHAR(number)`	Returns the character encoded by the ASCII code `number`. Example: `CHAR(65) = 'A'`
CONTAINS	`CONTAINS(string, substring)`	Returns true if the given string contains the specified substring. Example: `CONTAINS(“Calculation”, “alcu”) = true`
ENDSWITH	`ENDSWITH(string, substring)`	Returns true if the given string ends with the specified substring. Trailing white spaces are ignored. Example: `ENDSWITH(“Tableau”, “leau”) = true`

FIND	`FIND (string, substring, [start])`	Returns the index position of `substring` in `string`, or 0 if the `substring` isn't found. If the optional argument `start` is added, the function ignores any instances of `substring` that appear before the index position `start`. The first character in the string is position 1.
		
		Examples:
		
		`FIND("Calculation", "alcu") = 2`
		`FIND("Calculation", "Computer") = 0`
		`FIND("Calculation", "a", 3) = 7`
		`FIND("Calculation", "a", 2) = 2`
		`FIND("Calculation", "a", 8) = 0`
		`FIND("Calculation", "a", 3) = 7`
		`FIND("Calculation", "a", 2) = 2`
		`FIND("Calculation", "a", 8) = 0`

FINDNTH	`FINDNTH (string, substring, occurrence)`	Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument. **Note**: FINDNTH is not available for all data sources. Example: `FINDNTH("Calculation", "a", 2) = 7`
LEFT	`LEFT (string, number)`	Returns the left-most number of characters in the string. Example: `LEFT("Matador", 4) = "Mata"`
LEN	`LEN(string)`	Returns the length of the string.

		Example: `LEN("Matador") = 7`
LOWER	`LOWER (string)`	Returns `string`, with all characters lowercase. Example: `LOWER("ProductVersion") = "productversion"`
LTRIM	`LTRIM (string)`	Returns the string with any leading spaces removed. Example: `LTRIM(" Matador ") = "Matador "`
MAX	`MAX(a, b)`	Returns the maximum of `a` and `b` (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, `MAX` finds the value that is highest in the sort sequence defined by the database for that column. It returns `Null` if either argument is `Null`. Example: `MAX ("Apple","Banana") = "Banana"`
MID	`(MID (string, start, [length])`	Returns the string starting at index position `start`. The first character in the string is position 1. If the optional argument `length` is added, the returned string includes only that number of characters. Examples: `MID("Calculation", 2) =`

```
"alculation"
MID("Calculation", 2, 5) ="alcul"
```

MIN	`MIN(a, b)`	Returns the minimum of `a` and `b` (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, `MIN` finds the value that is lowest in the sort sequence. It returns `Null` if either argument is `Null`. Example: `MIN ("Apple","Banana") = "Apple"`
REPLACE	`REPLACE (string, substring, replace- ment)`	Searches `string` for `substring` and replaces it with `replacement`. If `substring` is not found, the string is not changed. Example: `REPLACE("Version8.5", "8.5", "9.0") = "Version9.0"`
RIGHT	`RIGHT (string, number)`	Returns the right-most number of characters in `string`. Example: `RIGHT("Calculation", 4) = "tion"`
RTRIM	`RTRIM (string)`	Returns `string` with any trailing spaces removed. Example: `RTRIM(" Calculation ") = " Calculation"`

SPACE	`SPACE(number)`	Returns a string that is composed of the specified `number` of repeated spaces. Example: `SPACE(1) = " "`

SPLIT	`SPLIT(string, delimiter, token number)`	Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens.
		The string is interpreted as an alternating sequence of delimiters and tokens. So for the string `abc-defgh-i-jkl`, where the delimiter character is '-', the tokens are abc, defgh, i, and jlk. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.
		Examples:
		`SPLIT ('a-b-c-d', '-', 2) = 'b'`
		`SPLIT ('a|b|c|d', '|', -2) = 'c'`
		Note: The split and custom split commands are available for the following data sources types: Tableau data extracts, Microsoft Excel, Text File, PDF File, Salesforce, OData, Microsoft Azure Market Place, Google Analytics, Vertica, Oracle, MySQL, PostgreSQL, Teradata, Amazon Redshift, Aster Data, Google Big Query, Cloudera Hadoop Hive, Hortonworks Hive, and Microsoft SQL Server.
		Some data sources impose limits on splitting string. The following table shows which data sources support

negative token numbers (splitting from the right) and whether there is a limit on the number of splits allow per data source. A SPLIT function that specifies a negative token number and would be legal with other data sources will return this error with these data sources: “Splitting from right is not support by the data source.”

Data Source	Left/Right Constraints	Maximum Number of Splits	Version limitations
Tableau Data Extract	Both	Infinite	
Microsoft Excel	Both	Infinite	
Text file	Both	Infinite	
Salesforce	Both	Infinite	
OData	Both	Infinite	
Google Analytics	Both	Infinite	
Tableau Data Server	Both	Infinite	Supported in version 9.0.
Vertica	Left only	10	
Oracle	Left only	10	
MySQL	Both	10	
PostgreSQL	Left only prior to ver-	10	

	sion 9.0; both for version 9.0 and above		
Teradata	Left only	10	Version 14 and later
Amazon Redshift	Left only	10	
Aster Database	Left only	10	
Google BigQuery	Left only	10	
Hortonworks Hadoop Hive	Left only	10	
Cloudera Hadoop	Left only	10	Impala supported starting in version 2.3.0.
Microsoft SQL Server	Both	10	2008 and later

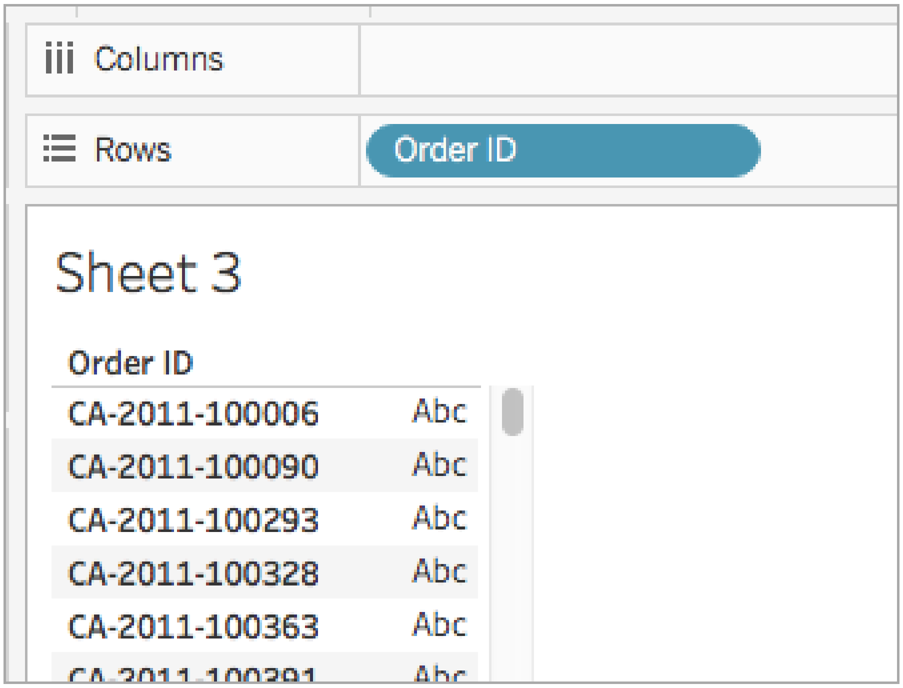
STARTSWITH	`STARTSWITH(string, substring)`	Returns true if `string` starts with `substring`. Leading white spaces are ignored. Example: `STARTSWITH("Joker", "Jo") = true`
TRIM	`TRIM`	Returns the string with leading and trailing spaces

	(string)	removed. Example: `TRIM(" Calculation ") = "Calculation"`
UPPER	UPPER (string)	Returns string, with all characters uppercase. Example: `UPPER("Calculation") = "CALCULATION"`

Create a string calculation

Follow along with the steps below to learn how to create a string calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. From the **Data** pane, under Dimensions, drag **Order ID** to the **Rows** shelf.



Notice that every order ID contains values for country (CA and US, for example), year (2011), and order number (100006). For this example, you will create a calculation to pull only the order number from the field.

4. Select **Analysis** > **Create Calculated Field**.
5. In the calculation editor that opens, do the following:
   - Name the calculated field **Order ID Numbers**.
   - Enter the following formula:

     ```
     RIGHT([Order ID], 6)
     ```

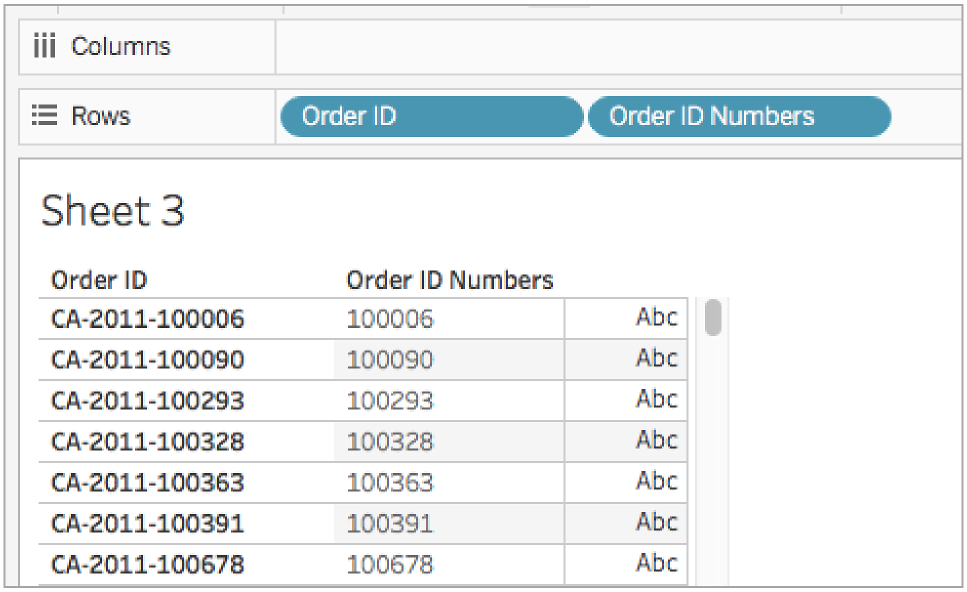
     This formula takes the specified digits (6) from the right of the string and pulls them into a new field.

     Therefore, `RIGHT('CA-2011-100006' , 6)` = '100006'.
   - When finished, click **OK**.

     The new calculated field appears under **Dimensions** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.
6. From the **Data** pane, drag **Order ID Numbers** to the **Rows** shelf. Place it to the right of

Order ID.

Notice how the fields differ now.



See Also

Functions in Tableau on page 2147

Tableau Functions (Alphabetical) on page 2342

Tableau Functions (by Category) on page 2267

Formatting Calculations in Tableau on page 2135

Date Functions

This article introduces date functions and their uses in Tableau. It also demonstrates how to create a date calculation using an example.

Create a date calculation

Follow along with the steps below to learn how to create a date calculation.

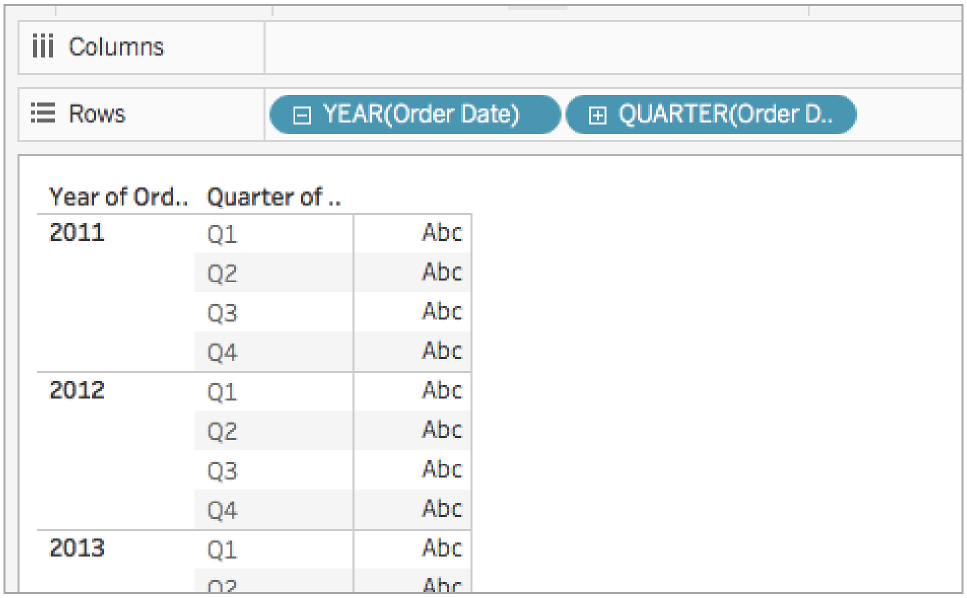
1. In Tableau Desktop, connect to the **Sample-Superstore** saved data source, which comes with Tableau.

2. Navigate to a worksheet.

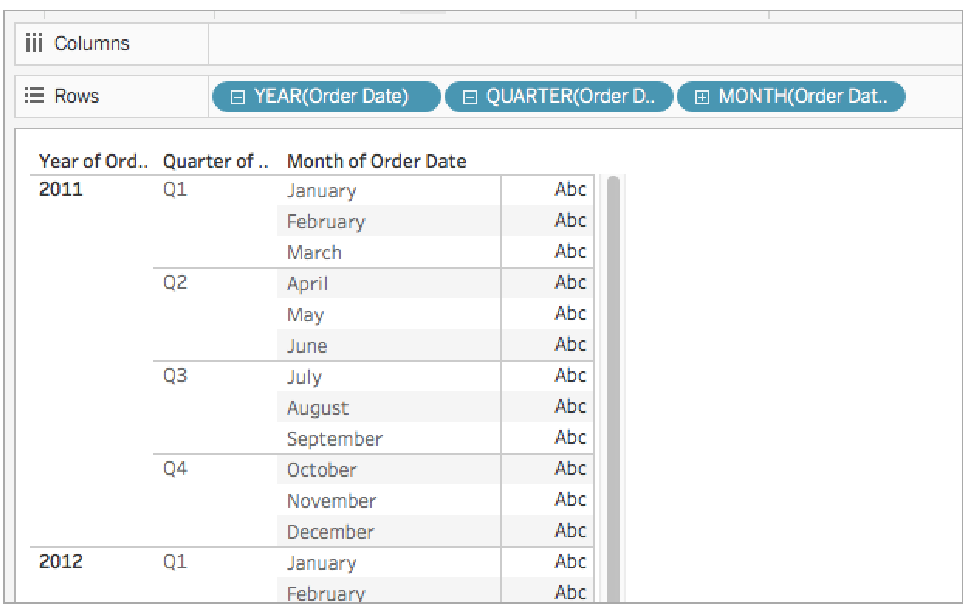
3. From the **Data** pane, under Dimensions, drag **Order Date** to the **Rows** shelf.

4. On the **Rows** shelf, click the plus icon (**+**) on the **YEAR(Order Date)** field.

   QUARTER(Order Date) is added to the Rows shelf and the view updates.



5. On the **Rows** shelf, click the plus icon (**+**) on the **QUARTER(Order Date)** field to drill down to MONTH(Order Date).



6. Select **Analysis** > **Create Calculated Field**.

7. In the calculation editor that opens, do the following:

   - Name the calculated field, Quarter Date.
   - Enter the following formula:

     ```
     DATETRUNC('quarter', [Order Date])
     ```

   - When finished, click **OK**.

     The new date calculated field appears under **Dimensions** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

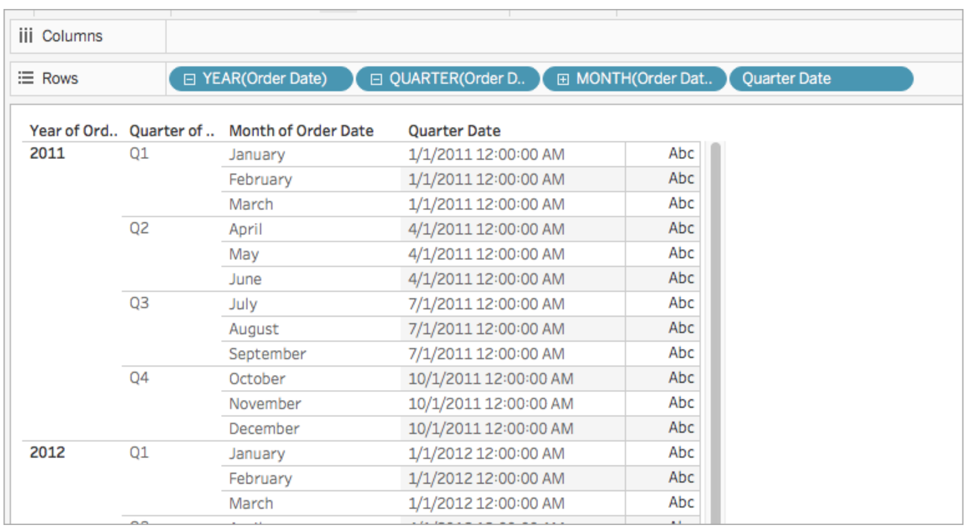
8. From the **Data** pane, under Dimensions, drag **Quarter Date** to the **Rows** shelf and place it to the right of MONTH(Order Date).

   The visualization updates with year values. This is because Tableau rolls date data up to the highest level of detail.

9. On the Rows shelf, right-click **YEAR(Quarter Date)** and select **Exact Date**.

10. On the Rows shelf, right-click **YEAR(Quarter Date)** again and select **Discrete**.

    The visualization updates with the exact quarter date for each row in the table.



Why use date functions

Date functions allow you to manipulate dates in your data source.

For example, you might have a date field with year, month, and day for each value (2004-04-15). From these existing values, you can create new date values with a date function, such as the DATETRUNC function. For example, you can find the date of the beginning of the quarter for any existing date value.

The date calculation might look something like this:

```
DATETRUNC('quarter', [Order Date])
```

So, if the original date is '3/27/2011', using the above calculation would return '1/1/2011' to indicate that Q1 started on January 1. If the original date is '5/3/2011', then the calculation would return '4/1/2011' to indicate that Q2 started on April 1, four months into the year.

See the **Create a date calculation** on page 2170 section below for an example.

Gregorian Calendar vs. ISO 8601 Standard

If you are using a .hyper extract, date functions can be calculated using the traditional Gregorian calendar or the ISO 8601 Standard. For more information on creating a .hyper extract, see **Extract Upgrade to .hyper Format** on page 1031

The ISO 8601 format is an international standard for calculating dates and times that differs from the Gregorian calendar due to how the starting week of a year (Week 1) is calculated. In the Gregorian Calendar, the user can define on which day a week begins. In the ISO 8601 Standard, the week always begins on a Monday.

In a Gregorian Calendar when a new year starts, Week 1 of the year is counted as starting on the 1st of January, regardless of where in the weekday the 1st of January occurs. If January 1st falls on a Saturday, then Week 1 will have one day in it and Week 2 will begin on the following Sunday.

In the ISO 8601 format, Week 1 of a new year begins on a Monday and has four or more days in January. For example, if January 1st falls on a Saturday, then Week 1 will not begin until the following Monday, January 3rd. Calculating dates this way makes sure that there are a consistent number of days in Week 1 of a new year.

For more information on working with ISO 8601 dates, see **ISO-8601 Week-Based Calendar** on page 1273.

Date functions available in Tableau:

Function	Syntax	Description
DATEADD	`DATEADD (date_part, interval, date)`	Returns the specified date with the specified number `interval` added to the specified `date_part` of that date. **Supports ISO 8601 dates.** Example: `DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM` This expression adds three months to the date `#2004-04-15#`.

DATEDIFF	`DATEDIFF (date_part, date1, date2, [start_of_week])`	Returns the difference between `date1` and `date2` expressed in units of `date_part`.
		
		The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.
		
		Supports ISO 8601 dates.
		
		Examples:
		
		`DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday')= 1`
		`DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday')= 0`
		
		The first expression returns 1 because when

		`start_of_week` is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0 because when `start_of_week` is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.

DATENAME	`DATENAME (date_part, date, [start_of_ week])`	Returns `date_part` of `date` as a string. The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.
		
		Supports ISO 8601 dates.
		
		Examples:
		
		`DATENAME('year', #2004-04-15#) = "2004"`
		`DATENAME('month', #2004-04-15#) = "April"`

DATEPARSE	`DATEPARSE (date_ format, [date_ string])`	Returns [date_string] as a date. The date_format argument will describes how the [string] field is arranged. Because of the variety of ways the string field can be ordered, the date_format must match exactly. For a full explanation, see Convert a field to a date field. Example: `DATEPARSE('yyyy-MM-dd', #2004-04-15#) = "April 4, 2004"`

		Note: This function is available through the following connectors: non-legacy Excel and text file connections, Amazon EMR Hadoop Hive, Cloudera Hadoop, Google Sheets, Hortonworks Hadoop Hive, MapR Hadoop Hive, MySQL, Oracle, PostgreSQL, and Tableau extracts. Some formats may not be available for all connections. **Note**: DATEPARSE is not supported on Hive variants. Only Denodo, Drill, and Snowflake are supported.

DATEPART	`DATEPART (date_part, date, [start_of_ week])`	Returns `date_part` of `date` as an integer.
		
		The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.
		
		Note: When the `date_part` is weekday, the `start_of_week` parameter is ignored. This is because Tableau relies on a fixed weekday ordering to apply offsets.
		
		Supports ISO 8601 dates.
		
		Examples:
		
		`DATEPART('year', #2004-04-15#) = 2004`
		`DATEPART('month', #2004-04-`

		`15#) = 4`

DATETRUNC	`DATETRUNC(date_part, date, [start_of_week])`	Truncates the specified date to the accuracy specified by the date_part. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.
		
		Supports ISO 8601 dates.
		
		Examples:
		
		`DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM`
		`DATETRUNC('month', #2004-04-15#) = 2004-04-01 12:00:00 AM`

DAY	`DAY(date)`	Returns the day of the given date as an integer. Example: `DAY(#2004-04-12#) = 12`
ISDATE	`ISDATE(string)`	Returns true if a given string is a valid date. Example: `ISDATE("April 15, 2004") = true`

MAKEDATE	`MAKEDATE (year, month, day)`	Returns a date value constructed from the specified year, month, and date. Available for Tableau Data Extracts. Check for availability in other data sources. Example: `MAKEDATE(2004, 4, 15) = #April 15, 2004#`

MAKEDATETIME	`MAKEDATETIME (date, time)`	Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime.
		
		Note: This function is available only for MySQL-compatible connections (which for Tableau are MySQL and Amazon Aurora).
		
		Examples:
		
		`MAKEDATETIME("1899-12-30", #07:59:00#) = #12/30/1899 7:59:00 AM#`
		`MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#`

MAKETIME	`MAKETIME (hour, minute, second)`	Returns a date value constructed from the specified hour, minute, and second. Available for Tableau Data Extracts. Check for availability in other data sources. Example: `MAKETIME(14, 52, 40) = #14:52:40#`

MAX	`MAX (expression) or MAX (expr1, expr2)`	Usually applied to numbers but also works on dates. Returns the maximum of `a` and `b` (`a` and `b` must be of the same type). Returns `Null` if either argument is `Null`.
		Examples:
		`MAX(#2004-01-01# ,#2004-03-01#) = 2004-03-01 12:00:00 AM`
		`MAX([ShipDate1], [ShipDate2])`

MIN	`MIN (expression) or MIN (expr1, expr2)`	Usually applied to numbers but also works on dates. Returns the minimum of `a` and `b` (`a` and `b` must be of the same type). Returns `Null` if either argument is `Null`.
		Examples:
		`MIN(#2004-01-01# ,#2004-03-01#) = 2004-01-01 12:00:00 AM`
		`MIN([ShipDate1], [ShipDate2])`

MONTH	`MONTH(date)`	Returns the month of the given date as an integer. Example: `MONTH(#2004-04-15#) = 4`
NOW	`NOW( )`	Returns the current local system date and time. Example: `NOW( ) = 2004-04-15 1:08:21 PM`
QUARTER	`QUARTER ( )`	Returns the quarter of the given date as an integer. Example:

		`WEEK (#2004-04-15#) = 16`
TODAY	`TODAY( )`	Returns the current date. Example: `TODAY( ) = 2004-04-15`
WEEK	`WEEK( )`	Returns the week of the given date as an integer. Example: `WEEK (#2004-04-15#) = 16`
YEAR	`YEAR (date)`	Returns the year of the given date as an integer. Example: `YEAR(#2004-04-15#) = 2004`
ISOQUARTER	`ISOQUARTER (date)`	Returns the ISO8601 week-based quarter of a given date as an integer. Example: `ISOQUARTER (#2005-03-29#) = 2`
ISOWEEK	`ISOWEEK (date)`	Returns the ISO8601 week-based week of a given date as an integer. Example: `ISOWEEK (#2004-03-29#) = 14`
ISOWEEKDAY	`ISOWEEKDAY (date)`	Returns the ISO8601 week-based weekday of a given date as an integer. Example:

		`ISOWEEKDAY (#2004-03-29#) = 1`
ISOYEAR	`ISOYEAR (date)`	Returns the ISO8601 week-based year of a given date as an integer. Example: `ISOYEAR (#2003-12-29#) = 2004`

`date_part` values

Many date functions in Tableau use `date_part`, which is a constant string argument.

The valid `date_part` values that you can use are:

date_part	Values
`'year'`	Four-digit year
`'quarter'`	1-4
`'month'`	1-12 or "January", "February", and so on
`'dayofyear'`	Day of the year; Jan 1 is 1, Feb 1 is 32, and so on
`'day'`	1-31
`'weekday'`	1-7 or "Sunday", "Monday", and so on
`'week'`	1-52
`'hour'`	0-23
`'minute'`	0-59
`'second'`	0-60
`'iso-year'`	Four-digit ISO 8601 year
`'iso-quarter'`	1-4

date_part	Values
'iso-week'	1-52, start of week is always Monday
'iso-weekday'	1-7, start of week is always Monday

For more information on formatting date functions, see the **Literal Expressions** section of the **At a glance: calculation syntax** on page 2137 table.

> **Note**: Date functions do not take account of the configured fiscal year start. See **Fiscal Dates** on page 1258 for more information.

> **Note**: `date_part`returns an integer. The values below are inputs that date_part accepts and the format of that input.

See Also

Fiscal Dates on page 1258

Dates and Times on page 1245

Date Properties for a Data Source on page 1247

Custom Dates on page 1256

Custom Date Formats on page 1261

Tableau Functions (Alphabetical) on page 2342

Tableau Functions (by Category) on page 2267

Formatting Calculations in Tableau on page 2135

Functions in Tableau on page 2147

Type Conversion

This article introduces type conversion functions and their uses in Tableau. It also demonstrates how to create a type conversion calculation using an example.

Why use type conversion functions

Type conversion functions allow you to convert fields from one data type to another. For example, you can convert numbers to strings, such as age values (numbers) to string values so that Tableau does not try to aggregate them.

The calculation for such a task might look similar to the following:

```
STR([Age])
```

Type conversion functions available in Tableau:

The result of any expression in a calculation can be converted to a specific data type. The conversion functions are `STR()`, `DATE()`, `DATETIME()`, `INT()`, and `FLOAT()`. For example, if you want to cast a floating point number like 3.14 as an integer, you could write `INT (3.14)`. The result would be `3`, which is an integer. The casting functions are described below.

> A boolean can be cast to an integer, float, or string, It cannot be cast to a date. `True` is 1, 1.0, or “1”, while False is 0, 0.0 or “0”. `Unknown` maps to `Null`.

Function	Syntax	Description

DATE	`DATE (expression)`	Returns a date given a number, string, or date expression.
		Examples:
		`DATE([Employee Start Date])`
		`DATE("April 15, 2004") = #April 15, 2004#`
		`DATE("4/15/2004")`
		`DATE(#2006-06-15 14:52#) = #2006-06-15#`
		Quotation marks are required in the second and third examples.

DATETIME	`DATETIME (expression)`	Returns a datetime given a number, string, or date expression. Example: `DATETIME(“April 15, 2005 07:59:00”)`

		`= April 15, 2005 07:59:00`

DATEPARSE	`DATEPARSE (format, string)`	Converts a string to a datetime in the specified format. Support for some locale-specific formats is determined by the computer's system settings. Letters that appear in the data and do not need to be parsed should be surrounded by single quotes (' '). For formats that do not have delimiters between values (for example, MMddyy), verify that they are parsed as expected. The format must be a constant string, not a field value. This function returns `Null` if the data does not match the format.
		
		This function is available for several connectors. For more information, see **Convert a Field to a Date Field** on page 980.
		
		Examples:
		
		`DATEPARSE ("dd.MMMM.yyyy", "15.April.2004") = #April 15, 2004#`
		`DATEPARSE ("h'h' m'm' s's'", "10h 5m 3s") = #10:05:03#`

FLOAT	`FLOAT (expression)`	Casts its argument as a floating point number. Examples: `FLOAT(3)` = `3.000` `FLOAT([Age])` converts every value in the `Age` field to a floating point number.
INT	`INT (expression)`	Casts its argument as an integer. For expressions, this function truncates results to the closest integer toward zero. Examples:

		`INT(8.0/3.0) = 2`
		`INT(4.0/1.5) = 2`
		`INT(0.50/1.0) = 0`
		`INT(-9.7) = -9`
		
		When a string is converted to an integer it is first converted to a float and then rounded.

STR	`STR (expression)`	Casts its argument as a string. Example: `STR([Age])` This expression takes all of the values in the measure called `Age` and converts them to strings.

Create a type conversion calculation

Follow along with the steps below to learn how to create a type conversion calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. Select **Analysis** > **Create Calculated Field**.
4. In the calculation editor that opens, do the following:
    - Name the calculated field, Postal Code String.
    - Enter the following formula:

      ```
      STR([Postal Code])
      ```

      This calculation converts the Postal Code field from a number to a string.
    - When finished, click **OK**.

The new calculated field appears under Dimensions in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

Converting this field from a number to a string ensures that Tableau treats it as a string and not a number (i.e. Tableau does not aggregate it).

See Also

Data Types on page 325

Convert a Field to a Date Field on page 980

Formatting Calculations in Tableau on page 2135

Functions in Tableau on page 2147

Tableau Functions (Alphabetical) on page 2342

Tableau Functions (by Category) on page 2267

Logical Functions

This article introduces logical functions and their uses in Tableau. It also demonstrates how to create a logical calculation using an example.

Why use logical calculations

Logical calculations allow you to determine if a certain condition is true or false (boolean logic). For example, you might want to quickly see if sales for each country you distribute your merchandise to were above or below a certain threshold.

The logical calculation might look something like this:

```
SUM(Sales) > 1,000,000
```

Logical functions available in Tableau:

Function	Syntax	Description
IN	`<expr1> IN <expr2>`	Returns TRUE if any value in `<expr1>` matches any value in `<expr2>`. The values in `<expr1>` can be a Set, list of literal values, or a combined field.

		Examples:
		`SUM([Cost]) IN (1000, 15, 200)`
		`[SET] IN [COMBINED FIELD]`

AND	IF <expr1> AND <expr2> THEN <then> END	Performs a logical conjunction on two expressions. Example: `IF (ATTR([Market]) = "Africa" AND SUM([Sales]) > [Emerging Threshold] )THEN "Well Performing"`
CASE	CASE <expression> WHEN <value1> THEN <return1> WHEN <value2> THEN <return2> ... ELSE <default return> END	Performs logical tests and returns appropriate values. The CASE function evaluates `expression`, compares it to a sequence of values, `value1`, `value2`, etc., and returns a result. When a value that matches `expression` is encountered, CASE returns the corresponding return value. If no match is found, the default return expression is used. If there is no default return and no values match, then Null is returned. CASE is often easier to use than IIF or IF THEN ELSE. Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function , although the CASE function will generally be more concise. Many times you can use a group to get the same results as a complicated case function. Examples: `CASE [Region] WHEN 'West' THEN 1`

		`WHEN 'East' THEN 2 ELSE 3 END`
		
		`CASE LEFT(DATENAME('weekday',[Order Date]),3) WHEN 'Sun' THEN 0 WHEN 'Mon' THEN 1 WHEN 'Tue' THEN 2 WHEN 'Wed' THEN 3 WHEN 'Thu' THEN 4 WHEN 'Fri' THEN 5 WHEN 'Sat' THEN 6 END`

ELSE	`IF <expr> THEN <then> ELSE <else> END`	Tests a series of expressions returning the <then> value for the first true <expr>. Example: `If [Profit] > 0 THEN 'Profitable' ELSE 'Loss' END`
ELSEIF	`IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...] [ELSE <else>] END`	Tests a series of expressions returning the <then> value for the first true <expr>. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END`
END	`IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...] [ELSE <else>] END`	Tests a series of expressions returning the <then> value for the first true <expr>. Must be placed at the end of an expression. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END`

IF	`IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...] [ELSE <else>] END`	Tests a series of expressions returning the <then> value for the first true <expr>. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END`
IFNULL	`IFNULL (expr1, expr2)`	Returns <expr1> if it is not null, otherwise returns <expr2>. Example: `IFNULL([Profit], 0)`
IIF	`IIF(test, then, else, [unknown])`	Checks whether a condition is met, and returns one value if TRUE, another value if FALSE, and an optional third value or NULL if unknown. Example: `IIF([Profit] > 0, 'Profit', 'Loss')`
ISDATE	`ISDATE (string)`	Returns true if a given string is a valid date. Example: `ISDATE("2004-04-15") = True`
ISNULL	`ISNULL (expression)`	Returns true if the expression is NULL (does not contain valid data). Example: `ISNULL([Profit])`
MAX	`MAX(expres-`	Returns the maximum of a single expression across all records or the maximum of two expressions for each

	`sion)` or `Max (expr1, expr2)`	record. Example: `MAX([Sales])`
MIN	`MIN(expression)` or `MIN (expr1, expr2)`	Returns the minimum of an expression across all records or the minimum of two expressions for each record. Example: `MIN([Profit])`
NOT	`IF NOT <expr> THEN <then> END`	Performs logical negation on an expression. Example: `IF NOT [Profit] > 0 THEN "Unprofitable" END`
OR	`IF <expr1> OR <expr2> THEN <then> END`	Performs a logical disjunction on two expressions. Example: `IF [Profit] < 0 OR [Profit] = 0 THEN "Needs Improvement" END`
THEN	`IF <expre> THEN <then> [ELSEIF ,expr2> THEN <then2>...] [ELSE <else>] END`	Tests a series of expressions returning the <then> value for the first true <expr>. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Break even' ELSE 'unprofitable' END`
WHEN	`CASE <expr>`	Finds the first <value> that matches <expr> and returns

	WHEN <Value1> THEN <return1> ... [ELSE <else>] END	the corresponding <return>. Example: CASE [RomanNumberal] WHEN 'I' THEN 1 WHEN 'II' THEN 2 ELSE 3 END
ZN	ZN(expres-sion)	Returns <expression> if it is not null, otherwise returns zero. Example: ZN([Profit])

Create a logical calculation

Follow along with the steps below to learn how to create a logical calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. From the **Data** pane, drag **State** to the **Rows** shelf.
4. From the **Data** pane, drag **Category** to the **Rows** shelf and place it to the right of State.
5. From the **Data** pane, drag **Sales** to the **Columns** shelf.
6. Select **Analysis** > **Create Calculated Field**.
7. In the calculation editor that opens, do the following:
   - Name the calculated field, KPI.
   - Enter the following formula:

     ```
     SUM([Profit]) > 0
     ```

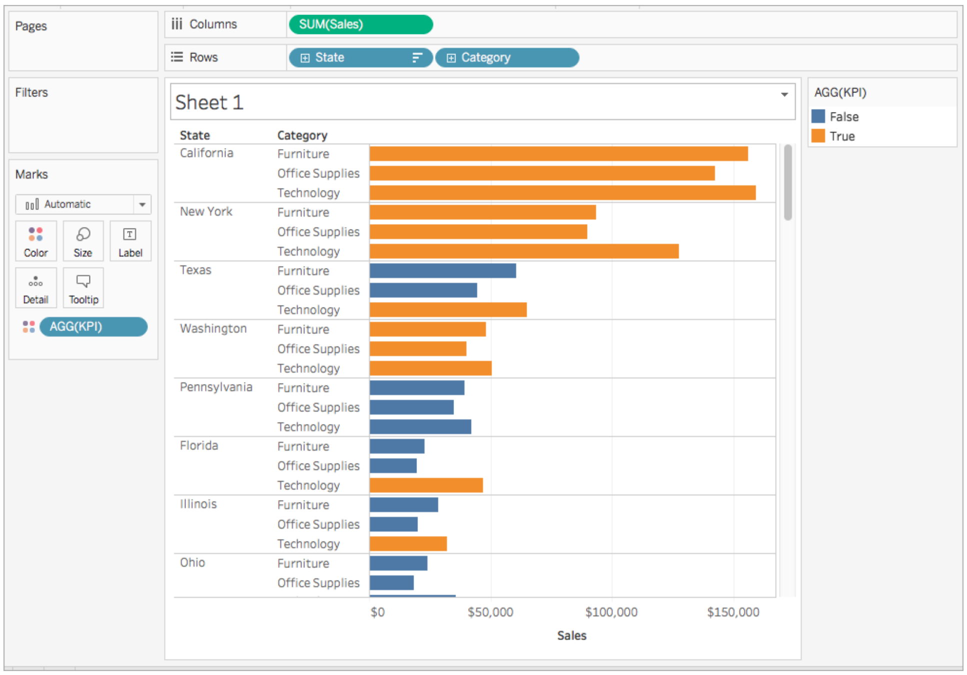
     This calculation quickly checks if a member is great than zero. If so, it returns true; if not, it returns false.

- When finished, click **OK**.

  The new calculated field appears under Measures in the Data pane. Just like your other fields, you can use it in onr or more visualizations.

8. From the **Data** pane, drag **KPI** to **Color** on the Marks card.

   You can now see which categories are losing money in each state.



See Also

Visualize Key Progress Indicators on page 1684

Logic Calculations free video tutorial

Functions in Tableau on page 2147

Tableau Functions (by Category) on page 2267

Tableau Functions (Alphabetical) on page 2342

Aggregate Functions in Tableau

This article introduces aggregate functions and their uses in Tableau. It also demonstrates how to create an aggregate calculation using an example.

Why use aggregate functions

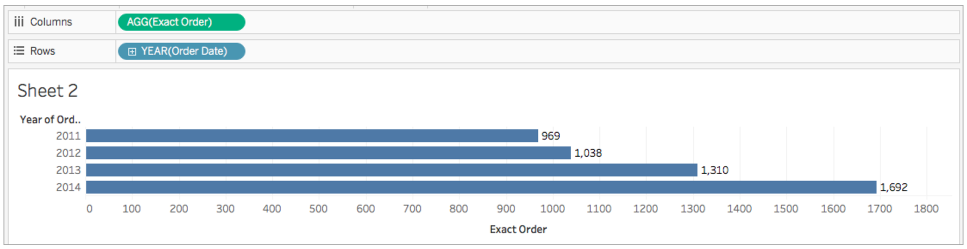
Aggregate functions allow you to summarize or change the granularity of your data.

For example, you might want to know exactly how many orders your store had for a particular year. You can use the COUNTD function to summarize the exact number of orders your company had, and then break the visualization down by year.

The calculation might look something like this:

```
COUNTD(Order ID)
```

The visualization might look something like this:



Aggregate functions available in Tableau

Aggregations and floating-point arithmetic: The results of some aggregations may not always be exactly as expected. For example, you may find that the Sum function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential distraction by using the ROUND function (see **Number Functions** on page 2147) or by formatting the number to show fewer decimal places.

Function	Syntax	Definition
ATTR	`ATTR (expression)`	Returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.
AVG	`AVG (expression)`	Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are

		ignored.
COLLECT	`COLLECT (spatial)`	An aggregate calculation that combines the values in the argument field. Null values are ignored. **Note**: The COLLECT function can only be used with spatial fields. Example: `COLLECT ([Geometry])`
CORR	`CORR (expression 1, expression2)`	Returns the Pearson correlation coefficient of two expressions. The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship. CORR is available with the following data sources: • Tableau data extracts (you can create an extract from any data source) • Cloudera Hive • EXASolution • Firebird (version 3.0 and later) • Google BigQuery • Hortonworks Hadoop Hive • IBM PDA (Netezza) • Oracle • PostgreSQL • Presto • SybaseIQ • Teradata • Vertica

		For other data sources, consider either extracting the data or using WINDOW_CORR. See **Table Calculation Functions** on page 2219. **Note**: The square of a CORR result is equivalent to the R-Squared value for a linear trend line model. See **Trend Line Model Terms** on page 2537. Example: You can use CORR to visualize correlation in a disaggregated scatter plot. The way to do this is to use a table-scoped level of detail expression. For example: `{CORR(Sales, Profit)}` With a level of detail expression, the correlation is run over all rows. If you used a formula like `CORR(Sales, Profit)` (without the surrounding brackets to make it a level of detail expression), the view would show the correlation of each individual point in the scatter plot with each other point, which is undefined. See **Table-Scoped** on page 2461
COUNT	`COUNT (expression)`	Returns the number of items in a group. Null values are not counted.
COUNTD	`COUNTD (expression)`	Returns the number of distinct items in a group. Null values are not counted. This function is not available in the following cases: workbooks created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, and workbooks that use Microsoft Access data sources. Extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.
COVAR	`COVAR`	Returns the *sample covariance* of two expressions.

	`(expression 1, expression2)`	Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (available with the COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population. COVAR is available with the following data sources: • Tableau data extracts (you can create an extract from any data source) • Cloudera Hive • EXASolution • Firebird (version 3.0 and later) • Google BigQuery • Hortonworks Hadoop Hive • IBM PDA (Netezza) • Oracle • PostgreSQL • Presto • SybaseIQ • Teradata • Vertica For other data sources, consider either extracting the data or using WINDOW_COVAR. See **Table Calculation Functions** on page 2219. If expression1 and expression2 are the same—for example, COVAR([profit], [profit])—COVAR returns a value that indicates how widely values are distributed.

		Note: The value of COVAR(X, X) is equivalent to the value of VAR(X) and also to the value of STDEV(X)^2. Example: The following formula returns the sample covariance of **Sales** and **Profit**. `COVAR([Sales], [Profit])`
COVARP	`COVARP (expression 1, expression2)`	Returns the *population covariance* of two expressions. Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the COVAR function) is appropriate. COVARP is available with the following data sources: • Tableau data extracts (you can create an extract from any data source) • Cloudera Hive • EXASolution • Firebird (version 3.0 and later) • Google BigQuery • Hortonworks Hadoop Hive • IBM PDA (Netezza) • Oracle • PostgreSQL

<table>
<tr><td></td><td></td><td>

Presto
SybaseIQ
Teradata
Vertica

For other data sources, consider either extracting the data or using WINDOW_COVARP. See Table Calculation Functions on page 2219.
If expression1 and expression2 are the same—for example, COVARP([profit], [profit])—COVARP returns a value that indicates how widely values are distributed.
Note: The value of COVARP(X, X) is equivalent to the value of VARP(X) and also to the value of STDEVP(X)^2.
Example:
The following formula returns the population covariance of Sales and Profit.
<pre><code>COVARP([Sales], [Profit])</code></pre>
</td></tr>
<tr><td>MAX</td><td><code>MAX(expression)</code></td><td>Returns the maximum of an expression across all records. If the expression is a string value, this function returns the last value where last is defined by alphabetical order.</td></tr>
<tr><td>MEDIAN</td><td><code>MEDIAN(expression)</code></td><td>
Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. This function is not available for workbooks created before Tableau Desktop 8.2 or that use legacy connections. It is also not available for connections using any of the following data sources:

Access
Amazon Redshift
Cloudera Hadoop

</td></tr>
</table>

		- HP Vertica - IBM DB2 - IBM PDA (Netezza) - Microsoft SQL Server - MySQL - SAP HANA - Teradata For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.
MIN	`MIN (expression)`	Returns the minimum of an expression across all records. If the expression is a string value, this function returns the first value where first is defined by alphabetical order.
PERCENTILE	`PERCENTILE (expression, number)`	Returns the percentile value from the given expression corresponding to the specified number. The number must be between 0 and 1 (inclusive)—for example, 0.66, and must be a numeric constant. This function is available for the following data sources. - Non-legacy Microsoft Excel and Text File connections. - Extracts and extract-only data source types (for example, Google Analytics, OData, or Salesforce). - Sybase IQ 15.1 and later data sources. - Oracle 10 and later data sources. - Cloudera Hive and Hortonworks Hadoop Hive data sources. - EXASolution 4.2 and later data sources. For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.

STDEV	`STDEV (expression)`	Returns the statistical standard deviation of all values in the given expression based on a sample of the population.
STDEVP	`STDEVP (expression)`	Returns the statistical standard deviation of all values in the given expression based on a biased population.
SUM	`SUM (expression)`	Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored.
VAR	`VAR (expression)`	Returns the statistical variance of all values in the given expression based on a sample of the population.
VARP	`VARP (expression)`	Returns the statistical variance of all values in the given expression on the entire population.

Create an aggregate calculation

Follow along with the steps below to learn how to create an aggregate calculation.

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet and select **Analysis** > **Create Calculated Field**.
3. In the calculation editor that opens, do the following:
   - Name the calculated field **Margin**.
   - Enter the following formula:

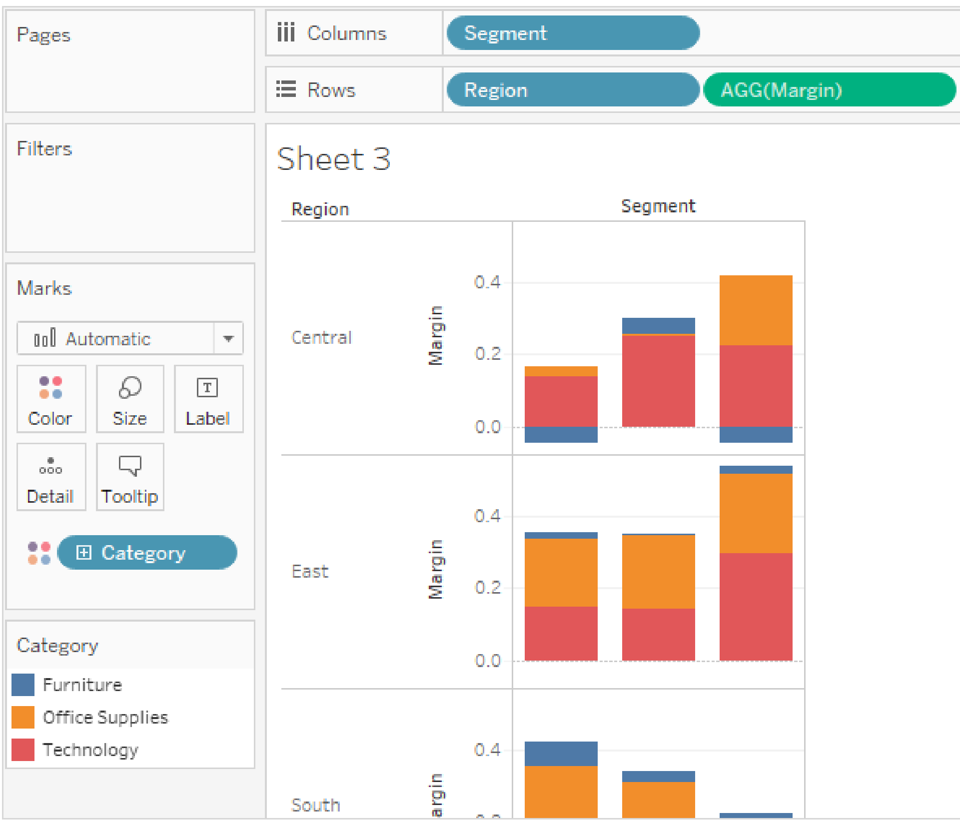
     ```
     IIF(SUM([Sales]) !=0, SUM([Profit])/SUM([Sales]),
     0)
     ```

     Note: You can use the function reference to find and add aggregate functions and other functions (like the logical IIF function in this example) to the calculation formula. For more information, see **Use the functions reference in the calculation editor** on page 2495.
   - When finished, click **OK**.

The new aggregate calculation appears under **Measures** in the **Data** pane. Just like your other fields, you can use it in one or more visualizations.

> **Note**: Aggregation calculations are always measures.

When **Margin** is placed on a shelf or card in the worksheet, its name is changed to **AGG (Margin)**, which indicates that it is an aggregate calculation and cannot be aggregated any further.



Rules for aggregate calculations

The rules that apply to aggregate calculations are as follows:

- For any aggregate calculation, you cannot combine an aggregated value and a disaggregated value. For example, SUM(Price)*[Items] is not a valid expression because SUM(Price) is aggregated and Items is not. However, SUM(Price*Items) and SUM (Price)*SUM(Items) are both valid.
- Constant terms in an expression act as aggregated or disaggregated values as

appropriate. For example: SUM(Price*7) and SUM(Price)*7 are both valid expressions.

- All of the functions can be evaluated on aggregated values. However, the arguments to any given function must either all be aggregated or all disaggregated. For example: MAX(SUM(Sales),Profit) is not a valid expression because Sales is aggregated and Profit is not. However, MAX(SUM(Sales),SUM(Profit)) is a valid expression.
- The result of an aggregate calculation is always a measure.
- Like predefined aggregations, aggregate calculations are computed correctly for grand totals. Refer to Grand Totals for more information.

See also

Understanding Calculations: Aggregate Calculations

Data Aggregation in Tableau on page 341

Functions in Tableau on page 2147

Tableau Functions (by Category) on page 2267

Tableau Functions (Alphabetical) on page 2342

Pass-Through Functions (RAWSQL)

These RAWSQL pass-through functions can be used to send SQL expressions directly to the database, without first being interpreted by Tableau. If you have custom database functions that Tableau doesn't know about, you can use these pass-through functions to call these custom functions.

Your database usually will not understand the field names that are shown in Tableau. Because Tableau does not interpret the SQL expressions you include in the pass-through functions, using the Tableau field names in your expression may cause errors. You can use a substitution syntax to insert the correct field name or expression for a Tableau calculation into pass-through SQL. For example, if you had a function that computed the median of a set of values, you could call that function on the Tableau column [Sales] like this:

```
RAWSQLAGG_REAL(“MEDIAN(%1)”, [Sales])
```

Because Tableau does not interpret the expression, you must define the aggregation. You can use the RAWSQLAGG functions described below when you are using aggregated expressions.

RAWSQL pass-through functions will not work with published data sources or with Tableau extracts.

These functions may return different results starting in Tableau Desktop 8.2 than they did in earlier versions of Tableau Desktop. This is because Tableau now uses ODBC for pass-through functions instead of OLE DB. ODBC truncates when returning real values as integer; OLE DB rounds when returning real values as integer.

RAWSQL Functions

The following RAWSQL functions are available in Tableau.

RAWSQL_BOOL(“sql_expr”, [arg1], …[argN])

Returns a Boolean result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

```
RAWSQL_BOOL("%1 > %2", [Sales], [Profit])
```

RAWSQL_DATE(“sql_expr”, [arg1], …[argN])

Returns a Date result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Order Date].

```
RAWSQL_DATE(“%1”, [Order Date])
```

RAWSQL_DATETIME(“sql_expr”, [arg1], …[argN])

Returns a Date and Time result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

```
RAWSQL_DATETIME("%1", [Order Date])
```

RAWSQL_INT(“sql_expr”, [arg1], …[argN])

Returns an integer result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

```
RAWSQL_INT(“500 + %1”, [Sales])
```

RAWSQL_REAL(“sql_expr”, [arg1], …[argN])

Returns a numeric result from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

```
RAWSQL_REAL(“-123.98 * %1”, [Sales])
```

RAWSQL_SPATIAL

Returns a Spatial from a given SQL expression that is passed directly to the underlying data source. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Geometry].

```
RAWSQL_SPATIAL("%1", [Geometry])
```

RAWSQL_STR("sql_expr", [arg1], …[argN])

Returns a string from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Customer Name].

Example

```
RAWSQL_STR("%1", [Customer Name])
```

RAWSQLAGG_BOOL("sql_expr", [arg1], …[argN])

Returns a Boolean result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQLAGG_BOOL("SUM(%1) >SUM(%2)", [Sales], [Profit])

RAWSQLAGG_DATE("sql_expr", [arg1], …[argN])

Returns a Date result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Order Date].

Example

```
RAWSQLAGG_DATE("MAX(%1)", [Order Date])
```

RAWSQLAGG_DATETIME("sql_expr", [arg1], …[argN])

Returns a Date and Time result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

```
RAWSQLAGG_DATETIME("MIN(%1)", [Delivery Date])
```

RAWSQLAGG_INT(“sql_expr”, [arg1,] …[argN])

Returns an integer result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

```
RAWSQLAGG_INT(“500 + SUM(%1)”, [Sales])
```

RAWSQLAGG_REAL(“sql_expr”, [arg1,] …[argN])

Returns a numeric result from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

```
RAWSQLAGG_REAL(“SUM( %1)”, [Sales])
```

RAWSQLAGG_STR(“sql_expr”, [arg1,] …[argN])

Returns a string from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Discount].

Example

```
RAWSQLAGG_STR(“AVG(%1)”, [Discount])
```

User Functions

This article introduces user functions and their uses in Tableau. It also demonstrates how to create a user calculation using an example.

Why use user functions

User functions can be used to create user filters or row-level security filters that affect visualizations published to Tableau Server or Tableau Online, so that only certain people can see your visualization.

For example, if you have a visualization that shows the sales performance for each employee in your department published on Tableau Server or Tableau Online, you might want to only allow employees to see their own sales numbers when they access that visualization.

In this case, you can use the ISMEMBEROF function to create a field that returns true if the username of the person signed in to the server is a member of a specified group (on the server), such as the "Managers" group, for example. Then when you filter the view using this calculated field, only a person who is part of that group can see the data.

The calculation in this case might look something like the following:

```
ISMEMBEROF('Managers')
```

> **Note**: If your group or user names contain certain non-alphanumeric you must use HTML URL Encoding for the special characters when using the functions below.

Some special characters are permitted without HTML URL encoding, such as underscores, parentheses, and exclamation points. `_ ( ) !` Many other characters must be encoded.

For example, the function `ISMEMBEROF("USERS+")` needs to be written as `ISMEMBEROF ("USERS%2B")`, because the '%2B' is the HTML URL Encoding for the '+' symbol. For information on HTML URL encoding, please see HTML URL Encoding Reference at the W3schools Web Developer site.

User functions available in Tableau:

Function	Syntax	Description
FULLNAME	`FULLNAME( )`	Returns the full name for the current user. This is the Tableau Server or Tableau Online full name when the user is signed in; otherwise the local or network full name for the Tableau Desktop user. Example: `[Manager]=FULLNAME( )` If manager Dave Hallsten is signed in, this example returns True only if the Manager field in the view contained Dave Hallsten. When used as a filter, this calculated field

		can be used to create a user filter that only shows data that is relevant to the person signed in to the server.
ISFULLNAME	`ISFULLNAME (string)`	Returns true if the current user's full name matches the specified full name, or false if it does not match. This function uses the Tableau Server or Online full name when the user is signed in; otherwise it uses the local or network full name for the Tableau Desktop user. Example: `ISFULLNAME("Dave Hallsten")` This example returns true if Dave Hallsten is the current user, otherwise it returns false.

ISMEMBEROF	`ISMEMBEROF (string)`	Returns true if the person currently using Tableau is a member of a group that matches the given string. If the person currently using Tableau is signed in, the group membership is determined by groups on Tableau Server or Tableau Online. If the person is not signed in, this function returns NULL.
		Note: The function will return a "True" value if the given string is "All Users", whether signed in to Tableau Server or Tableau Online.
		The ISMEMBEROF() function will also accept Active Directory domains. The Active Directory domain must be declared in the calculation with the group name.
		Example:
		`IF ISMEMBEROF('domain.lan\Sales')`
		`THEN "Sales" ELSE "Other" END`

ISUSERNAME	`ISUSERNAME (string)`	Returns true if the current user's username matches the specified username, or false if it does not match. This

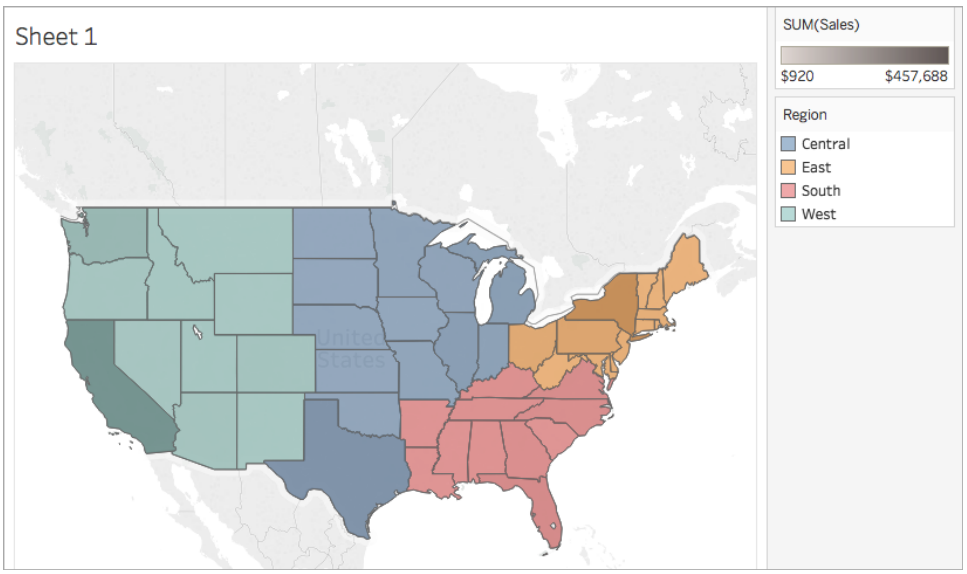
		function uses the Tableau Server or Online username when the user is signed in; otherwise it uses the local or network username for the Tableau Desktop user. Example: `ISUSERNAME("dhallsten")` This example returns true if dhallsten is the current user; otherwise it returns false. Note: "All Users" will always return as true.
USERDOMAIN	`USERDOMAIN()`	Returns the domain for the current user when the user is signed on to Tableau Server. Returns the Windows domain if the Tableau Desktop user is on a domain. Otherwise this function returns a null string. Example: `[Manager]=USERNAME() AND [Domain]=USERDOMAIN()`
USERNAME	`USERNAME()`	Returns the username for the current user. This is the Tableau Server or Tableau Online username when the user is signed in; otherwise it is the local or network username for the Tableau Desktop user. Example: `[Manager]=USERNAME( )` If the manager dhallsten was signed in, this function would only return True when the Manager field in the view is dhallsten. When used as a filter this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.

Create a user calculation

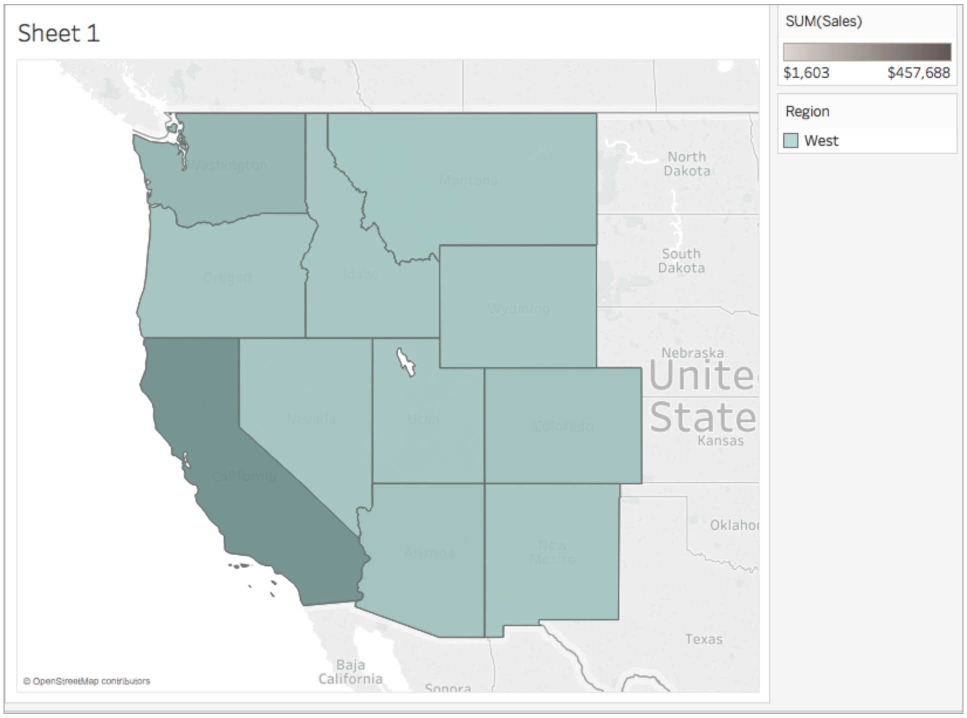
User calculations work directly with the users and groups you have set up on Tableau Server or Tableau Online. You can create user calculations to use as filters so users only see the data that is relevant to them.

For example, if you have a map viz similar to the following, which shows sales data for 48 US states, you can create a user calculation to show only parts of the map that are relevant to each user, such as data relevant for a regional manager versus a national manager. (A national manager should be able to see data for the entire country, while a regional manager should only be able to see data for the region they manage).

When the national manger is signed in, they see the following visualization:



When the western regional manager is signed in, they see only sales for their region:



To create a user function that performs similarly to this example, follow the steps below.

Before you begin

To follow along with this example you must have access to Tableau Server or Online. You must also be a Server or Site Administrator.

Step 1: Create the users and groups

1. Sign in to Tableau Server or Tableau Online.
2. In Tableau Server or Tableau Online, add the following users:
   - Sadie Pawthorne
   - Chuck Magee
   - Fred Suzuki
   - Roxanne Rodriguez

   For more information, see Add Users to a Site in the Tableau Server Help.
3. Create a new group called **National Managers**.

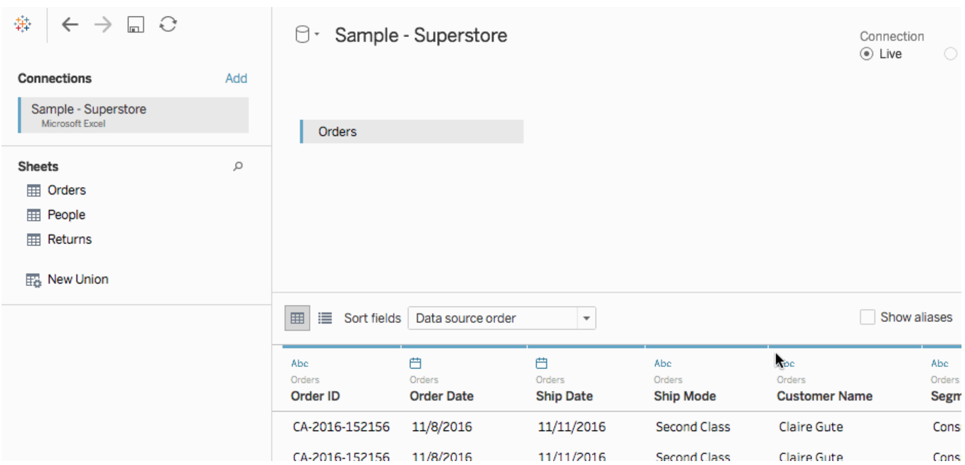
For more information, see Create a Local Group in the Tableau Server Help.

4. Add yourself to the National Managers group.

   For more information, see Add Users to a Group in the Tableau Server Help.

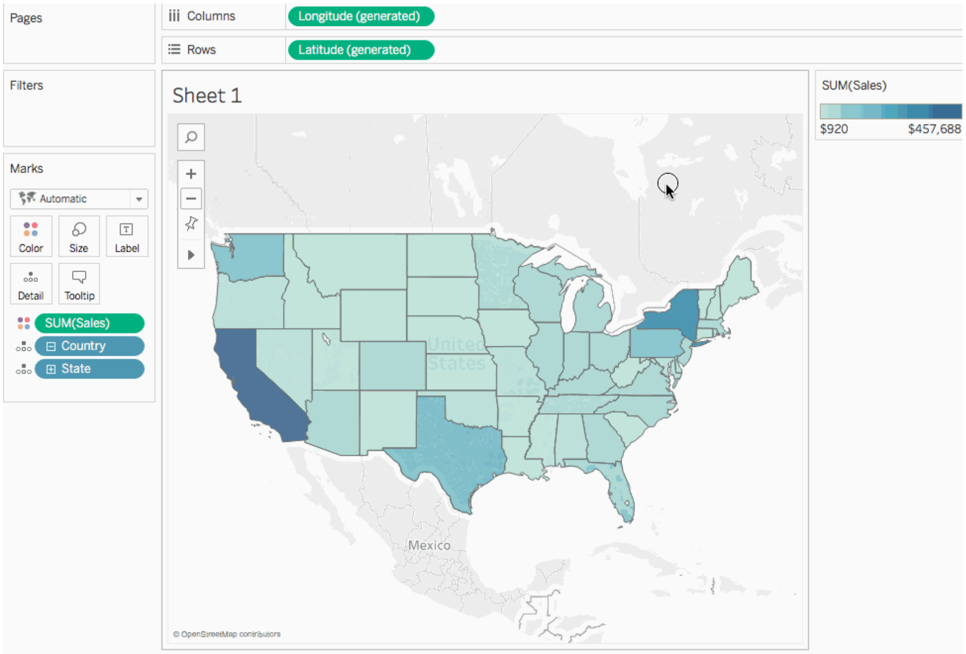
Step 2: Create the visualization

1. Open Tableau Desktop, and connect to the **Sample-Superstore** data source, which comes with Tableau.
2. In the bottom left corner of the workspace, click the Data Source tab.
3. On the Data Source page, from the Connections pane on the left, drag the People sheet to the join area.
4. Click the join icon and select **Left**.



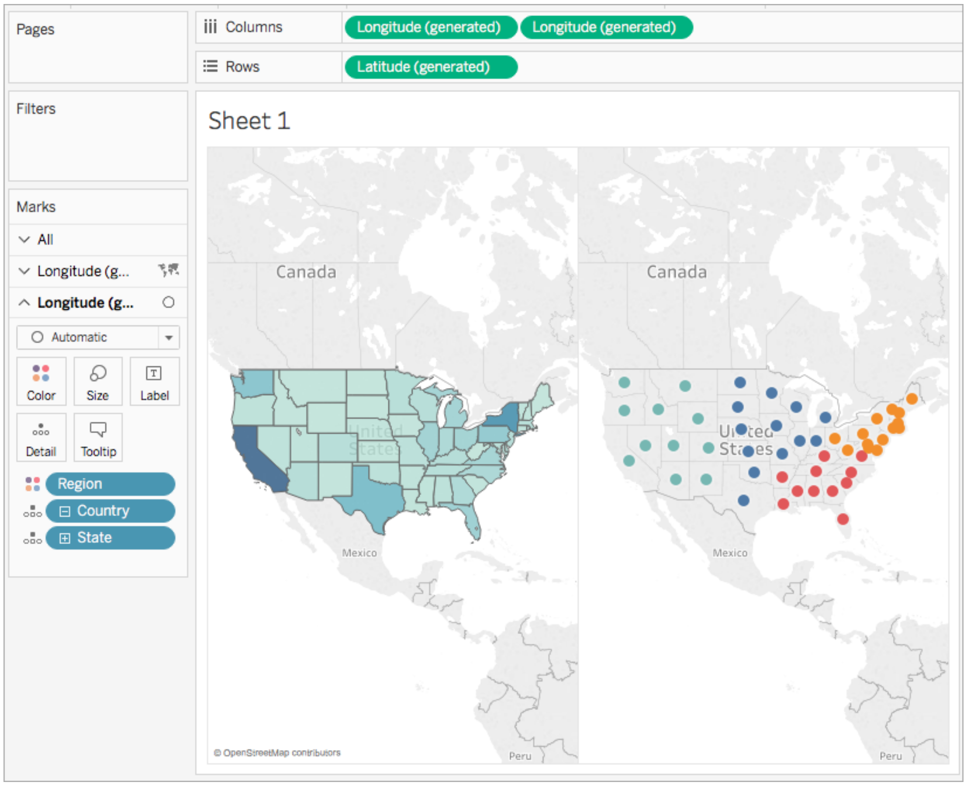
5. Navigate to a new worksheet.
6. In the **Data** pane, under Dimensions, double-click **State**.

   A map view is created.
7. From the **Data** pane, under Measures, drag **Sales** to **Color** on the Marks card.
8. On the Columns shelf, select the Longitude field and hold down Control (Comman on Mac) on your keyboard to copy it. Drag the copy to the right of the original on the Columns shelf.

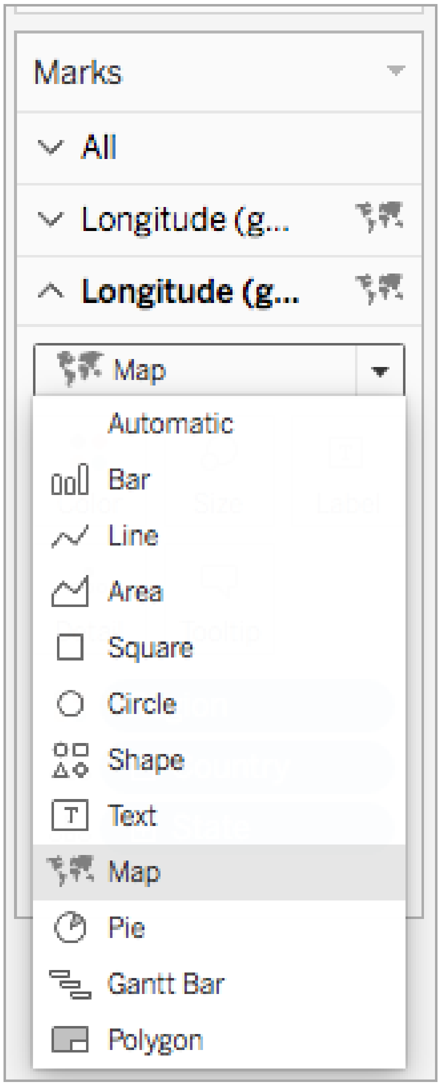


9. On the Marks card, click the second (bottom) **Longitude** tab.
10. From the **Data** pane, drag **Region** to **Color** on the Marks card.

    The map view on the right updates with new colors.



11. On the Marks card, click the mark type drop-down and select **Map**.

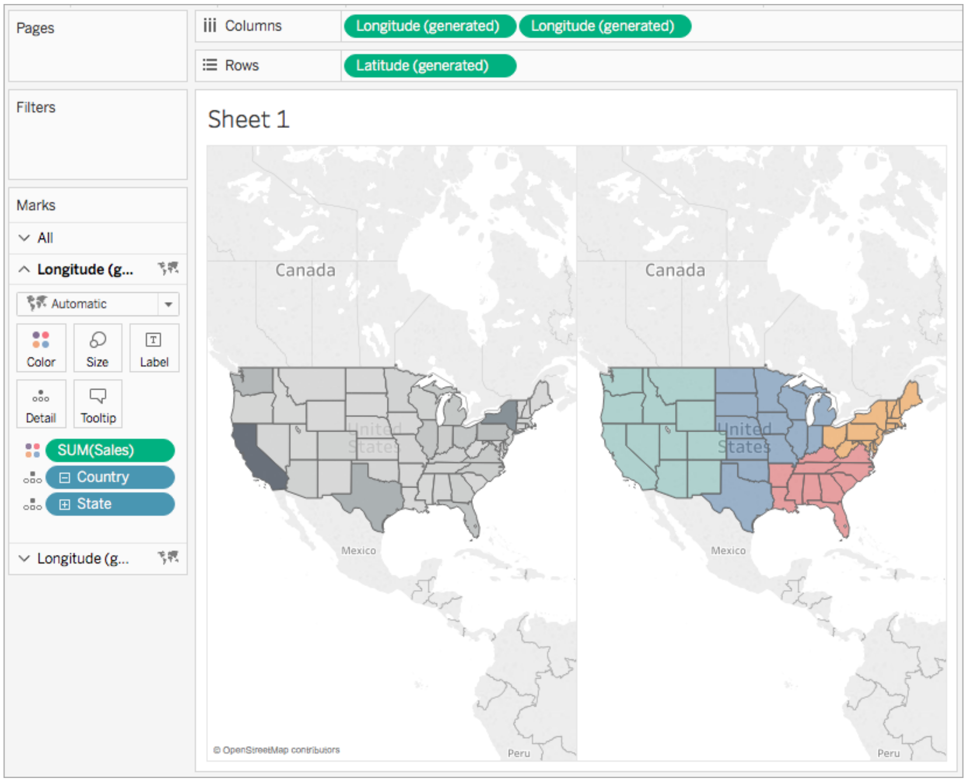


12. On the Marks card, click **Color**, and, under Opacity, adjust the slider to **50%**.

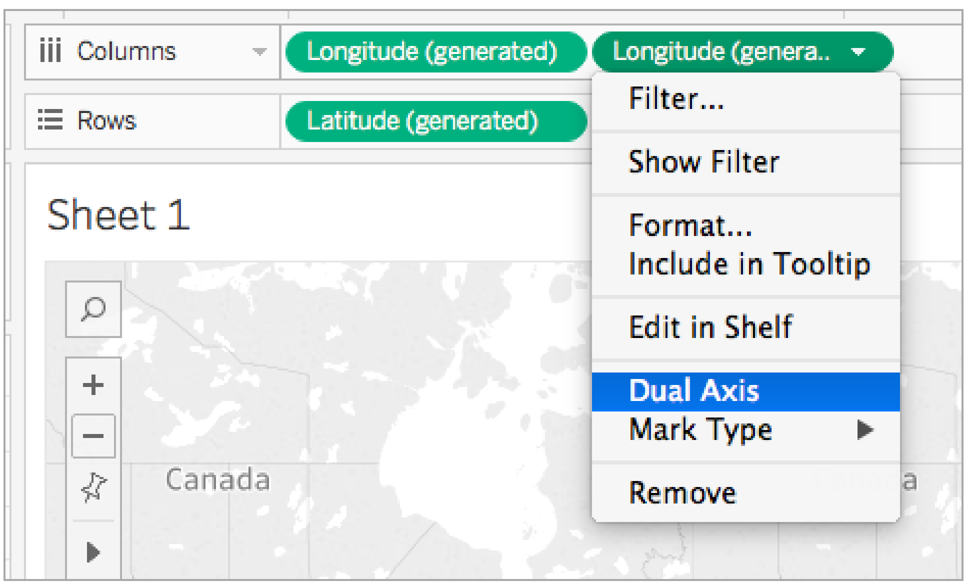
13. On the Marks card, click the first **Longitude** tab.

14. On the Marks card, click **Color** > **Edit Colors**, and then select **Gray** from the color palette drop-down list.

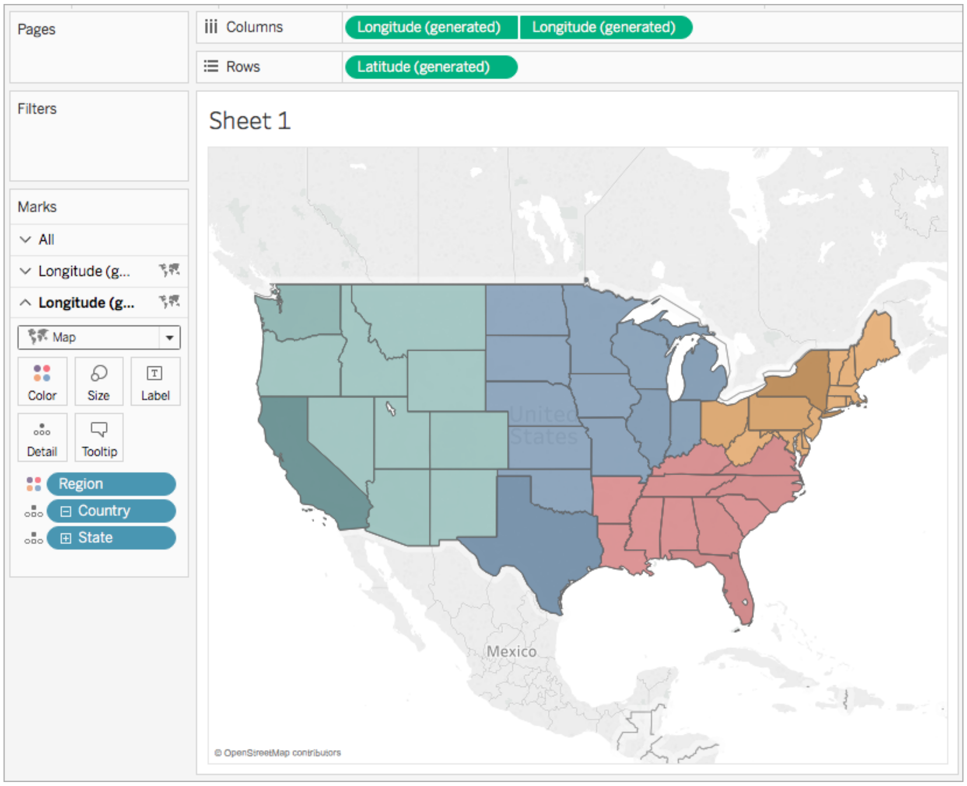
    The map view on the left updates.



15. On the Columns shelf, right-click the **Longitude** field on the right and select **Dual Axis**.



The map looks like the following:



Step 3: Create the User Calculation

1. Select **Analysis** > **Create Calculated Field**.
2. In the calculation editor that opens, do the following:
   - Name the calculated field, User Filter.
   - Enter the following formula:

     ```
     [Regional Manager] = USERNAME() OR ISMEMBEROF
     ("National Managers")
     ```

     This calculation checks if a person is included in the Region (People) field, or if a person is included in the National Managers group. If so, it returns true.
   - When finished, click **OK**.

   The new user calculation appears under Dimensions in the Data pane. Just like your other fields, you can use it in one or more visualizations.

Step 4: Add the user calculation to the Filters shelf

1. From the Data pane, under Dimensions, drag User Filter to the Filters shelf.
2. In the Filter dialog box that opens, select **True**, and then click **OK**.

   Note: If you are not signed in to Tableau Server or Tableau Online, the True option is not visible. In Tableau Desktop, sign in to Tableau Server or Tableau Online to select it. See **Sign in to Tableau Server or Tableau Online** on page 364 for more information.

Step 5: Test the calculation

1. In Tableau Desktop, in the bottom-right corner of the workspace, click the Filter as User drop-down and change the user to **Sadie Pawthorne**.

   The map updates to show only the West region of the United States because Sadie is assigned to the West region in the People sheet.
2. Select the Filter as User drop-down again and change the user to Roxanne Rodriguez.

   The map updates to show only the Central region of the United States because Roxanne is assigned to the Central region in the People sheet.
3. Select the Filter as User drop-down again and change the user to Chuck Magee.

   The map updates to show only the East region of the United States because Chuck is assigned to the East region in the People sheet.
4. Select the Filter as User drop-down again and change the user to Fred Suzuki.

   The map updates to show only the South region of the United States because Fred is assigned to the South region in the People sheet.
5. Select the Filter as User drop-down one more time and change the user back to yourself.

   The map updates to show all data because you are a part of the National Managers group on the server.

This behavior persists when you publish the view to Tableau Server or Tableau Online. Users not listed in the National Managers group, or in the People sheet in the Sample Superstore data source see only a blank visualization.

See Also

User Calculations free training video

Functions in Tableau on page 2147

Tableau Functions (by Category) on page 2267

Tableau Functions (Alphabetical) on page 2342

Table Calculation Functions

This article introduces table calculation functions and their uses in Tableau. It also demonstrates how to create a table calculation using the calculation editor.

Why use table calculation functions

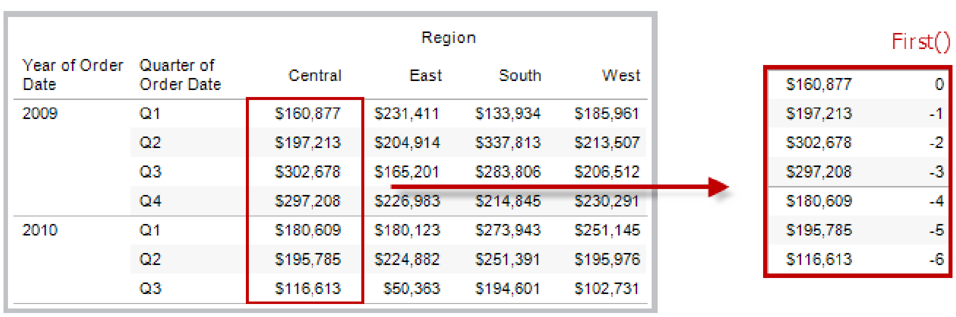
Table calculation functions allow you to perform computations on values in a table.

For example, you can calculate the percent of total an individual sale is for the year, or for several years.

Table calculation functions available in Tableau

FIRST( )

Returns the number of rows from the current row to the first row in the partition. For example, the view below shows quarterly sales. When FIRST() is computed within the Date partition, the offset of the first row from the second row is -1.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

First()

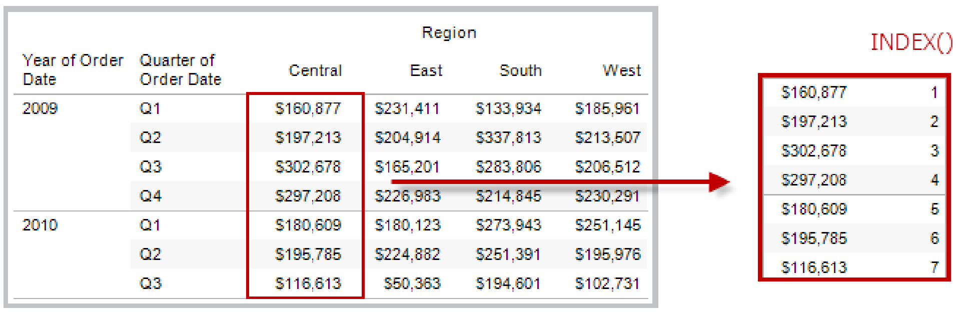
$160,877	0
$197,213	-1
$302,678	-2
$297,208	-3
$180,609	-4
$195,785	-5
$116,613	-6

Example

When the current row index is 3, `FIRST() = -2`.

INDEX( )

Returns the index of the current row in the partition, without any sorting with regard to value. The first row index starts at 1. For example, the table below shows quarterly sales. When INDEX() is computed within the Date partition, the index of each row is 1, 2, 3, 4..., etc.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

INDEX()

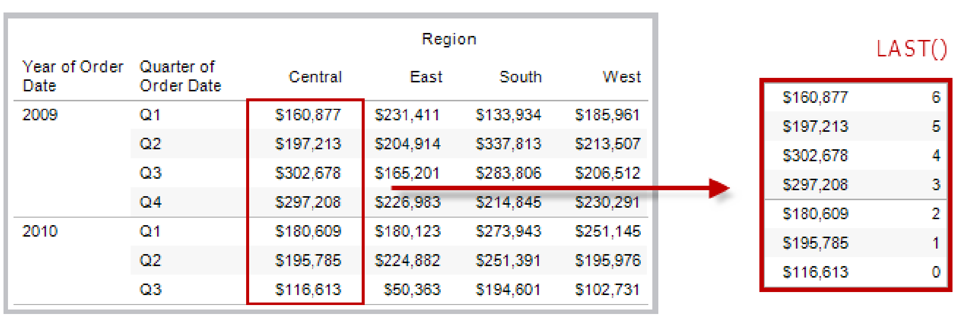
$160,877	1
$197,213	2
$302,678	3
$297,208	4
$180,609	5
$195,785	6
$116,613	7

Example

For the third row in the partition, `INDEX() = 3`.

LAST( )

Returns the number of rows from the current row to the last row in the partition. For example, the table below shows quarterly sales. When LAST() is computed within the Date partition, the offset of the last row from the second row is 5.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

LAST()

$160,877	6
$197,213	5
$302,678	4
$297,208	3
$180,609	2
$195,785	1
$116,613	0

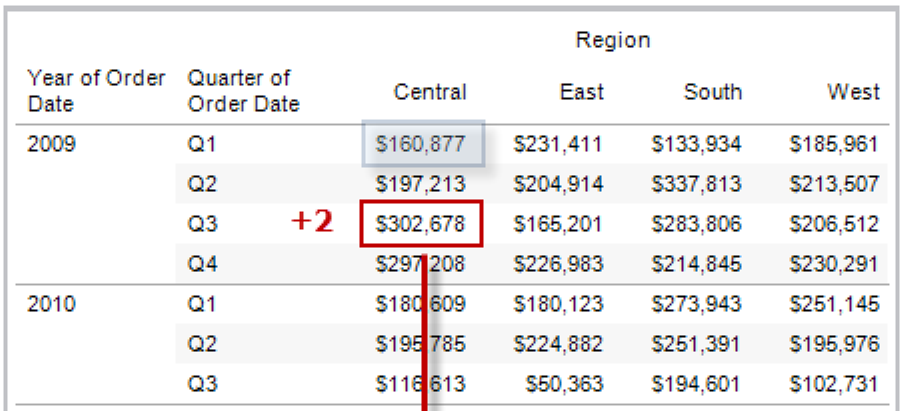
Example

When the current row index is 3 of 7, `LAST() = 4`.

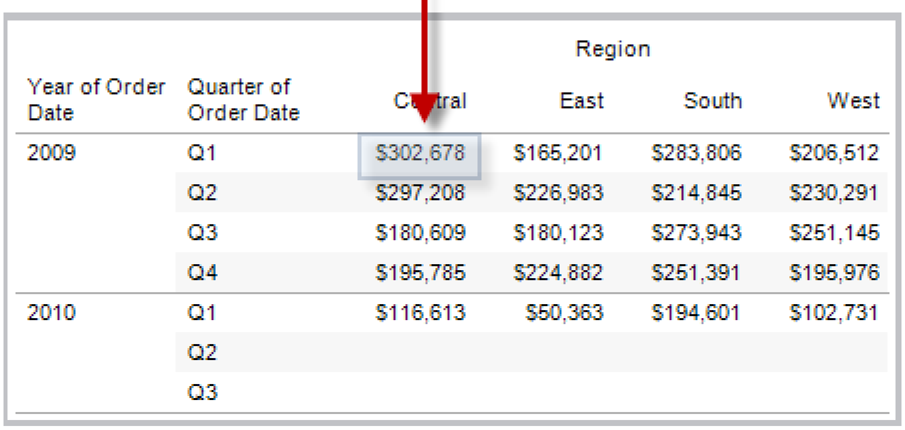
LOOKUP(expression, [offset])

Returns the value of the expression in a target row, specified as a relative offset from the current row. Use FIRST() + n and LAST() - n as part of your offset definition for a target relative to the first/last rows in the partition. If `offset` is omitted, the row to compare to can be set on the field menu. This function returns NULL if the target row cannot be determined.

The view below shows quarterly sales. When `LOOKUP (SUM(Sales), 2)` is computed within the Date partition, each row shows the sales value from 2 quarters into the future.



Year of Order Date	Quarter of Order Date		Region: Central	East	South	West
2009	Q1		$160,877	$231,411	$133,934	$185,961
	Q2		$197,213	$204,914	$337,813	$213,507
	Q3	+2	$302,678	$165,201	$283,806	$206,512
	Q4		$297,208	$226,983	$214,845	$230,291
2010	Q1		$180,609	$180,123	$273,943	$251,145
	Q2		$195,785	$224,882	$251,391	$195,976
	Q3		$116,613	$50,363	$194,601	$102,731



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$302,678	$165,201	$283,806	$206,512
	Q2	$297,208	$226,983	$214,845	$230,291
	Q3	$180,609	$180,123	$273,943	$251,145
	Q4	$195,785	$224,882	$251,391	$195,976
2010	Q1	$116,613	$50,363	$194,601	$102,731
	Q2				
	Q3				

Example

`LOOKUP(SUM([Profit]), FIRST()+2)` computes the SUM(Profit) in the third row of the partition.

MODEL_PERCENTILE(target_expression, predictor_expression(s))

Returns the probability (between 0 and 1) of the expected value being less than or equal to the observed mark, defined by the target expression and other predictors. This is the Posterior Predictive Distribution Function, also known as the Cumulative Distribution Function (CDF).

This function is the inverse of MODEL_QUANTILE. For information on predictive modeling functions, see **How Predictive Modeling Functions Work in Tableau** on page 2606.

Example

The following formula returns the quantile of the mark for sum of sales, adjusted for count of orders.

```
MODEL_PERCENTILE(SUM([Sales]), COUNT([Orders]))
```

MODEL_QUANTILE(quantile, target_expression, predictor_expression(s))

Returns a target numeric value within the probable range defined by the target expression and other predictors, at a specified quantile. This is the Posterior Predictive Quantile.

This function is the inverse of MODEL_PERCENTILE. For information on predictive modeling functions, see **How Predictive Modeling Functions Work in Tableau** on page 2606.

Example

The following formula returns the median (0.5) predicted sum of sales, adjusted for count of orders.

```
MODEL_QUANTILE(0.5, SUM([Sales]), COUNT([Orders]))
```

PREVIOUS_VALUE(expression)

Returns the value of this calculation in the previous row. Returns the given expression if the current row is the first row of the partition.

Example

`SUM([Profit]) * PREVIOUS_VALUE(1)` computes the running product of SUM(Profit).

RANK(expression, ['asc' | 'desc'])

Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

Example

The following image shows the effect of the various ranking functions (RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, and RANK_UNIQUE) on a set of values. The data set contains information on 14 students (StudentA through StudentN); the **Age** column shows the current age of each student (all students are between 17 and 20 years of age). The remaining columns show the effect of each rank function on the set of age values, always assuming the default order (ascending or descending) for the function.

Student	Age	RANKofAge	RANK_DENSEofAge	RANK_MODIFIEDofAge	RANK_PERCENTILEofAge	RANK_UNIQUEofAge
StudentA	19	4	2	7	79%	4
StudentB	18	8	3	12	50%	8
StudentC	19	4	2	7	79%	5
StudentD	18	8	3	12	50%	9
StudentE	17	13	4	14	14%	13
StudentF	18	8	3	12	50%	10
StudentG	19	4	2	7	79%	6
StudentH	20	1	1	3	100%	1
StudentI	19	4	2	7	79%	7
StudentJ	20	1	1	3	100%	2
StudentK	20	1	1	3	100%	3
StudentL	17	13	4	14	14%	14
StudentM	18	8	3	12	50%	11
StudentN	18	8	3	12	50%	12

RANK_DENSE(expression, ['asc' | 'desc'])

Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (3, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_MODIFIED(expression, ['asc' | 'desc'])

Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 3, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_PERCENTILE(expression, ['asc' | 'desc'])

Returns the percentile rank for the current row in the partition. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is ascending.

With this function, the set of values (6, 9, 9, 14) would be ranked (0.00, 0.67, 0.67, 1.00).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_UNIQUE(expression, ['asc' | 'desc'])

Returns the unique rank for the current row in the partition. Identical values are assigned different ranks. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 3, 1).

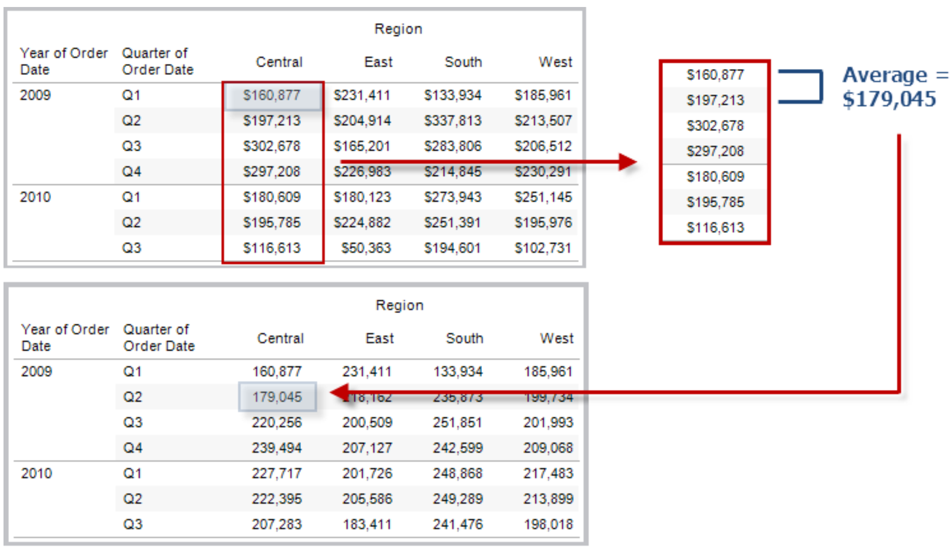
Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RUNNING_AVG(expression)

Returns the running average of the given expression, from the first row in the partition to the current row.

The view below shows quarterly sales. When `RUNNING_AVG(SUM([Sales])` is computed within the Date partition, the result is a running average of the sales values for each quarter.



Example

`RUNNING_AVG(SUM([Profit]))` computes the running average of SUM(Profit).

RUNNING_COUNT(expression)

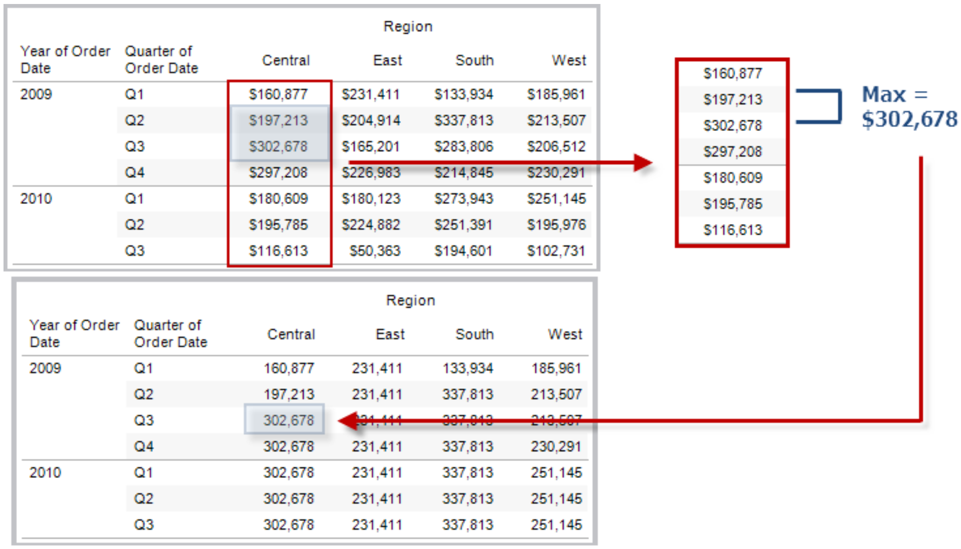
Returns the running count of the given expression, from the first row in the partition to the current row.

Example

`RUNNING_COUNT(SUM([Profit]))` computes the running count of SUM(Profit).

RUNNING_MAX(expression)

Returns the running maximum of the given expression, from the first row in the partition to the current row.

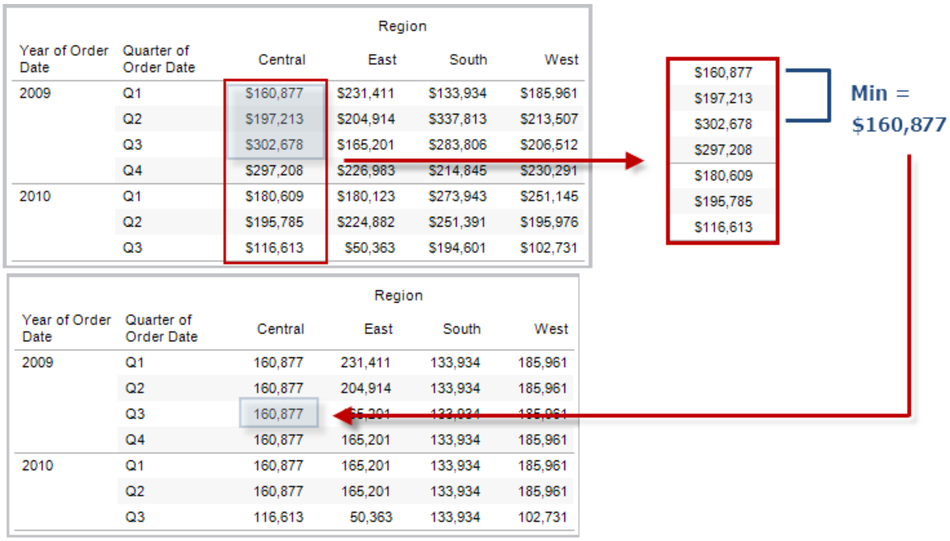


Example

`RUNNING_MAX(SUM([Profit]))` computes the running maximum of SUM(Profit).

RUNNING_MIN(expression)

Returns the running minimum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

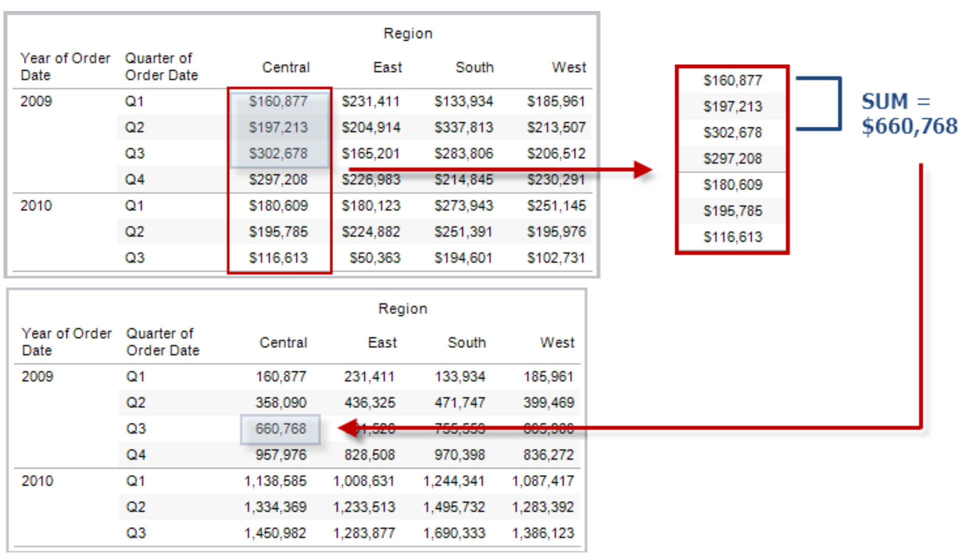
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	160,877	204,914	133,934	185,961
	Q3	160,877	165,201	133,934	185,961
	Q4	160,877	165,201	133,934	185,961
2010	Q1	160,877	165,201	133,934	185,961
	Q2	160,877	165,201	133,934	185,961
	Q3	116,613	50,363	133,934	102,731

Example

`RUNNING_MIN(SUM([Profit]))` computes the running minimum of SUM(Profit).

RUNNING_SUM(expression)

Returns the running sum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

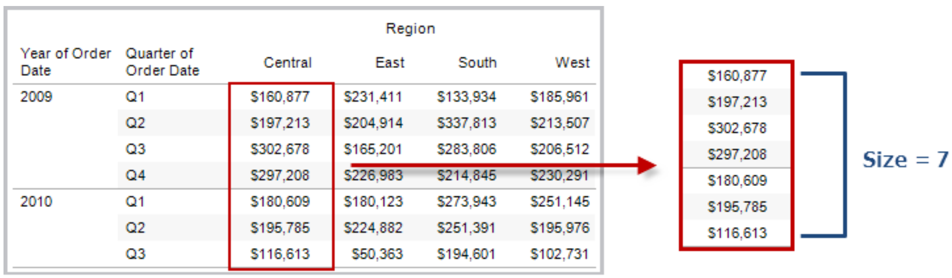
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	358,090	436,325	471,747	399,469
	Q3	660,768	601,526	755,553	605,981
	Q4	957,976	828,508	970,398	836,272
2010	Q1	1,138,585	1,008,631	1,244,341	1,087,417
	Q2	1,334,369	1,233,513	1,495,732	1,283,392
	Q3	1,450,982	1,283,877	1,690,333	1,386,123

Example

`RUNNING_SUM(SUM([Profit]))` computes the running sum of SUM(Profit)

SIZE()

Returns the number of rows in the partition. For example, the view below shows quarterly sales. Within the Date partition, there are seven rows so the Size() of the Date partition is 7.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Example

`SIZE() = 5` when the current partition contains five rows.

SCRIPT_BOOL

Returns a Boolean result from the specified expression. The expression is passed directly to a running analytics extension service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.).

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_BOOL("is.finite(.arg1)", SUM([Profit]))
```

The next example returns True for store IDs in Washington state, and False otherwise. This example could be the definition for a calculated field titled IsStoreInWA.

```
SCRIPT_BOOL('grepl(".*_WA", .arg1, perl=TRUE)',ATTR([Store ID]))
```

A command for Python would take this form:

```
SCRIPT_BOOL("return map(lambda x : x > 0, _arg1)", SUM([Profit]))
```

SCRIPT_INT

Returns an integer result from the specified expression. The expression is passed directly to a running analytics extension service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_INT("is.finite(.arg1)", SUM([Profit]))
```

In the next example, k-means clustering is used to create three clusters:

```
SCRIPT_INT('result <- kmeans(data.frame(.arg1,.arg2,.arg3,.arg4),
3);result$cluster;', SUM([Petal length]), SUM([Petal width]),SUM
([Sepal length]),SUM([Sepal width]))
```

A command for Python would take this form:

```
SCRIPT_INT("return map(lambda x : int(x * 5), _arg1)", SUM
([Profit]))
```

SCRIPT_REAL

Returns a real result from the specified expression. The expression is passed directly to a running analytics extension service instance. In

R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_REAL("is.finite(.arg1)", SUM([Profit]))
```

The next example converts temperature values from Celsius to Fahrenheit.

```
SCRIPT_REAL('library(udunits2);ud.convert(.arg1, "celsius",
"degree_fahrenheit")',AVG([Temperature]))
```

A command for Python would take this form:

```
SCRIPT_REAL("return map(lambda x : x * 0.5, _arg1)", SUM
([Profit]))
```

SCRIPT_STR

Returns a string result from the specified expression. The expression is passed directly to a running analytics extension service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_STR("is.finite(.arg1)", SUM([Profit]))
```

The next example extracts a state abbreviation from a more complicated string (in the original form 13XSL_CA, A13_WA):

```
SCRIPT_STR('gsub(".*_", "", .arg1)',ATTR([Store ID]))
```

A command for Python would take this form:

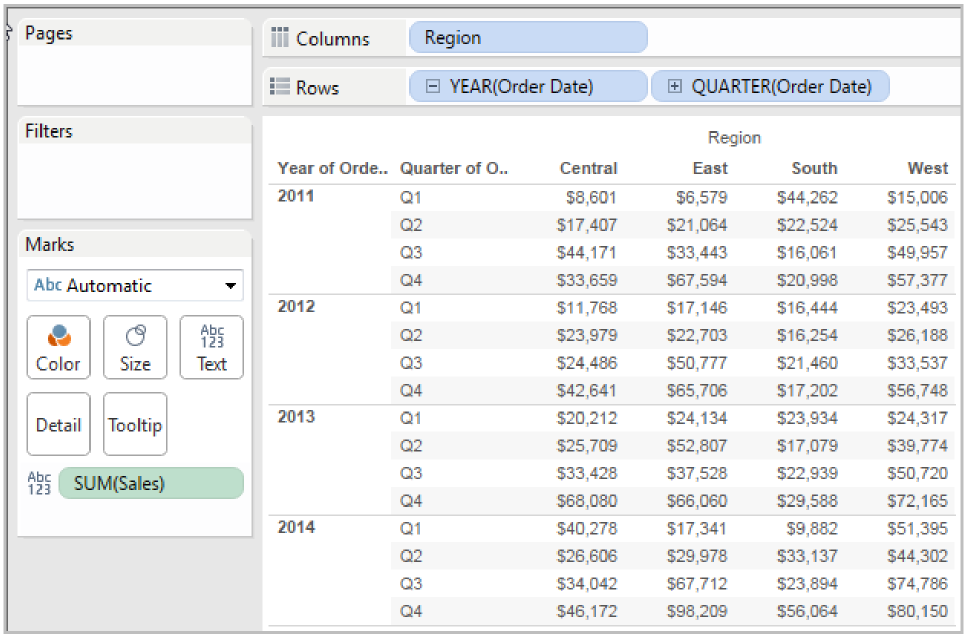
```
SCRIPT_STR("return map(lambda x : x[:2], _arg1)", ATTR([Region]))
```

TOTAL(expression)

Returns the total for the given expression in a table calculation partition.

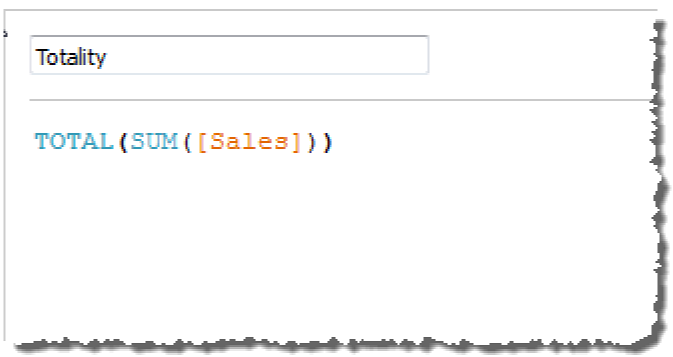
Example

Assume you are starting with this view:

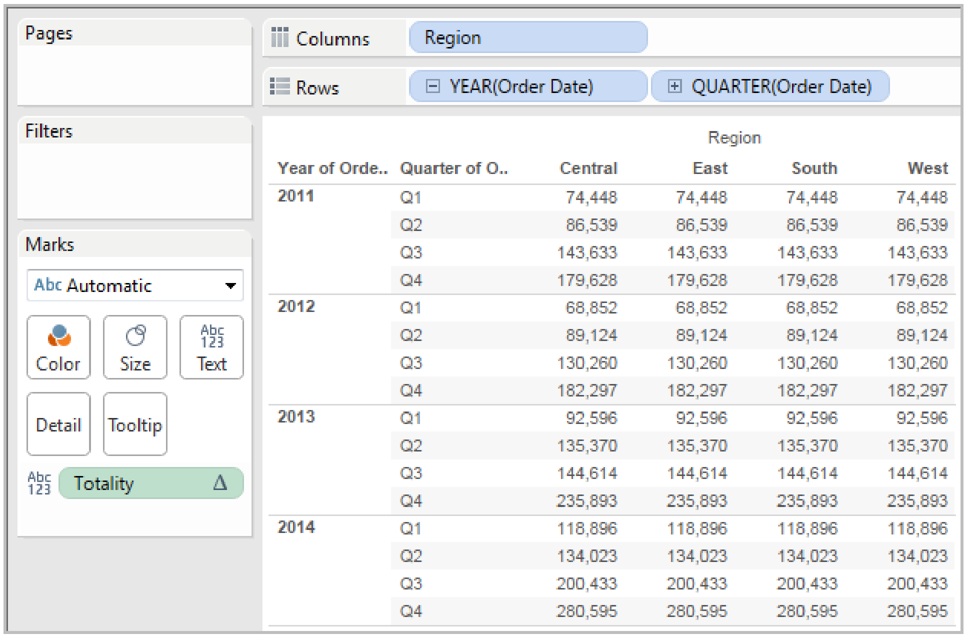


Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	$8,601	$6,579	$44,262	$15,006
	Q2	$17,407	$21,064	$22,524	$25,543
	Q3	$44,171	$33,443	$16,061	$49,957
	Q4	$33,659	$67,594	$20,998	$57,377
2012	Q1	$11,768	$17,146	$16,444	$23,493
	Q2	$23,979	$22,703	$16,254	$26,188
	Q3	$24,486	$50,777	$21,460	$33,537
	Q4	$42,641	$65,706	$17,202	$56,748
2013	Q1	$20,212	$24,134	$23,934	$24,317
	Q2	$25,709	$52,807	$17,079	$39,774
	Q3	$33,428	$37,528	$22,939	$50,720
	Q4	$68,080	$66,060	$29,588	$72,165
2014	Q1	$40,278	$17,341	$9,882	$51,395
	Q2	$26,606	$29,978	$33,137	$44,302
	Q3	$34,042	$67,712	$23,894	$74,786
	Q4	$46,172	$98,209	$56,064	$80,150

You open the calculation editor and create a new field which you name **Totality**:

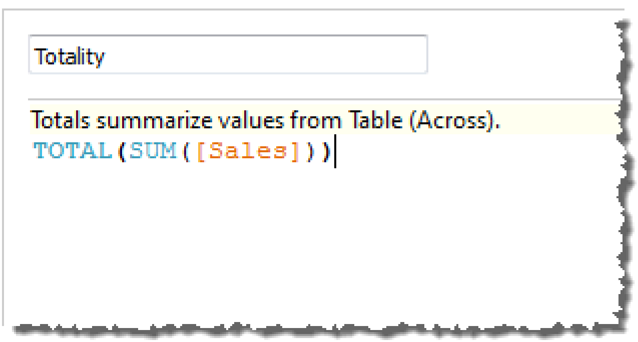


You then drop **Totality** on Text, to replace **SUM(Sales)**. Your view changes such that it sums values based on the default **Compute Using** value:



		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	74,448	74,448	74,448	74,448
	Q2	86,539	86,539	86,539	86,539
	Q3	143,633	143,633	143,633	143,633
	Q4	179,628	179,628	179,628	179,628
2012	Q1	68,852	68,852	68,852	68,852
	Q2	89,124	89,124	89,124	89,124
	Q3	130,260	130,260	130,260	130,260
	Q4	182,297	182,297	182,297	182,297
2013	Q1	92,596	92,596	92,596	92,596
	Q2	135,370	135,370	135,370	135,370
	Q3	144,614	144,614	144,614	144,614
	Q4	235,893	235,893	235,893	235,893
2014	Q1	118,896	118,896	118,896	118,896
	Q2	134,023	134,023	134,023	134,023
	Q3	200,433	200,433	200,433	200,433
	Q4	280,595	280,595	280,595	280,595

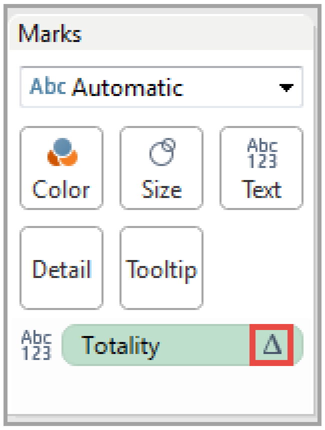
This raises the question, What is the default **Compute Using** value? If you right-click (Control-click on a Mac) **Totality** in the Data pane and choose **Edit**, there is now an additional bit of information available:



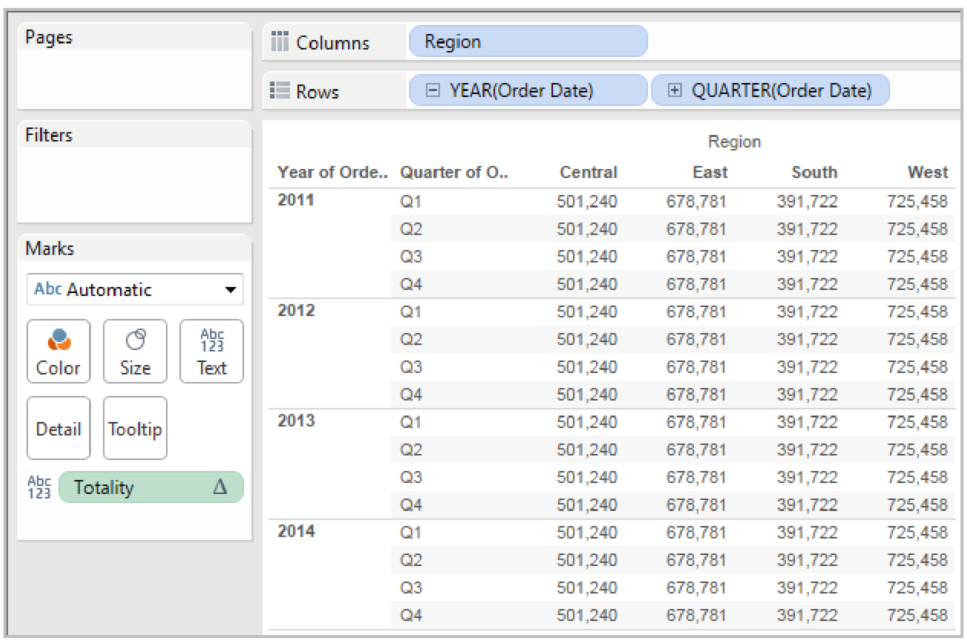
The default **Compute Using** value is **Table (Across)**. The result is that **Totality** is summing the values across each row of your table. Thus, the value that you see across each row is the sum of the values from the original version of the table.

The values in the 2011/Q1 row in the original table were $8601, $6579, $44262, and $15006. The values in the table after **Totality** replaces **SUM(Sales)** are all $74,448, which is the sum of the four original values.

Notice the triangle next to Totality after you drop it on Text:



This indicates that this field is using a table calculation. You can right-click the field and choose **Edit Table Calculation** to redirect your function to a different **Compute Using** value. For example, you could set it to **Table (Down)**. In that case, your table would look like this:

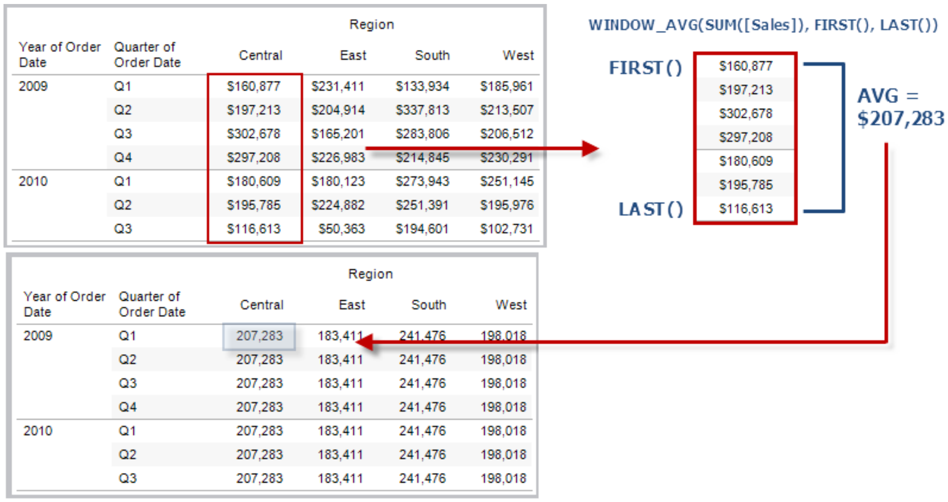


		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2012	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2013	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2014	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458

WINDOW_AVG(expression, [start, end])

Returns the average of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window average within the Date partition returns the average sales across all dates.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	207,283	183,411	241,476	198,018
	Q2	207,283	183,411	241,476	198,018
	Q3	207,283	183,411	241,476	198,018
	Q4	207,283	183,411	241,476	198,018
2010	Q1	207,283	183,411	241,476	198,018
	Q2	207,283	183,411	241,476	198,018
	Q3	207,283	183,411	241,476	198,018

Example

`WINDOW_AVG(SUM([Profit]), FIRST()+1, 0)` computes the average of SUM(Profit) from the second row to the current row.

WINDOW_CORR(expression1, expression2, [start, end])

Returns the Pearson correlation coefficient of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship.

There is an equivalent aggregation fuction: CORR. See **Tableau Functions (Alphabetical) on page 2342**.

Example

The following formula returns the Pearson correlation of **SUM(Profit)** and **SUM(Sales)** from the five previous rows to the current row.

```
WINDOW_CORR(SUM[Profit]), SUM([Sales]), -5, 0)
```

WINDOW_COUNT(expression, [start, end])

Returns the count of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0)` computes the count of SUM(Profit) from the second row to the current row

WINDOW_COVAR(expression1, expression2, [start, end])

Returns the *sample covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end arguments are omitted, the window is the entire partition.

Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (with the WINDOW_COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

There is an equivalent aggregation fuction: COVAR. See **Tableau Functions (Alphabetical)** on page 2342.

Example

The following formula returns the sample covariance of **SUM(Profit)** and **SUM(Sales)** from the two previous rows to the current row.

```
WINDOW_COVAR(SUM([Profit]), SUM([Sales]), -2, 0)
```

WINDOW_COVARP(expression1, expression2, [start, end])

Returns the *population covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the WINDOW_COVAR function) is appropriate.

There is an equivalent aggregation fuction: COVARP. **Tableau Functions (Alphabetical)** on page 2342.

Example

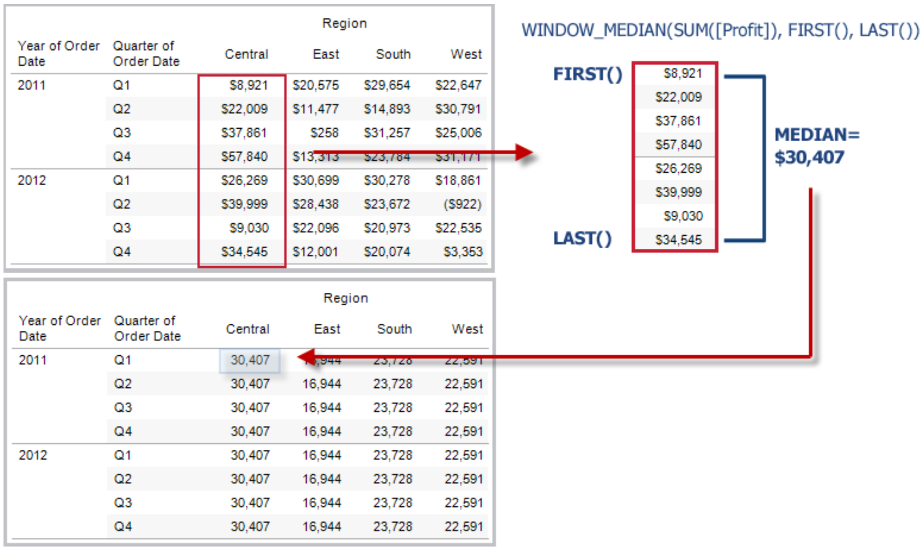
The following formula returns the population covariance of **SUM(Profit)** and **SUM(Sales)** from the two previous rows to the current row.

```
WINDOW_COVARP(SUM([Profit]), SUM([Sales]), -2, 0)
```

WINDOW_MEDIAN(expression, [start, end])

Returns the median of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly profit. A window median within the Date partition returns the median profit across all dates.



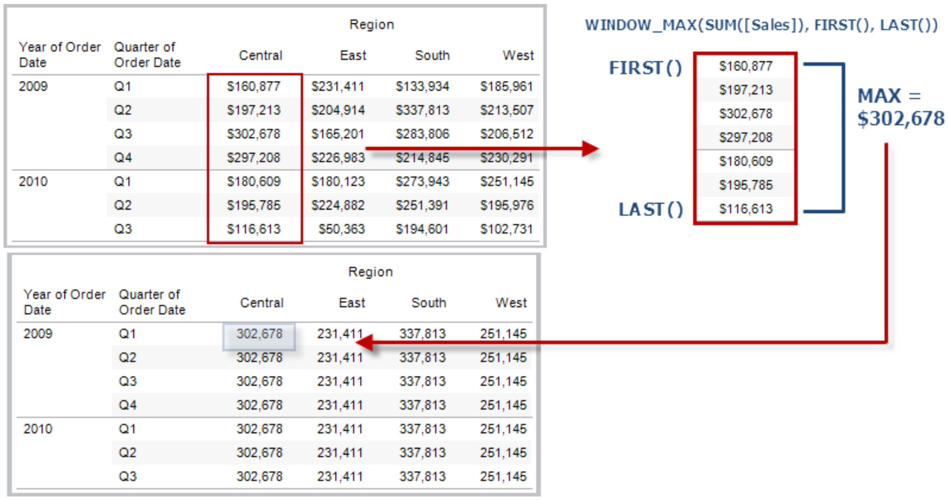
Example

`WINDOW_MEDIAN(SUM([Profit]), FIRST()+1, 0)` computes the median of SUM (Profit) from the second row to the current row.

WINDOW_MAX(expression, [start, end])

Returns the maximum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window maximum within the Date partition returns the maximum sales across all dates.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	302,678	231,411	337,813	251,145
	Q2	302,678	231,411	337,813	251,145
	Q3	302,678	231,411	337,813	251,145
	Q4	302,678	231,411	337,813	251,145
2010	Q1	302,678	231,411	337,813	251,145
	Q2	302,678	231,411	337,813	251,145
	Q3	302,678	231,411	337,813	251,145

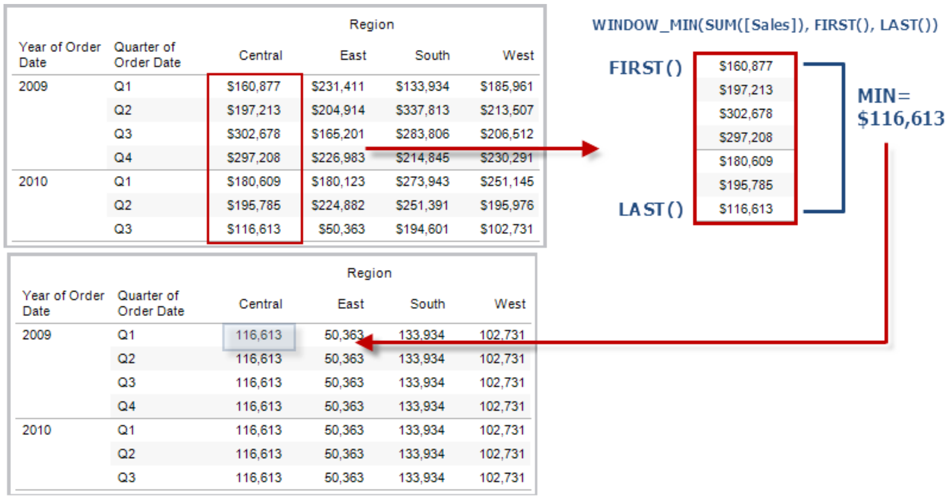
Example

`WINDOW_MAX(SUM([Profit]), FIRST()+1, 0)` computes the maximum of SUM (Profit) from the second row to the current row.

WINDOW_MIN(expression, [start, end])

Returns the minimum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window minimum within the Date partition returns the minimum sales across all dates.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	116,613	50,363	133,934	102,731
	Q2	116,613	50,363	133,934	102,731
	Q3	116,613	50,363	133,934	102,731
	Q4	116,613	50,363	133,934	102,731
2010	Q1	116,613	50,363	133,934	102,731
	Q2	116,613	50,363	133,934	102,731
	Q3	116,613	50,363	133,934	102,731

Example

`WINDOW_MIN(SUM([Profit]), FIRST()+1, 0)` computes the minimum of SUM(Profit) from the second row to the current row.

WINDOW_PERCENTILE(expression, number, [start, end])

Returns the value corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_PERCENTILE(SUM([Profit]), 0.75, -2, 0)` returns the 75th percentile for SUM(Profit) from the two previous rows to the current row.

WINDOW_STDEV(expression, [start, end])

Returns the sample standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_STDEV(SUM([Profit]), FIRST()+1, 0)` computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_STDEVP(expression, [start, end])

Returns the biased standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

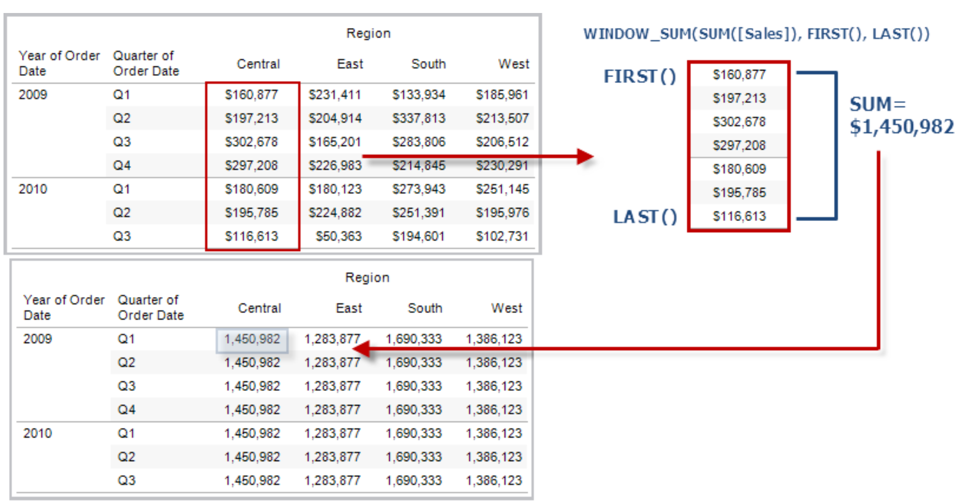
Example

`WINDOW_STDEVP(SUM([Profit]), FIRST()+1, 0)` computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_SUM(expression, [start, end])

Returns the sum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window sum computed within the Date partition returns the summation of sales across all quarters.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	1,450,982	1,283,877	1,690,333	1,386,123
	Q2	1,450,982	1,283,877	1,690,333	1,386,123
	Q3	1,450,982	1,283,877	1,690,333	1,386,123
	Q4	1,450,982	1,283,877	1,690,333	1,386,123
2010	Q1	1,450,982	1,283,877	1,690,333	1,386,123
	Q2	1,450,982	1,283,877	1,690,333	1,386,123
	Q3	1,450,982	1,283,877	1,690,333	1,386,123

Example

`WINDOW_SUM(SUM([Profit]), FIRST()+1, 0)` computes the sum of SUM(Profit) from the second row to the current row.

WINDOW_VAR(expression, [start, end])

Returns the sample variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VAR((SUM([Profit])), FIRST()+1, 0)` computes the variance of SUM (Profit) from the second row to the current row.

WINDOW_VARP(expression, [start, end])

Returns the biased variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VARP(SUM([Profit]), FIRST()+1, 0)` computes the variance of SUM (Profit) from the second row to the current row.

Create a table calculation using the calculation editor

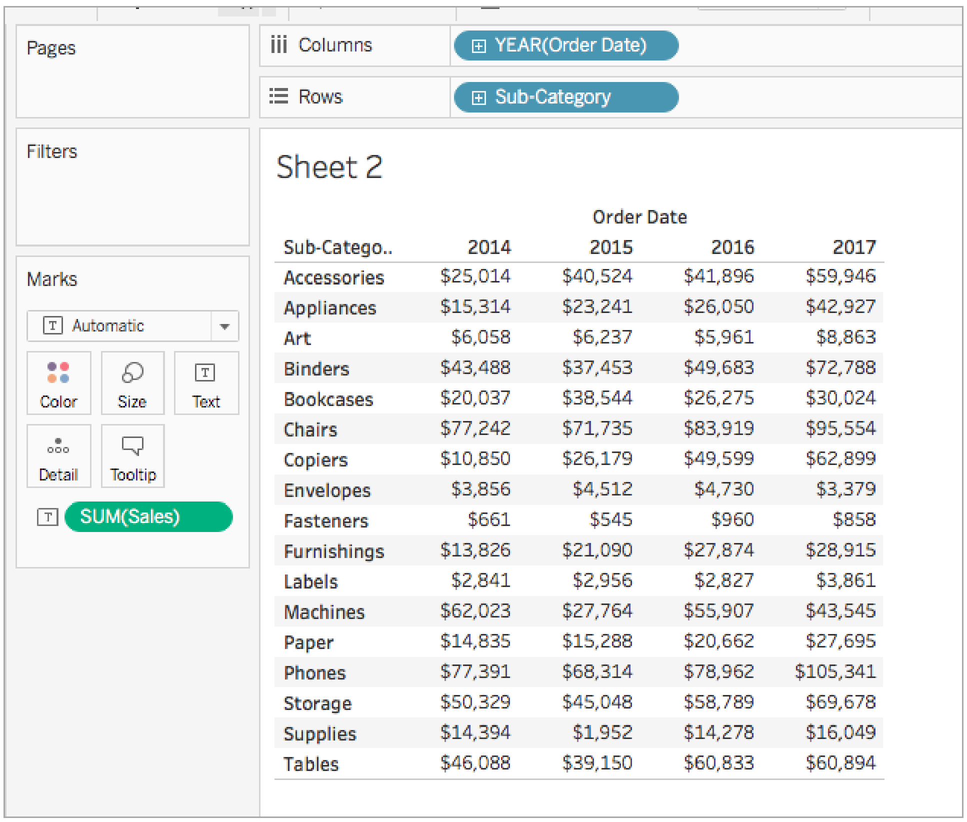
Follow along with the steps below to learn how to create a table calculation using the calculation editor.

Note: There are several ways to create table calculations in Tableau. This example demonstrates only one of those ways. For more information, see **Transform Values with Table Calculations** on page 2406.

Step 1: Create the visualization

1. In Tableau Desktop, connect to the **Sample-Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. From the **Data** pane, under Dimensions, drag **Order Date** to the **Columns** shelf.
4. From the **Data** pane, under Dimensions, drag **Sub-Category** to the **Rows** shelf.
5. From the **Data** pane, under Measures, drag **Sales** to **Text** on the Marks card.

   Your visualization updates to a text table.



Sub-Catego..	2014	2015	2016	2017
Accessories	$25,014	$40,524	$41,896	$59,946
Appliances	$15,314	$23,241	$26,050	$42,927
Art	$6,058	$6,237	$5,961	$8,863
Binders	$43,488	$37,453	$49,683	$72,788
Bookcases	$20,037	$38,544	$26,275	$30,024
Chairs	$77,242	$71,735	$83,919	$95,554
Copiers	$10,850	$26,179	$49,599	$62,899
Envelopes	$3,856	$4,512	$4,730	$3,379
Fasteners	$661	$545	$960	$858
Furnishings	$13,826	$21,090	$27,874	$28,915
Labels	$2,841	$2,956	$2,827	$3,861
Machines	$62,023	$27,764	$55,907	$43,545
Paper	$14,835	$15,288	$20,662	$27,695
Phones	$77,391	$68,314	$78,962	$105,341
Storage	$50,329	$45,048	$58,789	$69,678
Supplies	$14,394	$1,952	$14,278	$16,049
Tables	$46,088	$39,150	$60,833	$60,894

Step 2: Create the table calculation

1. Select **Analysis** > **Create Calculated Field**.
2. In the calculation editor that opens, do the following:

- Name the calculated field, Running Sum of Profit.
- Enter the following formula:

  ```
  RUNNING_SUM(SUM([Profit]))
  ```

  This formula calculates the running sum of profit sales. It is computed across the entire table.

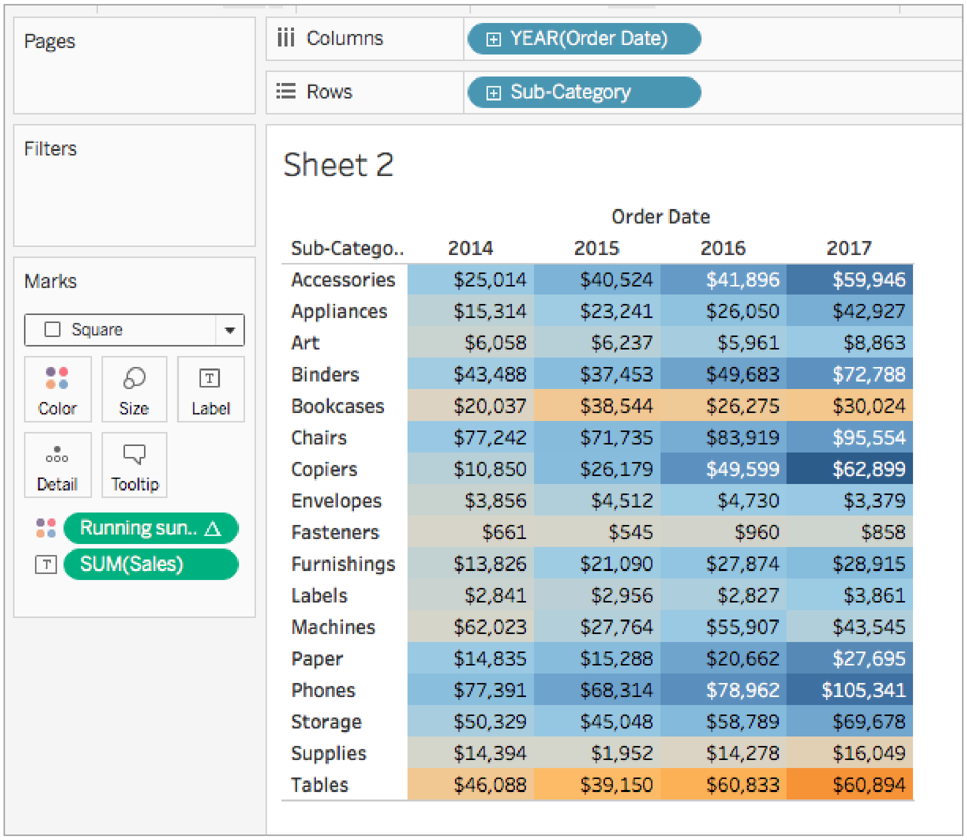
- When finished, click **OK**.

  The new table calculation field appears under Measures in the Data pane. Just like your other fields, you can use it in one or more visualizations.

Step 3: Use the table calculation in the visualization

1. From the Data pane, under Measures, drag **Running Sum of Profit** to **Color** on the Marks card.
2. On the Marks card, click the Mark Type drop-down and select **Square**.

The visualization updates to a highlight table:

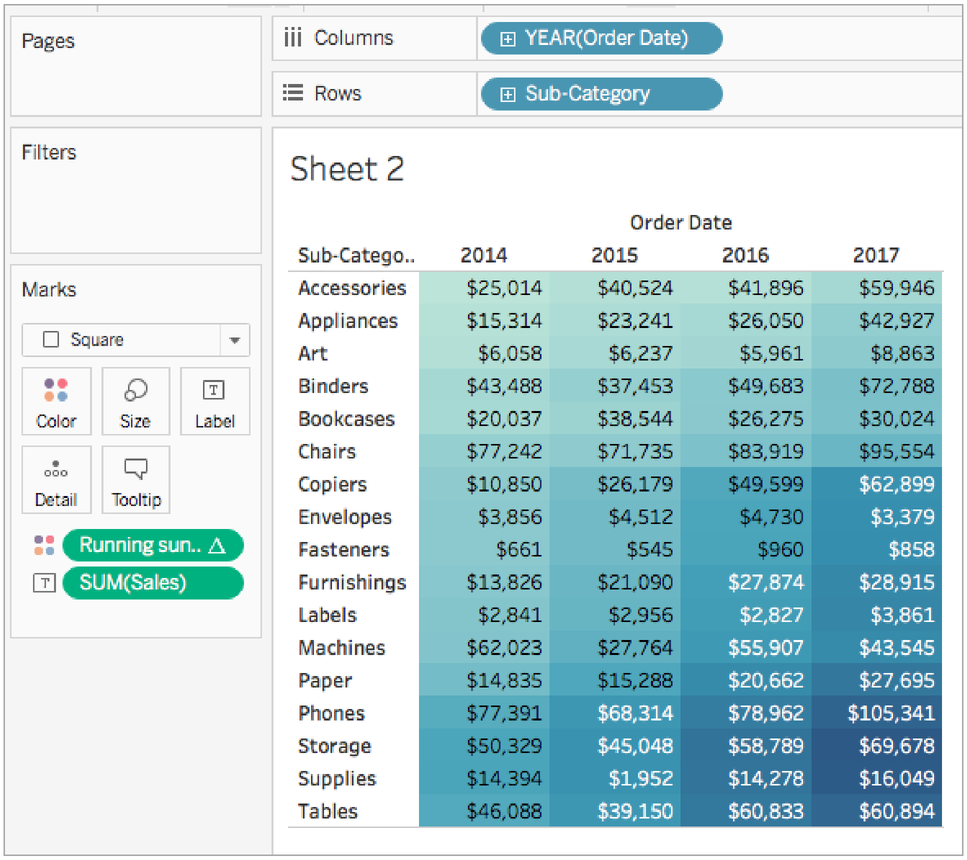


Step 4: Edit the table calculation

1. On the Marks card, right-click **Running Sum of Profit** and select **Edit Table Calculation**.

2. In the Table Calculation dialog box that opens, under Compute Using, select **Table (down)**.

The visualization updates to the following:



	Order Date			
Sub-Catego..	2014	2015	2016	2017
Accessories	$25,014	$40,524	$41,896	$59,946
Appliances	$15,314	$23,241	$26,050	$42,927
Art	$6,058	$6,237	$5,961	$8,863
Binders	$43,488	$37,453	$49,683	$72,788
Bookcases	$20,037	$38,544	$26,275	$30,024
Chairs	$77,242	$71,735	$83,919	$95,554
Copiers	$10,850	$26,179	$49,599	$62,899
Envelopes	$3,856	$4,512	$4,730	$3,379
Fasteners	$661	$545	$960	$858
Furnishings	$13,826	$21,090	$27,874	$28,915
Labels	$2,841	$2,956	$2,827	$3,861
Machines	$62,023	$27,764	$55,907	$43,545
Paper	$14,835	$15,288	$20,662	$27,695
Phones	$77,391	$68,314	$78,962	$105,341
Storage	$50,329	$45,048	$58,789	$69,678
Supplies	$14,394	$1,952	$14,278	$16,049
Tables	$46,088	$39,150	$60,833	$60,894

See Also

Create a table calculation on page 2416

Table Calculation Types on page 2419

Customize Table Calculations on page 2446

Quick Table Calculations on page 2440

Functions in Tableau on page 2147

Tableau Functions (by Category) on page 2267

Tableau Functions (Alphabetical) on page 2342

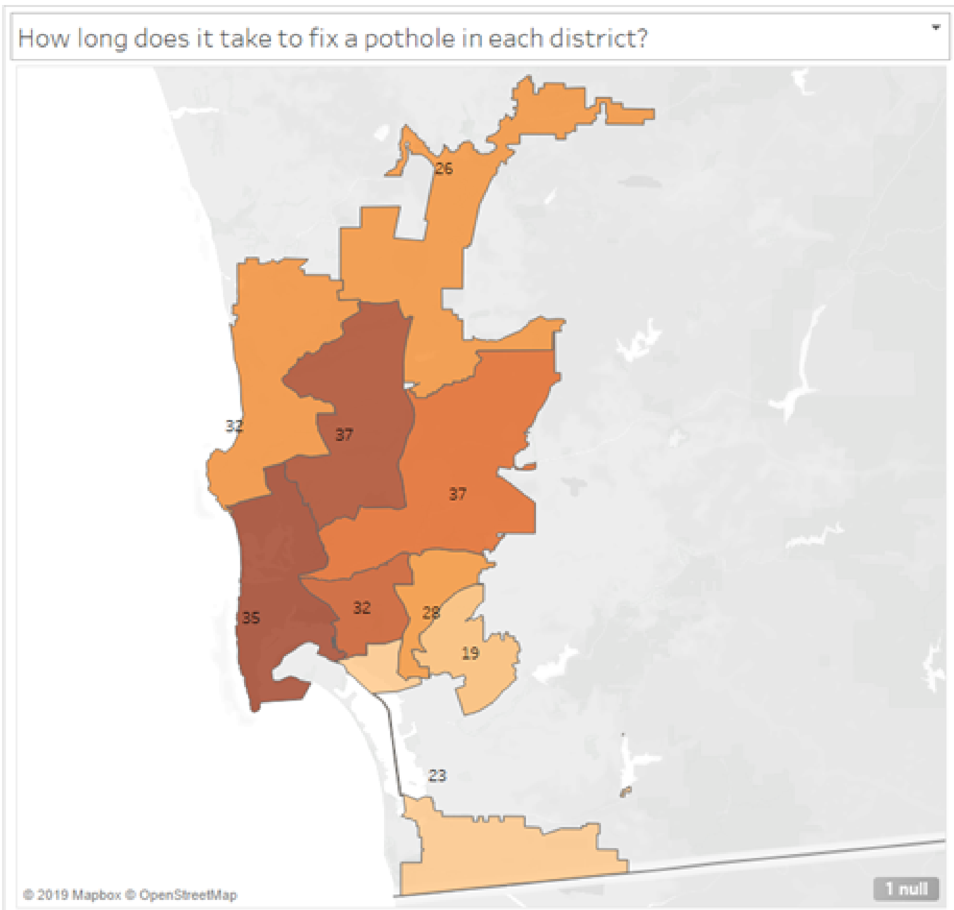
Spatial Functions

This article introduces spatial functions and their uses in Tableau. It also demonstrates how to create a spatial calculation using an example.

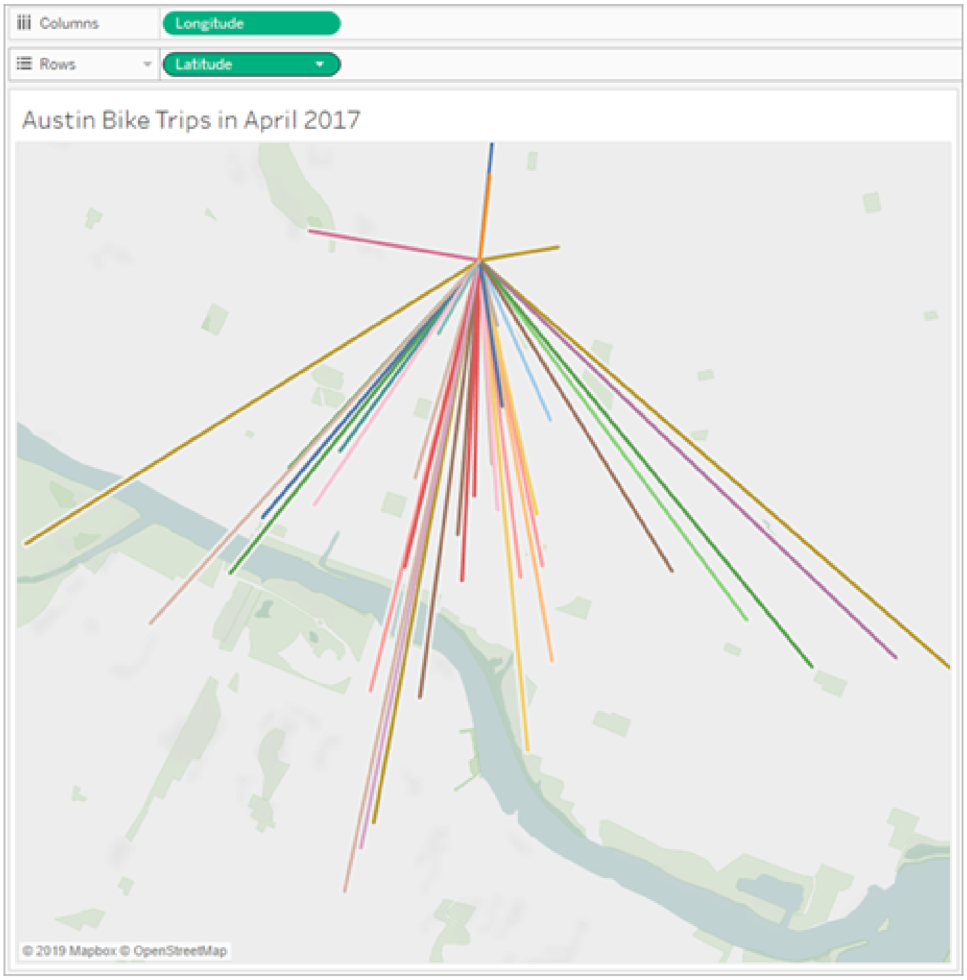
Why use spatial functions

Spatial functions allow you to perform advanced spatial analysis and combine spatial files with data in other formats like text files or spreadsheets. For example, you might have a spatial file of city council districts, and a text file containing latitude and longitude coordinates of reported potholes. You can use a spatial calculation when creating your data source to join these files and analyze which district takes the longest to repair potholes.

The visualization might look something like this:



Another example might be to create a line that connects two data points for origin-destination maps. For example, you might have a spreadsheet of bikeshare data that tells you where cyclists began and ended their trips. The visualization might look something like this:



List of spatial functions in Tableau

Function	Syntax	Definition
Area	`Area (Geometry, "units")`	Returns the total surface area of a spatial polygon. Supported unit names: meters ("meters," "metres" "m), kilometers ("kilometers," "kilometres," "km"), miles ("miles" or "mi"), feet ("feet," "ft").
MakeLine	`MakeLine(geometry1,geometry2)`	Generates a line mark between two points; useful for building origin-

		destination maps.
MakePoint	`MakePoint(latitude, longitude)`	Converts data from latitude and longitude columns into spatial objects.
MakePoint	`MakePoint(coordinatesX, coordinatesY, SRID)`	Converts data from projected geographic coordinates into spatial objects. SRID is a spatial reference identifier that uses ESPG reference system codes to specify coordinate systems. If SRID is not specified, WGS84 is assumed and parameters are treated as latitude/longitude in degrees. This function can only be created with a live connection and will continue to work when a data source is converted to an extract.
Distance	`Distance(geometry1,geometry2,"units")`	Returns distance measurement between two points in a specified unit. Supported unit names: meters ("meters," "metres" "m), kilometers ("kilometers," "kilometres," "km"), miles ("miles" or "miles"), feet ("feet," "ft"). This function can only be created with a live connection and will continue to work when a data source is converted to an extract.
Buffer	`Buffer(Spatial Point, distance, "units")`	Returns a circular shape with a radius determined by the distance and unit values defined in the calculation. **Note:** The Buffer

calculation will only work with a Point spatial object.

Buffer supports the same unit names as the Distance function.

Use a spatial calculation

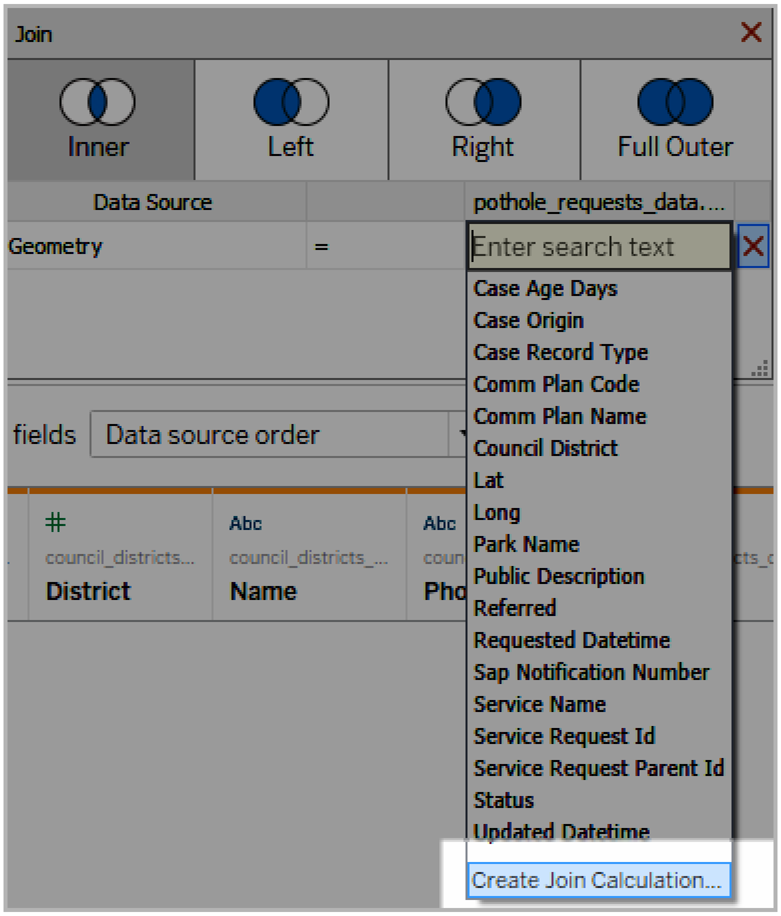
Create a spatial data source using MakePoint

You can use MakePoint to spatially-enable a data source so that it can be joined with a spatial file using a spatial join. To use MakePoint, your data must contain latitude and longitude coordinates.

1. Open Tableau and connect to a spatial data source. Under Connections, click Add to add a second, non-spatial data source.

   The two data sources are added to the canvas.
2. Click the Join icon.
3. In the Join dialog box that appears, do the following:
   - Select a join type.
   - Under Data Source, select a spatial field from your spatial file to join by. Spatial fields have a globe icon next to them.

4. For the non-spatial data source, select Create Join Calculation as the join clause.



The calculation might look something like this:

```
MAKEPOINT(Latitude,Longitude)
```

5. After adding the spatial function to your text file, select the Intersects join clause operator to create a data source for spatial analysis.
6. When finished, close the Join dialog box.

For more information on spatial joins, see **Join Spatial Files in Tableau** on page 1815

Create a visualization using MakeLine

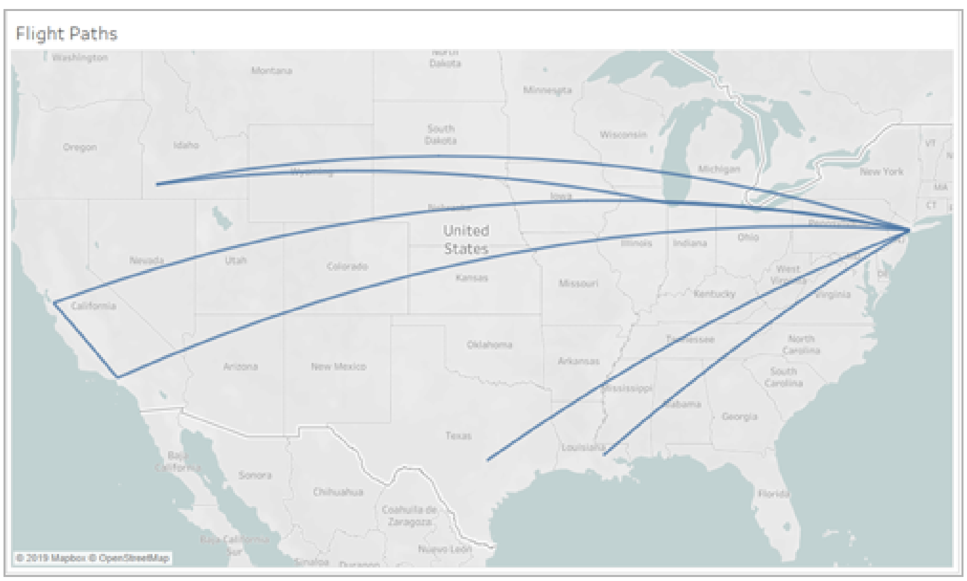
In Tableau Desktop, download the Flight Path workbook from Tableau Public, available here.

1. Navigate to a new worksheet.
2. Select Analysis > Create Calculated Field.
3. In the calculation that opens, do the following:
   - Name the calculated field Flight Paths
   - Enter the following formula

```
MAKELINE(MAKEPOINT([Lat],[Long]),MAKEPOINT([Dest Lat],
[Dest Lng]))
```

   This formula takes latitude and longitude coordinates from your origin and destination cities and turns them into geographic points for spatial analysis. Those coordinates are used to build to two-point lines between origin and destination.
4. When finished, click **OK**.

   The new calculated field appears under Dimensions in the Data pane. Just like your other fields, you can use it in one or more visualizations.
5. From the Data pane, double-click Flight Paths to add it to your visualization, which should automatically render as a map.

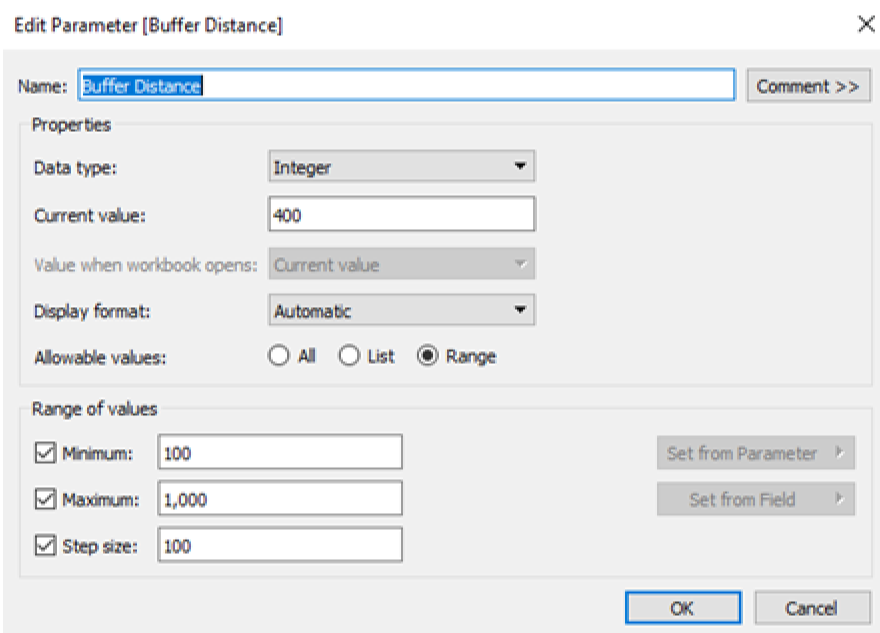


The calculation automatically produces curved geodesic lines when the lines span longer expanses of the globe.

Visualize an area with Buffer

In Tableau Desktop, download the Flight Path workbook from Tableau Public, available here.

1. Navigate to a new worksheet.
2. Right click the Measures Pane and select Create Parameter.
3. In the Parameter dialog that opens, set the options below:
   - Name the parameter Buffer Distance
   - Set the Data Type to Integer
   - Set Allowable values to Range
   - Set the Minimum range to 100, the Maximum range to 1000, and the step size to 100.



4. When finished, click **OK**

   This parameter will allow us to customize the radius of our buffer, ranging from 100 to 1000 miles. Right click the parameter and select **Show Parameter Control.**
5. Select Analysis > Create Calculated Field.
6. In the calculation that opens, do the following:
   - Name the calculated field Buffer
   - Enter the following formula

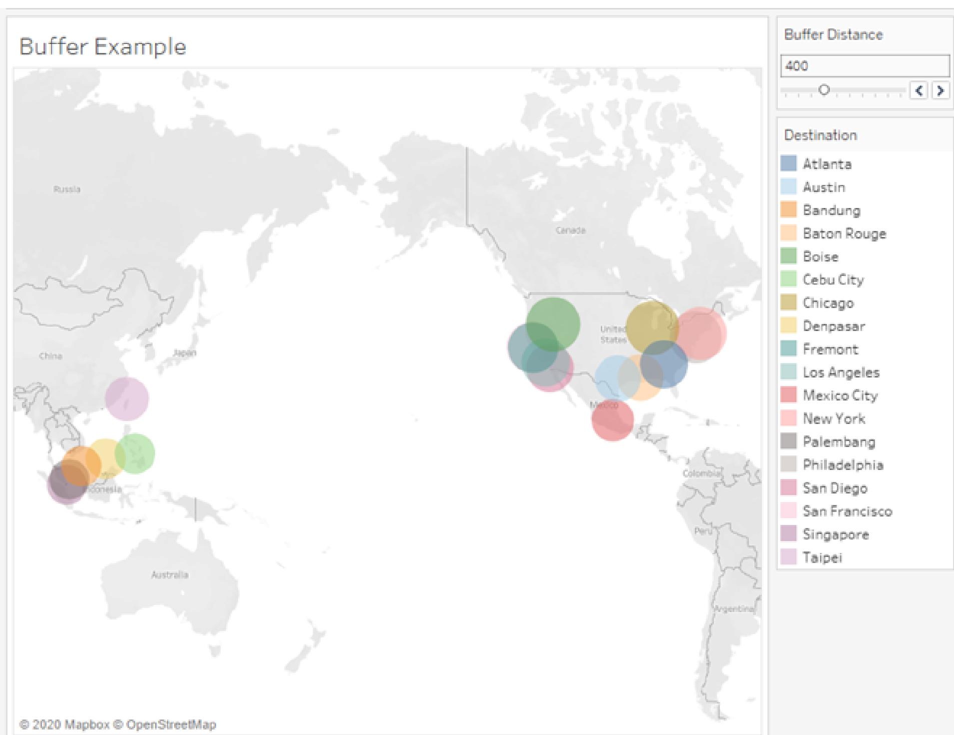
```
BUFFER(MAKEPOINT([Dest Lat],[Dest Lng]),[Buffer
Distance],"miles")
```

The buffer formula takes point spatial data and converts it into shapes with a radius in miles determined by the Buffer Distance parameter.

Note: Because Buffer can only be used with point spatial data, we are converting the latitude and longitude data into a point with Makepoint, as demonstrated in the previous example.

7. When finished, click **OK**.

   The new calculated field appears under Dimensions in the Data pane. Just like your other fields, you can use it in one or more visualizations.
8. From the Data pane, double-click Buffer to add it to your visualization, which should automatically render as a map.
9. Drag Destination to the Color panel on the Marks card to complete the visualization.



> If your view does not look like the image above, make sure that the Mark type is set to **Map** and not **Circle**. For more information on Marks type, see **Change the Type of Mark in the View** on page 1332.

Predictive Modeling Functions

This article introduces predictive modeling functions and their uses in Tableau. It also demonstrates with an example how to create table calculations using the predictive modeling functions.

Why use predictive modeling functions

Predictive modeling functions can help you quickly generate predictions that can be manipulated, visualized, and exported like data using table calculations.

Before, you may have had to integrate Tableau with R and Python in order to perform advanced statistical calculations and visualize them in Tableau. Now, you can select targets and predictors by updating the variables and visualizing multiple models with different combinations of predictors. The data can be filtered, aggregated, and transformed at all levels of detail, with inputs and predictions automatically recalculated to match the data in the view.

For more information about predictive modeling functions in Tableau, see **How Predictive Modeling Functions Work in Tableau** on page 2606

Predictive modeling functions available in Tableau:

Function	Syntax	Description
MODEL_ QUANTILE	`MODEL_QUANTILE( model_specification (optional), quantile, target_expression, predictor_expression(s))`	Returns a target numeric value within the probable range defined by the target expression and other predictors, at a specified quantile. This is the Posterior Predictive Quantile. Example:

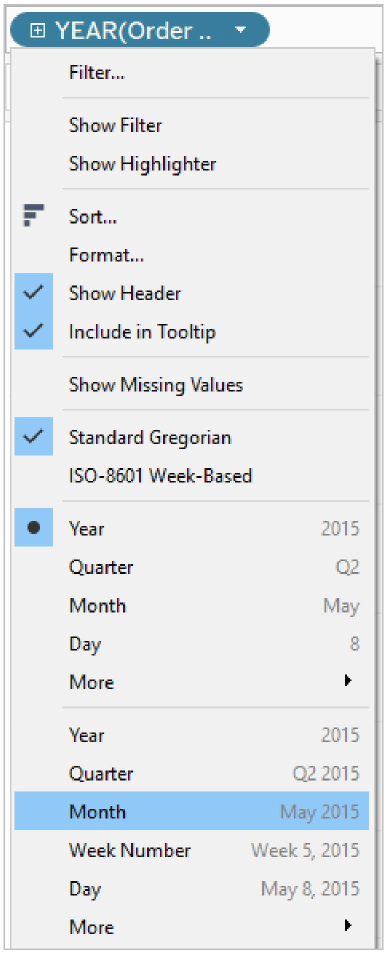
		`MODEL_QUANTILE(0.5, SUM([Sales]), COUNT([Orders]))`
MODEL_PERCENTILE	`MODEL_PERCENTILE(model_specification (optional), target_expression, predictor_expression(s))`	Returns the probability (between 0 and 1) of the expected value being less than or equal to the observed mark, defined by the target expression and other predictors. This is the Posterior Predictive Distribution Function, also known as the Cumulative Distribution Function (CDF). Example: `MODEL_PERCENTILE( SUM([Sales]), COUNT([Orders]))`

Create a prediction calculation

Follow along with the steps below to learn how to create a simple prediction calculation using the MODEL_QUANTILE function. For a more detailed example, see **Example - Explore Female Life Expectancy with Predictive Modeling Functions** on page 2637

Step 1: Create a Visualization

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to a worksheet.
3. From the **Data** pane, drag the **Order Date** dimension to the Columns shelf.
4. Open the measure's context menu to change its list level to Month and Year:



5. Drag **Sales** to the Rows shelf.

Step 2: Create the Calculated Field

1. Click to open the **Analysis** menu at the top, and then select **Create Calculated Field**.
2. In the Calculation Editor, do the following:
   - Name the calculation: **Predict Median Sales.**
   - Enter the following formula:

     ```
     MODEL_QUANTILE(0.5, SUM([Sales]),ATTR(DATETRUNC('month',
     [Order Date])))
     ```

     Remember: The MODEL_QUANTILE function takes a given quantile and predicts values based on the predictors you input.

Let's break this down:

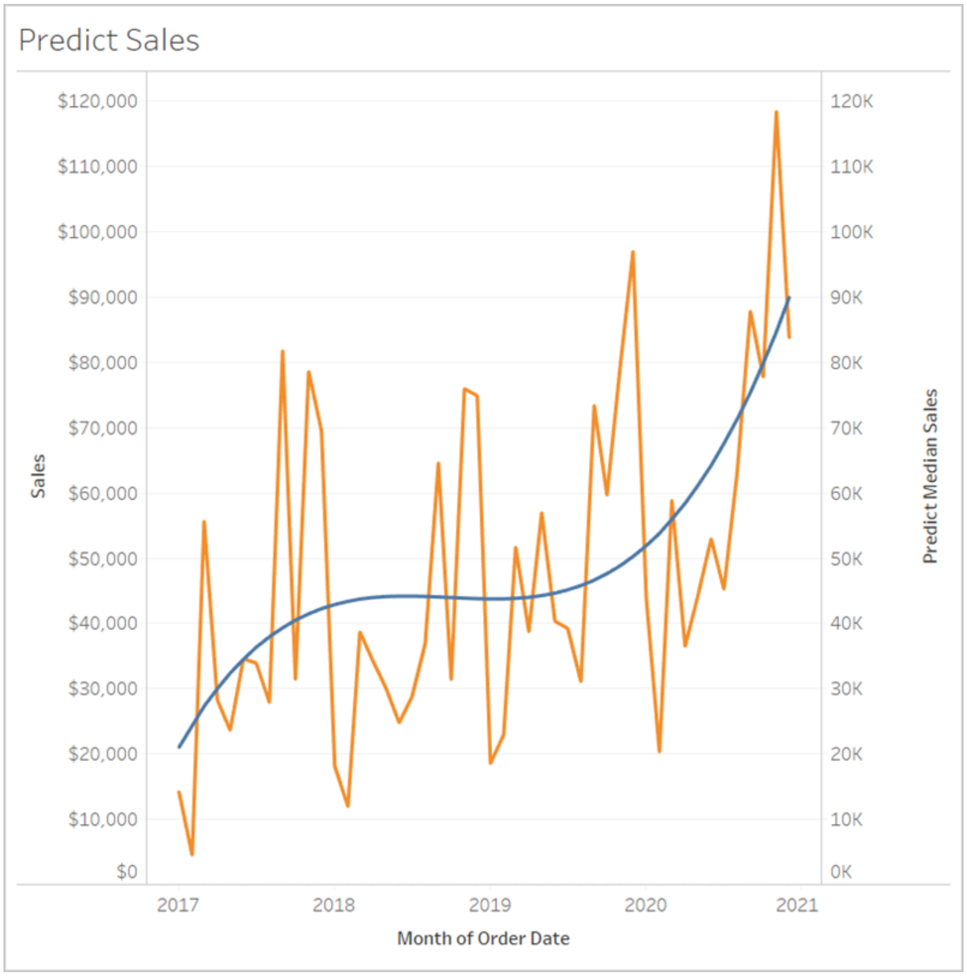
- In this case, the quantile = 0.5, which predicts the median.
- We want to predict sales, so the target expression is SUM([Sales]).
- We want to base the prediction on past performance, so we include date as a predictor, which is the last argument in the calculation.

3. When finished, click **OK**.

   The prediction calculation is now added as a calculated field in the Data pane.

Step 3: Add the Prediction Calculation to the View

1. Drag the prediction calculation to the Rows shelf, to the right of SUM(Sales).
2. Right-click (Control-click on Mac) the measure and select **Dual Axis**.
3. To align the two axes in a dual axes chart to use the same scale, right-click (Control-click on Mac) the secondary axis, in this case **Predict Median Sales**, and select **Synchronize Axis**. This aligns the scale of the two axes.



That's all there is to it. To find out how you can extend a date axis and predict the future, see **Predictive Modeling Functions in Time Series Visualizations** on page 2649.

Rules for Prediction Calculations

- You can't mix aggregate and non-aggregate arguments. If the target expression is an aggregate, so must the predictor.
- The functions are best used to predict values for individual records, on vizzes where each mark represents a discrete entity, such as a person, a product, a sale, etc.
- The functions are best used to predict values for aggregated target expressions using SUM and COUNT.
- The functions aren't recommended to predict values for aggregated target expressions using AVG, MEDIAN, MIN, or MAX.
- The functions should use predictors that are at the same level of detail or higher than the viz.

Additional Functions

REGEXP_REPLACE(string, pattern, replacement)

Returns a copy of the given string where the regular expression pattern is replaced by the replacement string. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern and the replacement must be constants.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_REPLACE('abc 123', '\s', '-') = 'abc-123'

REGEXP_MATCH(string, pattern)

Returns true if a substring of the specified string matches the regular expression pattern. This function is available for Text File, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Impala 2.3.0 (through Cloudera Hadoop data sources), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_MATCH('-([1234].[The.Market])-','\[\s*(\w*\.)(\w*\s*\])')=true

REGEXP_EXTRACT(string, pattern)

Returns the portion of the string that matches the regular expression pattern. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT('abc 123', '[a-z]+\s+(\d+)') = '123'

REGEXP_EXTRACT_NTH(string, pattern, index)

Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If index is 0, the entire string is returned. This function is available for Text File, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT_NTH('abc 123', '([a-z]+)\s+(\d+)', 2) = '123'

Hadoop Hive Specific Functions

Note: Only the PARSE_URL and PARSE_URL_QUERY functions are available for Cloudera Impala data sources.

GET_JSON_OBJECT(JSON string, JSON path)

Returns the JSON object within the JSON string based on the JSON path.

PARSE_URL(string, url_part)

Returns a component of the given URL string where the component is defined by url_part. Valid url_part values include: 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'AUTHORITY', 'FILE' and 'USERINFO'.

Example

PARSE_URL('http://www.tableau.com', 'HOST') = 'www.tableau.com'

PARSE_URL_QUERY(string, key)

Returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

Example

PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4', 'page') = '1'

XPATH_BOOLEAN(XML string, XPath expression string)

Returns true if the XPath expression matches a node or evaluates to true.

Example

XPATH_BOOLEAN('<values> <value id="0">1</value><value id="1">5</value>', 'values/value [@id="1"] = 5') = true

XPATH_DOUBLE(XML string, XPath expression string)

Returns the floating-point value of the XPath expression.

Example

XPATH_DOUBLE('<values><value>1.0</value><value>5.5</value> </values>', 'sum(value/*)') = 6.5

XPATH_FLOAT(XML string, XPath expression string)

Returns the floating-point value of the XPath expression.

Example

XPATH_FLOAT('<values><value>1.0</value><value>5.5</value> </values>','sum(value/*)') = 6.5

XPATH_INT(XML string, XPath expression string)

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_INT('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_LONG(XML string, XPath expression string)

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_LONG('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_SHORT(XML string, XPath expression string)

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_SHORT('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_STRING(XML string, XPath expression string)

Returns the text of the first matching node.

Example

XPATH_STRING('<sites ><url domain="org">http://www.w3.org</url> <url domain="com">http://www.tableau.com</url></sites>', 'sites/url[@domain="com"]') = 'http://www.tableau.com'

Google BigQuery Specific Functions

DOMAIN(string_url)

Given a URL string, returns the domain as a string.

Example

DOMAIN('http://www.google.com:80/index.html') = 'google.com'

GROUP_CONCAT(expression)

Concatenates values from each record into a single comma-delimited string. This function acts like a SUM() for strings.

Example

GROUP_CONCAT(Region) = "Central,East,West"

HOST(string_url)

Given a URL string, returns the host name as a string.

Example

HOST('http://www.google.com:80/index.html') = 'www.google.com:80'

LOG2(number)

Returns the logarithm base 2 of a number.

Example

LOG2(16) = '4.00'

LTRIM_THIS(string, string)

Returns the first string with any leading occurrence of the second string removed.

Example

LTRIM_THIS('[-Sales-]','[-') = 'Sales-]'

RTRIM_THIS(string, string)

Returns the first string with any trailing occurrence of the second string removed.

Example

RTRIM_THIS('[-Market-]','-]') = '[-Market'

TIMESTAMP_TO_USEC(expression)

Converts a TIMESTAMP data type to a UNIX timestamp in microseconds.

Example

TIMESTAMP_TO_USEC(#2012-10-01 01:02:03#)=1349053323000000

USEC_TO_TIMESTAMP(expression)

Converts a UNIX timestamp in microsseconds to a TIMESTAMP data type.

Example

USEC_TO_TIMESTAMP(1349053323000000) = #2012-10-01 01:02:03#

TLD(string_url)

Given a URL string, returns the top level domain plus any country domain in the URL.

Example

TLD('http://www.google.com:80/index.html') = '.com'

TLD('http://www.google.co.uk:80/index.html') = '.co.uk'

FORMAT() Function Workarounds in Tableau

Tableau has no FORMAT() function for formatting fields, but it does provide a variety of means to change the structure and appearance of fields in a workbook:

- For information on formatting geographic fields, see **Assign Geographic Roles** on page 1834.
- For information on formatting date or number fields, see **Set the default number format** on page 1159. And for information creating custom date formats, see **Custom Date Formats** on page 1261.
- For information on symbols and conventions that you can use to specify field formats, see **Literal expression syntax** on page 2144.

- For information on formatting numbers and null values, see **Format Numbers and Null Values** on page 2945.

Tableau also provides a range of string functions that you can use to customize the appearance of string fields in a view. See **String Functions** on page 2160.

Tableau Functions (by Category)

The Tableau functions in this reference are organized by category. Click a category to browse its functions. Or press Ctrl+F (Command-F on a Mac) to open a search box that you can use to search the page for a specific function.

Number functions

Function	Syntax	Description

ABS	`ABS(number)`	Returns the absolute value of the given number.
		Examples:
		`ABS(-7) = 7`
		`ABS([Budget Variance])`
		The second example returns the absolute value for all the numbers contained in the `Budget Variance` field.

ACOS	`ACOS(number)`	Returns the arc cosine of the given number. The result is in radians. Example: `ACOS(-1) = 3.14159265358979`
ASIN	`ASIN(number)`	Returns the arc sine of a given number. The result is in radians. Example:

		`ASIN(1) = 1.5707963267949`
ATAN	`ATAN(number)`	Returns the arc tangent of a given number. The result is in radians. Example: `ATAN(180) = 1.5652408283942`
ATAN2	`ATAN2(y number, x number)`	Returns the arc tangent of two given numbers (x and y). The result is in radians. Example: `ATAN2(2, 1) = 1.10714871779409`
CEILING	`CEILING (number)`	Rounds a number to the nearest integer of equal or greater value. Example: `CEILING(3.1415) = 4`

Availability by data source:

Data Source	**Support**
Microsoft Access	Not supported
Microsoft Excel	Supported
Text File	Supported
Statistical File	Supported
Tableau Server	Supported
Actian Vector	Not supported

	Amazon Aurora for MySQL	Not supported
	Amazon EMR Hadoop Hive	Supported
	Amazon Redshift	Not supported
	Aster Database	Not supported
	Cloudera Hadoop	Supported
	DataStax Enterprise	Supported
	EXASOL	Not supported
	Firebird	Not supported
	Google Analytics	Supported
	Google BigQuery	Supported
	Google Cloud SQL	Not supported
	Google Sheets	Not supported
	Hortonworks Hadoop Hive	Supported
	IBM BigInsights	Not supported
	IBM DB2	Not supported
	IBM PDA (Netezza)	Not supported
	MapR Hadoop Hive	Supported
	MarkLogic	Not supported
	Microsoft Analysis Services	Not supported
	Microsoft PowerPivot	Not supported
	Microsoft SQL Server	Supported
	MySQL	Not supported
	Oracle	Not supported

		Oracle Essbase	Not supported
		Actian Matrix (ParAccel)	Not supported
		Pivotal Greenplum	Not supported
		PostgreSQL	Not supported
		Progress OpenEdge	Not supported
		Salesforce	Supported
		SAP HANA	Not supported
		SAP Sybase ASE	Not supported
		SAP Sybase IQ	Not supported
		Spark SQL	Supported
		Splunk	Not supported
		Teradata	Not supported
		Teradata OLAP Connector	Not supported
		Vertica	Not supported

COS `COS(number)`

Returns the cosine of an angle. Specify the angle in radians.

Example:

```
COS(PI( ) /4) = 0.707106781186548
```

COT `COT(number)`

Returns the cotangent of an angle. Specify the angle in radians.

Example:

```
COT(PI( ) /4) = 1
```

DEGREES `DEGREES`

Converts a given number in radians to degrees.

	`(number)`	Example: `DEGREES(PI( )/4) = 45.0`
DIV	DIV(integer1, integer2)	Returns the integer part of a division operation, in which integer1 is divided by integer2. Example: `DIV(11,2) = 5`

EXP	`EXP(number)`	Returns e raised to the power of the given number.
		Examples:
		`EXP(2) = 7.389`
		`EXP(-[Growth Rate]*[Time])`

FLOOR	`FLOOR (number)`	Rounds a number to the nearest integer of equal or lesser value. Example: `FLOOR(3.1415) = 3`

Availability by data source:

Data Source	Support
Microsoft Access	Not supported
Microsoft Excel	Supported
Text File	Supported
Statistical File	Supported
Tableau Server	Supported

Actian Vector	Not supported
Amazon Aurora for MySQL	Not supported
Amazon EMR Hadoop Hive	Supported
Amazon Redshift	Not supported
Aster Database	Not supported
Cloudera Hadoop	Supported
DataStax Enterprise	Supported
EXASOL	Not supported
Firebird	Not supported
Google Analytics	Supported
Google BigQuery	Supported
Google Cloud SQL	Not supported
Hortonworks Hadoop Hive	Supported
IBM BigInsights	Not supported
IBM DB2	Not supported
IBM Netezza	Not supported
MapR Hadoop Hive	Supported
MarkLogic	Not supported
Microsoft Analysis Services	Not supported
Microsoft PowerPivot	Not supported
Microsoft SQL Server	Supported
MySQL	Not supported
Oracle	Not supported

		Oracle Essbase	Not supported
		ParAccel	Not supported
		Pivotal Greenplum	Not supported
		PostgreSQL	Not supported
		Progress OpenEdge	Not supported
		Salesforce	Supported
		SAP HANA	Not supported
		SAP Sybase ASE	Not supported
		SAP Sybase IQ	Not supported
		Spark SQL	Supported
		Splunk	Not supported
		Teradata	Not supported
		Teradata OLAP Connector	Not supported
		Vertica	Not supported
HEXBINX	`HEXBINX (number, number)`	Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. HEXBINX and HEXBINY are binning and plotting functions for hexagonal bins. Hexagonal bins are an efficient and elegant option for visualizing data in an x/y plane such as a map. Because the bins are hexagonal, each bin closely approximates a circle and minimizes variation in the distance from the data point to the center of the bin. This makes the clustering both more accurate and informative. Example:	

```
HEXBINX([Longitude], [Latitude])
```

HEXBINY	`HEXBINY (number, number)`	Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately. Example: `HEXBINY([Longitude], [Latitude])`
LN	`LN(number)`	Returns the natural logarithm of a number. Returns `Null` if number is less than or equal to 0.
LOG	`LOG(number [, base])`	Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used.

MAX	`MAX(number, number)`	Returns the maximum of the two arguments, which must be of the same type. Returns `Null` if either argument is `Null`. `MAX` can also be applied to a single field in an aggregate calculation.
		
		Examples:
		
		`MAX(4,7)`
		`MAX(Sales,Profit)`
		`MAX([First Name],[Last Name])`

MIN	`MIN(number, number)`	Returns the minimum of the two arguments, which must be of the same type. Returns `Null` if either argument is `Null`. `MIN` can also be applied to a single field in an aggregate calculation.
		
		Examples:
		
		`MIN(4,7)`
		`MIN(Sales,Profit)`
		`MIN([First Name],[Last Name])`

PI	`PI( )`	Returns the numeric constant pi: 3.14159.

POWER	`POWER (number, power)`	Raises the number to the specified power.
		Examples:
		`POWER(5,2) = 52 = 25`
		`POWER(Temperature, 2)`
		You can also use the ^ symbol:
		`5^2 = POWER(5,2) = 25`

RADIANS	`RADIANS (number)`	Converts the given number from degrees to radians. Example: `RADIANS(180) = 3.14159`
ROUND	`ROUND (number, [decimals])`	Rounds numbers to a specified number of digits. The `decimals` argument specifies how many decimal points of precision to include in the final result. If `decimals` is omitted, `number` is rounded to the nearest integer. Example: This example rounds every `Sales` value to an integer: `ROUND(Sales)` Some databases, such as SQL Server, allow specification of a negative `length`, where -1 rounds `number` to 10's, -2 rounds to 100's, and so on. This is not true of all databases. For example, it is not true of Excel or Access.
SIGN	`SIGN(number)`	Returns the sign of a number: The possible return values are -1 if the number is negative, 0 if the number is zero, or 1 if the number is positive.

Example:

If the average of the profit field is negative, then

```
SIGN(AVG(Profit)) = -1
```

SIN `SIN(number)`

Returns the sine of an angle. Specify the angle in radians.

Examples:

```
SIN(0) = 1.0
SIN(PI( )/4) = 0.707106781186548
```

SQRT `SQRT(number)`

Returns the square root of a number.

Example:

```
SQRT(25) = 5
```

SQUARE `SQUARE (number)`

Returns the square of a number.

Example:

```
SQUARE(5) = 25
```

TAN `TAN(number)`

Returns the tangent of an angle. Specify the angle in radians..

Example:

```
TAN(PI ( )/4) = 1.0
```

ZN `ZN (expression)`

Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values.

Example:

```
ZN([Profit]) = [Profit]
```

String functions

Function	Syntax	Definition
ASCII	`ASCII (string)`	Returns the ASCII code for the first character of `string`. Example: `ASCII('A') = 65`
CHAR	`CHAR(num-ber)`	Returns the character encoded by the ASCII code `number`. Example: `CHAR(65) = 'A'`
CONTAINS	`CONTAINS (string, substring)`	Returns true if the given string contains the specified substring. Example: `CONTAINS("Calculation", "alcu") = true`
ENDSWITH	`ENDSWITH (string, substring)`	Returns true if the given string ends with the specified substring. Trailing white spaces are ignored. Example: `ENDSWITH("Tableau", "leau") = true`

FIND	`FIND (string, substring, [start])`	Returns the index position of `substring` in `string`, or 0 if the `substring` isn't found. If the optional argument `start` is added, the function ignores any instances of `substring` that appear before the index position `start`. The first character in the string is position 1.
		Examples:
		`FIND("Calculation", "alcu") = 2`
		`FIND("Calculation", "Computer") = 0`
		`FIND("Calculation", "a", 3) = 7`
		`FIND("Calculation", "a", 2) = 2`
		`FIND("Calculation", "a", 8) = 0`
		`FIND("Calculation", "a", 3) = 7`
		`FIND("Calculation", "a", 2) = 2`
		`FIND("Calculation", "a", 8) = 0`

FINDNTH	`FINDNTH (string, substring, occurrence)`	Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument. **Note**: FINDNTH is not available for all data sources. Example: `FINDNTH("Calculation", "a", 2) = 7`
LEFT	`LEFT (string, number)`	Returns the left-most number of characters in the string. Example: `LEFT("Matador", 4) = "Mata"`
LEN	`LEN(string)`	Returns the length of the string. Example:

		`LEN("Matador") = 7`
LOWER	`LOWER (string)`	Returns `string`, with all characters lowercase. Example: `LOWER("ProductVersion") = "productversion"`
LTRIM	`LTRIM (string)`	Returns the string with any leading spaces removed. Example: `LTRIM(" Matador ") = "Matador "`
MAX	`MAX(a, b)`	Returns the maximum of `a` and `b` (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, `MAX` finds the value that is highest in the sort sequence defined by the database for that column. It returns `Null` if either argument is `Null`. Example: `MAX ("Apple","Banana") = "Banana"`

MID	`(MID (string, start, [length])`	Returns the string starting at index position `start`. The first character in the string is position 1. If the optional argument `length` is added, the returned string includes only that number of characters.
		Examples:
		`MID("Calculation", 2) = "alculation"`
		`MID("Calculation", 2, 5) ="alcul"`

MIN	`MIN(a, b)`	Returns the minimum of `a` and `b` (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, `MIN` finds the value that is lowest in the sort sequence. It returns `Null` if either argument is `Null`. Example: `MIN ("Apple","Banana") = "Apple"`
REPLACE	`REPLACE (string, substring, replace-ment)`	Searches `string` for `substring` and replaces it with `replacement`. If `substring` is not found, the string is not changed. Example: `REPLACE("Version8.5", "8.5", "9.0") = "Version9.0"`
RIGHT	`RIGHT (string, number)`	Returns the right-most number of characters in `string`. Example: `RIGHT("Calculation", 4) = "tion"`
RTRIM	`RTRIM (string)`	Returns `string` with any trailing spaces removed. Example: `RTRIM(" Calculation ") = " Calculation"`
SPACE	`SPACE(num-ber)`	Returns a string that is composed of the specified `number` of repeated spaces. Example:

```
SPACE(1) = " "
```

SPLIT	`SPLIT (string, delimiter, token number)`	Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens.
		
		The string is interpreted as an alternating sequence of delimiters and tokens. So for the string `abc-defgh-i-jkl`, where the delimiter character is '-', the tokens are abc, defgh, i, and jlk. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.
		
		Examples:
		
		`SPLIT ('a-b-c-d', '-', 2) = 'b'`
		`SPLIT ('a|b|c|d', '|', -2) = 'c'`
		
		Note: The split and custom split commands are available for the following data sources types: Tableau data extracts, Microsoft Excel, Text File, PDF File, Salesforce, OData, Microsoft Azure Market Place, Google Analytics, Vertica, Oracle, MySQL, PostgreSQL, Teradata, Amazon Redshift, Aster Data, Google Big Query, Cloudera Hadoop Hive, Hortonworks Hive, and Microsoft SQL Server.
		
		Some data sources impose limits on splitting string. The following table shows which data sources support negative token numbers (splitting from the right) and whether there is a limit on the number of splits allow per data source. A SPLIT function that specifies a negative token number and would be legal with other data sources will return this error with these data sources:

“Splitting from right is not support by the data source.”

Data Source	Left/Right Constraints	Maximum Number of Splits	Version limitations
Tableau Data Extract	Both	Infinite	
Microsoft Excel	Both	Infinite	
Text file	Both	Infinite	
Salesforce	Both	Infinite	
OData	Both	Infinite	
Google Analytics	Both	Infinite	
Tableau Data Server	Both	Infinite	Supported in version 9.0.
Vertica	Left only	10	
Oracle	Left only	10	
MySQL	Both	10	
PostgreSQL	Left only prior to version 9.0; both for version 9.0 and above	10	
Teradata	Left only	10	Version 14

			and later
Amazon Redshift	Left only	10	
Aster Database	Left only	10	
Google BigQuery	Left only	10	
Hortonworks Hadoop Hive	Left only	10	
Cloudera Hadoop	Left only	10	Impala supported starting in version 2.3.0.
Microsoft SQL Server	Both	10	2008 and later

STARTSWITH	`STARTSWITH (string, substring)`	Returns true if `string` starts with `substring`. Leading white spaces are ignored. Example: `STARTSWITH("Joker", "Jo") = true`
TRIM	`TRIM (string)`	Returns the string with leading and trailing spaces removed. Example: `TRIM(" Calculation ") = "Calculation"`

UPPER	`UPPER (string)`	Returns string, with all characters uppercase. Example: `UPPER("Calculation") = "CALCULATION"`

Date functions

Tableau provides a variety of date functions. Many of the examples use the `#` symbol with date expressions. See **Literal expression syntax** on page 2144 for an explanation of this symbol. Additionally, many date functions use `date_part`, which is a constant string argument. The valid `date_part` values that you can use are:

date_part	Values
`'year'`	Four-digit year
`'quarter'`	1-4
`'month'`	1-12 or "January", "February", and so on
`'dayofyear'`	Day of the year; Jan 1 is 1, Feb 1 is 32, and so on
`'day'`	1-31
`'weekday'`	1-7 or "Sunday", "Monday", and so on
`'week'`	1-52
`'hour'`	0-23
`'minute'`	0-59
`'second'`	0-60

Note: Date functions do not take account of the configured fiscal year start. See .**Fiscal Dates** on page 1258.

Function	Syntax	Description
DATEADD	`DATEADD (date_part, interval, date)`	Returns the specified date with the specified number `interval` added to the specified `date_part` of that date. **Supports ISO 8601 dates.** Example: `DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM` This expression adds three months to the date `#2004-04-15#`.

DATEDIFF	`DATEDIFF (date_part, date1, date2, [start_of_week])`	Returns the difference between `date1` and `date2` expressed in units of `date_part`.
		
		The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.
		
		Supports ISO 8601 dates.
		
		Examples:
		
		`DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday')= 1`
		`DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday')= 0`
		
		The first expression returns 1 because when `start_of_week` is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0

<table>
<tr><td></td><td></td><td>because when <code>start_of_week</code> is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.</td></tr>
<tr><td>DATENAME</td><td><code>DATENAME (date_part, date, [start_of_ week])</code></td><td>Returns <code>date_part</code> of <code>date</code> as a string. The <code>start_of_week</code> parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If <code>start_of_week</code> is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1247.

Supports ISO 8601 dates.

Examples:

<code>DATENAME('year', #2004-04-15#) = "2004"</code>
<code>DATENAME('month', #2004-04-15#) = "April"</code></td></tr>
<tr><td>DATEPARSE</td><td><code>DATEPARSE (date_ format, [date_ string])</code></td><td>Returns [date_string] as a date. The date_format argument will describes how the [string] field is arranged. Because of the variety of ways the string field can be ordered, the date_format must match exactly. For a full explanation, see Convert a field to a date field.

Example:

<code>DATEPARSE('yyyy-MM-dd', #2004-04-15#) = "April 4, 2004"</code>

Note: This function is available through the following connectors: non-legacy Excel and</td></tr>
</table>

<table>
<tr>
<td></td>
<td></td>
<td>
text file connections, Amazon EMR Hadoop Hive, Cloudera Hadoop, Google Sheets, Hortonworks Hadoop Hive, MapR Hadoop Hive, MySQL, Oracle, PostgreSQL, and Tableau extracts. Some formats may not be available for all connections.
Note: DATEPARSE is not supported on Hive variants. Only Denodo, Drill, and Snowflake are supported.
</td>
</tr>
<tr>
<td>DATEPART</td>
<td><code>DATEPART (date_part, date, [start_of_week])</code></td>
<td>
Returns <code>date_part</code> of <code>date</code> as an integer.
The <code>start_of_week</code> parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If <code>start_of_week</code> is omitted, the start of week is determined by the data source. See Date Properties for a Data Source on page 1247.
Note: When the <code>date_part</code> is weekday, the <code>start_of_week</code> parameter is ignored. This is because Tableau relies on a fixed weekday ordering to apply offsets.
Supports ISO 8601 dates.
Examples:
<pre>
DATEPART('year', #2004-04-15#)
= 2004
DATEPART('month', #2004-04-15#)
= 4
</pre>
</td>
</tr>
</table>

DATETRUNC	`DATETRUNC (date_part, date, [start_of_ week])`	Truncates the specified date to the accuracy specified by the date_part. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.
		Supports ISO 8601 dates.
		Examples:
		`DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM`
		`DATETRUNC('month', #2004-04-15#) = 2004-04-01 12:00:00 AM`

DAY	`DAY(date)`	Returns the day of the given date as an integer. Example: `DAY(#2004-04-12#) = 12`
ISDATE	`ISDATE (string)`	Returns true if a given string is a valid date. Example: `ISDATE("April 15, 2004") = true`
MAKEDATE	`MAKEDATE (year, month, day)`	Returns a date value constructed from the specified year, month, and date. Available for Tableau Data Extracts. Check for

		availability in other data sources. Example: `MAKEDATE(2004, 4, 15) = #April 15, 2004#`

MAKEDATETIME	`MAKEDATETIME (date, time)`	Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime.
		
		Note: This function is available only for MySQL-compatible connections (which for Tableau are MySQL and Amazon Aurora).
		
		Examples:
		
		`MAKEDATETIME("1899-12-30", #07:59:00#) = #12/30/1899 7:59:00 AM#`
		`MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#`

MAKETIME	`MAKETIME (hour, minute, second)`	Returns a date value constructed from the specified hour, minute, and second. Available for Tableau Data Extracts. Check for availability in other data sources. Example: `MAKETIME(14, 52, 40) = #14:52:40#`
MAX	`MAX (expression) or MAX`	Usually applied to numbers but also works on dates. Returns the maximum of `a` and `b` (`a` and `b` must be of the same type). Returns `Null` if either argument

	`(expr1, expr2)`	is `Null`.
		Examples:
		`MAX(#2004-01-01# ,#2004-03-01#) = 2004-03-01 12:00:00 AM`
		`MAX([ShipDate1], [ShipDate2])`

MIN	`MIN (expression) or MIN (expr1, expr2)`	Usually applied to numbers but also works on dates. Returns the minimum of `a` and `b` (`a` and `b` must be of the same type). Returns `Null` if either argument is `Null`.
		Examples:
		`MIN(#2004-01-01# ,#2004-03-01#) = 2004-01-01 12:00:00 AM`
		`MIN([ShipDate1], [ShipDate2])`

MONTH	`MONTH(date)`	Returns the month of the given date as an integer. Example: `MONTH(#2004-04-15#) = 4`
NOW	`NOW( )`	Returns the current local system date and time. Example: `NOW( ) = 2004-04-15 1:08:21 PM`
QUARTER	`QUARTER ( )`	Returns the quarter of the given date as an integer. Example: `WEEK (#2004-04-15#) = 16`
TODAY	`TODAY( )`	Returns the current date.

		Example: `TODAY( ) = 2004-04-15`
WEEK	`WEEK( )`	Returns the week of the given date as an integer. Example: `WEEK (#2004-04-15#) = 16`
YEAR	`YEAR (date)`	Returns the year of the given date as an integer. Example: `YEAR(#2004-04-15#) = 2004`
ISOQUARTER	`ISOQUARTER (date)`	Returns the ISO8601 week-based quarter of a given date as an integer. Example: `ISOQUARTER (#2005-03-29#) = 2`
ISOWEEK	`ISOWEEK (date)`	Returns the ISO8601 week-based week of a given date as an integer. Example: `ISOWEEK (#2004-03-29#) = 14`
ISOWEEKDAY	`ISOWEEKDAY (date)`	Returns the ISO8601 week-based weekday of a given date as an integer. Example: `ISOWEEKDAY (#2004-03-29#) = 1`
ISOYEAR	`ISOYEAR`	Returns the ISO8601 week-based year of a given

	`(date)`	date as an integer. Example: `ISOYEAR (#2003-12-29#) = 2004`

Logical functions

Function	Syntax	Description

IN	`<expr1> IN <expr2>`	Returns TRUE if any value in `<expr1>` matches any value in `<expr2>`.
		The values in `<expr1>` can be a Set, list of literal values, or a combined field.
		Examples:
		`SUM([Cost]) IN (1000, 15, 200)`
		`[SET] IN [COMBINED FIELD]`

AND	`IF <expr1> AND <expr2> THEN <then> END`	Performs a logical conjunction on two expressions. Example: `IF (ATTR([Market]) = "Africa" AND SUM([Sales]) > [Emerging Threshold] )THEN "Well Performing"`
CASE	`CASE <expression> WHEN <value1> THEN <return1>`	Performs logical tests and returns appropriate values. The CASE function evaluates `expression`, compares it to a sequence of values, `value1`, `value2`, etc., and returns a result. When a value that matches `expression` is encountered, CASE returns the corresponding return value. If no match is found, the default return expression is

	`WHEN <value2> THEN <return2> ... ELSE <default return> END`	used. If there is no default return and no values match, then Null is returned.
		
		CASE is often easier to use than IIF or IF THEN ELSE.
		
		Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function , although the CASE function will generally be more concise.
		
		Many times you can use a group to get the same results as a complicated case function.
		
		Examples:
		
		`CASE [Region] WHEN 'West' THEN 1 WHEN 'East' THEN 2 ELSE 3 END`
		
		`CASE LEFT(DATENAME('weekday',[Order Date]),3) WHEN 'Sun' THEN 0 WHEN 'Mon' THEN 1 WHEN 'Tue' THEN 2 WHEN 'Wed' THEN 3 WHEN 'Thu' THEN 4 WHEN 'Fri' THEN 5 WHEN 'Sat' THEN 6 END`

ELSE	`IF <expr> THEN <then> ELSE <else> END`	Tests a series of expressions returning the <then> value for the first true <expr>. Example: `If [Profit] > 0 THEN 'Profitable' ELSE 'Loss' END`
ELSEIF	`IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...]`	Tests a series of expressions returning the <then> value for the first true <expr>. Example:

	[ELSE <else>] END	IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END
END	IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...] [ELSE <else>] END	Tests a series of expressions returning the <then> value for the first true <expr>. Must be placed at the end of an expression. Example: IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END
IF	IF <expr> THEN <then> [ELSEIF <expr2> THEN <then2>...] [ELSE <else>] END	Tests a series of expressions returning the <then> value for the first true <expr>. Example: IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Breakeven' ELSE 'Loss' END
IFNULL	IFNULL (expr1, expr2)	Returns <expr1> if it is not null, otherwise returns <expr2>. Example: IFNULL([Profit], 0)
IIF	IIF(test, then, else, [unknown])	Checks whether a condition is met, and returns one value if TRUE, another value if FALSE, and an optional third value or NULL if unknown. Example: IIF([Profit] > 0, 'Profit', 'Loss')

ISDATE	`ISDATE(string)`	Returns true if a given string is a valid date. Example: `ISDATE("2004-04-15") = True`
ISNULL	`ISNULL(expression)`	Returns true if the expression is NULL (does not contain valid data). Example: `ISNULL([Profit])`
MAX	`MAX(expression)` or `Max(expr1, expr2)`	Returns the maximum of a single expression across all records or the maximum of two expressions for each record. Example: `MAX([Sales])`
MIN	`MIN(expression)` or `MIN(expr1, expr2)`	Returns the minimum of an expression across all records or the minimum of two expressions for each record. Example: `MIN([Profit])`
NOT	`IF NOT <expr> THEN <then> END`	Performs logical negation on an expression. Example: `IF NOT [Profit] > 0 THEN "Unprofitable" END`
OR	`IF <expr1> OR <expr2> THEN <then>`	Performs a logical disjunction on two expressions. Example:

	`END`	`IF [Profit] < 0 OR [Profit] = 0 THEN "Needs Improvement" END`
THEN	`IF <expre> THEN <then> [ELSEIF ,expr2> THEN <then2>...] [ELSE <else>] END`	Tests a series of expressions returning the <then> value for the first true <expr>. Example: `IF [Profit] > 0 THEN 'Profitable' ELSEIF [Profit] = 0 THEN 'Break even' ELSE 'unprofitable' END`
WHEN	`CASE <expr> WHEN <Value1> THEN <return1> ... [ELSE <else>] END`	Finds the first <value> that matches <expr> and returns the corresponding <return>. Example: `CASE [RomanNumberal] WHEN 'I' THEN 1 WHEN 'II' THEN 2 ELSE 3 END`
ZN	`ZN(expres-sion)`	Returns <expression> if it is not null, otherwise returns zero. Example: `ZN([Profit])`

Aggregate functions

Aggregations and floating-point arithmetic: The results of some aggregations may not always be exactly as expected. For example, you may find that the Sum function returns a value such as -1.42e-14 for a column of numbers that you know should sum to exactly 0. This happens because the Institute of Electrical and Electronics Engineers (IEEE) 754 floating-point standard requires that numbers be stored in binary format, which means that numbers are sometimes rounded at extremely fine levels of precision. You can eliminate this potential distraction by using the ROUND function (see **Number Functions** on page 2147) or by formatting the number to show fewer decimal places.

Function	Syntax	Definition
ATTR	`ATTR (expression)`	Returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.
AVG	`AVG (expression)`	Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored.
COLLECT	`COLLECT (spatial)`	An aggregate calculation that combines the values in the argument field. Null values are ignored. **Note**: The COLLECT function can only be used with spatial fields. Example: `COLLECT ([Geometry])`
CORR	`CORR (expression 1, expression2)`	Returns the Pearson correlation coefficient of two expressions. The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship. CORR is available with the following data sources: • Tableau data extracts (you can create an extract from any data source) • Cloudera Hive • EXASolution • Firebird (version 3.0 and later)

		• Google BigQuery • Hortonworks Hadoop Hive • IBM PDA (Netezza) • Oracle • PostgreSQL • Presto • SybaseIQ • Teradata • Vertica For other data sources, consider either extracting the data or using WINDOW_CORR. See **Table Calculation Functions** on page 2219. **Note**: The square of a CORR result is equivalent to the R-Squared value for a linear trend line model. See **Trend Line Model Terms** on page 2537. Example: You can use CORR to visualize correlation in a disaggregated scatter plot. The way to do this is to use a table-scoped level of detail expression. For example: `{CORR(Sales, Profit)}` With a level of detail expression, the correlation is run over all rows. If you used a formula like `CORR(Sales, Profit)` (without the surrounding brackets to make it a level of detail expression), the view would show the correlation of each individual point in the scatter plot with each other point, which is undefined. See **Table-Scoped** on page 2461
COUNT	`COUNT (expression)`	Returns the number of items in a group. Null values are not counted.
COUNTD	`COUNTD`	Returns the number of distinct items in a group. Null val-

	`(expression)`	ues are not counted. This function is not available in the following cases: workbooks created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, and workbooks that use Microsoft Access data sources. Extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.
COVAR	`COVAR (expression 1, expression2)`	Returns the *sample covariance* of two expressions. Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (available with the COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population. COVAR is available with the following data sources: • Tableau data extracts (you can create an extract from any data source) • Cloudera Hive • EXASolution • Firebird (version 3.0 and later) • Google BigQuery • Hortonworks Hadoop Hive • IBM PDA (Netezza) • Oracle • PostgreSQL • Presto • SybaseIQ • Teradata • Vertica

<table>
<tr><td></td><td></td><td>
For other data sources, consider either extracting the data or using WINDOW_COVAR. See Table Calculation Functions on page 2219.
If expression1 and expression2 are the same—for example, COVAR([profit], [profit])—COVAR returns a value that indicates how widely values are distributed.
Note: The value of COVAR(X, X) is equivalent to the value of VAR(X) and also to the value of STDEV(X)^2.
Example:
The following formula returns the sample covariance of Sales and Profit.
<pre><code>COVAR([Sales], [Profit])</code></pre>
</td></tr>
<tr><td>COVARP</td><td><code>COVARP (expression 1, expression2)</code></td><td>
Returns the population covariance of two expressions.
Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the COVAR function) is appropriate.
COVARP is available with the following data sources:

Tableau data extracts (you can create an extract from any data source)
Cloudera Hive

</td></tr>
</table>

		• EXASolution • Firebird (version 3.0 and later) • Google BigQuery • Hortonworks Hadoop Hive • IBM PDA (Netezza) • Oracle • PostgreSQL • Presto • SybaseIQ • Teradata • Vertica For other data sources, consider either extracting the data or using WINDOW_COVARP. See **Table Calculation Functions** on page 2219. If expression1 and expression2 are the same—for example, COVARP([profit], [profit])—COVARP returns a value that indicates how widely values are distributed. **Note**: The value of COVARP(X, X) is equivalent to the value of VARP(X) and also to the value of STDEVP(X)^2. Example: The following formula returns the population covariance of **Sales** and **Profit**. `COVARP([Sales], [Profit])`
MAX	`MAX (expression)`	Returns the maximum of an expression across all records. If the expression is a string value, this function returns the last value where last is defined by alphabetical order.
MEDIAN	`MEDIAN (expression)`	Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. This function is not available for workbooks

		created before Tableau Desktop 8.2 or that use legacy connections. It is also not available for connections using any of the following data sources: • Access • Amazon Redshift • Cloudera Hadoop • HP Vertica • IBM DB2 • IBM PDA (Netezza) • Microsoft SQL Server • MySQL • SAP HANA • Teradata For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.
MIN	`MIN (expression)`	Returns the minimum of an expression across all records. If the expression is a string value, this function returns the first value where first is defined by alphabetical order.
PERCENTILE	`PERCENTILE (expression, number)`	Returns the percentile value from the given expression corresponding to the specified number. The number must be between 0 and 1 (inclusive)—for example, 0.66, and must be a numeric constant. This function is available for the following data sources. • Non-legacy Microsoft Excel and Text File connections. • Extracts and extract-only data source types (for example, Google Analytics, OData, or Salesforce). • Sybase IQ 15.1 and later data sources. • Oracle 10 and later data sources. • Cloudera Hive and Hortonworks Hadoop Hive data sources.

		• EXASolution 4.2 and later data sources. For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.
STDEV	`STDEV (expression)`	Returns the statistical standard deviation of all values in the given expression based on a sample of the population.
STDEVP	`STDEVP (expression)`	Returns the statistical standard deviation of all values in the given expression based on a biased population.
SUM	`SUM (expression)`	Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored.
VAR	`VAR (expression)`	Returns the statistical variance of all values in the given expression based on a sample of the population.
VARP	`VARP (expression)`	Returns the statistical variance of all values in the given expression on the entire population.

User functions

Function	Syntax	Description
FULLNAME	`FULLNAME( )`	Returns the full name for the current user. This is the Tableau Server or Tableau Online full name when the user is signed in; otherwise the local or network full name for the Tableau Desktop user. Example: `[Manager]=FULLNAME( )` If manager Dave Hallsten is signed in, this example returns True only if the Manager field in the view contained Dave Hallsten. When used as a filter, this calculated field

		can be used to create a user filter that only shows data that is relevant to the person signed in to the server.
ISFULLNAME	`ISFULLNAME(string)`	Returns true if the current user's full name matches the specified full name, or false if it does not match. This function uses the Tableau Server or Online full name when the user is signed in; otherwise it uses the local or network full name for the Tableau Desktop user. Example: `ISFULLNAME(“Dave Hallsten”)` This example returns true if Dave Hallsten is the current user, otherwise it returns false.
ISMEMBEROF	`ISMEMBEROF(string)`	Returns true if the person currently using Tableau is a member of a group that matches the given string. If the person currently using Tableau is signed in, the group membership is determined by groups on Tableau Server or Tableau Online. If the person is not signed in, this function returns NULL. **Note**: The function will return a "True" value if the given string is "All Users", whether signed in to Tableau Server or Tableau Online. The ISMEMBEROF() function will also accept Active Directory domains. The Active Directory domain must be declared in the calculation with the group name. Example: `IF ISMEMBEROF(‘domain.lan\Sales’) THEN “Sales” ELSE “Other” END`
ISUSERNAME	`ISUSERNAME(string)`	Returns true if the current user's username matches the specified username, or false if it does not match. This

		function uses the Tableau Server or Online username when the user is signed in; otherwise it uses the local or network username for the Tableau Desktop user. Example: `ISUSERNAME(“dhallsten”)` This example returns true if dhallsten is the current user; otherwise it returns false. Note: "All Users" will always return as true.
USERDOMAIN	`USERDOMAIN()`	Returns the domain for the current user when the user is signed on to Tableau Server. Returns the Windows domain if the Tableau Desktop user is on a domain. Otherwise this function returns a null string. Example: `[Manager]=USERNAME() AND [Domain]=USERDOMAIN()`
USERNAME	`USERNAME( )`	Returns the username for the current user. This is the Tableau Server or Tableau Online username when the user is signed in; otherwise it is the local or network username for the Tableau Desktop user. Example: `[Manager]=USERNAME( )` If the manager dhallsten was signed in, this function would only return True when the Manager field in the view is dhallsten. When used as a filter this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.

Table calculations

FIRST( )

Returns the number of rows from the current row to the first row in the partition. For example, the view below shows quarterly sales. When FIRST() is computed within the Date partition, the offset of the first row from the second row is -1.

Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

First()

$160,877	0
$197,213	-1
$302,678	-2
$297,208	-3
$180,609	-4
$195,785	-5
$116,613	-6

Example

When the current row index is 3, `FIRST() = -2`.

INDEX( )

Returns the index of the current row in the partition, without any sorting with regard to value. The first row index starts at 1. For example, the table below shows quarterly sales. When INDEX() is computed within the Date partition, the index of each row is 1, 2, 3, 4..., etc.

Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

INDEX()

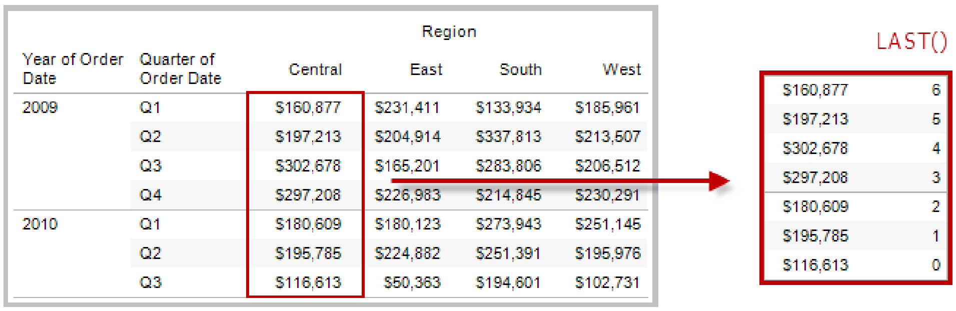
$160,877	1
$197,213	2
$302,678	3
$297,208	4
$180,609	5
$195,785	6
$116,613	7

Example

For the third row in the partition, `INDEX() = 3`.

LAST( )

Returns the number of rows from the current row to the last row in the partition. For example, the table below shows quarterly sales. When LAST() is computed within the Date partition, the offset of the last row from the second row is 5.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

LAST()

$160,877	6
$197,213	5
$302,678	4
$297,208	3
$180,609	2
$195,785	1
$116,613	0

Example

When the current row index is 3 of 7, `LAST() = 4`.

LOOKUP(expression, [offset])

Returns the value of the expression in a target row, specified as a relative offset from the current row. Use FIRST() + n and LAST() - n as part of your offset definition for a target relative to the first/last rows in the partition. If `offset` is omitted, the row to compare to can be set on the field menu. This function returns NULL if the target row cannot be determined.

The view below shows quarterly sales. When `LOOKUP (SUM(Sales), 2)` is computed within the Date partition, each row shows the sales value from 2 quarters into the future.

Year of Order Date	Quarter of Order Date	Central	East	South	West
		Region			
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

+2

Year of Order Date	Quarter of Order Date	Central	East	South	West
		Region			
2009	Q1	$302,678	$165,201	$283,806	$206,512
	Q2	$297,208	$226,983	$214,845	$230,291
	Q3	$180,609	$180,123	$273,943	$251,145
	Q4	$195,785	$224,882	$251,391	$195,976
2010	Q1	$116,613	$50,363	$194,601	$102,731
	Q2				
	Q3				

Example

`LOOKUP(SUM([Profit]), FIRST()+2)` computes the SUM(Profit) in the third row of the partition.

MODEL_PERCENTILE(target_expression, predictor_expression(s))

Returns the probability (between 0 and 1) of the expected value being less than or equal to the observed mark, defined by the target expression and other predictors. This is the Posterior Predictive Distribution Function, also known as the Cumulative Distribution Function (CDF).

This function is the inverse of MODEL_QUANTILE. For information on predictive modeling functions, see **How Predictive Modeling Functions Work in Tableau** on page 2606.

Example

The following formula returns the quantile of the mark for sum of sales, adjusted for count of orders.

```
MODEL_PERCENTILE(SUM([Sales]), COUNT([Orders]))
```

MODEL_QUANTILE(quantile, target_expression, predictor_expression(s))

Returns a target numeric value within the probable range defined by the target expression and other predictors, at a specified quantile. This is the Posterior Predictive Quantile.

This function is the inverse of MODEL_PERCENTILE. For information on predictive modeling functions, see **How Predictive Modeling Functions Work in Tableau** on page 2606.

Example

The following formula returns the median (0.5) predicted sum of sales, adjusted for count of orders.

```
MODEL_QUANTILE(0.5, SUM([Sales]), COUNT([Orders]))
```

PREVIOUS_VALUE(expression)

Returns the value of this calculation in the previous row. Returns the given expression if the current row is the first row of the partition.

Example

`SUM([Profit]) * PREVIOUS_VALUE(1)` computes the running product of SUM(Profit).

RANK(expression, ['asc' | 'desc'])

Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

Example

The following image shows the effect of the various ranking functions (RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, and RANK_UNIQUE) on a set of values. The data set contains information on 14 students (StudentA through StudentN); the **Age** column shows the current age of each student (all students are between 17 and 20 years of age). The remaining columns show the effect of each rank function on the set of age values, always assuming the default order (ascending or descending) for the function.

Student	Age	RANKofAge	RANK_DENSEofAge	RANK_MODIFIEDofAge	RANK_PERCENTILEofAge	RANK_UNIQUEofAge
StudentA	19	4	2	7	79%	4
StudentB	18	8	3	12	50%	8
StudentC	19	4	2	7	79%	5
StudentD	18	8	3	12	50%	9
StudentE	17	13	4	14	14%	13
StudentF	18	8	3	12	50%	10
StudentG	19	4	2	7	79%	6
StudentH	20	1	1	3	100%	1
StudentI	19	4	2	7	79%	7
StudentJ	20	1	1	3	100%	2
StudentK	20	1	1	3	100%	3
StudentL	17	13	4	14	14%	14
StudentM	18	8	3	12	50%	11
StudentN	18	8	3	12	50%	12

RANK_DENSE(expression, ['asc' | 'desc'])

Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (3, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_MODIFIED(expression, ['asc' | 'desc'])

Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 3, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_PERCENTILE(expression, ['asc' | 'desc'])

Returns the percentile rank for the current row in the partition. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is ascending.

With this function, the set of values (6, 9, 9, 14) would be ranked (0.00, 0.67, 0.67, 1.00).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_UNIQUE(expression, ['asc' | 'desc'])

Returns the unique rank for the current row in the partition. Identical values are assigned different ranks. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 3, 1).

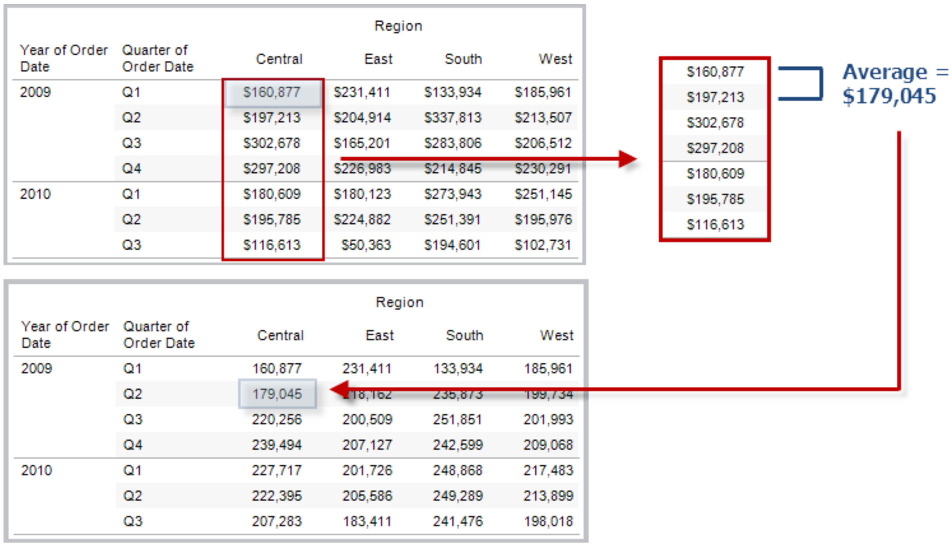
Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RUNNING_AVG(expression)

Returns the running average of the given expression, from the first row in the partition to the current row.

The view below shows quarterly sales. When `RUNNING_AVG(SUM([Sales])` is computed within the Date partition, the result is a running average of the sales values for each quarter.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	179,045	[illegible]	[illegible]	[illegible]
	Q3	220,256	200,509	251,851	201,993
	Q4	239,494	207,127	242,599	209,068
2010	Q1	227,717	201,726	248,868	217,483
	Q2	222,395	205,586	249,289	213,899
	Q3	207,283	183,411	241,476	198,018

Example

`RUNNING_AVG(SUM([Profit]))` computes the running average of SUM(Profit).

RUNNING_COUNT(expression)

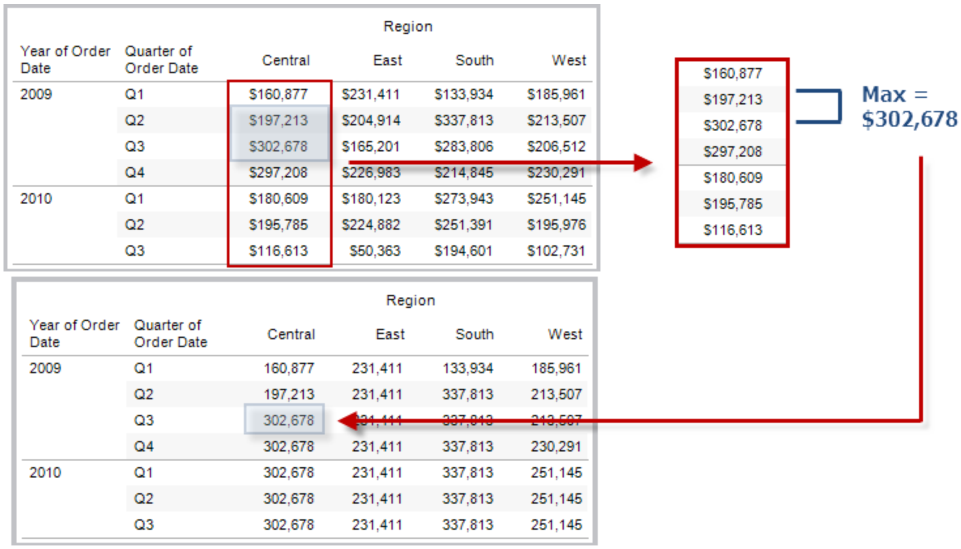
Returns the running count of the given expression, from the first row in the partition to the current row.

Example

`RUNNING_COUNT(SUM([Profit]))` computes the running count of SUM(Profit).

RUNNING_MAX(expression)

Returns the running maximum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

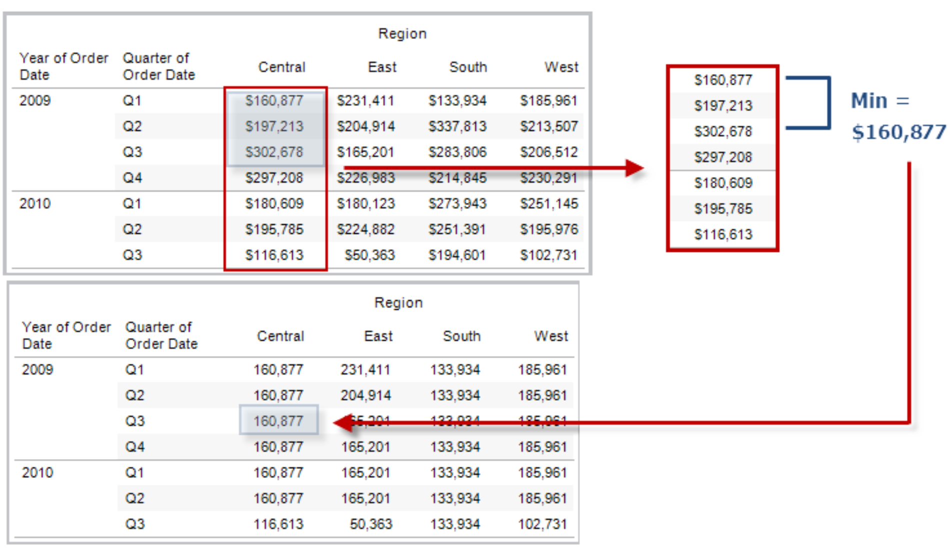
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	197,213	231,411	337,813	213,507
	Q3	302,678	231,411	337,813	213,507
	Q4	302,678	231,411	337,813	230,291
2010	Q1	302,678	231,411	337,813	251,145
	Q2	302,678	231,411	337,813	251,145
	Q3	302,678	231,411	337,813	251,145

Example

`RUNNING_MAX(SUM([Profit]))` computes the running maximum of SUM(Profit).

RUNNING_MIN(expression)

Returns the running minimum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

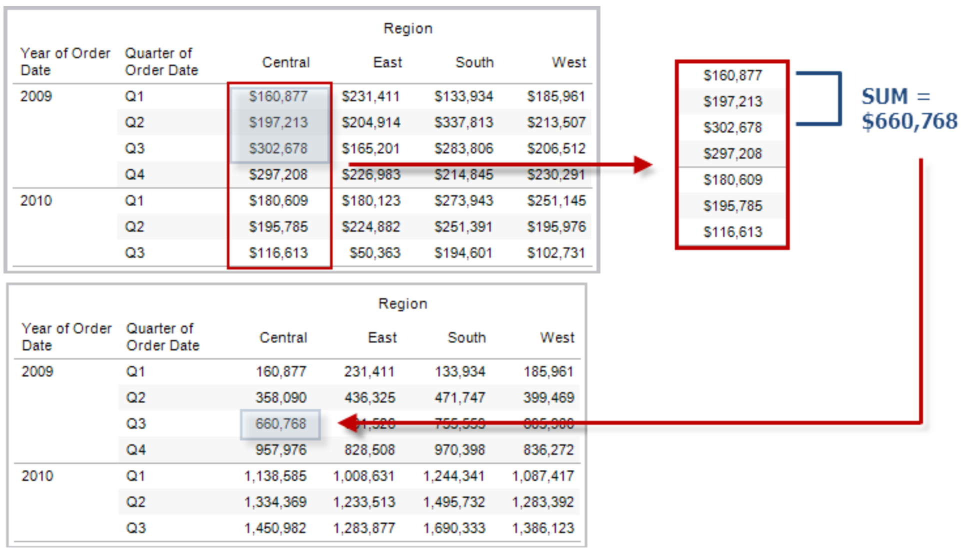
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	160,877	204,914	133,934	185,961
	Q3	160,877	165,201	133,934	185,961
	Q4	160,877	165,201	133,934	185,961
2010	Q1	160,877	165,201	133,934	185,961
	Q2	160,877	165,201	133,934	185,961
	Q3	116,613	50,363	133,934	102,731

Example

`RUNNING_MIN(SUM([Profit]))` computes the running minimum of SUM(Profit).

RUNNING_SUM(expression)

Returns the running sum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

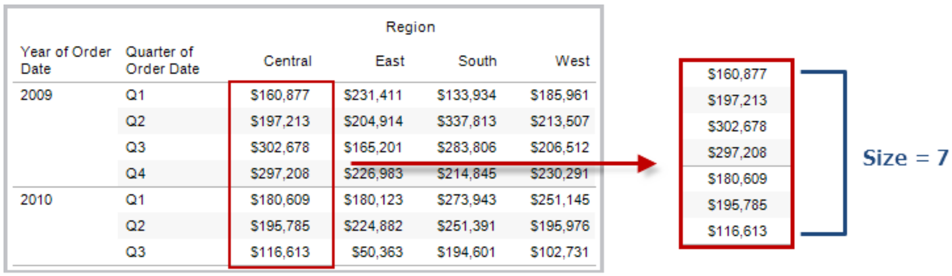
Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	358,090	436,325	471,747	399,469
	Q3	660,768	[illegible]	[illegible]	[illegible]
	Q4	957,976	828,508	970,398	836,272
2010	Q1	1,138,585	1,008,631	1,244,341	1,087,417
	Q2	1,334,369	1,233,513	1,495,732	1,283,392
	Q3	1,450,982	1,283,877	1,690,333	1,386,123

Example

`RUNNING_SUM(SUM([Profit]))` computes the running sum of SUM(Profit)

SIZE()

Returns the number of rows in the partition. For example, the view below shows quarterly sales. Within the Date partition, there are seven rows so the Size() of the Date partition is 7.



Example

`SIZE() = 5` when the current partition contains five rows.

SCRIPT_BOOL

Returns a Boolean result from the specified expression. The expression is passed directly to a running analytics extension service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.).

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_BOOL("is.finite(.arg1)", SUM([Profit]))
```

The next example returns True for store IDs in Washington state, and False otherwise. This example could be the definition for a calculated field titled IsStoreInWA.

```
SCRIPT_BOOL('grepl(".*_WA", .arg1, perl=TRUE)',ATTR([Store ID]))
```

A command for Python would take this form:

```
SCRIPT_BOOL("return map(lambda x : x > 0, _arg1)", SUM([Profit]))
```

SCRIPT_INT

Returns an integer result from the specified expression. The expression is passed directly to a running analytics extension service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_INT("is.finite(.arg1)", SUM([Profit]))
```

In the next example, k-means clustering is used to create three clusters:

```
SCRIPT_INT('result <- kmeans(data.frame(.arg1,.arg2,.arg3,.arg4),
3);result$cluster;', SUM([Petal length]), SUM([Petal width]),SUM
([Sepal length]),SUM([Sepal width]))
```

A command for Python would take this form:

```
SCRIPT_INT("return map(lambda x : int(x * 5), _arg1)", SUM
([Profit]))
```

SCRIPT_REAL

Returns a real result from the specified expression. The expression is passed directly to a running analytics extension service instance. In

R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_REAL("is.finite(.arg1)", SUM([Profit]))
```

The next example converts temperature values from Celsius to Fahrenheit.

```
SCRIPT_REAL('library(udunits2);ud.convert(.arg1, "celsius",
"degree_fahrenheit")',AVG([Temperature]))
```

A command for Python would take this form:

```
SCRIPT_REAL("return map(lambda x : x * 0.5, _arg1)", SUM
([Profit]))
```

SCRIPT_STR

Returns a string result from the specified expression. The expression is passed directly to a running analytics extension service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_STR("is.finite(.arg1)", SUM([Profit]))
```

The next example extracts a state abbreviation from a more complicated string (in the original form 13XSL_CA, A13_WA):

```
SCRIPT_STR('gsub(".*_", "", .arg1)',ATTR([Store ID]))
```

A command for Python would take this form:

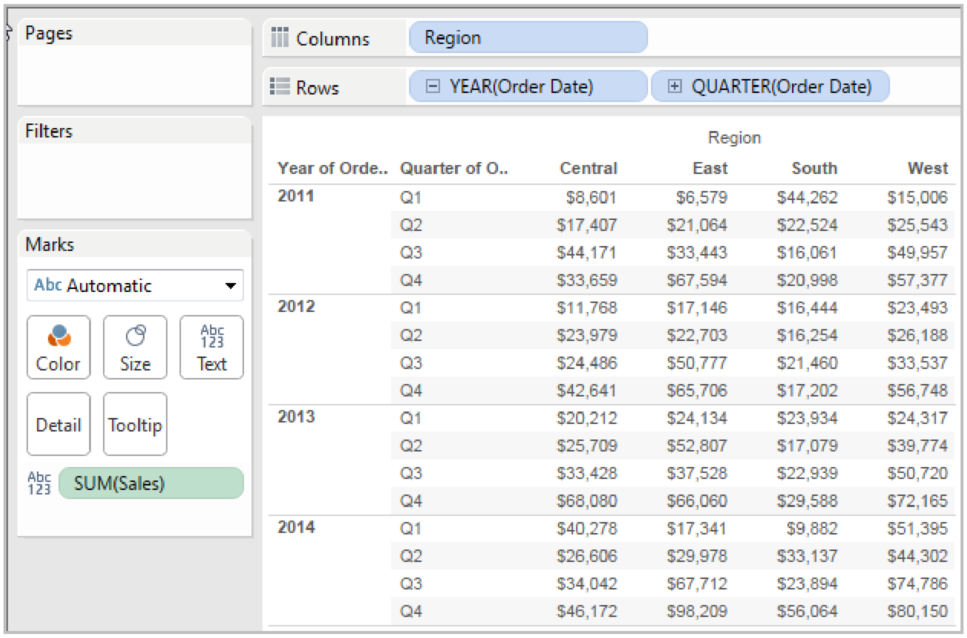
```
SCRIPT_STR("return map(lambda x : x[:2], _arg1)", ATTR([Region]))
```

TOTAL(expression)

Returns the total for the given expression in a table calculation partition.

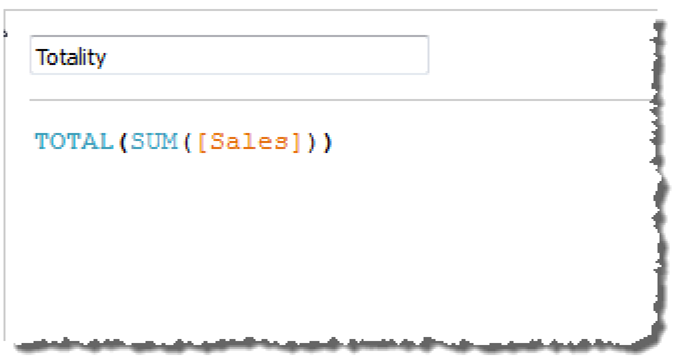
Example

Assume you are starting with this view:

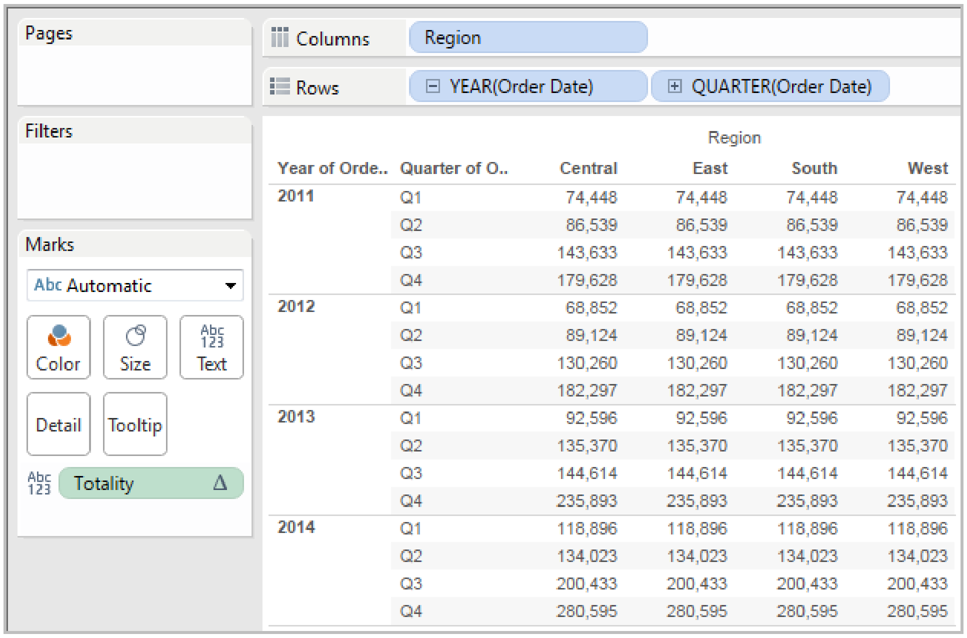


		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	$8,601	$6,579	$44,262	$15,006
	Q2	$17,407	$21,064	$22,524	$25,543
	Q3	$44,171	$33,443	$16,061	$49,957
	Q4	$33,659	$67,594	$20,998	$57,377
2012	Q1	$11,768	$17,146	$16,444	$23,493
	Q2	$23,979	$22,703	$16,254	$26,188
	Q3	$24,486	$50,777	$21,460	$33,537
	Q4	$42,641	$65,706	$17,202	$56,748
2013	Q1	$20,212	$24,134	$23,934	$24,317
	Q2	$25,709	$52,807	$17,079	$39,774
	Q3	$33,428	$37,528	$22,939	$50,720
	Q4	$68,080	$66,060	$29,588	$72,165
2014	Q1	$40,278	$17,341	$9,882	$51,395
	Q2	$26,606	$29,978	$33,137	$44,302
	Q3	$34,042	$67,712	$23,894	$74,786
	Q4	$46,172	$98,209	$56,064	$80,150

You open the calculation editor and create a new field which you name **Totality**:

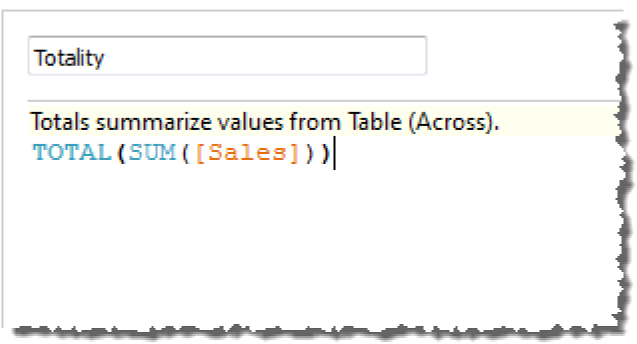


You then drop **Totality** on Text, to replace **SUM(Sales)**. Your view changes such that it sums values based on the default **Compute Using** value:



		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	74,448	74,448	74,448	74,448
	Q2	86,539	86,539	86,539	86,539
	Q3	143,633	143,633	143,633	143,633
	Q4	179,628	179,628	179,628	179,628
2012	Q1	68,852	68,852	68,852	68,852
	Q2	89,124	89,124	89,124	89,124
	Q3	130,260	130,260	130,260	130,260
	Q4	182,297	182,297	182,297	182,297
2013	Q1	92,596	92,596	92,596	92,596
	Q2	135,370	135,370	135,370	135,370
	Q3	144,614	144,614	144,614	144,614
	Q4	235,893	235,893	235,893	235,893
2014	Q1	118,896	118,896	118,896	118,896
	Q2	134,023	134,023	134,023	134,023
	Q3	200,433	200,433	200,433	200,433
	Q4	280,595	280,595	280,595	280,595

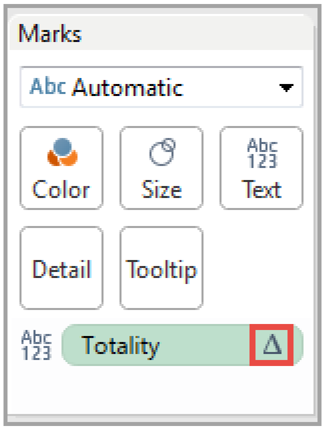
This raises the question, What is the default **Compute Using** value? If you right-click (Control-click on a Mac) **Totality** in the Data pane and choose **Edit**, there is now an additional bit of information available:



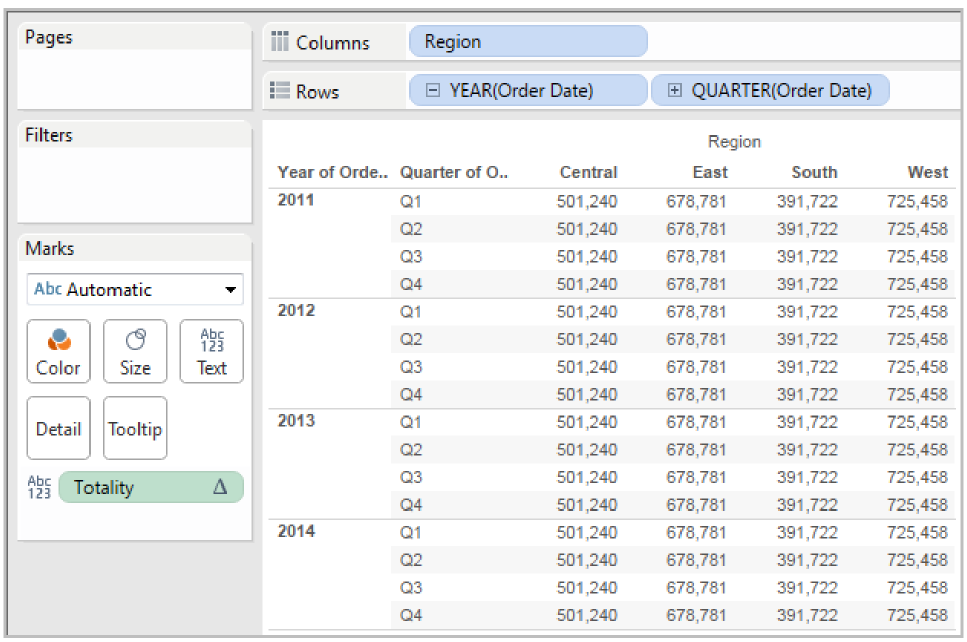
The default **Compute Using** value is **Table (Across)**. The result is that **Totality** is summing the values across each row of your table. Thus, the value that you see across each row is the sum of the values from the original version of the table.

The values in the 2011/Q1 row in the original table were $8601, $6579, $44262, and $15006. The values in the table after **Totality** replaces **SUM(Sales)** are all $74,448, which is the sum of the four original values.

Notice the triangle next to Totality after you drop it on Text:



This indicates that this field is using a table calculation. You can right-click the field and choose **Edit Table Calculation** to redirect your function to a different **Compute Using** value. For example, you could set it to **Table (Down)**. In that case, your table would look like this:

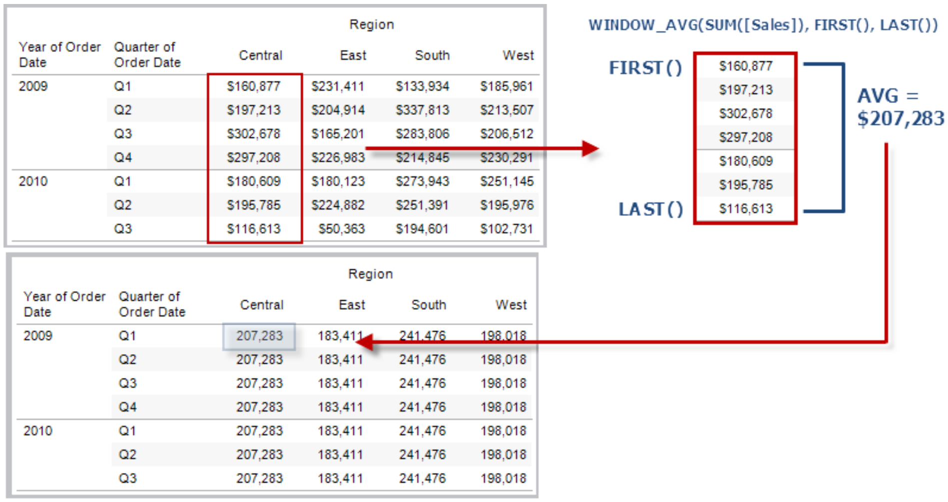


		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2012	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2013	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2014	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458

WINDOW_AVG(expression, [start, end])

Returns the average of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window average within the Date partition returns the average sales across all dates.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	207,283	183,411	241,476	198,018
	Q2	207,283	183,411	241,476	198,018
	Q3	207,283	183,411	241,476	198,018
	Q4	207,283	183,411	241,476	198,018
2010	Q1	207,283	183,411	241,476	198,018
	Q2	207,283	183,411	241,476	198,018
	Q3	207,283	183,411	241,476	198,018

Example

`WINDOW_AVG(SUM([Profit]), FIRST()+1, 0)` computes the average of SUM(Profit) from the second row to the current row.

WINDOW_CORR(expression1, expression2, [start, end])

Returns the Pearson correlation coefficient of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship.

There is an equivalent aggregation fuction: CORR. See **Tableau Functions (Alphabetical)** on page 2342.

Example

The following formula returns the Pearson correlation of **SUM(Profit)** and **SUM(Sales)** from the five previous rows to the current row.

```
WINDOW_CORR(SUM[Profit]), SUM([Sales]), -5, 0)
```

WINDOW_COUNT(expression, [start, end])

Returns the count of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0)` computes the count of SUM (Profit) from the second row to the current row

WINDOW_COVAR(expression1, expression2, [start, end])

Returns the *sample covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end arguments are omitted, the window is the entire partition.

Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (with the WINDOW_COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

There is an equivalent aggregation fuction: COVAR. See **Tableau Functions (Alphabetical)** on page 2342.

Example

The following formula returns the sample covariance of **SUM(Profit)** and **SUM(Sales)** from the two previous rows to the current row.

```
WINDOW_COVAR(SUM([Profit]), SUM([Sales]), -2, 0)
```

WINDOW_COVARP(expression1, expression2, [start, end])

Returns the *population covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the WINDOW_COVAR function) is appropriate.

There is an equivalent aggregation fuction: COVARP. **Tableau Functions (Alphabetical)** on page 2342.

Example

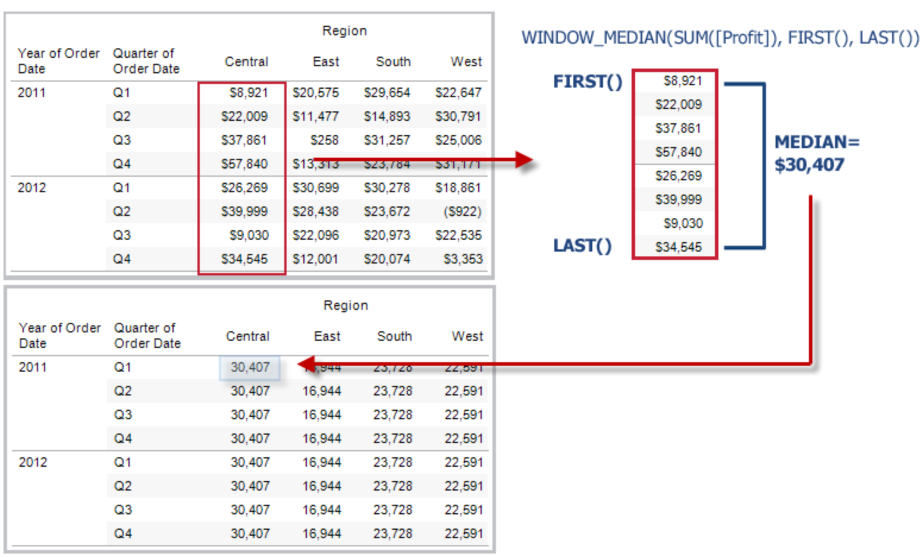
The following formula returns the population covariance of **SUM(Profit)** and **SUM(Sales)** from the two previous rows to the current row.

```
WINDOW_COVARP(SUM([Profit]), SUM([Sales]), -2, 0)
```

WINDOW_MEDIAN(expression, [start, end])

Returns the median of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly profit. A window median within the Date partition returns the median profit across all dates.



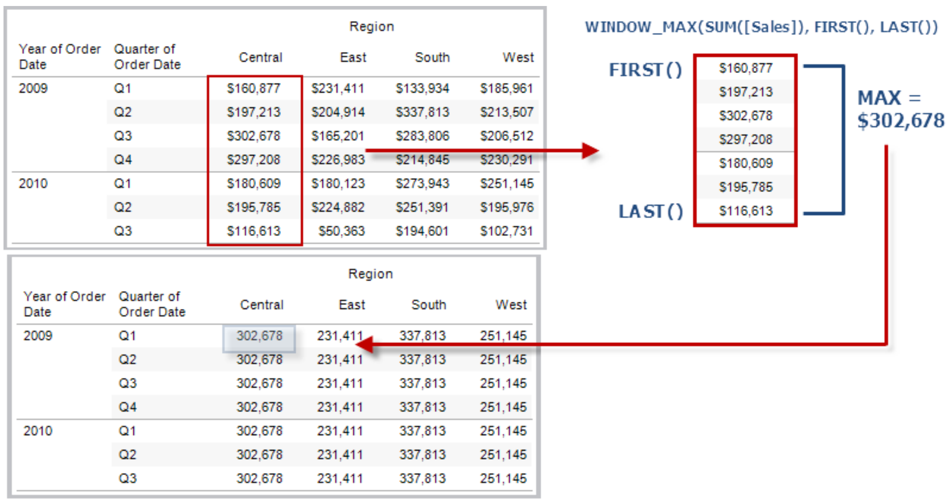
Example

`WINDOW_MEDIAN(SUM([Profit]), FIRST()+1, 0)` computes the median of SUM (Profit) from the second row to the current row.

WINDOW_MAX(expression, [start, end])

Returns the maximum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window maximum within the Date partition returns the maximum sales across all dates.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	302,678	231,411	337,813	251,145
	Q2	302,678	231,411	337,813	251,145
	Q3	302,678	231,411	337,813	251,145
	Q4	302,678	231,411	337,813	251,145
2010	Q1	302,678	231,411	337,813	251,145
	Q2	302,678	231,411	337,813	251,145
	Q3	302,678	231,411	337,813	251,145

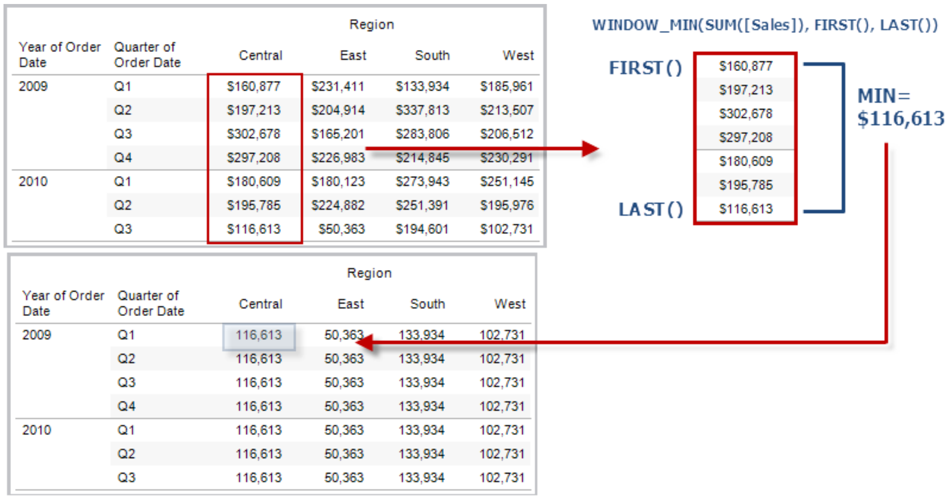
Example

`WINDOW_MAX(SUM([Profit]), FIRST()+1, 0)` computes the maximum of SUM (Profit) from the second row to the current row.

WINDOW_MIN(expression, [start, end])

Returns the minimum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window minimum within the Date partition returns the minimum sales across all dates.



Example

`WINDOW_MIN(SUM([Profit]), FIRST()+1, 0)` computes the minimum of SUM(Profit) from the second row to the current row.

WINDOW_PERCENTILE(expression, number, [start, end])

Returns the value corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_PERCENTILE(SUM([Profit]), 0.75, -2, 0)` returns the 75th percentile for SUM(Profit) from the two previous rows to the current row.

WINDOW_STDEV(expression, [start, end])

Returns the sample standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_STDEV(SUM([Profit]), FIRST()+1, 0)` computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_STDEVP(expression, [start, end])

Returns the biased standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

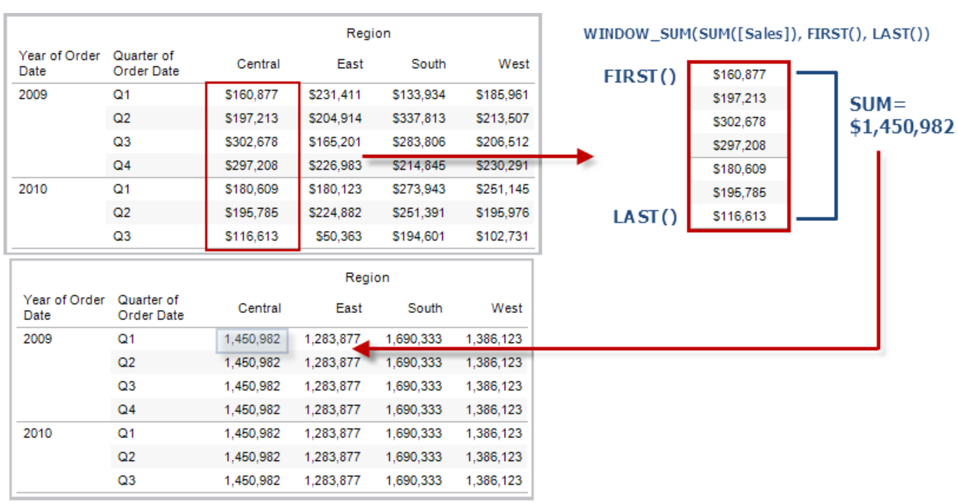
Example

`WINDOW_STDEVP(SUM([Profit]), FIRST()+1, 0)` computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_SUM(expression, [start, end])

Returns the sum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window sum computed within the Date partition returns the summation of sales across all quarters.



Example

`WINDOW_SUM(SUM([Profit]), FIRST()+1, 0)` computes the sum of SUM(Profit) from the second row to the current row.

WINDOW_VAR(expression, [start, end])

Returns the sample variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VAR((SUM([Profit])), FIRST()+1, 0)` computes the variance of SUM (Profit) from the second row to the current row.

WINDOW_VARP(expression, [start, end])

Returns the biased variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VARP(SUM([Profit]), FIRST()+1, 0)` computes the variance of SUM (Profit) from the second row to the current row.

Pass-Through functions (RAWSQL)

These RAWSQL pass-through functions can be used to send SQL expressions directly to the database, without first being interpreted by Tableau. If you have custom database functions that Tableau doesn’t know about, you can use these pass-through functions to call these custom functions.

Your database usually will not understand the field names that are shown in Tableau. Because Tableau does not interpret the SQL expressions you include in the pass-through functions, using the Tableau field names in your expression may cause errors. You can use a substitution syntax to insert the correct field name or expression for a Tableau calculation into pass-through SQL. For example, if you had a function that computed the median of a set of values, you could call that function on the Tableau column [Sales] like this:

```
RAWSQLAGG_REAL(“MEDIAN(%1)”, [Sales])
```

Because Tableau does not interpret the expression, you must define the aggregation. You can use the RAWSQLAGG functions described below when you are using aggregated expressions.

> RAWSQL pass-through functions will not work with published data sources.

> These functions may return different results starting in Tableau Desktop 8.2 than they did in earlier versions of Tableau Desktop. This is because Tableau now uses ODBC for pass-through functions instead of OLE DB. ODBC truncates when returning real values as integer; OLE DB rounds when returning real values as integer.

RAWSQL Functions

The following RAWSQL functions are available in Tableau.

RAWSQL_BOOL(“sql_expr”, [arg1], …[argN])

Returns a Boolean result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQL_BOOL(“IIF(%1 > %2, True, False)”, [Sales], [Profit])

RAWSQL_DATE(“sql_expr”, [arg1], …[argN])

Returns a Date result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Order Date].

```
RAWSQL_DATE(“%1”, [Order Date])
```

RAWSQL_DATETIME(“sql_expr”, [arg1], …[argN])

Returns a Date and Time result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

RAWSQL_DATETIME(“MIN(%1)”, [Delivery Date])

RAWSQL_INT(“sql_expr”, [arg1], …[argN])

Returns an integer result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

```
RAWSQL_INT(“500 + %1”, [Sales])
```

RAWSQL_REAL(“sql_expr”, [arg1], …[argN])

Returns a numeric result from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

```
RAWSQL_REAL("-123.98 * %1", [Sales])
```

RAWSQL_SPATIAL

Returns a Spatial from a given SQL expression that is passed directly to the underlying data source. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Geometry].

```
RAWSQL_SPATIAL("%1", [Geometry])
```

RAWSQL_STR("sql_expr", [arg1], …[argN])

Returns a string from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Customer Name].

Example

```
RAWSQL_STR("%1", [Customer Name])
```

RAWSQLAGG_BOOL("sql_expr", [arg1], …[argN])

Returns a Boolean result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQLAGG_BOOL("SUM(%1) >SUM(%2)", [Sales], [Profit])

RAWSQLAGG_DATE(“sql_expr”, [arg1], …[argN])

Returns a Date result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Order Date].

Example

```
RAWSQLAGG_DATE(“MAX(%1)”, [Order Date])
```

RAWSQLAGG_DATETIME(“sql_expr”, [arg1], …[argN])

Returns a Date and Time result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

```
RAWSQLAGG_DATETIME(“MIN(%1)”, [Delivery Date])
```

RAWSQLAGG_INT(“sql_expr”, [arg1,] …[argN])

Returns an integer result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

```
RAWSQLAGG_INT(“500 + SUM(%1)”, [Sales])
```

RAWSQLAGG_REAL(“sql_expr”, [arg1,] …[argN])

Returns a numeric result from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

`RAWSQLAGG_REAL("SUM( %1)", [Sales])`

RAWSQLAGG_STR("sql_expr", [arg1,] …[argN])

Returns a string from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Discount].

Example

`RAWSQLAGG_STR("AVG(%1)", [Discount])`

Spatial functions

Spatial functions allow you to perform advanced spatial analysis and combine spatial files with data in other formats like text files or spreadsheets.

Function	Syntax	Description
DISTANCE	`DISTANCE(<geometry1>,<geometry2>,"<units>")`	Returns distance measurement between two points in a specified unit. Supported unit names: meters ("meters," "metres" "m), kilometers ("kilometers," "kilometres," "km"), miles ("miles" or "miles"), feet ("feet," "ft"). This function can only be created with a live connection and will continue to work when a data source is converted to an extract.

<table>
<tr><td></td><td></td><td>Example:
<pre>DISTANCE([Origin
MakePoint],
[Destination
MakePoint],
"km")</pre></td></tr>
<tr><td>MAKELINE</td><td><code>MAKELINE(<geo-
metry1>,<geometry2>)</code></td><td>Generates a line mark between two points; useful for building origin-destination maps.

CASE is often easier to use than IIF or IF THEN ELSE.

Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function , although the CASE function will generally be more concise.

Many times you can use a group to get the same results as a complicated case function.

Examples:
<pre>MAKELINE
(OriginPoint,
DestinationPoin</pre></td></tr>
</table>

		`t)`
		
		`MAKELINE (MAKEPOINT (OriginLat], [OriginLong]),MAKEPOINT ([DestinationLat], [DestinationLong] )`

MAKEPOINT	`MAKEPOINT(<latitude>, <longitude>)`	Converts data from latitude and longitude columns into spatial objects. You can use MAKEPOINT to spatially-enable a data source so that it can be joined with a spatial file using a spatial join. For more information, see **Join Spatial Files in Tableau** on page 1815 To use MAKEPOINT, your data must contain latitude and longitude coordinates. Example: `MAKEPOINT ([AirportLatitude], [AirportLongitude])`

MAKEPOINT(X,Y, SRID)	`MAKEPOINT(<xcoordinate>, <ycoordinate>, <SRID>`	Converts data from projected geographic coordinates into spatial objects. SRID is a spatial reference identifier that uses ESPG reference system codes to specify coordinate systems. If SRID is not specified, WGS84 is assumed and parameters are treated as latitude/longitude in degrees. This function can only be created with a live connection and will continue to work when a data source is converted to an extract. Example: `MAKEPOINT ([Xcoord], [Ycoord],3493)`

Additional functions

- **Regular Expressions** below
- **Hadoop Hive Specific Functions** on page 2337
- **Google BigQuery Specific Functions** on page 2340

Regular Expressions

REGEXP_REPLACE(string, pattern, replacement)

Returns a copy of the given string where the regular expression pattern is replaced by the replacement string. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern and the replacement must be constants.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_REPLACE('abc 123', '\s', '-') = 'abc-123'

REGEXP_MATCH(string, pattern)

Returns true if a substring of the specified string matches the regular expression pattern. This function is available for Text File, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Impala 2.3.0 (through Cloudera Hadoop data sources), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_MATCH('-([1234].[The.Market])-','\[\s*(\w*\.)(\w*\s*\])')=true

REGEXP_EXTRACT(string, pattern)

Returns the portion of the string that matches the regular expression pattern. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT('abc 123', '[a-z]+\s+(\d+)') = '123'

REGEXP_EXTRACT_NTH(string, pattern, index)

Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If index is 0, the entire string is returned. This function is available for Text File, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT_NTH('abc 123', '([a-z]+)\s+(\d+)', 2) = '123'

Hadoop Hive Specific Functions

> **Note**: Only the PARSE_URL and PARSE_URL_QUERY functions are available for Cloudera Impala data sources.

GET_JSON_OBJECT(JSON string, JSON path)

Returns the JSON object within the JSON string based on the JSON path.

PARSE_URL(string, url_part)

Returns a component of the given URL string where the component is defined by url_part. Valid url_part values include: 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'AUTHORITY', 'FILE' and 'USERINFO'.

Example

PARSE_URL('http://www.tableau.com', 'HOST') = 'www.tableau.com'

PARSE_URL_QUERY(string, key)

Returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

Example

PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4', 'page') = '1'

XPATH_BOOLEAN(XML string, XPath expression string)

Returns true if the XPath expression matches a node or evaluates to true.

Example

XPATH_BOOLEAN('<values> <value id="0">1</value><value id="1">5</value>', 'values/value [@id="1"] = 5') = true

XPATH_DOUBLE(XML string, XPath expression string)

Returns the floating-point value of the XPath expression.

Example

XPATH_DOUBLE('<values><value>1.0</value><value>5.5</value> </values>', 'sum (value/*)') = 6.5

XPATH_FLOAT(XML string, XPath expression string)

Returns the floating-point value of the XPath expression.

Example

XPATH_FLOAT('<values><value>1.0</value><value>5.5</value> </values>','sum(value/*)') = 6.5

XPATH_INT(XML string, XPath expression string)

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_INT('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_LONG(XML string, XPath expression string)

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_LONG('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_SHORT(XML string, XPath expression string)

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_SHORT('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_STRING(XML string, XPath expression string)

Returns the text of the first matching node.

Example

XPATH_STRING('<sites ><url domain="org">http://www.w3.org</url> <url domain="com">http://www.tableau.com</url></sites>', 'sites/url[@domain="com"]') = 'http://www.tableau.com'

Google BigQuery Specific Functions

DOMAIN(string_url)

Given a URL string, returns the domain as a string.

Example

DOMAIN('http://www.google.com:80/index.html') = 'google.com'

GROUP_CONCAT(expression)

Concatenates values from each record into a single comma-delimited string. This function acts like a SUM() for strings.

Example

GROUP_CONCAT(Region) = "Central,East,West"

HOST(string_url)

Given a URL string, returns the host name as a string.

Example

HOST('http://www.google.com:80/index.html') = 'www.google.com:80'

LOG2(number)

Returns the logarithm base 2 of a number.

Example

LOG2(16) = '4.00'

LTRIM_THIS(string, string)

Returns the first string with any leading occurrence of the second string removed.

Example

LTRIM_THIS('[-Sales-]','[-') = 'Sales-]'

RTRIM_THIS(string, string)

Returns the first string with any trailing occurrence of the second string removed.

Example

RTRIM_THIS('[-Market-]','-]') = '[-Market'

TIMESTAMP_TO_USEC(expression)

Converts a TIMESTAMP data type to a UNIX timestamp in microseconds.

Example

TIMESTAMP_TO_USEC(#2012-10-01 01:02:03#)=1349053323000000

USEC_TO_TIMESTAMP(expression)

Converts a UNIX timestamp in microsseconds to a TIMESTAMP data type.

Example

USEC_TO_TIMESTAMP(1349053323000000) = #2012-10-01 01:02:03#

TLD(string_url)

Given a URL string, returns the top level domain plus any country/region domain in the URL.

Example

TLD('http://www.google.com:80/index.html') = '.com'

TLD('http://www.google.co.uk:80/index.html') = '.co.uk'

Want to learn more about functions?

Read the functions topics.

See also

Tableau Functions (Alphabetical) on the next page

Tableau Functions (Alphabetical)

The Tableau functions in this reference are organized alphabetically. Click a letter to see functions that start with it. If no functions start with that letter, the functions that start with the next letter in the alphabet are shown. You can also press Ctrl+F (Command-F on a Mac) to open a search box that you can use to search the page for a specific function.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

ABS(number)

Returns the absolute value of the given number.

Examples

`ABS(-7) = 7`

`ABS([Budget Variance])`

The second example returns the absolute value for all the numbers contained in the `Budget Variance` field.

ACOS(number)

Returns the arc cosine of the given number. The result is in radians.

Example

`ACOS(-1) = 3.14159265358979`

ASCII(string)

Return the ASCII code for the first character of `string`.

Example

`ASCII('A') = 65`

ASIN(number)

Returns the arc sine of a given number. The result is in radians.

Example

`ASIN(1) = 1.5707963267949`

ATAN(number)

Returns the arc tangent of a given number. The result is in radians.

Example

`ATAN(180) = 1.5652408283942`

ATAN2(y number, x number)

Returns the arc tangent of two given numbers (x and y). The result is in radians.

Example

`ATAN2(2, 1) = 1.10714871779409`

ATTR(expression)

Returns the value of the expression if it has a single value for all rows. Otherwise returns an asterisk. Null values are ignored.

AVG(expression)

Returns the average of all the values in the expression. AVG can be used with numeric fields only. Null values are ignored.

CASE expression WHEN value1 THEN return1 WHEN value2 THEN return2...ELSE default return END

Use the CASE function to perform logical tests and return appropriate values. CASE is often easier to use than IIF or IF THEN ELSE. The CASE function evaluates `expression`, compares it to a sequence of values, `value1`, `value2`, etc., and returns a result. When a value that matches `expression` is encountered, CASE returns the corresponding return value. If no match is found, the default return expression is used. If there is no default return and no values match, then Null is returned.

Typically, you use an IF function to perform a sequence of arbitrary tests, and you use a CASE function to search for a match to an expression. But a CASE function can always be rewritten as an IF function , although the CASE function will generally be more concise.

Many times you can use a group to get the same results as a complicated case function.

Examples

```
CASE [Region] WHEN 'West' THEN 1 WHEN 'East' THEN 2 ELSE 3 END
```

```
CASE LEFT(DATENAME('weekday',[Order Date]),3) WHEN 'Sun' THEN 0
WHEN 'Mon' THEN 1 WHEN 'Tue' THEN 2 WHEN 'Wed' THEN 3 WHEN 'Thu'
THEN 4 WHEN 'Fri' THEN 5 WHEN 'Sat' THEN 6 END
```

CEILING(number)

Rounds a number to the nearest integer of equal or greater value.

Example

```
CEILING(3.1415) = 4
```

Availability by data source

Data Source	**Support**
Microsoft Access	Not supported
Microsoft Excel	Supported
Text File	Supported
Statistical File	Supported
Tableau Server	Supported
Actian Vectorwise	Not supported
Amazon Aurora	Not supported
Amazon EMR Hadoop Hive	Supported
Amazon Redshift	Not supported
Aster Database	Not supported
Cloudera Hadoop	Supported

DataStax Enterprise	Supported
EXASOL	Not supported
Firebird	Not supported
Google Analytics	Supported
Google BigQuery	Supported
Google Cloud SQL	Not supported
Hortonworks Hadoop Hive	Supported
IBM BigInsights	Not supported
IBM DB2	Not supported
IBM PDA (Netezza)	Not supported
MapR Hadoop Hive	Supported
MarkLogic	Not supported
Microsoft Analysis Services	Not supported
Microsoft PowerPivot	Not supported
Microsoft SQL Server	Not supported
MySQL	Not supported
Oracle	Not supported
Oracle Essbase	Not supported
Actian Matrix (ParAccel)	Not supported
Pivotal Greenplum	Not supported
PostgreSQL	Not supported
Progress OpenEdge	Not supported
Salesforce	Supported

SAP HANA	Not supported
SAP Sybase ASE	Not supported
SAP Sybase IQ	Not supported
Spark SQL	Supported
Splunk	Not supported
Teradata	Not supported
Teradata OLAP Connector	Not supported
Vertica	Not supported

CHAR(number)

Returns the character encoded by the ASCII code `number`.

Example

```
CHAR(65) = 'A'
```

COLLECT (spatial)

An aggregate calculation that combines the values in the argument field. Null values are ignored.

Note: The COLLECT function can only be used with spatial fields.

Example

```
COLLECT ([Geometry])
```

CONTAINS(string, substring)

Returns true if the given string contains the specified substring.

Example

```
CONTAINS(“Calculation”, “alcu”) = true
```

CORR(expression 1, expression2)

Returns the Pearson correlation coefficient of two expressions.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship.

CORR is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASOL
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_CORR. See **Table Calculation Functions** on page 2219.

> **Note**: The square of a CORR result is equivalent to the R-Squared value for a linear trend line model. See **Trend Line Model Terms** on page 2537.

Example

You can use CORR to visualize correlation in a disaggregated scatter plot. The way to do this is to use a table-scoped level of detail expression. For example:

```
{CORR(Sales, Profit)}
```

With a level of detail expression, the correlation is run over all rows. If you used a formula like `CORR(Sales, Profit)` (without the surrounding brackets to make it a level of detail expression), the view would show the correlation of each individual point in the scatter plot with each other point, which is undefined.

See **Table-Scoped** on page 2461

COS(number)

Returns the cosine of an angle. Specify the angle in radians.

Example

```
COS(PI( ) /4) = 0.707106781186548
```

COT(number)

Returns the cotangent of an angle. Specify the angle in radians.

Example

```
COT(PI( ) /4) = 1
```

COUNT(expression)

Returns the number of items in a group. Null values are not counted.

COUNTD(expression)

Returns the number of distinct items in a group. Null values are not counted. This function is not available in the following cases: workbooks created before Tableau Desktop 8.2 that use Microsoft Excel or text file data sources, workbooks that use the legacy connection, and workbooks that use Microsoft Access data sources. Extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.

COVAR(expression 1, expression2)

Returns the *sample covariance* of two expressions.

Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (available with the COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

COVAR is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASOL
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- IBM PDA (Netezza)
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_COVAR. See **Table Calculation Functions** on page 2219.

If expression1 and expression2 are the same—for example, COVAR([profit], [profit])—COVAR returns a value that indicates how widely values are distributed.

> **Note**: The value of COVAR(X, X) is equivalent to the value of VAR(X) and also to the value of STDEV(X)^2.

Example

The following formula returns the sample covariance of **Sales** and **Profit**.

```
COVAR([Sales], [Profit])
```

COVARP(expression 1, expression2)

Returns the *population covariance* of two expressions.

Covariance quantifies how two variables change together. A positive covariance indicates that the variables tend to move in the same direction, as when larger values of one variable tend to correspond to larger values of the other variable, on average. Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the COVAR function) is appropriate.

COVARP is available with the following data sources:

- Tableau data extracts (you can create an extract from any data source)
- Cloudera Hive
- EXASOL
- Firebird (version 3.0 and later)
- Google BigQuery
- Hortonworks Hadoop Hive
- IBM PDA (Netezza)
- Oracle
- PostgreSQL
- Presto
- SybaseIQ
- Teradata
- Vertica

For other data sources, consider either extracting the data or using WINDOW_COVARP. See **Table Calculation Functions** on page 2219.

If expression1 and expression2 are the same—for example, COVARP([profit], [profit])—COVARP returns a value that indicates how widely values are distributed.

> **Note**: The value of COVARP(X, X) is equivalent to the value of VARP(X) and also to the value of STDEVP(X)^2.

Example

The following formula returns the population covariance of **Sales** and **Profit**.

```
COVARP([Sales], [Profit])
```

DATE(expression)

Returns a date given a number, string, or date expression.

Examples

```
DATE([Employee Start Date])
```

```
DATE("April 15, 2004") = #April 15, 2004#
```

```
DATE("4/15/2004")
```

```
DATE(#2006-06-15 14:52#) = #2006-06-15#
```

Quotation marks are required in the second and third examples.

DATEADD(date_part, interval, date)

Returns the specified date with the specified number `interval` added to the specified `date_part` of that date.

Example

```
DATEADD('month', 3, #2004-04-15#) = 2004-07-15 12:00:00 AM
```

This expression adds three months to the date `#2004-04-15#`.

DATEDIFF(date_part, date1, date2, [start_of_week])

Returns the difference between `date1` and `date2` expressed in units of `date_part`.

The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If it is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.

Example

```
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'monday')= 1
```

```
DATEDIFF('week', #2013-09-22#, #2013-09-24#, 'sunday')= 0
```

The first expression returns 1 because when `start_of_week` is 'monday', then 22 September (a Sunday) and 24 September (a Tuesday) are in different weeks. The second expression returns 0 because when `start_of_week` is 'sunday' then 22 September (a Sunday) and 24 September (a Tuesday) are in the same week.

DATENAME(date_part, date, [start_of_week])

Returns `date_part` of `date` as a string. The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.

Examples

```
DATENAME('year', #2004-04-15#) = "2004"
```

```
DATENAME('month', #2004-04-15#) = "April"
```

DATEPARSE(format, string)

Converts a string to a datetime in the specified format. Support for some locale-specific formats is determined by the computer's system settings. Letters that appear in the data and do not need to be parsed should be surrounded by single quotes (' '). For formats that do not have delimiters between values (for example, MMddyy), verify that they are parsed as expected. The format must be a constant string, not a field value. This function returns `Null` if the data does not match the format.

This function is available for several connectors. For more information, see **Convert a Field to a Date Field** on page 980.

Examples

```
DATEPARSE ("dd.MMMM.yyyy", "15.April.2004") = #April 15, 2004#
```

```
DATEPARSE ("h'h' m'm' s's'", "10h 5m 3s") = #10:05:03#
```

DATEPART(date_part, date, [start_of_week])

Returns `date_part` of `date` as an integer.

The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.

> **Note:** When the `date_part` is weekday, the `start_of_week` parameter is ignored. This is because Tableau relies on a fixed weekday ordering to apply offsets.

Examples

```
DATEPART('year', #2004-04-15#) = 2004
```

```
DATEPART('month', #2004-04-15#) = 4
```

DATETIME(expression)

Returns a datetime given a number, string, or date expression.

Example

```
DATETIME("April 15, 2005 07:59:00") = April 15, 2005 07:59:00
```

DATETRUNC(date_part, date, [start_of_week])

Truncates the specified date to the accuracy specified by the date_part. This function returns a new date. For example, when you truncate a date that is in the middle of the month at the month level, this function returns the first day of the month. The `start_of_week` parameter, which you can use to specify which day is to be considered the first day or the week, is optional. Possible values are 'monday', 'tuesday', etc. If `start_of_week` is omitted, the start of week is determined by the data source. See **Date Properties for a Data Source** on page 1247.

Examples

```
DATETRUNC('quarter', #2004-08-15#) = 2004-07-01 12:00:00 AM
```

```
DATETRUNC('month', #2004-04-15#) = 2004-04-01 12:00:00 AM
```

DAY(date)

Returns the day of the given date as an integer.

Example

```
DAY(#2004-04-12#) = 12
```

DEGREES(number)

Converts a given number in radians to degrees.

Example

```
DEGREES(PI( )/4) = 45.0
```

DISTANCE(Geometry1, Geometry2, "Units")

Returns distance measurement between two points in a specified unit. Supported unit names: meters ("meters," "metres" "m), kilometers ("kilometers," "kilometres," "km"), miles ("miles" or "miles"), feet ("feet," "ft").

This function can only be created with a live connection and will continue to work when a data source is converted to an extract.

Examples

```
DISTANCE ({ EXCLUDE [Branch Name] : COLLECT([Selected Point]) },
[unselected point], 'km')
```

```
DISTANCE([Origin MakePoint],[Destination MakePoint], "miles")
```

DIV(integer1, integer2)

Returns the integer part of a division operation, in which integer1 is divided by integer2.

Example

```
DIV(11,2) = 5
```

DOMAIN(string_url)

***Note**: Supported only when connected to Google BigQuery*

Given a URL string, returns the domain as a string.

Example

DOMAIN('http://www.google.com:80/index.html') = 'google.com'

ENDSWITH(string, substring)

Returns true if the given string ends with the specified substring. Trailing white spaces are ignored.

Example

```
ENDSWITH("Tableau", "leau") = true
```

EXP(number)

Returns e raised to the power of the given number.

Examples

```
EXP(2) = 7.389
```

```
EXP(-[Growth Rate]*[Time])
```

FIND(string, substring, [start])

Returns the index position of `substring` in `string`, or 0 if the `substring` isn't found. If the optional argument `start` is added, the function ignores any instances of `substring` that appear before the index position `start`. The first character in the string is position 1.

Examples

```
FIND("Calculation", "alcu") = 2

FIND("Calculation", "Computer") = 0

FIND("Calculation", "a", 3) = 7

FIND("Calculation", "a", 2) = 2

FIND("Calculation", "a", 8) = 0
```

FINDNTH(string, substring, occurrence)

Returns the position of the nth occurrence of substring within the specified string, where n is defined by the occurrence argument.

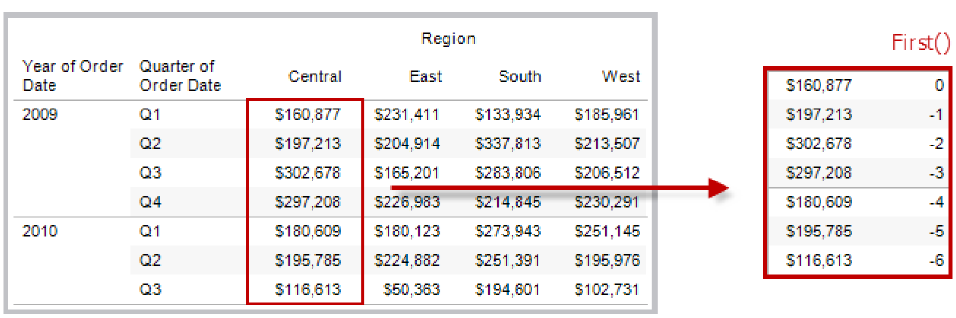
> **Note**: FINDNTH is not available for all data sources.

Example

```
FINDNTH("Calculation", "a", 2) = 7
```

FIRST( )

Returns the number of rows from the current row to the first row in the partition. For example, the view below shows quarterly sales. When FIRST() is computed within the Date partition, the offset of the first row from the second row is -1.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

First()

$160,877	0
$197,213	-1
$302,678	-2
$297,208	-3
$180,609	-4
$195,785	-5
$116,613	-6

Example

When the current row index is 3, `FIRST() = -2`.

FLOAT(expression)

Casts its argument as a floating point number.

Examples

`FLOAT(3)` = `3.000`

`FLOAT([Age])` converts every value in the `Age` field to a floating point number.

FLOOR(number)

Rounds a number to the nearest integer of equal or lesser value.

Example

`FLOOR(3.1415) = 3`

Availability by data source

Data Source	Support
Microsoft Access	Not supported
Microsoft Excel	Supported
Text File	Supported
Statistical File	Supported
Tableau Server	Supported
Actian Vector	Not supported
Amazon Aurora	Not supported
Amazon EMR Hadoop Hive	Supported
Amazon Redshift	Not supported
Aster Database	Not supported
Cloudera Hadoop	Supported

DataStax Enterprise	Supported
EXASOL	Not supported
Firebird	Not supported
Google Analytics	Supported
Google BigQuery	Supported
Google Cloud SQL	Not supported
Hortonworks Hadoop Hive	Supported
IBM BigInsights	Not supported
IBM DB2	Not supported
IBM Netezza	Not supported
MapR Hadoop Hive	Supported
MarkLogic	Not supported
Microsoft Analysis Services	Not supported
Microsoft PowerPivot	Not supported
Microsoft SQL Server	Not supported
MySQL	Not supported
Oracle	Not supported
Oracle Essbase	Not supported
ParAccel	Not supported
Pivotal Greenplum	Not supported
PostgreSQL	Not supported
Progress OpenEdge	Not supported
Salesforce	Supported

SAP HANA	Not supported
SAP Sybase ASE	Not supported
SAP Sybase IQ	Not supported
Spark SQL	Supported
Splunk	Not supported
Teradata	Not supported
Teradata OLAP Connector	Not supported
Vertica	Not supported

FULLNAME( )

Returns the full name for the current user. This is the Tableau Server or Tableau Online full name when the user is signed in; otherwise the local or network full name for the Tableau Desktop user.

Example

```
[Manager]=FULLNAME( )
```

If manager Dave Hallsten is signed in, this example returns True only if the Manager field in the view contained Dave Hallsten. When used as a filter, this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.

GET_JSON_OBJECT(JSON string, JSON path)

Note: *Supported only when connected to Hadoop Hive.*

Returns the JSON object within the JSON string based on the JSON path.

GROUP_CONCAT(expression)

Note*: Supported only when connected to Google BigQuery*

Concatenates values from each record into a single comma-delimited string. This function acts like a SUM() for strings.

Example

GROUP_CONCAT(Region) = "Central,East,West"

HEXBINX(number, number)

Maps an x, y coordinate to the x-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.

HEXBINX and HEXBINY are binning and plotting functions for hexagonal bins. Hexagonal bins are an efficient and elegant option for visualizing data in an x/y plane such as a map. Because the bins are hexagonal, each bin closely approximates a circle and minimizes variation in the distance from the data point to the center of the bin. This makes the clustering both more accurate and informative.

Example

```
HEXBINX([Longitude], [Latitude])
```

HEXBINY(number, number)

Maps an x, y coordinate to the y-coordinate of the nearest hexagonal bin. The bins have side length 1, so the inputs may need to be scaled appropriately.

Example

```
HEXBINY([Longitude], [Latitude])
```

HOST(string_url)

***Note**: Supported only when connected to Google BigQuery*

Given a URL string, returns the host name as a string.

Example

HOST('http://www.google.com:80/index.html') = 'www.google.com:80'

IF test THEN value END / IF test THEN value ELSE else END

Use the IF THEN ELSE function to perform logical tests and return appropriate values. The IF THEN ELSE function evaluates a sequence of test conditions and returns the value for the first condition that is true. If no condition is true, the ELSE value is returned. Each test must be a boolean: either be a boolean field in the data source or the result of a logical expression. The final ELSE is optional, but if it is not provided and there is no true test expression, then the function returns Null. All of the value expressions must be of the same type.

Examples

```
IF [Cost]>[Budget Cost] THEN 'Over Budget' ELSE 'Under Budget'
END
```

```
IF [Budget Sales]!=0 THEN [Sales]/[Budget Sales] END
```

IF test1 THEN value1 ELSEIF test2 THEN value2 ELSE else END

Use this version of the IF function to perform logical tests recursively. There is no built-in limit to the number of ELSEIFvalues you can use with an IF function, though individual databases may impose a limit on IF function complexity. While an IF function can be rewritten as a series of nested IIF statements, there are differences in how the expressions will be evaluated. In particular, an IIF statement distinguishes TRUE, FALSE and UNKNOWN, whereas an IF statement only worries about TRUE and not true (which includes both FALSE and UNKNOWN).

Example

When you create bins from a measure, Tableau creates bins of equal size by default. For example, say you have a measure that represents age. When you create bins from that measure, Tableau makes all the bins of equal size. You can specify how big you want the bins to be, but you cannot specify a separate range of values for each bin. A way around this constraint is to create a calculated field to define bins. Then you can create one bin for ages 0 - 20, another for ages 21 - 32, and so on. The following procedure shows how you could do that.

1. Create a new calculated field by choosing **Analysis** > **Create Calculated Field**.
2. Name the field **Age Groups** and type the following in the definition area

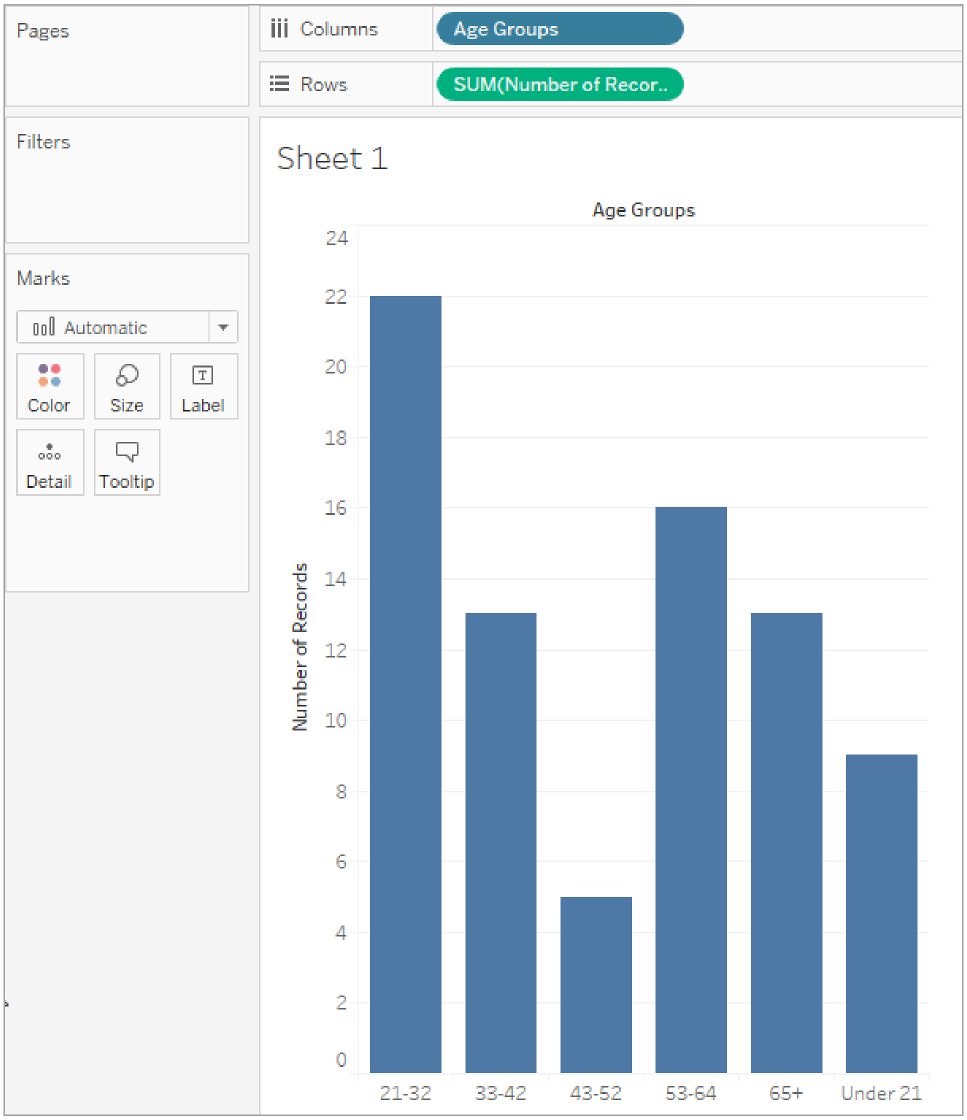
   ```
   IF
   [Age] < 21 THEN 'Under 21'
   ELSEIF
   [Age] <= 32 THEN '21-32'
   ELSEIF
   [Age] <= 42 THEN '33-42'
   ELSEIF
   ```

```
[Age] <= 52 THEN '43-52'
ELSEIF
[Age] <= 64 THEN '53-64'
ELSE '65+'
END
```

Confirm that the status message indicates that the formula is valid, and then click **OK**.

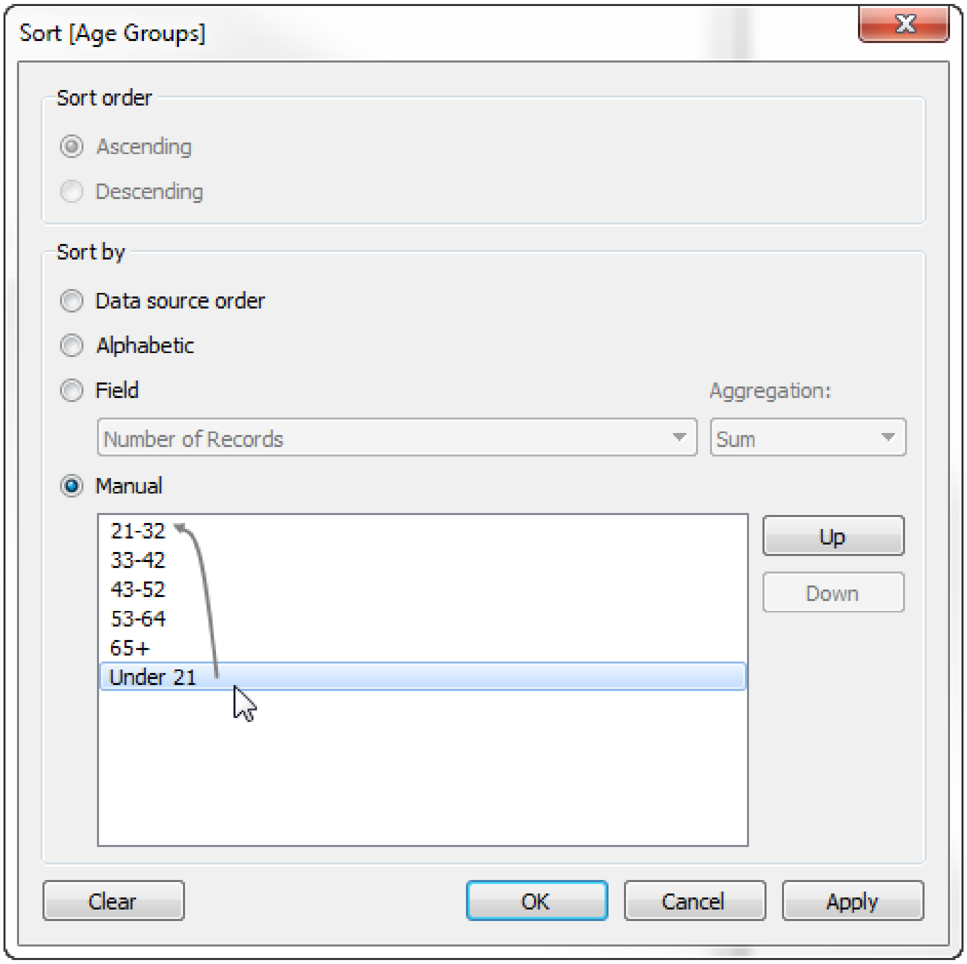
3. From the Measures area of the **Data** pane, drag **Number of Records** to **Rows**.
4. From the Dimensions area of the **Data** pane, drag **Age Groups** to **Columns**.

   The records are now divided among the six bins that you defined:



Unfortunately, the **Under 21** bin is at far right, when you would expect it to be at far left. Tableau's smart enough to put the bins with entirely numerical names in the right order, but it can't guess that the bin name beginning with 'Under' belongs at the left. Fix the problem with a manual sort.

5. Click the down arrow at the right side of the Age Groups field on Columns and then click Sort. Choose Manual and then move the Under 21 bin up to the top of the list:



Your view is now complete.

IIF(test, then, else, [unknown])

Use the IIF function to perform logical tests and return appropriate values. The first argument, `test`, must be a boolean: either a boolean field in the data source, or the result of a logical expression using operators (or a logical comparison of AND, OR, or NOT). If `test` evaluates to TRUE, then IIF returns the `then` value. If `test` evaluates to FALSE, then IIF returns the `else` value.

A boolean comparison may also yield the value UNKNOWN (neither TRUE nor FALSE), usually due to the presence of Null values in test. The final argument to IIF is returned in the event of an UNKNOWN result for the comparison. If this argument is left out, Null is returned.

Examples

```
IIF(7>5, 'Seven is greater than five', 'Seven is less than five')
```

```
IIF([Cost]>[Budget Cost], 'Over Budget', 'Under Budget')

IIF([Budget Sales]!=0,[Sales]/[Budget Sales],0)

IIF(Sales>=[Budget Sales], 'Over Cost Budget and Over Sales
Budget', 'Over Cost Budget and Under Sales Budget','Under Cost
Budget')
```

IFNULL(expression1, expression2)

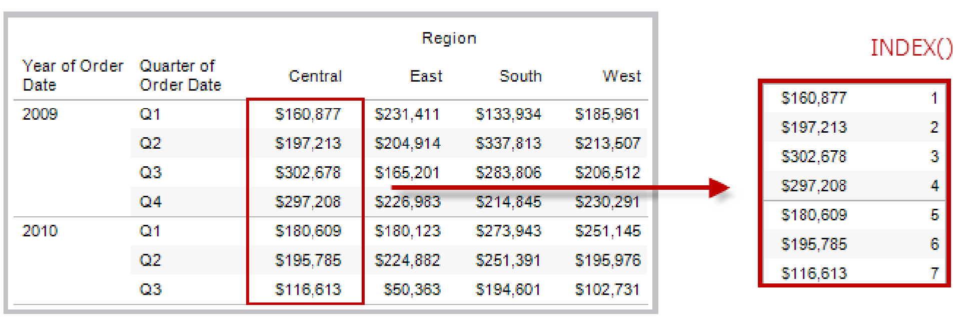
The IFNULL function returns the first expression if the result is not null, and returns the second expression if it is null.

Example

```
IFNULL([Profit], 0) = [Profit]
```

INDEX( )

Returns the index of the current row in the partition, without any sorting with regard to value. The first row index starts at 1. For example, the table below shows quarterly sales. When INDEX() is computed within the Date partition, the index of each row is 1, 2, 3, 4..., etc.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

INDEX()

$160,877	1
$197,213	2
$302,678	3
$297,208	4
$180,609	5
$195,785	6
$116,613	7

Example

For the third row in the partition, `INDEX() = 3`.

INT(expression)

Casts its argument as an integer. For expressions, this function truncates results to the closest integer toward zero.

Examples

```
INT(8.0/3.0)=2
```

`INT(4.0/1.5) = 2`

`INT(0.50/1.0) = 0`

`INT(-9.7) = -9`

When a string is converted to an integer it is first converted to a float and then rounded.

ISDATE(string)

The ISDATE function returns `TRUE` if the string argument can be converted to a date and `FALSE` if it cannot.

Examples

```
ISDATE('January 1, 2003') = TRUE
```

```
ISDATE('Jan 1 2003') = TRUE
```

```
ISDATE('1/1/03') = TRUE
```

```
ISDATE('Janxx 1 2003') = FALSE
```

ISFULLNAME(string)

Returns true if the current user's full name matches the specified full name, or false if it does not match. This function uses the Tableau Server or Online full name when the user is signed in; otherwise it uses the local or network full name for the Tableau Desktop user.

Example

```
ISFULLNAME(“Dave Hallsten”)
```

This example returns true if Dave Hallsten is the current user, otherwise it returns false.

ISNULL(expression)

The ISNULL function returns `TRUE` if the expression is `Null` and `FALSE` if it is not.

Example

The following example uses ISNULL in combination with IIF to replace null values with 0's.

```
IIF(ISNULL([Sales]), 0,[Sales] )
```

ISMEMBEROF(string)

Returns true if the person currently using Tableau is a member of a group that matches the given string. If the person currently using Tableau is signed in, the group membership is determined by groups on Tableau Server or Tableau Online. If the person is not signed in, this function returns false.

Example

```
IF ISMEMBEROF(“Sales”) THEN “Sales” ELSE “Other” END
```

ISUSERNAME(string)

Returns true if the current user's username matches the specified username, or false if it does not match. This function uses the Tableau Server or Online username when the user is signed in; otherwise it uses the local or network username for the Tableau Desktop user.

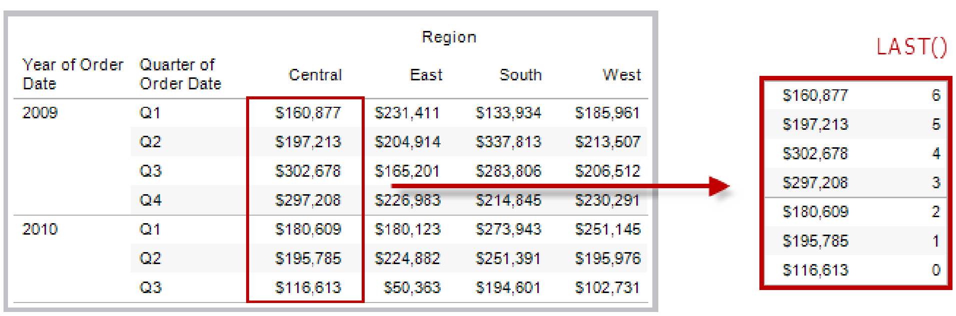
Example

```
ISUSERNAME(“dhallsten”)
```

This example returns true if dhallsten is the current user; otherwise it returns false.

LAST( )

Returns the number of rows from the current row to the last row in the partition. For example, the table below shows quarterly sales. When LAST() is computed within the Date partition, the offset of the last row from the second row is 5.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

LAST()

$160,877	6
$197,213	5
$302,678	4
$297,208	3
$180,609	2
$195,785	1
$116,613	0

Example

When the current row index is 3 of 7, `LAST() = 4`.

LEFT(string, number)

Returns the left-most number of characters in the string.

Example

```
LEFT("Matador", 4) = "Mata"
```

LEN(string)

Returns the length of the string.

Example

```
LEN("Matador") = 7
```

LN(number)

Returns the natural logarithm of a number. Returns `Null` if number is less than or equal to 0.

LOG(number [, base])

Returns the logarithm of a number for the given base. If the base value is omitted, base 10 is used.

LOG2(number)

***Note**: Supported only when connected to Google BigQuery*

Returns the logarithm base 2 of a number.

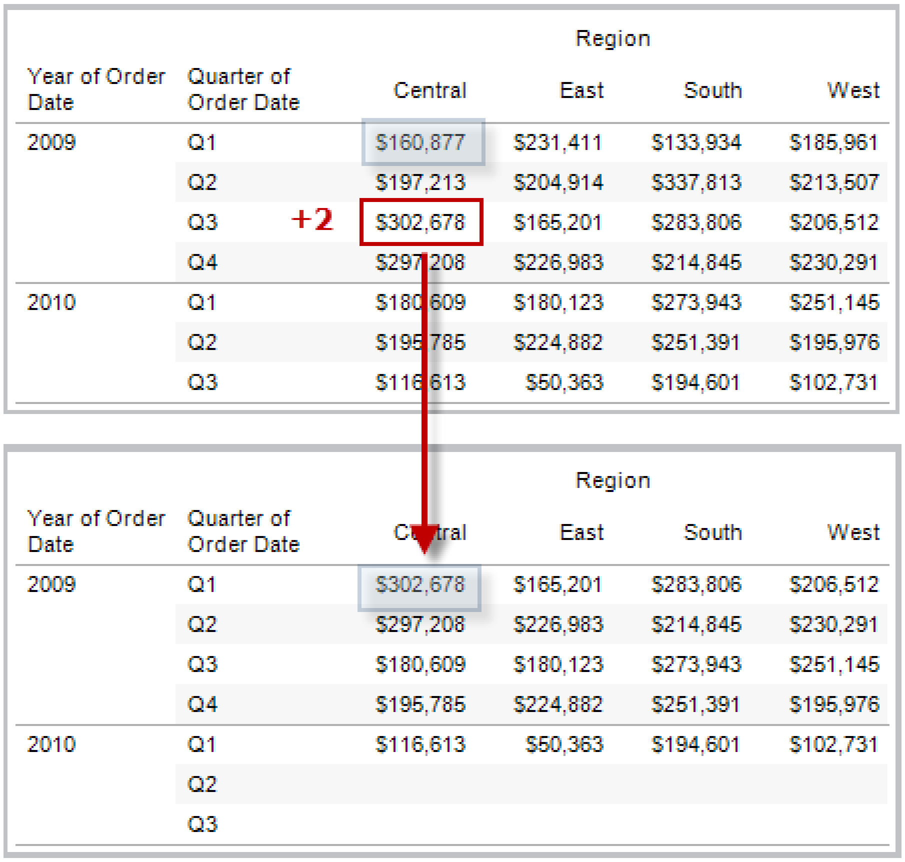
Example

LOG2(16) = '4.00'

LOOKUP(expression, [offset])

Returns the value of the expression in a target row, specified as a relative offset from the current row. Use FIRST() + n and LAST() - n as part of your offset definition for a target relative to the first/last rows in the partition. If `offset` is omitted, the row to compare to can be set on the field menu. This function returns NULL if the target row cannot be determined.

The view below shows quarterly sales. When `LOOKUP (SUM(Sales), 2)` is computed within the Date partition, each row shows the sales value from 2 quarters into the future.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3 +2	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$302,678	$165,201	$283,806	$206,512
	Q2	$297,208	$226,983	$214,845	$230,291
	Q3	$180,609	$180,123	$273,943	$251,145
	Q4	$195,785	$224,882	$251,391	$195,976
2010	Q1	$116,613	$50,363	$194,601	$102,731
	Q2				
	Q3				

Example

`LOOKUP(SUM([Profit]), FIRST()+2)` computes the SUM(Profit) in the third row of the partition.

LOWER(string)

Returns `string`, with all characters lowercase.

Example

```
LOWER("ProductVersion") = "productversion"
```

LTRIM(string)

Returns the string with any leading spaces removed.

Example

```
LTRIM(" Matador ") = "Matador "
```

LTRIM_THIS(string, string)

***Note**: Supported only when connected to Google BigQuery*

Returns the first string with any leading occurrence of the second string removed.

Example

LTRIM_THIS('[-Sales-]','[-') = 'Sales-]'

MAKEDATE(year, month, day)

Returns a date value constructed from the specified year, month, and date.

Available for Tableau Data Extracts. Check for availability in other data sources.

Example

```
MAKEDATE(2004, 4, 15) = #April 15, 2004#
```

MAKEDATETIME(date, time)

Returns a datetime that combines a date and a time. The date can be a date, datetime, or a string type. The time must be a datetime. This function is available only for MySQL-compatible connections (which for Tableau are, in addition to MySQL, Amazon Aurora and Amazon Aurora).

Examples

```
MAKEDATETIME("1899-12-30", #07:59:00#) = #12/30/1899 7:59:00 AM#
```

```
MAKEDATETIME([Date], [Time]) = #1/1/2001 6:00:00 AM#
```

MAKELINE(geometry1,geometry2)

MAKELINE(Geometry1,Geometry2)

Generates a line mark between two spatial points; useful for building origin-destination maps.

Example

```
MAKELINE(MAKEPOINT(OriginLat],[OriginLong) ),MAKEPOINT
([DestinationLat],[DestinationLong] ) )
```

```
MAKELINE(Geometry1, Geometry2 )
```

MAKEPOINT(Latitude,Longitude)

Converts data from latitude and longitude columns into spatial objects. You can use MAKEPOINT to spatially-enable a data source so that it can be joined with a spatial file using a spatial join. To use MAKEPOINT, your data must contain latitude and longitude coordinates.

Example

```
MAKEPOINT([AirportLatitude],[AirportLongitude])
```

MAKEPOINT(Xcoordinate,Ycoordinate,SRID)

Converts data from projected geographic coordinates into spatial objects. SRID is a spatial reference identifier that uses ESPG reference system codes to specify coordinate systems. If SRID is not specified, WGS84 is assumed and parameters are treated as latitude/longitude in degrees.

This function can only be created with a live connection and will continue to work when a data source is converted to an extract.

Example

```
MAKEPOINT([Xcoord],[Ycoord],3493)
```

MAKETIME(hour, minute, second)

Returns a date value constructed from the specified hour, minute, and second.

Available for Tableau Data Extracts. Check for availability in other data sources.

Example

```
MAKETIME(14, 52, 40) = #14:52:40#
```

MAX(a, b)

Returns the maximum of `a` and `b` (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, `MAX` finds the value that is highest in the sort sequence defined by the database for that column. It returns `Null` if either argument is `Null`.

Example

```
MAX ("Apple","Banana") = "Banana"
```

MAX(expression) or MAX(expr1, expr2)

Usually applied to numbers but also works on dates. Returns the maximum of `a` and `b` (`a` and `b` must be of the same type). Returns `Null` if either argument is `Null`.

Examples

`MAX(#2004-01-01# ,#2004-03-01#) = 2004-03-01 12:00:00 AM`

`MAX([ShipDate1], [ShipDate2])`

MAX(number, number)

Returns the maximum of the two arguments, which must be of the same type. Returns `Null` if either argument is `Null`. `MAX` can also be applied to a single field in an aggregate calculation.

Examples

`MAX(4,7)`

`MAX(Sales,Profit)`

`MAX([First Name],[Last Name])`

MEDIAN(expression)

Returns the median of an expression across all records. Median can only be used with numeric fields. Null values are ignored. This function is not available for workbooks created before Tableau Desktop 8.2 or that use legacy connections. It is also not available for connections using any of the following data sources:

- Access
- Amazon Redshift
- Cloudera Hadoop
- IBM DB2
- IBM PDA (Netezza)
- Microsoft SQL Server
- MySQL
- SAP HANA
- Teradata
- Vertica

For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.

MID(string, start, [length])

Returns the string starting at index position `start`. The first character in the string is position 1. If the optional argument `length` is added, the returned string includes only that number of characters.

Examples

```
MID("Calculation", 2) = "alculation"
```

```
MID("Calculation", 2, 5) ="alcul"
```

MIN(a, b)

Returns the minimum of `a` and `b` (which must be of the same type). This function is usually used to compare numbers, but also works on strings. With strings, `MIN` finds the value that is lowest in the sort sequence. It returns `Null` if either argument is `Null`.

Example

```
MIN ("Apple","Banana") = "Apple"
```

MIN(expression) or MIN(expr1, expr2)

Usually applied to numbers but also works on dates. Returns the minimum of `a` and `b` (`a` and `b` must be of the same type). Returns `Null` if either argument is `Null`.

Examples

```
MIN(#2004-01-01# ,#2004-03-01#) = 2004-01-01 12:00:00 AM
```

```
MIN([ShipDate1], [ShipDate2])
```

MIN(number, number)

Returns the minimum of the two arguments, which must be of the same type. Returns `Null` if either argument is `Null`. `MIN` can also be applied to a single field in an aggregate calculation.

Examples

```
MIN(4,7)
```

```
MIN(Sales,Profit)
```

```
MIN([First Name],[Last Name])
```

MODEL_PERCENTILE(target_expression, predictor_expression(s))

Returns the probability (between 0 and 1) of the expected value being less than or equal to the observed mark, defined by the target expression and other predictors. This is the Posterior Predictive Distribution Function, also known as the Cumulative Distribution Function (CDF).

Example

```
MODEL_PERCENTILE( SUM([Sales]),COUNT([Orders]))
```

MODEL_QUANTILE(quantile, target_expression, predictor_expression(s))

Returns a target numeric value within the probable range defined by the target expression and other predictors, at a specified quantile. This is the Posterior Predictive Quantile.

Example

```
MODEL_QUANTILE(0.5, SUM([Sales]),COUNT([Orders]))
```

MONTH(date)

Returns the month of the given date as an integer.

Example

```
MONTH(#2004-04-15#) = 4
```

NOW( )

Returns the current date and time.

The return varies depending on the nature of the connection:

- For a live, unpublished connection, NOW returns the data source server time.
- For a live, published connection, NOW returns the data source server time.
- For an unpublished extract, NOW returns the local system time.
- For a published extract, NOW returns the local time of the Tableau Server Data Engine. When there are multiple worker machines indifferent time zones, this can produce inconsistent results.

Example

```
NOW( ) = 2004-04-15 1:08:21 PM
```

PARSE_URL(string, url_part)

Note: *Supported only when connected to Hadoop Hive and Cloudera Impala.*

Returns a component of the given URL string where the component is defined by url_part. Valid url_part values include: 'HOST', 'PATH', 'QUERY', 'REF', 'PROTOCOL', 'AUTHORITY', 'FILE' and 'USERINFO'.

Example

PARSE_URL('http://www.tableau.com', 'HOST') = 'www.tableau.com'

PARSE_URL_QUERY(string, key)

Note: *Supported only when connected to Hadoop Hive and Cloudera Impala.*

Returns the value of the specified query parameter in the given URL string. The query parameter is defined by the key.

Example

PARSE_URL_QUERY('http://www.tableau.com?page=1&cat=4', 'page') = '1'

PERCENTILE(expression, number)

Returns the percentile value from the given expression corresponding to the specified number. The number must be between 0 and 1 (inclusive)—for example, 0.66, and must be a numeric constant.

This function is available for the following data sources.

- Non-legacy Microsoft Excel and Text File connections.
- Extracts and extract-only data source types (for example, Google Analytics, OData, or Salesforce).
- Sybase IQ 15.1 and later data sources.
- Oracle 10 and later data sources.
- Cloudera Hive and Hortonworks Hadoop Hive data sources.
- EXASOL 4.2 and later data sources.

For other data source types, you can extract your data into an extract file to use this function. See **Extract Your Data** on page 1014.

PI( )

Returns the numeric constant pi: 3.14159.

POWER(number, power)

Raises the number to the specified power.

Examples

`POWER(5,2) = 52 = 25`

`POWER(Temperature, 2)`

You can also use the ^ symbol:

`5^2 = POWER(5,2) = 25`

PREVIOUS_VALUE(expression)

Returns the value of this calculation in the previous row. Returns the given expression if the current row is the first row of the partition.

Example

`SUM([Profit]) * PREVIOUS_VALUE(1)` computes the running product of SUM(Profit).

RADIANS(number)

Converts the given number from degrees to radians.

Example

`RADIANS(180) = 3.14159`

RANK(expression, ['asc' | 'desc'])

Returns the standard competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

Example

The following image shows the effect of the various ranking functions (RANK, RANK_DENSE, RANK_MODIFIED, RANK_PERCENTILE, and RANK_UNIQUE) on a set of values. The data set contains information on 14 students (StudentA through StudentN); the **Age** column shows the current age of each student (all students are between 17 and 20 years of age). The remaining columns show the effect of each rank function on the set of age values, always assuming the default order (ascending or descending) for the function.

Student	Age	RANKofAge	RANK_DENSEofAge	RANK_MODIFIEDofAge	RANK_PERCENTILEofAge	RANK_UNIQUEofAge
StudentA	19	4	2	7	79%	4
StudentB	18	8	3	12	50%	8
StudentC	19	4	2	7	79%	5
StudentD	18	8	3	12	50%	9
StudentE	17	13	4	14	14%	13
StudentF	18	8	3	12	50%	10
StudentG	19	4	2	7	79%	6
StudentH	20	1	1	3	100%	1
StudentI	19	4	2	7	79%	7
StudentJ	20	1	1	3	100%	2
StudentK	20	1	1	3	100%	3
StudentL	17	13	4	14	14%	14
StudentM	18	8	3	12	50%	11
StudentN	18	8	3	12	50%	12

RANK_DENSE(expression, ['asc' | 'desc'])

Returns the dense rank for the current row in the partition. Identical values are assigned an identical rank, but no gaps are inserted into the number sequence. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (3, 2, 2, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_MODIFIED(expression, ['asc' | 'desc'])

Returns the modified competition rank for the current row in the partition. Identical values are assigned an identical rank. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 3, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_PERCENTILE(expression, ['asc' | 'desc'])

Returns the percentile rank for the current row in the partition. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is ascending.

With this function, the set of values (6, 9, 9, 14) would be ranked (0.25, 0.75, 0.75, 1.00).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RANK_UNIQUE(expression, ['asc' | 'desc'])

Returns the unique rank for the current row in the partition. Identical values are assigned different ranks. Use the optional `'asc' | 'desc'` argument to specify ascending or descending order. The default is descending.

With this function, the set of values (6, 9, 9, 14) would be ranked (4, 2, 3, 1).

Nulls are ignored in ranking functions. They are not numbered and they do not count against the total number of records in percentile rank calculations.

For information on different ranking options, see **Rank calculation** on page 2431.

RAWSQL_BOOL(“sql_expr”, [arg1], …[argN])

Returns a Boolean result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQL_BOOL(“IIF(%1 > %2, True, False)”, [Sales], [Profit])

RAWSQL_DATE(“sql_expr”, [arg1], …[argN])

Returns a Date result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Order Date].

```
RAWSQL_DATE("%1", [Order Date])
```

RAWSQL_DATETIME("sql_expr", [arg1], …[argN])

Returns a Date and Time result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

RAWSQL_DATETIME("MIN(%1)", [Delivery Date])

RAWSQL_INT("sql_expr", [arg1], …[argN])

Returns an integer result from a given SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

```
RAWSQL_INT("500 + %1", [Sales])
```

RAWSQL_REAL("sql_expr", [arg1], …[argN])

Returns a numeric result from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

```
RAWSQL_REAL("-123.98 * %1", [Sales])
```

RAWSQL_SPATIAL

Returns a Spatial from a given SQL expression that is passed directly to the underlying data source. Use %n in the SQL expression as a substitution syntax for database values.

Example

In this example, %1 is equal to [Geometry].

```
RAWSQL_SPATIAL("%1", [Geometry])
```

RAWSQL_STR(“sql_expr”, [arg1], …[argN])

Returns a string from a given SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Customer Name].

Example

```
RAWSQL_STR(“%1”, [Customer Name])
```

RAWSQLAGG_BOOL(“sql_expr”, [arg1], …[argN])

Returns a Boolean result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values.

Example

In the example, %1 is equal to [Sales] and %2 is equal to [Profit].

RAWSQLAGG_BOOL(“SUM(%1) >SUM(%2)”, [Sales], [Profit])

RAWSQLAGG_DATE(“sql_expr”, [arg1], …[argN])

Returns a Date result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Order Date].

Example

```
RAWSQLAGG_DATE(“MAX(%1)”, [Order Date])
```

RAWSQLAGG_DATETIME(“sql_expr”, [arg1], …[argN])

Returns a Date and Time result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Delivery Date].

Example

```
RAWSQLAGG_DATETIME("MIN(%1)", [Delivery Date])
```

RAWSQLAGG_INT("sql_expr", [arg1,] …[argN])

Returns an integer result from a given aggregate SQL expression. The SQL expression is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales].

Example

```
RAWSQLAGG_INT("500 + SUM(%1)", [Sales])
```

RAWSQLAGG_REAL("sql_expr", [arg1,] …[argN])

Returns a numeric result from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Sales]

Example

```
RAWSQLAGG_REAL("SUM( %1)", [Sales])
```

RAWSQLAGG_STR("sql_expr", [arg1,] …[argN])

Returns a string from a given aggregate SQL expression that is passed directly to the underlying database. Use %n in the SQL expression as a substitution syntax for database values. In this example, %1 is equal to [Discount].

Example

```
RAWSQLAGG_STR("AVG(%1)", [Discount])
```

REGEXP_REPLACE(string, pattern, replacement)

Returns a copy of the given string where the regular expression pattern is replaced by the replacement string. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern and the replacement must be constants.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_REPLACE('abc 123', '\s', '-') = 'abc-123'

REGEXP_MATCH(string, pattern)

Returns true if a substring of the specified string matches the regular expression pattern. This function is available for Text File, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Impala 2.3.0 (through Cloudera Hadoop data sources), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_MATCH('-([1234].[The.Market])-','\[\s*(\w*\.)(\w*\s*\])')=true

REGEXP_EXTRACT(string, pattern)

Returns the portion of the string that matches the regular expression pattern. This function is available for Text File, Hadoop Hive, Google BigQuery, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), Snowflake, and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for

Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT('abc 123', '[a-z]+\s+(\d+)') = '123'

REGEXP_EXTRACT_NTH(string, pattern, index)

Returns the portion of the string that matches the regular expression pattern. The substring is matched to the nth capturing group, where n is the given index. If index is 0, the entire string is returned. This function is available for Text File, PostgreSQL, Tableau Data Extract, Microsoft Excel, Salesforce, Vertica, Pivotal Greenplum, Teradata (version 14.1 and above), and Oracle data sources.

For Tableau data extracts, the pattern must be a constant.

For information on regular expression syntax, see your data source's documentation. For Tableau extracts, regular expression syntax conforms to the standards of the ICU (International Components for Unicode), an open source project of mature C/C++ and Java libraries for Unicode support, software internationalization, and software globalization. See the Regular Expressions page in the online ICU User Guide.

Example

REGEXP_EXTRACT_NTH('abc 123', '([a-z]+)\s+(\d+)', 2) = '123'

REPLACE(string, substring, replacement)

Searches `string` for `substring` and replaces it with `replacement`. If `substring` is not found, the string is not changed.

Example

```
REPLACE("Version8.5", "8.5", "9.0") = "Version9.0"
```

RIGHT(string, number)

Returns the right-most number of characters in `string`.

Example

```
RIGHT("Calculation", 4) = "tion"
```

ROUND(number, [decimals])

Rounds numbers to a specified number of digits. The `decimals` argument specifies how many decimal points of precision to include in the final result. If `decimals` is omitted, `number` is rounded to the nearest integer.

Example

This example rounds every `Sales` value to an integer:

```
ROUND(Sales)
```

Some databases, such as SQL Server, allow specification of a negative `length`, where -1 rounds `number` to 10's, -2 rounds to 100's, and so on. This is not true of all databases. For example, it is not true of Excel or Access.

RTRIM(string)

Returns `string` with any trailing spaces removed.

Example

```
RTRIM(" Calculation ") = " Calculation"
```

RTRIM_THIS(string, string)

***Note**: Supported only when connected to Google BigQuery*

Returns the first string with any trailing occurrence of the second string removed.

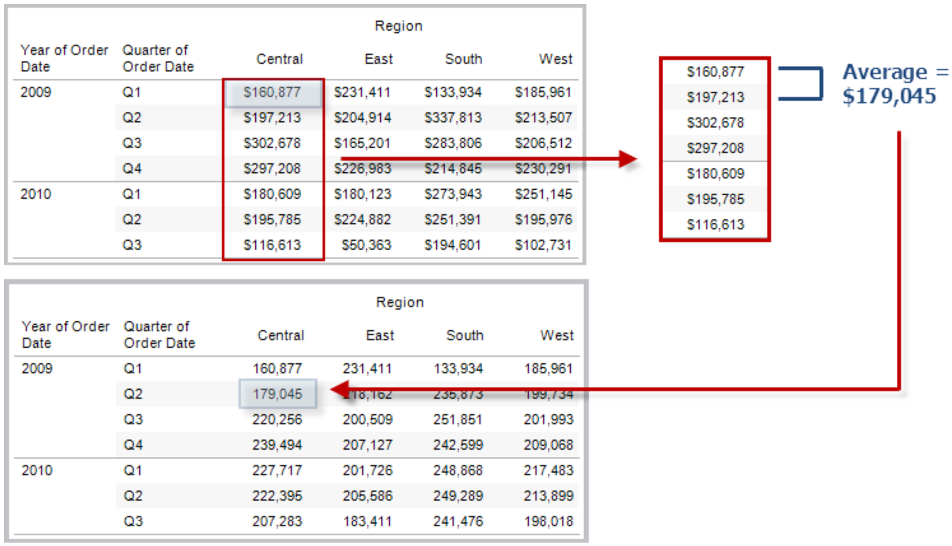
Example

RTRIM_THIS('[-Market-]','-]') = '[-Market'

RUNNING_AVG(expression)

Returns the running average of the given expression, from the first row in the partition to the current row.

The view below shows quarterly sales. When `RUNNING_AVG(SUM([Sales])` is computed within the Date partition, the result is a running average of the sales values for each quarter.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	179,045	[illegible]	235,873	199,734
	Q3	220,256	200,509	251,851	201,993
	Q4	239,494	207,127	242,599	209,068
2010	Q1	227,717	201,726	248,868	217,483
	Q2	222,395	205,586	249,289	213,899
	Q3	207,283	183,411	241,476	198,018

Example

`RUNNING_AVG(SUM([Profit]))` computes the running average of SUM(Profit).

RUNNING_COUNT(expression)

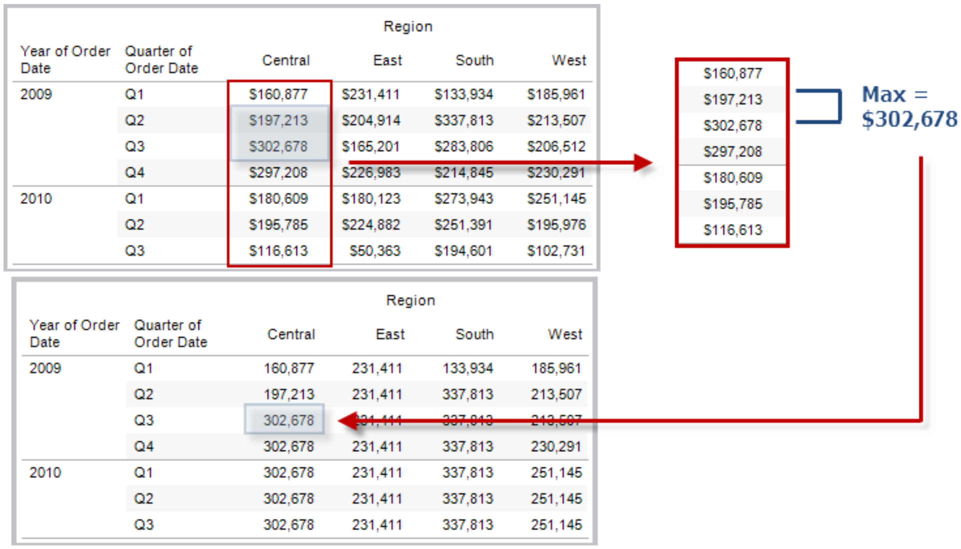
Returns the running count of the given expression, from the first row in the partition to the current row.

Example

`RUNNING_COUNT(SUM([Profit]))` computes the running count of SUM(Profit).

RUNNING_MAX(expression)

Returns the running maximum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

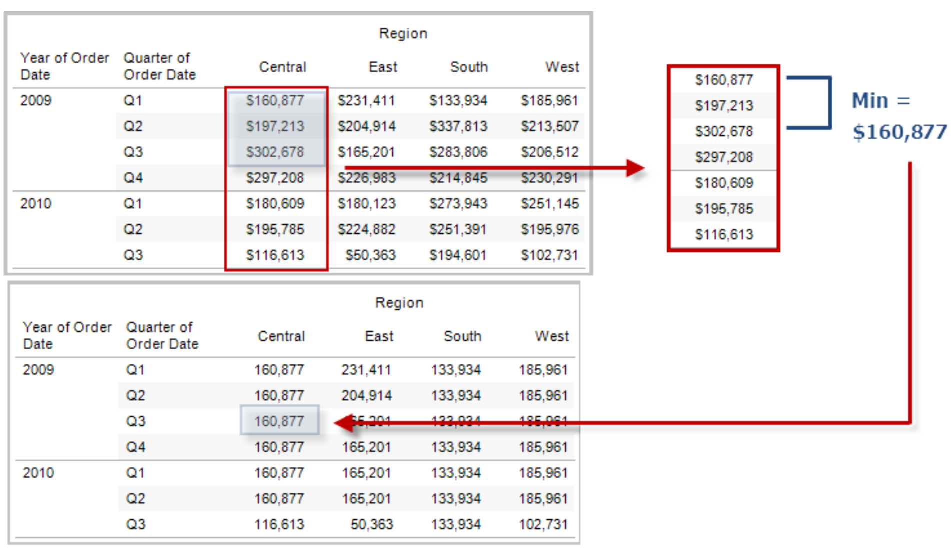
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	197,213	231,411	337,813	213,507
	Q3	302,678	231,411	337,813	213,507
	Q4	302,678	231,411	337,813	230,291
2010	Q1	302,678	231,411	337,813	251,145
	Q2	302,678	231,411	337,813	251,145
	Q3	302,678	231,411	337,813	251,145

Example

`RUNNING_MAX(SUM([Profit]))` computes the running maximum of SUM(Profit).

RUNNING_MIN(expression)

Returns the running minimum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

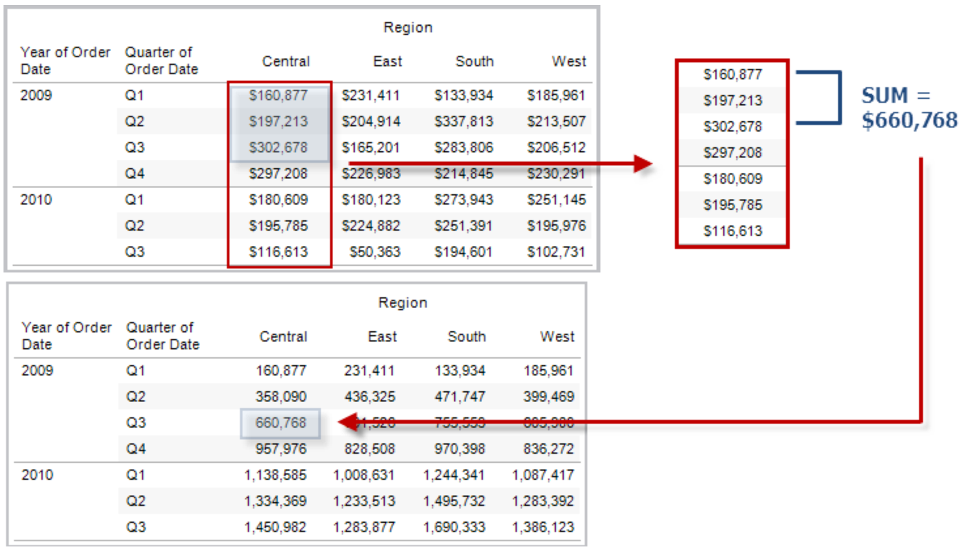
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	160,877	204,914	133,934	185,961
	Q3	160,877	165,201	133,934	185,961
	Q4	160,877	165,201	133,934	185,961
2010	Q1	160,877	165,201	133,934	185,961
	Q2	160,877	165,201	133,934	185,961
	Q3	116,613	50,363	133,934	102,731

Example

`RUNNING_MIN(SUM([Profit]))` computes the running minimum of SUM(Profit).

RUNNING_SUM(expression)

Returns the running sum of the given expression, from the first row in the partition to the current row.



Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Year of Order Date	Quarter of Order Date	Region: Central	East	South	West
2009	Q1	160,877	231,411	133,934	185,961
	Q2	358,090	436,325	471,747	399,469
	Q3	660,768	[illegible]	[illegible]	[illegible]
	Q4	957,976	828,508	970,398	836,272
2010	Q1	1,138,585	1,008,631	1,244,341	1,087,417
	Q2	1,334,369	1,233,513	1,495,732	1,283,392
	Q3	1,450,982	1,283,877	1,690,333	1,386,123

Example

`RUNNING_SUM(SUM([Profit]))` computes the running sum of SUM(Profit)

SCRIPT_BOOL

Returns a Boolean result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.).

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_BOOL("is.finite(.arg1)", SUM([Profit]))
```

The next example returns True for store IDs in Washington state, and False otherwise. This example could be the definition for a calculated field titled IsStoreInWA.

```
SCRIPT_BOOL('grepl(".*_WA", .arg1, perl=TRUE)',ATTR([Store ID]))
```

A command for Python would take this form:

```
SCRIPT_BOOL("return map(lambda x : x > 0, _arg1)", SUM([Profit]))
```

SCRIPT_INT

Returns an integer result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_INT("is.finite(.arg1)", SUM([Profit]))
```

In the next example, k-means clustering is used to create three clusters:

```
SCRIPT_INT('result <- kmeans(data.frame(.arg1,.arg2,.arg3,.arg4),
3);result$cluster;', SUM([Petal length]), SUM([Petal width]),SUM
([Sepal length]),SUM([Sepal width]))
```

A command for Python would take this form:

```
SCRIPT_INT("return map(lambda x : int(x * 5), _arg1)", SUM
([Profit]))
```

SCRIPT_REAL

Returns a real result from the specified expression. The expression is passed directly to a running external service instance. In

R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_REAL("is.finite(.arg1)", SUM([Profit]))
```

The next example converts temperature values from Celsius to Fahrenheit.

```
SCRIPT_REAL('library(udunits2);ud.convert(.arg1, "celsius",
"degree_fahrenheit")',AVG([Temperature]))
```

A command for Python would take this form:

```
SCRIPT_REAL("return map(lambda x : x * 0.5, _arg1)", SUM
([Profit]))
```

SCRIPT_STR

Returns a string result from the specified expression. The expression is passed directly to a running external service instance.

In R expressions, use .arg*n* (with a leading period) to reference parameters (.arg1, .arg2, etc.)

In Python expressions, use _arg*n* (with a leading underscore).

Examples

In this R example, .arg1 is equal to SUM([Profit]):

```
SCRIPT_STR("is.finite(.arg1)", SUM([Profit]))
```

The next example extracts a state abbreviation from a more complicated string (in the original form 13XSL_CA, A13_WA):

```
SCRIPT_STR('gsub(".*_", "", .arg1)',ATTR([Store ID]))
```

A command for Python would take this form:

```
SCRIPT_STR("return map(lambda x : x[:2], _arg1)", ATTR([Region]))
```

SIGN(number)

Returns the sign of a number: The possible return values are -1 if the number is negative, 0 if the number is zero, or 1 if the number is positive.

Example

If the average of the profit field is negative, then

```
SIGN(AVG(Profit)) = -1
```

SIN(number)

Returns the sine of an angle. Specify the angle in radians.

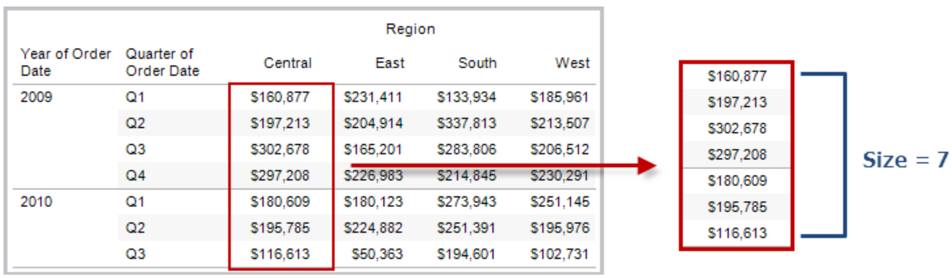
Example

```
SIN(0) = 1.0
```

`SIN(PI( )/4) = 0.707106781186548`

SIZE()

Returns the number of rows in the partition. For example, the view below shows quarterly sales. Within the Date partition, there are seven rows so the Size() of the Date partition is 7.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

Example

`SIZE() = 5` when the current partition contains five rows.

SPACE(number)

Returns a string that is composed of the specified `number` of repeated spaces.

Example

`SPACE(1) = " "`

SPLIT(string, delimiter, token number)

Returns a substring from a string, using a delimiter character to divide the string into a sequence of tokens.

The string is interpreted as an alternating sequence of delimiters and tokens. So for the string `abc-defgh-i-jkl`, where the delimiter character is '-', the tokens are abc, defgh, i, and jlk. Think of these as tokens 1 through 4. SPLIT returns the token corresponding to the token number. When the token number is positive, tokens are counted starting from the left end of the string; when the token number is negative, tokens are counted starting from the right.

Examples

`SPLIT ('a-b-c-d', '-', 2) = 'b'`

`SPLIT ('a|b|c|d', '|', -2) = 'c'`

Note: The split and custom split commands are available for the following data sources types: Tableau data extracts, Microsoft Excel, Text File, PDF File, Salesforce, OData, Microsoft Azure Market Place, Google Analytics, Vertica, Oracle, MySQL, PostgreSQL, Teradata, Amazon Redshift, Aster Data, Google Big Query, Cloudera Hadoop Hive, Hortonworks Hive, and Microsoft SQL Server.

Some data sources impose limits on splitting string. The following table shows which data sources support negative token numbers (splitting from the right) and whether there is a limit on the number of splits allow per data source. A SPLIT function that specifies a negative token number and would be legal with other data sources will return this error with these data sources: “Splitting from right is not support by the data source.”

Data Source	Left/Right Constraints	Maximum Number of Splits	Version Limitations
Tableau Data Extract	Both	Infinite	
Microsoft Excel	Both	Infinite	
Text file	Both	Infinite	
Salesforce	Both	Infinite	
OData	Both	Infinite	
Google Analytics	Both	Infinite	
Tableau Data Server	Both	Infinite	Supported in version 9.0.
Vertica	Left only	10	
Oracle	Left only	10	
MySQL	Both	10	
PostgreSQL	Left only prior to version 9.0; both for version 9.0 and above	10	

Teradata	Left only	10	Version 14 and later
Amazon Redshift	Left only	10	
Aster Database	Left only	10	
Google BigQuery	Left only	10	
Hortonworks Hadoop Hive	Left only	10	
Cloudera Hadoop	Left only	10	Impala supported starting in version 2.3.0.
Microsoft SQL Server	Both	10	2008 and later

SQRT(number)

Returns the square root of a number.

Example

`SQRT(25) = 5`

SQUARE(number)

Returns the square of a number.

Example

`SQUARE(5) = 25`

STARTSWITH(string, substring)

Returns true if `string` starts with `substring`. Leading white spaces are ignored.

Example

`STARTSWITH(“Joker”, “Jo”) = true`

STDEV(expression)

Returns the statistical standard deviation of all values in the given expression based on a sample of the population.

STDEVP(expression)

Returns the statistical standard deviation of all values in the given expression based on a biased population.

STR(expression)

Casts its argument as a string.

Example

`STR([Age])` takes all of the values in the measure called `Age` and converts them to strings.

SUM(expression)

Returns the sum of all values in the expression. SUM can be used with numeric fields only. Null values are ignored.

TAN(number)

Returns the tangent of an angle. Specify the angle in radians..

Example

```
TAN(PI ( )/4) = 1.0
```

TIMESTAMP_TO_USEC(expression)

***Note**: Supported only when connected to Google BigQuery*

Converts a TIMESTAMP data type to a UNIX timestamp in microseconds.

Example

TIMESTAMP_TO_USEC(#2012-10-01 01:02:03#)=1349053323000000

TLD(string_url)

***Note**: Supported only when connected to Google BigQuery*

Given a URL string, returns the top level domain plus any country domain in the URL.

Example

TLD('http://www.google.com:80/index.html') = '.com'

TLD('http://www.google.co.uk:80/index.html') = '.co.uk'

TODAY( )

Returns the current date.

Example

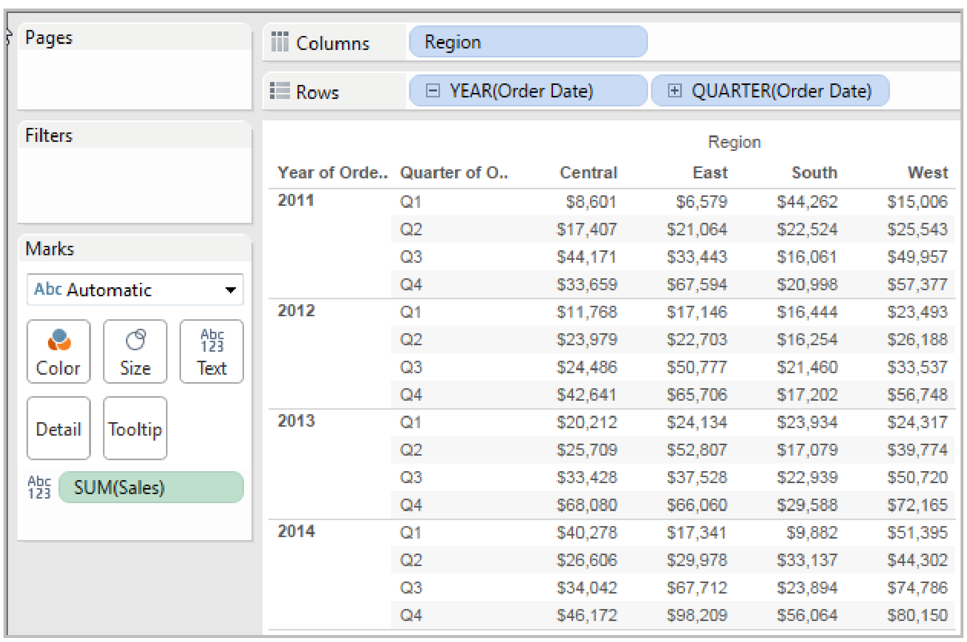
```
TODAY( ) = 2004-04-15
```

TOTAL(expression)

Returns the total for the given expression in a table calculation partition.

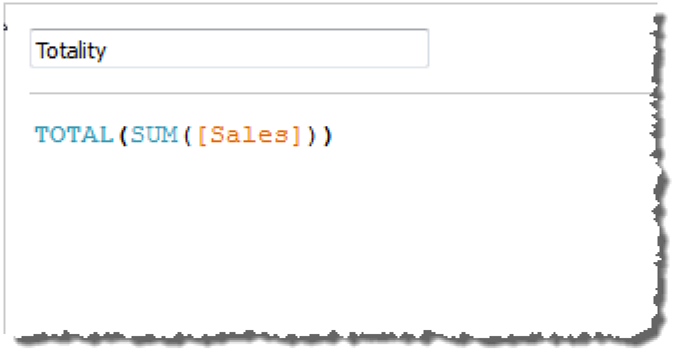
Example

Assume you are starting with this view:

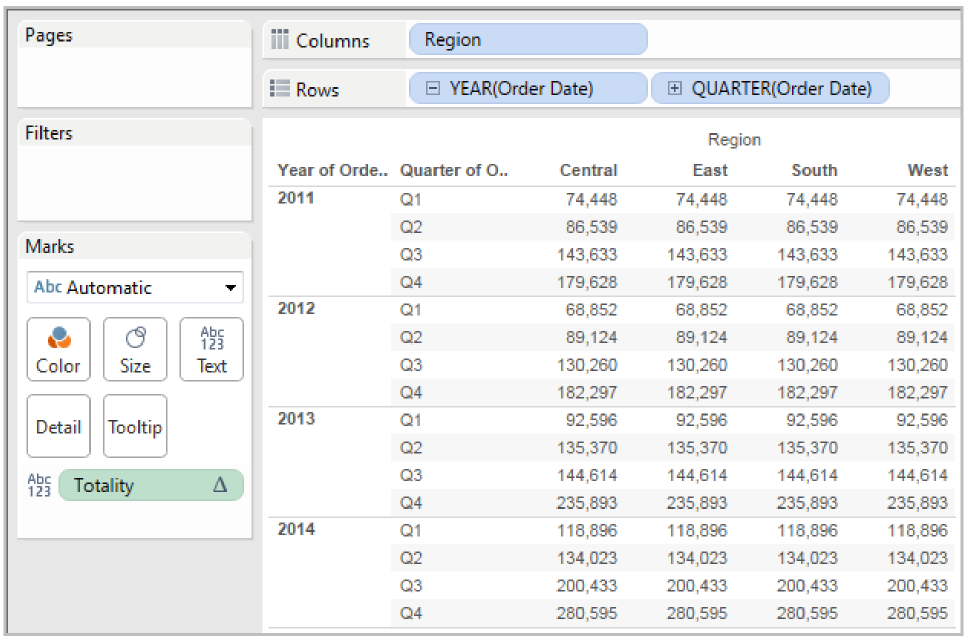


		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	$8,601	$6,579	$44,262	$15,006
	Q2	$17,407	$21,064	$22,524	$25,543
	Q3	$44,171	$33,443	$16,061	$49,957
	Q4	$33,659	$67,594	$20,998	$57,377
2012	Q1	$11,768	$17,146	$16,444	$23,493
	Q2	$23,979	$22,703	$16,254	$26,188
	Q3	$24,486	$50,777	$21,460	$33,537
	Q4	$42,641	$65,706	$17,202	$56,748
2013	Q1	$20,212	$24,134	$23,934	$24,317
	Q2	$25,709	$52,807	$17,079	$39,774
	Q3	$33,428	$37,528	$22,939	$50,720
	Q4	$68,080	$66,060	$29,588	$72,165
2014	Q1	$40,278	$17,341	$9,882	$51,395
	Q2	$26,606	$29,978	$33,137	$44,302
	Q3	$34,042	$67,712	$23,894	$74,786
	Q4	$46,172	$98,209	$56,064	$80,150

You open the calculation editor and create a new field which you name **Totality**:

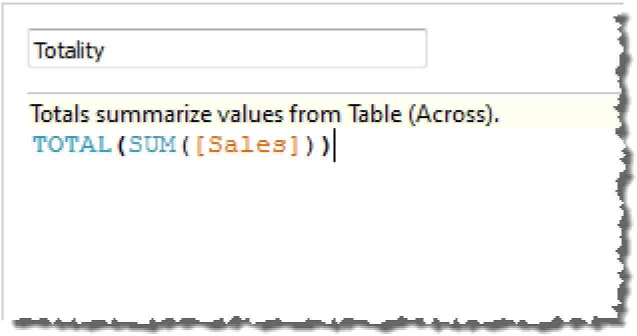


You then drop **Totality** on Text, to replace **SUM(Sales)**. Your view changes such that it sums values based on the default **Compute Using** value:



		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	74,448	74,448	74,448	74,448
	Q2	86,539	86,539	86,539	86,539
	Q3	143,633	143,633	143,633	143,633
	Q4	179,628	179,628	179,628	179,628
2012	Q1	68,852	68,852	68,852	68,852
	Q2	89,124	89,124	89,124	89,124
	Q3	130,260	130,260	130,260	130,260
	Q4	182,297	182,297	182,297	182,297
2013	Q1	92,596	92,596	92,596	92,596
	Q2	135,370	135,370	135,370	135,370
	Q3	144,614	144,614	144,614	144,614
	Q4	235,893	235,893	235,893	235,893
2014	Q1	118,896	118,896	118,896	118,896
	Q2	134,023	134,023	134,023	134,023
	Q3	200,433	200,433	200,433	200,433
	Q4	280,595	280,595	280,595	280,595

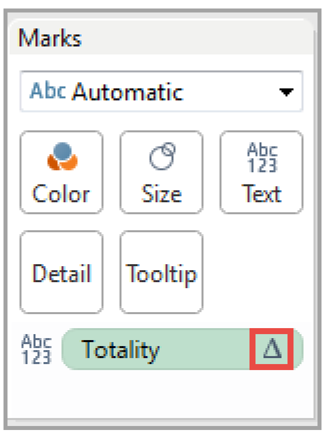
This raises the question, What is the default **Compute Using** value? If you right-click (Control-click on a Mac) **Totality** in the Data pane and choose **Edit**, there is now an additional bit of information available:



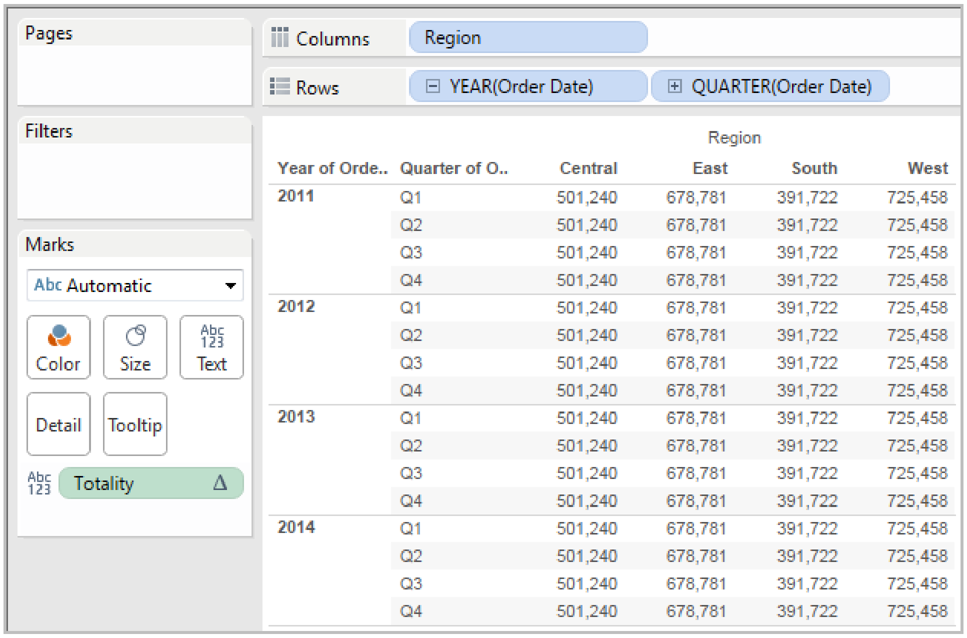
The default **Compute Using** value is **Table (Across)**. The result is that **Totality** is summing the values across each row of your table. Thus, the value that you see across each row is the sum of the values from the original version of the table.

The values in the 2011/Q1 row in the original table were $8601, $6579, $44262, and $15006. The values in the table after **Totality** replaces **SUM(Sales)** are all $74,448, which is the sum of the four original values.

Notice the triangle next to Totality after you drop it on Text:



This indicates that this field is using a table calculation. You can right-click the field and choose **Edit Table Calculation** to redirect your function to a different **Compute Using** value. For example, you could set it to **Table (Down)**. In that case, your table would look like this:



		Region			
Year of Orde..	Quarter of O..	Central	East	South	West
2011	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2012	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2013	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458
2014	Q1	501,240	678,781	391,722	725,458
	Q2	501,240	678,781	391,722	725,458
	Q3	501,240	678,781	391,722	725,458
	Q4	501,240	678,781	391,722	725,458

TRIM(string)

Returns the string with leading and trailing spaces removed. For example, `TRIM(" Calculation ") = "Calculation"`

UPPER(string)

Returns string, with all characters uppercase.

Example

```
UPPER("Calculation") = "CALCULATION"
```

USEC_TO_TIMESTAMP(expression)

***Note**: Supported only when connected to Google BigQuery*

Converts a UNIX timestamp in microsseconds to a TIMESTAMP data type.

Example

USEC_TO_TIMESTAMP(1349053323000000) = #2012-10-01 01:02:03#

USERDOMAIN()

Returns the domain for the current user when the user is signed on to Tableau Server. Returns the Windows domain if the Tableau Desktop user is on a domain. Otherwise this function returns a null string.

Example

```
[Manager]=USERNAME() AND [Domain]=USERDOMAIN()
```

USERNAME( )

Returns the username for the current user. This is the Tableau Server or Tableau Online username when the user is signed in; otherwise it is the local or network username for the Tableau Desktop user.

Example

```
[Manager]=USERNAME( )
```

If the manager dhallsten was signed in, this function would only return True when the Manager field in the view is dhallsten. When used as a filter this calculated field can be used to create a user filter that only shows data that is relevant to the person signed in to the server.

VAR(expression)

Returns the statistical variance of all values in the given expression based on a sample of the population.

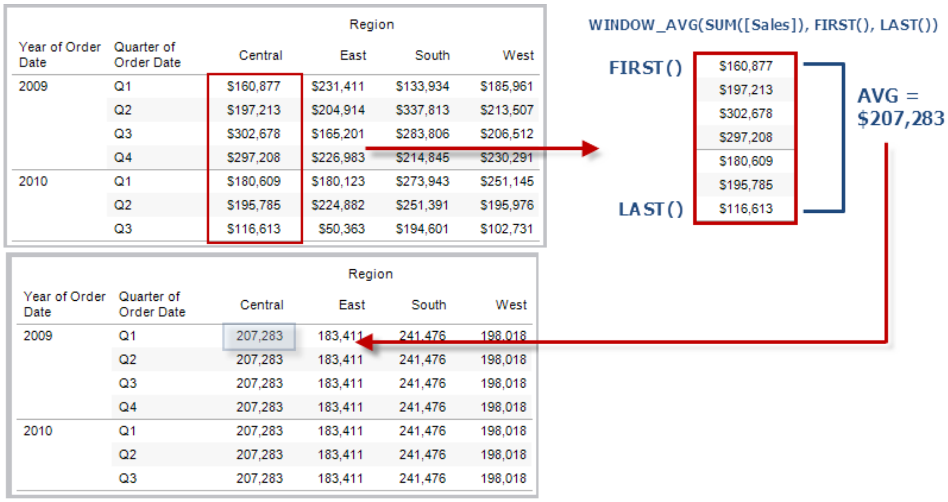
VARP(expression)

Returns the statistical variance of all values in the given expression on the entire population.

WINDOW_AVG(expression, [start, end])

Returns the average of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window average within the Date partition returns the average sales across all dates.



Example

`WINDOW_AVG(SUM([Profit]), FIRST()+1, 0)` computes the average of SUM(Profit) from the second row to the current row.

WINDOW_CORR(expression1, expression2, [start, end])

Returns the Pearson correlation coefficient of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

The Pearson correlation measures the linear relationship between two variables. Results range from -1 to +1 inclusive, where 1 denotes an exact positive linear relationship, as when a positive change in one variable implies a positive change of corresponding magnitude in the other, 0 denotes no linear relationship between the variance, and −1 is an exact negative relationship.

There is an equivalent aggregation fuction: CORR.

Example

The following formula returns the Pearson correlation of **SUM(Profit)** and **SUM(Sales)** from the five previous rows to the current row.

```
WINDOW_CORR(SUM[Profit]), SUM([Sales]), -5, 0)
```

WINDOW_COUNT(expression, [start, end])

Returns the count of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_COUNT(SUM([Profit]), FIRST()+1, 0)` computes the count of SUM(Profit) from the second row to the current row

WINDOW_COVAR(expression1, expression2, [start, end])

Returns the *sample covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end arguments are omitted, the window is the entire partition.

Sample covariance uses the number of non-null data points n - 1 to normalize the covariance calculation, rather than n, which is used by the population covariance (with the WINDOW_COVARP function). Sample covariance is the appropriate choice when the data is a random sample that is being used to estimate the covariance for a larger population.

There is an equivalent aggregation fuction: COVAR.

Example

The following formula returns the sample covariance of **SUM(Profit)** and **SUM(Sales)** from the two previous rows to the current row.

```
WINDOW_COVAR(SUM([Profit]), SUM([Sales]), -2, 0)
```

WINDOW_COVARP(expression1, expression2, [start, end])

Returns the *population covariance* of two expressions within the window. The window is defined as offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If start and end are omitted, the entire partition is used.

Population covariance is sample covariance multiplied by (n-1)/n, where n is the total number of non-null data points. Population covariance is the appropriate choice when there is data available for all items of interest as opposed to when there is only a random subset of items, in which case sample covariance (with the WINDOW_COVAR function) is appropriate.

There is an equivalent aggregation fuction: COVARP. See **Tableau Functions (Alphabetical)** on page 2342.

Example

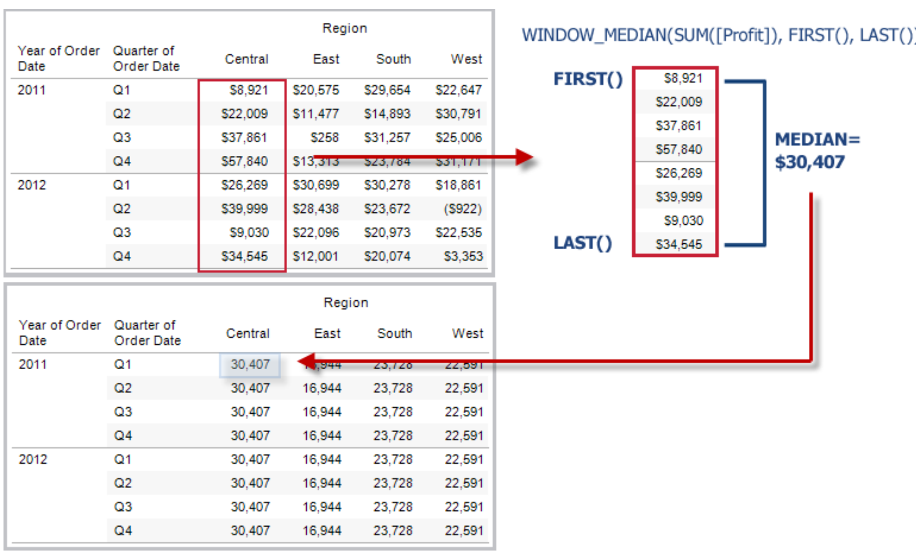
The following formula returns the population covariance of **SUM(Profit)** and **SUM(Sales)** from the two previous rows to the current row.

```
WINDOW_COVARP(SUM([Profit]), SUM([Sales]), -2, 0)
```

WINDOW_MEDIAN(expression, [start, end])

Returns the median of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly profit. A window median within the Date partition returns the median profit across all dates.



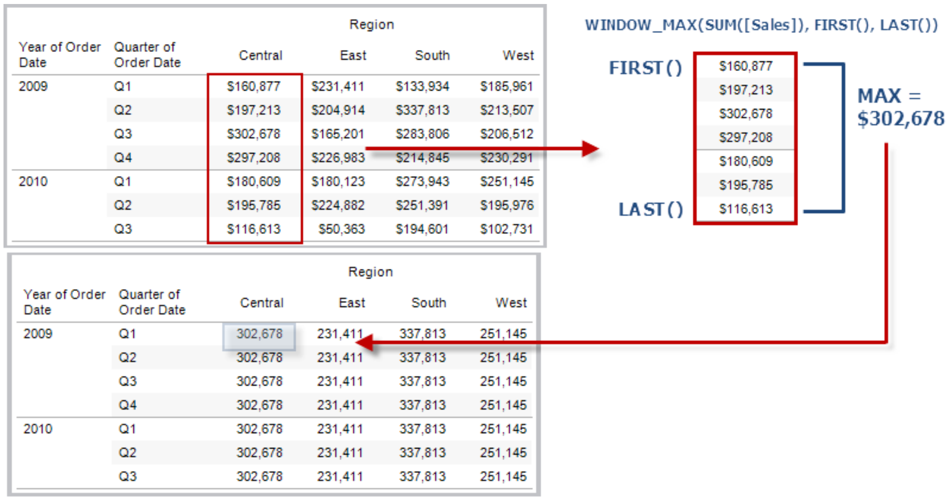
Example

`WINDOW_MEDIAN(SUM([Profit]), FIRST()+1, 0)` computes the median of SUM (Profit) from the second row to the current row.

WINDOW_MAX(expression, [start, end])

Returns the maximum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window maximum within the Date partition returns the maximum sales across all dates.



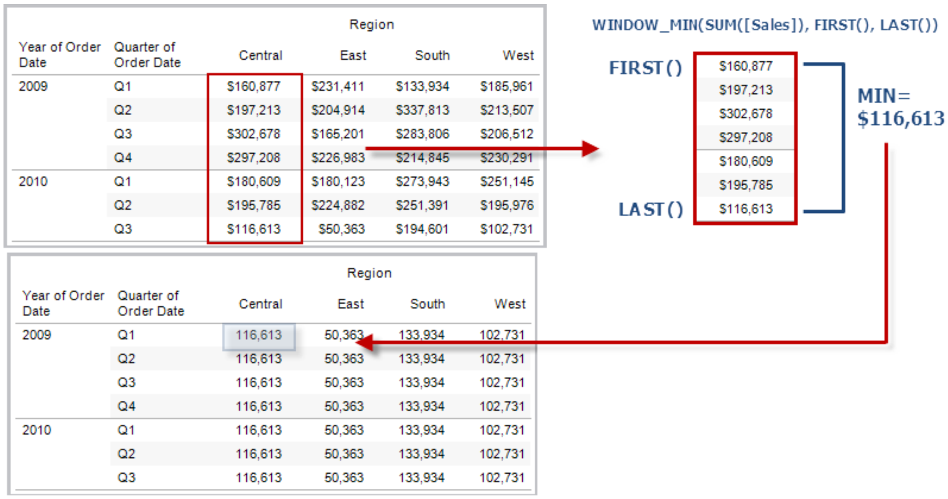
Example

`WINDOW_MAX(SUM([Profit]), FIRST()+1, 0)` computes the maximum of SUM (Profit) from the second row to the current row.

WINDOW_MIN(expression, [start, end])

Returns the minimum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window minimum within the Date partition returns the minimum sales across all dates.



Example

`WINDOW_MIN(SUM([Profit]), FIRST()+1, 0)` computes the minimum of SUM (Profit) from the second row to the current row.

WINDOW_PERCENTILE(expression, number, [start, end])

Returns the value corresponding to the specified percentile within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_PERCENTILE(SUM([Profit]), 0.75, -2, 0)` returns the 75th percentile for SUM(Profit) from the two previous rows to the current row.

WINDOW_STDEV(expression, [start, end])

Returns the sample standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_STDEV(SUM([Profit]), FIRST()+1, 0)` computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_STDEVP(expression, [start, end])

Returns the biased standard deviation of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

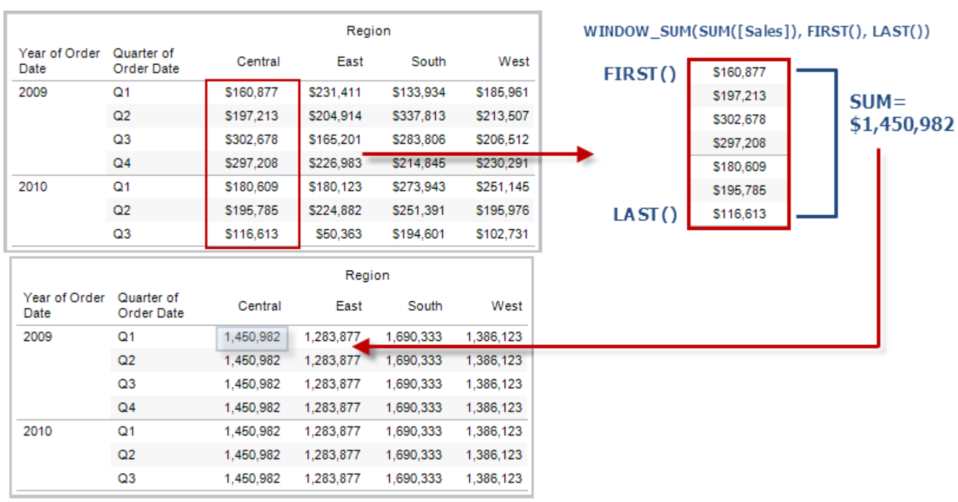
Example

`WINDOW_STDEVP(SUM([Profit]), FIRST()+1, 0)` computes the standard deviation of SUM(Profit) from the second row to the current row.

WINDOW_SUM(expression, [start, end])

Returns the sum of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

For example, the view below shows quarterly sales. A window sum computed within the Date partition returns the summation of sales across all quarters.



		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	$160,877	$231,411	$133,934	$185,961
	Q2	$197,213	$204,914	$337,813	$213,507
	Q3	$302,678	$165,201	$283,806	$206,512
	Q4	$297,208	$226,983	$214,845	$230,291
2010	Q1	$180,609	$180,123	$273,943	$251,145
	Q2	$195,785	$224,882	$251,391	$195,976
	Q3	$116,613	$50,363	$194,601	$102,731

		Region			
Year of Order Date	Quarter of Order Date	Central	East	South	West
2009	Q1	1,450,982	1,283,877	1,690,333	1,386,123
	Q2	1,450,982	1,283,877	1,690,333	1,386,123
	Q3	1,450,982	1,283,877	1,690,333	1,386,123
	Q4	1,450,982	1,283,877	1,690,333	1,386,123
2010	Q1	1,450,982	1,283,877	1,690,333	1,386,123
	Q2	1,450,982	1,283,877	1,690,333	1,386,123
	Q3	1,450,982	1,283,877	1,690,333	1,386,123

Example

`WINDOW_SUM(SUM([Profit]), FIRST()+1, 0)` computes the sum of SUM(Profit) from the second row to the current row.

WINDOW_VAR(expression, [start, end])

Returns the sample variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VAR((SUM([Profit])), FIRST()+1, 0)` computes the variance of SUM (Profit) from the second row to the current row.

WINDOW_VARP(expression, [start, end])

Returns the biased variance of the expression within the window. The window is defined by means of offsets from the current row. Use FIRST()+n and LAST()-n for offsets from the first or last row in the partition. If the start and end are omitted, the entire partition is used.

Example

`WINDOW_VARP(SUM([Profit]), FIRST()+1, 0)` computes the variance of SUM (Profit) from the second row to the current row.

XPATH_BOOLEAN(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns true if the XPath expression matches a node or evaluates to true.

Example

XPATH_BOOLEAN('<values> <value id="0">1</value><value id="1">5</value>', 'values/value [@id="1"] = 5') = true

XPATH_DOUBLE(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns the floating-point value of the XPath expression.

Example

XPATH_DOUBLE('<values><value>1.0</value><value>5.5</value> </values>', 'sum (value/*)') = 6.5

XPATH_FLOAT(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns the floating-point value of the XPath expression.

Example

XPATH_FLOAT('<values><value>1.0</value><value>5.5</value> </values>','sum(value/*)') = 6.5

XPATH_INT(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_INT('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_LONG(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_LONG('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_SHORT(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns the numerical value of the XPath expression, or zero if the XPath expression cannot evaluate to a number.

Example

XPATH_SHORT('<values><value>1</value><value>5</value> </values>','sum(value/*)') = 6

XPATH_STRING(XML string, XPath expression string)

Note: *Supported only when connected to Hadoop Hive*

Returns the text of the first matching node.

Example

XPATH_STRING('<sites ><url domain="org">http://www.w3.org</url> <url domain="com">http://www.tableau.com</url></sites>', 'sites/url[@domain="com"]') = 'http://www.tableau.com'

YEAR (date)

Returns the year of the given date as an integer.

Example

```
YEAR(#2004-04-15#) = 2004
```

ZN(expression)

Returns the expression if it is not null, otherwise returns zero. Use this function to use zero values instead of null values.

Example

```
ZN([Profit]) = [Profit]
```

Want to learn more about functions?

Read the functions topics.

See also

Tableau Functions (by Category) on page 2267

Functions in Tableau on page 2147

Transform Values with Table Calculations

This article explains the basics of table calculations and how to create them in Tableau.

What is a table calculation?

For a 4 minute introduction to table calculations in Tableau, click this video link.

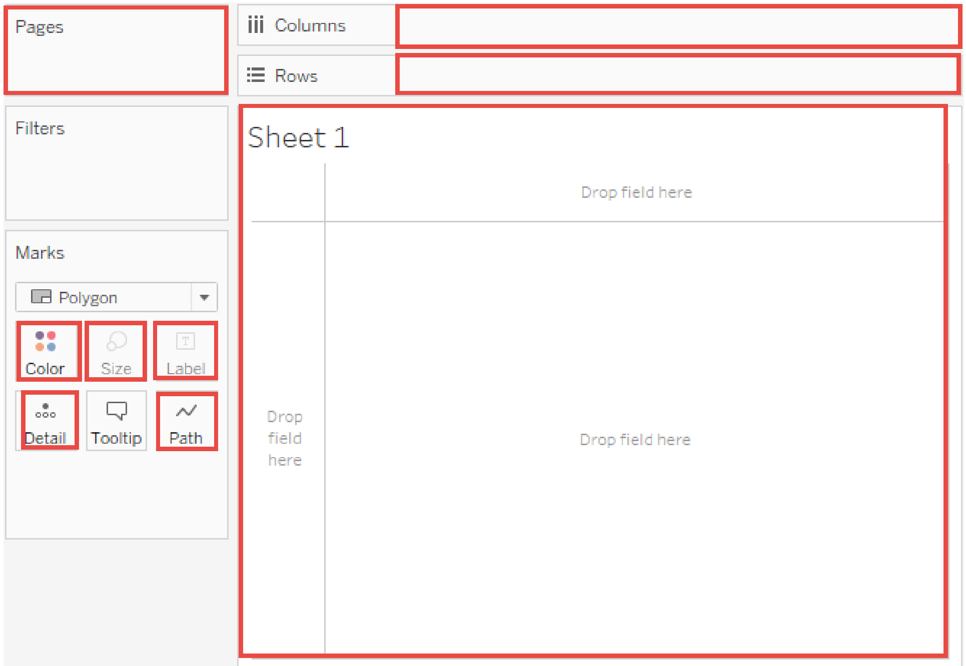
For related resources and videos, see Intro to Tableau Calculations. Use your tableau.com account to sign in.

A table calculation is a transformation you apply to the values in a visualization. Table calculations are a special type of calculated field that computes on the local data in Tableau. They are calculated based on what is currently in the visualization and do not consider any measures or dimensions that are filtered out of the visualization.

You can use table calculations for a variety of purposes, including:

- Transforming values to rankings
- Transforming values to show running totals
- Transforming values to show percent of total

For any Tableau visualization, there is a virtual table that is determined by the dimensions in the view. This table is not the same as the tables in your data source. Specifically, the virtual table is determined by the dimensions within the “level of detail,” which means the dimensions on any of the following shelves or cards in a Tableau worksheet:



The basics: addressing and partitioning

When you add a table calculation, you must use all dimensions in the level of detail either for partitioning (scoping) or for addressing (direction).

The dimensions that define how to group the calculation (the scope of data it is performed on) are called **partitioning fields**. The table calculation is performed separately within each partition.

The remaining dimensions, upon which the table calculation is performed, are called **addressing fields**, and determine the direction of the calculation.

Partitioning fields break the view up into multiple sub-views (or sub-tables), and then the table calculation is applied to the marks within each such partition. The direction in which the calculation moves (for example, in calculating a running sum, or computing the difference between values) is determined by the addressing fields. So when you order the fields in the Specific Dimensions section of the Table Calculation dialog box from top to bottom, you are specifying the direction in which the calculation moves through the various marks in the partition.

When you add a table calculation using the Compute Using options, Tableau identifies some dimensions as addressing and others as partitioning automatically, as a result of your selections. But when you use Specific Dimensions, then it’s up to you to determine which dimensions are for addressing and which for partitioning.

Table (across)

Computes across the length of the table and restarts after every partition.

For example, in the following table, the calculation is computed across columns (YEAR(Order Date)) for every row (MONTH(Order Date)).

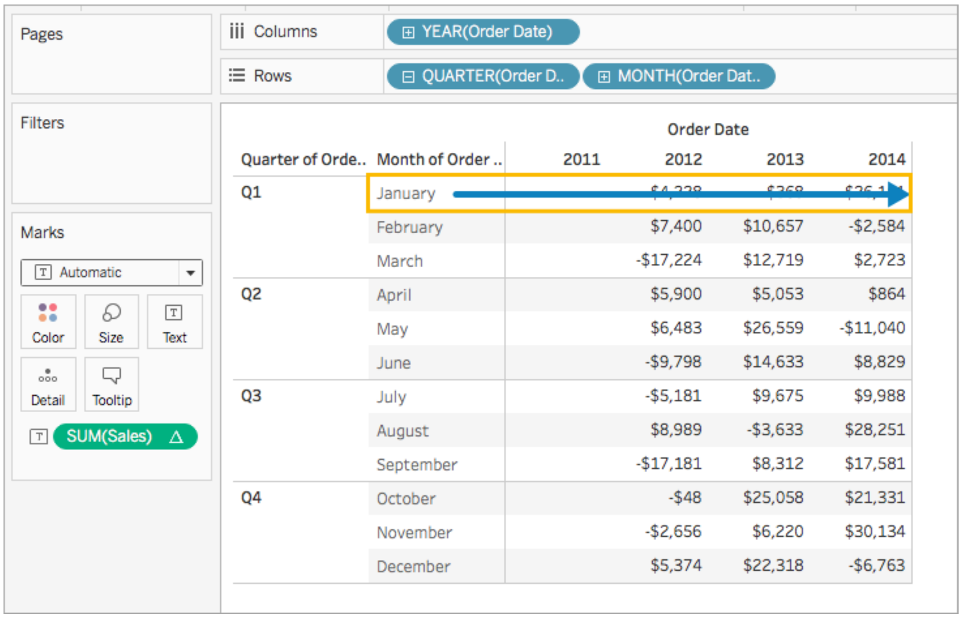


Table (down)

Computes down the length of the table and restarts after every partition.

For example, in the following table, the calculation is computed down rows (MONTH(Order Date)) for every column (YEAR(Order Date)).

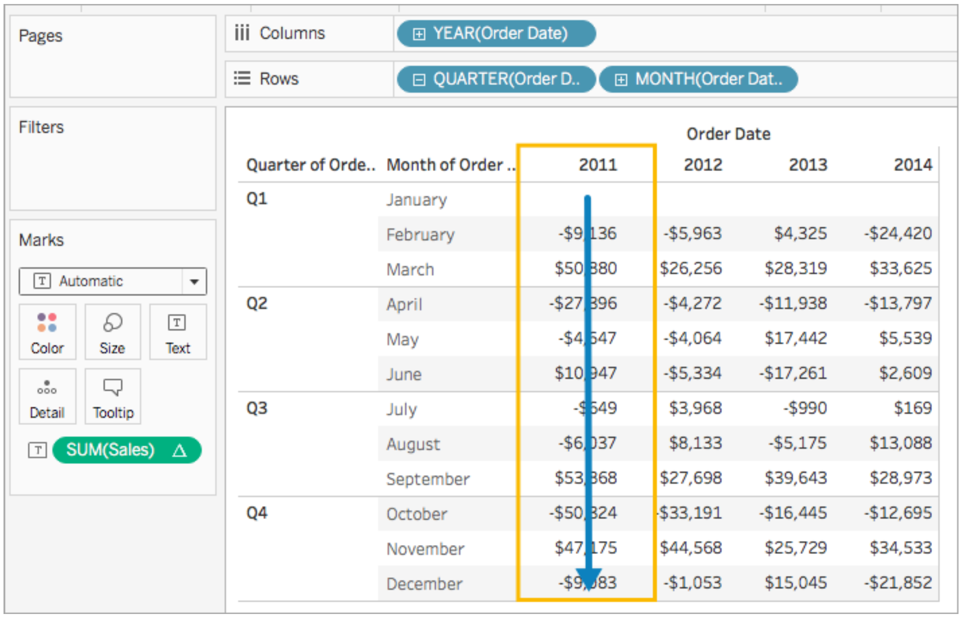


Table (across then down)

Computes across the length of the table, and then down the length of the table.

For example, in the following table, the calculation is computed across columns (YEAR(Order Date)), down a row (MONTH(Order Date)), and then across columns again for the entire table.

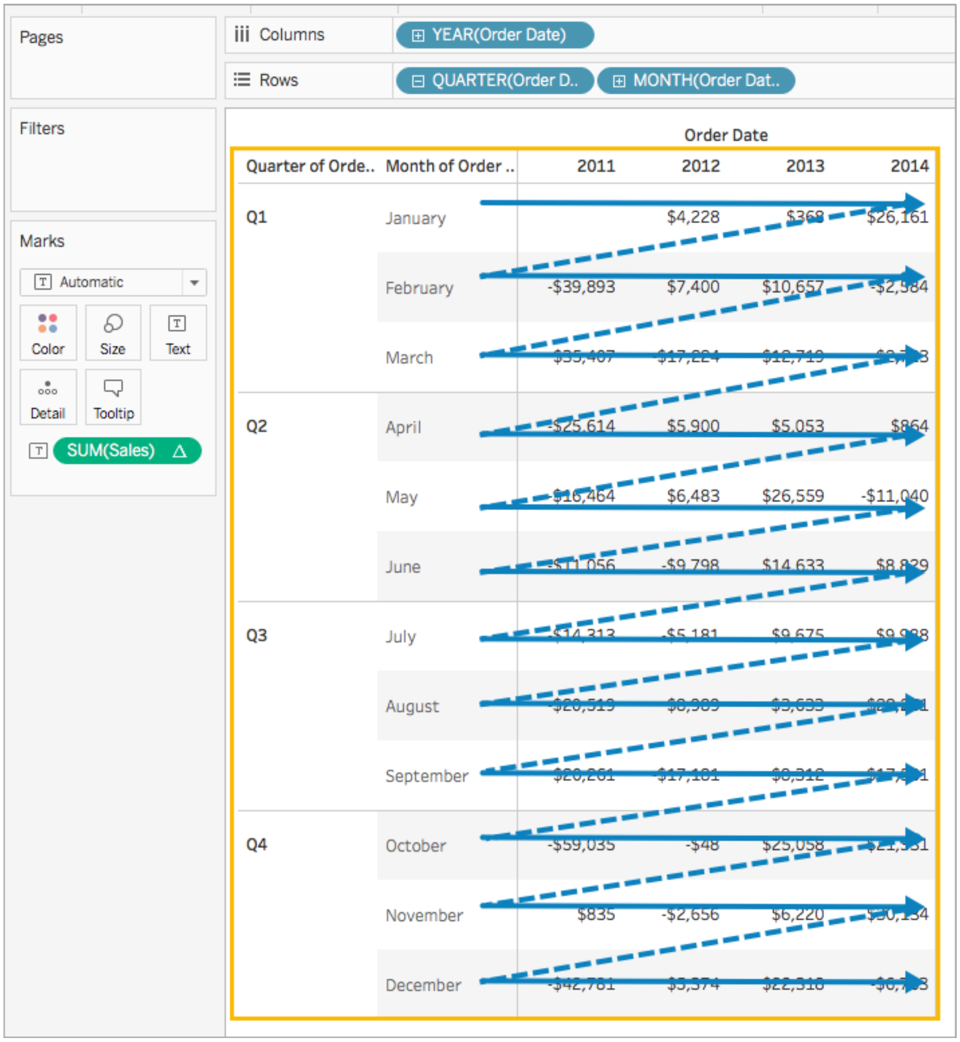
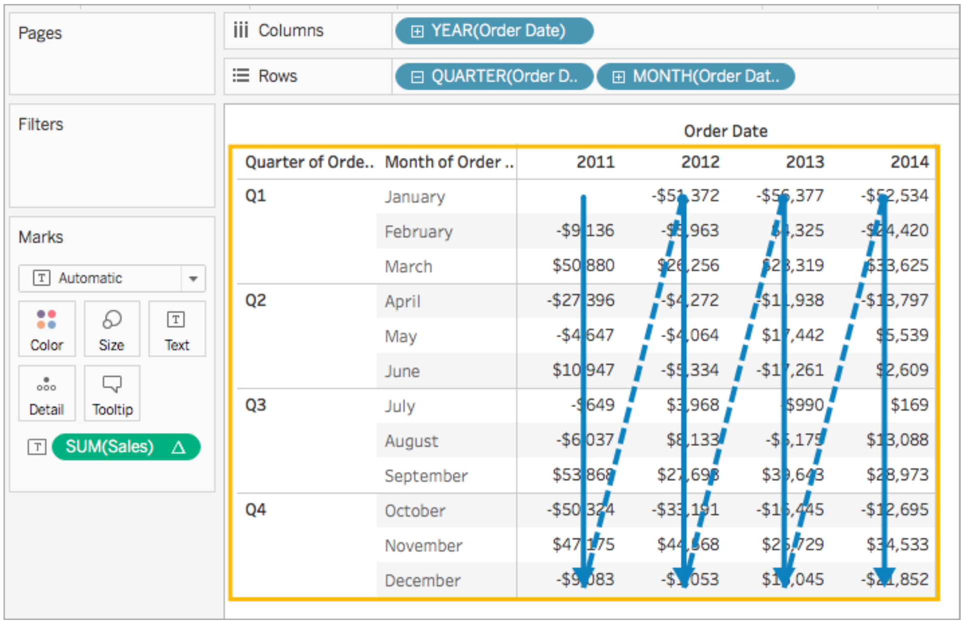


Table (down then across)

Computes down the length of the table, and then across the length of the table.

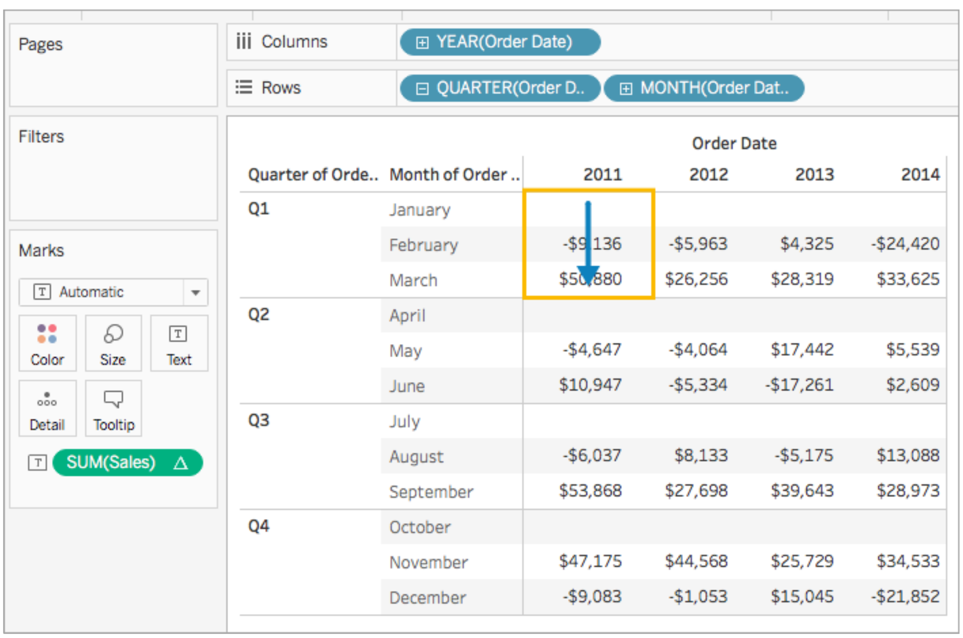
For example, in the following table, the calculation is computed down rows (MONTH(Order Date)), across a column (YEAR(Order Date), and then down rows again.



Pane (down)

Computes down an entire pane.

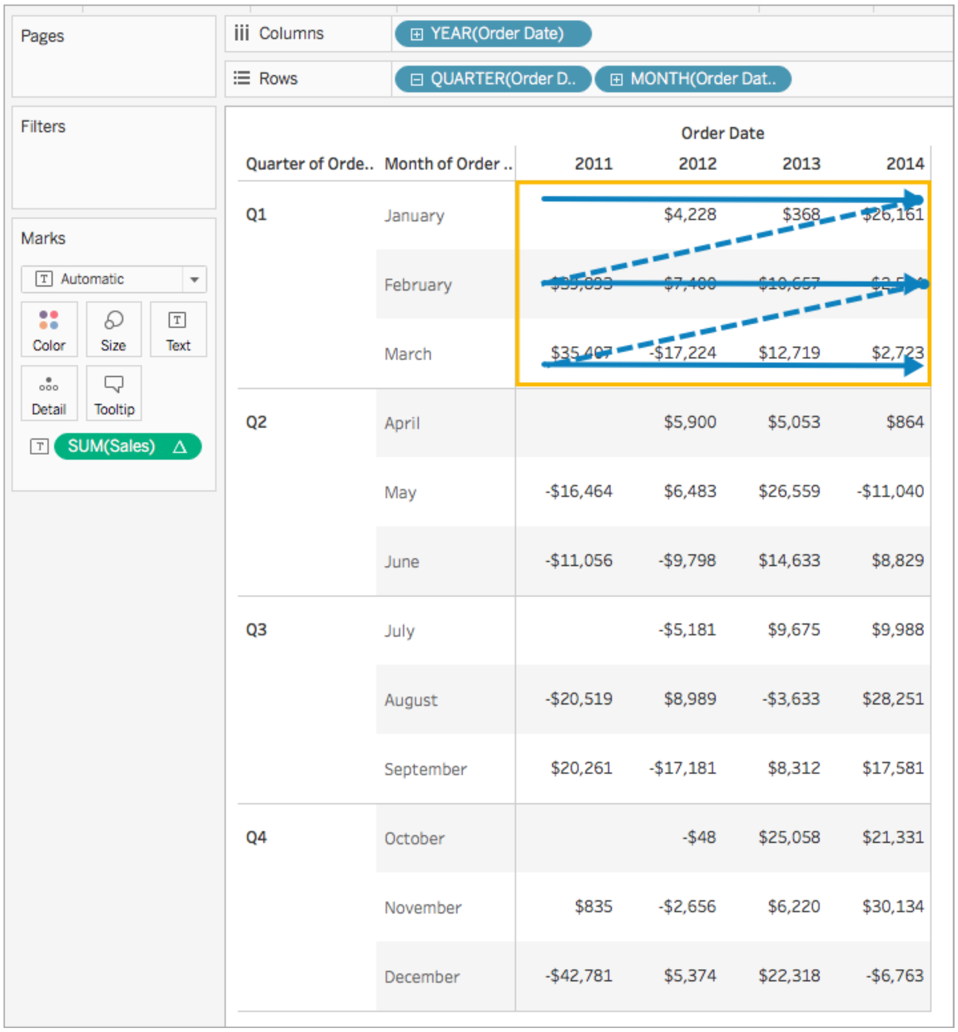
For example, in the following table, the calculation is computed down rows (MONTH(Order Date)) for a single pane.



Pane (across then down)

Computes across an entire pane and then down the pane.

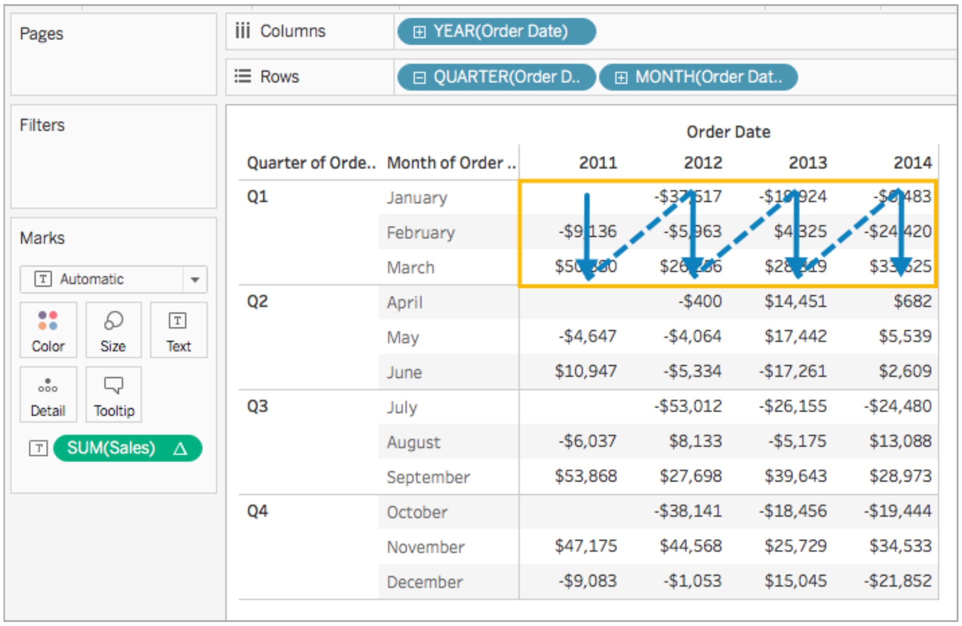
For example, in the following table, the calculation is computed across columns (YEAR(Order Date)) for the length of the pane, down a row (MONTH(Order Date)), and then across columns for the length of the pane again.



Pane (down then across)

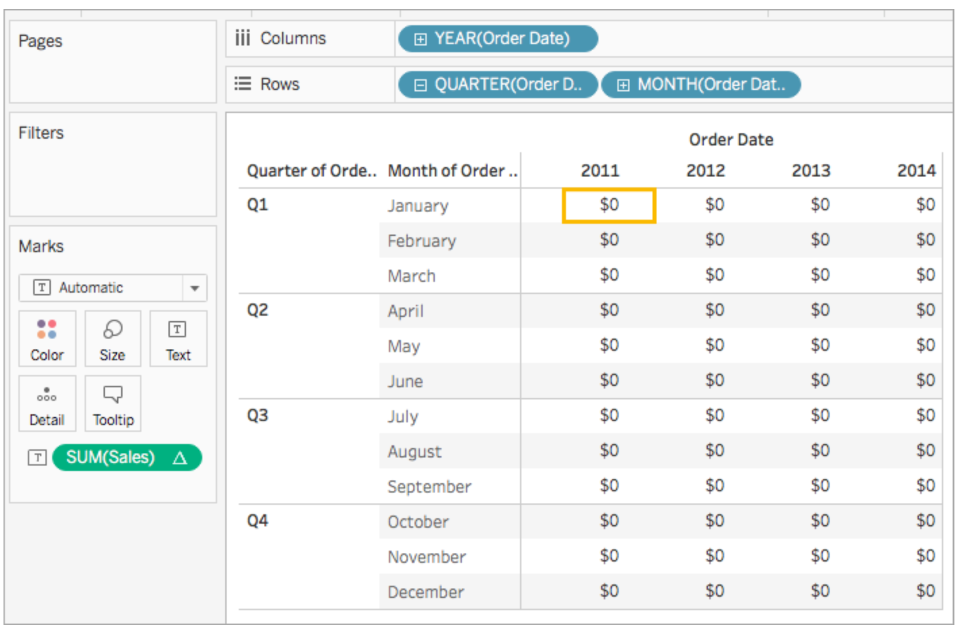
Computes down an entire pane and then across the pane.

For example, in the following table, the calculation is computed down rows (MONTH(Order Date)) for the length of the pane, across a column (YEAR(Order Date)), and then down the length of the pane agian.



Cell

Computes within a single cell.

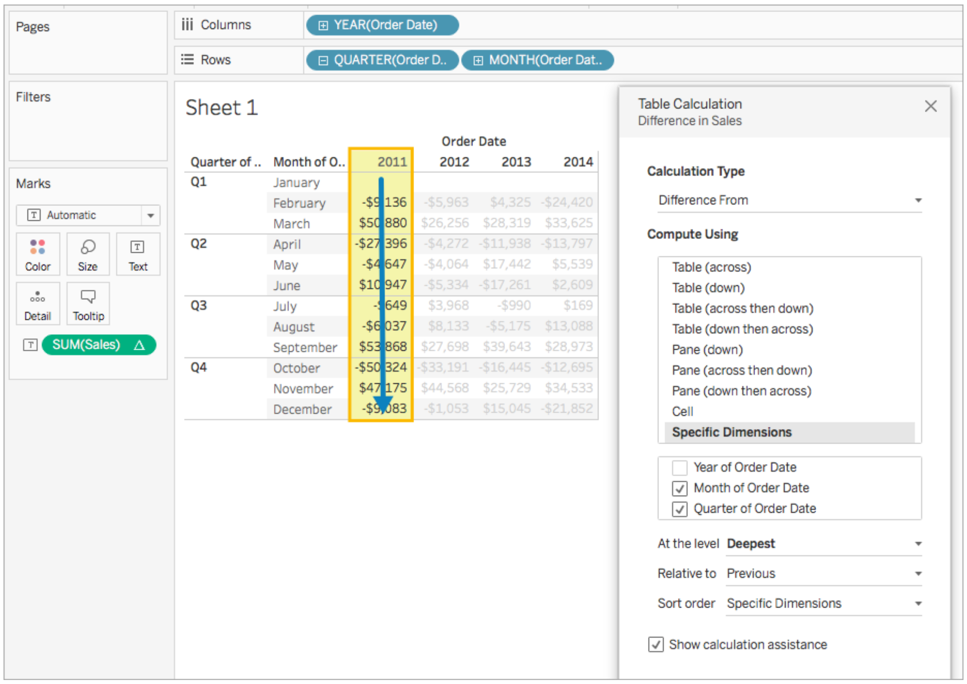


Specific Dimensions

Computes only within the dimensions you specify.

For example, in the following visualization the dimensions, Month of Order Date and Quarter of Order Date, are the addressing fields (since they are selected), and Year of Order Date is the partitioning field (since it is not selected). So the calculation transforms the difference from each month across all quarters within a year. The calculation starts over for every year.

Note that if all dimensions are selected, then the entire table is in scope.



At the level

The **At the level** option is only available when you select **Specific Dimensions** in the Table Calculations dialog box, and when more than one dimension is selected in the field immediately below the **Compute Using** options —that is, when more than one dimension is defined as an addressing field.

This option is not available when you're defining a table calculation with **Compute Using**, because those values establish partitions by position. But with **Specific Dimensions**, because the visual structure and the table calculation are not necessarily aligned, the **At the level** option is available to let you fine-tune your calculation.

Use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension. How is this different from just using that dimension for partitioning? In fact, it is partitioning, but it's partitioning by position rather than by value, which is how partitioning is defined with the **Compute Using** options.

The choices available from the At the level drop-down list in the example above are:

Deepest	Specifies that the calculation should be performed at the level of finest granularity. This is the default option.
Quarter of Order Date	Specifies that the calculation should be performed at the quarter level.
Month of Order Date	Specifies that the calculation should be performed at the month level.

Create a table calculation

To learn how to create a table calculation, follow along with the steps in the example below. To learn how to create quick table calculations, see **Quick Table Calculations** on page 2440.

Step 1: Build the visualization

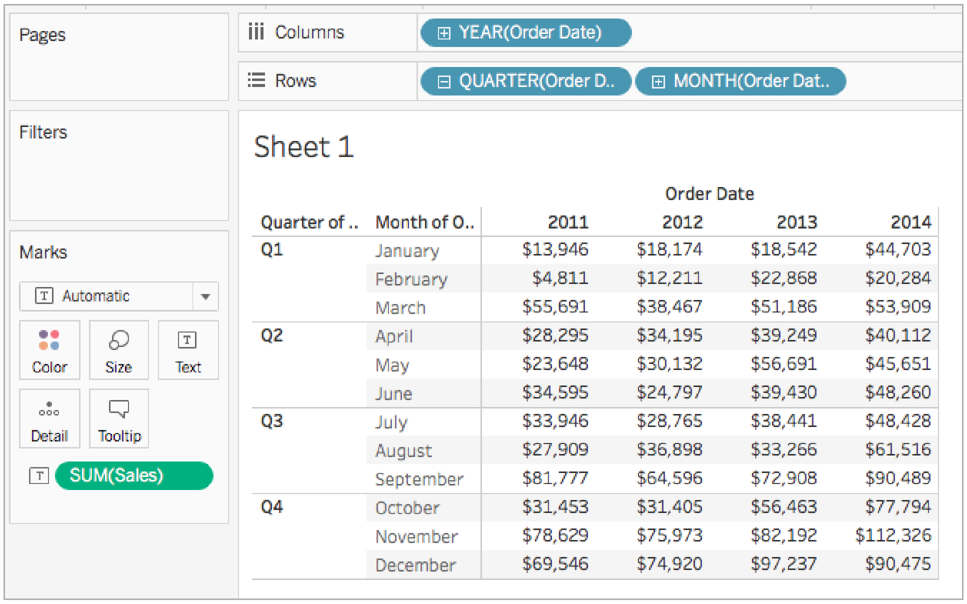
1. Open Tableau and connect to the **Sample-Superstore** saved data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, under Dimensions, drag **Order Date** to the **Rows** shelf.

   The dimension updates to YEAR(Order Date).
4. On the Rows shelf, right-click **YEAR(Order Date)** and select **Quarter**.
5. On the Rows shelf, click the **+** icon on **QUARTER(Order Date)**.

   MONTH(Order Date) is added to the shelf.
6. From the **Data** pane, under Dimensions, drag **Order Date** to the **Columns** shelf.

   The dimension updates to YEAR(Order Date) again.
7. From the **Data** pane, under Measures, drag **Sales** to **Text** on the Marks card.

The updates to look like this:



Step 2: Add the table calculation

1. On the Marks card, right-click SUM(Sales) and select **Add Table Calculation**.
2. In the Table Calculation dialog box that opens, do the following:
   - For **Calculation Type**: select **Difference From**.

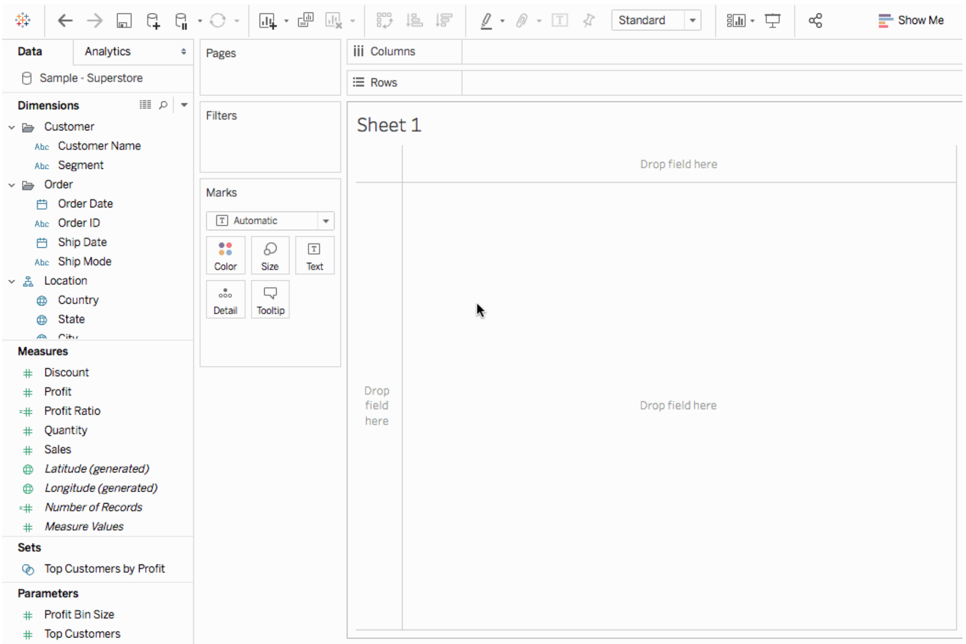
     For more information about the types of table calculations you can use in Tableau, and how you can configure them, see **Table Calculation Types** on page 2419.
   - For **Compute Using**, select **Table (across)**.

     For more information about these options, see **The basics: addressing and partitioning** on page 2408 section.

     Note that as you select how to compute the calculation, the visualization updates with visual indicators to guide you.
   - When finished, click the X in the top corner of the Table Calculation dialog box to exit it.

     The calculation is applied to the values in the visualization.

Check your work!



Edit a table calculation

To edit a table calculation:

1. Right-click the measure in the view with the table calculation applied to it and select **Edit Table Calculation**.
2. In the Table Calculation dialog box that appears, make your changes.
3. When finished, click the X in the top corner of the Table Calculation dialog box to exit it.

Remove a table calculation

To remove a table calculation:

- Right-click the measure in the view with the table calculation applied to it and select **Clear Table Calculation**.

  The table calculation is removed from the measure and the visualization updates with the original values.

See Also

Table calculations on page 2114

Quick Table Calculations on page 2440

Table Calculation Types below

Table Calculation Types

This article describes the types of table calculations available in Tableau and when to use them. It uses simple examples to demonstrate how each calculation transforms data in a table. For more information on how to create and configure table calculations, see **Create a table calculation** on page 2416.

Difference From calculation

A **Difference From** table calculation computes the difference between the current value and another value in the table for each mark in the visualization.

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different.

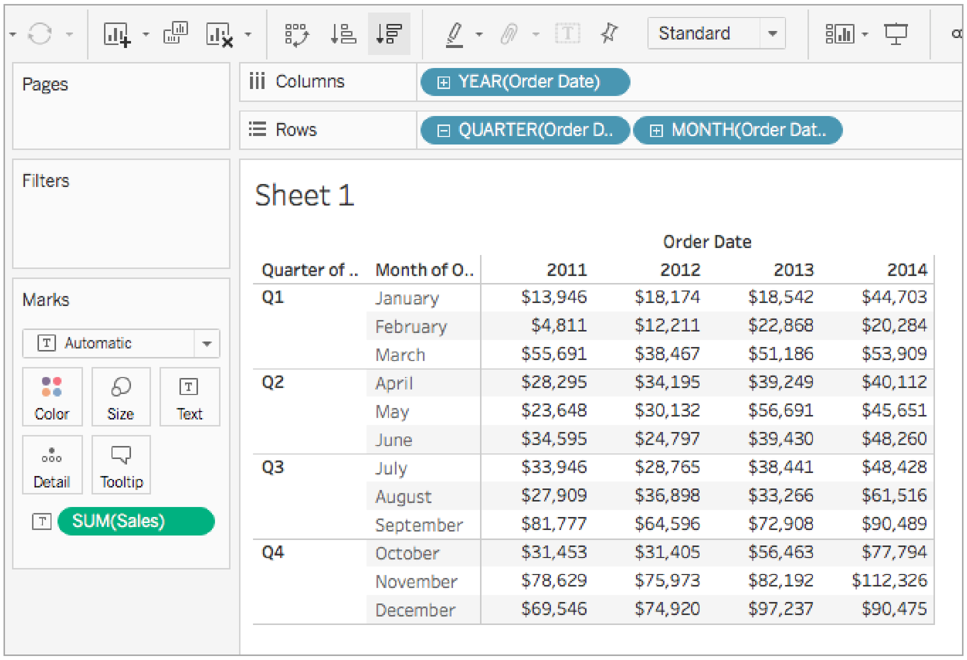
To specify from which value the difference should be calculated:

1. Right-click a measure in the view and select **Add Table Calculation**.
2. In the Table Calculation dialog box, for **Relative to**, select one of the following options:

Previous	Calculates the difference between the current value and the previous value in the partition. This is the default value.
Next	Calculates the difference between the current value and the next value in the partition.
First	Calculates the difference between the current value and the first value in the partition.
Last	Calculates the difference between the current value and the last value in the partition.

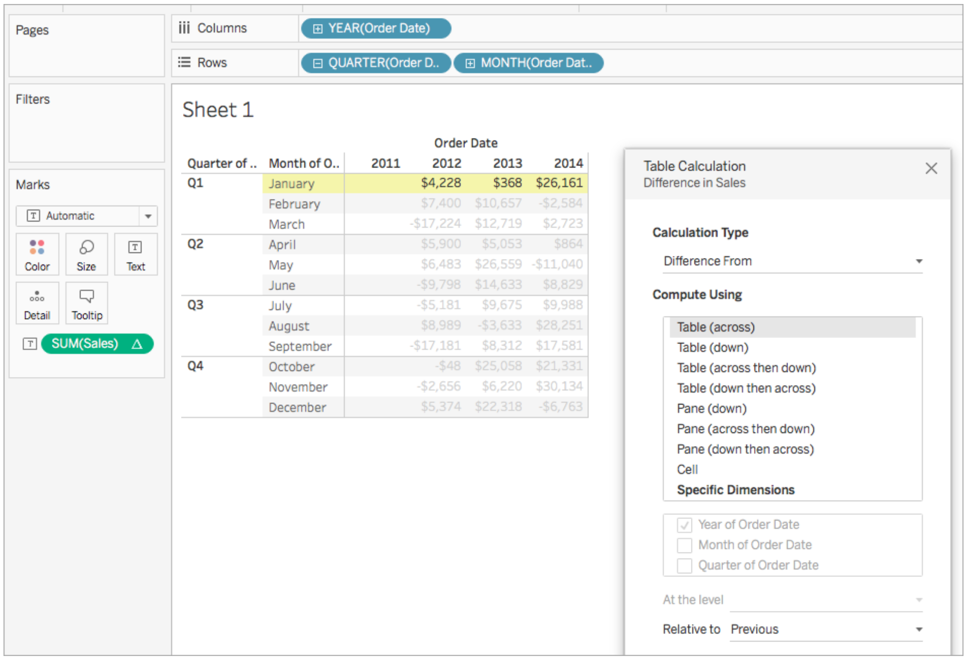
Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



		Order Date			
Quarter of ..	Month of O..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$4,811	$12,211	$22,868	$20,284
	March	$55,691	$38,467	$51,186	$53,909
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$23,648	$30,132	$56,691	$45,651
	June	$34,595	$24,797	$39,430	$48,260
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$27,909	$36,898	$33,266	$61,516
	September	$81,777	$64,596	$72,908	$90,489
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$78,629	$75,973	$82,192	$112,326
	December	$69,546	$74,920	$97,237	$90,475

You can use a Difference From table calculation to calculate how sales fluctuate (how much they go up or down) between the years for each month.



You can see that in January, there was a 368 USD difference between sales in 2012 and 2013, and a 26,161 USD difference between sales in 2013 and 2014.

Tip: When calculating year-over-year growth, the first year doesn't have a previous year to compare to, so the column is left blank. Hide the column that you don’t want to show to keep the calculation intact. For details, see **Hide rows and columns** on page 225.

Why? If you filtered out the first year to remove it from the view, it would also remove it from the calculation so the second year doesn't have a previous year to compare to and is left blank. Instead of filtering, hiding the column keeps the calculation intact.

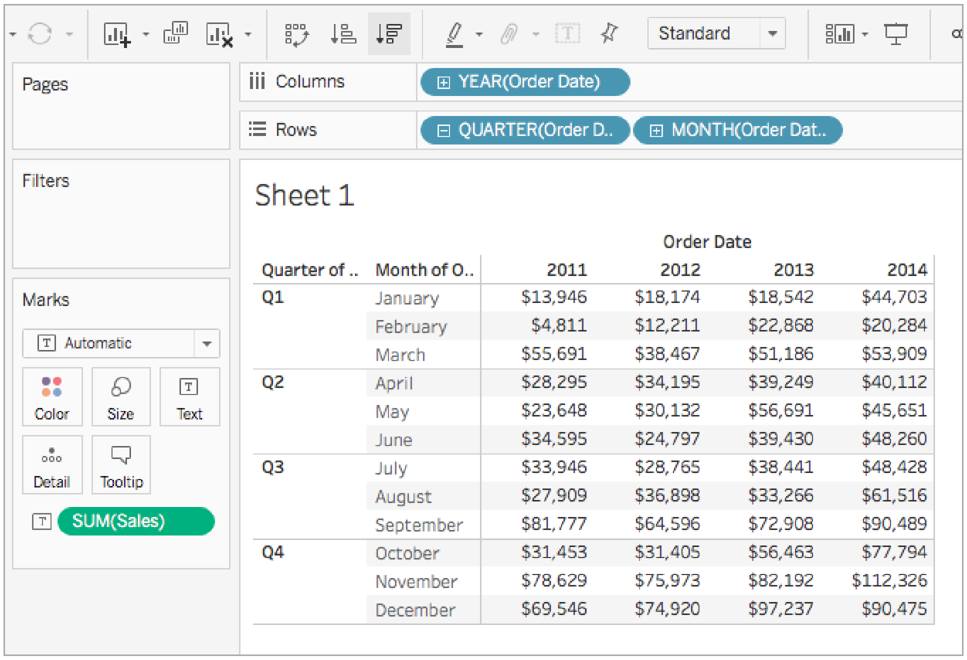
Moving calculation

For each mark in the view, a **Moving Calculation** table calculation (sometimes referred to as a *rolling* calculation) determines the value for a mark in the view by performing an aggregation (sum, average, minimum, or maximum) across a specified number of values before and/or after the current value.

A moving calculation is typically used to smooth short-term fluctuations in your data so that you can see long-term trends. For example, with securities data there are so many fluctuations every day that it is hard to see the big picture through all the ups and downs. You can use a moving calculation to define a range of values to summarize using an aggregation of your choice.

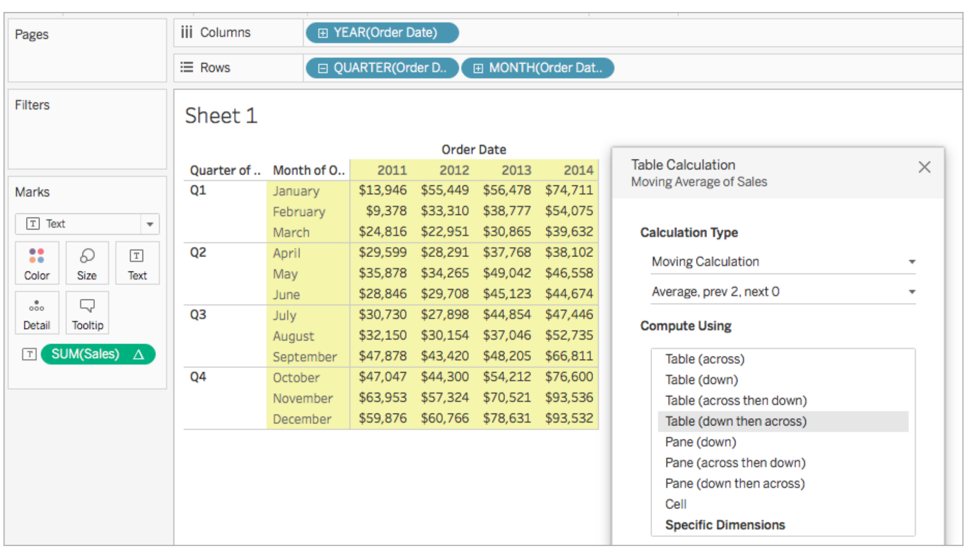
Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



		Order Date			
Quarter of ..	Month of O..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$4,811	$12,211	$22,868	$20,284
	March	$55,691	$38,467	$51,186	$53,909
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$23,648	$30,132	$56,691	$45,651
	June	$34,595	$24,797	$39,430	$48,260
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$27,909	$36,898	$33,266	$61,516
	September	$81,777	$64,596	$72,908	$90,489
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$78,629	$75,973	$82,192	$112,326
	December	$69,546	$74,920	$97,237	$90,475

You can use a Moving calculation to find out how sales totals are trending over time. To do this, you can transform each monthly total so that it averages the monthly total for it and the two previous months over time.



		Order Date			
Quarter of ..	Month of O..	2011	2012	2013	2014
Q1	January	$13,946	$55,449	$56,478	$74,711
	February	$9,378	$33,310	$38,777	$54,075
	March	$24,816	$22,951	$30,865	$39,632
Q2	April	$29,599	$28,291	$37,768	$38,102
	May	$35,878	$34,265	$49,042	$46,558
	June	$28,846	$29,708	$45,123	$44,674
Q3	July	$30,730	$27,898	$44,854	$47,446
	August	$32,150	$30,154	$37,046	$52,735
	September	$47,878	$43,420	$48,205	$66,811
Q4	October	$47,047	$44,300	$54,212	$76,600
	November	$63,953	$57,324	$70,521	$93,536
	December	$59,876	$60,766	$78,631	$93,532

You can see the average sales over time. For example, the value listed for December 2011 is the average sales for October, November, and December, 2011. The value listed for January, 2012 is the average sales for November and December, 2011, and January, 2012.

Add Secondary Calculation

With **Running Total** and **Moving Calculation** table calculations, you have the option to transform values twice to obtain the result you want—that is, to add a secondary table calculation on top of the primary table calculation. For example, you could add an initial table calculation to calculate the running total for sales per month within each individual year, and then a secondary calculation to calculate the year-over-year percent difference for each month from one year to the next.

For an example showing how to create a secondary calculation, see **Running Total calculation** on page 2433.

Percent Difference From calculation

A **Percent Difference From** table calculation computes the difference between the current value and another value in the table as a percentage for each mark in the visualization.

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different.

To specify from which value the difference should be calculated:

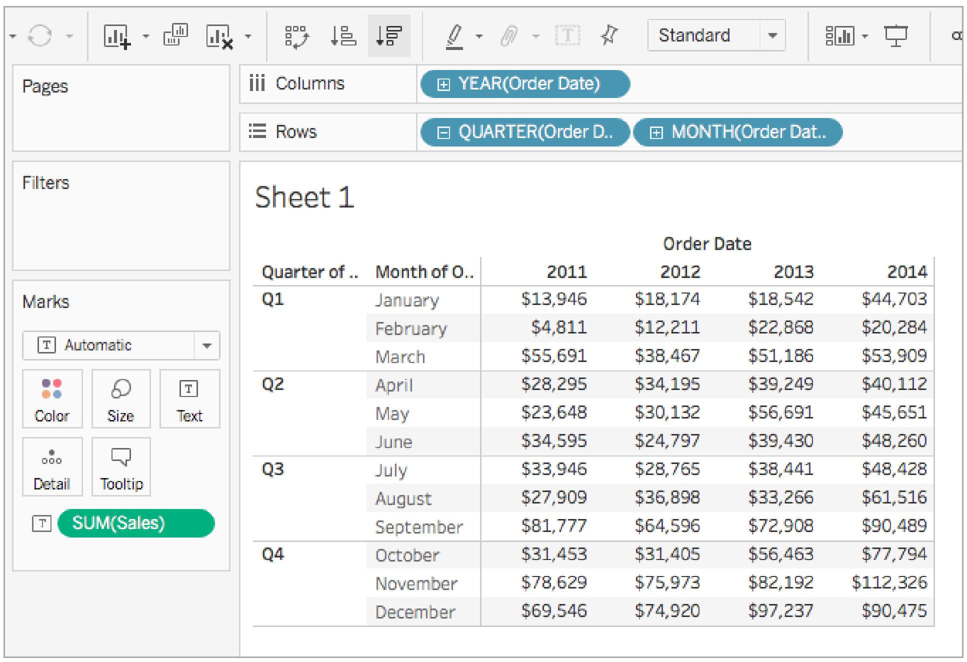
1. Right-click a measure in the view and select **Add Table Calculation**.
2. In the Table Calculation dialog box, for **Relative to**, select one of the following options:

Previous	Calculates the difference between the current value and the previous value in the partition. This is the default value.
Next	Calculates the difference between the current value and the next value in the partition.
First	Calculates the difference between the current value and the first value in the partition.

Last	Calculates the difference between the current value and the last value in the partition.

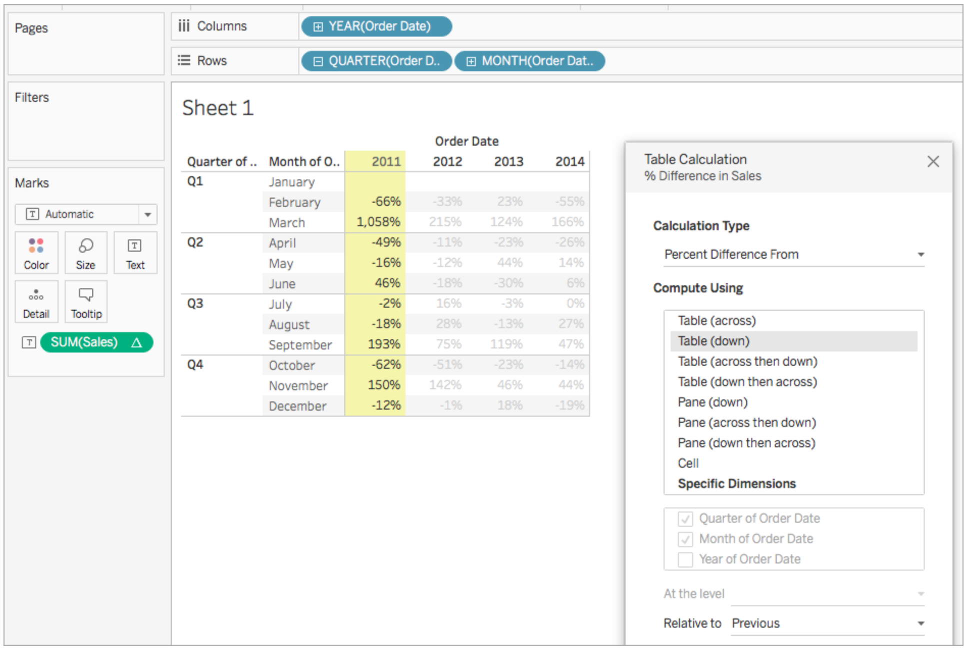
Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



		Order Date			
Quarter of ..	Month of O..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$4,811	$12,211	$22,868	$20,284
	March	$55,691	$38,467	$51,186	$53,909
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$23,648	$30,132	$56,691	$45,651
	June	$34,595	$24,797	$39,430	$48,260
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$27,909	$36,898	$33,266	$61,516
	September	$81,777	$64,596	$72,908	$90,489
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$78,629	$75,973	$82,192	$112,326
	December	$69,546	$74,920	$97,237	$90,475

You can use a Percent Difference From table calculation to calculate how sales fluctuate (how much they go up or down) between the years for each month. Values are calculated as percentages.



You can see that between January and February, 2011, there was a -66% difference in sales, but between February and March, 2011, there was a huge improvement of 1,058% sales.

Percent From calculation

A **Percent From** table calculation computes a value as a percentage of some other value—typically, as a percentage of the previous value in the table—for each mark in the visualization.

With a **Difference From**, **Percent Difference From**, or **Percent From** calculation, there are always two values to consider: the current value, and the value from which the difference should be calculated. In most cases, you want to calculate the difference between the current value and the previous value, as in the procedure above. But in some cases you may want something different.

To specify from which value the difference should be calculated:

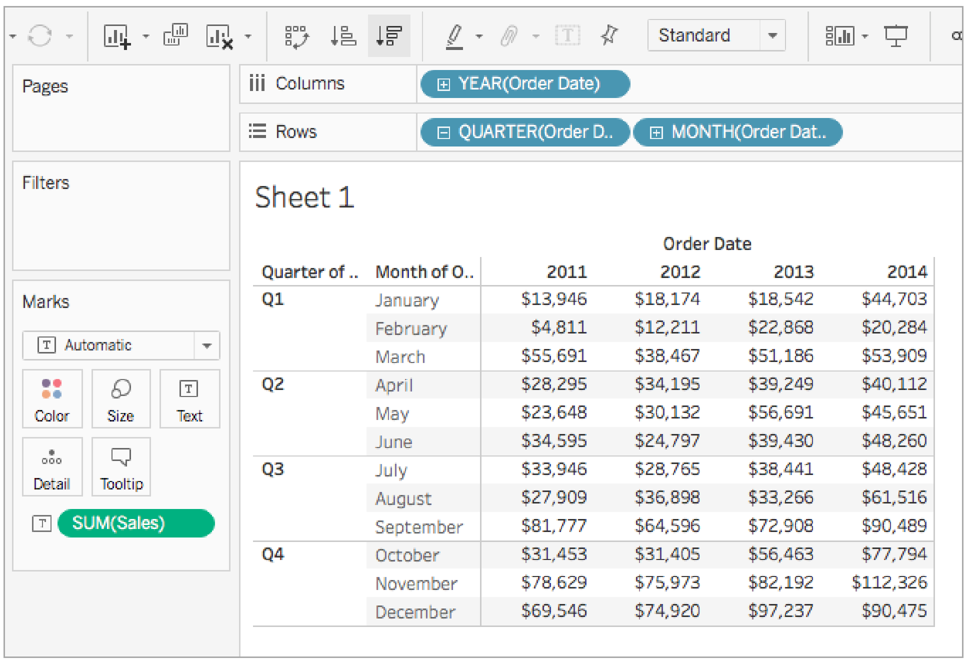
1. Right-click a measure in the view and select **Add Table Calculation**.
2. In the Table Calculation dialog box, for **Relative to**, select one of the following options:

Previous	Calculates the difference between the current value and the previous value in the partition. This is the default value.
Next	Calculates the difference between the current value and the next value in the partition.

First	Calculates the difference between the current value and the first value in the partition.
Last	Calculates the difference between the current value and the last value in the partition.

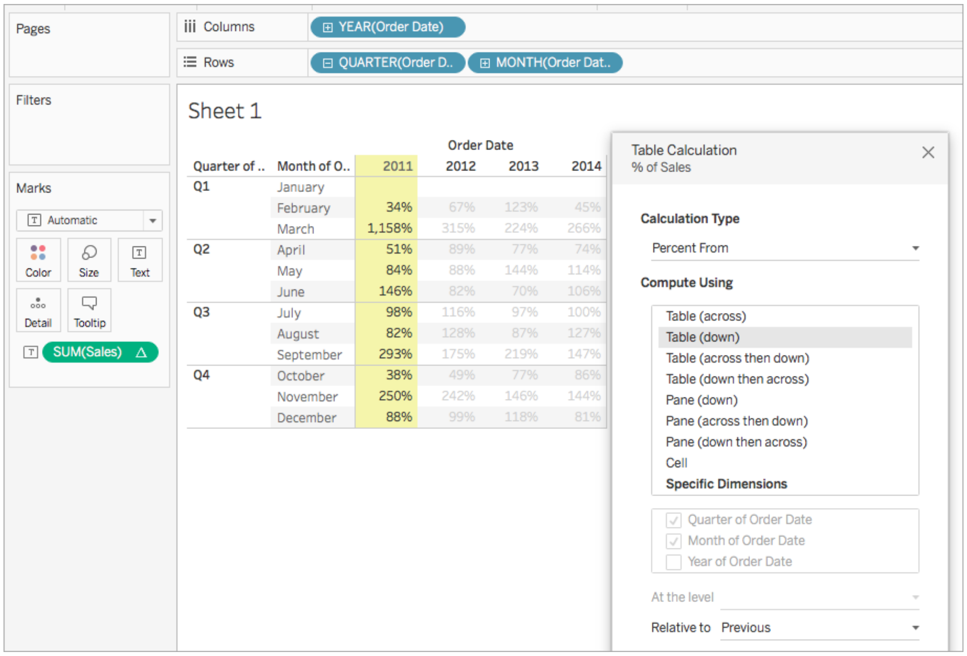
Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



		Order Date			
Quarter of ..	Month of O..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$4,811	$12,211	$22,868	$20,284
	March	$55,691	$38,467	$51,186	$53,909
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$23,648	$30,132	$56,691	$45,651
	June	$34,595	$24,797	$39,430	$48,260
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$27,909	$36,898	$33,266	$61,516
	September	$81,777	$64,596	$72,908	$90,489
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$78,629	$75,973	$82,192	$112,326
	December	$69,546	$74,920	$97,237	$90,475

You can use a Percent From table calculation to calculate the percentage of a previous value. For example, you can calculate what percentage of sales in January 2011, was made in February 2011.



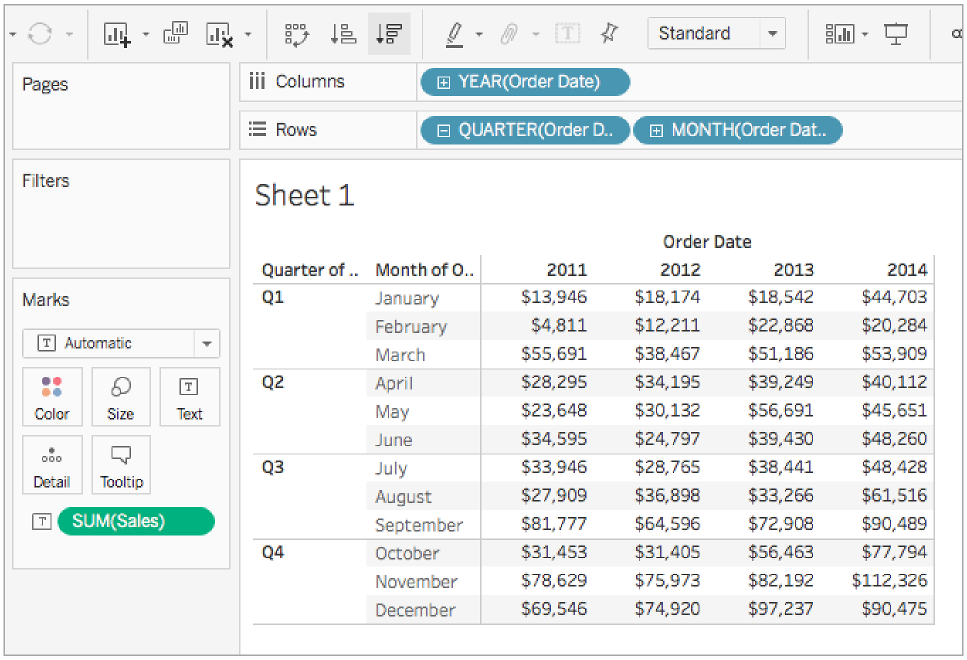
You can see that February, 2011 made 34% of the sales made in January, 2011; March, 2011 made 1,158% of the sales made in February, and so on.

Percent of Total calculation

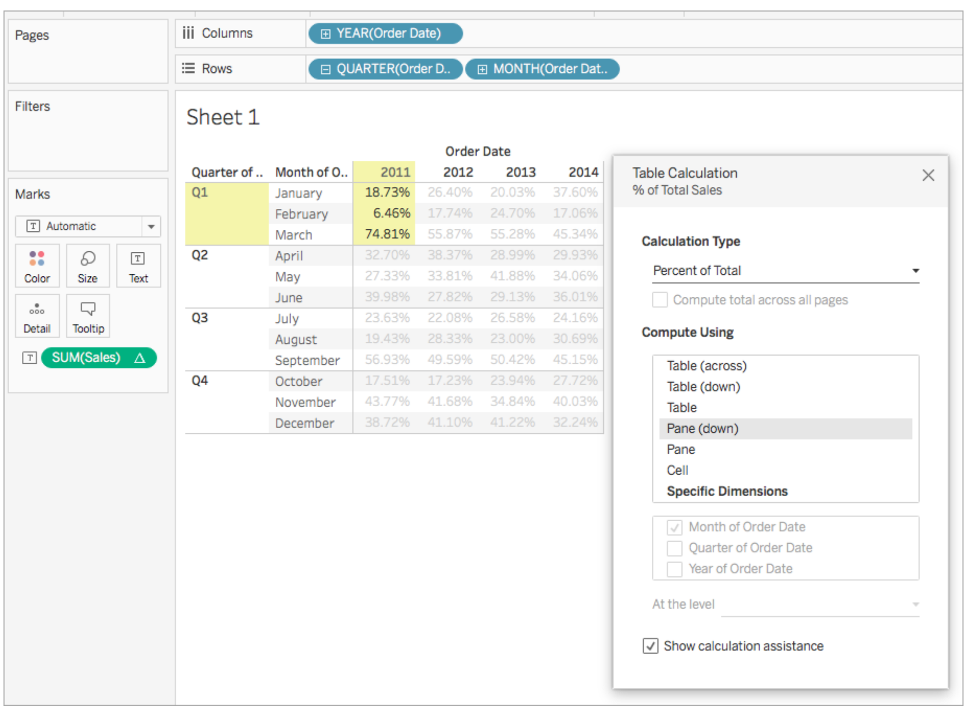
For each mark in the view, a **Percent of Total** table calculation computes a value as a percentage of all values in the current partition.

Example

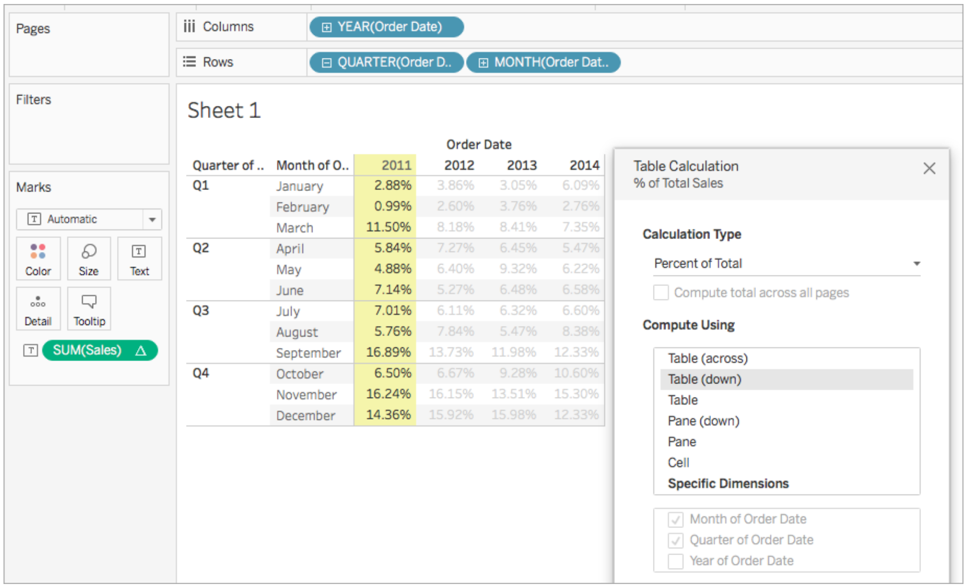
Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



You can use a Percent of Total table calculation to calculate the percentage of total sales each month makes within a quarter. For example, you can see that January, 2011 makes up 18.73% of sales made in Q1.



Or you can calculate the percentage of total sales each month makes within a year. For example, you can see that January, 2011 makes up 2.88% of sales made in 2011.

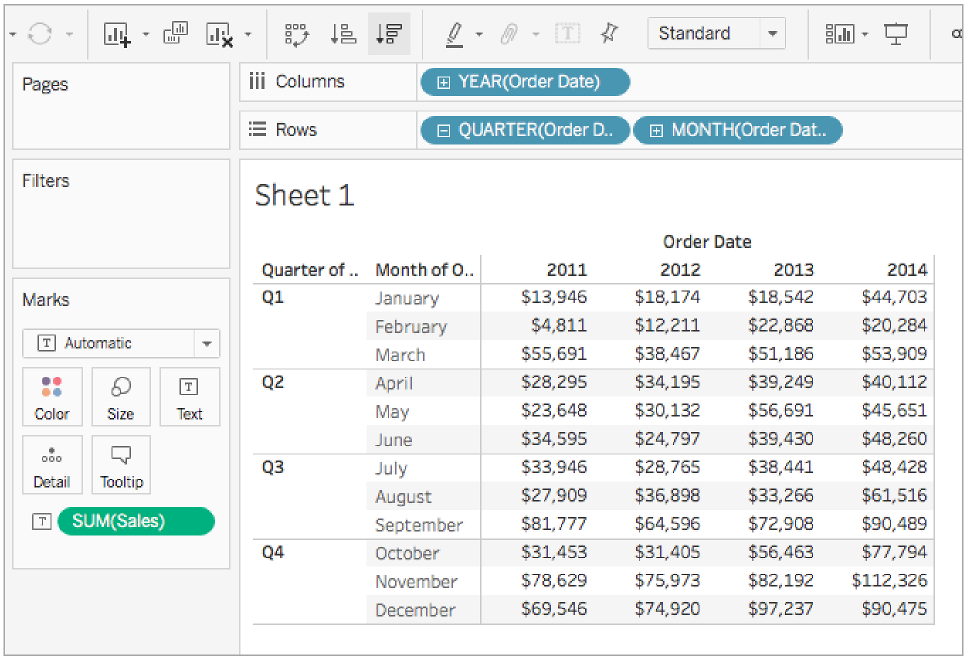


Percentile calculation

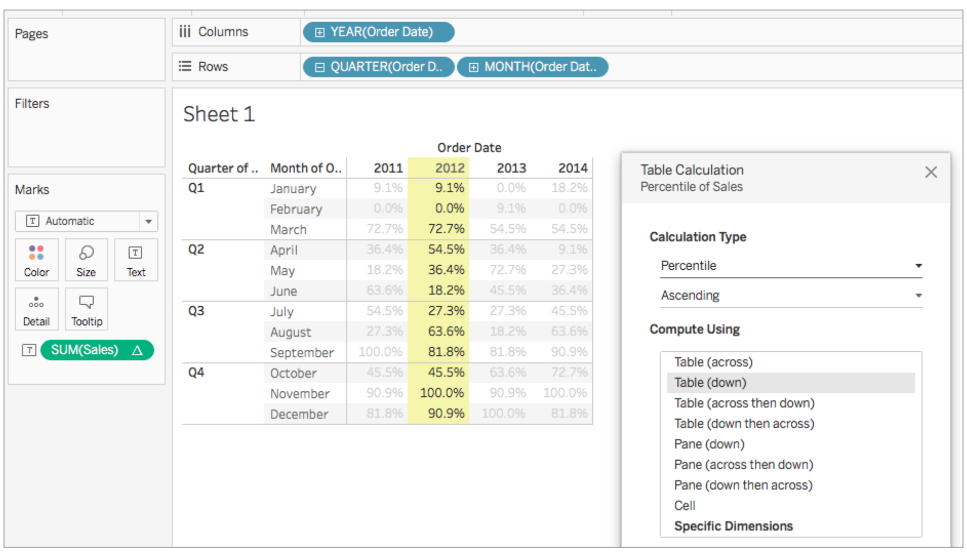
For each mark in the view, a **Percentile** table calculation computes a percentile rank for each value in a partition.

Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



You can use a Percentile table calculation to rank the total sales for each month in a year as a percentage, rather than a whole number (for example, 1 through 10).



Since February made a very small amount of sales in 2012 compared to the overall total, it is ranked as 0.0% (or number 1 out of 12, since this example is Ascending, and therefore ranked from least to most). Sales in January, 2012 were a bit higher and were therefore ranked as 9.1% (or number 2 out of 12 months). Since November made the most sales in 2012, it is ranked as 100% (or number 12 out of 12).

Descending vs. Ascending

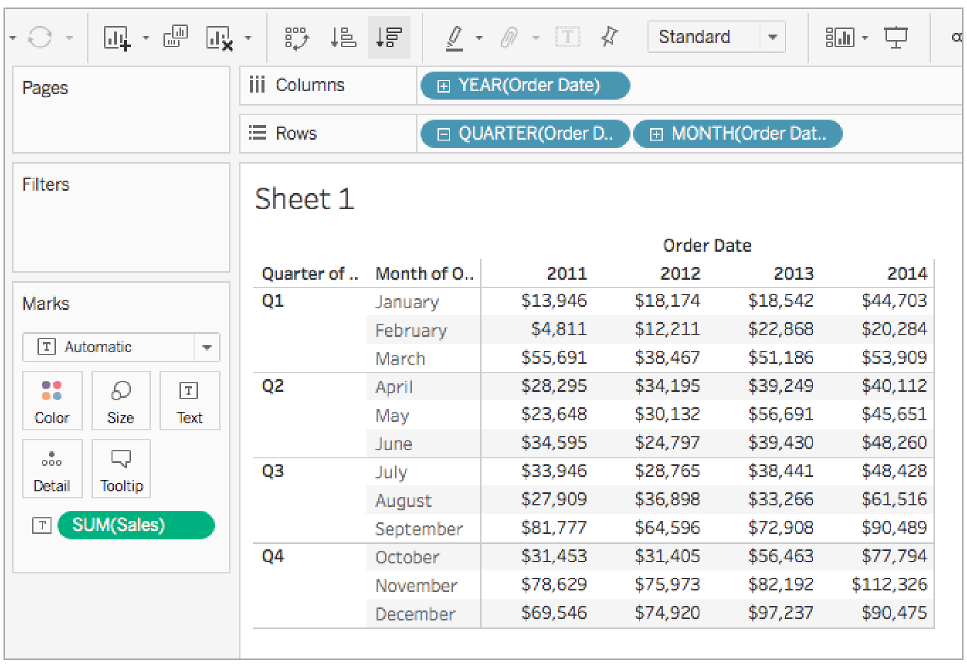
Ascending order ranks values from least to most. **Descending** order ranks values from most to least.

Rank calculation

For each mark in the view, a **Rank** table calculation computes a ranking for each value in a partition.

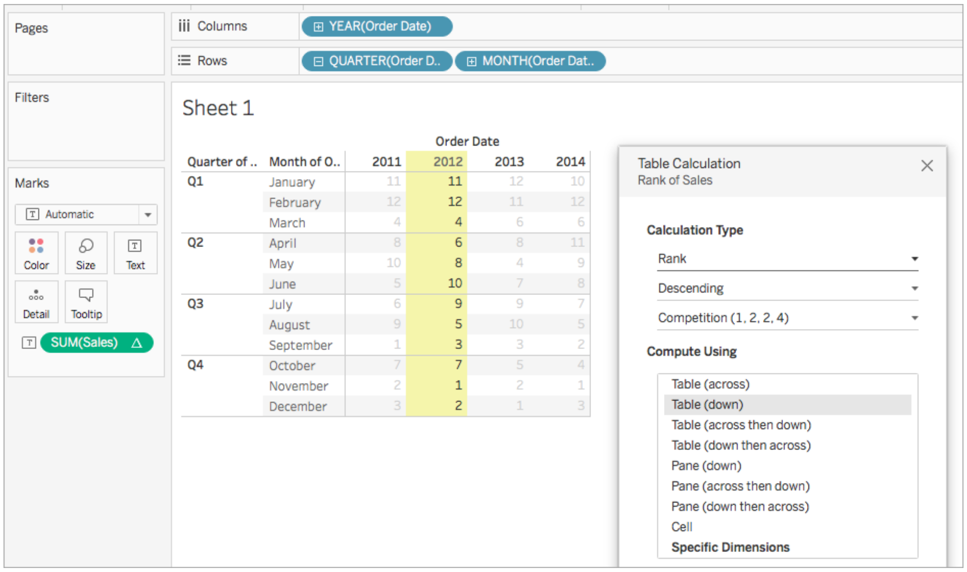
Example

Consider the text table below. It shows the total sales per month for 2011, 2012, 2013, and 2014 for a large store chain.



		Order Date			
Quarter of ..	Month of O..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$4,811	$12,211	$22,868	$20,284
	March	$55,691	$38,467	$51,186	$53,909
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$23,648	$30,132	$56,691	$45,651
	June	$34,595	$24,797	$39,430	$48,260
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$27,909	$36,898	$33,266	$61,516
	September	$81,777	$64,596	$72,908	$90,489
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$78,629	$75,973	$82,192	$112,326
	December	$69,546	$74,920	$97,237	$90,475

You can use a Rank table calculation to calculate a ranking for each month in a year.



You can see that, since November made the most amount of sales in 2012, it is ranked as number 1 (because the rank is in descending order, meaning it is ordered from most to least). Since February made the least amount of sales in 2012, it is ranked number 12.

Descending vs. Ascending

Ascending order ranks values from least to most. **Descending** order ranks values from most to least. For Rank table calculation, the default value is **Descending**.

Rank Type

One issue with **Rank** calculations is that there may be more than one mark with the same value. What would happen, for example, if Tables in the Central region and Appliances in the South region both had sales of exactly $36,729? Tableau lets you specify how to handle such cases by including an additional field in the Table Calculation dialog box when you set **Calculation Type** to **Rank**.

The choices are listed below. The number sequence at the beginning of each option show how each option would rank a hypothetical set of four values where two of the values are identical:

Option	Result
Competition (1, 2, 2, 4)	Identical values are assigned an identical rank. The highest value is ranked 1 and then the next two,

	identical values, are both are ranked 2. The next value is then ranked 4.
Modified Competition (1, 3, 3, 4)	Identical values are assigned an identical rank. The highest value is ranked 1 and then the next two, identical values, are both are ranked 3. The next value is then ranked 4.
Dense (1, 2, 2, 3)	Duplicate values are all given the same rank, which is the next number in the ranking sequence. The next value after the duplicate values is computed as though the duplicate values were a single value.
Unique (1, 2, 3, 4)	Duplicate values are given unique rankings, according to the direction in which the ranking is being com-puted.

Running Total calculation

For each mark in the view, a **Running Total** table calculation aggregates values cumulatively in a partition. It can do this by summing values, averaging values, or replacing all values with either the lowest or highest actual value.

Suppose you are starting with the following text view, which shows sales totals broken out by year (from left to right) and by quarter and month (from top to bottom):

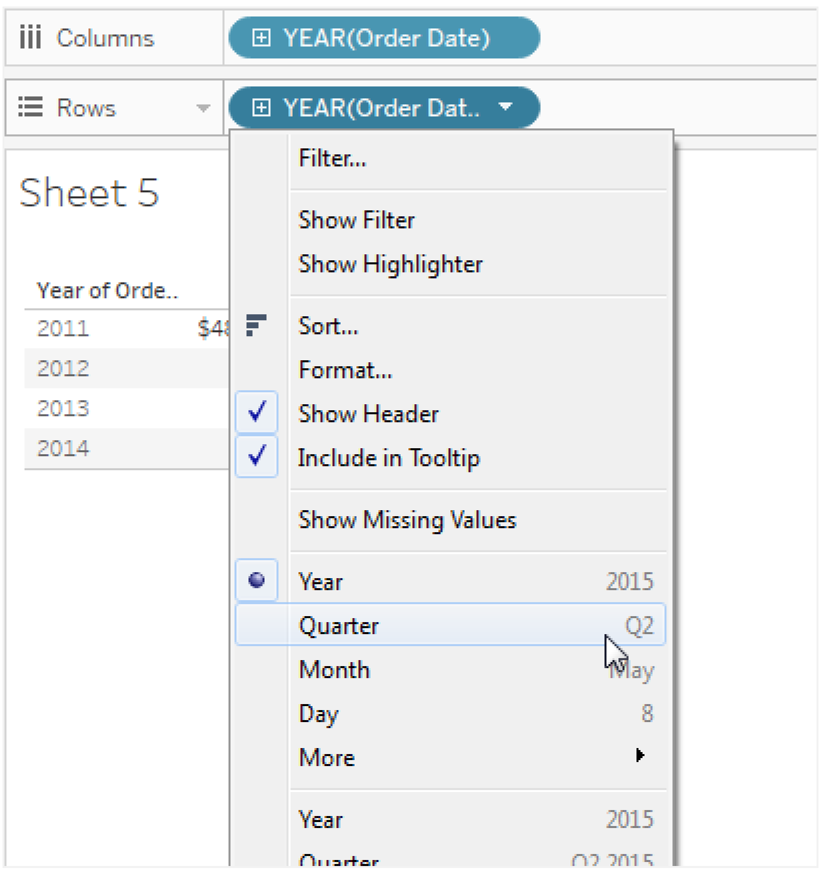
		Order Date			
Quarter of O..	Month of Or..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$4,811	$12,211	$22,868	$20,284
	March	$55,691	$38,467	$51,186	$53,909
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$23,648	$30,132	$56,691	$45,651
	June	$34,595	$24,797	$39,430	$48,260
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$27,909	$36,898	$33,266	$61,516
	September	$81,777	$64,596	$72,908	$90,489
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$78,629	$75,973	$82,192	$112,326
	December	$69,546	$74,920	$97,237	$90,475

Instead of absolute sales values, you want to see a running total of sales for each year, such that each month's sales are added to all previous months' sales.

Create the Basic View

1. Connect to the **Sample - Superstore** data source.
2. Click and drag the **Order Date** field in the Data pane and drag it to the Columns shelf.

   The default date level is YEAR(Order Date).
3. Click and drag **Order Date** again and drop it this time on the Rows shelf.
4. Click on the right side of the field to open the context menu. Then choose **Quarter**.

   You will see two options named **Quarter**. Be sure to choose the first one.

   The field should now read **QUARTER(Order Date)**.

Note: If you are creating the view on the web, the menu looks a bit different.

5. Click and drag **Order Date** a third time and drop it on the Rows shelf to the right of **QUARTER(Order Date)**.
6. Click on the right side of the field to open the context menu and this time choose **Month** (again, choose the first of two options named **Month**). The field should now read **MONTH(Order Date)**.
7. Drag **Sales** from the Data pane and drop it on Text on the Marks card.

You now have the basic view, showing Sales by Order Date over a four-year period, by month, quarter, and year.

Add a Running Total table calculation to the basic view

1. Click the **SUM(Sales)** field on the Marks card and choose **Add table calculation**.
2. In the Table Calculation dialog box, choose **Running Total** as the **Calculation Type**.
3. Choose **Table (Down)** from the **Compute Using** list.

   The highlighting in the view shows how this **Compute Using** value sets the scope of the calculation in the view:

		Order Date			
Quarter of O..	Month of Or..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$18,757	$30,385	$41,410	$64,987
	March	$74,448	$68,852	$92,596	$118,896
Q2	April	$102,743	$103,047	$131,845	$159,008
	May	$126,391	$133,179	$188,536	$204,659
	June	$160,987	$157,976	$227,967	$252,919
Q3	July	$194,933	$186,741	$266,407	$301,347
	August	$222,842	$223,640	$299,673	$362,863
	September	$304,620	$288,236	$372,581	$453,352
Q4	October	$336,073	$319,640	$429,044	$531,146
	November	$414,702	$395,613	$511,236	$643,472
	December	$484,247	$470,533	$608,474	$733,947

Comparing the values in the original text view with the values in this view shows that the result is correct. The monthly values ascend steadily and the December value (484,247) is the same value you see if you show column grand totals (from the Analysis menu, select **Totals** > **Show column grand totals**).

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

The Running Total doesn't have to be a sum

For a **Running Total** table calculation, Tableau can update values cumulatively in other ways than summing. Choose one of the options from the drop-down list just below the **Calculation Type** field:

Option	**Meaning**
Sum	Each value is added to the previous value.
Average	The running total averages the current and all previous values.
Minimum	All values are replaced with the lowest value in the original partition.
Maximum	All values are replaced with the highest value in the original partition.

Restarting every option

The **Restarting every** option is only available when you select **Specific Dimensions** in the Table Calculations dialog box and when more than one dimension is selected in the field immediately below the **Compute Using** options—that is, when more than one dimension is defined as an addressing field.

This option is not available when you're defining a table calculation with **Compute Using**.

You can use this setting to set a break (that is, restart of the calculation) in the view, based on a particular dimension.

Restarting every can be useful in the following situations:

- With dates or other hierarchies, if you restart every month, as you bring in Year or Quarter, Tableau knows to partition automatically.
- With non-hierarchies, **Restarting every** affects the sorting. If you want to address on

Products and partition by **State**, but you want the products sorted by **SUM(Sales)** within each state, you need to include **States** as an addressing field under Specific Dimensions, but then restart every state. Otherwise, the sort by **SUM(Sales)** would be based on each product's sum of sales across all states.

For example, if you take the result of the **Running Total** calculation you added above, you can see the effect of **Restarting every** by doing the following:

1. Click the **SUM(Sales)** field on the Marks card and choose **Edit table calculation**.
2. In the Table Calculation dialog box, choose **Specific Dimensions**.

   Notice that two dimensions are now checked in the list box of dimensions: **Quarter of Order Date** and **Month of Order Date**. These are the addressing fields, and because more than one field is being used for addressing, Restarting every is now available.

   The choices available from the At the level drop-down list are:

None	Specifies that the calculation should be performed at the level of greatest granularity. This is the default option. This option does not change the view.
Quarter of Order Date	Specifies that the calculation should be performed at the quarter level.

3. If you choose Quarter of Order Date, the view updates to show the effect of this change:

		Order Date			
Quarter of O..	Month of Or..	2011	2012	2013	2014
Q1	January	$13,946	$18,174	$18,542	$44,703
	February	$18,757	$30,385	$41,410	$64,987
	March	$74,448	$68,852	$92,596	$118,896
Q2	April	$28,295	$34,195	$39,249	$40,112
	May	$51,944	$64,327	$95,940	$85,764
	June	$86,539	$89,124	$135,370	$134,023
Q3	July	$33,946	$28,765	$38,441	$48,428
	August	$61,856	$65,664	$71,706	$109,944
	September	$143,633	$130,260	$144,614	$200,433
Q4	October	$31,453	$31,405	$56,463	$77,794
	November	$110,082	$107,377	$138,655	$190,120
	December	$179,628	$182,297	$235,893	$280,595

The calculation now restarts after every quarter. If you click out of the Table Calculations dialog box (to dismiss the highlighting) you can see this more clearly.

4. Click the X in the upper-right corner of the Table Calculations dialog box to close it.

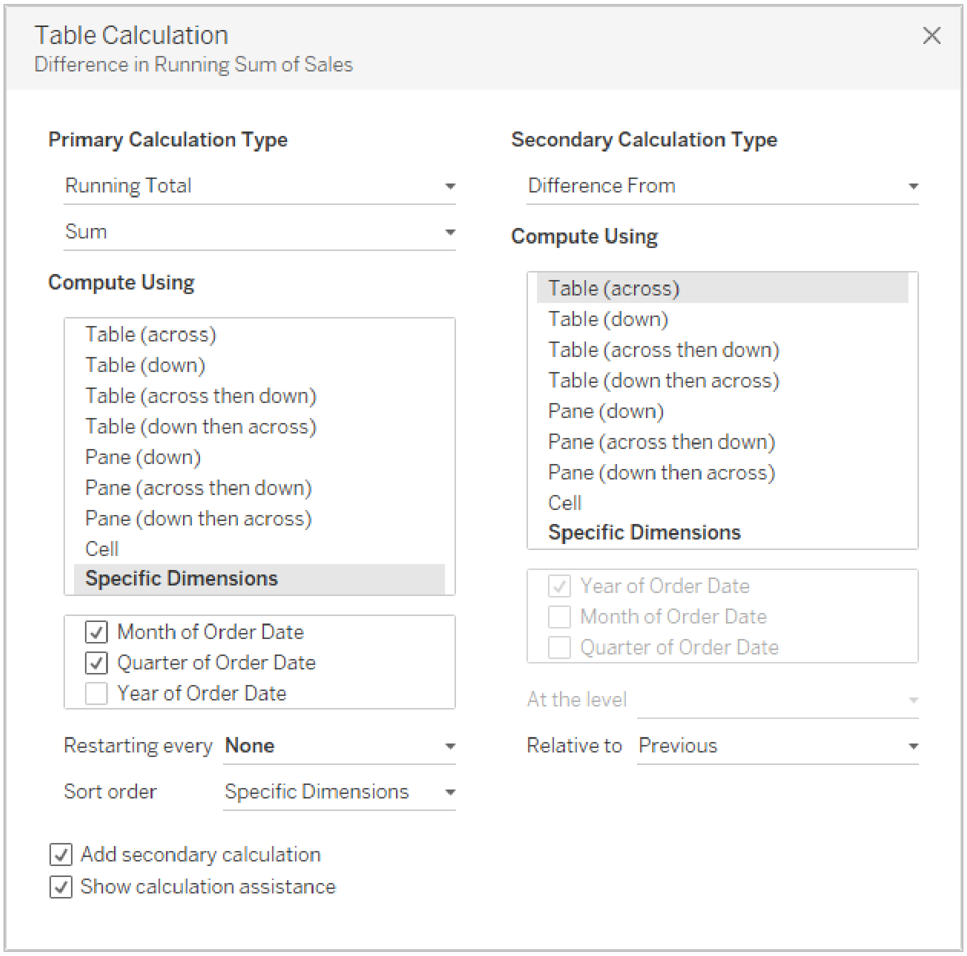
Add Secondary Calculation

With **Running Total** and **Moving Calculation** table calculations, you have the option to transform values twice to obtain the result you want—that is, to add a secondary table calculation on top of the primary table calculation. For example, you could add an initial table calculation to calculate the running total for sales per month within each individual year, and then a secondary calculation to calculate the year-over-year percent difference for each month from one year to the next.

To do this, first add the primary table calculation, as shown above. Then continue as follows:

1. Click the **SUM(Sales)** field on the Marks card and select **Edit table calculation**.
2. In the Table Calculation dialog box, click **Add Secondary Calculation**.

   The Table Calculation dialog box expands to show a second panel:



3. In the second panel, choose **Percent Difference From** as the **Secondary Calculation Type**.

4. You do not need to change the **Compute Using** selection: **Table (Across)** is the right option.

5. Click the X in the upper-right corner to dismiss the Table Calculation dialog box.

   Now your view shows what you needed: a year-over-year percent difference of a running total:

		Order Date			
Quarter of O..	Month of Or..	2011	2012	2013	2014
Q1	January		$4,228	$368	$26,161
	February		$11,628	$11,025	$23,576
	March		-$5,596	$23,745	$26,299
Q2	April		$304	$28,798	$27,163
	May		$6,787	$55,357	$16,123
	June		-$3,011	$69,991	$24,952
Q3	July		-$8,192	$79,666	$34,940
	August		$797	$76,033	$63,191
	September		-$16,384	$84,345	$80,771
Q4	October		-$16,433	$109,404	$102,102
	November		-$19,089	$115,623	$132,236
	December		-$13,715	$137,941	$125,473

See Also

Create a table calculation on page 2416

The basics: addressing and partitioning on page 2408

Quick Table Calculations

Quick table calculations allow you to quickly apply a common table calculation to your visualization using the most typical settings for that calculation type. This article demonstrates how to apply a quick table calculation to a visualization using an example.

The following quick table calculations are available in Tableau for you to use:

- Running total
- Difference
- Percent difference
- Percent of total
- Rank
- Percentile
- Moving average
- YTD total
- Compound growth rate
- Year of year growth
- YTD growth

For more information about some of these, see **Table Calculation Types** on page 2419.

How does a quick table calculation differ from a table calculation?

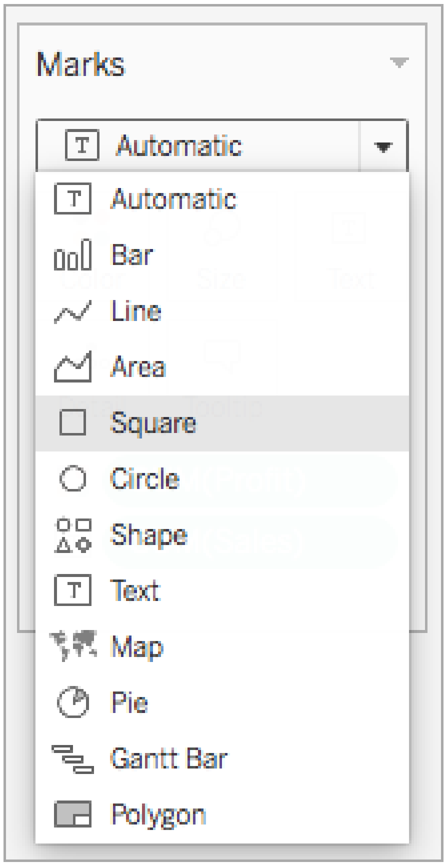
Quick table calculations are table calculations that you can apply quickly to your visualization in Tableau. They are applied to the visualization with the most typical settings for the calculation type you choose so that you can continue on with your analysis. With traditional table calculations, you can apply the same settings, but you must apply them manually.

Apply a quick table calculation to the visualization

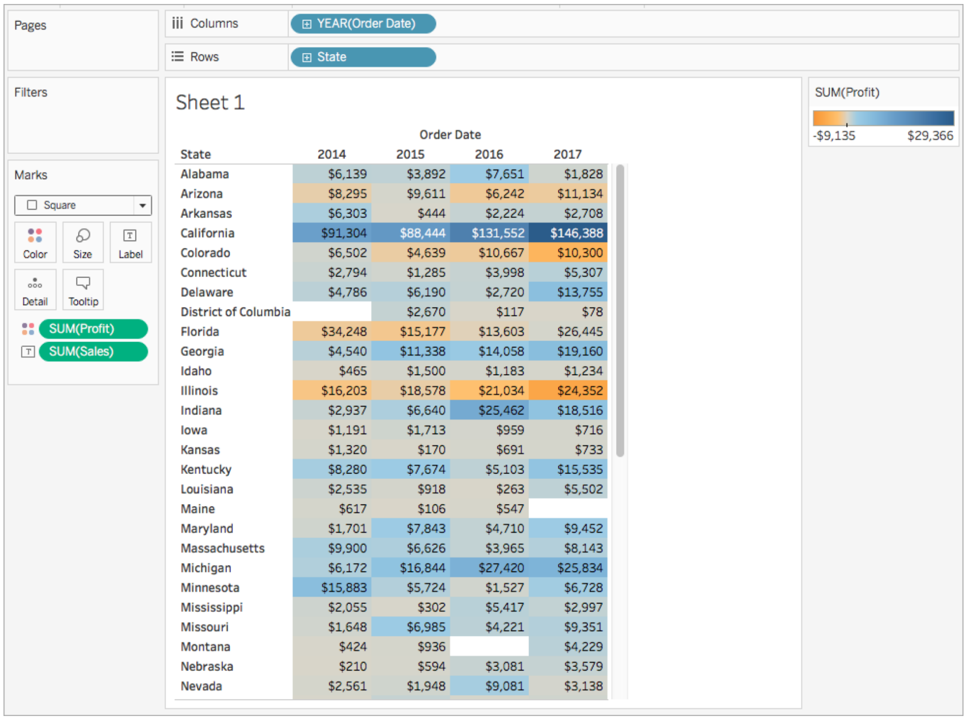
Follow along with the steps below to learn how to apply a quick table calculation to a visualization.

Step 1Set up the visualization

1. Open Tableau Desktop and connect to the **Sample-Superstore** data source, which comes with Tableau.
2. Navigate to a new worksheet.
3. From the **Data** pane, under Dimensions, drag **Order Date** to the **Columns** shelf.
4. From the **Data** pane, under Dimensions, drag **State** to the **Rows** shelf.
5. From the **Data** pane, under Measures, drag **Sales** to **Text** on the Marks Card.
6. From the **Data** pane, under Measures, drag **Profit** to **Color** on the Marks Card.
7. On the Marks card, click the Mark Type drop-down and select **Square**.



The visualization updates to look like this:

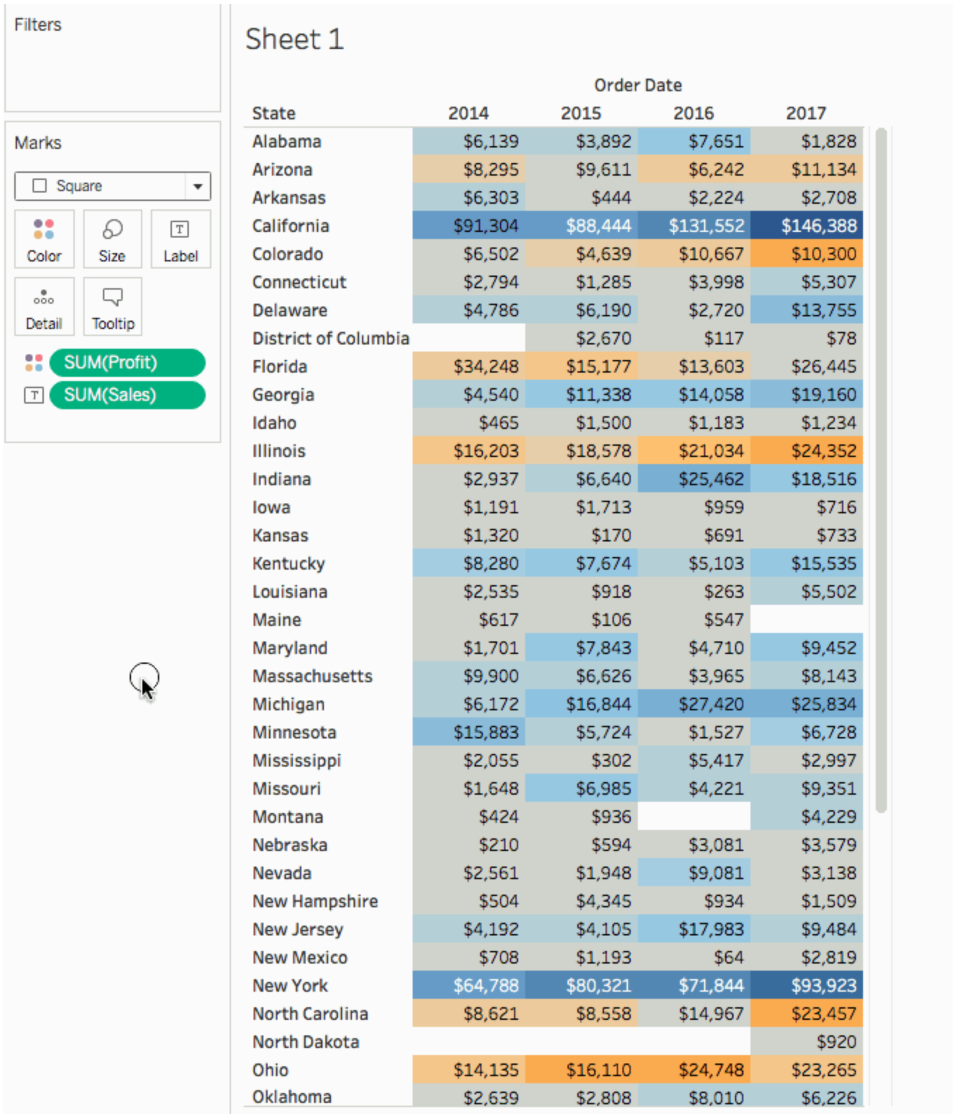


Step 2Apply the quick table calculation

1. On the Marks card, right-click **SUM(Profit)** and select **Quick Table Calculation** > **Moving Average**.

   Note: You can only perform quick table calculations on measures in the view.

   A delta symbol appears on the field to indicate that a quick table calculation is being applied to the field. The colors in the visualization update to show the moving average of profit across the years.



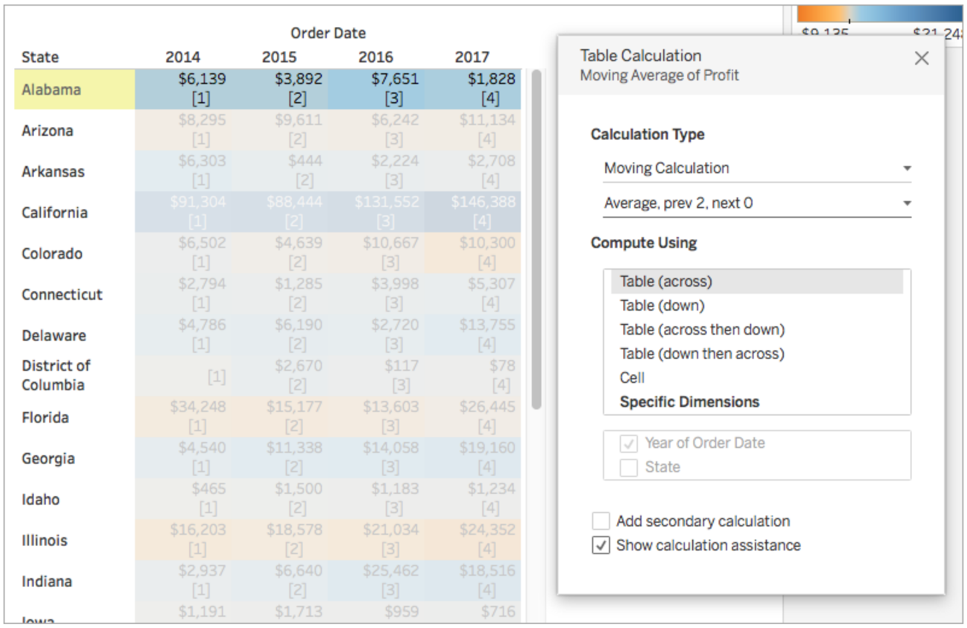
State	2014	2015	2016	2017
Alabama	$6,139	$3,892	$7,651	$1,828
Arizona	$8,295	$9,611	$6,242	$11,134
Arkansas	$6,303	$444	$2,224	$2,708
California	$91,304	$88,444	$131,552	$146,388
Colorado	$6,502	$4,639	$10,667	$10,300
Connecticut	$2,794	$1,285	$3,998	$5,307
Delaware	$4,786	$6,190	$2,720	$13,755
District of Columbia		$2,670	$117	$78
Florida	$34,248	$15,177	$13,603	$26,445
Georgia	$4,540	$11,338	$14,058	$19,160
Idaho	$465	$1,500	$1,183	$1,234
Illinois	$16,203	$18,578	$21,034	$24,352
Indiana	$2,937	$6,640	$25,462	$18,516
Iowa	$1,191	$1,713	$959	$716
Kansas	$1,320	$170	$691	$733
Kentucky	$8,280	$7,674	$5,103	$15,535
Louisiana	$2,535	$918	$263	$5,502
Maine	$617	$106	$547	
Maryland	$1,701	$7,843	$4,710	$9,452
Massachusetts	$9,900	$6,626	$3,965	$8,143
Michigan	$6,172	$16,844	$27,420	$25,834
Minnesota	$15,883	$5,724	$1,527	$6,728
Mississippi	$2,055	$302	$5,417	$2,997
Missouri	$1,648	$6,985	$4,221	$9,351
Montana	$424	$936		$4,229
Nebraska	$210	$594	$3,081	$3,579
Nevada	$2,561	$1,948	$9,081	$3,138
New Hampshire	$504	$4,345	$934	$1,509
New Jersey	$4,192	$4,105	$17,983	$9,484
New Mexico	$708	$1,193	$64	$2,819
New York	$64,788	$80,321	$71,844	$93,923
North Carolina	$8,621	$8,558	$14,967	$23,457
North Dakota				$920
Ohio	$14,135	$16,110	$24,748	$23,265
Oklahoma	$2,639	$2,808	$8,010	$6,226

Step 3(Optional) Customize the quick table calculation

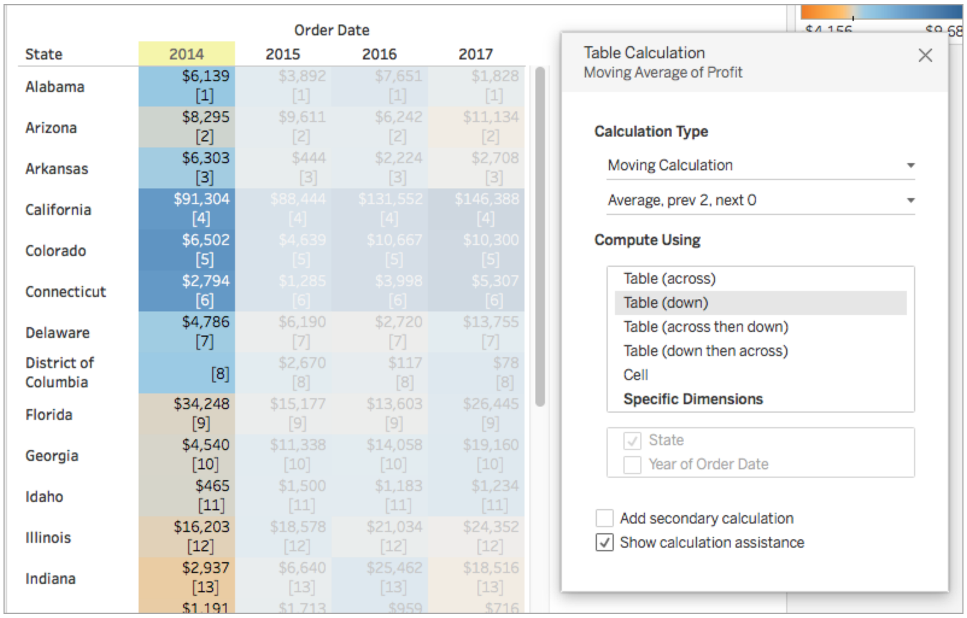
1. On the Marks card, right-click **Sum(Profit)** and select **Edit Table Calculation**.
2. In the dialog box that opens, you can configure the following options:
   - The calculation type
   - How to aggregate the values
   - How to compute the calculation (how to address and partition the calculation)

For more information about these options, see **Table Calculation Types** on page 2419 and **The basics: addressing and partitioning** on page 2408

The visualization updates as you make changes to the calculation. Highlighting and numbering are used to demonstrate how the calculation is being computed. For example, in the following image, the calculation is being computed across the table, for each State.



If this setting is changed, the visualization and highlighting update to indicate the change.



See Also

Understanding Calculations: Table calculations

Transform Values with Table Calculations on page 2406

Customize Table Calculations below

Customize Table Calculations

You can always customize a table calculation by editing it in the Table Calculations dialog box, but there are other, more specialized ways to customize a table calculation.

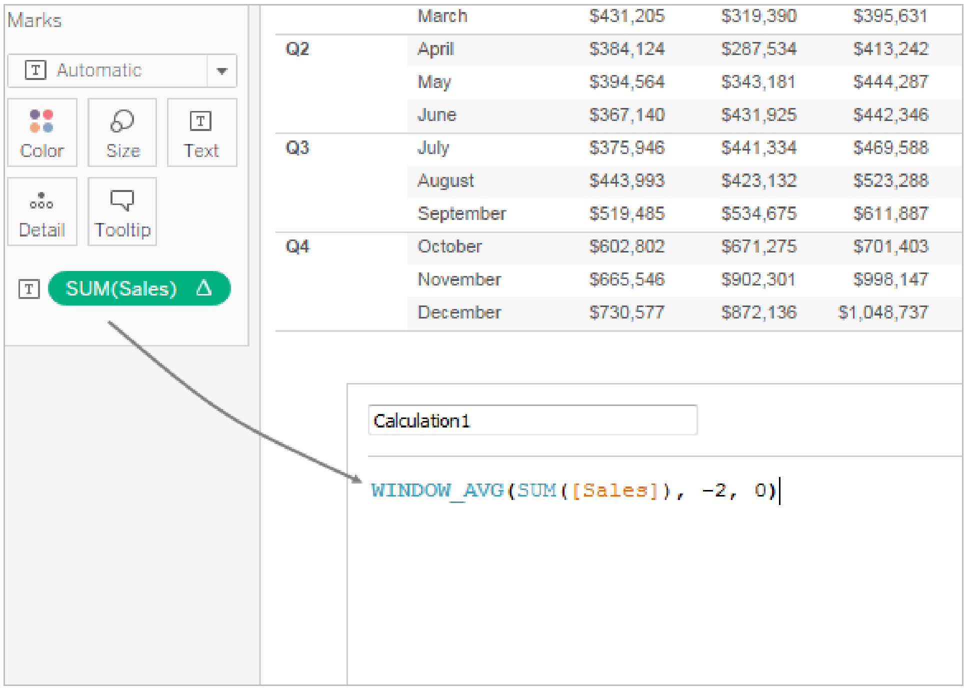
Customizing a Table Calculation Using Its Context Menu

Click any field in the view to see a context menu listing ways to customize the field. For a field in the view that has a table calculation, you can change the **Compute Using** option—that is, the option that determines the direction and scope of the calculation relative to the visual structure of the view. Do this by clicking the field and then choosing an option from the **Compute Using** list.

For **Difference From**, **Percent Difference From**, and **Percent From** table calculations, you can also specify a different field from which the difference should be computed. Do this by clicking the field and then choosing an option from the **Relative to** list. The options are **Previous**, **Next**, **First**, and **Last**.

Customizing a Table Calculation Using the Calculation Editor

You can customize a table calculation by dragging it into the calculation editor:



When you edit a table calculation in the calculation editor, you can click **Default Table Calculation** in the lower-right corner of the editor to open the calculation in the Table Calculation dialog box. This will allow you to create a new named calculated field that uses the same table calculation as the one you are basing the calculation on.

Nested Table Calculations

A nested table calculation can be one of two types of calculated fields:

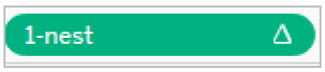
- A calculated field that includes more than one calculated field with a table calculation (as in the example below), or
- A calculated field that itself has a table calculation and includes at least one calculated field with a table calculation.

With nested table calculations, you can set **Compute Using** configurations for individual calculations independently.

Here is a scenario you can try, using the Sample - Superstore data source that is included with Tableau Desktop, which results in a nested table calculation.

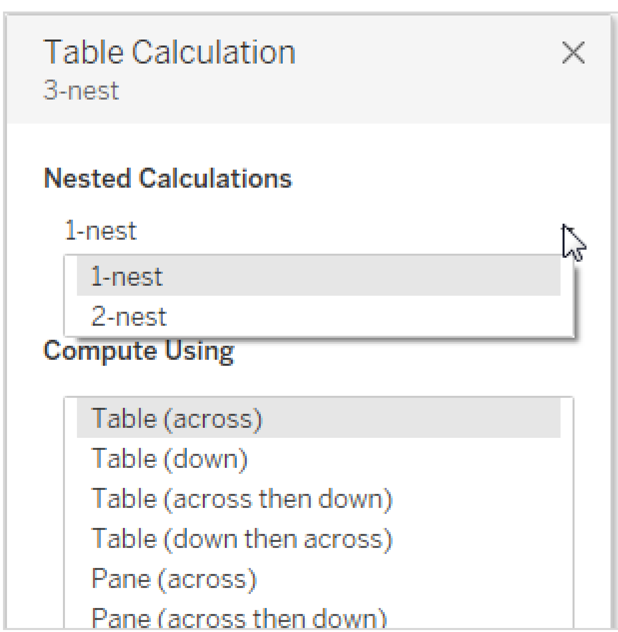
1. Drag **Sub-Category** to Columns and **Region** to Rows.
2. Create a calculated field, *1-nest*, with the definition `TOTAL(SUM([Sales]))`.

   TOTAL is a table calculation function, so this calculated field automatically has a table calculation—when you use it in the view, the field will have the tell-tale table triangle, indicating a table calculation:

   For information on table calculation functions, see **Table Calculation Functions** on page 2219.
3. Create a second calculated field, *2-nest*, with the definition `TOTAL(SUM([Profit]))`.
4. Create a third calculated field, *3-nest*, with the definition `[1-nest] + [2-nest]`.
5. Drag *3-nest* and drop it to the right of `Sub-Category` on Columns.
6. Click *3-nest* on Columns and choose **Edit Table Calculation**.

   In the Table Calculations dialog box, you can now separately configure the underlying table calculations:

Create Level of Detail Expressions in Tableau

Level of Detail expressions (also known as LOD expressions) allow you to compute values at the data source level and the visualization level. However, LOD expressions give you even more control on the level of granularity you want to compute. They can be performed at a more granular level (INCLUDE), a less granular level (EXCLUDE), or an entirely independent level (FIXED).

This article explains the types of LOD expressions you can use in Tableau, as well as when to use them, and how to format them. It also uses an example to demonstrate how to create a simple LOD expression.

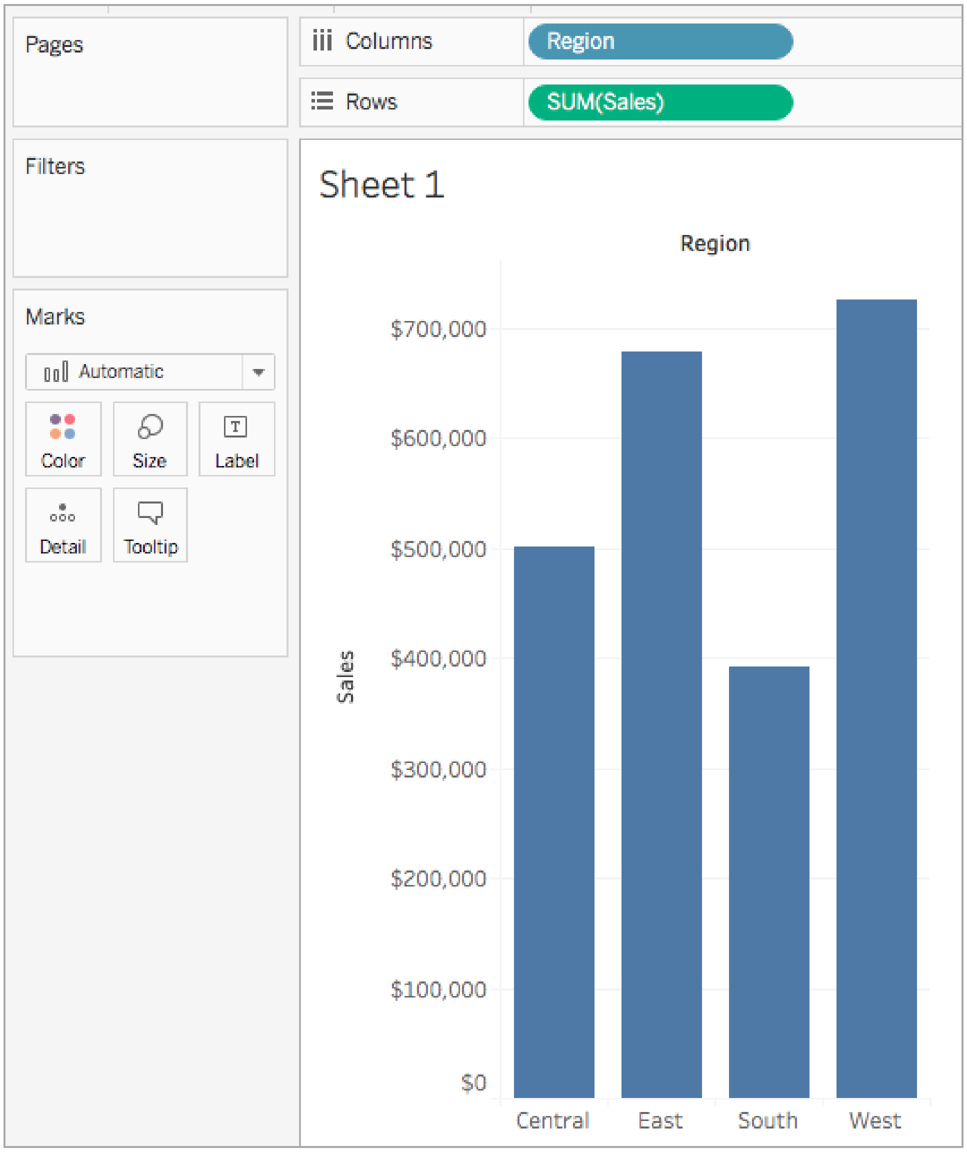
How to create LOD expressions

Follow along with the steps below to learn how to create and use an LOD expression in Tableau.

Step 1: Set up the Visualization

1. Open Tableau Desktop and connect to the **Sample-Superstore** saved data source.
2. Navigate to a new worksheet.
3. From the **Data** pane, under Dimensions, drag **Region** to the **Columns** Shelf.
4. From the **Data** pane, under Measures, drag **Sales** to the **Rows** Shelf.

A bar chart showing the sum of sales for each region appears.



Step 2: Create the LOD expression

Instead of the sum of all sales per region, perhaps you want to also see the average sales per customer for each region. You can use an LOD expression to do this.

1. Select **Analysis** > **Create Calculated Field**.
2. In the Calculation editor that opens, do the following:

- Name the calculation, Sales Per Customer.
- Enter the following LOD expression:

```
{ INCLUDE [Customer Name] : SUM([Sales]) }
```

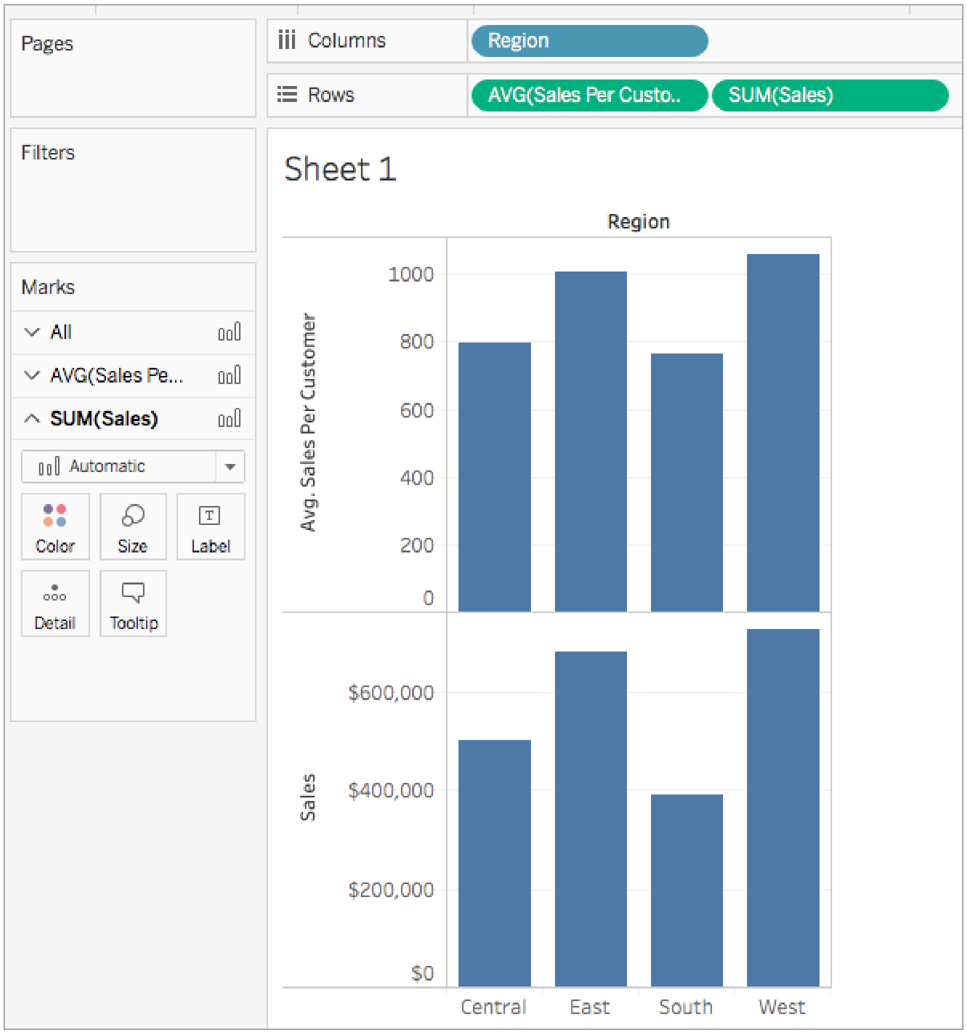
3. When finished, click **OK**.

   The newly created LOD expression is added to the Data pane, under Measures. To learn more about the types of LOD expressions you can use, see the **Types of LOD expressions** on page 2453 section.

Step 3: Use the LOD expression in the visualization

1. From the **Data** pane, under Measures, drag **Sales Per Customer** to the **Rows** shelf and place it to the left of SUM(Sales).
2. On the Rows shelf, right-click **Sales Per Customer** and select **Measure (Sum)** > **Average**.

   You can now see both the sum of all sales and the average sales per customer for each region. For example, you can see that in the Central region, the sales totaled approximately 500,000 USD with an average sale for each customer being approximately 800 USD.



Use a Quick LOD expression

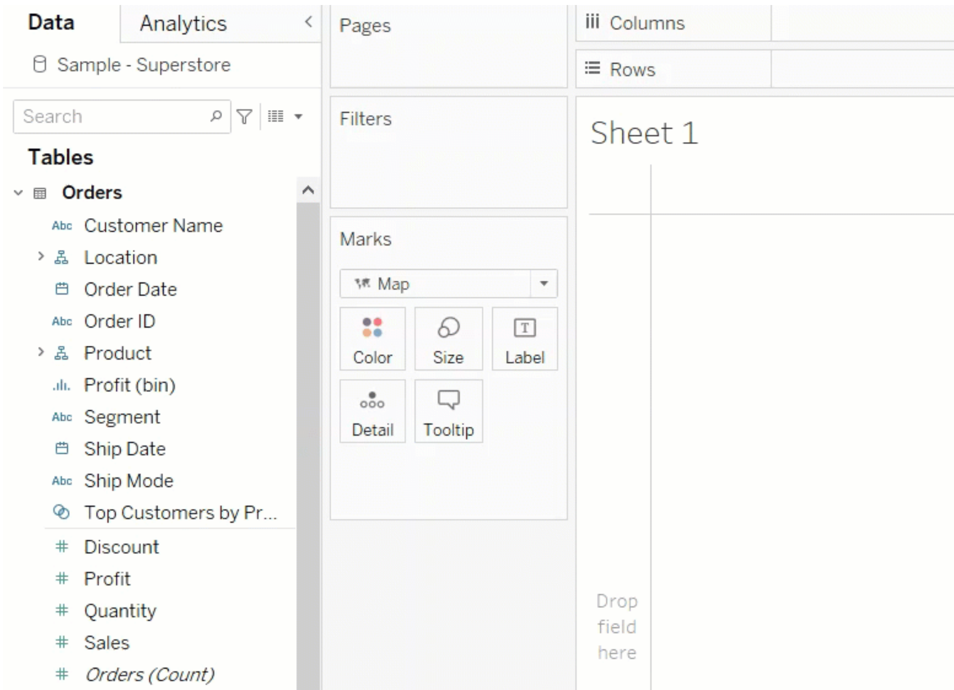
As of Tableau 2021.1, you can create a FIXED LOD expression without needing to enter the full calculation into the calculation dialog.

There are two ways to create a quick LOD calculation.

1. Control-click (or Command-click on a Mac) the measure you want to aggregate and drag it to the dimension you want to aggregate on. A new field will appear with a new

FIXED LOD calculation.

2. As a second option, select the measure you want to aggregate and then Control-click (or Command-click on a Mac) to select the dimension you want to aggregate on.
   - Right-click on the selected fields and select **Create > Quick LOD.**
   - Select **OK** in the calculation dialog that appears to finish creating the LOD.



Types of LOD expressions

There are three types of LOD expressions you can create in Tableau:

- **FIXED** below
- **INCLUDE** on page 2455
- **EXCLUDE** on page 2459

You can also scope an LOD expression to the table. This is called a **Table-Scoped** on page 2461 LOD expression.

FIXED

FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view.

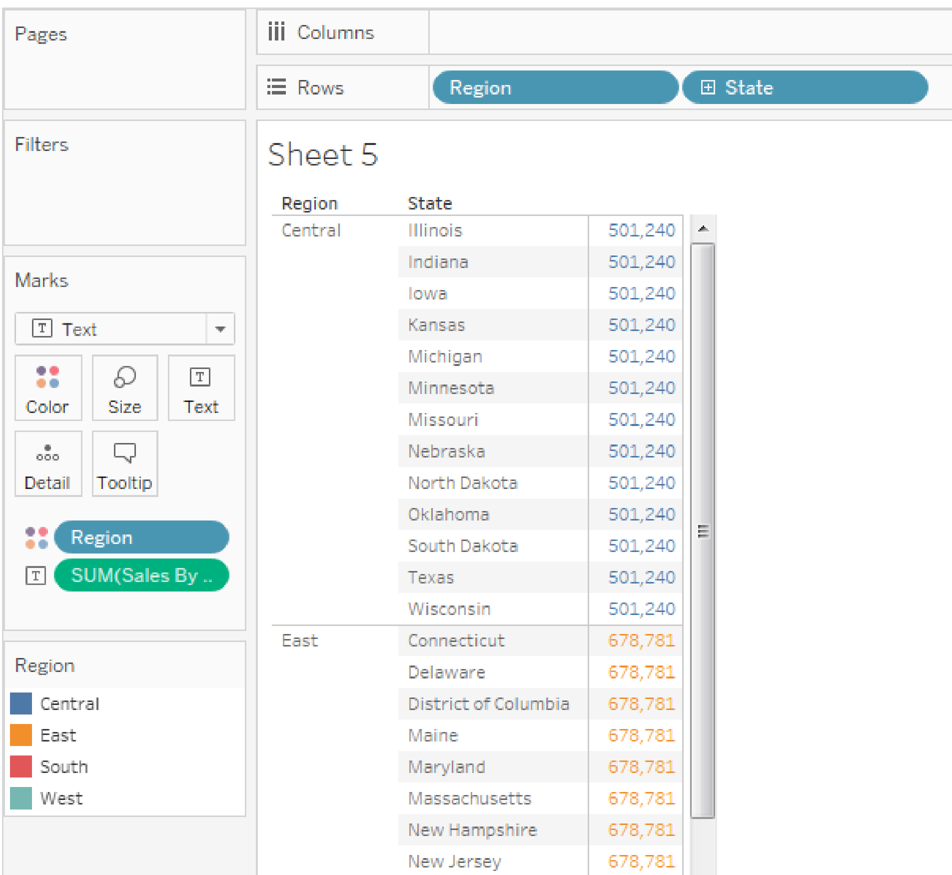
Example

The following FIXED level of detail expression computes the sum of sales per region:

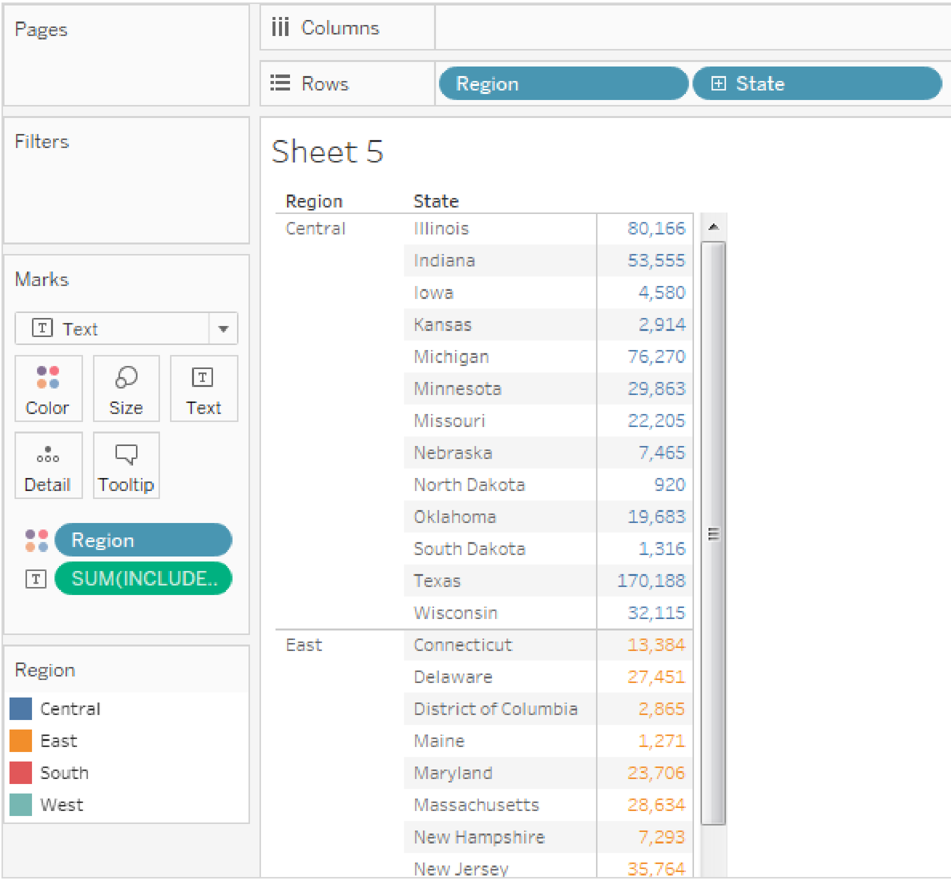
```
{FIXED [Region] : SUM([Sales])}
```

This level of detail expression, named [**Sales by Region**], is then placed on Text to show total sales per region.

The view level of detail is [**Region**] plus [**State**], but because FIXED level of detail expressions do not consider the view level of detail, the calculation only uses the dimension referenced in the calculation, which in this case is Region. Because of this, you can see that the values for the individual states in each region are identical. For more information about why this happens, see **Aggregation and Level of Detail Expressions** on page 2469.



If the INCLUDE keyword had been used in the level of detail expression instead of FIXED, the values would be different for each state, because Tableau would add the dimension in the expression ([**Region**]) with any additional dimensions in the view ([**State**]) when determining values for the expression. The result would be as follows:



INCLUDE

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

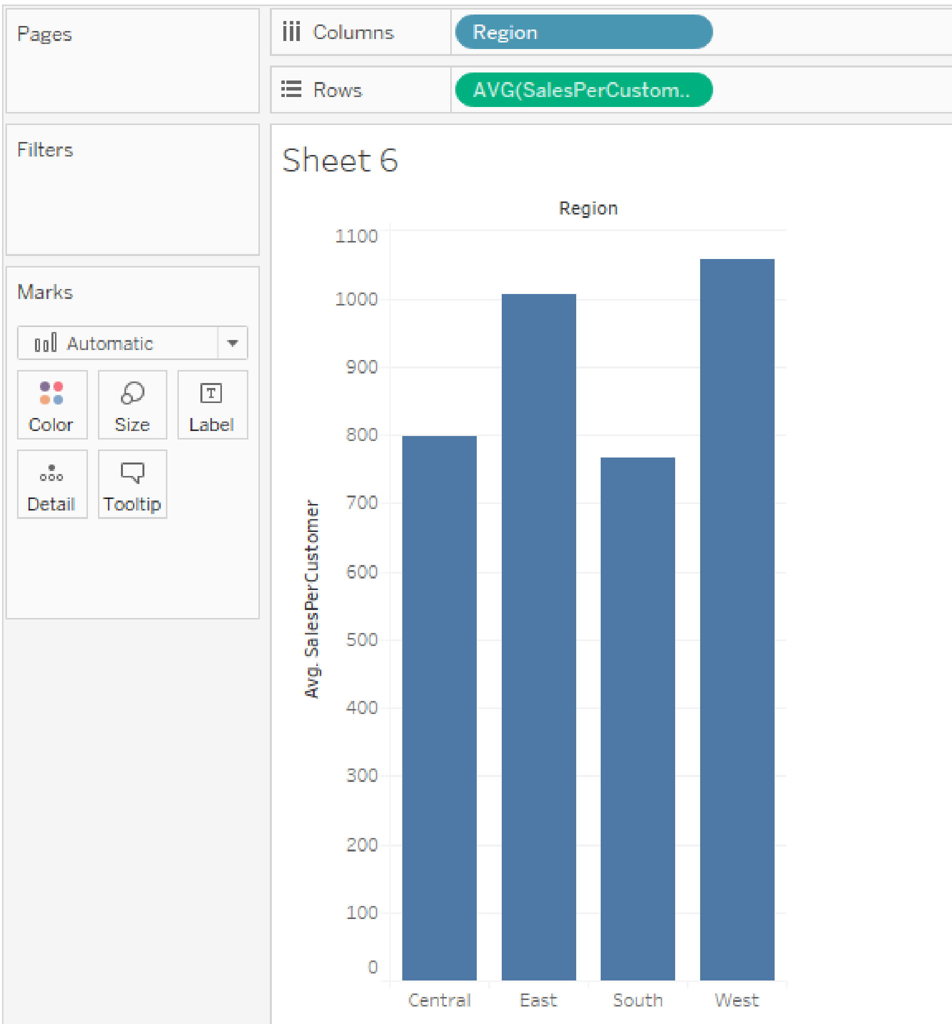
INCLUDE level of detail expressions can be useful when you want to calculate at a fine level of detail in the database and then re-aggregate and show at a coarser level of detail in your view. Fields based on INCLUDE level of detail expressions will change as you add or remove dimensions from the view.

Example 1

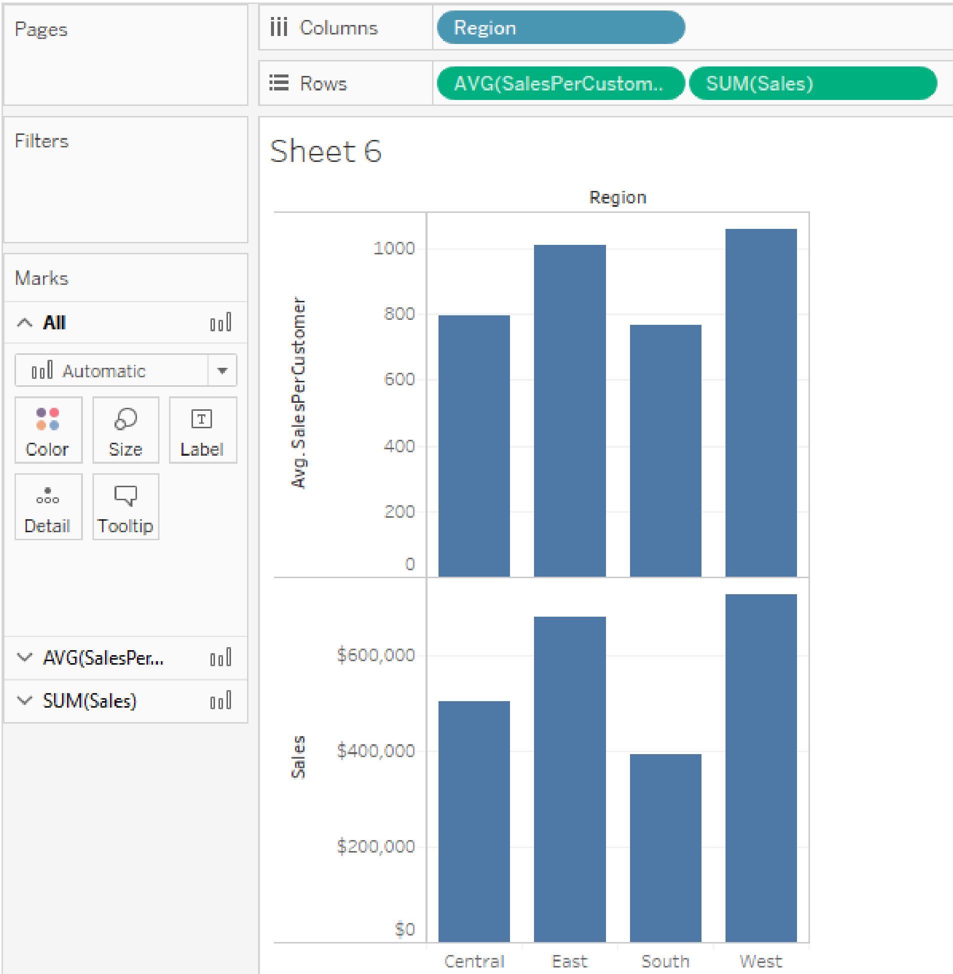
The following INCLUDE level of detail expression computes total sales per customer:

```
{ INCLUDE [Customer Name] : SUM([Sales]) }
```

When that calculation is placed on the **Rows** shelf, aggregated as AVG, and the [**Region**] dimension is placed on the **Columns** shelf, the view shows the average customer sales amount per region:



If the **[Sales]** measure is then dragged to the **Rows** shelf, the result illustrates the difference between the total sale for each region and the average sale per customer for each region:

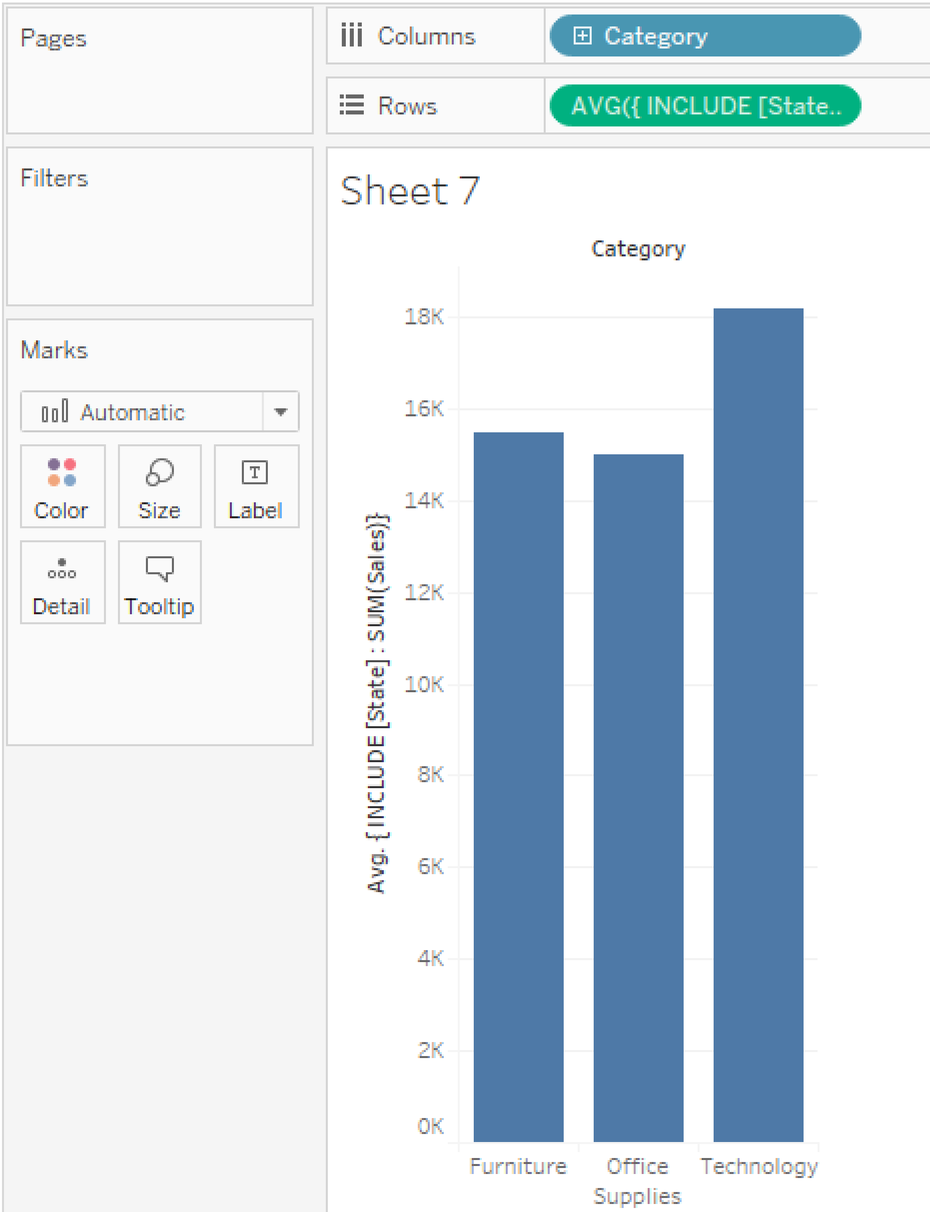


Example 2

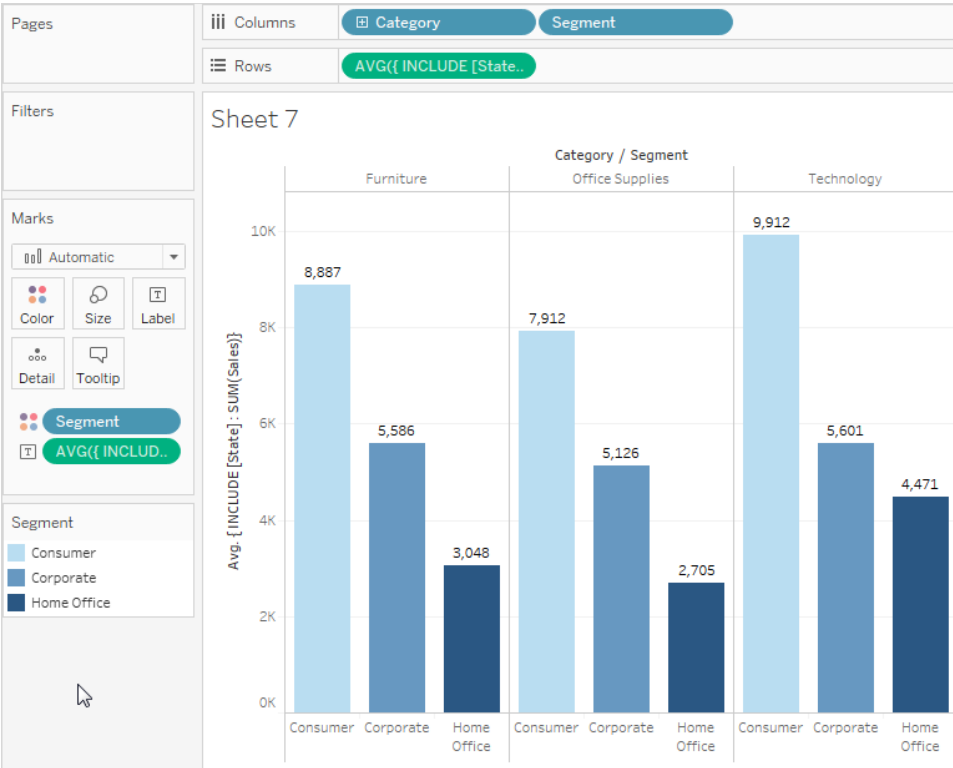
The following INCLUDE level of detail expression calculates sum of sales on a per-state basis:

```
{ INCLUDE [State] : SUM(Sales)}
```

The calculation is placed on the Rows shelf and is aggregated as an average. The resulting visualization averages the sum of sales by state across categories.



When **Segment** is added to the **Columns** shelf and the calculation is moved to **Label**, the LOD expression results update. Now you can see how the average sum of sales per state varies across categories and segments.



EXCLUDE

EXCLUDE level of detail expressions declare dimensions to omit from the view level of detail.

EXCLUDE level of detail expressions are useful for ‘percent of total’ or ‘difference from overall average’ scenarios. They are comparable to such features as Totals and Reference Lines.

EXCLUDE level of detail expression cannot be used in row-level expressions (where there are no dimensions to omit), but can be used to modify either a view level calculation or anything in between (that is, you can use an EXCLUDE calculation to remove dimension from some other level of detail expression).

Example 1

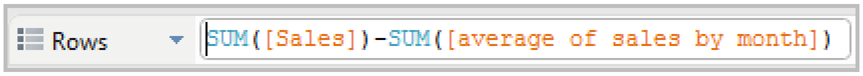
The following EXCLUDE level of detail expression computes the average sales total per month and then excludes the month component:

```
{EXCLUDE [Order Date (Month / Year)] : AVG({FIXED [Order Date
(Month / Year)] : SUM([Sales])})}
```

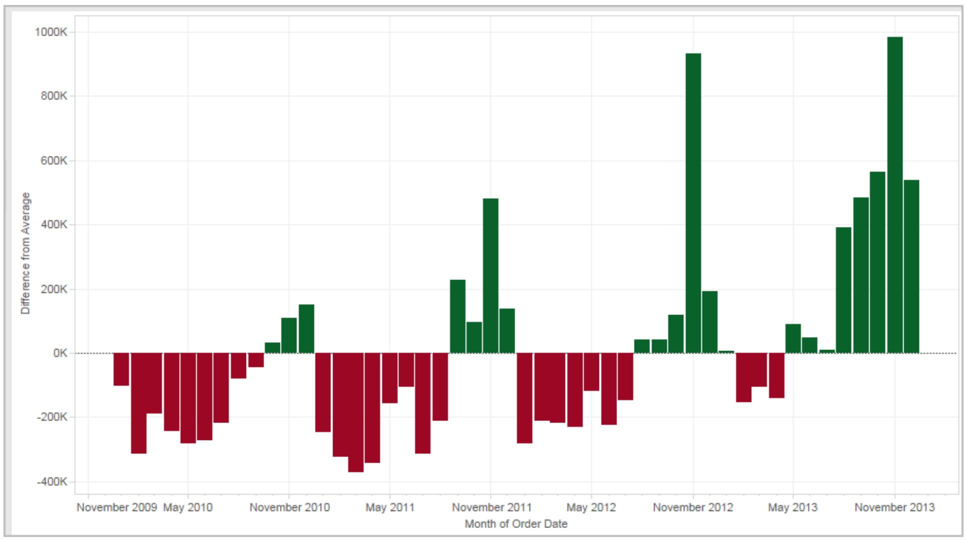
For more information on creating Month / Year date fields, see **Custom Dates** on page 1256.

Notice that this is a nested level of detail expression—that is, a level of detail expression within another level of detail expression.

Saved as [**average of sales by month**], the calculation can then be subtracted from the sum of sales per month by means of an ad-hoc calculation on the **Rows** shelf:



With **Month([Order Date])** on the **Columns** shelf, this creates a view that shows the difference between actual sales per month over a four-year period and the average monthly sales for the entire four-year period:



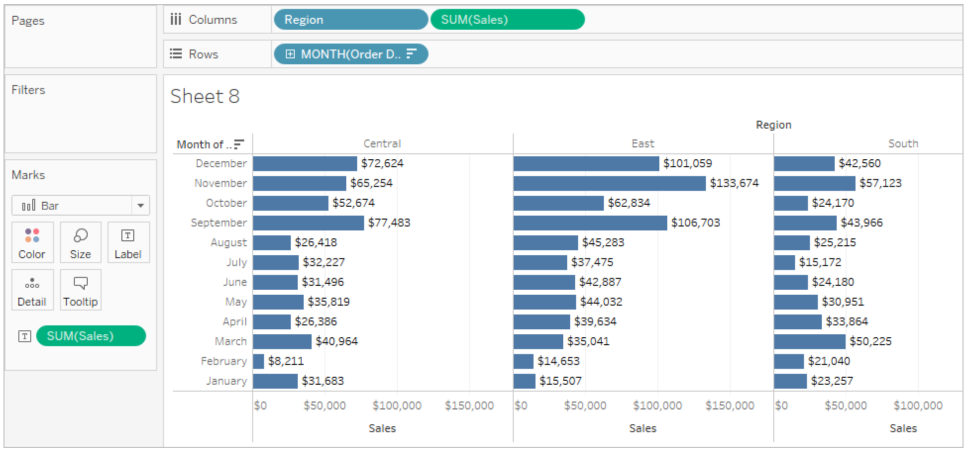
Example 2

The following level of detail expression excludes [**Region**] from a calculation of the sum of [**Sales**]:

```
{EXCLUDE [Region]: SUM([Sales])}
```

The expression is saved as [**ExcludeRegion**].

To illustrate how this expression might be useful, first consider the following view, which breaks out the sum of sales by region and by month:



Dropping [**ExcludeRegion**] on Color shades the view to show total sales by month but without the regional component:

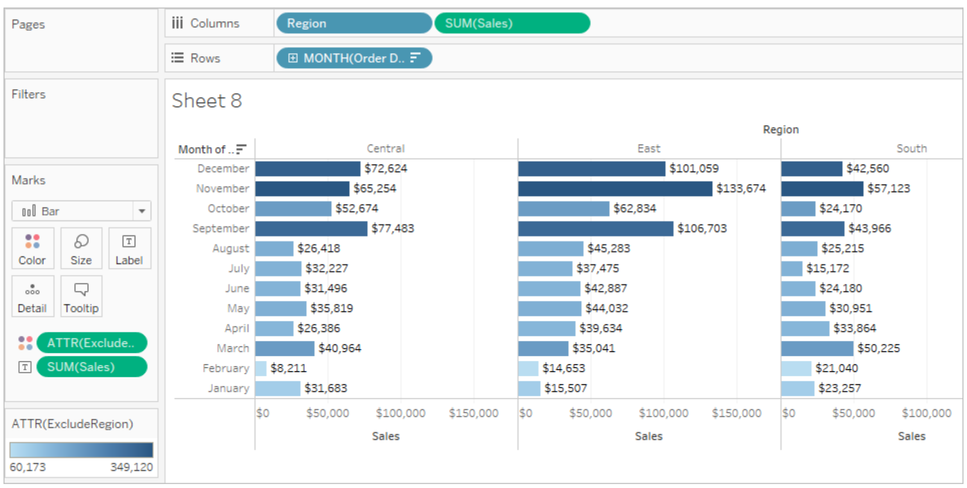


Table-Scoped

It is possible to define a level of detail expression at the table level without using any of the scoping keywords. For example, the following expression returns the minimum (earliest) order date for the entire table:

```
{MIN([Order Date])}
```

This is equivalent to a FIXED level of detail expression with no dimension declaration:

```
{FIXED : MIN([Order Date])}
```

LOD expression syntax

Level of Detail Expression Syntax

A level of detail expression has the following structure:

{[FIXED | INCLUDE | EXCLUDE] <*dimension declaration* > **:** <*aggregate expression*>}

The elements in a level of detail expression are described in the following table.

<table>
<tr><th>Element</th><th>Description</th></tr>
<tr><td>{ }</td><td>The entire level of detail expression is enclosed in curly braces.</td></tr>
<tr><td>[FIXED | INCLUDE | EXCLUDE]</td><td>The first element after the opening curly brace is one of the following scoping keywords:

FIXED

FIXED level of detail expressions compute values using the specified dimensions without reference to the view level of detail—that is, without reference to any other dimensions in the view.

FIXED level of detail expressions also ignore all the filters in the view other than context filters, data source filters, and extract filters.

Example: <code>{ FIXED [Region] : SUM([Sales]) }</code>

For more information about FIXED level of detail expressions, and for some example FIXED level of detail scenarios, see the FIXED on page 2453 section.
INCLUDE

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

INCLUDE level of detail expressions are most useful when including a dimension that isn’t in the view.

Example: <code>{ INCLUDE [Customer Name] : SUM([Sales]) }</code>

For more information about INCLUDE level of detail expressions, and for some example INCLUDE level of detail scenarios, see the INCLUDE on page 2455 section.
EXCLUDE
</td></tr>
</table>

	EXCLUDE level of detail expressions explicitly remove dimensions from the expression—that is, they subtract dimensions from the view level of detail. EXCLUDE level of detail expressions are most useful for eliminating a dimension in the view. Example: `{EXCLUDE [Region]: SUM([Sales])}` For more information about EXCLUDE level of detail expressions, and for some example EXCLUDE level of detail scenarios, see the **EXCLUDE** on page 2459 section . • **Table-Scoped** In the case of a table-scoped level of detail expression, no scoping keyword is required. For more inforamtion, see the **Table-Scoped** on page 2461 section.

<dimen-sion declar-ation>	Specifies one or more dimensions to which the aggregate expression is to be joined. Use commas to separate dimensions. For example:
	
	`[Segment], [Category], [Region]`
	
	For level of detail expressions, you can use any expression that evaluates as dimension in a dimensionality declaration, including Date expressions.
	
	This example will aggregate the sum of **Sales** at the Year level:
	
	`{FIXED YEAR([Order Date]) : SUM(Sales)}`
	
	This example will aggregate the sum of **Sales** for the **[Order Date]** dimension, truncated to the day date part. Because it is an INCLUDE expression, it will also use the dimensions in the view to aggregate the value:
	
	`{INCLUDE DATETRUNC('day', [Order Date]) : AVG (Profit)}`
	
	Note: It is strongly recommended that you drag fields into the calculation editor when creating dimension declarations, instead of typing them. For example, if you see **YEAR([Order Date])** on a shelf and then type that as the dimension declaration, it will not match the

<table>
<tr>
<td></td>
<td>
field on the shelf. But if you drag the field from the shelf into the expression, it will become DATEPART('year', [Order Date]), and that will match the field on the shelf.
With named calculations (that is, calculations that you save to the Data pane, as opposed to ad-hoc calculations, which you do not name), Tableau cannot match the name of a calculation to its definition. So if you create a named calculation, MyCalculation, defined as follows:
<code>MyCalculation = YEAR([Order Date])</code>
And then you created the following EXCLUDE level of detail expression and used it in the view:
<code>{EXCLUDE YEAR([Order Date]) : SUM(Sales)}</code>
Then MyCalculation would not be excluded.
Similarly, if the EXCLUDE expression specified MyCalculation:
<code>{EXCLUDE MyCalculation : SUM(Sales)}</code>
Then YEAR([Order Date]) would not be excluded.
</td>
</tr>
<tr>
<td>:</td>
<td>A colon separates the dimension declaration from the aggregate expression.</td>
</tr>
<tr>
<td><aggregate expression></td>
<td>The aggregate expression is the calculation performed to define the target dimensionality.</td>
</tr>
</table>

See Also

Introduction to Level of Detail Expressions

A Deeper Look at LOD Expressions

Diving into LOD Expressions

Top 15 LOD Expressions

Understanding LOD Expressions

How Level of Detail Expressions Work in Tableau on the facing page

How Level of Detail Expressions Work in Tableau

This article explains how level of detail expressions are computed and how they function in Tableau. For more information about LOD expressions and how they work, see the Understanding Level of Detail (LOD) Expressions whitepaper on the Tableau website.

Row Level Expressions and View Level Expressions

In Tableau, expressions referencing **unaggregated** datasource columns are computed for each row in the underlying table. In this case, the dimensionality of the expression is *row level*. An example of a row-level expression is:

```
[Sales] / [Profit]
```

This calculation will be evaluated in each row of the database. For each row, the Sales value in that row will be divided by the Profit value in that row, producing a new column with the result of the multiplication (a profit ratio).

If you create a calculation with this definition, save it with the name [**ProfitRatio**], and then drag it from the **Data** pane to a shelf, Tableau typically aggregates the calculated field for the view:

```
SUM([ProfitRatio])
```

By contrast, expressions referencing **aggregated** data source columns are computed at the dimensionality defined by the dimensions in the view. In this case, the dimensionality of the expression is view level. An example of a view-level expression is:

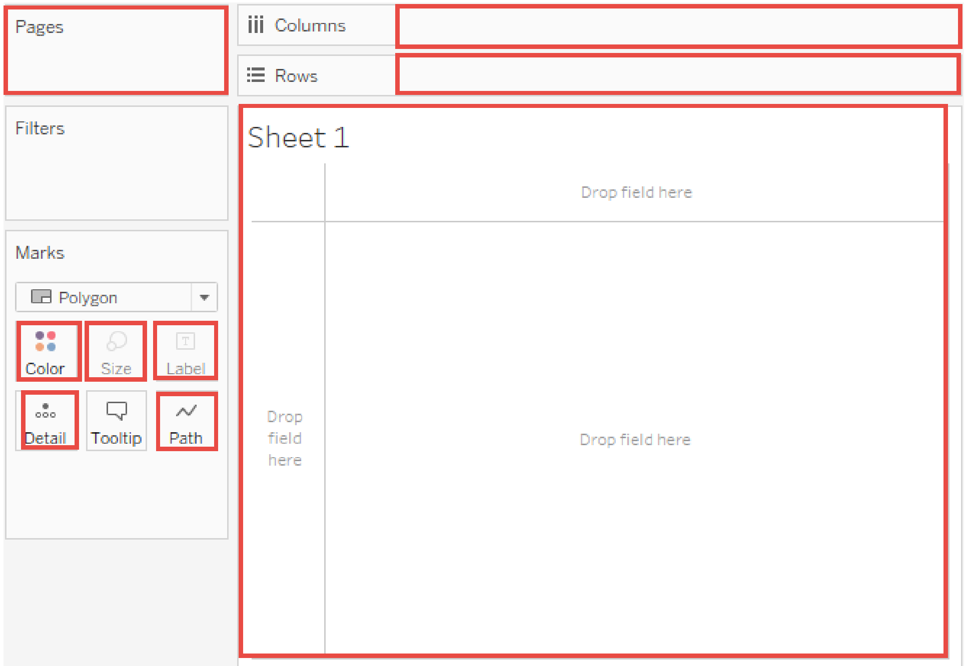
```
SUM(Sales) / SUM(Profit)
```

If you drag this calculation to a shelf (or type it directly on a shelf as an ad-hoc calculation), Tableau encloses it in an AGG function:

AGG(SUM(Sales) / SUM(Profit))

This is what is known as an aggregate calculation. For details, see **Aggregate Functions in Tableau** on page 2192.

Dimension and set fields placed on any of the locations highlighted in the following image contribute to the view level of detail:



Before level of detail expressions were supported in Tableau, it was not possible to create calculations at a level of detail other than the view level. For example, if you attempt to save the following expression, Tableau displays the error message: “Cannot mix aggregate and non-aggregate arguments with this function”:

```
[Sales] – AVG([Sales])
```

The user’s intent in this case was to compare store sales for each individual store to the average of sales for all stores. This can now be accomplished with a level of detail expression:

```
[Sales] – {AVG([Sales])}
```

This is what is known as a table-scoped level of detail expression. See **Table-Scoped** on page 2461

Limitations for Level of Detail Expressions

The following limitations and constraints apply for level of detail expressions. Also see **Data Source Constraints for Level of Detail Expressions** on page 2471.

- Level of detail expressions that reference floating-point measures can behave unreliably when used in a view that requires comparison of the values in the expression. For details, see **Understanding data types in calculations** on page 2146.
- Level of detail expressions are not shown on the Data Source page. See **Data Source Page** on page 188.
- When referencing a parameter in a dimensionality declaration, always use the parameter name, and not the parameter value.
- With data blending, the linking field from the primary data source must be in the view before you can use a level of detail expression from the secondary data source. See **Troubleshoot Data Blending** on page 930.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

Level of Detail Expressions Can Be Dimensions or Measures

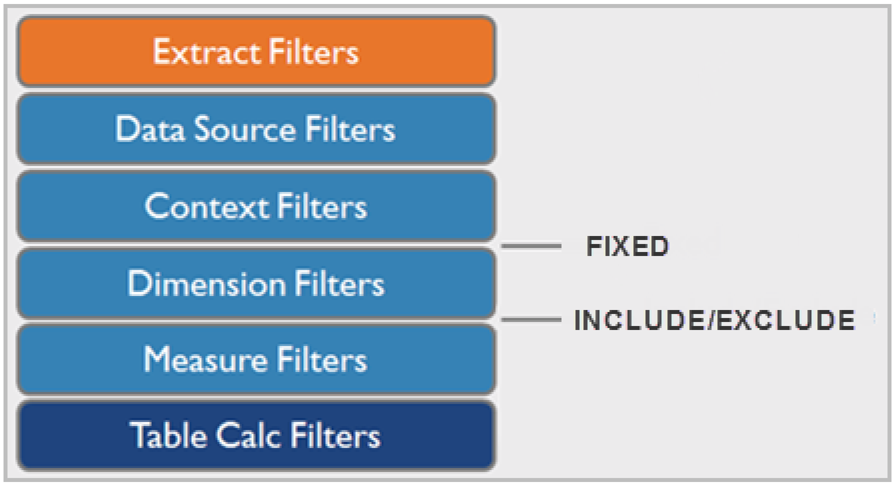
When you save a level of detail expression, Tableau adds it to either the Dimensions or the Measures area in the Data pane.

FIXED level of detail expressions can result in measures or dimensions, depending on the underlying field in the aggregate expression. So MIN([Date])} will be a dimension because [Date] is a dimension, and {fixed Store : SUM([Sales])} will be a measure because [Sales] is a measure. When a FIXED level of detail expression is saved as a measure you have the option of moving it to dimensions.

INCLUDE and EXCLUDE level of detail expressions are always measures.

Filters and Level of Detail Expressions

There are several different kinds of filters in Tableau and they get executed in the following order from top to bottom.



The text on the right shows where level of detail expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you're creating a Tableau Extract from a data source. Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

If you're familiar with SQL, you can think of measure filters as equivalent to the HAVING clause in a query, and dimension filters as equivalent to the WHERE clause.

FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to **Improve View Performance with Context Filters** on page 1439, they will be ignored. For example, consider if you have the following calculation on one shelf in a view, along with [**State**] on a different shelf:

```
SUM([Sales]) / ATTR({FIXED : SUM([Sales])})
```

This calculation will give you the ratio of a state's sales to total sales.

If you then put [**State**] on the Filters shelf to hide some of the states, the filter will affect only the numerator in the calculation. Since the denominator is a FIXED level of detail expression, it will still divide the sales for the states still in the view against the total sales for all states—including the ones that have been filtered out of the view.

INCLUDE and EXCLUDE level of detail expressions are considered after Dimension filters. So if you want filters to apply to your FIXED level of detail expression but don't want to use Context Filters, consider rewriting them as INCLUDE or EXCLUDE expressions.

Aggregation and Level of Detail Expressions

The level of detail of the view determines the number of marks in your view. When you add a level of detail expression to the view, Tableau must reconcile two levels of detail—the one in the view, and the one in your expression.

The behavior of a level of detail expression in the view varies depending on whether the expression's level of detail is coarser, finer, or the same as the level of detail in the view. What do we mean by “coarser” or “finer” in this case?

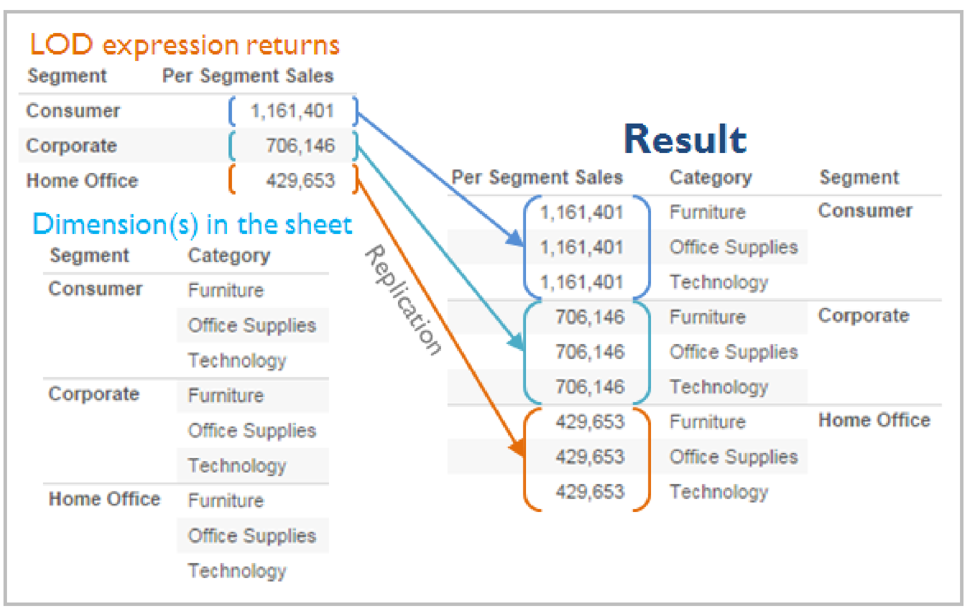
Level of Detail Expression is Coarser Than View Level of Detail

An expression has a coarser level of detail than the view when it references a subset of the dimensions in the view. For example, for a view that contained the dimensions [**Category**] and [**Segment**], you could create a level of detail expression that uses only one of these dimensions:

```
{FIXED [Segment] : SUM([Sales])}
```

In this case, the expression has a coarser level of detail than the view. It bases its values on one dimension ([**Segment**]), whereas the view is basing its view on two dimensions ([**Segment**] and [**Category**]).

The result is that using the level of detail expression in the view causes certain values to be replicated—that is, to appear multiple times.



Replicated values are useful for comparing specific values against average values within a category. For example the following calculation subtracts average sales for a customer from the average sales overall:

```
[Sales] - {FIXED [Customer Name] : AVG([Sales])}
```

When values are being replicated, changing the aggregation for the relevant field in the view (for example, from AVG to SUM) will not change the result of the aggregation.

Level of Detail Expression is Finer Than View Level of Detail

An expression has a finer level of detail than the view when it references a superset of the dimensions in the view. When you use such an expression in the view, Tableau will aggregate results up to the view level. For example, the following level of detail expression references two dimensions:

```
{FIXED [Segment], [Category] : SUM([Sales])}
```

When this expression is used in a view that has only [Segment] as its level of detail, the values must be aggregated. Here's what you would see if you dragged that expression to a shelf:

```
AVG([{FIXED [Segment]], [Category]] : SUM([Sales]])}])
```

An aggregation—in this case, average—is automatically assigned by Tableau. You can change the aggregation as needed.

Adding a Level of Detail Expression to the View

Whether a level of detail expression is aggregated or replicated in the view is determined by the expression type (FIXED, INCLUDE, or EXCLUDE) and whether the expression's granularity is coarser or finer than the view's.

- INCLUDE level of detail expressions will have either the same level of detail as the view or a finer level of detail than the view. Therefore, values will never be replicated.
- FIXED level of detail expressions can have a finer level of detail than the view, a coarser level of detail, or the same level of detail. The need to aggregate the results of a FIXED level of detail depends on what dimensions are in the view.
- EXCLUDE level of detail expressions always cause replicated values to appear in the view. When calculations including EXCLUDE level of detail expressions are placed on a shelf, Tableau defaults to the ATTR aggregation (as opposed to SUM or AVG) to

indicate that the expression is not actually being aggregated and that changing the aggregation will have no effect on the view.

Level of detail expressions are always automatically wrapped in an aggregate when they are added to a shelf in the view unless they're used as dimensions. So if you double-click on a shelf and type

```
{FIXED[Segment], [Category] : SUM([Sales])}
```

and then press Enter to commit the expression, what you now see on the shelf is

```
SUM({FIXED[Segment], [Category] : SUM([Sales])})
```

But if you double-click into the shelf to edit the expression, what you see in edit mode is the original expression.

If you wrap a level of detail expression in an aggregation when you create it, Tableau will use the aggregation you specified rather than assigning one when any calculation including that expression is placed on a shelf. When no aggregation is needed (because the expression's level of detail is coarser than the view's), the aggregation you specified is still shown when the expression is on a shelf, but it is ignored.

Data Source Constraints for Level of Detail Expressions

For some data sources, only more recent versions support level of detail expressions. Some data sources do not support level of detail expressions at all.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

Data Source	Support
Actian Vectorwise	Not supported.
Amazon EMR Hadoop Hive	Supported for Hive 0.13 and later.
Amazon Redshift	Supported.
Aster Database	Supported for version 4.5 and later.
Cloudera Hadoop	Supported for Hive 0.13 and later.
Cloudera Impala	Supported for Impala 1.2.2 and later.

Cubes (multidimensional data sources)	Not supported.
DataStax Enterprise	Not supported.
EXASOL	Supported.
Firebird	Supported for version 2.0 and later.
Generic ODBC	Limited. Depends on the specific data source.
Google Big Query	Supported for standard SQL, not supported for legacy SQL.
Hortonworks Hadoop Hive	Supported for Hive 0.13 and later. On version 1.1 of HIVE level of detail expressions that produce cross joins are not reliable. Cross join occur when there is no explicit field to join on. For example, for a level of detail expression `{fixed [Product Type] : sum (sales)}` when the view only contains one dimension [**Ship Mode**], Tableau creates a cross-join. A cross join produces rows which combine each row from the first table with each row from the second table.
IBM BigInsights	Supported.
IBM DB2	Supported for version 8.1 and later.
MarkLogic	Supported for version 7.0 and later.
Microsoft Access	Not supported.
Microsoft Jet-based connections (legacy connectors for Microsoft Excel, Microsoft Access, and text)	Not supported.

Microsoft SQL Server	SQL Server 2005 and later.
MySQL	Supported.
IBM PDA (Netezza)	Supported version 7.0 and later.
Oracle	Supported version 9i and later.
Actian Matrix (ParAccel)	Supported version 3.1 and later.
Pivotal Greenplum	Supported for version 3.1 and later.
PostgreSQL	Supported version 7 and later.
Progress OpenEdge	Supported.
SAP HANA	Supported.
SAP Sybase ASE	Supported.
SAP Sybase IQ	Supported version 15.1 and later.
Spark SQL	Supported.
Splunk	Not supported.
Tableau Data Extract	Supported.
Teradata	Supported.
Vertica	Supported for version 6.1 and later.

See Also

Create Level of Detail Expressions in Tableau on page 2449

Understanding Level of Detail (LOD) Expressions

Table-Scoped Level of Detail Expressions

It is possible to define a level of detail expression at the table level without using any of the scoping keywords. For example, the following expression returns the minimum (earliest) order date for the entire table:

```
{MIN([Order Date])}
```

This is equivalent to a FIXED level of detail expression with no dimension declaration:

```
{FIXED : MIN([Order Date])}
```

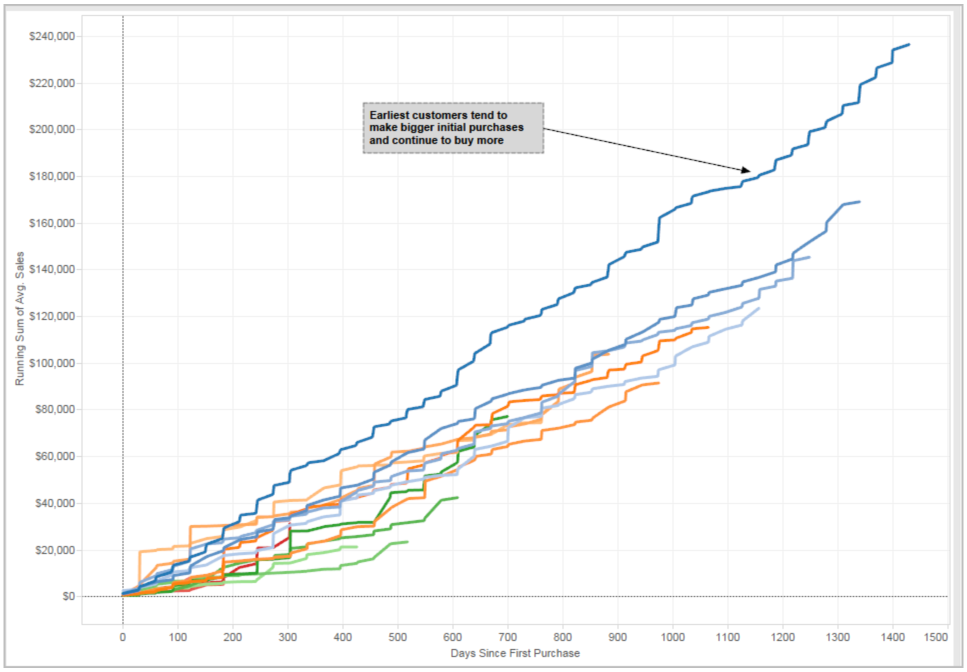
FIXED Level of Detail Expressions

FIXED level of detail expressions compute a value using the specified dimensions, without reference to the dimensions in the view.

The following examples can be recreated using the **Sample - Superstore** data source.

Example 1

The following view shows the interval between a customer's first purchase date and any subsequent purchase:



Step-by-Step

Here's how to build the view above, using the Sample - Superstore data source provided with Tableau Desktop.

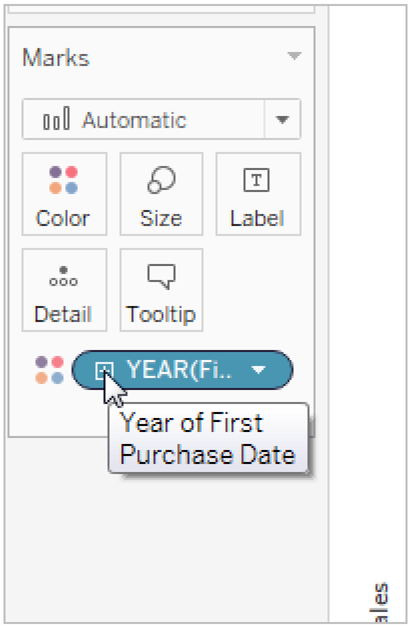
1. Create two calculated fields: a FIXED level of detail expression, and a date subtraction.

Name	Formula
First Purchase Date	{FIXED [Customer Name] : MIN([Order Date])}
Days Since First Purchase	DATETRUNC('day', [Order Date])-DATETRUNC('day', [First Purchase Date])

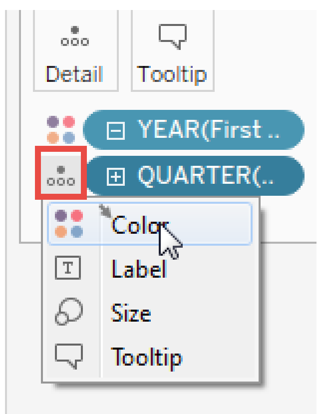
2. Drag **Days Since First Purchase** from the Measures area of the Data pane to the Dimensions area.

   Because this calculated field contains a subtraction operation, the result is a number, and so Tableau assigns it to the measure category. But you will be using it as a dimension.

3. Drag **Days Since First Purchase** to Columns.
4. Click **Days Since First Purchase** on Columns and choose **Continuous**.
5. Drag **Sales** to Rows.
6. Change the aggregation for **Sales** on Rows from SUM to AVG.
7. Add a quick table calculation to **Sales** on Rows: Running Total.
8. Drag **First Purchase Date** to Color.
9. Click the + in the **YEAR(First Purchase Date)** field on Color to add the next level down in the date hierarchy: **QUARTER(First Purchase Date)**.

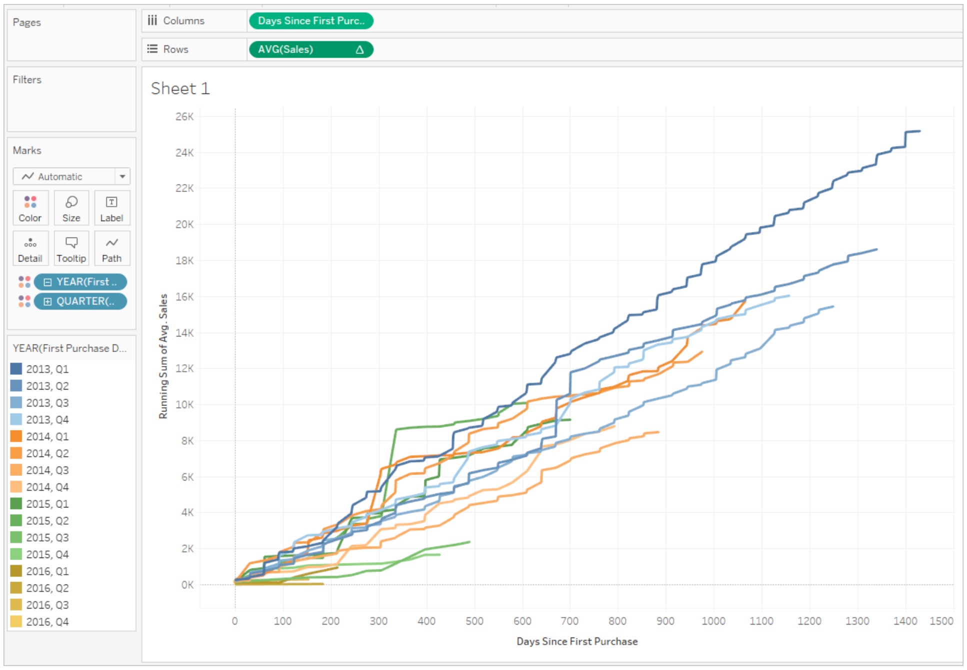


10. Tableau won't automatically put two fields on Color, but you can make it happen by clicking the icon to the left of the **QUARTER(First Purchase Date)** field and choosing Color:



11. Drag the Color legend over to the left side of the view, below the Marks card.

    Your view should now look like this:



The visualization yields insights as you interact with it. Select individual quarters in the color legend on the left to see how customers acquired in specific individual quarters continued to spend in subsequent quarters. Notice that customers acquired in the early going (2013) tend to spend at a higher rate, even allowing for the fact that they had more time to do so—thus, the blue lines (for 2013) are higher up on the x-axis than other lines. If you had used a standard date value on the y-axis, the view would have been easier to create, but the lines would not all start at the same coordinates and so it would be more difficult to compare acquisition rates.

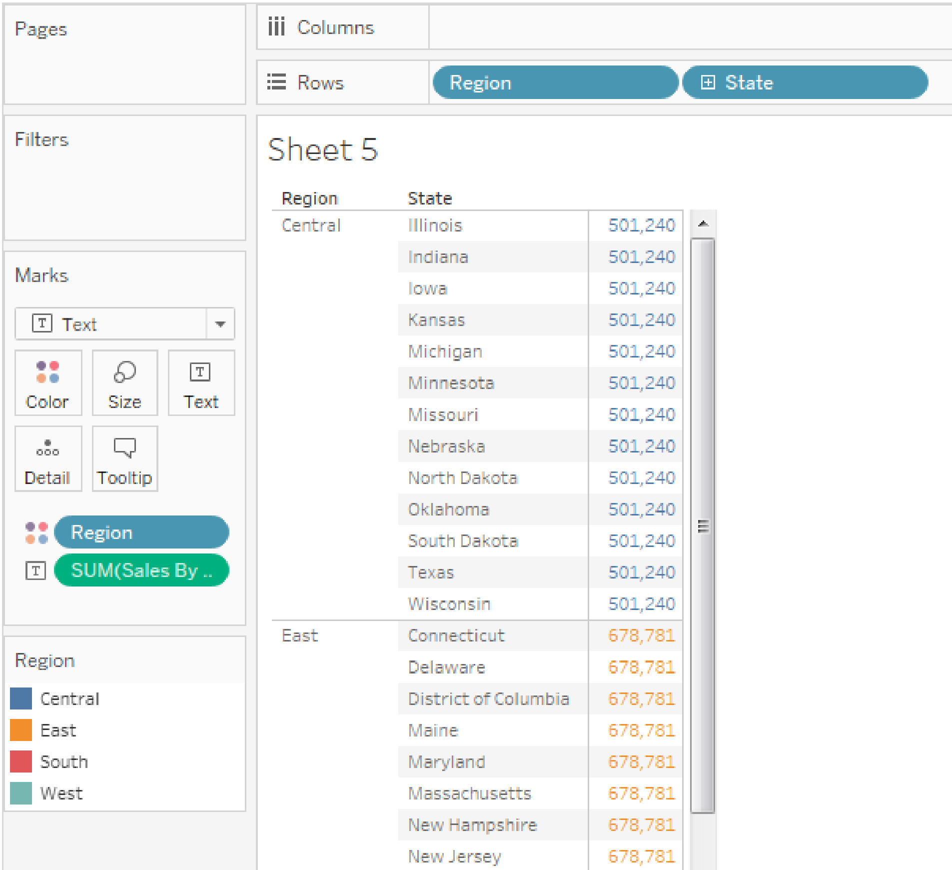
You can also drag either YEAR(First Purchase Date) or QUARTER(First Purchase Date) off color to see seasonal or annual trends more clearly.

Example 2

The following FIXED level of detail expression computes the sum of sales per region:

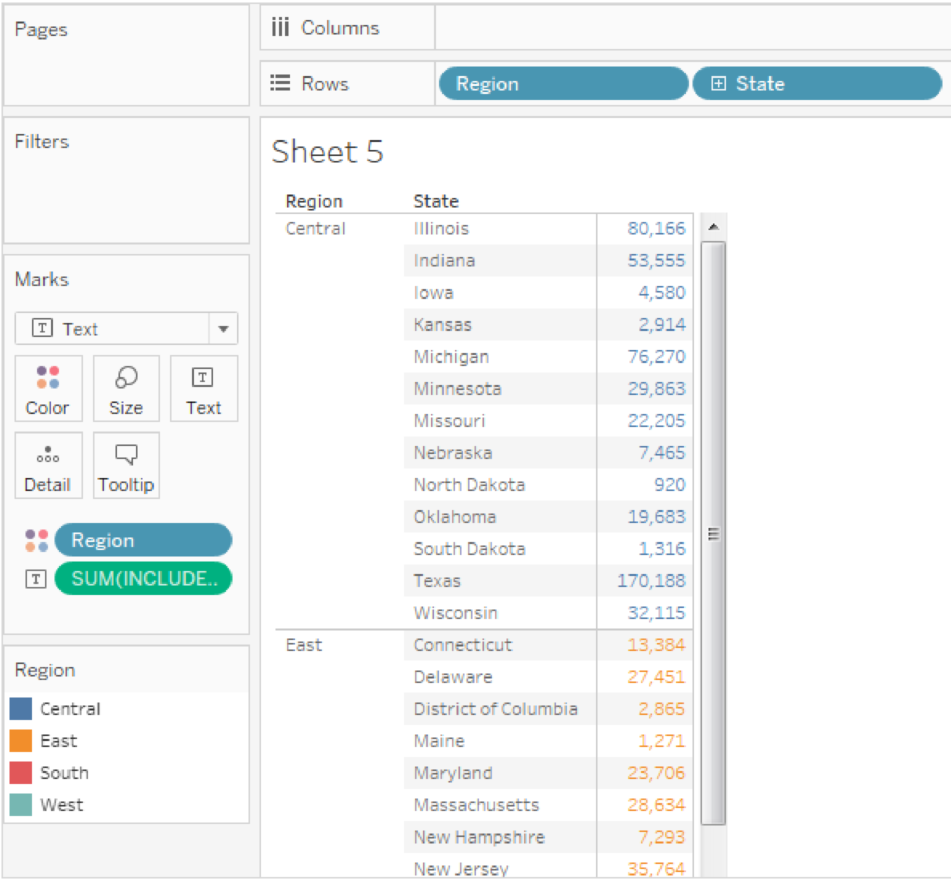
```
{FIXED [Region] : SUM([Sales])}
```

This level of detail expression, named [**Sales by Region**], is then placed on Text to show total sales per region:



The view level of detail is [**Region**] plus [**State**], but because FIXED level of detail expressions do not consider the view level of detail, the calculation only uses the [**Region**] dimension, and so the values for the individual states in each region are identical. See **Level of Detail Expressions and Aggregation** on page 2488 for an explanation of why this is so.

If the INCLUDE keyword had been used in the level of detail expression instead of FIXED, the values would be different for each state, because Tableau would add the dimension in the expression ([**Region**]) with any additional dimensions in the view ([**State**]) in determining values for the expression. The result would be:



INCLUDE Level of Detail Expressions

INCLUDE level of detail expressions compute values using the specified dimensions in addition to whatever dimensions are in the view.

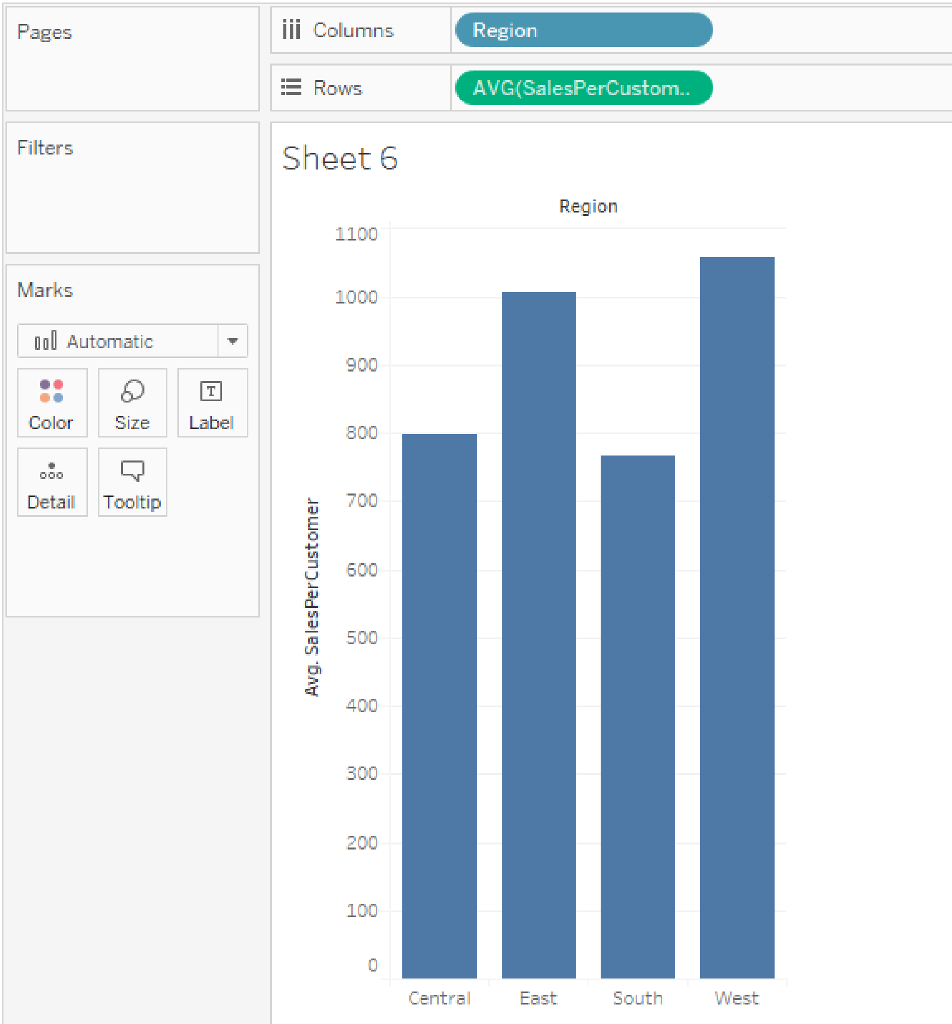
INCLUDE level of detail expressions can be useful when you want to calculate at a fine level of detail in the database and then re-aggregate and show at a coarser level of detail in your view. Fields based on INCLUDE level of detail expressions will change as you add or remove dimensions from the view.

Example 1

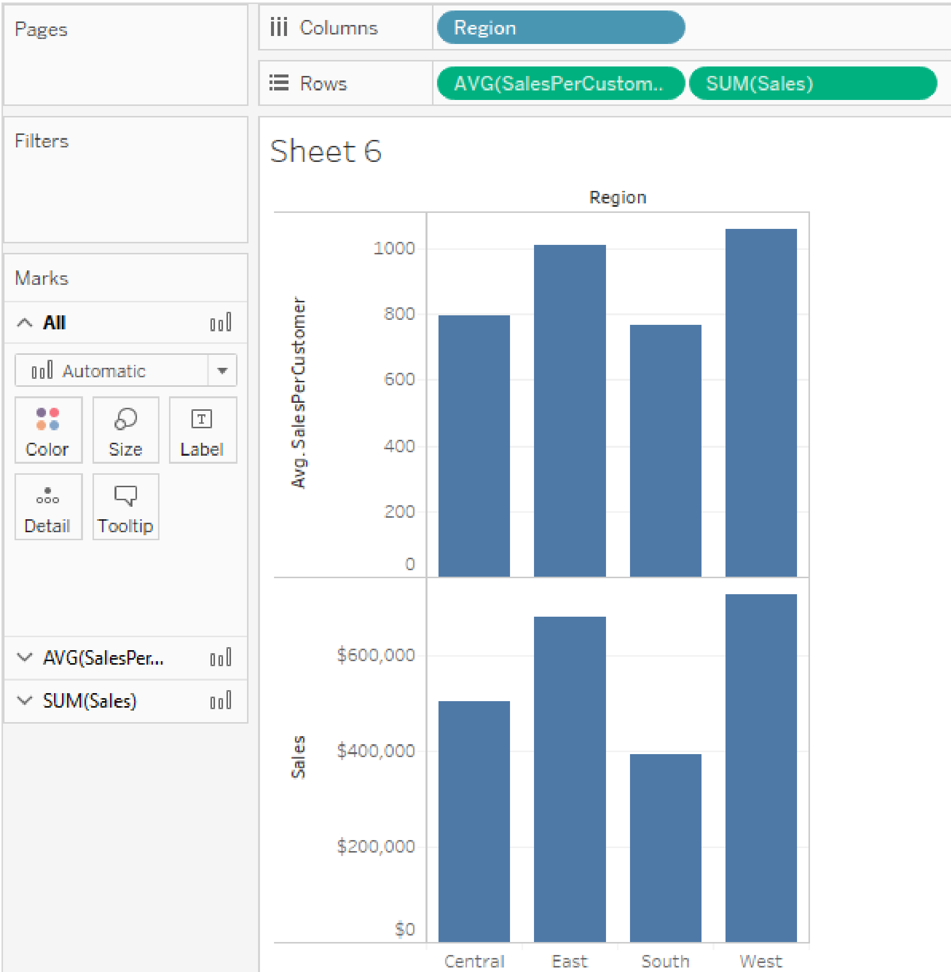
The following INCLUDE level of detail expression computes total sales per customer:

```
{ INCLUDE [Customer Name] : SUM([Sales]) }
```

When that calculation is placed on the **Rows** shelf, aggregated as AVG, and the [**Region**] dimension is placed on the **Columns** shelf, the view shows the average customer sales amount per region:



If the **[Sales]** measure is then dragged to the **Rows** shelf, the result illustrates the difference between the sum of sales (somewhere between $390K and $700K per region) and the average sales per customer (between $750 and $1100 per region):

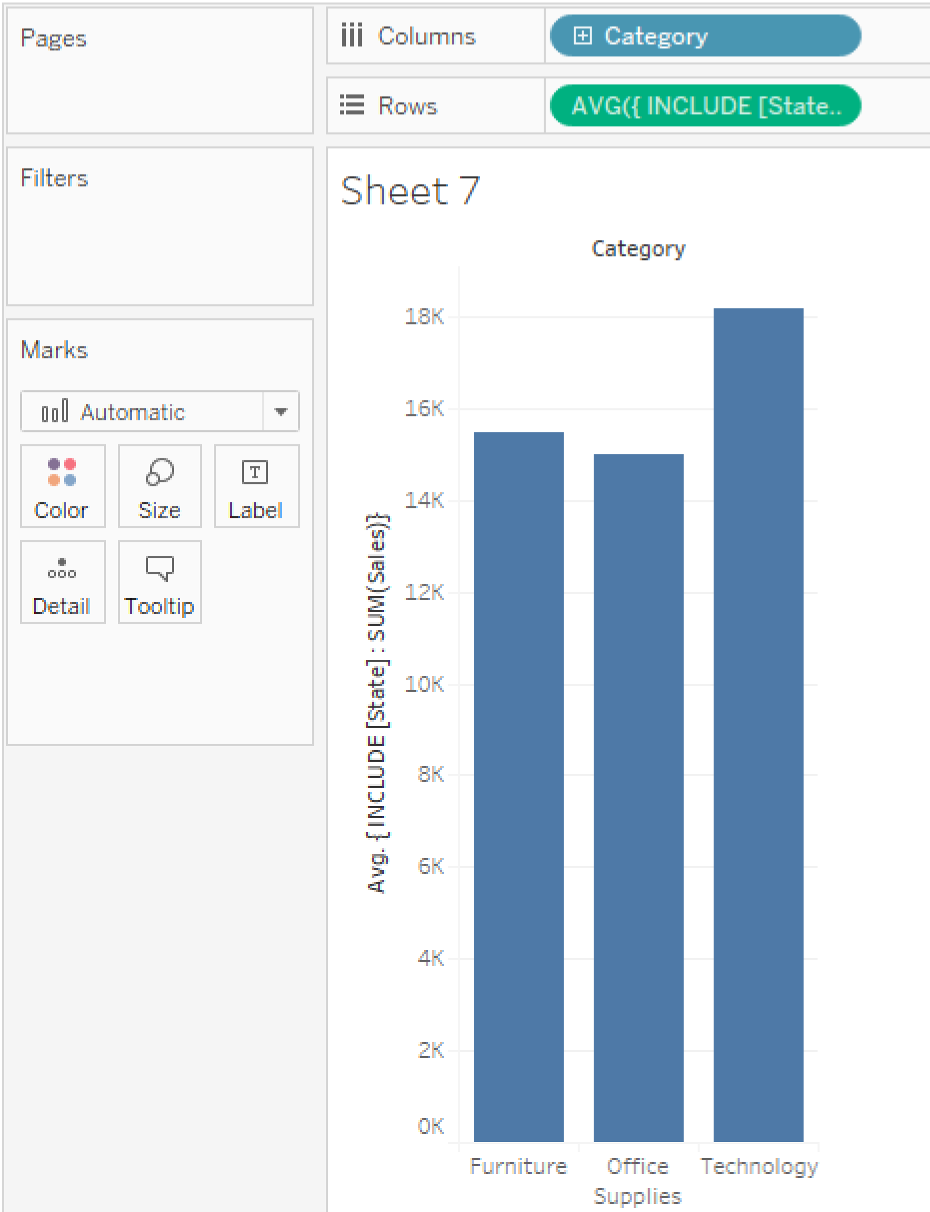


Example 2

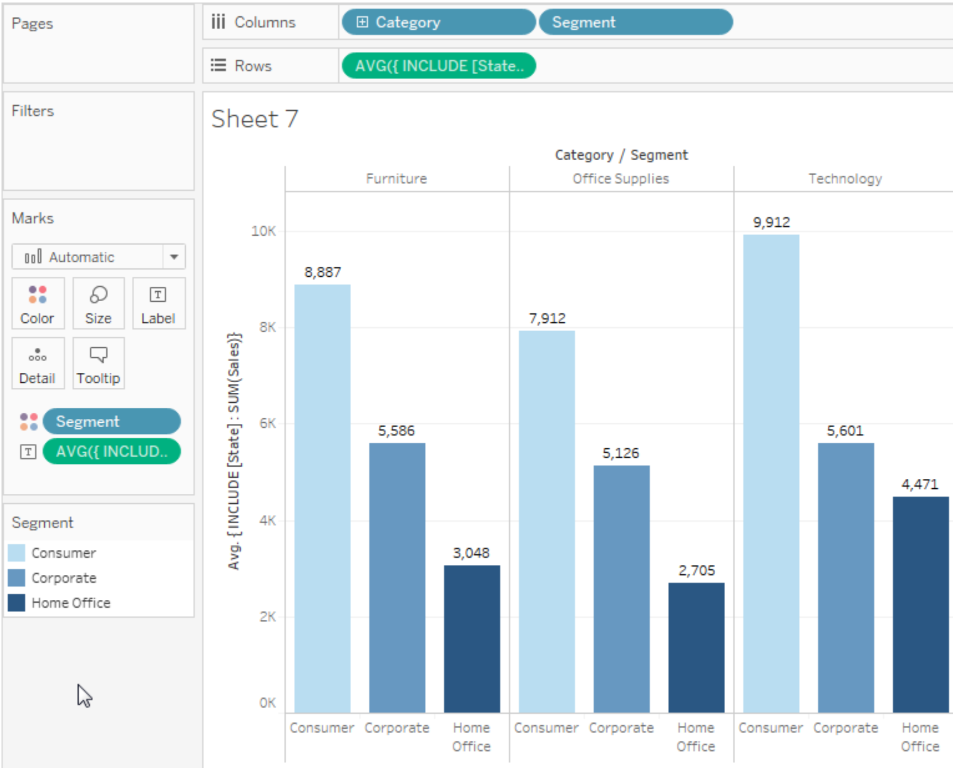
The following INCLUDE level of detail expression calculates sum of sales on a per-state basis:

```
{ INCLUDE [State] : SUM(Sales)}
```

Type the calculation directly on the **Rows** shelf, and then change the aggregation to AVG using the field's context menu. The resulting view averages the sum of sales by state across categories.



The view can be further enhanced by adding [**Segment**] to **Columns** and then copying the ad-hoc calculation from **Rows** to **Label** (using Ctrl+drag). In the image below we also dragged **Segment** to **Color** and then edited the colors for aesthetic effect. Now we can see how the average sum of sales per state varies across categories and segments.



EXCLUDE Level of Detail Expressions

For foundational concepts about level of detail (LOD) expressions, see **Create Level of Detail Expressions in Tableau** on page 2449 and **How Level of Detail Expressions Work in Tableau** on page 2465.

Level of detail expressions are used to modify or control the level of detail (granularity) of a calculation. Most fields and calculations are tied to the granularity of the view or of the data source. LOD expressions allow you to set the granularity of a calculation explicitly.

- FIXED LOD expressions establish a specific dimension or dimensions for the rest of the calculation, such as returning the maximum blood pressure for each patient instead of the overall maximum in the data set by fixing the maximum to Patient ID.
- INCLUDE LOD expressions ensure that a dimension that isn't necessarily present in the view is considered in the calculation, such as looking at the average blood pressure for each patient including their sex, displayed in a view that doesn't contain sex information.
- EXCLUDE LOD expressions remove some of the detail from the view for the purposes of the calculation, such as looking at the average blood pressure for patients on a certain medication without considering individual patients, even if the view is by patient.

Note: Whenever a FIXED LOD is used, it will give the same result regardless of the visualization. When an INCLUDE or EXCLUDE LOD is used, the value might change depending on the structure of the view. The INCLUDE keyword takes the context of the view and adds a dimension for the calculation to consider (and the EXCLUDE removes a dimension present in the view from the context of the calculation), so using the calculation in another context can offer different results.

Consider a view (A) built with patient ID, age, and sex, and another view (B) with patient ID, age, and *weight*.

- A FIXED LOD expression will have the same value in both cases because the calculation ignores what is in the view and cares only about the dimension in the LOD expression.
- An INCLUDE LOD that includes *weight* will have a different value in each view, because it adds a dimension to view A but not to view B, which already had weight.
- An EXCLUDE LOD that excludes *weight* will have a different value in each view, because it removes a dimension from view B but doesn't change the context of view A, which already didn't have weight.

EXCLUDE

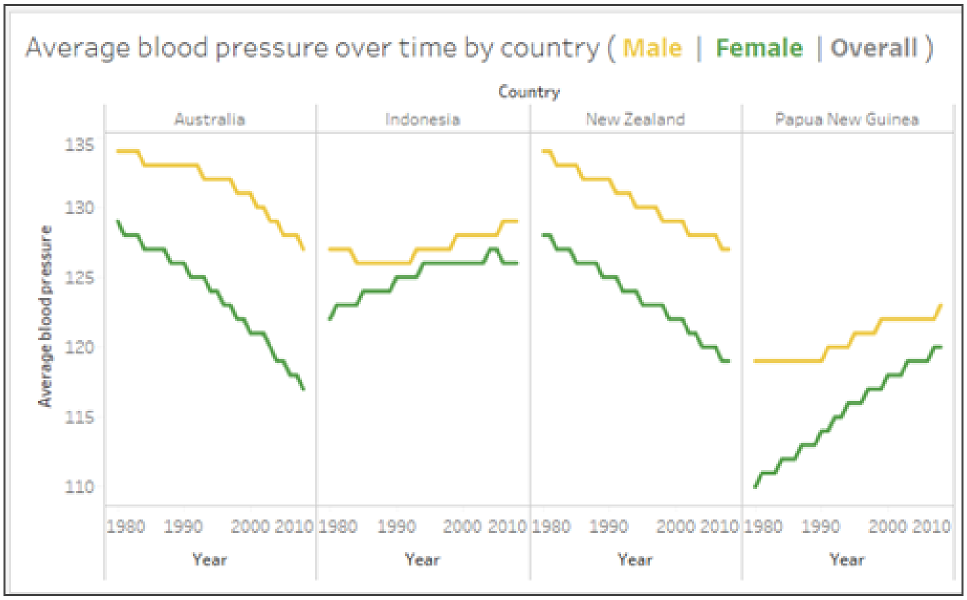
EXCLUDE level of detail expressions prevent the calculation from using one or more of the dimensions present in the view.

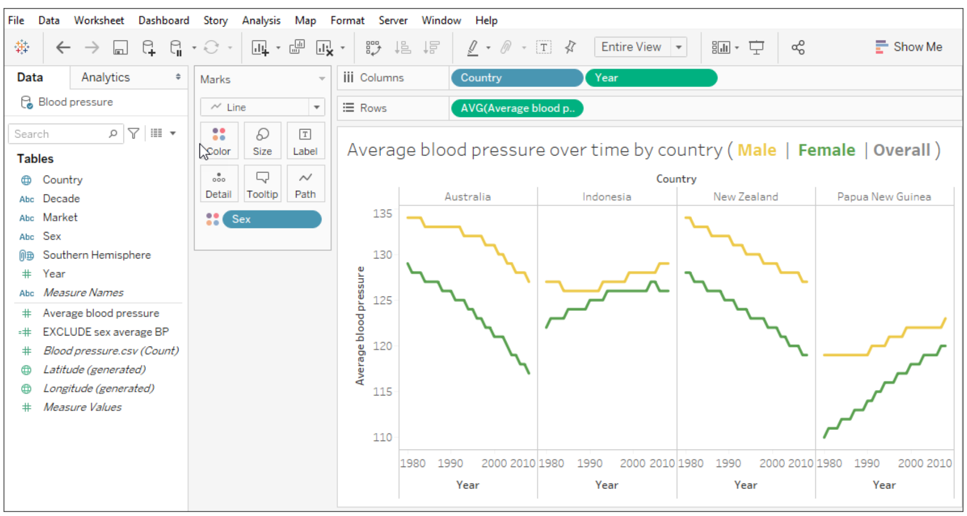
EXCLUDE level of detail expressions are useful for ‘percent of total’ or ‘difference from overall average’ scenarios. They are comparable to such features as Totals and Reference Lines.

EXCLUDE level of detail expression cannot be used in row-level expressions (where there are no dimensions to omit), but can be used to modify either a view level calculation or anything in between (that is, you can use an EXCLUDE calculation to remove dimension from some other level of detail expression).

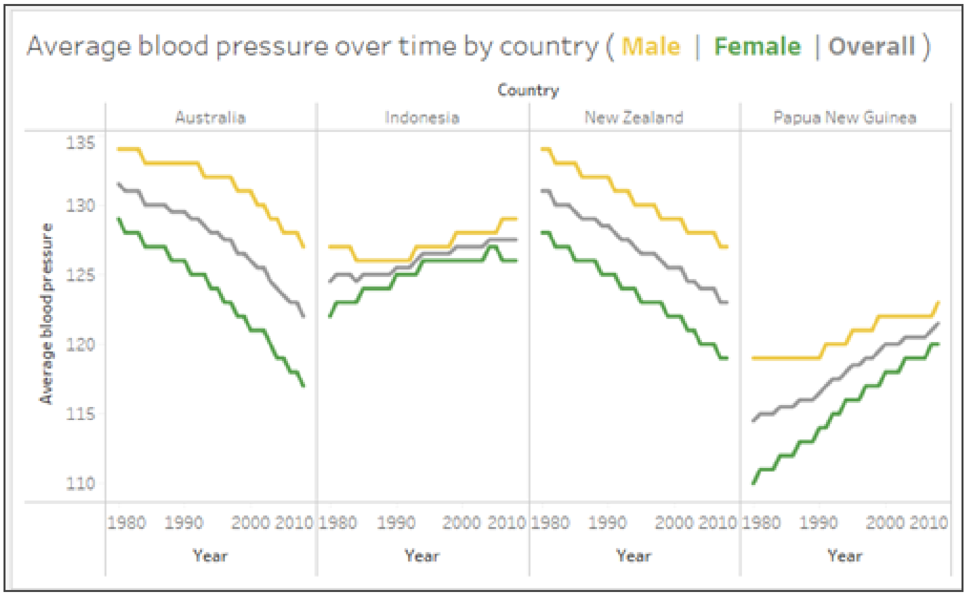
Example 1

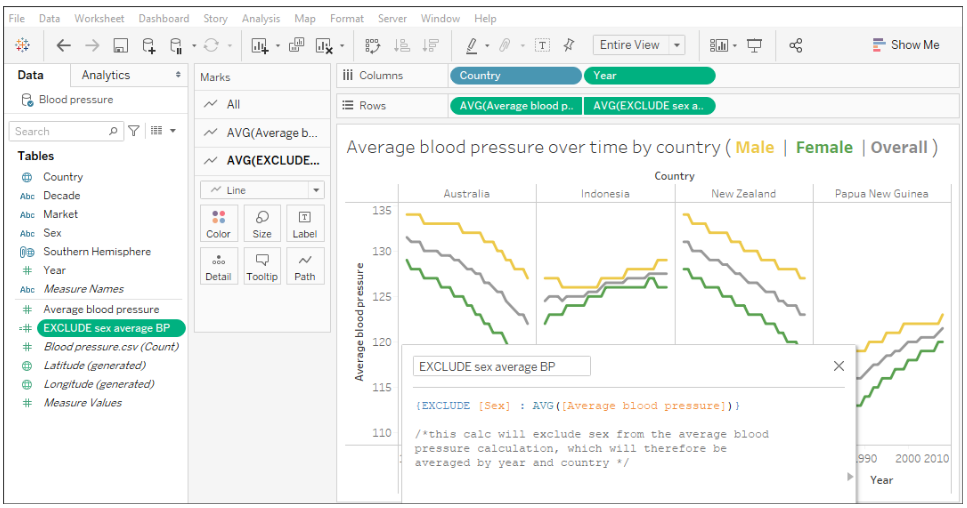
The image below shows the change in average blood pressure over time for four countries, broken out by sex.





To see the average blood pressure for each country over time but without being split by male and female, use an EXCLUDE level of detail expression `{EXCLUDE [Sex] : AVG [Average blood pressure]}`. This is plotted as the grey line in the viz.





Bringing an average line out from the Analytics pane wouldn't work, because it would simply be a horizontal line for the average blood pressure per country, ignoring sex and year.

> **Note**: This example takes the average of averages which is an incorrect analytical practice. It is done here purely for illustration purposes. No analysis should be made from this visualization.

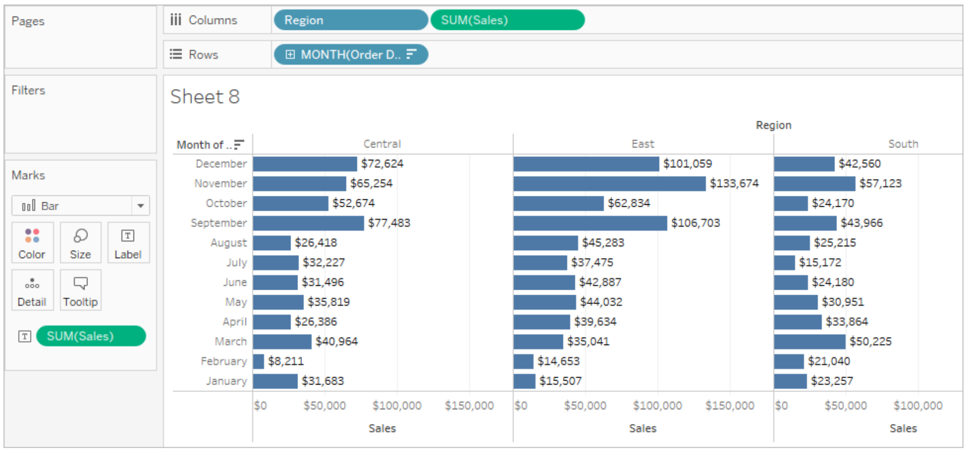
Example 2

The following level of detail expression excludes [**Region**] from a calculation of the sum of [**Sales**]:

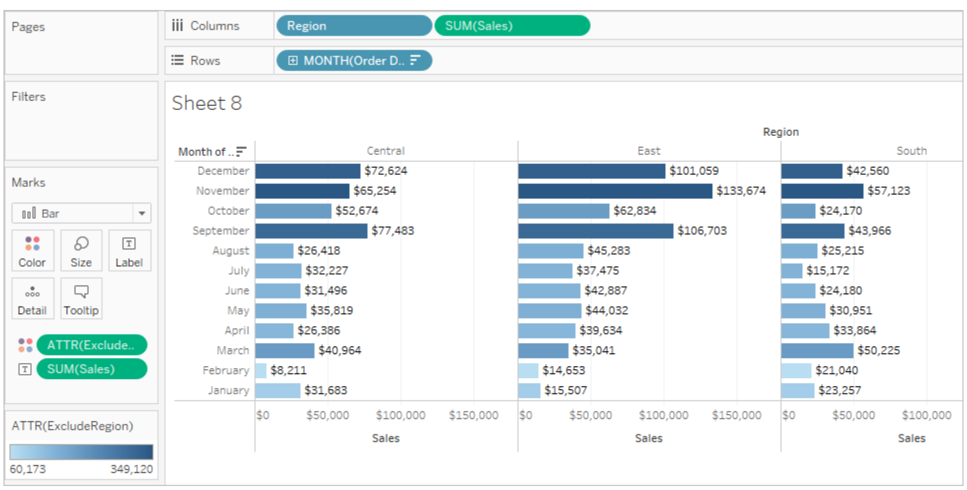
```
{EXCLUDE [Region]: SUM([Sales])}
```

The expression is saved as [**ExcludeRegion**].

To illustrate how this expression might be useful, first consider the following view, which breaks out the sum of sales by region and by month:



Dropping [**ExcludeRegion**] on Color shades the view to show total sales by month but without the regional component:



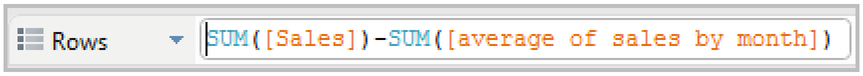
Example 3

The following EXCLUDE level of detail expression computes the average sales total per month and then excludes the month component:

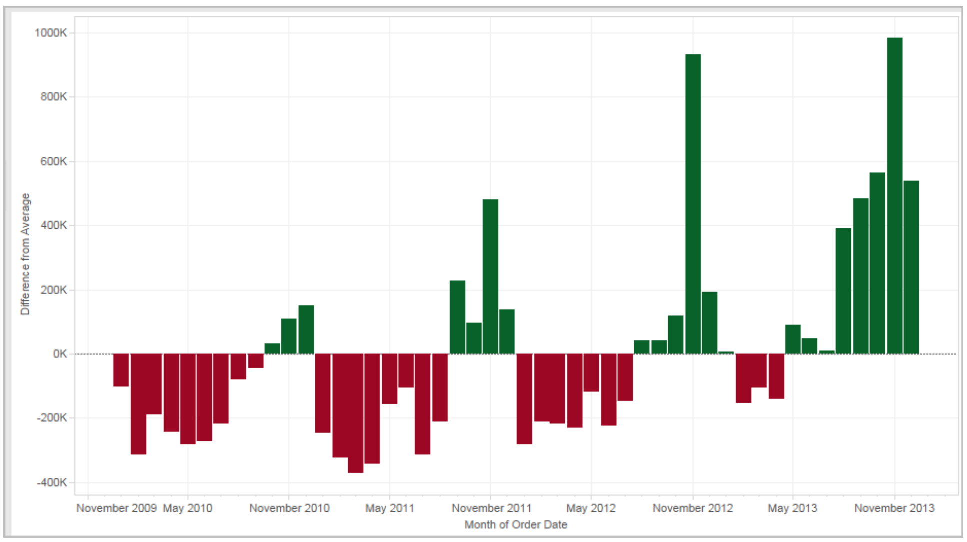
```
{EXCLUDE [Order Date (Month / Year)] : AVG({FIXED [Order Date
(Month / Year)] : SUM([Sales])})}
```

Notice that this is a nested level of detail expression—that is, a level of detail expression within another level of detail expression.

Saved as [**average of sales by month**], the calculation can then be subtracted from the sum of sales per month by means of an ad-hoc calculation on the **Rows** shelf:



With **Month([Order Date])** on the **Columns** shelf, this creates a view that shows the difference between actual sales per month over a four-year period and the average monthly sales for the entire four-year period:



Level of Detail Expressions and Aggregation

The level of detail of the view determines the number of marks in your view. When you add a level of detail expression to the view, Tableau must reconcile two levels of detail—the one in the view, and the one in your expression.

The behavior of a level of detail expression in the view varies depending on whether the expression's level of detail is coarser, finer, or the same as the level of detail in the view. What do we mean by “coarser” or “finer” in this case?

Level of Detail Expression is Coarser Than View Level of Detail

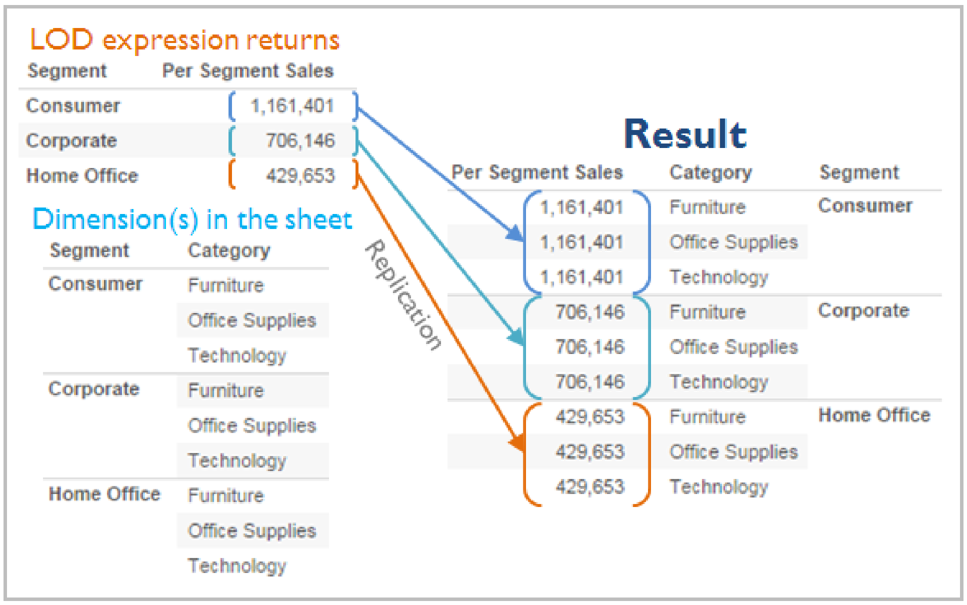
An expression has a coarser level of detail than the view when it references a subset of the dimensions in the view. For example, for a view that contained the dimensions [**Category**] and

[**Segment**], you could create a level of detail expression that uses only one of these dimensions:

```
{FIXED [Segment] : SUM([Sales])}
```

In this case, the expression has a coarser level of detail than the view. It bases its values on one dimension ([**Segment**]), whereas the view is basing its view on two dimensions ([**Segment**] and [**Category**]).

The result is that using the level of detail expression in the view causes certain values to be replicated—that is, to appear multiple times.



Replicated values are useful for comparing specific values against average values within a category. For example the following calculation subtracts average sales for a customer from the average sales overall:

```
[Sales] - {FIXED [Customer Name] : AVG([Sales])}
```

When values are being replicated, changing the aggregation for the relevant field in the view (for example, from AVG to SUM) will not change the result of the aggregation.

Level of Detail Expression is Finer Than View Level of Detail

An expression has a finer level of detail than the view when it references a superset of the dimensions in the view. When you use such an expression in the view, Tableau will aggregate results up to the view level. For example, the following level of detail expression references two dimensions:

```
{FIXED [Segment], [Category] : SUM([Sales])}
```

When this expression is used in a view that has only [Segment] as its level of detail, the values must be aggregated. Here's what you would see if you dragged that expression to a shelf:

```
AVG([{FIXED [Segment]], [Category]] : SUM([Sales]])}])
```

An aggregation—in this case, average—is automatically assigned by Tableau. You can change the aggregation as needed.

Adding a Level of Detail Expression to the View

Whether a level of detail expression is aggregated or replicated in the view is determined by the expression type (FIXED, INCLUDE, or EXCLUDE) and whether the expression's granularity is coarser or finer than the view's.

- INCLUDE level of detail expressions will have either the same level of detail as the view or a finer level of detail than the view. Therefore, values will never be replicated.
- FIXED level of detail expressions can have a finer level of detail than the view, a coarser level of detail, or the same level of detail. The need to aggregate the results of a FIXED level of detail depends on what dimensions are in the view.
- EXCLUDE level of detail expressions always cause replicated values to appear in the view. When calculations including EXCLUDE level of detail expressions are placed on a shelf, Tableau defaults to the ATTR aggregation (as opposed to SUM or AVG) to indicate that the expression is not actually being aggregated and that changing the aggregation will have no effect on the view.

Level of detail expressions are always automatically wrapped in an aggregate when they are added to a shelf in the view unless they're used as dimensions. So if you double-click on a shelf and type

```
{FIXED[Segment], [Category] : SUM([Sales])}
```

and then press Enter to commit the expression, what you now see on the shelf is

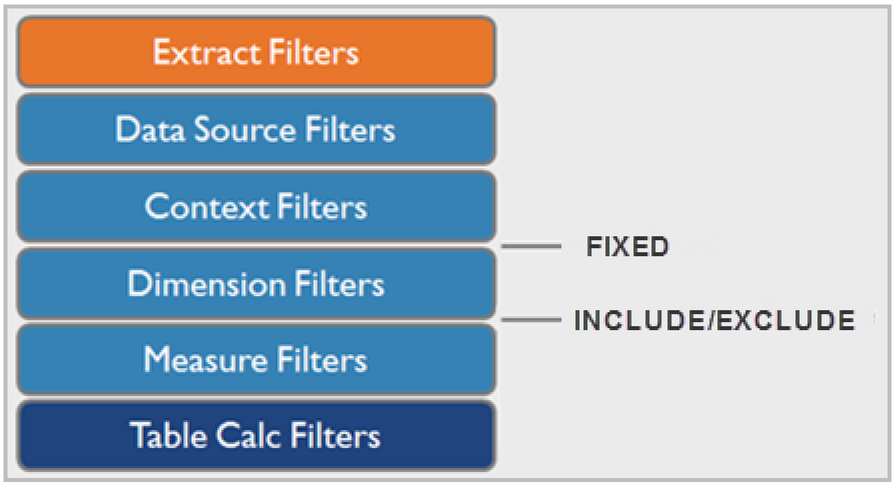
```
SUM({FIXED[Segment], [Category] : SUM([Sales])})
```

But if you double-click into the shelf to edit the expression, what you see in edit mode is the original expression.

If you wrap a level of detail expression in an aggregation when you create it, Tableau will use the aggregation you specified rather than assigning one when any calculation including that expression is placed on a shelf. When no aggregation is needed (because the expression's level of detail is coarser than the view's), the aggregation you specified is still shown when the expression is on a shelf, but it is ignored.

Filters and Level of Detail Expressions

There are several different kinds of filters in Tableau and they get executed in the following order from top to bottom.



The text on the right shows where level of detail expressions are evaluated in this sequence.

Extract Filters (in orange) are only relevant if you're creating a Tableau Extract from a data source. Table calculations filters (dark blue) are applied after calculations are executed and therefore hide marks without filtering out the underlying data used in the calculations.

If you're familiar with SQL, you can think of measure filters as equivalent to the HAVING clause in a query, and dimension filters as equivalent to the WHERE clause.

FIXED calculations are applied before dimension filters, so unless you promote the fields on your Filter shelf to **Improve View Performance with Context Filters** on page 1439, they will be ignored. For example, consider if you have the following calculation on one shelf in a view, along with [**State**] on a different shelf:

```
SUM([Sales]) / ATTR({FIXED : SUM([Sales])})
```

This calculation will give you the ratio of a state's sales to total sales.

If you then put [**State**] on the Filters shelf to hide some of the states, the filter will affect only the numerator in the calculation. Since the denominator is a FIXED level of detail expression, it will still divide the sales for the states still in the view against the total sales for all states—including the ones that have been filtered out of the view.

INCLUDE and EXCLUDE level of detail expressions are considered after Dimension filters. So if you want filters to apply to your FIXED level of detail expression but don't want to use Context Filters, consider rewriting them as INCLUDE or EXCLUDE expressions.

Data Source Constraints for Level of Detail Expressions

For some data sources, only more recent versions support level of detail expressions. Some data sources do not support level of detail expressions at all.

In addition, some data sources have complexity limits. Tableau will not disable calculations for these databases, but query errors are a possibility if calculations become too complex.

Data Source	Support
Amazon EMR Hadoop Hive	Supported for Hive 0.13 and later.
Aster Database	Supported for version 4.5 and later.
Cloudera Hadoop	Supported for Hive 0.13 and later.
Cloudera Impala	Supported for Impala 1.2.2 and later.
Cubes (multidimensional data sources)	Not supported.
DataStax Enterprise	Not supported.
Firebird	Supported for version 2.0 and later.
Generic ODBC	Limited. Depends on the specific data source.
Google Big Query	Supported for standard SQL, not supported for legacy SQL.
Hortonworks Hadoop Hive	Supported for Hive 0.13 and later. On version 1.1 of HIVE level of detail expressions that produce cross joins are not reliable.

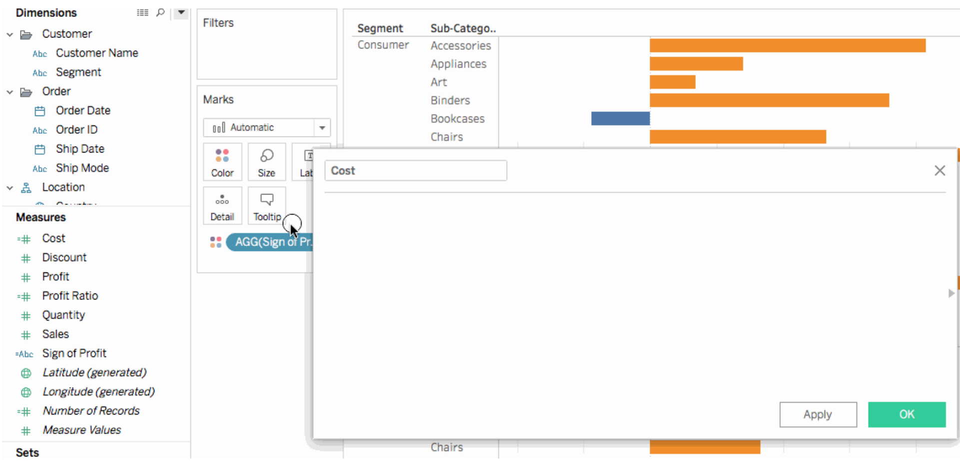
	Cross join occur when there is no explicit field to join on. For example, for a level of detail expression `{fixed [Product Type] : sum (sales)}` when the view only contains one dimension [**Ship Mode**], Tableau creates a cross-join. A cross join produces rows which combine each row from the first table with each row from the second table.
IBM DB2	Supported for version 8.1 and later.
MarkLogic	Supported for version 7.0 and later.
Microsoft Access	Not supported.
Microsoft Jet-based connections (legacy connectors for Microsoft Excel, Microsoft Access, and text)	Not supported.
Microsoft SQL Server	SQL Server 2005 and later.
Mongo DB	Not supported.
IBM PDA (Netezza)	Supported version 7.0 and later.
Oracle	Supported version 9i and later.
Actian Matrix (ParAccel)	Supported version 3.1 and later.
Pivotal Greenplum	Supported for version 3.1 and later.
PostgreSQL	Supported version 7 and later.
SAP Sybase IQ	Supported version 15.1 and later.
Splunk	Not supported.
Vertica	Supported for version 6.1 and later.

Tips for Working with Calculated Fields in Tableau

To help you become more efficient with creating and editing calculated fields in Tableau, this article lists several tips for working in the calculation editor.

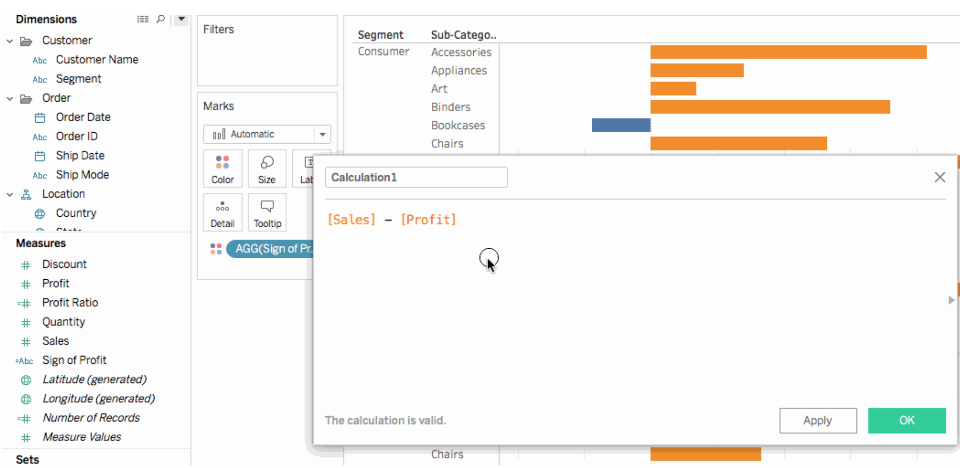
Drag and drop fields into the calculation editor

When creating fields in the calculation editor, you can drag existing fields from the Data pane into the editor at any time.



Drag and drop formulas from the calculation editor to the Data pane

When typing a calculation in the calculation editor, you can highlight all or part of the formula and drag it to the Data pane to create a new calculated field. You can then rename the field by typing a name. For more information, see **Ad-Hoc Calculations** on page 2497.

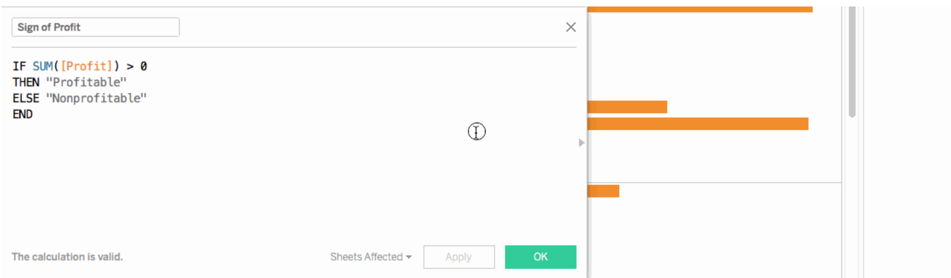


Use the functions reference in the calculation editor

When typing a calculation in the calculation editor, you can use the functions reference to browse all the functions available in Tableau.

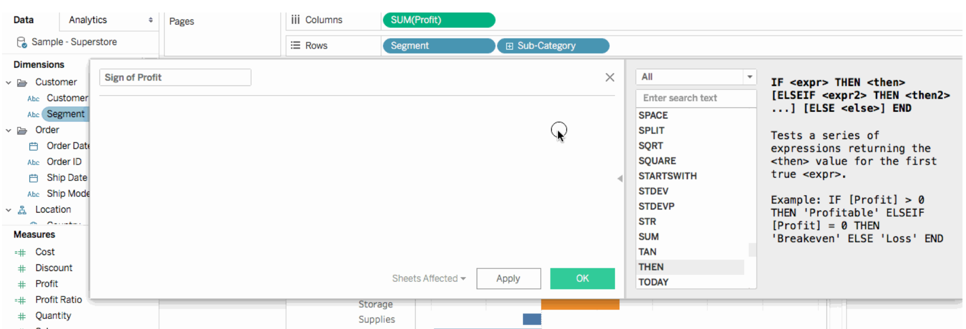
To open the functions reference:

- In the calculation editor, click the triangle icon on the right-side of the editor.



To add a function from the reference to a formula:

- In the function reference, double-click a function.



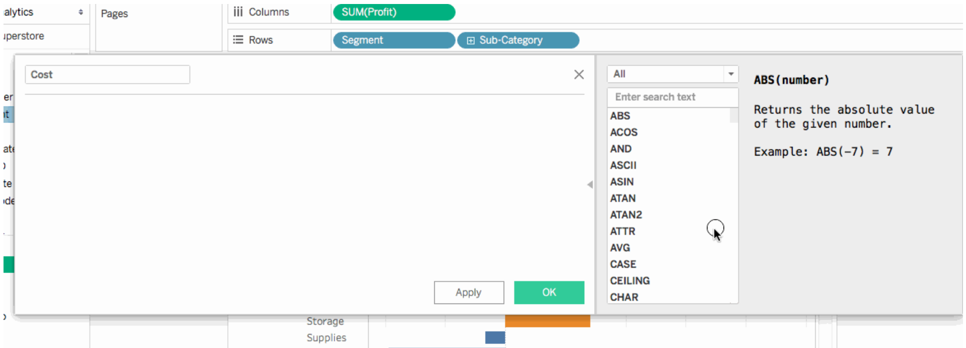
Take advantage of auto-complete for formulas

As you type a formula in the calculation editor, Tableau suggests options to complete items in your formula. Tableau suggests functions, fields in your data source, parameters, sets, and bins that begin with or contain the string you type. The list of suggestions update as you type.

To add an item from auto-complete to a formula:

- Press Enter on your keyboard to select the highlighted suggestion.

  Note: You can use the up and down arrows on your keyboard to move between items in the auto-complete list.

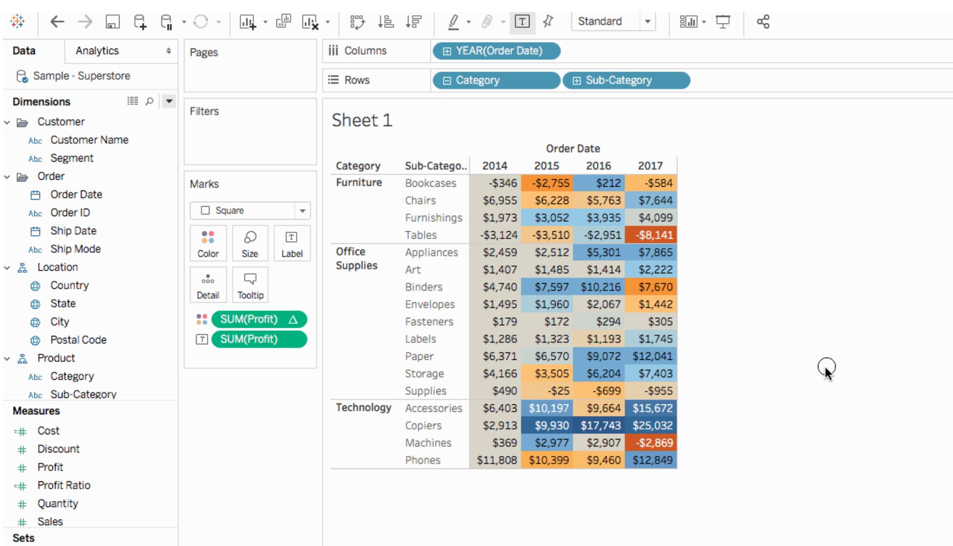


Drag table calculations into the calculation editor to edit them

When you create a table calculation, you can drag it into the calculation editor to review or make changes to the formula.

To edit a table calculation in the calculation editor:

1. Click the **Data** pane drop-down and select **Create Calculated Field**.
2. From the worksheet, drag the table calculation into the calculation editor.
3. When finished, click **OK**.



Resize text in the calculation editor

You can adjust the size of the text in the calculation editor as you create or edit calculations.

To increase text size in the calculation editor:

- Press the `CTRL` and + keys on your keyboard (`Command` + on a Mac)

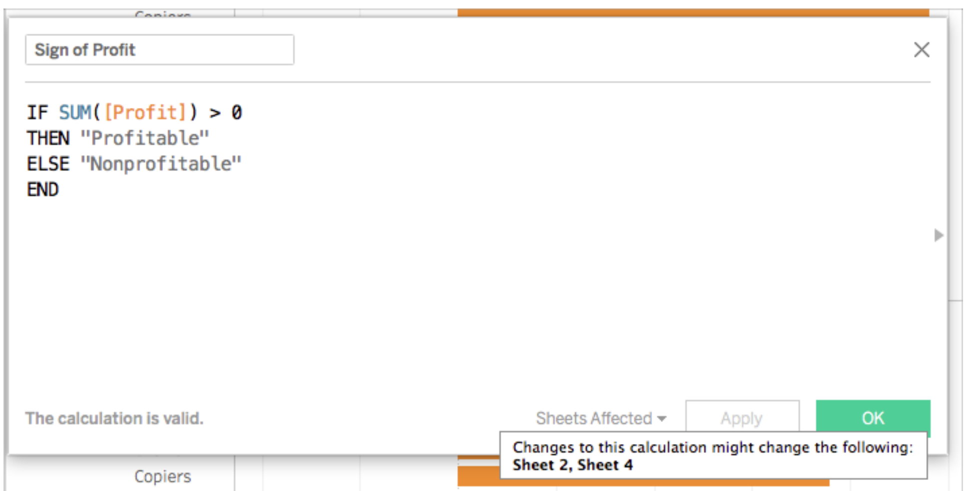
To decrease text size in the calculation editor:

- Press the `CTRL` and – keys on your keyboard (`Command` – on a Mac).

> **Note**: Text size persists until you close the editor. The next time you open the editor, text is at the default size.

See which sheets are using a calculated field

As you edit a calculated field, you can click **Sheets Affected** to see which other sheets are using the field. These sheets will also be updated when you commit your changes.

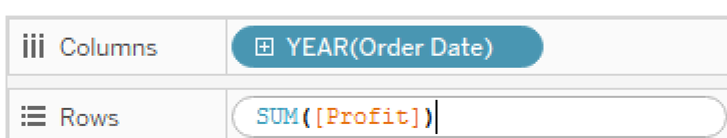


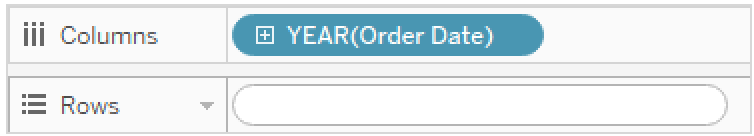
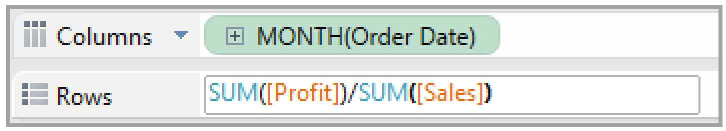
Ad-Hoc Calculations

Ad-hoc calculations are calculations that you can create and update as you work with a field on a shelf in the view. Ad-hoc calculations are also known as type-in or in-line calculations.

Create an Ad-Hoc Calculation

Double-click on an existing field to start editing.



You can also double-click on an empty shelf or on an empty part of a shelf to create a new calculation.	
Type to update the expression, or drag new fields into the expression from the **Data** pane or elsewhere in the view.	
Press Enter, Tab, or click outside the expression to commit the expression, close the calculation, and update the view. Press Ctrl+Enter (or Command+Enter on a Mac) to commit the expression and update the view without closing the calculation.	

Ad-hoc calculations are supported on the **Rows**, **Columns**, **Marks**, and **Measure Values** shelves; they are not supported on the **Filters** or **Pages** shelves.

If you are working in Tableau Desktop, errors in ad-hoc calculations are underlined in red. Hover over the error to see a suggestion for resolving it.

> **Note**: Ad-hoc calculations are not named, but are saved when you close the workbook. If you want to save an ad-hoc calculation for use in other workbook sheets, copy it to the **Data** pane. You will be prompted to name the calculation. {On server it just deposits it w/o rename prompt.} Once you name an ad-hoc calculation, it is the same as a calculation you created with the calculation editor, and is available on other sheets in the workbook. See **Get Started with Calculations in Tableau** on page 2095

Multi-Line Ad-Hoc Calculations

As you're typing an ad-hoc calculation, you can press Shift+Enter to start a new line. However, only the current line is ever visible in an ad-hoc calculation, so this can be confusing for anyone viewing or editing the calculation who has no way of knowing that it contains multiple lines.

The first line of a multi-line ad-hoc calculation can be a comment that serves as a title for the calculation. This is the only line that is visible on the shelf after the calculation is committed:

```
//City and State

[City] + ', ' + [State]
```

Ad-Hoc Calculations and Aggregation

If Tableau determines that the expression you enter is a measure (that is, returns a number), it automatically adds an aggregation to the expression when you commit the expression. For example, if you type `DATEDIFF('day',[Ship Date],[Order Date])` in an ad-hoc calculation and then press Enter, what you will see is the following:

```
SUM(DATEDIFF('day',[Ship Date],[Order Date]))
```

If you use a field that is already an aggregated field (for example, SUM([Profit])) in an ad-hoc calculation, the result is an aggregate calculation. For example, when you commit an ad-hoc calculation `SUM([Profit])/SUM([Sales])`, the result is:

```
AGG(SUM([Profit])/SUM([Sales]))
```

For more information on aggregate calculations, see **Aggregate Functions in Tableau** on page 2192.

Ad-Hoc Calculations for Insight and Experimentation

Typically you create ad-hoc calculations on-the-fly to do things like

- Test a hunch
- Try a what-if scenario
- Debug a complex calculation

Managing Ad-Hoc Calculations

Use the following keystrokes to manage ad-hoc calculations.

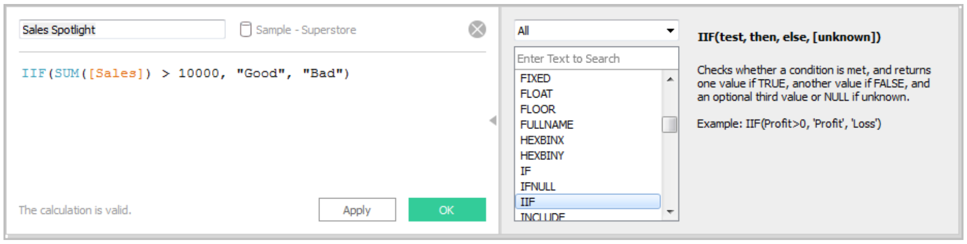
- Double-click in an existing field on the **Rows**, **Columns**, **Marks**, or **Measure Values** shelf to open it as an ad-hoc calculation.
- Double-click elsewhere on any of these shelves to create a new ad-hoc calculation from scratch.
- Press Esc to cancel an ad-hoc calculation.
- Press Enter to commit an ad-hoc calculation, which updates the view and closes the ad-hoc calculation. Press Ctrl+Enter to commit the change and update the view without closing the ad-hoc calculation.
- If you are working in Tableau Desktop, and there is a field to the right of the current ad-hoc calculation on the same shelf, pressing Tab opens that field as an ad-hoc calculation. If there is no field to the right of the current ad-hoc calculation on the shelf, pressing Tab opens a new ad-hoc calculation. Shift+Tab has the same functionality, except it moves to the left.
- When you double-click a named field on a shelf to edit it, you are not changing the original named field.

In addition, the following rules govern the use of ad-hoc calculations.

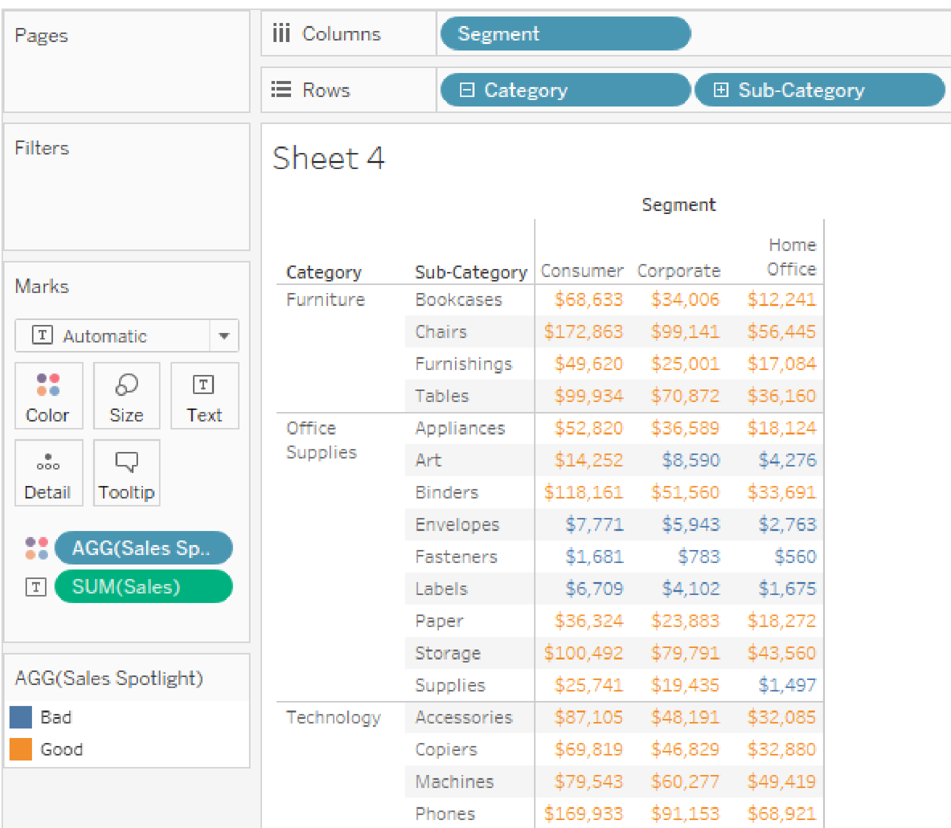
- Only one ad-hoc calculation can be open at a time.
- If a field that has a geographic role or a fiscal year setting associated with it is added to an ad-hoc calculation, the ad-hoc calculation inherits that role or setting.
- The right-click context menu for an ad-hoc calculation supports the same options that would be available for any other field in the view of the same type, including the ability to change aggregation or format.
- Ad-hoc calculations are not available when you create groups, sets, hierarchies, or parameters.
- Ad-hoc calculations are valid for creating trend lines, forecasts, and reference lines, bands, and distributions.

Example - Spotlighting Using Calculations

Spotlighting is a technique for showing discrete thresholds based on the values of a measure. For instance, you might want to color-code sales so that those over 10,000 appear green and those below 10,000 appear red. A spotlighting calculation is just a special case of a calculation that results in a discrete measure. A discrete measure is a calculation that is a dependent variable (and therefore a measure), but which results in a discrete result (as opposed to a continuous result). Thus the name discrete measure. Here is an example:



The formula in this example defines a discrete measure named **Sales Spotlight**. Discrete measures always appear with a blue **abc** icon in the **Data** pane. **Sales Spotlight** is classified as a measure in Tableau because it is a function of another measure; it is discrete because it produces discrete values (“Good” and “Bad”). Here is an example of this categorical measure in use:



Here, **Sales Spotlight** is on Color in the Marks card. It appears with the **AGG** prefix because it is an aggregate calculation. Values above 10,000 and below 10,000 are assigned different colors.

Spot Trends

Reference Lines, Bands, Distributions, and Boxes

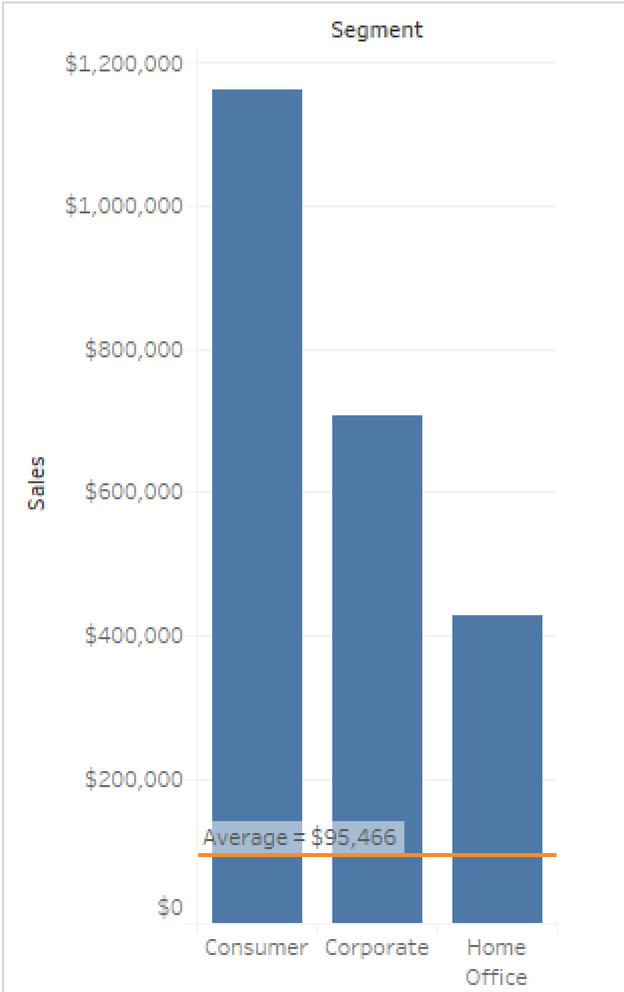
You can add a reference line, band, distribution, or box plot to identify a specific value, region, or range on a continuous axis in a Tableau view. For example, if you are analyzing the monthly sales for several products, you can include a reference line at the average sales mark so you can see how each product performed against the average.

Tableau lets you add as many reference lines, bands, distributions, and box plots to a view as you require.

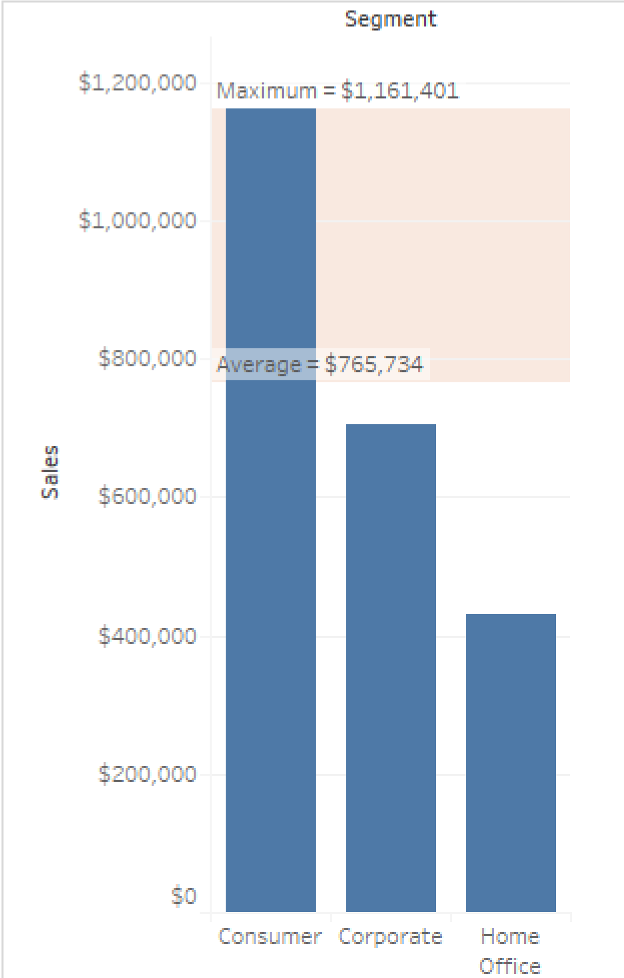
Types of Reference Lines, Bands, Distributions, and Boxes

You can add reference lines, bands, distributions, or (in Tableau Desktop but not on the web) box plots to any continuous axis in the view.

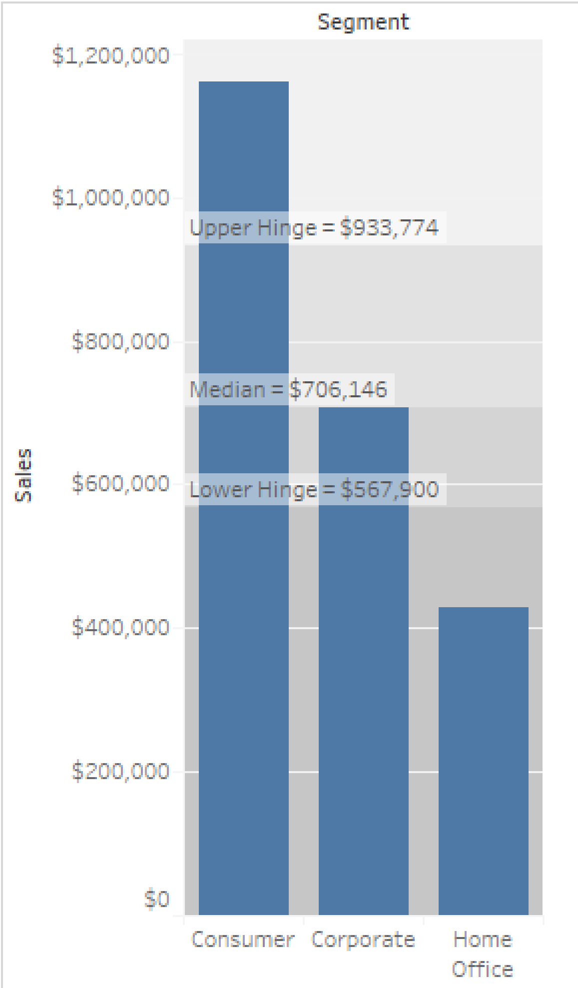
- **Reference Lines** - You can add a reference line at a constant or computed value on the axis. Computed values can be based on a specified field. You can also include confidence intervals with a reference line.



- **Reference Bands** - Reference bands shade an area behind the marks in the view between two constant or computed values on the axis.

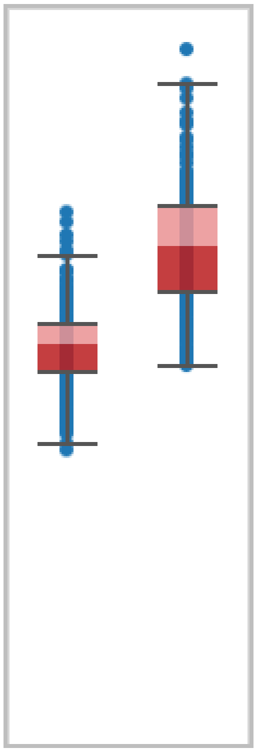


- **Reference Distributions** - Reference distributions add a gradient of shading to indicate the distribution of values along the axis. Distribution can be defined by percentages, percentiles, quantiles (as in the following image), or standard deviation.



Reference distributions can also be used to create bullet charts. See **Add a Bullet Graph** on page 2520 later in this article for specifics.

- **Box Plots** - Box plots (also known as box and whisker charts) are a standardized graphic for describing the distribution of values along an axis. Box plots show quartiles (also known as hinges) and whiskers. Tableau provides different box plot styles, and allows you to configure the location of the whiskers and other details.



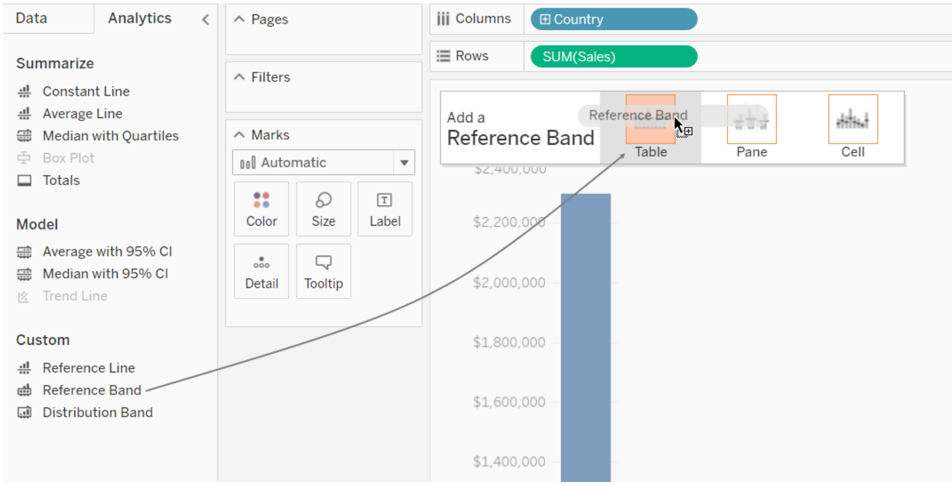
Add a Reference Line

You can add a reference line to any continuous axis in the view.

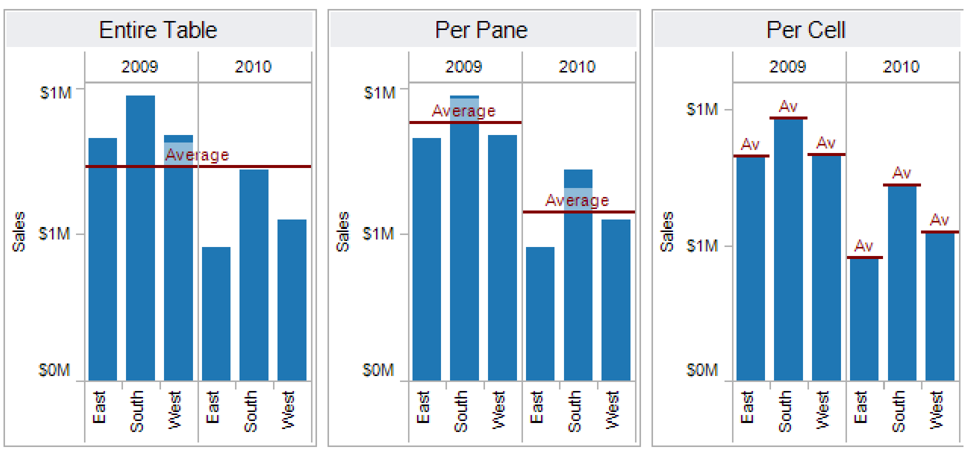
To add a reference line:

1. Drag **Reference Line** from the **Analytics** pane into the view. Tableau shows the possible destinations. The range of choices varies depending on the type of item and the current view.

   In a simple case, the drop target area offers three options:



The view above is from a web editing session. In Tableau Desktop, the process is the same but the user interface looks a bit different. The terms **Table**, **Pane** and **Cell** define the scope for the item:

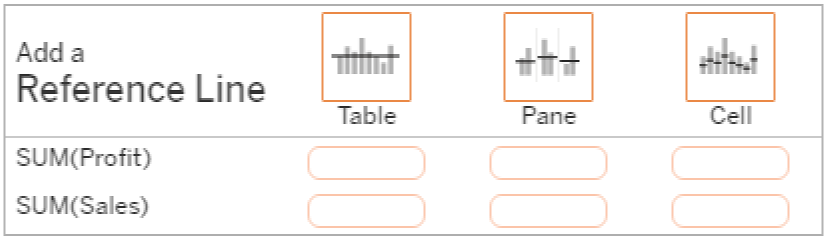


Adds a reference line to the entire table across all panes.

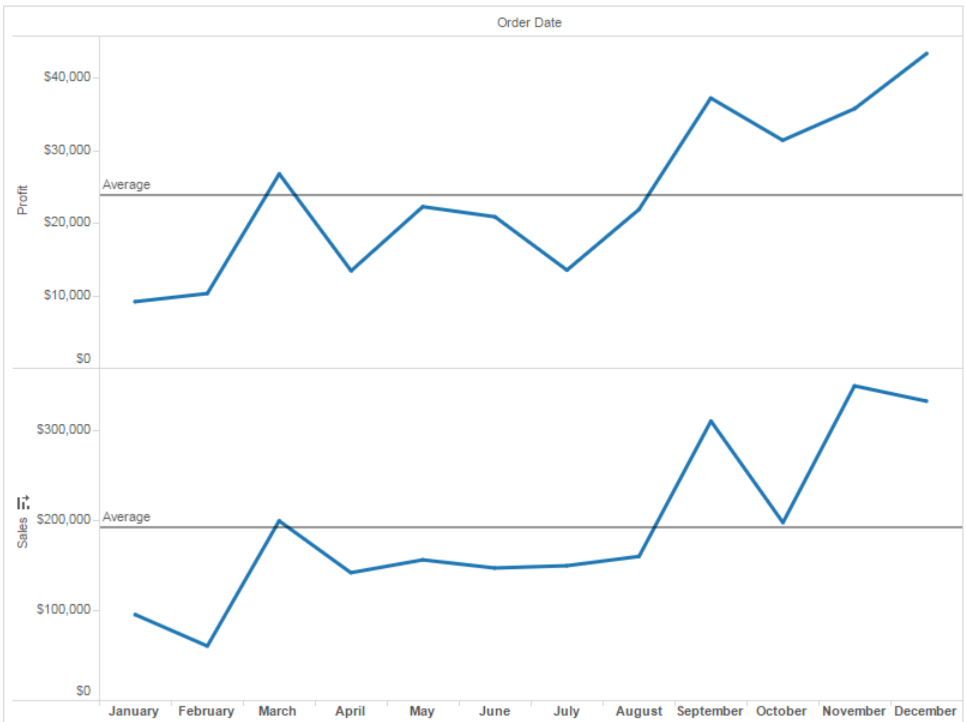
Adds a reference line on a per pane basis. Computed reference lines are recalculated for each pane in the view.

Adds a reference line within each cell. Computed reference lines are recalculated for each cell in the view.

For a more complicated view—for example, if the view contains a line chart with multiple or dual axes—Tableau shows you an expanded drop target area:



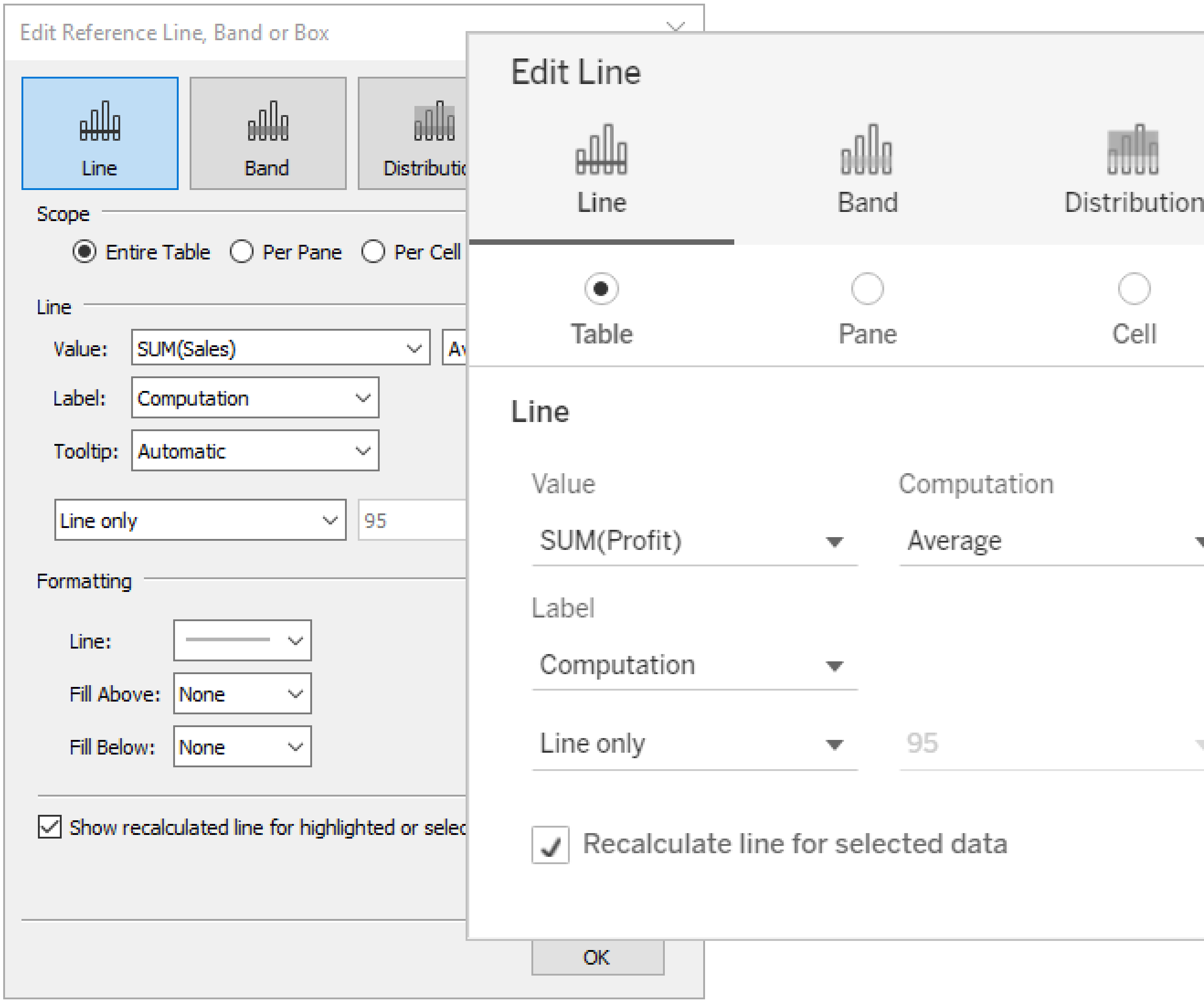
If you drop the item in one of the three larger boxes in the header (for example, the **Table** box), a separate reference line is added for each continuous field in the view:



But if you drop the item in any of the lower boxes that are aligned with a specific continuous field, the line is added on the corresponding axis, with the specified scope.

When you drop the line in the target area, Tableau displays a dialog box:

Tableau Desktop version **Web version**



2. The **Line** option is already selected at the top of the dialog box.
3. Select a continuous field from the **Value** field to use as the basis for your reference line. You can also select a parameter.

   You cannot select a continuous field that isn't currently in the view as the basis for your reference line. If you want to use such a continuous field, do the following:

   a. Drag the continuous field from the Data pane to the Details target on the Marks card.
   b. Change the continuous field's aggregation if necessary.

This will not change the view, but it will allow you to use that continuous field as the basis for your reference band.

c. Click on the reference line in the view and choose **Edit** to re-open the **Edit Line** dialog box.

4. Select an aggregation. The aggregations that are displayed depend on the continuous field you select:

   - **Total** - places a line at the aggregate of all the values in either the cell, pane, or the entire view. This option is particularly useful when computing a weighted average rather than an average of averages. It is also useful when working with a calculation with a custom aggregation. The total is computed using the underlying data and behaves the same as selecting one of the totals option the Analysis menu.
   - **Sum** - places a line at the SUM of all the values in either the cell, pane, or entire view.
   - **Constant**- places a line at the specified value on the axis.
   - **Minimum** - places a line at the minimum value.
   - **Maximum** - places a line at the maximum value.
   - **Average** - places a line at the average value along the axis.
   - **Median**- places a line at the median value.

5. Select how you want to label the line:

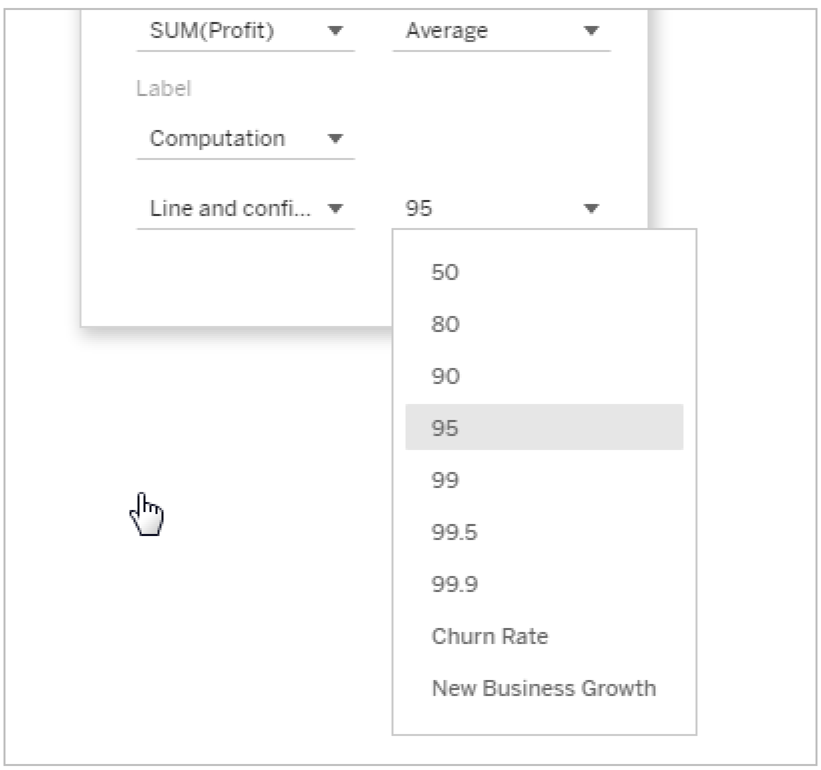
   - **None** –select this option to not show a label for the reference line.
   - **Value** – select this option to show a label corresponding to the line's value on the axis.
   - **Computation** – select this option to display the name of the continuous field that is the basis for your reference line and any computation that is performed.
   - **Custom** – select this option to build a custom label in the text box. You can use the menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as <Field Name> = <Value>.

6. Select how you want the tooltip to appear.

   - **None** –select this option to not show a tooltip for the reference line.
   - **Automatic** – select this option to show the default tooltip for the reference line.
   - **Custom** – select this option to build a custom label in the tooltip. You can use the menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as <Field Name> = <Value>.

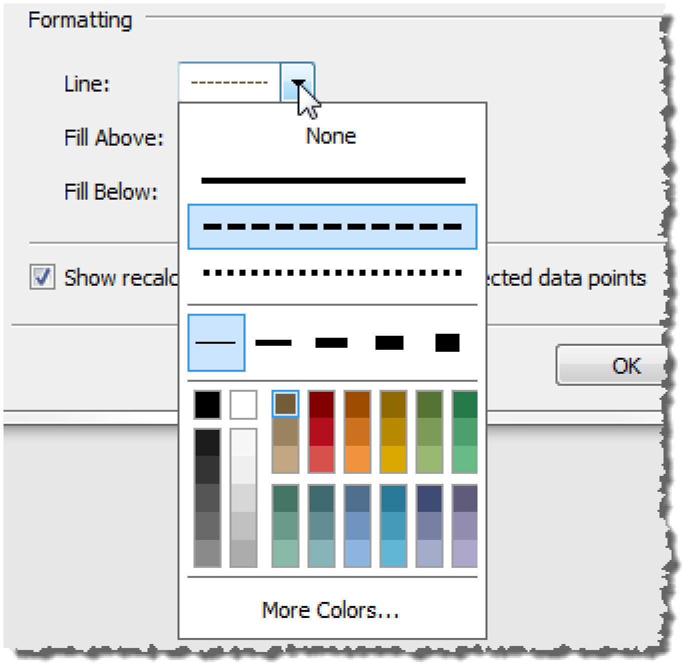
7. Specify whether to display the line with a confidence interval, just the line, or just the confidence interval.

   Confidence interval distribution bands shade the region in which the population average will fall *n* of the time, where *n* is the value you select in the drop-down on the right. You can choose one of the listed numeric values or select a parameter:

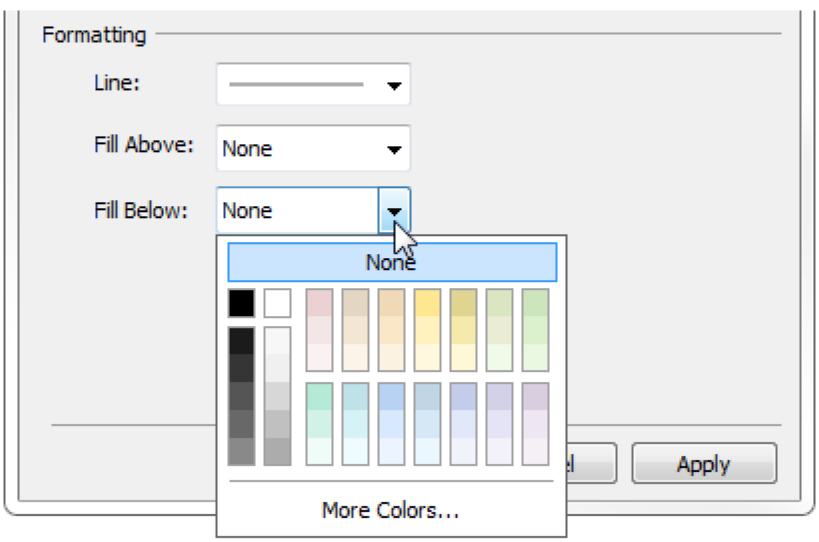
   

   The higher the value you select, the wider the bands will be.

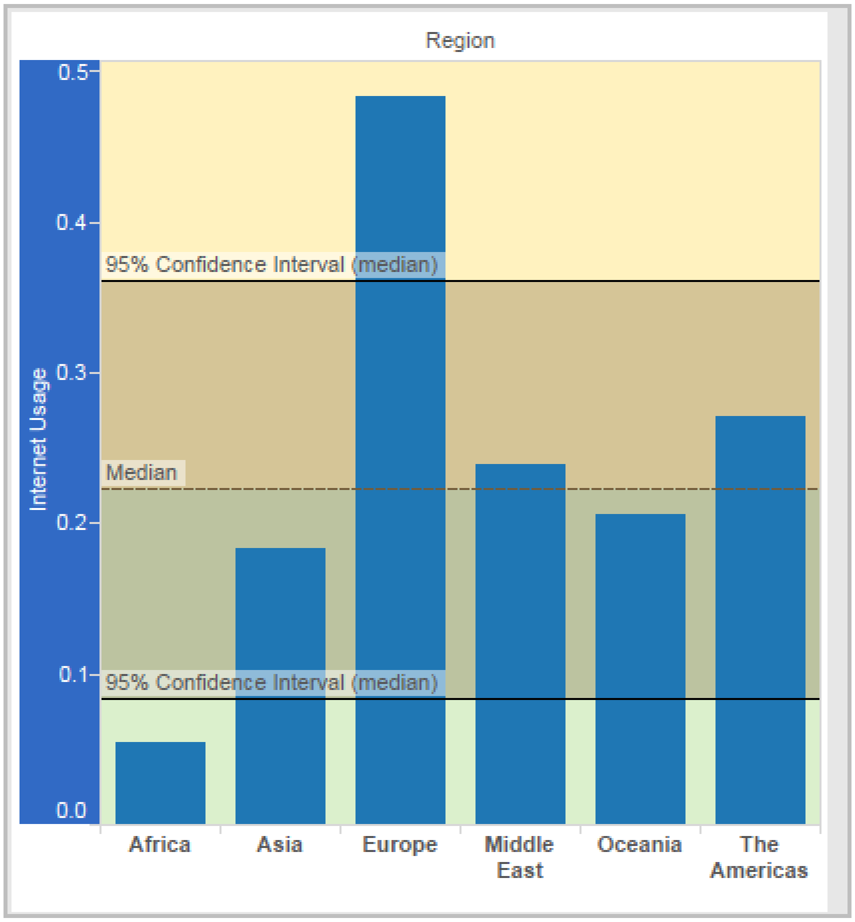
8. In Tableau Desktop, you can also specify formatting options for the line.



9. Optionally, add a fill color above and below the line.



When you are displaying a line and a confidence interval, the shading will be darker within the confidence interval, and lighter beyond it:



When you are displaying a confidence interval without a line, the fill colors are disregarded, though your settings are retained and then applied if you decide later to show a line.

10. Specify whether to **Show recalculated line for highlighted or selected data points**. For more information, see **Compare marks data with recalculated lines** on page 2712.

Add Reference Bands

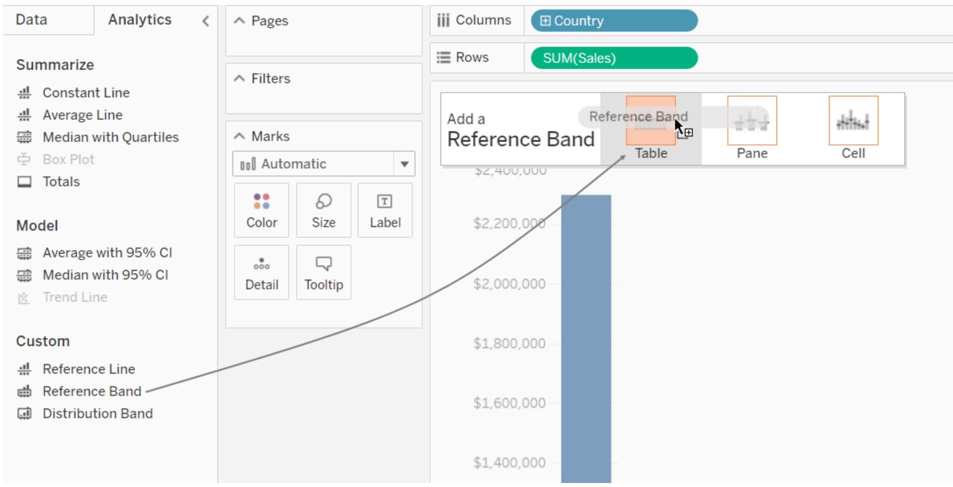
Reference bands are shaded areas behind the marks in the view between two constant or computed values on the axis. You can add reference bands to any continuous axis in the view.

To add a reference band:

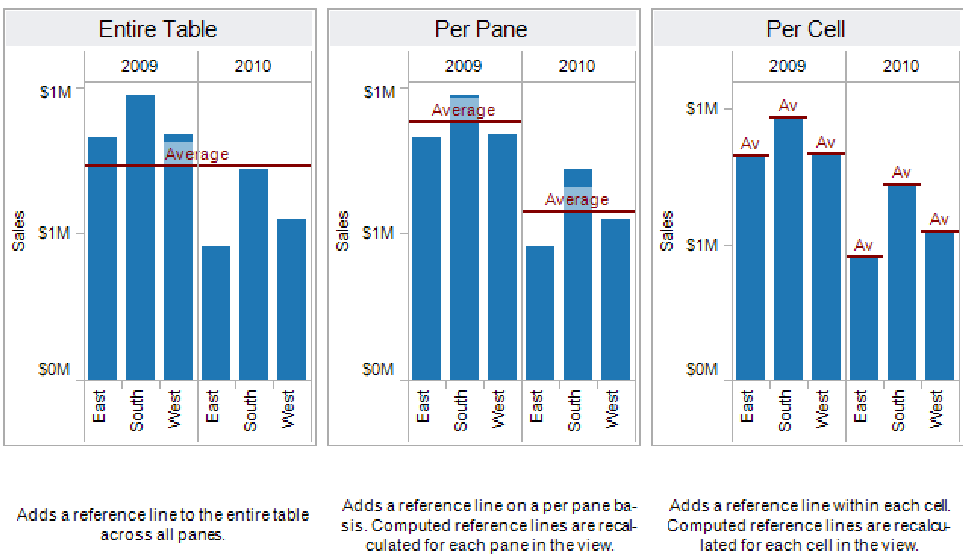
1. Drag **Reference Band** from the **Analytics** pane into the view. Tableau shows the possible destinations. The range of choices varies depending on the type of item and the

current view.

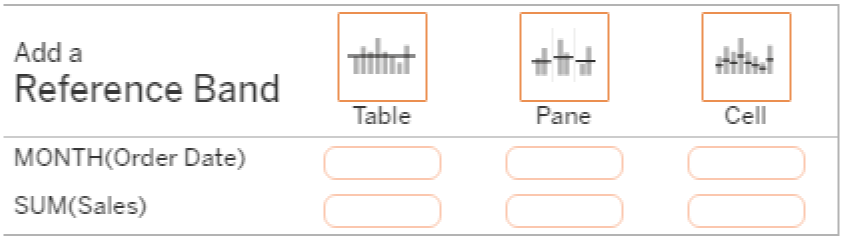
In a simple case, the drop target area would offer just three options:



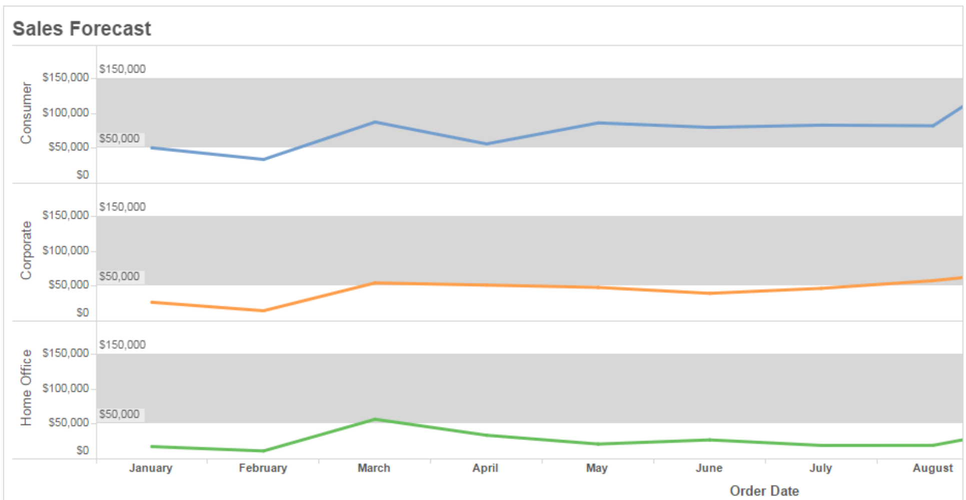
The terms **Table**, **Pane**, and **Cell** define the scope for the item:



For a more complicated view—for example, if the view contains multiple or dual axes—Tableau shows you an expanded drop target area that looks like this:



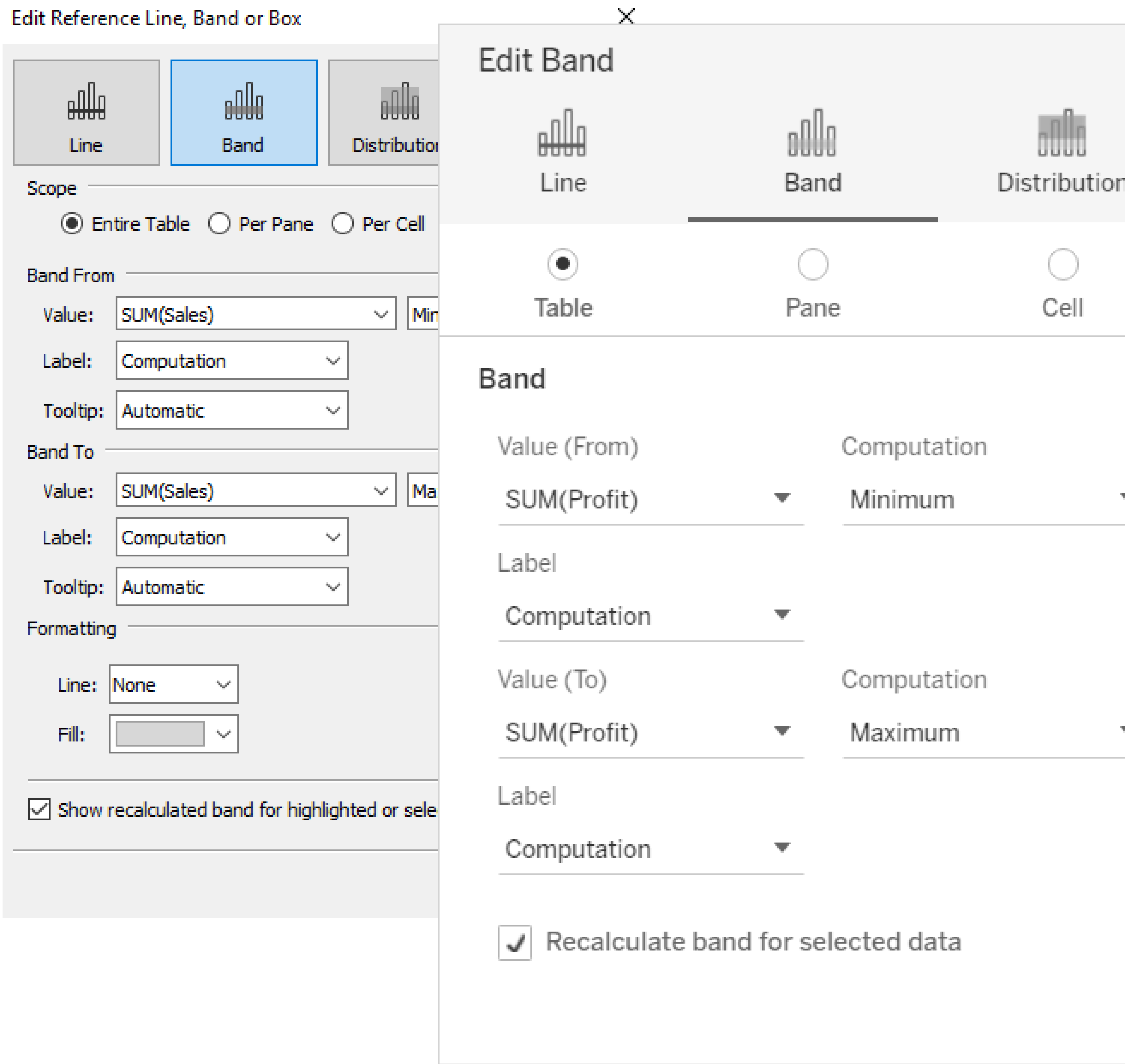
If you drop the item in one of the three larger boxes in the header (for example, the **Table** box), a separate set of bands is added for each continuous field in the view:



But if you drop the item in any of the lower boxes aligned with a specific continuous field, the band is added on the corresponding axis, with the specified scope.

When you drop the band in the target area, Tableau displays a dialog box:

Tableau Desktop version	**Web version**



2. The **Band** area is already selected at the top of the dialog box.

3. Select two continuous fields to use as the basis for your reference band one in each **Value** field. You can also select a parameter from the drop-down lists. Do not select the same continuous field and aggregation in both areas.

   You cannot select a continuous field that isn't currently in the view as the basis for your reference band. If you want to use such a continuous field, do the following:

a. Drag the continuous field from the Data pane to the Details target on the Marks card.

b. Change the continuous field's aggregation if necessary.

   This will not change the view, but it will allow you to use that continuous field as the basis for your reference band.

c. Click on the reference band in the view and choose **Edit** to re-open the **Edit Band** dialog box, and select the continuous field in in the **Value (From)** area and one in the **Value (To)** area.

4. Select a computation for each value. The aggregations that are displayed depend on the continuous field you select:

   - **Total** - extends the band to a value that is at the aggregate of all the values in either the cell, pane, or the entire view. This option is particularly useful when computing a weighted average rather than an average of averages. It is also useful when working with a calculation with a custom aggregation. The total is computed using the underlying data and behaves the same as selecting one of the totals option the Analysis menu.
   - **Sum** - extends the band to a value that is at the SUM of all the values in either the cell, pane, or entire view.
   - **Constant**- extends the band to a value that is at the specified value on the axis.
   - **Minimum** - extends the band to a value that is at the minimum value.
   - **Maximum** - extends the band to a value that is at the maximum value.
   - **Average** - extends the band to a value that is at the average value along the axis.
   - **Median**- extends the band to a value that is at the median value.

5. Select how you want to label the bands:

   - **None** –select this option to not show a label for the reference band.
   - **Value** – select this option to show a label corresponding to the band's value on the axis.
   - **Computation** – select this option to display the name of the continuous field that is the basis for your reference band and any computation that is performed.
   - **Custom** – select this option to build a custom label in the text box. You can use the

menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as `<Field Name> = <Value>`.

6. Select how you want the tooltip to appear.

   - **None** –select this option to not show a tooltip for the reference band.
   - **Automatic** – select this option to show the default tooltip for the reference band.
   - **Custom** – select this option to build a custom label in the tooltip. You can use the menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as `<Field Name> = <Value>`.

1. In Tableau Desktop, you can also specify formatting options for the bands. You can mark the two values with a line or select a shading color for the band.

2. Specify whether to **Show recalculated line for highlighted or selected data points**. For more information, see **Compare marks data with recalculated lines** on page 2712.

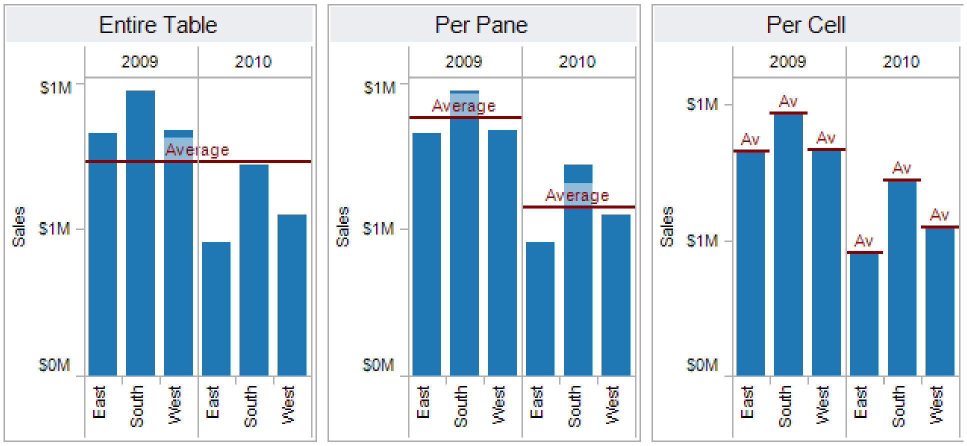
Add Reference Distributions

When you add a reference distribution, you specify one, two, or more values. With one value, the result is a line; with two or more values the result is a set of one, two, or more bands.

To add a reference distribution:

1. Drag **Distribution Band** from the **Analytics** pane into the view. Tableau shows the possible destinations. The range of choices varies depending on the type of item and the current view.

2. Select a scope for the distribution. The terms **Table**, **Pane**, and **Cell** define the scope for the item:



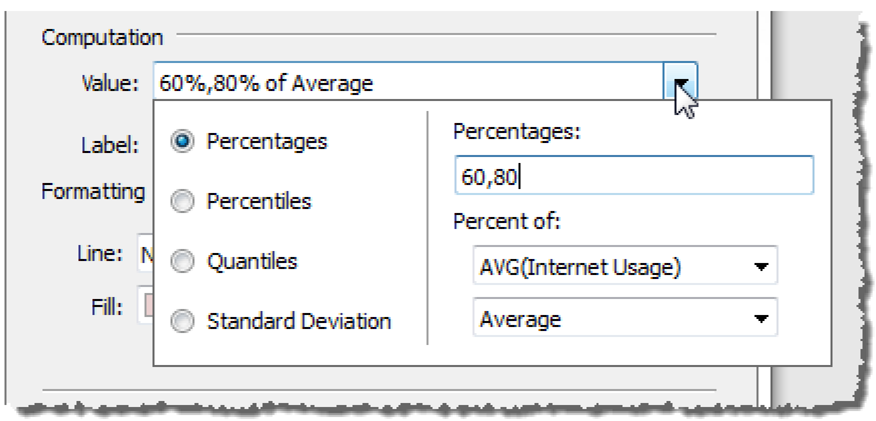
Adds a reference line to the entire table across all panes.

Adds a reference line on a per pane basis. Computed reference lines are recalculated for each pane in the view.

Adds a reference line within each cell. Computed reference lines are recalculated for each cell in the view.

3. Select the computation that will be used to create the distribution:

   - **Percentages** - shades the interval between the specified percentage values. Use a comma to separate two or more percentage values (for example, `60, 80`), and then specify which measure and aggregation to use for the percentages.



   - **Percentiles** - shades intervals at the specified percentiles. Choose *Enter a value* from the **Value** drop-down list, and then enter two or more numerical values, delimited by commas (for example, `60,80` or `25, 50, 75`).

   - **Quantiles** - breaks the view into the specified number of tiles using shading and lines. When you select this computation, you must also specify the number of tiles (from 3 to 10, inclusive). For example, if you select 3, Tableau calculates the boundaries between the first, second and third terciles by calling the general quantile function and asking for the 33.33 and the 66.66 quantiles. It then shades the three terciles differently.

Tableau uses estimation type 7 in the R standard to compute quantiles and percentiles.

- **Standard Deviation** - places lines and shading to indicated the specified number of standard deviations above and below the mean. When you select this option you must specify the factor, which is the number of standard deviations and whether the computation is on a sample or the population.

4. Specify how you want to label the distribution bands:

    - **None** –select this option to not show a label for the distribution bands.
    - **Value** – select this option to show a label corresponding to each distribution band's value on the axis.
    - **Computation** – select this option to display the name of the continuous field that is the basis for your distribution bands and any computation that is performed.
    - **Custom** – select this option to build a custom label in the text box. You can use the menu to the right of the text box to insert values such as the computation or the value. You can also type text directly into the box, so you could create a value such as <Field Name> = <Value>.

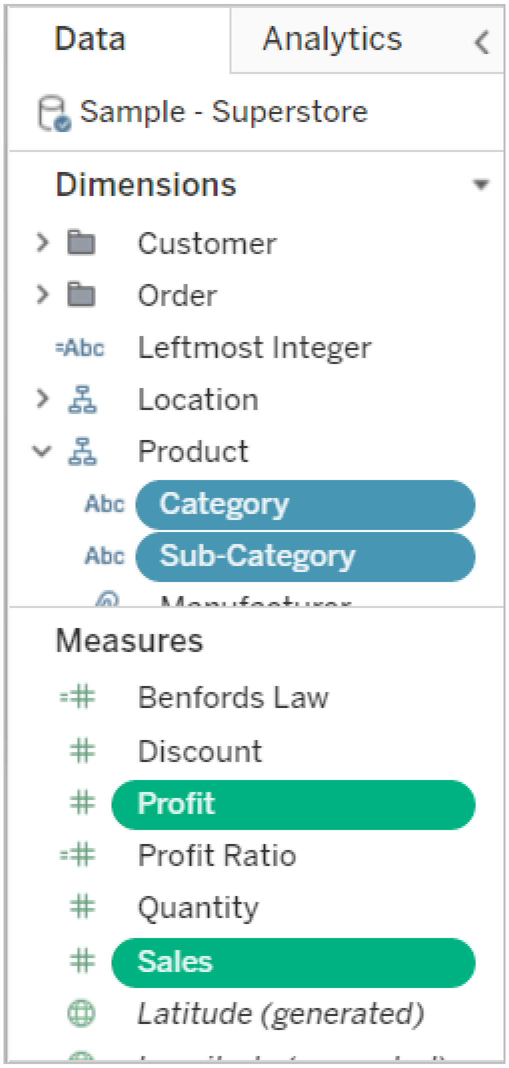
5. Specify whether to **Show recalculated band for highlighted or selected data points**. For more information, see **Compare marks data with recalculated lines** on page 2712 in the Tableau Desktop online help.

Add a Bullet Graph

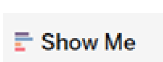
Reference distributions can also be used to create bullet graphs. A bullet graph is a variation of a bar graph developed to replace dashboard gauges and meters. The bullet graph is generally used to compare a primary measure to one or more other measures in the context of qualitative ranges of performance such as poor, satisfactory, and good. You can create a bullet graph by adding a distribution to indicate the qualitative ranges of performance, and a line to indicate the target. The following procedure uses Show Me to make this process easier.

1. Select one or more dimensions, and two measures in the **Data** pane. The bullet graph will compare measure values. For example, budget vs. actual; actual vs. target; etc. Select multiple fields in the **Data** pane by holding down the Ctrl key as you click fields. If

you are using the Superstore sample workbook, you can select the fields show below:

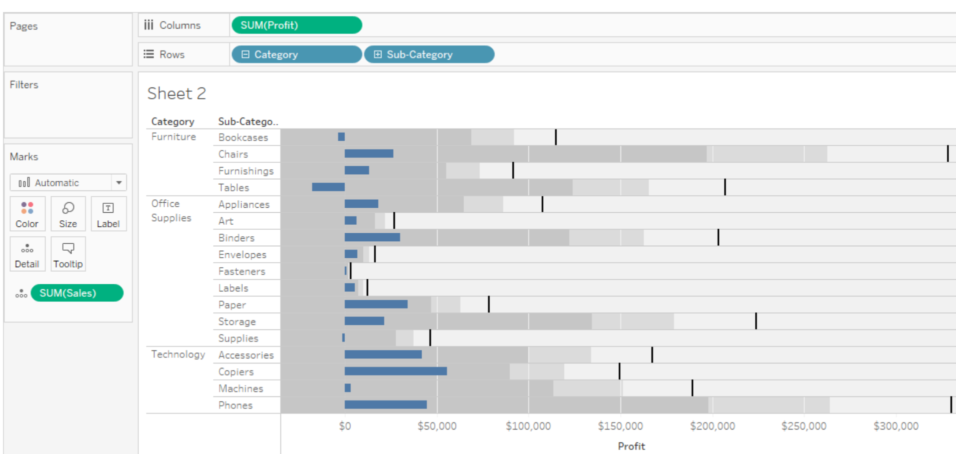


2. Click the **Show Me** button in the toolbar.



3. Select **Bullet Graph** in the Show Me pane.

Tableau adds a reference distribution that is defined at 60% and 80% of the Average of the measure on Detail. It also adds a reference line that marks the Average of that same measure. The other measure is placed on the Rows shelf.



You can edit either of these to change its definition. For example, you may want to add 100% to the set of distribution band values, or draw a line at a constant value. Click on the outer edge or a distribution band, or on the line, and choose **Edit**.

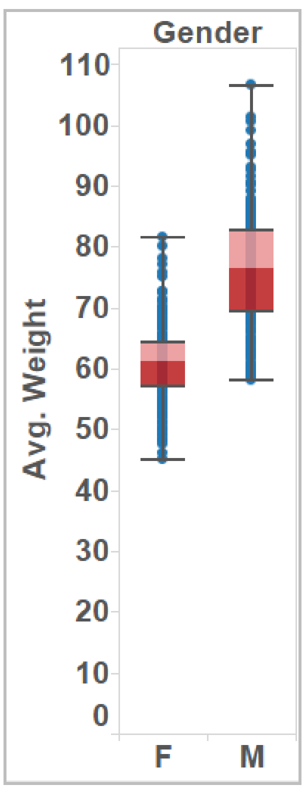
Add a Box Plot

In Tableau Desktop, but not on the web, you can add box plots to a continuous axis.

Use box plots, also known as box-and-whisker plots, to show the distribution of values along an axis.

Boxes indicate the middle 50 percent of the data (that is, the middle two quartiles of the data's distribution).

You can configure lines, called *whiskers*, to display all points within 1.5 times the interquartile range (in other words, all points within 1.5 times the width of the adjoining box), or all points at the maximum extent of the data, as shown in the following image:

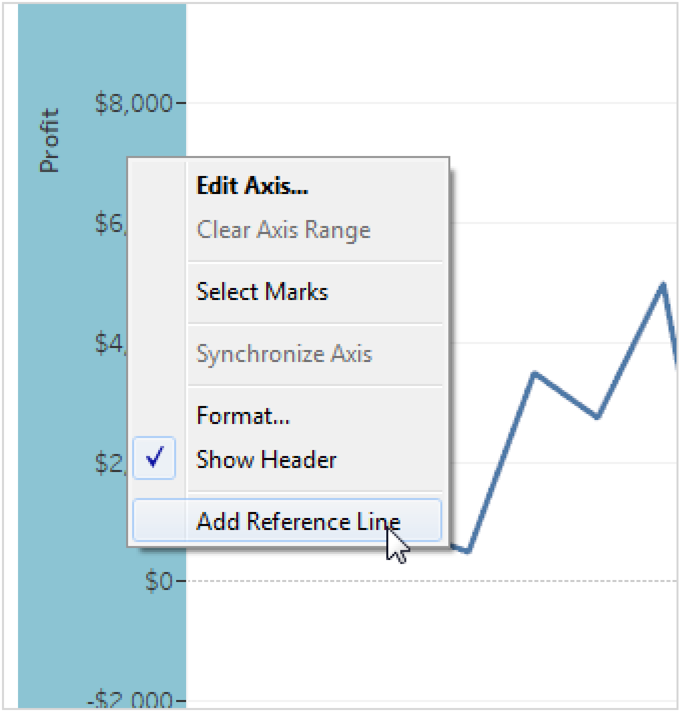


Boxplots are also available from the Show Me pane when you have at least one measure in the view:

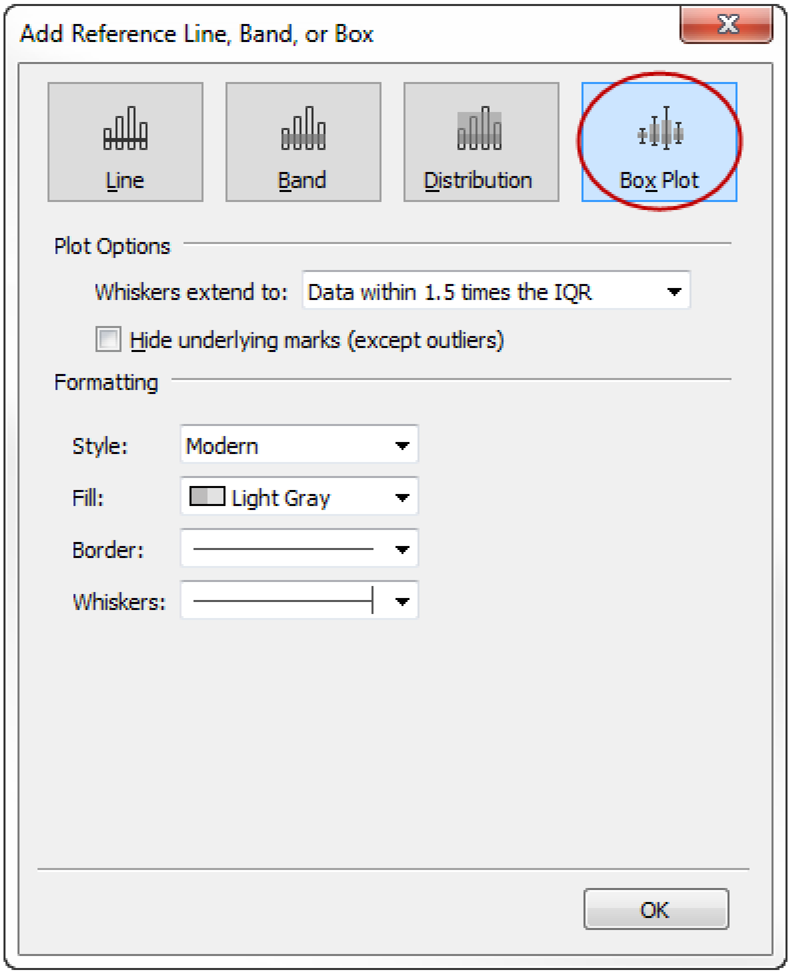
For information on Show Me, see **Use Show Me to Start a View** on page 1323

To add a box plot:

1. Right-click (Control-click on a Mac) on a quantitative axis and select **Add Reference Line**.



2. In the Add Reference Line, Band, or Box dialog box, select **Box Plot**.



3. Under Plot Options, specify placement for the whiskers:

   - **Data within 1.5 times the IQR** - places whiskers at a location that is 1.5 times the interquartile range—that is, 1.5 times further out than the width of the adjoining box. This is also known as a schematic box plot.

   - **Maximum extent of the data** - places whiskers at the farthest data point (mark) in the distribution. This is also known as a skeletal box plot.

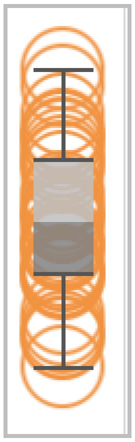
4. Specify whether to **Hide underlying marks (except outliers)**—that is, whether to hide all marks except those beyond the whiskers.

5. Configure the appearance of the plot by selecting a **Style** , **Fill**, **Border**, and **Whiskers**.

Box Plot Alternatives: Show Me Vs. Add Reference Line, Band, or Box

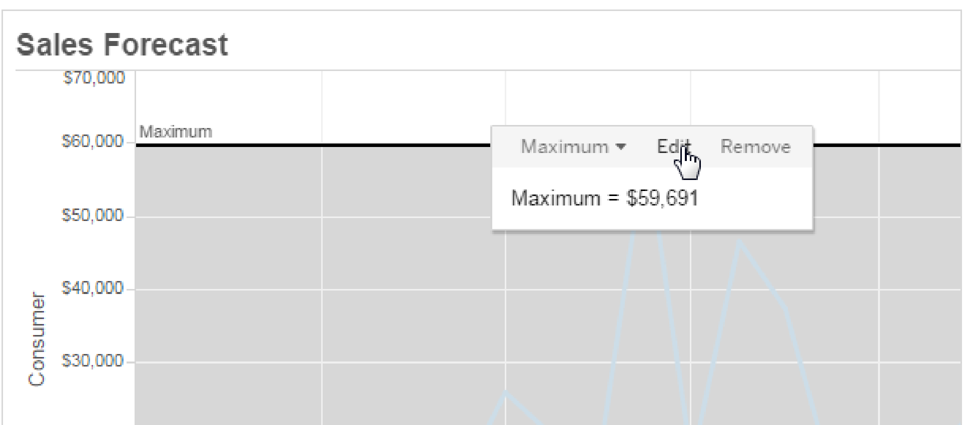
The difference between adding a box plot using Show Me and adding a box plot using Add Reference Line is that with Show Me, the box plot is your visualization, whereas with Add

Reference Line, Band, or Box, you are adding a box plot to an existing visualization. For example, you could create the following view by first selecting a circle view in Show Me, and then adding a box plot from Add Reference Line:



Edit Existing Reference Lines, Bands, and Distributions

You can edit existing lines, bands, or distributions. To do this, click on a line or on the outer edge of a band and choose **Edit** to reopen the edit dialog box for that object.



Remove Reference Lines, Bands, or Distributions

To remove a reference line, band, or distribution, click on a line or on the outer edge of a band and choose **Remove**. You can also drag a line or band off the view.

Add Trend Lines to a Visualization

You can show trend lines in a visualization to highlight trends in your data. You can publish a view that contains trend lines, and you add trend lines to a view as you edit it on the web.

When you add trend lines to a view, you can specify how you want them to look and behave.

> For a 5-minute walkthrough, see the Trend Lines free training video. Use your tableau.com account to sign in. To view more training and introductory videos, go to Free Training Videos on the Tableau website.

Add trend lines to a view

To add a trend line to a visualization:

1. Select the Analytics pane.
2. From the Analytics pane, drag **Trend Line** into the view, and then drop it on the Linear, Logarithmic, Exponential, Polynomial, or Power model types.

For more information on each of these model types, see **Trend Line Model Types** on page 2535.

About adding trend lines (and when you can't add them)

To add trend lines to a view, both axes must contain a field that can be interpreted as a number. For example, you cannot add a trend line to a view that has the Product **Category** dimension, which contains strings, on the **Columns** shelf and the **Profit** measure on the **Rows** shelf. However, you can add a trend line to a view of sales over time because both sales and time can be interpreted as numeric values.

For multidimensional data sources, the date hierarchies actually contain strings rather than numbers. Therefore, trend lines are not allowed. Additionally, the 'm/d/yy' and 'mmmm yyyy' date formats on all data sources do not allow trend lines.

If you have trend lines turned on and you modify the view in a way where trend lines are not allowed, the trend lines do not show. When you change the view back to a state that allows trend lines, they reappear.

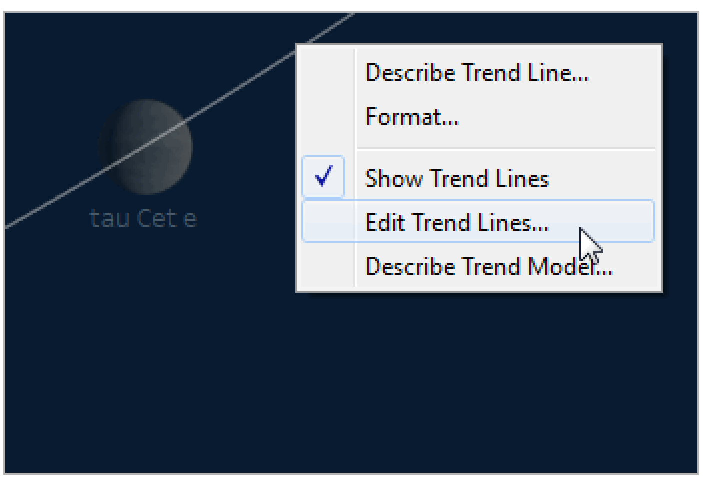
Tableau automatically stacks bar marks in many cases. However, trend lines cannot be turned on for stacked bars. You can turn off stacked marks by clearing the **Analysis** > **Stack Marks** option.

Edit a trend line

Once you add a trend line to the visualization, you can edit it to fit your analysis.

To edit a trend line:

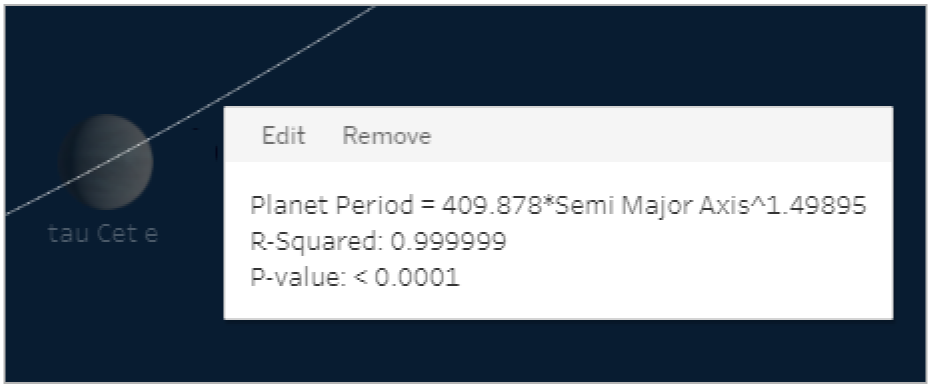
In Tableau Desktop: Right-click a trend line in the visualization, and select **Edit Trend Lines**.



In web editing mode:

1. In the visualization, click the trend line, and then hover your cursor over it.
2. In the tooltip that appears, select **Edit** to open the **Trend Line Options** dialog box.

> **Note**: To edit a trend line in Tableau Online or Tableau Server, you must have web editing permissions.



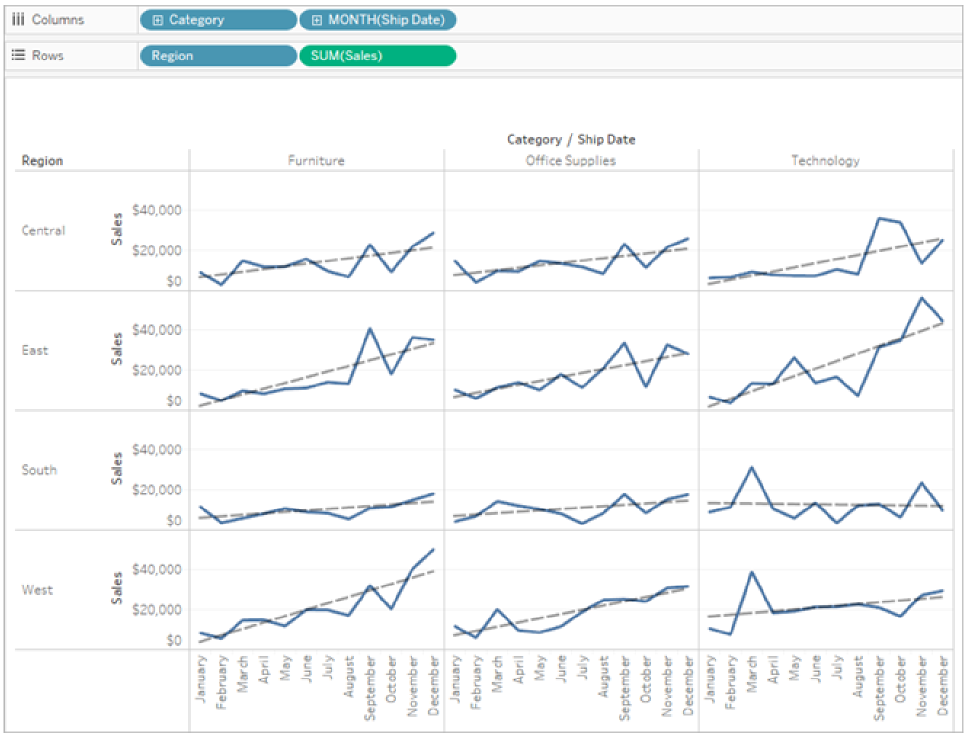
You can configure the following options in the Trend Line Options dialog box:

- Select a model type. For more information, see **Trend Line Model Types** on page 2535.
- Select which fields to use a factors in the trend line model. For more information, see **Choose which fields to use as factors in the trend line model** on the facing page.
- Decide whether to exclude color, using the **Allow a trend line per color** option. When you have color encodings in your view, you can use this option to add a single trend line that models all of the data, ignoring the color encoding.
- Decide whether to **Show Confidence Bands**. Tableau confidence bands show upper and lower 95% confidence lines by default when you add trend lines. Confidence lines are not supported for Exponential models.
- Select whether to **Force the y-intercept to zero**. This option is useful when you know that you want your trend line to begin at zero. This option is available only when both the **Rows** shelf and the **Columns** shelf contain a continuous field, as with a scatterplot.
- Decide whether to **show recalculated lines** when you select or highlight data in the visualization.

Choose which fields to use as factors in the trend line model

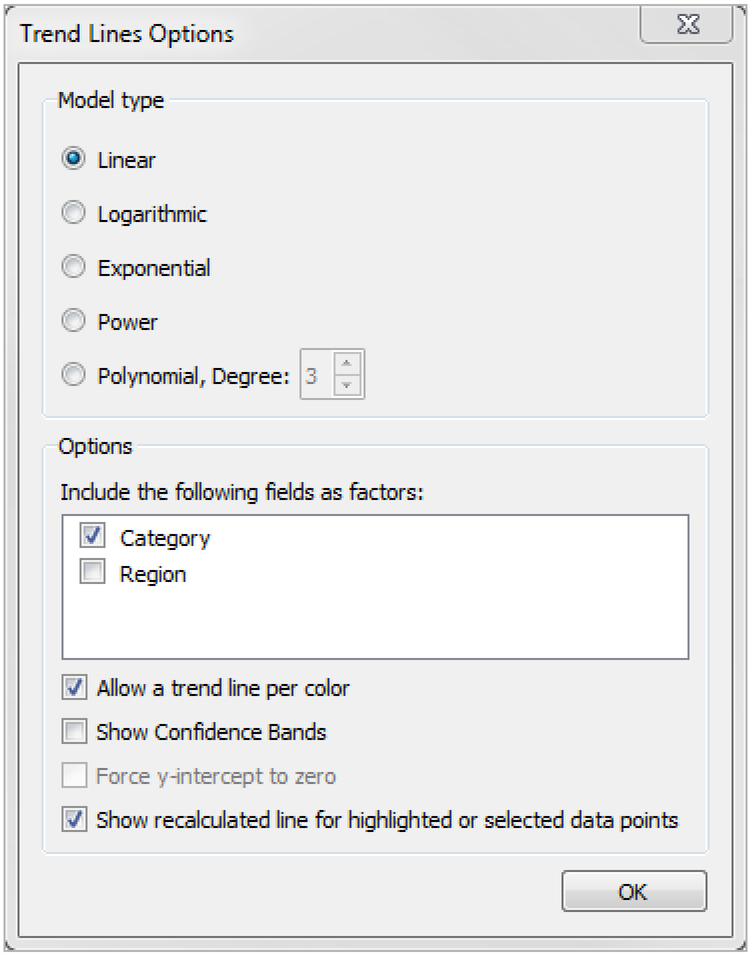
For trend models that are considering multiple fields, you can eliminate specific fields as factors in the trend line model.

Often you will want to remove factors because you want the trend line model to be based on the entire row in the table rather than broken up by the members or values of a field. Consider the following example. The view below shows the monthly sales for various products categories, broken out by region.

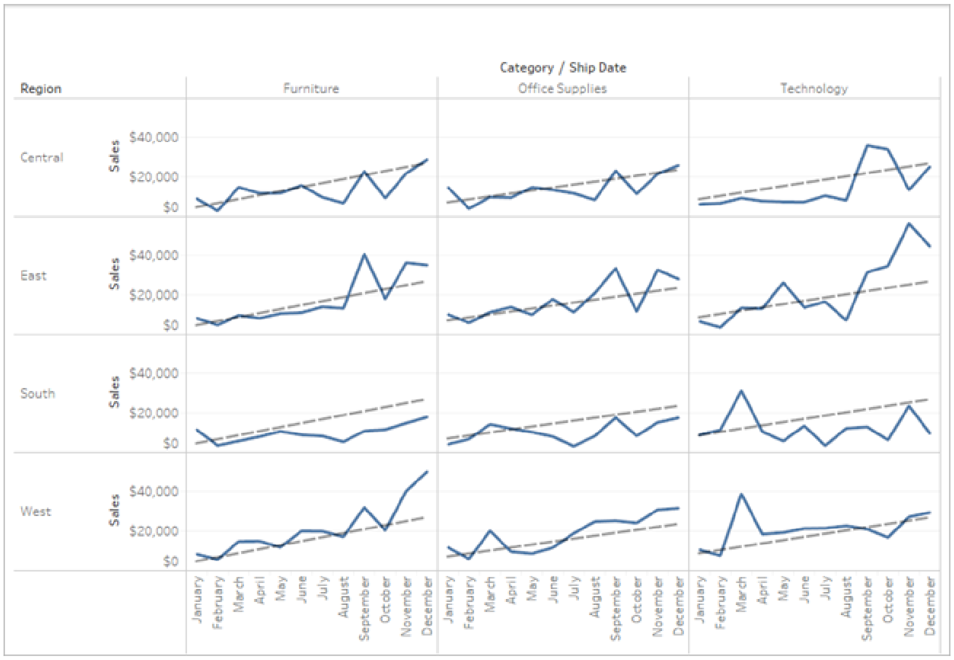


You can see that a separate model is created for each region.

Now remove **Region** as a factor in the model by deselecting it in the Trend Lines Options dialog box.



You can see that the trend line model within a category is now the same across all regions. This allows you to compare actual sales against a trend line that is the same for all regions.



Remove Trend Lines

To remove a trend line from a visualization, drag it off of the visualization area. You can also click a trend line and select **Remove**.

To remove all trend lines from the view, select **Analysis >Trend Lines >Show Trend Lines**.

> **Note**: In Tableau Desktop, trend line options are retained so that if you choose **Show Trend Lines** again from the Analysis menu, the options are as you last set them. However, if you close the workbook with trend lines turned off, trend line options revert to defaults.

See a description of a trend line or trend line model

After you add trend lines, you can display statistics on the trend line. For example, you can see the formula as well as r-squared and p values. For more information on the model types and terms used in the descriptions, see the **Trend Line Model Terms** on page 2537 and **Trend Line Model Types** on page 2535 sections.

To see a description of a trend line:

- Hover over any part of a trend line to see its description.

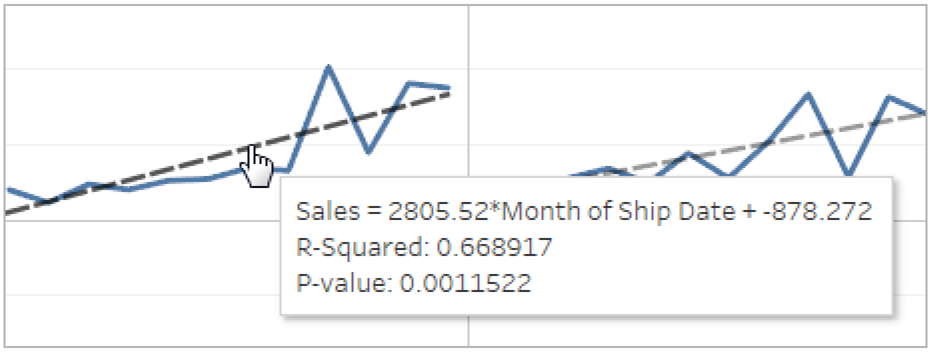
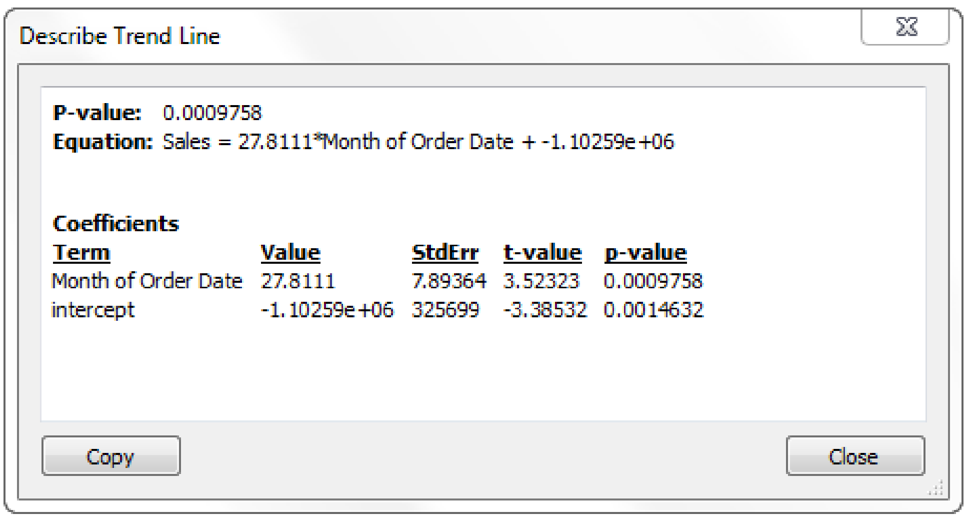


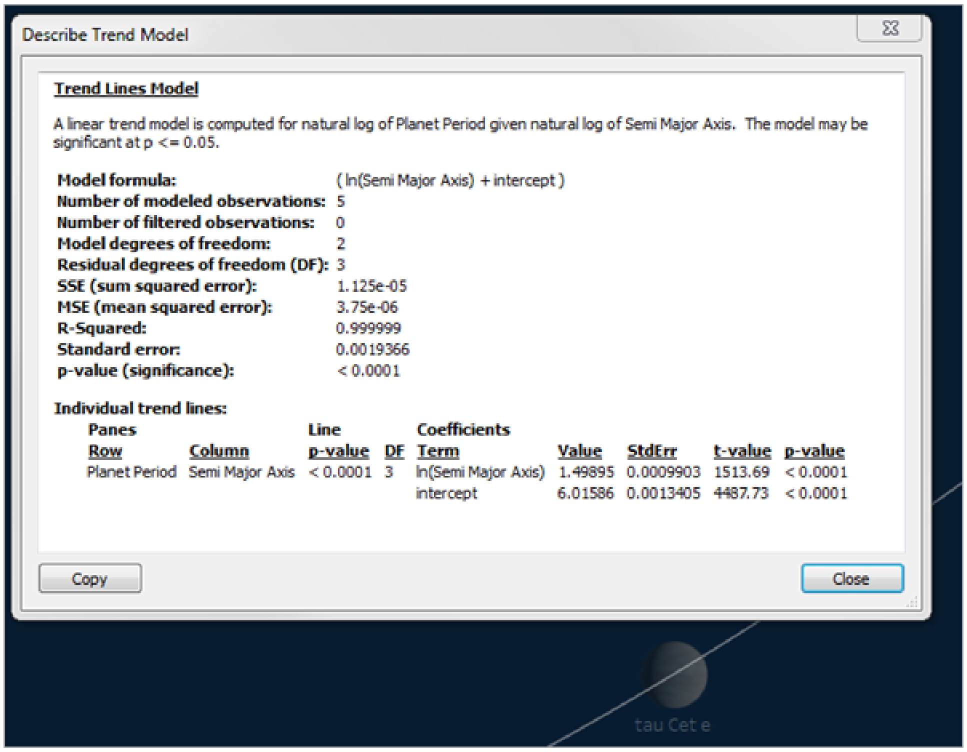
Tableau Desktop only

- Right-click the trend line in the visualization , and then select **Describe Trend Line**.



To view a full description of the model being used in the current view:

- Right-click a trend line in the visualization, and then select **Describe Trend Model**.



Trend Line Model Types

These model types are available for trend lines: **Linear** below, **Logarithmic** below, **Exponential** on the next page, **Power** on the next page, and **Polynomial** on the next page.

In the following formulas, X represents the explanatory variable, and Y the response variable.

Linear

With the linear model type the formula is:

```
Y = b0 + b1 * X
```

where `b1` is the slope and `b0` is the intercept of the line.

Logarithmic

With the logarithmic model type, the formula is:

```
Y = b0 + b1 * ln(X)
```

Because a logarithm is not defined for number less than zero, any marks for which the explanatory variable is negative are filtered before estimation of the model. Avoid using a model that discards some data unless you know that the data being filtered out is invalid. The trend line description reports how many marks were filtered before model estimation.

Exponential

With the exponential model type, the formula is:

`Y = exp(b0)* exp(b1 * X)`

With an exponential model, the response variable is transformed by the natural log before estimation of the model so the marks plotted in your view are found by plugging in various explanatory values to find values of `ln(Y)`.

`ln(Y) = b0 + b1 * X`

These values are then exponentiated to plot the trend line. What you see is the exponential model in the following form:

`Y = b2*exp(b1 * X)`

Where `b2` is the value of `exp(b0)`. Because a logarithm is not defined for numbers less than zero, any marks for which the response variable is negative are filtered before model estimation.

Power

With the power model type, the formula is:

`Y = b0 * X^b1`

With a power model, both variables are transformed by the natural log before estimation of the model resulting in this formula:

`ln(Y) = ln(b0) + b1 * ln(X)`

These values are then exponentiated to plot the trend line.

Because a logarithm is not defined for numbers less than zero, any marks for which the response variable or explanatory variable is negative are filtered before model estimation.

Polynomial

With the polynomial model type, the response variable is transformed into a polynomial series of the specified degree. The formula is:

`Y = b0 + b1 * X + b2 * X^2 + …`

With a polynomial model type, you must also select a **Degree** between 2 and 8. The higher polynomial degrees exaggerate the differences between the values of your data. If your data increases very rapidly, the lower order terms may have almost no variation compared to the higher order terms, rendering the model impossible to estimate accurately. Also, more complicated higher order polynomial models require more data to estimate. Check the model description of the individual trends line for a red warning message indicating that an accurate model of this type is not possible.

Trend Line Model Terms

When you view the description for a trend line model, there are several values listed. This section discusses what each of these values means.

Model formula

This is the formula for the full trend line model. The formula reflects whether you have specified to exclude factors from the model.

Number of modeled observations

The number of rows used in the view.

Number of filtered observations

The number of observations excluded from the model.

Model degrees of freedom

The number of parameters needed to completely specify the model. Linear, logarithmic, and exponential trends have model degrees of freedom of 2. Polynomial trends have model degrees of freedom of 1 plus the degree of the polynomial. For example a cubic trend has model degrees of freedom of 4, since we need parameters for the cubed, squared, linear and constant terms.

Residual degrees of freedom (DF)

For a fixed model, this value is defined as the number of observations minus the number of parameters estimated in the model.

SSE (sum squared error)

The errors are the difference between the observed value and the value predicted by the model. In the Analysis of Variance table, this column is actually the difference between the SSE of the simpler model in that particular row and the full model, which uses all the factors. This SSE also corresponds to the sum of the differences squared of the predicted values from the smaller model and the full model.

MSE (mean squared error)

The term MSE refers to "mean squared error" which is the SSE quantity divided by its corresponding degrees of freedom.

R-Squared

R-squared is a measure of how well the data fits the linear model. It is the ratio of the variance of the model's error, or unexplained variance, to the total variance of the data.

When the y-intercept is determined by the model, R-squared is derived using the following equation:

$$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

When the y-intercept is forced to 0, R-squared is derived using this equation instead:

$$1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} y_i^2}$$

In the latter case, the equation will not necessarily match Excel. This is because R-squared is not well defined in this case, and Tableau's behavior matches that of R instead of that of Excel.

Note: The R-Squared value for a linear trend line model is equivalent to the square of the result from the CORR function. See **Tableau Functions (Alphabetical)** on page 2342 for syntax and examples for CORR.

Standard error

The square root of the MSE of the full model. An estimate of the standard deviation (variability) of the "random errors" in the model formula.

p-value (significance)

The probability that an F random variable with the above degrees of freedom exceeds the observed F in this row of the Analysis of Variance table.

Analysis of Variance

This table, also known as the ANOVA table, lists information for each factor in the trend line model. The values are a comparison of the model without the factor in question to the entire model, which includes all factors.

Individual trend lines

This table provides information about each trend line in the view. Looking at the list you can see which, if any, are the most statistically significant. This table also lists coefficient statistics for each trend line. A row describes each coefficient in each trend line model. For example, a linear model with an intercept requires two rows for each trend line. In the Line column, the p-value and the DF for each line span all the coefficient rows. The DF column under the shows the residual degrees of freedom available during the estimation of each line.

Terms

The name of the independent term.

Value

The estimated value of the coefficient for the independent term.

StdErr

A measure of the spread of the sampling distribution of the coefficient estimate. This error shrinks as the quality and quantity of the information used in the estimate grows.

t-value

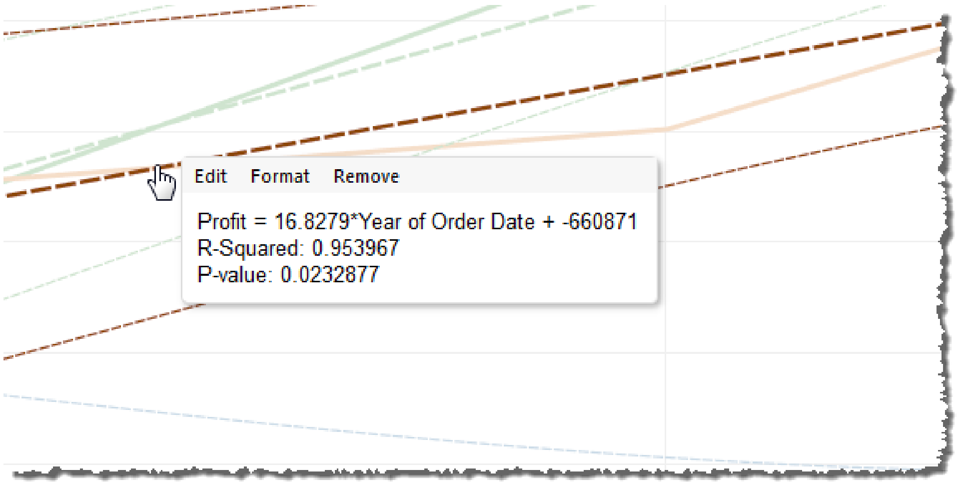
The statistic used to test the null hypothesis that the true value of the coefficient is zero.

p-value

The probability of observing a t-value that large or larger in magnitude if the true value of the coefficient is zero. So, a p-value of .05 gives us 95% confidence that the true value is not zero.

Assess Trend Line Significance

To see relevant information for any trend line in the view, hover the cursor over it:



The first line in the tooltip shows the equation used to compute a value of **Profit** from a value of **Year of Order Date**.

The second line, the R-Squared value, shows the ratio of variance in the data, as explained by the model, to the total variance in the data. For more details, see **Trend Line Model Terms** on page 2537.

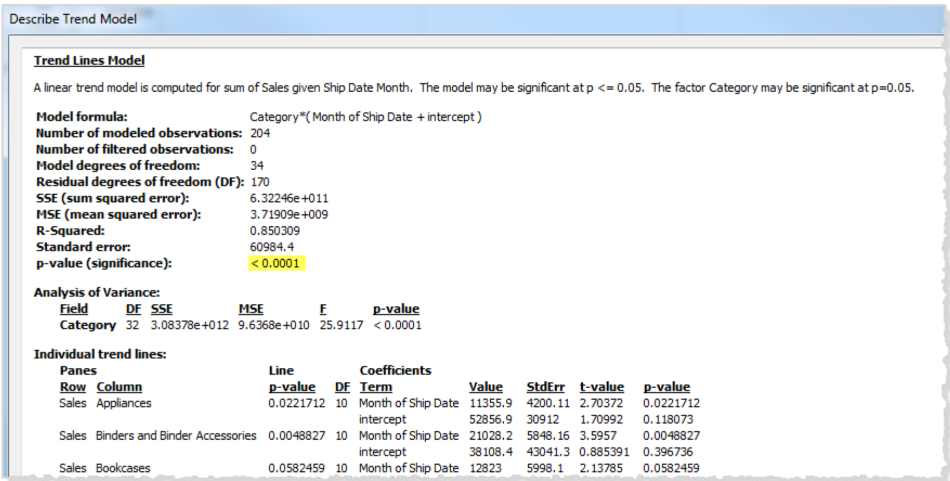
The third line, the P-value, reports the probability that the equation in the first line was a result of random chance. The smaller the p-value, the more significant the model is. A p-value of 0.05 or less is often considered sufficient.

Entire Model Significance

Once you've added a trend line to a view, you typically want to know the goodness of fit of the model, which is a measure of the quality of the model's predictions. In addition, you may be interested in the significance of each factor contributing to the model. To view these numbers, open the Describe Trend Model dialog box, right-click (Control-click on a Mac) in the view and select **Trend Lines >Describe Trend Model**.

When you are testing significance, you are concerned with the p-values. The smaller the p-value, the more significant the model or factor is. It is possible to have a model that has statistical significance but which contains an individual trend line or a term of an individual trend line that does not contribute to the overall significance.

Under Trend Lines Model, look for the line that shows the p-value (significance) of the model: The smaller the p-value, the less likely it is that the difference in the unexplained variance between models with and without the relevant measure or measures was a result of random chance.


Describe Trend Model

Trend Lines Model

A linear trend model is computed for sum of Sales given Ship Date Month. The model may be significant at p <= 0.05. The factor Category may be significant at p=0.05.

Model formula:	Category*(Month of Ship Date + intercept)
Number of modeled observations:	204
Number of filtered observations:	0
Model degrees of freedom:	34
Residual degrees of freedom (DF):	170
SSE (sum squared error):	6.32246e+011
MSE (mean squared error):	3.71909e+009
R-Squared:	0.850309
Standard error:	60984.4
p-value (significance):	< 0.0001

Analysis of Variance:

Field	DF	SSE	MSE	F	p-value
Category	32	3.08378e+012	9.6368e+010	25.9117	< 0.0001

Individual trend lines:

Panes		Line		Coefficients				
Row	Column	p-value	DF	Term	Value	StdErr	t-value	p-value
Sales	Appliances	0.0221712	10	Month of Ship Date	11355.9	4200.11	2.70372	0.0221712
				intercept	52856.9	30912	1.70992	0.118073
Sales	Binders and Binder Accessories	0.0048827	10	Month of Ship Date	21028.2	5848.16	3.5957	0.0048827
				intercept	38108.4	43041.3	0.885391	0.396736
Sales	Bookcases	0.0582459	10	Month of Ship Date	12823	5998.1	2.13785	0.0582459

This p-value for a model compares the fit of the entire model to the fit of a model composed solely of the grand mean (the average of data in the data view). That is, it assesses the explanatory power of the quantitative term f(x) in the model formula, which can be linear, polynomial, exponential, or logarithmic with the factors fixed. It is common to assess significance using the "95% confidence" rule. Thus, as noted above, a p-value of 0.05 or less is considered good.

Significance of Categorical Factors

In the Analysis of Variance table, sometimes referred to as the ANOVA table, each field that is used as a factor in the model is listed. For each field, among other values, you can see the p-value. In this case, the p-value indicates how much that field adds to the significance of the entire model. The smaller the p-value the less likely it is that the difference in the unexplained variance between models with and without the field was a result of random chance. The values displayed for each field are derived by comparing the entire model to a model that does not include the field in question.

The following image shows the Analysis of Variance table for a view of quarterly sales for the past two years of three different product categories.

Analysis of Variance:

Field	DF	SSE	MSE	F	p-value
Category	160	1.65811e+012	1.03632e+010	20.9732	< 0.0001
Region	136	2.64345e+012	1.94371e+010	39.3372	< 0.0001

As you can see, the p-values for **Category** and **Region** are both quite small. Both of these factors are statistically significant in this model.

For information on specific trend line terms, see **Trend Line Model Terms** on page 2537.

For ANOVA models, trend lines are defined by the mathematical formula:

```
Y = factor 1 * factor 2 * ...factorN * f(x) + e
```

The term `Y` is called the *response variable* and corresponds to the value you are trying to predict. The term `X` is the *explanatory variable*, and e (epsilon) is random error. The factors in the expression correspond to the categorical fields in the view. In addition, each factor is represented as a matrix. The `*` is a particular kind of matrix multiplication operator that takes two matrices with the same number of rows and returns a new matrix with the same number of rows. That means that in the expression `factor 1 * factor 2`, all combinations of the members of factor 1 and factor 2 are introduced. For example, if factor 1 and factor 2 both have three members, then a total of nine variables are introduced into the model formula by this operator.

Trend Line Assumptions

The p-values reported in Tableau trend lines depend on some assumptions about the data.

The first assumption is that, whenever a test is performed, the model for the mean is (at least approximately) correct.

The second assumption is that the "random errors" referred to in the model formula (see **Trend Line Model Types** on page 2535) are independent across different observations, and that they all have the same distribution. This constraint would be violated if the response variable had much more variability around the true trend line in one category than in another.

Assumptions Required to Compute Trend Lines

The Assumptions required to compute (using Ordinary Least Squares) each individual trend line are:

- Your model is an accurate functional simplification of the true data generating process (for example, no linear model for a log linear relationship).

- Your errors average to zero and are uncorrelated with your independent variable (for example, no error measuring the independent variable).
- Your errors have constant variance and are not correlated with each other (for example, no increase in error spread as your independent variable increases).
- Explanatory variables are not exact linear functions of each other (perfect multicollinearity).

Commonly Asked Questions About Trend Lines

This section describes some commonly asked questions regarding trend lines in Tableau.

How do I change the confidence level used in the model?

Tableau does not enforce a confidence level. It simply reports the significance of the whole model, or of a specific field, by showing the p-value. The p-value will measure the probability of obtaining the same trend result without taking the dimensions into account. For example, a trend of sales per time p-value of 0.05 means that there is 5% chance that the same value could be obtained without taking the time into consideration.

What does it mean if the p-value for the model is significant but the p-value for the specific field in the Analysis of Variance table is not significant?

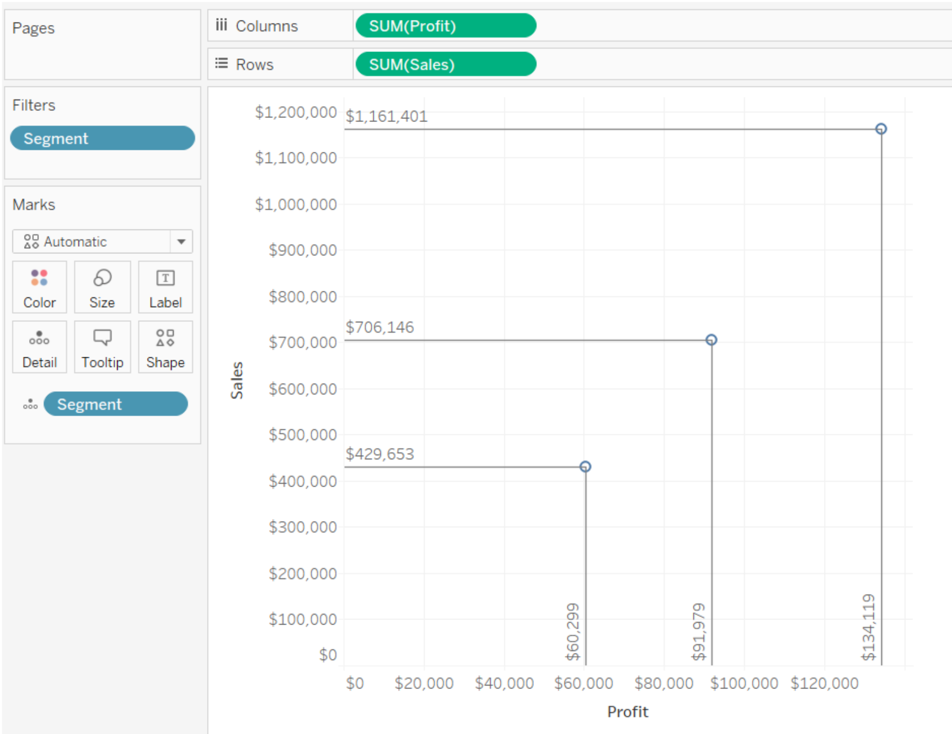
The p-value in the Analysis of Variance table indicates whether the field adds or detracts from the significance of the entire model. The smaller the p-value the less likely it is that the difference in the unexplained variance between models with and without the field was a result of random chance. The values displayed for each field are derived by comparing the entire model to a model that does not include the field in question. So, for the situation where the p-value for the model is significant but the p-value for the specific field is not, you know that the model is statistically significant, but you cannot be confident that the specific field in question adds anything to it. Consider whether you might not be better off removing the factor from the model.

What does it mean if the p-value for the specific field in the Analysis of Variance table is significant but the p-value for the model is not significant?

This could happen in a case when there is no "trend" within each pane. For example, the lines are flat, but the mean varies across a given factor.

Drop Lines

Use drop lines to distinguish marks and call out their position in the view. For example, in a view that is dense with scatter marks, you can turn on drop lines to show the position of a particular data point. When you add drop lines, a line is extended from the marks to one of the axes. You can show drop lines all the time or only when a mark is selected.



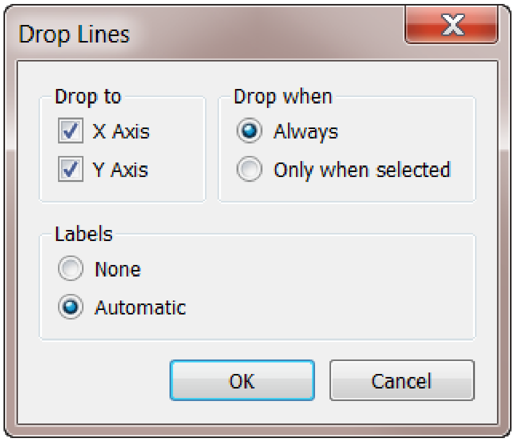
Note: Drop lines do not display when you publish a view to Tableau Server or Tableau Online.

To add drop lines to the view, right-click (control-click on Mac) the pane and select **Drop Lines** > **Show Drop Lines**.

By default, drop lines are set to only show when the mark is selected. You can change this setting and specify other options in the Drop Lines dialog box.

Right-click (Control-click on Mac) the pane and select **Drop Lines** > **Edit Drop Lines** to open the Drop Lines dialog box.

In the Drop Lines dialog box select an axis to draw the line to, whether to always show the drop lines, and whether to show labels.



Calculate Percentages in Tableau

Any analysis in Tableau can be expressed in terms of percentages. For example, rather than viewing sales for every product, you might want to view each product's sales as a percentage of the total sales for all products.

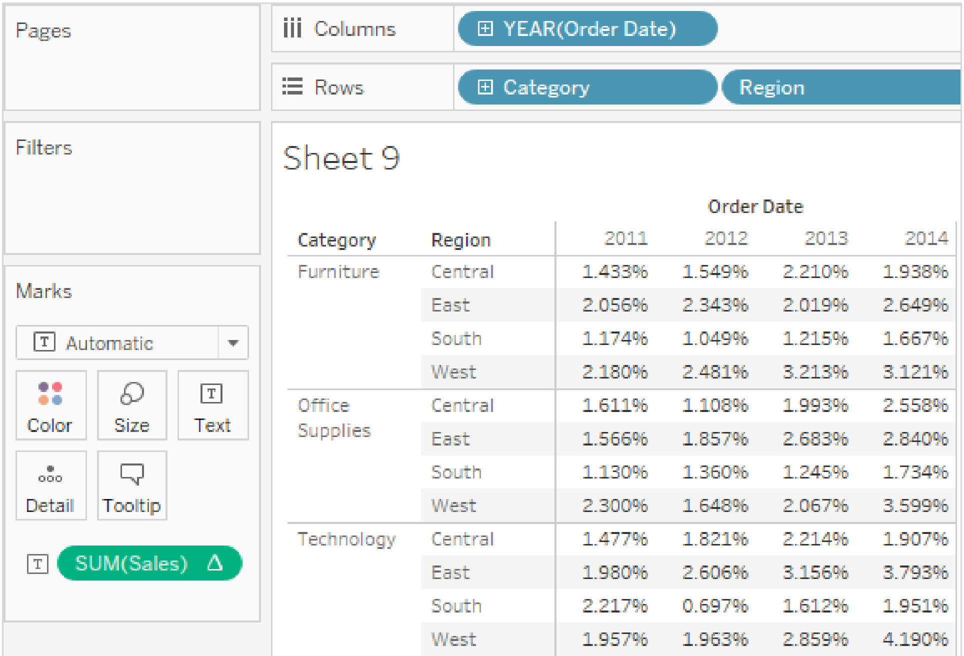
About percentages

There are two factors that contribute to the percentage calculation:

1. The data to which you compare all percentage calculations

Percentages are a ratio of numbers. The numerator is the value of a given mark. The denominator depends on the type of percentage you want, and is the number to which you compare all your calculations. The comparison can be based on the entire table, a row, a pane, and so on. By default, Tableau uses the entire table. Other percentage calculations are available via the **Percentage of** menu item. See **Percentage options** on page 2548.

The figure below is an example of a text table with percentages. The percentages are calculated with the **Sales** measure aggregated as a summation, and are based on the entire table.



		Order Date			
Category	Region	2011	2012	2013	2014
Furniture	Central	1.433%	1.549%	2.210%	1.938%
	East	2.056%	2.343%	2.019%	2.649%
	South	1.174%	1.049%	1.215%	1.667%
	West	2.180%	2.481%	3.213%	3.121%
Office Supplies	Central	1.611%	1.108%	1.993%	2.558%
	East	1.566%	1.857%	2.683%	2.840%
	South	1.130%	1.360%	1.245%	1.734%
	West	2.300%	1.648%	2.067%	3.599%
Technology	Central	1.477%	1.821%	2.214%	1.907%
	East	1.980%	2.606%	3.156%	3.793%
	South	2.217%	0.697%	1.612%	1.951%
	West	1.957%	1.963%	2.859%	4.190%

2. The aggregation

Percentages are computed on the basis of the aggregation for each measure. Standard aggregations include summation, average, and several others. See **Data Aggregation in Tableau** on page 341 for more information.

For example, if the aggregation applied to the **Sales** measure is a summation, then the default percentage calculation (percent of table) means that each number displayed is the SUM (Sales) for that mark divided by the SUM(Sales) for the entire table.

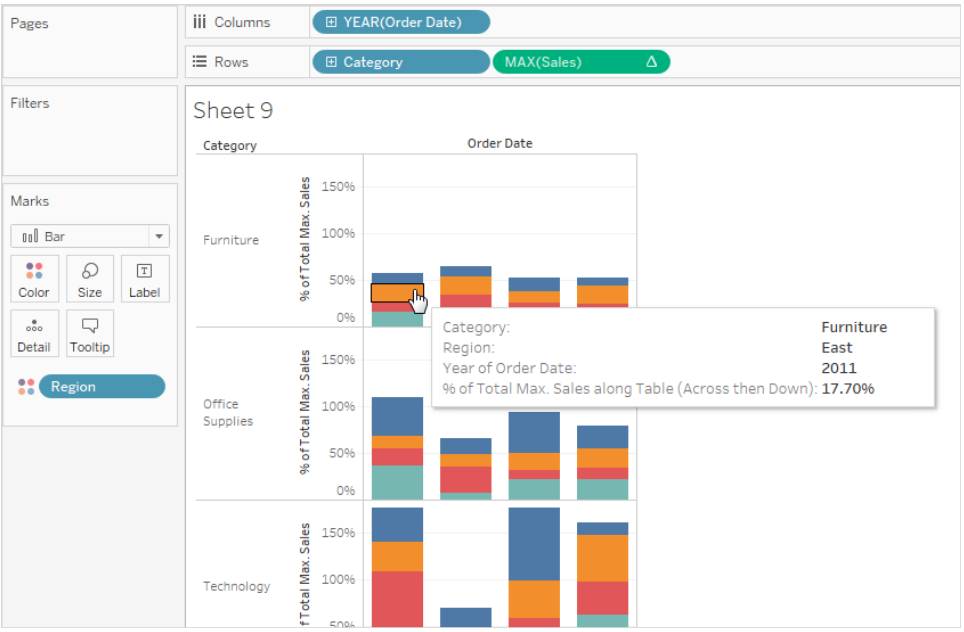
In addition to using predefined aggregations, you can use custom aggregations when calculating percentages. You define your own aggregations by creating a calculated field. Once the new field is created, you can use percentages on the field as you would any other field. See **Aggregate Functions in Tableau** on page 2192 for more information.

Percent calculations can also be applied to disaggregated data. In this case, all values are expressed as the percentage of a summation. You cannot choose any other aggregation.

Example

The view below shows a nested bar chart created using two dimensions and a measure that is aggregated as a maximum. Additionally, the data are color-encoded by a dimension and the default percentage calculation has been applied. Notice that the axis labels are modified to reflect the percent calculation.

The tooltip reveals that the maximum sales for furniture in the east in 2011 is 17.70% of the maximum for the entire table. What is the maximum for the table? If you recreate the view you'll see that the maximum occurs in the South, in the Technology category, in the year 2011. The tooltip for this bar segment would reveal a maximum sales of 100%.



The next view displays two disaggregated measures as a scatter plot. Again, the default percentage calculation has been applied as reflected by the modified axis labels.

The tooltip shows that the selected data point constitutes -0.475 percent of total profit and a 0.3552 percent of total sales. Percentage calculations are based on the entire data source.

How to calculate percentages

To calculate percentages in your visualization:

- Select **Analysis** > **Percentages Of**, and then select a percentage option.

Percentage options

Computing a percentage involves specifying a total on which the percentage is based. The default percentage calculation is based on the entire table. You can also choose a different option.

The option you choose is applied uniformly to all measures that appear on a worksheet. You cannot choose **Percent of Column** for one measure and **Percent of Row** for another.

> The percentage options on the Analysis menu correspond to the percentage table calculations. When you select a percentage option, you are actually adding a Percent of Total table calculation. See **Transform Values with Table Calculations** on page 2406 for more information.

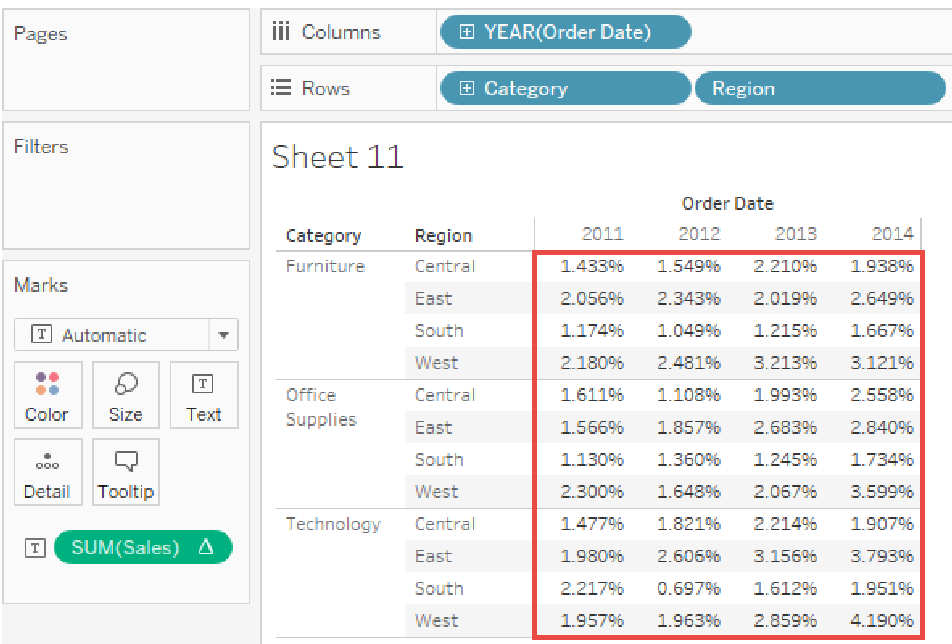
If you are unsure what the current percentage calculation means, display the grand totals. This provides more information about each row and column. For example, if you select **Percent of**

Row while displaying grand totals, you will see that the total for each row is exactly 100%. See **Show Totals in a Visualization** on page 1456 for more information on grand totals.

The percent calculation options are described in the following sections. In each case, the grand totals are displayed as well.

Percent of Table

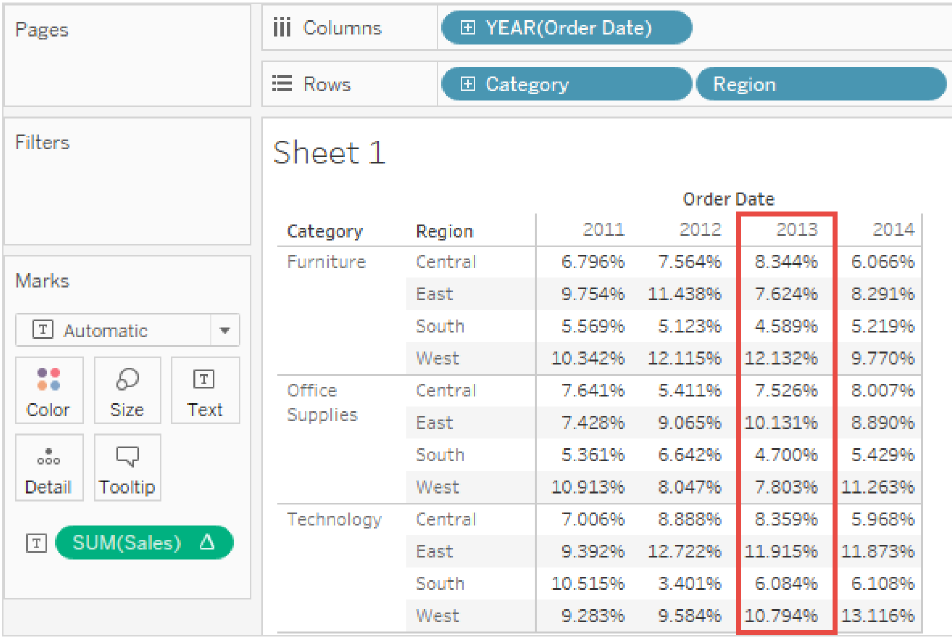
When you select **Percentage Of** > **Table** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for the entire worksheet (table). For example, Technology in the East region accounts for 3.79% of total sales in 2014. The grand totals for rows show that 2014 accounts for 31.95% of the total sales. Summing the grand totals for rows or for columns yields 100% of the total.



Category	Region	2011	2012	2013	2014
Furniture	Central	1.433%	1.549%	2.210%	1.938%
	East	2.056%	2.343%	2.019%	2.649%
	South	1.174%	1.049%	1.215%	1.667%
	West	2.180%	2.481%	3.213%	3.121%
Office Supplies	Central	1.611%	1.108%	1.993%	2.558%
	East	1.566%	1.857%	2.683%	2.840%
	South	1.130%	1.360%	1.245%	1.734%
	West	2.300%	1.648%	2.067%	3.599%
Technology	Central	1.477%	1.821%	2.214%	1.907%
	East	1.980%	2.606%	3.156%	3.793%
	South	2.217%	0.697%	1.612%	1.951%
	West	1.957%	1.963%	2.859%	4.190%

Percent of Column

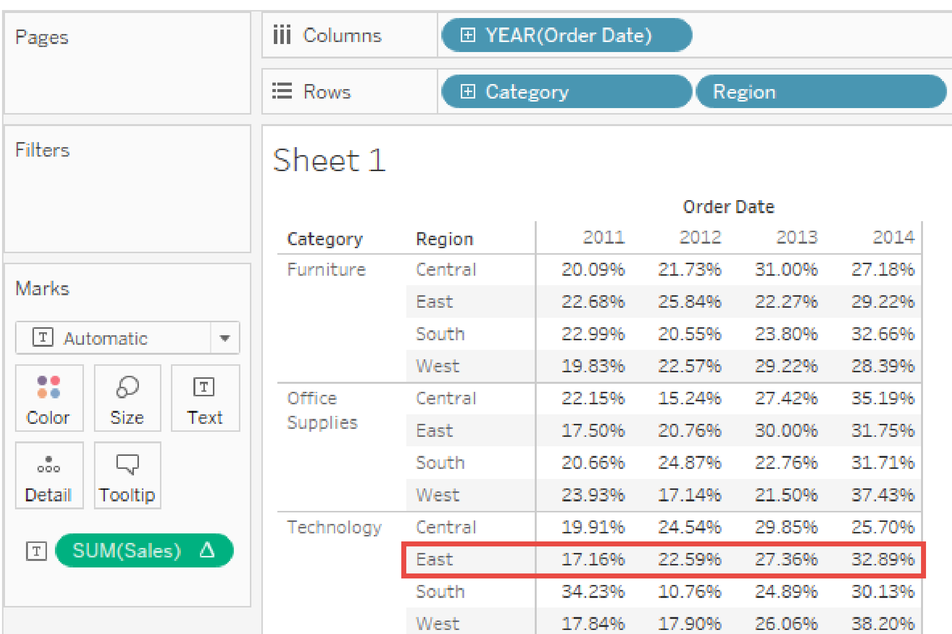
When you select **Percentage of** > **Column** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for the column. The values within the red box add up to 100%.



Category	Region	2011	2012	2013	2014
		Order Date			
Furniture	Central	6.796%	7.564%	8.344%	6.066%
	East	9.754%	11.438%	7.624%	8.291%
	South	5.569%	5.123%	4.589%	5.219%
	West	10.342%	12.115%	12.132%	9.770%
Office Supplies	Central	7.641%	5.411%	7.526%	8.007%
	East	7.428%	9.065%	10.131%	8.890%
	South	5.361%	6.642%	4.700%	5.429%
	West	10.913%	8.047%	7.803%	11.263%
Technology	Central	7.006%	8.888%	8.359%	5.968%
	East	9.392%	12.722%	11.915%	11.873%
	South	10.515%	3.401%	6.084%	6.108%
	West	9.283%	9.584%	10.794%	13.116%

Percent of Row

When you select **Percentage of Row**, each measure on the worksheet is expressed as a percentage of the total for the row. The values within the red box add up to 100%.

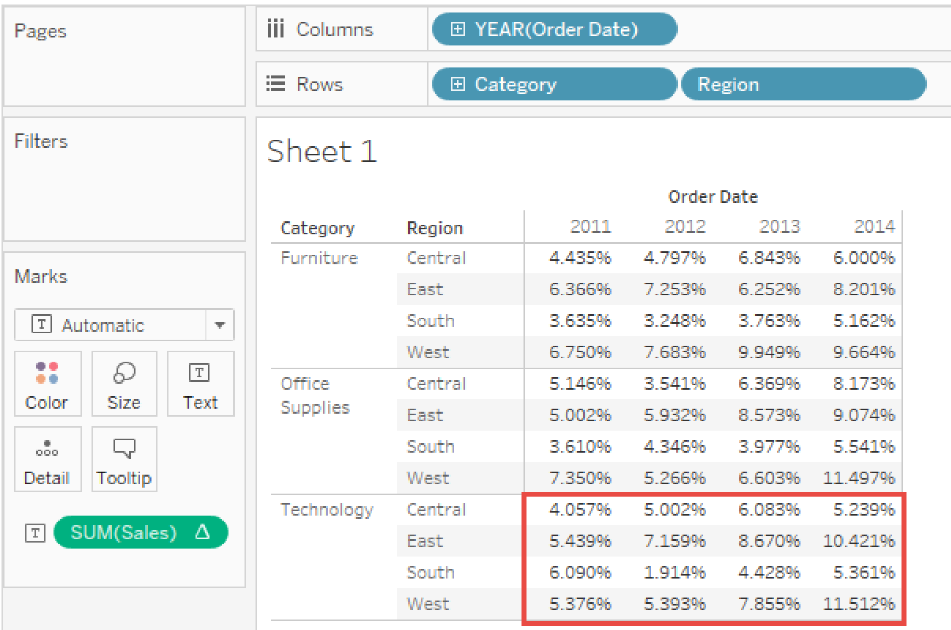


Category	Region	2011	2012	2013	2014
		Order Date			
Furniture	Central	20.09%	21.73%	31.00%	27.18%
	East	22.68%	25.84%	22.27%	29.22%
	South	22.99%	20.55%	23.80%	32.66%
	West	19.83%	22.57%	29.22%	28.39%
Office Supplies	Central	22.15%	15.24%	27.42%	35.19%
	East	17.50%	20.76%	30.00%	31.75%
	South	20.66%	24.87%	22.76%	31.71%
	West	23.93%	17.14%	21.50%	37.43%
Technology	Central	19.91%	24.54%	29.85%	25.70%
	East	17.16%	22.59%	27.36%	32.89%
	South	34.23%	10.76%	24.89%	30.13%
	West	17.84%	17.90%	26.06%	38.20%

Percent of Pane

When you select **Percentage of** > **Pane** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for the panes in the view. This option is equivalent to **Percent of Table** when the table consists of only a single pane.

In the following view, the red box constitutes a single pane; the values within the red box add up to 100%.

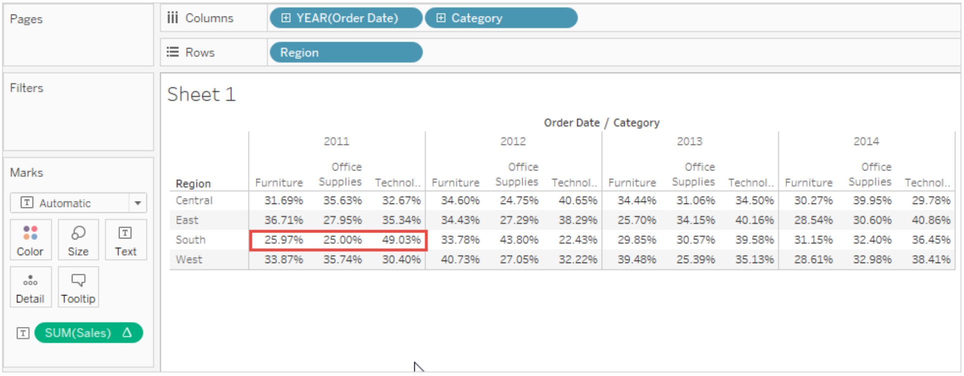


Category	Region	2011	2012	2013	2014
Furniture	Central	4.435%	4.797%	6.843%	6.000%
	East	6.366%	7.253%	6.252%	8.201%
	South	3.635%	3.248%	3.763%	5.162%
	West	6.750%	7.683%	9.949%	9.664%
Office Supplies	Central	5.146%	3.541%	6.369%	8.173%
	East	5.002%	5.932%	8.573%	9.074%
	South	3.610%	4.346%	3.977%	5.541%
	West	7.350%	5.266%	6.603%	11.497%
Technology	Central	4.057%	5.002%	6.083%	5.239%
	East	5.439%	7.159%	8.670%	10.421%
	South	6.090%	1.914%	4.428%	5.361%
	West	5.376%	5.393%	7.855%	11.512%

Percent of Row in Pane

When you select **Percentage of** > **Row in Pane** from the **Analysis** menu, each measure on the worksheet is expressed as a percentage of the total for a row within a pane. This option is equivalent to as **Percent of Row** when the table is only a single pane wide.

In the following view, the red box constitutes a row within a pane; the values within the red box add up to 100%.

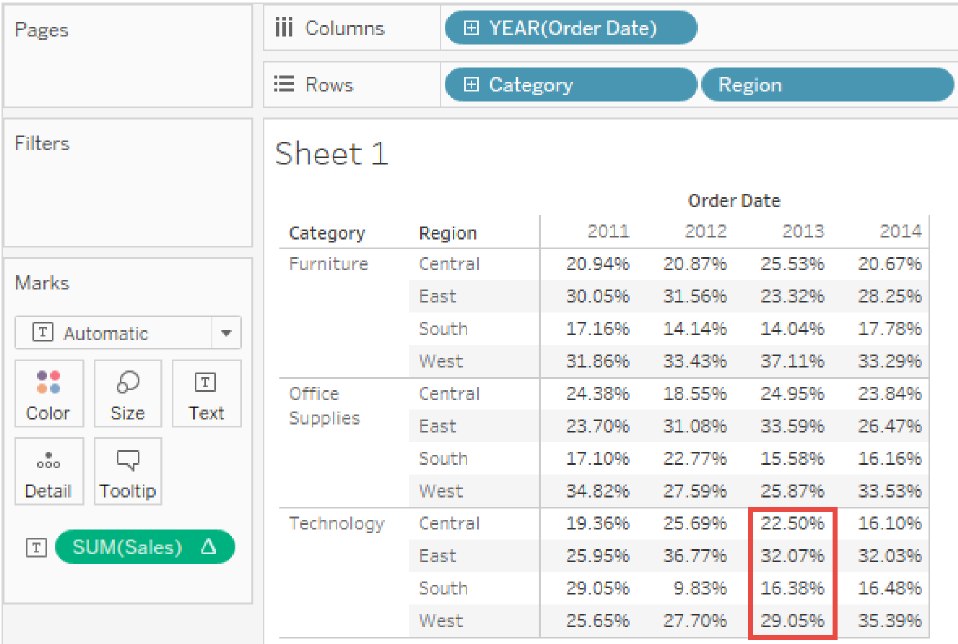


Note: If you place **Measure Names** as the inner dimension on the **Columns** shelf (that is, the dimension farthest to the right), Tableau will return 100% for each mark because you cannot total up the values for multiple measure names. For example, you can't total up the values for SUM(Sales) and SUM(Profit).

Percent of Column in Pane

When you select **Percentage of** > **Column in Pane** from the **Analysis** menu, each measure in the worksheet is expressed as a percentage of the total for a column within a pane. This option is equivalent to as **Percent of Column** when the table is only a single pane high.

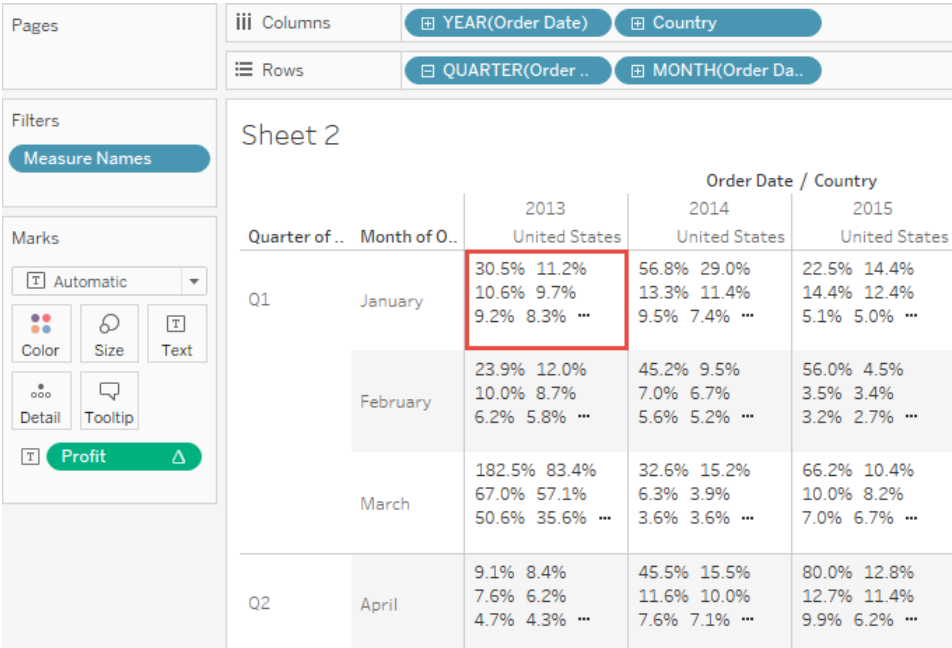
In the following view, the red box constitutes a column within a pane; the values within the red box add up to 100%.



If you place **Measure Names** as the inner dimension on the **Rows** shelf (that is, the dimension farthest to the right on the shelf), Tableau will return 100% for each mark because you cannot total up the values for multiple measure names. For example, you can't total up the values for SUM(Sales) and SUM(Profit).

Percent of Cell

When you select **Percentage Of** > **Cell** from the Analysis menu, each measure on the worksheet is expressed as a percentage of the total for each individual cell in the view. Most views show only one value per cell, in which case all cells show a percentage of 100%. But in some cases, as, for example, when you disaggregate data, a single cell can contain multiple values:



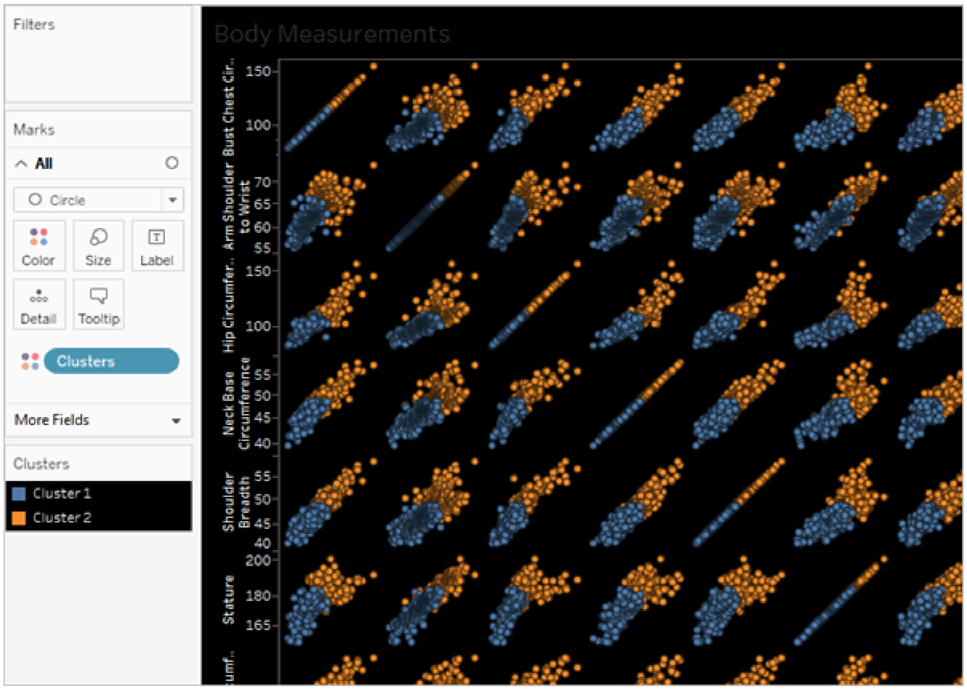
See Also

Transform Values with Table Calculations on page 2406

[Video] Aggregation, Granularity, and Ratio Calculations. Sign in with your tableau.com account to access the free video.

Find Clusters in Data

Cluster analysis partitions marks in the view into clusters, where the marks within each cluster are more similar to one another than they are to marks in other clusters.



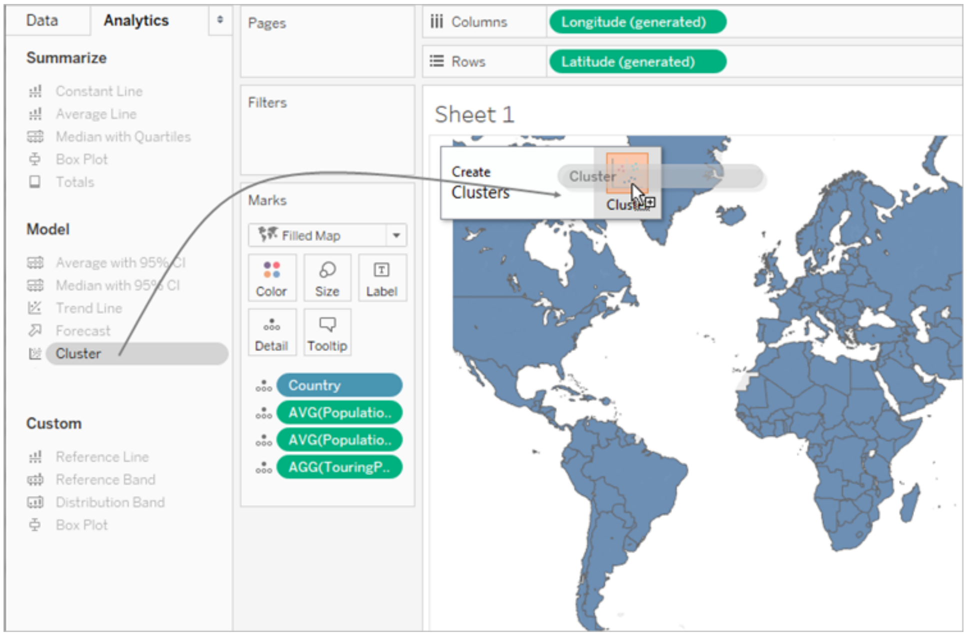
Watch a Video: To see related concepts demonstrated in Tableau, watch Clustering, a 2-minute free training video. Use your tableau.com account to sign in.

For an example that demonstrates the process of creating clusters with sample data, see **Example: Create clusters using World Economic Indicators data** on page 2571.

Create clusters

To find clusters in a view in Tableau, follow these steps.

1. Create a view.
2. Drag **Cluster** from the **Analytics** pane into the view, and drop it on in the target area in the view:



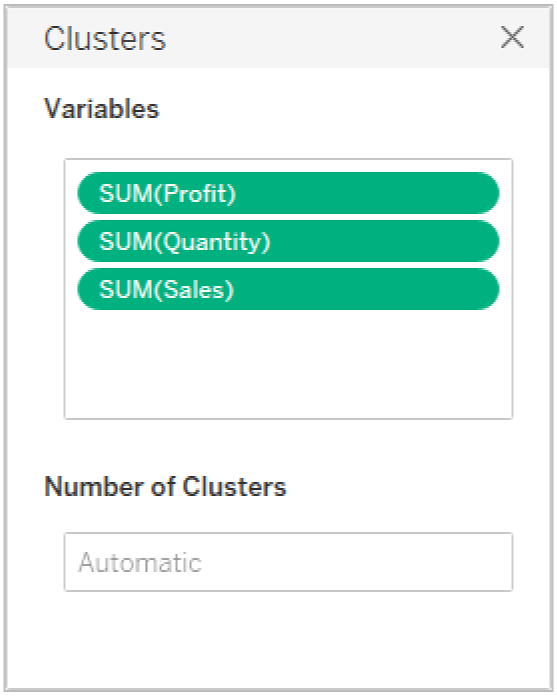
You can also double-click **Cluster** to find clusters in the view.

When you drop or double-click **Cluster**:

- Tableau creates a **Clusters** group on **Color**, and colors the marks in your view by cluster. If there is already a field on **Color,** Tableau moves that field to **Detail** and replaces it on **Color** with the clustering results.

  Tableau assigns each mark in the view to one of the clusters. In some cases, marks that do not fit well into a cluster are assigned to a "Not Clustered" cluster.

- Tableau displays the Clusters dialog box, where you can customize the cluster.



3. Customize the cluster results by doing either of the following in the Clusters dialog box.

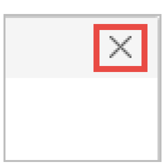
   - Drag new fields from the **Data** pane into the Variables area of the Clusters dialog box. You can also drag fields out of the Variables area to remove them.

     When you add variables, measures are aggregated using the default aggregation for the field; dimensions are aggregated using ATTR, which is the standard way that Tableau aggregates dimensions.

     To change the aggregation for a variable, right-click it.

   - Specify the number of clusters (between 2 and 50). If you do not specify a value, Tableau will automatically create up to 25 clusters.

4. When you finish customizing the cluster results, click the X in the upper-right corner of the Clusters dialog box to close it:



> **Note:** You can move the cluster field from Color to another shelf in the view. However, you cannot move the cluster field from the **Filters** shelf to the **Data** pane.

To rename the resulting clusters, you must first save the cluster as a group. For details, see **Create a group from cluster results** on page 2561 and **Edit clusters** on the facing page.

Clustering constraints

Clustering is available in Tableau Desktop, but is not available for authoring on the web (Tableau Server, Tableau Online). Clustering is also not available when any of the following conditions apply:

- When you are using a cube (multidimensional) data source.
- When there is a blended dimension in the view.
- When there are no fields that can be used as variables (inputs) for clustering in the view.
- When there are no dimensions present in an aggregated view.

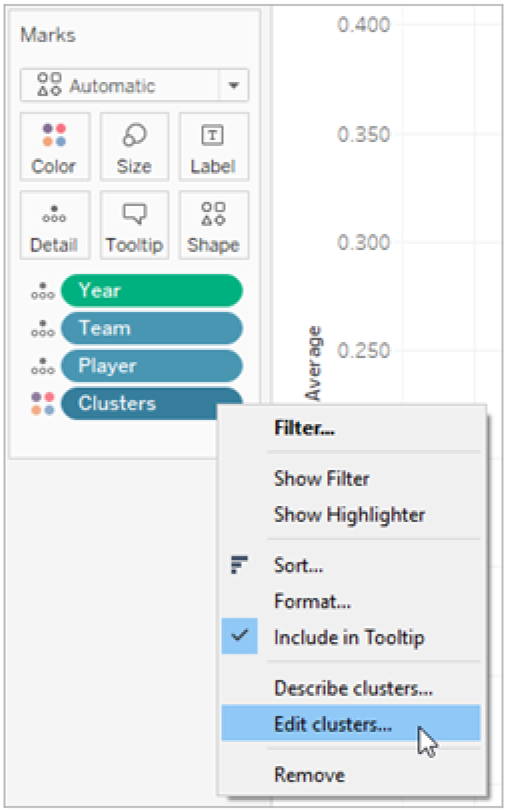
When any of those conditions apply, you will not be able to drag **Clusters** from the Analytics pane to the view.

In addition, the following field types cannot be used as variables (inputs) for clustering:

- Table calculations
- Blended calculations
- Ad-hoc calculations
- Generated latitude/longitude values
- Groups
- Sets
- Bins
- Parameters
- Dates
- Measure Names/Measure Values

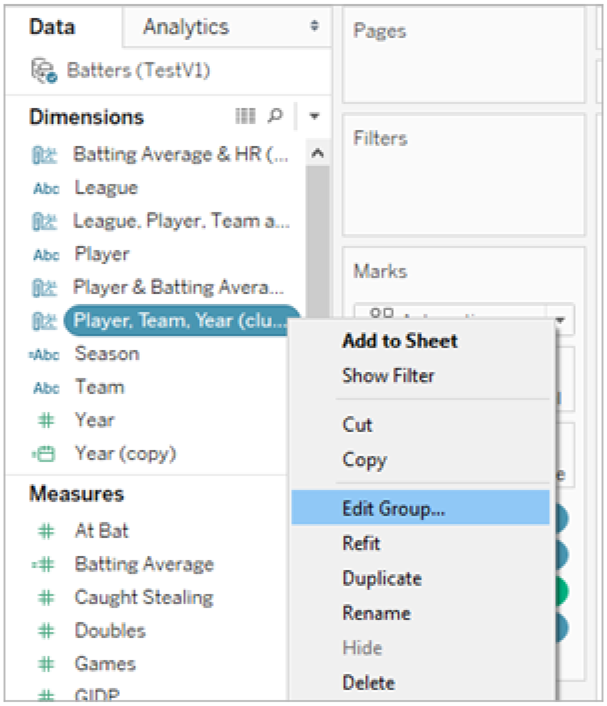
Edit clusters

To edit an existing cluster, right-click (Control-click on a Mac) a **Clusters** field on **Color** and select **Edit clusters**.

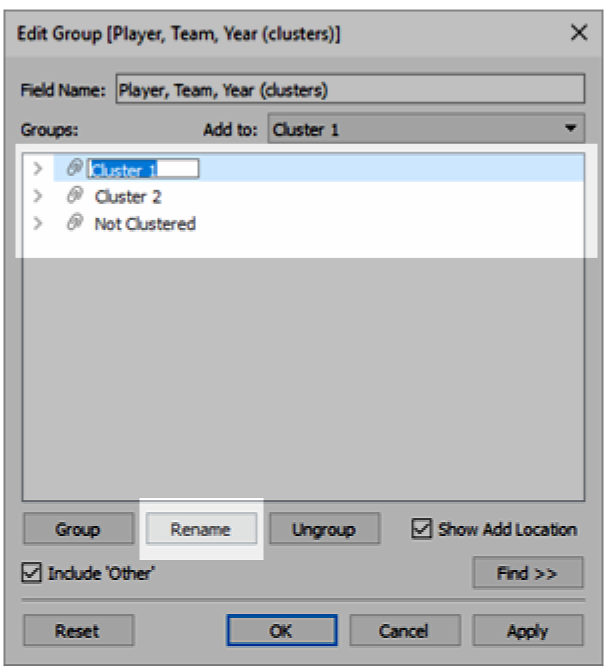


To change the names used for each cluster, you will first need to drag the Clusters field to the **Data** pane and save it as a group. For details, see **Create a group from cluster results** on page 2561.

Right-click the cluster group and select **Edit Group** to make changes to each cluster.



Select a cluster group in the list of Groups and click **Rename** to change the name.

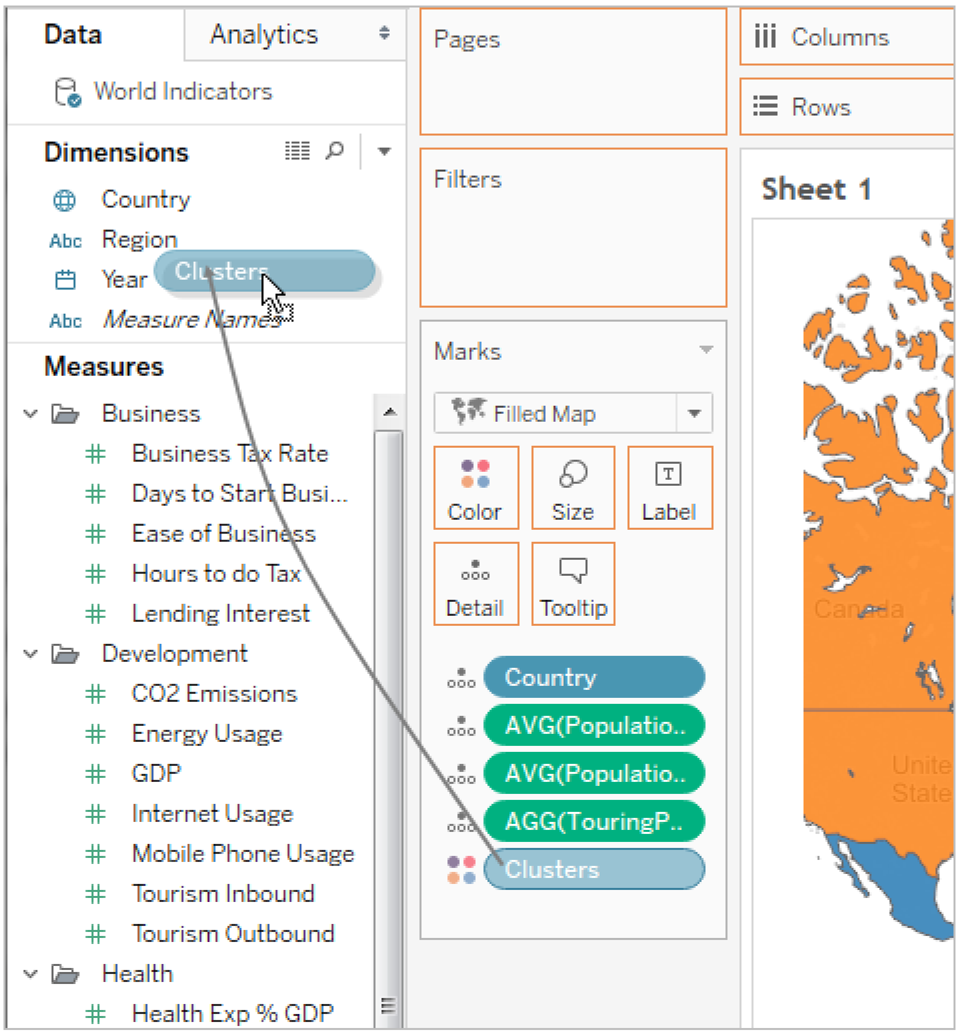


Create a group from cluster results

If you drag a cluster to the **Data** pane, it becomes a group dimension in which the individual members (Cluster 1, Cluster 2, etc.) contain the marks that the cluster algorithm has determined are more similar to each other than they are to other marks.

After you drag a cluster group to the **Data** pane, you can use it in other worksheets.

Drag **Clusters** from the **Marks** card to the **Data** pane to create a Tableau group:



After you create a group from clusters, the group and the original clusters are separate and distinct. Editing the clusters does not affect the group, and editing the group does not affect the cluster results. The group has the same characteristics as any other Tableau group. It is part of the data source. Unlike the original clusters, you can use the group in other worksheets in the workbook. So if you rename the saved cluster group, that renaming is not applied to the original clustering in the view. See **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1184.

Constraints on saving clusters as groups

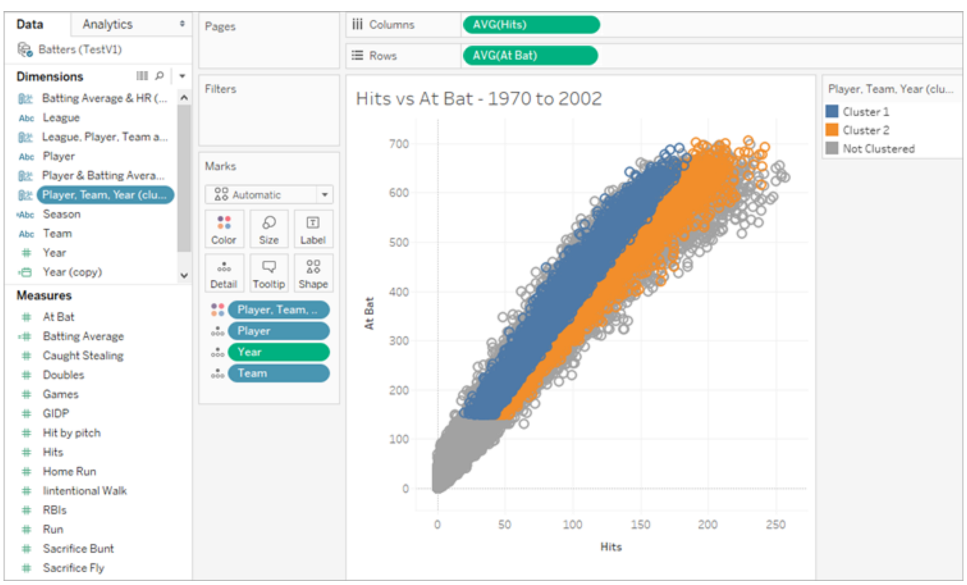
You will not be able to save Clusters to the **Data** pane under any of the following circumstances:

- When the measures in the view are disaggregated and the measures you are using as clustering variables are not the same as the measures in the view. For detials, see **How to Disaggregate Data** on page 350.
- When the Clusters you want to save are on the **Filters** shelf.
- When **Measure Names** or **Measure Values** is in the view.
- When there is a blended dimension in the view.

Refit saved clusters

When you save a Clusters field as a group, it is saved with its analytic model. You can use your cluster groups in other worksheets and workbooks, however, they don't automatically refresh.

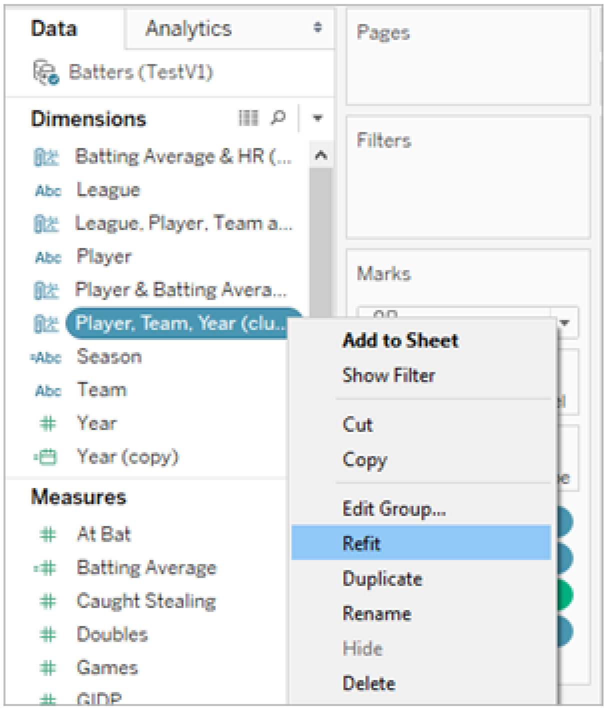
In this example, a saved cluster group and its analytic model has been applied to a different worksheet. As a result, some of the marks are not included in the clustering yet (indicated by gray marks).



If the underlying data changes, you can use the **Refit** option to refresh and recompute the data for a saved clusters group.

To refit a saved cluster

- Right-click a clusters group in the Data pane, and then click **Refit**.



Here's an example of updated clustering after refitting the saved cluster:

When you refit saved clusters, new clusters will be created and existing aliases for each cluster group category will be replaced with new, generic cluster aliases. Be aware that refitting saved clusters may alter your visualizations that use existing clusters and aliases.

How clustering works

Cluster analysis partitions the marks in the view into clusters, where the marks within each cluster are more similar to one another than they are to marks in other clusters. Tableau distinguishes clusters using color.

> **Note**: For additional insight into how clustering works in Tableau, see the blog post Understanding Clustering in Tableau 10.

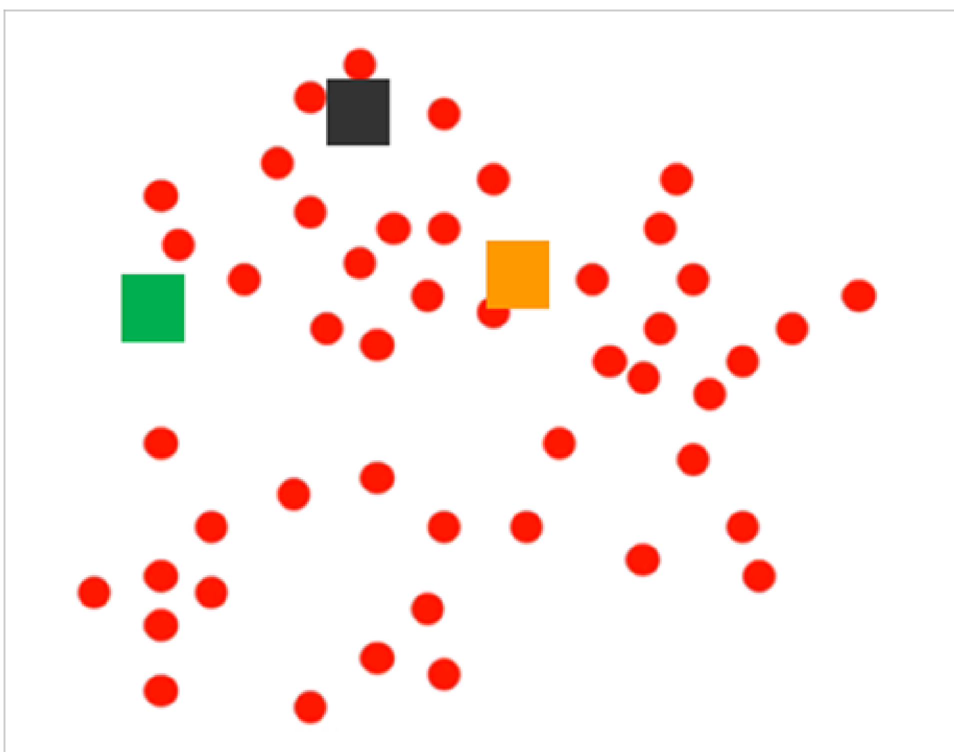
The clustering algorithm

Tableau uses the k-means algorithm for clustering. For a given number of clusters k, the algorithm partitions the data into k clusters. Each cluster has a center (centroid) that is the mean value of all the points in that cluster. K-means locates centers through an iterative procedure that minimizes distances between individual points in a cluster and the cluster center. In Tableau, you can specify a desired number of clusters, or have Tableau test different values of k and suggest an optimal number of clusters (see **Criteria used to determine the optimal number of clusters** on page 2567).

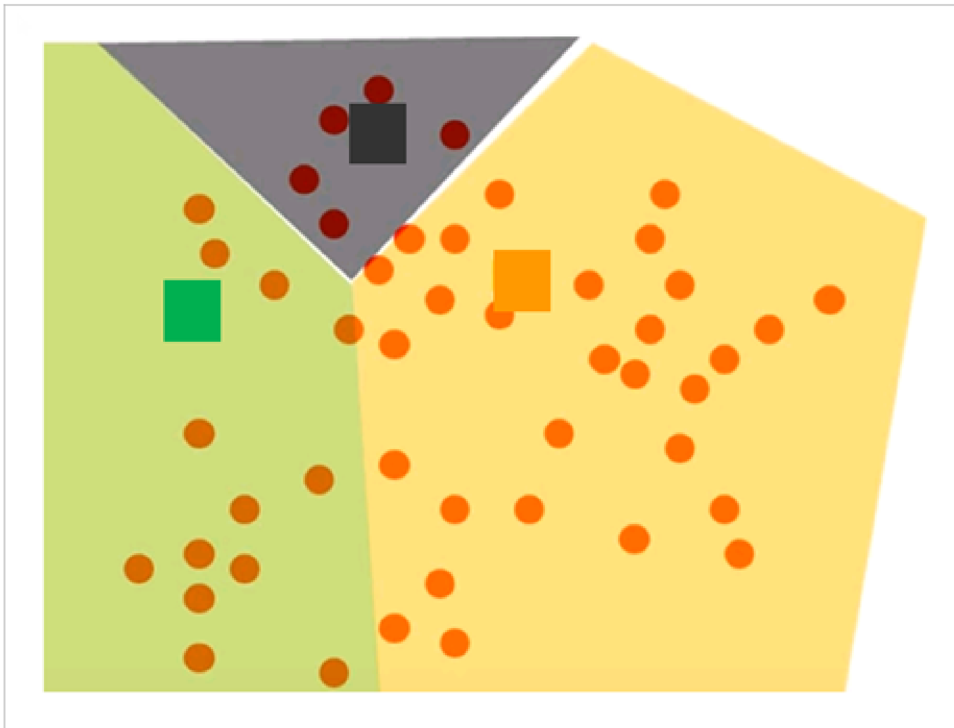
K-means requires an initial specification of cluster centers. Starting with one cluster, the method chooses a variable whose mean is used as a threshold for splitting the data in two. The centroids of these two parts are then used to initialize k-means to optimize the membership of the two clusters. Next, one of the two clusters is chosen for splitting and a variable within that cluster is chosen whose mean is used as a threshold for splitting that cluster in two. K-means is then used to partition the data into three clusters, initialized with the centroids of the two parts of the split cluster and the centroid of the remaining cluster. This process is repeated until a set number of clusters is reached.

Tableau uses Lloyd's algorithm with squared Euclidean distances to compute the k-means clustering for each k. Combined with the splitting procedure to determine the initial centers for each k > 1, the resulting clustering is deterministic, with the result dependent only on the number of clusters.

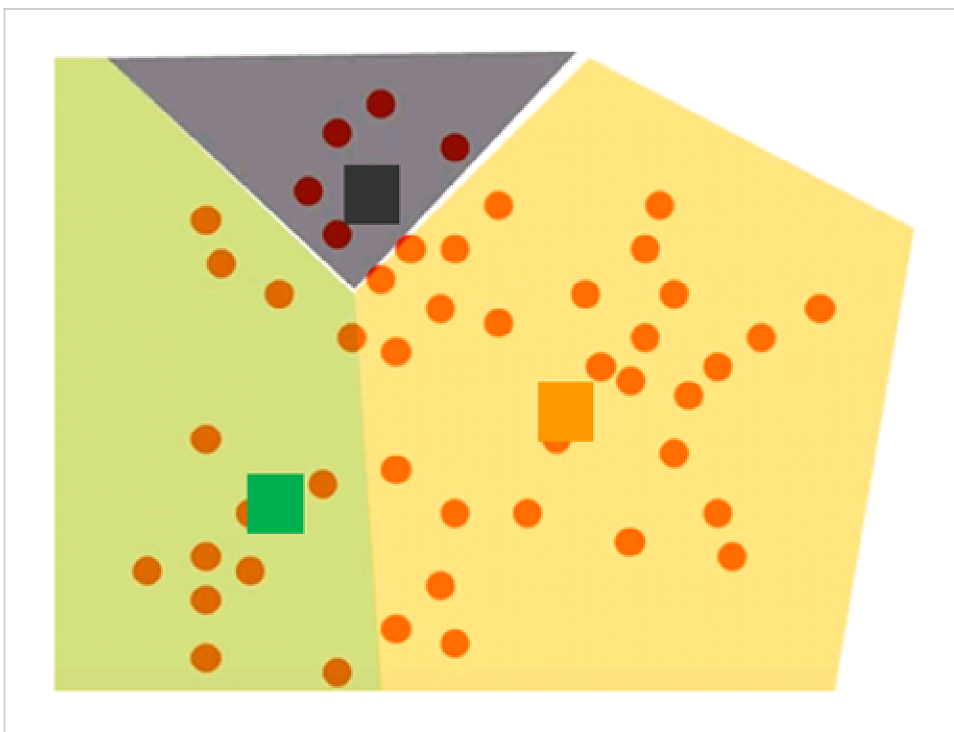
The algorithm starts by picking initial cluster centers:



It then partitions the marks by assigning each to its nearest center:



Then it refines the results by computing new centers for each partition by averaging all the points assigned to the same cluster:



It then reviews the assignment of marks to clusters and reassigns any marks that are now closer to a different center than before.

The clusters are redefined and marks are reassigned iteratively until no more changes are occurring.

Criteria used to determine the optimal number of clusters

Tableau uses the Calinski-Harabasz criterion to assess cluster quality. The Calinski-Harabasz criterion is defined as

$$\frac{SS_B}{SS_W} \times \frac{(N-k)}{(k-1)}$$

where SS_B is the overall between-cluster variance, SS_W the overall within-cluster variance, k the number of clusters, and N the number of observations.

The greater the value of this ratio, the more cohesive the clusters (low within-cluster variance) and the more distinct/separate the individual clusters (high between-cluster variance).

Since the Calinski-Harabasz index is not defined for k=1, it cannot be used to detect one-cluster cases.

If a user does not specify the number of clusters, Tableau picks the number of clusters corresponding to the first local maximum of the Calinski-Harabasz index. By default, k-means will be run for up to 25 clusters if the first local maximum of the index is not reached for a smaller value of k. You can set a maximum value of 50 clusters.

> **Note**: If a categorical variable (that is, a dimension) has more than 25 unique values, then Tableau will disregard that variable when computing clusters.

What values get assigned to the "Not Clustered" category?

When there are null values for a measure, Tableau assigns values for rows with null to a **Not Clustered** category. Categorical variables (that is, dimensions) that return * for ATTR (meaning that all values are not identical) are also not clustered.

Scaling

Tableau scales values automatically so that columns having a larger range of magnitudes don't dominate the results. For example, an analyst could be using inflation and GDP as input variables for clustering, but because GDP values are in trillions of dollars, this could cause the inflation values to be almost completely disregarded in the computation. Tableau uses a scaling method called *min-max normalization*, in which the values of each variable is mapped to a value between 0 and 1 by subtracting its minimum and dividing by its range.

Information on statistics models used for clusters

The Describe Clusters dialog box provides information about the models that Tableau computed for clustering. You can use these statistics to assess the quality of the clustering.

When the view includes clustering, you can open the Describe Clusters dialog box by right-clicking **Clusters** on the **Marks** card (Control-clicking on a Mac) and choosing **Describe Clusters**. The information in the Describe Clusters dialog box is read-only, though you can click **Copy to Clipboard** and then paste the screen contents into a writeable document.

Describe Clusters - Summary Tab

The Summary tab identifies the inputs that were used to generate the clusters and provides some statistics that characterize the clusters.

Inputs for Clustering

Variables

Identifies the fields Tableau uses to compute clusters. These are the fields listed in the Variables box in the Clusters dialog box.

Level of Detail

Identifies the fields that are contributing to the view's level of detail—that is, the fields that determine the level of aggregation. For details, see **How dimensions affect the level of detail in the view** on page 319.

Scaling

Identifies the scaling method used for pre-processing. Normalized is currently the only scaling method Tableau uses. The formula for this method, also known as min-max normalization, is `(x – min(x))/(max(x) - min(x))`.

Summary Diagnostics

Number of Clusters

The number of individual clusters in the clustering.

Number of Points

The number of marks in the view.

Between-group sum of squares

A metric quantifying the separation between clusters as a sum of squared distances between each cluster's center (average value), weighted by the number of data points assigned to the cluster, and the center of the data set. The larger the value, the better the separation between clusters.

Within-group sum of squares

A metric quantifying the cohesion of clusters as a sum of squared distances between the center of each cluster and the individual marks in the cluster. The smaller the value, the more cohesive the clusters.

Total sum of squares

Totals the between-group sum of squares and the within-group sum of squares. The ratio (between-group sum of squares)/(total sum of squares) gives the proportion of variance explained by the model. Values are between 0 and 1; larger values typically indicate a better model. However, you can increase this ratio just by increasing the number of clusters, so it could be misleading if you compare a five-cluster model with a three-cluster model using just this value.

Cluster Statistics

For each cluster in the clustering, the following information is provided.

Items

The number of marks within the cluster.

Centers

The average value within each cluster (shown for numeric items).

Most Common

The most common value within each cluster (only shown for categorical items).

Describe Clusters – Models Tab

Analysis of variance (ANOVA) is a collection of statistical models and associated procedures useful for analyzing variation within and between observations that have been partitioned into groups or clusters. In this case, analysis of variance is computed per variable, and the resulting analysis of variance table can be used to determine which variables are most effective for distinguishing clusters.

Relevant analysis of variance statistics for clustering include:

F-statistic

The F-statistic for one-way, or single-factor, ANOVA is the fraction of variance explained by a variable. It is the ratio of the between-group variance to the total variance.

The larger the F-statistic, the better the corresponding variable is distinguishing between clusters.

p-value

The p-value is the probability that the F-distribution of all possible values of the F-statistic takes on a value greater than the actual F-statistic for a variable. If the p-value falls below a specified significance level, then the null hypothesis (that the individual elements of the variable are random samples from a single population) can be rejected. The degrees of freedom for this F-distribution are (k - 1, N - k), where k is the number of clusters and N is the number of items (rows) clustered.

The lower the p-value, the more the expected values of the elements of the corresponding variable differ among clusters.

Model Sum of Squares and Degrees of Freedom

The Model Sum of Squares is the ratio of the between-group sum of squares to the model degrees of freedom. The between group sum of squares is a measure of the variation between cluster means. If the cluster means are close to each other (and therefore close to the overall mean), this value will be small. The model has k-1 degrees of freedom, where k is the number of clusters.

Error Sum of Squares and Degrees of Freedom

The Error Sum of Squares is the ratio of within-group sum of squares to the error degrees of freedom. The within-group sum-of-squares measures the variation between observations within each cluster. The error has N-k degrees of freedom, where N is the total number of observations (rows) clustered and k is the number of clusters.

The Error Sum of Squares can be thought of as the overall Mean Square Error, assuming that each cluster center represents the "truth" for each cluster.

Example: Create clusters using World Economic Indicators data

The Tableau clustering feature partitions marks in the view into clusters, where the marks within each cluster are more similar to one another than they are to marks in other clusters. This example shows how a researcher might use clustering to find an optimal set of marks (in this case, countries/regions) in a data source.

The objective

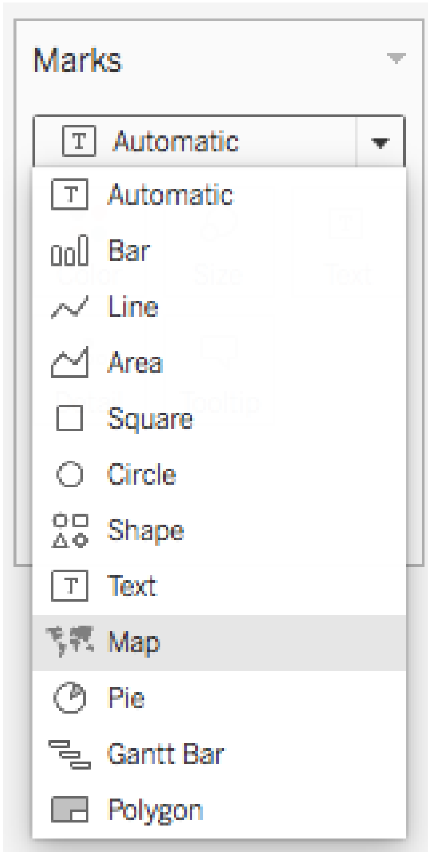
As life expectancy increases around the world, and as older people remain more active, senior tourism can be a lucrative market for companies that know how to find and appeal to potential customers. The World Indicators sample data set that comes with Tableau contains the kind of data that might help companies identify the countries or regions where there are enough of the right kind of customers.

Finding the right countries/regions

Here is an example of how Tableau clustering could help such a company identify the countries/regions where a senior tourism business could succeed. Imagine you are the analyst. Here is how you might proceed.

1. Open the **World Indicators** sample data source in Tableau Desktop.
2. Double-click **Country/Region** in the **Data** pane.

   Tableau automatically creates a map view, with a mark in each country/region.
3. On the **Marks** card, change the mark type to **Map**:



You should now see a map projection where all countries/regions are filled with a solid color:

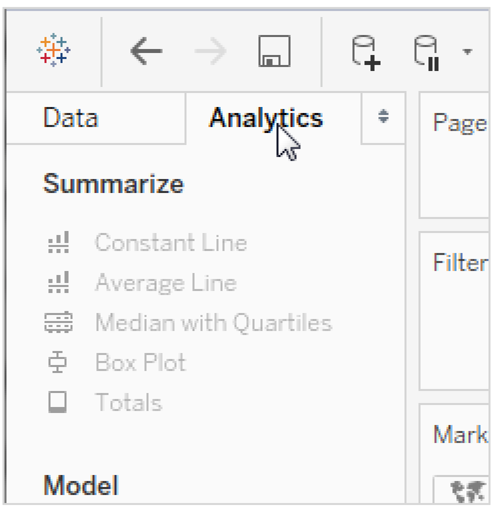
4. The next step is to identify the fields that you will use as variables for clustering. Here are the fields you choose:

Field	Reason for inclusion
Life Expectancy Female and **Life Expectancy Male**	Where people are living longer, there are more likely to be people who are interested in traveling later in life.
Population Urban	It is easier to market services in areas with greater population density.
Population 65+	The target population is older residents with the time and funds to travel.
TourismPerCapita	This is a measure that you must create as a named calculated field. The formula is: `SUM([Tourism Outbound])/SUM([Population Total])`

> **Tourism Outbound** aggregates the money (in US dollars) that residents of a country/region spend annually on international travel. But this total must be divided by the population of each country/region to determine the average amount each resident spends on international travel.

There is no guarantee that these are the ideal fields to choose, or that these fields will produce cluster results that are clear and unambiguous. Clustering is an iterative process—experimentation leads to discovery which leads, in turn, to more experimentation.

5. Drag these five fields from the **Data** pane to **Detail** on the **Marks** card.
6. Click to open the **Analytics** pane:



7. Drag **Cluster** from the **Analytics** pane and drop it in the view:

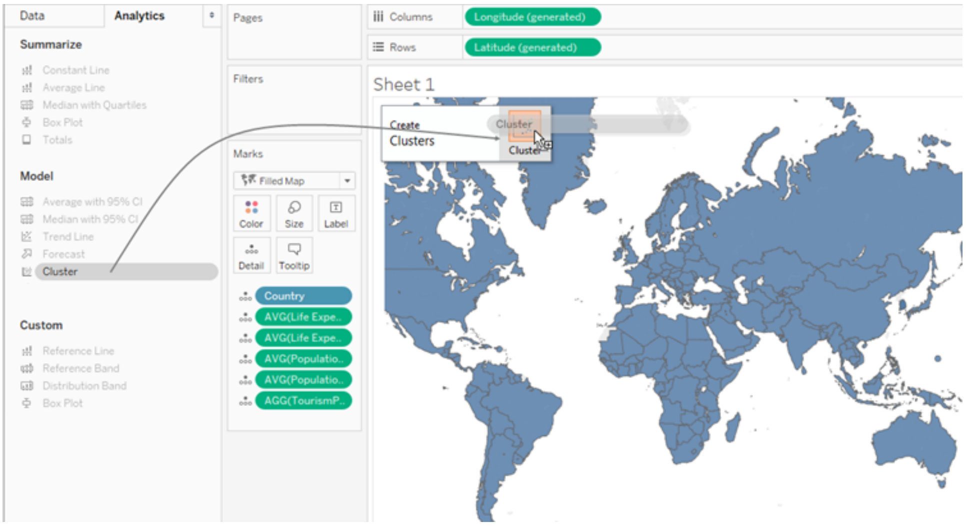
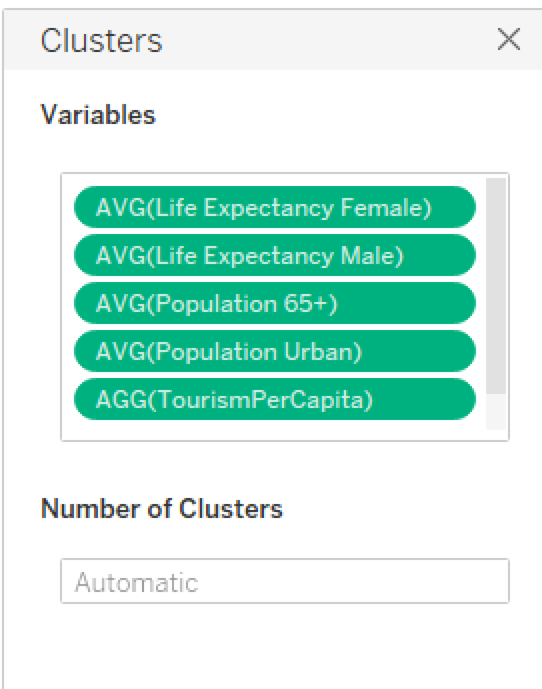
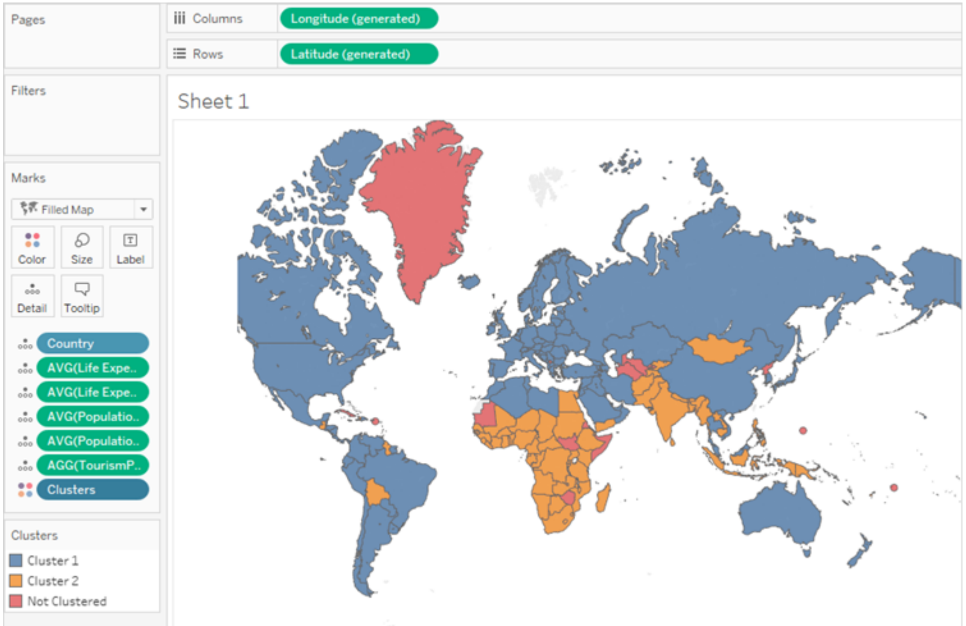


Tableau displays the Clusters dialog box and adds the measures in the view to the list of variables:



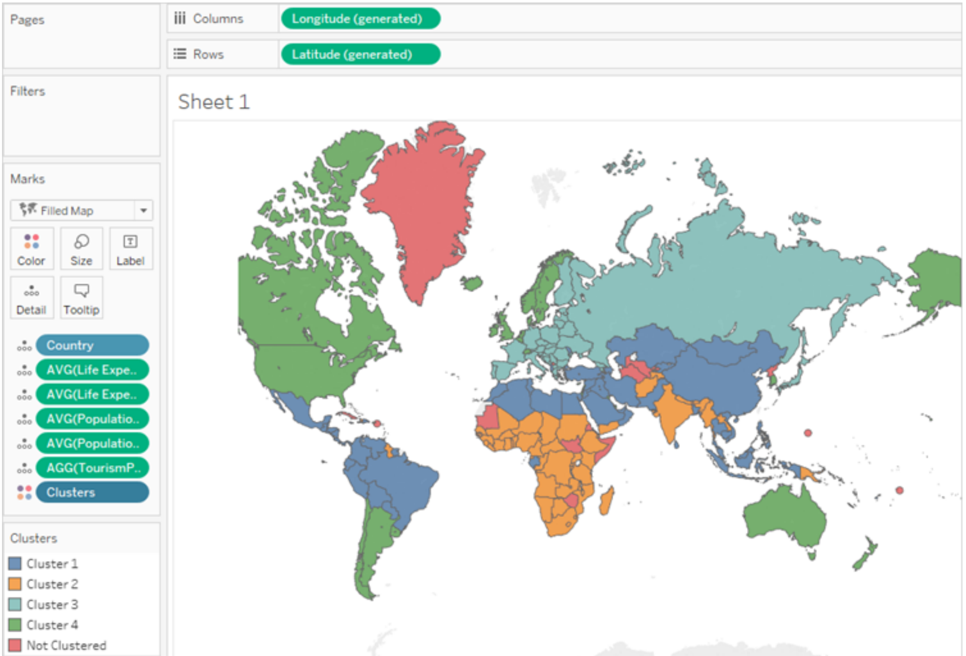
It also updates the view by adding clusters to **Color**. In this case, Tableau finds two distinct clusters, and is unable to assign certain countries/regions (colored reddish-pink) to either cluster:



> **Note**: See **How clustering works** on page 2564 for details on data that Tableau assigns to "Not clustered."

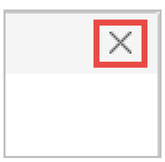
8. You decide that two clusters isn't enough—you don't have the resources to set up shop in half the countries/regions in the world. So you type `4` in the **Number of Clusters** field in the Clusters dialog box.

   The map becomes more interesting:



But how do these clusters relate to the variables you have chosen? Which one correlates best with the factors that support senior tourism? It's time to look at the statistics behind the clusters.

9. Close the Clusters dialog box by clicking the X in its upper-right corner:



10. Click the **Clusters** field on the **Marks** card and choose **Describe Clusters**.

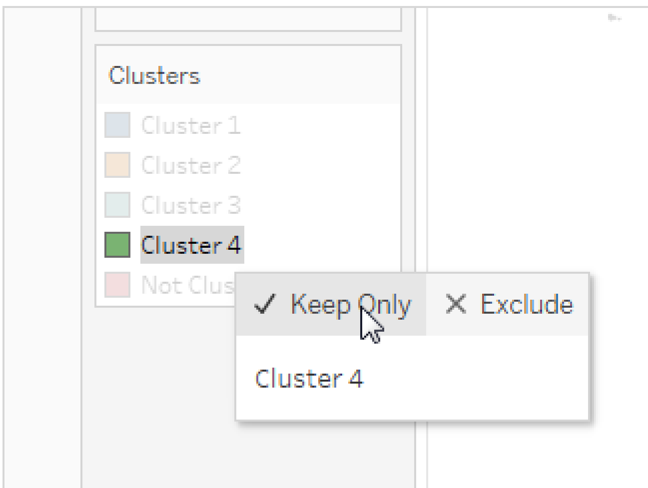
The table at the bottom of the **Models** tab in the Describe Clusters dialog box shows the average value for each variable in each cluster:

		Centers				
Clusters	Number of Items	Avg. Life Expectancy Female	Avg. Life Expectancy Male	Avg. Population 65+	Avg. Population Urban	TourismPerCapita
Cluster 1	69	74.216	69.003	0.054943	0.5325	197.12
Cluster 2	52	56.75	54.429	0.031889	0.33127	20.775
Cluster 3	29	79.164	71.706	0.15493	0.65532	320.92
Cluster 4	30	81.287	76.355	0.11606	0.87123	1360.4
Not Clustered	28					

Cluster 4 has the highest life expectancy (both male and female), the highest concentration of urban population, and the highest expenditure for international tourism: $1360.40 per capita. The only variable for which Cluster 4 does not have the highest value is **Population 65+**, where Cluster 3 has the advantage: 0.15493 (just under 16%) to 0.11606 (just over 11%) in Cluster 4.

The clustering algorithm does not know whether you are looking for the maximum value for these variables, the minimum value, or something in the middle—it just looks for correlation. But you know that higher values for these variables is the signal you're looking for, and Cluster 4 is the best choice.

11. You could attempt to pick out the Cluster 4 countries/regions from the map, but there is an easier way. Close the **Describe Clusters** dialog box and then click **Cluster 4** on the Color legend and choose **Keep Only**.



12. Choose Text Table from ShowMe.

You now see a list of the countries/regions in Cluster 4:

Clusters	Country	
Cluster 4	Argentina	Abc
	Australia	Abc
	Bahamas, The	Abc
	Belgium	Abc
	Canada	Abc
	Chile	Abc
	Curacao	Abc
	Cyprus	Abc
	Denmark	Abc
	Hong Kong SAR, China	Abc
	Iceland	Abc
	Ireland	Abc
	Israel	Abc
	Korea, Rep.	Abc
	Kuwait	Abc
	Lebanon	Abc
	Luxembourg	Abc
	Macao SAR, China	Abc
	Malta	Abc
	Netherlands	Abc
	New Zealand	Abc
	Norway	Abc
	Puerto Rico	Abc
	Qatar	Abc
	Singapore	Abc
	Sweden	Abc
	Switzerland	Abc
	United Kingdom	Abc
	United States	Abc
	Uruguay	Abc

This list is not the end of the process. You might try clustering again with a somewhat different set of variables and maybe a different number of clusters, or you might add some countries/regions to the list and remove others, based on other factors. For example, if your tours are mostly to tropical locales, you might remove countries/regions like Curacao and the Bahamas from the list, because tropical tours might not appeal to residents of those countries/regions.

Another option is to filter your data before you re-cluster, to only show countries/regions with populations above a certain threshold, or to target countries/regionss in a particular geographical area.

Forecasting

You can add a forecast to a view when there is at least one date dimension and one measure in the view. If you're interested in predictive modeling, also available in Tableau, see **How Predictive Modeling Functions Work in Tableau** on page 2606.

To turn forecasting on, right-click (control-click on Mac) on the visualization and choose **Forecast >Show Forecast**, or choose **Analysis >Forecast >Show Forecast**.

When no date dimension is present, you can add a forecast if there is a dimension field in the view that has integer values. For details, see **Forecasting When No Date is in the View** on page 2591.

You can forecast quantitative time-series data using exponential smoothing models in Tableau Desktop. With exponential smoothing, recent observations are given relatively more weight than older observations. These models capture the evolving trend or seasonality of your data and extrapolate them into the future. Forecasting is fully automatic, yet configurable. Many forecast results can become fields in your visualizations.

When a forecast is showing, future values for the measure are shown next to the actual values.

Watch a video: To see related concepts demonstrated in Tableau, watch Forecasting, a 6-minute free training video. Use your tableau.com account to sign in.

Forecasting Constraints

Forecasting is not supported for Multidimensional data sources. In Tableau Desktop, multidimensional data sources are supported only in Windows.

You can publish a view that contains a forecast, and see the forecast when you view or edit the view on the web, but you cannot modify or add a forecast when you are editing a view on the web.

In addition, you cannot add a forecast to a view if it contains any of the following:

- Table calculations
- Disaggregated measures

- Percent calculations
- Grand Totals or Subtotals
- Date values with aggregation set to Exact Date

How Forecasting Works in Tableau

Forecasting in Tableau uses a technique known as exponential smoothing. Forecast algorithms try to find a regular pattern in measures that can be continued into the future. If you're interested in predictive modeling, also available in Tableau, see **How Predictive Modeling Functions Work in Tableau** on page 2606.

> **Watch a video**: To see related concepts demonstrated in Tableau, watch Forecasting, a 6-minute free training video. Use your tableau.com account to sign in.

You typically add a forecast to a view that contains a date field and at least one measure. However, in the absence of a date, Tableau can create a forecast for a view that contains a dimension with integer values in addition to at least one measure.

For details on creating a forecast, see **Create a Forecast** on page 2587. For details on forecasting using an integer dimension, see **Forecasting When No Date is in the View** on page 2591.

Overview

All forecast algorithms are simple models of a real-world data generating process (DGP). For a high quality forecast, a simple pattern in the DGP must match the pattern described by the model reasonably well. Quality metrics measure how well the model matches the DGP. If the quality is low, the precision measured by the confidence bands is not important because it measures the precision of an inaccurate estimate.

Tableau automatically selects the best of up to eight models, the best being the one that generates the highest quality forecast. The smoothing parameters of each model are optimized before Tableau assesses forecast quality. The optimization method is global. Therefore, choosing locally optimal smoothing parameters that are not also globally optimal is not impossible. However, initial value parameters are selected according to best practices but are not further optimized. So it is possible for initial value parameters to be less than optimal. The eight models available in Tableau are among those described at the following location on the OTexts web site: A taxonomy of exponential smoothing methods.

When there is not enough data in the visualization, Tableau automatically tries to forecast at a finer temporal granularity, and then aggregates the forecast back to the granularity of the visualization. Tableau provides prediction bands which may be simulated or calculated from a closed form equation. All models with a multiplicative component or with aggregated forecasts have simulated bands, while all other models use the closed form equations.

Exponential Smoothing and Trend

Exponential smoothing models iteratively forecast future values of a regular time series of values from weighted averages of past values of the series. The simplest model, *Simple Exponential Smoothing*, computes the next level or smoothed value from a weighted average of the last actual value and the last level value. The method is exponential because the value of each level is influenced by every preceding actual value to an exponentially decreasing degree—more recent values are given greater weight.

Exponential smoothing models with trend or seasonal components are effective when the measure to be forecast exhibits trend or seasonality over the period of time on which the forecast is based. *Trend* is a tendency in the data to increase or decrease over time. *Seasonality* is a repeating, predictable variation in value, such as an annual fluctuation in temperature relative to the season.

In general, the more data points you have in your time series, the better the resulting forecast will be. Having enough data is particularly important if you want to model seasonality, because the model is more complicated and requires more proof in the form of data to achieve a reasonable level of precision. On the other hand, if you forecast using data generated by two or more different DGPs, you will get a lower quality forecast because a model can only match one.

Seasonality

Tableau tests for a seasonal cycle with the length most typical for the time aggregation of the time series for which the forecast is estimated. So if you aggregate by months, Tableau will look for a 12-month cycle; if you aggregate by quarters, Tableau will search for a four-quarter cycle; and if you aggregate by days, Tableau will search for weekly seasonality. Therefore, if there is a six-month cycle in your monthly time series, Tableau will probably find a 12-month pattern that contains two similar sub-patterns. However, if there is a seven-month cycle in your monthly time series, Tableau will probably find no cycle at all. Luckily, seven-month cycles are uncommon.

Tableau can use either of two methods for deriving season length. The original temporal method uses the natural season length of the temporal granularity (TG) of the view. Temporal granularity means the finest unit of time expressed by the view. For example, if the view contains either a continuous green date truncated to month or discrete blue year and month date parts, the temporal granularity of the view is month. The new non-temporal method, introduced with Tableau 9.3, uses periodic regression to check season lengths from 2 to 60 for candidate lengths.

Tableau automatically selects the most appropriate method for a given view. When Tableau is using a date to order the measures in a view, if the temporal granularity is quarterly, monthly, weekly, daily or hourly, the season lengths are almost certainly 4, 12, 13, 7 or 24, respectively. So only the length natural to the TG is used to construct the five seasonal exponential smoothing models supported by Tableau. The AIC of the five seasonal models and the three non-seasonal models are compared and the lowest returned. (For an explanation of the AIC metric, see Forecast Descriptions.)

When Tableau is using an integer dimension for forecasting, the second method is used. In this case there is no temporal granularity (TG), so potential season lengths must be derived from the data.

The second method is also used if the temporal granularity is yearly. Yearly series rarely have seasonality, but, if they do, it must also be derived from the data.

The second method is also used for views with temporal granularity of minute or second. If such series have seasonality, the season lengths are likely 60. However, when measuring a regular real world process, the process may have a regular repetition which does not correspond to the clock. So, for minutes and seconds, Tableau also checks for a length different from 60 in the data. This does not mean that Tableau can model two different season lengths at the same time. Rather, ten seasonal models are estimated, five with a season length of 60 and another five with the season length derived from the data. Whichever of the ten seasonal models or three non-seasonal models has the lowest AIC, that model is used to compute the forecast.

For series ordered by year, minute, or second, a single season length from the data is tested if the pattern is fairly clear. For integer ordered series, up to nine somewhat less clear potential season lengths are estimated for all five seasonal models, and the model with the lowest AIC is returned. If there are no likely season length candidates, only the non-seasonal models are estimated.

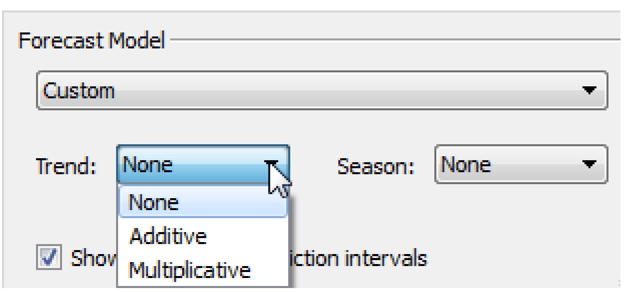
Since all selection is automatic when Tableau is deriving potential season lengths from the data, the default Model Type of “Automatic” in the Forecast Options Dialog Model Type menu does not change. Selecting “Automatic without seasonality” improves performance by eliminating all season length searching and estimation of seasonal models.

The heuristic that Tableau uses to decide when to use season lengths derived from the data depends on the distribution of errors for the periodic regression of each candidate season length. Since the assembly of season length candidates by periodic regression usually produces one or two clear winning lengths if seasonality actually exists in the data, the return of a single candidate indicates likely seasonality. In this case, Tableau estimates seasonal models with this candidate for year, minute and second granularity. The return of less than the maximum of ten candidates indicates possible seasonality. In this case, Tableau estimates seasonal models with all returned candidates for integer ordered views. The return of the maximum number of candidates indicates that errors for most length are similar. Therefore, the existence of any seasonality is unlikely. In this case, Tableau estimates only non-seasonal models for an integer-ordered or yearly ordered series, and only the seasonal models with a natural season length for other temporally ordered views.

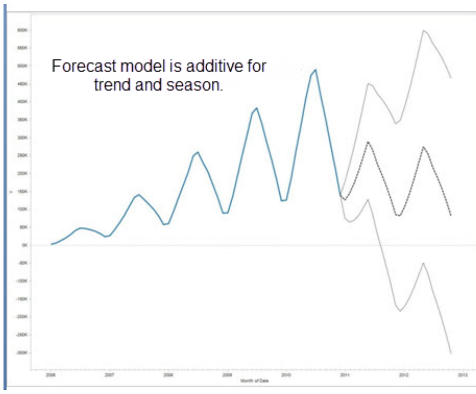
For Model Type “Automatic” in integer-, year-, minute- and second-ordered views, candidate season lengths are always derived from the data whether or not they are used. Since model estimation is much more time consuming than periodic regression, the performance impact should be moderate.

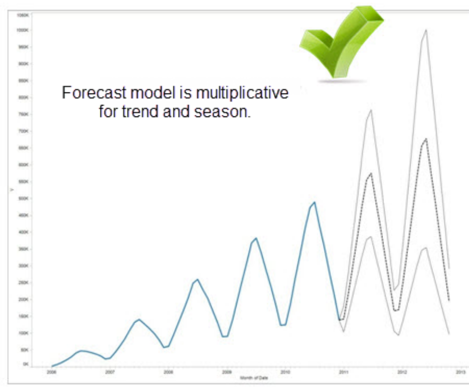
Model Types

In the Forecast Options dialog box, you can choose the model type Tableau users for forecasting. The **Automatic** setting is typically optimal for most views. If you choose **Custom** , then you can specify the trend and season characteristics independently, choosing either **None**, **Additive**, or **Multiplicative**:



An additive model is one in which the contributions of the model components are summed, whereas a multiplicative model is one in which at least some component contributions are multiplied. Multiplicative models can significantly improve forecast quality for data where the trend or seasonality is affected by the level (magnitude) of the data:





Keep in mind that you do not need to create a custom model to generate a forecast that is multiplicative: the **Automatic** setting can determine if a multiplicative forecast is appropriate for your data. However, a multiplicative model cannot be computed when the measure to be forecast has one or more values that are less than or equal to zero.

Forecasting with Time

When you are forecasting with a date, there can be only one base date in the view. Part dates are supported, but all parts must refer to the same underlying field. Dates can be on **Rows**, **Columns**, or **Marks** (with the exception of the Tooltip target).

Tableau supports three types of dates, two of which can be used for forecasting:

- Truncated dates reference a particular point in history with specific temporal granularity, such as February 2017. They are usually continuous, with a green background in the view. Truncated dates are valid for forecasting.
- Date parts refer to a particular member of a temporal measure such as February. Each date part is represented by a different, usually discrete field (with a blue background). Forecasting requires at least a Year date part. Specifically, it can use any of the following sets of date parts for forecasting:
  - *Year*
  - *Year + quarter*
  - *Year + month*
  - *Year + quarter + month*
  - *Year + week*
  - *Custom: Month/Year, Month/Day/Year*

Other date parts, such as *Quarter* or *Quarter* + *month*, are not valid for forecasting. See **Convert Fields between Discrete and Continuous** on page 1164 for more details about different date types.

- Exact dates refer to a particular point in history with maximum temporal granularity such as February 1, 2012 at 14:23:45.0. Exact dates are invalid for forecasting.

It is also possible to forecast without a date. See **Forecasting When No Date is in the View** on page 2591.

Granularity and Trimming

When you create a forecast, you select a date dimension that specifies a unit of time at which date values are to be measured. Tableau dates support a range of such time units, including Year, Quarter, Month, and Day. The unit you choose for the date value is known as the *granularity* of the date.

The data in your measure typically does not align precisely with your unit of granularity. You might set your date value to quarters, but your actual data may terminate in the middle of a quarter—for example, at the end of November. This can cause a problem because the value for this fractional quarter is treated by the forecasting model as a full quarter, which will typically have a lower value than a full quarter would. If the forecasting model is allowed to consider this data, the resulting forecast will be inaccurate. The solution is to trim the data, such that the trailing periods that could mislead the forecast are ignored. Use the Ignore Last option in the Forecast Options dialog box to remove—or *trim*—such partial periods. The default is to trim one period.

Getting More Data

Tableau requires at least five data points in the time series to estimate a trend, and enough data points for at least two seasons or one season plus five periods to estimate seasonality. For example, at least nine data points are required to estimate a model with a four quarter seasonal cycle (4 + 5), and at least 24 to estimate a model with a twelve month seasonal cycle (2 * 12).

If you turn on forecasting for a view that does not have enough data points to support a good forecast, Tableau can sometimes retrieve enough data points to produce a valid forecast by querying the datasource for a finer level of granularity:

- If your view contains fewer than nine years of data, by default, Tableau will query the data source for quarterly data, estimate a quarterly forecast, and aggregate to a yearly forecast to display in your view. If there are still not enough data points, Tableau will estimate a monthly forecast and return the aggregated yearly forecast to your view.

- If your view contains fewer than nine quarters of data, by default Tableau will estimate a monthly forecast and return the aggregated quarterly forecast results to your view.
- If your view contains fewer than nine weeks of data, by default, Tableau will estimate a daily forecast and return the aggregated weekly forecast results to your view.
- If your view contains fewer than nine days of data, by default, Tableau will estimate an hourly forecast and return the aggregated daily forecast results to your view.
- If your view contains fewer than nine hours of data, by default, Tableau will estimate an minutely forecast and return the aggregated hourly forecast results to your view.
- If your view contains fewer than nine minutes of data, by default, Tableau will estimate an secondly forecast and return the aggregated minutely forecast results to your view.

These adjustments happen behind the scene and require no configuration. Tableau does not change the appearance of your visualization, and does not actually change your date value. However, the summary of the forecast time period in the Forecast Describe and Forecast Options dialog will reflect the actual granularity used.

Tableau can only get more data when the aggregation for the measure you are forecasting is SUM or COUNT. See **Data Aggregation in Tableau** on page 341 for information on available aggregation types and information on how to change the aggregation type.

Create a Forecast

To create a forecast, your view must be using at least one date dimension and one measure.

To turn forecasting on, right-click (control-click on Mac) on the visualization and choose **Forecast >Show Forecast**, or choose **Analysis >Forecast >Show Forecast**.

> **Watch a video**: To see related concepts demonstrated in Tableau, watch Forecasting, a 6-minute free training video. Use your tableau.com account to sign in.

Each of the following examples indicate the structure that supports creating a forecast.

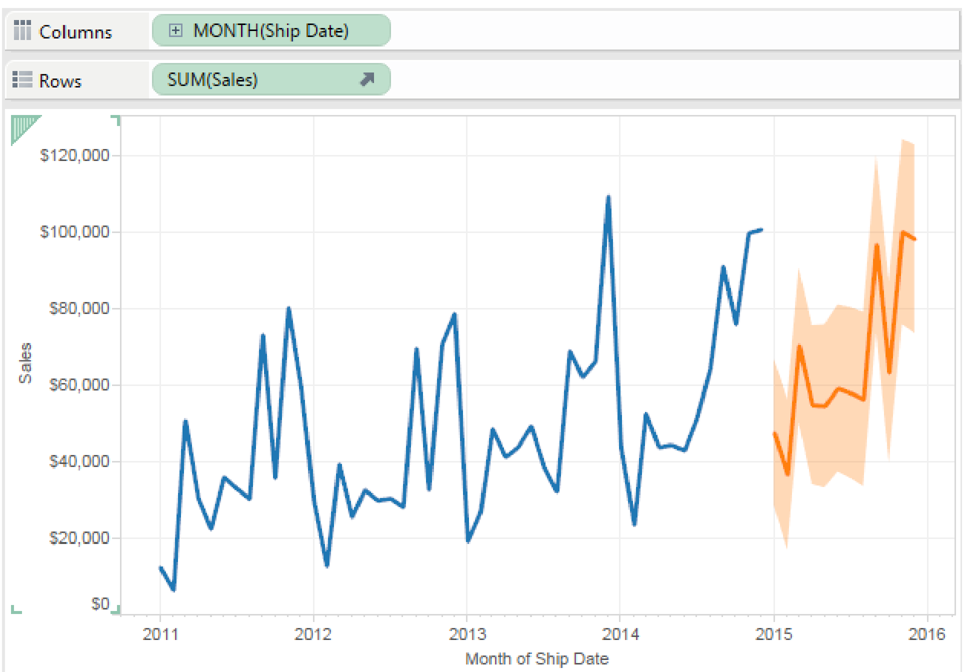
- The field you want to forecast is on the **Rows** shelf and a continuous date field is on the **Columns** shelf.
- The field you want to forecast is on the **Columns** shelf and a continuous date field is on

the **Rows** shelf.

- The field you want to forecast on either the **Rows** or **Columns** shelf, and discrete dates are on either the **Rows** or **Columns** shelf. At least one of the included date levels must be Year.
- The field you want to forecast is on the Marks card, and a continuous date or discrete date set is on **Rows**, **Columns** or **Marks**.

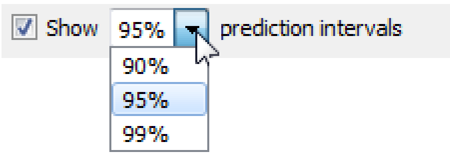
Note: You can also create a forecast when no date dimension is present if there is a dimension in the view that has integer values. See **Forecasting When No Date is in the View** on page 2591.

With forecasting on, Tableau visualizes estimated future values of the measure, in additional to actual historical values. The estimated values are shown by default in a lighter shade of the color used for the historical data:



Prediction Intervals

The shaded area in the image above shows the 95% prediction interval for the forecast. That is, the model has determined that there is a 95% likelihood that the value of sales will be within the shaded area for the forecast period. You can configure the confidence level percentile for the prediction bands, and whether prediction bands are included in the forecast, using the **Show prediction intervals** setting in the Forecast Options dialog box:

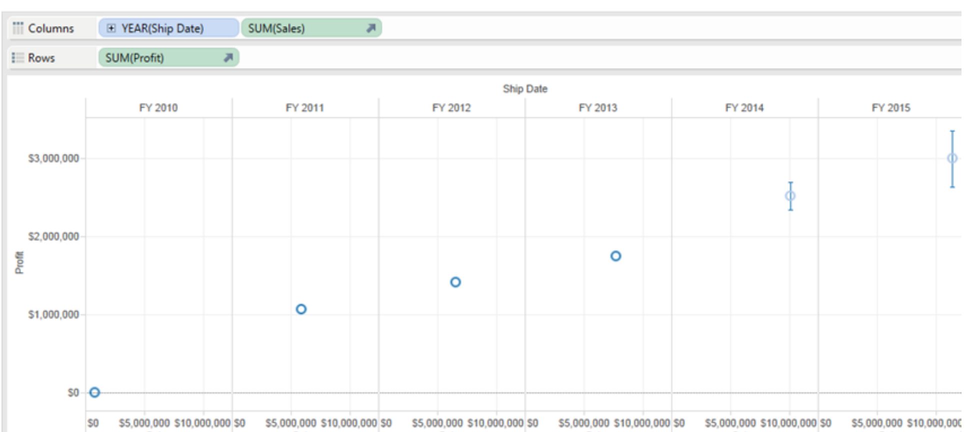


Clear the check box if you do not want to display prediction bands in forecasts. To set the prediction interval, select one of the values or enter a custom value. The lower the percentile you set for the confidence level, the narrower the prediction bands will be.

How your prediction intervals are displayed depends on the mark type of your forecasted marks:

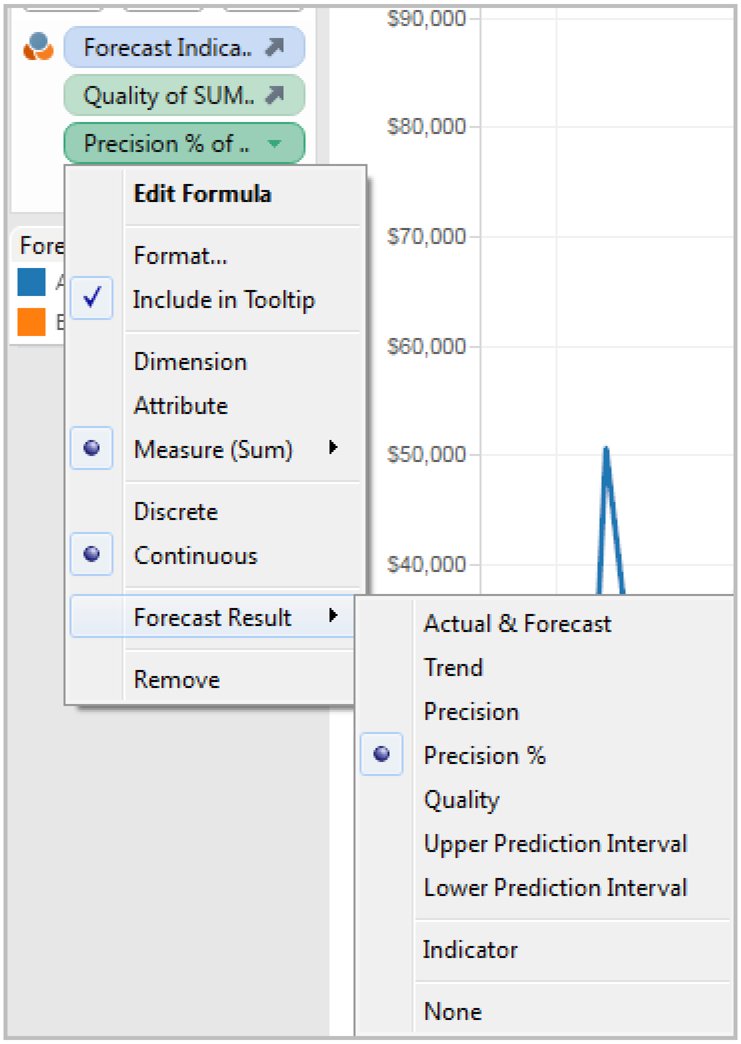
Forecast mark type	Prediction intervals displayed using
Line	Bands
Shape, square, circle, bar, or pie	Whiskers

In the following example, forecast data is indicated by lighter shaded circles, and the prediction intervals are indicated by lines ending in whiskers:



Enhancing Forecasts

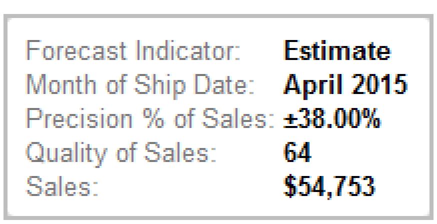
For each forecast value, consider verifying the quality or precision of your forecast by dragging another instance of the forecast measure from the **Data** pane to the Detail shelf on the Marks card and then after right-clicking the field to open the content menu, choosing one of the available options:



For descriptions of these options, see **Forecast Field Results** on the facing page.

You can repeat the process to add additional result types for each forecast value. See "Changing the Forecast Result" type in **Forecast Field Results** on the facing page for information on changing the result type.

By adding such result types to the Details shelf, you add information about the forecast to tooltips for all marks that are based on forecasted data.

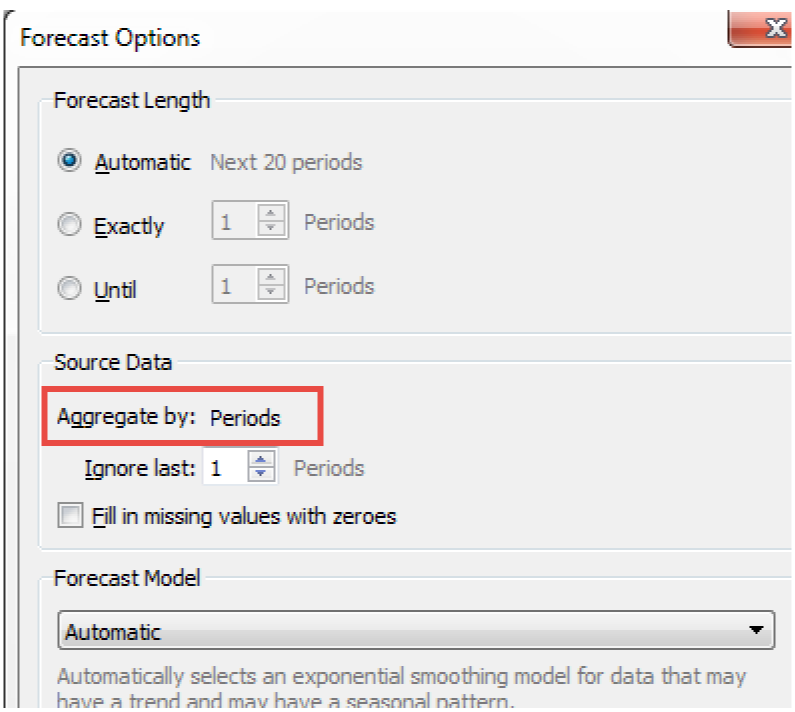


Forecasting When No Date is in the View

If a valid date is not in the view, Tableau will look for a dimension in the view that has integer values. If it finds such a dimension it will use that to forecast additional values for measures in the view. As with a date, when an integer dimension is selected to order the measures to be forecast, it is no longer used to partition the data. If there is more than one such integer dimension, Tableau will go in this order:

- An integer dimension on the Columns shelf. If there is more than one such dimension, it will use the first one (farthest to the left on the shelf).
- An integer dimension on the Rows shelf.
- An integer dimension on the Pages shelf.
- An integer dimension on the Marks card.

When Tableau is using an integer dimension to forecast, the Forecast Option and Forecast Description dialog boxes will automatically specify that forecasting is aggregating by periods:



Forecast Field Results

Tableau provides several types of forecast results. To view these result types in the view, right-click (control-click on Mac) on the measure field, choose **Forecast Result**, and then choose one of the options.

The options are:

- **Actual & Forecast**—Show the actual data extended by forecasted data.
- **Trend**—Show the forecast value with the seasonal component removed.
- **Precision**—Show the prediction interval distance from the forecast value for the configured confidence level.
- **Precision %**—Show precision as a percentage of the forecast value.
- **Quality**—Show the quality of the forecast, on a scale of 0 (worst) to 100 (best). This metric is scaled MASE, based on the MASE (Mean Absolute Scaled Error) of the forecast, which is the ratio of forecast error to the errors of a naïve forecast which assumes that the value of the current period will be the same as the value of the next period. The actual equation used for quality is:

  $$100 * \max(1 - MASE, 0)$$

  The Quality for a naïve forecast would be 0. The advantage of the MASE metric over the more common MAPE is that MASE is defined for time series which contain zero, wheras MAPE is not. In addition, MASE weights errors equally while MAPE weights positive and/or extreme errors more heavily.
- **Upper Prediction Interval**—Shows the value above which the true future value will lie confidence level percent of the time assuming a high quality model. The confidence level percentage is controlled by the Prediction Interval setting in the Forecast Options dialog box. See **Configure Forecast Options** on page 2594.
- **Lower Prediction Interval**—Shows 90, 95, or 99 confidence level below the forecast value. The actual interval is controlled by the **Prediction Interval** setting in the Forecast Options dialog box.
- **Indicator**—Show the string **Actual** for rows that were already on the worksheet when forecasting was inactive and **Estimate** for rows that were added when forecasting was activated.
- **None**—Do not show forecast data for this measure.

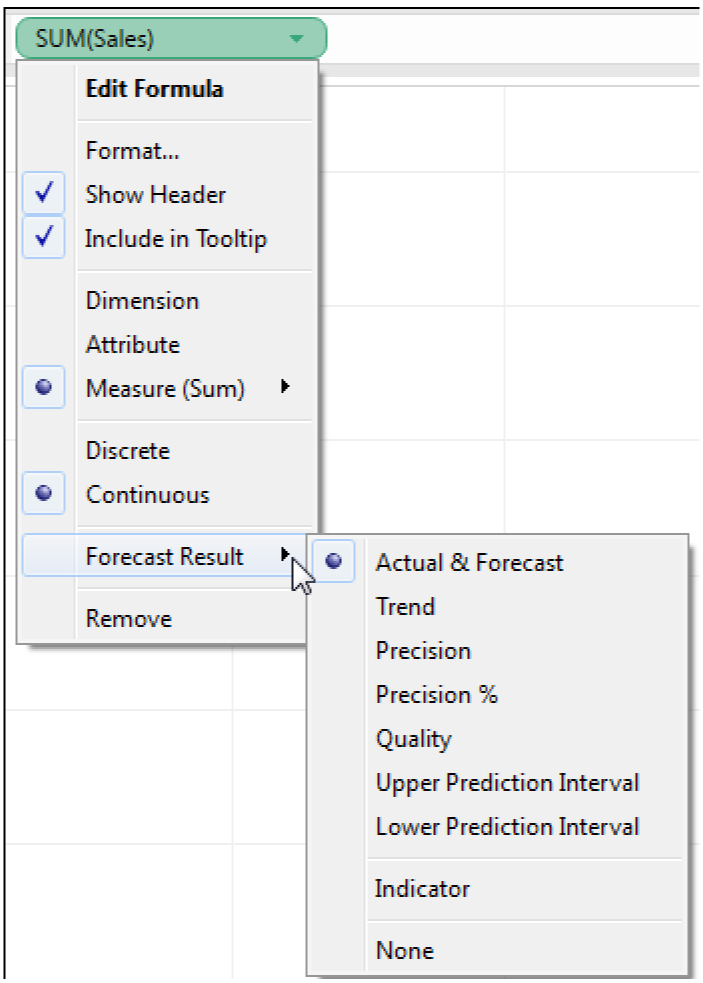
Forecast description information is also included in the worksheet description. See **Get Details About Fields and Sheets in a Workbook** on page 2732.

Forecasting a New Measure

When you add a new measure to a visualization that already has forecasting enabled, Tableau attempts to forecast future values.

Changing the Forecast Result Type

To change the forecast result type for a measure, right-click (control-click on Mac) on the measure field, select **Forecast Result**, and then choose a result type.



Configure Forecast Options

Use the Forecast Options dialog box to configure forecast options, including:

- The length of the forecast
- The range and temporal aggregation of source data from which to generate the forecast
- The forecast model
- Prediction intervals

When forecasting is enabled, you can open the Forecast Options dialog box by choosing **Analysis > Forecast > Forecast Options**.

Forecast Length

The **Forecast Length** section determines how far into the future the forecast extends. Select one of the following:

- **Automatic**: Tableau determines the forecast length based on the data.
- **Exactly**: Extends the forecast for the specified number of units.
- **Until**: Extends the forecast to the specified point in the future.

Source Data

Use the **Source Data** section to specify.

- **Aggregate by**: Specifies the temporal granularity of the time series. With the default value (**Automatic**), Tableau chooses the best granularity for estimation. This will typically match the temporal granularity of the visualization (that is, the date dimension that the forecast is based on). However, it is sometimes possible and desirable to estimate the forecast model at a finer granularity than the visualization when the time series in the visualization is too short to allow estimation.

  > **Note**: When you are using an integer dimension instead of a date dimension for forecasting, the Aggregate by value is always Periods. See **Forecasting When No Date is in the View** on page 2591.

- **Ignore last**: Specifies the number of periods at the end of the actual data that should be ignored in estimating the forecast model. Forecast data is used instead of actual data for these time periods. Use this feature to trim off unreliable or partial trailing periods which could mislead the forecast. When the estimation granularity specified in the **Source Data** section is finer than in the visualization, the trimmed periods are estimation periods. As a result, the trailing actual visualization period may become a forecast period, which is an aggregate of both actual and forecast periods of estimation granularity. In contrast, null values are not filled with zeros and must be filtered to allow forecast.

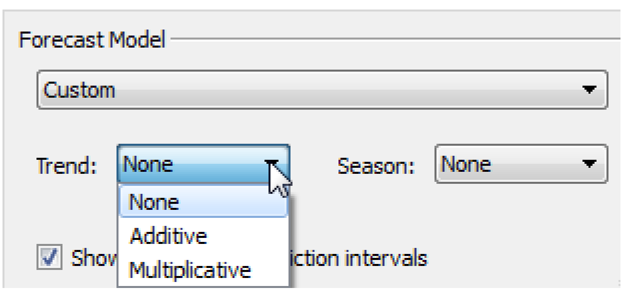
- **Fill in missing values with zeros**: If there are missing values in the measure you are attempting to forecast, you can specify that Tableau fill in these missing values with zero.

Forecast Model

The **Forecast Model** section specifies how the forecast model is to be produced.

Use the drop down to specify whether Tableau selects what it determines to be the best of all models (**Automatic**), the best of those with no seasonal component (**Automatic without seasonality**), or the model you specify (**Custom**).

When you choose the **Custom** option, two new fields appear in the Forecast Options dialog box, which you use to specify the trend and season characteristics for your model:



The choices are the same for both fields:

- **None**: When you select **None** for Trend, the model does not assess the data for trend. When you select **None** for Season, the model does not assess the data for seasonality.
- **Additive**: An additive model is one in which the combined effect of several independent factors is the sum of the isolated effects of each factor. You can assess the data in your view for additive trend, additive seasonality, or both.
- **Multiplicative** : A multiplicative model is one in which the combined effect of several independent factors is the product of the isolated effects of each factor. You can assess the data in your view for multiplicative trend, multiplicative seasonality, or both.

If there is more than one time series in your visualization, the Custom option forces them all to be forecast using the same custom model. Constraining the models in this way usually results in lower quality models than would be produced by automatic model selection.

Constraints on Multiplicative Models

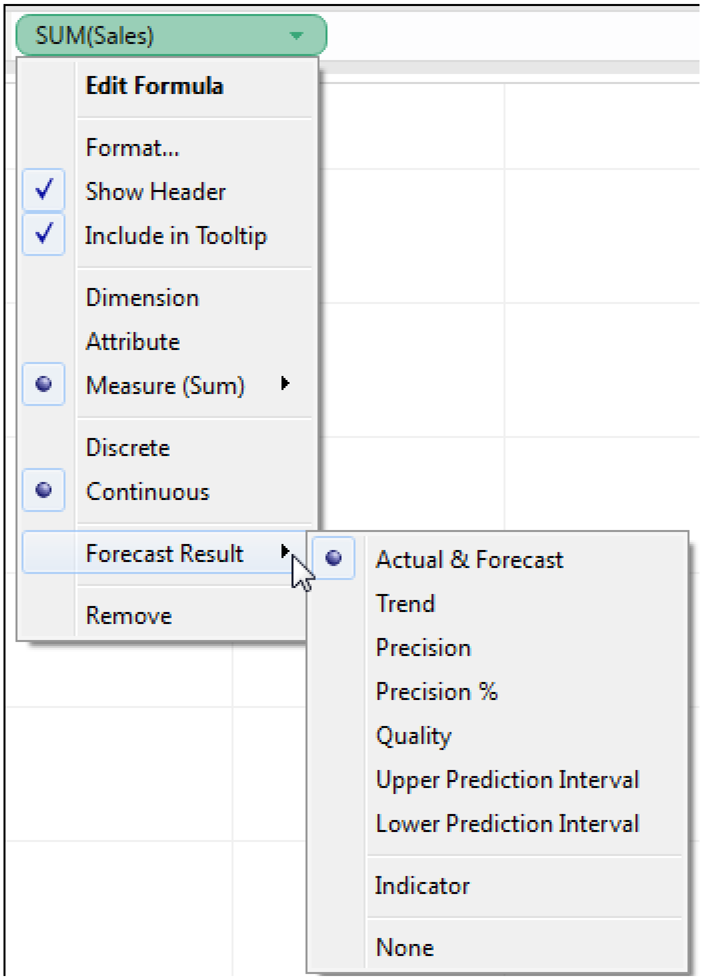
- You cannot use a multiplicative model when the measure to be forecast has one or more values that are less than or equal to zero, or even when some of the data points are too close to zero, relative to other data points.
- You cannot specify a model with multiplicative trend and additive season because the result may be numerically unstable.

Prediction Interval

You can set the prediction interval to 90, 95, or 99 percent, or enter a custom value. This value is used in two locations:

- In the prediction bands displayed with a forecast.
- For the prediction interval options (**Upper Prediction Interva**l and **Lower Prediction**

Interval) that are available as forecast result types for a measure in the view:



Forecast Summary

The text box at the bottom of the Forecast Options dialog box provides a description of the current forecast. The forecast summary updates whenever you change any of the forecast options above. If there is a problem with the forecast, the text box provides an error message that may help you resolve the issue. See **Resolving Forecasting Errors** on page 2603.

Forecast Descriptions

The **Describe Forecast** dialog box describes the forecast models that Tableau computed for your visualization.

When forecasting is enabled, you can open this dialog by selecting **Analysis > Forecast > Describe Forecast**.

The information in the Describe Forecast dialog box is read-only, though you can click **Copy to Clipboard** and then paste the screen contents into a document.

The **Describe Forecast** dialog box has two tabs: a **Summary** tab and a **Models** tab.

Describe Forecast - Summary Tab

The **Summary** tab describes the forecast models Tableau has created, as well as the general patterns Tableau discovered in the data.

Options Used To Create Forecasts

This section summarizes the options Tableau used to create forecasts. These options were either picked automatically by Tableau or specified in the Forecast Options dialog box.

- **Time series**—The continuous date field used to define the time series. In some cases this value might not actually be a date. See **Forecasting When No Date is in the View** on page 2591.
- **Measures**—The measures for which values are estimated.
- **Forecast forward**—The length and date range of the forecast.
- **Forecast based on**—The date range of the actual data used to create the forecast.
- **Ignore last**—The number of periods at the end of the actual data that are disregarded--forecast data is displayed for these periods.This value is determined by the **Ignore Last** option in the Forecast Options dialog box.
- **Seasonal pattern**—The length of the seasonal cycle that Tableau found in the data, or None if no seasonal cycle was found in any forecast.

Forecast Summary Tables

For each measure that is forecasted, a summary table is displayed describing the forecast. If the view is broken into multiple panes using dimensions, a column is inserted into each table that identifies the dimensions. The fields in summary forecast tables are:

- **Initial**—The value and prediction interval of the first forecast period.
- **Change From Initial**—The difference between the first and the last forecast estimate points. The interval for those two points is shown in the column header. When values are shown as percentages, this field shows the percentage change from the first forecast period.
- **Seasonal Effect**—These fields are displayed for models identified as having

seasonality--that is, a repeating pattern of variation over time. They show the high and low value of the seasonal component of the last full seasonal cycle in the combined time series of actual and forecast values. The seasonal component expresses the deviation from the trend and so varies around zero and sums to zero over the course of a season.

- **Contribution**—The extent to which Trend and Seasonality contribute to the forecast. These values are always expressed as percentages and add up to 100%.
- **Quality**—Indicates how well the forecast fits the actual data. Possible values are GOOD, OK, and POOR. A naïve forecast is defined as a forecast that estimates that the value of the next period will be identical to the value of the current period. Quality is expressed relative to a naïve forecast, such that OK means the forecast is likely to have less error than a naïve forecast, GOOD means that the forecast has less than half as much error, and POOR means that the forecast has more error.

Describe Forecast - Models Tab

The **Models** tab provides more exhaustive statistics and smoothing coefficient values for the Holt-Winters exponential smoothing models underlying the forecasts. For each measure that is forecasted, a table is displayed describing the forecast models Tableau created for the measure. If the view is broken into multiple panes using dimensions, a column is inserted into each table that identifies the dimensions. The table is organized into the following sections:

Model

Specifies whether the components **Level**, **Trend**, or **Season** are part of the model used to generate the forecast. The value for each component is one of the following:

- **None**—The component is not present in the model.
- **Additive**—The component is present and is added to the other components to create the overall forecast value.
- **Multiplicative**—The component is present and is multiplied by the other components to create the overall forecast value.

Quality Metrics

This set of values provides statistical information about the quality of the model.

Value	Definition

RMSE: Root mean squared error	$\sqrt{\left(\frac{1}{n}\right)\sum e(t)^2}$
MAE: Mean absolute error	$\frac{1}{n}\sum \left\|e\left(t\right)\right\|$
MASE: Mean absolute scaled error. MASE measures the magnitude of the error compared to the magnitude of the error of a naive one-step ahead forecast as a ratio. A naive forecast assumes that whatever the value is today will be same value tomorrow. So, a MASE of 0.5 means that your forecast is likely to have half as much error as a naive forecast, which is better than a MASE of 1.0, which is no better than a naive forecast. Since this is a normalized statistic that is defined for all values and weighs errors evenly, it is an excellent metric for comparing the quality of different forecast methods. The advantage of MASE over the more common MAPE metric is that MASE is defined for time series that contain zero, whereas MAPE is not. Also, MASE weights errors equally, whereas MAPE weights positive and/or extreme errors more heavily.	$\frac{\frac{1}{n}\sum\left\|e\left(t\right)\right\|}{\frac{1}{(n-1)}\sum_2^n\left\|Y\left(t\right)-Y\left(t-1\right)\right\|}$
MAPE: Mean absolute percentage error. MAPE measures the magnitude of the error compared to the magnitude of your data, as a percentage. So, a MAPE of 20% is better than a MAPE of 60%. Errors are the differences between the response values, which the model estimates, and the actual response values for each explanatory value in your data. Since this is a	$100\frac{1}{n}\sum\left\|\frac{e(t)}{A(t)}\right\|$

normalized statistic, it can be used to compare the quality of different models computed in Tableau. However, it is unreliable for some comparisons because it weights some kinds of error more heavily than others. Also, it is undefined for data with values of zero.	
AIC: Akaike information criterion. AIC is a model quality measure, developed by Hirotugu Akaike, that penalizes complex models to prevent overfitting. In this definition, *k* is the number of estimated parameters, including initial states, and *SSE* is the sum of the squared errors.	$n * \log(SSE/n) + 2 * (k+1)$

In the preceding definitions, the variables are as follow:

Variable	**Meaning**
t	Index of a period in a time series.
n	Time series length.
m	Number of periods in a season/cycle.
A(t)	Actual value of the time series at period t.
F(t)	Fitted or forecast value at period t.

Residuals are: e(t) = F(t)-A(t)

Smoothing Coefficients

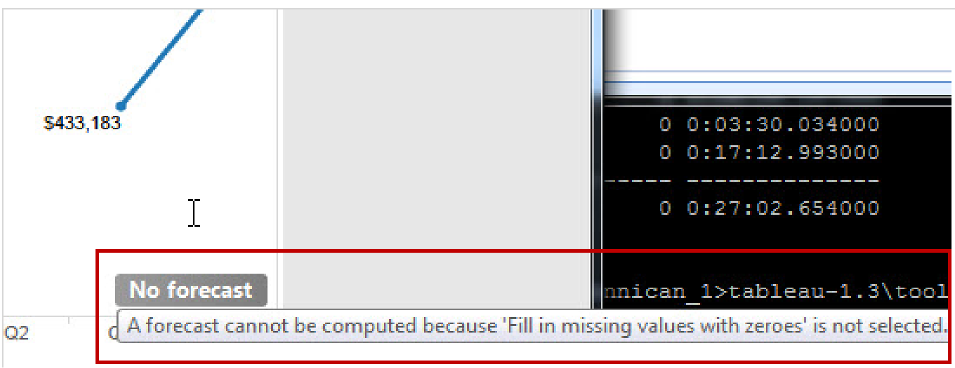
Depending on the rate of evolution in the level, trend, or seasonal components of the data, smoothing coefficients are optimized to weight more recent data values over older ones, such that within-sample one-step-ahead forecast errors are minimized. Alpha is the level smoothing coefficient, beta the trend smoothing coefficient, and gamma the seasonal smoothing coefficient. The closer a smoothing coefficient is to 1.00, the less smoothing is performed, allowing for rapid component changes and heavy reliance on recent data. The closer a smoothing coefficient is to 0.00, the more smoothing is performed, allowing for gradual component changes and less reliance on recent data.

Troubleshooting Forecasting

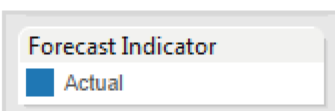
This topic discusses some issues that may arise as you use forecasting in Tableau.

Null Forecasts

A null forecast results when you are using forecasting and modify the view in a manner that is incompatible with forecasting. The most obvious indication that you have a null forecast is that no forecast data is displayed in your view and the text No Forecast is displayed. Hover the cursor over this message to see why Tableau is unable to create a forecast:



Another indication is that the **Forecast Indicator** field on the Marks card shows Actual values, but no Estimate values:



To diagnose a null forecast, open the Describe Forecast dialog box from the Analysis menu (**Analysis > Forecast > Describe Forecast**) to see the error message. Then see **Resolving Forecasting Errors** on the facing page for a suggested resolution.

Tableau also displays the error message in the Forecast Options dialog box (**Analysis > Forecast > Forecast Options**).

Because a forecast cannot be computed for a time series with null date values in the middle, actions that filter data, either explicitly or implicitly can trigger a null forecast. The **Keep Only** and **Exclude** commands on tooltips are examples of actions that can filter data implicitly—in some cases, these commands are removed when a forecast is shown. For example, if you have a time series of sales for each quarter from 2008 until 2012 and you exclude the Sales value for the second quarter of 2010, you will get a Null forecast because the time series is irregular. If, instead, you exclude the first quarter of 2008, you have shortened the time series but it remains regular. So, a valid forecast is still possible.

Resolving Forecasting Errors

If Tableau is unable to provide a forecast for your view, the problem can often be resolved by changing the Date value in the view (see **Change Date Levels** on page 1248).

Forecasting errors can result when the aggregation level of the time series (months, weeks, etc.) is either too fine or too coarse for the data to be forecast. This can lead to the "too much data" or "too little data" errors described below. Date aggregation can trigger a "too many Nulls" scenario when forecasting attempts to extract more data from the measure than is possible. For example, if the underlying granularity of the sales data is months but you aggregate by weeks, the result may be a significant number of Null values.

Other problems arise when the view’s aggregation and the aggregation specified for the forecast (using the **Aggregate by** field in the Forecast Options dialog box) are not compatible. Tableau can create a forecast when the forecast aggregation is a finer level of detail than the view's aggregation, but not when it is at a coarser level of detail; even when it is finer, the two values are only compatible if there is a strict hierarchy that Tableau can use (for example, quarters can be evenly divided into three months, but months can't be evenly divided into weeks). Avoid these scenarios by setting **Aggregate by** to Automatic.

The following list shows errors that can be result from invalid forecasts in Tableau, and provides advice on how to resolve them.

Error message	Suggestion for Resolution
A continuous date cannot be derived from the date fields in the view.	Forecasting requires a date field that can be interpreted continuously. If the date field is not explicitly continuous, then one of the included date levels must be Year. This error is returned if there are no dates in the view, or if the dates in the view don’t constitute a full hierarchy (for example, the date includes Year and Day, but not Month), or if they constitute a hierarchy that is not supported (for example, Year, Week, Day).
The time series is too short	Expand the time series in your view to include more date

to forecast.	values. This error is returned if there are fewer than four data points after trimming off unreliable or partial trailing periods which could mislead the forecast.
A forecast cannot be computed for a time series with Null date values.	Eliminate any Null values from the date field or fields in the view, either by filtering the date field or by using a less detailed date granularity (for example, by switching from months to quarters).
A forecast cannot be computed when the view contains multiple distinct date fields.	This error is returned if there are multiple date fields in the view. For example, if both Order Date and Ship Date are in the same view, forecasting is not supported.
The selected 'Aggregate by' value in Forecast Options is not compatible with the visualization.	The date in the view must be compatible with the value of **Aggregate by** in the Forecast Options dialog box. For example, if **Aggregate by** is set to Weeks and the date in the view is set to Months, this error occurs. Change one of the dates so that the two are compatible, or set **Aggregate by** to Automatic.
A forecast cannot be computed because there are too many missing values.	This error is returned if more than 40% of the data in a pane is missing. Selecting **Fill in missing values with zeros** in the Forecast Options dialog box will not resolve this error. Aggregate your data to a higher level of detail by removing dimensions or changing the date level, for example from 'weeks' to 'months'. Otherwise, you must modify the source data or use data from a different source.
There is no measure to forecast.	This error is returned if no measure that can be forecast is present in the view. Forecast measures must be on the Rows or Columns shelf, or on the Marks card.
The measure to forecast must be a number.	Some measures cannot be interpreted numerically and therefore cannot be forecast.

A forecast cannot be computed for a dimension.	The value to be forecast must be a measure, and not a dimension.
There is too much data to compute a forecast.	Forecasting is not possible when the result set from the query is too large. The limit is about 10,000 rows. To fix the forecast, aggregate the time series value at a higher level (for example, Month instead of Week) or filter the data.
A forecast cannot be computed because the data is divided into too many rows, columns, or colors.	Simplify the view to resolve the error by filtering or removing some of the dimensions.
A forecast cannot be computed because the view contains table calculations.	Create a version of the view that does not contain table calculations.
A forecast cannot be computed because there is a measure on the Filters shelf.	Remove the measure from the Filters shelf.
A forecast cannot be computed because **Aggregate Measures** is not selected.	**Aggregate Measures** is an option on the Analysis menu. See **Data Aggregation in Tableau** on page 341 and **How to Disaggregate Data** on page 350.
A forecast cannot be computed because the view contains percent calculations.	**Percentage of** is an option on the Analysis menu. See **Calculate Percentages in Tableau** on page 2545.
A forecast cannot be computed because Grand Totals or Subtotals is enabled.	These options are controlled from the **Totals** command in the Analysis menu. See **Show Totals in a Visualization** on page 1456.
A multiplicative model cannot be computed because the measure to be forecast has one or more values that are less than or equal to	You have created a custom model with Trend or Seasonality set to **Multiplicative**. Change this value, or set the Forecast Model to **Automatic**.

zero.	
A model with multiplicative trend and additive season is not allowed because it is numerically unstable.	You have created a custom model configured as described in the error message. Change the settings for the custom model, or set the Forecast Model to **Automatic**.
A seasonal model cannot be computed because the time series is too short.	Expand the time series in your view to include more date values.
The selected multiplicative model cannot be computed because some of the data is too close to zero relative to the rest of the data.	You have created a custom model configured as described in the error message. Change the settings for the custom model, or set the Forecast Model to **Automatic**.

Predictive Modeling

Predictive modeling functions in Tableau use linear regression to build predictive models and generate predictions about your data. Two table calculations, MODEL_PERCENTILE and MODEL_QUANTILE, can generate predictions and surface relationships within your data. These can be used to identify outliers, estimate values for sparse or missing data, and predict values for future time periods.

This section provides all the documentation you need to get started with predictive modeling functions in Tableau.

How Predictive Modeling Functions Work in Tableau

You can already add trend lines and forecasts to a visualization, but now you can go beyond this, using the power of a statistical engine to build a model that understands how your data is distributed around such a trend line, or line of best fit. Previously, users had to integrate Tableau with R and Python in order to perform advanced statistical calculations and visualize them in Tableau. Now, you can use the predictive modeling functions to make predictions from your data by including them in a table calculation. For more information about table calculations, see Transform Values with Table Calculations.

With these predictive modeling functions, you can select targets and predictors by updating the variables and visualizing multiple models with different combinations of predictors. The data can be filtered, aggregated, and transformed at any level of detail, and the model—and thus the prediction—will automatically recalculate to match your data.

For a detailed example that shows how to create prediction calculations using these functions, see **Example - Explore Female Life Expectancy with Predictive Modeling Functions** on page 2637.

Predictive modeling functions available in Tableau

Function	Syntax	Description
MODEL_ QUANTILE	`MODEL_QUANTILE( model_specification (optional), quantile, target_expression, predictor_expression(s))`	Returns a target numeric value within the probable range defined by the target expression and other predictors, at a specified quantile. This is the Posterior Predictive Quantile. Example: `MODEL_ QUANTILE (0.5, SUM ([Sales]), COUNT ([Orders]))`
MODEL_ PERCENTILE	`MODEL_PERCENTILE( model_specification (optional), target_expression, predictor_expression(s))`	Returns the probability (between 0 and 1) of the expected value being less than or equal to the observed mark, defined by the target

		expression and other predictors. This is the Posterior Predictive Distribution Function, also known as the Cumulative Distribution Function (CDF). Example: `MODEL_PERCENTILE( SUM([Sales]), COUNT([Orders]))`

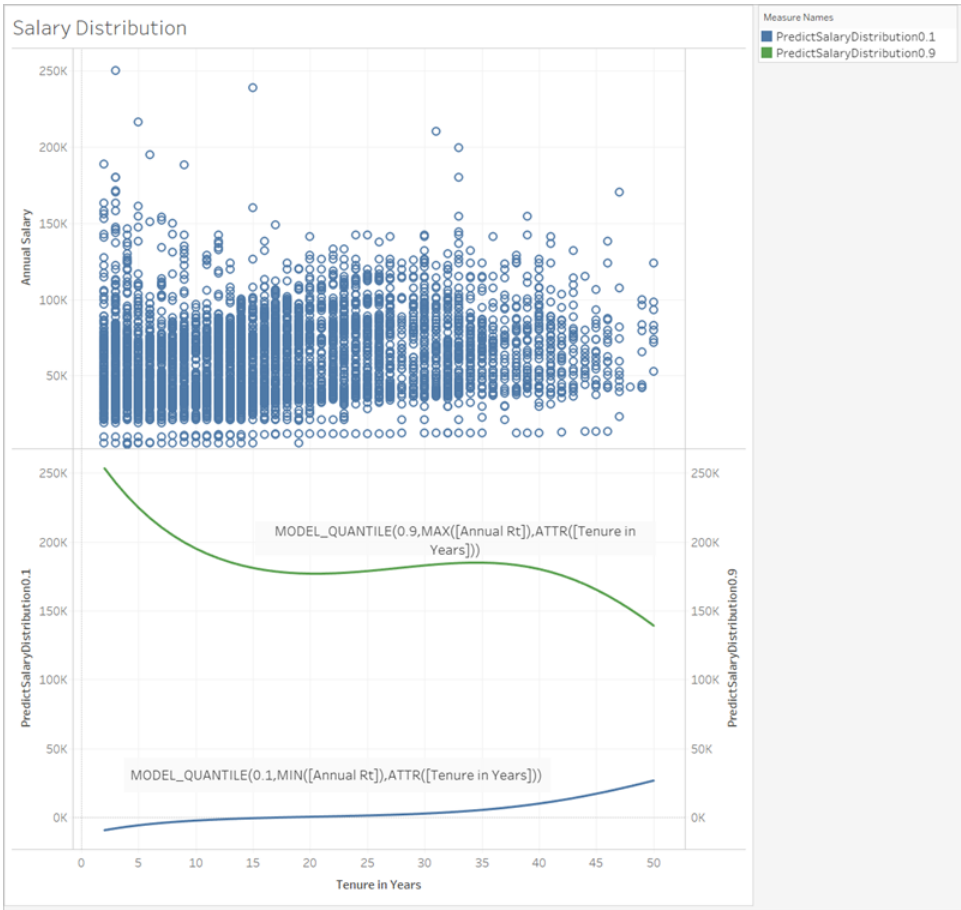
The power of predictive modeling functions

Let's look at an example using salary data, starting with MODEL_QUANTILE.

In the example below, MODEL_QUANTILE has been used to display the 10th percentile and the 90th percentile of the expected distribution for the same data set. Based on the existing data and using a linear regression model, the statistical engine has determined that there is a 90% probability that the maximum salary for each tenure will be below the green line, and a 10% probability that the minimum salary for each tenure will be below the blue line.

In other words, with the quantile set at 0.9, the model predicts all salaries will fall at or below the green line 90% of the time. The blue line is set at .1 or 10%, so only 10% of salaries will fall at or below the blue line, with the inverse of that (90%) above the blue line.

Effectively, this gives us a band that we can predict 80% of potentially generated future points or unobserved data will fall inside.



Next, let’s see how MODEL_PERCENTILE, the inverse of MODEL_QUANTILE, can help us understand the data further.

You can identify the outliers of the data set by using the MODEL_PERCENTILE function. MODEL_PERCENTILE tells you, as a percentile, where the observed mark falls within a range of probable values for each mark. If the percentile is very close to 0.5, the value observed is very close to the median value predicted. If the percentile is close to 0 or 1, the value observed is at the lower or upper boundaries of the model range and is relatively unexpected.

Below, we have applied MODEL_PERCENTILE as color to the salary visualization in the top half of the image, to help us understand which values are most expected.

Syntax of predictive modeling functions in detail

What is MODEL_QUANTILE?

MODEL_QUANTILE calculates the posterior predictive quantile, or the expected value at a specified quantile.

- **Quantile:** The first argument is a number between 0 and 1, indicating what quantile should be predicted. For example, 0.5 specifies that the median will be predicted.
- **Target expression:** The second argument is the measure to predict or "target."
- **Predictor expression(s):** The third argument is the predictor used to make the prediction. Predictors can be dimensions, measures, or both.

The result is a number within the probable range.

You can use MODEL_QUANTILE to generate a confidence interval, missing values such as future dates, or to generate categories that don't exist in your underlying data set.

What is MODEL_PERCENTILE?

MODEL_PERCENTILE calculates the posterior predictive distribution function, also known as the Cumulative Distribution Function (CDF). This calculates the quantile of a particular value between 0 and 1, the inverse of MODEL_QUANTILE.

- **Target expression:** The first argument is the measure to target, identifying which values to assess.
- **Predictor expression(s):** The second argument is the predictor used to make the prediction.
- Additional arguments are optional and are included to control the prediction.

Notice that the calculation syntax is similar, with MODEL_QUANTILE having the extra argument of a defined quantile.

The result is the probability of the expected value being less than or equal to the observed value expressed in the mark.

You can use MODEL_PERCENTILE to surface correlations and relationships within your database. If MODEL_PERCENTILE returns a value close to 0.5, the observed mark is near the median of the range of predicted values, given the other predictors that you've selected. If MODEL_PERCENTILE returns a value close to 0 or to 1, the observed mark is near the lower or upper range of what the model expects, given the other predictors that you've selected.
Note: Dimensions used as predictors can be *ordered* or *unordered.* An ordered dimension is any whose values can be sequenced, such as MONTH. An unordered dimension is any whose values don't have an inherent sequence, such as gender or color. This distinction matters when using Gaussian process regression. For more information on this model and others, see **Choosing a Predictive Model** on page 2615.

For advanced users, there are two other optional arguments you can include to control the prediction. For more information, see **Regularization and Augmentation in Predictive Modeling** on page 2659.

What's being computed?

The input used to build the model is a matrix where each mark is a row, and the columns are the target expression and predictor expressions evaluated for each mark. Whatever row is specified on the viz is what defines the row for the data set being computed by the statistical engine.

Let's look at the example below, where rows (and thus marks) are defined by job titles, and columns are the target expression MEDIAN([Annual Rt]). These are followed by the optional additional predictors MEDIAN([Tenure in Months (Measure)] and ATTR([Department Generic (group)].

Jobtitle	Actual Median Annual Rt	Median Tenure in Months (Measure)	Department
911 Lead Operator	60,425	287	Fire Department
911 Operator	53,134	141	Fire Department
911 Operator Supervisor	66,537	423	Fire Department
Account Executive	60,700	35	M-R
Account Executive Supervisor	73,900	70	M-R
Accountant I	52,268	75	*
Accountant II	61,451	102	*
Accountant Supervisor	71,300	225	*
Accounting Assistant I	32,003	113	*
Accounting Assistant II	34,011	61	*
Accounting Assistant III	46,758	206	*
Accounting Asst II	32,271	58	M-R
Accounting Asst III	41,810	60	M-R
Accounting Manager	88,100	123	*
Accounting Systems Admin	68,136	137	Finance

Note: Several job titles have an asterisk listed under Department. This is because those job titles exist in several departments but Tableau treats all of them as though they are in the same department. In addition, the statistical engine doesn't consider the number of records, and treats each mark as equally likely. This is because Tableau can't perform table calculations on unaggregated data, and can't aggregate the results of table calculations. For more information on aggregation, see Data Aggregation in Tableau.

For more information on the data used to build a model and generate predictions, see **Compute Using and Data Partitioning in Predictive Modeling** on page 2628.

What models are supported?

Predictive modeling functions support linear regression, regularized linear regression, and Gaussian process regression. These models support different use cases and prediction types, as well as have different limitations. For more information, see **Choosing a Predictive Model** on page 2615.

Choosing your predictors

A predictor can be any field in your data source—measure or dimension—including calculated fields.

Note: If using a dimension as a predictor, you must use a ATTR wrapper (e.g. ATTR ([State]rather than just [State]). This is because dimensions will generally only work if they are in the level of detail in the visualization either directly, or above it in a hierarchy.

For example, suppose you have a data set that includes the fields [City], [State], and [Region], where multiple[City] records exist within a [State], and multiple [State] records within a [Region].

In a visualization that uses [State] as a mark, the predictor ATTR([State]) or ATTR([Region]) will both work as predictors. However, the predictor ATTR([City]) will revert to *, since multiple cities exist within a visualized state and so cannot be used as predictors. In other words, including a predictor that's at a lower level of detail than the viz adds no value to the predictions. In most cases, a predictor at a lower level of detail than the viz will evaluate to * and so will all be treated identically.

However, if the same data set is used to generate a viz that uses [City] as a mark, ATTR([City]), ATTR([State]), and ATTR([Region]) can all be used successfully as predictors. For more information about the use of ATTR functions, see When to Use the Attribute (ATTR) Function.

Dimensions and measures do not need to be visualized (in the view or viz) in order to be included as predictors. For more detailed guidance, see **Choosing Predictors** on page 2617.

Recommendations

Prediction calculations are best used as follows:

- To predict values for individual records, where each mark in your viz represents a discrete entity such as a product, sale, person, etc., rather than aggregated data. This is because Tableau regards each mark as equally likely, even if one mark is made up of 100 records and the other marks are made up of one record each. The statistical engine does not weigh marks based on the number of records that make it up.
- To predict values for aggregated target expressions using SUM and COUNT.

Note: MODEL_QUANTILE and MODEL_PERCENTILE aren't recommended to predict values for aggregated target_expressions using AVG, MEDIAN, MIN, and MAX.

Limitations

- You must use a calculated field to extend a time series into the future. For more information, see Predicting the Future.
- Predictors should be at the same or higher level of detail than the view. That is, if your view aggregates by state, you should use state or region as a predictor, but not city. For more information, see **Choosing Predictors** on page 2617.

When will prediction calculations break?

Regardless of the model you're using, you must have at least three data points within each partition for the model to return a response.

If you have specified Gaussian process regression as your model, it can be used in predictive calculations with one **ordered** dimensional predictor and any number of **unordered** dimensional predictors. Measures aren't supported as predictors in Gaussian process regression calculations, but can be used in linear and regularized linear regression calculations. For more information on model selection, see **Choosing a Predictive Model** on the facing page.

If your calculation used ATTR[State] as a predictor and your viz also included State as a mark but no other field at a lower level of detail, such as City, then you would return an error. To prevent this, simply ensure there is not a one-to-one relationship between marks and predictor categories.

For more information on these and other prediction problems, see Resolve Errors in Predictive Modeling Functions.

FAQ

What about marks in multiple predictor-groups?

If a row is aggregated from data that exists within multiple predictor-groups, the value of the ATTR function is a special value of multiple values. For instance, all cities that exist in multiple states will have the same predicted value (unless there are other predictors that are distinct). When selecting your predictors, it's best to use predictors that are at the same level of detail as the viz or or higher. Again, for more information on ATTR functions, see When to Use the Attribute (ATTR) Function.

What if ATTR Aggregation returns a * value?

* is treated as a distinct value. If ATTR returns * for all marks, then you, in essence, have a predictor with a constant value, which will be ignored. This is the same as not including that predictor at all.

If ATTR returns * for some but not all marks, then it will be treated as a category where all the * values are considered to be the same. This scenario is identical to the above scenario, in which marks exist in multiple predictor-groups.

What about the "Compute Using" table calculation menu options?

This functions identically to Compute Using in other table calculations. For more information, see **Compute Using and Data Partitioning in Predictive Modeling** on page 2628.

Why am I getting an error?

There are several reasons why you might be encountering an error when using predictive modeling functions. For detailed troubleshooting steps, see Resolve Errors in Predictive

Modeling Functions.

Choosing a Predictive Model

Predictive modeling functions support linear regression, regularized linear regression, and Gaussian process regression. These models support different use cases and prediction types, as well as have different limitations.

Supported models

Linear regression

Linear regression (also known as ordinary least squares regression, or OLS) is best used when there are one or more predictors that have a linear relationship between the prediction and the prediction target, they aren't affected by the same underlying conditions, and they don't represent two instances of the same data (for example, sales expressed in both dollars and euros). Linear regression is the default model for predictive modeling functions in Tableau; if you don't specify a model, linear regression will be used. You can explicitly specify this model by including **"model=linear"** as the first argument in your table calculation.

Example:

```
MODEL_QUANTILE(
"model=linear",
0.5,
SUM([Sales]),
ATTR(DATETRUNC('month',([Order Date]))
)
```

Regularized linear regression

Regularized linear regression is best used when there's an approximate linear relationship between two or more independent variables—also known as multicollinearity. This is frequently observed in real-world data sets. To use this model instead of the default linear regression, include **"model=rl"** as the first argument in your table calculation.

Example:

```
MODEL_QUANTILE(
"model=rl",
0.5,
SUM([Sales]),
ATTR(DATETRUNC('month',([Order Date]))
)
```

Gaussian process regression

Gaussian process regression is best used when generating predictions across a continuous domain, such as time or space, or when there is a nonlinear relationship between the variable and the prediction target. Gaussian process regression in Tableau must have a single **ordered** dimension as a predictor but may include multiple **unordered** dimensions as predictors. Note that measures can't be used as predictors in Gaussian process regression in Tableau. To use this model instead of the default linear regression, include **"model=gp"** as the first argument in your table calculation.

Note: An *ordered* dimension is any whose values can be sequenced, such as MONTH. An *unordered* dimension is any whose values don't have an inherent sequence, such as gender or color.

Example:

```
MODEL_PERCENTILE(
"model=gp",
AVG([Days to Ship Actual]),
ATTR(DATETRUNC('month',([Order Date])))
)
```

As a simple heuristic, you can use the below criteria for selecting your model:

- **Linear regression (default):** Use when you have only one predictor, and that predictor has a linear relationship with your target metric.
- **Regularized linear regression:** Use when you have multiple predictors, especially

when those predictors have a linear relationship to the target metric and those predictors are likely affected by similar underlying relationships or trends.

- **Gaussian process regression:** Use when you have time or space predictors, or when you're using predictors that might not have a linear relationship with the target metric.

Choosing Predictors

When creating prediction calculations using the predictive modeling functions in Tableau, you'll need to choose predictors. As a reminder, a predictor is an input variable whose value is used to predict an outcome variable, also known as the target or response. Often, you'll be working with data you have extensive domain knowledge of and will already have a good sense of which fields are strongly correlated with your prediction target, and would make good predictors. However, it's still a good idea to take the time to evaluate your predictors and ensure that you're choosing them wisely. You will *always* want to include at least one predictor, and usually more than one.

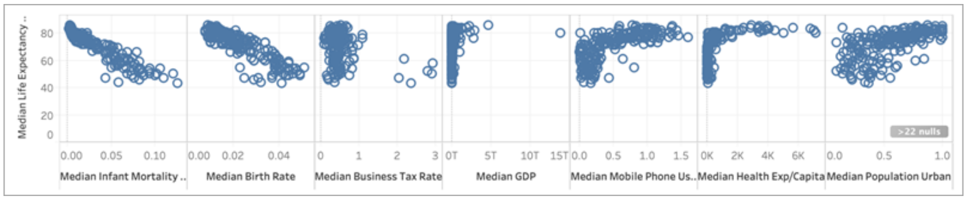
First, select your target. This sounds obvious, but making sure that you're selecting your predictors *based on what you want to predict* is a critical first step. For example, the fields that are most correlated with female life expectancy might be very different from the fields most correlated with male life expectancy. Similarly, the fields most correlated with sales could be very different from the fields most correlated with profit.

Another thing to keep in mind is that by default, the predictive modeling functions use linear regression as the underlying statistical model. With this model, the most correlated predictors are those that have a linear relationship with the target. For information on using another supported model, see **Choosing a Predictive Model** on page 2615.

To better understand how to choose the best predictors for the questions you're aiming to answer, let's look at female life expectancy data. To follow along, download the following workbook from Tableau Public: Choosing Predictors for Your Predictions.

Measures as predictors

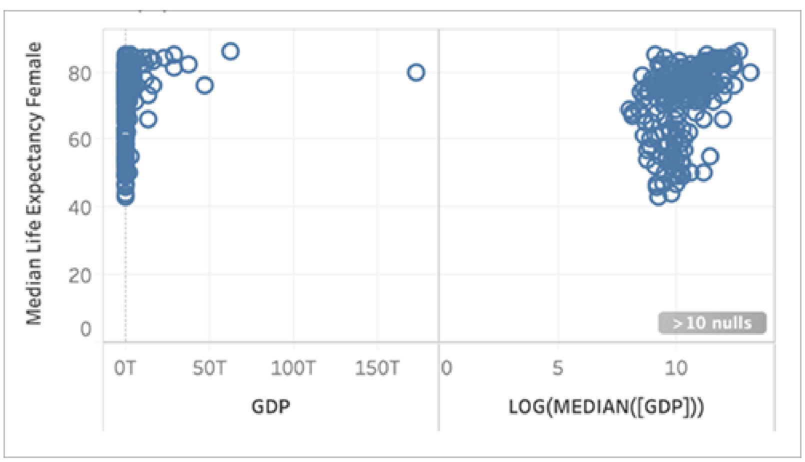
When using a measure as a predictor, you can evaluate its correlation with your target using Tableau. One way is to build a scatter plot. Below, we compare a country's median female life expectancy with a variety of other measures.



For some measures, like **Infant Mortality** and **Birth Rate**, there's a clear negative correlation with **Female Life Expectancy**, as seen in the negative slope of the scatter plot. For others, it's less clear. However, one thing that we can clearly see is an L-shaped distribution for **Median GDP**, **Median Mobile Phone Usage**, and **Median Health Exp/Capita**. This L-shaped distribution often indicates that using a log transformation can help you analyze your data more accurately. Another clue is if all a column's values are positive. In Tableau, you can use a log transformation by editing the pill and wrapping the expression in a LOG function:

```
LOG(MEDIAN([GDP]))
```

This takes us from the L-shaped distribution—where it's hard to differentiate between the extremes of the scale—to a more even distribution less compressed by the extremes of the scale.

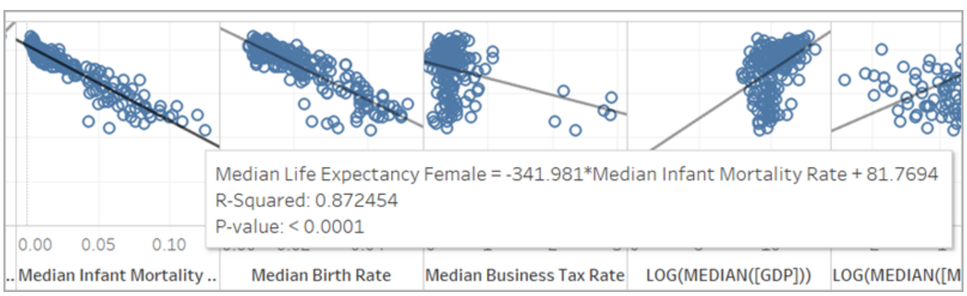


Repeating this with the other L-shaped distributions gives us the following:

Coefficient of determination, or R-squared value

The closer the marks are to lying on a straight line, the higher the correlation between the two measures. To help evaluate the correlation, you can add trend lines. From the **Analytics** pane, drag **Trend Line** into the view and drop it on **Linear.** Hovering your cursor over the trend line will tell you the R-squared value, or coefficient of determination, which indicates how much of the dependent variable (the target) is explained by the independent variable (the predictor). Predictors with R-squared values closer to 1 are better than predictors with R-squared values closer to 0.

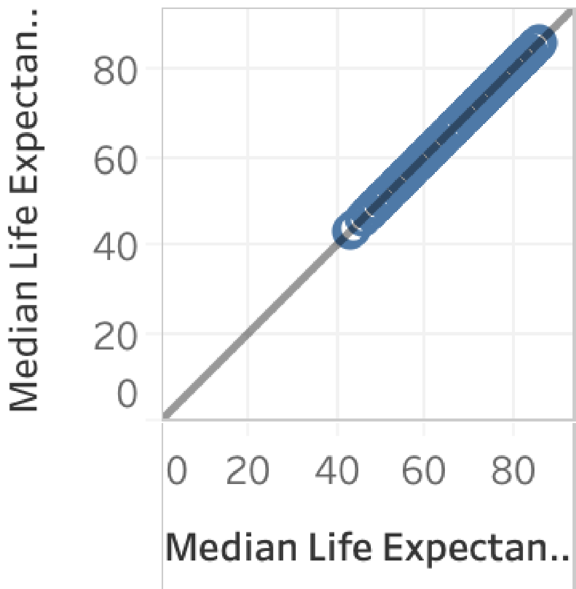
Looking at our scatter plots, we can see that the best predictor for median female life expectancy is median infant mortality, which has an R-squared value of 0.87:



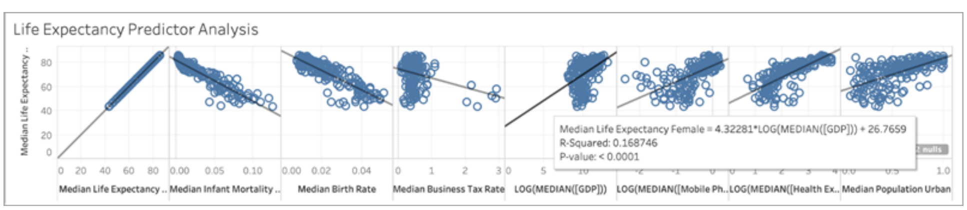
Other good predictors are **Median Birth Rate** (R-squared=0.76) and the log transformation of **Median Health Expenditure/Capita** (R-squared=0.56).

Note: The slope of the line does *not* clearly show you which predictors have the highest R-squared value. Since the scale of the x-axis is determined by the range of the specific variable selected, the slope of the line is greatly affected by the specific variables used.

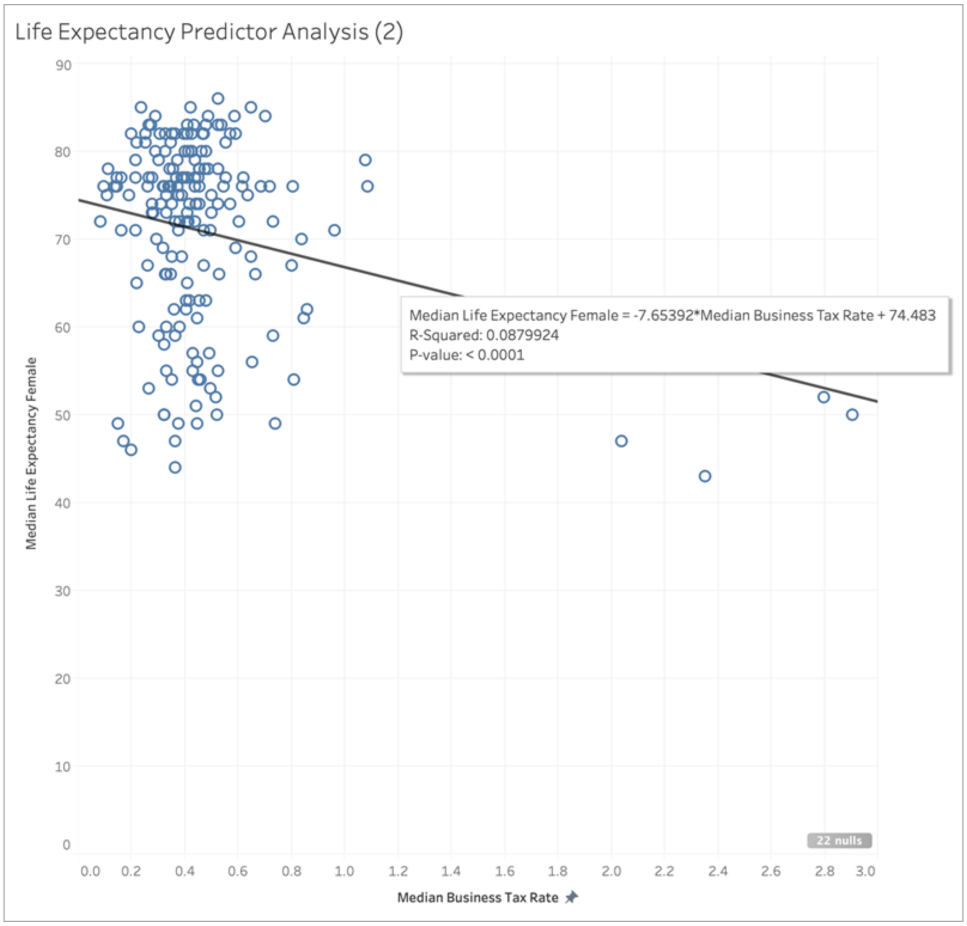
In the below image, we've visualized **Median Female Life Expectancy** against **Median Female Life Expectancy**, with the resulting viz being a perfectly straight line on a 45° angle: as expected, there is a perfect correlation between the value on the x-axis and the value on the y-axis, with an R-squared value of 1:



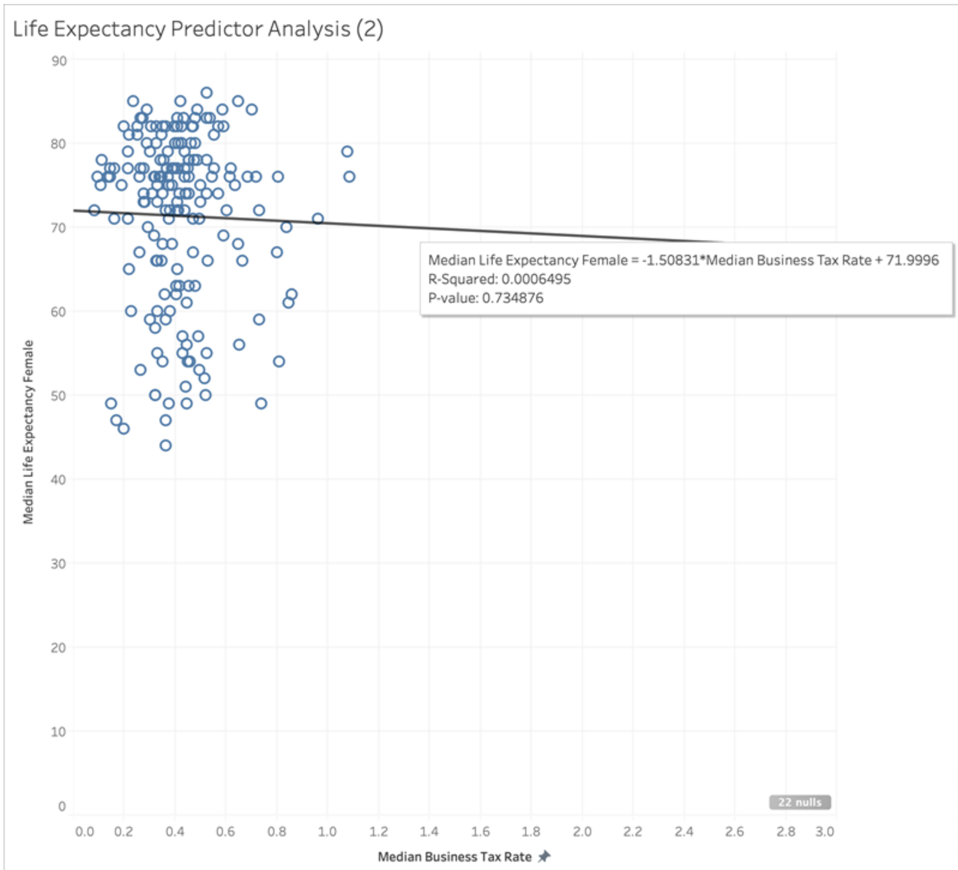
However, as shown below, even though LOG(MEDIAN([GDP])) has a steeper sloped trend line than the others, it has a low R-squared score of only 0.169. This is due to the scale of the x-axis for that pane:



Furthermore, let's take a look at how a few marks can significantly affect the slope of a trend line. Zooming in on the scatter plot for **Median Business Tax Rate** we can see that the majority of the marks have a tax rate between 0 and about 1, with six countries having much higher rates, between 2 and 3. The R-squared value for all marks is 0.0879:



However, let's see what happens if we remove that cluster of six marks:



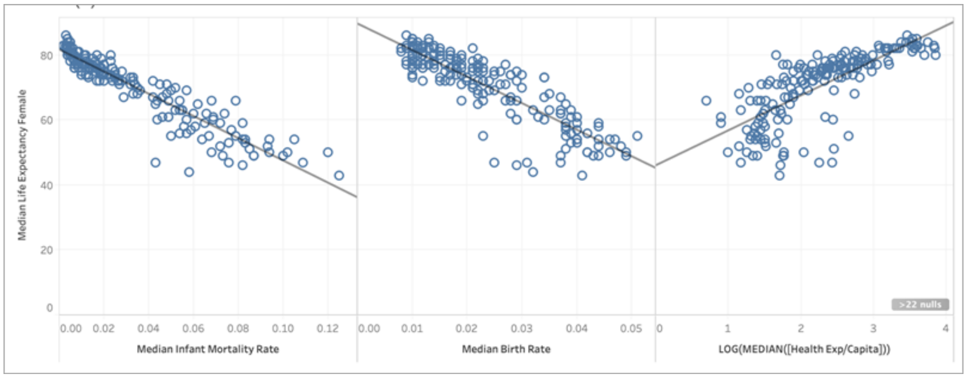
The trend line goes nearly flat, and the R-squared value drops to 0.0006—indicating that there's essentially no correlation between **Median Business Tax Rate** and **Median Female Life Expectancy**. As you visualize your data and use good statistical methods to select your predictors, it's important to carefully consider whether there are any outliers or other data characteristics that could affect your conclusions.

Note: For more examples of how summary statistics might not give you the full picture, read up on Anscombe's quartet.

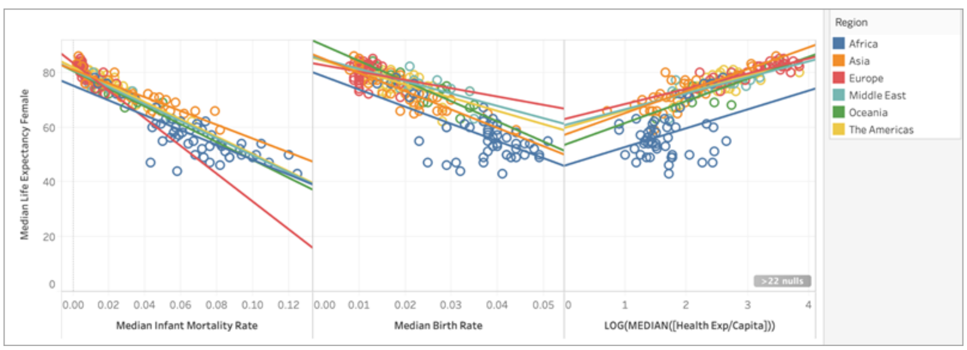
Dimensions as predictors

When using dimensions as predictors, you can use a similar procedure to determine correlation. However, you may find that there’s a significant discrepancy between different dimensions in their level of relationship to the target. For example, when subdividing by **Region**, one region may be a very good predictor for the target, but another region may have significantly less correlation. This doesn’t mean that you shouldn't use that dimension as a predictor, but you may want to consider whether using additional measures or dimensions will help improve your model, and in turn, your predictions.

Since we've determined the best predictors for our data set are **Median Infant Mortality**, **Median Birth Rate**, and the log transformation of **Median Health Expenditure/Capita**, let’s limit the viz to these three variables:



Next, let’s partition the data by adding **Region** to Color on the Marks card and see what happens to the viz:



Comparing R-squared values across predictors

Let's see how the R-squared values compare for each of the **Region** trend lines for each predictor:

Region	Median Infant Mortality Rate	Median Birth Rate	LOG(Median(Health Exp/Capita))
Africa	0.659	0.407	0.131
Asia	0.813	0.701	0.618
Europe	0.547	0.036	0.755
Middle East	0.859	0.491	0.459
Oceania	0.756	0.747	0.656
The Americas	0.797	0.396	0.528

In the table above, the lowest R-squared value for each predictor is highlighted in red, and the second-lowest in yellow.

Europe has the lowest R-squared values for Median Infant Mortality and Median Birth Rate and Africa has the lowest R-squared value for the log transformation of Median Health Expenditure/Capita (highlighted in red). Africa also has lower R-squared values for both Median Infant Mortality Rate and Median Birth Rate.

Adding a dimension can provide your model with more information, and adding more information can improve the quality of your prediction. However, within a given subdivision (a Region, in this case), the prediction quality could either improve or decline. In some cases, you may want to build an individual model for each subdivision based on the measures that are the best predictors for that specific group.

In this case, Infant Mortality has a reasonably strong correlation with Female Life Expectancy for all regions, although it's somewhat weaker in Africa and Europe; Median Birth Rate is a good predictor for Oceania and Asia, but has almost no correlation with Female Life Expectancy in Europe, and the log transformation of Median Health Expenditure is a reasonable predictor for all regions except Africa. We can expect that the model built with all four predictors (Infant Mortality, Birth Rate, Log(Health Expenditure), and Region) will have the least accurate predictions for countries in Europe and Africa; we may want to do some deeper dives into the data to see if there are additional or alternative predictors we could use to build models that are better fits for Europe and Africa.

Building your predictive modeling function

Now that we've found good predictors, we can build and apply a predictive modeling function to see it in action.

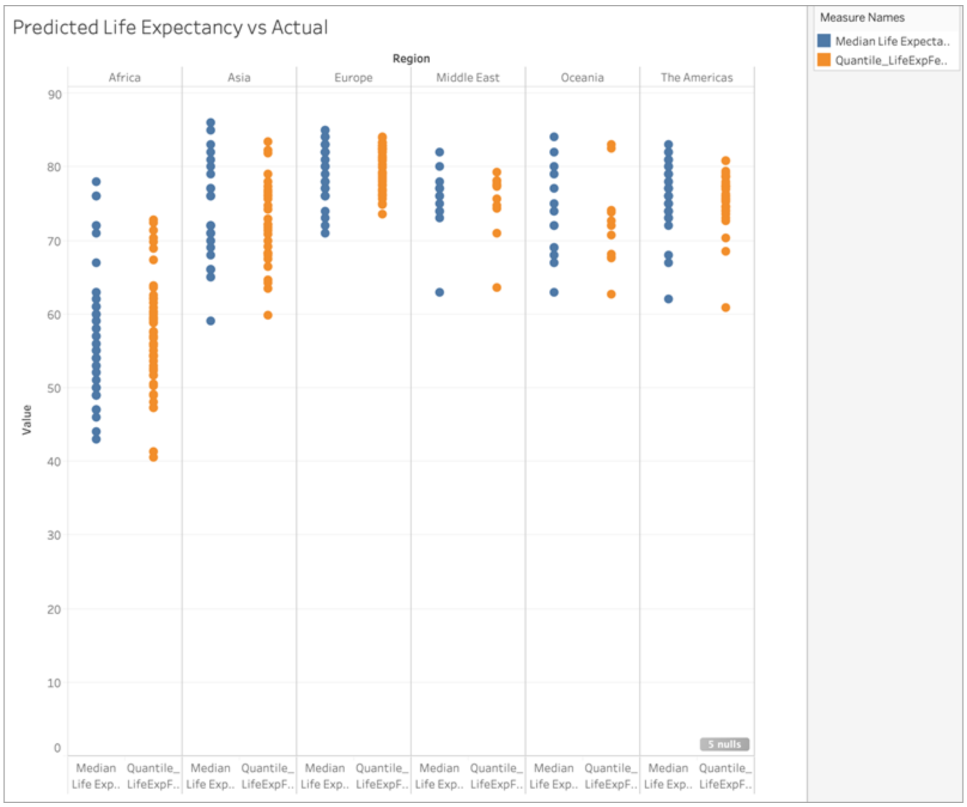
1. Open the **Analysis** menu at the top, and then select **Create Calculated Field**.
2. In the Calculation Editor, name the calculation, do the following:
    - Name the calculation: **Quantile_LifeExpFemale_ HealthExpend,BirthRate,Mortality,Region**
    - Enter the following formula:

    ```
    MODEL_QUANTILE(0.5,MEDIAN([Life Expectancy
    Female]),
    LOG(MEDIAN([Health Exp/Capita])),
    MEDIAN([Birth Rate]),
    ```

```
MEDIAN([Infant Mortality Rate]),
ATTR([Region]))
```

This calculation will return the median value (0.5) of the range of modeled median female life expectancies, based on the predictors we selected: **Health Expenditure**, **Birth Rate**, **Infant Mortality**, and **Region**.

Next, let's build a scatter plot that shows both the *actual* median female life expectancy and the *predicted* median female life expectancy:



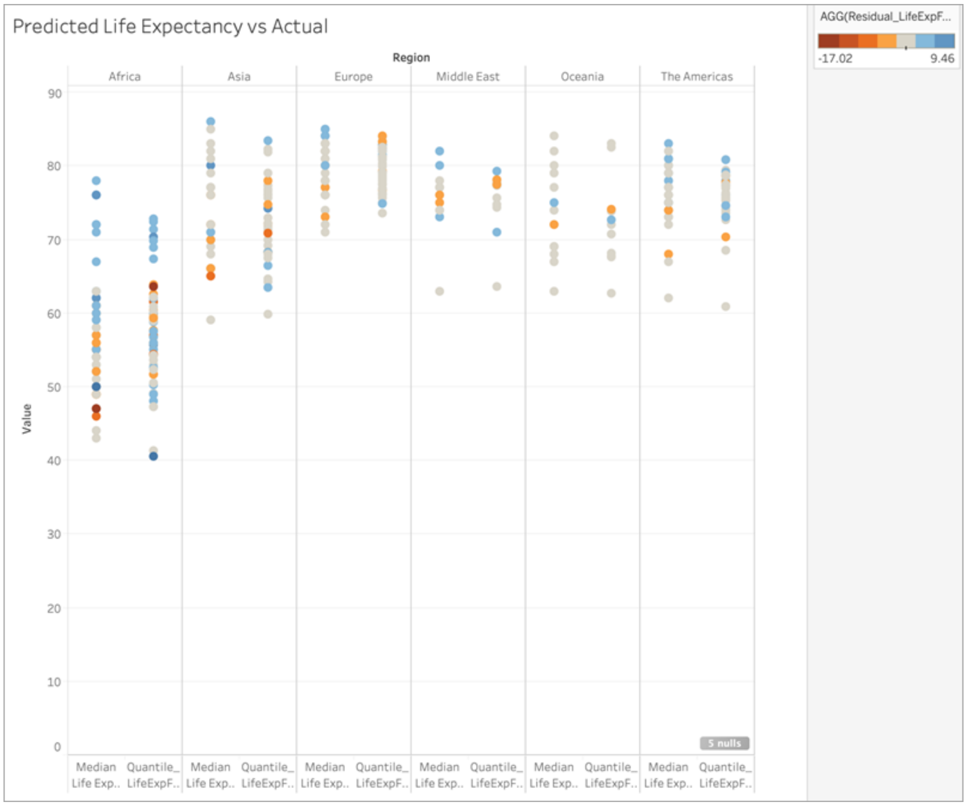
Nice! The predictions are pretty much in line with the actual values for each region.

But let's do another pass to figure out where the predictions are furthest off the mark. Create another calculation **called Residual_LifeExpFemale_ HealthExpend,BirthRate,Mortality,Region**, as follows:

```
MEDIAN([Life Expectancy Female]) - [Quantile_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region]
```

This residual calculation will return the difference between the *predicted* median and the *actual* median, helping us see the countries where there's the biggest discrepancy between the actual and predicted median female life expectancy.

Next, let's apply this residual calculation to **Color**:



You can see in the above viz that most countries, across most regions, have low discrepancies between predictions and actual values. Africa is the region with the largest number of countries with significant discrepancies, but let's do one more round to see what kind of differences we're looking at.

You can see that the differences range between -17 and +9, so let's divide the viz into groups where there's less than ±3 years difference, less than ±5 years difference, less than ±10 years difference, and more than ±10 years difference.

Create one more calculation, called **Grouped_Residual_LifeExpFemale_ HealthExpend,BirthRate,Mortality,Region**, as follows:

```
IF [Residual_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region] <= 3
AND [Residual_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region] >= -3
THEN "±3"
ELSEIF [Residual_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region] <= 5
AND [Residual_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region] >= -5
THEN "±5"
ELSEIF [Residual_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region] <= 10
AND [Residual_LifeExpFemale_
HealthExpend,BirthRate,Mortality,Region] >= -10
THEN "±10"
ELSE
"> ±10"
END
```

Again, let's add the calculation to **Color**:

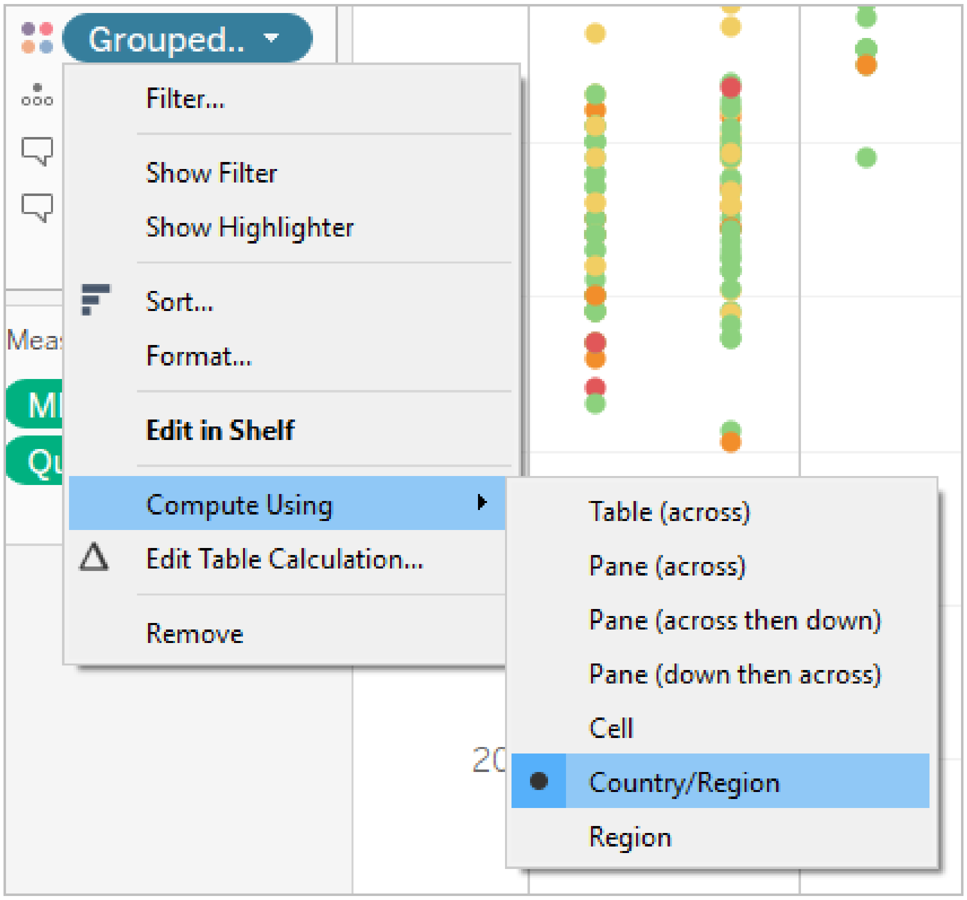
Notice that the vast majority of predictions are incorrect by less than 3 years, and that only a small handful are off by more than 10 years. Overall, pretty good!

This means that using this model would allow us to accurately identify those countries with median female life expectancies that are outliers, or to provide the modeled female life expectancy for a country where this data was missing.

Compute Using and Data Partitioning in Predictive Modeling

You make predictions from your data by including the predictive modeling functions, MODEL_ QUANTILE or MODEL_PERCENTILE, in a table calculation.

Remember that all table calculations must have a **Compute Using** direction specified. For an overview of how different addressing and partitioning dimensions can affect your results, see **Transform Values with Table Calculations** on page 2406.



Note: When you add a table calculation, you must use all dimensions in the level of detail either for partitioning (scoping) or for addressing (direction). The dimensions that define how to group the calculation (the scope of data it is performed on) are called partitioning fields. The table calculation is performed separately within each partition. The remaining dimensions, upon which the table calculation is performed, are called addressing fields, and determine the direction of the calculation. For more details, see The basics: addressing and partitioning.

In predictive modeling functions, the **Compute Using** option is used to partition (scope) the data set that will be used to build the predictive model.

Predictive modeling functions *do not* have a concept of addressing (direction), since the model returns a distinct result for each mark based on the predictors selected. That is, unlike **Running Total**, where the addressing dimension determines the order in which fields are added and results are returned, predictive modeling functions are inherently non-sequential. They calculate results using a model from the data defined by the function's target and predictors, at the level of detail specified by the visualization. Within that data, there's no concept of sequencing unless an ordered predictor, such as a date dimension, is used.

Additionally, the level of detail of the visualization is always used when defining the data used to build the model. All table calculations operate at the same level of detail as the viz itself, and predictive modeling functions are no exception.

Recommendations for predictive modeling functions

It's recommended you select a specific dimension to partition on when using predictive modeling functions. Since you may have multiple prediction calculations in a single viz or dashboard, selecting a specific partitioning dimension ensures that you're building models using the same underlying data set for each individual function, and so comparing results from like models.

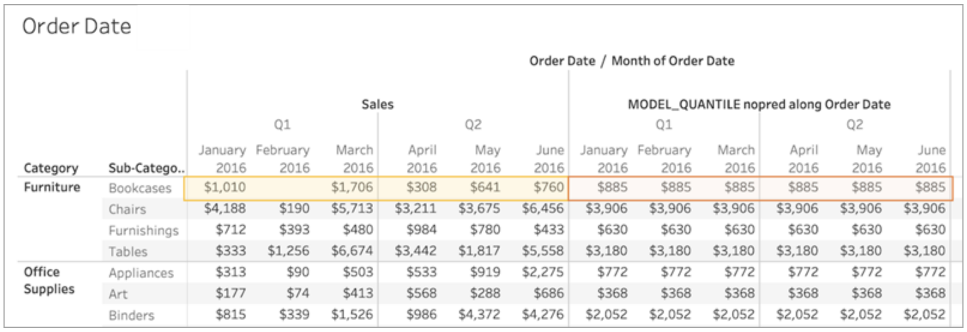
When working with predictive modeling functions in Tableau, it's critical to ensure that you maintain consistency across the different instantiations, both in different iterations of your model (e.g., as you select different predictors), and in different vizzes. Using the directional Compute Using options opens up the possibility that a small change in your visualized data will significantly affect the data being used to build the model, thus affecting its validity and its consistency across different vizzes.

Choosing dimensions

The following examples use the **Sample - Superstore** data source, which is included with Tableau Desktop.

When choosing a dimension, remember that Tableau will build a predictive model *across* that dimension. That is, if you select **Order Date** as your partitioning dimension, Tableau will use data *within* any other established partition, but *along* values of **Order Date**.

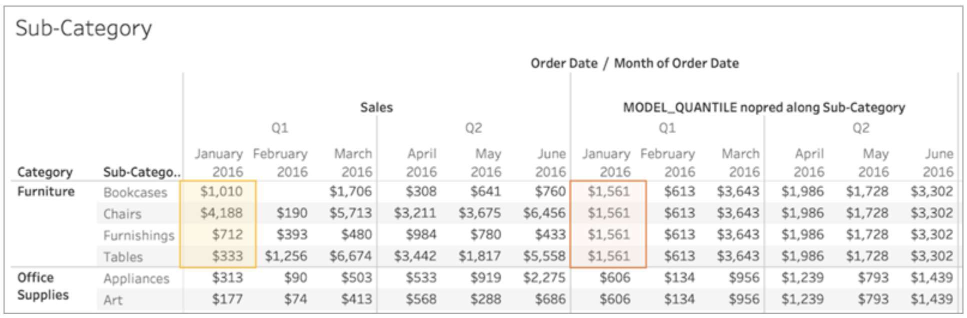
The image below shows the data that is used to build the model highlighted in yellow, and the model output highlighted in orange. In this case, since there aren't any predictors, all responses are identical within a given **Sub-Category**; selecting optimal predictors will help you generate more meaningful results. For more information on optimal predictors, see **Choosing Predictors** on page 2617.



Order Date

		Order Date / Month of Order Date											
		Sales						MODEL_QUANTILE nopred along Order Date					
		Q1			Q2			Q1			Q2		
Category	Sub-Catego..	January 2016	February 2016	March 2016	April 2016	May 2016	June 2016	January 2016	February 2016	March 2016	April 2016	May 2016	June 2016
Furniture	Bookcases	$1,010		$1,706	$308	$641	$760	$885	$885	$885	$885	$885	$885
	Chairs	$4,188	$190	$5,713	$3,211	$3,675	$6,456	$3,906	$3,906	$3,906	$3,906	$3,906	$3,906
	Furnishings	$712	$393	$480	$984	$780	$433	$630	$630	$630	$630	$630	$630
	Tables	$333	$1,256	$6,674	$3,442	$1,817	$5,558	$3,180	$3,180	$3,180	$3,180	$3,180	$3,180
Office Supplies	Appliances	$313	$90	$503	$533	$919	$2,275	$772	$772	$772	$772	$772	$772
	Art	$177	$74	$413	$568	$288	$686	$368	$368	$368	$368	$368	$368
	Binders	$815	$339	$1,526	$986	$4,372	$4,276	$2,052	$2,052	$2,052	$2,052	$2,052	$2,052

Similarly, if **Sub-Category** had been selected as a partitioning dimension, Tableau would use the data *within* a given month but *along* multiple subcategories, as below. If the data is further subdivided into panes, the pane boundaries would be respected when building a model.



Sub-Category

		Order Date / Month of Order Date											
		Sales						MODEL_QUANTILE nopred along Sub-Category					
		Q1			Q2			Q1			Q2		
Category	Sub-Catego..	January 2016	February 2016	March 2016	April 2016	May 2016	June 2016	January 2016	February 2016	March 2016	April 2016	May 2016	June 2016
Furniture	Bookcases	$1,010		$1,706	$308	$641	$760	$1,561	$613	$3,643	$1,986	$1,728	$3,302
	Chairs	$4,188	$190	$5,713	$3,211	$3,675	$6,456	$1,561	$613	$3,643	$1,986	$1,728	$3,302
	Furnishings	$712	$393	$480	$984	$780	$433	$1,561	$613	$3,643	$1,986	$1,728	$3,302
	Tables	$333	$1,256	$6,674	$3,442	$1,817	$5,558	$1,561	$613	$3,643	$1,986	$1,728	$3,302
Office Supplies	Appliances	$313	$90	$503	$533	$919	$2,275	$606	$134	$956	$1,239	$793	$1,439
	Art	$177	$74	$413	$568	$288	$686	$606	$134	$956	$1,239	$793	$1,439

A note on partitioning

Note that partitioning your data visually has significant effects on the data that's used to build a model and generate your predictions. Adding a higher level of detail (for example, including both **State** and **City** on a single shelf) will partition your data by the higher LOD. This is true regardless of the order in which the pills are placed on the shelf. For instance, these will return identical predictions:

Adding a pill that modifies the level of detail will partition your data if it's added to either the Rows or Columns shelf, or to Color, Size, Label, Detail, or Shape on the Marks card. Adding a pill at a different level of detail to Tooltip will *not* partition your data.

In the below example, the model is automatically partitioning by **Category** since the **Category** and **Sub-Category** pills are both on Rows. The prediction calculation is being computed across **Sub-Category** within the boundaries of the higher-level pill, **Category**.

Sub-Category

		Order Date / Month of Order Date											
		Sales						MODEL_QUANTILE nopred along Sub-Category					
		Q1			Q2			Q1			Q2		
Category	Sub-Catego..	January 2016	February 2016	March 2016	April 2016	May 2016	June 2016	January 2016	February 2016	March 2016	April 2016	May 2016	June 2016
Furniture	Bookcases	$1,010		$1,706	$308	$641	$760	$1,561	$613	$3,643	$1,986	$1,728	$3,302
	Chairs	$4,188	$190	$5,713	$3,211	$3,675	$6,456	$1,561	$613	$3,643	$1,986	$1,728	$3,302
	Furnishings	$712	$393	$480	$984	$780	$433	$1,561	$613	$3,643	$1,986	$1,728	$3,302
	Tables	$333	$1,256	$6,674	$3,442	$1,817	$5,558	$1,561	$613	$3,643	$1,986	$1,728	$3,302
Office Supplies	Appliances	$313	$90	$503	$533	$919	$2,275	$606	$134	$956	$1,239	$793	$1,439
	Art	$177	$74	$413	$568	$288	$686	$606	$134	$956	$1,239	$793	$1,439

This has implications for how your predictors are applied. Let's look at the example below. In this case, we have three MODEL_QUANTILE table calculations being applied:

Predict_Sales_City	**Predict_Sales_State**	**Predict_Sales_Region**
`MODEL_QUANTILE (0.5,sum([Sales]), ATTR([City]))`	`MODEL_QUANTILE (0.5,sum([Sales]), ATTR([State]))`	`MODEL_QUANTILE (0.5,sum([Sales]), ATTR([Region]))`

For all three, we've selected **Compute Using > City**. Let's take a look at some cities in North Carolina:

Region	State	City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
South	North Carolina	Asheville	$1,475	2,926	2,926	2,926
		Burlington	$12,681	2,926	2,926	2,926
		Cary	$505	2,926	2,926	2,926
		Chapel Hill	$14	2,926	2,926	2,926
		Charlotte	$13,693	2,926	2,926	2,926
		Concord	$5,112	2,926	2,926	2,926
		Durham	$880	2,926	2,926	2,926
		Fayetteville	$3,183	2,926	2,926	2,926
		Gastonia	$895	2,926	2,926	2,926
		Goldsboro	$35	2,926	2,926	2,926
		Greensboro	$2,407	2,926	2,926	2,926
		Greenville	$2,118	2,926	2,926	2,926
		Hickory	$41	2,926	2,926	2,926
		Jacksonville	$5,580	2,926	2,926	2,926
		Monroe	$1,213	2,926	2,926	2,926
		Raleigh	$4,508	2,926	2,926	2,926
		Thomasville	$151	2,926	2,926	2,926
		Wilmington	$743	2,926	2,926	2,926
		Wilson	$369	2,926	2,926	2,926
	South Carolina	Columbia	$6,168	1,414	1,414	1,414
		Florence	$1,065	1,414	1,414	1,414
		Mount Pleasant	$546	1,414	1,414	1,414
		North Charleston	$284	1,414	1,414	1,414
		Rock Hill	$12	1,414	1,414	1,414
		Summerville	$407	1,414	1,414	1,414
	Tennessee	Bartlett	$89	2,044	2,044	2,044
		Bristol	$437	2,044	2,044	2,044
		Chattanooga	$989	2,044	2,044	2,044
		Clarksville	$2,218	2,044	2,044	2,044
		Columbia	$5,163	2,044	2,044	2,044
		Franklin	$948	2,044	2,044	2,044

Notice that the results from all three calculations are identical within a given state, despite using different predictors.

If we remove **Region** from the Rows shelf, nothing happens to our results—they're still all identical within a given state:

State	City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
North Carolina	Asheville	$1,475	2,926	2,926	2,926
	Burlington	$12,681	2,926	2,926	2,926
	Cary	$505	2,926	2,926	2,926
	Chapel Hill	$14	2,926	2,926	2,926
	Charlotte	$13,693	2,926	2,926	2,926
	Concord	$5,112	2,926	2,926	2,926
	Durham	$880	2,926	2,926	2,926
	Fayetteville	$3,183	2,926	2,926	2,926
	Gastonia	$895	2,926	2,926	2,926
	Goldsboro	$35	2,926	2,926	2,926
	Greensboro	$2,407	2,926	2,926	2,926
	Greenville	$2,118	2,926	2,926	2,926
	Hickory	$41	2,926	2,926	2,926
	Jacksonville	$5,580	2,926	2,926	2,926
	Monroe	$1,213	2,926	2,926	2,926
	Raleigh	$4,508	2,926	2,926	2,926
	Thomasville	$151	2,926	2,926	2,926
	Wilmington	$743	2,926	2,926	2,926
	Wilson	$369	2,926	2,926	2,926
North Dakota	Fargo	$920			
Ohio	Akron	$2,730	3,261	3,261	3,261
	Bowling Green	$1,866	3,261	3,261	3,261
	Cincinnati	$1,612	3,261	3,261	3,261
	Cleveland	$6,346	3,261	3,261	3,261
	Columbus	$15,901	3,261	3,261	3,261

But when we remove **State** from the Rows shelf, we see different results for each calculation:

City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
Burbank	$3,247	4,326	6,260	4,667
Burlington	$21,668	4,326	9,366	9,647
Caldwell	$456	4,326	875	4,667
Camarillo	$1,468	4,326	6,260	4,667
Cambridge	$502	4,326	1,038	6,574
Canton	$818	4,326	3,507	2,528
Carlsbad	$419	4,326	683	4,667
Carol Stream	$1,306	4,326	2,926	2,528
Carrollton	$2,201	4,326	3,102	2,528
Cary	$505	4,326	2,084	2,465
Cedar Hill	$122	4,326	3,102	2,528
Cedar Rapids	$278	4,326	431	2,528
Champaign	$152	4,326	2,926	2,528
Chandler	$1,077	4,326	2,828	4,667
Chapel Hill	$14	4,326	2,084	2,465
Charlotte	$13,693	4,326	2,084	2,465
Charlottesville	$103	4,326	3,607	2,465

What's going on?

In the first example, **Region** and **State** on the Rows shelf are partitioning the cities. Therefore, the models for **Predict_Sales_City**, **Predict_Sales_State**, and **Predict_Sales_Region** are receiving the same data and generating the same predictions.

Since we've already visually partitioned the data within **State** and **Region**, none of our predictors add any value to the model and have no impact on the results:

Region	State	City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
South	North Carolina	Asheville	$1,475	2,926	2,926	2,926
		Burlington	$12,681	2,926	2,926	2,926
		Cary	$505	2,926	2,926	2,926
		Chapel Hill	$14	2,926	2,926	2,926
		Charlotte	$13,693	2,926	2,926	2,926
		Concord	$5,112	2,926	2,926	2,926
		Durham	$880	2,926	2,926	2,926
		Fayetteville	$3,183	2,926	2,926	2,926
		Gastonia	$895	2,926	2,926	2,926
		Goldsboro	$35	2,926	2,926	2,926
		Greensboro	$2,407	2,926	2,926	2,926
		Greenville	$2,118	2,926	2,926	2,926
		Hickory	$41	2,926	2,926	2,926
		Jacksonville	$5,580	2,926	2,926	2,926
		Monroe	$1,213	2,926	2,926	2,926
		Raleigh	$4,508	2,926	2,926	2,926
		Thomasville	$151	2,926	2,926	2,926
		Wilmington	$743	2,926	2,926	2,926
		Wilson	$369	2,926	2,926	2,926
	South Carolina	Columbia	$6,168	1,414	1,414	1,414
		Florence	$1,065	1,414	1,414	1,414
		Mount Pleasant	$546	1,414	1,414	1,414
		North Charleston	$284	1,414	1,414	1,414
		Rock Hill	$12	1,414	1,414	1,414
		Summerville	$407	1,414	1,414	1,414
	Tennessee	Bartlett	$89	2,044	2,044	2,044
		Bristol	$437	2,044	2,044	2,044
		Chattanooga	$989	2,044	2,044	2,044
		Clarksville	$2,218	2,044	2,044	2,044
		Columbia	$5,163	2,044	2,044	2,044
		Franklin	$948	2,044	2,044	2,044

When we remove **Region** from the Rows shelf, we're still partitioning by **State**—so there's no change to the data used to build the model. Again, since we've already visually partitioned the data within **State**, none of our predictors add any value to the model or have any impact on results:

State	City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
North Carolina	Asheville	$1,475	2,926	2,926	2,926
	Burlington	$12,681	2,926	2,926	2,926
	Cary	$505	2,926	2,926	2,926
	Chapel Hill	$14	2,926	2,926	2,926
	Charlotte	$13,693	2,926	2,926	2,926
	Concord	$5,112	2,926	2,926	2,926
	Durham	$880	2,926	2,926	2,926
	Fayetteville	$3,183	2,926	2,926	2,926
	Gastonia	$895	2,926	2,926	2,926
	Goldsboro	$35	2,926	2,926	2,926
	Greensboro	$2,407	2,926	2,926	2,926
	Greenville	$2,118	2,926	2,926	2,926
	Hickory	$41	2,926	2,926	2,926
	Jacksonville	$5,580	2,926	2,926	2,926
	Monroe	$1,213	2,926	2,926	2,926
	Raleigh	$4,508	2,926	2,926	2,926
	Thomasville	$151	2,926	2,926	2,926
	Wilmington	$743	2,926	2,926	2,926
	Wilson	$369	2,926	2,926	2,926
North Dakota	Fargo	$920			
Ohio	Akron	$2,730	3,261	3,261	3,261
	Bowling Green	$1,866	3,261	3,261	3,261
	Cincinnati	$1,612	3,261	3,261	3,261
	Cleveland	$6,346	3,261	3,261	3,261
	Columbus	$15,901	3,261	3,261	3,261

However, when we remove **State**, the data is de-partitioned and we see different predictions for each calculation. Let's take a closer look at what's happening there:

City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
Burbank	$3,247	4,326	6,260	4,667
Burlington	$21,668	4,326	9,366	9,647
Caldwell	$456	4,326	875	4,667
Camarillo	$1,468	4,326	6,260	4,667
Cambridge	$502	4,326	1,038	6,574
Canton	$818	4,326	3,507	2,528
Carlsbad	$419	4,326	683	4,667
Carol Stream	$1,306	4,326	2,926	2,528
Carrollton	$2,201	4,326	3,102	2,528
Cary	$505	4,326	2,084	2,465
Cedar Hill	$122	4,326	3,102	2,528
Cedar Rapids	$278	4,326	431	2,528
Champaign	$152	4,326	2,926	2,528
Chandler	$1,077	4,326	2,828	4,667
Chapel Hill	$14	4,326	2,084	2,465
Charlotte	$13,693	4,326	2,084	2,465
Charlottesville	$103	4,326	3,607	2,465

For **Predict_Sales_City**, we're using ATTR([City]) as a predictor. Since this is at the same level of detail as the viz, it adds no value and is disregarded. We are aggregating **Sales** for all cities, passing them to the statistical engine, and computing the predicted sales. Since no other predictors are included, we see the same result for each city; if we had included one or more measures, we'd see variation in the results.

City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
Burbank	$3,247	4,326	6,260	4,667
Burlington	$21,668	4,326	9,366	9,647
Caldwell	$456	4,326	875	4,667
Camarillo	$1,468	4,326	6,260	4,667
Cambridge	$502	4,326	1,038	6,574
Canton	$818	4,326	3,507	2,528
Carlsbad	$419	4,326	683	4,667
Carol Stream	$1,306	4,326	2,926	2,528
Carrollton	$2,201	4,326	3,102	2,528
Cary	$505	4,326	2,084	2,465
Cedar Hill	$122	4,326	3,102	2,528
Cedar Rapids	$278	4,326	431	2,528
Champaign	$152	4,326	2,926	2,528
Chandler	$1,077	4,326	2,828	4,667
Chapel Hill	$14	4,326	2,084	2,465
Charlotte	$13,693	4,326	2,084	2,465
Charlottesville	$103	4,326	3,607	2,465

For **Predict_Sales_State**, we're using ATTR([State]) as a predictor. The *predictor* is partitioning all the **City** data by **State**. We expect to see identical results *within* a state, but different results for *each* state.

But notice that isn't quite what we get. The cities of Cary, Chapel Hill, and Charlotte all have identical predictions of $2,084, as we expect. Burlington, however, shows us a different prediction of $9,366:

City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
Burbank	$3,247	4,326	6,260	4,667
Burlington	$21,668	4,326	9,366	9,647
Caldwell	$456	4,326	875	4,667
Camarillo	$1,468	4,326	6,260	4,667
Cambridge	$502	4,326	1,038	6,574
Canton	$818	4,326	3,507	2,528
Carlsbad	$419	4,326	683	4,667
Carol Stream	$1,306	4,326	2,926	2,528
Carrollton	$2,201	4,326	3,102	2,528
Cary	$505	4,326	2,084	2,465
Cedar Hill	$122	4,326	3,102	2,528
Cedar Rapids	$278	4,326	431	2,528
Champaign	$152	4,326	2,926	2,528
Chandler	$1,077	4,326	2,828	4,667
Chapel Hill	$14	4,326	2,084	2,465
Charlotte	$13,693	4,326	2,084	2,465
Charlottesville	$103	4,326	3,607	2,465

That's because a city named "Burlington" exists within multiple states (Iowa, North Carolina, and Vermont). Therefore, **State** resolves to *, meaning "more than one value." All marks where **State** resolves to * are evaluated together, so any other city that also exists in multiple states would also have a prediction of $9,366.

For **Predict_Sales_Region**, we're using ATTR([Region]) as a predictor. The *predictor* is partitioning all the **City** data by **Region**. You expect to see identical results *within* a region, but different results for *each* region:

City	Sales	Predict_Sales_City along City	Predict_Sales_State along City	Predict_Sales_Region along City
Burbank	$3,247	4,326	6,260	4,667
Burlington	$21,668	4,326	9,366	9,647
Caldwell	$456	4,326	875	4,667
Camarillo	$1,468	4,326	6,260	4,667
Cambridge	$502	4,326	1,038	6,574
Canton	$818	4,326	3,507	2,528
Carlsbad	$419	4,326	683	4,667
Carol Stream	$1,306	4,326	2,926	2,528
Carrollton	$2,201	4,326	3,102	2,528
Cary	$505	4,326	2,084	2,465
Cedar Hill	$122	4,326	3,102	2,528
Cedar Rapids	$278	4,326	431	2,528
Champaign	$152	4,326	2,926	2,528
Chandler	$1,077	4,326	2,828	4,667
Chapel Hill	$14	4,326	2,084	2,465
Charlotte	$13,693	4,326	2,084	2,465
Charlottesville	$103	4,326	3,607	2,465
Chattanooga	$989	4,326	2,115	2,465
Chesapeake	$5,997	4,326	3,607	2,465
Chester	$1,968	4,326	16,471	6,574
Cheyenne	$1,603	4,326	1,603	4,667
Chicago	$48,540	4,326	2,926	2,528

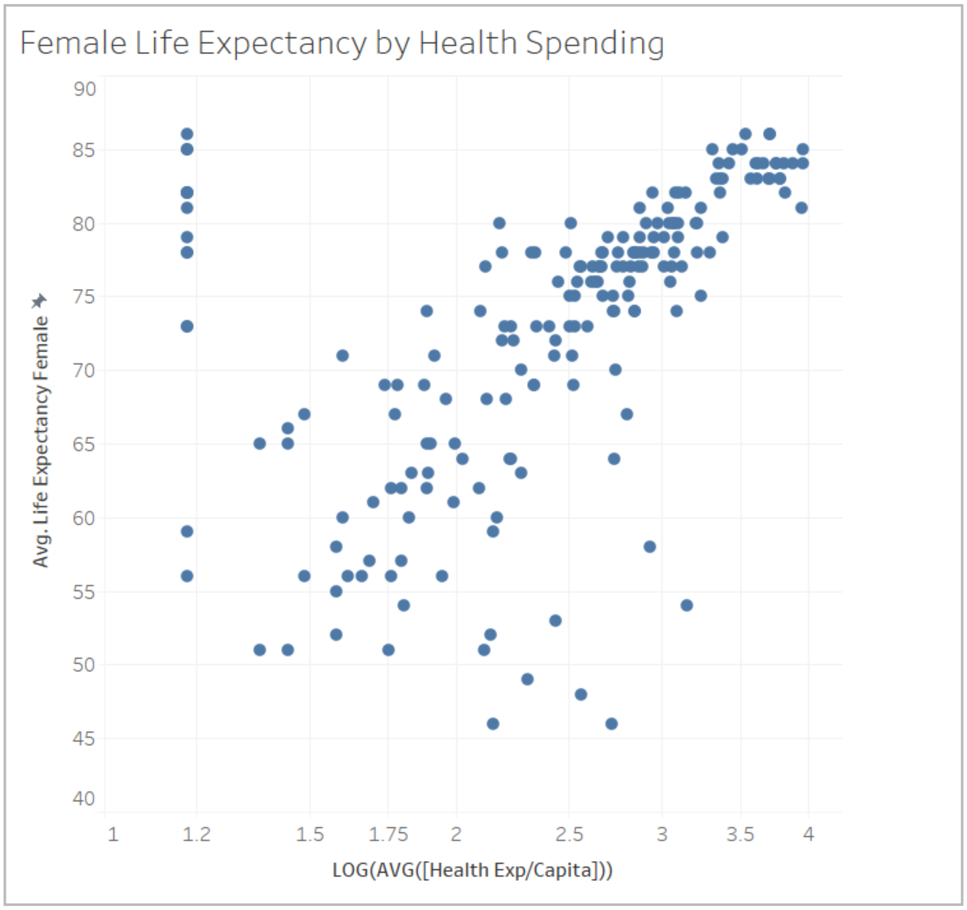
Again, since Burlington exists within multiple regions (Central, East, and South), **Region** resolves to *. Burlington's predictions will match only those cities that also exist within multiple regions.

As you can see, it's very important to ensure that any dimensional predictors are correctly aligned with both your visualization's level of detail and your partitioning. Subdividing your visualization by any dimension could have unintended effects on your predictions.

Example - Explore Female Life Expectancy with Predictive Modeling Functions

This example uses the World Indicators saved data source, which comes with Tableau. We'll use the MODEL_QUANTILE and MODEL_PERCENTILE predictive modeling functions to explore the relationships between health spending per capita, female life expectancy, birth rate.

Let's start with a visualization that compares each country's health spending with its female life expectancy. To follow along and access the pre-built views and dashboards, or to view the solution, download the following workbook from Tableau Public: Predictive Modeling of Female Life Expectancy.



Using MODEL_PERCENTILE

First we'll evaluate the life expectancy and health expenditures for all visible marks. This will allow Tableau to build a model from those marks and return the percentile for each within the model.

Step 1: Create the prediction calculation

If you also have Tableau Server or Tableau Online and you want to do your authoring on the web instead of in Tableau Desktop, publish the workbook to your Tableau server, click **Workbooks**, select the workbook, then under Actions, choose **Edit Workbook**.

Once you open the workbook, you'll see that it has several sheets. You'll be using those sheets to build your views.

1. In the starter workbook, click the **Percentile Starter** sheet.
2. Open the **Analysis** menu at the top, and then select **Create Calculated Field**.
3. In the Calculation Editor, do the following:

- Name the calculation: **Percentile Expectancy vs Spending**
  Note: If you're following along in the starter workbook, you'll see different names for the calculations used throughout this example. This is so you'll be able to name the fields as described here, without creating duplicates. Your solution may have different names, but the visualizations should look the same.
- Enter the following formula:

  ```
  MODEL_PERCENTILE(AVG([Life Expectancy Female]), LOG
  (MEDIAN([Health Exp/Capita])))
  ```

  This calculation uses average life expectancy as the target expression, and median health expenditure as the predictor. In this case, we used a logarithmic transformation on the health spending axis, as well as for the predictor.
  Note: Because this data set skews toward very large values, we transformed our data using a logarithmic scale, which is useful when analyzing data with a few values that are much higher than the rest, and will make it easier to identify trends and relationships in our data set.

4. Click **OK**.

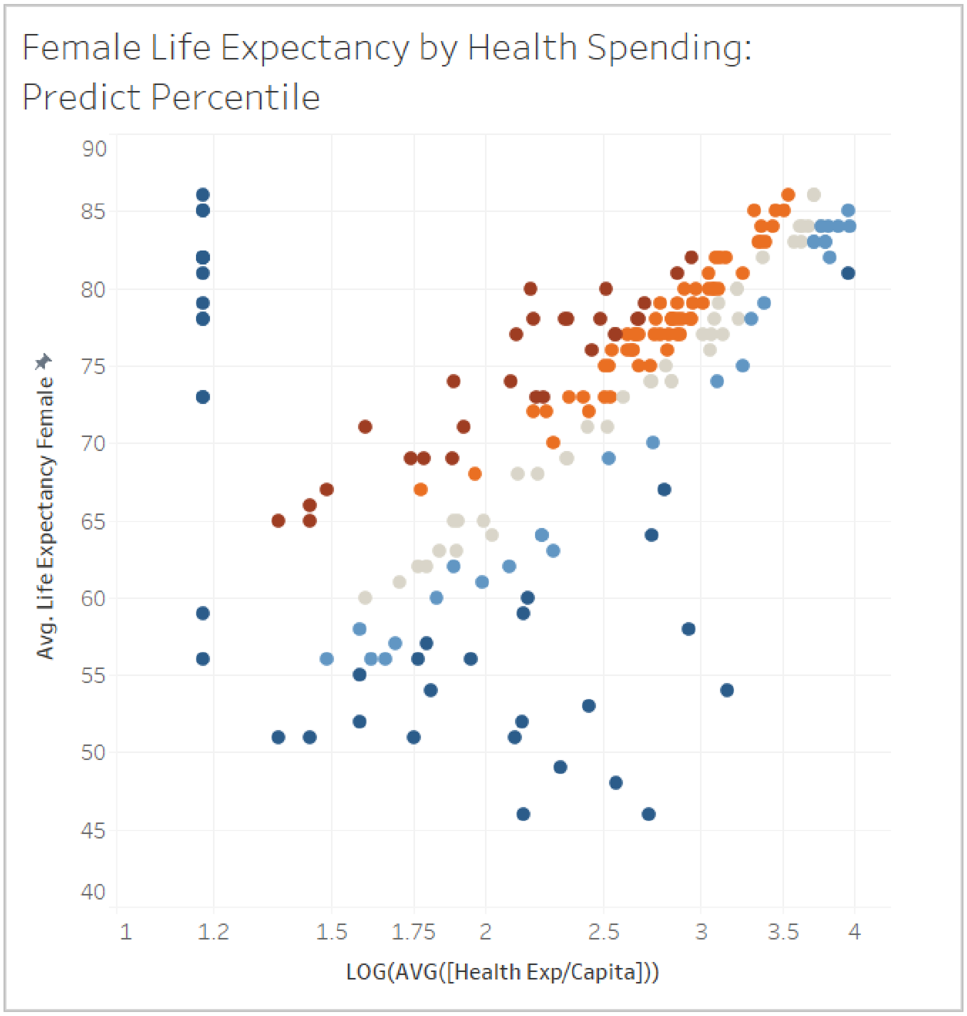
   The prediction calculation is now added as a calculated field in the Data pane.

Step 2: Add the prediction calculation to the view

In the viz above, you can see each country's health spending against its female life expectancy, filtered to 2012.

Now, let's add the MODEL_PERCENTILE calculation to the view and see what insights we can gain.

1. Drag **Percentile Expectancy vs Spending** to Color on the Marks card.
2. Click the drop-down arrow on the pill and select **Compute Using > Country/Region**.
3. Click **Color** on the Marks card, and then click **Edit Colors**.
   - Under Palette, select **Orange-Blue Diverging**.
   - Select the **Stepped Color** checkbox.
   - Select the **Reversed** checkbox.
4. Click **OK**.



You can see the distribution of countries where health expectancy is both higher and lower than expected based on the level of spending. Notice that generally, the dark red marks indicate that life expectancy is high relative to healthcare spending, dark blue means that life expectancy is low relative to healthcare spending, and grey means that life expectancy is close to what the model expects, based on the level of healthcare spending.

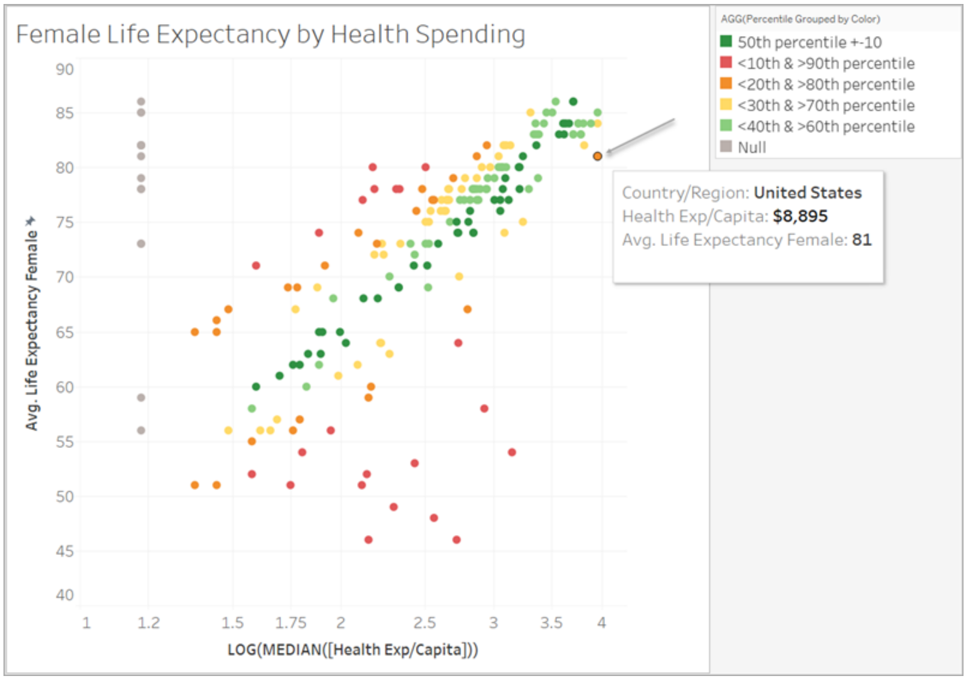
Step 3: Group the results by color

To simplify analysis, let's use the prediction calculation within a new calculation to group the results. We'll build groups so that marks above the 90th percentile and below the 10th percentile are grouped together, marks in the 80th-90th percentile range and 10th-20th percentile range are grouped together, and so on. We'll also highlight marks with a null value and address those later using the other predictive modeling function, MODEL_QUANTILE.

1. In the Calculation Editor, do the following:
   - Name the calculation: **Percentile by Color.**
   - Enter the following formula:

```
IF
ISNULL([Percentile Expectancy vs Spending])
THEN "Null"
ELSEIF [Percentile Expectancy vs Spending] >=0.9 OR
[Percentile Expectancy vs Spending] <=0.1
THEN "<10th & >90th percentile"
ELSEIF [Percentile Expectancy vs Spending] >=0.8 OR
[Percentile Expectancy vs Spending] <=0.2
THEN "<20th & >80th percentile"
ELSEIF [Percentile Expectancy vs Spending] >=0.7 OR
[Percentile Expectancy vs Spending] <=0.3
THEN "<30th & >70th percentile"
ELSEIF [Percentile Expectancy vs Spending] >=0.6 OR
[Percentile Expectancy vs Spending] <=0.4
THEN "<40th & >60th percentile"
ELSE "50th percentile +-10"
END
```

2. Add the new calculation to **Color** on the Marks card.
3. Click the drop-down arrow on the pill and select **Compute Using > Country/Region**.
4. Click Color on the Marks card, and then click **Edit Colors**.
   - Adjust the colors to better see the trend. In this case, let’s use the **Traffic Light** color palette, and use gray for Nulls.
5. Click **OK**.

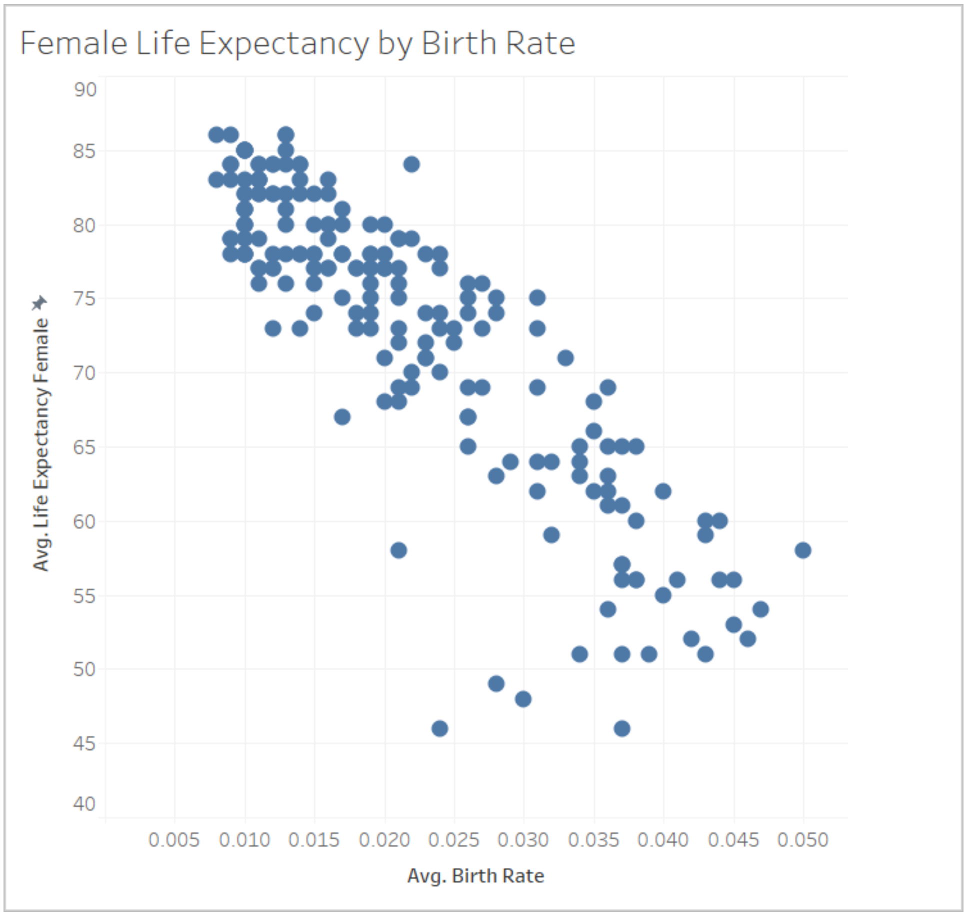


Looking at the orange mark in the corner, notice that the U.S. spends $8,895 per female for a life expectancy of 81 years. Moving along the X-axis to the left, you can see that other countries spend less and have the same life expectancy.

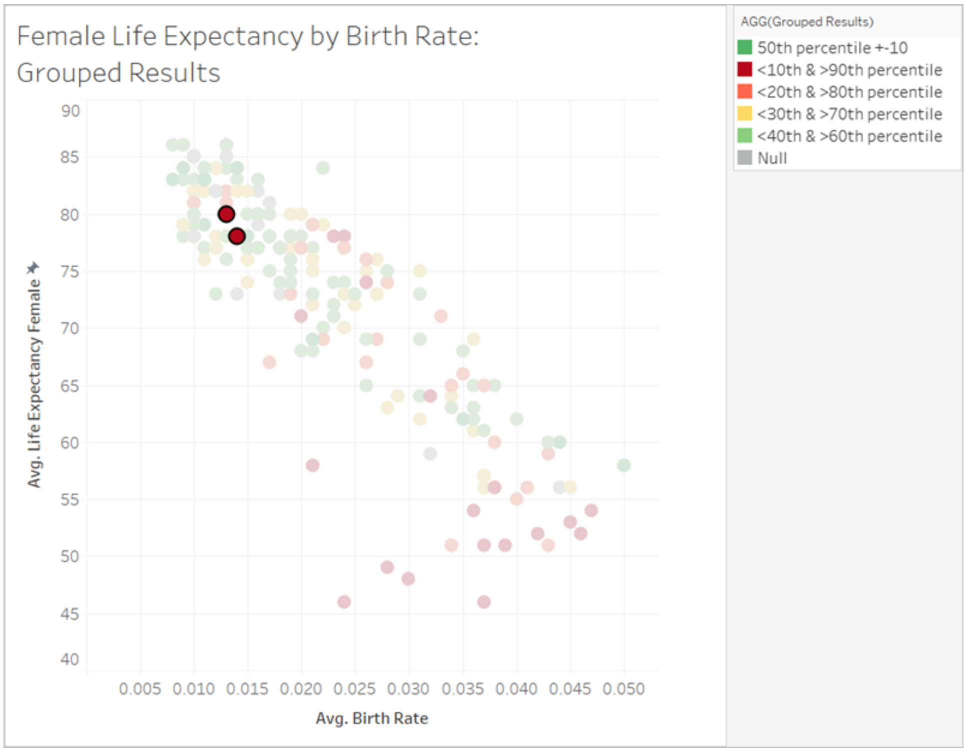
The model evaluates the strength of the relationship at each point, where the U.S. is close to the upper end of the model's expected range.

Step 4: Compare life expectancy with birth rate

Next, let's look at a viz that compares female life expectancy with birth rate. Notice that there is a negative correlation between birth rates and female life expectancy; however, this does not mean that higher birth rates cause lower female life expectancy. There are likely additional factors that affect both birth rates and female life expectancy that are not visible in this view of the data. But let's add the model and see where the model expects female life expectancy to be higher or lower given health expenditures.

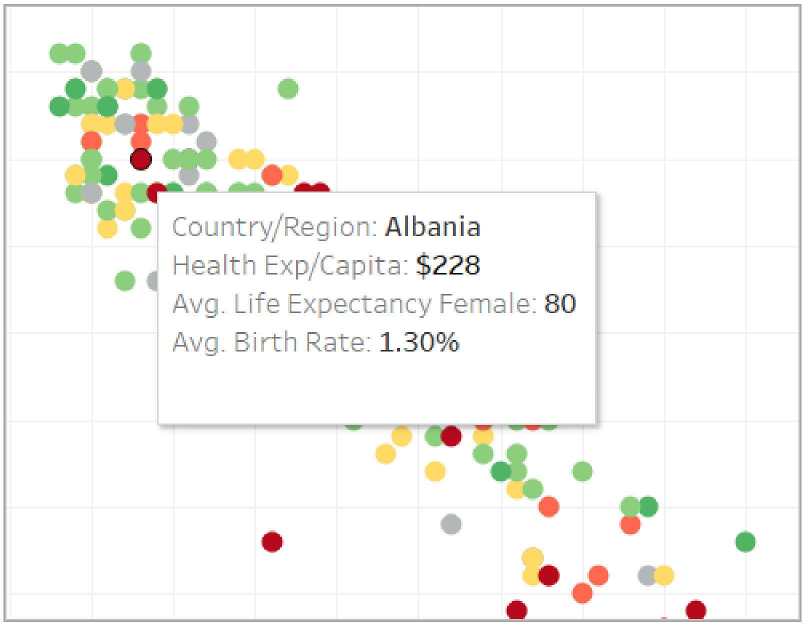


1. On **the Birth Rate** sheet, add the **Percentile by Color** prediction calculation to **Color** on the Marks card to bring it into the view.
2. Click the drop-down arrow on the pill and select **Compute Using > Country/Region**.
3. Click Color on the Marks card, and click **Edit Colors**. Edit the colors as before, using the **Traffic Light** palette and gray for Null.
4. Click **OK**.



Now the data is much more distributed. The red band in the lower right corner is where life expectancy is lowest but the birth rate is highest, and healthcare spending relative to life expectancy is low. Singling out the two red marks in the top left quadrant, which pertain to Albania and Armenia, you'll notice that both countries have high female life expectancy, lower birth rates, and low health expenditures.

As you can see, we were able to use MODEL_PERCENTILE to identify that these two countries are outliers: Even though they both had relatively low healthcare spending, they still have relatively high life expectancies, placed in the context of birth rate.



Now, let's see how you can use the other predictive modeling function, MODEL_QUANTILE, to continue your analysis.

Using MODEL_QUANTILE

MODEL_QUANTILE is used to generate numeric predictions, given a target percentile, target expression, and predictors. It's the inverse of MODEL_PERCENTILE.

Recall that we have several null values in our results, meaning that some countries don't have health expenditure data. We'll use MODEL_QUANTILE to estimate those missing values.

Step 1: Create the prediction calculations

We've been working with this calculation:

```
MODEL_PERCENTILE(AVG([Life Expectancy Female]), LOG(MEDIAN
([Health Exp/Capita])))
```

First, we want to invert this function in order to get a prediction for healthcare expenditure based on female life expectancy.

1. In the starter workbook, click the **Quantile Starter** sheet.
2. Open the **Analysis** menu at the top, and then select **Create Calculated Field**.
3. In the Calculation Editor, do the following:
   - Name the calculation: **Quantile of Life Expectancy vs Spending**
   - Enter the following formula:

```
POWER(10, MODEL_QUANTILE(0.5, LOG(MEDIAN([Health
Exp/Capita])), AVG([Life Expectancy Female])))
```

Let’s break this calculation down to better understand it:

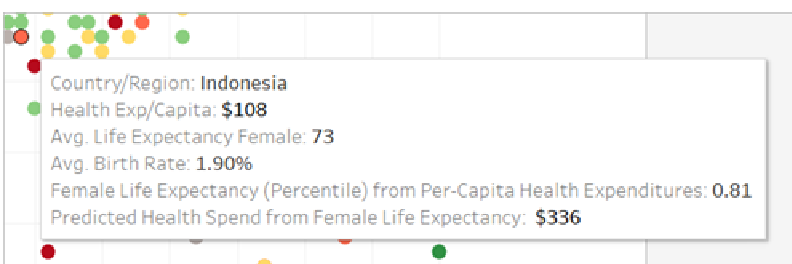
- We start with MODEL_QUANTILE, where the first argument is 0.5, specifying what percentile to predict.
- The target expression is median health expenditure per capita.
- The predictor is average female life expectancy.
- In addition, we wrapped the function inside a POWER function to convert the log-transformed target expression back into dollars.

4. Click **OK**.

   The prediction calculation is now added as a calculated field in the Data pane.

Step 2: Add the prediction calculation to the view

1. Drag **Quantile of Life Expectancy vs Spending** to Tooltip on the Marks card.
2. Click the drop-down arrow on the pill and select **Compute Using > Country/Region**.
3. Click **Tooltip** on the Marks card and add a row for the MODEL_QUANTILE prediction:
   - Name the tooltip row: **Predicted Health Spend from Female Life Expectancy:**
   - Click **Insert** and select the calculation to ensure the tooltip will dynamically show the mark’s unique prediction as you interact with the viz.
4. Click **OK**.



Right now our MODEL_QUANTILE calculation only has one predictor: female life expectancy. Notice as you move from left to right, viewing the tooltip for marks with the same life expectancy, that each one has the same predicted health spend. Indonesia’s predicted health spend from life expectancy is $336, but so is Fiji’s, Egypt’s, and other countries that have the same female life expectancy.

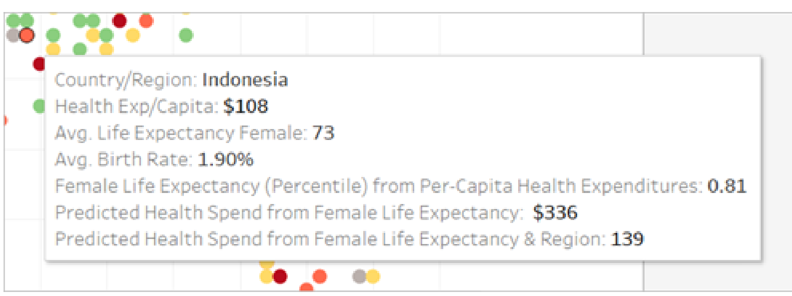
This is because the model provides the same estimated expenditure for each mark. Since we're only using one predictor (female life expectancy), the model returns the same results for all marks where the predictor has the same value. You can add more nuance to the model by adding predictors.

Step 3: Add a prediction with a second predictor

As you build your predictions, you should consider which fields will be good predictors for your target values and include them in your calculations. You can combine any number of dimensions and measures. For example, we could add GDP, population, and other fields as predictors to improve our predictions. In this case, let's add **Region**.

1. In the Calculation Editor, do the following:
   - Name the calculation: **Quantile Spend vs Expectancy & Region**
   - Enter the following formula, which is the same as the previous calculation, but adds **Region** as a predictor:

     ```
     POWER(10, MODEL_QUANTILE(0.5, LOG(MEDIAN([Health
     Exp/Capita])), AVG([Life Expectancy Female]), ATTR
     ([Region])))
     ```
2. Click **OK**.
3. Next, add the new calculation to Tooltip on the Marks card.
4. Click **Tooltip**, and add another line to describe the new prediction, such as **Predicted Health Spend from Female Life Expectancy & Region:**.



Now the tooltip shows both predictions.

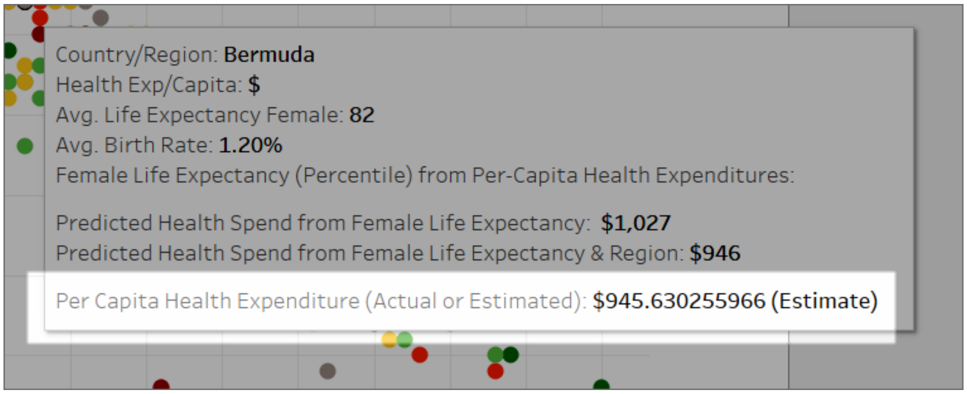
Step 4: Compare actual values to predicted values

As a final step in your analysis, you can also build prediction calculations that combine actual and predicted values. In our example, let’s show the actual health expenditures where available and the estimated expenditures where unavailable.

1. In the Calculation Editor, do the following:
   - Name the calculation: **Health Spend Actual vs Prediction Value**
   - Enter the following formula, which will return the numeric value of the prediction::

     ```
     ROUND(IFNULL(AVG([Health Exp/Capita]),[Quantile_HE/Cap_
     LEF,Region]),0)
     ```
2. Click **OK**.
3. Create another calculation as follows:
   - Name the calculation: **Health Spend Actual vs Prediction Tag**
   - Enter the following formula, which will act as a label for the above calculation:

     ```
     STR(IF ISNULL(AVG([Health Exp/Capita])) THEN "
     (Estimate)" ELSE "(Actual)" END)
     ```
4. Click **OK**.
5. Next, add both calculations to **Tooltip** on the Marks card.
6. Click **Tooltip**, and add another line to describe the new calculations:
   - **Per Capita Health Expenditure (Actual or Estimated):**
   - Insert the new calculations, one after the other.
7. Click **OK**.



Now, as you interact with the viz, you can see each country's per capita health spending, or view an estimate if the actual value was missing (null) from the data.

Just like that, you can use predictive modeling functions in Tableau to gain insights into your data.

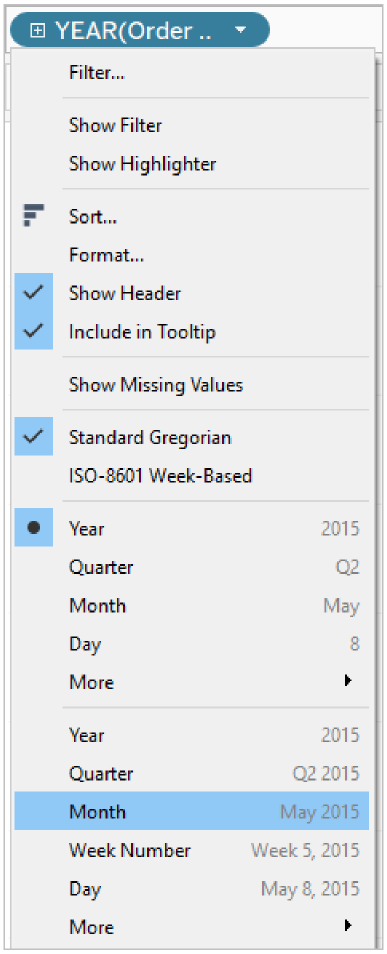
Predictive Modeling Functions in Time Series Visualizations

This document describes the steps required to extend a time axis into the future, so that you can make predictions for future dates using predictive modeling functions.

To see how, follow along with the below example.

Step 1: Create a visualization

1. In Tableau Desktop, connect to the **Sample - Superstore** saved data source, which comes with Tableau.
2. Navigate to the **Sheet 1** worksheet.
3. From the Data pane, drag the **Order Date** dimension to the Columns shelf.
4. Open the measure's context menu to change its list level to Month and Year:



5. Drag **Profit** to the Rows shelf.

Step 2: Create a prediction calculation

1. Click to open the **Analysis** menu at the top, and then select **Create Calculated Field**.
2. In the Calculation Editor, do the following:
   - Name the calculation: **Quantile Predict Median Profit**.
   - Enter the following formula:

```
MODEL_QUANTILE(0.5, SUM([Profit]),ATTR(DATETRUNC
('month', [Order Date])))
```

The MODEL_QUANTILE function takes a given quantile and predicts values based on the predictors you input. In this case, the quantile = 0.5, which predicts the median. We want to predict profit, so the target expression is SUM([Profit]). And since we want to base the prediction on past performance, we need to include date as a predictor.
Note: With Predictive Modeling functions, you cannot mix aggregate and non-aggregate arguments. Since the target expression must be an aggregate, so must the predictor. In this case, we used ATTR(DATETRUNC('month', [Order Date])). For more information about the use of ATTR functions, see When to Use the Attribute (ATTR) Function.

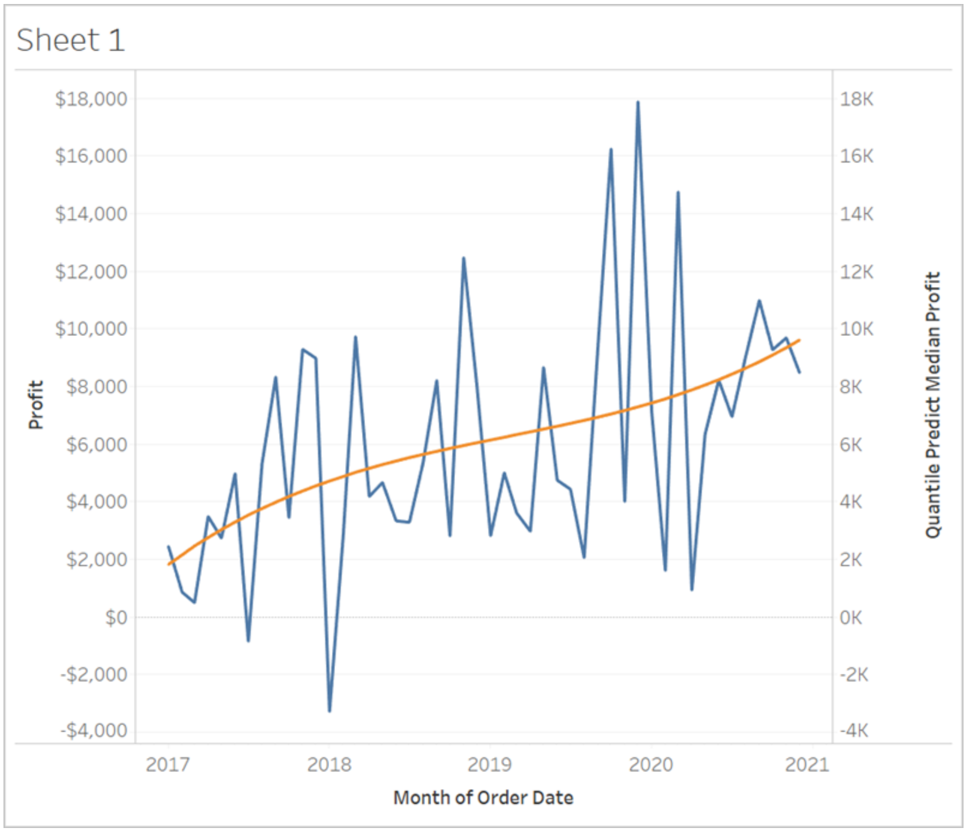
3. When finished, click **OK**.

   The prediction calculation is now added as a calculated field in the Data pane.

Step 3: Add the prediction calculation to the view

1. Drag the prediction calculation to the Rows shelf, to the right of SUM(Profit).
2. Right-click (Control-click on Mac) the measure and select **Dual Axis**.
3. To align the two axes in a dual axes chart to use the same scale, right-click (Control-click on Mac) the secondary axis, in this case **Quantile Predict Median Profit**, and select

Synchronize Axis. This aligns the scale of the two axes.



Step 4: Extend the date range and densify the data

The final step is to extend the horizontal axis into the future so that you can add marks beyond the current date range.

1. To do this, open the context menu of the MONTH(Order Date) pill, and select **Extend Date Range**.
2. In the Extend Date Range dialog, let's extend the axis by 16 months:

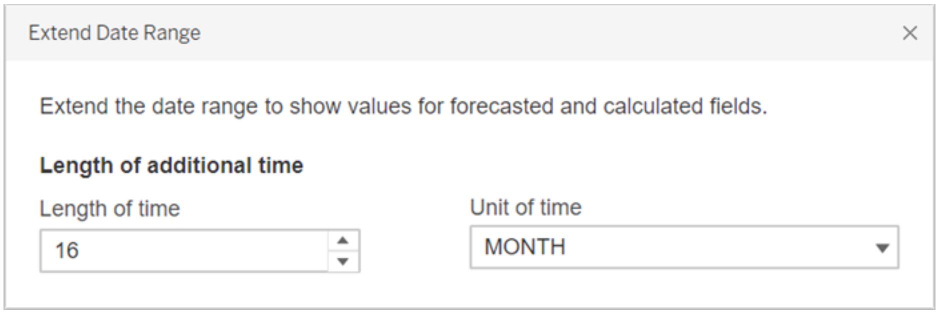
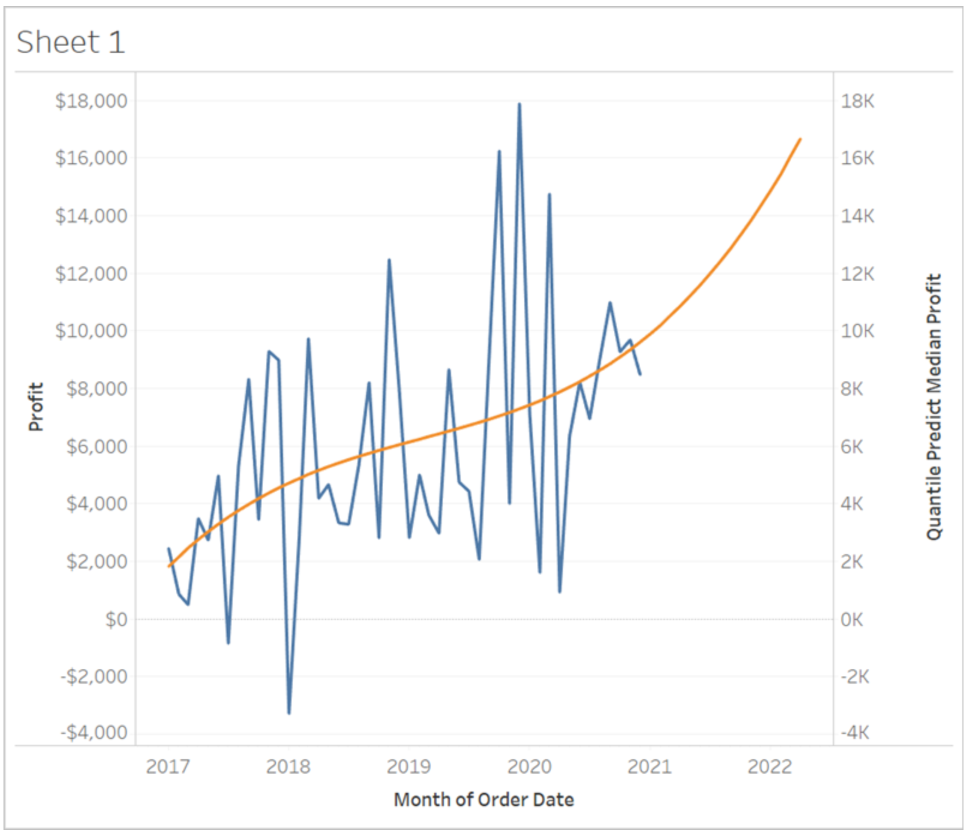


Tableau has automatically updated the view by extending the horizontal axis and includes predictions on these future dates:



Notice that the blue **Profit** line doesn't extend to the end of the graph. However, the orange line representing the prediction calculation, **Quantile Predict Median Profit**, *does* extend to the end of the graph. Computing the predicted profit on future dates is an example of data densification, which occurs when Tableau creates additional marks in the view to compensate for missing values. Data is not added to the underlying data source; marks are simply generated and added to the view.

Note: When you extend a date range into the future, Tableau automatically enables the **Infer Properties from Missing Values** option in the **Analysis** menu. This ensures that your prediction calculation will work on the missing values. For more detail on this menu option, see **Predictive Modeling with Generated Marks** below.

For more information on data densification, see this Tableau Conference presentation, You Did What to Your Data? A Deep Dive into Imputing and Densifying Data. Also check out this data densification post from Data Plus Science.

Predictive Modeling with Generated Marks

If you've been using Tableau for a while, you may have heard of the phrase "data densification." This refers to a process in which marks are generated by Tableau and added to the view, even though those marks aren't supported by records in the underlying data source. This might be done to extend a date axis, or if you're working with predictive modeling functions, to show predictions.

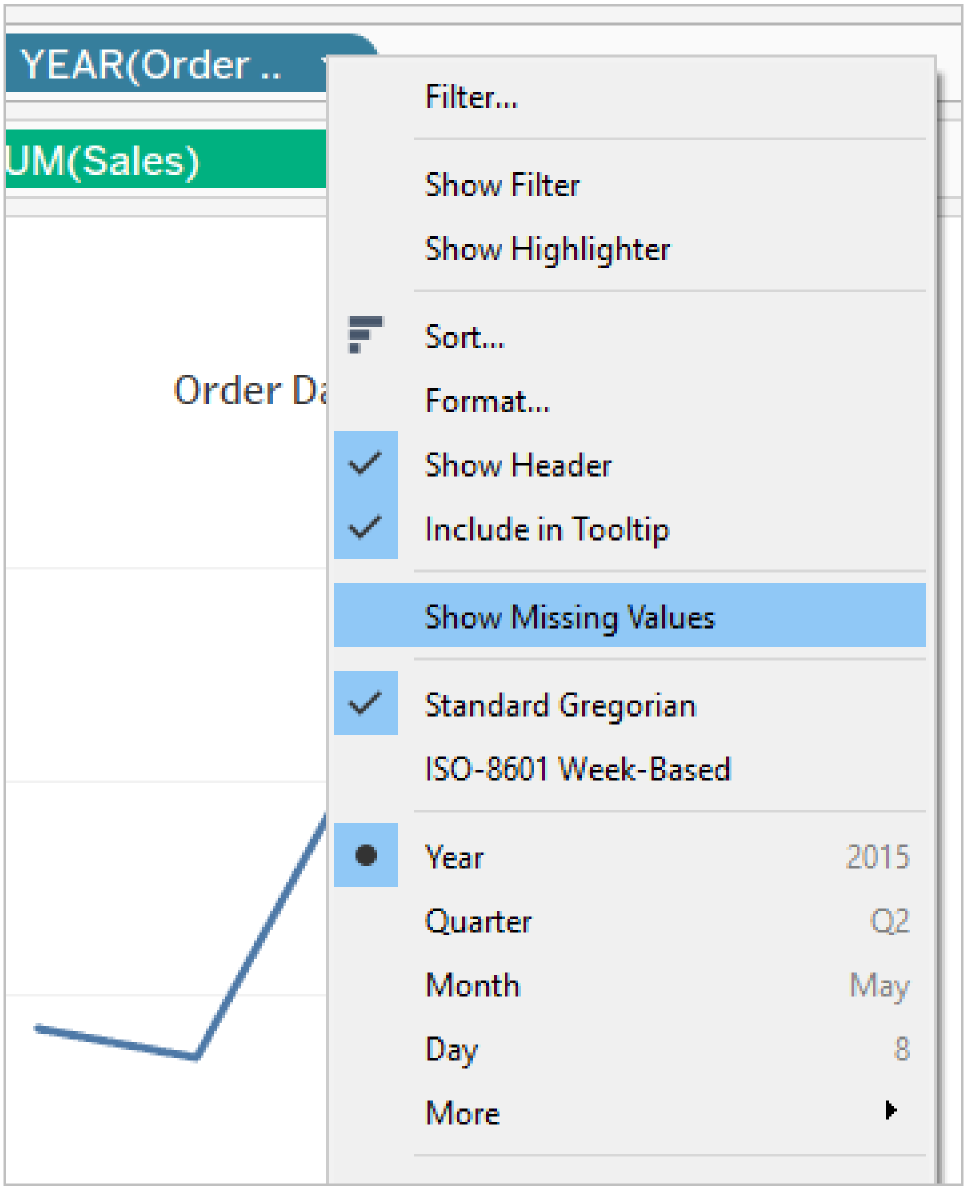
Watch a video: For an in-depth discussion on densifying data, check out this presentation from TC19: You Did What to Your Data? A Deep Dive into Imputing and Densifying Data.

Learn more: See this blog post on data densification from Data Plus Science.

Calculate predictions on missing values

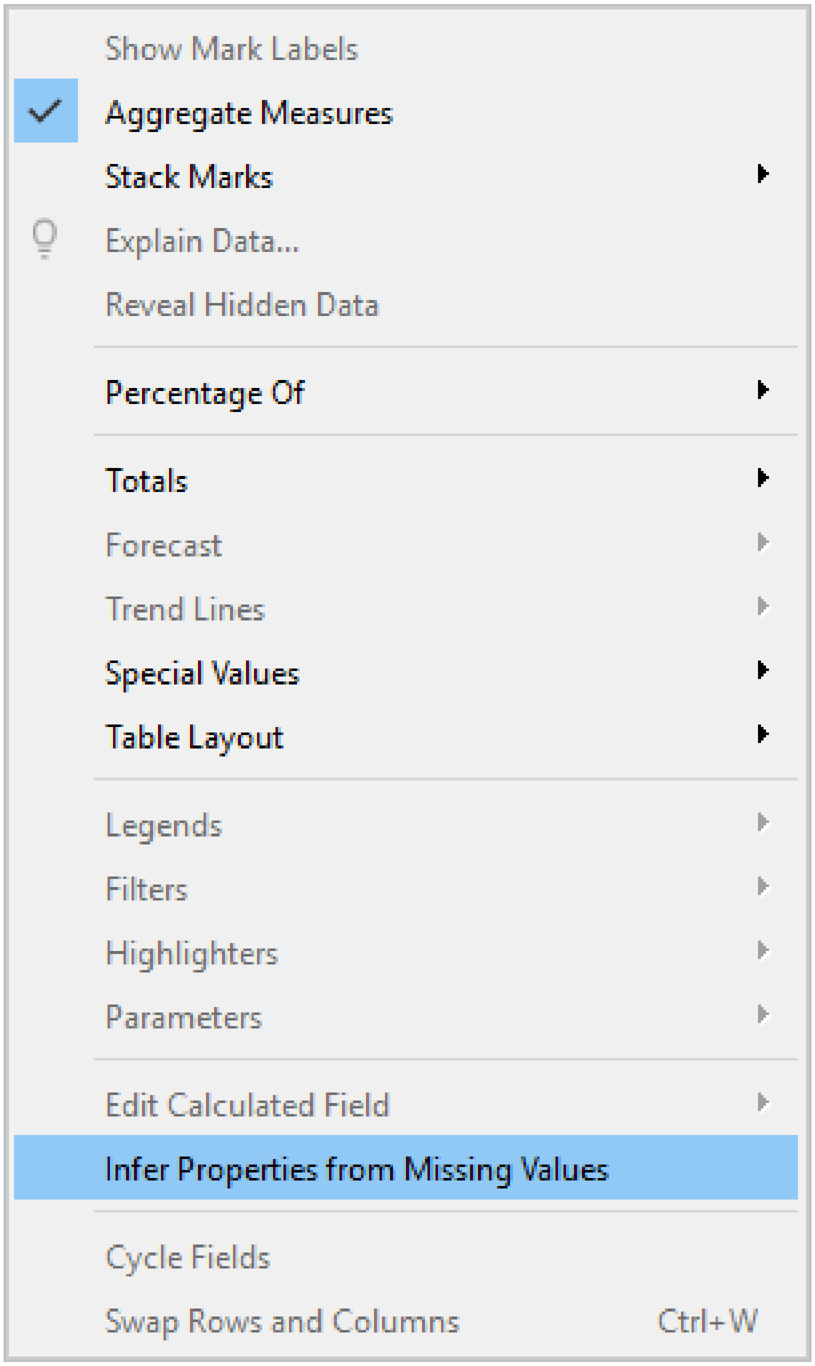
For example, you might want to add predictions for future dates. By default, missing values in Tableau aren't shown, but you can generate these marks as follows:

1. Right-click (control-click on Mac) the date or bin header.
2. Select **Show Missing Values**.



But this isn't enough to let you make predictions on those generated marks. If you tried to perform a calculation on them (whether that's a prediction calculation or not), Tableau would return null values. This is expected, since those marks are based on missing values that don't exist.

To make predictions on those missing values, open the **Analysis** menu at the top, and then select **Infer Properties from Missing Values**.



Note: The default setting is that **Infer Properties from Missing Values** is turned off, even when **Show Missing Values** is turned on. Any field or axis that can be expanded using **Show Missing Values** can have this setting applied.

Example of predictions on generated marks

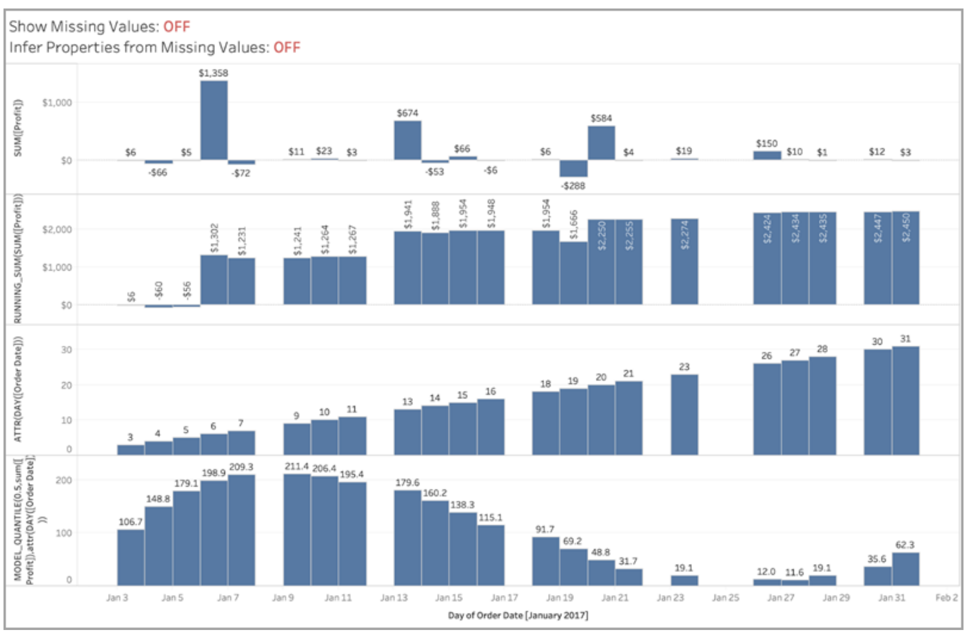
Now let's explore this behavior further. We'll compare three different illustrations showing how the **Show Missing Values** and **Infer Properties from Missing Values** settings can affect your visualization, depending on whether one or both are turned on or off. To follow along, download the following workbook from Tableau Public: Predictions on Missing Values.

We've included predictions using ATTR(DAY([Order Date])) as a predictor. This isn't the best predictor for the data (and yields inadequate predictions), but for the purposes of this article, it's a good illustration of **Infer Properties from Missing Values**.

Each viz includes the same four measures on the Rows shelf, as outlined below:

- Row 1: `SUM([Profit])`
- Row 2: `RUNNING_SUM(SUM([Profit]))`
- Row 3: `ATTR(DAY([Order Date]))`
- Row 4: `MODEL_QUANTILE(0.5, SUM([Profit]),ATTR(DAY([Order Date])))`

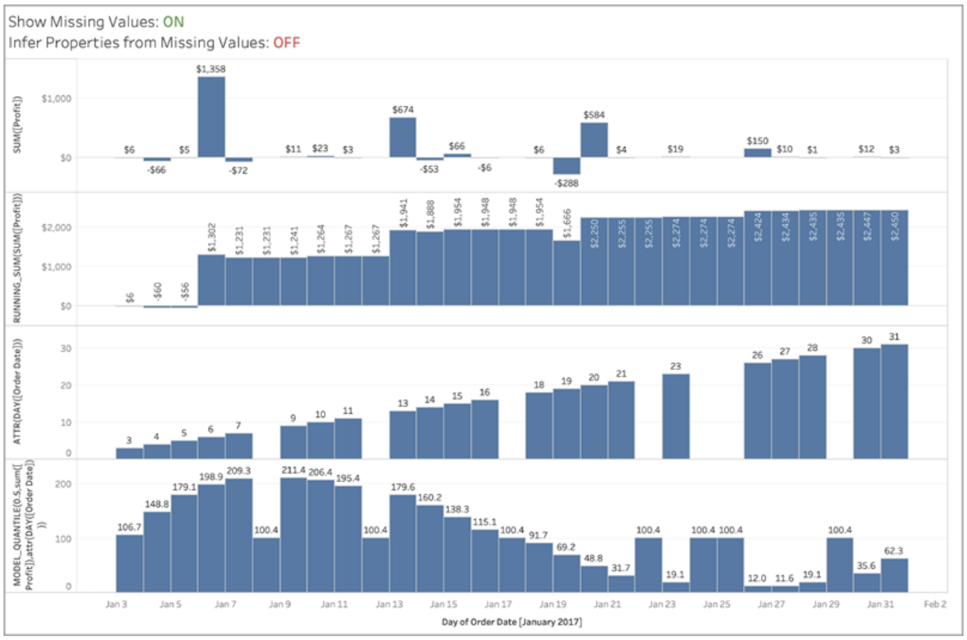
Illustration 1



In the above image, both **Show Missing Values** *and* **Infer Properties from Missing Values** *are turned off, which are the default settings in Tableau.*

You would see the same viz if **Infer Properties from Missing Values** *was turned on and* **Show Missing Values** *was turned off. This is because* **Infer Properties from Missing Values** *depends on* **Show Missing Values** *being turned on.*

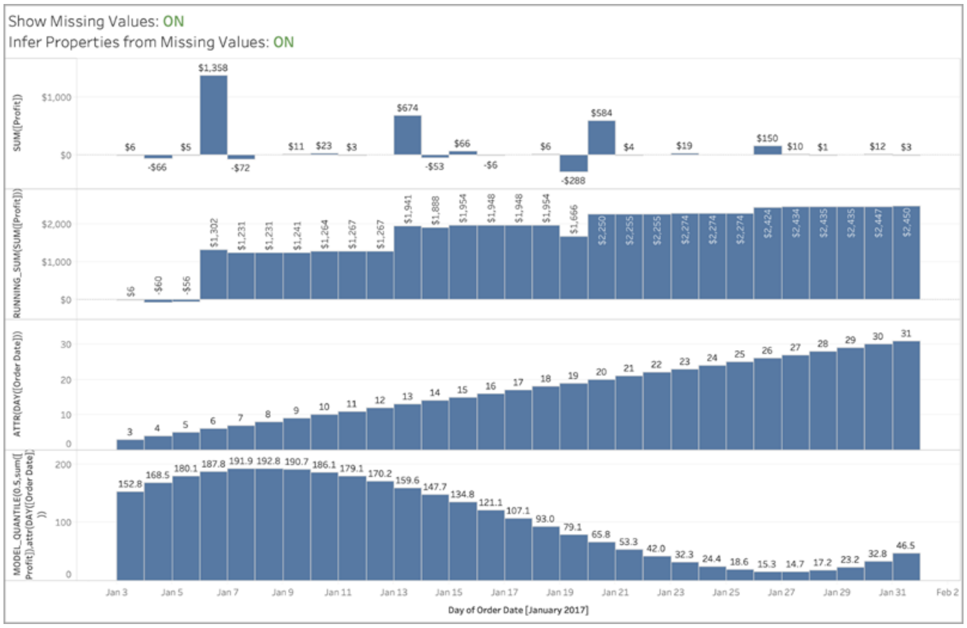
Illustration 2



In the above image, **Show Missing Values** *is turned on and* **Infer Properties from Missing Values** *is turned off. The default setting is that* **Infer Properties from Missing Values** *is turned off, even when* **Show Missing Values** *is turned on.*

Note that in this situation, we do not compute a value for ATTR on DAY([Order Date]) for the missing values (Row 3). We do generate a prediction for the densified dates, but they are identical for all missing dates, since we're not able to infer the actual ATTR(DAY([Order Date])) as shown in Row 3. Effectively, those marks are being computed as if DAY([Order Date]) is null.

Illustration 3



In this image, both **Show Missing Values** *and* **Infer Properties from Missing Values** *are turned on, illustrating the* **Infer Properties from Missing Values** *setting in action.*

As you can see, since we're able to infer the ATTR(DAY([Order Date])) (Row 3), we're able to use it in predictions in Row 4, returning a nice smooth curve of predictions.

Regularization and Augmentation in Predictive Modeling

Many people who use Tableau may not use predictive modeling, let alone look for ways to improve a predictive model's fit and prediction quality. This article is for advanced users who are interested in this area of data science.

In addition to the default arguments such as target expression (the measure to predict) and predictor expression (the measure and/or dimensions used to make the prediction), you can add two more arguments to fine-tune your predictions: *lambda*, a regularization parameter, and *augmentation*. This involves adding the new arguments to the syntax of your calculation.

Which models work with regularization and augmentation?

As a reminder, predictive modeling functions in Tableau support three models: linear regression (also known as ordinary least squares regression, or OLS), regularized linear regression (or ridge regression), and Gaussian process regression. If you're using linear or ridge regression, then augmentation allows you to increase the ability of your models to pick up non-linear patterns. If you're using ridge regression, then the regularization parameter is a scalar you can use to adjust the regularization effect on your model.

Regularization and augmentation are not applicable to Gaussian process regression.

Before discussing regularization and augmentation further, let's review these two models:

Linear regression is best used when there are one or more predictors that have a linear relationship between the prediction and the prediction target, they aren't affected by the same underlying conditions, and they don't represent two instances of the same data (for example, sales expressed in both dollars and euros).

Regularized linear regression is used to improve stability, reduce the impact of collinearity, and improve computational efficiency and generalization. In Tableau, L2 regularization is used. For more information on L2 regularization, see this lesson on Ridge Regression.

What is regularization?

Ridge regression is a specific kind of regularized linear regression. Regularization imposes a penalty on the size of the model's coefficients. The strength of the regularization is controlled by lambda, a scalar used to fine-tune the overall impact of regularization. The higher the value, the heavier the penalty (i.e. the greater the regularization).

Ridge regression addresses some of the problems of linear regression:

- It removes pathologies introduced by multicollinearity among predictors.
- If the least-square problem is ill-conditioned, such as if the number of data points is less than the number of features, then lambda will select a unique solution.
- It provides a way of improving the generalization of the linear model.

By default, ridge regression in Tableau has **lambda=0.5** because this value works well in many cases. To change the lambda value, simply edit the table calculation (examples below).

What is augmentation?

Augmentation in MODEL_QUANTILE and MODEL_PERCENTILE is a simple example of data augmentation: predictors are expanded to higher order polynomials. In Tableau, there are a couple types of polynomial augmentations built in to the predictive modeling functions.

- For **ordered dimensions**, Legendre polynomials up to order 3 allow the linear model to pick up quadratic and cubic relations between the augmented predictor and response.

- For **measures**, 2nd degree Hermite polynomials allow the linear model to pick up quadratic relations between the augmented predictor and response.

In linear regression, only ordered dimensions are augmented by default with **augmentation=on**; in ridge regression where **model=rl**, only measures are augmented by default. To override the setting and disable augmentation for each predictor in your calculation, use **augmentation=off**; no higher order polynomials will be added.

Turning off augmentations is advantageous when the data set is very small because the augmentations could overfit any noise present in the original data, and also because the resulting relationship is simpler and more intuitive.
Note: Augmentation is turned on/off on a per-predictor basis (i.e. within the predictor argument of your calculation). Lambda (and model) are applied at the higher level (outside any predictor expressions).

Configuring lambda and augmentation in your calculation

Now that you know about the regularization parameter (or lambda) and data augmentation, let's see them in the context of a prediction calculation:

```
MODEL_QUANTILE("model=rl, lambda=0.05", 0.5, SUM([Profit]),
"augmentation=off", SUM([Sales]))
```

Below is a table that quickly summarizes if changing the augmentation and lambda from the default affects the linear models:

	Augmentation	Lambda
Ridge regression	Yes	Yes
Linear regression	Yes	Not applicable

Considerations for regularization and augmentation

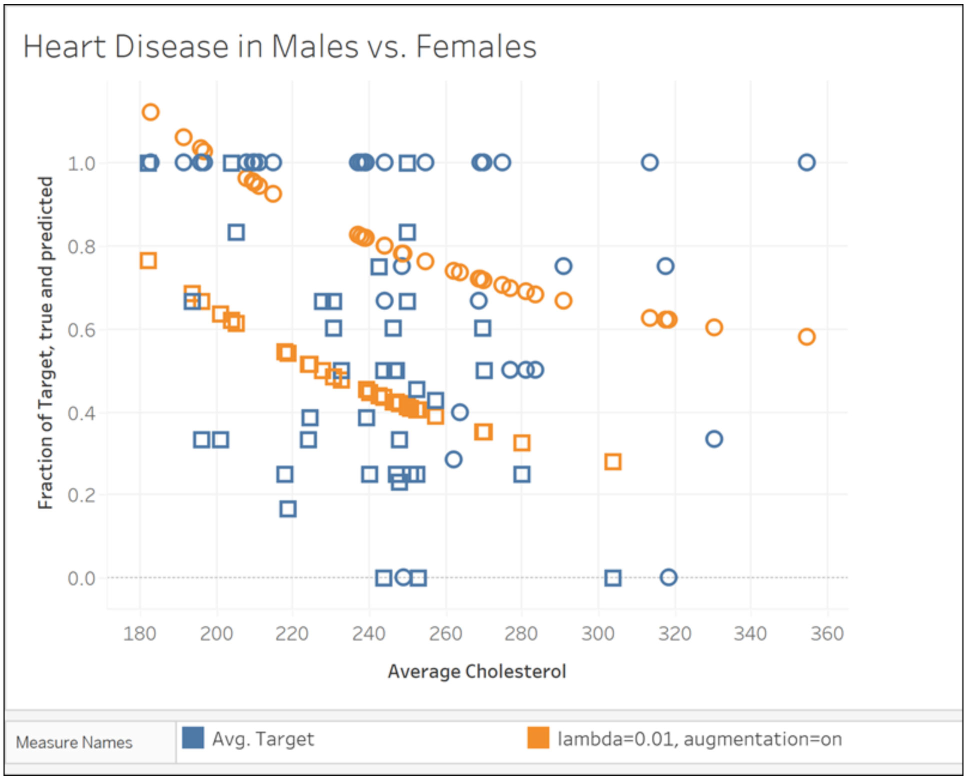
- If you have the wrong model for your data, then changing the regularization parameter or augmentation isn't likely to produce significantly better results. Consider reviewing if the data types are correct (measures vs. dimensions). If the underlying data is a time series, for instance, consider using Gaussian process regression instead, by changing the model in your table calculation with **model=gp**.

- Because OLS is not regularized, there is no lambda value that can be changed.
- If your data set is extremely small and you have dimensions (especially high cardinality dimensions), then consider using ridge regression by adding model=rl to your table calculation.
- All things being equal (for the same data set, given augmentation is on or off), a low lambda may improve fit, but hurt generalization (cause overfitting).
- Conversely, a high lambda may push the fit to be a constant model, with no dependence on any of the predictor. This will reduce the model capacity (cause underfitting).

Example 1

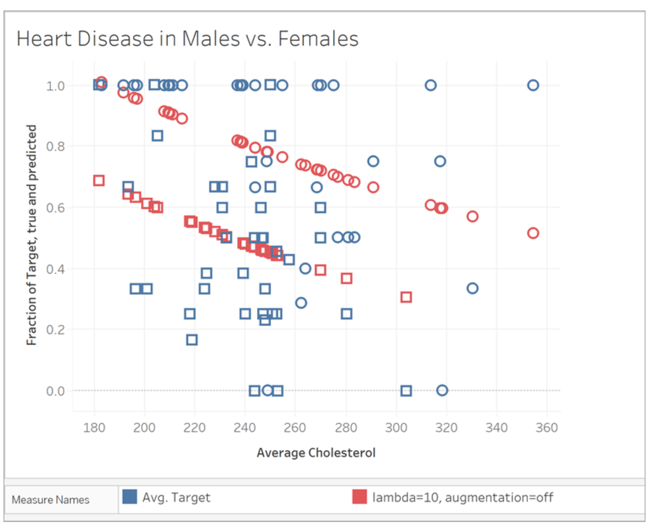
This example shows the relationship between average cholesterol and heart disease in males and females, where males are represented by square marks and females are represented by circles.

In the first visualization, the blue marks indicate the prediction target and the orange marks are the modelled values. You can see that the data is very noisy, and that with augmentation turned on and a small lambda value of 0.01, we see unrealistic heart disease rates greater than 1. There is far too steep of a dependence, probably due to all the outliers in the noisy data.



MODEL_QUANTILE("model=rl, lambda=0.01", 0.5, AVG([Target]), ATTR([Sex]), "augmentation=on", AVG ([Chol]))

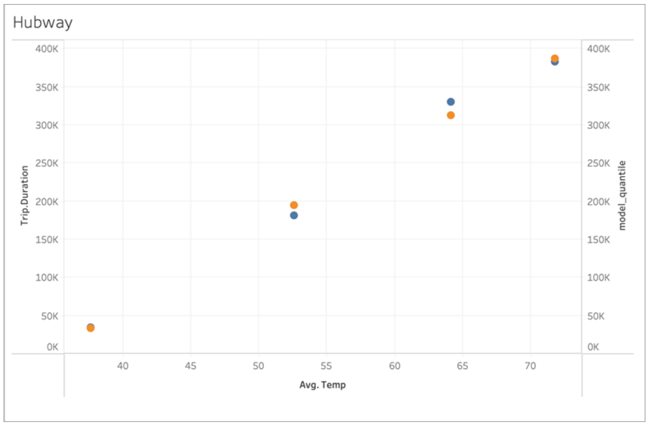
In the next visualization, we compare the prediction target with a different model, with augmentation turned off and a lambda value off 10. Notice that this model is more realistic, and no marks exceed a disease rate of 1.



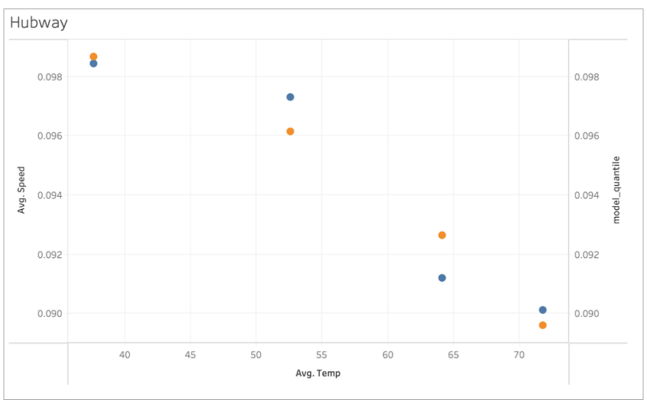
MODEL_QUANTILE("model=rl, lambda=10", 0.5, AVG([Target]), ATTR([Sex]), "augmentation=off", AVG ([Chol]))

Example 2

Next, let’s look at another real-world example using ridership data for Boston’s Bluebikes (formerly Hubway) bike sharing system. In this case, linear regression works well. Compare the following visualizations, aggregated to quarters of 2017:



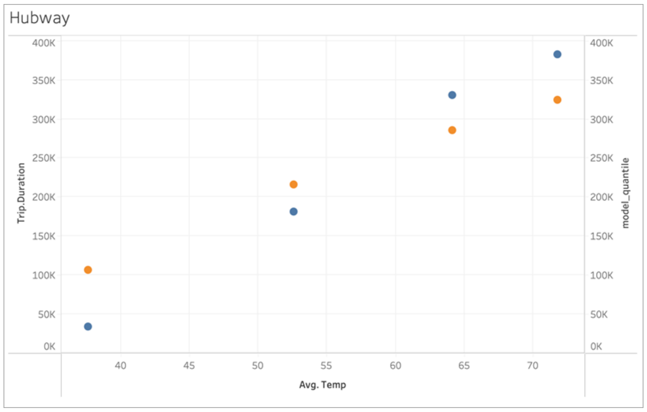
MODEL_QUANTILE('model=rl, lambda=0.05', 0.5, sum([Trip.Duration]), AVG([Temp]))



MODEL_QUANTILE('model=rl, lambda=0.05', 0.5, AVG([Speed]), AVG([Temp]))

Neither is prone to overfitting much, so the dependence on lambda is weak for a small lambda.

Now look at this last visualization:



MODEL_QUANTILE('model=rl, lambda=2', 0.5, sum([Trip.Duration]), AVG([Temp]))

Notice that as lambda increases, the fit flattens to no slope (that is, becomes over-regularized or “underfit”).

Resolve Errors in Predictive Modeling Functions

When using predictive modeling functions in Tableau, you may encounter errors when trying to add the calculations to a viz. Often, this is due to the combination of predictors and visualized elements, leading to a mathematically invalid computation.

Generally, the first thing to try is to double check your Compute Using dimension. Next, verify there isn't a mismatch between the level of detail specified in your predictors and the level of detail in the view. Also verify that your predictors provide relevant data to the model, given the current visualization.

In this article, we describe several common issues in detail, and the ways in which you can address them.

Errors with predictive modeling functions are usually caused by one or more of the following issues, depending on the model used.

General errors

- **Invalid Compute Using dimension** on the next page
- **Dimension predictors conflict with level of detail visualized** on page 2667
- **Multiple predictors conflict with each other** on page 2667
- **There's not enough data to build a model** on page 2668
- **A successful model couldn't be constructed from the data provided** on page 2668

Gaussian process regression errors

- **Gaussian process regression requires exactly one ordered dimension predictor** on page 2669
- **Gaussian process regression only supports dimensions as predictors** on page 2670
- **There's no training data for one or more prediction targets** on page 2671
- **There are more than 5,000 marks** on page 2671

Invalid Compute Using dimension

When visualizing a predictive modeling function, a Compute Using dimension must be specified. It's recommended you select a specific dimension (such as **Region** or **Category**) rather than a directional dimension (such as **Table Across Then Down** or **Pane Across**).

Since you may have multiple predictive modeling functions included in a viz or dashboard, selecting a specific partitioning dimension will help ensure you're building models using the same underlying data set for each individual function—in other words, comparing results from like to like models.

For more information on addressing and partitioning, see **Transform Values with Table Calculations** on page 2406.

When working with predictive modeling functions, you must ensure you maintain consistency across the different instantiations, both in different iterations of your model (e.g., as you select different predictors), and in different vizzes. Using the directional Compute Using dimensions opens up the possibility that a trivial change in your visualized data will significantly affect the data being used to build the model, thus affecting its validity and consistency across different vizzes.

You may see null values if no Compute Using dimension has been specified, or if an invalid Compute Using dimension has been selected.

Invalid Compute Using dimensions include:

- Cell: This dimension will always result in a single data point being used to build a model and will not be successful.
- A higher-level nested dimension (e.g., if **Region** and **State** are both visualized on an axis, where marks are generated for **State** but grouped by **Region**.) If **Region** is selected as the partitioning dimension, it will return null values, since there are no marks formalized for **Region** in this view.

*In the above viz, valid Compute Using dimensions are **State** and **Category**. Partitioning the data by **State** or by **Category** yields multiple valid marks that can be used to build a model.*

***Region**, while listed as a Compute Using option, doesn't generate any marks on this view's level of detail, and can't be used to partition the data.*

For more information, see **Compute Using and Data Partitioning in Predictive Modeling** on page 2628.

Dimension predictors conflict with level of detail visualized

Dimension predictors must be at the same level of detail being visualized or *higher*. That is, if you're visualizing data by **State**, you can use **State**, **Region**, or **Country** as predictors, but using **City** as a predictor will cause an error. Since multiple cities exist within a state, this predictor will resolve to * for all marks and won't return meaningful information; as such, Tableau ignores it.

Each dimensional predictor must be enclosed within an ATTR function, for example:

```
MODEL_PERCENTILE(SUM([Sales]), ATTR([State]), ATTR([Category])
```

Multiple predictors conflict with each other

Each predictor used within a given predictive modeling function must provide independent information to the calculation. If two predictors are selected that are exactly equivalent, scaled versions, or inverses of each other, the predictive modeling function will return an error when it's visualized.

For example, a data set that tracks weather conditions over time might have two fields: **IsRain** and **IsClear**. These two fields are inverses of each other; that is, **IsClear** = True when **IsRain** =

False, and vice versa. Including both **IsClear** and **IsRain** as predictors on a single predictive modeling function will return an error.

The above is true for both measures and dimensions as predictors.

In both cases, the error can be resolved by modifying your predictors to remove the highly correlated predictors. For more information, see **Choosing Predictors** on page 2617.

There's not enough data to build a model

There must be at least three marks within a partition to allow predictive modeling functions to build a model and generate predictions. If there are two or fewer marks in the data set defined by the viewed data and the Compute Using dimension, then the predictive modeling function will return null results.

To resolve this issue, revise your level of detail, your data filtering parameters, or your Compute Using dimension so that each partition has more than two marks.

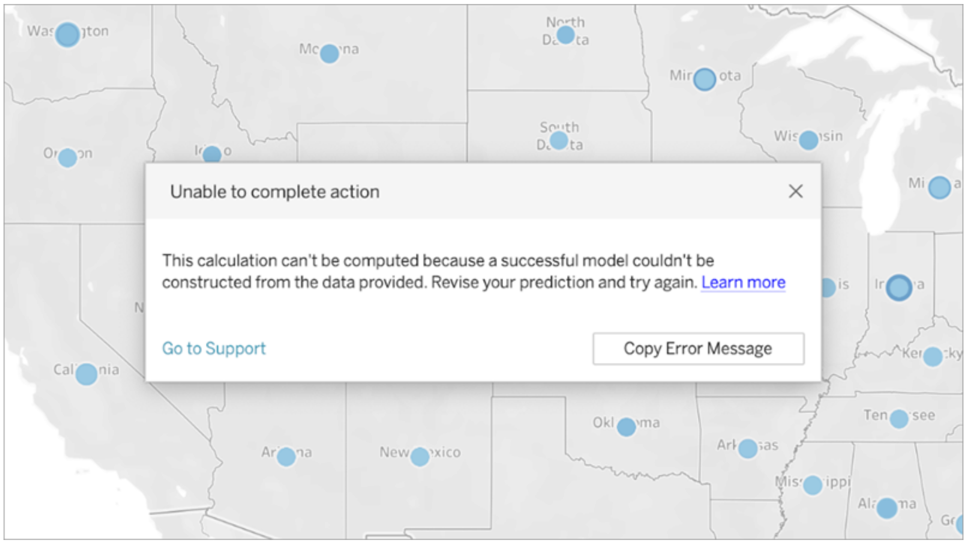
A successful model couldn't be constructed from the data provided

This error appears if there's a conflict between the predictor(s) selected, the visualized level of detail, and/or the selected Compute Using dimension, such that it is impossible to use the predictors to build a rational model.

For example, let's look at this MODEL_QUANTILE calculation:

```
MODEL_QUANTILE(0.5, MEDIAN([Profit]), ATTR(MONTH([Order Date])))
```

Applying this calculation to a viz where each mark represents a **State** value will return the below error:



Since **Order Date** is not used to visualize or segment the data, the predictive modeling function can’t use the predictor specified to build a model.

To resolve this:

- Update your visualization to include the predictor as a dimension.
- Update your function to remove the non-viable predictor.

It’s worth noting that this error reflects a mismatch between the visualization and the table calculation's specific predictors. This same calculation would work without issue if applied to a visualization that included **Order Date** as a dimension.

Gaussian process regression requires exactly one ordered dimension predictor

To use Gaussian process regression, you must include exactly one ordered dimension as a predictor. You can include additional *unordered* dimensions as predictors. Measures can't be used as predictors with Gaussian process regression.

If you encounter this error, add an ordered dimension predictor to your table calculation, remove the excess ordered dimension predictor, or specify that one of the ordered dimensions should be treated as an unordered dimension.

As Gaussian process regression is best used on time-series data, this will usually take the form of a time-based predictor such as ATTR(DATETRUNC('month',[Order Date])). Any date_part can be used in the DATETRUNC expression, but the date_part must be at the same or higher level of detail as the viz. That is, if the viz uses WEEK([Order Date]) as the level of aggregation, the date_part must be 'week','month','quarter', etc.; it can't be 'dayofyear', 'day', 'weekday', 'hour', etc.

Note that since ATTR(MONTH([Date])) returns string values, it will not function as an ordered dimensional predictor unless it is manually specified. This is done by including **"ordered"** immediately before the predictor, as below:

```
MODEL_PERCENTILE(
"model=gp",
SUM([Sales]),
"ordered",ATTR(MONTH([Order Date]))
)
```

If you want to include multiple time dimensions in your predictors, you must specify which ones you wish to use as unordered dimensions. This can be done by including **"unordered"** directly before that dimension, as below:

```
MODEL_PERCENTILE(
"model=gp",
SUM([Sales]),
ATTR(DATETRUNC('month',[Order Date]))
"unordered",ATTR(DATETRUNC('year',[Order Date]))
)
```

Attempting to use the above calculation without specifying **"unordered"** in front of ATTR (DATETRUNC('year',[Order Date]) will return an error.

Additionally, the addressing direction (or "Compute Using") for Gaussian process regression must be set to the same field as the ordered dimension predictor.

Only one addressing direction (or "Compute Using") can be selected for all models.

Gaussian process regression only supports dimensions as predictors

You can't use any measures as predictors in Gaussian process calculations. Remove the measure predictor.

This limitation only applies to predictive modeling functions that specify Gaussian process regression. Linear regression (or OLS, the default) and regularized linear regression functions both support measures as predictors.

There's no training data for one or more prediction targets

Gaussian process regression requires that each data partition has at least *n* data points in order to build a predictive model and return predictions. If you encounter this error, revise your predictions.

There are more than 5,000 marks

Gaussian process regressions are only supported on vizzes with fewer than 5,000 marks. To resolve this issue, update your level of detail to decrease the number of marks, or select a different statistical model.

Integrate Einstein Discovery Predictions in Tableau

Reveal dynamic predictions from Einstein Discovery in your Tableau views and dashboards. In Tableau 2021.1 and later, you can access the power of Einstein Discovery predictions in these ways:

- Connect to the Einstein Discovery analytics extension to interact with Einstein Discovery-powered predictive models from Tableau.
- Embed predictions in a Tableau workbook by pasting generated table calculation scripts from Einstein Discovery into calculated fields in Tableau.
- Give users dynamic, on-demand predictions based on their Tableau data using the Einstein Discovery dashboard extension.
- In Tableau Prep, add Einstein Discovery prediction steps to flows to enrich your flow output with predictions and, optionally, improvements and top factors.

Note: For an introduction to Einstein Discovery in Tableau, see the Einstein Discovery in Tableau page and demo on tableau.com.

What is Einstein Discovery?

Einstein Discovery is an analytics tool that augments your data analysis with the power of artificial intelligence using machine learning models and comprehensive statistical analysis. Einstein Discovery quickly sifts through millions of rows of data to find important correlations, predict outcomes, and suggest ways to improve those predicted outcomes.

For more information about Einstein Discovery, see Getting Started with Discovery, and Explain, Predict, and Take Action with Einstein Discovery in Salesforce help. You can also expand your knowledge with the Gain Insight with Einstein Discovery trail in Trailhead.

> **Note**: Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.

Requirements for access

To configure and use Einstein Discovery predictions in Tableau, you and anyone who will be viewing predictions in a Tableau workbook will need certain licenses, access, and permissions in Salesforce and Tableau.

Salesforce Requirements

Tableau integration with Einstein Discovery is supported in Tableau version 21.1 or later. Integration requires:

requirement	description
Salesforce license	One of the following licenses: • Einstein Discovery in Tableau license • Tableau CRM Plus license • Einstein Predictions license These licenses are available for an extra cost.
Salesforce user account	Account that is configured to access Einstein Discovery. If you use the Einstein Discovery in Tableau license, your user account must have the **View Einstein Discovery Recommendations Via Connect API** system permission assigned to it. If you use either the Tableau CRM Plus license or Einstein Predictions license: • To get predictions using already deployed Einstein Discovery predictions, the account must have the **View Einstein Discovery Recommendations** system permission assigned to it.

requirement	description
	• To build, deploy, and manage predictions in Einstein Discovery, the account must have the **Manage Einstein Discovery permission** assigned to it. To configure user accounts, see Set Up Einstein Discovery in Salesforce help.
Administrator settings	Salesforce administrators will need to: • **Analytics Extension**: Configure Salesforce to create a connected app for Tableau Server (basic). Required for Tableau Server only. • **Dashboard Extension**: Add the Tableau Server or Tableau Online website domain to the Salesforce Cross-Origin Resource Sharing (CORS) allowlist.

Tableau Requirements

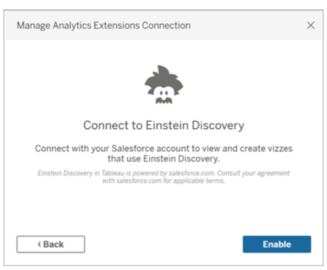
requirement	description
Tableau license and permissions	Authors who need to configure the Einstein Discovery dashboard extension will need a Creator or Explorer license and permissions that allow them to edit and save workbooks. Creators, Explorers, and Viewers will need to be signed in to the Salesforce org account to access predictions in the Einstein Discovery dashboard extension in Tableau.
Tableau user account	In Tableau Online and Tableau Server, users can save Salesforce user account credentials along with their Tableau user account.
Administrator settings	Tableau Server administrators will need to configure Tableau Server for OAuth (enables access to predictions from Tableau). For more information, see Configure Einstein Discovery Integration in Tableau Server help. To let users see predictions in published table calculations,

requirement	description
	Tableau Server administrators need to enable analytics extensions for the site. For more information, see Configure Connection with Analytics Extensions in Tableau Server help. For more information on Tableau Server configuration, see Configure Einstein Discovery Integration (Tableau Server). For more information on Tableau Online configuration, see Configure Einstein Discovery Integration (Tableau Online).

Analytics extension support for Einstein Discovery predictions

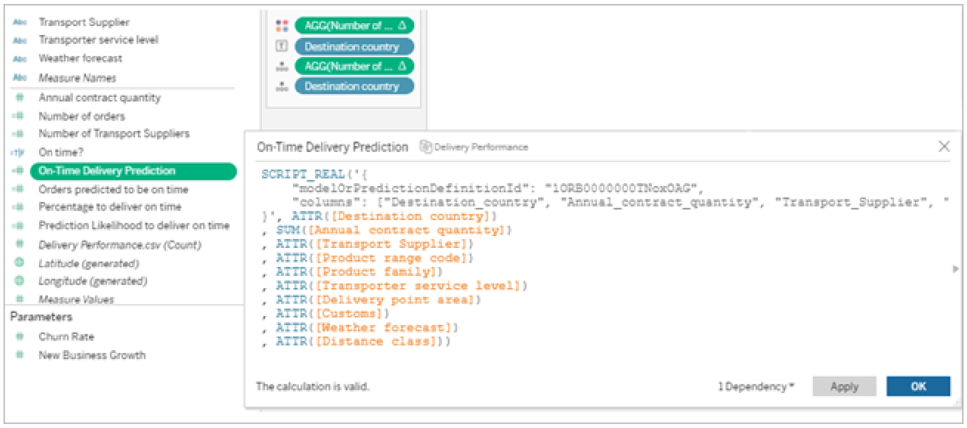
Einstein Discovery in Tableau brings trusted, intuitive predictions and improvements from Einstein Discovery to every Tableau user across your organization. Workbook authors can build and share visualizations and dashboards that dynamically integrate predictions from predictive models deployed in Salesforce Einstein Discovery.

Authors set up the Einstein Discovery analytics extension for use in a workbook, and then can embed Einstein Discovery-powered predictions directly in Tableau using calculated fields.



Access Einstein Discovery predictions in Tableau calculated fields

With Tableau connected to the Einstein Discovery analytics extension, you can embed predictions directly in calculated fields. In Salesforce, use Model Manager to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook. The table calculation script accesses predictive models deployed in Salesforce by calling the prediction ID and passing the data required for the model.

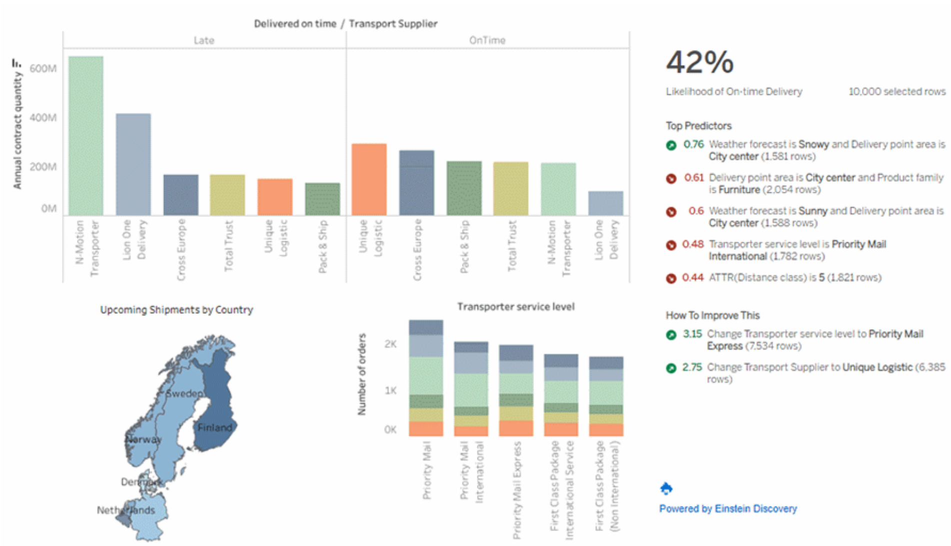


You can create prediction-enhanced visualizations in Tableau Desktop, Tableau Online, or Tableau Server, or publish them to the web.

For more details, see **Einstein Discovery** on page 2693 in **Pass Expressions with Analytics Extensions** on page 2690.

Einstein Discovery dashboard extension in Tableau

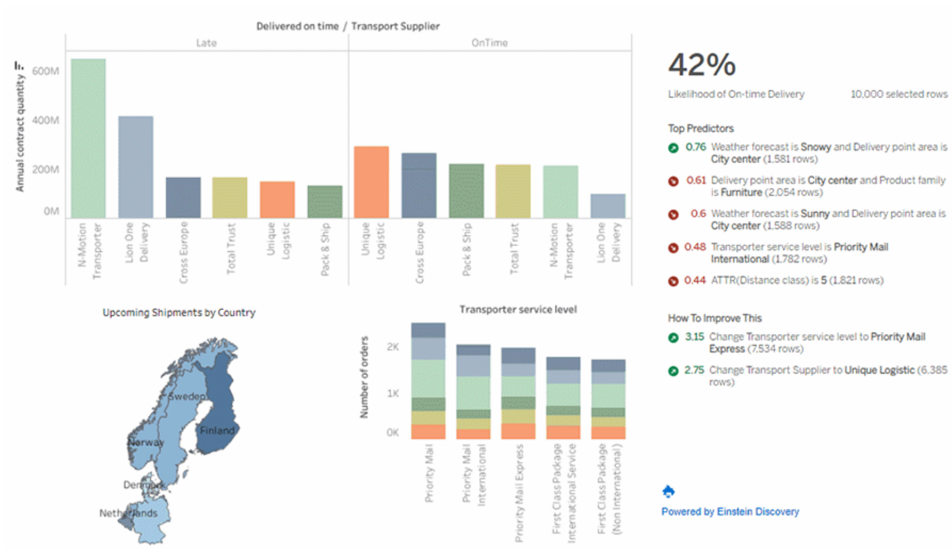
Integrate on-demand, interpretable predictions from Einstein Discovery in Tableau dashboards. Authors create worksheets and the dashboard, and then configure the Einstein Discovery dashboard extension to use a selected Einstein Discovery prediction definition, which uses source data available in the dashboard to derive predictions. Dashboard viewers and users can click different marks in the Tableau visualization to explore predictions.



For more information, see **Explore Predictions in Tableau with the Einstein Discovery Dashboard Extension** below.

Explore Predictions in Tableau with the Einstein Discovery Dashboard Extension

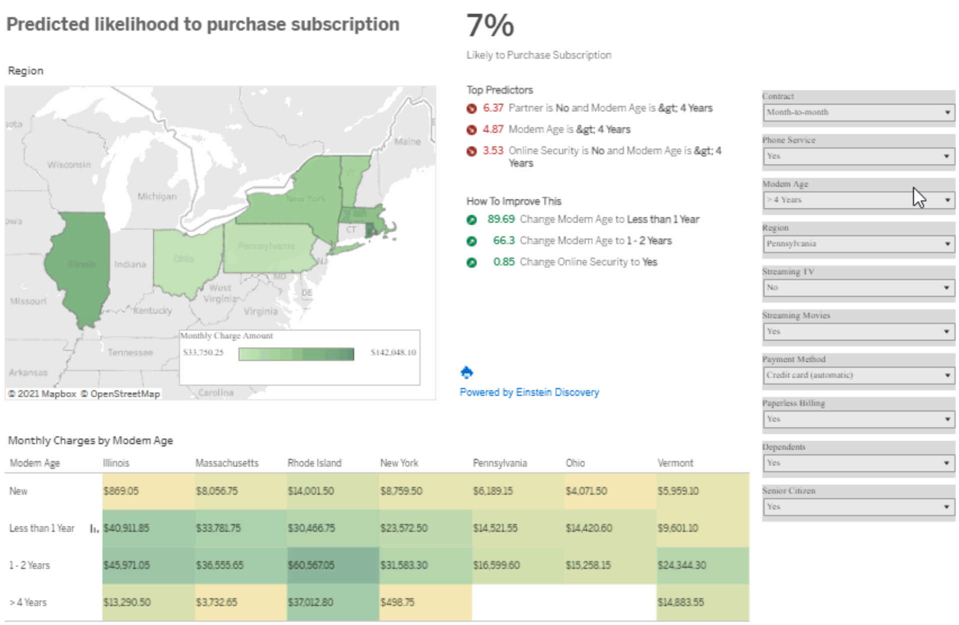
Use the Einstein Discovery dashboard extension to surface real-time predictions in Tableau. The dashboard extension generates predictions interactively, on-demand, using an Einstein Discovery predictive model on source data in your Tableau workbook.



Click marks in the dashboard to see dynamic predictions, key drivers, and possible ways to improve predictions based on an Einstein Discovery predictive model

Dashboard authors can configure the dashboard extension to run predictions on aggregated data in a worksheet, as demonstrated in the image above, or use parameters to let dashboard users explore "what-if" scenarios based on a single set of input values.

For information on how to add the Einstein Discovery dashboard extension, see **Add and configure the Einstein Discovery dashboard extension** on page 2679 in this topic. Also see **Requirements for access** on page 2672.



Select inputs in parameters to see how those values might affect predictions

For more information about Einstein Discovery, see Getting Started with Discovery, and Explain, Predict, and Take Action with Einstein Discovery in Salesforce help. You can also expand your knowledge with the Gain Insight with Einstein Discovery trail in Trailhead.

Note: Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.

Use the Einstein Discovery dashboard extension

To explore predictions in a dashboard that uses the Einstein Discovery dashboard extension, select different marks or parameter values in the view. Predictions update based on the data in your selection.

You might need to sign into the Salesforce org account that has access to the prediction definition used by the dashboard extension.

In Tableau Desktop, you will need to sign in to your Salesforce account every time you use the Einstein Discovery dashboard extension. If Salesforce signs you out of your session automatically, you may need to reload the dashboard extension (click the drop-down arrow on the Extension object and then select **Reload**).

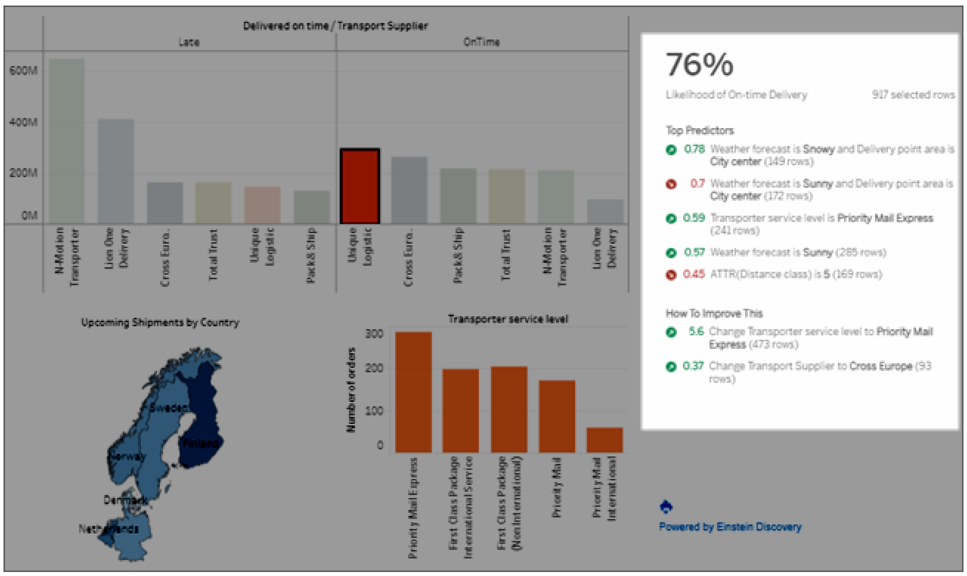
In Tableau Online and Tableau Server, you should need to sign in only once if your credentials are being saved with your Tableau account settings.

For related information, see Use Dashboard Extensions in Tableau help and Explain, Predict, and Take Action with Einstein Discovery in Salesforce help.

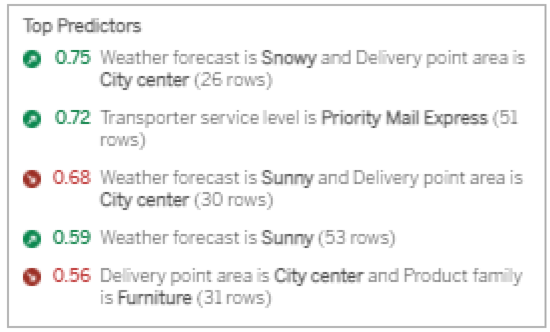
How to read Einstein Discovery predictions

The dashboard extension has three sections:

- **Prediction**—The predicted outcome
- **Top Predictors**: Conditions that affect the predicted outcome
- **How To Improve This**: Suggested actions to take to improve the predicted outcome



Section	Description	Example
Prediction The prediction reflects the goal of the use case.	The goal is to either maximize or minimize the outcome. For example, your goal can be to minimize shipping time or to maximize the average sales revenue per shipment.	

Section	Description	Example
Top Predictors Conditions that most significantly affect the predicted outcome, in decreasing order of magnitude.	A **condition** is a data value associated with a field. A **predictor** consists of one or two conditions. If two conditions are shown, they are joined by *and* to represent the intersection of those conditions. Green (arrow up) indicates that the predictor improves the predicted outcome. Red (arrow down) indicates that the predictor worsens the predicted outcome.	
How To Improve This Suggested actions the user can take to improve the predicted outcome.	Improvements are associated with factors over which users can possibly control, such as the shipping method or a subscriber's membership level.	

Add and configure the Einstein Discovery dashboard extension

To configure and use the Einstein Discovery dashboard extension in Tableau, you will need:

- Access to a Salesforce org and Tableau Desktop, Tableau Server, or Tableau Online.
- An Einstein Discovery prediction definition that is deployed in Salesforce.
- Source data in Tableau with fields that match the model fields required by the Einstein Discovery prediction definition.
- A worksheet in Tableau that contains the source data for the prediction.
- Additional worksheets that can be used as filters in the dashboard.
- The Einstein Discovery dashboard extension in a dashboard.

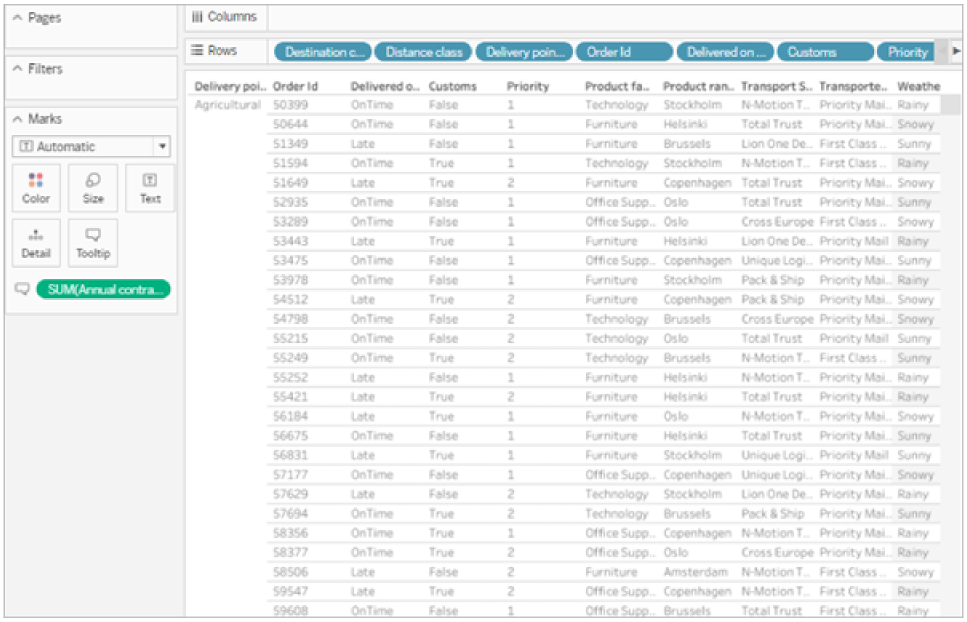
To configure and use Einstein Discovery predictions in Tableau, you and anyone who will be viewing predictions in a Tableau workbook will need certain licenses, access, and permissions in Salesforce and Tableau. For more information, see **Requirements for access** on page 2672.

Prepare the model and the workbook

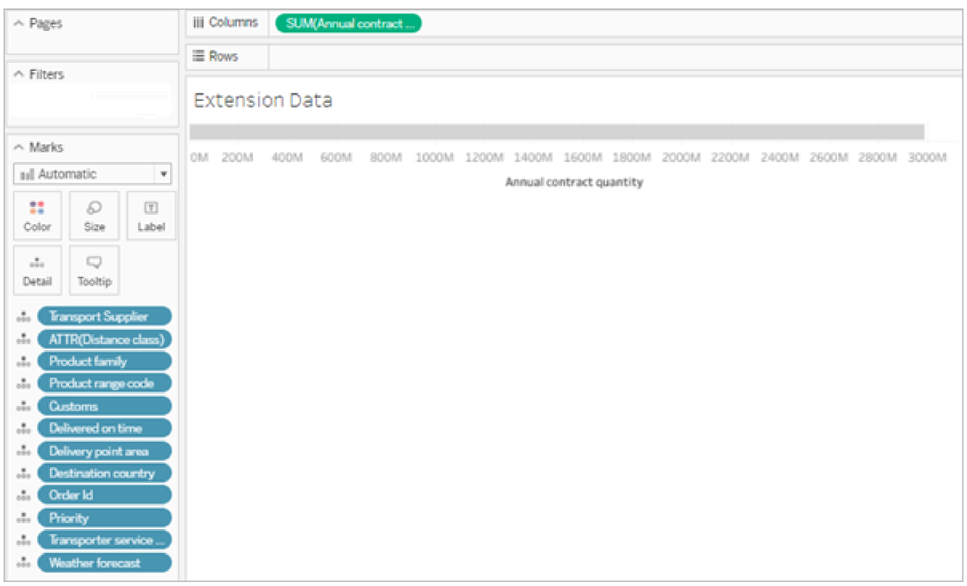
1. Build and deploy the Einstein Discovery prediction definition that you want to use. You can also use a prediction definition that someone else has built and deployed. For more information, see Build, Deploy, and Manage Models in Salesforce help.
2. Create a Tableau workbook that uses a data source with fields that can be mapped to all model fields required by the Einstein Discovery prediction definition.
3. Create a worksheet that contains the source data for the dashboard extension.

   This source data must include all of the fields required by the Einstein Discovery prediction definition to predict an outcome. Also, the source data must match the granularity that the Einstein Discovery prediction definition expects. For example, if the prediction definition expects sales per individual order, then your Tableau data must include fields at the level of detail of individual orders.

   - **Worksheets** support predictions for multiple rows of data (bulk predictions). The worksheet can contain the necessary fields in the Rows or Columns shelves, or on Marks card properties. The worksheet will not be visible in the dashboard; a simple view will suffice. For example, you could create a text table.

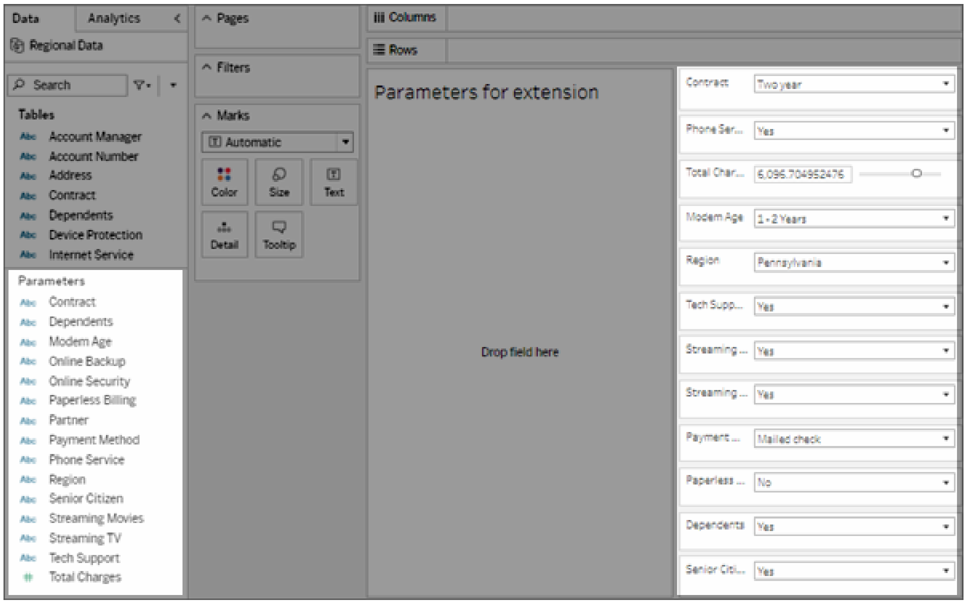


Or a basic bar chart.



The worksheet used as source data for the model cannot exceed 50,000 rows of data. Fields in the view can't be Measure Names or Measure Values.

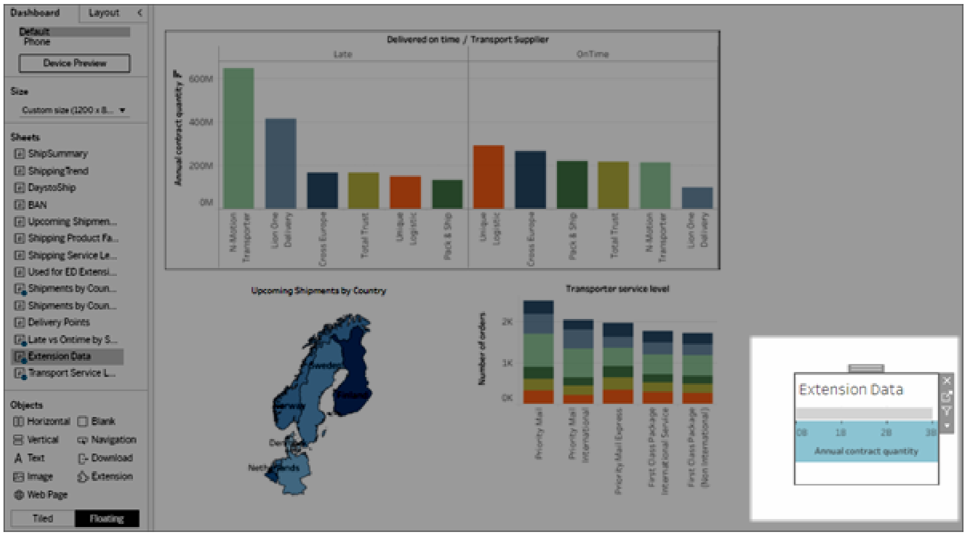
- **Parameters** support single-row predictions. To set up the worksheet, show the parameters in the view. You don't need to add fields to the view for this worksheet. For example:



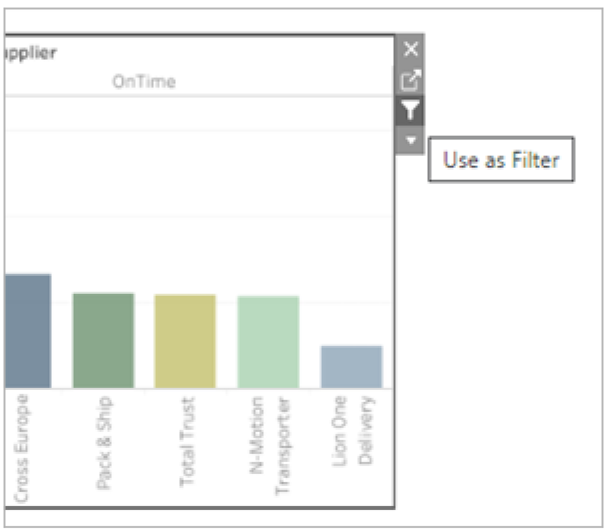
4. Create one or more worksheets with visualizations that you can set to **Use as Filter** in the dashboard.

   The visualization can use a subset of the fields used by the model. Clicking marks in these views in the dashboard will refresh the predictions in the dashboard extension.

5. Create the dashboard. In the **Objects** section, click **Floating**. From the **Sheets** section, drag the source data worksheet to the canvas. Resize and hide that sheet so users won't see it in the dashboard. Also drag one or more sheets to the canvas to serve as filters in the dashboard.



In the Sheet object menu, set at least one worksheet with a visualization to **Use as Filter**.



Configure the extension

1. From the **Objects** section, select **Floating**, and then drag the **Extension** object to the dashboard canvas. In the Extensions Gallery, select **Einstein Discovery**.

   Sign in to your Salesforce account. After you sign in, a web page opens asking if you want to allow your Salesforce account to access Tableau. Click **Allow** to continue, and then close the resulting tab in your browser.

   On Tableau Desktop, you will need to sign in to your Salesforce account every time you use the Einstein Discovery dashboard extension. If Salesforce signs you out of your session automatically, you may need to reload the dashboard extension (click the drop-down arrow on the Extension object and then select **Reload**). This time-out setting is configurable. For more information, see Edit Session Settings in Profiles in Salesforce help.

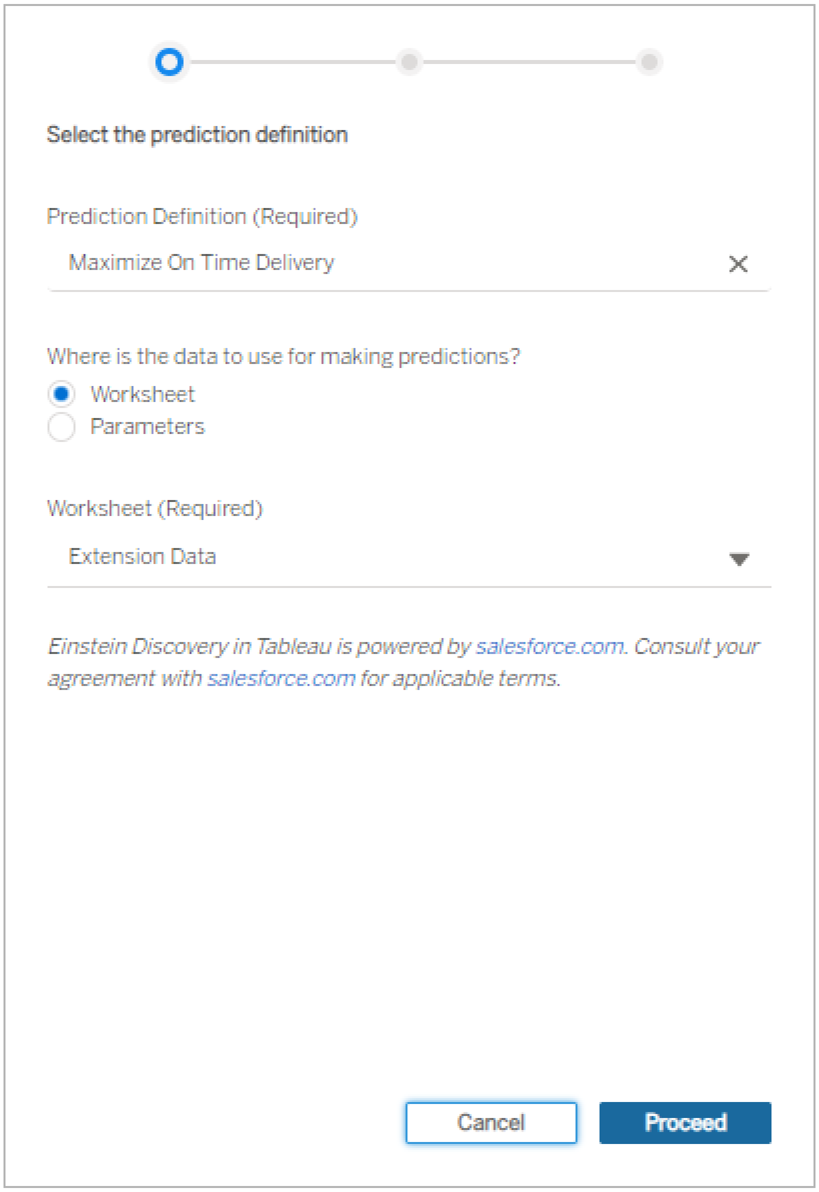
   In Tableau Online and Tableau Server, you should need to sign in only once if your credentials are being saved with your Tableau account settings.

2. Configure the dashboard extension.

   For **Prediction Definition**, click *Search Predictions* and then select the name of a prediction model deployed in Salesforce.

   Select **Worksheet** or **Parameters** to be the source data for the predictions. Worksheet provides predictions based on a selected mark with aggregated values. Parameters supports interactive, “what if” predictive analysis on a single set of input values.

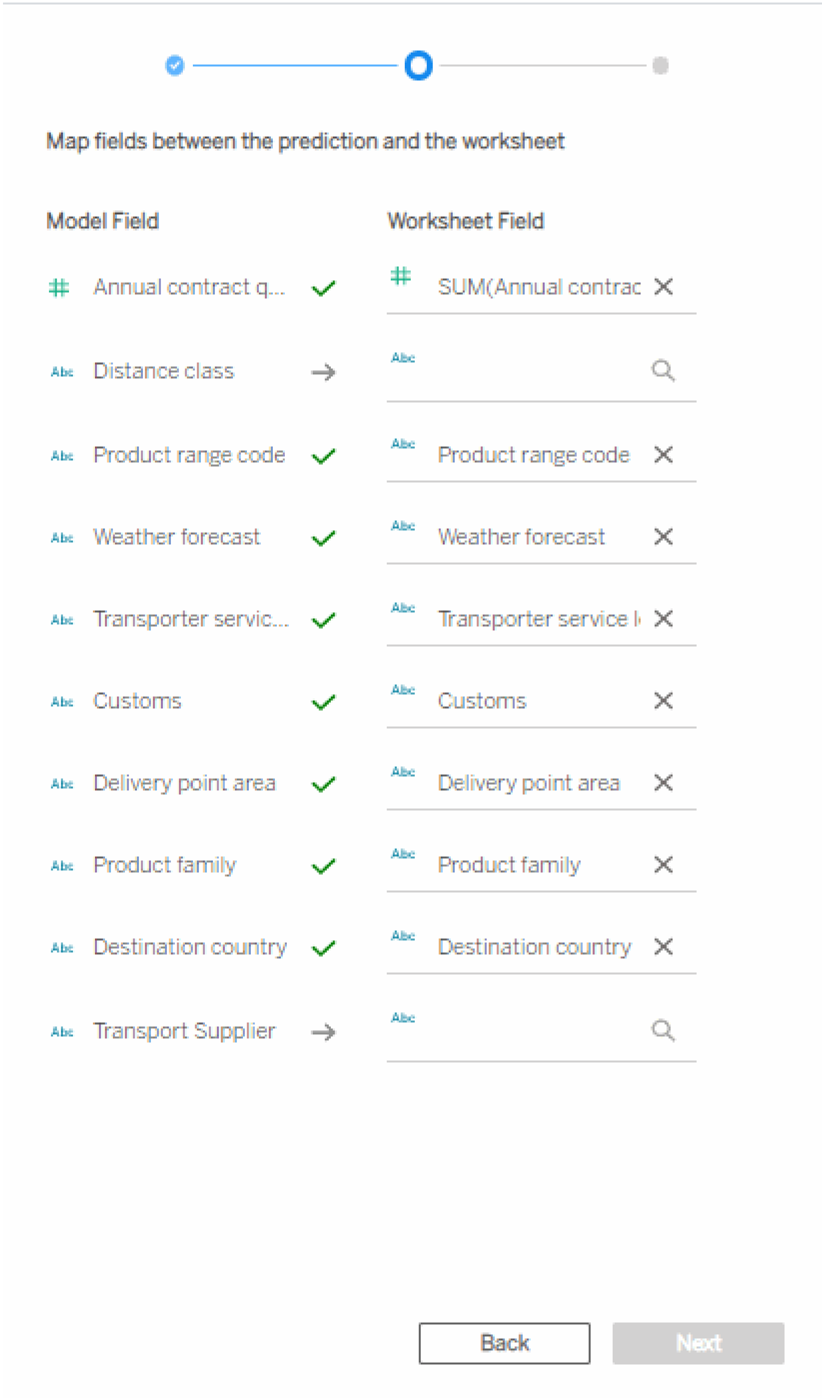
   For **Worksheet**, click **Select an Option**, and then select the name of the worksheet with the source data. For **Parameters**, no other setting is needed.



Click **Proceed**.

3. Map the fields from the model to the fields in the worksheet. The extension will automatically map fields based on names when possible.

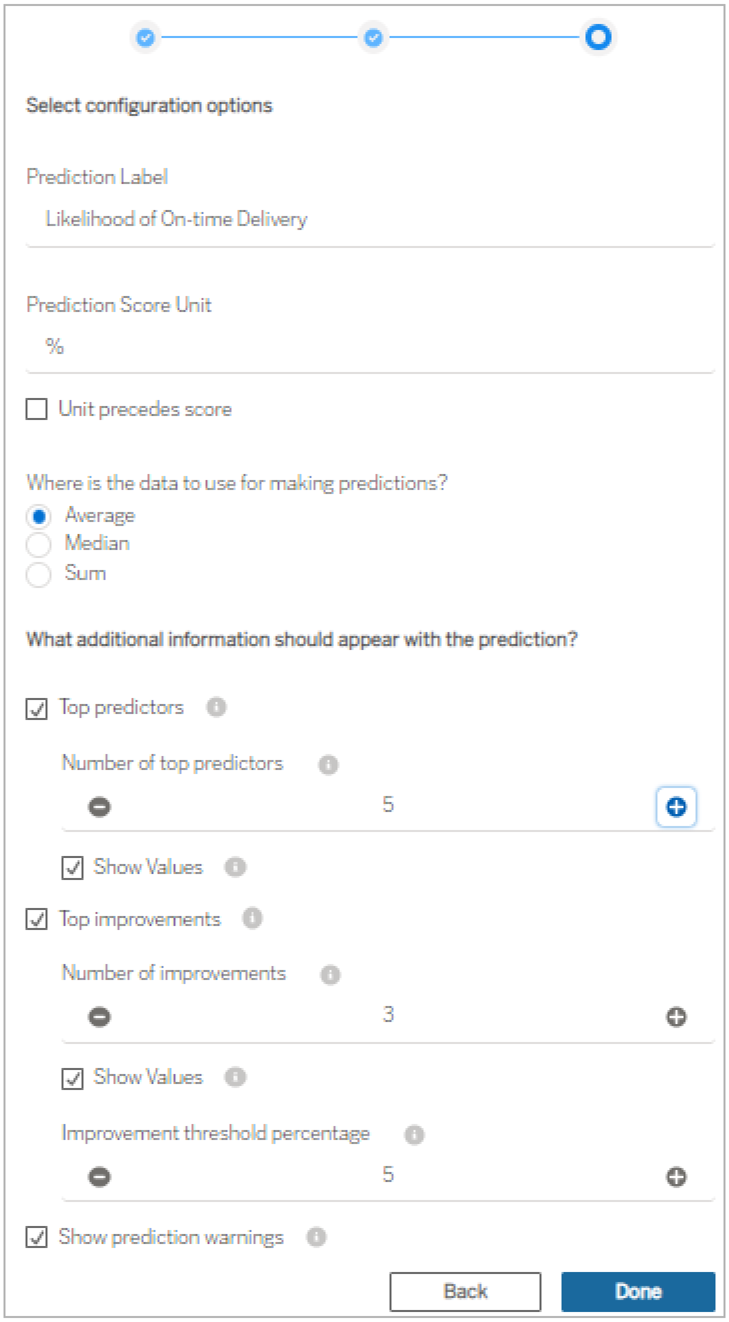
   To add or change mappings, click in the search box next to a prediction field and select a name from the list of available worksheet fields or parameters.



You must map all of the fields to continue to the next set of configuration settings. If you don't see a matching field listed, try these troubleshooting steps.

Click **Next** to continue.

4. Select options for how predictions are displayed.



Prediction Label: Click the field and type a label that represents the outcome that you are predicting, for example, *Likelihood of On Time Delivery*. The deployed model name is displayed by default.

Prediction Score Unit: Type a symbol or text that indicates the unit for the prediction score. For example, %. If the unit should be displayed before the score, select **Unit precedes score**. For example, a currency symbol (such as $) would precede a currency amount.

Aggregation: Specify how you want to aggregate the selected records (**Average**, **Median**, or **Sum**) in the source data worksheet. Tableau aggregates your selected data into a single set of values, which it submits to the Einstein Discovery prediction definition as input. The granularity must match the granularity expected for the model.

Top predictors: Specify whether to display top predictors that impact the prediction. You can also specify the number of top predictors displayed, and whether the impact values are displayed next to each predictor.

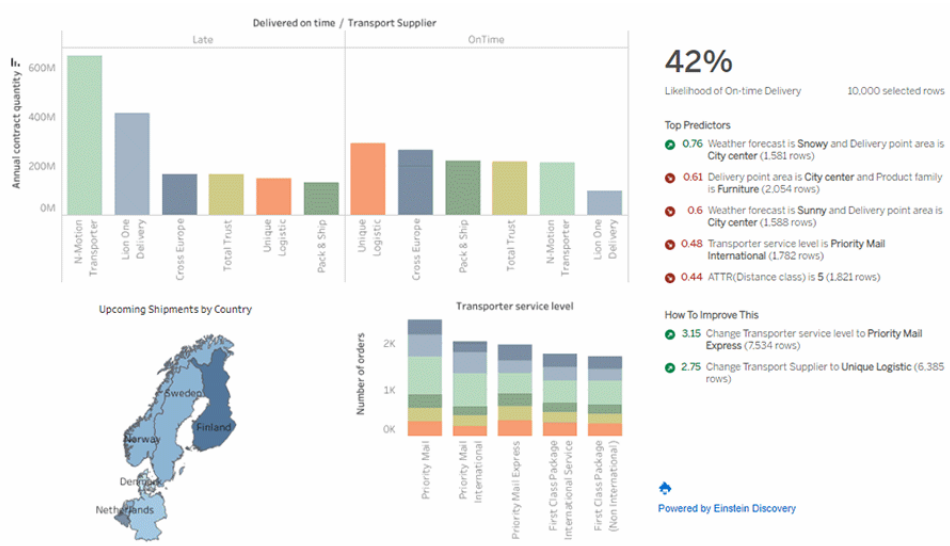
Top improvements: Specify whether to display information on ways to improve the probability of the predicted outcome. You can also specify the number of improvements displayed, and whether the impact values are displayed next to each improvement. For **Improvement threshold percentage**, specify a percentage number (0-99) to display improvements only if they improve the prediction by that percentage.

Show prediction warnings: Specify whether to show warnings about the predictive model, such as missing values in required fields or out-of-bounds values.

Consider showing everything to start with and then fine tuning what you end up sharing with your dashboard audience.

5. Click **Done**. Tableau submits the data immediately to the target Einstein Discovery prediction definition, and then displays the result.
6. In the dashboard, make sure at least one worksheet is set to **Use as Filter**. Every worksheet in the dashboard that is set to **Use as Filter** will drive the extension to update the predictions based on the current selection.

Test your dashboard by clicking and selecting different marks and areas in the dashboard (or by selecting different parameter values) to see what outcome Einstein Discovery predicts for that subset of the data.



Save the workbook in Tableau Online or Tableau Server. Or, publish the dashboard from Tableau Desktop to your Tableau Online or Tableau Server site to share it with more people. Explorers and Viewers with access to the same Salesforce org and an Einstein Discovery in Tableau license, a Tableau CRM Plus license, or an Einstein Predictions license will be able to use the extension.

Troubleshoot field mapping

If a field you are expecting to use is not in the list of available fields, you might need to check the data type for the field in Tableau.

Einstein Discovery and Tableau do not always recognize dimensions, measures, and data types in the same way, so you might have to change the data type and role (dimension or measure) in the Data pane of the worksheet to match how Einstein Discovery interprets the field.

1. Close the **Extension: Einstein Discovery** window.
2. Go to the source data worksheet. In the Data pane, change the data type and role of a problematic field in Tableau to match the Einstein Discovery field data type and role in the mapping settings. After you change the data type or role, you will need to replace incorrect fields in the view with the updated fields from the Data pane.

3. In the dashboard, click the drop-down menu for the extension, and then select **Configure**. Proceed to the field mapping settings and select the fields from the list.
4. If you still don't see the fields you expect in the list, go back to the source data worksheet. Remove the fields that are missing for mapping from the Marks card, or the Rows, or Columns shelves.

   Next, drag the missing fields from the Data pane to Tooltip on the Marks card.

5. Repeat Step 3 to select the fields for mapping.

Pass Expressions with Analytics Extensions

Tableau supports a set of functions that you can use to pass expressions to analytics extensions for integration with Einstein Discovery, RServe, MATLAB, and Python. Previously, this feature was referred to as "external services". In version 2020.1 (and later), this functionality is referred to as *analytics extensions*.

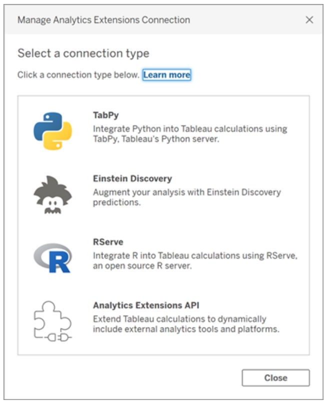
Beginning with version 2021.2, you can create multiple analytics extensions connections for any site. Versions prior to 2021.2 are limited to a single analytics extension for any site. **Note:** Einstein Discovery analytics extensions are still limited to one for each site. For more information about configuring analytics extensions, see Configure Connections with Analytics Extensions (Tableau Online Help) or config_r_tabpy.htm (Tableau Server Help).

Once you have created a connection to an analytic extension, you'll communicate with your external service through calculated fields. Each calculated field uses the **SCRIPT functions for expressions** on page 2697, operating as a table calculation.

Configure an analytics extension

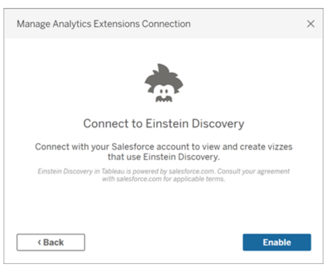
Tableau allows connections to analytics extensions via a set of SCRIPT functions. Before you can pass any SCRIPT functions to your analytics extension, you need to configure your connection to the service.

1. In Tableau Desktop, click the **Help** menu, and then select **Settings and Performance** > **Manage Analytics Extension Connection**.

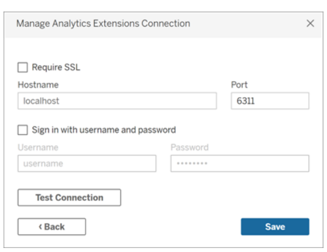


Select the analytics extension you want to connect to. **MATLAB** connections are managed by the Analytics Extension API.

2. Select and configure your analytics service.
   - If you are using **Einstein Discovery**, select Enable. Your default browser will open, allowing you to authorize the connection to Salesforce. After connecting, go to the **Einstein Discovery** on page 2693 section of this help topic to continue.



   - For all other services (Tabpy, Matlab, and RServe), the following dialog will appear.



3. Enter or select a server name using a domain or an IP address.

If your server is using SSL encryption, select the **Require SSL** option. Tableau will read certificates installed in the operating system key store to establish the secured connection.

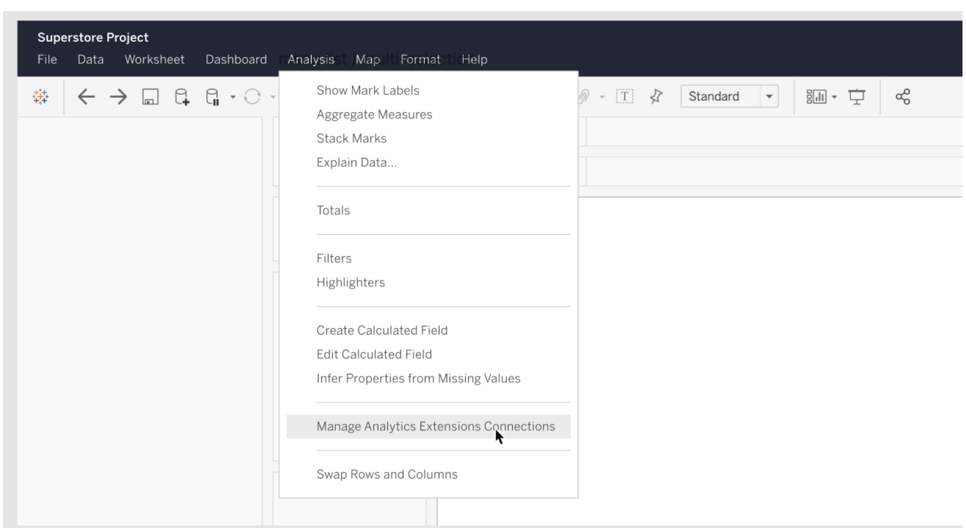
> **Note**: Tableau only supports **PEM** encoded certificates when connecting to **TabPy** or **RServe**.

4. Specify a port.
   - Port 6311 is the default port for plaintext RServe servers.
   - Port 4912 is the default port for SSL-encrypted RServe servers.
   - Port 9004 is the default port for TabPy.
5. Click **Test Connection**.
6. Click **Save**.

If no connection can be established, an error message is displayed. Click **Show Details** in the message to see any diagnostic information returned by the server.

Using Analytics Extensions in Web Authoring

When you are using **web authoring** to create or edit a workbook, select **Analysis** > **Manage Analytics Extension Connections** and select the analytics extension you want to use for the workbook. Only one analytics extension can be used per workbook.



See the links below for information on setting up Analytics Extensions on Tableau Online or Tableau Server.

- For Tableau Server, see "Configure Connection with Analytics Extension" (Windows | Linux).
- For Tableau Online, see Configure Connection with Analytics Extension.

Einstein Discovery

Einstein Discovery is a powerful analytics tool in Salesforce that makes use of machine learning to provide insights and predictions. Einstein Discovery requires the Einstein Analytics Plus license, Tableau CRM Plus license, or the Einstein Predictions license, which are available for an extra cost. For related information, see **Requirements for access** on page 2672.

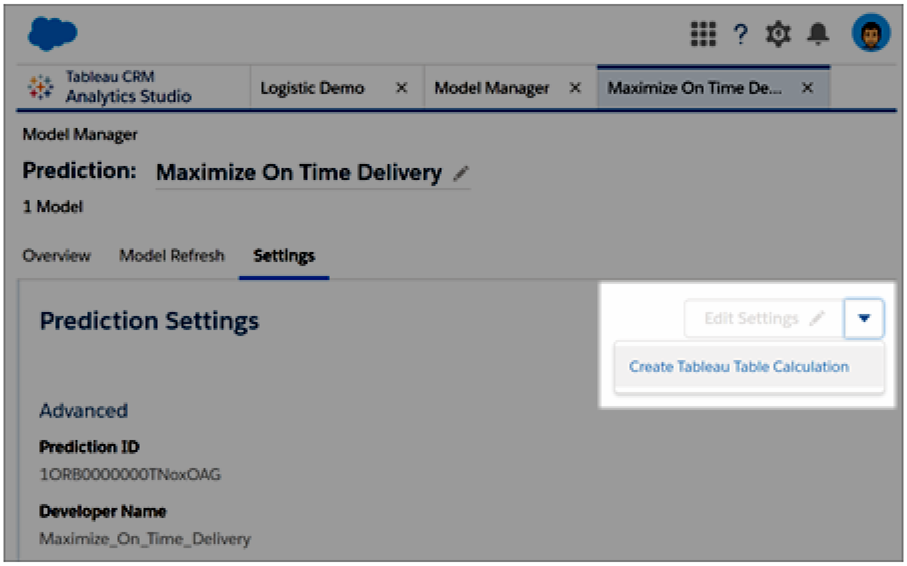
For more information about Einstein Discovery, see Getting Started with Discovery, and Explain, Predict, and Take Action with Einstein Discovery in Salesforce help. You can also expand your knowledge with the Gain Insight with Einstein Discovery trail in Trailhead.

Note: Einstein Discovery in Tableau is powered by salesforce.com. Consult your agreement with salesforce.com for applicable terms.

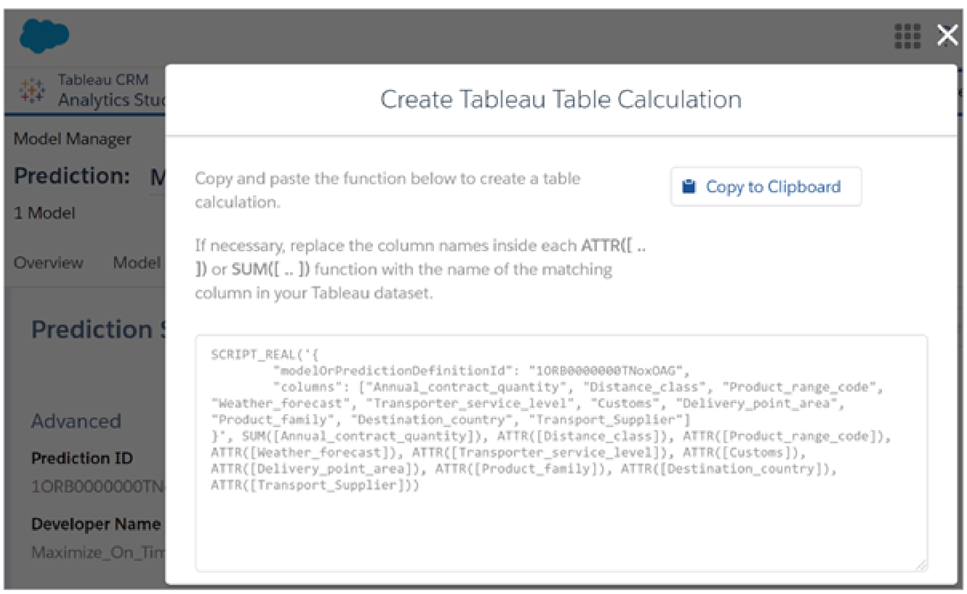
Integrate Einstein Discovery models into Tableau

With Tableau connected to the Einstein Discovery analytics extension, you can embed predictions directly in calculated fields. In Salesforce, use Model Manager to auto-generate a Tableau table calculation script, and then paste that script into a calculated field for use in a Tableau workbook. The table calculation script accesses predictive models deployed in Salesforce by calling the prediction ID and passing the data required for the model.

1. Open your Einstein Discovery Salesforce account and navigate to the Model Manager. Select a model to integrate into Tableau. On the model tab, select **Edit Settings** > **Create Tableau Table Calculation.**



Model Manager generates a script containing the ID of the prediction definition you want to use, the fields (variables) used in the model, and the corresponding source fields in Tableau.

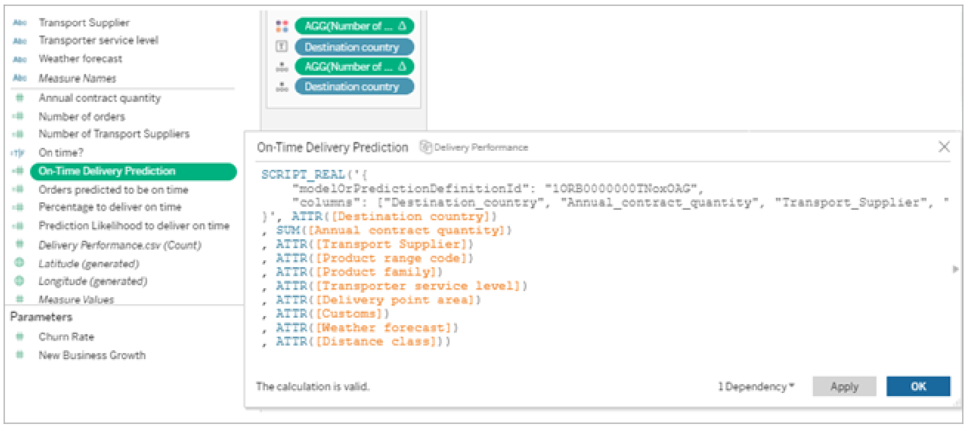


2. Click **Copy to Clipboard.**

3. In a Tableau workbook, select **Analysis** > **Create Calculated Field**, and then paste the generated script in the Calculation Editor.

If the field names are different from the Salesforce dataset the model was trained on, these fields are displayed with a red underline. You will need to edit these field names to match the correct field names in Tableau.

Replace the fields names inside each ATTR or SUM function with the name of the matching field in your Tableau data, then save the calculated field.



The calculation can be used just like any other Tableau calculation. For more information, see **SCRIPT functions for expressions** on page 2697 below.

When the script runs, it sends a prediction request (with the input data) to the prediction definition in Salesforce. Einstein Discovery returns the results to Tableau, where it appears in your worksheet or dashboard.

RServe

R is an open source software programming language and a software environment for statistical computing and graphics.

> **Watch a Video**: To see related concepts demonstrated in Tableau, watch the free training videos How to Integrate R and Tableau (4:40 minutes) and Using R with Tableau (4:44 minutes). Use your tableau.com account to sign in.

Implementing RServe connections

For R, you need access to a server which allows applications to access R functionality. See RServe for details. For information on installing, running, and configuring RServe, as well as on optimizing R scripts and R security, see the Tableau Community post R Implementation Notes. For more information about using R with Tableau, see the blog post Tableau 8.1 and R.

Tableau Desktop can connect to R via either plaintext or SSL-encrypted traffic. For direction on which type to use, consult your system administrator.

> **Note**: For R integration, Tableau has been tested with R versions 3.4.4 through 3.5.1, and with RServe versions 0.6-8 through 1.7.3.

Intermediate certificate chain for RServe analytics extension

As of Tableau Server version 2020.1, you must install a full certificate chain on Tableau Desktop computers (Windows and Mac) that are connecting to a Rserve external connection through Tableau Server. This requirement is due to how Rserve manages the handshake on secure connections.

Importing a root certificate on the Tableau Desktop is not sufficient, the entire certificate chain must be imported onto the client computer.

MATLAB

MATLAB is a high-level language that connects through the **Analytics Extension API** that enables you to perform computationally intensive tasks faster than with traditional programming languages such as C, C++, and Fortran. You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. For information on how to configure your MATLAB server to work with Tableau please contact MATLAB support.

To read more about the possibilities of MATLAB in Tableau, see Put your MATLAB models and algorithms to work in Tableau.

TabPy

Python is a widely used high-level programming language for general-purpose programming. By sending Python commands to an analytics extension, you can do things like predicting customer churn or running sentiment analysis.

Tableau Python Server (TabPy) is part of Tableau's expanding range of extensibility options. To install TabPy, visit this GitHub page. To read more about the possibilities of Python in Tableau, see Building advanced-analytics applications with TabPy.

SCRIPT functions for expressions

In Tableau Desktop, a set of four SCRIPT functions are available for passing expressions to analytics extensions and obtaining a result. The functions are:

`SCRIPT_BOOL`

`SCRIPT_INT`

`SCRIPT_REAL`

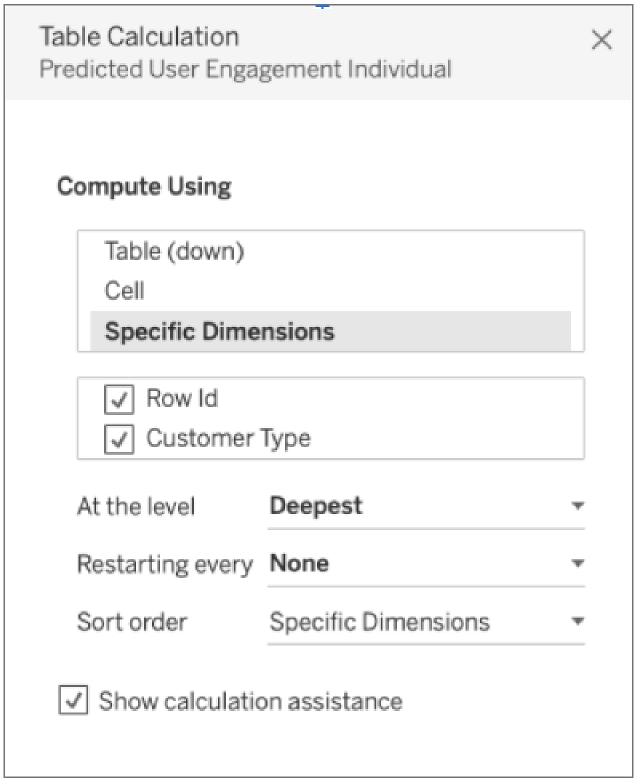
`SCRIPT_STR`

See **SCRIPT_BOOL** on page 2228 for details and examples.

Because these SCRIPT functions are table calculations, addressing and partitioning concepts apply. (For an explanation of these concepts, see **The basics: addressing and partitioning** on page 2408.) Tableau makes one call to an the analytics extension per partition. This means that for each dimension the calculation is partitioning on, one call will be made for every member of that dimension present in the visualization. For this reason, it is recommended to put dimensions on addressing unless partitioning is necessary.

Because connecting to an analytics extension involves some network overhead, it is often most efficient to pass data all at once rather than as individual row values whenever possible. For example if you set addressing to **Cell** (for example, by clicking the field in the view and choosing **Compute Along** > **Cell**), Tableau will make a separate call per row (created by each mark in the visualization) to the analytics extension. Depending on the size of the data, this can result in a very large number of individual calls.

The easiest way to ensure efficient calls are made to the analytics extension is to navigate to **Edit Table Calculation** and select **Specific Dimensions** then check the box next to each dimension to put that dimension on addressing. This will generate a single analytics extension call for the entire visualization. Any dimension left un-checked will be partitioned on, generating a separate call for each member in the visualization. If there are many marks in the visualization, it may be advantageous to select **Pause Auto Updates** while selecting dimensions to prevent frequent runs of the calculation before it is properly computed.



A calculation can be set to address specific dimensions by default by editing the calculated field and selecting the **Default Table Calculation** option in the bottom right-hand corner. In the **Compute Using** menu, select **Advanced** and then move the dimensions expected to the right hand space under **Addressing**. Note that this will cause the calculation to error if all addressed dimensions are not present in the visualization, though this can be corrected by editing the table calculation as described above and un-checking the dimensions not present.

Share a workbook that requires an analytics extension connection

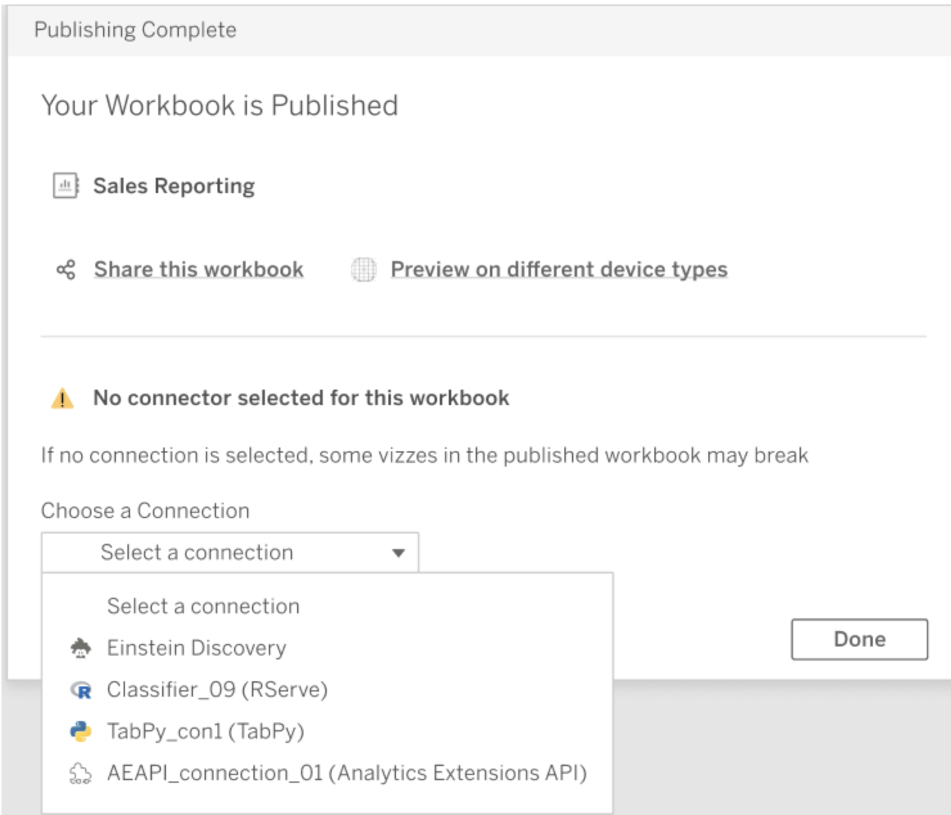
You might need to send a workbook that contains analytics extension functionality to other users, who may be using different copies of Tableau Desktop on other computers. Or, users may download a workbook from Tableau Server that contains analytics extension functionality. If you share your workbook with someone, they must configure analytics extension connections on their computers.

Publish a workbook that requires an analytics extension connection

Before you publish a workbook that relies on an analytics extension connection to Tableau Server or Tableau Online, you must configure Tableau Server or Tableau Online to have its own analytics extension connection.

- For Tableau Server, see "Configure Connection with Analytics Extension" (Windows | Linux).
- For Tableau Online, see Configure Connection with Analytics Extension.

Upon successfully publishing, you must select which of the configured analytics extensions to associate with the workbook.



Troubleshooting Analytics Extensions Connections

This topic describes errors you can receive when Tableau is connected to an external analytics extension service.

Note: Tableau technical support cannot assist with writing, troubleshooting, or debugging external scripts.

Errors that can occur when communicating with any analytics extension

Error Message	Comments
Unsupported type passed as an argument to SCRIPT function.	Tableau can only export the following Tableau data types to R: • Number (Decimal) • Number (Whole) • Boolean • String • Date • Date/Time
Unexpected number of results returned by SCRIPT function. Function expected %2 values; %1 values were returned.	The script result must be either a scalar or vector of length one that is replicated for all rows, or a vector of length equal to the number of rows in the Tableau result table.
Certificate file is not valid for the specified connection (%1 %2:%3). Check that the same certificate is used for both server and client, the certificate file is not corrupted and has valid dates.	Verify that the certificate you are using to connect to the external service is encoded and formatted correctly. You must connect with a valid PEM-encoded x509 certificate with the extension .crt.
The result returned by the SCRIPT function is of an unexpected type.	Occurs when an invalid data type is received. Tableau can only import the following data types from an Rserve server: • RDouble

	• RInteger • RLogical • RCharacter For TabPy, Tableau can import the following data types: • List of floating point numbers • List of integers • List of Booleans • List of strings This error is also reported if the result is null or if there was a script execution error for which Tableau could not collect an explanation from the external server.
An error occurred while communicating with the external service.	Tableau runs all external scripts inside of the "try" external function. This error is displayed along with an external-service-generated error message when the “try” function traps an evaluation error.
This external service connection does not support authentication. Try connecting without specifying a password.	
Authentication failed. Please provide a valid external service username and password.	
An unsupported	Either disable external service authentication or change to plain-

authentication type is enabled in the external service.	text password authentication.
No external service connection con-figured. Specify a server name and try again.	See **Pass Expressions with Analytics Extensions** on page 2690.
The calculation '%1' contains a SCRIPT_ function that requires an internal service connection. Con-figure your external service connection to enable custom scripts.	See **Pass Expressions with Analytics Extensions** on page 2690.
The workbook you are attempting to publish contains SCRIPT_ functions that require an external service con-nection. Custom scripts are not allowed in public workbooks.	You cannot publish workbooks containing external service scripts to Tableau Public.
Tableau Public does not support running custom scripts that require external ser-	The "Tableau Public" in this error refers to Tableau Desktop Public.

vices. To take advantage of external services, upgrade to Tableau Desktop Professional Edition.	
Tableau Reader does not support running custom scripts that require external services. To take advantage of external services, upgrade to Tableau Desktop Professional Edition.	You cannot view workbooks containing external service scripts in Tableau Reader.

Errors that can occur only with an Rserve Server

Error Message	Comments
Unrecognized Rserve signature.	The Rserve header signature must be "Rsrv".
Unrecognized Rserve version.	The Rserve header version must be "0103".
Unrecognized Rserve protocol.	The Rserve header protocol must be "QAP1".
Authentication failure when connecting to R.	Tableau attempted and failed to authenticate with Rserve. Verify that you entered a valid password.
Incorrect number of	

bytes in para-meter/body.	
The length of data which Tableau read does not equal the length promised by the header.	
Unrecognized response type.	The transport protocol type when reading a result was not SEXP as expected.
Excessively long <type>vector.	Unreasonably large number measuring the length of data sent to or from Rserve possibly indicating a corrupt protocol header.
Invalid Rserve com-mand.	Tableau may have improperly implemented the Rserve protocol.
Response from server was Error " << (uint32_t)status << ". See Rsrv.h for details.	Various error conditions are documented in the comments in Rsr-v.h.
Excessively long header offset.	This may be due to a garbled header with an unreasonable offset to response data.
Rserve socket failed.	A login, script evaluation, read pending check, result read, or Tableau field to R script argument assignment threw a non-stand-ard exception.
The external service uses an unsupported authentication type.	To support running both SSL-enabled and plaintext connections to an Rserve instance, the Rserve Server administrator must add the following entry to the Rserve configuration file on the Rserve Server: `plaintext enable`

A crypto-system error has occurred on RServe.	An error has occurred with RServe's encryption system. Check the RServe logs for more details.
RServe has initiated closing due to a security violation.	RServe closed the connection due to a security violation. Check the RServe logs for more details.
The External Service Configuration is set for a secure connection but missing a certificate.	When establishing a secure connection to an External Service, you are required to upload the certificate with which you expect this External Service to be signed. Connections to servers that are not signed by this certificate will be refused. This ensures that you are only sending your data to trusted servers.
Certificate file is not valid for the specified connection (%1 %2:%3). Check that the same certificate is used for both server and client, the certificate file is not corrupted and has valid dates	The certificate provided is not valid for this connection configuration. Check to see that it matches the certificate signing the External Service server, that it is PEM encoded, and that it is valid for today's date.
External service connection certificate is invalid	The certificate provided is invalid. Check to see that the certificate is PEM encoded, and that it is valid for today's date.
External service connection failed to create SSL/TLS context	Tableau failed to initialize a secure environment for this given External Service configuration. Check to see that the host and port are correct, and that the certificate is valid.
External service failed to connect with SSL/TLS	Tableau failed to establish a secure connection to the External Service. Verify that the External Service is configured to support secure connection, and that the certificate provided matches the

	one signing the External Service.
External service server protocol is unrecognized	A protocol was prepended to the host that is not recognized or supported by Tableau. If connecting to RServe, no protocol is expected. If connecting to TabPy, the HTTP protocol is expected when establishing a plaintext connection, and the HTTPS protocol is expected when establishing a secure connection.

Other issues

SCRIPT functions run even in logical statements that evaluate as false

A function that sends an expression to a running external service instance will be executed even when it is within a logical statement that would otherwise prevent it from being executed. This is true for logical functions such as IF, IIF, and CASE. For example:

```
IF 1==0 THEN
  [[R script code]]
ELSE
  "1 does not equal 0"
END
```

Explore and Inspect Data in a View

When you are ready to explore and analyze your view, Tableau offers a selection of dynamic data inspection tools that help you isolate the data of interest. For example, if you have a dense data view, you can focus on a particular region, select a group of outliers, and view the underlying data source rows for each mark.

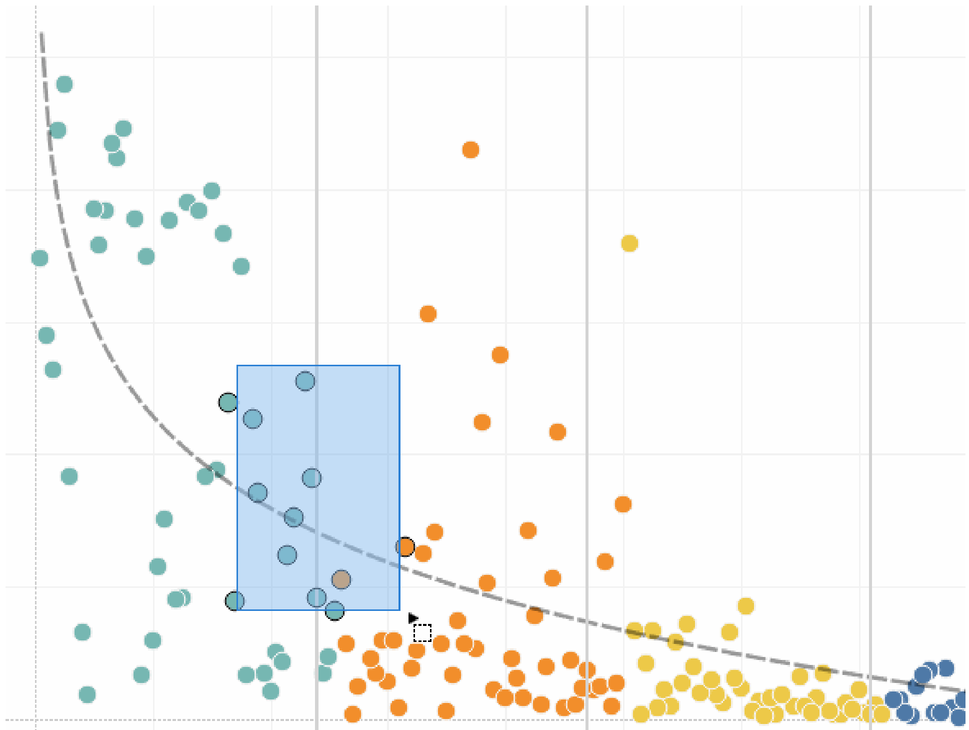
Watch a Video: To see related concepts demonstrated in Tableau, watch Getting Started with Visual Analytics, a 6-minute free training video. Use your tableau.com account to sign in.

Select marks to highlight data points in the view

The simplest way to add highlighting to a view is to select the marks you want to highlight. When you manually select a mark in the view, all other marks are dimmed to draw attention to the selection. Your selection is saved with the workbook.

For more information on how to use selection tools, see **Select marks** on page 2722. Also see **Legend Highlighting** on page 1499, **Highlight Data Points in Context** on page 1502, **Highlight Toolbar Button** on page 1505, and **Highlight Actions** on page 1498.

You can select multiple marks by holding down the Ctrl key on your keyboard (Command key on Mac) while you select each mark. You can also use one of the advanced selection tools to select marks within a specific shape.

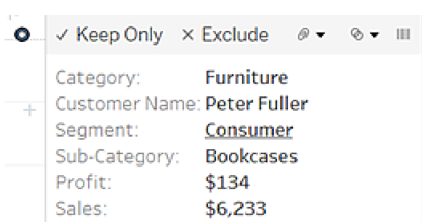


You can also select marks and headers to filter what is being displayed in the view. For more details, see **Select to keep or exclude data points in your view** on page 1412 and **Select headers to filter data** on page 1413.

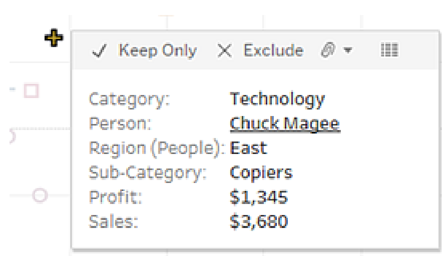
Analyze selected marks using tooltips

When you select one or multiple marks in a view, and then continue to hover in the same spot, several options for inspecting your data are available in the tooltip.

Tableau Desktop version



Web version



Tooltips provide additional information about the marks in your view, and provide the following data analysis options:

- Keep only the selected marks in the view.

  For more information, see **Select to keep or exclude data points in your view** on page 1412.

- Exclude the selected marks from the view.

  For more information, see **Select to keep or exclude data points in your view** on page 1412.

- Create a group based on the selected marks. For more information, see **Correct Data Errors or Combine Dimension Members by Grouping Your Data** on page 1184.

- Create a set that contains the selected marks (Tableau Desktop only). For more information, see **Create Sets** on page 1186.

- View the underlying data of the marks selected. For more information, see **View Underlying Data** on page 2728.

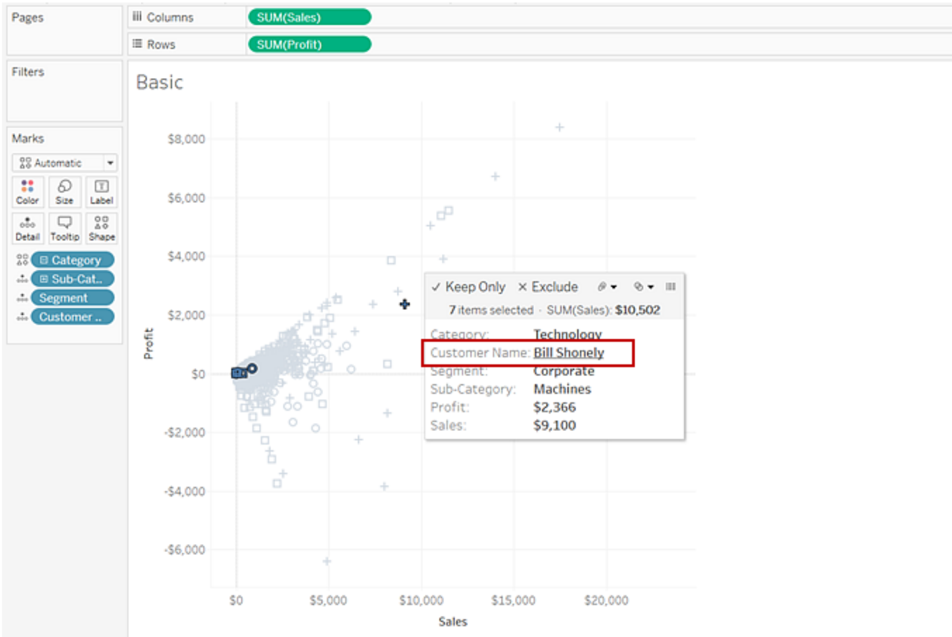
For related information on Tooltips, see **Tooltips** on page 250.

For details on related analysis tools, see **Zoom and Pan Views, and Select Marks** on page 2719, **Undo and Redo** on page 2725, and **Drop Lines** on page 2543.

Highlight data by category in tooltips

If you include discrete (categorical) dimensions or measures in your tooltip, you can use them to select marks in the view that have the same values. If Tooltip selection is turned on for your worksheet, then these fields show as underlined text when you mouse over them. Clicking on the tooltip selects all the marks in the view that have the same value.

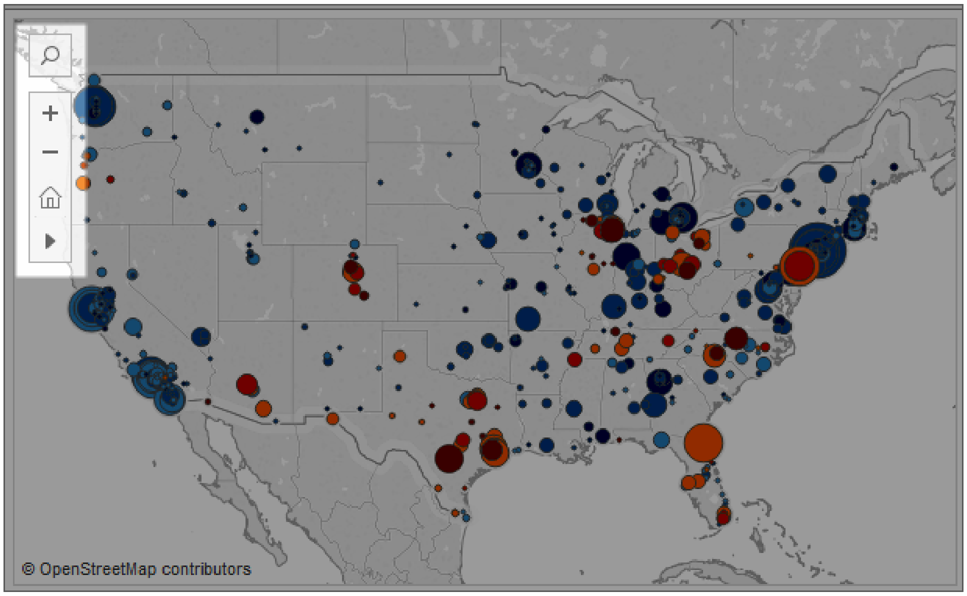
In the example below, when you click on Bill Shonely (the value for Customer Name) in the tooltip, any mark in the view that also includes Bill Shonely as a customer is selected and all other marks are dimmed.



To turn this functionality off or on, select the **Allow selection by category** check box in the Edit Tooltips dialog box. For more information about setting tooltip properties, see **Add tooltips to marks** on page 1366.

Explore data in maps

Tableau maps can help you quickly find locations and analyze data worldwide. There are many ways you can explore and interact with map views. You can zoom in and out, pan, and select marks with the view toolbar, and even search for locations worldwide with map search.



For more details, see **Explore Data in Maps** on page 2003, **Search for Locations in Your Map** on page 2006, and **Measure Distances Between Data Points and Locations in a Map** on page 2007.

Add analytics objects to the view (trend lines, forecasts, reference lines and bands)

Drag reference lines, box plots, trend lines forecasts, and other items into your view from the **Analytics** pane, which appears on the left side of the workspace. Toggle between the **Data** pane and the **Analytics** pane by clicking the tabs at the top of the side bar. For more details, see **Apply Advanced Analysis to a View (Analytics Pane)** on page 213.

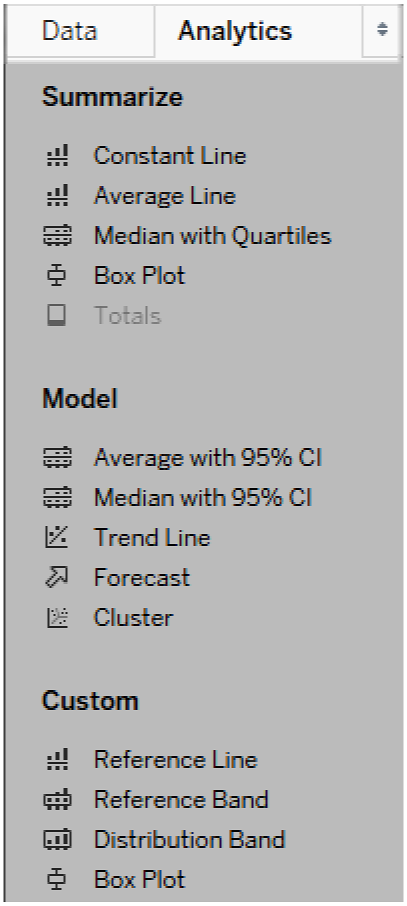


Tableau Desktop Analytics pane

In Tableau Desktop, options for adding Analytics objects to the view are available the **Analytics** pane or menu, or in context in the view. For example, reference lines and bands are available when you edit an axis, and trend lines and forecasts are available from the Analysis menu.

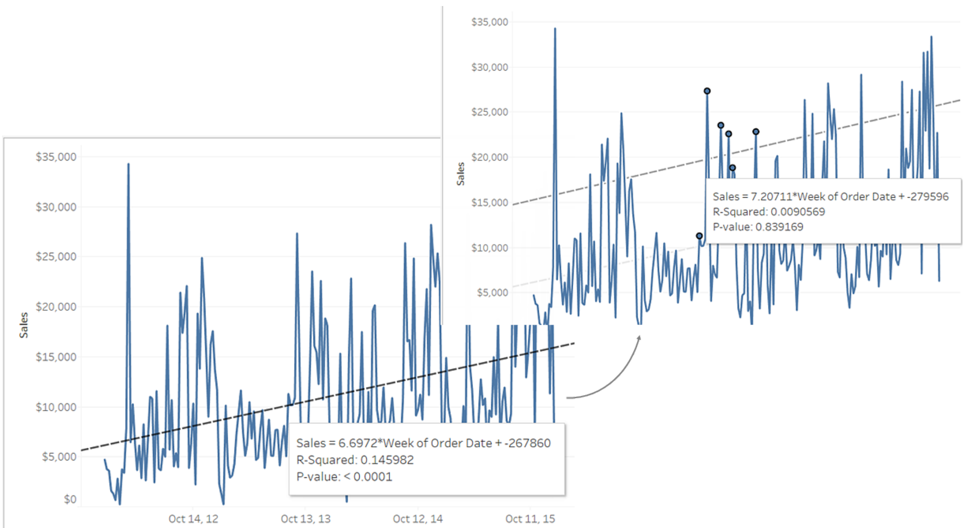
The **Analytics** pane provides drag-and-drop access for the various options.

On the web, most Analytics objects are available from the Analytics pane.

Compare marks data with recalculated lines

If there is an analytics object in your view, such as an average line, constant line, trend line, reference line or distribution line or band, selecting one or multiple marks in the view lets you instantly compare the analytical data for the selected marks to all data in the view.

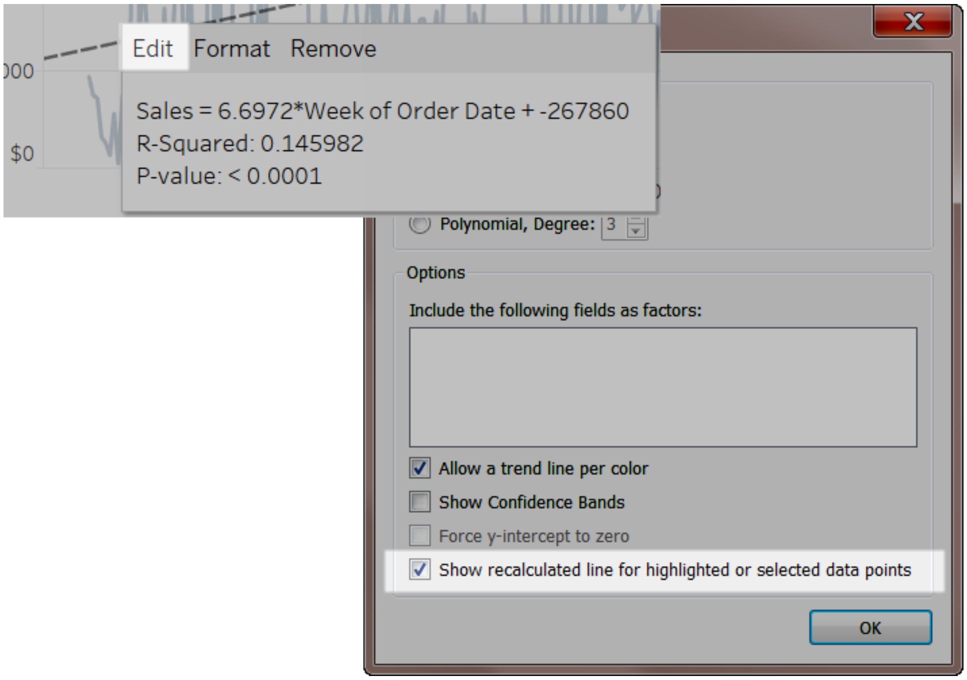
For example, selecting marks in a view that contains a trend line creates a second, recalculated trend line, the value of which is determined by the selected marks only, so you can compare that trend to the overall trend.



Recalculated lines are displayed by default when you select marks in a view that already contains analytics objects. In Tableau Desktop, if you don't want recalculated lines to be created you can turn them off.

Turn off recalculated lines (Tableau Desktop only):

1. Select an analytics object in the view, such as a trend line, and click **Edit**.
2. In the **Edit** dialog box, clear **Show recalculated line for highlighted or selected data points**.



Alternatively, you can right-click (control-click on Mac) an analytics object in the view and clear **Show Recalculated Line**.

When you return to the view and select or highlight marks, recalculated lines will not appear. For more information, see **Highlight Actions** on page 1498.

Recalculated lines and highlight actions

Recalculated lines also work with highlighting actions, both in worksheets and dashboards.

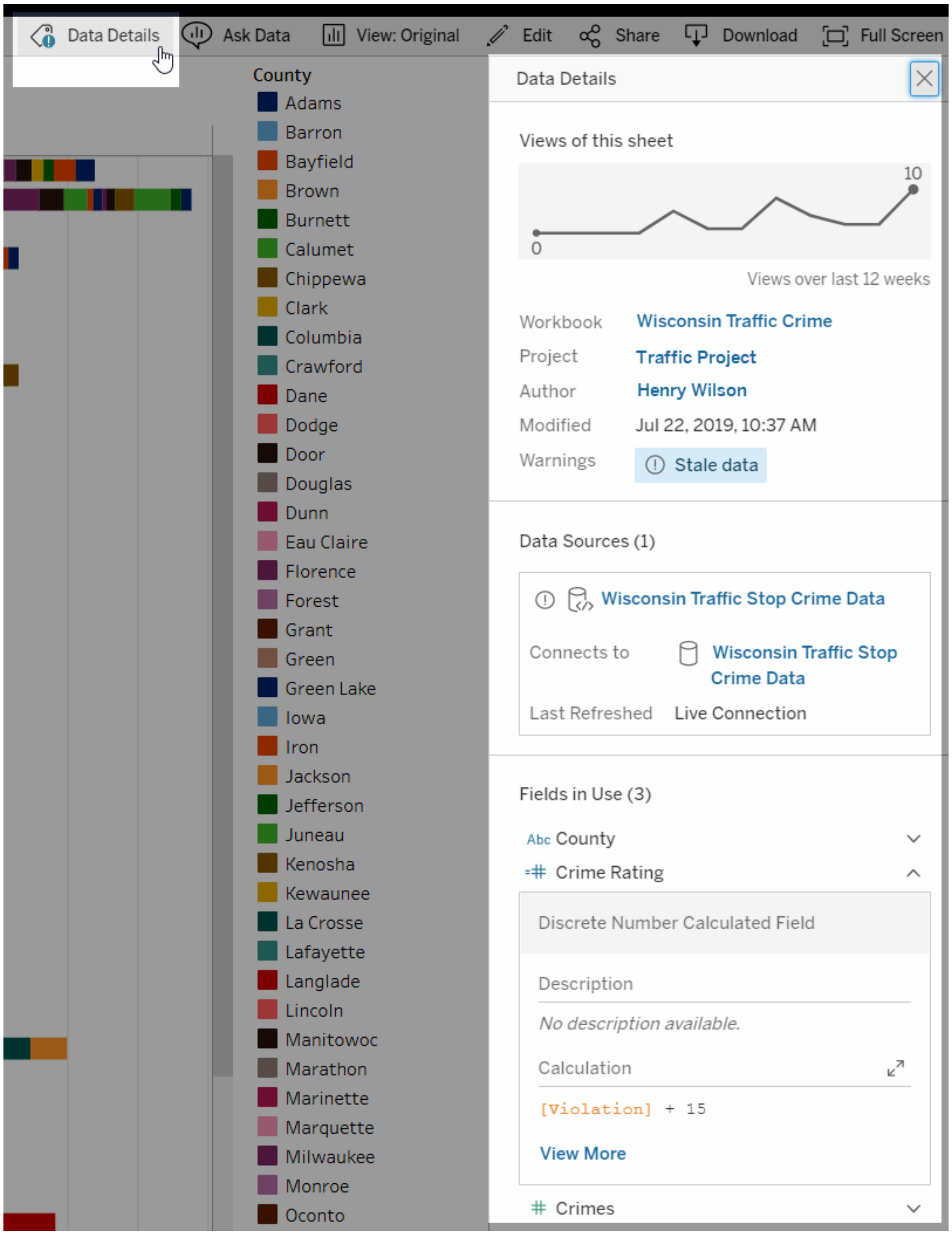
For example, if you create a highlight action in a dashboard, selecting marks in one sheet will highlight marks in the other sheets in the dashboard. If those other sheets have trend lines, reference lines, or other analytics objects, recalculated lines will appear as the views update.

Use Data Details to see visualization information

> Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When the Tableau Catalog is enabled in your environment, you can use the Data Details pane to see visualization information. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

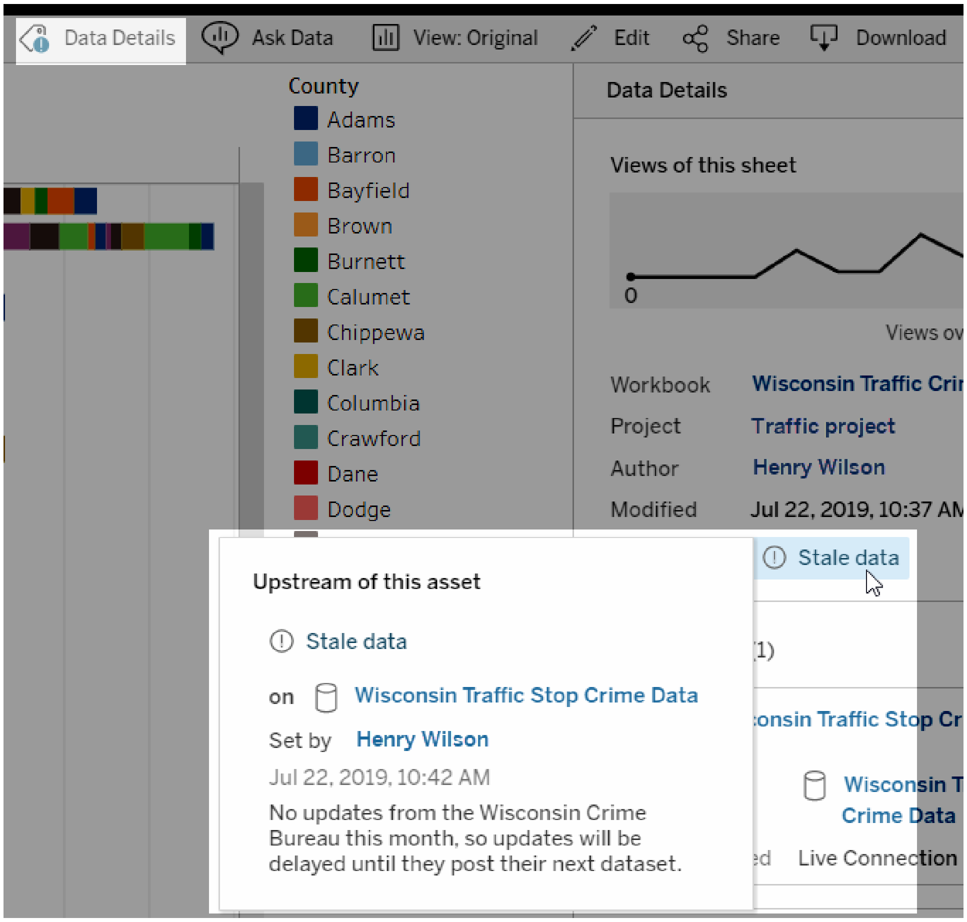
When you're looking at a published view and Tableau Catalog is enabled in your environment, you can see information about the data used in the view by selecting **Data Details** in the toolbar above the view in workbooks and dashboards. Details include:

- A sparkline that shows the number of times a sheet or dashboard was viewed over the last 12 weeks.
- Information about the workbook (name, project, author, date modified, data quality warnings).
- The data sources used in the sheet or dashboard.
- A list of the fields in use. Select a field to see the field metadata, such as a description or calculation.



The links in the **Data Details** pane take you to that item, which opens in a separate window.

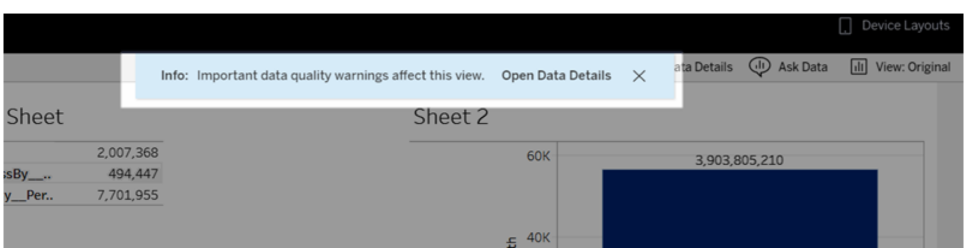
If there's a data quality warning in the lineage of any of the data used in a view, workbook, or dashboard, the **Data Details** tab shows a blue circle with a white exclamation mark in it. Click the **Data Details** tab, then click the data quality warning to see the details of the warning.



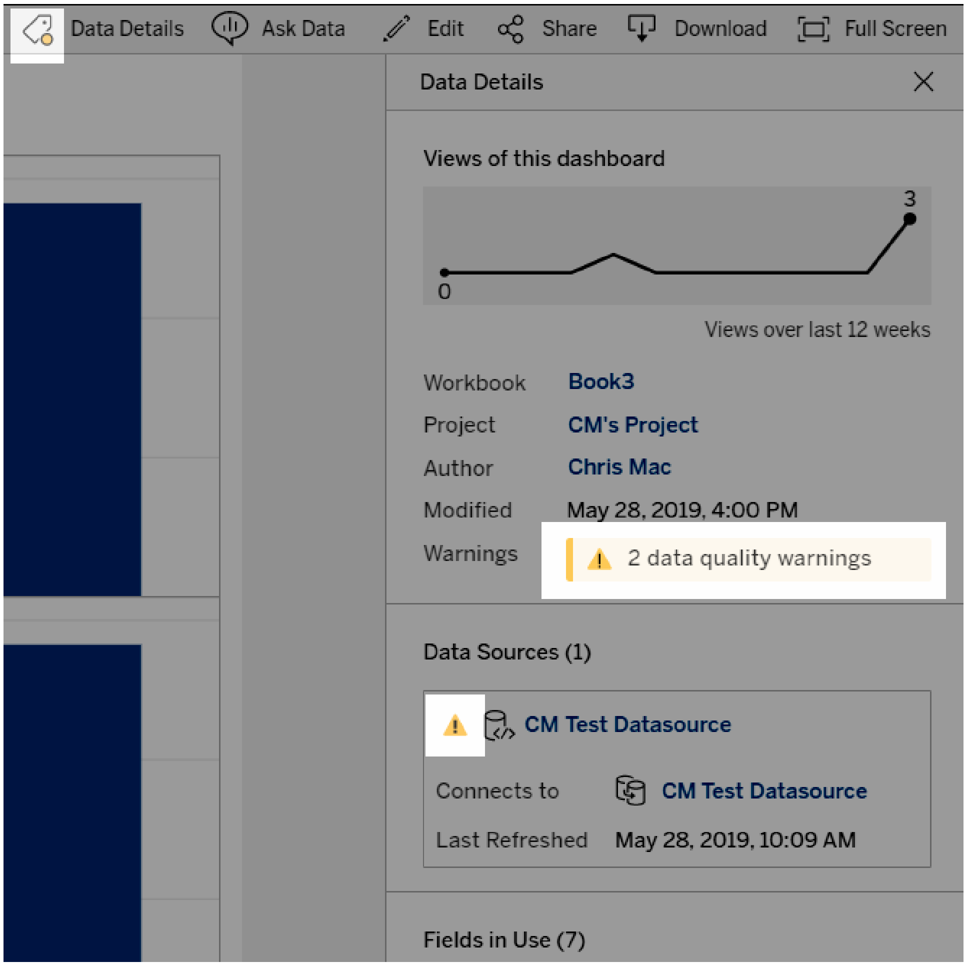
High visibility data quality warnings

When a data quality warning is tagged as important by, for example, the data item owner or an admin:

- A notification appears when anyone opens a published view that uses the affected data, with a link to the Data Details pane:



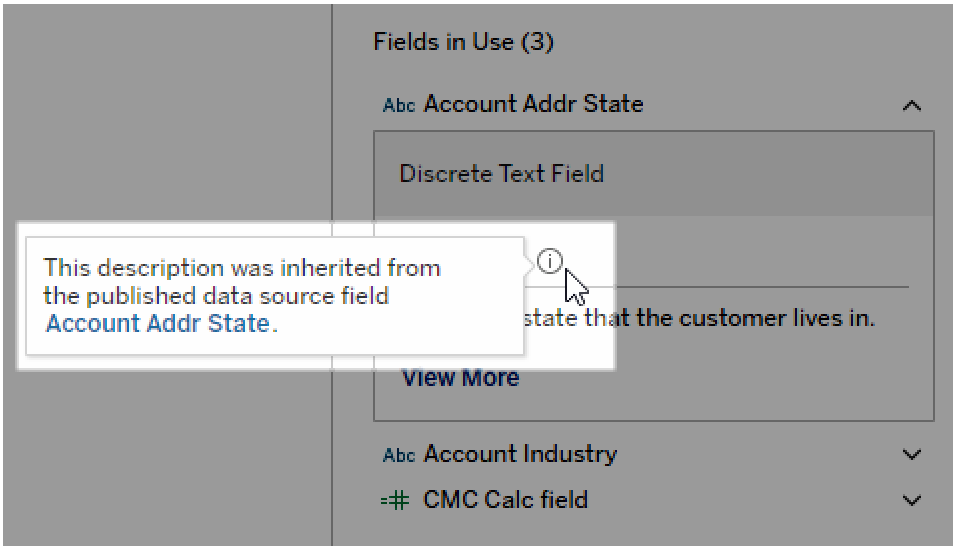
- On the Data Details pane (and elsewhere in Tableau), the warning's visibility is raised by using a yellow warning symbol:



Fields can show upstream descriptions

If a field used in a view doesn't have a description in the workbook, Catalog can look up and display the description from the field or column upstream from it, if one exists. If a published data source is upstream and has a description for this field, the Data Details pane displays that field description. If not, this field shows the description from the column that the field references in the upstream table.

When a field get its description from an upstream source, an information icon appears next to **Description**, telling where the description came from.

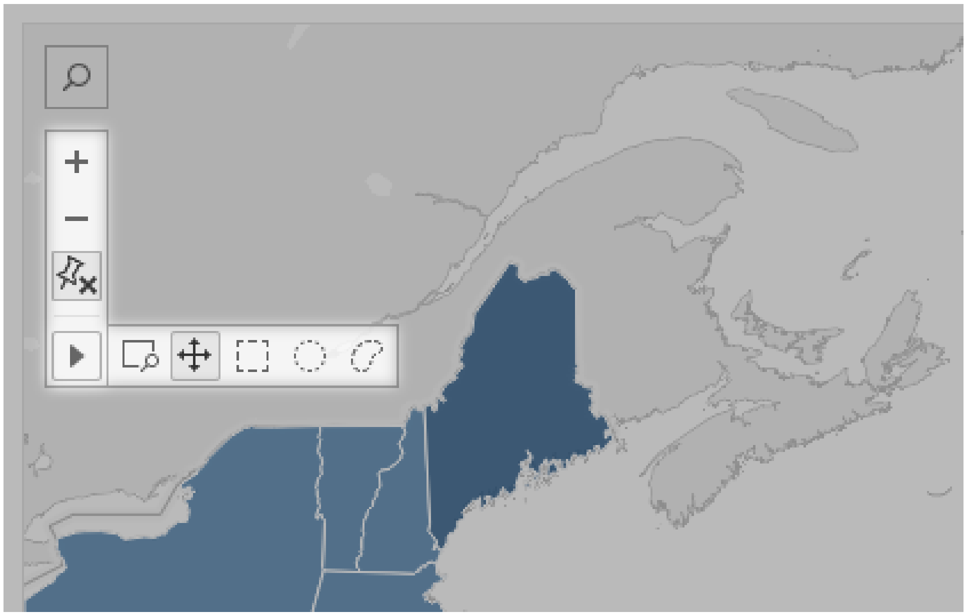


Some fields, such as a calculated field, group field, or set field, don't support showing a description from an upstream source. Also, workbooks and data sources don't support displaying upstream field or column descriptions.

For information about adding descriptions, see "Add or edit descriptions" in **Manage Web Content** on page 3205.

Zoom and Pan Views, and Select Marks

In the upper-left corner of a view, the view toolbar lets you zoom in and out, pan the view, and select data marks.



Show or hide the view toolbar

> **Tip:** If you don't see the view toolbar in Tableau Online or Tableau Server, ask the content owner to show it using these instructions. Or use keyboard shortcuts to zoom, pan, and select marks.

When you hover over a map view, the view toolbar appears by default, but you can change that behavior. For more information, see **Customize How People Interact with your Map** on page 1989. You can also show the toolbar in other types of views, such as scatter or box plots.

1. If you're working with a dashboard, select the zone on which you'd like to show or hide the toolbar.
2. From the top menu, select **Worksheet** > **Show View Toolbar**. Then select one of the following options:
   - **Automatic** shows the toolbar when you hover over a map view.
   - **Show on hover** shows the toolbar when you hover over any type of view.
   - **Hide** doesn't show the toolbar on the selected view. (To improve usability on mobile devices, however, the toolbar will appear when users interact with scrollable views or multiple selections.)

Tip: In Tableau Desktop, you can quickly toggle the toolbar on or off. Right-click anywhere in a view, and select **Show View Toolbar** or **Hide View Toolbar**.

Zoom and pan

The view toolbar provides zoom controls and a pan tool that lets you see a different areas of a zoomed view.

Zoom in and out

On the view toolbar, click the Zoom In button + to zoom in and the Zoom Out button − to zoom out.

If the view toolbar is hidden, double-click the view to zoom in; to zoom out, hold down Shift, and then double-click the view.

Zoom to a specific area

To zoom in to a specific area of the view, click the Zoom Area tool button on the view toolbar, and then drag in the view.

If the view toolbar is hidden, hold down Ctrl + Shift (Wind0ws) or Command+ Shift (macOS) to use the Zoom Area tool.

Reset the view

After you zoom in or out, the axes in the view are fixed to a specific range. To quickly reset the axes so they automatically zoom the view to all of your data, do one of the following:

- In Tableau Desktop, click the Reset Axes button on the view toolbar.
- In Tableau Server or Tableau Online, click the Zoom Home button on the view toolbar.

Pan a view

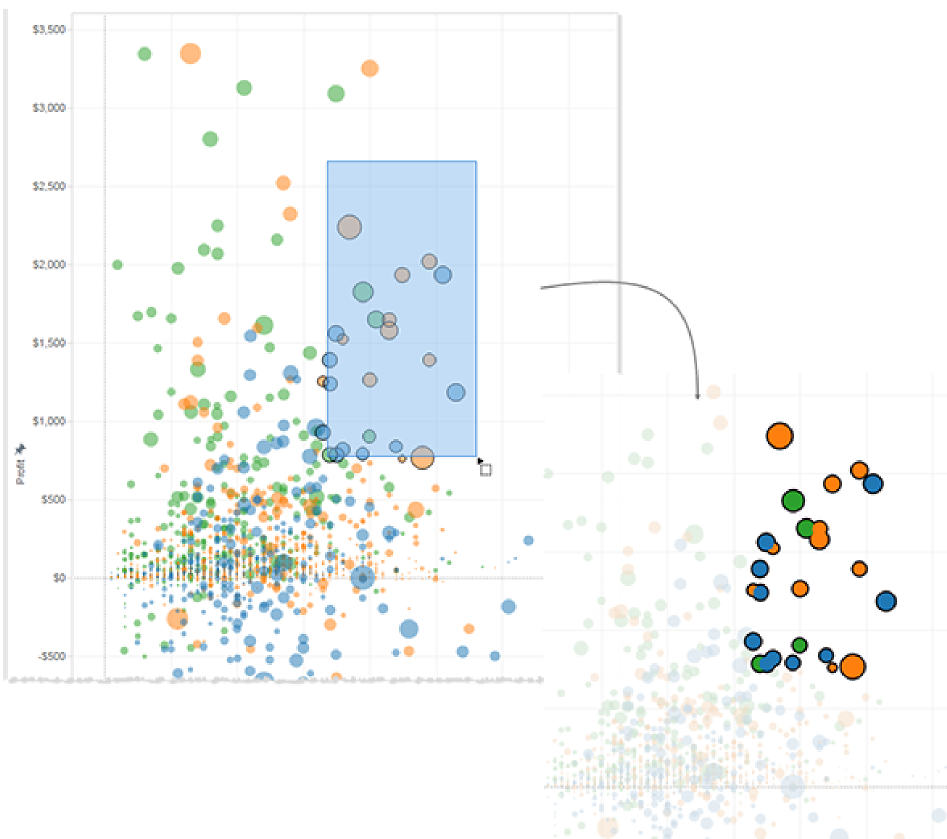
To pan, do one of the following:

- Hold down Shift, and then drag across the view.
- On the view toolbar, hover over the arrow, select the Pan tool , and then drag across the view.

Select marks

Select marks to see information about them in tooltips that appear. You can also quickly view underlying data or filter selected marks from a view. For more information, see **Tooltips** on page 250.

When you select marks in the view, all other marks are dimmed to draw attention to the selection. Your selection is saved with the workbook.

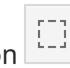


Select individual or multiple marks

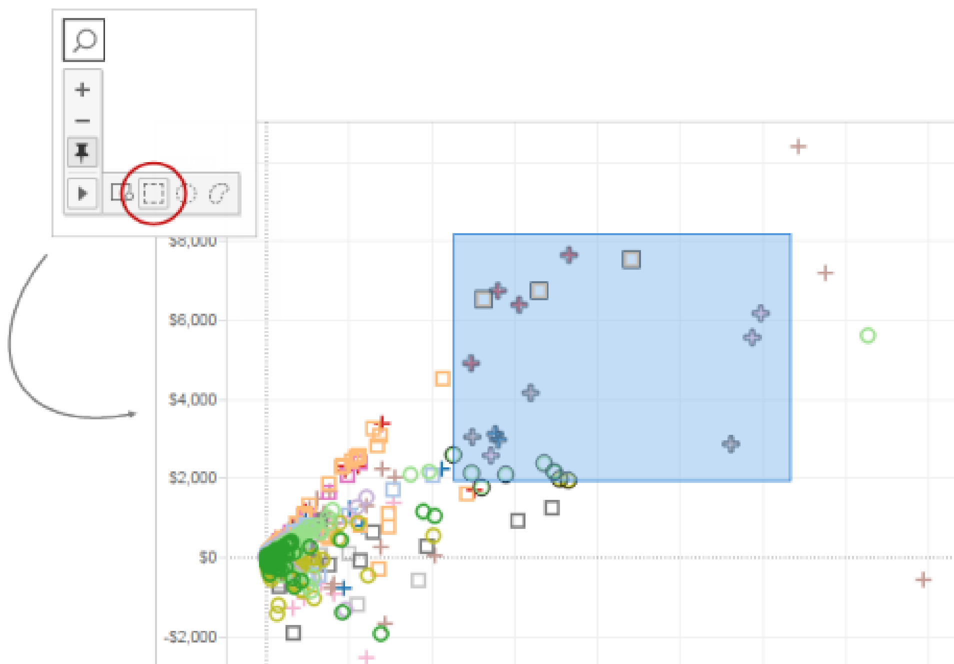
To select an individual mark, click it. To add more individual marks to a selection, hold down the Ctrl key (Windows) or the Command key (macOS).

To select multiple marks at once, drag across the view while using the Rectangular, Radial, and Lasso tools on the view toolbar.

Make rectangular selections

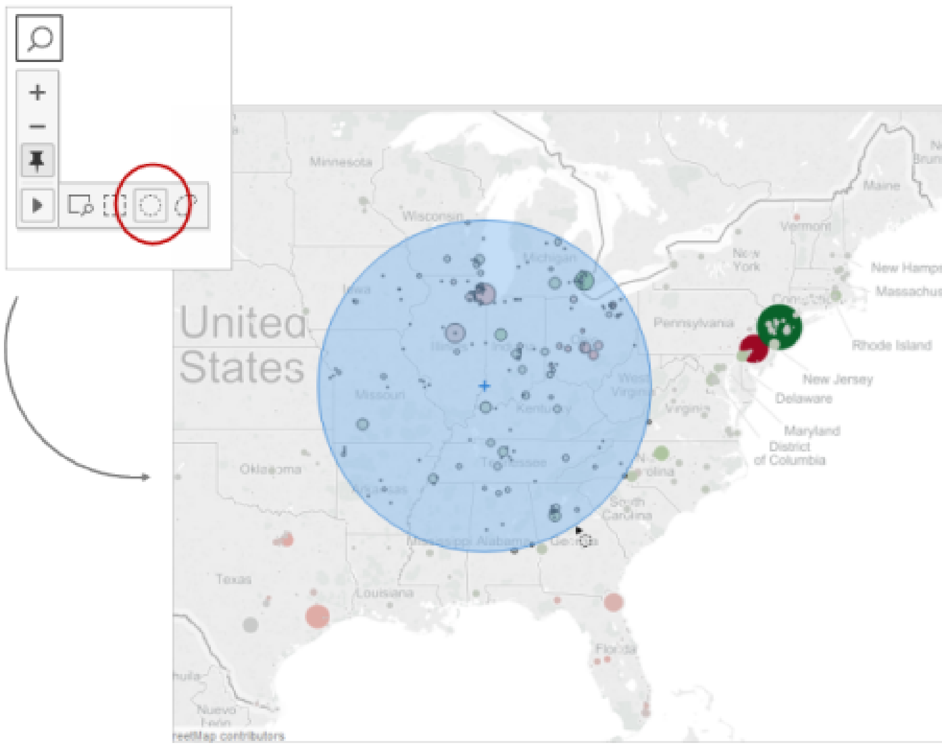
The Rectangular tool selects marks within a rectangular shape. To use the Rectangular tool, hover over the arrow on the view toolbar, click the Rectangular tool button , and then drag across the view.

> **Note:** The Rectangular tool is the default tool in most views and may not appear in the view toolbar. In this case, simply drag across the view to use the rectangular tool.



Make radial selections

The Radial tool selects marks within a circular area. To use the Radial tool, hover over the arrow on the view toolbar, click the Radial tool button , and then drag across the view.



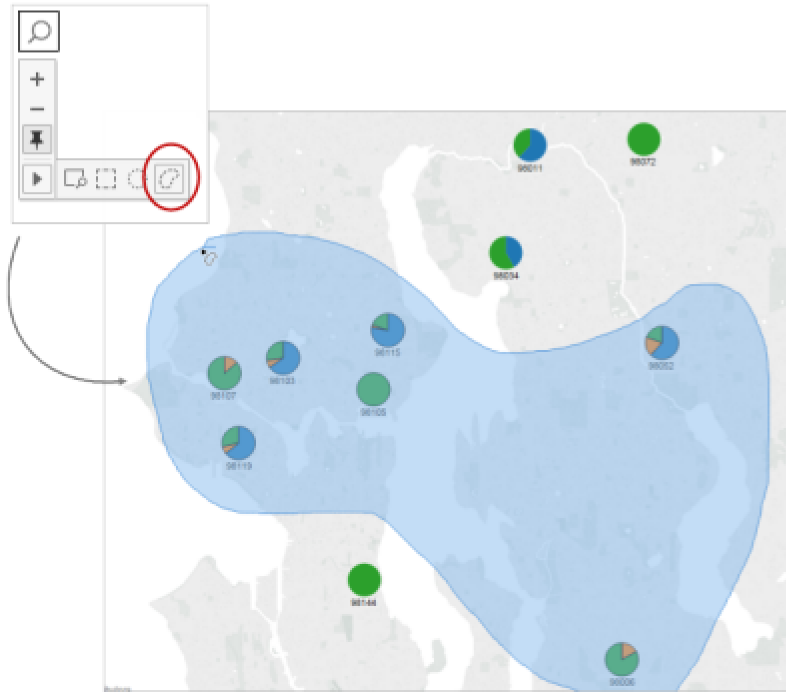
You can also measure distance in a map view with the Radial tool. For more information, see **Measure Distances Between Data Points and Locations in a Map** on page 2007.

Make lasso selections

With the Lasso tool, you can select multiple marks by drawing a freehand shape around them. This tool is useful when you want to include only certain marks, and exclude others around them.

To use the Lasso tool, hover over the arrow on the view toolbar, click the Lasso tool button

, and then draw a freehand shape around the marks you want to select.



Undo and Redo

You can perform unlimited undo and redo of almost all actions in Tableau Desktop, as well as the content viewing and authoring modes of Tableau Online and Tableau Server. For example, you can quickly return to a sheet you were just viewing, or you can cycle through changes to a data source you're editing.

Tableau saves your undo and redo history across all worksheets until you exit; the history isn't saved between sessions.

Keyboard shortcuts

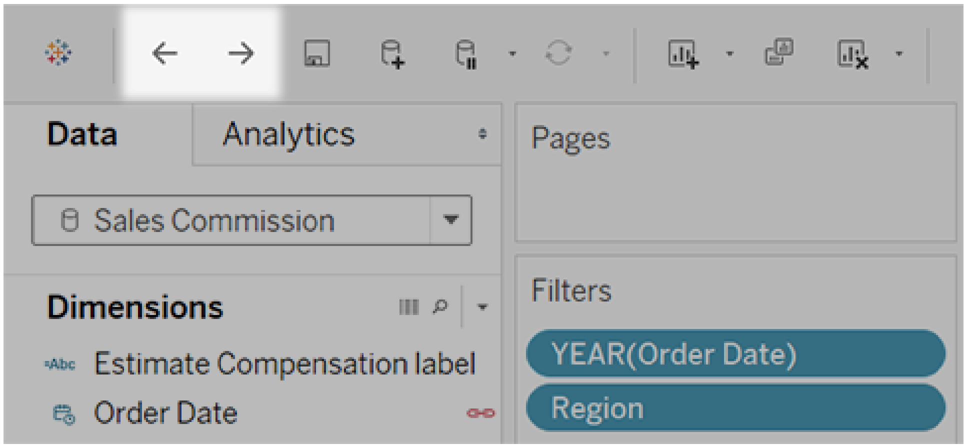
To quickly undo or redo changes, use these keyboard shortcuts:

- On Windows, press Ctrl+Z to undo and Ctrl+Y to redo.
- On macOS, press Command+Z to undo and Command+Shift+Z to redo.

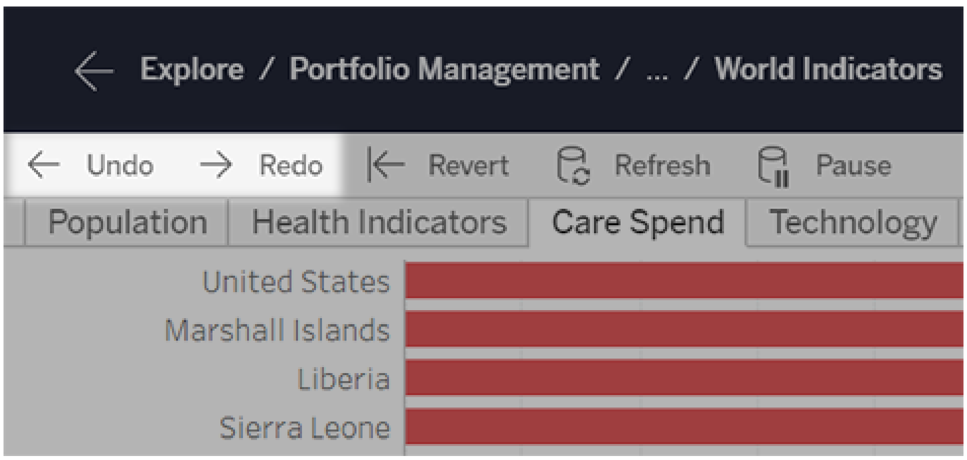
Undo and Redo buttons

In the upper-left corner of Tableau Desktop, Tableau Online, and Tableau Server, you can click Undo and Redo buttons.

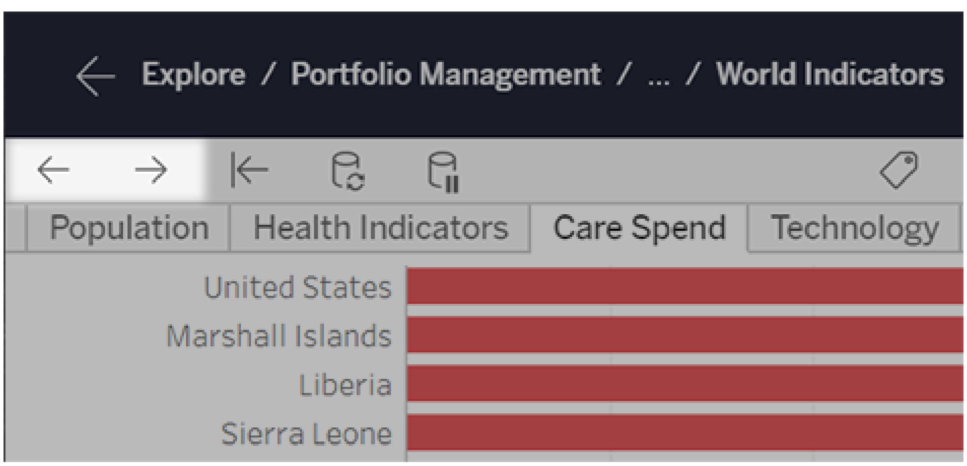
Here's how those buttons appear in Tableau Desktop:



And here's how they appear in Tableau Online and Tableau Server:



If your browser window is more narrow, the buttons appear like so:



I don't see the toolbar for these buttons

Well, you've got a few options. Either use the keyboard shortcuts above, or do the following:

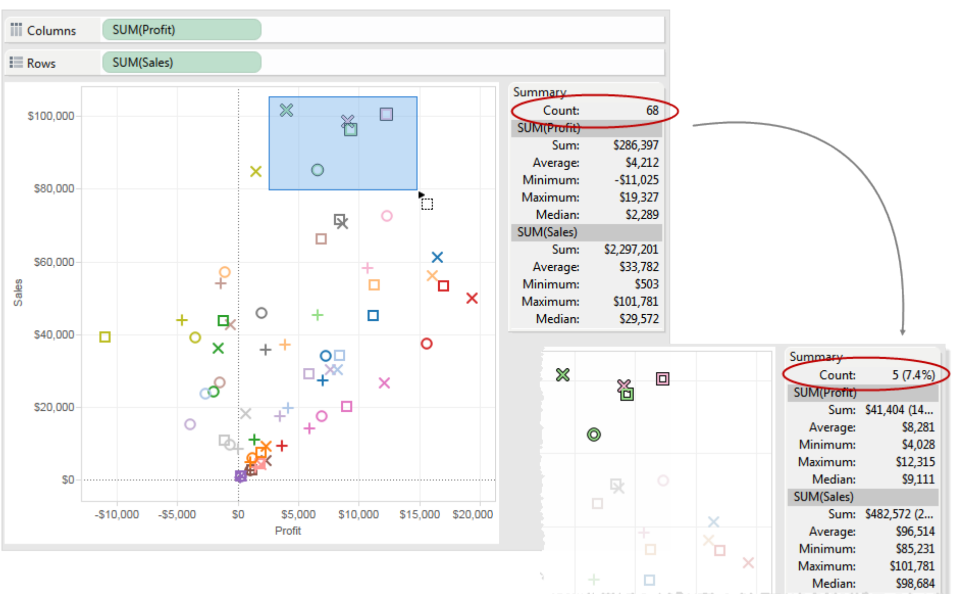
- For Tableau Desktop, select **Window** > **Show Toolbar**.
- For an embedded view from Tableau Online or Tableau Server, ask the content owner to unhide the toolbar using the “toolbar” URL parameter.

Now start making some wild changes in Tableau so you can undo and redo them to your heart's content!

Summary Card

The **Summary Card**, available on the **Show/Hide Cards** toolbar menu , provides a quick view of information about a selection or the entire data source.

When you select data in the view, the **Summary Card** updates to show you information only for the data within the selection:



By default, the **Summary Card** shows Sum, Average, Minimum, Maximum, and Median values for the data in the view. The summary card values update to show these values for the current selection of data points. (Average is computed by summing all relevant values and then dividing by the total number of values. Median is computed by sorting values from lowest to highest and then selecting the middle value.) The Count value at the top of the card indicates the number of marks in the view or selection.

You can use the drop-down menu for the **Summary Card** to show additional statistics:

- **Standard Deviation**

  A measure of data spread around its average, measured in the same units as the data itself. The sample standard deviation is an unbiased estimate of the population standard deviation given a slight correction. This standard deviation includes the correction.

- **First Quartile**

  A measure of location that is commonly used with other quartiles to provide a robust measure of spread. Robust in this case means not as sensitive to outliers as the standard deviation. The first quartile is the 25th percentile, typically the lower line in a boxplot.

- **Third Quartile**

  A measure of location that is commonly used with other quartiles to provide a robust measure of spread. Robust in this case means not as sensitive to outliers as the standard deviation. The third quartile is the 75th percentile, typically the upper line in a boxplot.

- **Skewness**

  A measure of the tendency of your data to have extreme values to one side. Positive skewness means the extreme values are to the right, while negative skewness means the extreme values are to the left.

- **Excess Kurtosis**

  A measure of the tendency of your data to have more extreme or outlying values than a normal distribution. A normal distribution has a kurtosis of 3 so this value is kurtosis minus three.

View Underlying Data

View Data allows you to inspect your data in a spreadsheet-like layout. You can view data either for the data source as a whole, or to see the underlying data for an individual mark or a group of marks. In a worksheet, the rows that you see in the View Data window are always scoped to the current selection or the current view.

The View Data window displays as much of the data as possible by default, up to 10,000 rows. Field names are shown as column headers and can be dragged and dropped to change their display order. Click a column header to sort the values in that column.

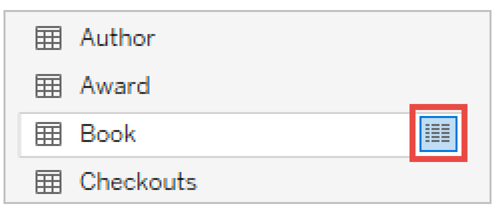
> **Important: View Data column order has changed in Tableau 2020.2 and later**: When you open a data source from a previous version of Tableau in Tableau 2020.2 or later, the column order may be different. Columns may be displayed differently in the View Data window, and the column order may be different when you export it to CSV format. This change in how column order is handled in the View Data window is required to support relationships and logical tables. If you use scripts that depend on a customized column order, we recommend using the Tableau Extensions API for View Data to download underlying data to CSV format.

View Data from the data source

The View Data icon is available on the Data Source page and on the Data pane on a sheet. Viewing data from the data source before or during analysis can be useful to give you a sense of the scope of each table, its fields and their values, and general impressions about the data.

Data Source page

In the Data Source page, hover over a table or sheet in the left pane to display the View Data icon .

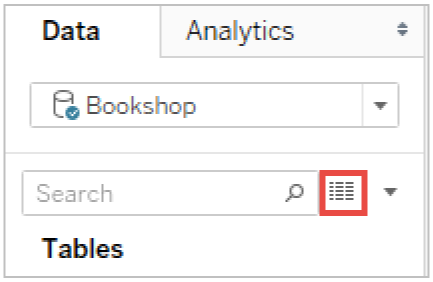


The View Data window displays the data for that table.

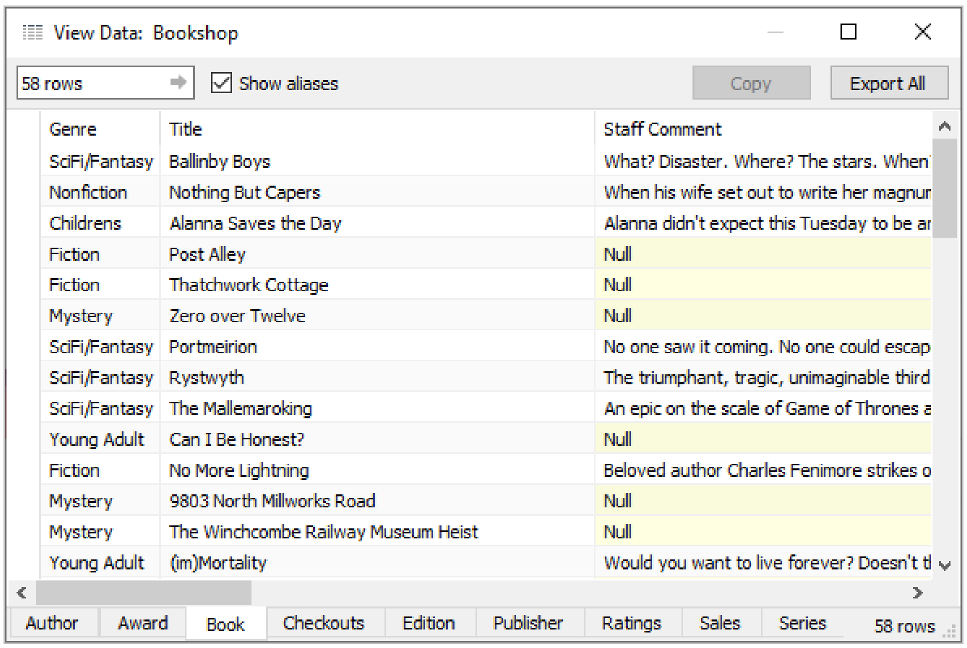
Be aware that the two icons at the top of the data grid in the Data Source page toggle the data grid presentation between “Preview the data source” and “Manage metadata” and neither are View Data.

Data pane

In a worksheet, the View Data icon is located at the top of the Data pane, below the data source list and to the right of the Search box.



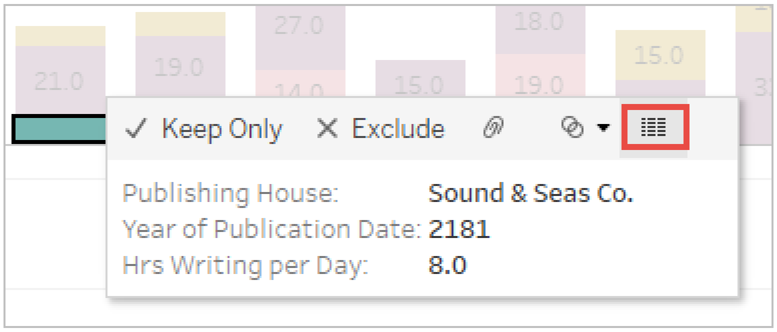
The View Data window displays a tab for every table in the data source. Tables that are joined or unioned make up a single tab, as they are represented as a single logical table in the data model.



View Data for a mark

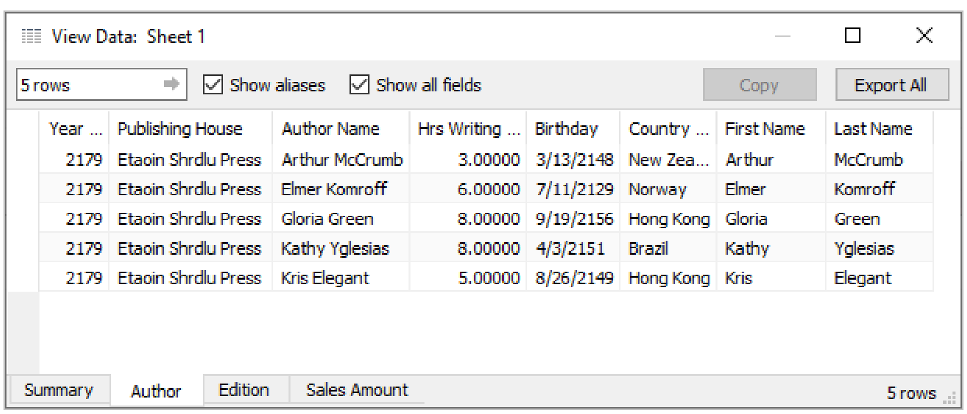
Tableau visualizations are, essentially, created by aggregating measures based on the structure of the dimensions in the view and displaying the results as a mark. You can use View Data to reveal the underlying data represented in that mark.

Click on a mark to display its the tooltip. In the tooltip menu, click the View Data icon . You can also select multiple marks simultaneously and see their combined View Data results.



The View Data window displays multiple tabs:

- The summary tab which shows the dimensions that define the view and the aggregated measure or measures that make up the mark.
- A tab for each table that has a measure present in the view, showing all fields from that table.
  - If “Show all fields” is checked, each tab will also show all dimensions that define the view, regardless of which table they’re from.
- Some calculations used in the view may get their own tab as well:
  - Calculations that do not belong with a table in the Data pane, such aggregate calculations or calculations that span tables.
  - LOD calculations (`FIXED` or `INCLUDE` only), even if they reside in a table in the Data pane. (Ad-hoc LOD calculations are not shown in View Data.)
  - All relevant fields used in the calculation will be displayed, regardless of their original table, as well as the value of the calculation itself.



Year ...	Publishing House	Author Name	Hrs Writing ...	Birthday	Country ...	First Name	Last Name
2179	Etaoin Shrdlu Press	Arthur McCrumb	3.00000	3/13/2148	New Zea...	Arthur	McCrumb
2179	Etaoin Shrdlu Press	Elmer Komroff	6.00000	7/11/2129	Norway	Elmer	Komroff
2179	Etaoin Shrdlu Press	Gloria Green	8.00000	9/19/2156	Hong Kong	Gloria	Green
2179	Etaoin Shrdlu Press	Kathy Yglesias	8.00000	4/3/2151	Brazil	Kathy	Yglesias
2179	Etaoin Shrdlu Press	Kris Elegant	5.00000	8/26/2149	Hong Kong	Kris	Elegant

View Data from the Analysis menu

To see the underlying data for the entire view, in the **Analysis** menu, select **View Data**. The View Data window is displayed with similar results for viewing data for a mark, but instead it displays all of the data in use in the view.

Copy or Export All from the View Data dialog

You can copy the data out in the View Data window for use in a spreadsheet or another Tableau workbook (really, anywhere you can paste text).

- Select the rows or columns of interest (or the empty square to the left of the first column header to select the entire display) and click the Copy button, then paste the data into a spreadsheet, another Tableau workbook, or another destination of your choice.
- Click the **Export All** button to export the current tab as a .csv file. **Export All** is not limited to 10,000 rows like the View Data window display.

If you are a workbook author and want to ensure your audience can't copy out the underlying data, you may want to consider turning off the command buttons in the tooltips or denying workbook permissions to the summary or full data.

Why can't I see the View Data icon?

If you are in Tableau Desktop and cannot find the View Data icon for a mark, check that tooltips are enabled and the tooltip menu command buttons are not turned off. Alternatively, right click on a mark and select View Data from the menu. This method will work even if tooltips or command buttons aren't enabled.

If you are looking at a published viz and cannot find the View Data icon for a mark, it may be that tooltips are not present, the command buttons have been turned off, or you may not have the necessary permissions.

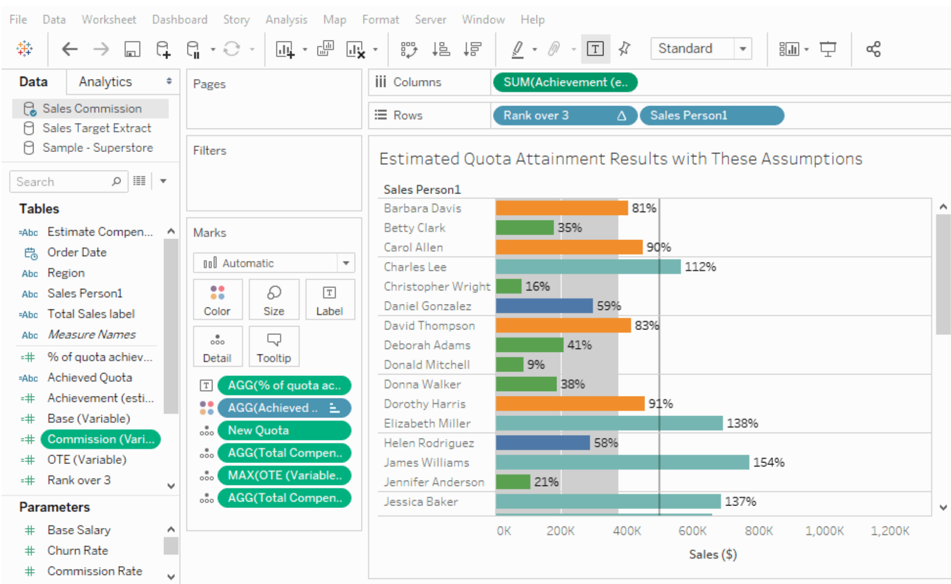
Get Details About Fields and Sheets in a Workbook

When you want to see details about all of the elements in use in a viz, select the **Worksheet** menu, and then click **Describe Sheet**. For details about a field (such as data role, data type, domain, aggregation, calculation formula), click **Describe** in a field's context menu in the **Data** pane. You can then share those details with someone else by copying that summary text.

> **Note**: For related information about using Describe, see these blog posts from Tableau Zen Masters Ryan Sleeper (PlayfairData) and Bridget Cogley (TableauFit): 3 Ways the Describe Feature Can Help You Use Tableau and 10 Intro Tips to Tableau When You're Handed Someone Else's Workbooks.

Display sheet details

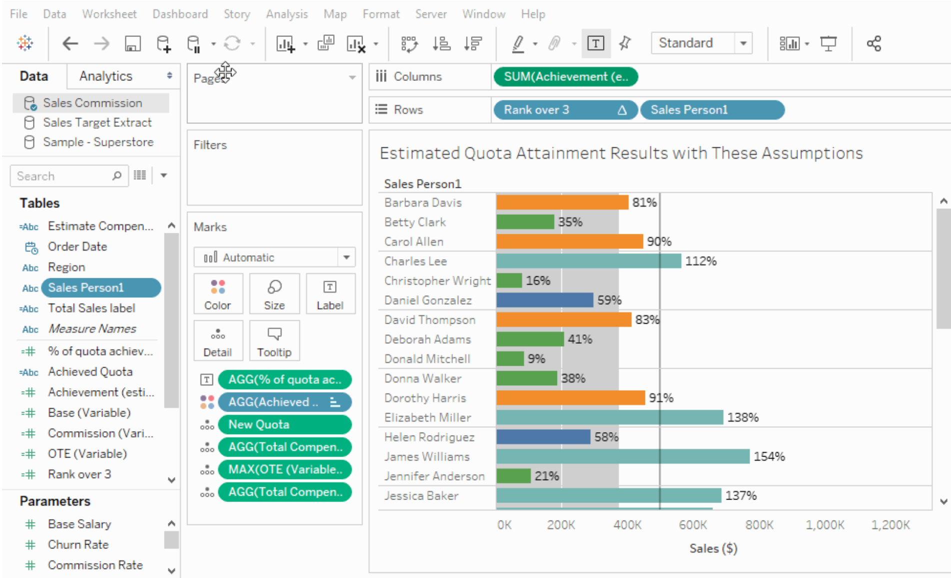
When you click **Worksheet** > **Describe Sheet**, you can view a description of the workbook, data source, fields and layout of the current worksheet. This summary includes the Caption in the first line, but provides other important summary information. You can then copy and paste this information into other applications.



> **Note:** If you have Trend Lines turned on, the Describe Sheet dialog box includes information about the trend line model. For more information, see **Assess Trend Line Significance** on page 2540. If you have Forecasting turned on, the Describe Sheet dialog box includes information about estimated data. For more information, see **Forecast Descriptions** on page 2597.

Display field details

When you click **Describe** in the context menu of a field in the Data pane, you can view a description of the workbook, data source, fields and layout of the current worksheet. This summary includes the Caption in the first line, but provides other important summary information. You can then copy and paste this information into other applications.


File Data Worksheet Dashboard Story Analysis Map Format Server Window Help
Standard
Data Analytics
Sales Commission
Sales Target Extract
Sample - Superstore
Search
Tables
Estimate Compen...
Order Date
Region
Sales Person1
Total Sales label
Measure Names
% of quota achiev...
Achieved Quota
Achievement (esti...
Base (Variable)
Commission (Vari...
OTE (Variable)
Rank over 3
Parameters
Base Salary
Churn Rate
Commission Rate
Pages
Filters
Marks
Automatic
Color
Size
Label
Detail
Tooltip
AGG(% of quota ac..
AGG(Achieved ..
New Quota
AGG(Total Compen..
MAX(OTE (Variable..
AGG(Total Compen..
Columns
SUM(Achievement (e..
Rows
Rank over 3
Sales Person1
Estimated Quota Attainment Results with These Assumptions
Sales Person1
Barbara Davis 81%
Betty Clark 35%
Carol Allen 90%
Charles Lee 112%
Christopher Wright 16%
Daniel Gonzalez 59%
David Thompson 83%
Deborah Adams 41%
Donald Mitchell 9%
Donna Walker 38%
Dorothy Harris 91%
Elizabeth Miller 138%
Helen Rodriguez 58%
James Williams 154%
Jennifer Anderson 21%
Jessica Baker 137%
0K 200K 400K 600K 800K 1,000K 1,200K
Sales ($)

Dashboards

A dashboard is a collection of several views, letting you compare a variety of data simultaneously. For example, if you have a set of views that you review every day, you can create a dashboard that displays all the views at once, rather than navigate to separate worksheets.

Like worksheets, you access dashboards from tabs at the bottom of a workbook. Data in sheets and dashboards is connected; when you modify a sheet, any dashboards containing it change, and vice versa. Both sheets and dashboards update with the latest available data from the data source.

Best Practices for Effective Dashboards

A well-designed dashboard can align your organization's efforts, help uncover key insights, and speed up decision-making. Use this topic for tips on best practices for creating effective dashboards in Tableau.

What's your goal?

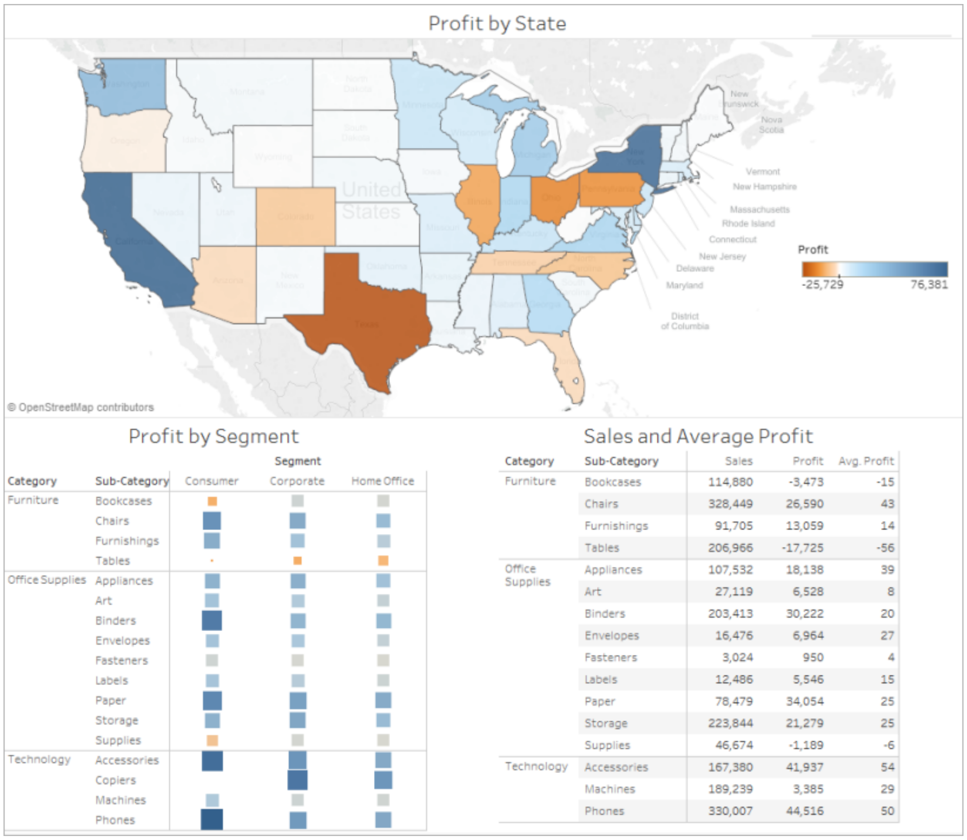
Know your purpose and audience

The best visualizations have a clear purpose and work for their intended audience. What will you be trying to say with this dashboard? Are you presenting a conclusion or a key question?

In addition to knowing what you're trying to say, it's important to know who you're saying it to. Does your audience know this subject matter extremely well or will it be new to them? What kind of cues will they need? Thinking about these questions before you head into the design phase can help you create a successful dashboard.

Leverage the most-viewed spot

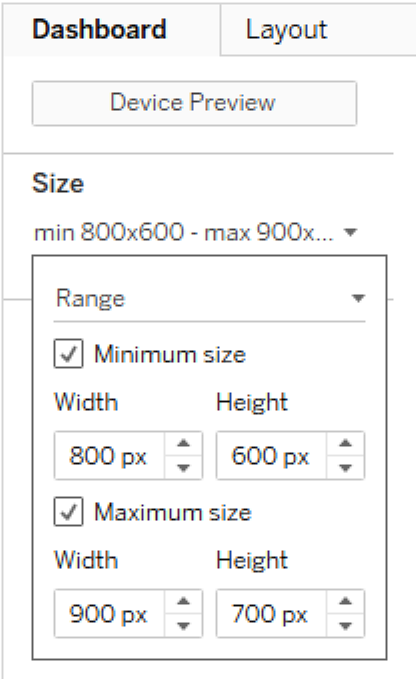
Most viewers scan web content starting at the top left of a web page. Once you know your dashboard's main purpose, be sure to place your most important view so that it occupies or spans the upper-left corner of your dashboard. In the dashboard below, the author decided that the map view holds the key message.



Design for the real world

Author at your final display size

By default, Tableau dashboards are set to use a fixed size and if you keep this setting, be sure to construct your visualization at the size it will be viewed at. You can also set **Size** to **Automatic**, which makes Tableau automatically adapt the overall dimensions of a visualization based on screen size. This means that if you design a dashboard at 1300 x 700 pixels, Tableau will resize it for smaller displays—and sometimes this results in scrunched views or scrollbars. The **Range** sizing feature is helpful for avoiding this.

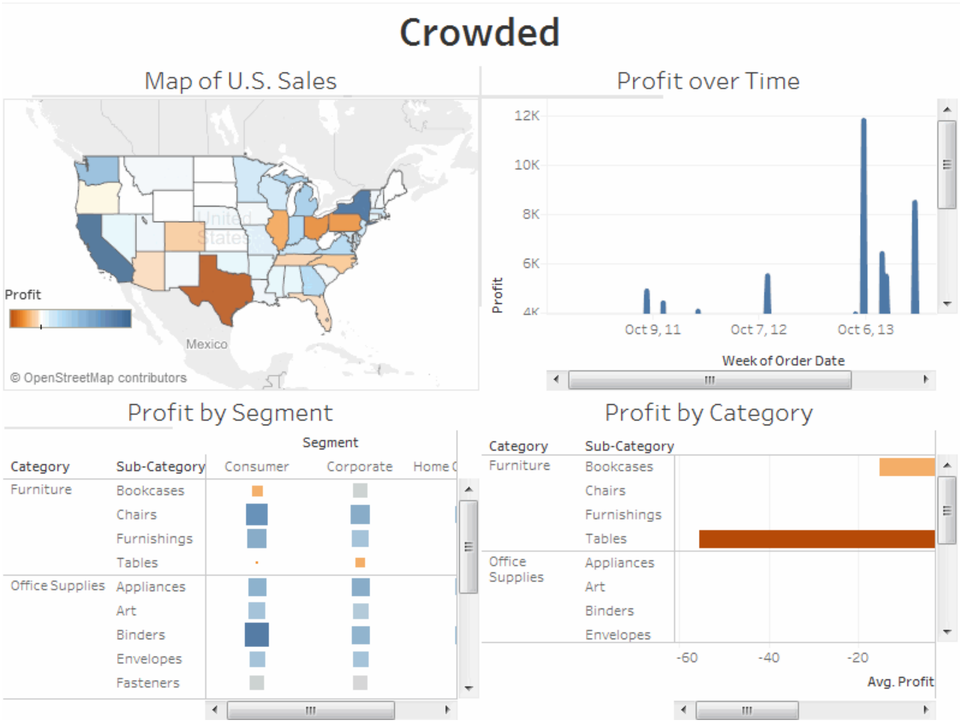


If you're using Tableau Desktop to create dashboards, you can also design for specific device layouts, so that on tablets, for example, your dashboard contains one set of views and objects, and on phones it displays another. See **Create Dashboard Layouts for Different Device Types** on page 2801 for steps.

Limit the number of views

In general, it's a good idea to limit the number of views you include in your dashboard to two or three. If you add too many views, visual clarity and the big picture can get lost in the details. If you find that the scope of your story needs to grow beyond two or three views, you can always create more dashboards.

Too many views can also interfere with the performance of your dashboard after it's published. See **Make Visualizations Faster** on page 3006 for more details on performance.



Click the image to replay it.

Add interactivity to encourage exploration

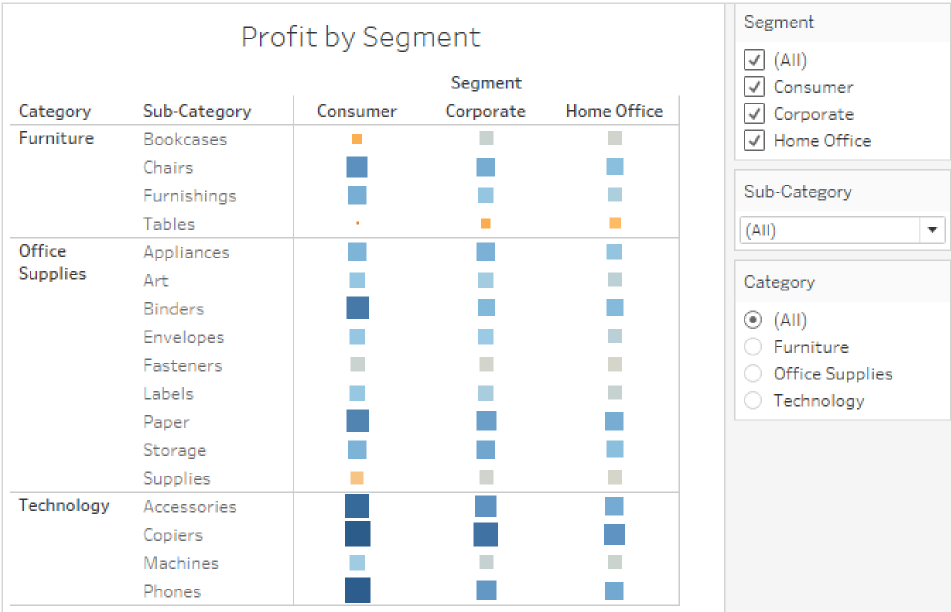
Show filters

Filters help users specify which data is shown in the view.

To turn on filters for a field:

- In Tableau Desktop—Right-click the field in the Data window and select **Show Filter**.
- In Tableau Server or Tableau Online—In the toolbar click **Show/Hide Cards** > **Filters**.

You can customize each filter for different types of data. For example, you can show filters as multi-select check boxes, single select radio buttons, or drop-down lists, etc. You can include a search button, the option to show all fields, null controls, and more. You can also edit the title of a filter to give your viewers clear instructions for interacting with the data.



Enable highlighting

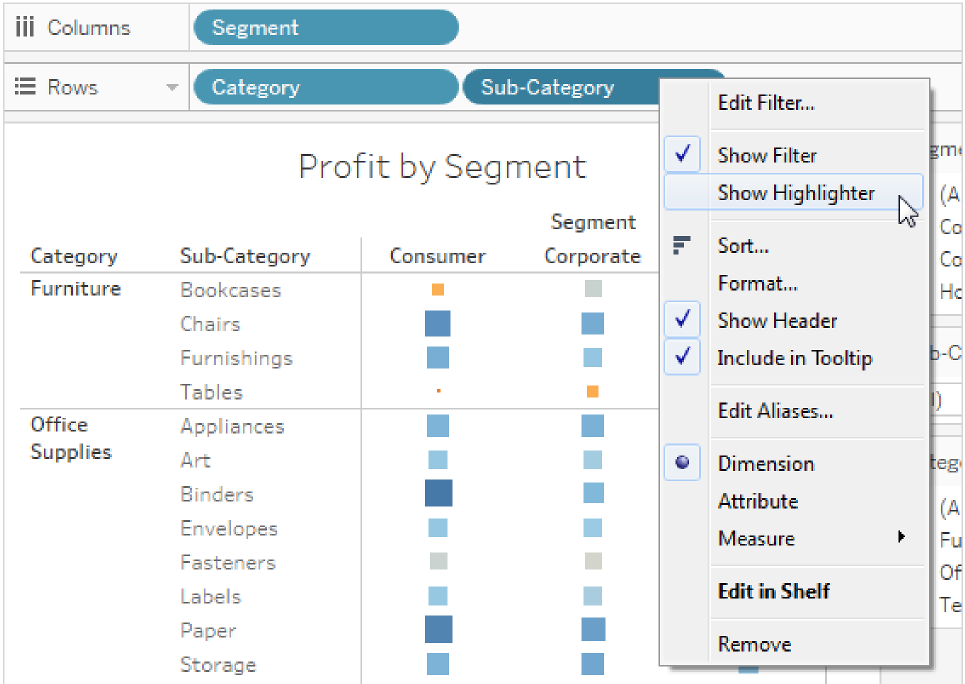
You can use the **Highlight** button on the toolbar to set up highlighting between views. When highlighting is turned on, a selection in one view will highlight related data in the other views. You can turn on highlighting for all fields or select specific fields. For more information about the different methods you can use to highlight data, see Highlight Actions.

You can also display a highlighter that allows your customers to highlight parts of a view based on what they enter or select.

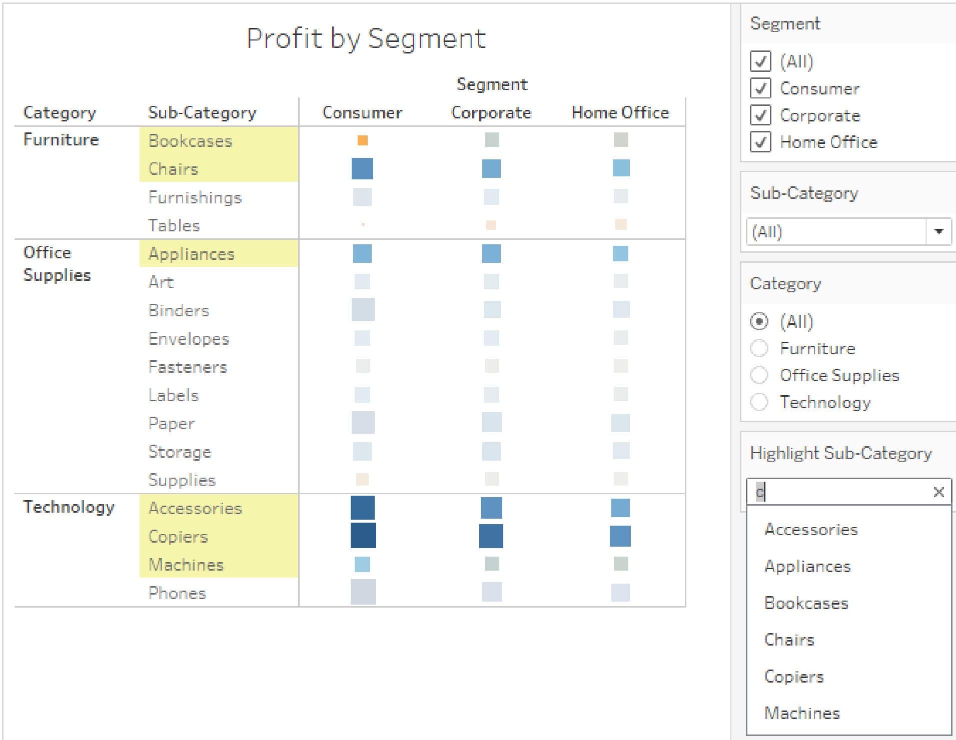
To display a highlighter:

1. Go to the worksheet where the view is (or select **Go to Sheet** from the dashboard).

2. Right-click the field you want to highlight and choose **Show Highlighter**:



In the highlighter, your users will be able to select or enter terms to highlight data in the view:



Create a Dashboard

After you've created one or more sheets, you can combine them in a dashboard, add interactivity, and much more.

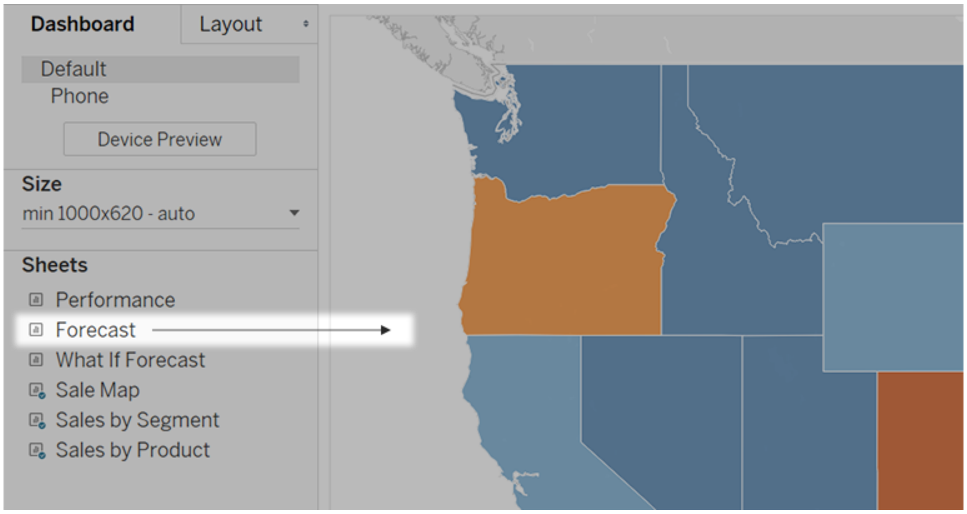
Tip: To quickly get up and running, see **Dashboard Starters for Cloud-based Data** on page 2750.

Create a dashboard, and add or replace sheets

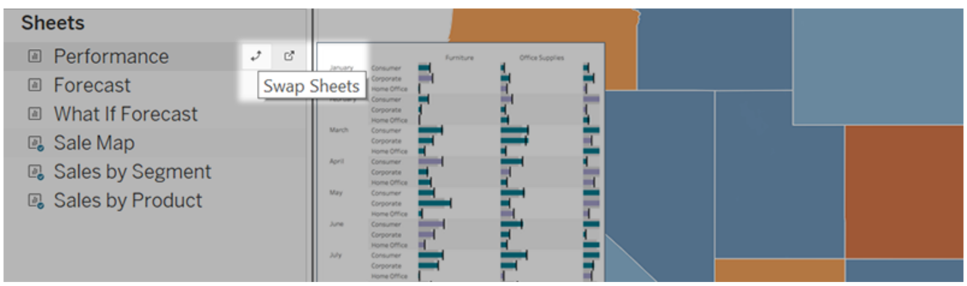
You create a dashboard in much the same way you create a new worksheet.

1. At the bottom of the workbook, click the **New Dashboard** icon:

2. From the **Sheets** list at left, drag views to your dashboard at right.



3. To replace a sheet, select it in the dashboard at right. In the Sheets list at left, hover over the replacement sheet, and click the **Swap Sheets** button.

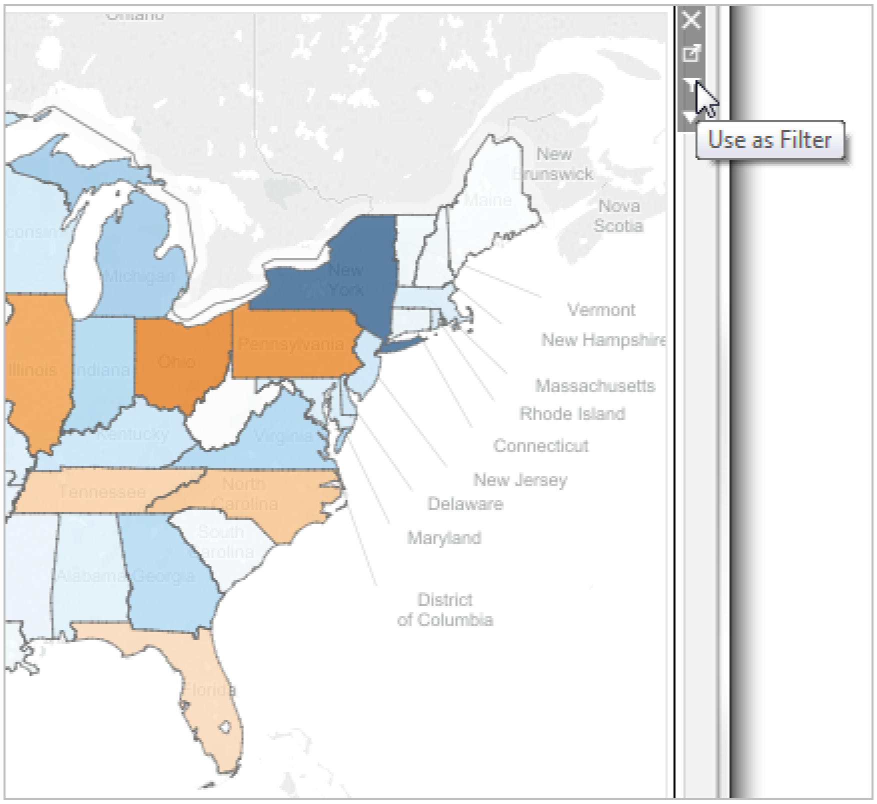


Note: When you replace a sheet, Tableau retains any padding, border, or background color. However, you may need to adjust sheet size if content differs significantly. You may also need to delete dashboard items specific to the previous sheet, such as filters, which become blank.

Add interactivity

You can add interactivity to dashboards to enhance users' data insights. Try these techniques:

- In the upper corner of sheet, enable the **Use as Filter** option to use selected marks in the sheet as filters for other sheets in the dashboard.



- When authoring in Tableau Desktop, add actions to use multiple sheets as filters, navigate from one sheet to another, display web pages, and more. For details, see **Actions and Dashboards** on page 1566.

Add dashboard objects and set their options

In addition to sheets, you can add dashboard objects that add visual appeal and interactivity. Here's guidance about each type:

- **Horizontal** and **Vertical** objects provide layout containers that let you group related objects together and fine-tune how your dashboard resizes when users interact with them.
- **Text** objects can provide headers, explanations, and other information.
- **Image** objects add to the visual flavor of a dashboard, and you can link them to specific target URLs. (While Web Page objects can also be used for images, they are better for complete web pages. The Image object provides image-specific fitting, linking, and alt-text options.)
- **Web Page** objects display target pages in the context of your dashboard. Be sure to review these web security options, and be aware that some web pages don't allow

themselves to be embedded—Google is one example.

> **Note:** For security reasons, your Tableau administrator may prevent Web Page and Image objects from displaying target URLs.

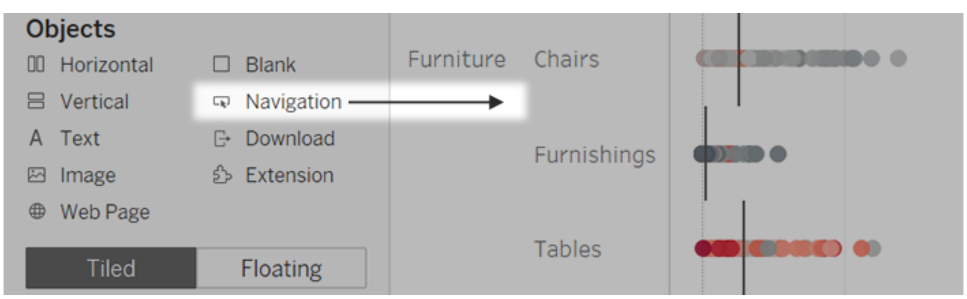
- **Blank** objects help you adjust spacing between dashboard items.
- **Navigation** objects let your audience navigate from one dashboard to another, or to other sheets or stories. You can display text or an image to indicate the button's destination to your users, specify custom border and background colors, and provide informational tooltips.
- **Download** objects let your audience quickly create a PDF file, PowerPoint slide, or PNG image of an entire dashboard, or a crosstab of selected sheets. Formatting options are similar to Navigation objects.

> **Note:** Crosstab download is possible only after publishing to Tableau Online or Tableau Server.

- Extension objects let you add unique features to dashboards or integrate them with applications outside Tableau.
- Ask Data objects let users enter conversational queries for specific data source fields, which authors optimize for specific audiences such as sales, marketing, and support staff.

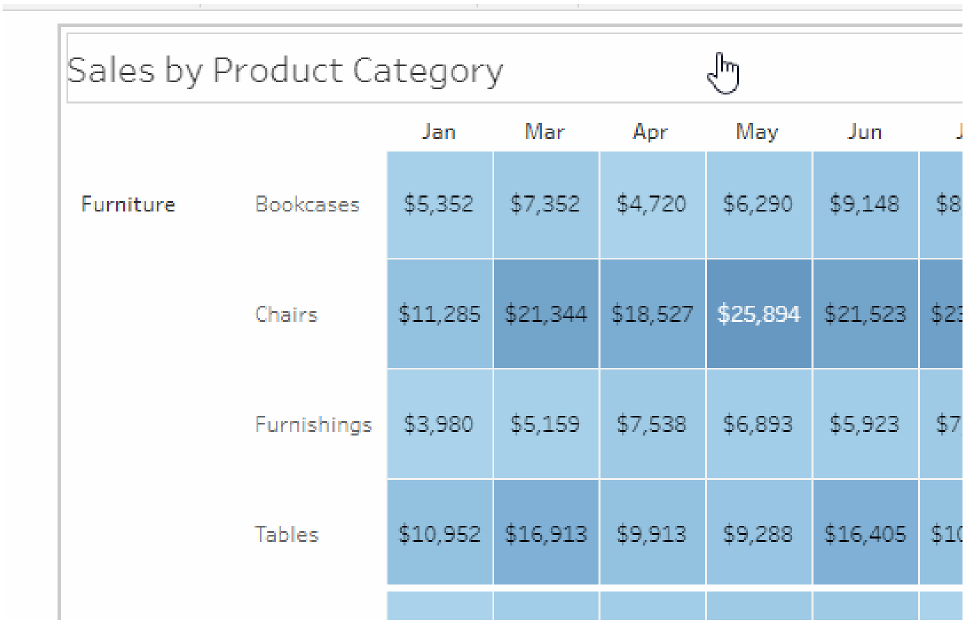
Add an object

From the **Objects** section at left, and drag an item to the dashboard on the right:



Set options for objects

Click the object container to select it. Then click the arrow in the upper corner to open the shortcut menu. (The menu options vary depending on the object.)



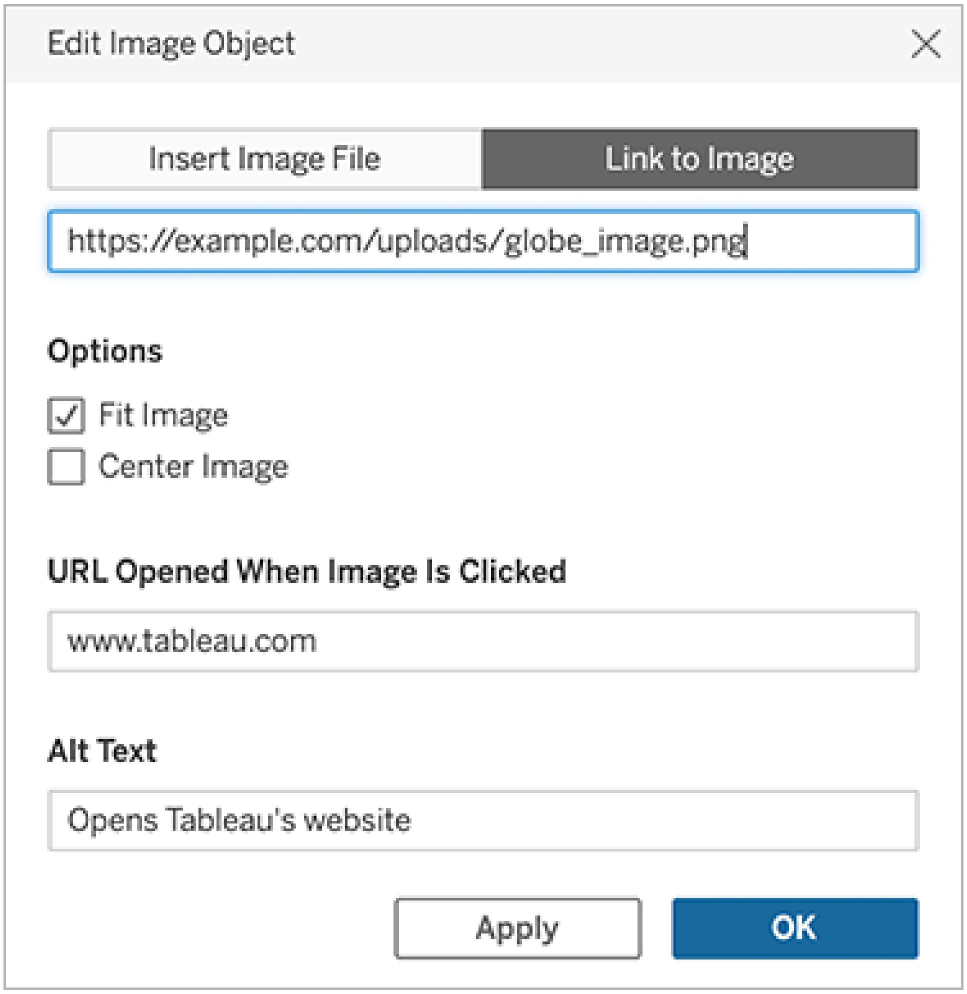
Detailed options for Image objects

With the Image object, you can either insert image files into dashboards or link to images posted on the web. In either case, you can specify a URL the image opens when clicked, adding interactivity to your dashboard.

> **Note:** URLs for web-based images require the HTTPS prefix for improved security. For image URLs with other prefixes, use the Web Page object.

1. From the Objects section at left, drag an Image object to your dashboard at right. Or, on an existing Image object in a dashboard, click the pop-up menu in the upper corner, and choose **Edit Image**.
2. Click either **Insert Image File** to embed an image file into the workbook or **Link to Image** to link to a web-based image.

   Consider linking to a web-based image when:
   - The image is very large and your dashboard audience will view it in a browser. (Unlike web-based images, inserted images must be downloaded every time a sheet opens, slowing performance.)
   - The image is an animated GIF file. (Inserted images don't support animated GIFs.)



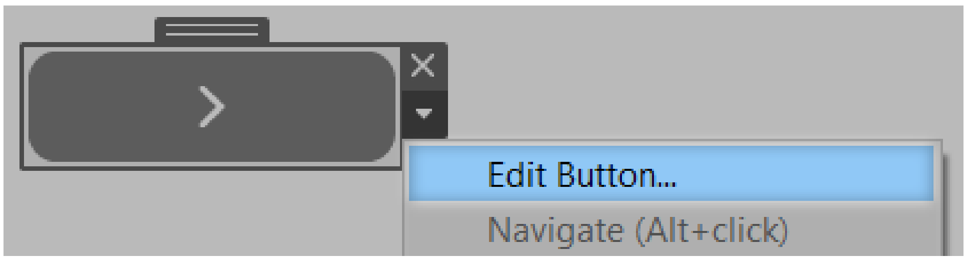
3. If you're inserting an image, click **Choose** to select the file. If you're linking to an image, enter its web URL.
4. Set remaining image fitting, URL linking, and alt text options. (Alt text describes the image in screen-reading applications for improved accessibility.)

Detailed options for Navigation and Download objects

Navigation and Download objects have several unique options that help you visually indicate a navigation destination or file format.

A navigation button using text for the button style

1. In the upper corner of the object, click the object menu, and choose **Edit Button**.



2. Do one of the following:
    - From the **Navigate to** menu, choose a sheet outside the current dashboard.
    - From the **Export to** menu, choose a file format.
3. Choose image or text for **Button Style**, specify the image or text you want to appear, and then set related formatting options.
4. For **Tooltip text**, add explanatory text that appears when viewers hover over the button. This text is optional and typically best used with image buttons. (For example, you might enter "Open Sales viz" to clarify the destination for a navigation button that appears as a miniature sales chart.)

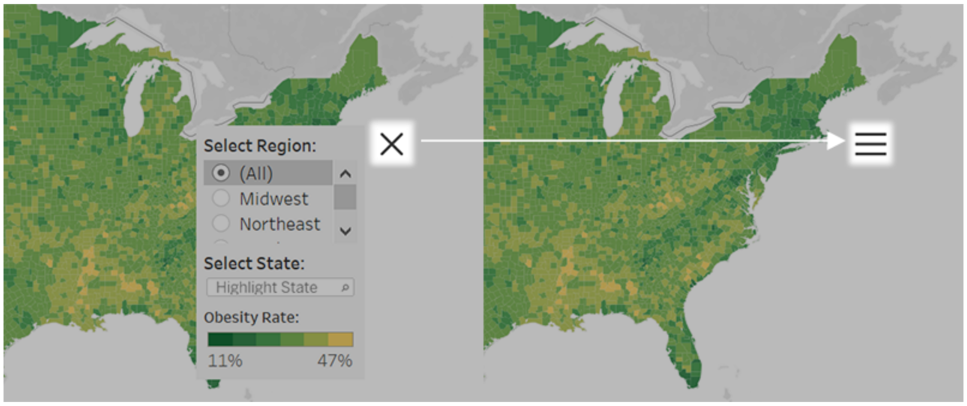
Note: When viewing a published dashboard, simply clicking a button navigates or exports. When authoring a dashboard, however, you need to Alt-click (Windows) or Option-click (macOS).

Show and hide objects by clicking a button

Show/Hide buttons let dashboard viewers toggle the visibility of dashboard objects, revealing them only when necessary.

How hidden objects affect layouts

When a floating object is hidden, it simply reveals any objects beneath it. Show/Hide buttons can be particularly helpful when you want to temporarily hide a floating group of filters to reveal more of a visualization.

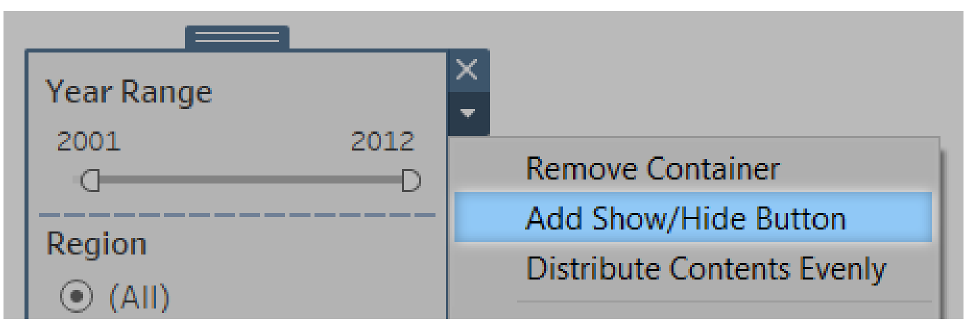


When a tiled object is hidden, the results depend on the object's level in the layout hierarchy.

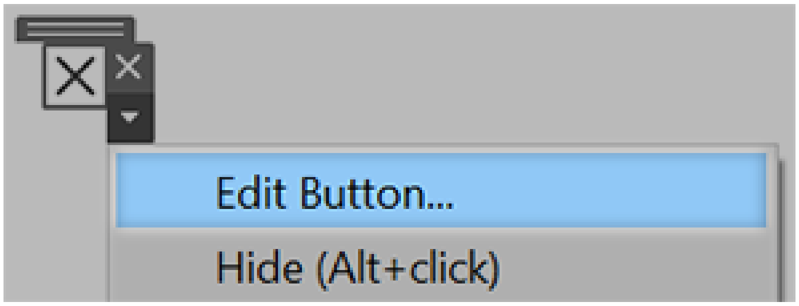
- In most cases, you'll want to place objects you plan to hide in a Horizontal or Vertical layout container, because hidden objects will have their space filled in by other objects in the container.
- By contrast, in the Tiled layout container at the very top of the layout hierarchy, a hidden object leaves blank space behind.

Add and configure a Show/Hide button

1. Select a dashboard object.
2. From the pop-up menu in the upper corner of the object, select **Add Show/Hide Button**.



3. From the button menu, choose **Edit Button**.



4. Set these options:
   - **Dashboard Item to Show/Hide** specifies the target object. (An object can be the target of only one Show/Hide button at a time. Choose **None** if you want to target the object with another Show/Hide button.)
   - **Button Style** specifies whether image or text displays for the button.
   - **Button Appearance** specifies how the button looks when the item is both shown and hidden. Click **Item Shown** and **Item Hidden** to choose different images or text for each state.
   - **Tooltip text** provides explanatory text that appears when viewers hover over the button. (For example, you might enter "Show or hide filters" for a container with filter menus.)
5. If necessary, drag the button to a different location or resize it to better fit your layout.

Note: When viewing a published dashboard, simply clicking a Show/Hide button toggles object visibility. When authoring a dashboard, however, you need to Alt-click (Windows) or Option-click (macOS).

Security for Web Page objects

If you include Web Page objects in your dashboard, you can optimize security by doing the following.

Use HTTPS protocol when possible

As a best practice, use HTTPS (https://) in your URLs. This ensures that the connection from your dashboard to the web page is encrypted. Also, if Tableau Server is running HTTPS and you use HTTP in the URL, your users' browsers won't be able to display the web page that the URL points to. If you don't specify a protocol, HTTP will be assumed.

Security options for Web Page objects (Tableau Desktop only)

Choose **Help** > **Settings and Performance** > **Set Dashboard Web View Security**, and set the options below. (In some organizations, these options are controlled across all machines by a Tableau administrator.)

> **Note:** Any changes you make to these security options apply to both existing and newly created web page objects.

- **Enable JavaScript** Allows JavaScript support in the web view. Clearing this option may cause some web pages that require JavaScript to function improperly in the dashboard.
- **Enable Plug-ins** Enables any plug-ins the web page uses, such as an Adobe Flash or QuickTime player.
- **Block Pop-ups** When selected, blocks pop-ups.
- **Enable URL Hover Actions** Allows URL hover actions. For more information, see **URL Actions** on page 1509.
- **Enable Web Page Objects and Web Images** Allows the display of target URLs in Web Page and Image objects. If you deselect this option, Web Page objects and Image objects that link to the web remain in the dashboard but show no content.

Dashboard Starters for Cloud-based Data

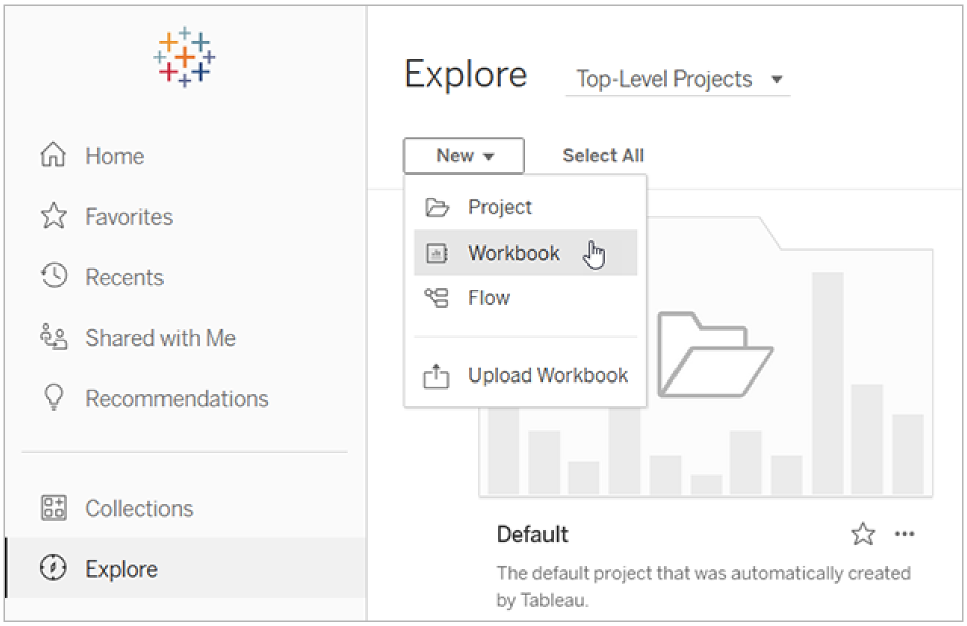
Dashboard Starters help you quickly author and analyze data from cloud-based systems like Salesforce, ServiceNow ITSM, LinkedIn Sales Navigator, Marketo, and Oracle Eloqua.

Simply create a new workbook and choose from several beautiful, informative designs that are tailor-made for key business metrics.

Create a Dashboard Starter

It's easy to create a Dashboard Starter. When the process is complete, you can edit the resulting workbook just like any other, quickly customizing it for your needs.

1. In Tableau Online, open the Explore page, then click the **New** button and select **Workbook**.



> **Note**: Site administrators can also access Dashboard Starters from the Home page, under **Dashboard Starters**.

2. At the top of the Connect to Data window, click **Dashboard Starter**.
3. From the list of pre-built designs, find an option that reflects the data source and business metrics you need, and click **Use Dashboard**.
4. To quickly see how a workbook looks with sample data, click **Continue without signing in**. Or click **Continue** to create a workbook with your data.
5. Specify a name and containing project for the workbook.
6. If you chose to create a workbook with your data, connect to your data source. While Tableau prepares an extract of your data, sample data appears so you can explore the

layout.

That's it—you've created a rich, interactive dashboard in seconds!

Change permissions to share Dashboard Starters with colleagues

To avoid exposure of confidential data, workbooks for Dashboard Starters are visible only to authors and administrators by default. To share a Dashboard Starter with your colleagues, follow these simple steps:

1. In Tableau Online, navigate to the workbook for the Dashboard Starter.
2. Select the workbook, click **Actions**, and choose **Permissions**.
3. Give **View** permissions to any user or group you want to see the dashboard. For more information, see Edit Permissions in Tableau Online Help.

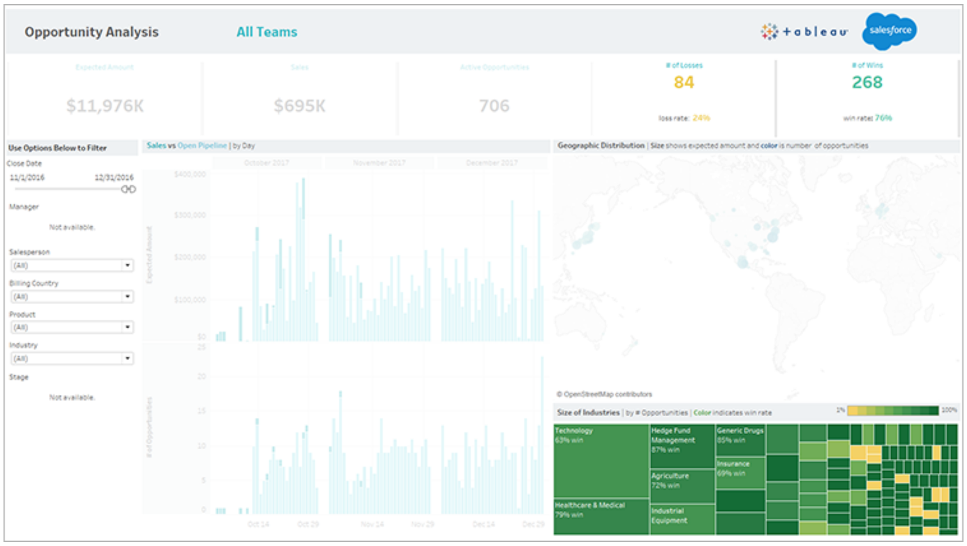
Replace sample data with your data

If you chose to quickly load sample data in a dashboard, you can replace it with your data at any time.

1. In Tableau Online, navigate to the workbook for the Dashboard Starter.
2. On the **Data Sources** tab, select the data source. From the Actions menu, choose **Edit Connection**.
3. For authentication, select **Embedded credentials in the connection**, and either choose an existing user account or add a new one. Then click **Save**.
4. On the **Refresh Schedules** tab, select the schedule. From the Actions menu, choose **Run Now**.

Fix grayed-out views by replacing field names

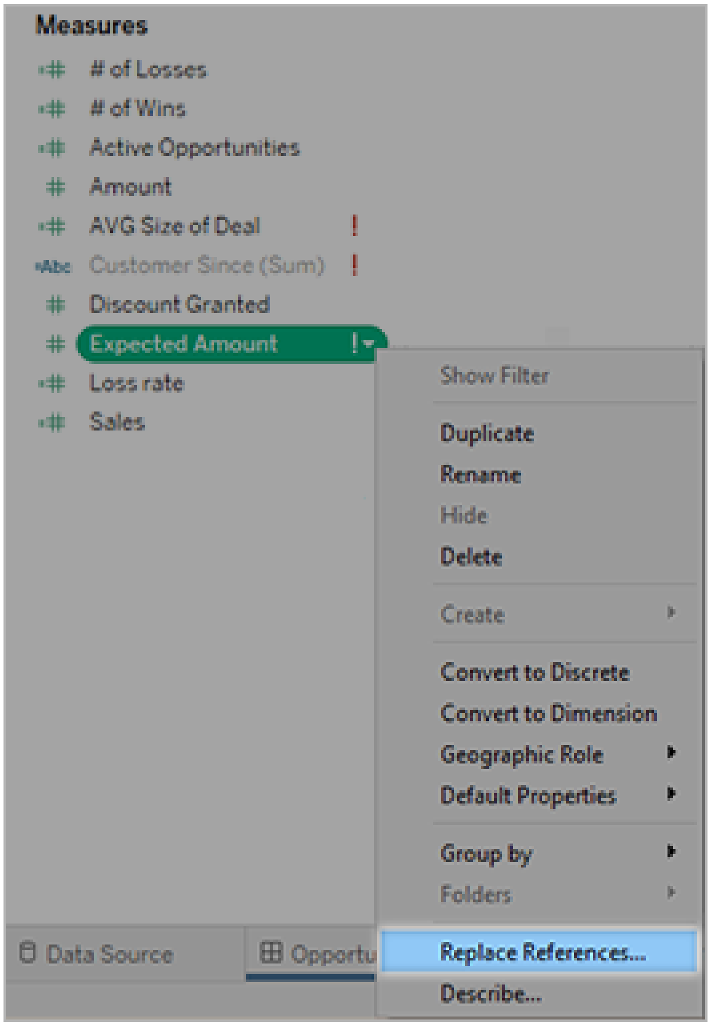
If your organization has customized the data structure for a cloud-based system, you may need to match those changes in Dashboard Starters after your data loads in them. For example, if your organization has renamed the Salesforce “Account” field to “Customer”, you'll need to make a corresponding change in Dashboard Starters to avoid grayed-out views like this:



Fortunately, the fix is pretty straightforward:

1. Above the dashboard, click **Edit**.
2. Navigate directly to the grayed-out sheet.
3. In the Data pane at left, look for red exclamation points (!) next to field names, which indicate that your organization uses different names.
4. Right-click each of those fields, and choose **Replace References**. Then select the

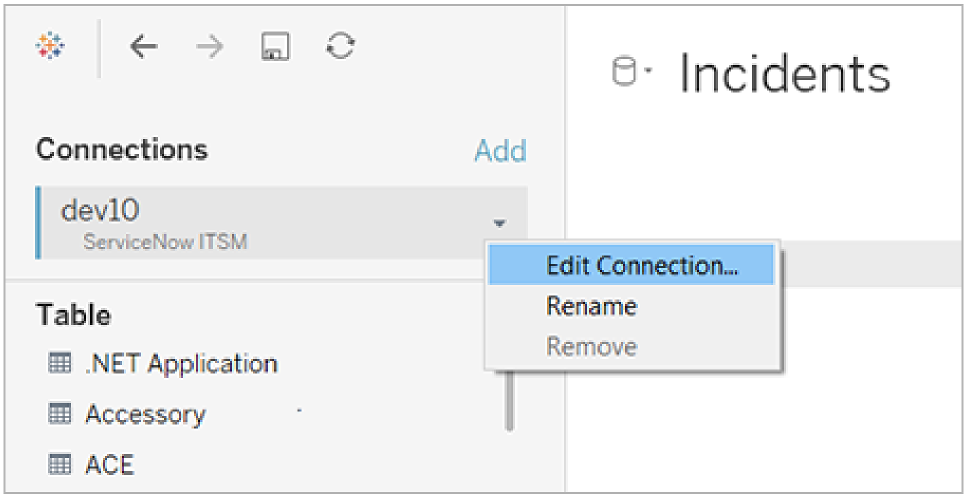
correct field name from the list.



Fix empty dashboards by changing default date ranges

If a dashboard looks completely empty, the likely cause is a default date range that doesn't correspond to the dates in your source data.

1. Download the workbook, and open it in Tableau Desktop.
2. Click the **Data Source** tab.
3. In the upper-left corner, click the arrow next to the data source name, and choose **Edit Connection**. Then sign in.



4. Specify a date range that reflects the dates in your data, and click **Connect**.

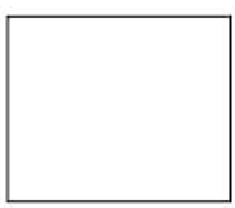
5. Choose **Server** > **Publish Data Source** to update extracts of the data on Tableau Online.

Size and Lay Out Your Dashboard

After you create a dashboard, you might need to resize and reorganize it to work better for your users.

Control overall dashboard size

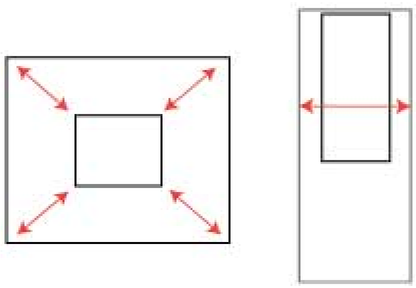
Dashboard size options



Fixed size (default): The dashboard remains the same size, regardless of the size of the window used to display it. If the dashboard is larger than the window, it becomes scrollable. You can pick from a preset size, such as Desktop Browser (the default), Small Blog, and iPad.

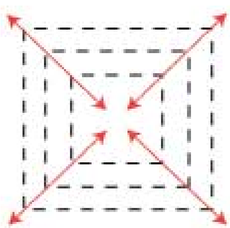
Fixed size dashboards let you specify the exact location and position of objects, which can be useful if there are floating objects. Select this setting if you know the precise size at which your dashboard will be displayed.

Published dashboards that use a fixed size can load faster because they're more likely to use a cached version on the server. (Dashboards with variable sizes need to be freshly rendered for every browser request.) For other performance tips, see **Optimize Workbook Performance** on page 2993.



Range: The dashboard scales between minimum and maximum sizes that you specify. If the window used to display the dashboard is smaller than the minimum size, scroll bars are displayed. If it's larger than the maximum size, white space is displayed.

Use this setting when you're designing for two different display sizes that need the same content and have similar shapes—such as small- and medium-sized browser windows. Range also works well for mobile dashboards with vertical layouts, where the width may change to account for different mobile device widths, but the height is fixed to allow for vertical scrolling.



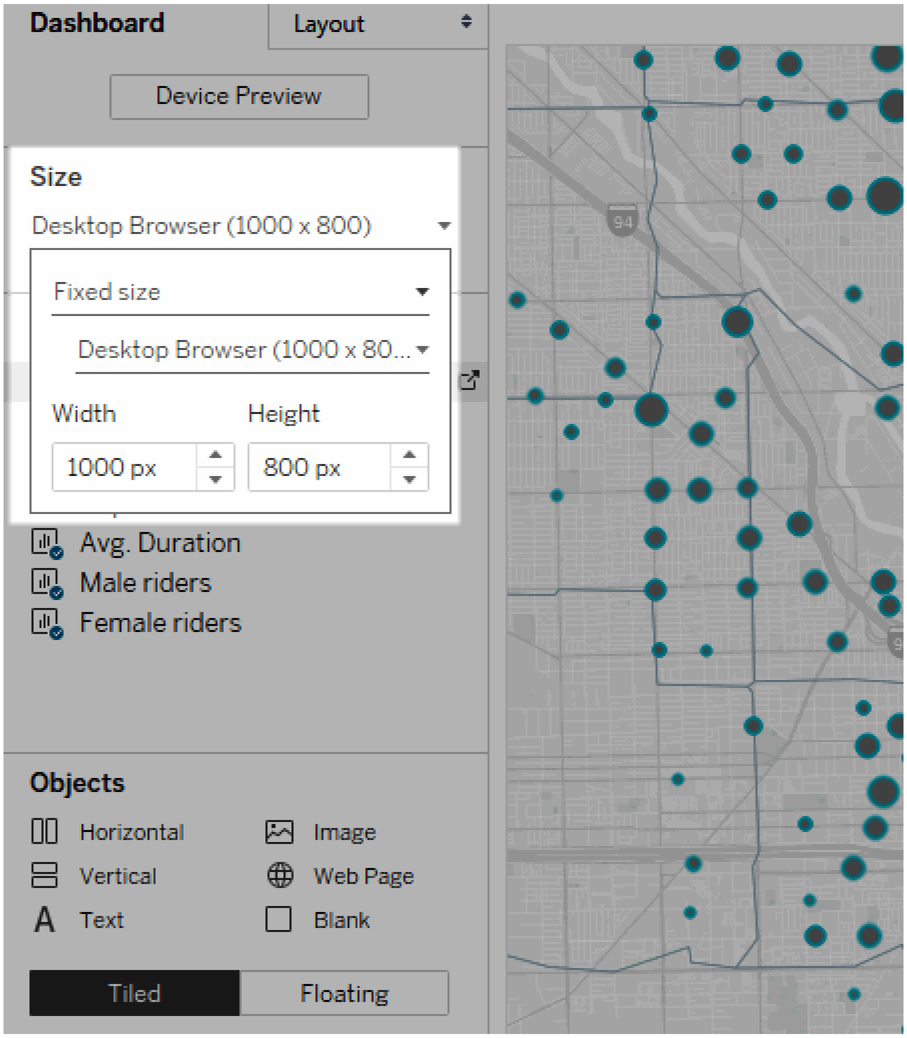
Automatic: The dashboard automatically resizes to fill the window used to display it.

Use this setting if you want Tableau to take care of any resizing. For best results, use a tiled dashboard layout.

Tip: If you use Tableau Desktop, see **Create Dashboard Layouts for Different Device Types** on page 2801 to create unique layouts optimized for desktop computers, tablets, and phones. In addition to adapting to different screen sizes, each device layout can contain different items.

Set overall dashboard size

- Under **Size** on the Dashboard pane, select the dashboard's dimensions (such as **Desktop Browser**) or sizing behavior (for example, **Automatic**).



Group items using layout containers

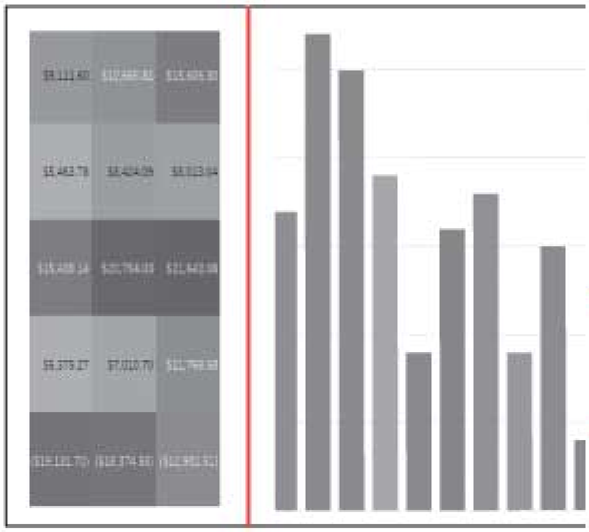
Layout containers let you group related dashboard items together so you can quickly position them. As you change the size and placement of items inside a container, other container items automatically adjust.

Layout container types

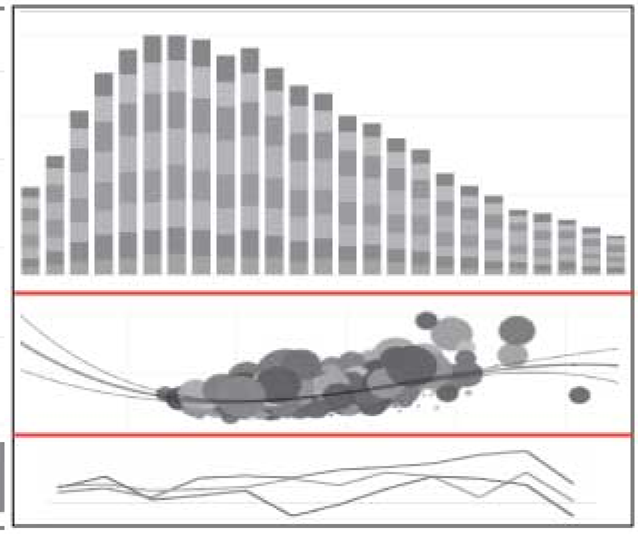
A horizontal layout container resizes the width of the views and objects it contains; a vertical layout container adjusts height.

Horizontal layout container	Vertical layout container

The two views below are arranged in a horizontal layout container.

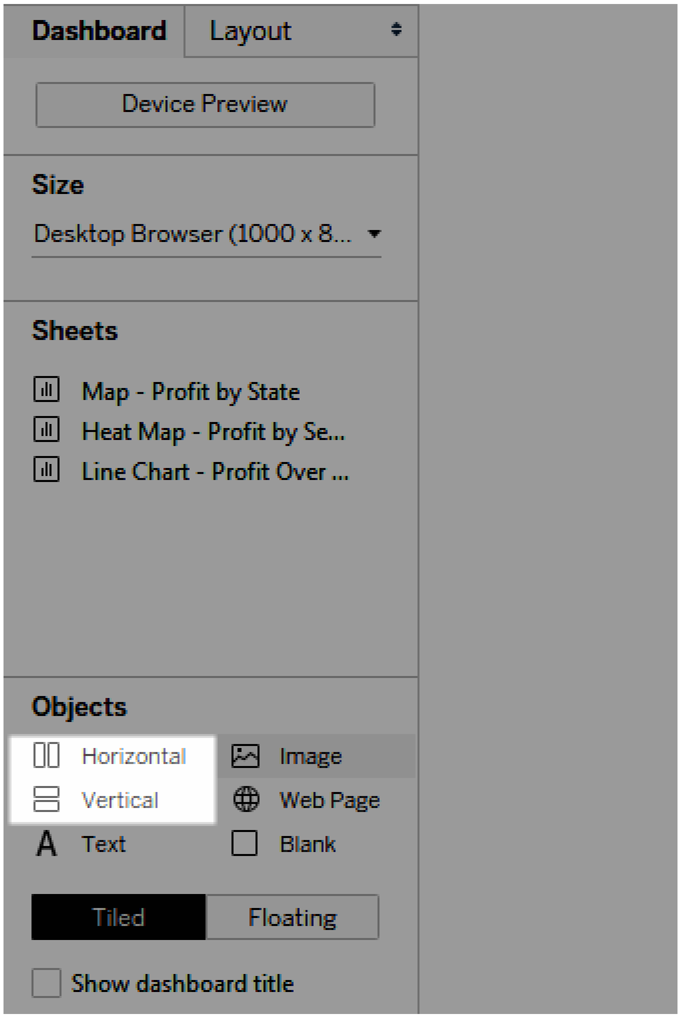


The three views below are stacked in a vertical layout container.

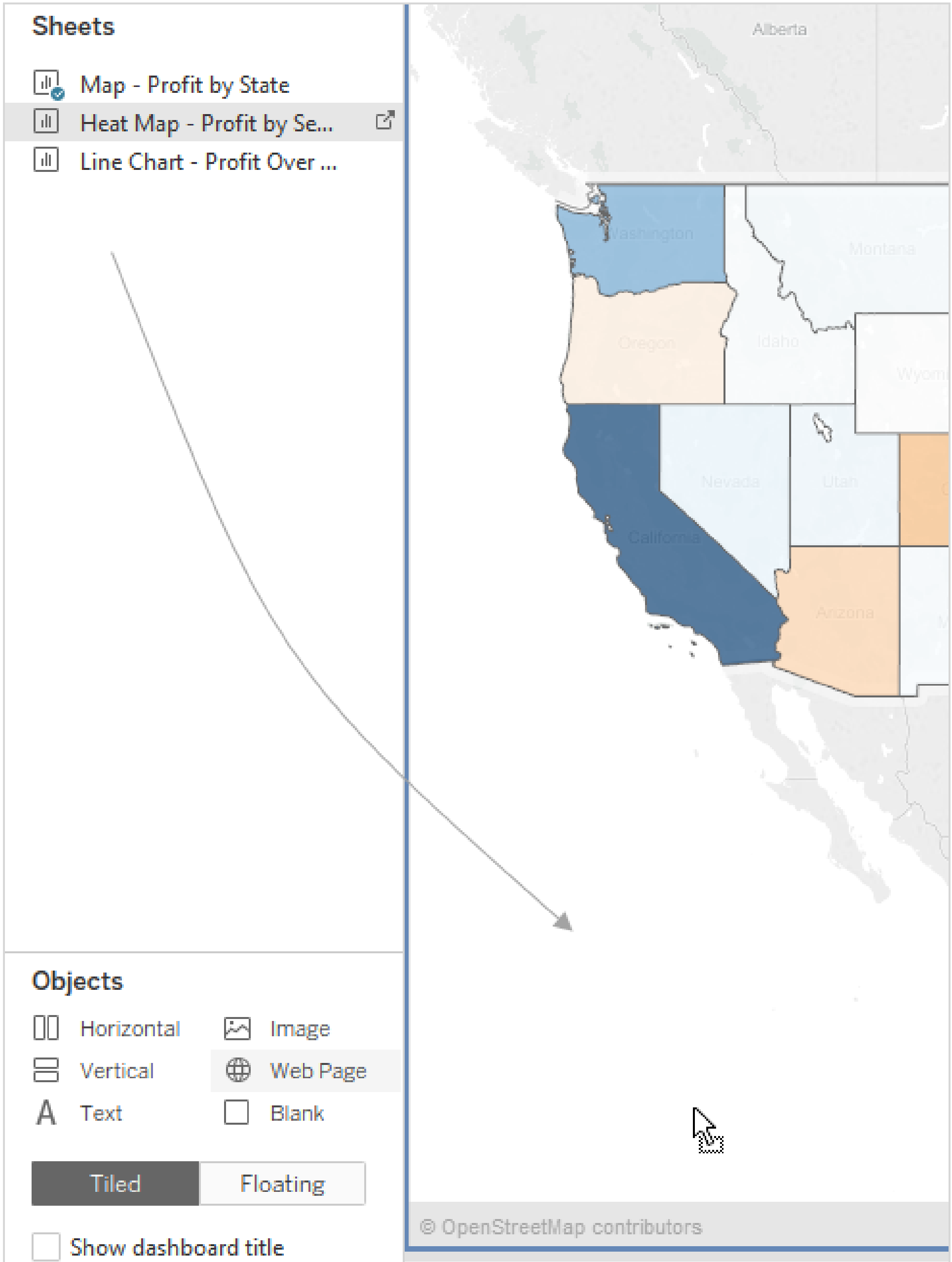


Add a layout container

1. Under **Objects** on the Dashboard pane, select **Horizontal** or **Vertical**.
2. Drag the container to the dashboard.

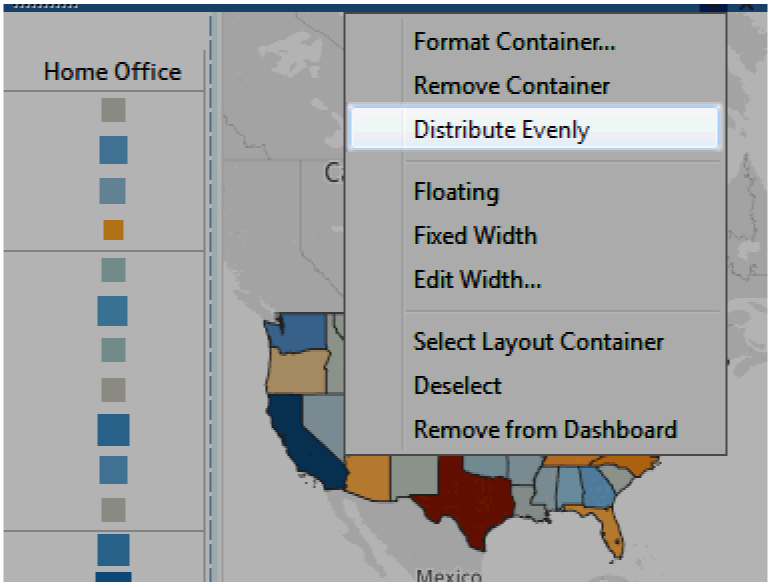


3. Add views and objects to the layout container.



Evenly distribute a layout container's items

1. Select the layout container. If you have trouble doing this, select an individual item within the container and choose **Select Container** from its shortcut menu.
2. With the layout container selected, choose **Distribute Evenly** from its shortcut menu:

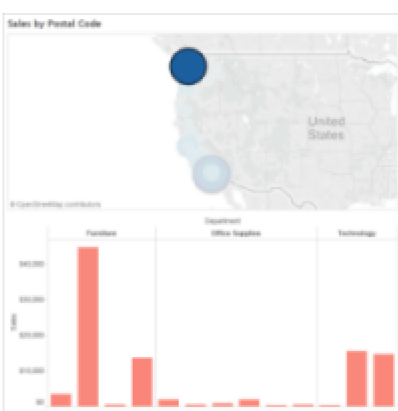
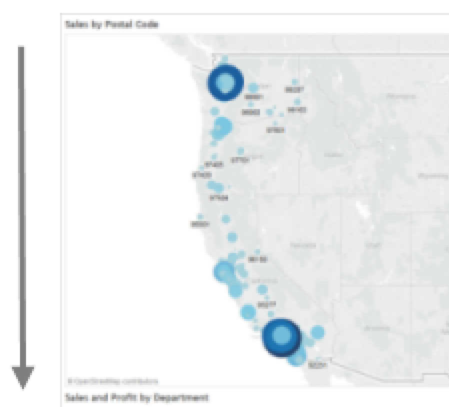


Items that are already within the layout container arrange themselves evenly; any items you add will do the same.

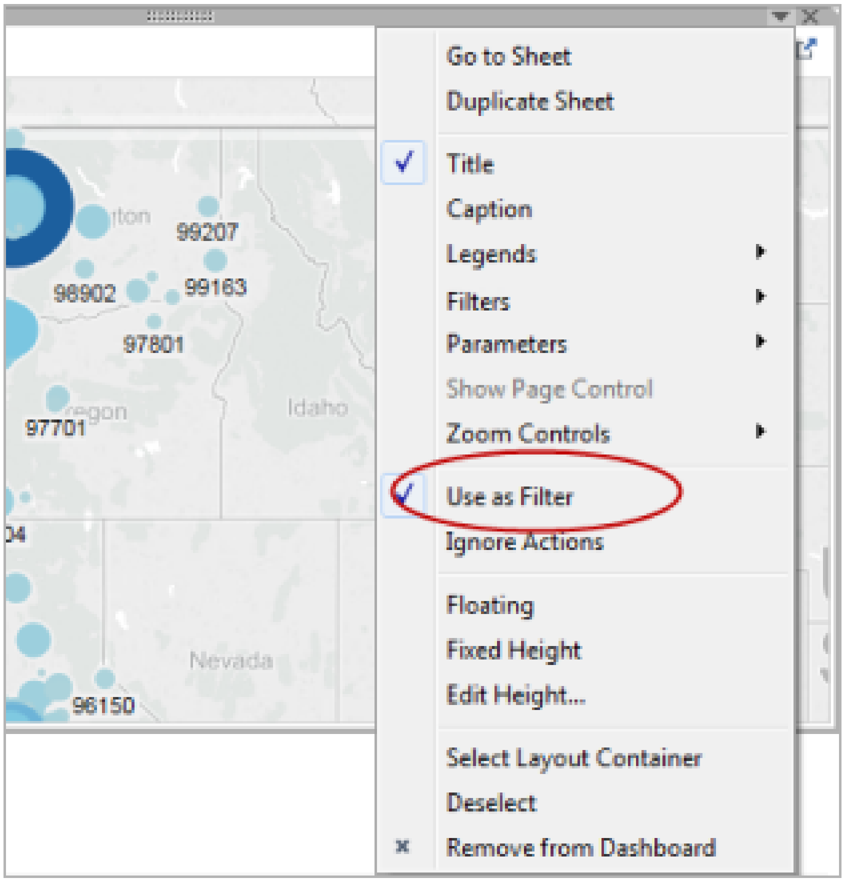
Automatically resize sheets in layout containers

If you add multiple sheets with related data to a layout container, whenever marks are selected in one sheet, you can automatically resize related sheets.

In this example, when a mark is selected in the map, the bar chart updates to display profit and sales for that mark; when no marks are selected in the map, it automatically expands to fill the layout container.

With mark selected	**Without a selection**
	

1. Add multiple sheets with related data to a layout container.
2. From the drop-down menu of the sheet you want to expand, choose **Use as Filter**.



3. Choose **Dashboard** > **Actions**, and double-click the generated filter you just created.
4. In the **Target Sheets** section of the Edit Filter Action dialog box, select the other sheets in the layout container.
5. To control how target sheets resize when no marks are selected in the source sheet, select one of the following:
   - **Show all values** returns target sheets to their original size, showing all data.
   - **Exclude all values** collapses target sheets under their titles, hiding all data.

Remove a layout container to independently edit items it contains

1. Select the container either on the dashboard or in the **Item hierarchy** area of the Layout pane.
2. From the drop-down menu at the top of the container, select **Remove Container**.

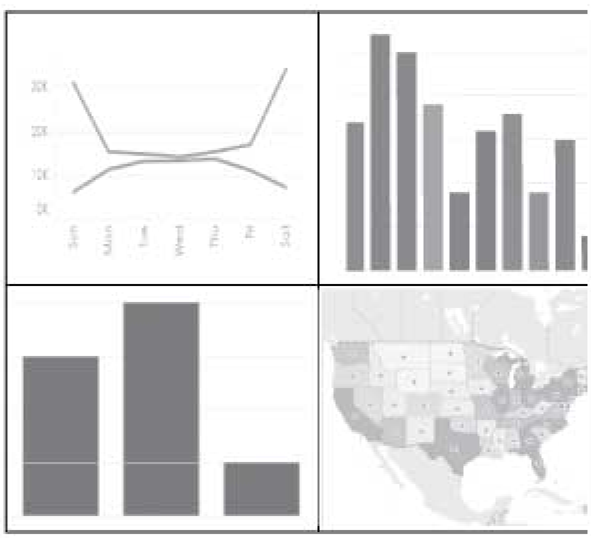
Tile or float dashboard items

Tiled vs. floating layouts

Each object, layout container, and view that you place on a dashboard is either tiled (the default) or floating.

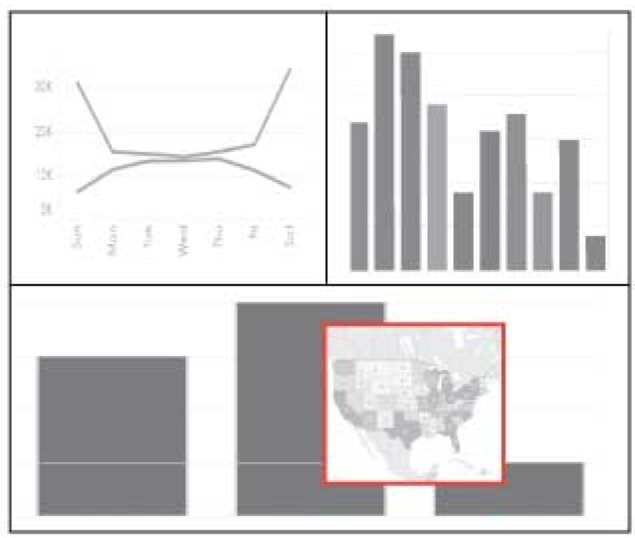
Tiled layout

Tiled items don't overlap; they become part of a single-layer grid that resizes based on the overall dashboard size.



Floating layout

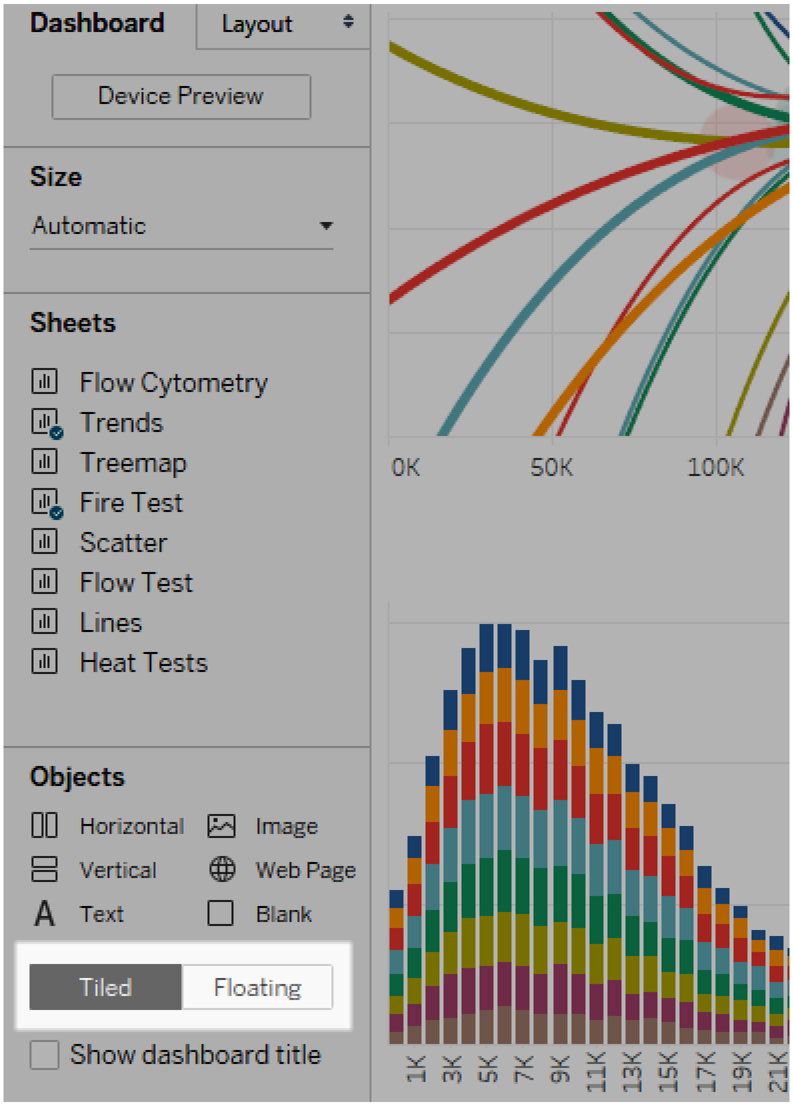
Floating items can be layered over other objects. In the example below, a map floats over tiled views.



For best results, give floating objects and views a fixed size and position.

Float or tile a new item

1. Under **Objects** in the Dashboard pane, click the layout option you want to use: **Floating** or **Tiled**.



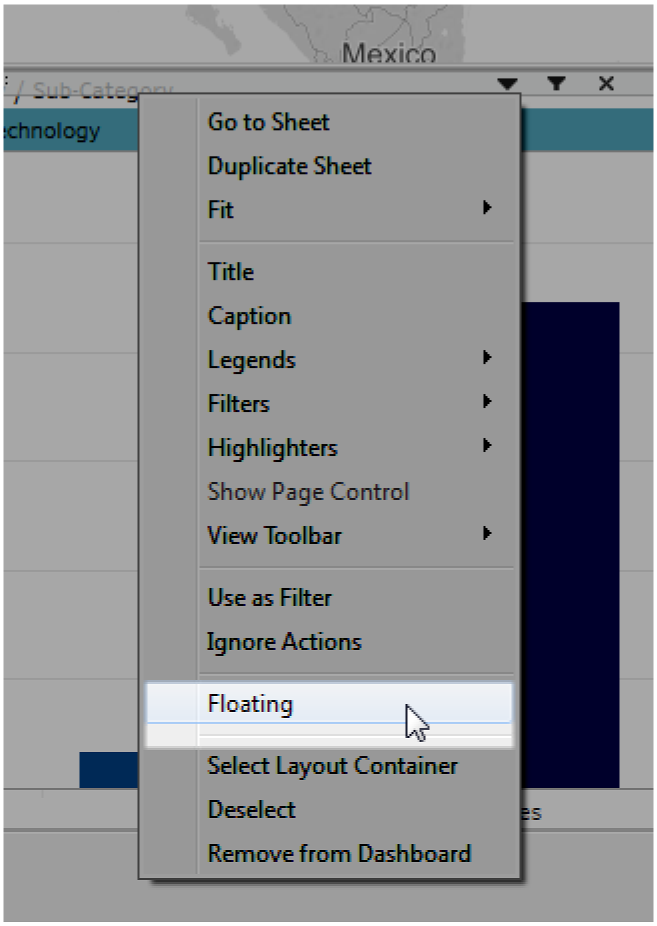
2. Drag the view or object onto the dashboard on the right.

Tip: You can also assign a floating layout to an item by holding down **Shift** on your keyboard as you drag it onto the dashboard.

Switch an existing item from tiled to floating

1. Select the item in the dashboard.

2. In the item's shortcut menu, select **Floating**:

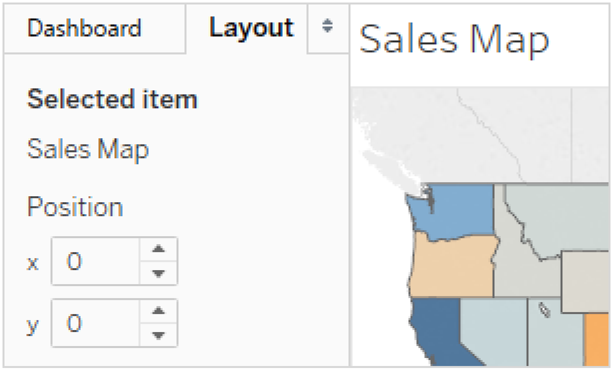


Size, position, reorder, and rename individual dashboard items

Set an item's size and position

If you want a high degree of control over the placement of every item in a dashboard, choose **Fixed size** or **Range** for overall dashboard sizing, and then set each item's size and position using the Layout pane.

1. Select the dashboard item you want to position and size. In the Layout pane, the item's name appears under **Selected item**.



> **Tip:** To quickly toggle between the Dashboard and Layout panes, press the T key.

2. Define the item's position **x** and **y** position in pixels as an offset from the top left corner of the dashboard. For example, to place an object in the top left corner of the dashboard, specify x = 0 and y = 0.

   The values you enter can be positive or negative but must be whole numbers.

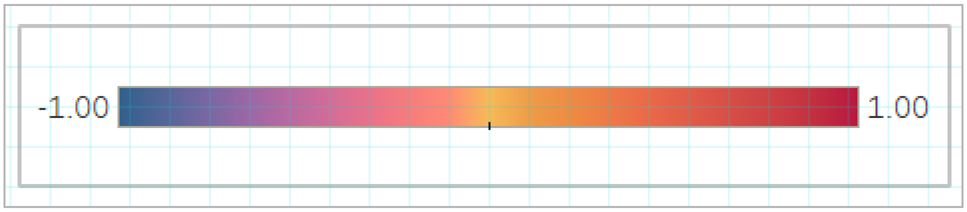
3. To resize an item, click and drag it in the dashboard. You can also use the settings under **Size**.

4. To hide the title of the item you have selected, clear the **Show title** check box. (The similar **Show dashboard title** check box on the Dashboard pane controls the overall dashboard title.)

> **Tip:** To quickly position floating items, press arrow keys to move 1 pixel, or Shift+arrow keys to move 10 pixels. To quickly resize items, add Alt (Windows) or Option (macOS) to these shortcuts.

Align items with a grid

To present a visually consistent design, arrange and size dashboard items over a grid.

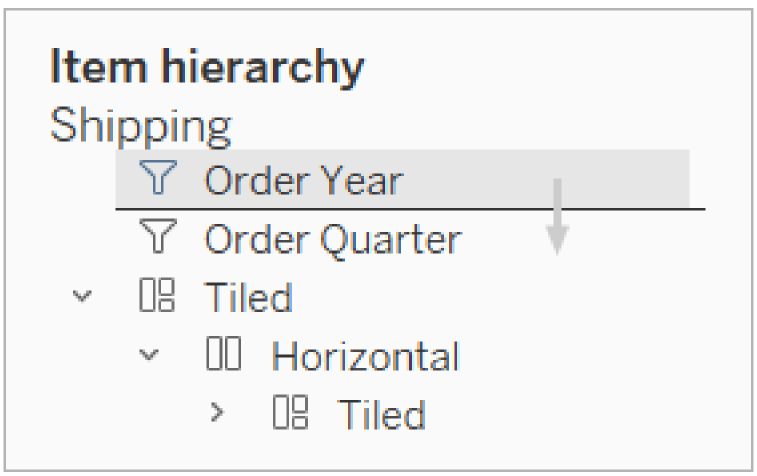


1. Choose **Dashboard** > **Show Grid**.
2. To change the grid size, choose **Dashboard** > **Grid Options**.

Tip: To quickly toggle the grid on and off, press the G key.

Reorder floating objects

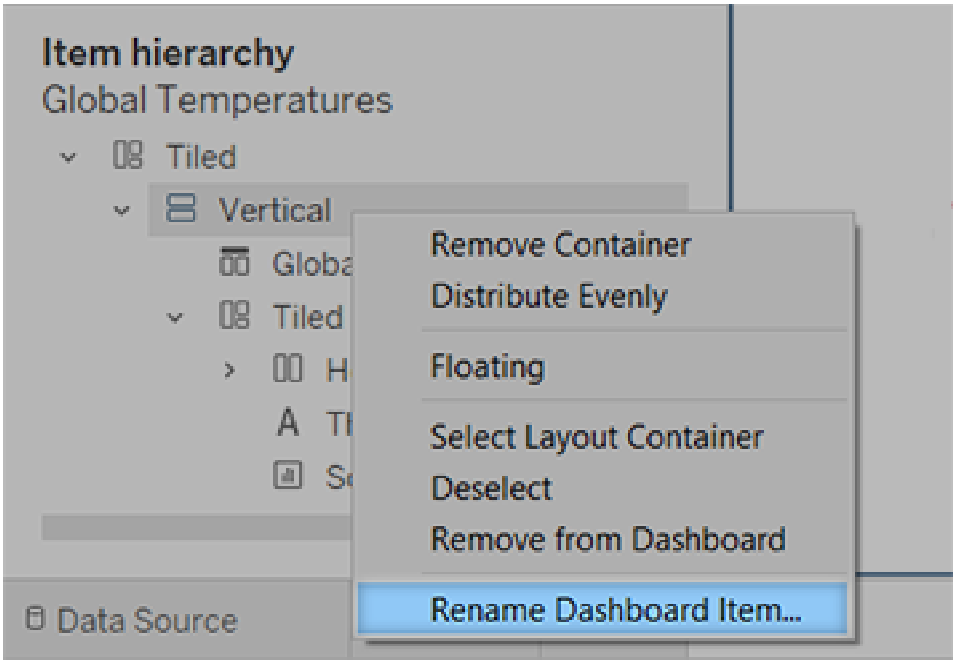
The **Item hierarchy** area on the Layout pane shows the relationships between objects on your dashboard. Drag floating items in the hierarchy to change how they layer over each other. Items at the top of the list appear in the front, while items at the bottom appear in back.



Rename items

To better distinguish between dashboard items, you can rename them.

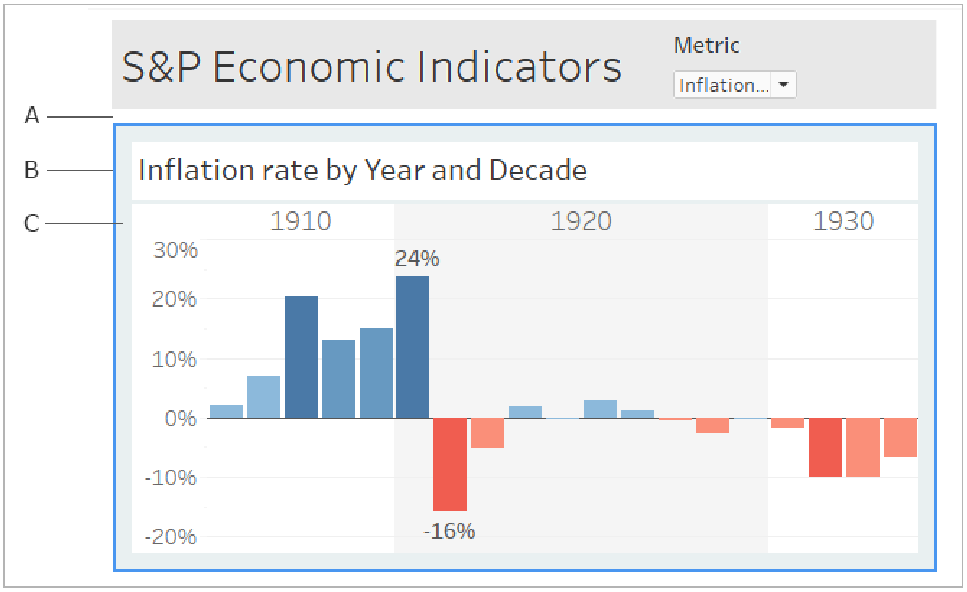
In the **Item hierarchy** area of the Layout pane, right-click an item, and choose **Rename Dashboard Item**. On the dashboard itself, this command is also available from the drop-down menu for each object.



Add padding, borders, and background colors around items

Padding lets you precisely space items on dashboard, while borders and background colors let you visually highlight them. Inner padding sets the spacing between item contents and the perimeter of the border and background color; outer padding provides additional spacing beyond the border and background color.

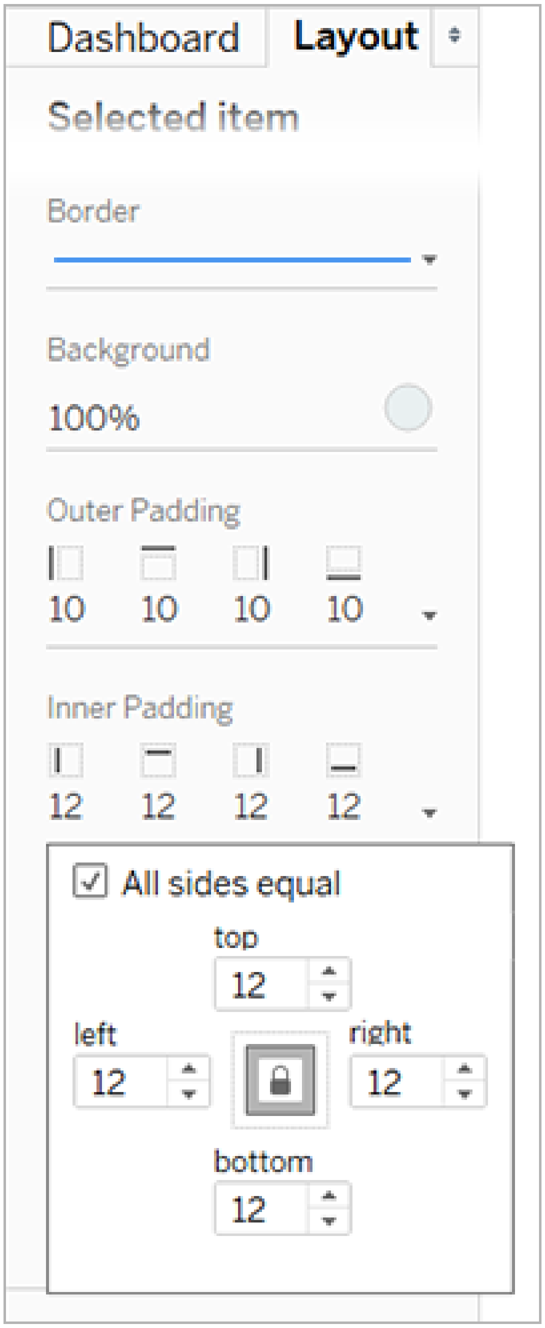
> **Note:** If you can't change the border or background color for a particular dashboard item, change its formatting at the worksheet level.



A. Outer padding B. Blue border C. Inner padding with light blue background

1. Select an individual item, or your entire dashboard.
2. On the **Layout** tab at left, specify border style and color, background color and opacity,

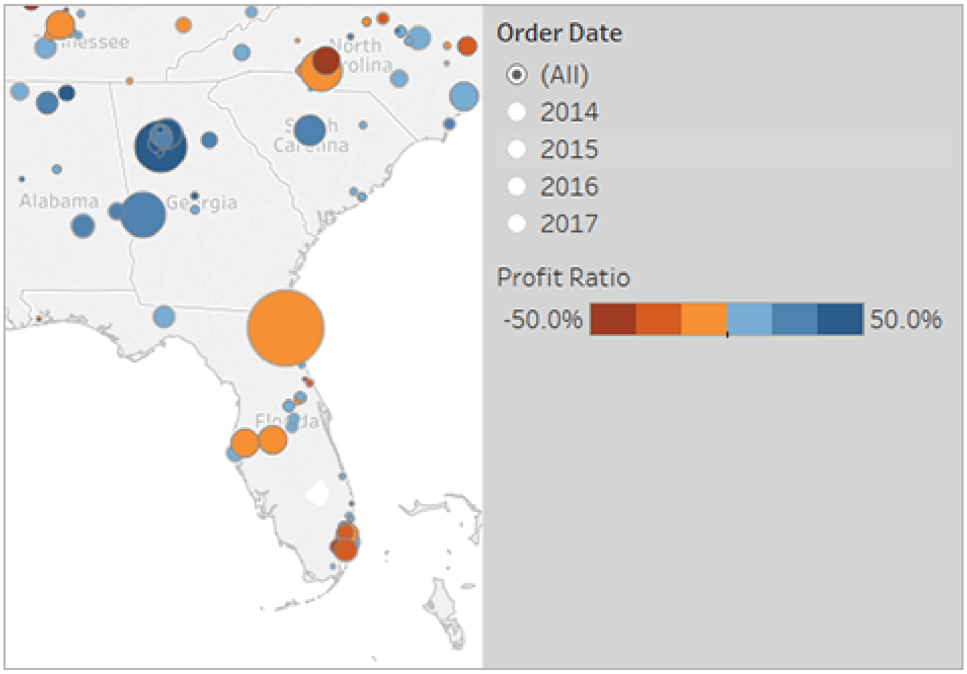
or padding size in pixels.



Here are some handy dashboard spacing tips:

- To precisely align one dashboard item with another, deselect **All sides equal**, and adjust padding for only one side.

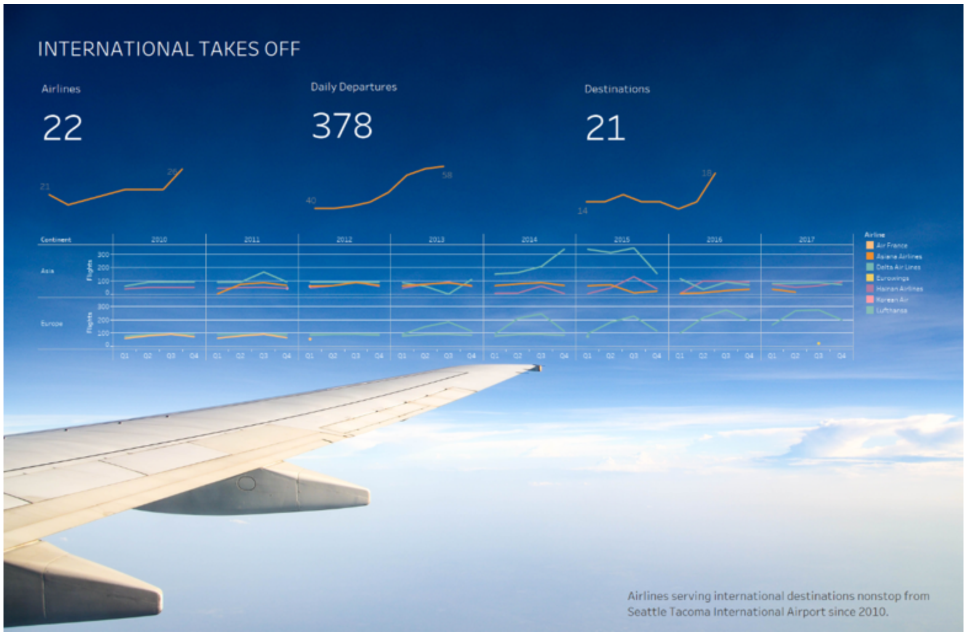
- To create seamless designs, specify zero outer padding for adjoining items.



Visually integrate elements with transparency

Transparent elements create a seamless visual look for your dashboards, revealing underlying objects and images.

> **Tip:** To create a background image that shows through floating transparent elements, add an image object as the only tiled item. Fixed-size dashboards ensure that the floating elements maintain a consistent visual relationship with the image, regardless of screen size.



Make worksheet backgrounds transparent

1. In your dashboard, select the sheet.
2. Choose **Format** > **Shading**, click the **Worksheet** menu, and choose **None** for the background color.

   If the sheet appears opaque, change the background color to None for the underlying dashboard, object, or layout container. You can quickly adjust these items on the Layout tab for the dashboard.
3. To smoothly integrate the transparent sheet with other dashboard items, choose **Format** > **Borders** and **Format** > **Lines**, and either remove borders and lines or change their colors.

Additional steps for transparent maps

1. Choose **Format** > **Shading**, click the **Pane** menu, and choose **None** for the color.
2. Choose **Map** > **Map Layers**, and deselect the opaque **Base** layer. Then, to make transparent maps more distinct when zoomed out, select the **Coastline** layer.

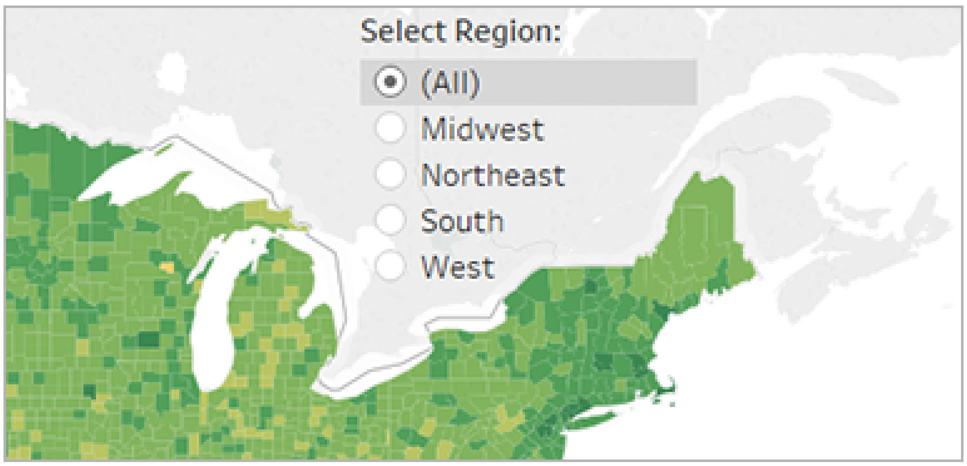
   Depending on the map style, you also may need to experiment with turning other layers on or off.

Make a sheet partially transparent

1. In the **Layout** pane for the dashboard, select the sheet.
2. Click the **Background** color, and set the color and opacity.

Float transparent legends, filters, highlighters, and parameters

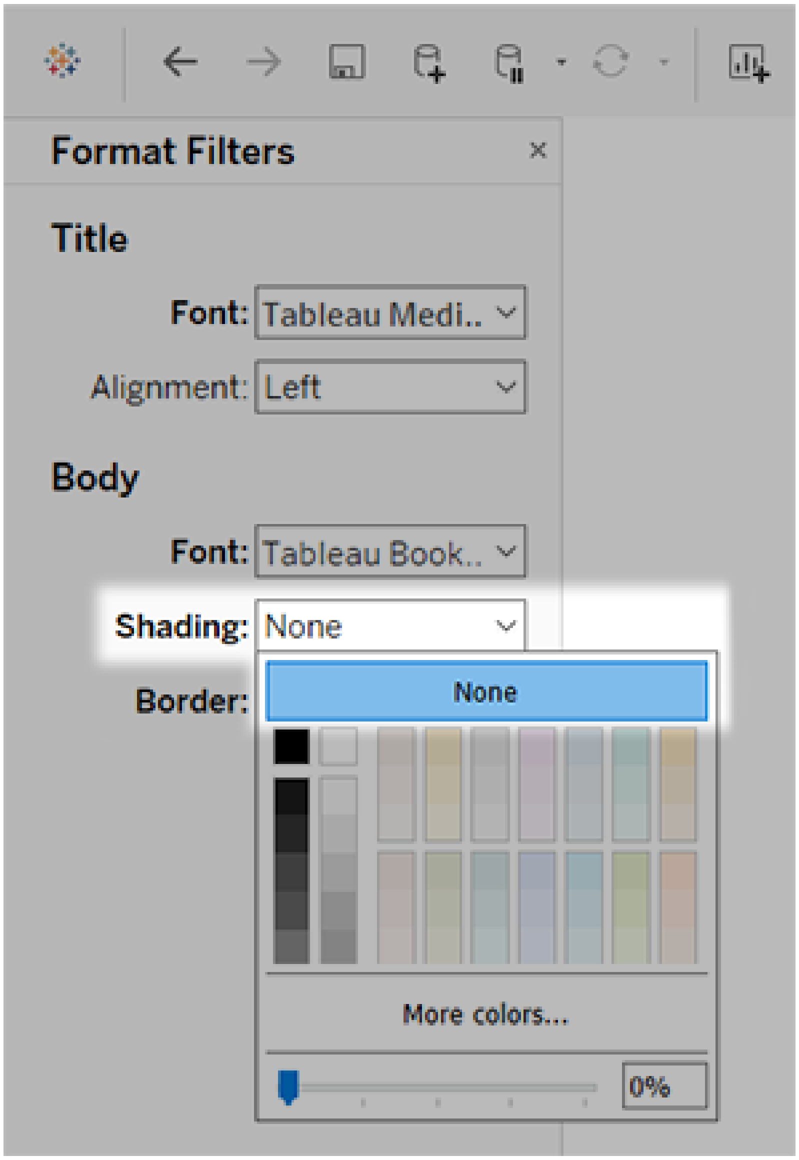
To visually connect filters, parameters, and highlighters to related data, float these items, which are transparent by default. Text always remains fully opaque, maintaining legibility.



If a floating object continues to display a color, check these object- and worksheet-level settings:

- Select the object, and on the **Layout** tab, click the **Background** color, and choose **None**.
- Click the **Format** menu, and then choose **Legends**, **Filters**, **Highlighters**, or

Parameters. Then, in the Format pane at left, choose **Shading** > **None**.



For more information, see **Format at the Worksheet Level** on page 2892.

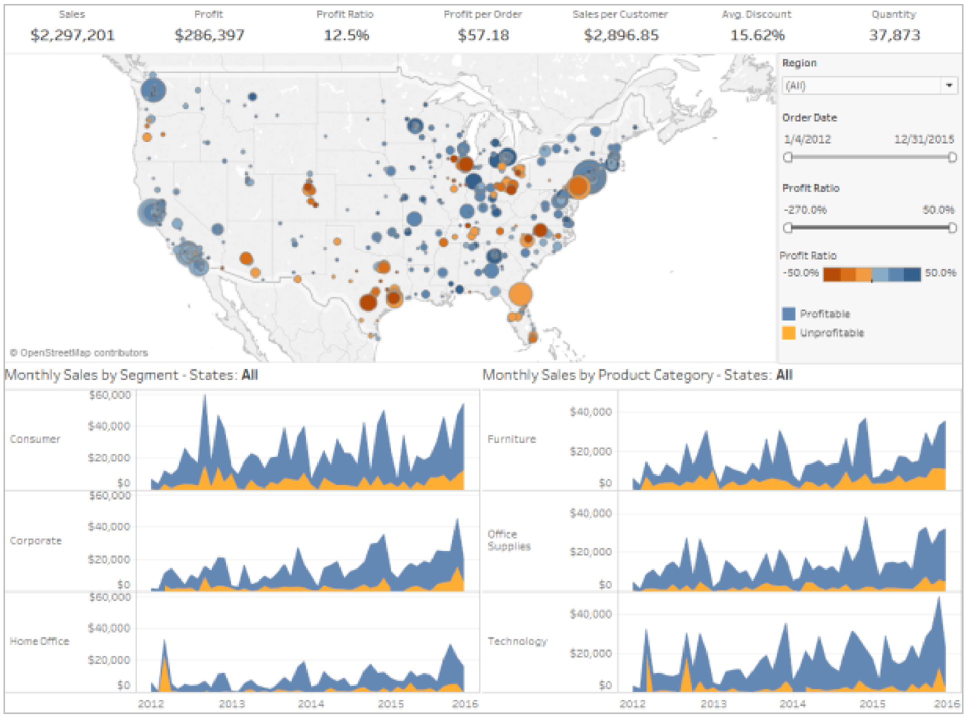
Refine Your Dashboard

After you've created and laid out your dashboard, take a moment to step back and evaluate it. This topic lists some areas to check and refine.

Is everything in the right spot?

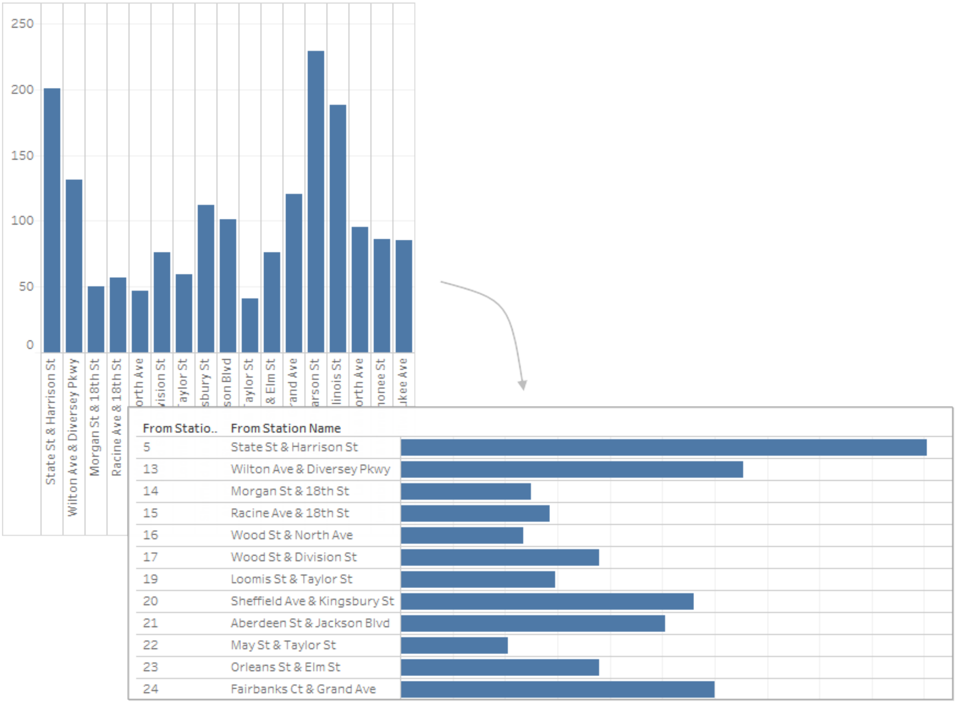
Confirm placement

Make sure that the view with your key finding or main question spans or occupies the upper-left corner of your dashboard. In your views, the most important data should be on the X or Y axis and your less important data should be on color, size, or shape.



Check label orientation and readability

Important labels should be horizontal so that your viewers can easily read them.



Review sizing and layout

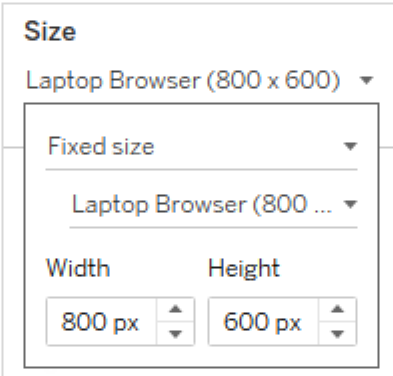
Test your work and adjust sizing

Once you finish your dashboard, it's a best practice to test it and experience it as your users will. For example, will they be using Tableau Reader on a laptop to view your dashboard, will the dashboard be published on Tableau Server, or will it be embedded in a web application and viewed from a tablet?

After you test your dashboard, you'll probably find some things to change—sizing and how and where items are positioned are usually among them.

Tableau dashboards are set to a fixed default size that's intended to work well on a typical desktop. However, when you publish (to the web, in a blog, for a presentation, etc.) you may find yourself more limited.

You can specify the overall size of the dashboard using the settings under **Size** in the **Dashboard** area on the left. Select a new size using the drop-down menu.



- **Automatic** - The dashboard automatically resizes to fill the window it's displayed in.
- **Fixed size** - The dashboard always remains a specific size. If the dashboard is larger than the window the dashboard becomes scrollable.
- **Range** - The dashboard scales between minimum and maximum sizes that you specify, after which scroll bars or white space display.

Customize image display (Tableau Desktop only)

When you add an **Image** object to a dashboard, you can customize how the image displays by selecting an option on the image's menu. For example, you can select whether to **Fit Image**, which scales the image to the size of image object on the dashboard.

By default, any high-resolution images that you use in a dashboard are displayed in standard resolution and will maintain the same size whether they're viewed on standard or Retina displays.

To take advantage of high-resolution images in a dashboard that are intended to be viewed on a Retina display, you can use the @2x naming convention:

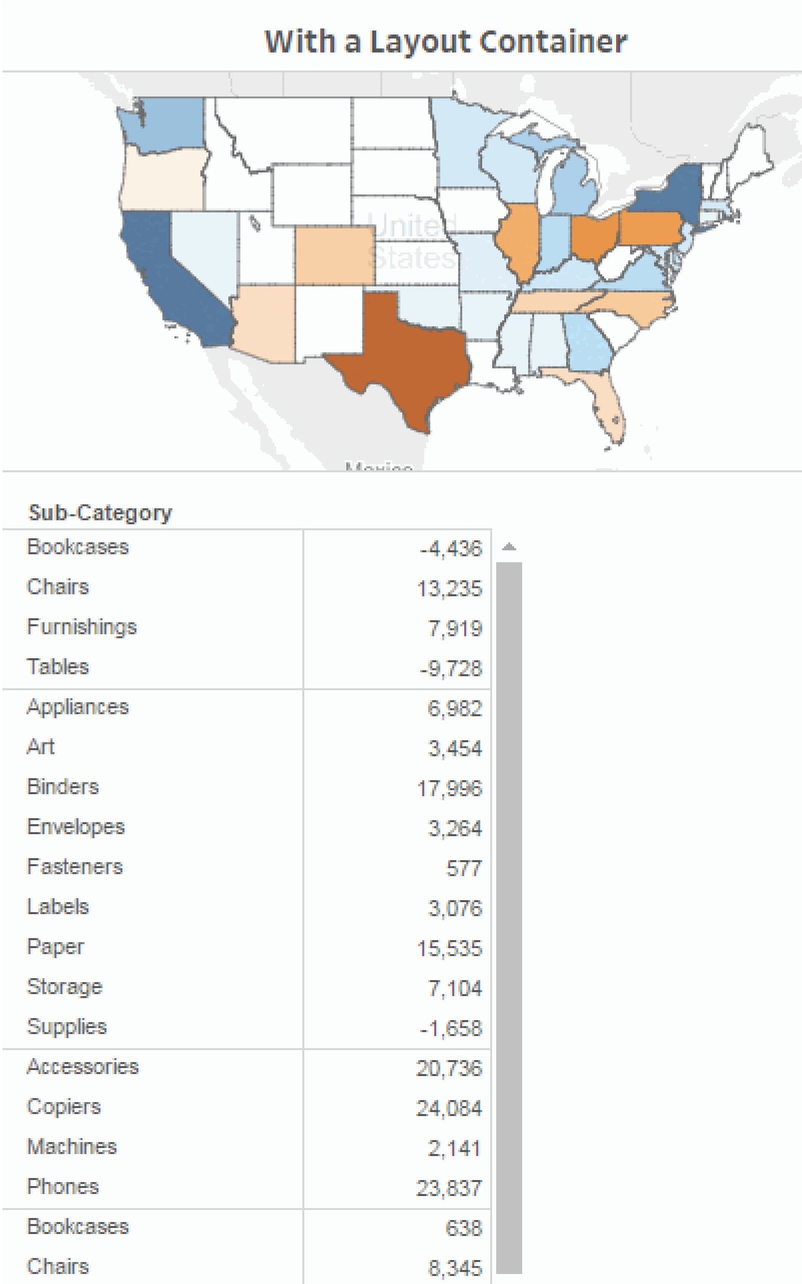
1. Navigate to the high-resolution image on your computer with Retina display.
2. Rename your high-resolution image to include the @2x modifier by using *<image-name>*@2x*<file-extension>* format. For example, logo@2x.png.
3. In your workbook, click and drag an **Image** object onto your dashboard.
4. When you're prompted, select the image you just renamed using the @2x naming convention and then click **Open**.

Note: EMF image file formats are not compatible with Tableau Desktop on the Mac.

Use a layout container

A layout container can create a smoother experience for your users by helping dashboard objects reposition and resize when one view is filtering another.

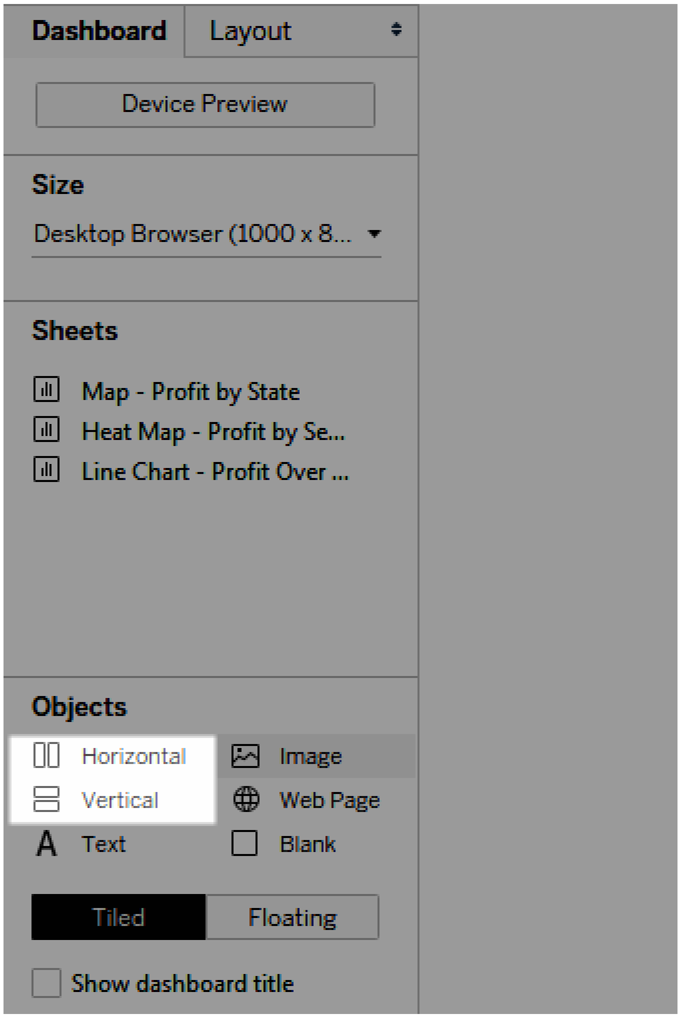
The image below shows how a dashboard behaves when two views are placed in a vertical layout container versus how they behave when they are not placed in a layout container. Notice how, in the dashboard that uses a layout container, the views adjust vertically as different filters are applied.



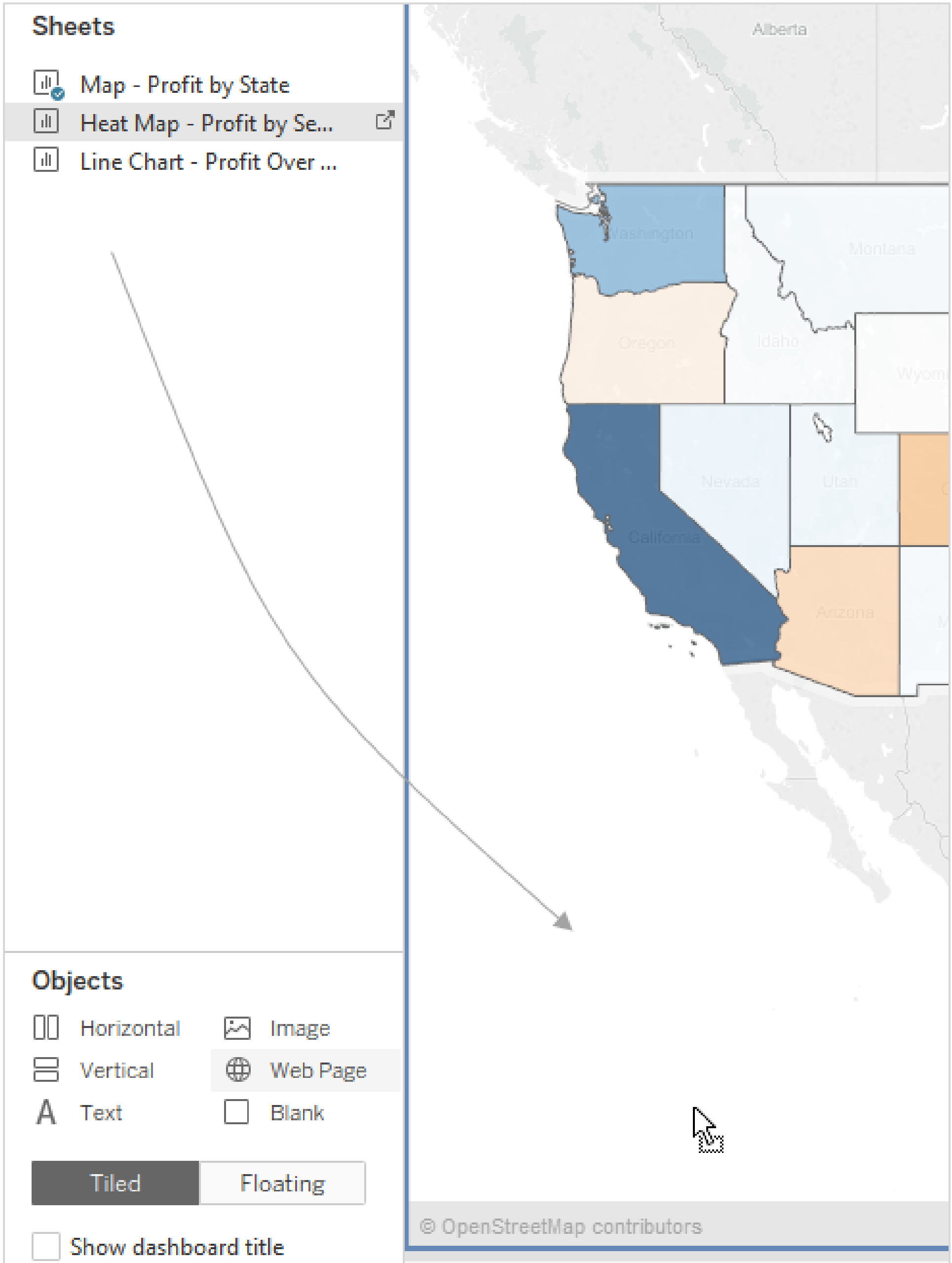
Click the image to replay it.

To add a layout container

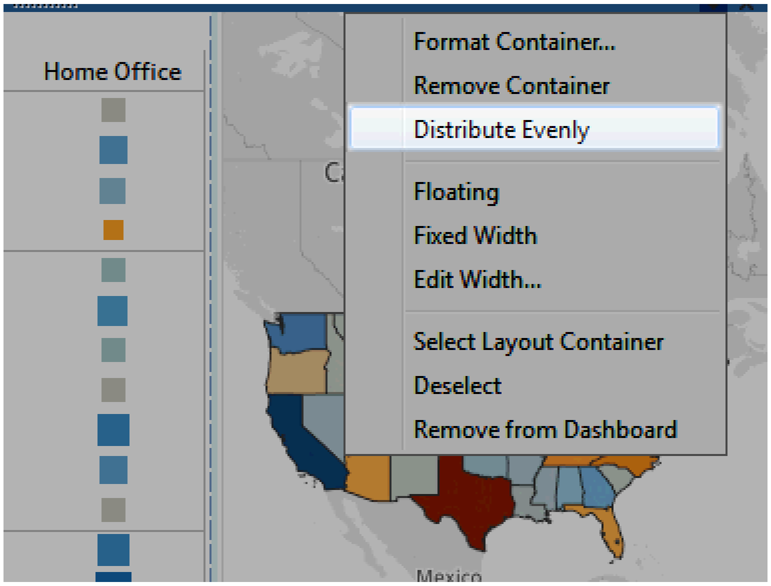
1. Drag a horizontal or vertical layout container to the dashboard.



2. Add sheets and objects to the layout container. As you hover over the layout container, a blue box indicates that the object is being added to the flow of the layout container.



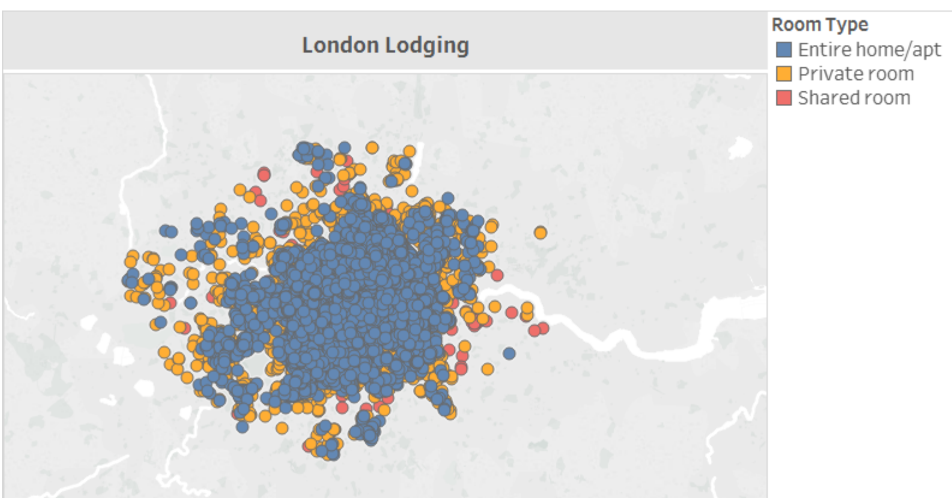
3. To position items evenly, choose **Distribute Evenly** from the layout container's shortcut menu:



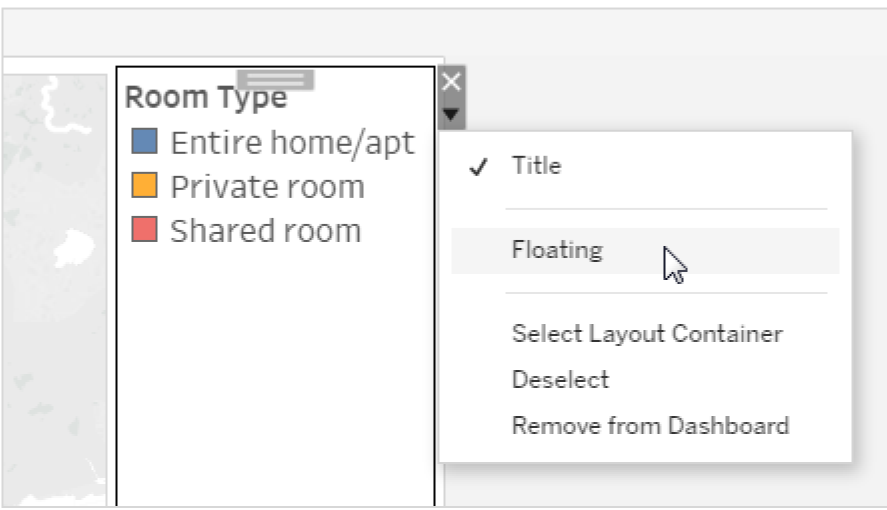
The **Distribute Evenly** command is only available for layout containers that you explicitly add to a dashboard. It's not available for layout containers that are automatically added when you add items.

Buy space with a floating layout

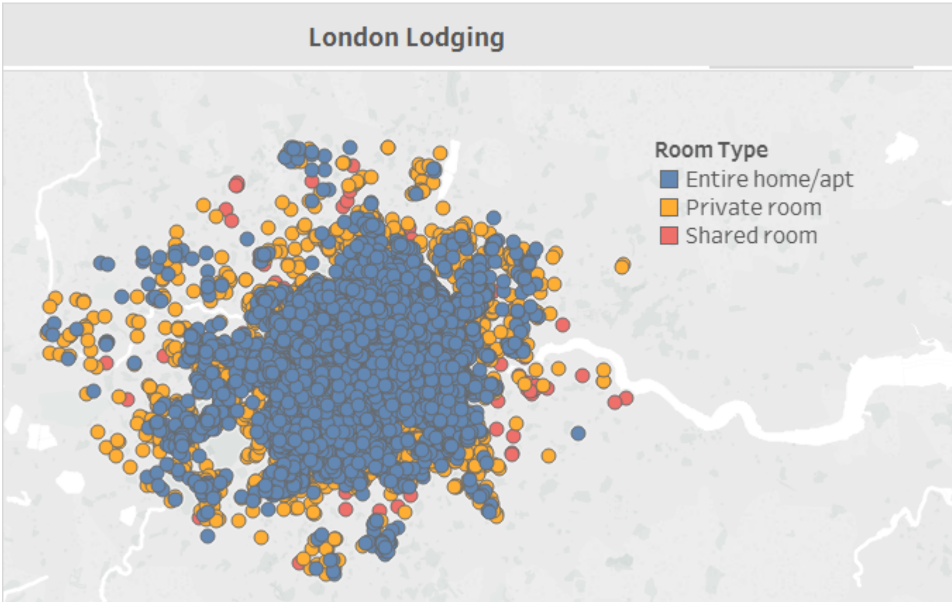
Tableau uses a tiled layout by default in dashboards, which means that each view, legend, and object is arranged in a single-layer grid, similar to a tiled floor.



If you want an item to overlap another, you can use a **Floating** layout for one or more items instead:



Legends can sometimes be floated over other items instead of remaining tiled.



See **Size and Lay Out Your Dashboard** on page 2755 for more details.

Tableau Desktop only: If you create device layouts for a dashboard, use a **Tiled** layout. It will give you the most control over where objects appear. See **Create Dashboard Layouts for Different Device Types** on page 2801 for steps.

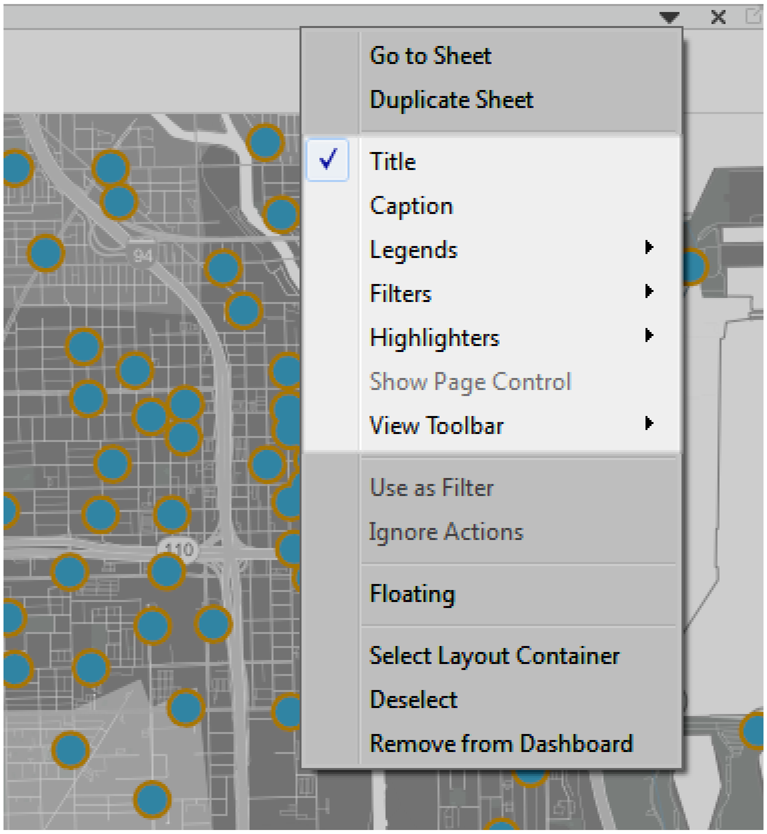
Highlight the essentials

Show just what your users need

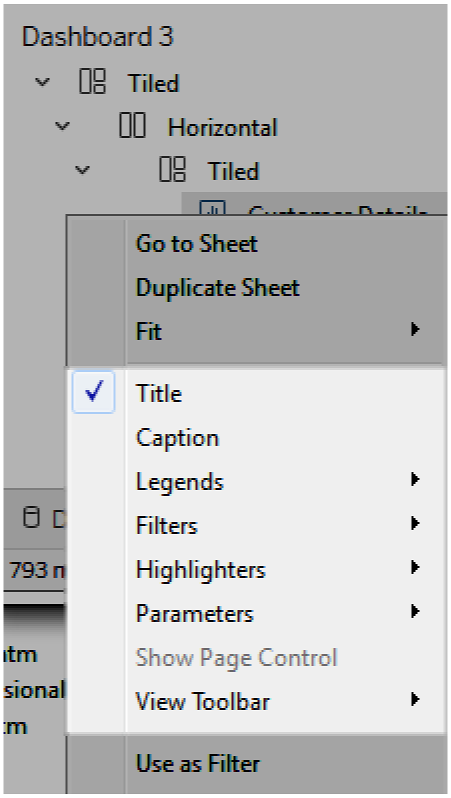
Legends, titles, captions, and filters can be helpful for your viewers. Make sure you give your users what they need and no more.

To change what's displayed for your users:

1. Select a view in the dashboard.
2. Click the drop-down menu in the upper-right corner of the selected view and select the items you want to show. For example, you can show the title, caption, legends, and a variety of filters.



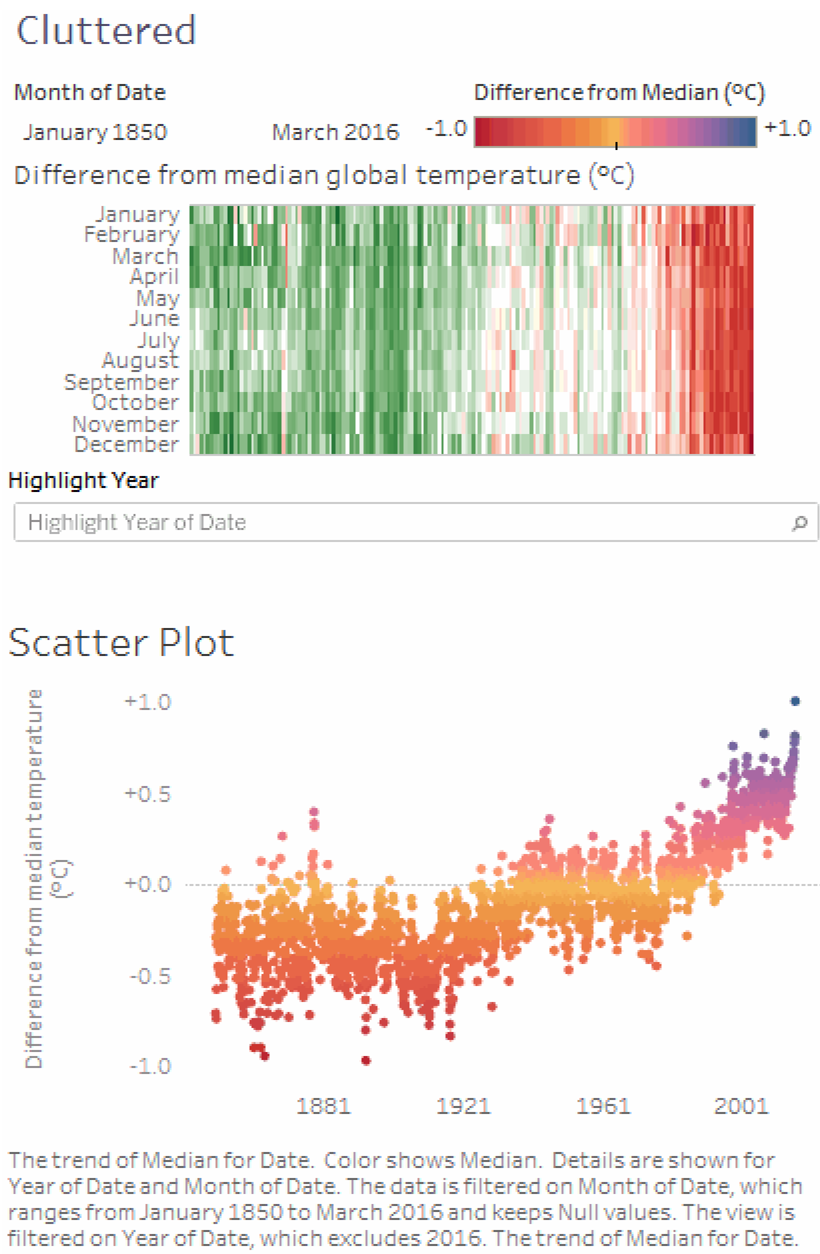
Alternatively, you can right-click an item in the **Layout** section of the **Dashboard** pane to access all of these same commands.



Note: Filters are only available for the fields used in the original view.

Eliminate clutter

In general, you should try to use no more than two color palettes in a single dashboard, and ideally, just one palette if your data is quantitative. Remove unnecessary text, lines, or shading that don't provide actionable information for your viewers. Make sure every legend you provide is really needed, remembering to take into account the final, published size of your dashboard.



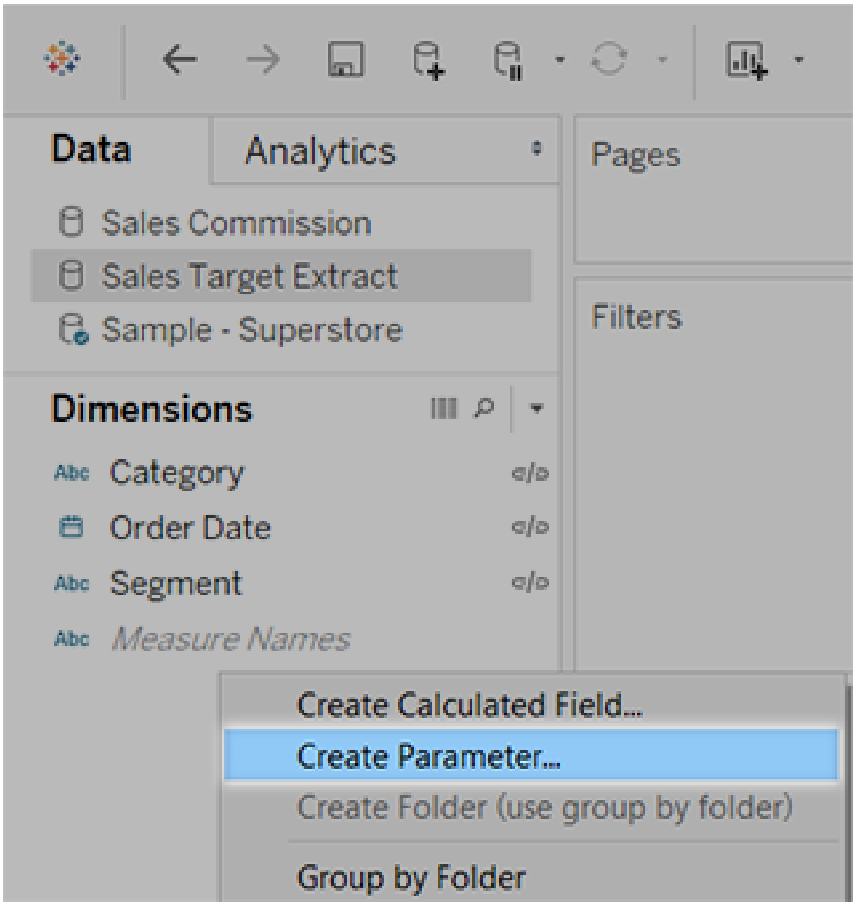
Click the image to replay it.

Create a Sheet Selection Menu for a Dashboard

Using a combination of a parameter and a calculated field, you can create a drop-down menu that lets users select an individual view that automatically expands to fill a dashboard.

Tip: To show sheets using separate buttons rather than a single menu, instead use Show/Hide buttons. This is a much simpler approach, though each button will reveal a separate floating layout container. (Tiled objects aren't supported.) To leave a dashboard and open a completely different sheet, use Navigation objects.

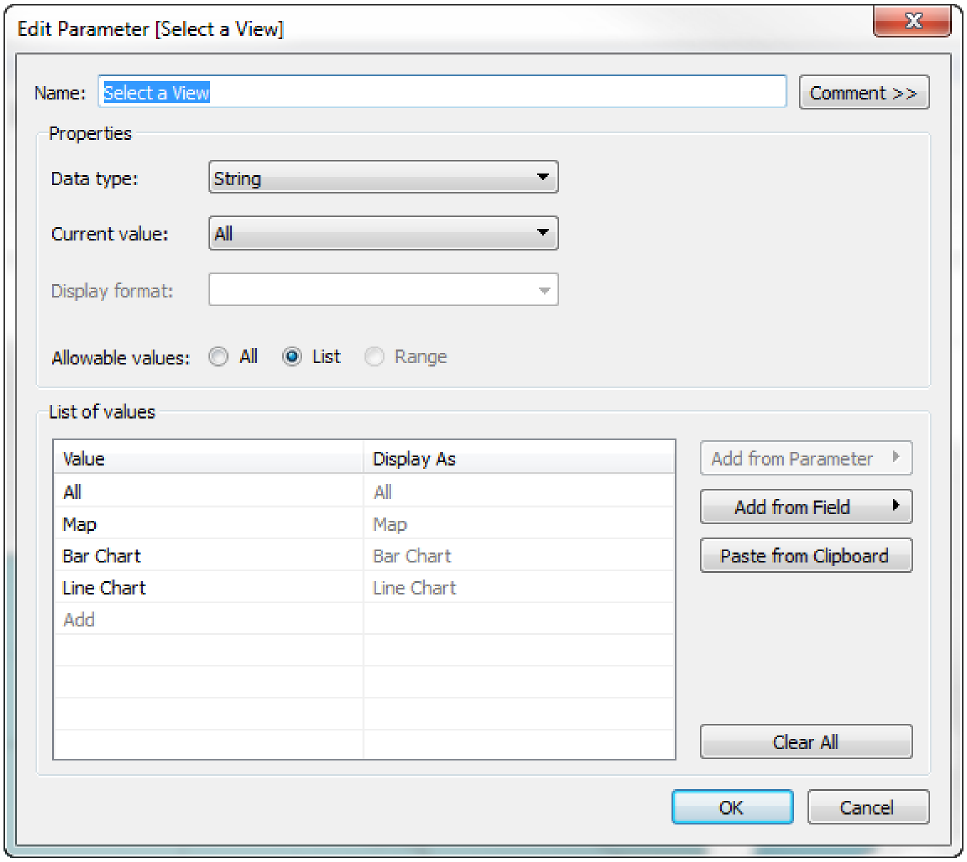
1. On an individual worksheet, right-click an empty area of the **Data** pane at left, and select **Create Parameter**.



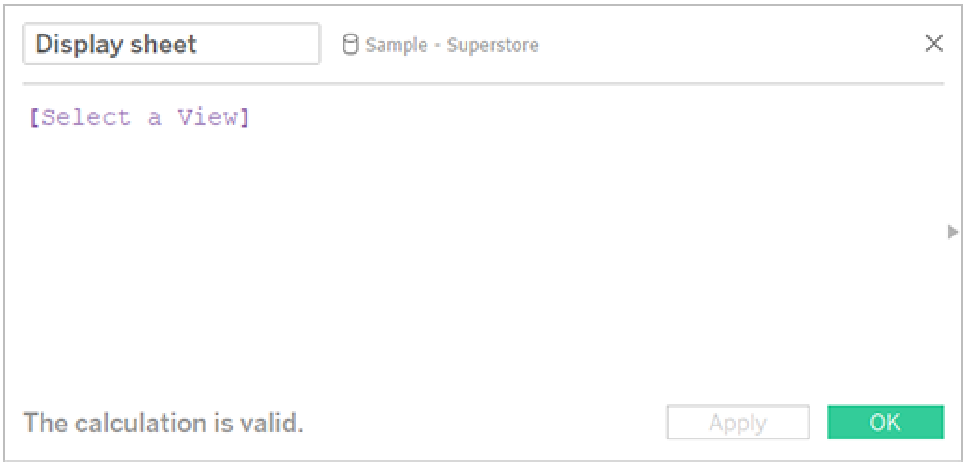
2. In the Create Parameter dialog box, do the following:

   - Enter a name that will appear appear above the menu, like **Select a View**.
   - For **Data type**, select **String**.
   - For **Allowable values**, select **List**.
   - Under **List of values**, type **All** for the first value, and then add values with the

name of each view in the dashboard.



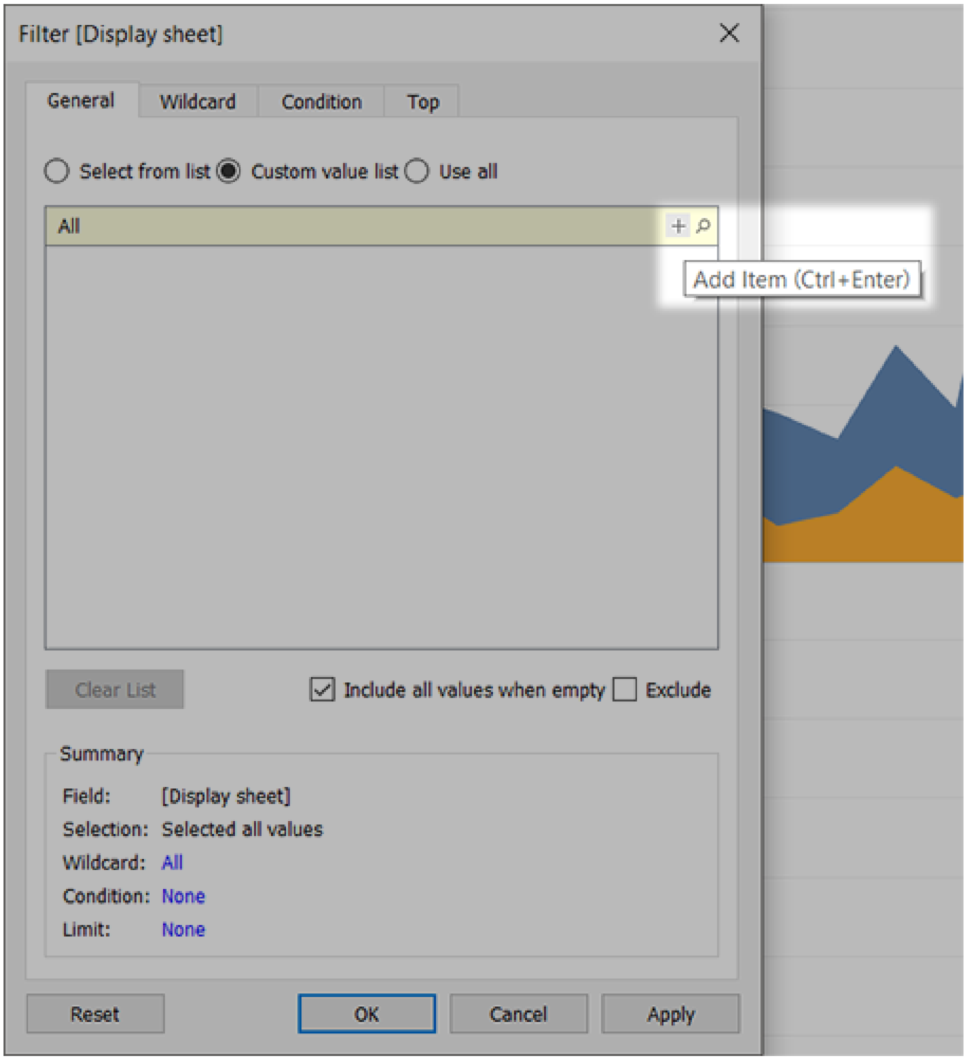
3. Click **OK**.
4. On any sheet, right-click an empty area of the Data pane at left, and select **Create Calculated Field**.
5. Give the calculation a descriptive name like **Display sheet**. In the formula text box, enter the name of the parameter you created above. Then click **OK**.



6. Open a sheet you plan to add to your dashboard, and drag the new calculation to the Filters shelf. Then do the following in the Filter dialog box:

    - Select **Custom Value List**.
    - Type **All** in the text box, and click the **Add Item** button.
    - Type the current view's name (like "Map") in the text box, and click the **Add Item** button.

    When you're done, click **OK**.



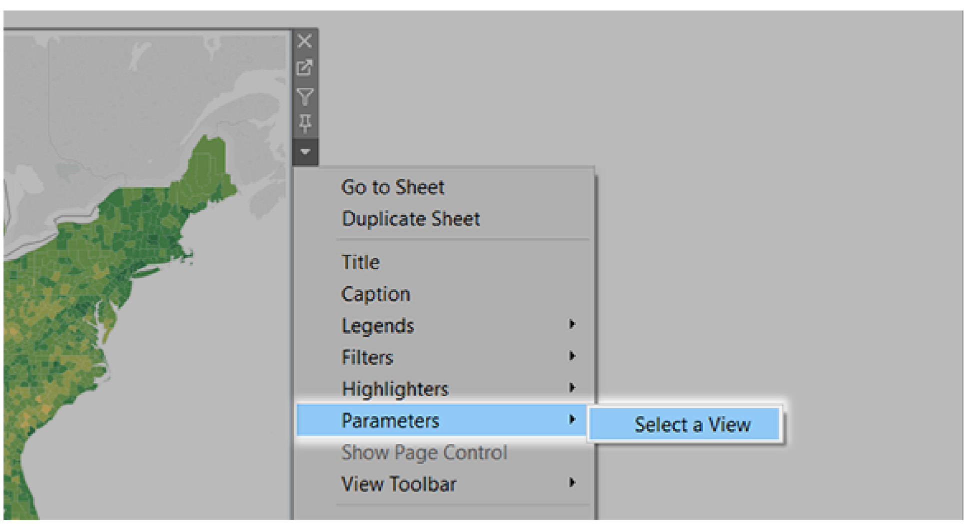
7. Repeat step 6 for every sheet you plan to add to your dashboard.

8. Select **Dashboard** > **New Dashboard**.

9. From the **Objects** section at lower left, drag a **Vertical** or **Horizontal** layout container to the dashboard.

10. Now drag each sheet to the layout container, identified by the dark blue outline.

11. To display the sheet selector, from the drop-down menu at the top of a view, choose **Parameters** > [new parameter name].



12. To ensure that selected views fill the dashboard, do the following:

- From the drop-down menu at the top of each view, ensure that **Fix Width** is *not* selected.
- Right-click the title area of each view in the dashboard, and select **Hide Title**.

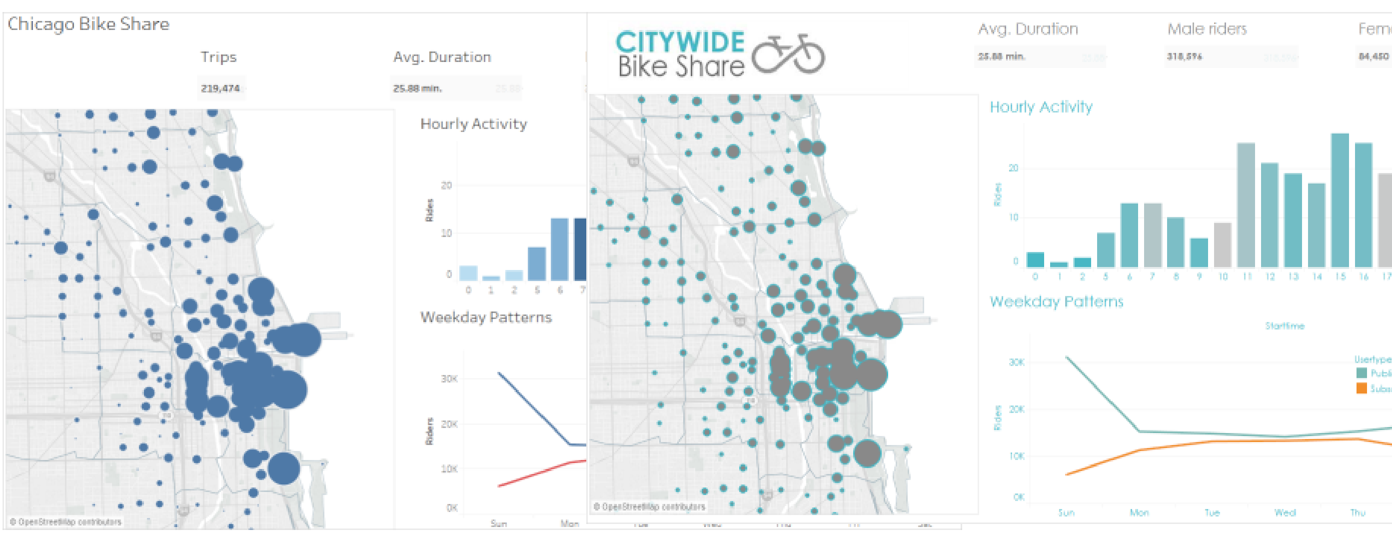
Your sheet selector is ready to go! For similar examples, see **Use Parameters to Make Views More Interactive** on page 1227.

Rebrand a Dashboard

Tableau Desktop builds visual best practices into its default formatting settings so that your visualizations look great from the start. But you might want to change formatting settings for a dashboard—for example, to align them with your organization's brand.

This article steps you through a dashboard rebranding process for a fictitious organization called Citywide Bike Share. Its brand includes turquoise, orange, and gray for its colors and Century Gothic for its font. The overall look is light and uncluttered and, of course, it's all about bikes!

Before **After**

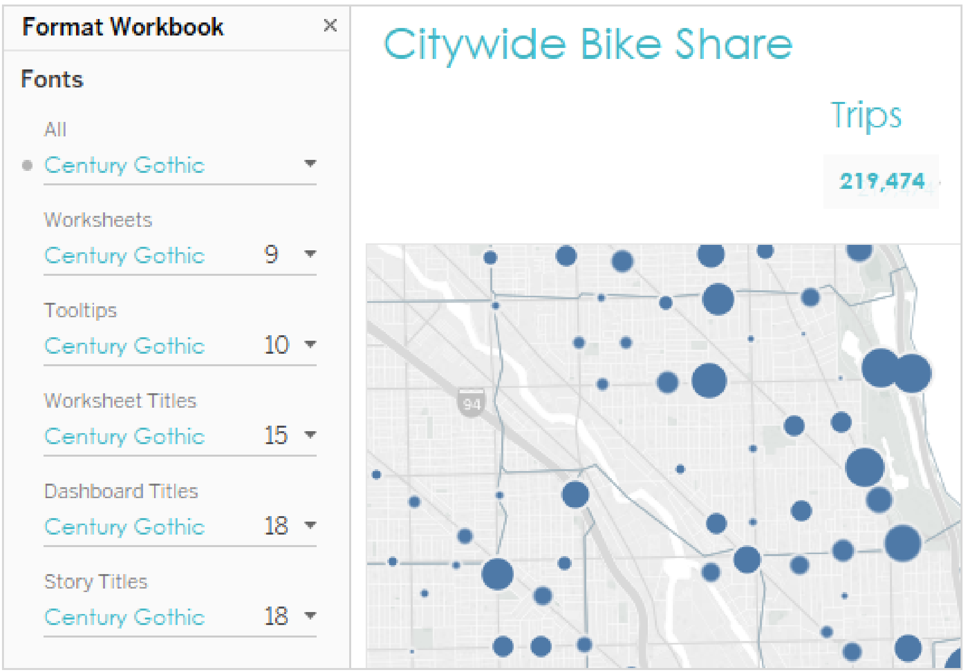


Change the fonts and font color of your titles

Fonts and colors are usually an important part of an organization's brand. The quickest way to make a fast, large-scale change to all the titles in your dashboard is to change them at the workbook level. A workbook is the largest possible container for formatting settings.

1. Click **Format** > **Workbook**.
2. Make your changes under **Fonts**, in the **Format Workbook** pane.

   Here we've changed the font to Century Gothic and the font color to turquoise. We've chosen to change every title in the workbook, but you could also change the look of individual titles.



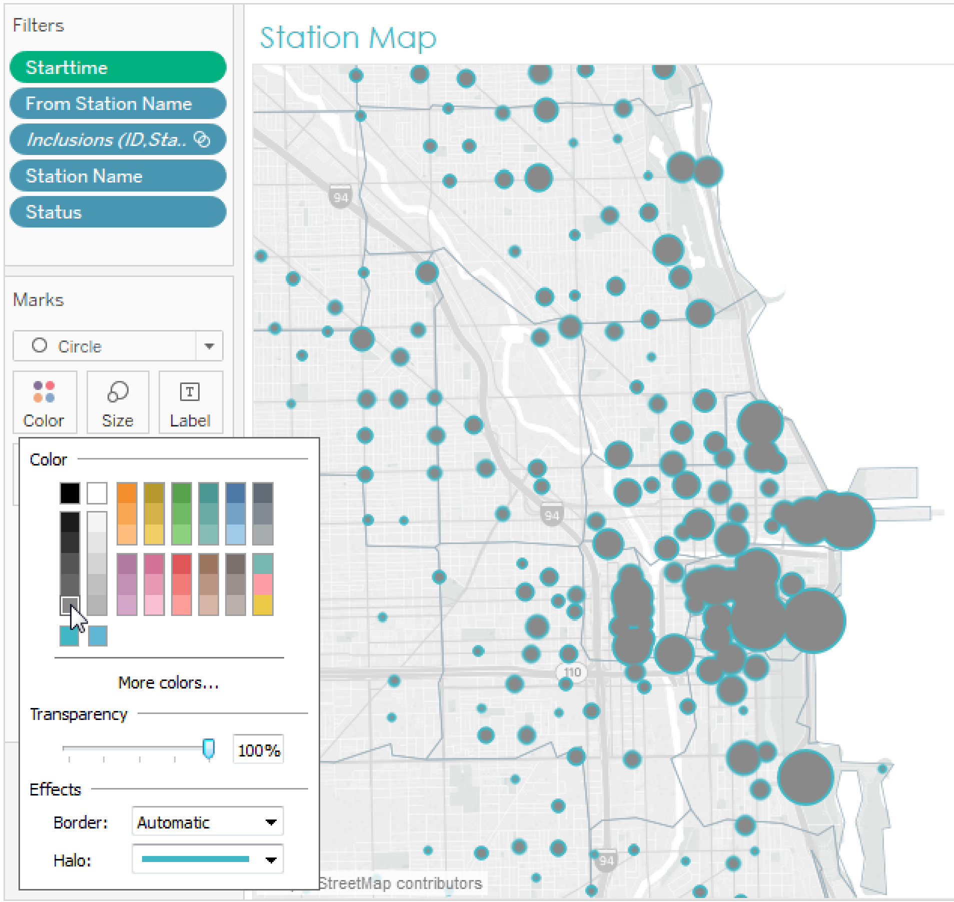
   If the font you want to use doesn't exist in Tableau by default, you can add it. See **Use Custom Fonts** on page 2934 for details.

Change the marks colors

Marks represent data in a view. In this rebranding scenario, the marks show bike share locations. Changing marks colors is another way to communicate your organization's brand to your audience. For certain types of marks, you can also add and format halos, which is a colored ring around a mark.

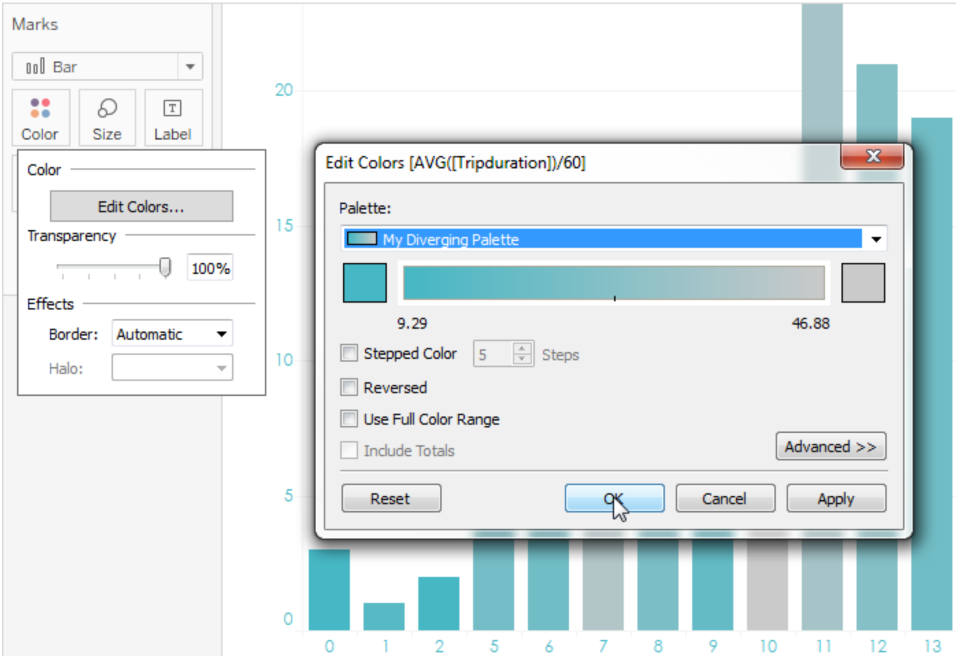
1. Go a view on your dashboard and click **Go to Sheet** from the view's shortcut menu.
2. In the view, click **Color** on the **Marks** card.
3. Make your new color selection.

Here we've changed the marks in the Chicago Ride Share map to use gray with a turquoise halo:

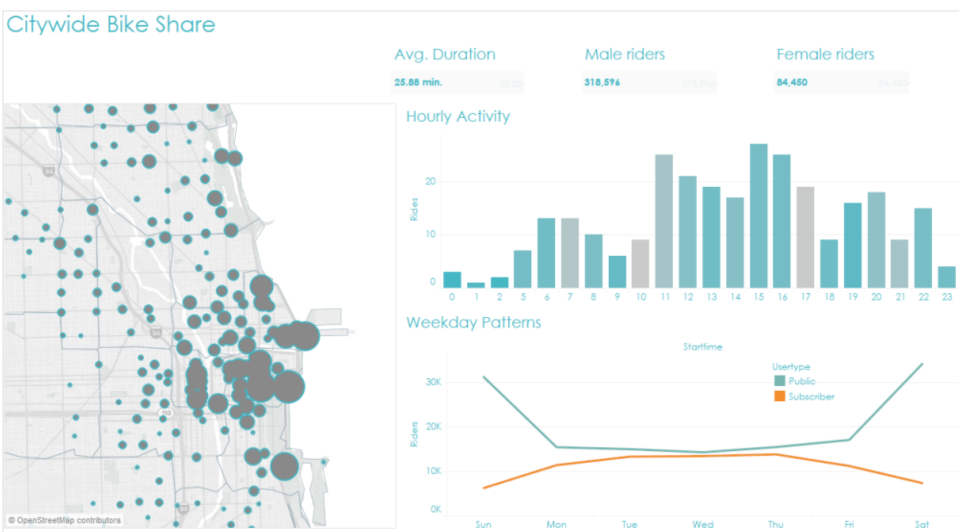


4. Repeat the above steps for every view in your dashboard.

If a color you want to use isn't available, you can create your own custom palette. See **Create Custom Color Palettes** on page 2969 for details. For example, here we've changed the colors in the bar chart by created a custom diverging palette.



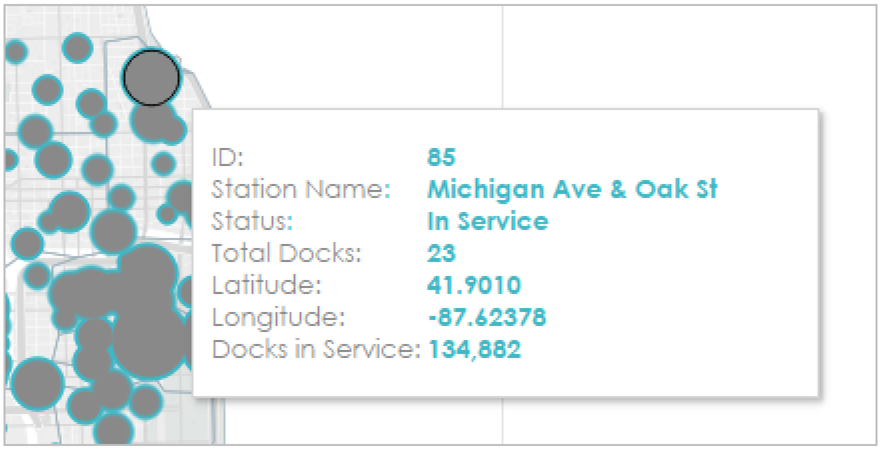
Now, with just a few changes to fonts and marks colors, the dashboard branding is already taking shape.



Customize your tooltips

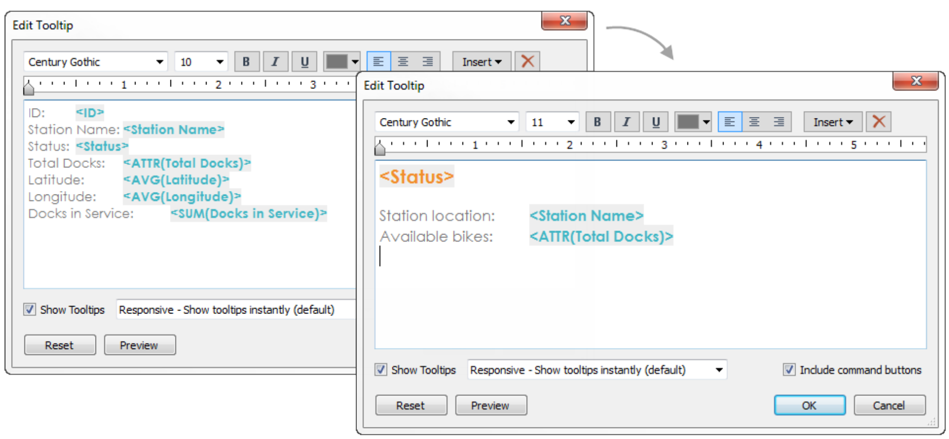
Tooltips appear when you hover the mouse pointer over a mark. Tooltips appear by default for most views. They're a great way to reinforce your brand and tell a better story.

Here's what the map tooltips looked like by default:

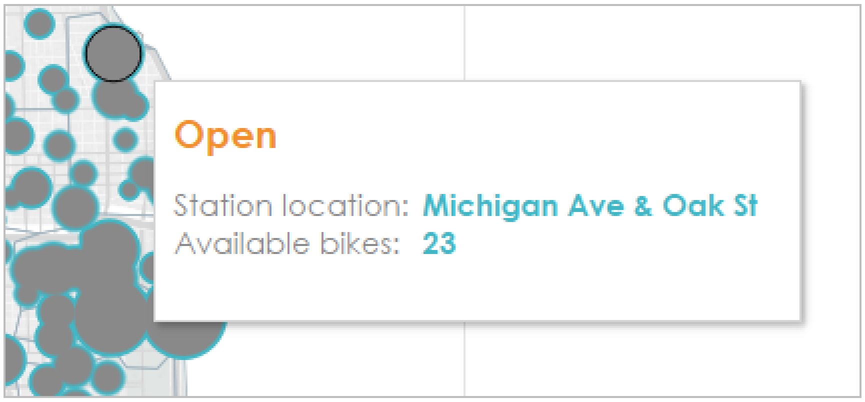


1. Go to the view's worksheet and click **Worksheet** > **Tooltip**.
2. Confirm that the tooltip has the details you want to display. You can change fonts, order, wording, alignment, and color.

   For this rebranding scenario, we reduced the number of items displayed in the tooltip so that it's easier for users to quickly see what they're most interested in. We also reworded a few items and added the color orange as a contrasting, brand-appropriate color:



Here's what the map tooltips look like now:

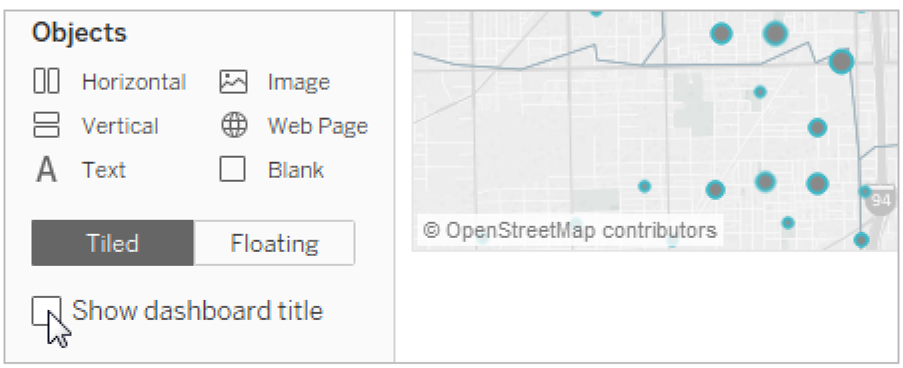


Add an image or logo

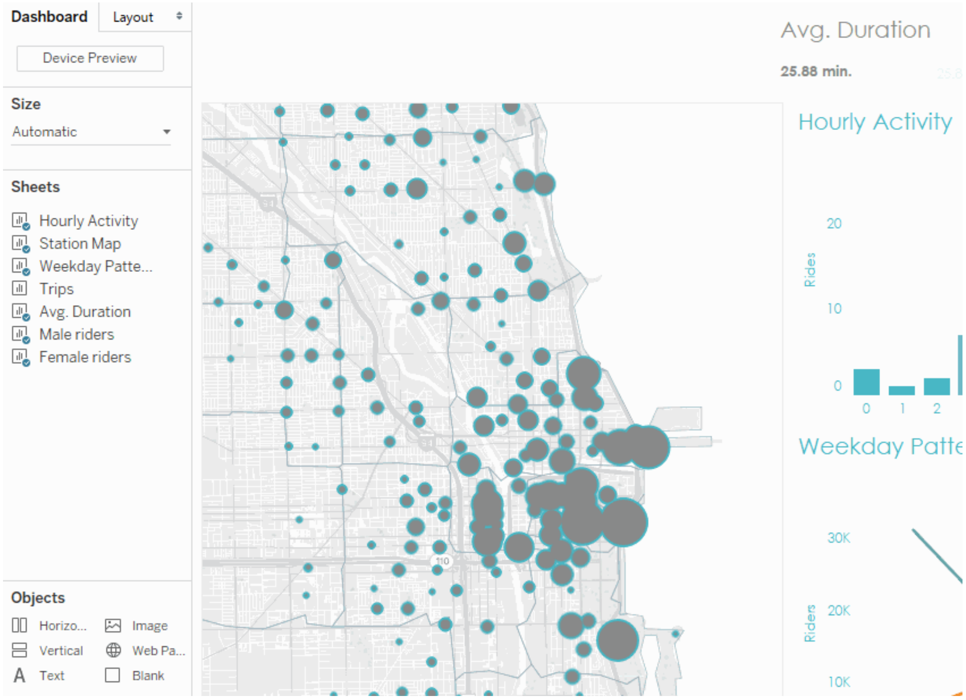
Adding your own images can also be a way to bring your organization's brand into a workbook. For example, you can import a logo and use it at the top of your dashboard instead of displaying the default dashboard title.

To remove the title and add an image:

1. In the Dashboard pane, stop displaying the dashboard title by clearing the **Show dashboard title** check box in the lower left corner.

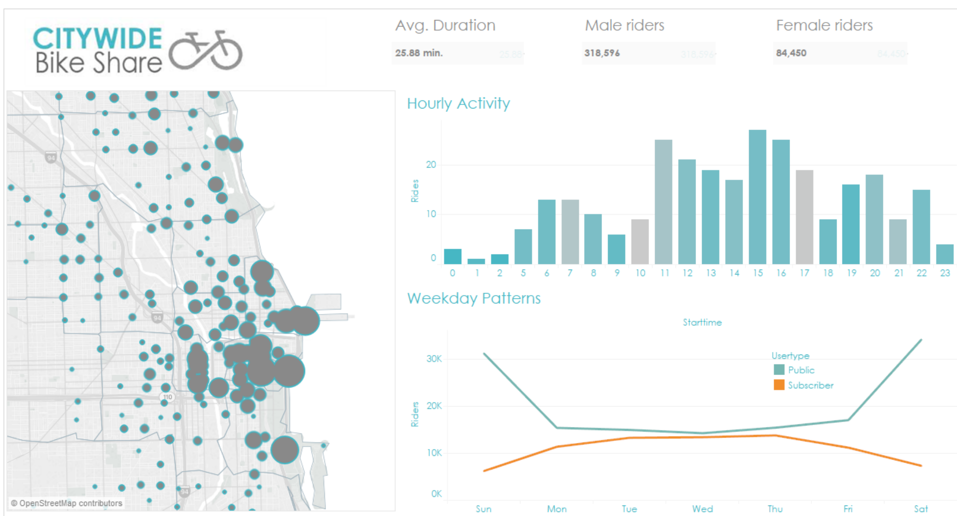


2. Drag the **Image** object to the area where you want to display your logo. You may also need to adjust or remove other dashboard objects.



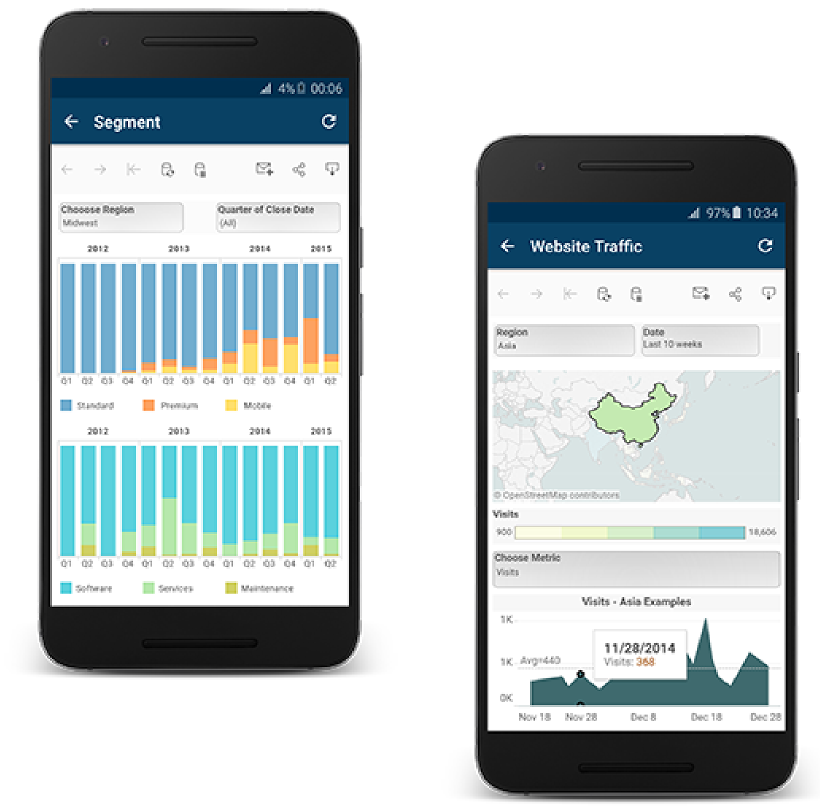
Click the image to replay it.

Here's our final, rebranded workbook:



Create Dashboard Layouts for Different Device Types

Dashboards can include layouts for different types of devices that span a wide range of screen sizes. When you publish these layouts to Tableau Server or Tableau Online, people viewing your dashboard experience a design optimized for their phone, tablet, or desktop. As the author, you only have to create a single dashboard and deliver a single URL.



Tip: In addition to optimizing layouts for mobile, optimize workbook performance to better meets the needs of mobile users, who often have limited bandwidth and are in a hurry.

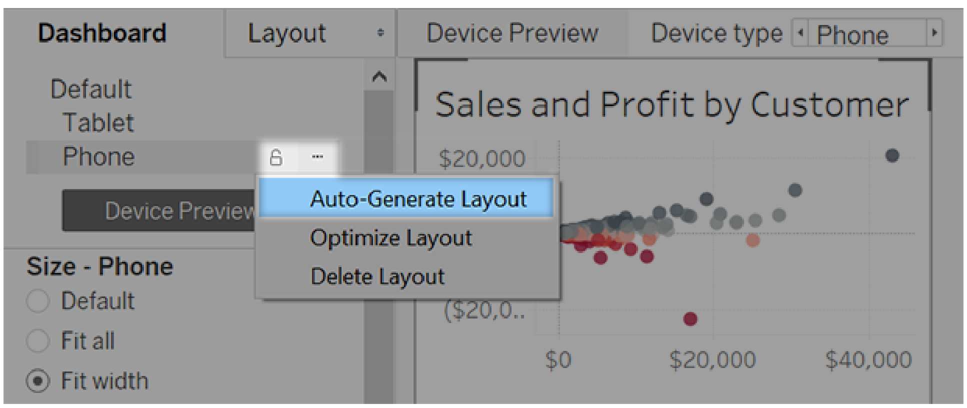
How the Default dashboard relates to device layouts

Device layouts appear on the Dashboard tab, under Default. Initially, each device layout contains every item in the Default dashboard and derives its size and layout from Default as well.

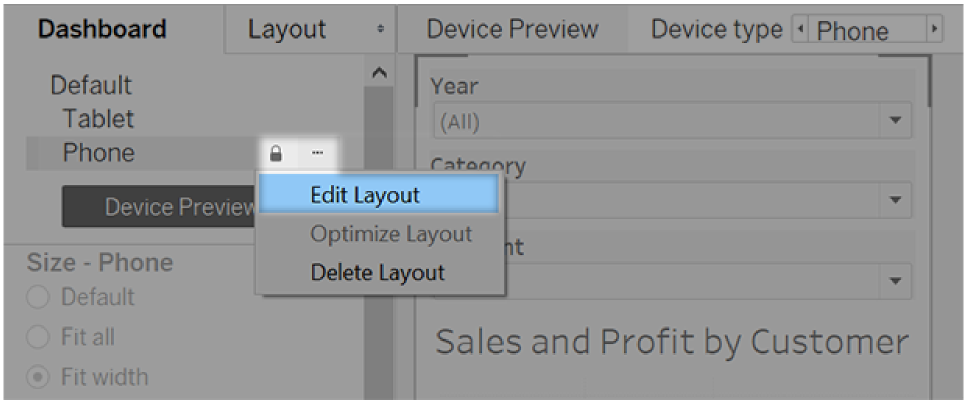
Think of the Default dashboard as the parent, and the device layouts (desktop, tablet, and phone) as its children. Any view, filter, action, legend or parameter that you want to add to a device layout must first exist in the Default dashboard.

Phone layouts and the Default dashboard

To save time with a unique Phone layout option that automatically reflects changes to the Default dashboard, either click the open lock icon , or choose **Auto-Generate Layout** from the pop-up menu.

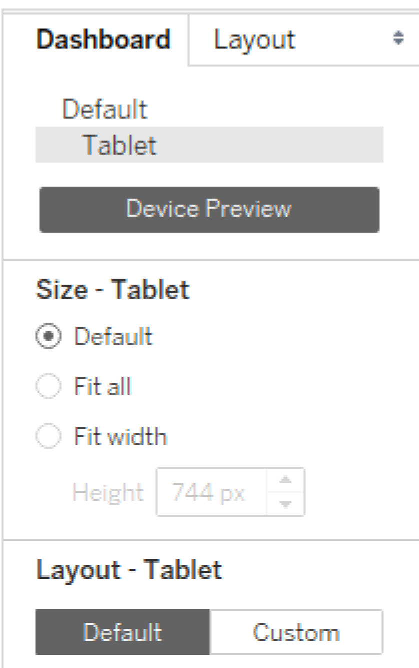


If you instead click the closed lock icon or choose **Edit Layout** from the menu, the Phone layout becomes fully independent, so you'll need to manually add and arrange items to reflect changes to the Default dashboard.



Desktop and Tablet layouts and the Default dashboard

Unlike Phone layouts, you need to manually add Desktop and Tablet layouts to a dashboard. Desktop and Tablet layouts are always fully independent from the Default dashboard, so each device layout can contain a unique arrangement of objects.



Automatically add phone layouts

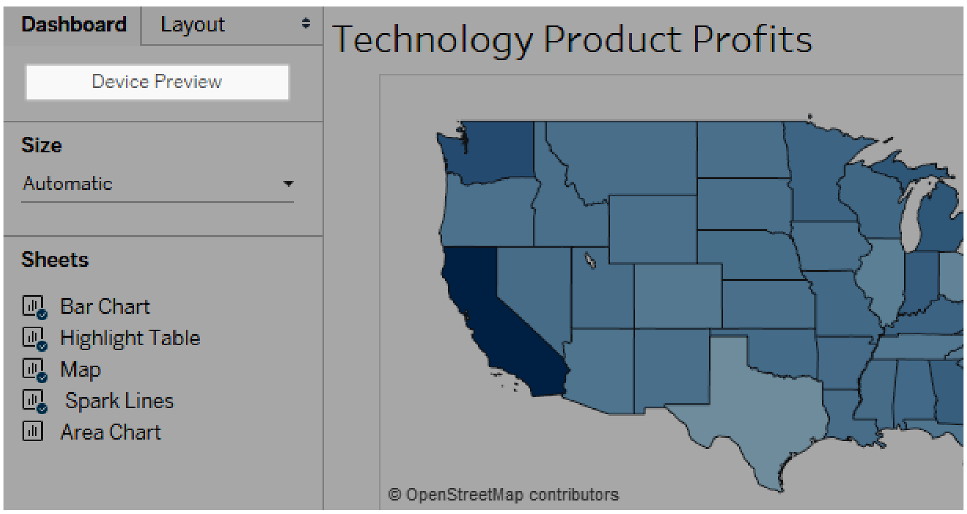
Two options let you automatically add phone layouts:

- To create phone layouts whenever you open old dashboards that lack them, choose **Dashboard** > **Add Phone Layouts to Existing Dashboards**
- To create phone layouts whenever you create a new dashboard, choose **Dashboard** > **Add Phone Layouts to New Dashboards**. (This option is on by default.)

Preview and manually add device layouts

1. Open a dashboard.

2. On the **Dashboard** tab on the left, click **Device Preview**.

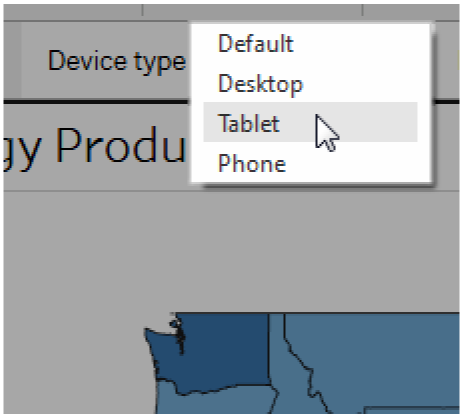


In device preview mode, these options appear above the dashboard:

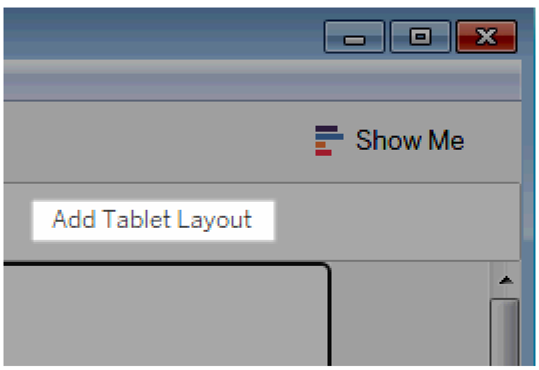
3. Take a moment to click through the **Device types** and **Models** and explore the different screen sizes. Then set these options:

   - To see how the dashboard will look in landscape vs. portrait mode, click . Usually, landscape is optimal for tablets and portrait is best for phones.
   - Select **Tableau Mobile app** to see how the dashboard will look with the app instead of the browser. This option is available for iOS or Android devices and shrinks the dashboard slightly, leaving space for the app controls.

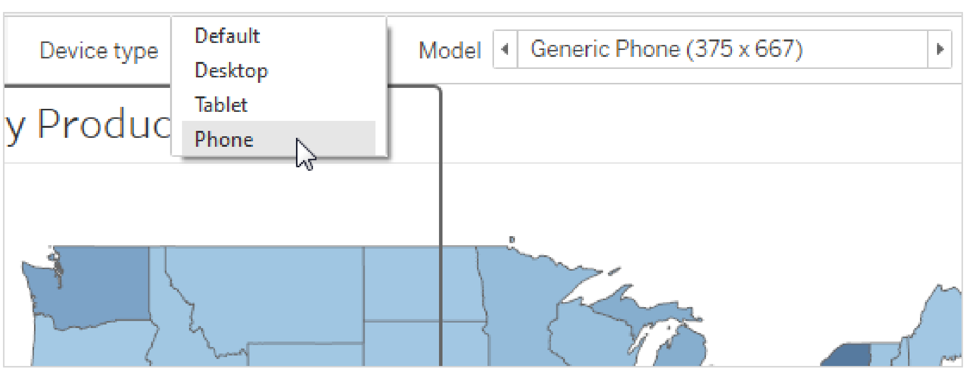
4. Choose a **Device type**, such as **Tablet**.



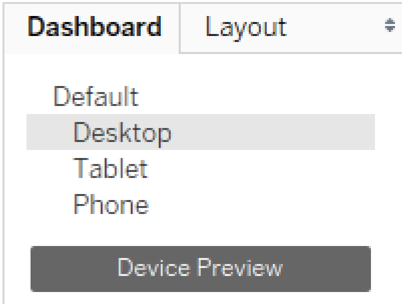
5. In the upper-right corner, click the **Add Layout** button for the device type you selected (for example, **Add Tablet Layout**).



6. Add an additional layout by selecting a new **Device type** and clicking **Add Layout**.



Creating a layout for each device type gives you the most control over your users' experience as they view your dashboard from different devices. After you publish a dashboard with all three layouts, users won't see the default dashboard layout; instead, they'll always see the appropriate device-specific layout.

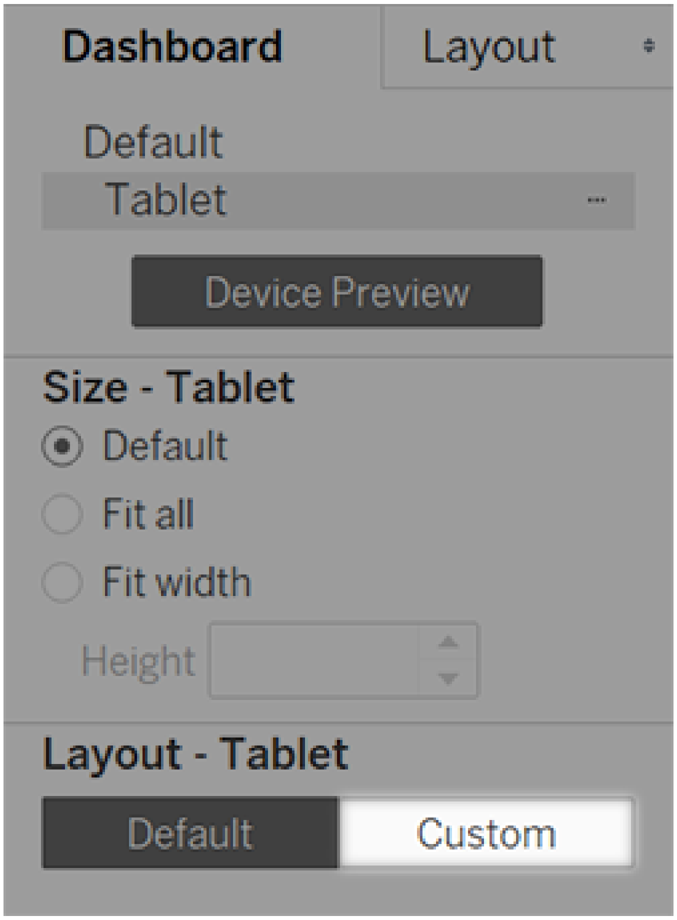


Note: If you make changes to a view, double-check related devices layouts to ensure that they look as you expect.

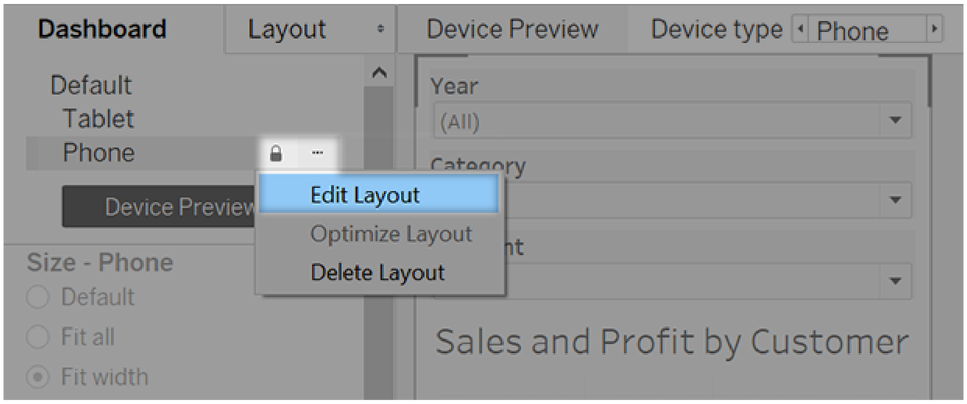
Customize a device layout

After you've added a device layout to your dashboard, you can start rearranging objects to create the look you want.

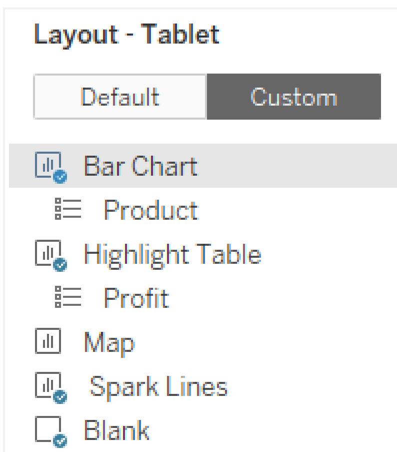
1. For Desktop and Tablet layouts, click **Custom**:



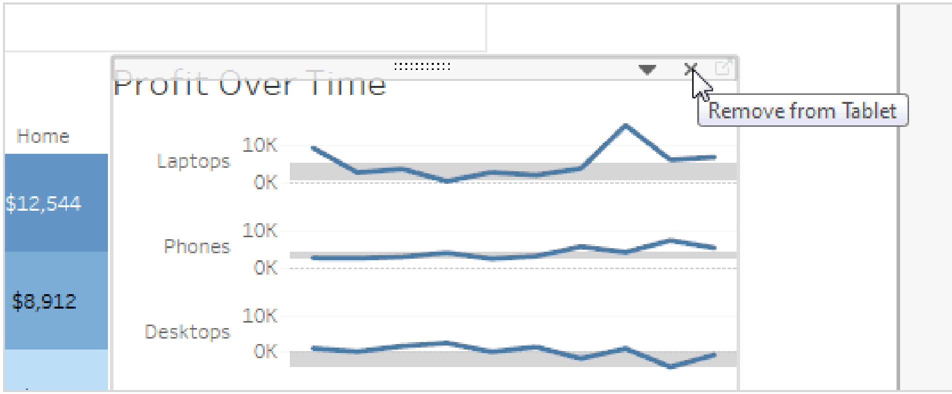
For Phone layouts, either click the lock icon , or choose **Edit Layout** from the pop-up menu:



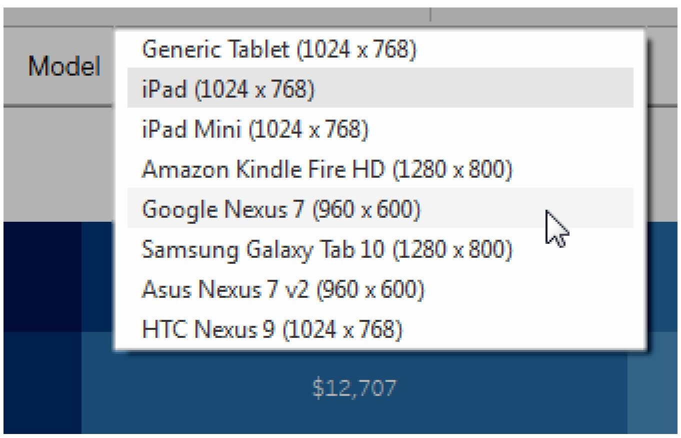
2. Anything you can add to your layout is listed on the left, under **Layout**. If an item has a blue check mark, it means that it's part of the device layout that you're currently working on.



3. If you remove an item, it's only removed from the current device layout. It still exists on the default dashboard and can be added to the device layout again.

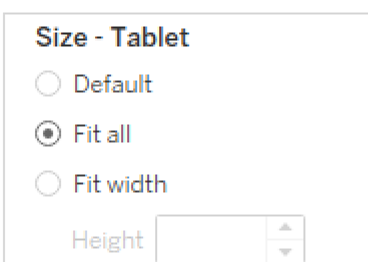


4. Click through the **Device model** options to see how the layout will appear on different models.



Ultimately, it's the size of the web browser that loads the dashboard that determines which layout appears on the device. For details, see **Confirm which layout a device will display** on page 2814.

5. At left, explore the options under **Size**.



Default: The height and width of the device layout mimics whatever the default dashboard is using. For example, if you're creating a tablet layout and the default dashboard is set to a fixed size of Desktop Browser (1000 x 800), setting Size to Default for the tablet layout will make it use 1000 x 800 as well.

Fit all: All items are automatically resized to fit the device frame size. The device frame size is determined by the Device type, Model, and orientation (portrait or landscape) settings.

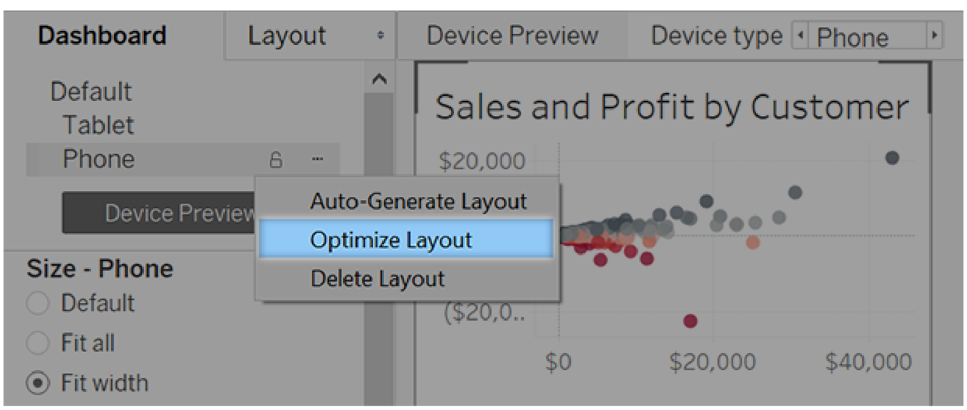
Fit width (recommended for phones): Items are automatically resized to fit the width of the device frame, but the height is fixed. This is a great option for phone layouts and vertical scrolling.

Optimize for phones

The small screens of mobile phones benefit from further optimization. Try these techniques.

Optimize manual phone layouts

If you've chosen to edit a phone layout yourself, you can quickly optimize the placement of filters, remove white space, and more. On the **Dashboard** tab, click the pop-up menu to the right of **Phone**, and select **Optimize Layout**.



Be aware that this command only rearranges items currently in the phone layout. If you want to continuously update the phone layout to reflect all changes to the default dashboard, select **Auto-Generate Layout**.

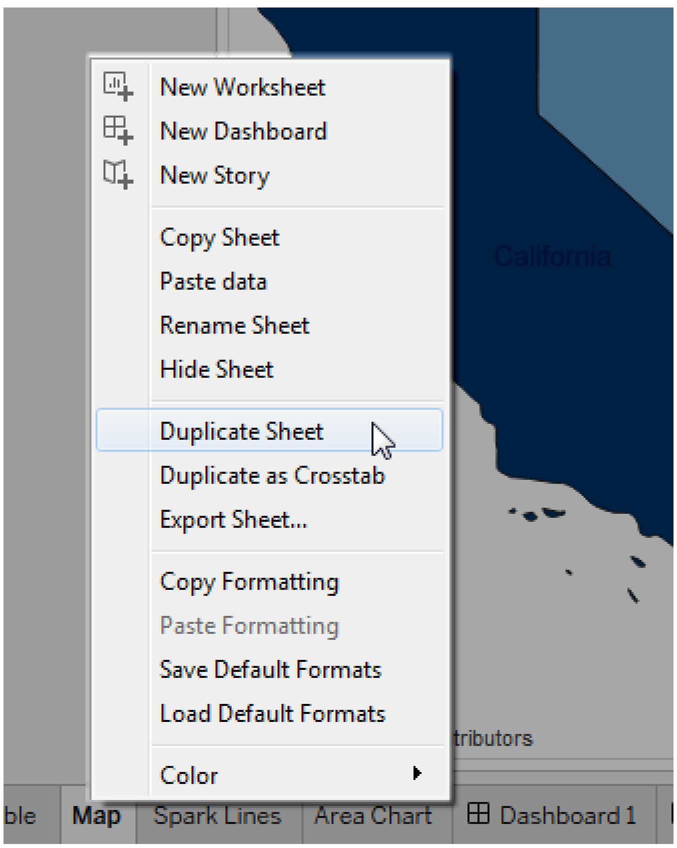
Add links that trigger instant messages and phone calls

To let phone users quickly contact key people about dashboard content, add URL actions to objects that automatically trigger SMS messages and telephone calls. Use the link format `sms:`*`phone-number`* or `tel:`*`phone-number`*. Be sure to include country and area codes if necessary.

Create phone-specific versions of views

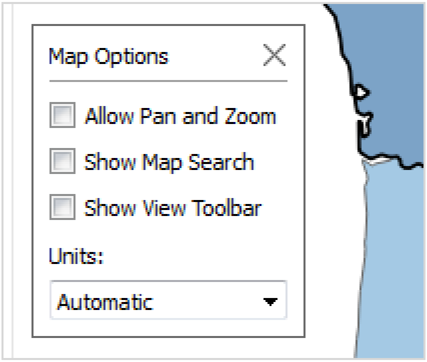
Create duplicates of certain views in the default dashboard—one optimized for desktop viewing and a second optimized for phones.

1. Go to the worksheet for a view, click its tab, and select **Duplicate Sheet**.



2. Customize the view for mobile viewing.

   For maps for example, you may want to zoom in to a specific region by default, or you may want to disable panning, zooming, and other functionality. See **Customize How People Interact with your Map** on page 1989.



3. Now add the new view to the default dashboard so that it can be available to the device layouts you're creating.

Shorten titles

Short titles work best for mobile viewing. To edit a title, double-click it.

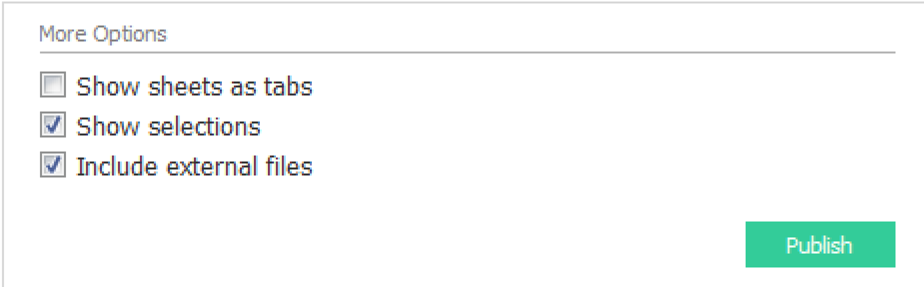
Optimize white space

White space is another visual element to consider. While screen real estate on a phone is scarce and you want to make the most of it, you may also want to provide additional safe places for your users to tap or initiate scrolling, so they don't select filters and other items unintentionally.

To add white space, use padding or Blank objects. For more information, see **Size and Lay Out Your Dashboard** on page 2755.

Publish the dashboard

1. Click **Server** > **Publish Workbook**. If you're not already signed in, you're prompted for your credentials.
2. In the Publish Workbook to Tableau Server dialog box, make sure the **Show sheets as tabs** check box is cleared.



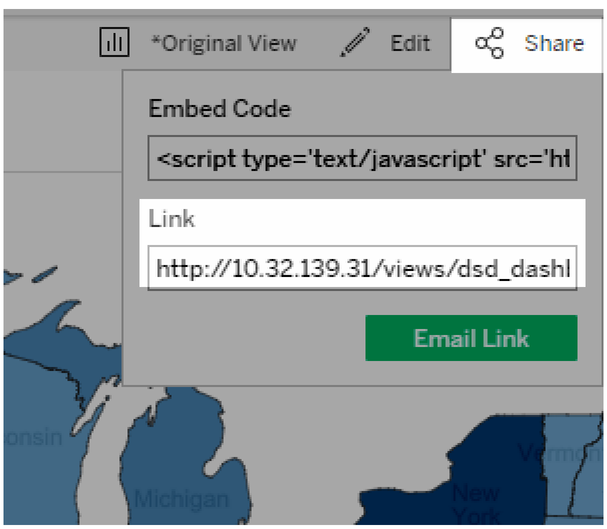
When this check box is selected for device-specific dashboards, the tabs' sizing requirements interfere with the server's ability to correctly detect the size of the web browser and load the correct layout.

3. Click **Publish**.

Test the dashboard

After you publish the dashboard to Tableau Server or Tableau Online, test the dashboard by viewing it from different browser sizes.

1. Open the dashboard on Tableau Server or Tableau Online.
2. In the upper-right corner of the page, click **Share** and copy the contents of the **Link** text box.



3. Paste the string into a web browser URL. The string should include the following:
   `embed=y`

4. With the embed code string as your browser URL, test the different layouts by changing the size of your web browser window and refreshing it.

Confirm which layout a device will display

The dashboard layout a device displays is based on the smallest dimension (height or width) of the iframe in which the Tableau view appears. Sometimes Desktop, Tablet, or Phone layouts may appear on other types of devices. For example, a Tablet layout may appear on a desktop computer if the display or browser window is small.

If the smallest iframe dimension is ...	This device layout appears ...
500 pixels or less	Phone
Between 501 and 800 pixels	Tablet
Greater than 800 pixels	Desktop

If Tableau Online and Tableau Server users find a Phone or Tablet layout too limiting, they can click **See Desktop Layout** in the toolbar. This toggle button lets users switch back to the mobile device layout at any time.

Build Accessible Dashboards

If you want to make your dashboards accessible to as many people as possible, or if you work in an environment that is subject to US Section 508 requirements, other accessibility-related laws and regulations, you can use Tableau to create dashboards that conform to the Web Content Accessibility guidelines (WCAG 2.0 AA).

To make an accessible dashboard:

1. Create a dashboard in either Tableau Desktop or in web authoring on Tableau Server or Tableau Online
2. Publish and embed that dashboard into a web page that conforms to Web Content Accessibility guidelines (WCAG 2.0 AA)

This topic explains how screen readers help users to navigate dashboards, the order that screen readers read objects or views in a dashboard, what items in a dashboard are accessible, and how to take an existing dashboard and make it more accessible for all users.

Dashboard keyboard navigation and focus order

Tableau contains keyboard shortcuts that helps users navigate using only a keyboard. For more information, see **Keyboard Accessibility for Tableau on the Web** on page 3164.

When creating a dashboard, note that screen readers read views or objects in a dashboard in the order in which they were added. For example, if your dashboard contains a text object containing information that helps explain the dashboard, add this object to your dashboard first.

You can also edit the dashboard hierarchy directly by editing the XML directly. For more information, see the Tableau community post How can I set the focus order of the views and objects in a dashboard?

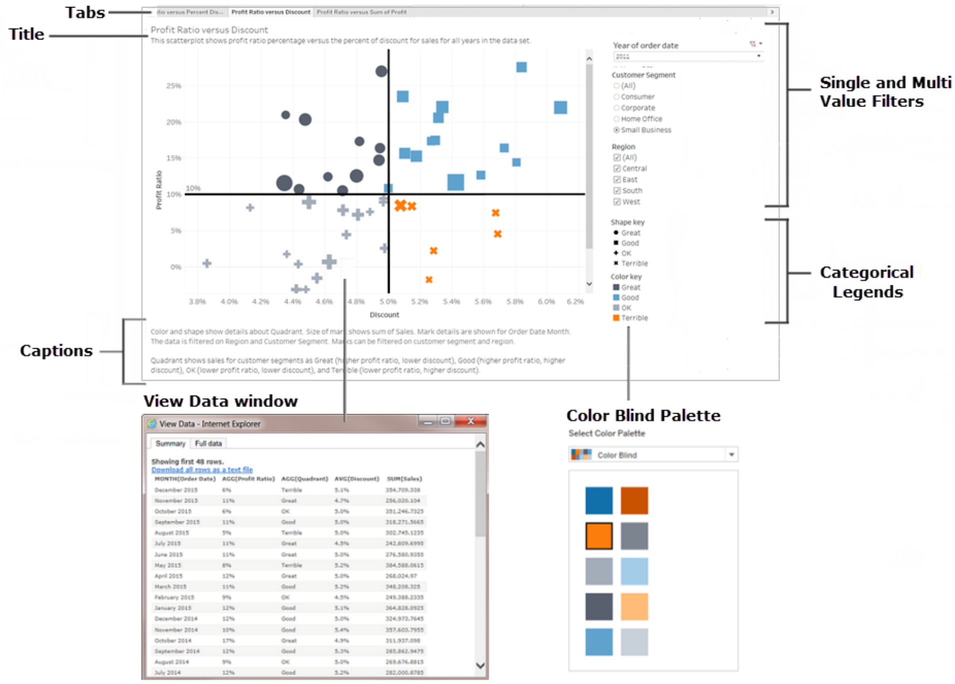
Create accessible dashboards

Many items and objects in the Tableau workspace support keyboard navigation and are compatible with assistive technologies like screen reading software. You can use the following items on a dashboard you want to be accessible:

- **Add dashboard objects and set their options** on page 2743
  - Text objects
  - Web page objects
  - Button objects
  - Image objects
- Toolbar
- Dashboard titles
- Views, specifically:
  - Title
  - Workbook tabs
  - View Data window
  - Captions (when visible)

- Categorical legends

  Single and multi-value filters

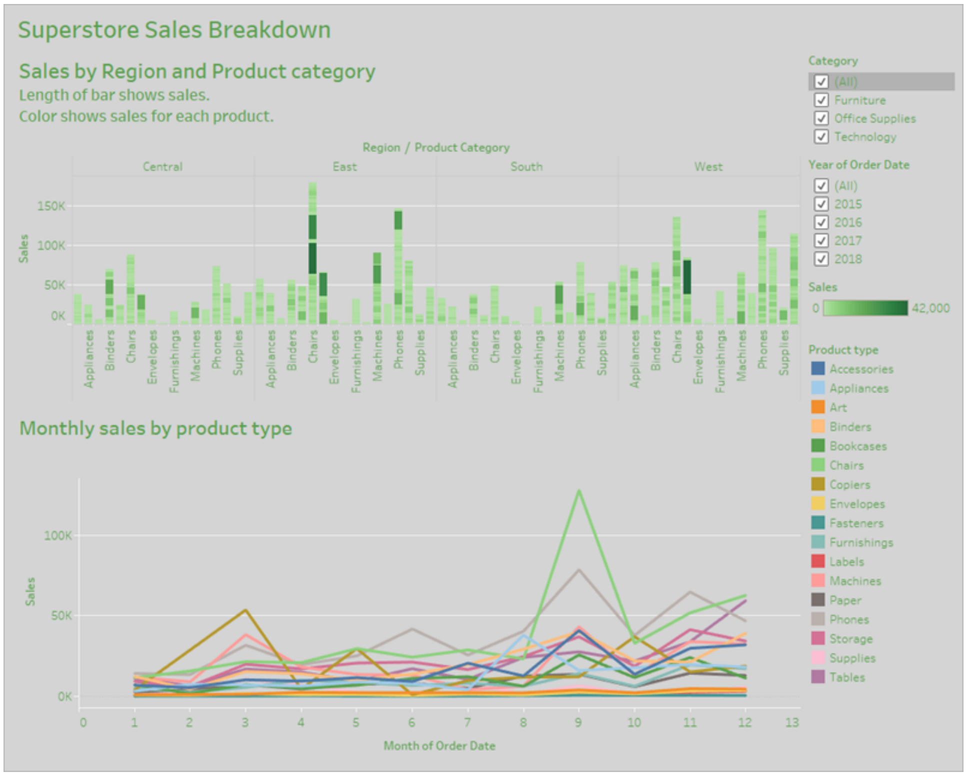
- 

For more information about views, see **Build Data Views for Accessibility** on page 1709

> **Note**: Interactivity within a view (mark selection, tooltips, etc) is not accessible.

Walkthrough: make existing dashboards more accessible

Let's start with a finished workbook and work backward to make it more accessible.



For this workbook to be more accessible, we will:

1. **Format for accessibility at the dashboard and workbook level** on the next page
2. **Aggregate and reduce marks** on page 2820
3. **Eliminate redundant information** on page 2822
4. **Use color and shape to differentiate marks** on page 2822
5. **Use filters to reduce the number of marks in a view** on page 2827
6. **Add captions and legends** on page 2831

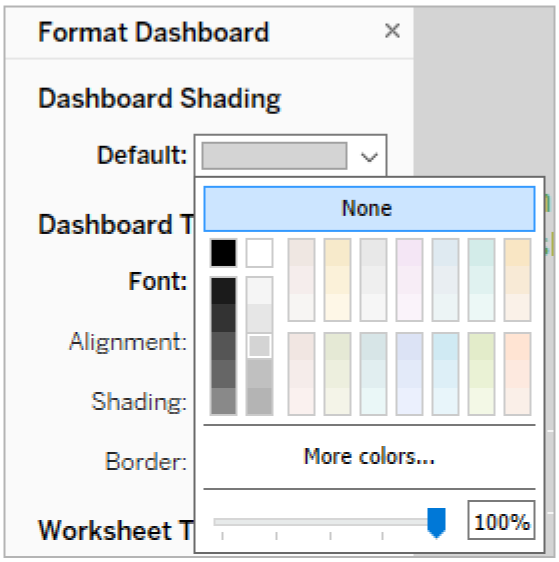
This dashboard is designed to show total sales for each sales region and product type. The line chart at the bottom shows monthly sales by product type. The author of this dashboard wanted to use color to brand the dashboard in the Superstore company's colors of green and gray.

Unfortunately, this dashboard has several features that pose accessibility problems for users, including the company color scheme. The steps that follow show a few ways to make this dashboard more accessible.

Format for accessibility at the dashboard and workbook level

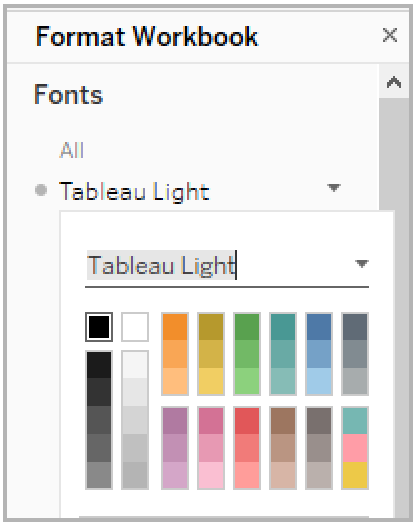
The green text on a gray background does not have a high enough contrast ratio to work for users with visual contrast sensitivity loss. To make this workbook accessible to all users and make the contrast ratio as high as possible, we're going to make the background white and change the text colors on our chart to black.

To update the dashboard shading color, change **Dashboard Shading** in the **Format>Dashboard** menu to **None**.



For text formatting changes, we could manually change the formatting on each worksheet. But that will take a lot of time. This dashboard only has two views, but dashboards often contain more.

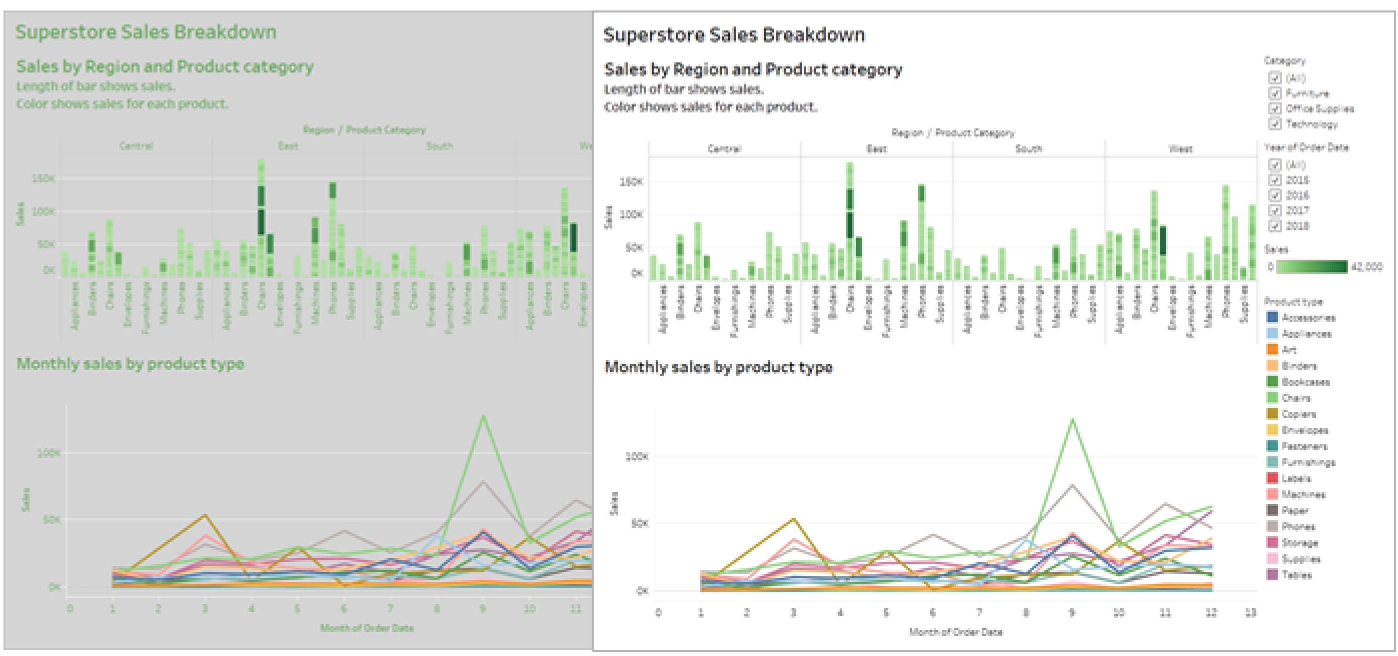
To update the text on our dashboard to black, select **Format>Workbook** and change **Fonts>All** to **Black**.



This changes the font color in every sheet of our workbook to black, which is then applied to the views included in the dashboard. For external information on color contrast, see Color Contrast and Why You Should Rethink It.

Less accessible - low contrast

More accessible - higher contrast



For more information about adding background colors around items, see Add padding, borders, and background colors around items.

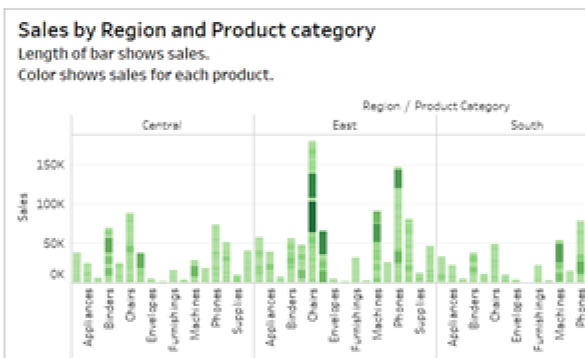
Aggregate and reduce marks

You might have a lot of information that you want to communicate with your dashboard or view. However, a dense view with a lot of marks, like the bar chart at the top of our view, can be difficult to understand using a screen reader or keyboard.

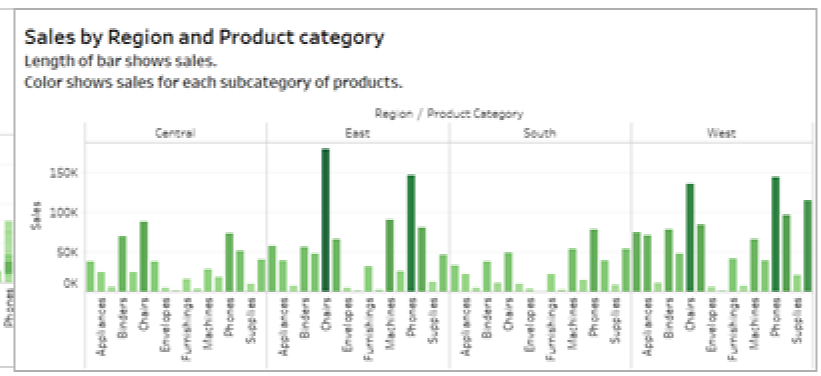
The top view in our dashboard displays over 5000 marks because each bar in the chart contains the name of each product in that category. If a view uses more than 1000 marks, it causes the view to be rendered on the server instead of the browser, and server-rendered views are not supported for WCAG conformance. Limit the number of marks to only those that emphasize the most important data points.

To make this view more accessible, make the stacked bar chart into a bar chart by removing the field **Product Name** from **Details** in the **Marks** card. This reduced the number of marks in the view from over 5000 to 68.

Less accessible - too many marks



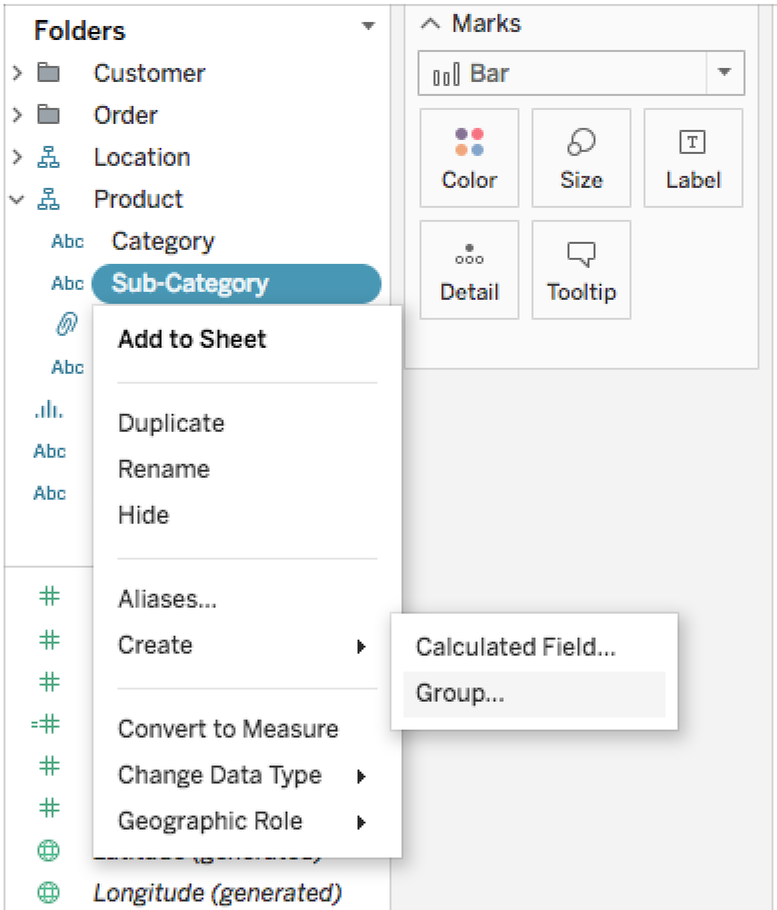
More accessible - aggregated view



Users can also access the **View Data** window (enabled by default) to review the underlying data for the marks or can download the data from that page to an accessible application to view it that way.

To reduce cognitive overload and reduce the numbers of marks in our field even further, we can create groups of related members in a field. To create a group of similar products:

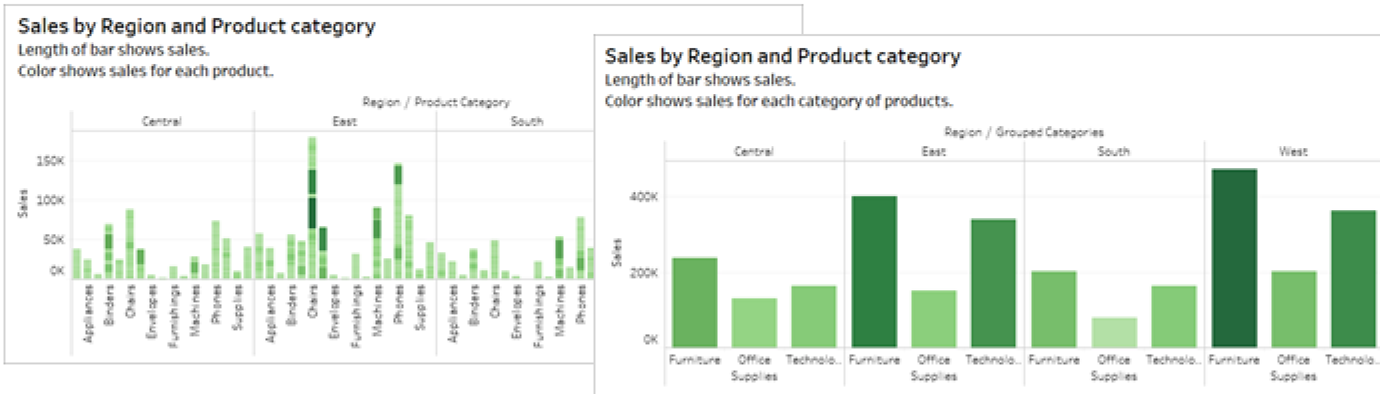
1. Right click on the field and select **Create** > **Group**.



2. In the Create Group dialog box, select several members that you want to group, and then click **Group**.

For more information on grouping, see **Group Your Data** on page 1173. Here's what the view looked like before grouping and aggregating, compared to after:

Less accessible - too many marks	**More accessible - aggregated and grouped view**

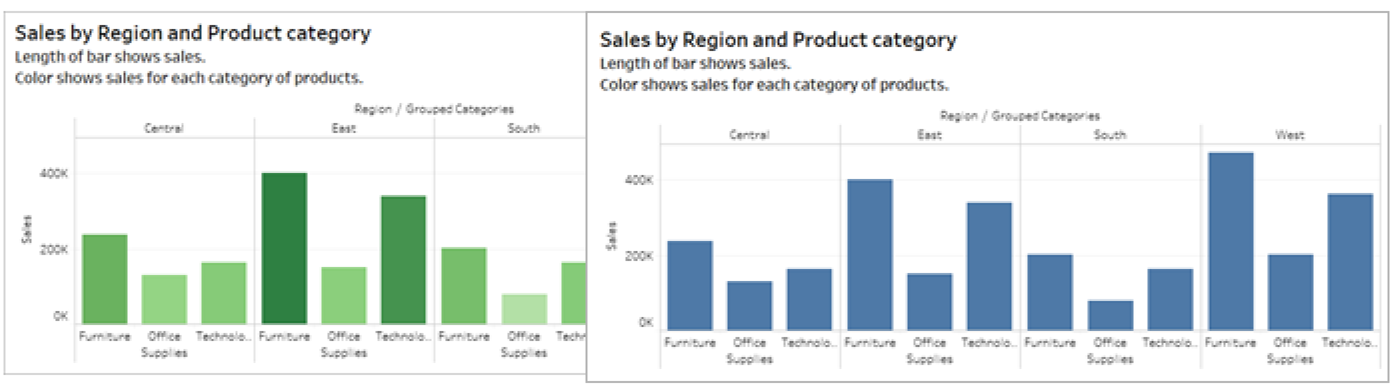


For information and examples about how to build this type of view, see **Keep it simple** on page 1719 in **Author Views for Accessibility** on page 1719.

Eliminate redundant information

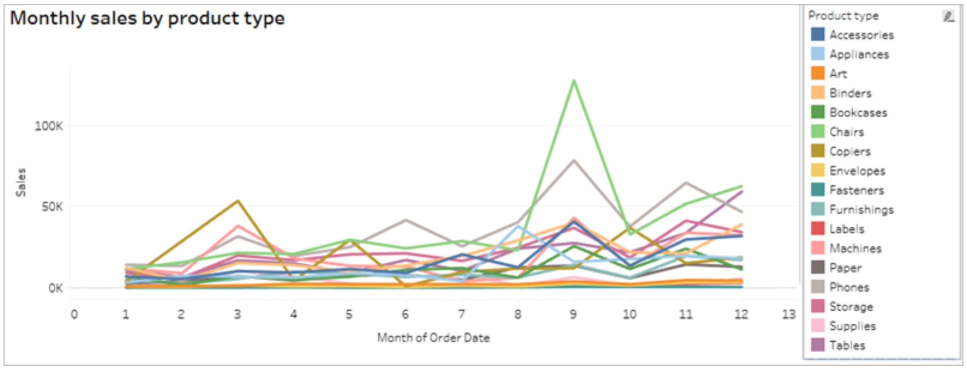
The aggregated and grouped bar chart reduces cognitive overload for users, but still uses color to convey meaning when color isn't necessary. Since the size of each bar already represents sales numbers, color can be removed from the chart by removing the sales field from Color in the **Marks** card. Removing color also removes the color legend that the screen reader reads.

Less accessible - unnecessary marks	**More accessible - unnecessary marks removed**



Use color and shape to differentiate marks

The line chart in this dashboard contains marks that are only distinguished by color, which makes it impossible or nearly impossible for people with colorblindness to distinguish marks from one another. The line path provides helpful position information, but the colors do not provide clear distinction between the line marks.

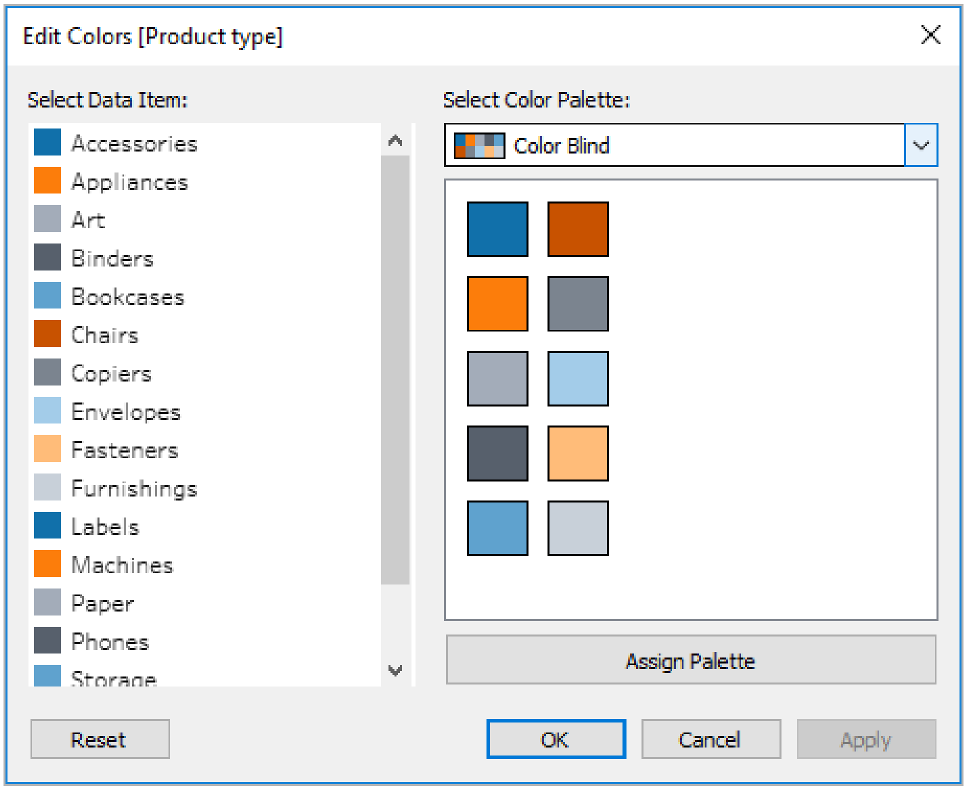


Make this line chart more accessible by applying the **Color Blind** palette and adding shapes to reinforce what is being shown in color.

To apply the Color Blind palette:

1. Select **Color** on the **Marks** card.

2. Select **Color Blind** from the **Color Palette** menu.

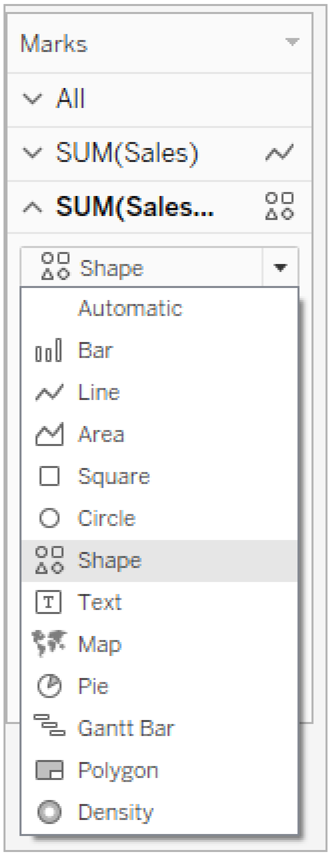


3. Select **Assign Palette** and **Apply** to apply it to the view.

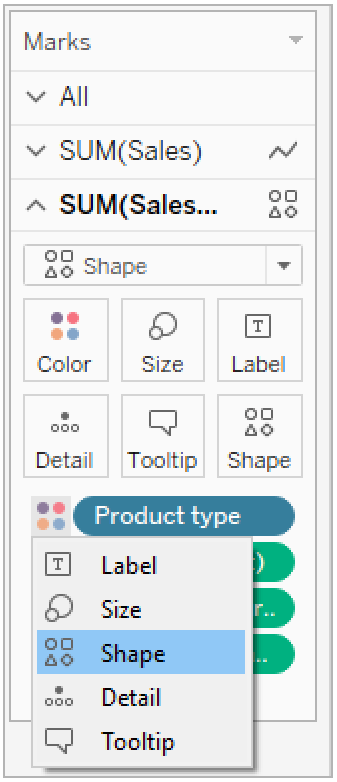
To add shapes in addition to color to your line chart, create a combination chart, or dual axis chart:

1. Control-drag (command-drag on a Mac) the field in the **Rows** shelf to copy it and place it to the right of the first field. This creates two tabs in your Marks card, one for each field in the Rows shelf.

2. On the marks card, select the bottom tab, titled **[Field Name (2)]**. Select **Shape** from the drop down menu in the Marks card.



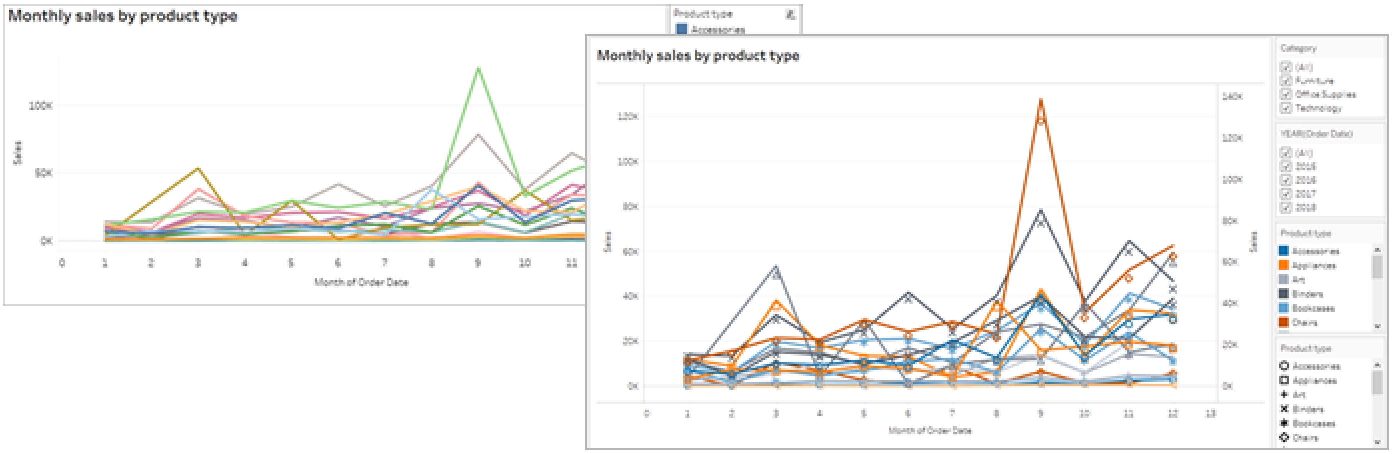
3. Select the Color mark to the right of the field name and select Shape.

4. 

This creates separate shapes for each of the marks in the field. At this point, there are two different charts; one a line chart of colors, another a scatter plot with shapes.

To merge these two charts together into a dual axis chart, in the Rows shelf, select the field on the right and select **Dual Axis**. For more information on combination charts, see **Build a Combination Chart** on page 1668.

Not easily accessible	**More accessible - Color Blind palette and shapes**



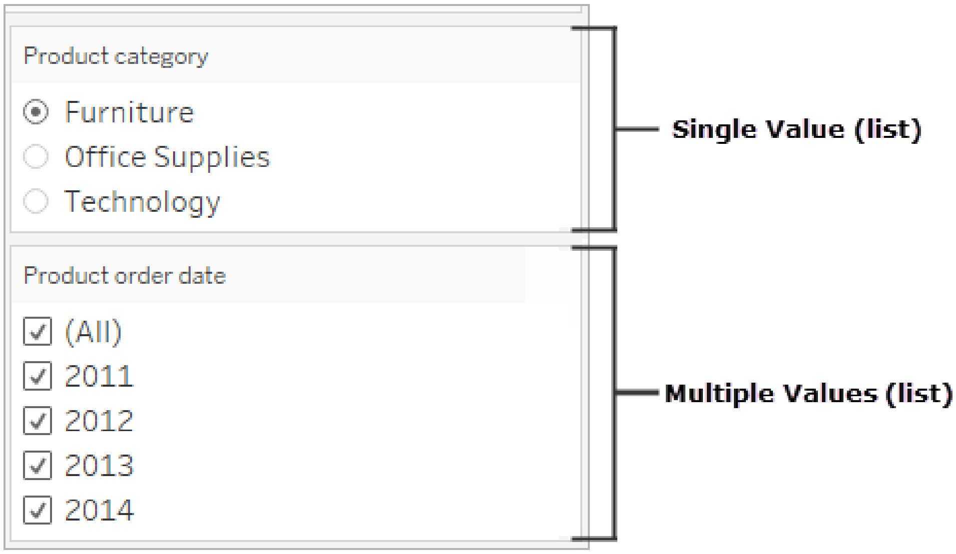
The chart is improved, but the line colors and shapes have started repeating after the 10th mark applied. This makes the view look busy and hard to make sense of, leading to cognitive overload. Let's fix it by adding filters and legends.

Use filters to reduce the number of marks in a view

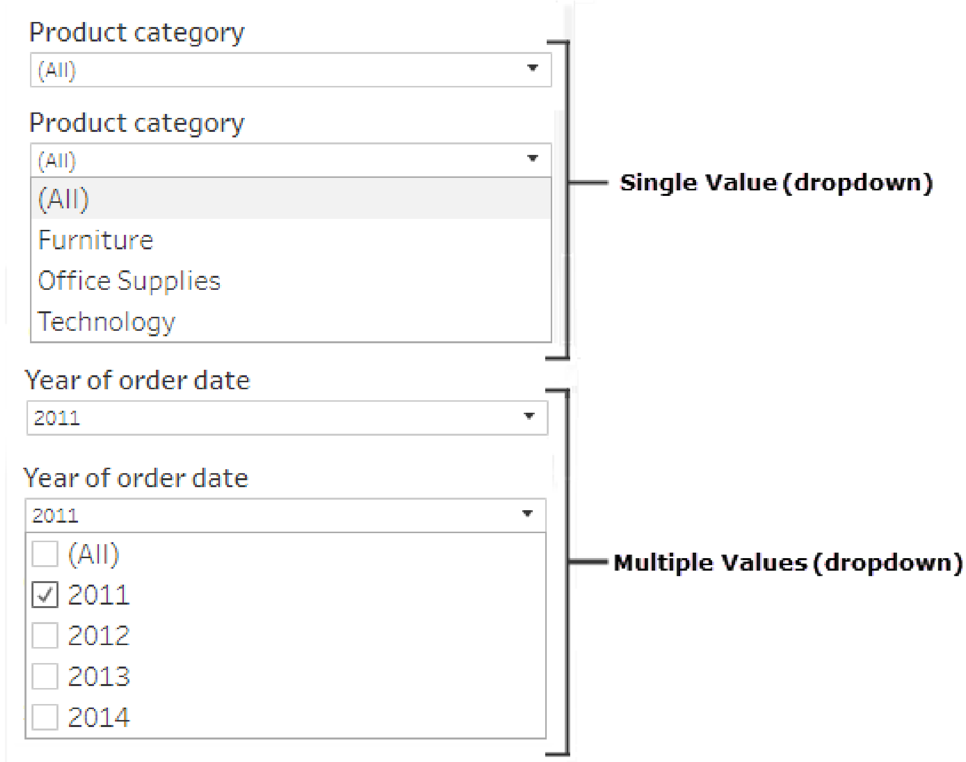
Using filters helps to focus the number of marks in the view to only what you want users to see.

You can show the following filter modes to allow users to control what data is in the view. The filter modes that Tableau currently supports for WCAG conformance are:

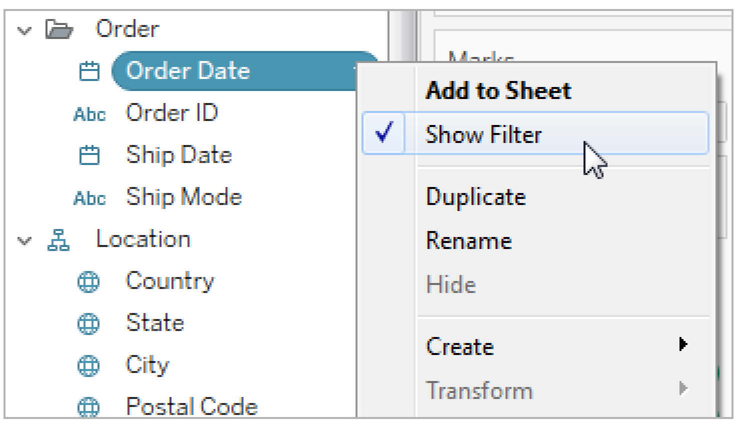
- **Single Value (list)**: a filter with radio buttons. Only one item can be selected at a time. Giving your users single-value filters is a great way to reduce the number of marks in the view.
- **Multiple Values (list)**: a filter with a list of items (with check boxes) that can be selected at the same time. Multiple-value filters will allow more marks to be shown at the same time.



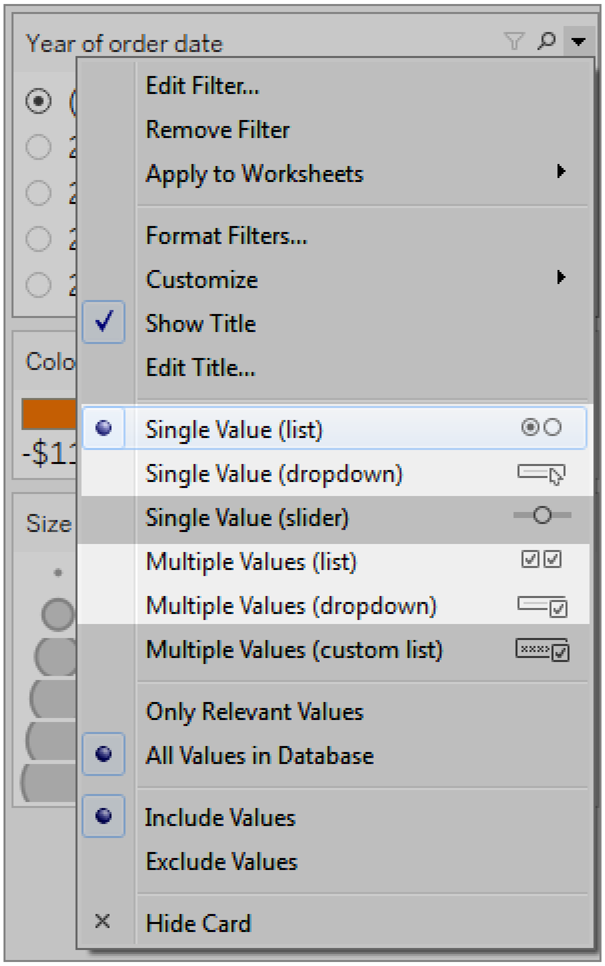
- **Single Value (drop-down)**: a filter with a dropdown list of items. Only one item can be selected at a time. Giving your users single-value filters is a great way to reduce the number of marks in the view.
- **Multiple Values (drop-down)**: a filter with a dropdown list of items that can be selected at the same time. Multiple-value filters will allow more marks to be shown at the same time.



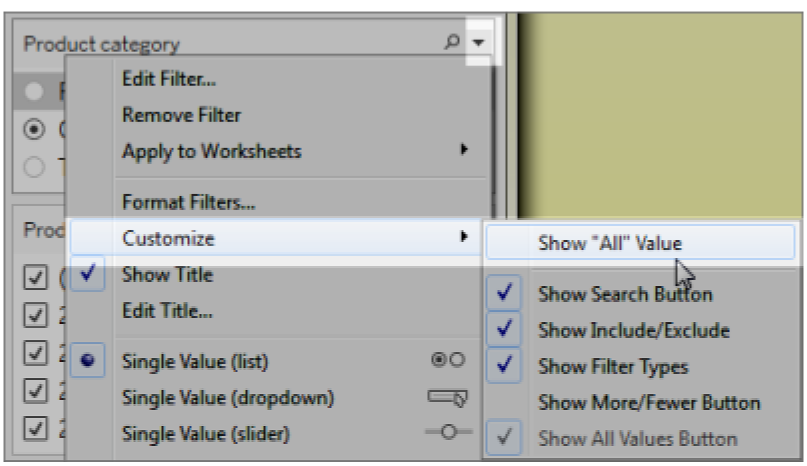
To show a filter, right-click the field you want to use as a filter, and then select **Show Filter**.



To select the filter mode, in the drop-down menu for a filter, select a **Single Value (list)**, **Single Value (dropdown)**, **Multiple Values (list)**, or **Multiple Values (dropdown)** filter.

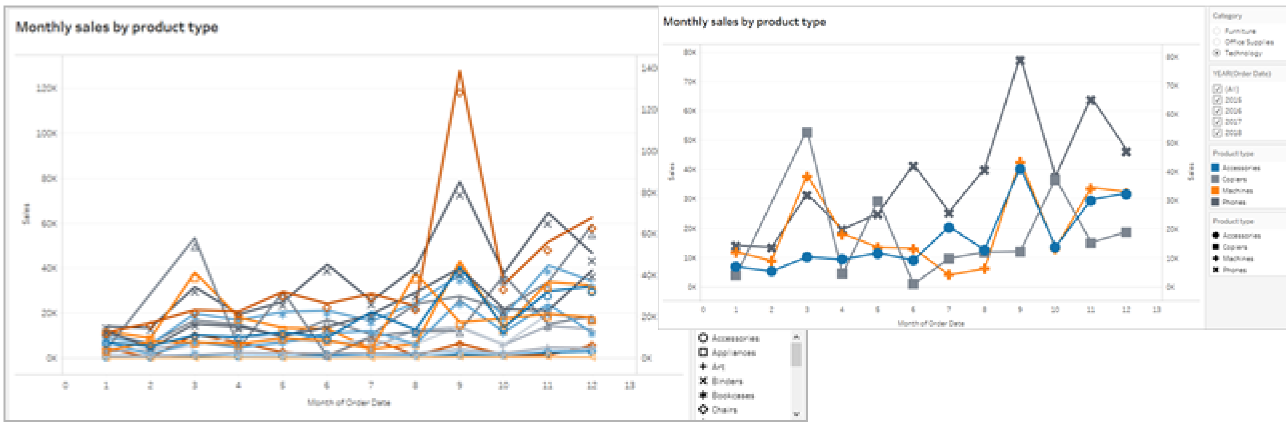


Because the line chart is a Single Value list, we can remove the **All** option from the filter. In the drop-down menu for the filter, select **Customize** and then clear the **Show "All" Value** option.



This reduces the level of marks in the view to prevent cognitive overload for dashboard consumers.

Less accessible - too many marks	**More accessible - less marks in the view**



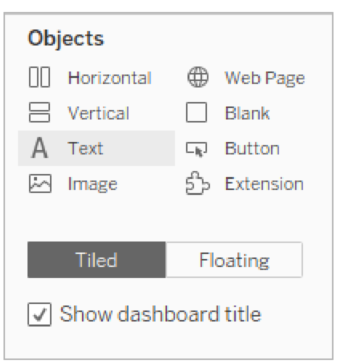
To learn more, see Use filters to reduce the number of marks in a view.

Another way to make the viz easier to consume is to

Add captions and legends

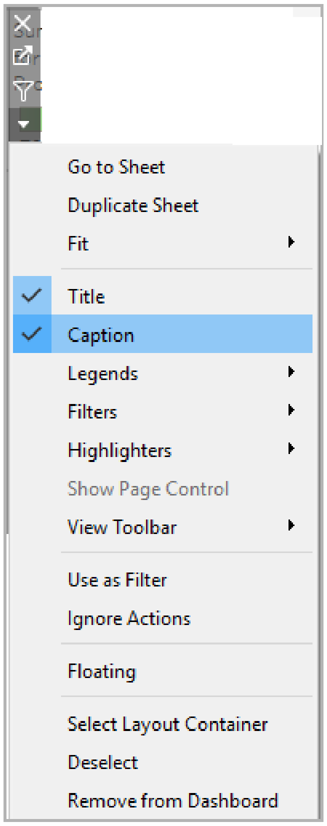
To add clarity for our users, we can add a **Text object** with a description of the dashboard and instructions about how to filter and use each view in the dashboard, including how to use the filters and categorical legends that we've added. To add a text object:

1. Grab Text from the Objects menu and drag it onto the dashboard.



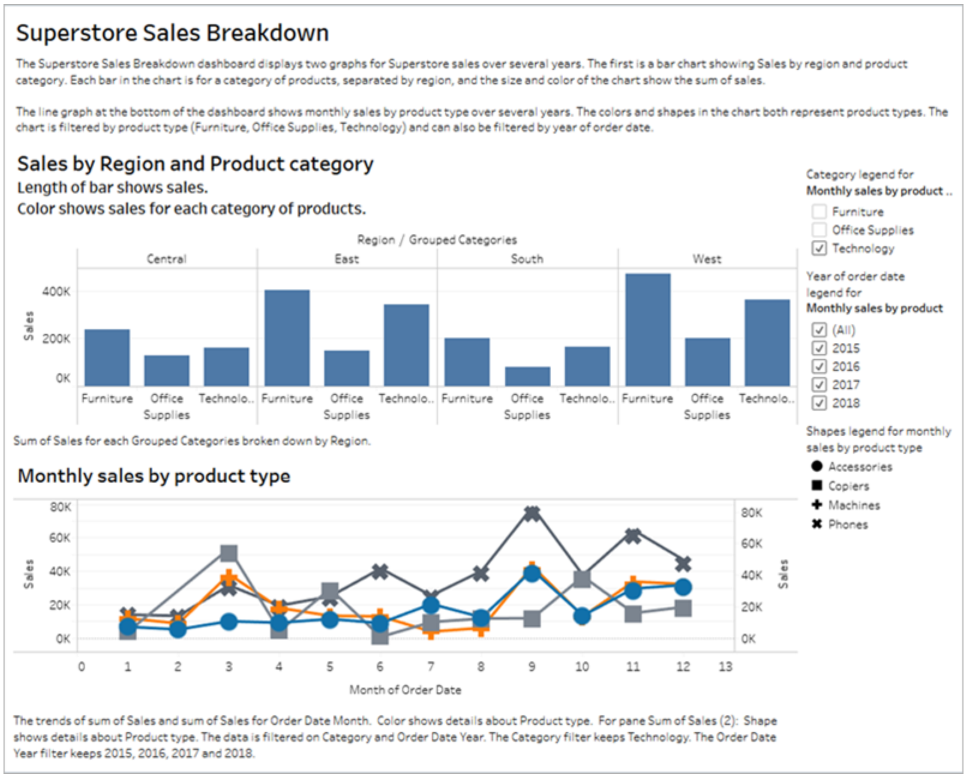
2. Use **Edit Text** window to add text that explains the dashboard's purpose and how to use it.

You can also add chart captions to the dashboard. The default caption text gives a summary of what is shown in the view, but you can add or edit this text to provide better context. To do this, click on a chart in the dashboard and select **Caption** from the **More Options**s drop-down menu.



Last, we can update the titles of our legends to reference their associated charts and reduce the amount of redundant text that screen readers consume. To do this, double-click on the legend title, or right-click and select **Edit Title** to add or edit text in the **Edit Filter Title** window.

The result is a dashboard that includes an explanation of the dashboard and chart captions and eliminates redundant text for users.



Publish and embed the dashboard

After you create your views, to make them WCAG-conformant, you must:

- Publish views to Tableau Server or Tableau Online, and then embed the view in a WCAG-compliant web page. For information about how to publish a view, see **Comprehensive Steps to Publish a Workbook** on page 3058.
- Web editing is not supported in WCAG-conformant views, so Web Edit permissions must be turned off in the published workbook.
- In the embedded views, hide the Custom Views, Subscriptions, and Alerts buttons in the toolbar.
- Make sure users have permissions to access to the embedded views and are able to view the underlying data in the View Data window.

For information about each of these steps, see Publish and embed the view in Author Views for Accessibility.

Manage Sheets in Dashboards and Stories

Hiding, showing, and navigating to sheets helps you more easily manage workbooks and design dashboards.

Hide and show sheets

To streamline the editing process for large workbooks, hide and show sheets as you work.

> For Viz in Tooltip, see **Hide or show a Viz in Tooltip worksheet** on page 1483.

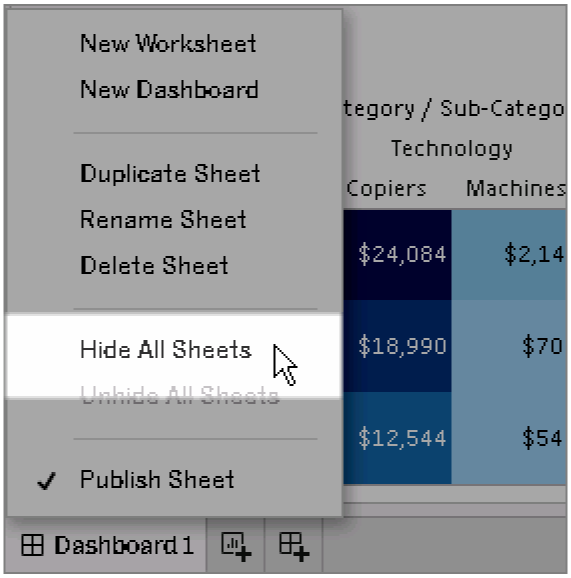
What sheets can I hide?

While editing a workbook, you can hide any sheet contained in a dashboard or story. (That includes dashboard sheets in stories.)

Sheets that aren't in a dashboard or story are always visible while editing workbooks. However, you can hide any sheet in workbooks you publish to Tableau Online or Tableau Desktop. See **Comprehensive Steps to Publish a Workbook** on page 3058.

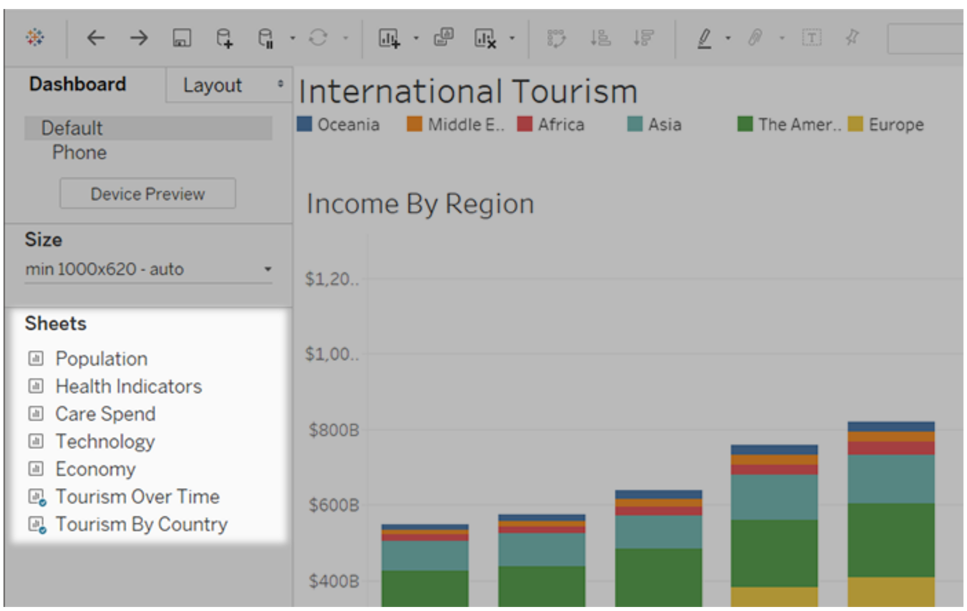
Hide or unhide all sheets

To hide all of a dashboard or story's sheets, right-click (Windows) or Control-click (macOS) the dashboard's tab at the bottom of the screen, and select **Hide All Sheets**. You can later select **Unhide All Sheets** if needed.

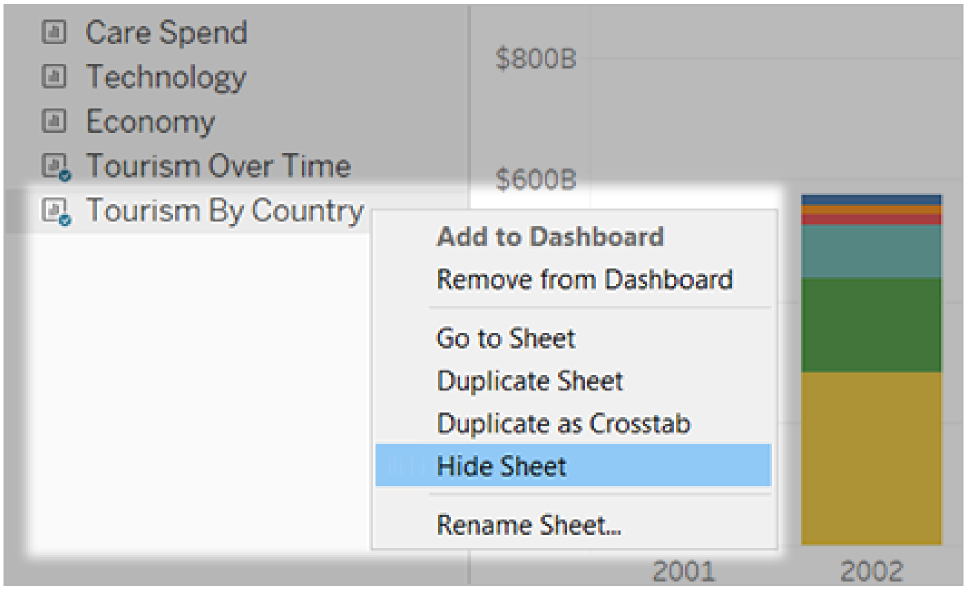


Hide an individual sheet

1. Look for the sheet in the tabs at the bottom of the screen. Or, if you're viewing a dashboard or story, look in the **Sheets** list at left:



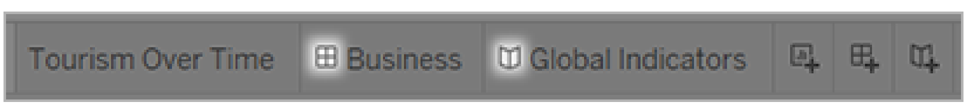
2. Right-click (Windows) or Control-click (macOS) the sheet name, and select **Hide Sheet**.



Note: From the sheets list to the left of a dashboard or story, you can hide only sheets used in that particular dashboard or story. You can identify these sheets by the blue checkmark on the sheet icon.

Unhide an individual sheet

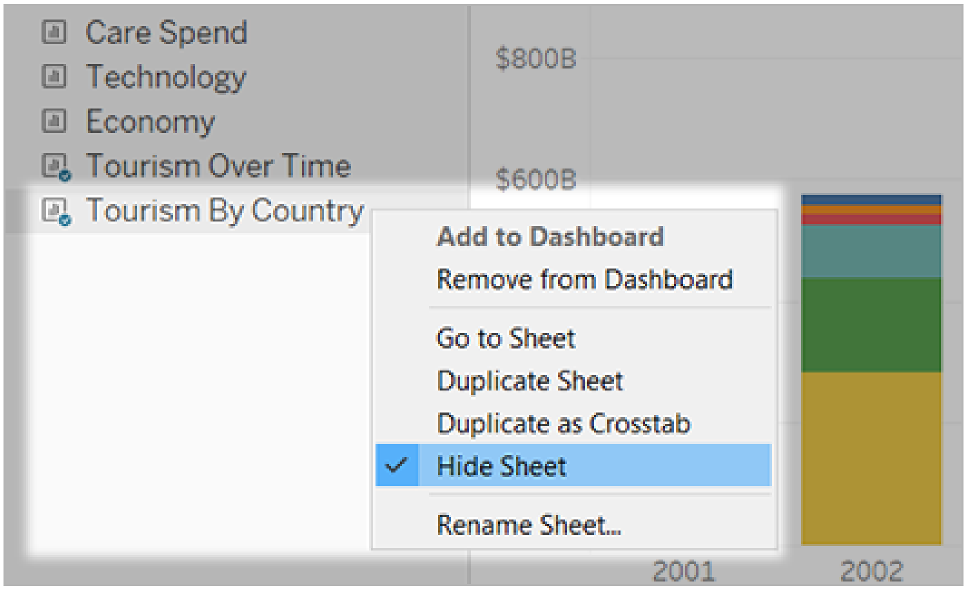
1. At the bottom of the screen, click the tab for the dashboard or story that contains the sheet. You can identify dashboards and stories by grid and book icons, respectively.



Note: If a story has a hidden dashboard that in turn has a hidden sheet, first go to the hidden dashboard from the story, and then to the hidden sheet from the dashboard.

2. In the list at left, right-click (Windows) or Control-click (macOS) the sheet, and clear the

Hide Sheet checkbox.



Navigate between sheets, dashboards, and stories

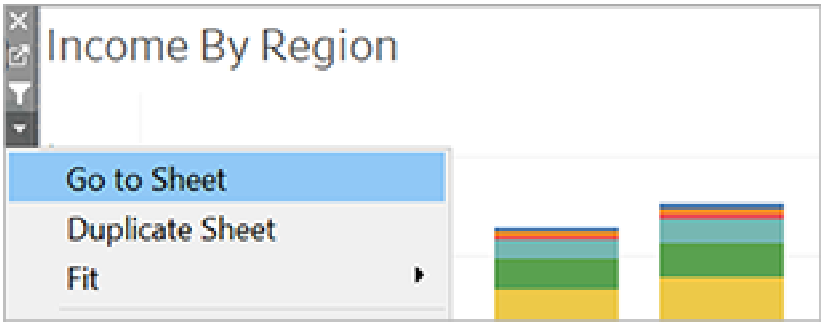
Tableau lets you quickly navigate back and forth between sheets and related dashboards and stories, helping you better optimize the design of each.

Go from a dashboard or story to a sheet it contains

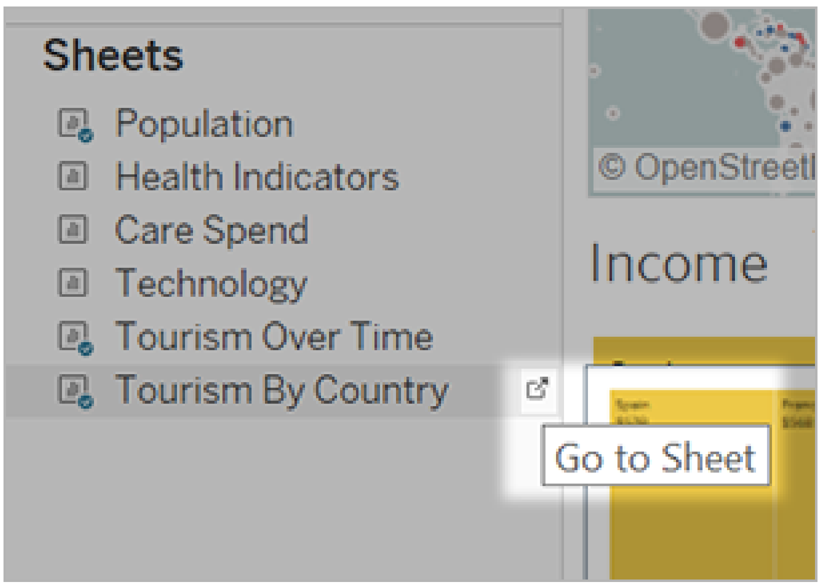
Tip: If a sheet is hidden, this technique temporarily shows it so you can get a closer look at it. The sheet is hidden again when you switch to another sheet.

Do either of the following:

- On the dashboard itself, select the item that references the hidden sheet. Then click the drop-down arrow in the top left or right corner, and select **Go to Sheet**.

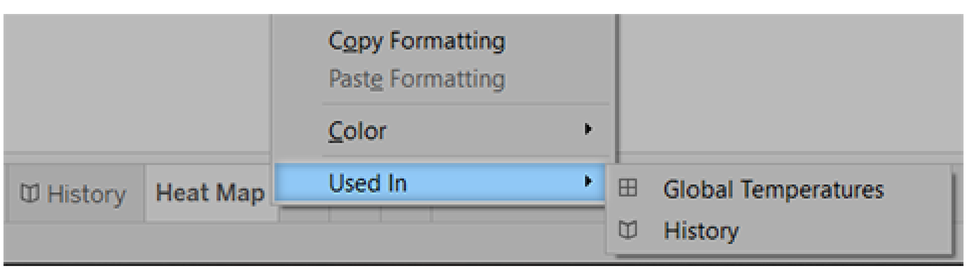


- In the **Sheets** list to the left of the dashboard, click the icon to the right of the sheet name.



Go from a sheet to related dashboards and stories

To quickly go to dashboards and stories a sheet appears in, right-click the sheet tab and open the **Used in** menu.



Use Dashboard Extensions

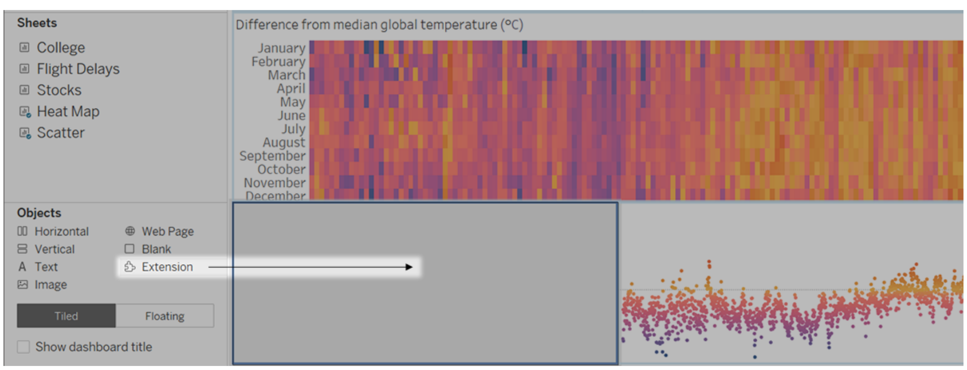
Extensions let you add unique features to dashboards or directly integrate them with applications outside Tableau. Adding extensions is easy; you incorporate them into dashboard layouts just like other dashboard objects.

Extensions expand dashboard functionality with the help of web applications created by third-party developers. If you're a developer and want to create your own extensions, see the Tableau Extensions API documentation on GitHub.

> **Note**: Tableau administrators can turn off dashboard extensions for Tableau Desktop, Tableau Server, and Tableau Online.

Add an extension to a dashboard

1. In a Tableau workbook, open a dashboard sheet.
2. From the **Objects** section, drag **Extension** to the dashboard.



3. In the “Add an Extension” dialog box, do either of the following:
   - Search for and select an extension.
   - Click **Access Local Extensions**, and navigate to a .trex file you previously downloaded.
4. If prompted, allow or deny the dashboard extension access to data in the workbook. For more information, see **Data security, Network-enabled, and Sandboxed extensions** on page 2841.

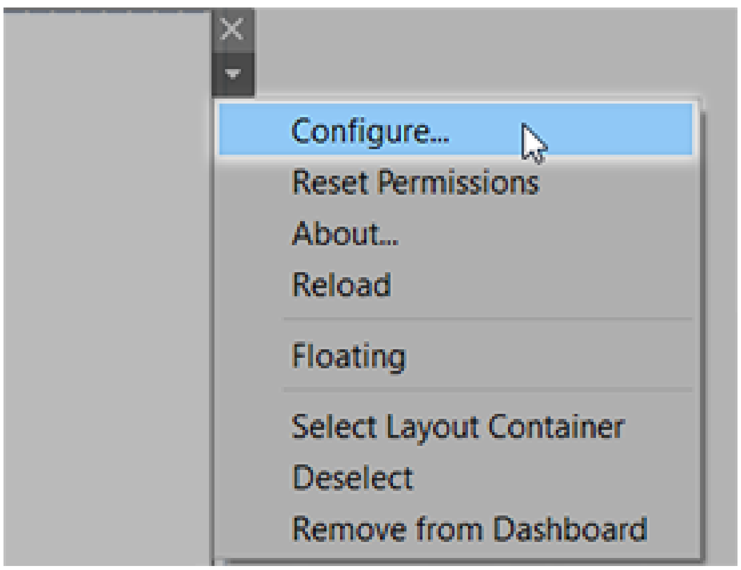
If you allow access, follow any on screen instructions for configuring the extension.

> **Note:** If you're using Tableau Server or Tableau Online, Extension objects will appear blank in prints, PDFs, and images of dashboards (including images in subscription emails).

Configure a dashboard extension

Some dashboard extensions provide configuration options that let you customize features.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Configure**.
2. Follow the on-screen instructions to configure the extension.



Reload a dashboard extension

If a dashboard extension becomes unresponsive, you might need to reload it, which is similar to refreshing a web page in a browser.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reload**.

   The dashboard extension is refreshed and set to its original state.

2. If reloading the extension fails to return it to a useable state, try removing it from the dashboard and adding it again.

Data security, Network-enabled, and Sandboxed extensions

Dashboard extensions are web applications that come in two forms:

- *Network-enabled extensions* run on web servers located outside of your local network.
- *Sandboxed extensions* run in a protected environment without access to any other resource or service on the web.

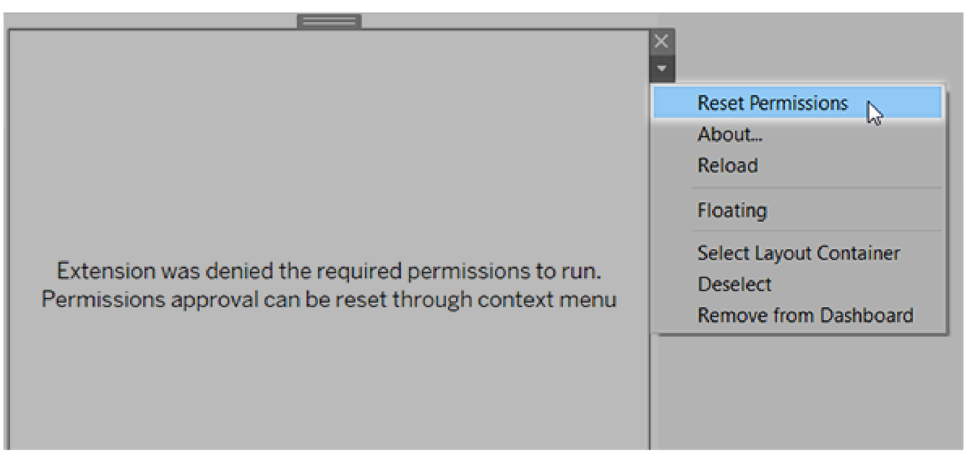
Before adding a Network-enabled extension or viewing a dashboard with one, be certain that you trust the website that hosts it. By default, dashboard extensions use the HTTPS protocol, which guarantees an encrypted channel for sending and receiving data, and ensures some privacy and security.

For more information about data security when using dashboard extensions, see Extension Security - Best Practices for Deployment.

Allow or deny data access to a Network-enabled extension

Depending on how an extension is designed, it can access either visible data in a view, or full underlying data, table and field names from data sources, and information about data source connections. When you add an extension, or view a dashboard with one, you're given an opportunity to allow or deny the extension to run and access this data.

If you're viewing a dashboard with an extension that requires full data access, and that access has been denied, a message appears in place of the extension. If you trust the extension and want to use it, you can reset permissions and allow the extension to run.



1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reset Permissions**.
2. Click either **Allow** to let the extension run and access data, or **Deny** to prevent the extension from running.

Ensure that JavaScript is enabled in Tableau Desktop

Dashboard extensions interact with data using the Tableau Extensions API library, a JavaScript library. If you want to use extensions, be sure that JavaScript is enabled in the dashboard security settings:

Choose **Help** > **Settings and Performance** > **Set Dashboard Web View Security** > **Enable JavaScript**.

Ensure that extensions run on Tableau Online or Tableau Server

You can add extensions to workbooks you publish from Tableau Desktop or directly in the web-authoring mode of Tableau Online and Tableau Server. A Tableau administrator must allow extensions to run on a site and add Network-enabled extensions to a safe list. Administrators should only allow extensions that you have tested and trust.

If you want to use a dashboard extension on Tableau Online or Tableau Server, direct your administrator to Manage Dashboard Extensions in Tableau Online or Manage Dashboard Extensions in Tableau Server.

Supported web browsers for Sandboxed extensions

Sandboxed extensions run in all browsers supported Tableau Server and Tableau Online except Internet Explorer 11.

Supported versions of Tableau Server for Sandboxed extensions

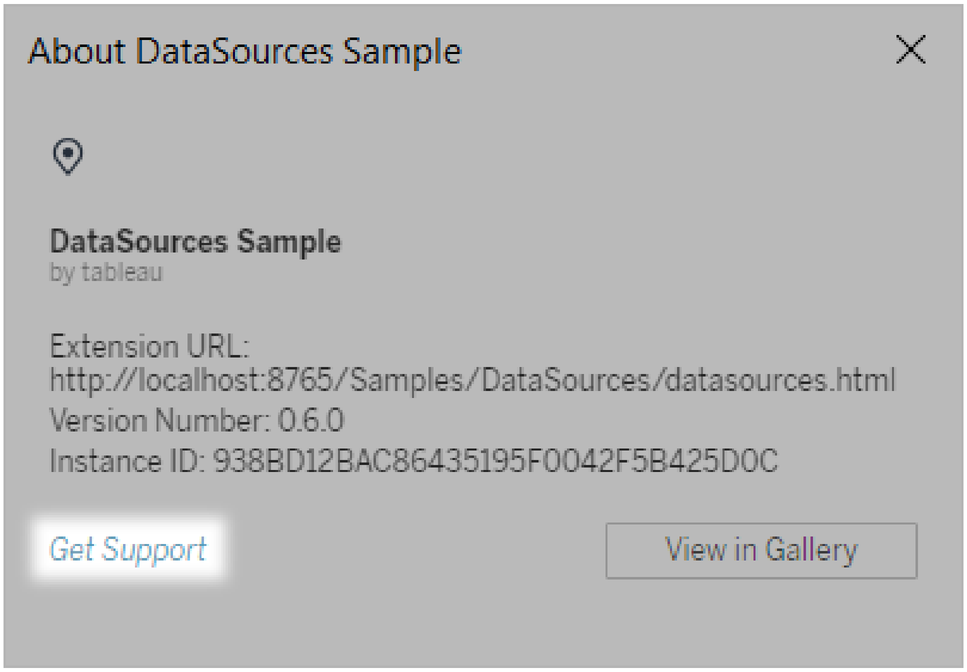
You can use Sandboxed extensions in Tableau Server 2019.4 and later.

Get support for dashboard extensions

To get help for an extension, you'll need to contact the developer or company who created it.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **About.**

2. Click **Get Support** to go to the support page of the extension developer.



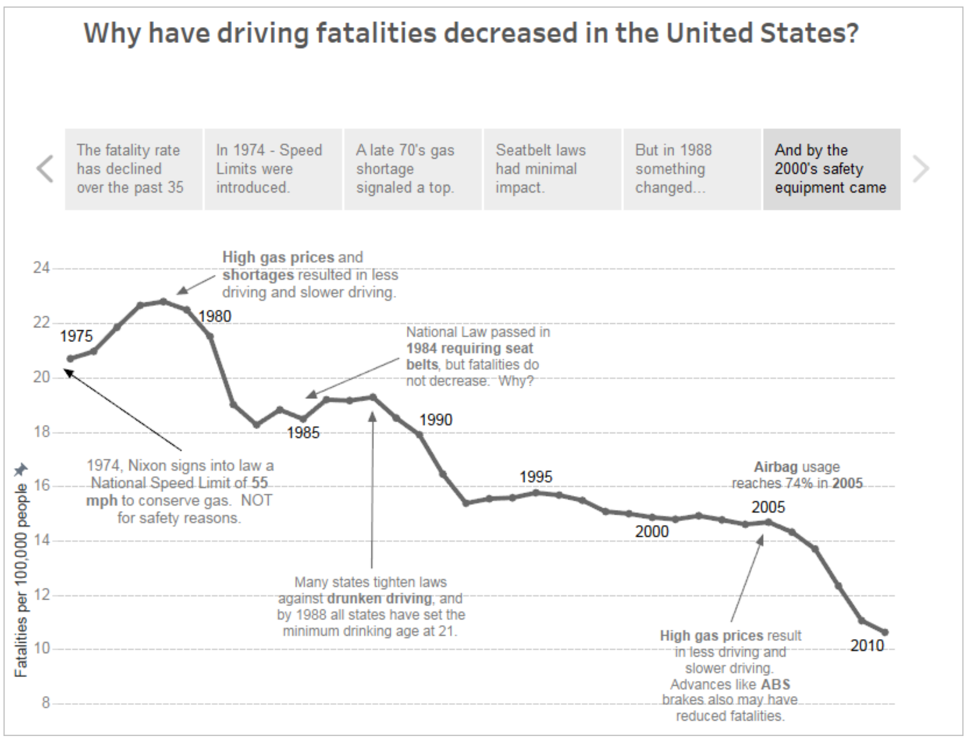
Note: Tableau doesn't provide support for extensions or for other programs that interface with the Extensions API. However, you can submit questions and ask for help in the Tableau developer community.

Stories

In Tableau, a **story** is a sequence of visualizations that work together to convey information. You can create stories to tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or to simply make a compelling case.

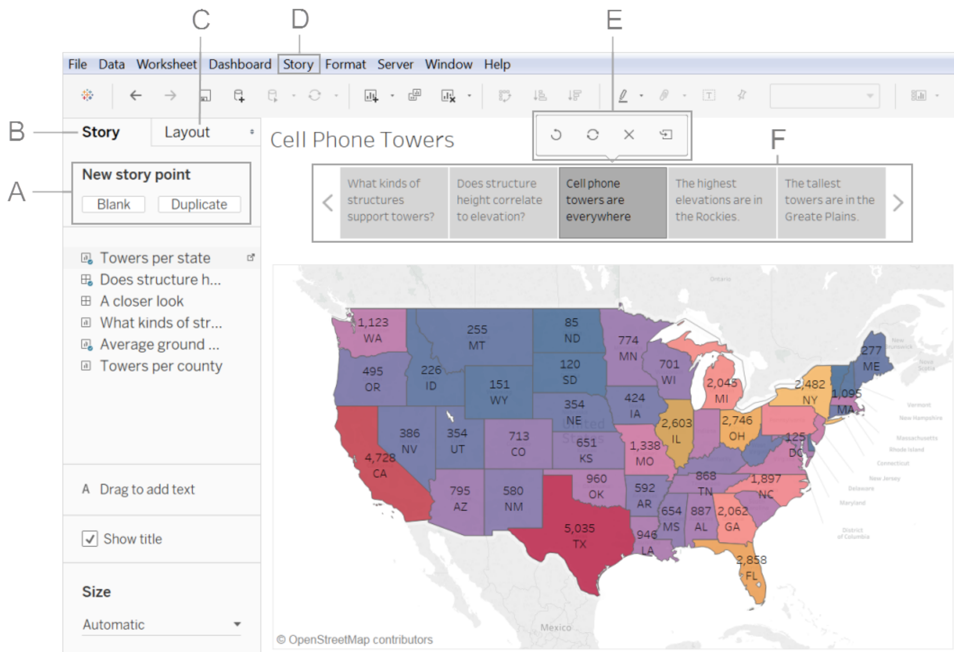
A story is a sheet, so the methods you use to create, name, and manage worksheets and dashboards also apply to stories (for more details, see **Workbooks and Sheets** on page 261). At the same time, a story is also a collection of sheets, arranged in a sequence. Each individual sheet in a story is called a **story point**.

When you share a story —for example, by publishing a workbook to Tableau Public, Tableau Server, or Tableau Online—users can interact with the story to reveal new findings or ask new questions of the data.



The Story Workspace

As you work on a story, you can use the following controls, elements, and features.



A. Options for adding a new story point: Choose **Blank** to add a new point or **Duplicate** to use the current story point as the starting place for your next point.

B. The Story pane: Use this pane to drag dashboards, sheets, and text descriptions to your story sheet. This is also where you set the size of your story and display or hide the title.

C. The Layout pane: This is where you choose your navigator style and display or hide the forward and back arrows.

D. The Story menu: Use this menu in Tableau Desktop to format the story or copy or export the current story point as an image. You can also clear the entire story here or show or hide the navigator and story title.

E. The Story toolbar: This toolbar appears when you mouse-over the navigator area. Use it to revert changes, apply updates to a story point, delete a story point, or create a new story point out of the current, customized one.

F. The navigator: The navigator allows you to edit and organize your story points. It's also how your audience will step through your story. To change the style of the navigator, use the Layout pane.

For more information on these options, see **Create a Story** on page 2852.

Best Practices for Telling Great Stories

A good data story brings data and facts to life. Use this article for tips on best practices to use with Tableau's story points feature.

What's your story's purpose?

Before you start to build your story, take some time to think about the purpose of your story and what you want your viewers' journey to be. Is it a call to action, is it a simple narrative, or are you presenting a case?

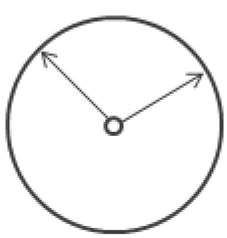
If you're presenting a case, decide whether you want to present data points that lead up to a conclusion at the end, or start with a conclusion then show the supporting data points. The latter approach works well for a busy audience.

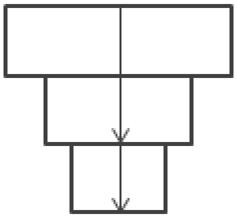
Finally, sketching out your story first on paper or a whiteboard can help you quickly identify problems with your sequence.

The seven types of data stories

When you use the story feature, you are building a sequence of points. Each point can contain a view, dashboard, or even just text. Some stories show the same view throughout the story, with text annotations and different filters applied to different points to support the narrative arc.

The following table describes seven different data story approaches you can take and provides an example for each. Each data story type is also illustrated in the Data Story Examples Workbook on Tableau Public. A single story can also use more than one approach—see **Example - A Story That Examines a Trend** on page 2861.

Data Story Type	Description
 Change Over Time	**What it does**: Uses a chronology to illustrate a trend. **Discussions it starts**: Why did this happen, or why does it keep happening? What can we do prevent or make this happen? **Example**: Arsenal's Injury Crisis

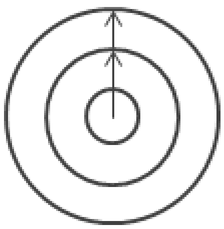


Drill Down

What it does: Sets context so that your audience better understands what's going on in a particular category.

Discussions it starts: Why is this person, place or thing different? How does the performance of this person, place, or thing compare?

Examples: Tell Me About Will, The Simpsons Vizipedia

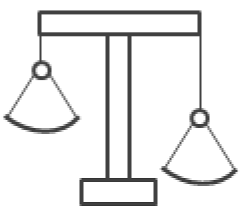


Zoom Out

What it does: Describes how something your audience cares about relates to the bigger picture.

Discussion it starts: How does something you care about compare to the bigger picture? What effect does one area have on the bigger picture?

Example: Vancouver Cyclists

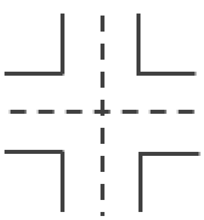


Contrast

What it does: Shows how two or more subjects differ.

Discussions it starts: Why are these items different? How can we make A perform like B? Which area should we focus on and which area is doing fine?

Example: The Pyramids of Egypt

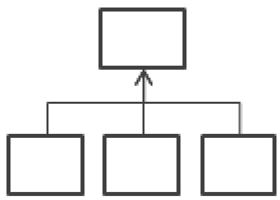


Intersections

What it does: Highlights important shifts when one category overtakes another.

Discussions it starts: What causes these shifts? Are these shifts good or bad? How do these shifts affect other aspects of our plan?

Example: US vs. THEM

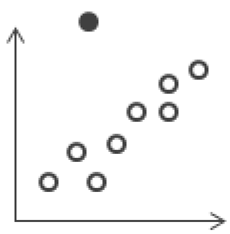


Factors

What it does: Explains a subject by dividing it into types or categories.

Discussions it starts: Is there a particular category we should focus on more? How much do these items affect the metric we care about?

Example: Planet Earth



Outliers

What it does: Shows anomalies or where things are exceptionally different.

Discussions it starts: Why is this item different?

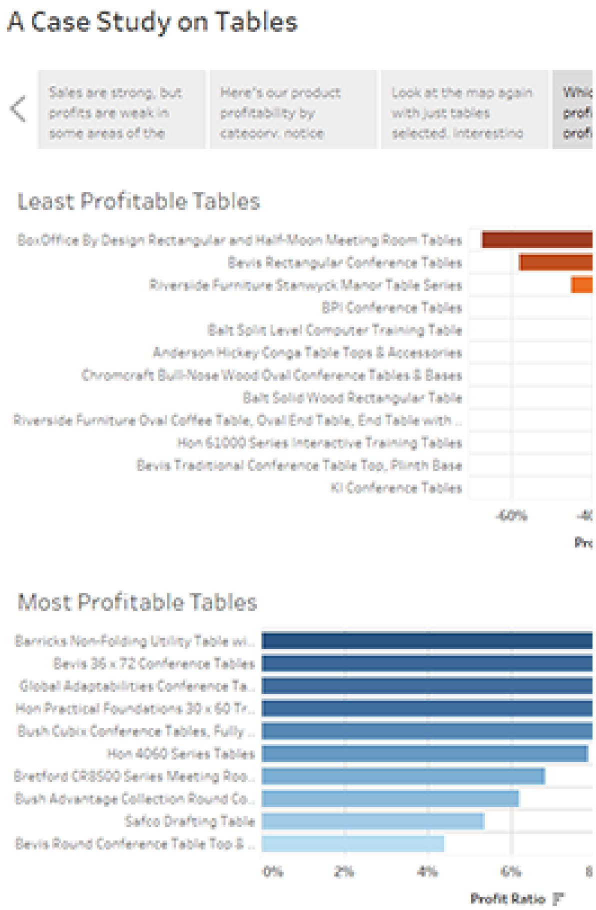
Example: SOS Children's Villages

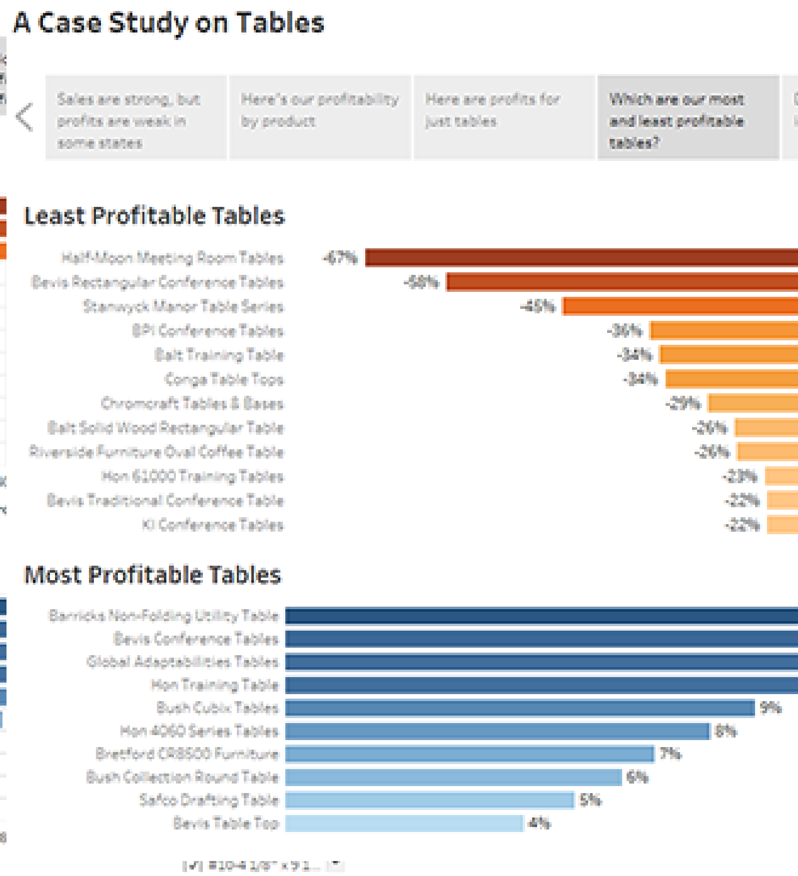
Keep it simple

A common error is trying to cram too many views and dashboards into a single story. The result is too many points for your viewers to take in.

The clarity of each story point is also important. Take a step back and consider your story from the perspective of someone who's never seen it. Every element should serve a purpose. If captions, titles, legends, or grid lines aren't necessary, get rid of them!

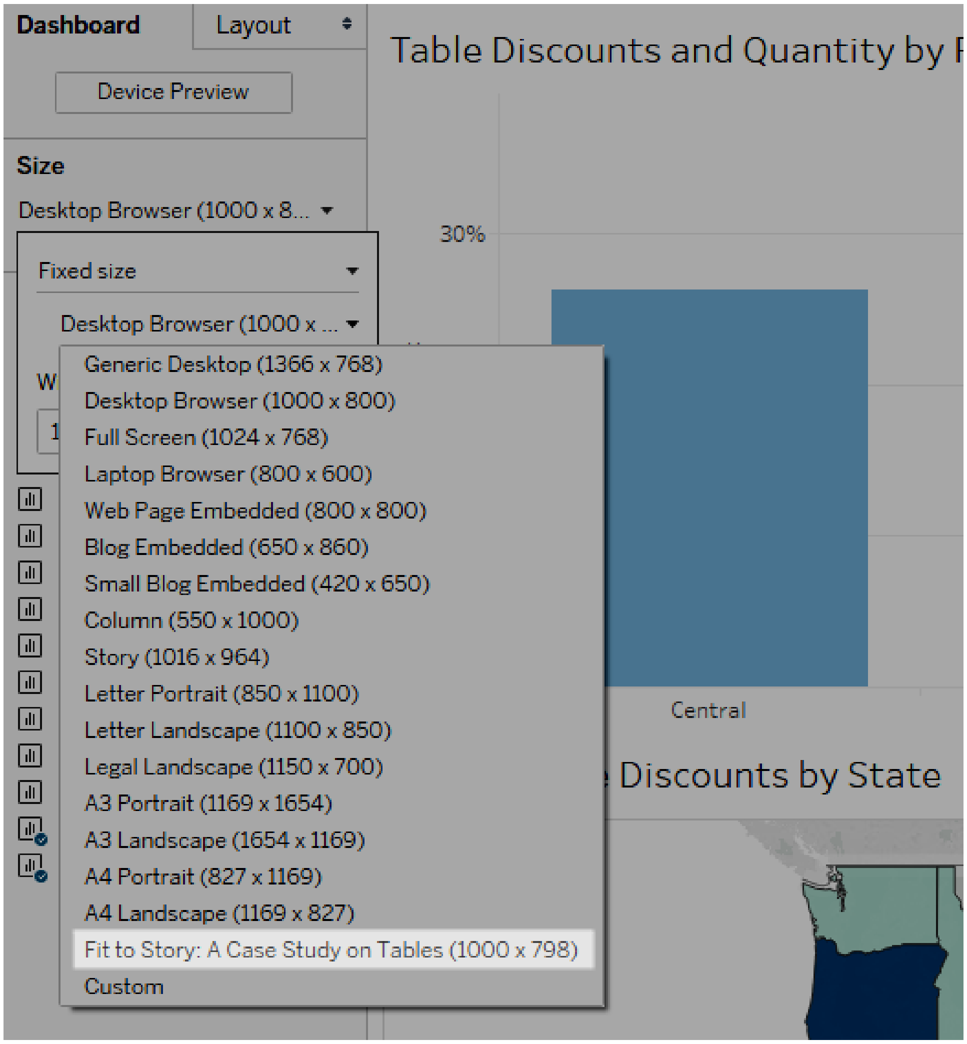
Before	**After**





Use 'Fit to' in your dashboards

Dashboards are a common ingredient in Tableau stories. For dashboards that you plan to include in your story, you can use the **Fit to** option under **Size** on the Dashboard pane. It will resize your dashboard so that it's the right size for the story you're creating.



Plan for fast load times

The most wonderful story in the world won't have much impact if it takes too long to load once it's published. People find long waits frustrating.

Filtering is a common culprit for slow load times. Although filters are effective in restricting the amount of data being analyzed, they also impact query performance. For example, **Exclude** filters tend to be slower than **Keep Only** filters. This is because **Exclude** filters load all of the data for a dimension instead of just what you want to keep. See **Filter Your Data Carefully** on page 3000 in the Tableau Desktop online help for more details. Knowing **Tableau's Order of Operations** on page 330 can also shave time off your load times.

Some of the most critical performance decisions you make as an author begin before you even create your first view or story, in the data preparation stage. Take a moment to familiarize yourself with the data you're working with. See **Know Your Data at the Database Level** on page 2994 and **Test Your Data and Use Extracts** on page 2996 in the Tableau Desktop online help for tips on what to look for and smart ways to work with your data.

Create a Story

Use stories to make your case more compelling by showing how facts are connected, and how decisions relate to outcomes. You can then publish your story to the web, or present it to an audience.

Each story point can be based on a different view or dashboard, or the entire story can be based on the same visualization seen at different stages, with different filters and annotations.

Create a story point

1. Click the **New Story** tab.

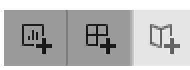
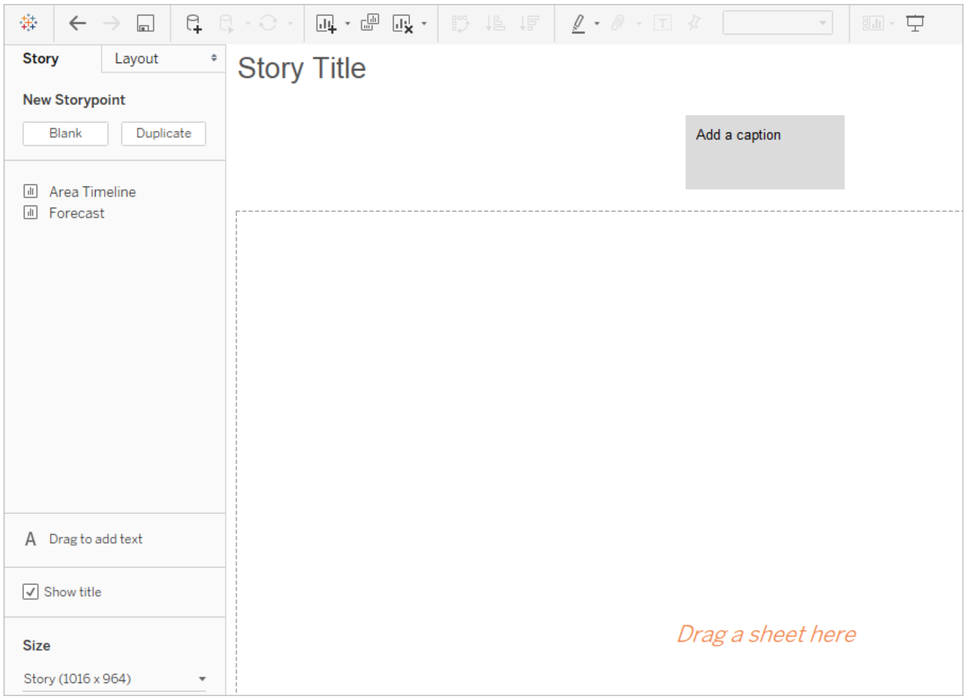
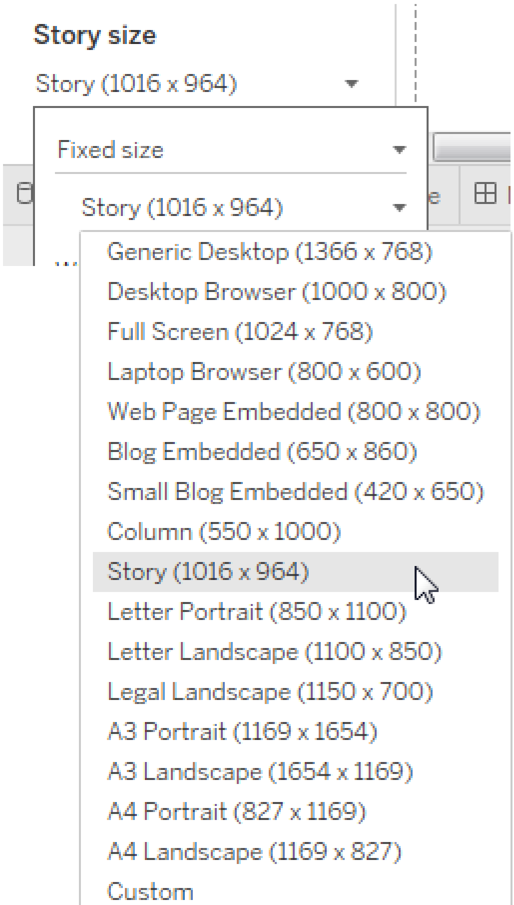
   

   Tableau opens a new story as your starting point:



2. In the lower-left corner of the screen, choose a size for your story. Choose from one of the predefined sizes, or set a custom size, in pixels:



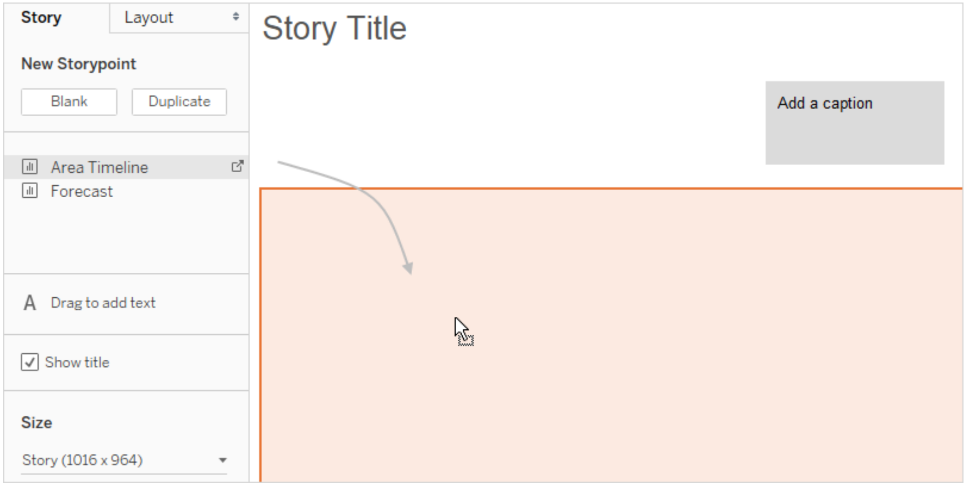
Note: Choose the size your story will be viewed at, not the size you're authoring in.

3. By default, your story gets its title from the sheet name. To edit it, right-click the sheet tab, and choose **Rename Sheet**.

   If you're using Tableau Desktop, you can also rename a story by double-clicking the title.

4. To start building your story, double-click a sheet on the left to add it to a story point.

   In Tableau Desktop, you can also drag sheets into your story point.



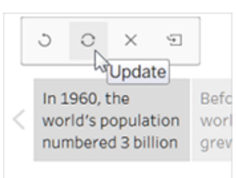
When you add a sheet to a story point, that sheet remains connected to the original sheet. If you modify the original sheet, your changes will automatically be reflected on the story points that use it.

> If you are using Tableau Server or Tableau Online to author on the web and the original sheet has **Pause Auto Updates** enabled, the story sheet will be blank until auto-updates are resumed.

5. Click **Add a caption** to summarize the story point.

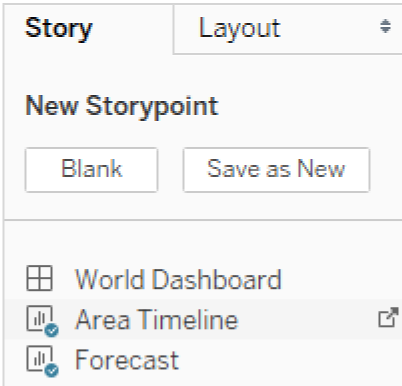
   In Tableau Desktop, you can highlight a key takeaway for your viewers by dragging a text object to the story worksheet and typing a comment.

6. To further highlight the main idea of this story point, you can change a filter or sort on a field in the view. Then save your changes by clicking **Update** on the story toolbar above the navigator box:

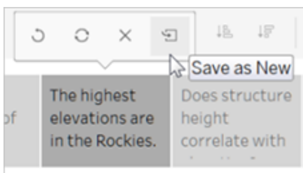


7. Add another story point by doing one of the following:

   - Click **Blank** to use a fresh sheet for the next story point.



   - Start customizing a story point and click **Save as New** on the toolbar above the navigator box.



   - Click **Duplicate** to use the current story point as the basis for a new one.

Explore layout options

You can refine the look of your story using the options on the **Layout** tab.

1. Click the **Layout** tab.
2. Choose a navigator style that best suits your story, and show or hide the next and previous arrows.

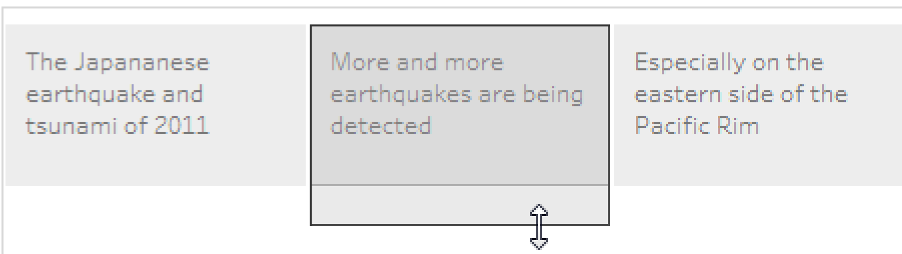
Format a story

Resize captions (Tableau Desktop only)

Sometimes the text in one or more of your captions is too long to fit inside the height of the navigator. In this case, you can re-size the captions vertically and horizontally.

1. In the navigator, select a caption.
2. Drag the border left or right to resize the caption horizontally, down to resize vertically, or select a corner and drag diagonally to resize the caption both horizontally and vertically.

   All captions in the navigator update to the new size.

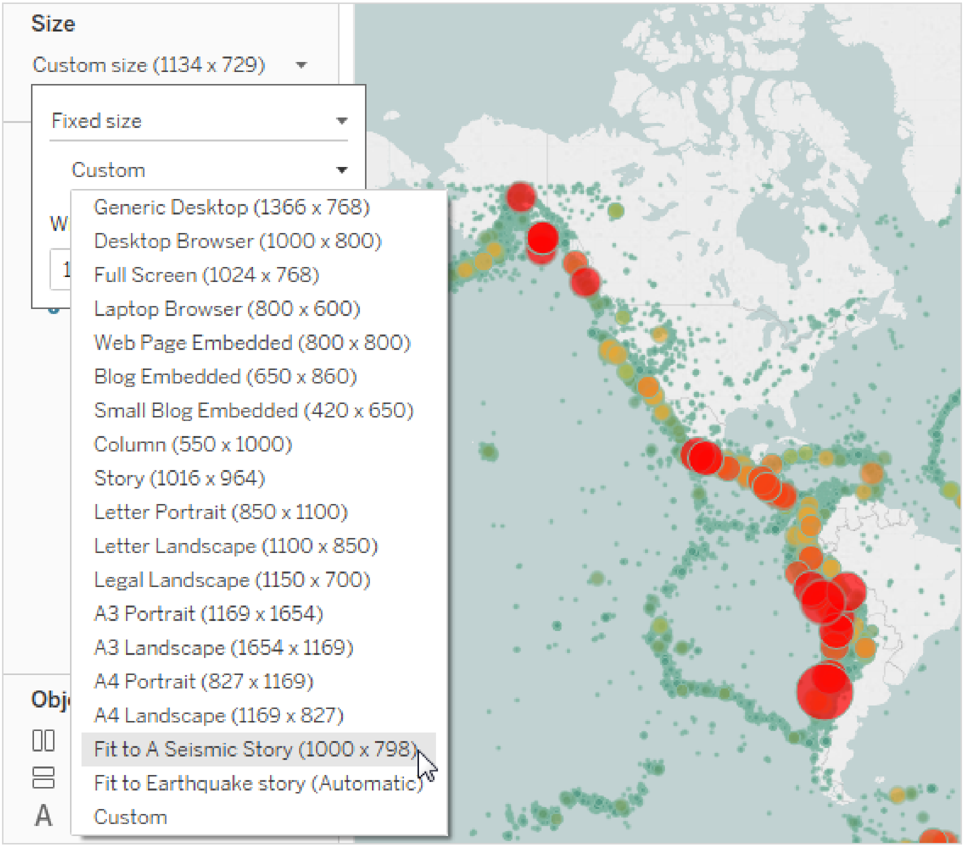


When you resize a caption, you can only select the left, right, or bottom border of the caption.

Fit a dashboard to a story

You can fit a dashboard to the exact size of a story. For example, if your story is exactly 800 by 600 pixels, you can shrink or expand a dashboard to fit inside that space.

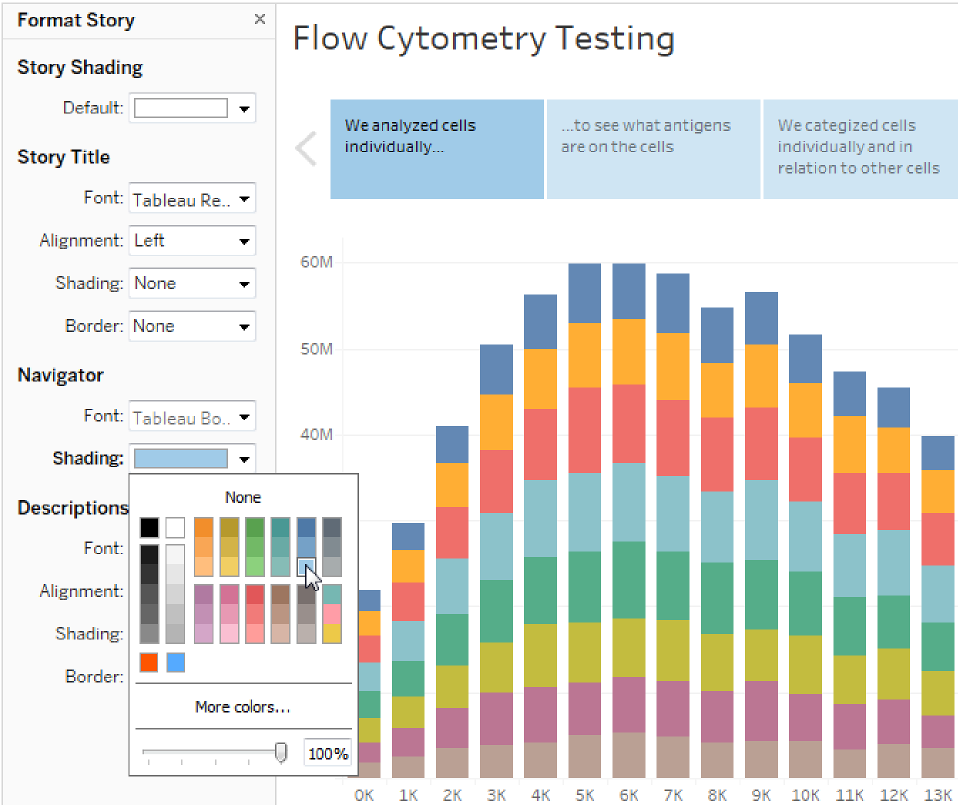
Click the **Size** drop-down menu and select the story you want the dashboard to fit inside.



-

Format a story's shading, title, and text objects (Tableau Desktop only)

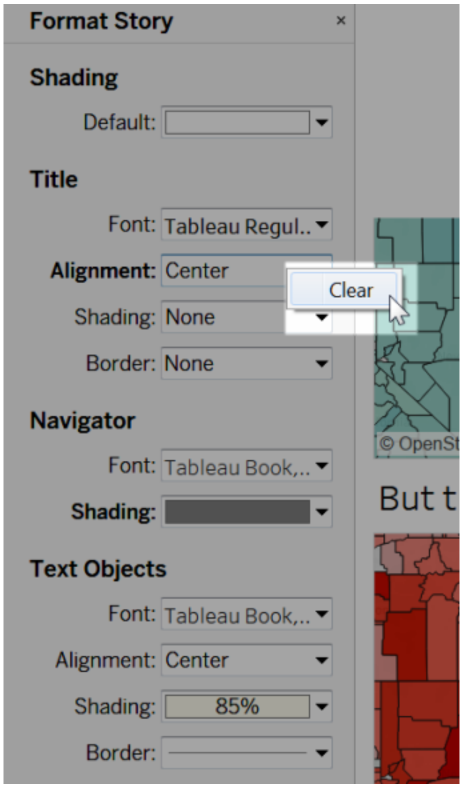
To open the **Format Story** pane, select **Format** > **Story**.



Clear all formatting (Tableau Desktop only)

- To reset a story to its default format settings, click the **Clear** button at the bottom of the **Format Story** pane.
- To clear a single format setting, right-click (Windows) or control-click (macOS) the format setting you want to undo in the **Format Story** pane. Then select **Clear**.

  For example, if you want to clear the alignment of the story title, right-click (control-click on Mac) **Alignment** in the **Title** section, and then select **Clear**.



Delete a story point

Click the X in the toolbar above the point's caption:

Present your story

1. In Tableau Desktop, click the **Presentation Mode** button  on the toolbar. Or, publish the story to Tableau Online or Tableau Server, and click the **Full Screen** button in the upper-right corner of the browser.

2. To step through your story, click the arrow to the right of the story points. Or, in Tableau Desktop, use the arrow keys on your keyboard.
3. To exit Presentation or Full Screen mode, press **Esc**.

Example - A Story That Examines a Trend

The example in this article walks you through building a story about earthquake trends over time.

The story feature in Tableau is a great way to showcase this type of analysis because it has a step-by-step format which lets you move your audience through time.

Rather than showing you how to create all the views and dashboards from scratch, this example starts from an existing workbook. What you'll do is pull the story together. To follow along and access the pre-built views and dashboards, download the following workbook from Tableau Public: An Earthquake Trend Story.

Frame the story

A successful story is well-framed, meaning its purpose is clear. In this example, the story's purpose is to answer the following question: Are big earthquakes becoming more common?

There are several approaches you could take—see **Best Practices for Telling Great Stories** on page 2847 for a list—but the one used here as an overall approach is Change over Time, because it works especially well for answering questions about trends. As you build the story you'll notice that other data story types, such as Drill Down and Outliers, are blended in to support the overall approach.

Build the story

Create a story worksheet

1. Use Tableau Desktop to open the Earthquake Trend Story workbook that you downloaded.

   If you also have Tableau Server or Tableau Online and you want to do your authoring on the web instead of in Tableau Desktop, publish the workbook to your Tableau server, click **Workbooks**, select the workbook, then under **Actions** choose **Edit Workbook**.

Once you open the workbook, you'll see that it has three dashboards. You'll be using those dashboards to build your story. The workbook also has a finished version of the story.

> **Tip**: To see the individual views that are in a dashboard, right-click the dashboard's tab and select **Unhide all Sheets**.

2. Click the **New Story** tab.

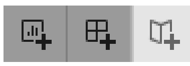
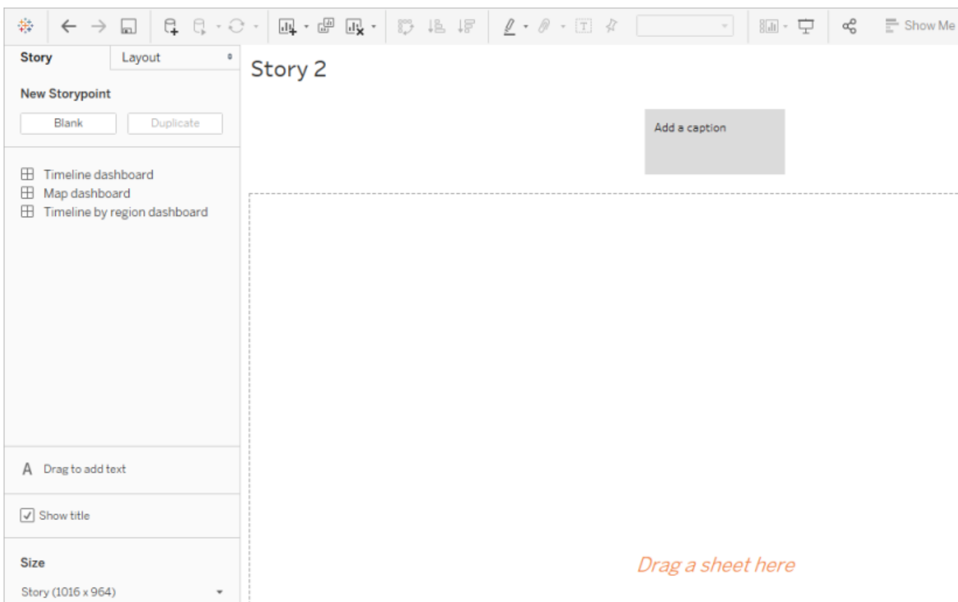


Tableau opens a new worksheet as your starting point.



3. Right-click the **Story 2** tab, choose **Rename Sheet**, and type **Earthquake story** as the worksheet name.

State the question

Story titles are in view at all times and they're a handy way to keep your story's purpose front and center. By default, Tableau uses the worksheet name as the story title. In Tableau Desktop you can override that by doing the following:

1. Double-click the title.
2. In the Edit Title dialog box, replace **<Sheet Name>** with the following:

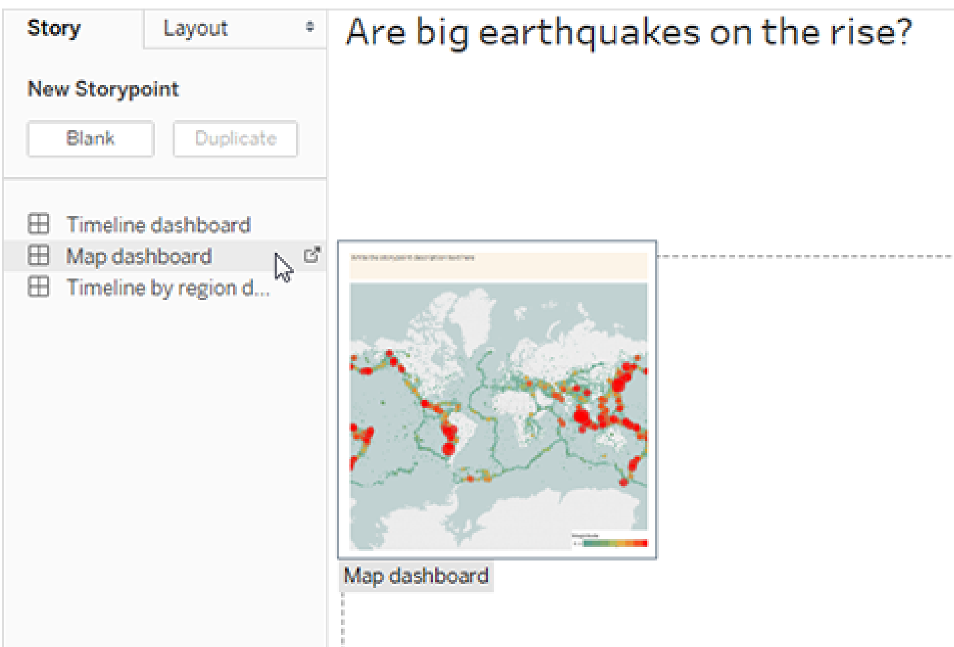
   Are big earthquakes on the rise?
3. Click **OK**.

   If you're authoring in Tableau Server or Tableau Online, the story tab is the only place where you can change the title.

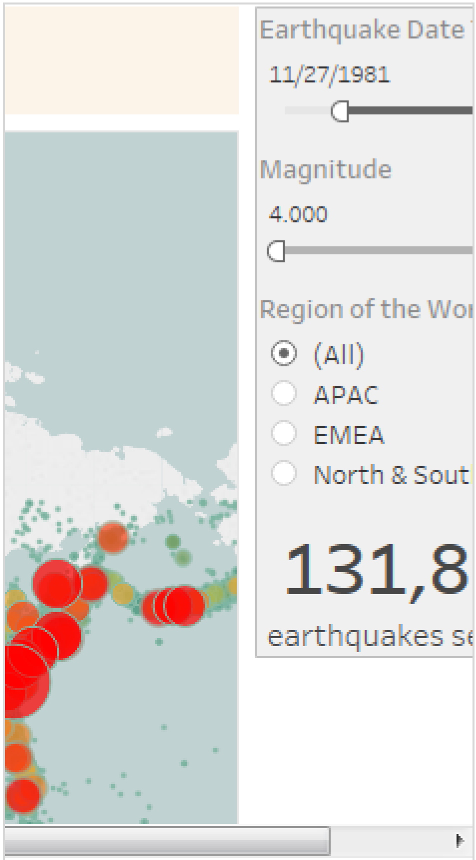
Start big

To help orient your audience, the first story point you create will show the broadest possible viewpoint—all earthquakes, across the entire planet.

1. On the Story pane, double-click **Map dashboard** to place it on the story sheet. If you're using Tableau Desktop, you can also use drag-and-drop to add views and dashboards to a story sheet.

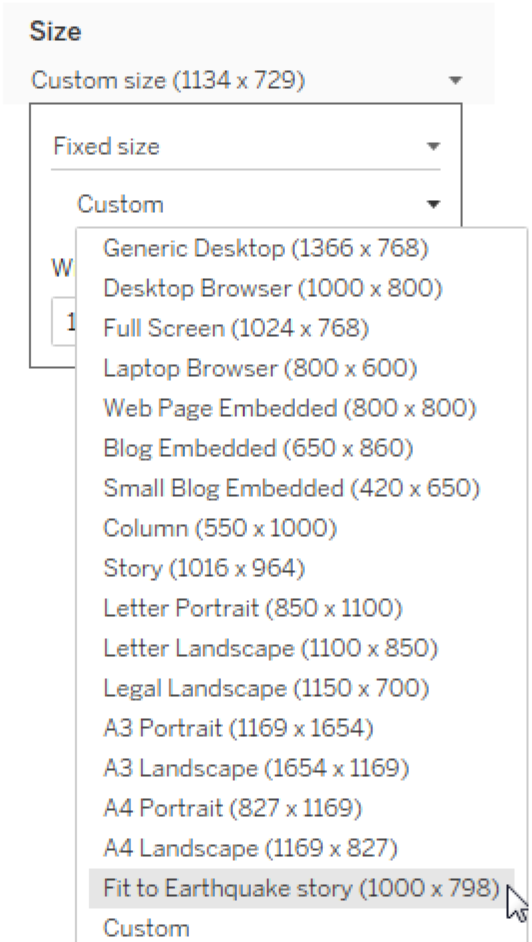


   Notice how there's a horizontal scroll bar and the legend isn't fully displayed.



There's a special setting you can use on your dashboards to prevent this from happening.

2. Select **Map dashboard** and under **Size** on the Dashboard pane, select **Fit to Earthquake story**. This setting is designed to make dashboards the perfect size for a story.

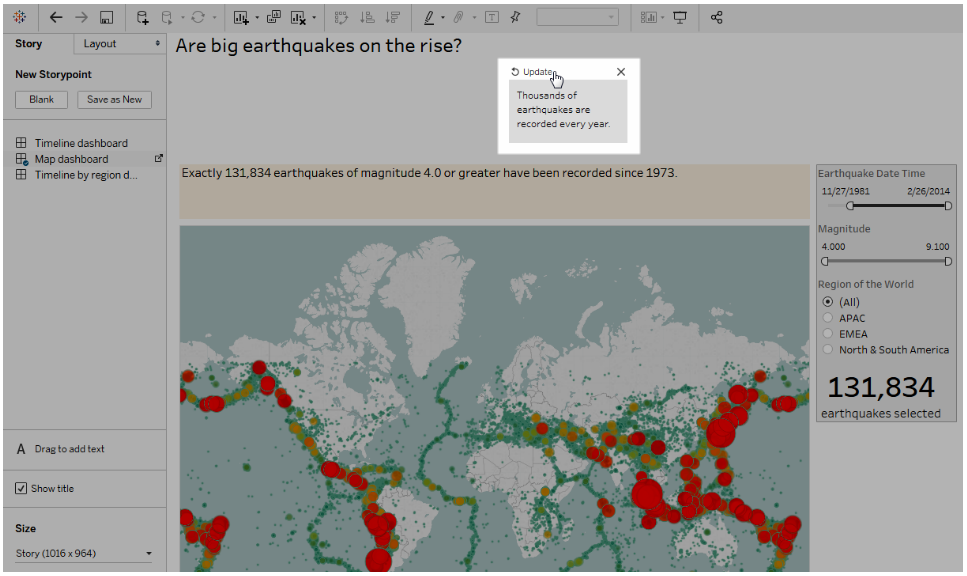


Look at the Earthquake story again. You see that its size has been adjusted and the scroll bars are gone.

3. If you're using Tableau Desktop, add a description for this story point, such as *Exactly 131,834 earthquakes of magnitude 4.0 or greater have been recorded since 1973*.

4. Add caption text by clicking the area that reads **Write the story point description text here**.

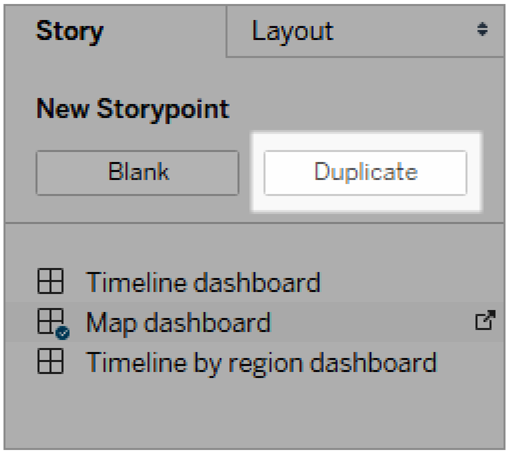
5. Click **Update** on the caption to save your changes to the story point.



Drill down

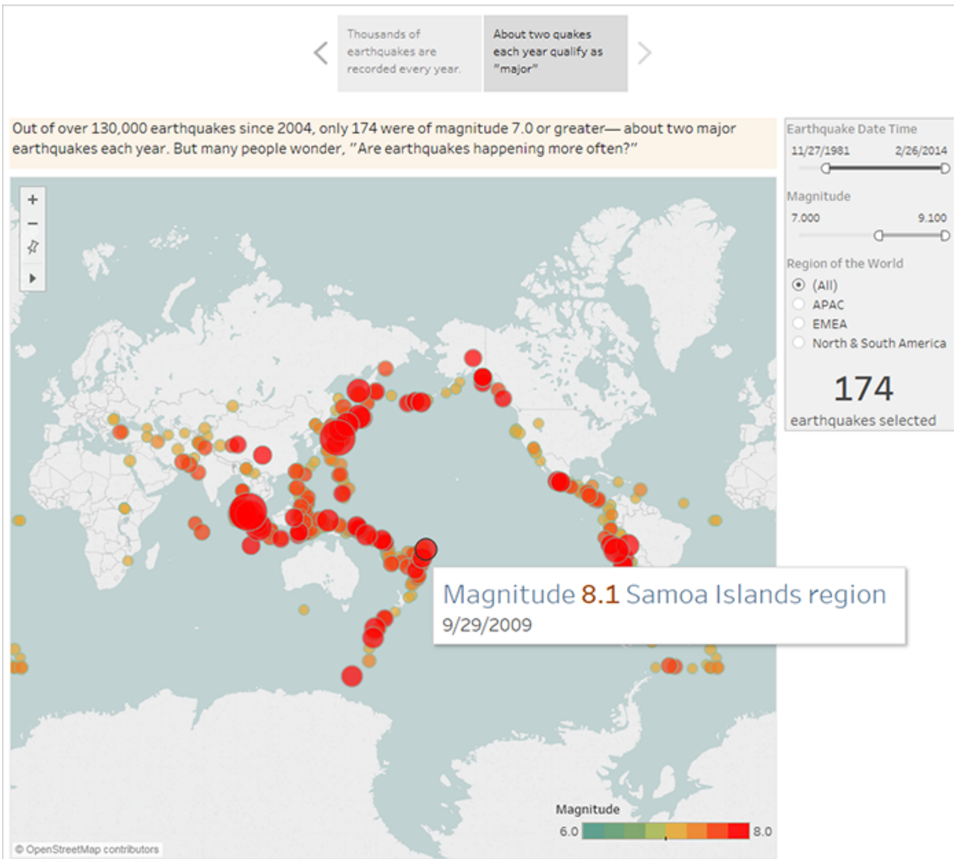
Just like the plot of a good novel needs to move the action along, so does a data story. Starting with your next story point, you'll use the drill-down technique in order to narrow down the scope of the story and keep the narrative moving.

1. To use your first story point as a baseline for your next, click **Duplicate** under **New Storypoint** on the left.



2. Change the **Magnitude** filter to **7.000 – 9.100** so that the map filters out smaller earthquakes. The map pans to show the Pacific "Ring of Fire," where the majority of the large earthquakes occurred.

3. Add a caption, such as *About two quakes each year qualify as "major"*
4. If you're using Tableau Desktop, edit the description to describe what you've done in this story point. For example: *Out of over 130,000 earthquakes since 2004, only 174 were of magnitude 7.0 or greater—about two major earthquakes each year. But many people wonder, "Are earthquakes happening more often?"*
5. Click **Update** in the story toolbar above the caption to save your changes.

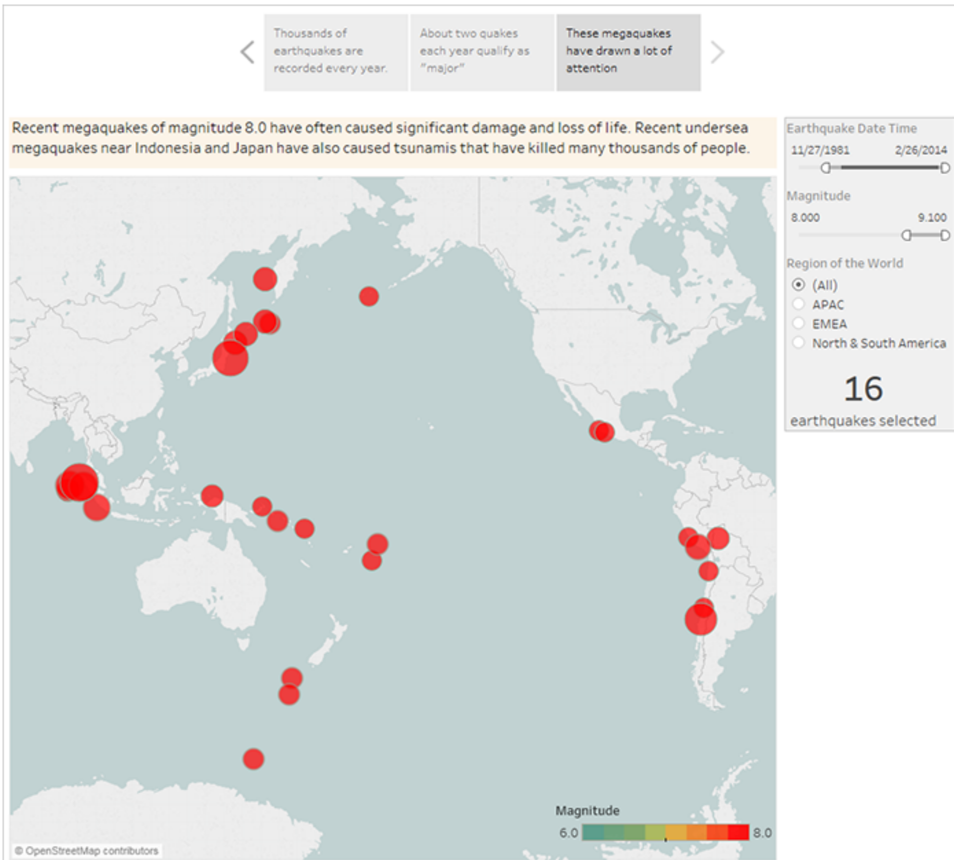


In the next story point, you're going to drill down further, narrowing the story's focus so that a specific type of earthquake—the "megaquake"—comes into view.

1. Click **Duplicate** in your second story point to use it as the baseline for your third story point.
2. Change the **Magnitude** filter to **8.000 – 9.100** so that the map filters out everything except the megaquakes.
3. Add the caption and description text.

- Caption: *These megaquakes have drawn a lot of attention*
- Description (Tableau Desktop only): *Recent megaquakes of magnitude 8.0 and higher have often caused significant damage and loss of life. The undersea megaquakes near Indonesia and Japan also caused tsunamis that have killed many thousands of people.*

4. Click **Update** to save your changes.



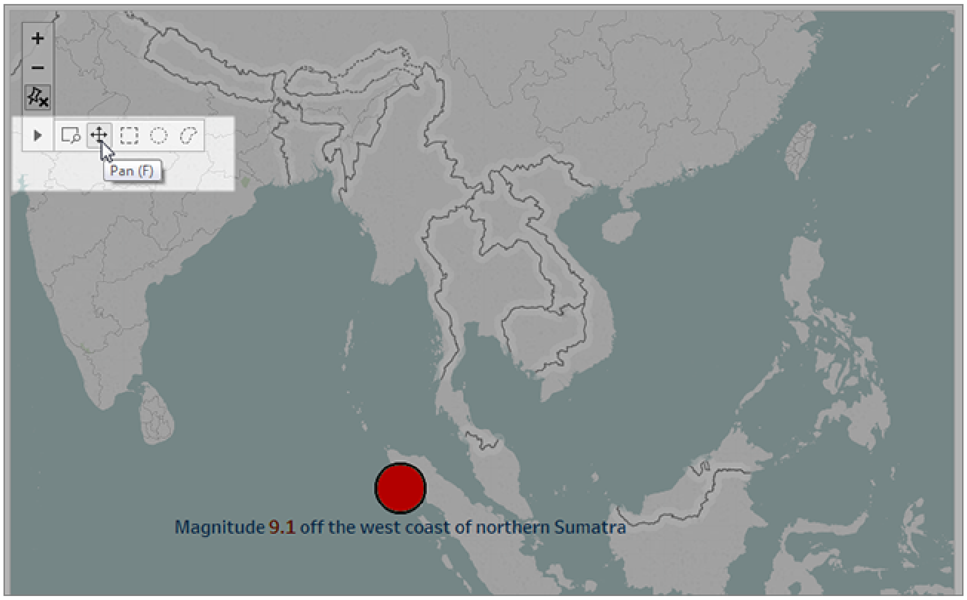
Highlight outliers

In the next two story points, you're going to further engage your audience by examining data points at the far end of the scale: the two most deadly earthquakes in recent history.

1. As you've done before, use **Duplicate** to create a new story point as your starting point.
2. Adjust **Magnitude** to **9.000–9.100** and you'll see just two data points.
3. Select one of the marks, such as the Indian Ocean earthquake and tsunami of 2004 that

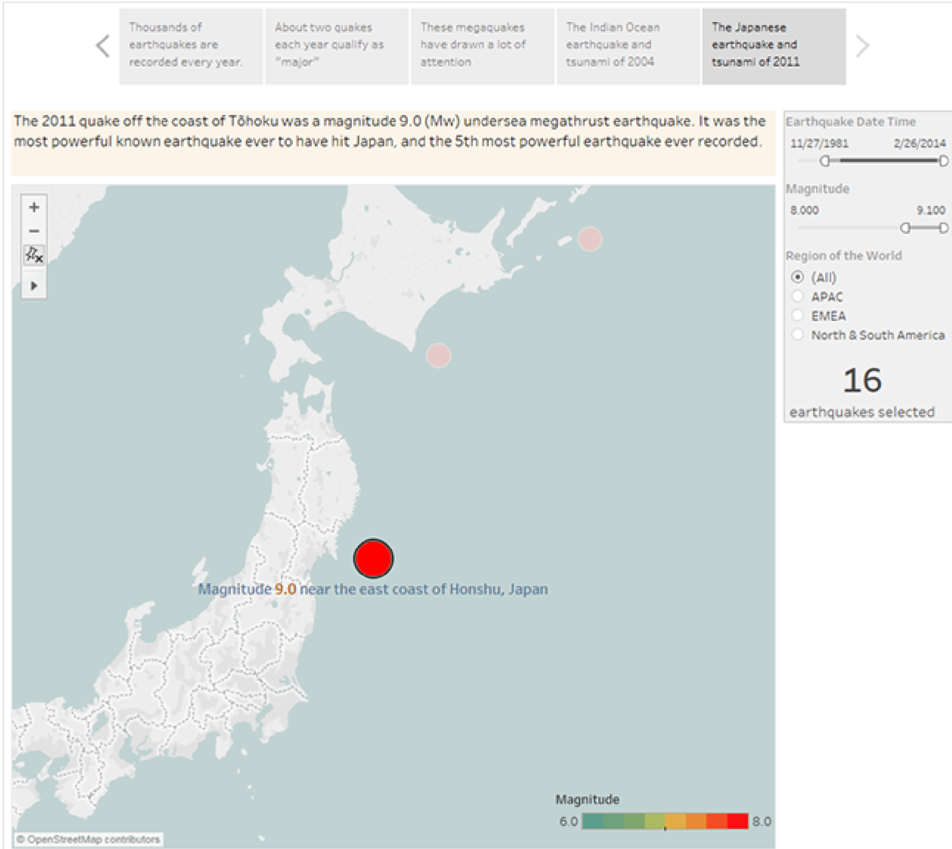
had a magnitude of 9.1.

4. Use the pan tool on the maps menu to center it in your story point.



5. Add caption and description text. For example:
   - Caption: *The Indian Ocean earthquake and tsunami of 2004*
   - Description (Tableau Desktop only): *The 2004 Indian Ocean earthquake was an undersea megathrust earthquake that occurred on December 26, 2004. It is the third largest earthquake ever recorded and had the longest duration of faulting ever observed, between 8.3 and 10 minutes.*
6. Click **Update** to save your changes.
7. Repeat the preceding steps for the Japanese earthquake and tsunami of 2011, using the following as caption and description text.
   - Caption: *The Japanese earthquake and tsunami of 2011*
   - Description (Tableau Desktop only): *The 2011 quake off the coast of Tõhoku was a magnitude 9.0 undersea megathrust earthquake. It was the most powerful known earthquake ever to have hit Japan, and the 5th most powerful earthquake ever recorded.*

   Notice that you've already created a compelling visual story using just a single dashboard—all by filtering the data and zooming and panning the map.

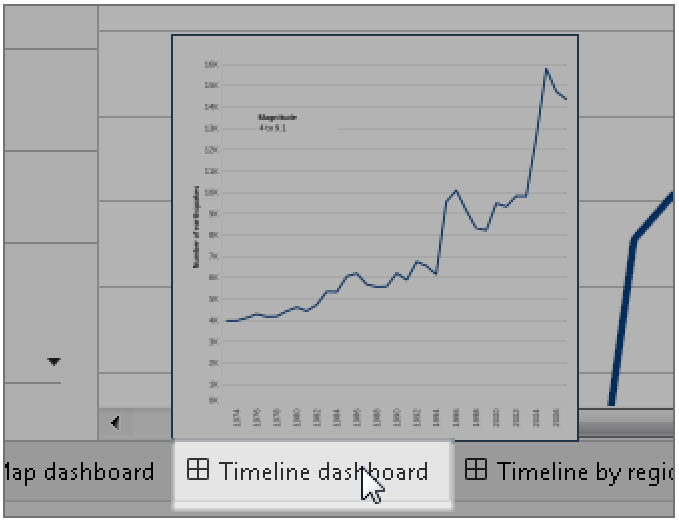


We still haven't answered the key question, however: Are big earthquakes on the rise? The next story points will dig in to that angle.

Show a trend

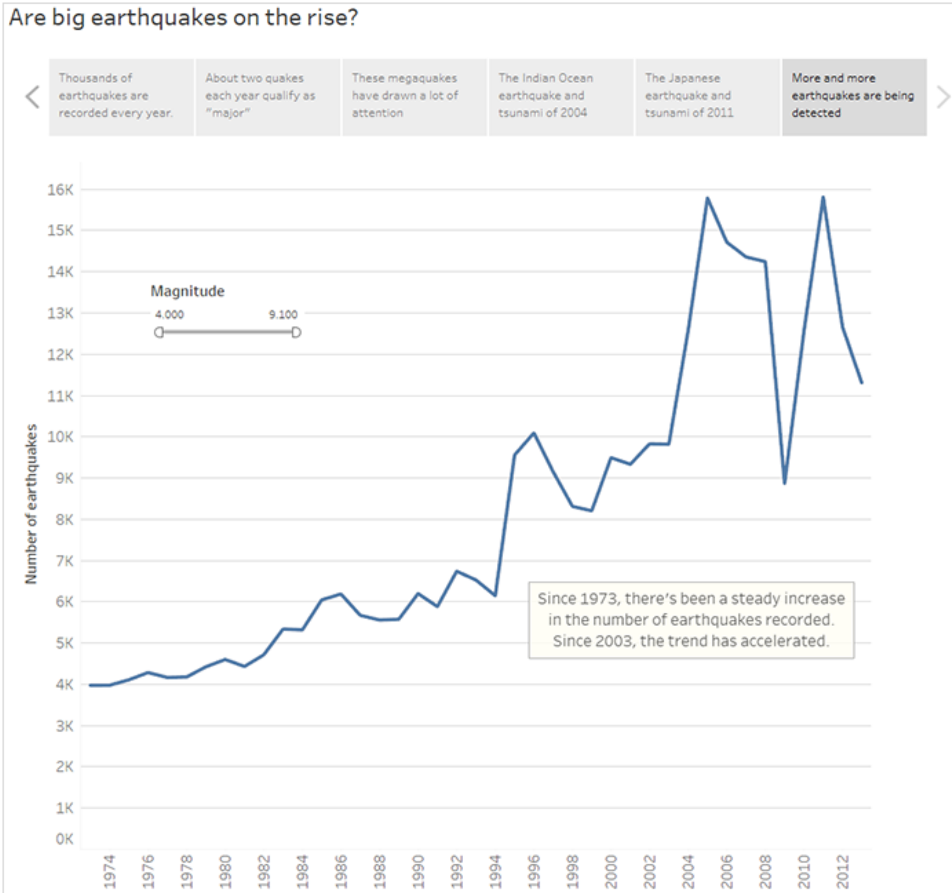
In the next story point, you'll switch to a line chart (the Timeline dashboard) to show your audience a trend you spotted when you were initially creating views and dashboards.

1. Switch from the story you're building to **Timeline dashboard**.



2. On the Timeline dashboard, set size to **Fit to Earthquake story**.
3. Go back to your story and click **Blank** to create a fresh story point.
4. Double-click the **Timeline dashboard** to add it to your story sheet.

   More earthquakes are being reported over time since 1973. In fact, it's increased significantly!
5. Add a caption, such as: *More and more earthquakes are being detected*
6. Use **Drag to add text** to add a description of the trend (Tableau Desktop only): *Since 1973, there's been a steady increase in the number of earthquakes recorded. Since 2003, the trend has accelerated.*

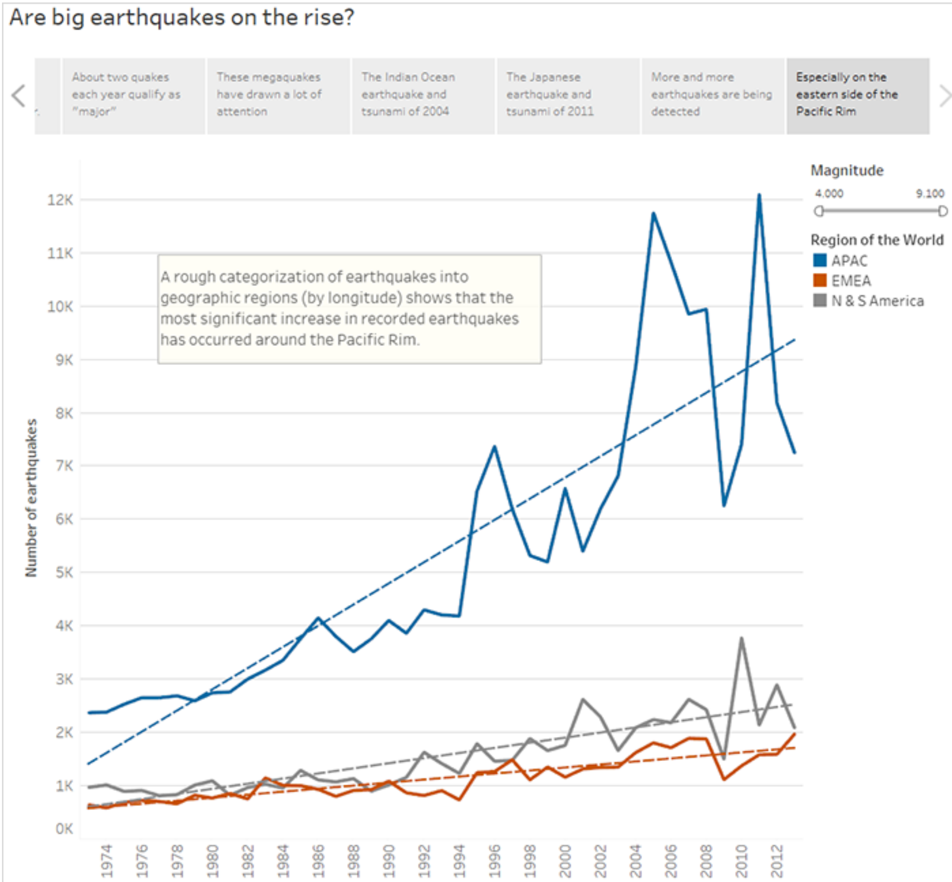


Offer your analysis

From your earlier work in this story with the **Map dashboard** you know that there are regional differences in earthquake frequency. In your next story point, you'll pull in the **Timeline by region dashboard**, which breaks out earthquakes by region, and adds trend lines, which help reduce the variability in the data.

1. Click **Blank** to create a new story sheet.
2. Double-click the **Timeline by region dashboard** to the story sheet. The APAC region clearly stands out.
3. Add a caption then use **Drag to add text** to add a comment that points out the large number of earthquakes in the APAC region.
   - Caption: *Especially on the eastern side of the Pacific Rim*
   - Description (Tableau Desktop only): *A rough categorization of earthquakes into*

geographic regions (by longitude) shows that the most significant increase in recorded earthquakes has occurred around the Pacific Rim.



Answer the question

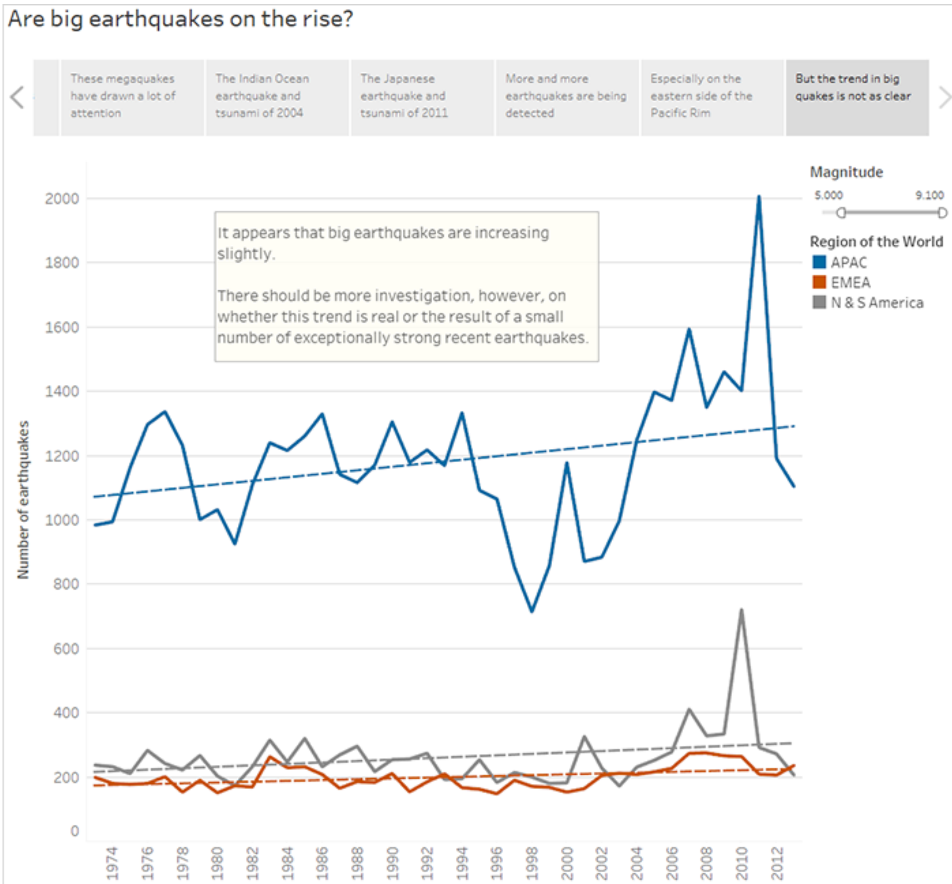
Thus far, your data story has concluded that earthquake frequency in the Pacific Rim has increased since 1973, but your original question was about whether big earthquakes are becoming more frequent.

To answer this question, in your final story point, you'll filter out weaker earthquakes and see what the resulting trend line is.

1. Click **Duplicate** to create a new story sheet.
2. Set the **Magnitude** filter to **5.000–9.100**. Notice how the trend lines have flattened out but there's still a slight increase.
3. Add a caption then use **Drag to add text** to add your answer to the story point.

Caption: *But the trend in big quakes is not as clear*

Description (Tableau Desktop only): *It appears that big earthquakes are increasing slightly. There should be more investigation, however, on whether this trend is real or the result of a small number of exceptionally strong recent earthquakes.*



As is often the case with a data story, the story ends with additional questions.

Yes, there's a trend, but it's slight. More big earthquakes (magnitude 5.000 - 9.100) have been reported in recent years, especially in the Asia-Pacific region, but could that be natural variation? That might be a good topic for another story.

Format Your Work

Formatting is an important part of both your analysis and presentation. You can format almost everything you see on a worksheet including the fonts, shading, alignment, borders, and graph lines. For example, in a text table you may want to add banded shading to help you visually separate consecutive groups of rows or columns. In a scatter view with reference lines you may want to change the line thickness and color. All of these settings can be changed using the Format window.

Most often you will want to specify format settings for the entire worksheet, all rows, or all columns. However, Tableau also allows you to format individual parts of the view as well. For example, you can format specific fields, resize the cells and the table, and edit individual axes.

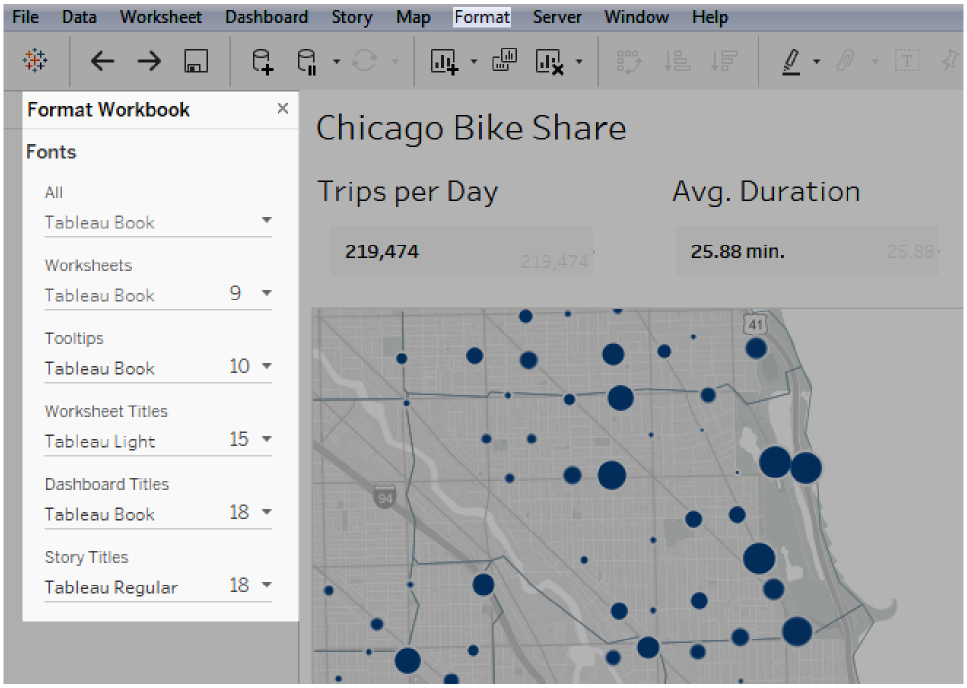
Visual Best Practices

The fonts, colors, shading, alignment, borders, and grid lines in your visualization are important parts of both your analysis and the story you're telling. Tableau products are designed so that you can create great-looking visualizations that use visual best practices by default, freeing you from the need to think about things like fonts and colors—unless you want to.

If you do want to customize, you can control the look of almost everything you see on a worksheet. This article outlines visual best practices and tips to keep in mind while you customize, from ideal workflow to how to get the most out of tooltips.

Format from largest to smallest

As you change the look and feel of your work, use a "biggest to smallest" workflow. Start by formatting fonts and titles at the workbook level, then move on to the worksheet level. Save formatting the individual parts of a view for last. A workbook is the largest possible "container" for formatting changes and making changes at that level first will save you time.



See **Format at the Workbook Level** on page 2882, **Format at the Worksheet Level** on page 2892, and **Format Text and Numbers** on page 2923 for details.

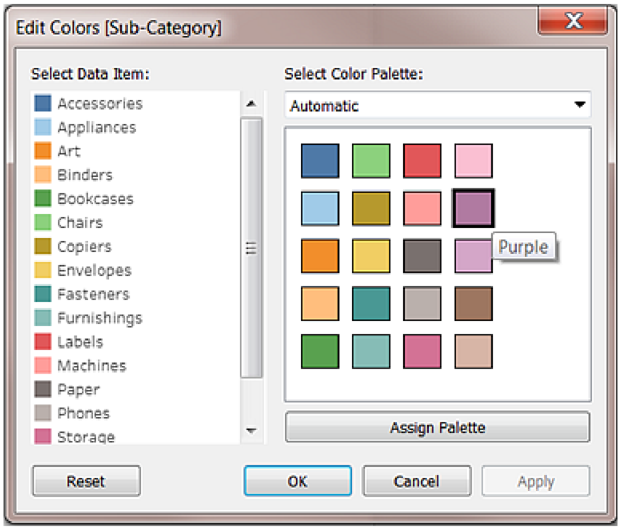
Change color with purpose

If you want to change the colors used by your visualization, Tableau makes it easy: just click the **Color** card, then **Edit Colors**. For example, maybe you want to emphasize a key finding. You can use neutral colors with a single, bright color to highlight what you want your viewers to pay attention to. If you want to match your company's brand, you can create a custom palette. See **Create Custom Color Palettes** on page 2969 for steps.

When you change colors, keep the following best practices in mind..

Discrete fields and categorical palettes

Discrete fields are ones where the field's values are unique. One example of this is customer names. Tableau automatically associates discrete fields with *categorical* palettes, which have colors that are designed to be distinct from one another yet also work well together in the visualization as a whole.



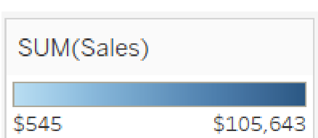
Change colors for individual fields

Once you have selected a categorical palette, you can manually change the colors associated with specific fields.

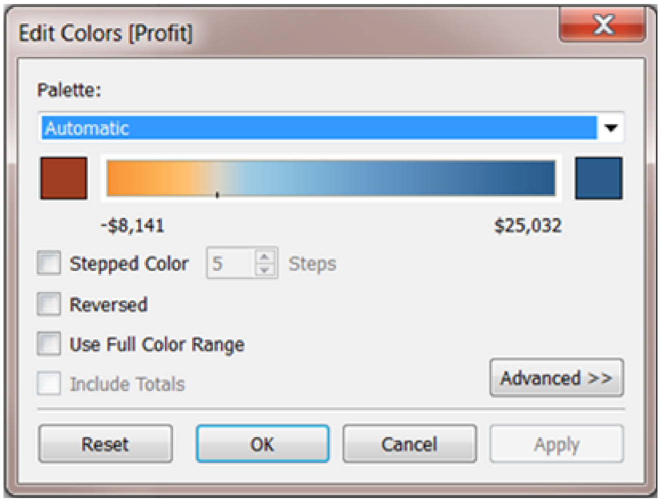
1. From the **Edit Colors** dialog, double click on any of the fields in the **Select Data Item** panel.
2. The color palette for your operating system will open, allowing you to define a new color for the item selected.

Continuous fields and quantitative palettes

With continuous fields, the field's values are part of a whole (such as sales over time). Tableau automatically associates these fields with *quantitative* palettes, where colors are along a continuum or range. If the field's values are positive, a single-color range is best:

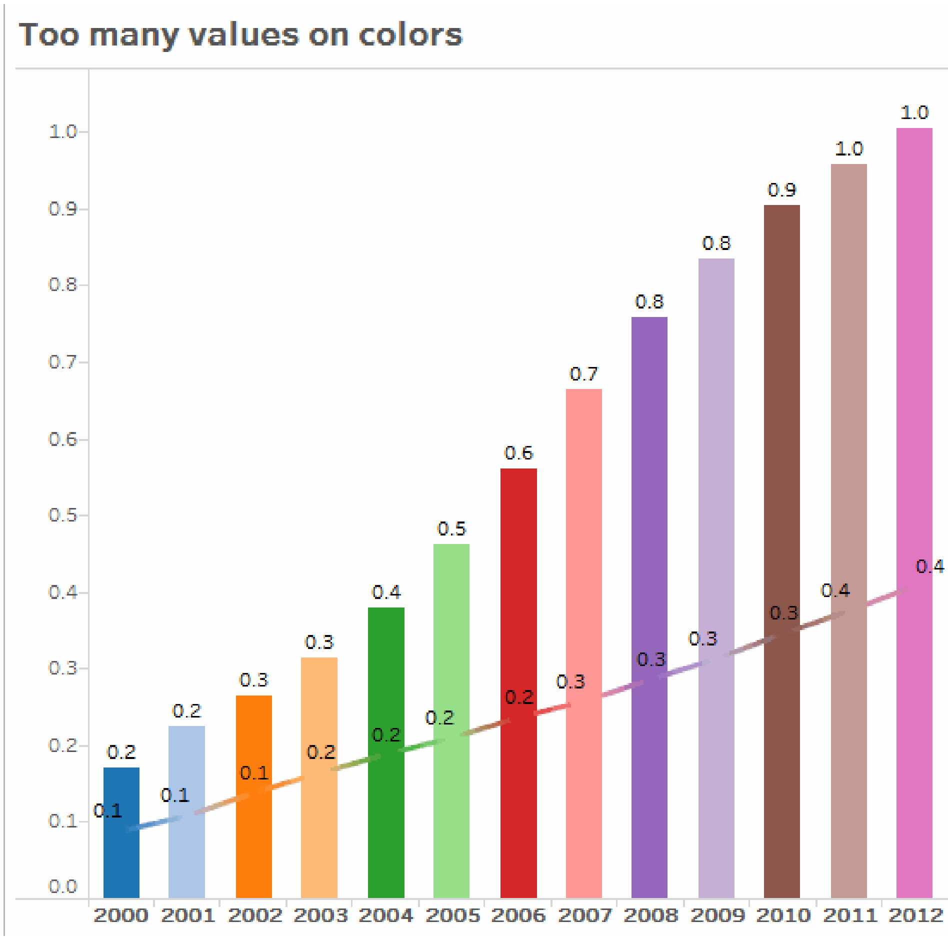


If there are both positive and negative values, choose a two-color range:



Limit colors

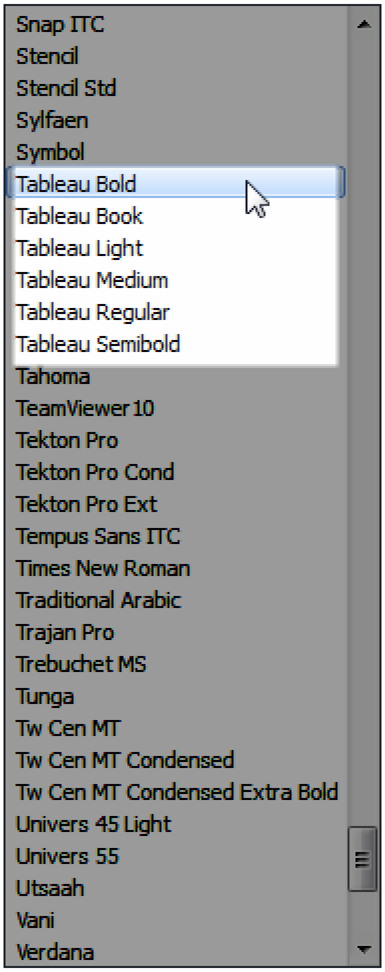
Color used correctly can enhance analysis. Too many colors can create visual overload for your users and impede analysis. In the view below, the stacked bars represent cell phone subscriptions and the trend line is internet usage over time. In the first image, Years is on the **Colors** shelf. Notice how hard it is to pick out the trend line among all the other colors and how easy it is to spot the trend line in the second image, where there are only two colors.



Click the image to replay it.

Use fonts that optimize online readability

The fonts in the Tableau typeface were designed to be paired with data visualizations and are optimized for legibility at small sizes. Tableau visualizations use them by default.



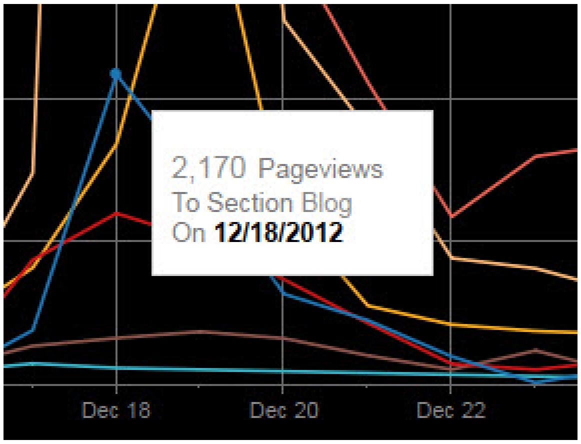
If you need to use a different font, choose one that optimizes readability online, such as the following:

- Arial
- Trebuchet MS
- Verdana
- Times New Roman
- Lucida sans

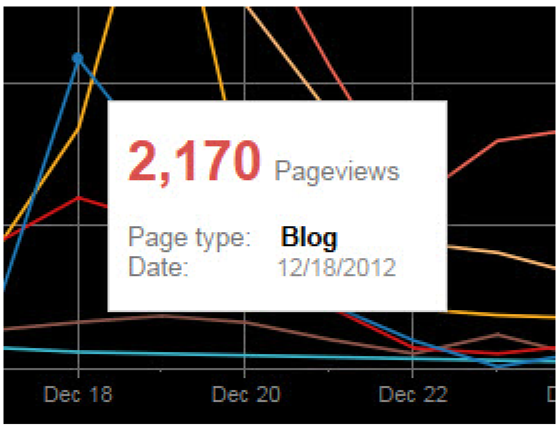
Reinforce your story with tooltips

A person looking at your dashboard will intuitively use his or her mouse to explore marks, and this makes tooltips appear. Tooltips appear by default for most views and they are a great way to reinforce your data story. You can customize your tooltips by clicking **Worksheet** > **Tooltip**.

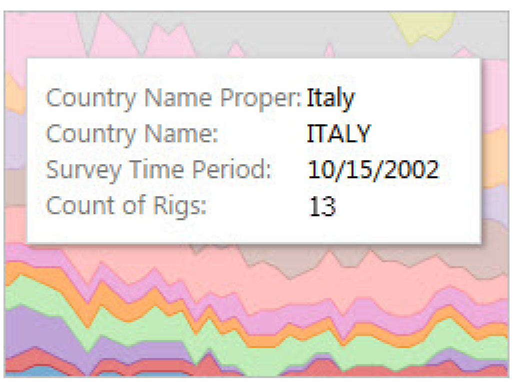
For example, here's a tooltip for a view on pageviews for a web site:



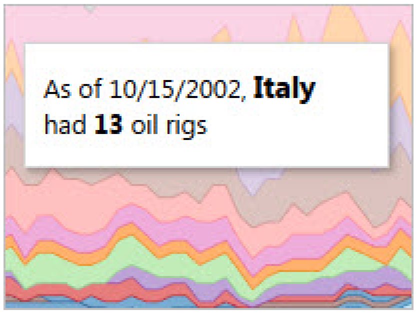
Here's an example of how the above tooltip could be rewritten and formatted to highlight what's important to the people looking at this view:



You can also rewrite your tooltips to tell a mini-story. For example, here's a tooltip about the number of oil rigs in Italy.



Below, the tooltip is rewritten as a phrase and key elements have been put in bold to draw the viewer's attention.



Consider your axes

By default, an axis range in Tableau will automatically adjust, based on the data in your visualization. Filtering and other actions can change the axis range. If you are trying to create a specific comparison between two views, changing axes ranges can make analysis difficult for your users. If this is the case, you can set your axes to a specific, fixed range. For steps on how to do this, see **Examples: Use Different Axis Ranges (Uniform, Independent, Fixed)** on page 2913.

If you're working with data that has a very large range, a fixed axis may not be practical. If this is the case, add grid lines to your visualization. Grid lines can help your users stay oriented when the axis automatically adjusts. See **Format lines** on page 2897 for more information.

Format at the Workbook Level

You can quickly change how fonts, titles, and lines look in every view in a workbook by specifying format settings at the workbook level, instead of the worksheet level.

For example, you might want to use a specific font, size, and color so that all views adhere to your company's brand. You might also want to remove grid lines from your views—or make them more noticeable by increasing their pixel size or color.

You can also change the theme used by your workbook. Themes control items like the default font, colors, and line thickness. When you create a new workbook, it automatically uses the Default theme, which uses visual best practices.

Change lines in your workbook

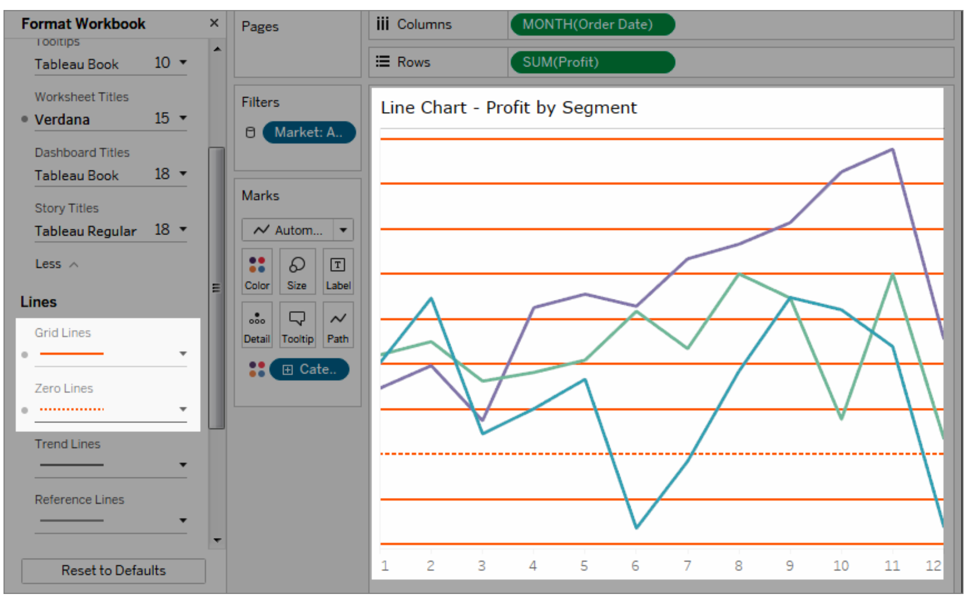
You can change every type of line for all views in your workbook, or you can change just certain types of lines, such as trend lines. You can also turn off certain types of lines, including grid lines.

When you make changes to your workbook's line settings, a gray dot appears next to the setting in the **Format Workbook** pane. You can quickly switch back to default settings using the **Reset to Defaults** button.

> In Tableau, line opacity is linked to line color. If you set a line's opacity at the workbook level, all worksheets in the workbook will receive the workbook's color setting for that line as well. If you need to change line opacity and maintain different color settings for a line type, change opacity at the worksheet level instead of at the workbook level.

1. On the **Format** menu, select **Workbook**.

2. The **Format Workbook** pane replaces the **Data** pane on the left and provides a series of drop-down lists where you can change all line settings in a workbook.

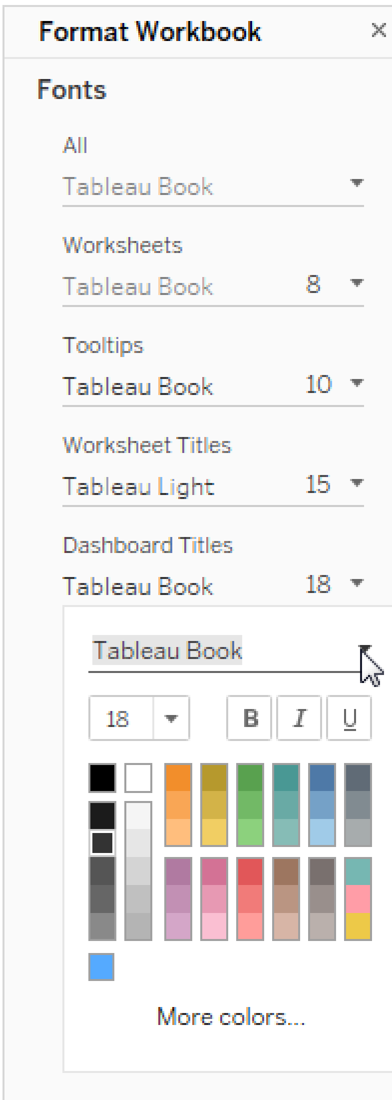


To confirm that your change has been applied to every view in your workbook, you can use thumbnails. Right-click a thumbnail, and select **Refresh All Thumbnails**.

Change fonts in your workbook

You can change all fonts in your workbook or you can change fonts for only certain areas, such as just worksheet titles.

1. On the **Format** menu, select **Workbook**.
2. The **Format Workbook** pane replaces the **Data** pane on the left and provides a series of drop-down lists where you can change all font settings in a workbook, as well as the font settings for titles of worksheets, stories, and dashboards.



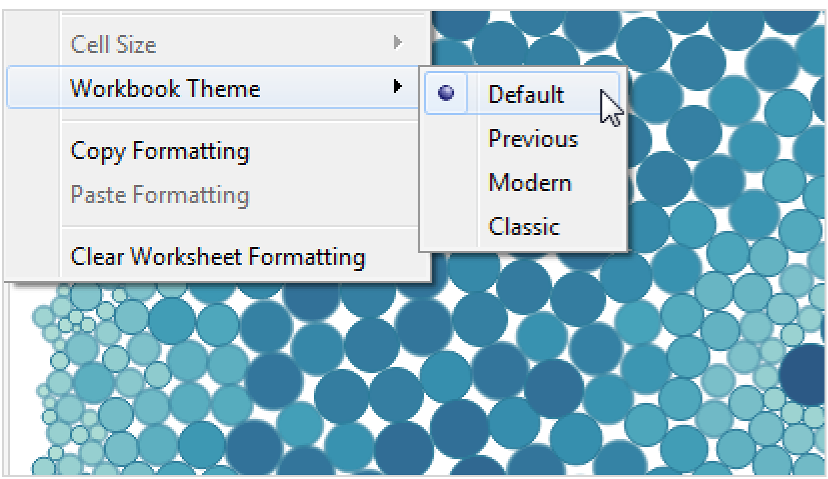
> If you have made font changes at the worksheet level, such as on a filter card or a worksheet title, changing the font at the workbook level will overwrite those changes.

Upgrade or change your workbook theme (Tableau Desktop only)

If your workbook is set to a theme other than Default, it will retain that theme when you upgrade from one version of Tableau to another, but you can easily change the theme to Default. Conversely, you can change your workbook's theme from Default to a different, earlier theme.

If you upgraded from version 9.3 to version 10.x and your theme was set to Default in version 9.3, it will be set to Previous in version 10.x. After you switch to the Default theme, check your workbook to confirm that items are sized appropriately. You might need to make some manual adjustments.

1. On the **Format** menu, select **Workbook Theme**.
2. Choose a theme.



Each theme is associated with a different version of Tableau Desktop.

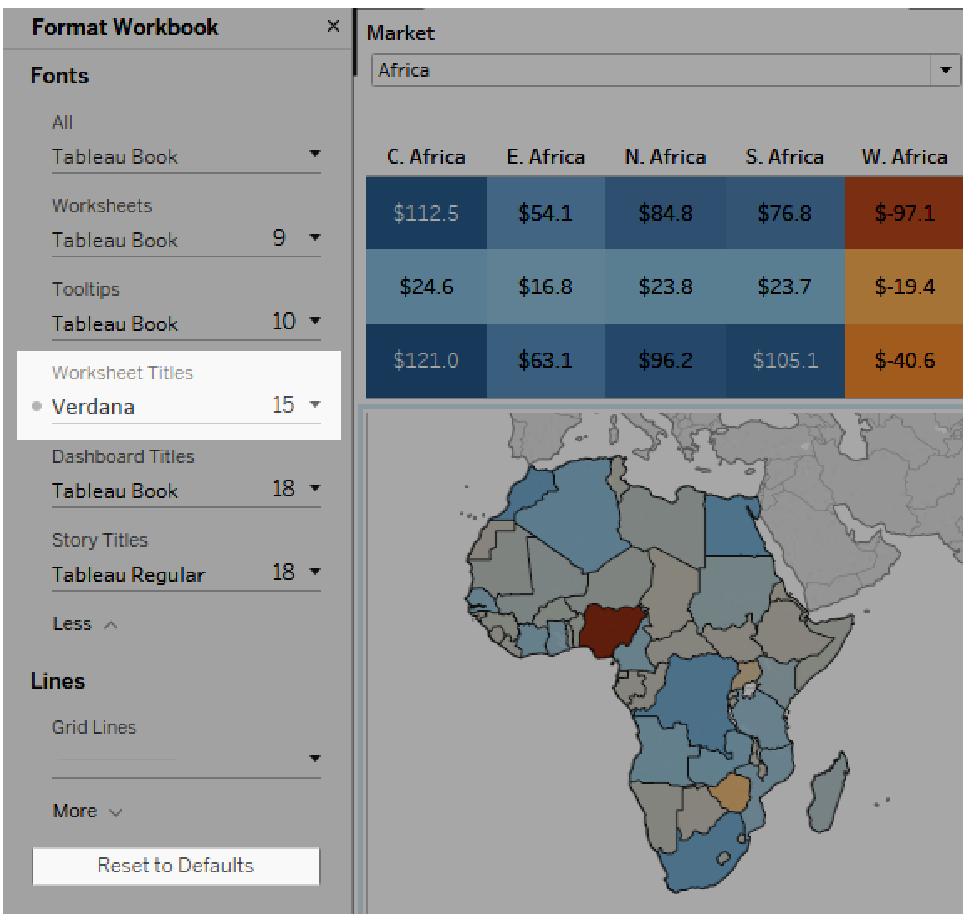
This workbook theme...	Is associated with these Tableau versions...
Default	Version 10.x

Previous	Versions 8.0.x – 9.3.x
Modern	Versions 3.5 – 7.0.x
Classic	Versions 1.0 – 3.2

Reset a workbook to its default settings

When you make changes to your workbook's font settings, a gray dot appears next to the setting in the **Format Workbook** pane. You can quickly switch back to default settings using the **Reset to Defaults** button.

1. On the **Format** menu, select **Workbook**.
2. In the **Format Workbook** pane, click **Reset to Defaults**.



Format Animations

Animate visualizations to better highlight changing patterns in your data, reveal spikes and outliers, and see how data points cluster and separate.

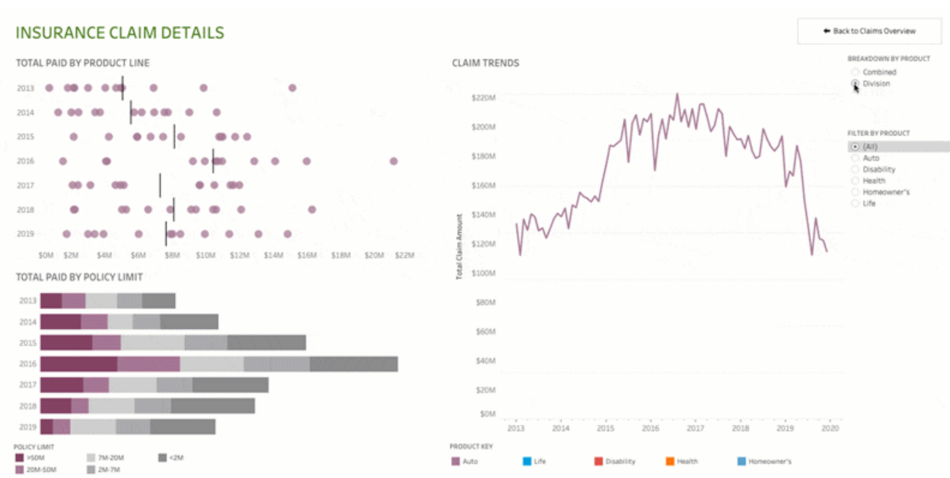
Animations visually transition between filter, sort, and zoom settings, different pages, and changes to filter, parameter, and set actions. As visualizations animate in response to these changes, viewers can more clearly see how data differs, helping them make better informed decisions.

Understanding simultaneous and sequential animations

When you author animations, you can choose between two different styles: simultaneous or sequential. Here are examples of each type.

Simultaneous animations

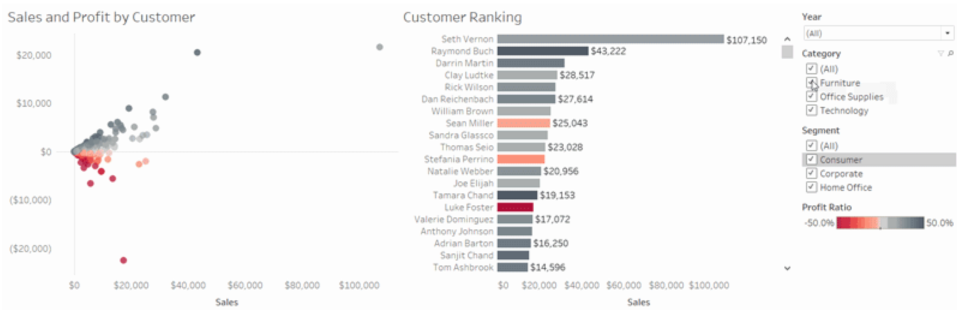
The default simultaneous animations are faster and work well when showing value changes in simpler charts and dashboards.



Click the image above to replay the animation.

Sequential animations

Sequential animations take more time but make complex changes clearer by presenting them step-by-step.



Click the image above to replay the animation.

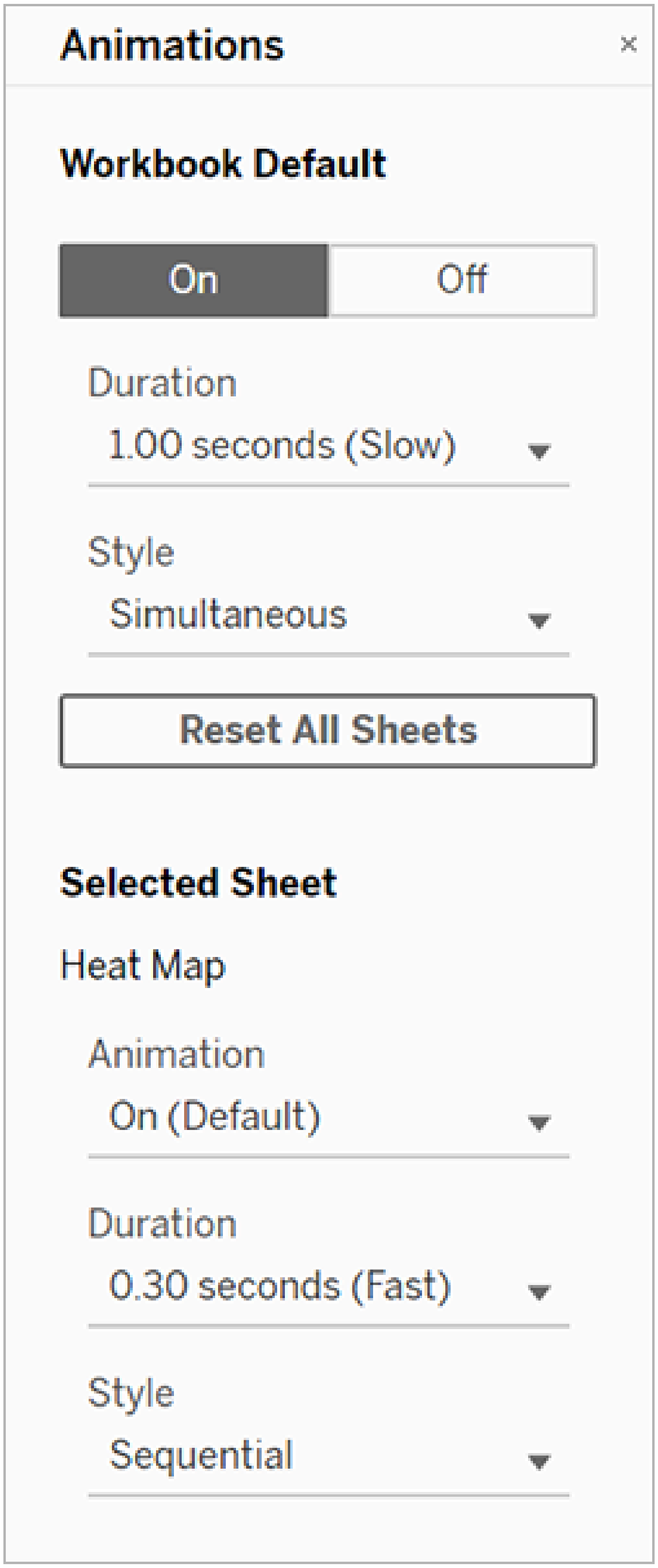
Animate visualizations in a workbook

When you create a new workbook, Tableau enables animations for your viz by default. You can turn animations on or off at the user and workbook level.

1. Choose **Format** > **Animations**.
2. If you want to animate every sheet, under **Workbook Default**, click **On**. Then do the following:
   - For **Duration**, choose a preset, or specify a custom duration of up to 10 seconds.
   - For **Style**, choose **Simultaneous** to play all animations at once or **Sequential** to fade out marks, move and sort them, and then fade them in.
3. To override workbook defaults for a particular sheet, change the settings under **Selected Sheet**.

   > **Note:** In the Selected Sheet section, “(Default)” indicates a setting that

automatically reflects the related Workbook Default setting.


Animations
Workbook Default
On
Off
Duration
1.00 seconds (Slow)
Style
Simultaneous
Reset All Sheets
Selected Sheet
Heat Map
Animation
On (Default)
Duration
0.30 seconds (Fast)
Style
Sequential

Reset animation settings for a workbook

You can reset animations to return an entire workbook to the default animation settings. Be aware that this turns animations off by default.

1. Choose **Format** > **Animations**.
2. In the middle of the **Animations** pane, click **Reset All Sheets**.

Completely disable all animations

When you create a new workbook, animations are enabled by default. If you find animations distracting while viewing vizzes, you can completely disable them so they never play. (This isn't a system-wide setting; each user needs to apply it separately.)

- In Tableau Desktop, choose **Help** > **Settings and Performance**, and deselect **Enable Animations**.
- In Tableau Online or Tableau Server, click your profile image or initials in the top right corner of the browser, and choose **My Account Settings**. Then scroll down to the bottom of the page, deselect **Enable animations**, and click **Save Changes**.

Note: When animations are disabled, you can still choose **Format** > **Animations** in authoring mode and adjust settings—but they will have no effect.

Why animations won't play

Server rendering

Animations won't play if a viz is server-rendered. To ensure that vizzes render on a client computer or mobile device, use these techniques:

- If you're a viz author, reduce viz complexity.
- If you're a Tableau Server administrator, increase the complexity threshold for client-side rendering.

Note: On computers with lower processing power, animations may appear choppy, but users can continue to interact with vizzes without any delays in responsiveness.

Unsupported browsers and features

Animations are supported by all web browsers except Internet Explorer.

The following Tableau features don't animate:

- Maps, polygons, and density marks in web browsers
- Pie and text marks
- Axes and headers
- Forecasts, trends, and reference lines
- Page history trails (If a viz includes these, turn off animations to avoid unexpected behavior.)

Format at the Worksheet Level

You can format settings for fonts, alignment, shading, borders, lines and tooltips at the worksheet level. For example, you might want to remove all the borders in a text table, or add shading to every other column in a view.

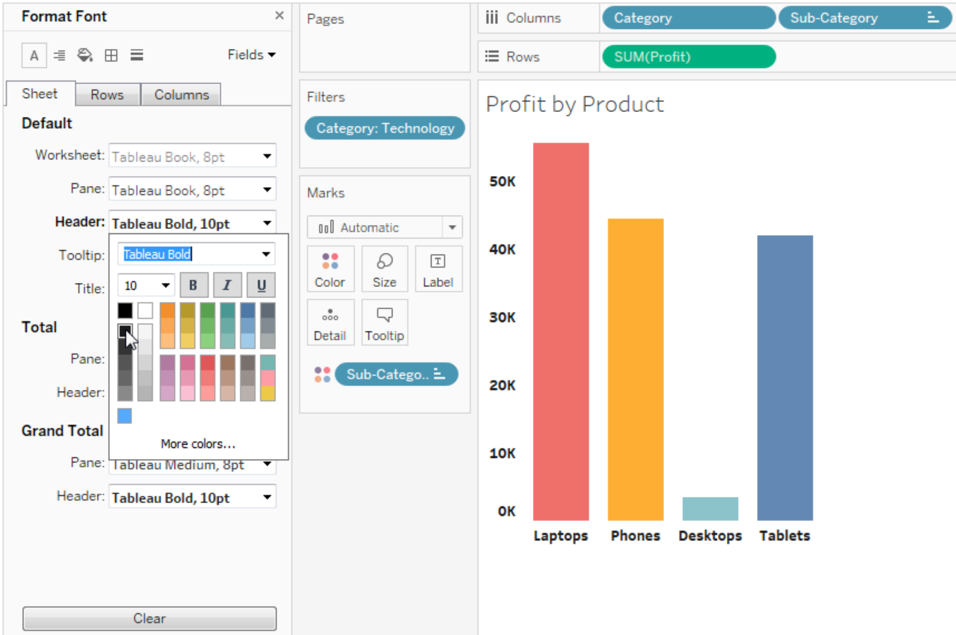
When you make formatting changes at this level, they apply only to the view you're working on. See **Format at the Workbook Level** on page 2882 for how to make changes that apply to every view in your workbook.

Access worksheet formatting settings

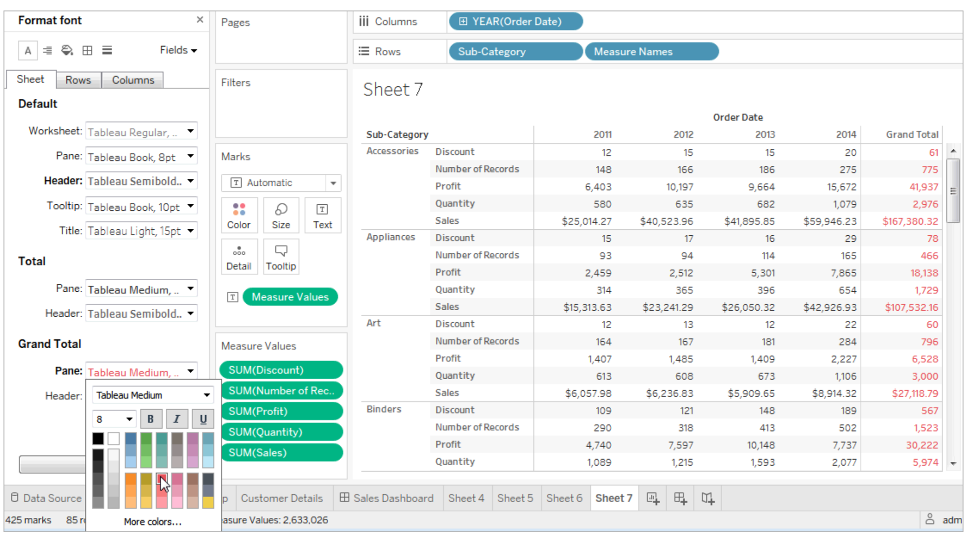
1. Display a worksheet or dashboard.
2. From the **Format** menu, choose the part of the view that you want to format, such as **Font**, **Borders**, or **Filters**.

Format fonts

For a view, you can specify the font, style, size, and color for either the pane text or header text, or both. For example, in the view below, the header text is set to use the Tableau Bold font.



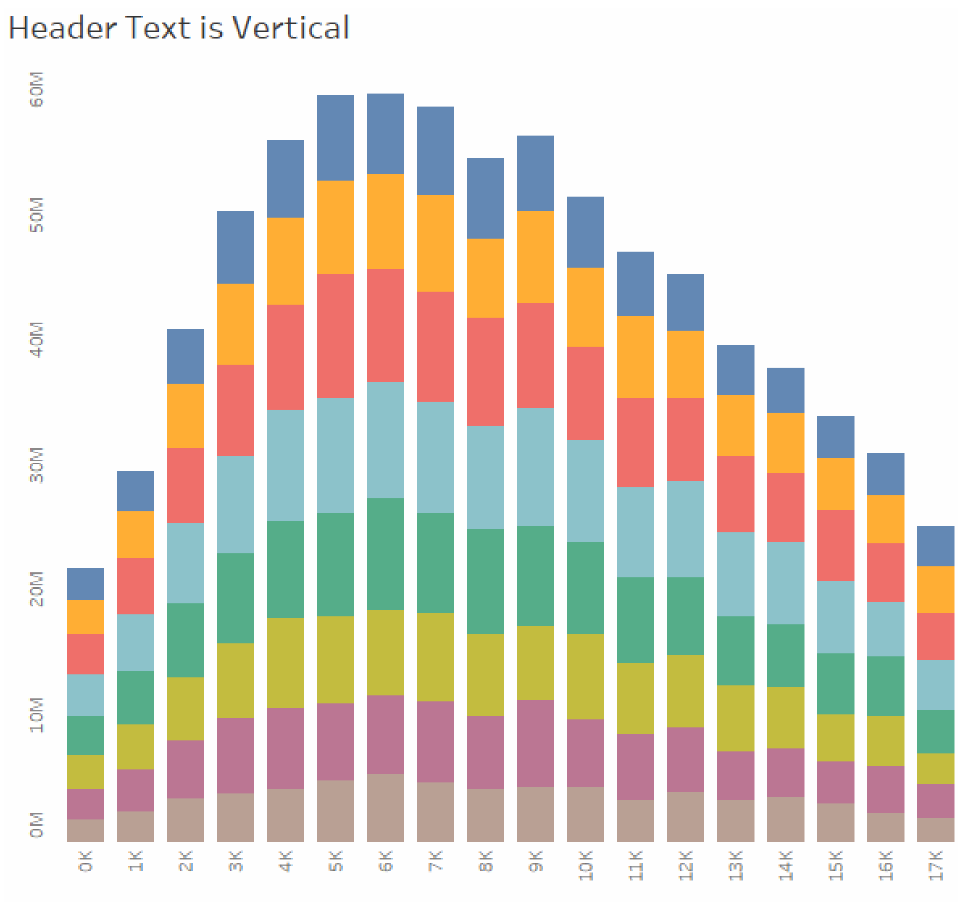
If you have totals or grand totals in the view, you can specify special font settings to make these values stand out from the rest of the data. This is particularly useful when you are working with a text table. The view below shows a text table in which the grand totals are formatted to be dark red.



Format text alignment

Tableau uses visual best practices to determine how text is aligned in a view, but you can also customize text. For example, you can change the direction of header text so that it is horizontal (normal) instead of vertical (up).

> **Note**: Tableau adheres to regional standards when determining when to begin or end line breaks.



Click the image to replay it.

For each text area you can specify the following alignment options:

Horizontal - Controls whether text aligns to the left, right, or center.

Vertical Alignment - Controls whether text aligns at the top, middle, or bottom.

Direction - Rotates text so that it runs horizontally (normal), top-to-bottom (up), or bottom-to-top (down).

Wrap - Controls whether long headers wrap to the next line or are abbreviated, but does not control text marks.

Note: If cells are not large enough to show more than one row of text, turning on wrapping will have no visible effect. If this happens, you can hover the cursor over a cell until a double-sided arrow appears, and then click and drag down to expand the cell size.

Format shading

Shading settings control the background color of the pane and headers for totals, grand totals, as well as for the worksheet areas outside those areas.

You can also use shading to add *banding*, alternating color from row to row or column to column. Banding is useful for text tables because the alternating shading helps your eye distinguish between consecutive rows or columns.

Table with Banding

Clinton St & Washington Blvd	1,033	2,873
Canal St & Jackson Blvd	942	2,355
Clinton St & Madison St	994	1,946
Franklin St & Jackson Blvd	1,649	1,829
Daley Center Plaza	1,147	1,772
Michigan Ave & Lake St	2,919	1,522
Dearborn St & Adams St	1,456	1,464
Clark St & Randolph St	956	1,447
Clinton St & Lake St	701	1,370
Dearborn St & Monroe St	838	1,353
State St & Kinzie St	1,957	1,304
Kingsbury St & Erie St	877	1,049

Click the image to replay it.

For row and column banding, you can use the following options:

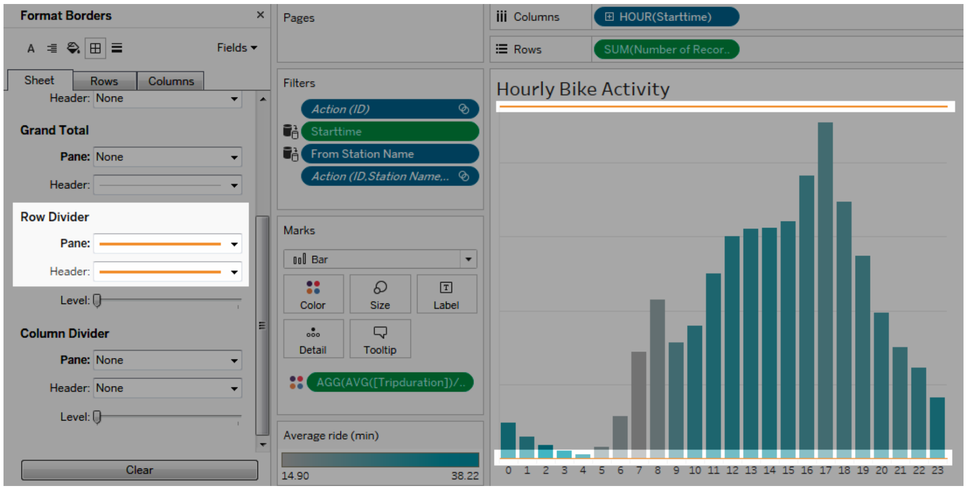
Pane and Header- The color the bands use.

Band Size - The thickness of the bands.

Level - If you have nested tables with multiple dimensions on the rows and columns shelves, this option allows you to add banding at a particular level.

Format borders

Borders are the lines that surround the table, pane, cells, and headers in a view. You can specify the border style, width, and color for the cell, pane, and header areas. Additionally, you can format the row and column dividers. For example, in this view the **Row Divider** borders are formatted to use an orange color:



Row and column dividers serve to visually break up a view and are most commonly used in nested text tables. You can modify the style, width, color, and level of the borders that divide each row or each column using the row and column divider drop-downs. The level refers to the header level you want to divide by.

Format lines

You can control the appearance of the lines that are part of the view, such as grid lines and zero lines, as well as lines that help you inspect data, such as trend lines, reference lines, and drop lines.

For example, you can set trend lines to use a red color and an increased thickness:

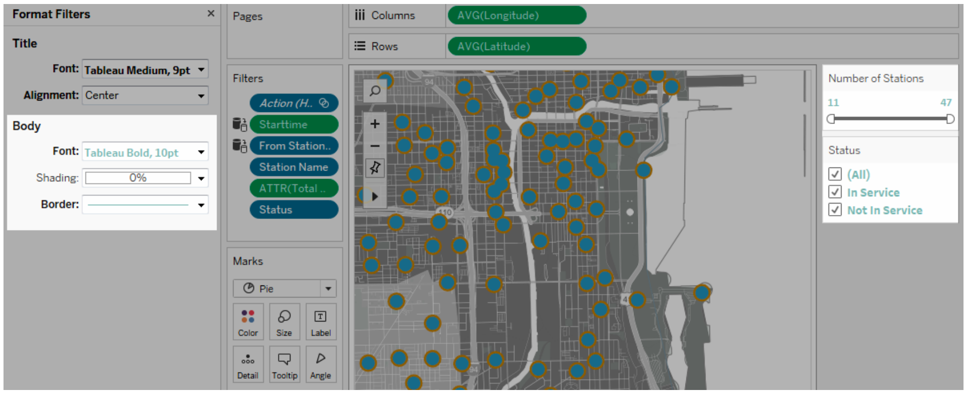
Format highlighters

The highlighter on your worksheet can be formatted to use a different font, style, color, background color, font size, and border. Formatting highlighters allows you to better integrate them into your dashboard or worksheet style. You can also edit the title that displays on each highlighter that shows in the view.

For more information about using highlighters, see **Highlight Data Points in Context** on page 1502.

Format a filter card

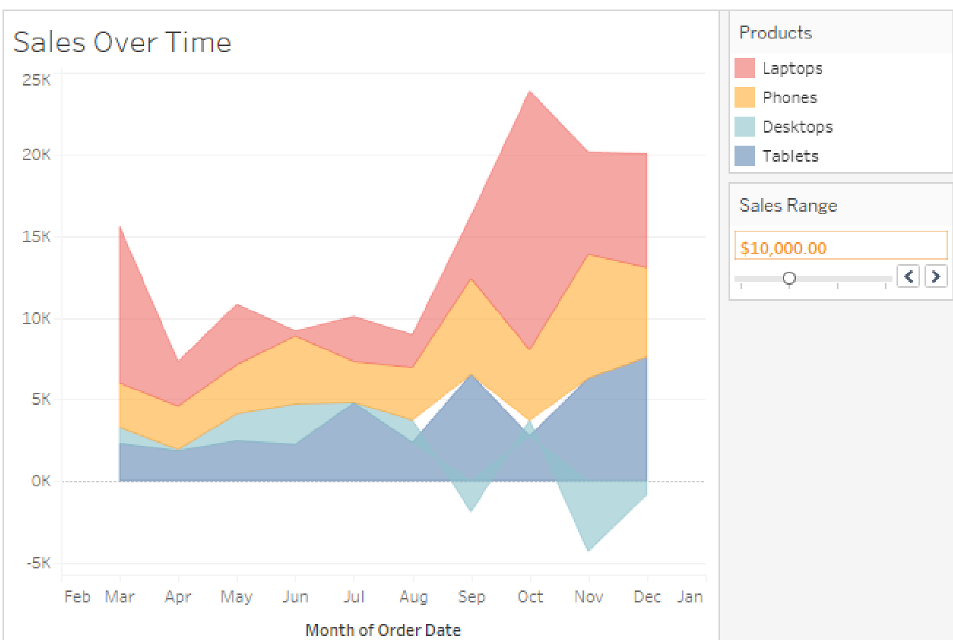
Filter cards contain controls that let users interact with your view. You can change filter cards to use custom formatting. For example, the body text in the filters below is formatted to use the Tableau Bold font, in aqua.



Note: For filters and parameters, title formatting appears only on dashboards or in views published to the web.

Format a parameter control card

Parameter controls are similar to filter cards in that they contain controls that let users modify the view. If you create a parameter control, you can customize how it looks. For example, in the view below, the Sales Range parameter is formatted so that the sales amount appears in orange.



Copy and paste worksheet formatting

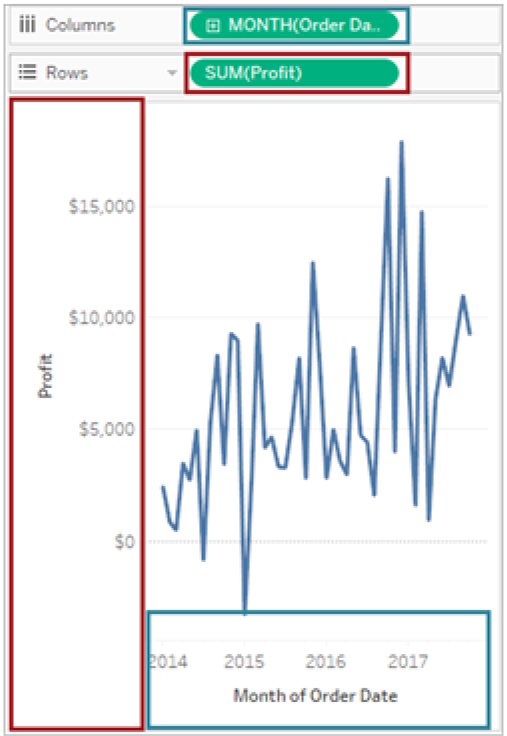
After you format a worksheet, you can copy its formatting settings and paste them into other worksheets. The settings that you can copy are anything you can set in the **Format** pane, with the exception of reference lines and annotations. Adjustments like manual sizing and level of zoom are not copied.

1. Select the worksheet from which you want to copy formatting.
2. Right-click (control-click on Mac) the worksheet tab and select **Copy Formatting**.
3. Select the worksheet you want to paste the formatting into.
4. Right-click (control-click on Mac) the worksheet tab and select **Paste Formatting**.

Edit Axes

When you add a continuous field (a field with a green background) to the Columns or Rows shelf, it adds an axis to the view. An axis shows data points that lie within a range of values. For each axis, you can specify the range, scale, and tick mark properties.

Double-click an axis to open the **Edit Axis** dialog box and change the axis configuration and formatting. If you can't edit the axis, see **Why can't I edit my axis?** on page 2902

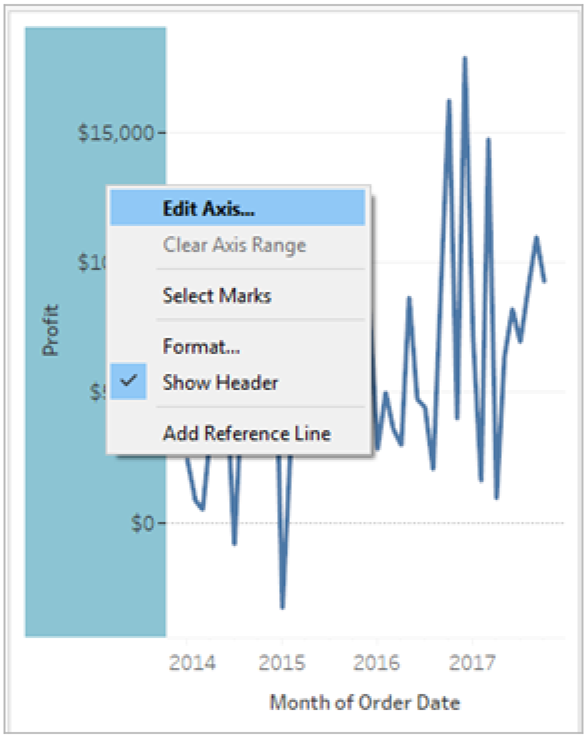


In this example, the Profit field (a continuous measure) on the Rows shelf creates a vertical axis, and the Order Date field (a continuous date dimension) on the Columns shelf creates a horizontal axis.

Note: For more information on formatting headers and field labels, see **Format at the Worksheet Level** on page 2892 and **Format Fields and Field Labels** on page 2943. For details on continuous and discrete fields, see **Dimensions and Measures, Blue and Green** on page 314.

Edit an axis range

To edit an axis range, double-click the axis that you want to edit.



Note: In Tableau Desktop, you can right-click (control-click on Mac) the axis, and then select **Edit Axis**. In web authoring, you can click the arrow button on an axis, and then select **Edit Axis**.

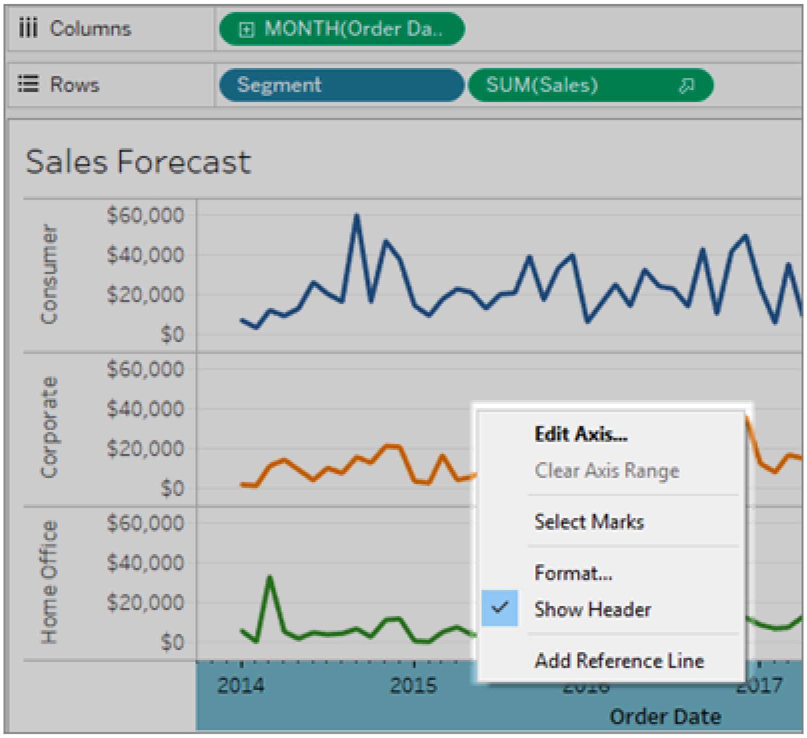
When you select an axis, the marks associated with the axis are not selected so that you can edit and format the axis without modifying the marks.

To select the marks associated with the axis, right-click the axis and select **Select Marks**. You can also select each mark individually. For more information, see **Select marks** on page 2722.

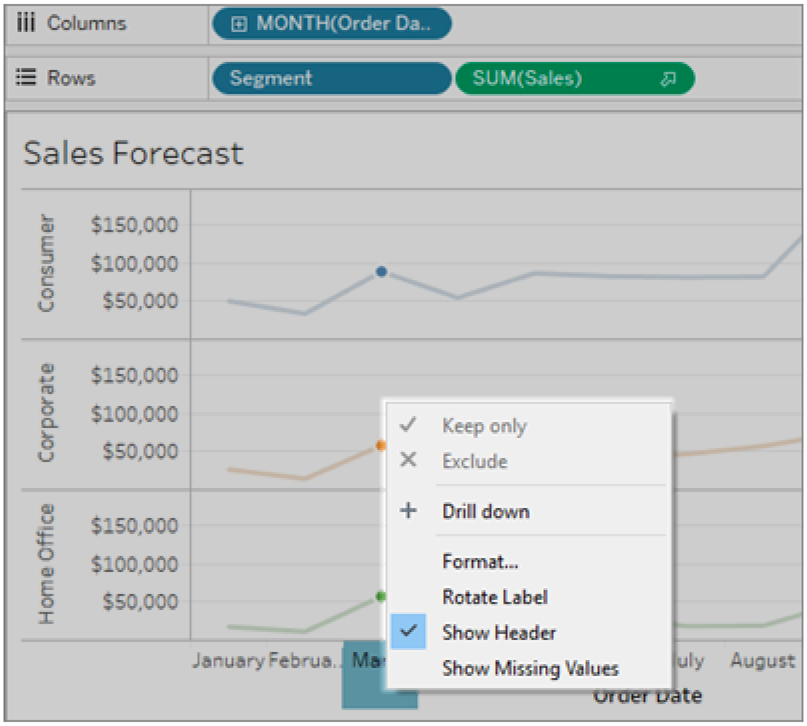
Why can't I edit my axis?

If you can't edit an axis, it's most likely a header rather than axis.

- Continuous fields (green background) on the Rows and Columns shelves create axes in the view. When you right-click an axis, you will see this menu:



- Discrete fields (blue background) on the Rows and Columns shelves create headers, not axes. When you right-click a header, you will see this menu:



You can edit the formatting of headers by right-clicking a header, and then selecting **Format**.

For more details on formatting headers and field labels, see **Format at the Worksheet Level** on page 2892 and **Format Fields and Field Labels** on page 2943. For details on continuous and discrete fields, see **Dimensions and Measures, Blue and Green** on page 314.

Hide and reshow axes and headers

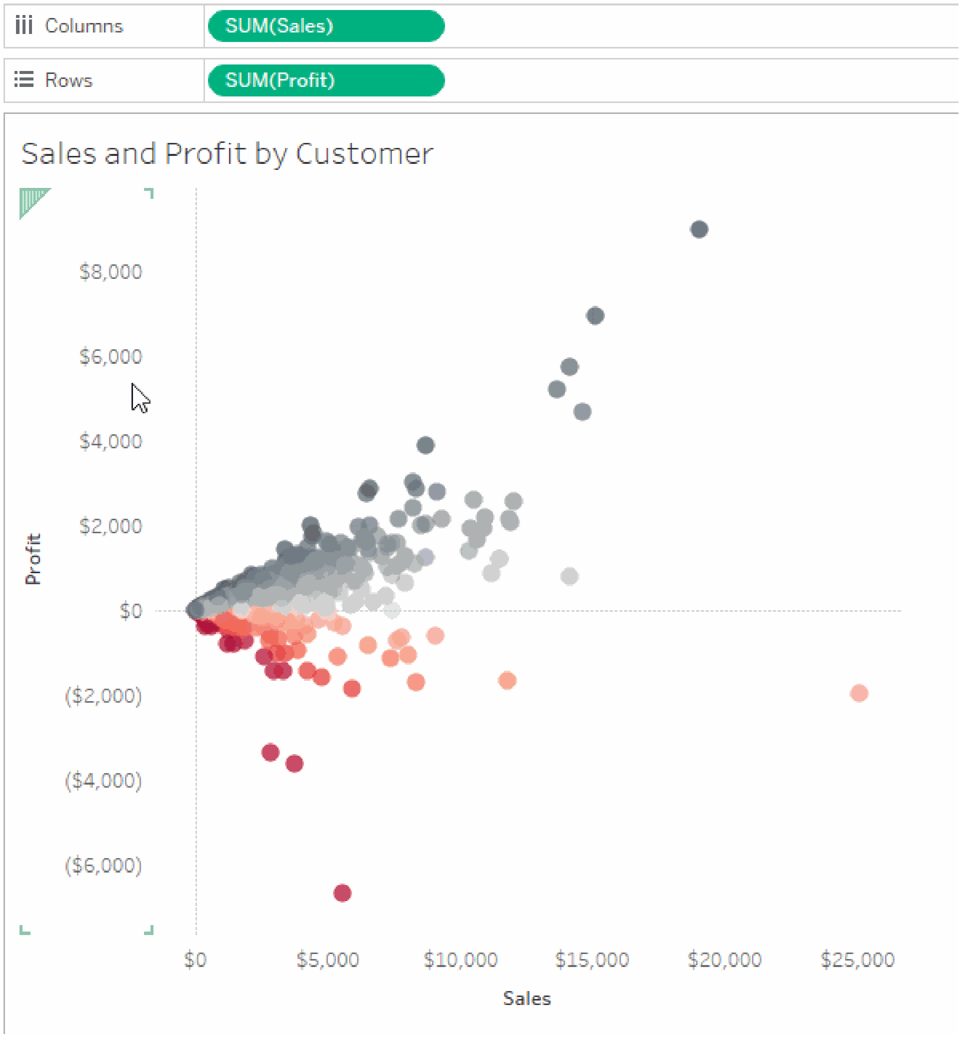
If you have hidden an axis or a header in the view, you can show it again from its related field in the view.

To hide an axis

Right-click (control-click on Mac) the axis in the view, and then clear the check mark next to the **Show Header** option.

To show a previously hidden axis

Right-click (control-click on Mac) the continuous field in the view that is associated with axis you want to show, click its drop down menu, and then click **Show Header**.



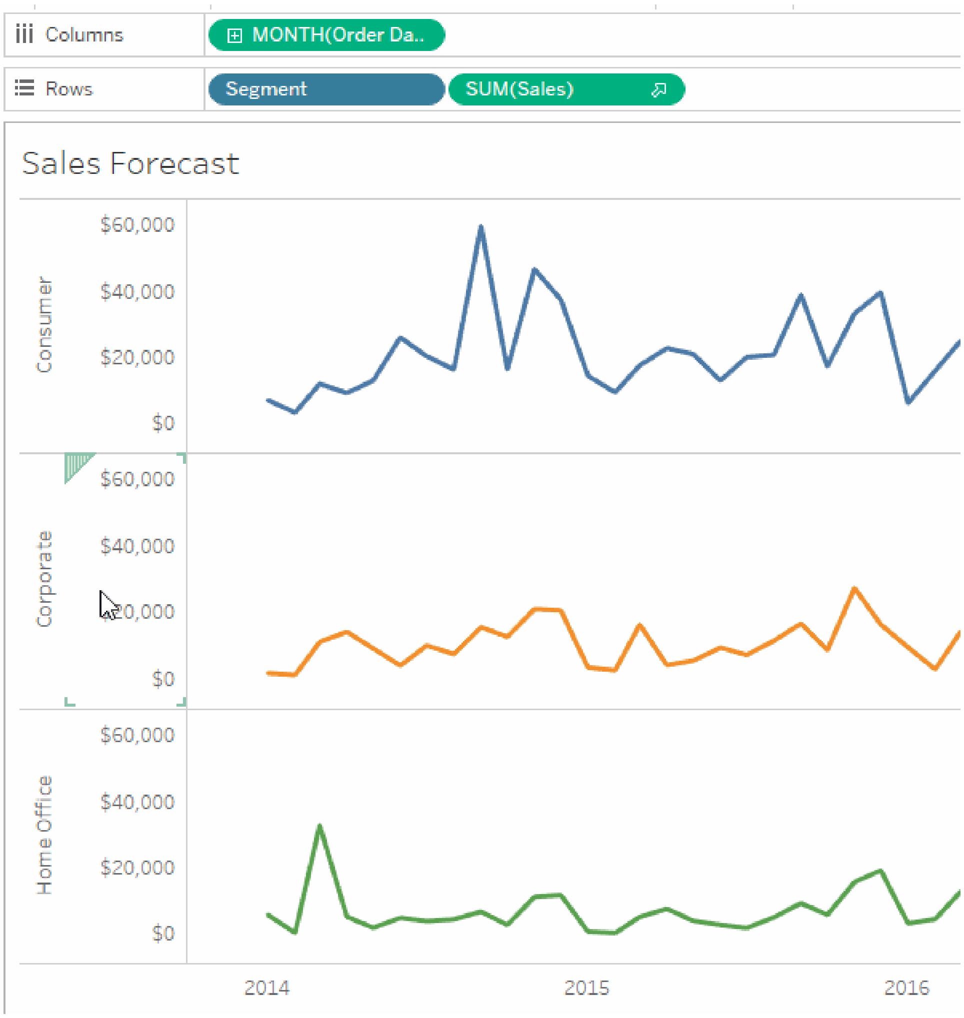
For related details, see **Axes** on page 246 in **Parts of the View** on page 241.

To hide a header

Right-click (control-click on Mac) a header in the view, and then clear the check mark next to the **Show Header** option.

To show previously hidden header

Right-click (control-click on Mac) the discrete field in the view that is associated with headers that you want to show, click its drop down menu, and then click **Show Header**.



For related details, see **Headers** on page 242 in **Parts of the View** on page 241.

For details on continuous and discrete fields, see **Dimensions and Measures, Blue and Green** on page 314.

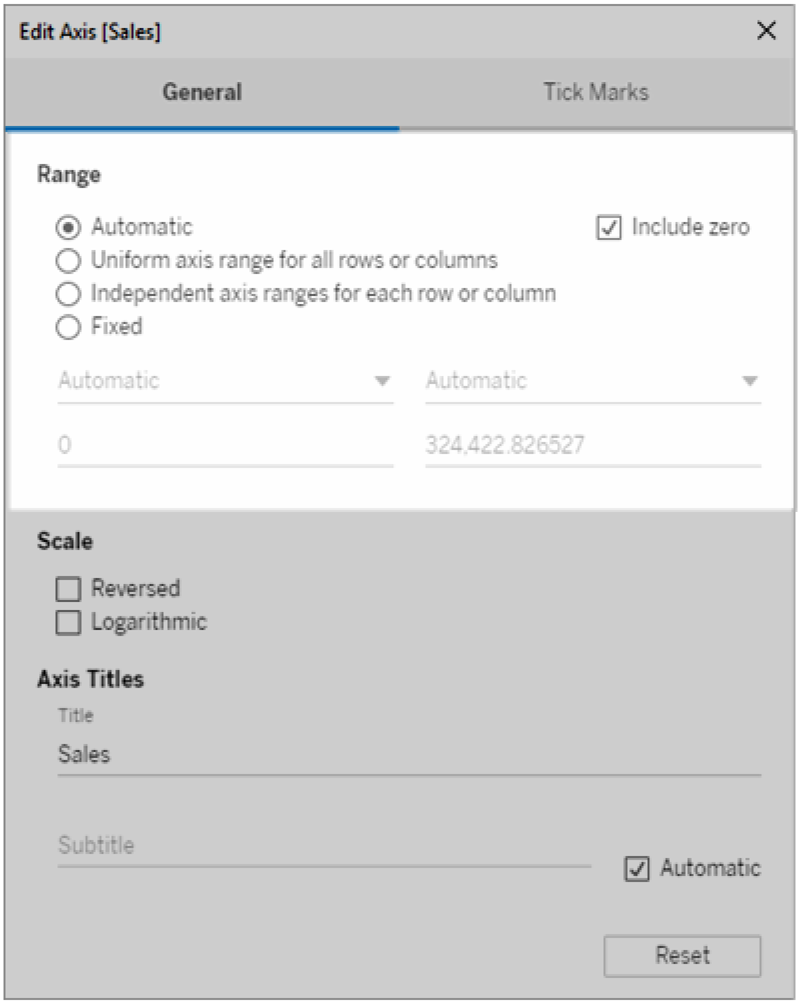
Change an axis range

You can limit the axis range to focus the view on certain data points.

For example, assume your view shows sales over four years. The automatic axis might show a range from 0 to $750,000. If sales never go below $470,000, you can adjust the axis range to start at $470,000 to keep the focus on where the data points actually lie.

To change an axis range:

1. Double-click the axis that you want to edit. You can also right-click (control-click on Mac) the axis, and then select **Edit Axis**.
2. In the **Edit Axis** dialog box, select one of the following options:
   - **Automatic.**Automatically bases the axis range on the data used in the view.
   - **Uniform axis range for all rows or columns.**Sets the axis range uniformly to the maximum data range for all panes in the view.
   - **Independent axis ranges for each row or column.** Makes the axis range independent. The axis range varies for each pane in the view, depending on the range of data in each pane.
   - **Fixed.** Specifies to start and/or end the axis at a specific value. You can fix both ends of the axis or only one end. When you fix both ends of the axis, the axis range is determined by the values you specify. When you fix only one end of the axis, you must set the other end of the axis as automatic, uniform, or independent.
3. You can also specify whether to include zero. When you clear the **Include zero** check box, the axis range adjusts to show only the range of values in the data.
4. Click **OK**.



Change the axis scale to reversed or logarithmic

Axis scale options include **Reversed** and **Logarithmic**.

When you select **Logarithmic**, you can then specify **Positive** or **Symmetric**. Symmetric displays data that contains positive, 0, or negative values on a log scale axis, and is best used to visualize large negative values on a logarithmic scale, as well as large positive values, or both. For related details, see **About the symmetric log axis transform** on page 2910.

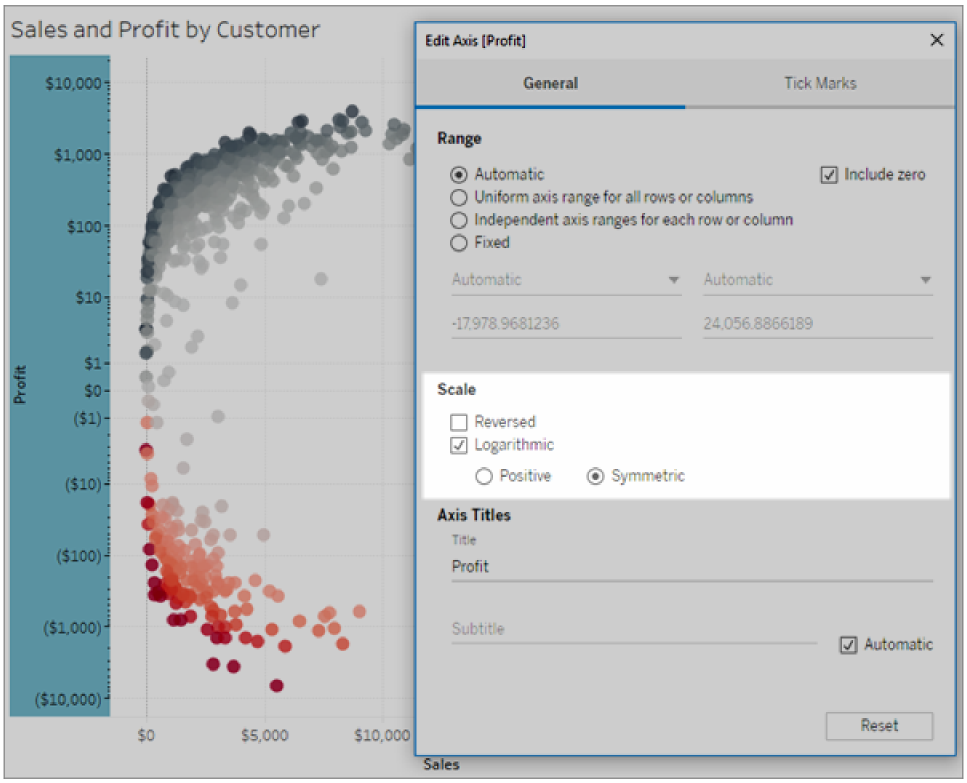
To change the scale of an axis:

1. Double-click the axis that you want to edit. You can also right-click (control-click on Mac) the axis, and then select **Edit Axis**.
2. In the **Edit Axis** dialog box, on the General tab, select **Reversed** or **Logarithmic** for

the axis scale.

The option to choose Symmetric or Positive is only available when the axis range extends outside of the closed interval [-10, 10].

Logarithmic: Select **Positive** or **Symmetric** for the log scale. The **Symmetric** option displays data that contains positive, 0, and negative values on a log scale axis. By default, the tick marks are drawn at powers of ten, but you can specify any base that is greater than 1.



3. Click the **Tick Marks** tab and select one of the following options:

   - **Automatic** - the major tick marks are drawn at powers of 10.
   - **Fixed** - the major tick marks are drawn at a specified exponent. Type a number into the Powers of text box.
   - **None** - major tick marks are not shown.

4. When finished, click **OK**.

You can also reverse the axis by selecting **Reverse** in the Scale area on the General tab of the Edit Axis dialog box.

About the symmetric log axis transform

The symmetric log axis applies an area hyperbolic sine transform to the axis:

```
asinh(x) = ln(x + √(x^2 +1))
```

For large values (|x| >> 1),

```
asinh(x)~ln(x), x > 0
```

```
asinh(x)~-ln(x), x < 0
```

Because *asinh* is defined for all real values (negative numbers in particular) this transform allows you to visualize large negative values on a logarithmic scale, as well as large positive values, or both.

However, while a log axis can be used to visualize the scale of very small values, a symmetric log axis cannot be used to visualize the scale of very small values.

For small values (|x| < 1),

```
asinh(x) ≈ x
```

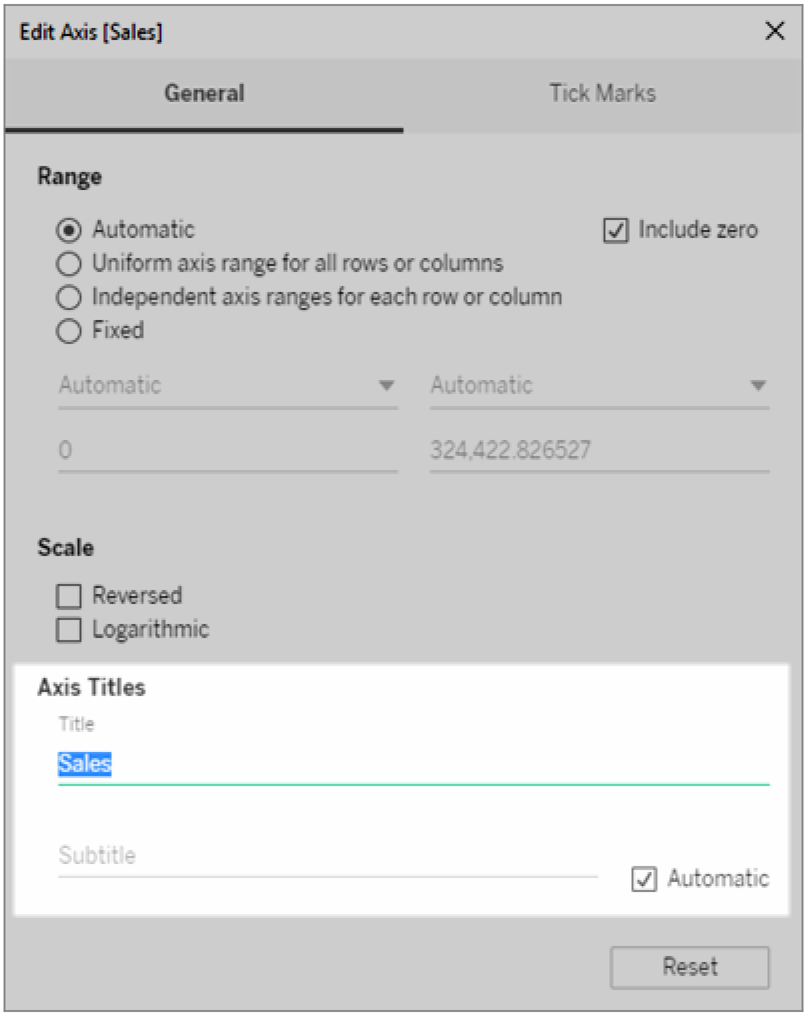
Using a symmetric log axis scale for small values has a similar effect to using a linear scale (i.e., a normal quantitative axis).

Change the appearance of an axis

Every axis has a title that is automatically generated based on the fields in the view. You can specify a custom axis title and add a subtitle using the Edit Axis dialog box. You can also specify the scale of the axis, such as whether to use a logarithmic scale or whether to reverse the axis.

To change the appearance of an axis:

1. Right-click (control-click on Mac) on the axis that you want to edit and select **Edit Axis**.
2. In the Edit Axis dialog box, type a new title in the **Title** text box.

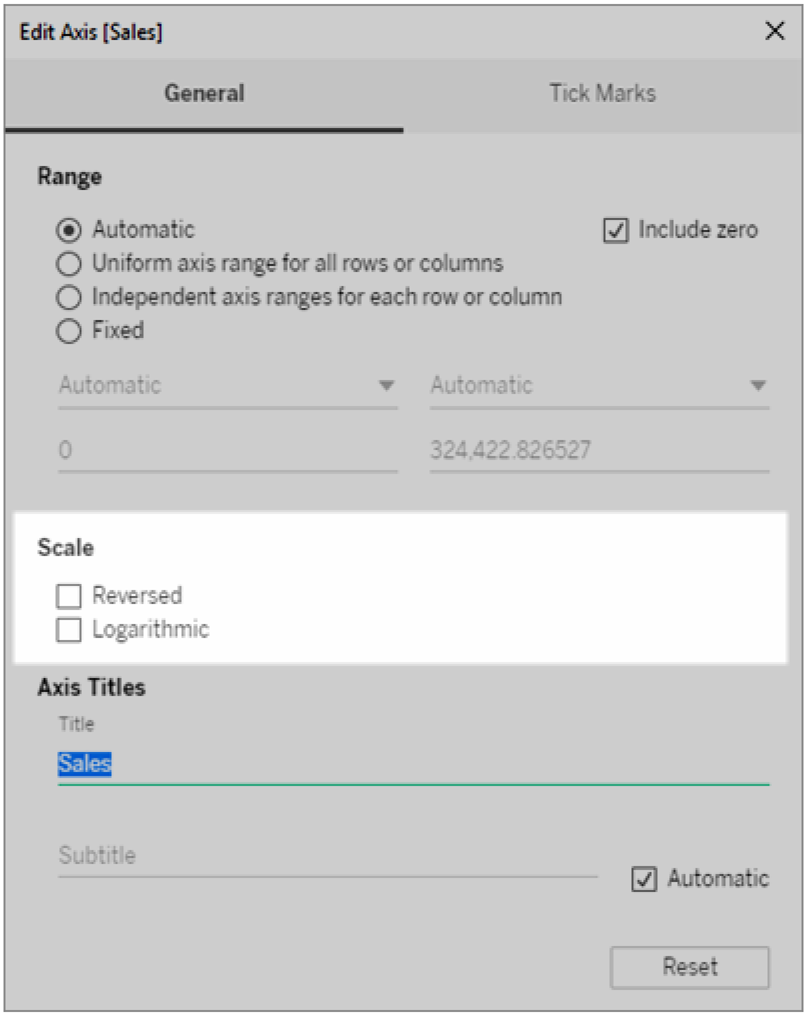


3. To add a custom subtitle, clear the **Automatic** check box.

4. To change the scale of the axis, select one of the following options:

   - **Reversed**- select this option to reverse the order of values on the axis.

   - **Logarithmic** - select this option to use a logarithmic scale on the axis.



5. Click **OK**.

Format tick marks

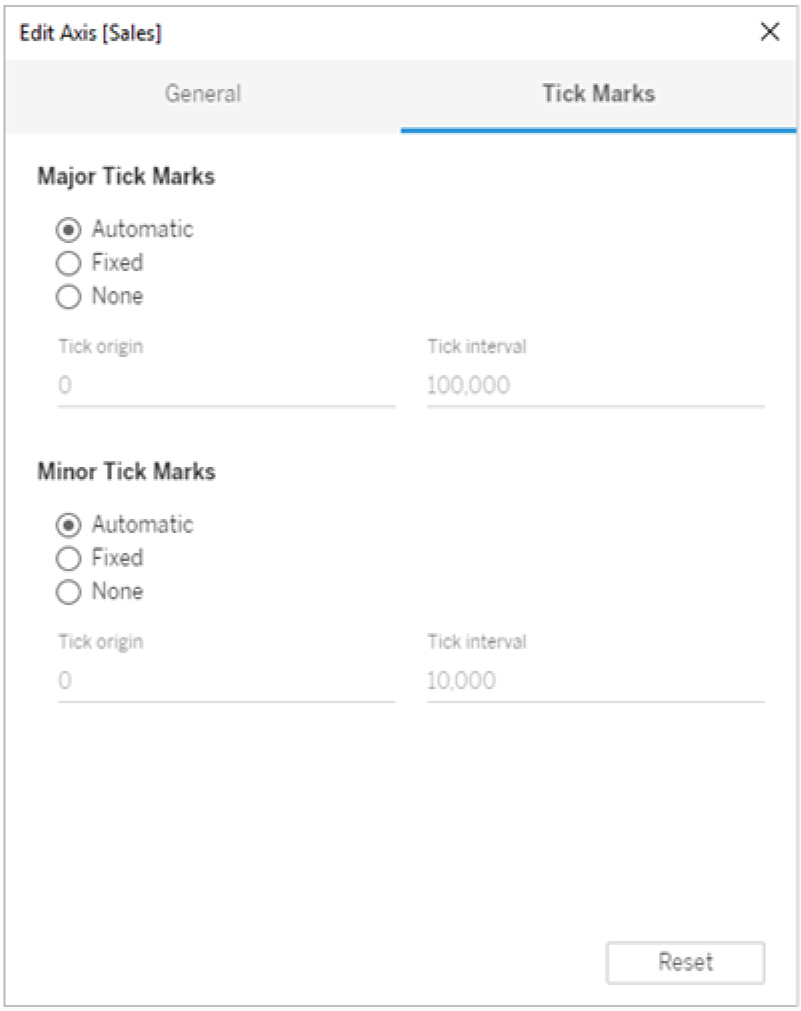
You can specify how often the tick marks are displayed along the axis. Tableau allows you to modify both the Major and Minor tick marks. Major tick marks are accompanied by unit labels while Minor tick marks simply represent smaller increments between the major marks. You can choose to use automatic or fixed tick marks or have none at all.

To format tick marks:

1. Right-click (control-click on Mac) the axis you want to edit and select **Edit Axis**.
2. In the Edit Axis dialog box, select the **Tick Marks** tab.
3. For both **Major** and **Minor tick marks**, select from one of the following options:

- **Automatic** - select this option to automatically show tick marks based on the data in the view.
- **Fixed** - select this option to specify how often the tick mark should display and the starting value.
- **None** - select this option to hide the tick marks completely.

4. Click **OK**.



Examples: Use Different Axis Ranges (Uniform, Independent, Fixed)

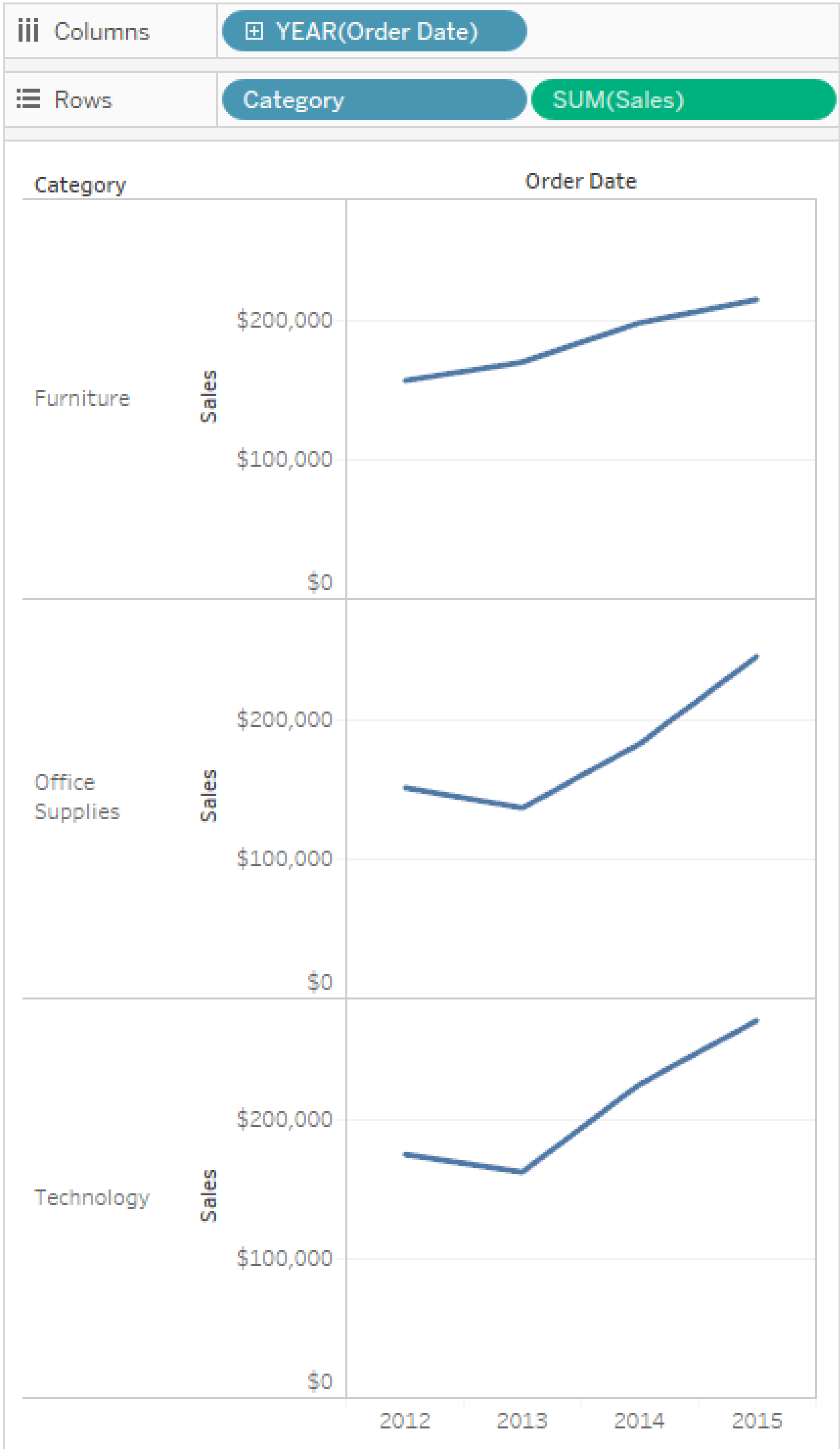
In this example you will build three views using the same data. However, each view will use a different axis range format. These views use the **Sample-Superstore** data source to display the aggregated total sales for three product categories over the course of four years.

Build a view with a uniform axis range

A uniform axis range means that the same range is applied to each row or column in your view. The range is automatically generated based on the underlying data values.

1. Place the **Order Date** dimension on the **Columns** shelf and the **Category** dimension on the **Rows** shelf.
2. Place the **Sales** measure on the **Rows** shelf.

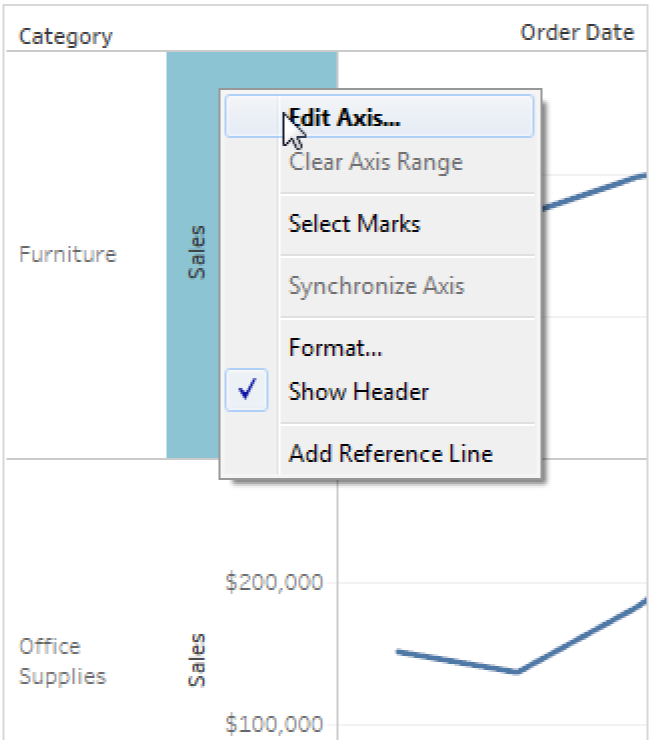
   The measure is automatically aggregated as a summation and an axis is added to the view. By default the view uses a uniform axis range. Notice that the axis range is the same, from zero to 250,000 for each product category.



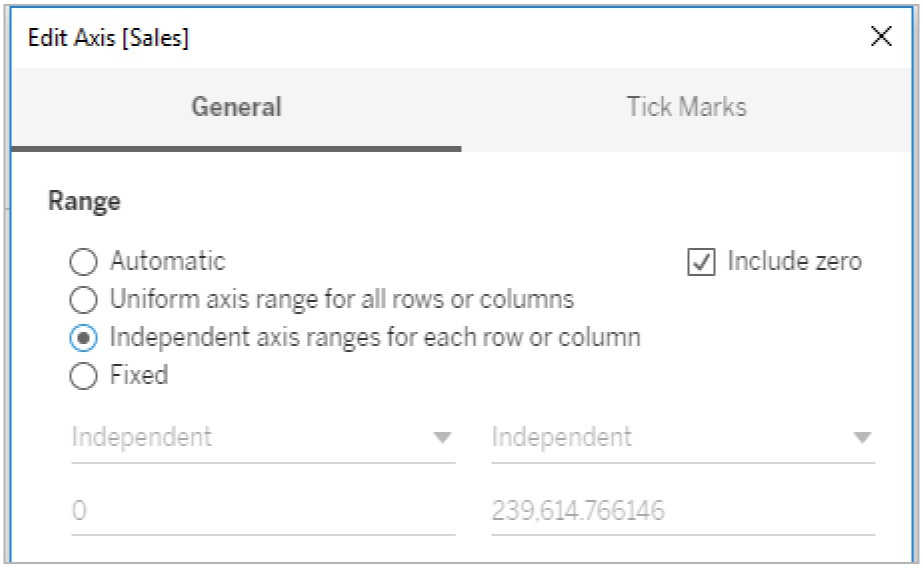
Build a view using independent axis ranges

When you use an independent axis range, each row or column will have its own axis range based on the underlying data values.

1. Right-click (control-click on Mac) the SUM(Sales) axis in the view and select **Edit Axis**.

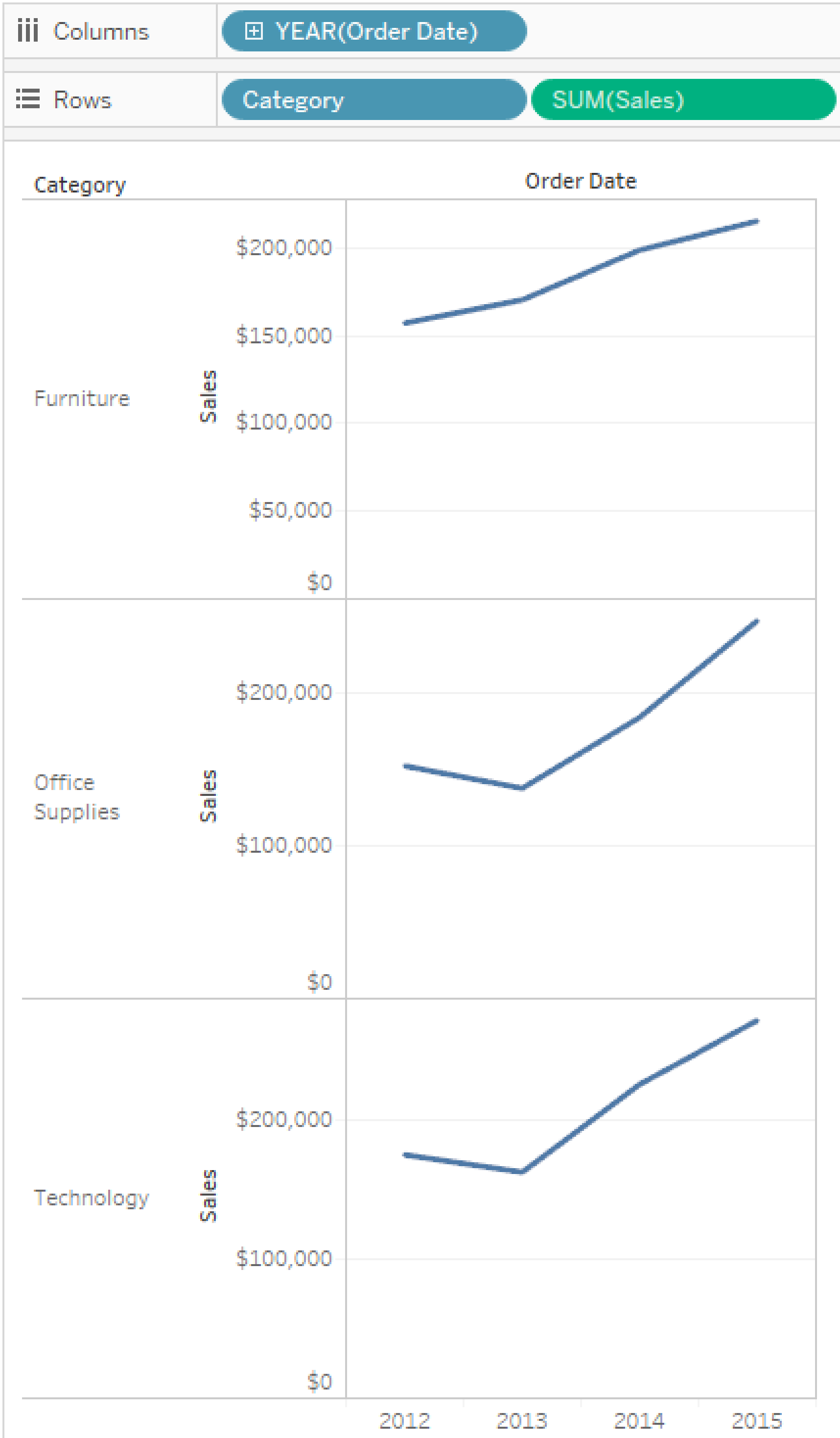


2. In the Edit Axis dialog box, select **Independent axis ranges for each row or column**.



3. Click the X to close the dialog box with the current settings.

   The axis range for each product category are now independent from each other. The Technology and Office Supplies categories still range from zero to 250,000, but the Furniture category ranges from zero to a little over 200,000.


Columns
YEAR(Order Date)
Rows
Category
SUM(Sales)
Category
Order Date
Furniture
Office Supplies
Technology
Sales
$200,000
$150,000
$100,000
$50,000
$0
2012
2013
2014
2015

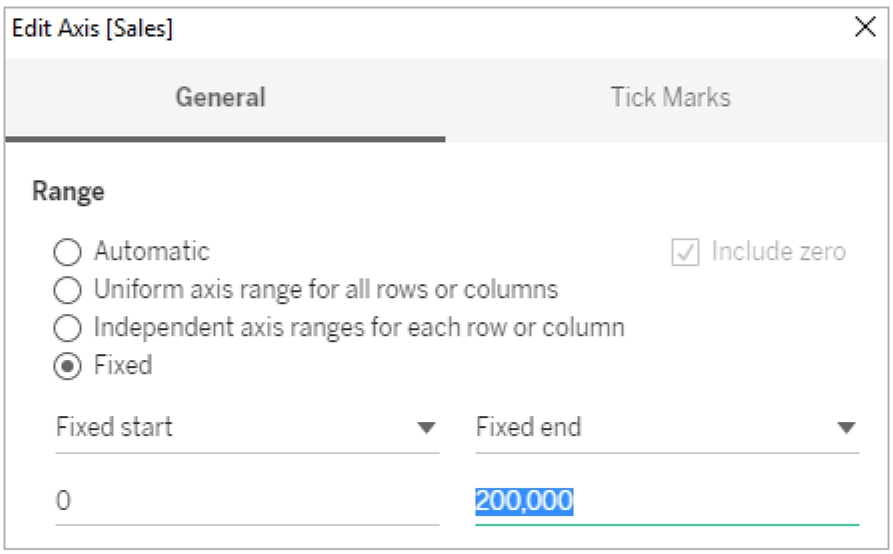
Build a view using a fixed axis range

You can fix one or both ends of an axis in order to customize how much of the data you can see in the view.

The first example shows how to fix both ends of the axis. The second shows how to fix only one end of the axis, and then define the axis range for all panes in the view.

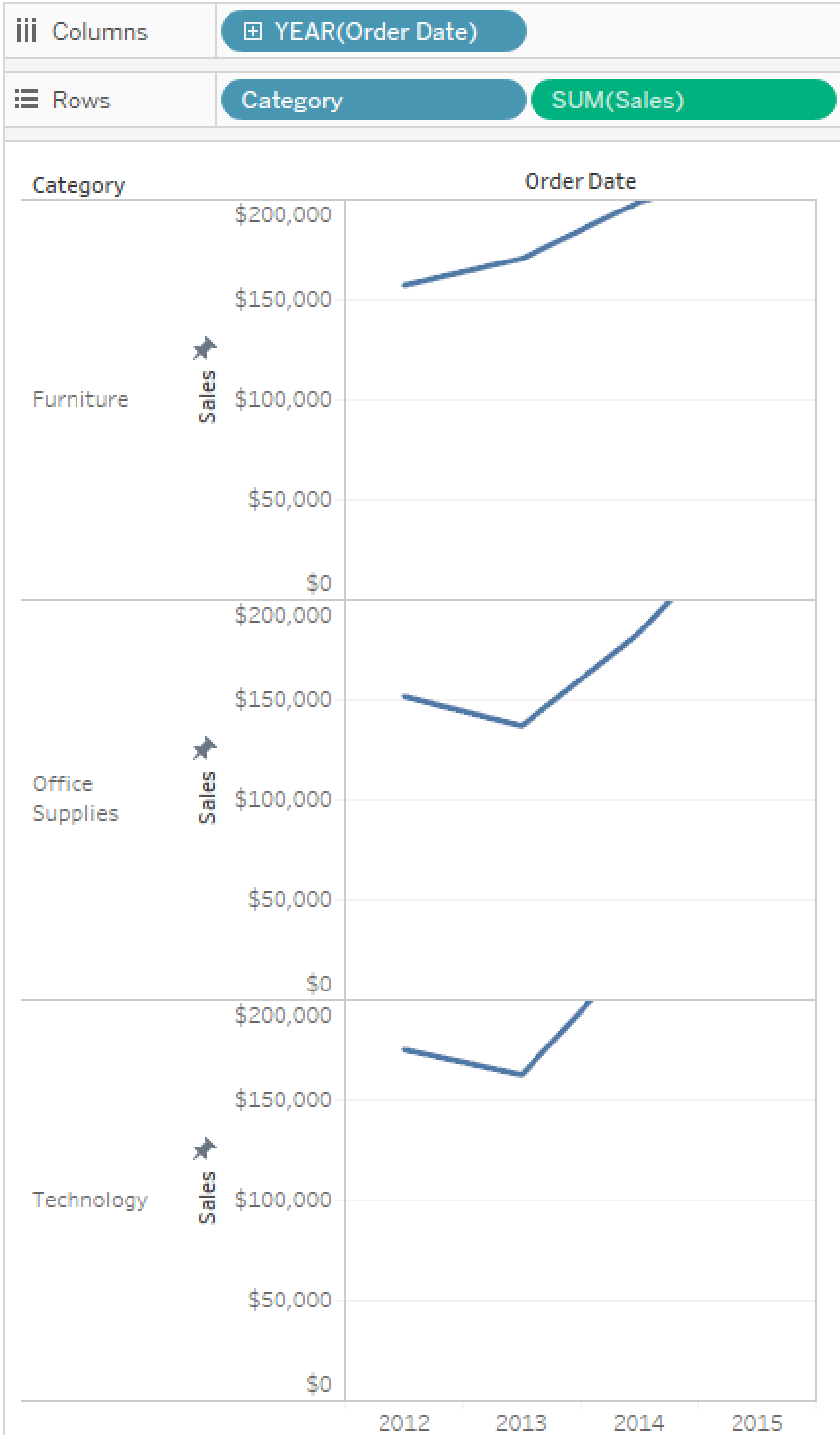
To fix both ends of the axis:

1. Right-click (control-click on Mac) the **SUM(Sales)** axis in the view and select **Edit Axis**.
2. In the Edit Axis dialog box, select **Fixed**, and then define **Start** and **End** values either by typing into the text boxes or by dragging the sliders toward each other. For this example, type **0** for **Start** and **200,000** for **End**.



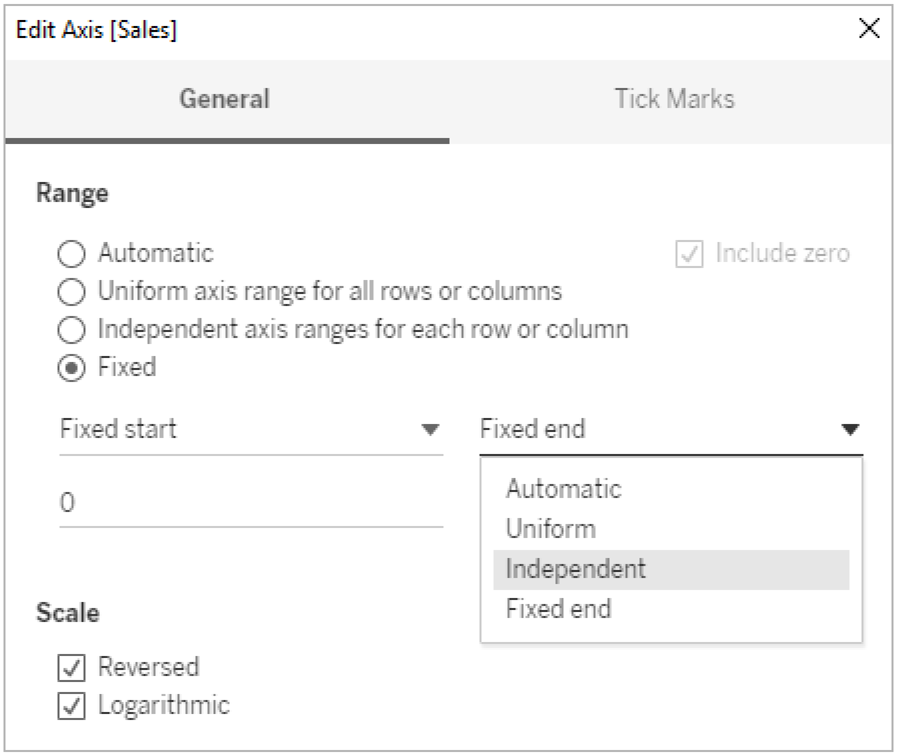
3. Click the X to close the dialog box with the current settings.

   The categories now display only up to $200,000 in sales. The axes are marked with a pin symbol , which indicates that you have limited the axis range and that some data might not be displayed.



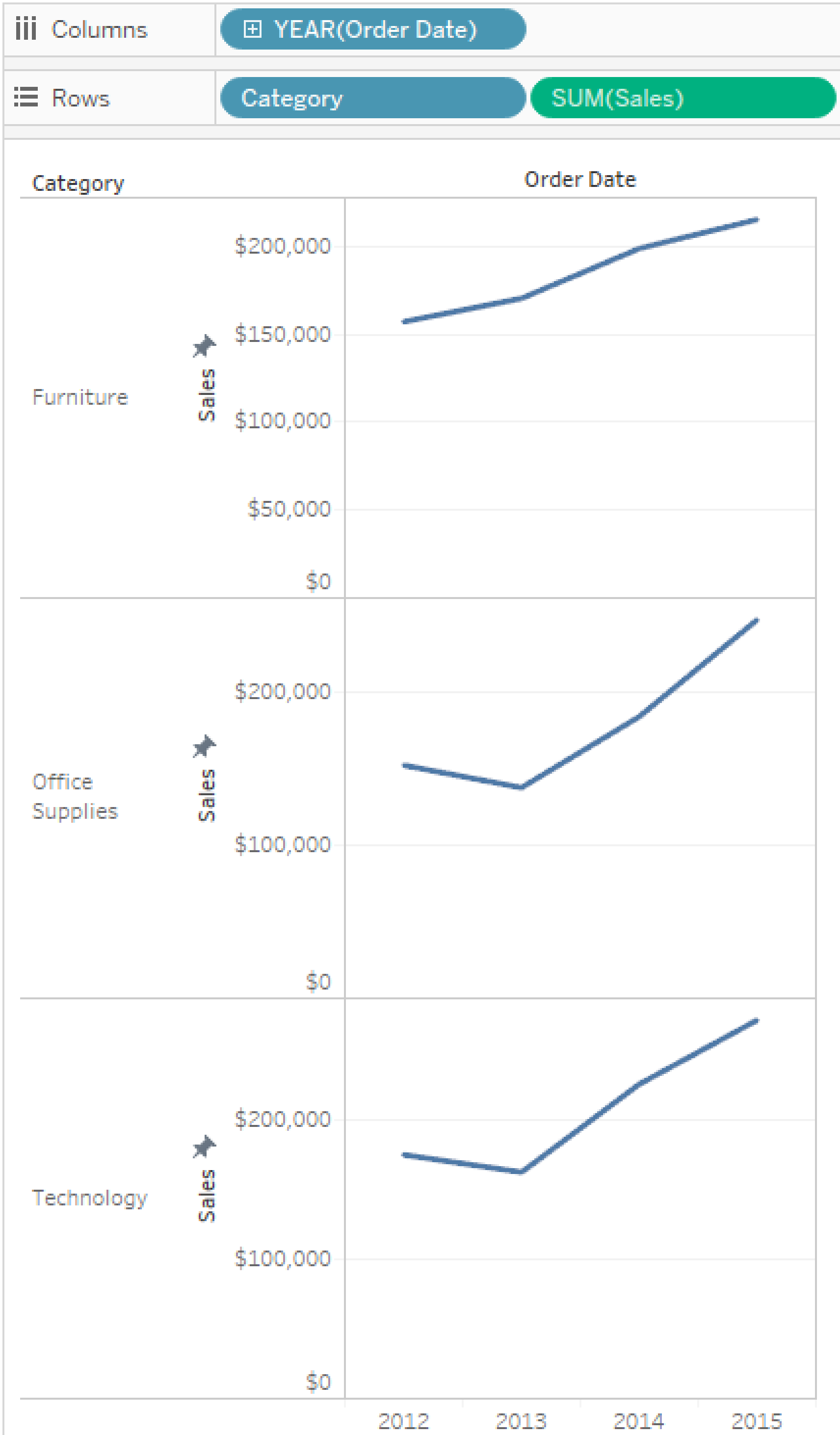
To fix one end of the axis:

1. Right-click (control-click on Mac) the **SUM(Sales)** axis in the view and select **Edit Axis**.
2. In the Edit Axis dialog box , select **Fixed**, click the **Fixed End** drop-down menu, and then select **Independent**.



3. Click the X to close the dialog box with the current settings.

   Notice that the categories now have slightly different axis ranges. For example, Office Supplies has an axis range from 0 to 250,000 dollars, while Furniture only has a range from 0 to 200,000 dollars.



For more information about the difference between automatic, uniform, and independent axis ranges, see **Edit Axes** on page 2900.

Format Text and Numbers

To learn how to format specific items, click the links below.

Format Text

The proper font selection can take your visualization from good to great, and Tableau is here to help. Whether you're changing the font size, position, color, or direction, Tableau provides a wide range of customization options for your text.

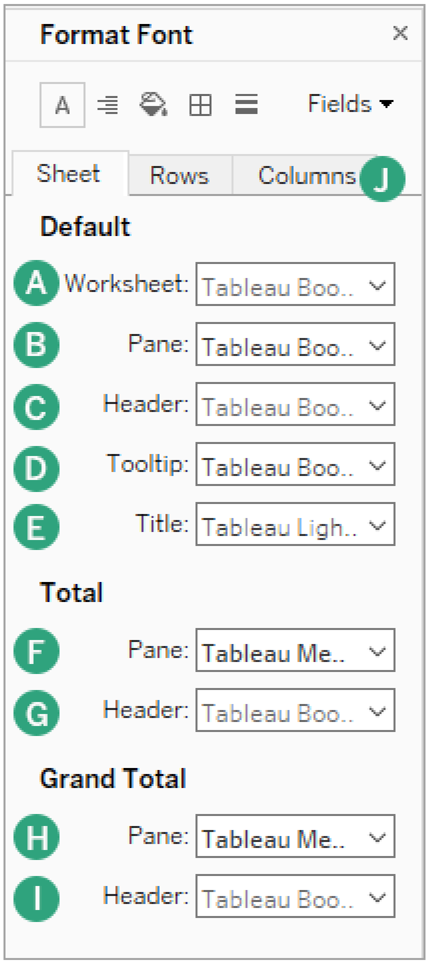
When you make formatting changes at this level, they apply only to the view you're working on. See **Format at the Workbook Level** on page 2882 for how to make changes that apply to every view in your workbook.

To access worksheet-level format settings, select the **Format** menu, then choose the part of the view, such as **Font,** or **Border,** that you want to format.

Format fonts

Select **Format** > **Font** . A **Format Font** control pane will open. This control pane will let you customize the text in the view.

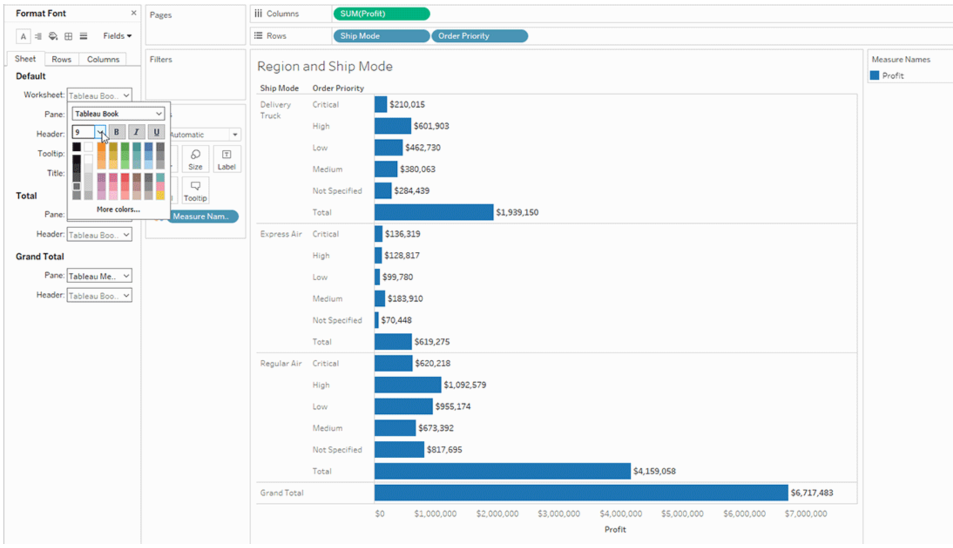
Click the letters below for more information.



A. Worksheet

Editing the font at the worksheet level applies the changes you make to all text fields in view except the Header and Tooltip fields.

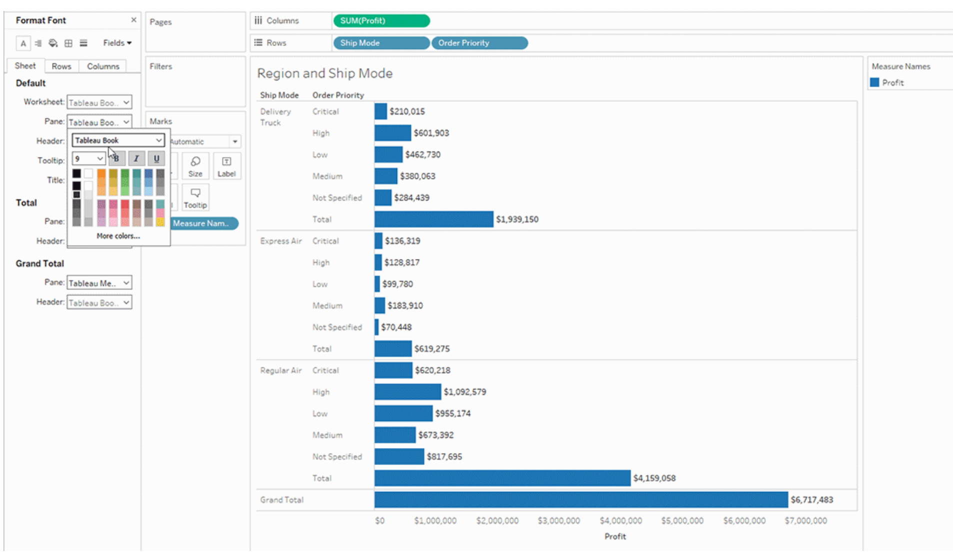
> **Note**:Tableau treats Worksheet formatting as a new default, and any changes to any other fields (such as Title, Pane, or Header) will overwrite the Worksheet format.



Back to top.

B. Pane

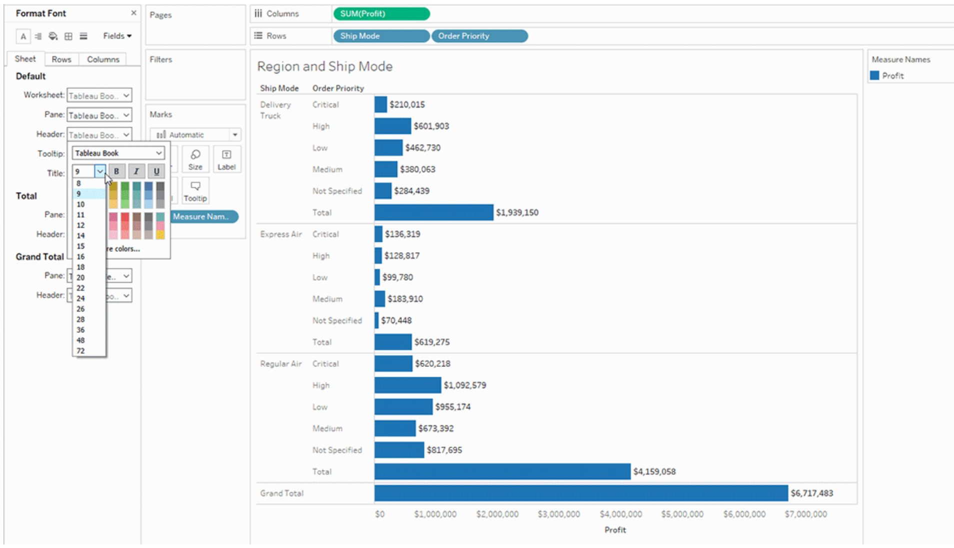
Adjusting the font here will effect the pane view, which is where Tableau displays the results of your visualizations.



Back to top.

C. Header

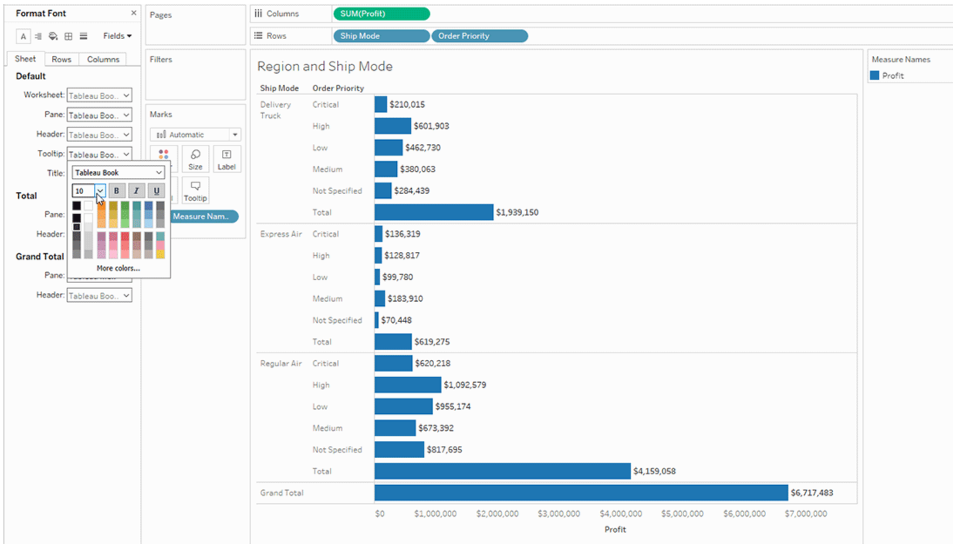
Select this option to adjust font for the headers in your view. Headers display the dimensions used in your viz.



Back to top.

D. Tooltip

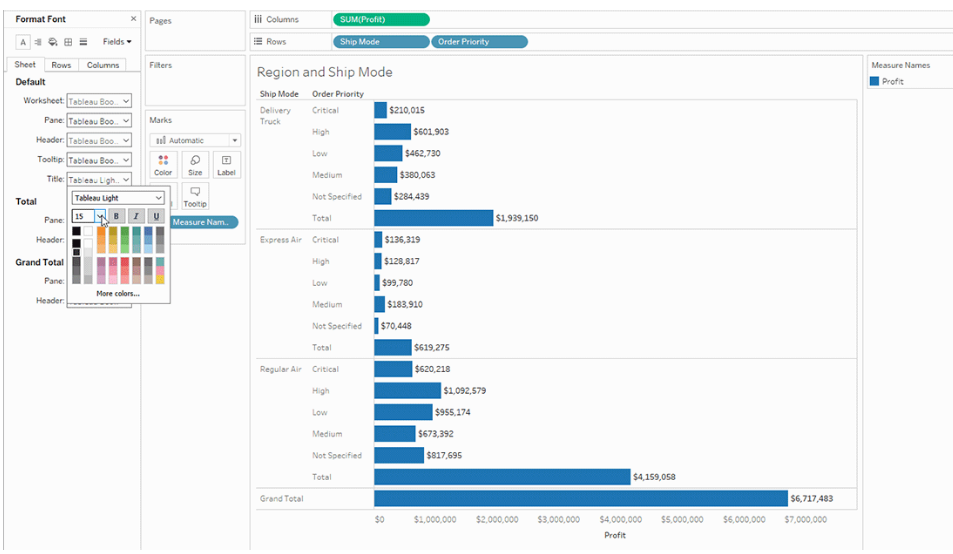
Adjusting the font here will apply changes to all the text in the tooltip, the field that pops up when someone hovers over a mark in your view. If you want more control over the fields in the tooltip, see **Format tooltips** on page 2932



Back to top.

E. Title

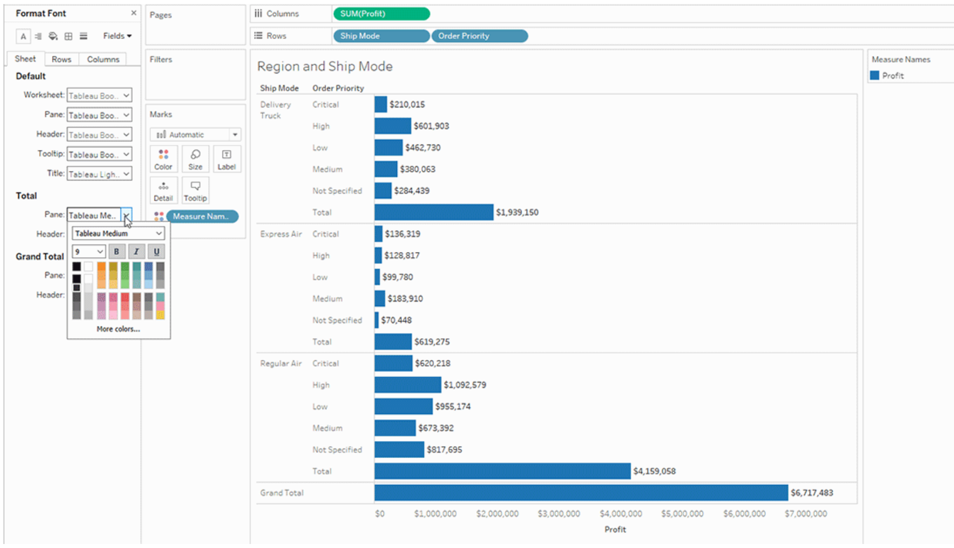
Select this formatting option to quickly adjust the appearance of the Title. For more information, see **Edit worksheet titles, captions, and legend titles** on page 2933



Back to top.

F. Total Pane

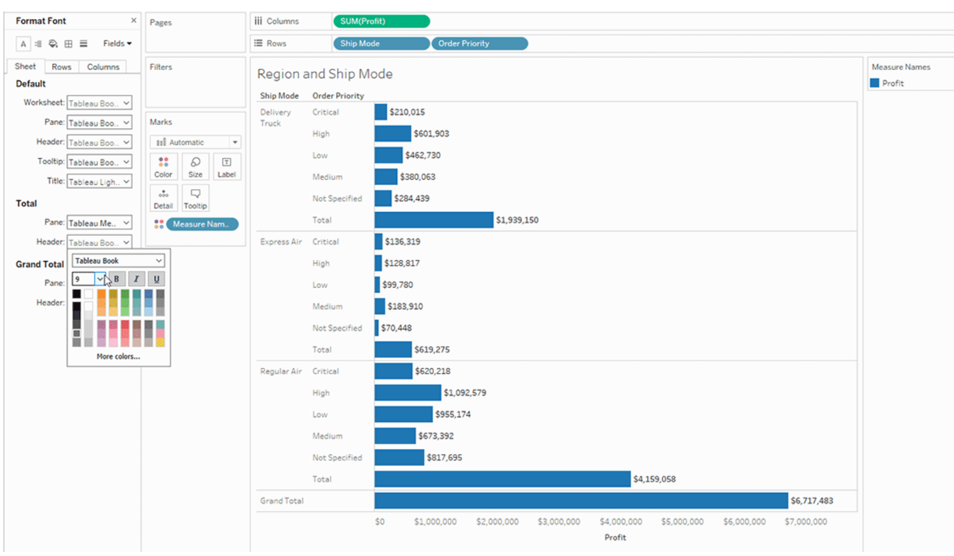
Adjusting the font for totals at the pane level will change both the subtotals and grand totals. For information on setting up totals, see **Show Totals in a Visualization** on page 1456.



Back to top.

G. Total Header

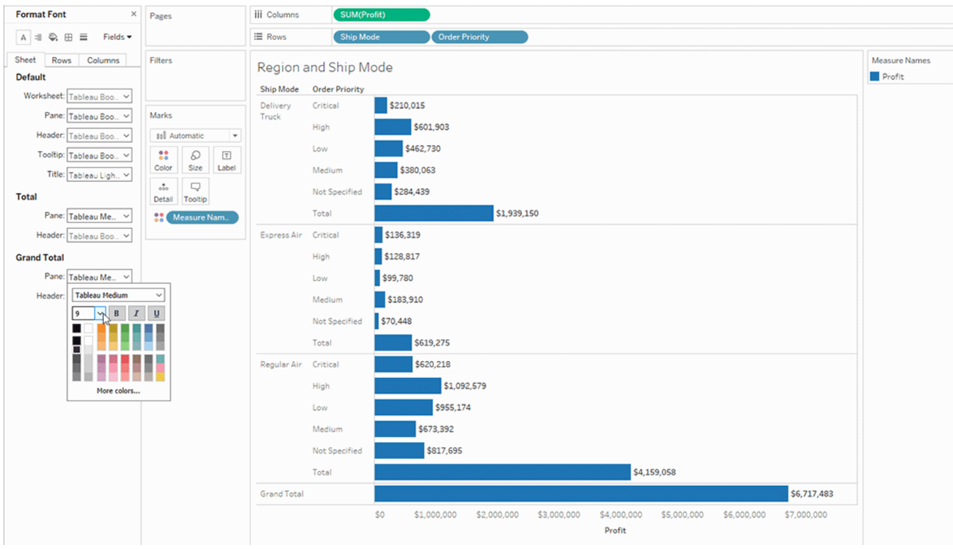
Adjusting the font for totals at the header changes the appearance of the **Total** and **Grand Total** label in the view.



Back to top.

H. Grand Total Pane

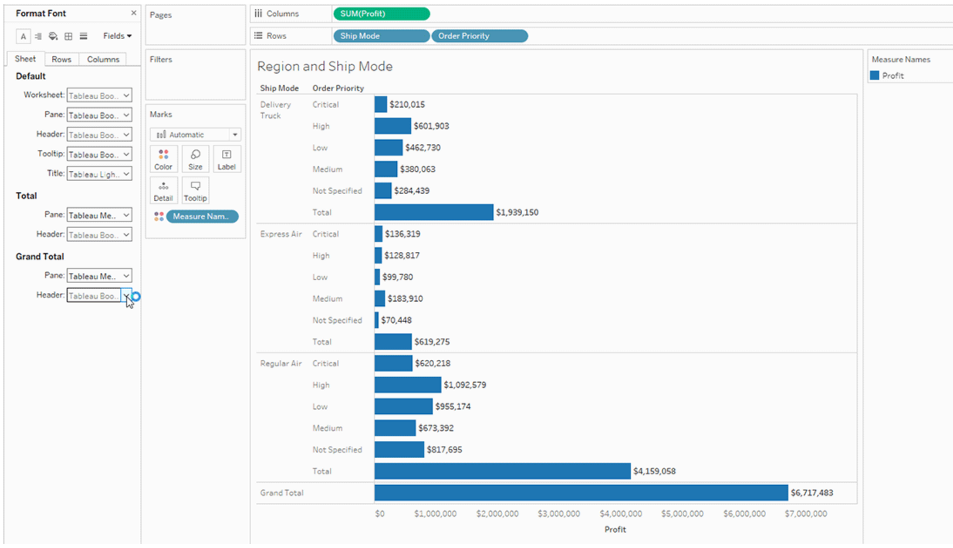
Select this option to adjust the font for the grand total result in your view.



Back to top.

I. Grand Total Header

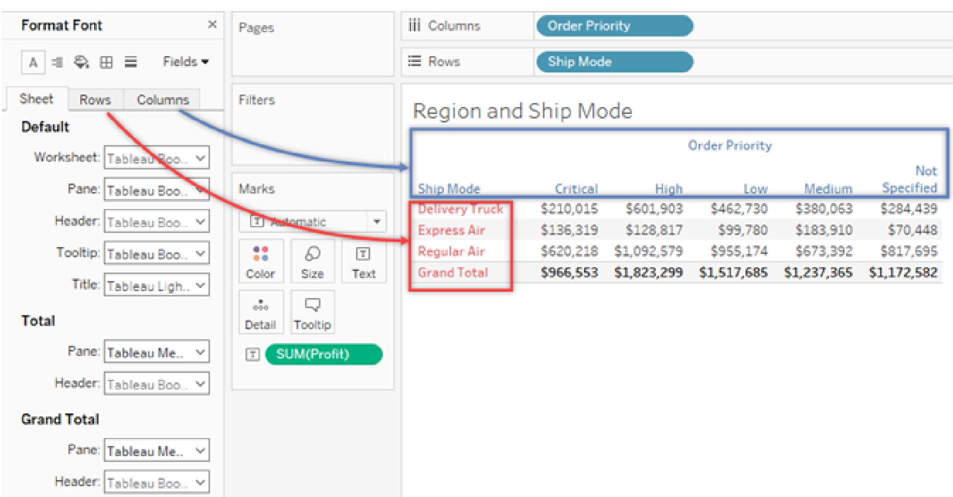
Select this option to adjust the font for the grand total label in your view.



Back to top.

J. Row and Column Text Formatting

If you have dimensions on both the row and column shelves, you can independently format the fonts through the **Rows** and **Columns** tabs on the Formatting pane.



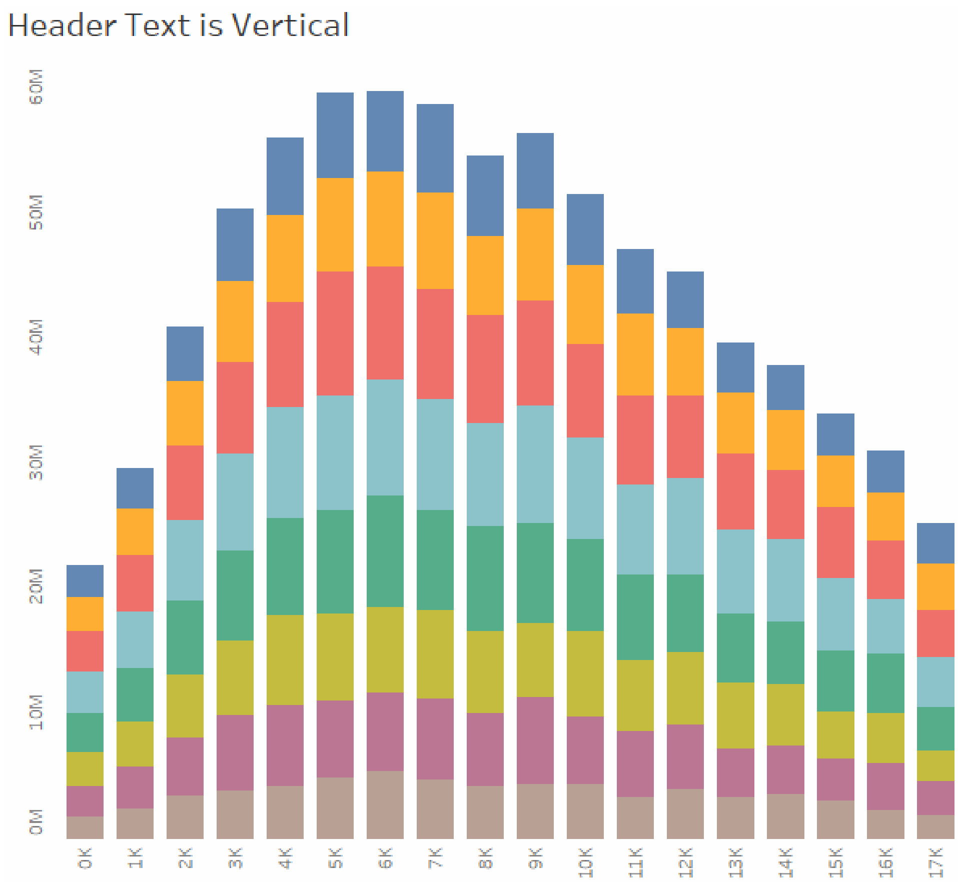
The format font options are limited to the Header, Total, and Grand Total selections.

Back to top.

Format text alignment

Tableau uses visual best practices to determine how text is aligned in a view, but you can also customize text. For example, you can change the direction of header text so that it is horizontal (default) instead of vertical (up).

To format the text alignment, select **Format** > **Alignment** to open the text alignment pane.



For each text area you can specify the following alignment options:

Horizontal - Controls whether text aligns to the left, right, or center.

Vertical Alignment - Controls whether text aligns at the top, middle, or bottom.

Direction - Rotates text so that it runs horizontally (default), top-to-bottom (up), or bottom-to-top (down).

Wrap - Controls whether long headers wrap to the next line or are abbreviated. It does not control text marks in the visualization.

> **Note:** If cells are not large enough to show more than one row of text, turning on wrapping will have no visible effect. If this happens, you can hover the cursor over a cell until a double-sided arrow appears, and then click and drag down to expand the cell size.

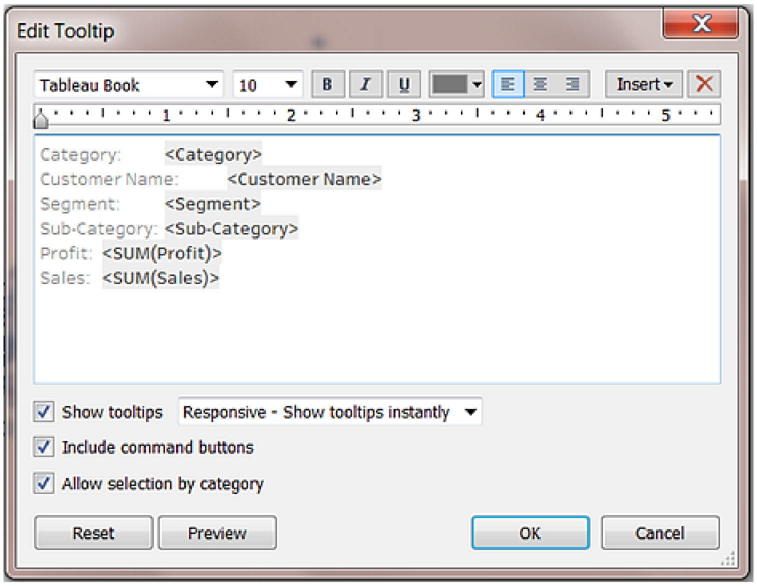
> **Note**: Tableau adheres to regional standards when determining when to begin or end line breaks.

Format tooltips

Tooltips are details that appear when you rest the pointer over one or more marks in the view. Tooltips also offer convenient tools to quickly filter or remove a selection, select marks that have the same value or view underlying data. You can edit the tooltip to include both static and dynamic text. You can also modify which fields are included in the automatic tooltip.

To edit a tooltip:

1. Select **Worksheet** > **Tooltip**. Tooltips are specified for each sheet and can be formatted using the formatting tools along the top of the Edit Tooltip dialog box. Use the **Insert** menu at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.



2. Use the formatting tools along the top of the Edit Tooltip dialog box.

3. (Optional) Use the **Insert** drop-down list at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.

   The **All Fields** option on the **Insert** menu adds all field names and values that are used in the view. Inserting the All Fields parameter updates the tooltip as you change the view. You can also use the All Fields option to exclude fields.

4. (Optional) Select the **Include command buttons** option to show filtering and view data options in the tooltip. For example, including command buttons will add **Keep Only**, **Exclude**, and **View Data** buttons to the bottom of the tooltip. These command buttons are available both in Tableau Desktop and when the view is published to the web or viewed on a mobile device.
5. (Optional) Select the **Allow selection by category** check box to select marks in a view that have the same value by clicking on a discrete field in a tooltip. For more information see **Tooltips** on page 250.

Edit worksheet titles, captions, and legend titles

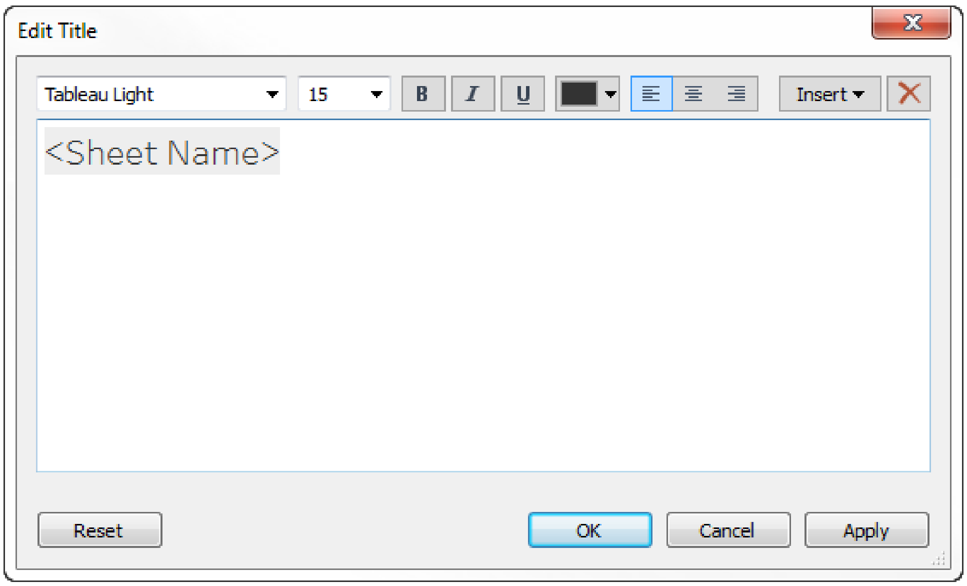
1. Do one of the following:

   - On Tableau Server or Tableau Online web editing, double click the item you want to change.

   - On Tableau Desktop, Right-click (control-click on Mac) the item you want to change and select **Edit <item>**, for example, **Edit Title**.

- On a worksheet, hover on the title, click the drop-down arrow on the right-hand side and select **Edit Title** or **Edit Caption** from the context menu.

2. In the Edit dialog box, modify the text and format the font, size, style, color, and alignment. Use the **Insert** menu to add dynamic text such as sheet properties and field values. Click **OK**.



To reset the title back to the default, click **Reset**.

Use Custom Fonts

Tableau provides a variety of fonts for you to use in your visualizations, including the fonts in its default typeface: Tableau. However you might want to use a font other than what's provided. For example, you may want to use a font that's unique to your company's brand.

To use a custom font in Tableau, the first step is to install it on the computers that are running Tableau. This includes Tableau Desktop and, if you're publishing to Tableau Server, any computers running Tableau Server.

If a font specified in Tableau Desktop isn't also installed on all nodes of Tableau Server, Tableau Server substitutes a default font.

To ensure that Tableau Server can render the font correctly, make sure the font is on the list of fonts installed with most browsers. These commonly installed fonts are called "web-safe" fonts, as detailed here.

> **Note:** If you download a workbook from Tableau Server, custom fonts are not downloaded with the workbook.

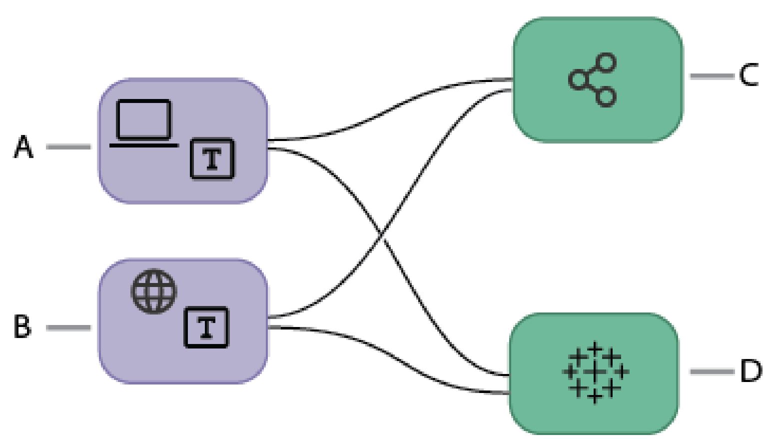
For more information about installing fonts, see the following links:

- How to install or remove a font in Windows (Microsoft Knowledge Base)
- Mac Basics: Font Book (Apple Support)

After you install the font it appears in the formatting pane in Tableau Desktop, where you can use it to change fonts at the workbook or worksheet levels. See **Format at the Workbook Level** on page 2882, **Format at the Worksheet Level** on page 2892, and **Format Text and Numbers** on page 2923 for details.

Publish custom fonts

Publishing a workbook with a custom font correctly depends on knowing how the author and published workbook interact.



Publishing a custom font work flow

A. Workbook is created in Tableau Desktop.

or

B. Workbook is created using Web Authoring.

then

C. Workbook is published to Tableau

Server.

or

D. Workbook is published to Tableau Online.

If	Then
A- The workbook is created in Tableau Desktop.	You can use any font installed on your computer in the workbook.
B- The workbook is created using Web Authoring.	Only fonts installed on the Tableau Server or Tableau Online are functional when formatting your text.
C- The workbook is published to Tableau Server.	Any custom font must be installed on all server nodes. If the font is not installed on Tableau Server, the uninstall font will be replaced with a substitute font upon publishing.
D- The workbook is published to Tableau Online.	Only fonts supported by Tableau Online will appear. Any unsupported fonts in the workbook will be replaced with a substitute font upon publishing. For a list of supported fonts, see Knowledge Base: Fonts not displaying as expected.

Troubleshoot custom fonts in published workbooks

Once published, any viewer of the published workbook will need to have the custom fonts installed on their computer. This is why it is a best practice to limit fonts to either Tableau fonts or web safe fonts.

Issue	Possible Solution
Some fonts are accurate in the view, but others are not.	When a view has many marks, the server will render the marks on the server side and display the marks as a static image. This can result in some fonts in the view appearing inconsistent with the rest of the view. To resolve this issue, reduce the number of marks per view.
The spacing on the font appears different for different viewers.	If the viewer has the custom font installed on their computer, the viewer will see the font displayed accurately. However, each browser uses a slightly different method for rendering fonts, so seeing slight differences in fonts between browsers is expected.
A different font appears for viewers.	If the viewer does not have the custom font installed on their computer, any uninstalled fonts in the workbook will be replaced with a substitute font selected by the browser.

Format Titles, Captions, Tooltips, and Legends

There are several ways text can appear on a sheet. You can show titles and captions on any sheet. Another option is to **Add tooltips to marks** on page 1366. A legend card appears in the worksheet when you encode marks by dropping them on the **Color** or **Size** cards. It's also possible to **Add Annotations** on page 1465 for a mark, point, or area in the view.

These text elements can be customized for text properties like size, color, alignment, and font, as well as element properties like shading and borders.

Note: Some of these formatting options are only available in Tableau Desktop.

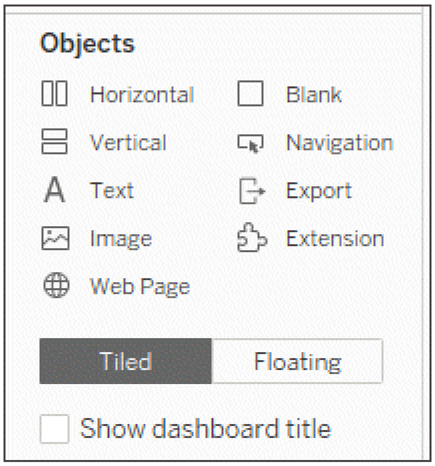
Some parts of the view have their own formatting and editing controls, such as an axis, and the text like labels and headers.

Show titles and other text elements

Titles and captions

If a title or caption isn't showing, open the Worksheet menu and select **Show Title** or **Show Caption**.

For dashboards and stories, there are equivalent options in their respective menus (such as Dashboard> Show Title). There is also a check box to toggle the dashboard or story title on or off, available in the left-hand pane.



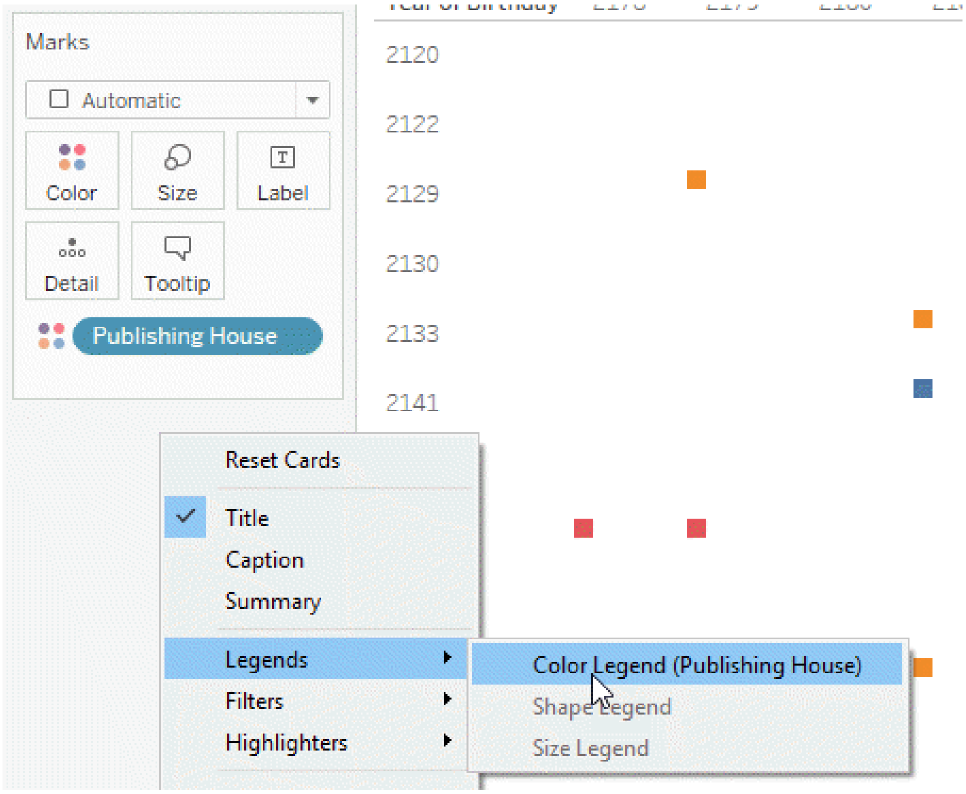
Legends and other elements

By default, a legend card is created whenever a field is encoded by color, size, or shape. If a legend is removed or hidden, however, it can be brought back.

Open the **Analysis** menu > **Legends** and select the desired option. Only legends that are relevant to the view can be selected. That is, if nothing is on the Size shelf, the size legend cannot be turned on.

On Worksheets

These items can also be toggled by right-clicking anywhere in the grey area of the sheet, such as under the Marks card.

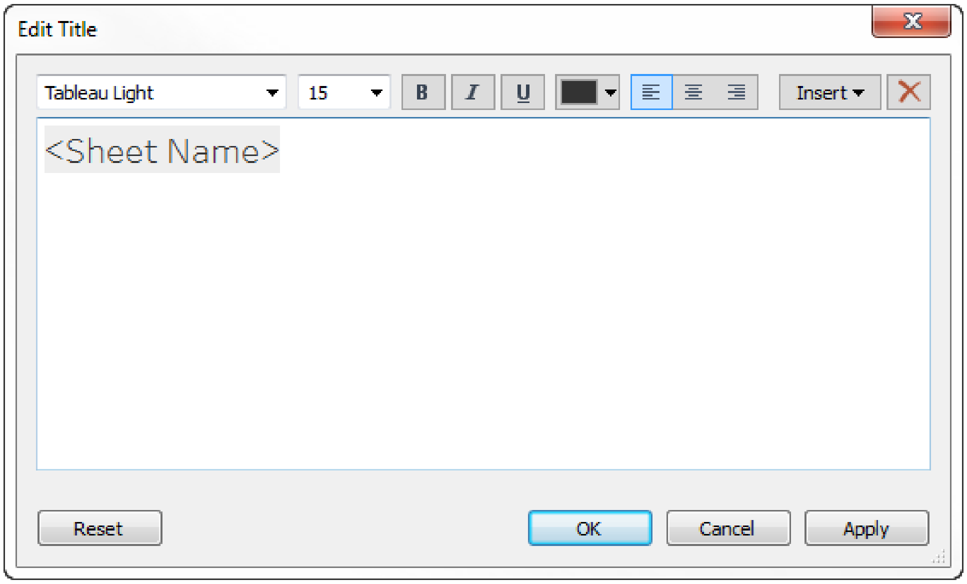


For more information about adding elements such as legends, filters, and parameter controls to a worksheet, see **Additional Shelves, Legends, Cards, and Controls** on page 239.

Edit titles and captions

1. On a worksheet, hover on the title, click the drop-down arrow on the right-hand side and select **Edit Title** or **Edit Caption** from the context menu. Alternatively:
    a. Right-click (control-click on Mac) the item you want to change and select **Edit**.
    b. In web editing, double click the worksheet title. Note that captions and legend titles cannot be edited in web editing
2. In the Edit Title dialog box, modify the text and format the font, size, style, color, and alignment. If desired, use the **Insert** menu to add dynamic text such as sheet properties and

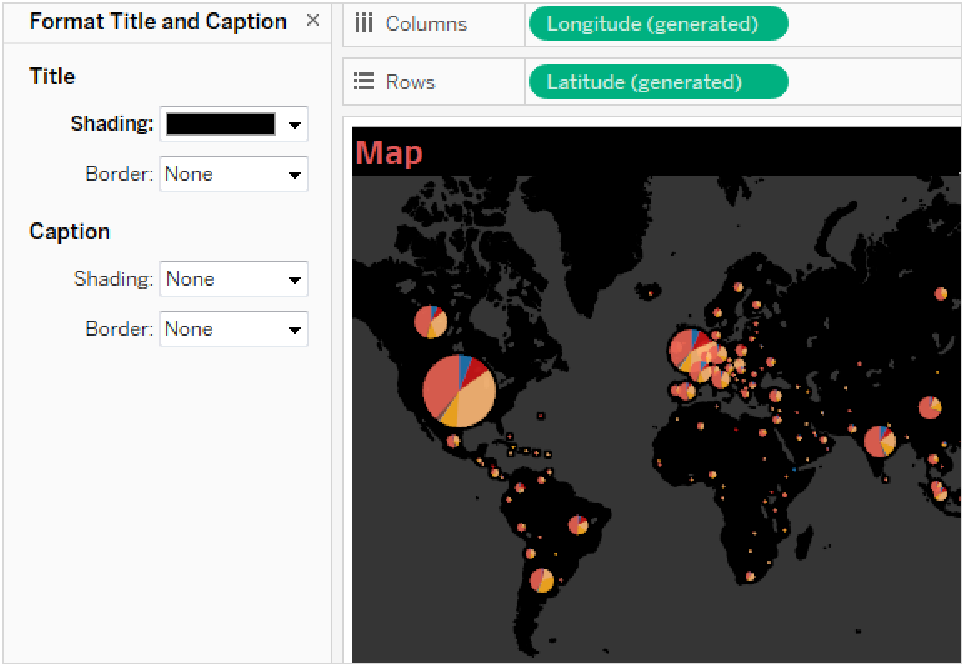
field values. To reset a title or caption back to the default, click **Reset**.



Format titles and captions

For information on showing or hiding a title, see **Titles** on page 252.

1. On a worksheet, right-click (control-click on Mac) the title, caption, or legend and select **Format <item>**—for example, **Format Title**.
2. In the **Format** pane, use the drop-down lists to change the default shading or border.



Edit dashboard titles

For information on showing or hiding a title, see **Titles** on page 252.

1. On a dashboard, double-click the title.
2. In the Edit Title dialog box that appears, format the title as you like. You can enter a new title for the dashboard or a description, as well as change the font type, size, emphasis, color, and alignment. You can also type in a website URL and Tableau will automatically create a hyperlink.

   > **Note**: If you edit the hyperlink, make sure to delete the entire hyperlink and then reenter the new one. If you do not do this, the text of the hyperlink will update, but the URL path will not.

3. When finished, click **OK**. The dashboard title updates with the changes.

Format tooltips

Tooltips are details that appear when you rest the pointer over one or more marks in the view. Tooltips also offer convenient tools to quickly filter or remove a selection, select marks that have the same value or view underlying data.

- For more information on tooltips, see **Tooltips** on page 250.
- To show a visualization in a tooltip, see **Create Views in Tooltips (Viz in Tooltip)** on page 1475.
- To see related concepts demonstrated in Tableau, watch Basic Tooltips, a 7-minute free training video. Use your tableau.com account to sign in.

You can edit the tooltip to include both static and dynamic text. You can also modify which fields are included in the automatic tooltip.

1. On the Marks card select **Tooltip**. Tooltips are specified for each sheet and can be formatted using the formatting tools along the top of the Edit Tooltip dialog box.
2. Use the **Insert** menu at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.
3. Use the formatting tools along the top of the Edit Tooltip dialog box.

There are several optional customizations as well.

Dynamic text values

Use the **Insert** drop-down list at the top of the dialog box to add dynamic text such as field values, sheet properties, and more.

The **All Fields** option on the **Insert** menu adds all field names and values that are used in the view. Inserting the All Fields parameter updates the tooltip as you change the view. You can also use the All Fields option to exclude fields.

Command buttons

Select the **Include command buttons** option to show filtering and view data options in the tooltip.

For example, including command buttons will add **Keep Only**, **Exclude**, and **View Data** buttons to the tooltip. These command buttons are available both in Tableau Desktop and when the view is published to the web or viewed on a mobile device. Unchecking **Include command buttons** will also hide the aggregation summary if multiple marks are selected.

Selection by category

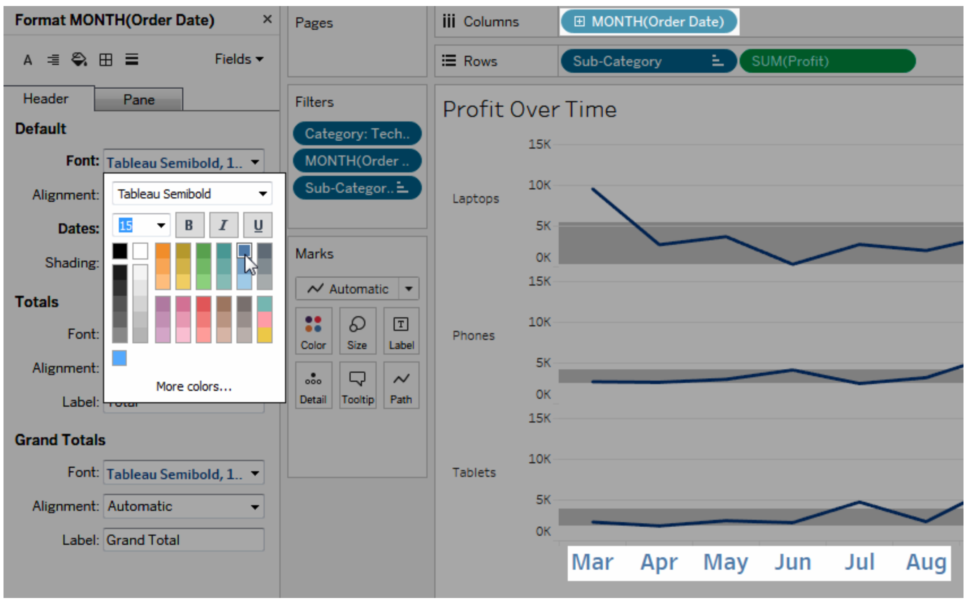
Select the **Allow selection by category** check box to select marks in a view that have the same value by clicking on a discrete field in a tooltip. For more information see **Tooltips** on page 250.

Format Fields and Field Labels

You can format a specific field in a view, as well as individual field labels. When formatting is done this way, your formatting changes are scoped to only where the field appears in the view, or only an individual label.

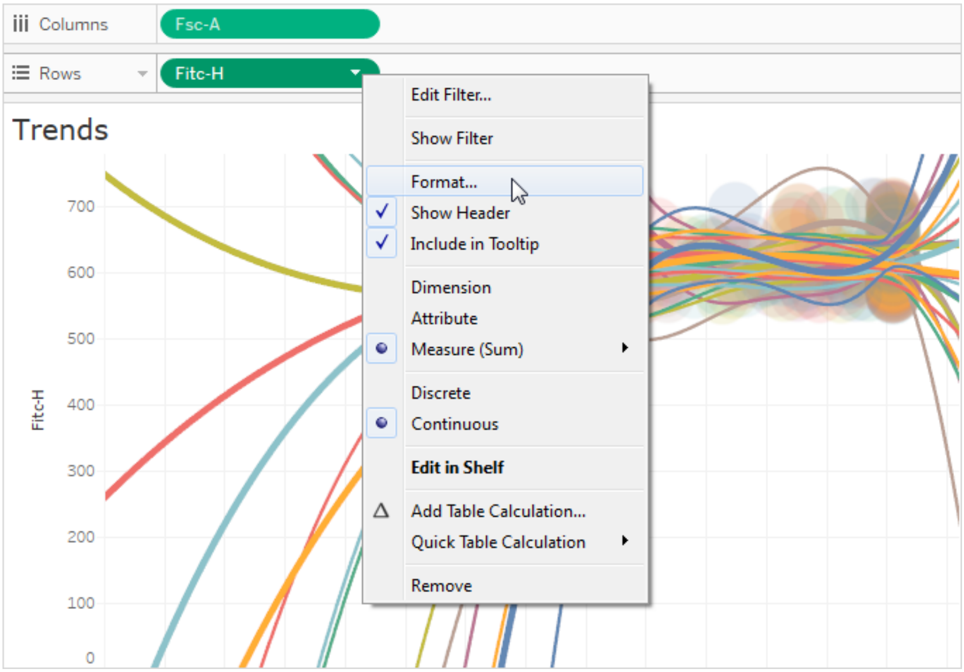
Format a field

In the view below, the **Month(Order Date)** field has been formatted so that the headers use the Tableau Semibold font, in blue. Notice that the header values along the Profit axis are not affected.



To format a specific field:

1. Right-click (control-click on Mac) the field and select **Format**.



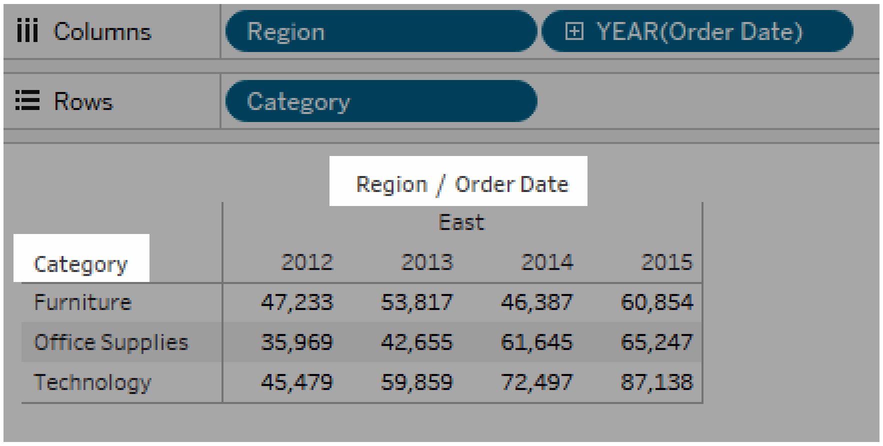
   The **Format** pane opens to settings for the selected field.

2. Make your changes in the **Format** pane.

   For discrete fields, such as Region or Customer Name, you can specify font and alignment properties for both header and pane areas. For continuous fields, such as Profit or Sales, you can format font properties for the pane and axis as well as number and tick mark colors. For more information about other axis options, see **Edit Axes** on page 2900. The view is updated as you make changes so you can quickly see the colors and formats that work with your view.

Format a field label

Field labels are row and column headings that indicate the data fields used to create a view. By default, Tableau shows field labels, but you can choose to hide them. When field labels are showing they appear in three different parts of the view: rows, columns, and the corner. The view below shows an example of each of these types of field labels.



You can format the font, shading, alignment, and separators for each of these types of field labels.

To format a specific field label:

1. Right-click (control-click on Mac) the field label in the view and select **Format**.
2. In the **Format** pane, specify the settings of the font, shading, and alignment field labels.

> **Note:** When you have multiple dimensions on the rows or columns shelves, the field labels appear adjacent to each other in the table. Each field label is separated from the others with a forward slash symbol. Specify a different separator in the **Format** pane.

Format Numbers and Null Values

When you format a measure, you can specify the number format for both the axis and the pane text. You can select from a set of standard formats, such as number, currency, scientific, and percentage; or you can define a custom number format using Microsoft Excel style format codes.

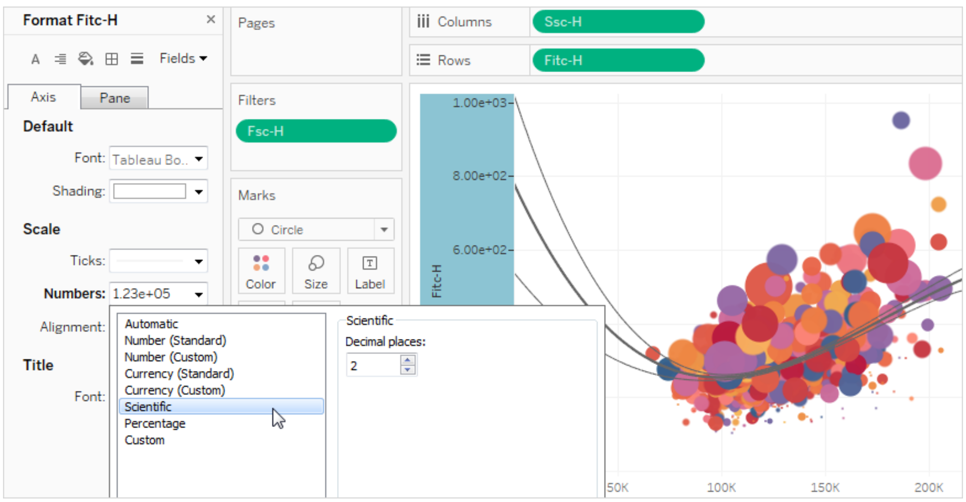
When a measure contains null values, the nulls are usually plotted as zero. You can use formatting, however, to handle the null values in a different way, such as hiding them.

For Tableau Desktop

Specify a number format

1. Right-click (control-click on Mac) a measure or axis in the view and select **Format**.
2. In the **Format** pane, click the **Numbers** drop-down menu.
3. Select a number format.

   Some formats require additional settings. For example, if you select **Scientific**, you must also specify the number of decimal places.



Here are the number formats and associated options available in Tableau.

Number Format	Format Options
Automatic: format is automatically selected based on either the format specified by the data source or the	None.

Number Format	Format Options
data contained in the field.	
Number (Standard): format is based on locale selected.	Locale: number format changes based on the geographical location selected.
Number (Custom): format is customized to your choice.	Decimal Places: the number of decimal places to display. Negative Values: how negative values are displayed. Units: the number is displayed using the specified units. For example, if the number

Number Format	Format Options
	is 20,000 and the units are thousands, the number will be displayed as 20K. Prefix/Suffix: characters that precede and follow each displayed number. Include thousands separators: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Currency (Standard): format and currency	Locale: currency format based on the

Number Format	Format Options
symbol is based on locale selected.	geographical location selected.
Currency (Custom): format and currency symbol is customized to your choice.	Decimal Places: the number of decimal places to display. Negative Values: how negative values are displayed. Units: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number is displayed as 20K.

Number Format	Format Options
	Prefix/Suffix: characters that precede and follow each displayed number. Include thousands separators: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Scientific: numbers are displayed in scientific notation.	Decimal: the number of decimal places to display.
Percentage: numbers are displayed as a percentage with the percent symbol. The value of 1 is	Decimal: the number of decimal places to display.

Number Format	Format Options
interpreted as 100% and 0 as 0%	
Custom: format is based entirely on what is specified in the format options.	Custom: type in the format you want to use. This format can be specified by a non-conditional Excel style number code. See Number format codes in the Microsoft Knowledge Base for details.

Set the default number format for a field

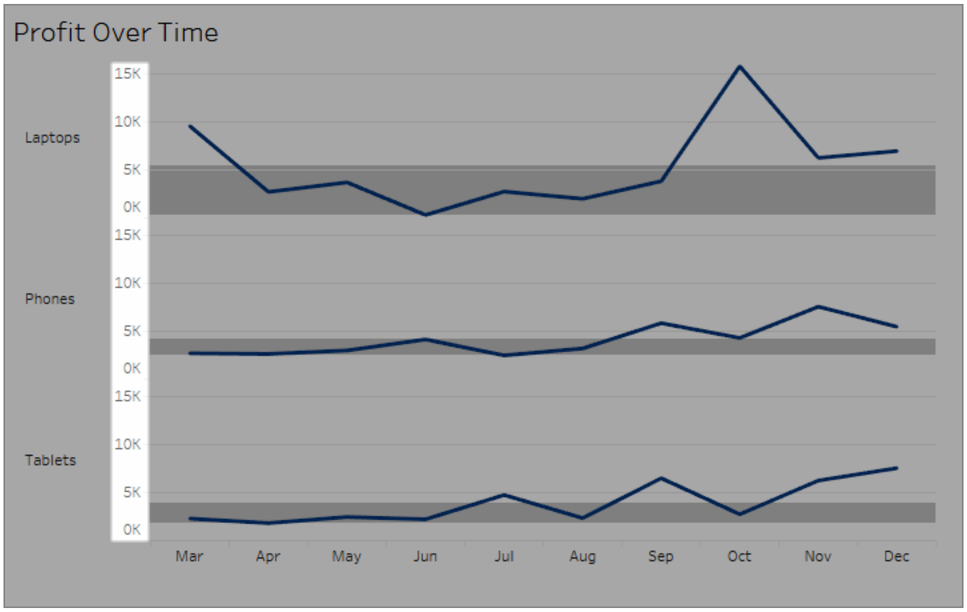
Right-click (control-click on Mac) the field in the **Data** pane and select **Default Properties** > **Number Format**.

In the subsequent dialog box, specify a number format to be used whenever the field is added to the view. The default number format is saved with the workbook. It is also exported when you export the connection information.

> **Note:**Formatting numbers using the **Format** pane overrides any number formatting applied elsewhere.

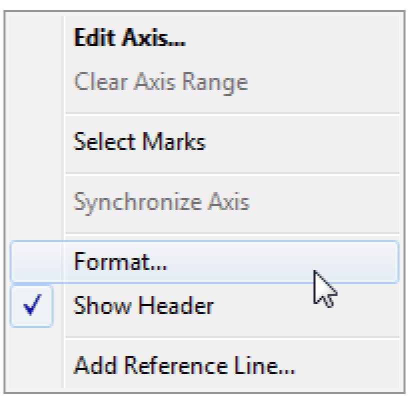
Format a measure as currency

The view in the following image shows profit over time. Notice that the profit figures on the vertical axis are not formatted as currency.



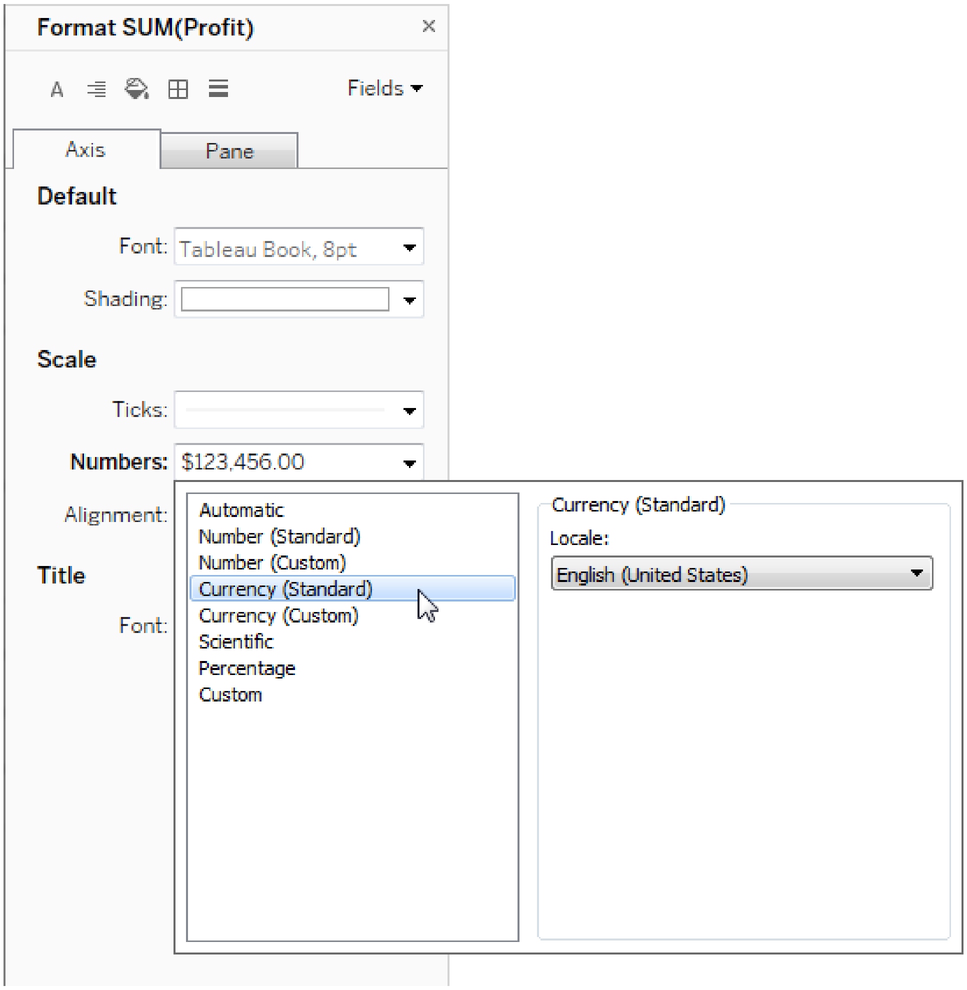
To format the numbers as currency:

1. Right-click the **Profit** axis and choose **Format**.



2. On the **Axis** tab in the **Format** pane, under **Scale**, select the **Numbers** drop-down list, and then select one of the following:

   Currency (Standard) to add a dollar sign and two decimal places to the figures.



Currency (Custom) to specify the number of decimal places, how to show negative values, the units, whether to include a prefix or suffix, and whether to include a separator character.

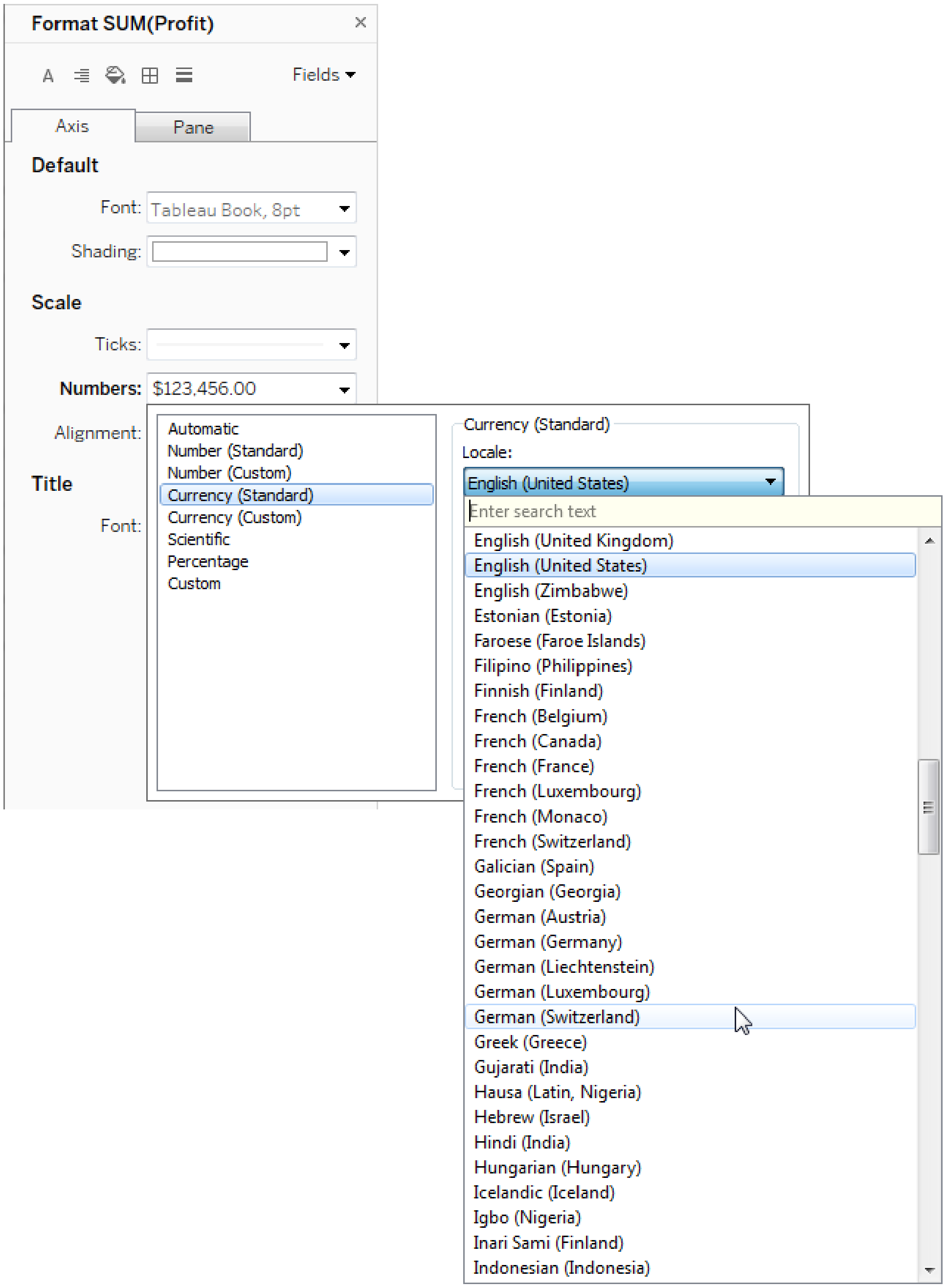
Use locale to specify number formats

By default, Tableau uses your computer's locale and language settings to format numbers. But you can explicitly set a different locale in the **Format** pane.

The following steps show how to set Swiss German currency, using the same view as in the previous section.

1. Right-click the **Profit** axis and select **Format**.
2. On the **Axis** tab, under **Scale**, select the **Numbers** drop-down list and then select **Currency (Standard)**.

3. In the **Locale** drop-down list, items appear in a **Language (Country)** format. For this example, select **German (Switzerland)**.



The view updates to show the sales figures in Swiss Francs, formatted for the German language.

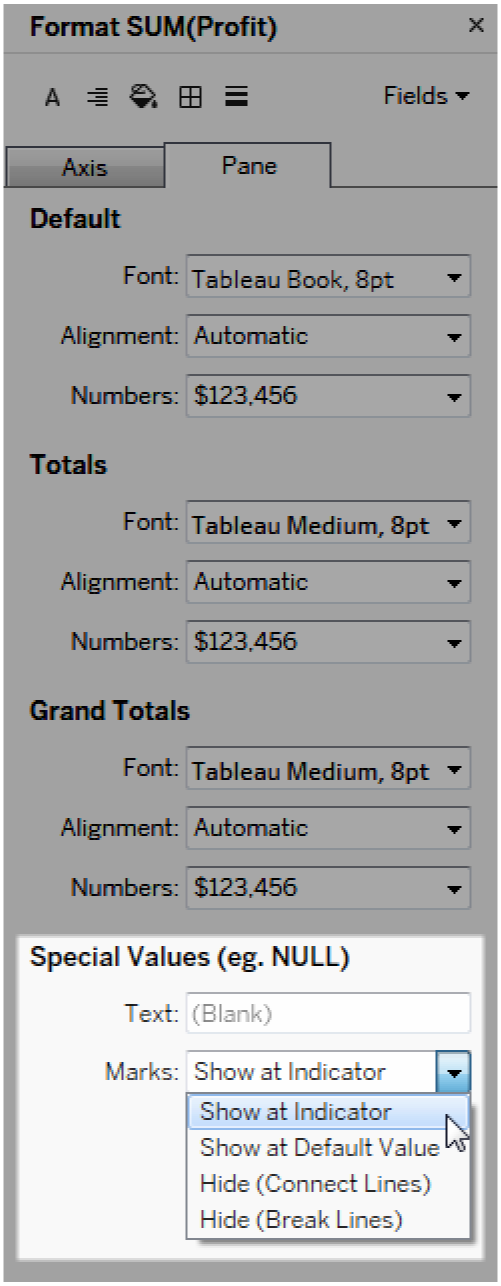
> **Tip:** You can change the default currency setting so that every time you drag the **Profit** measure to a view it uses the settings you want. In the **Data** pane, right-click **Profit** (or other monetary measure), and select **Default Properties** > **Number Format**. Then format the field as shown above.

Format null values

When a measure contains null values, they are usually plotted in a view as zero. However, sometimes that changes the view and you'd rather just suppress null values altogether. You can format each measure to handle null values in a unique way.

To format null values for a specific field:

1. Right-click the field in the view that has the null values (Control-click on a Mac) and choose **Format**.
2. Go to the **Pane** tab.
3. In the **Special Values** area, specify whether to show the null values using an indicator in the lower right corner of the view, plot them at a default value (e.g., zero for number fields), hide the values but connect lines, or hide and break lines to indicate that null values exist.



4. If you specify text in the **Text** field, it will appear in the view for null values when mark labels are turned on. See **Show, Hide, and Format Mark Labels** on page 1378.

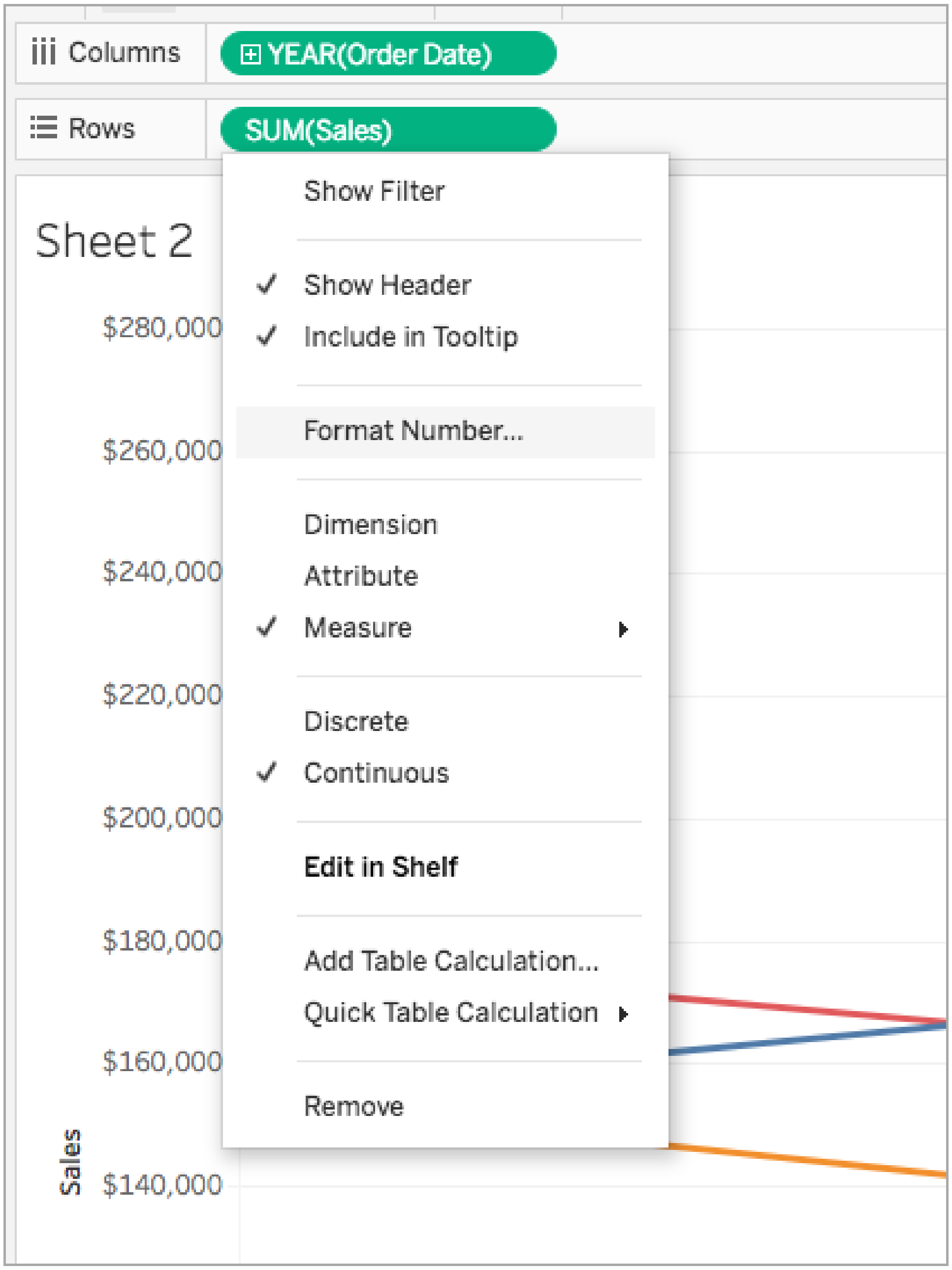
Note: The Special Values area is not available for dimensions or discrete measures.

For Tableau Server or Tableau Online

Specify a number format

When authoring a view on the web, you can specify the number format for a field used in the view.

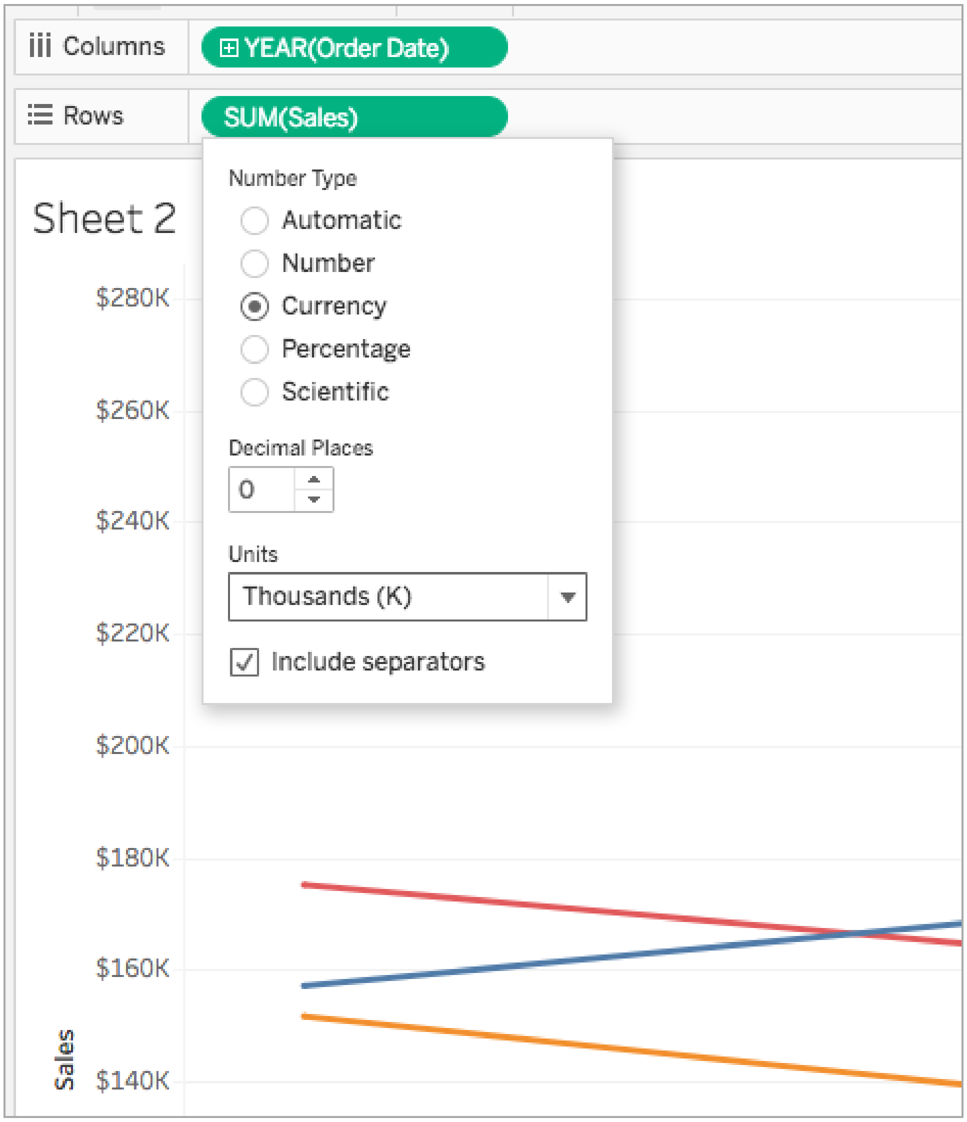
1. In web editing mode, right-click a measure in the view and select **Format Number**.



2. In the dialog box that appears, select a number format.

   Some formats provide additional settings. For example, if you select **Currency**, you can also specify the number of decimal places, as well as the units, and whether or not to include separators, such as commas.

In this example, Sales is formatted as a Currency with zero decimal places and thousand (k) units. Sales numbers in the view update with these settings. Labels and tooltips update as well.



Here are the number formats and associated options available in Tableau.

Number Format	Format Options
Automatic:	None.

Number Format	Format Options
format is automatically selected based on either the format specified by the data source or the data contained in the field.	
Number (Custom): format is customized to your choice.	**Decimal Places**: the number of decimal places to display. **Units**: the number is displayed using the specified units. For example, if the number is 20,000 and the units are thousands, the number

Number Format	Format Options
	will be displayed as 20K. **Include separators**: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Currency (Custom): format and currency symbol is customized to your choice.	**Decimal Places**: the number of decimal places to display. **Units**: the number is displayed using the specified units. For example, if the number is 20,000

Number Format	Format Options
	and the units are thousands, the number is displayed as 20K. **Include separators**: whether the number shows separators every thousand (example: 100,000 vs. 100000).
Percentage (Custom): numbers are displayed as a percentage with the percent symbol. The value of 1 is interpreted as 100% and 0 as 0%	**Decimal Places**: the number of decimal places to display.
Scientific	**Decimal**

Number Format	Format Options
(Custom): numbers are displayed in scientific notation.	**Places**: the number of decimal places to display.

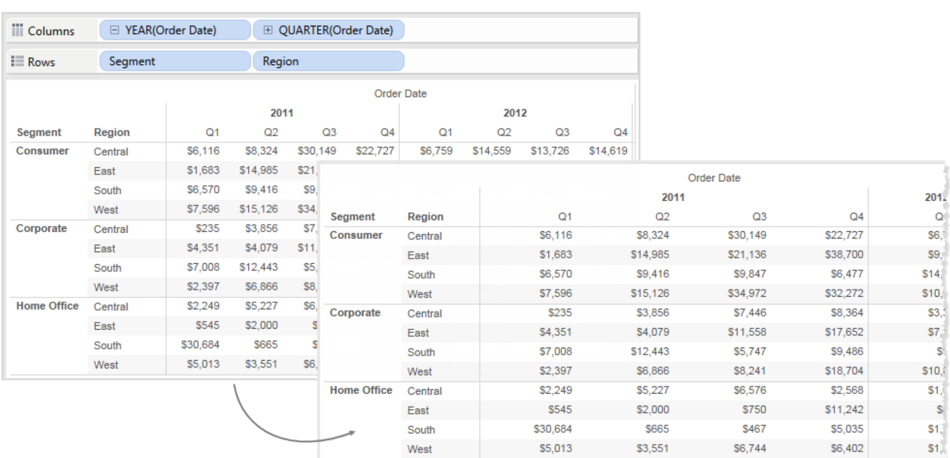
Resize Tables and Cells

You can change the size of the rows, columns, and cells that compose a table. The best way to resize your table depends on the view type and the table components you want to resize.

Use commands to resize rows and columns

1. Select **Format** > **Cell Size**.
2. Choose the **Taller**, **Shorter**, **Wider**, or **Narrower** command. To quickly apply these commands, see **Shortcuts for resizing rows and columns (Tableau Desktop)** on page 3350.

For example, in the view below, we used the **Wider** and **Taller** commands to make the view more readable.

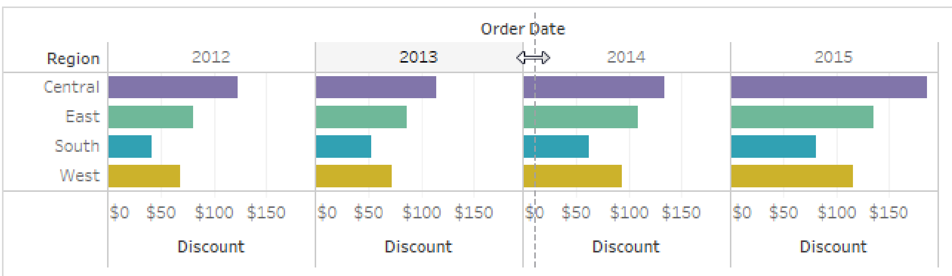


Note: For a given field, all members will have the same width and the same height. You can't resize individual field members.

Manually resize rows and columns

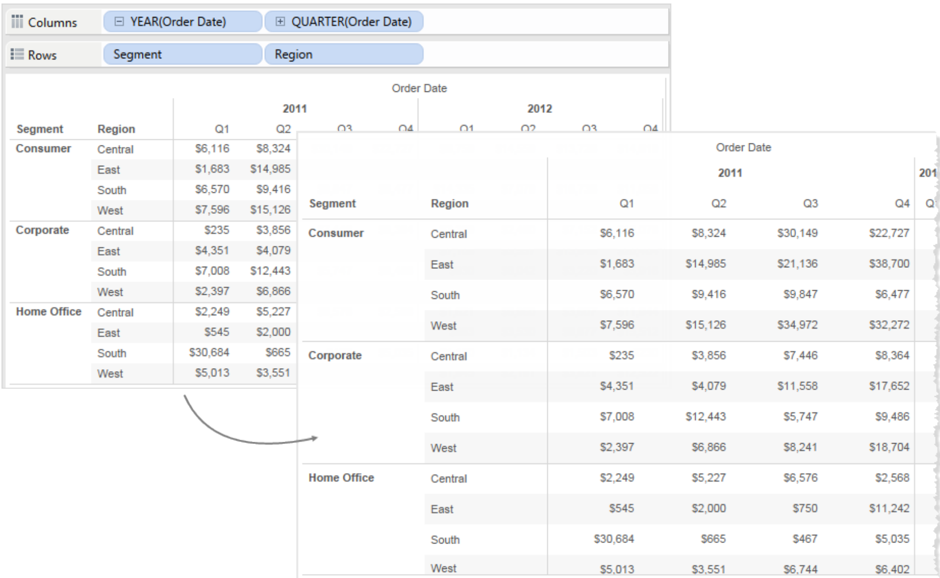
To manually resize the widths or heights of row and column headers or axes:

1. Place your cursor over the vertical or horizontal border of a header or axis.
2. When you see the resize cursor , click and drag the border left and right or up and down.



Resize an entire table

You can increase or decrease the size of the entire table by selecting **Bigger** or **Smaller** on the **Format** > **Cell Size** menu. For example, to increase the width of the columns and the height of the rows for the view shown below, you can select **Format** > **Cell Size** > **Bigger** . This option increases both the width and height of the panes in a visually appropriate way. Notice that the size of the row headers increases horizontally when you resize the table.



Resize cells

Any table you can create in Tableau has the cell as its basic component. For a text table, the cell is what you would expect. It is the intersection of a row and a column, and is where the text is displayed.

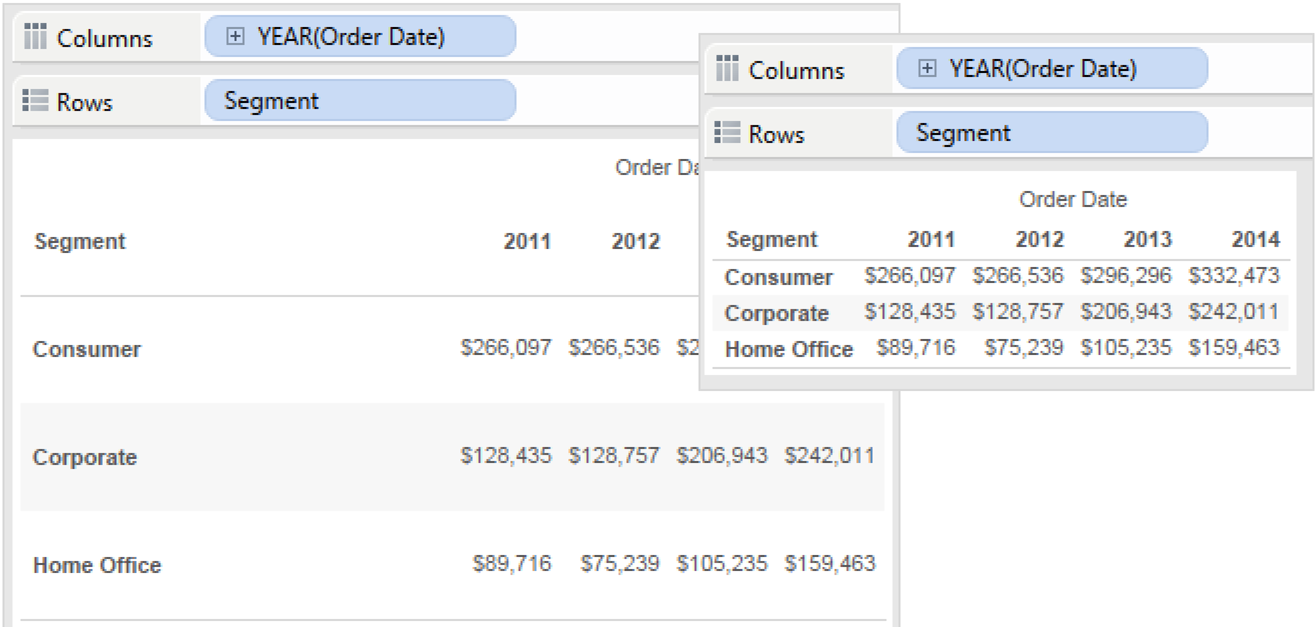
However, depending on the view you construct, identifying the cell is not always possible or useful, such as in the case of a scatter plot.

Manipulating cells to enhance your data view is useful when dimensions are the inner fields on both the **Rows** and **Columns** shelves. In this case, there are two shortcuts you can select on the **Format** > **Cell Size** menu:

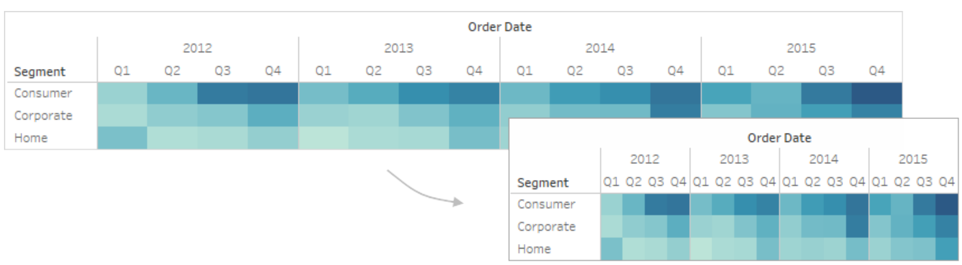
- **Square Cell** – Adjusts the view so the cell has a 1:1 aspect ratio. This results in a square cell, which is useful for heat maps.
- **Text Cell** – Adjusts the view so the cell has a 3:1 aspect ratio. This is useful for text tables.

For example, you can see the text table below with **Square Cell** and **Text Cell** applied. **Text Cell** enforces a cell aspect ratio of 3:1 and results in a compact table that is easy to read.

Square Cell **Text Cell**



The heat map shown below is modified by selecting **Square Cell** on the **Format** > **Cell Size** menu. This enforces a cell aspect ratio of 1:1 and results in a compact table that is easy to analyze. You can also use the **Size** slider on the **Marks** card to adjust the size of each mark.



After changing the cell size, you can use **Ctrl+B** and **Ctrl+Shift+B** to decrease or increase the table size while maintaining the cell aspect ratio. On a Mac, the keystrokes are **Command-B** and **Shift-Command-B**.

Define Table Structure

In addition to the standard formatting, there are some other settings that define the table structure. You can modify these settings by selecting **Analysis** > **Table Layout** > **Advanced** to open the Table Options dialog box. There you can specify the aspect ratio, the default number format, row and column attributes, and the default label orientation for labels along the bottom of the view. While these settings apply to the whole view, some can be overridden using the **Format** pane.

Set the aspect ratio

The aspect ratio refers to the ratio of the pane width to the pane height. You can choose to constrain the aspect ratio to a specified amount or not constrain it at all. An unconstrained axis range can be useful because it means that the axes don't have to be the same length.

Anytime you manually resize a row or column, you are unconstraining the aspect ratio. The aspect ratio setting only applies to views containing continuous axes on both the row and column shelves. Nominative axes are not affected by the aspect ratio settings.

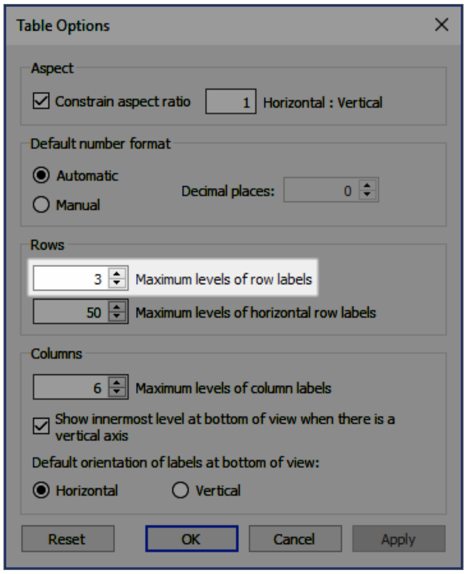
Set the default number format

You can define the number of decimal places to appear by default for numbers in the view. If you select **Automatic**, Tableau automatically decides the number of decimal places based on the data in the field. If you select **Manual,** you can decide to show up to 16 decimal places.

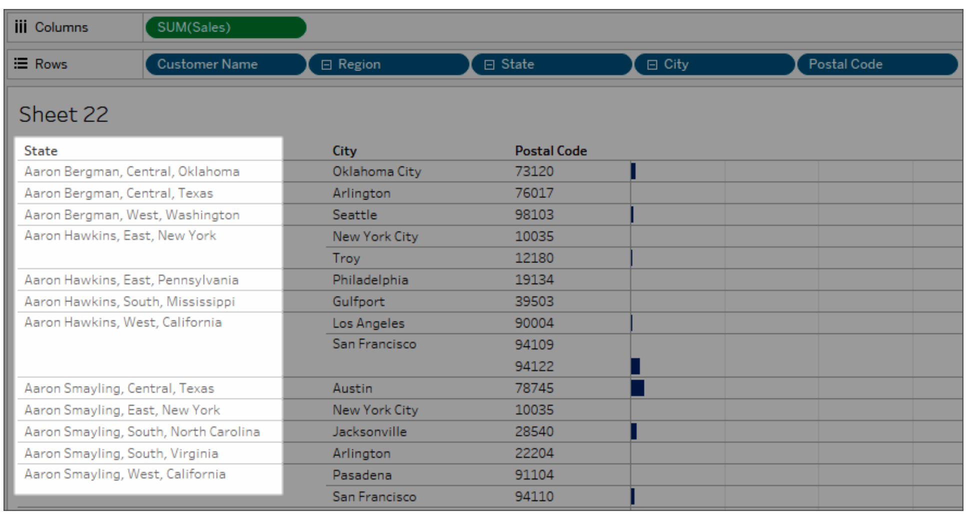
Set row attributes

Select from the following row attributes:

- Maximum levels of row labels: determines the number of fields that can be added to the Rows shelf before the headers are combined on the same level.



For example, if you set Maximum levels of row labels to 3 and place 5 discrete fields on the Rows shelf, then the first, second, and third row headers will appear in the same header, separated by commas (they will be combined at the same header level).



In this example, Customer Name, Region, State, City, and Postal Code are the 5 discrete fields on the Rows shelf. Maximum levels of row labels is set to 3. In the resulting view, Customer Name is combined with Region and State in the first column.

- Maximum levels of horizontal row labels: determines the number of fields that can be placed on the Rows shelf before headers are automatically oriented vertically rather than horizontally when the table doesn't fit the view.

Set column attributes

Select from the following column attributes:

- Maximum levels of column labels: determines the number of fields that can be placed on the Columns shelf before Tableau begins to combine the labels. For example, if you set Maximum levels of column labels to 8 and place 9 discrete fields on the Columns shelf, then the first and second column headers will appear in the same header, separated by a comma (they will be combined at the same header level).
- Show innermost level at bottom of view when there is a vertical axis: displays the innermost level of column headers at the bottom of the view (as opposed to the top) when a vertical axis is added to the view.
- Default orientation of labels at bottom of view: determines whether labels at the bottom of the view are oriented horizontally or vertically by default. You can toggle between the horizontal and vertical options by pressing Ctrl + L on your keyboard (Command-L on a Mac).

Create Custom Color Palettes

Tableau Desktop comes with color palettes that have been carefully designed to work well together and effectively apply color to data in many situations, such as on maps, heat maps, bar charts, etc. You can also create and use your own custom color palettes by modifying the **Preferences.tps** file that comes with Tableau Desktop. For example, you can create a custom categorical palette that matches your company's brand.

About the preferences file

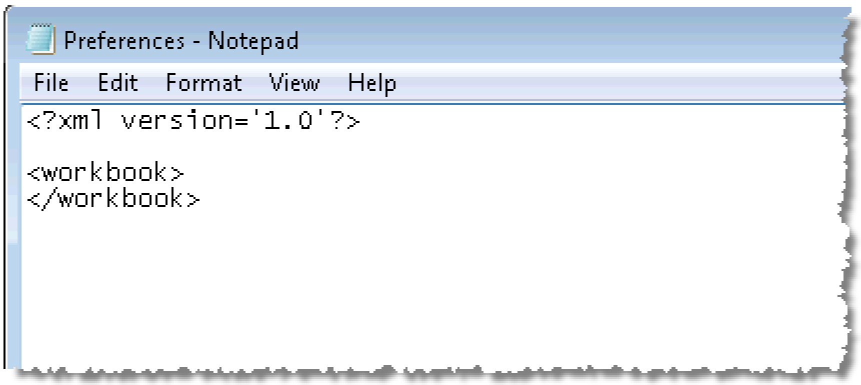
You can add as many custom palettes as you like to your **Preferences.tps** file, each with as many colors as you want. When you modify **Preferences.tps** to add colors, use the standard HTML format for the new colors (hexadecimal value #RRGGBB or Red Green Blue format). When you save the workbook and restart Tableau Desktop, the color palette names you added to **Preferences.tps** appear in the **Select Color Palette** drop-down list (Edit Color dialog). You can use a new palette like you would any other.

Tableau doesn't test or support custom color palettes, so be sure to back up your workbooks before you continue. Also, there is no guarantee that custom color palettes you create will work with future Tableau Desktop upgrades.

> When you edit your **Preferences.tps** file, be sure to use straight quotation marks (`'  '` or `"  "`) to delimit the palette name and type, not curly quotation marks (“ ” or ‘ ’).

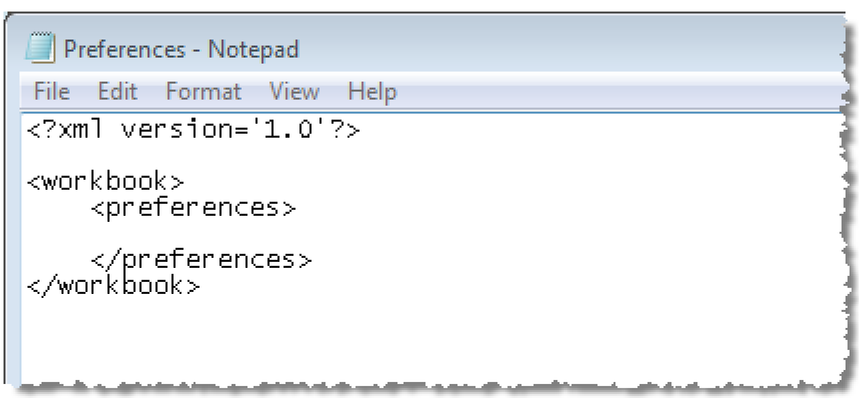
Edit the preferences file

The **Preferences.tps** file is located in your My Tableau Repository. The file is a basic XML file that you can open in a text editor to modify. An unedited preferences file looks like this:



To edit your preferences file:

1. Go to the My Tableau Repository folder in your Documents directory, and open the **Preferences.tps** file.
2. Between the opening and closing `workbook` tags, insert opening and closing `preferences` tags.

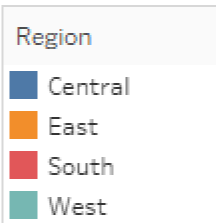


3. Follow one of the next three procedures to create a custom color palette.

Create custom color palettes

Create a custom categorical color palette

A categorical color palette contains several distinct colors that can be assigned to discrete dimension members. For example, when you put a discrete dimension such as **Region** on the **Color** card, the categorical color legend is used.



The following is an example of what to add between the `preferences` tags to add a categorical color palette. Note that the `type` attribute is specified as regular, which identifies this palette as a categorical palette.

To create a custom categorical color palette:

1. In the **Preferences.tps** file, between the "preferences" tags, paste the following. Be sure to use straight quotation marks, not curly quotation marks, to delimit the palette's name and type.

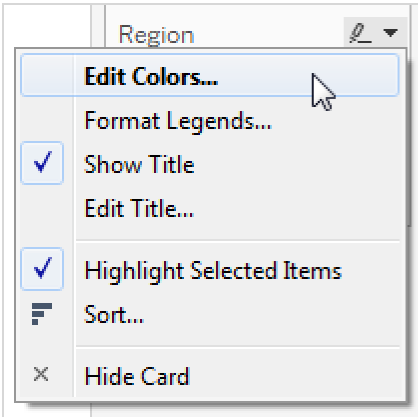
   ```
   <color-palette name="My Categorical Palette" type="regular" >

   <color>#eb912b</color>
   <color>#7099a5</color>
   <color>#c71f34</color>
   <color>#1d437d</color>
   <color>#e8762b</color>
   <color>#5b6591</color>
   <color>#59879b</color>
   </color-palette>
   ```

2. Save the **Preferences.tps** file and then restart Tableau Desktop.

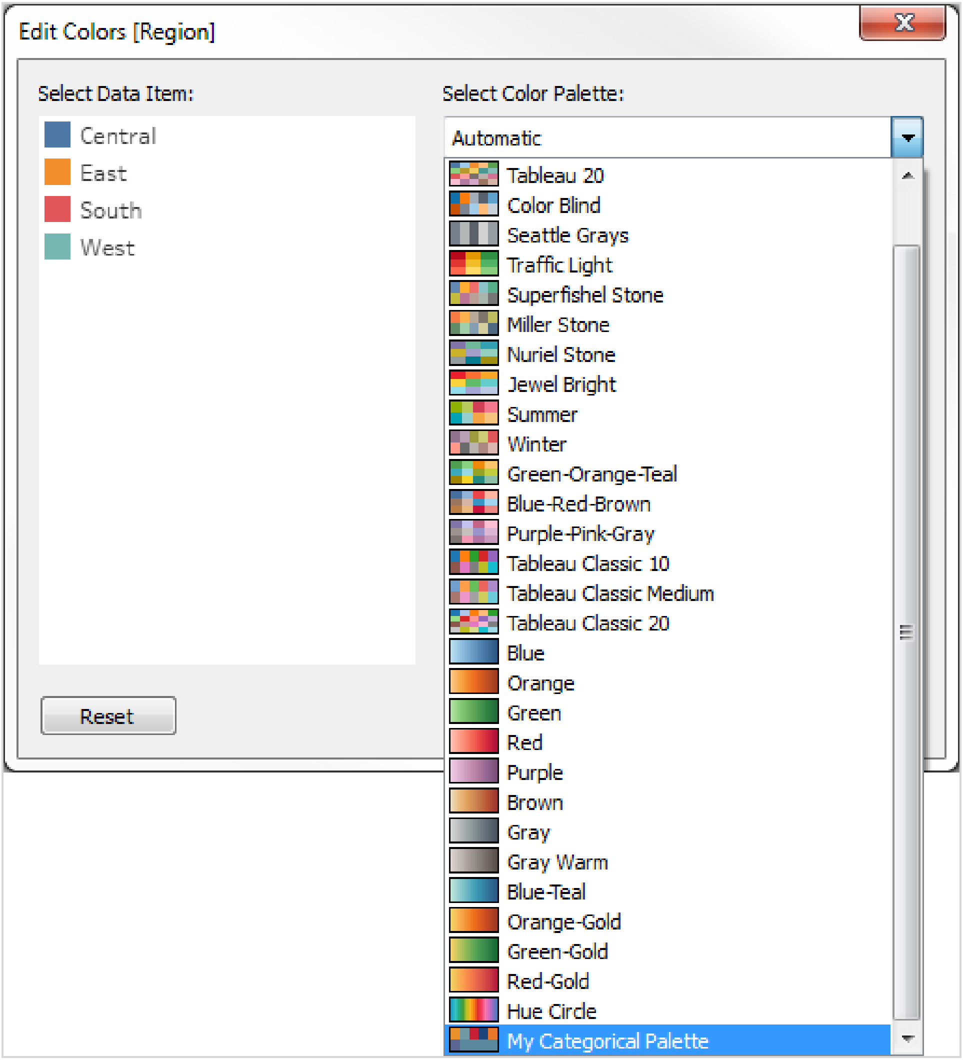
3. Open a data source, such as the **Superstore - Sample** data source.

4. From the **Data** pane, drag a discrete dimension, such as **Region**, to **Color**.

5. Click the color legend menu arrow and select **Edit Colors**.

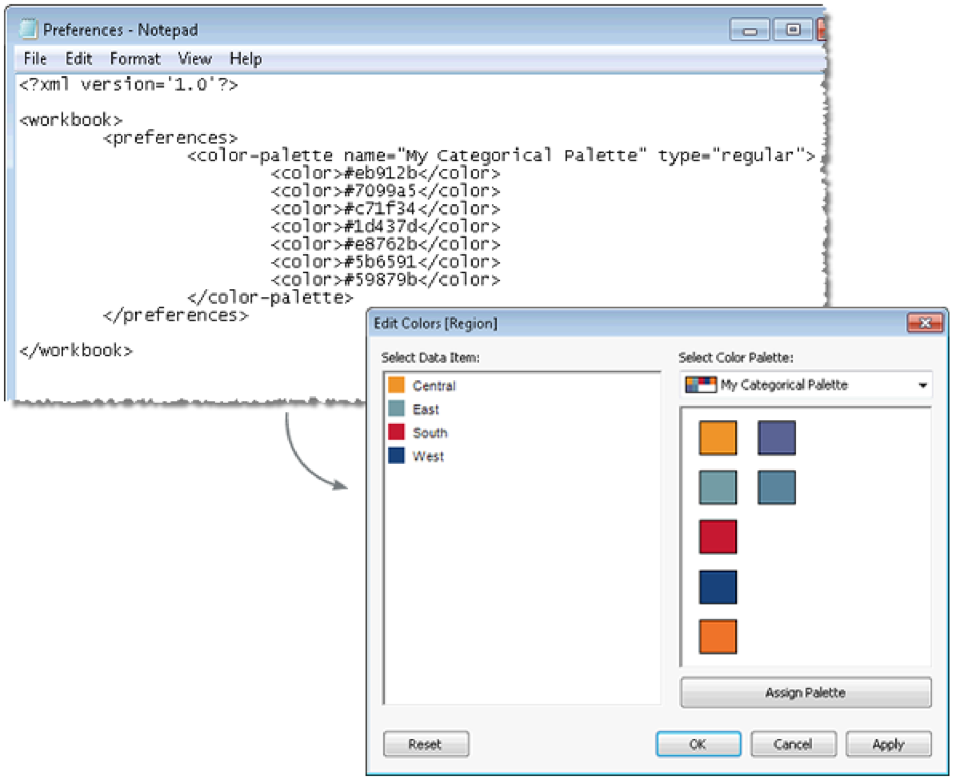


6. In the Edit Colors dialog box, from the palette drop-down list, select your new custom palette.



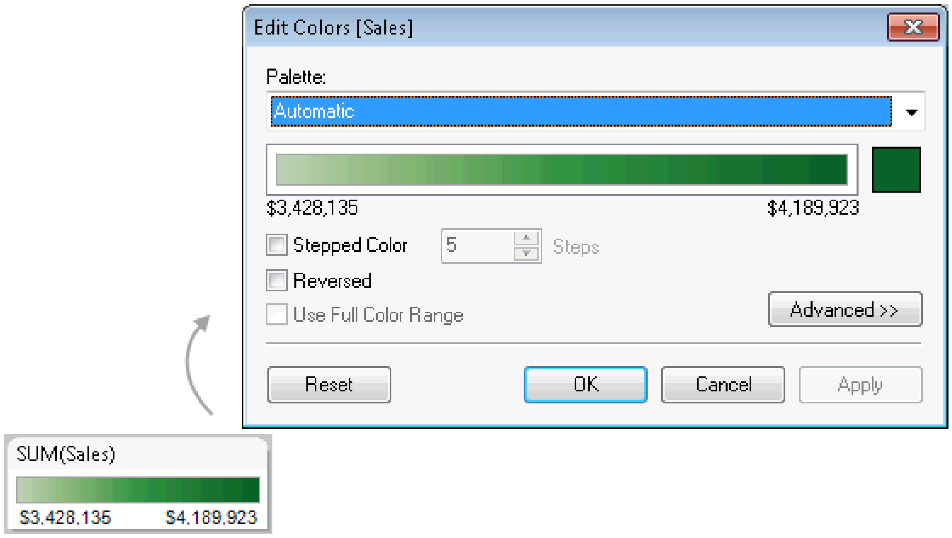
7. Click the **Assign Palette** button to assign the custom colors to each respective field.

8. Click **OK**.



Create a custom sequential color palette

Another type of palette is the sequential color palette. Typically, this type of palette shows a single color, varying in intensity. This type of color palette is used for continuous fields, typically for measures.



The following is an example of what to add between the `preferences` tags to add a sequential color palette. Note that the `type` attribute is specified as `ordered-sequential`, which identifies this palette as a sequential palette. Also, for sequential palettes you must specify at least two variants of the color in the sequential color range.

To create a custom sequential color palette:

1. In the **Preferences.tps** file, between the `preferences` tags, paste the following. Be sure to use straight quotation marks, not curly quotation marks, to delimit the palette's name and type.

```
<color-palette name="My Sequential Palette" type="ordered-sequential" >
 <color>#eb912b</color>
 <color>#eb9c42</color>
 <color>#ebad67</color>
 <color>#eabb86</color>
 <color>#eacba8</color>
 <color>#ebd8c2</color>
 </color-palette>
```

2. Save the **Preferences.tps** file and then restart Tableau Desktop.

3. Open a data source, such as the **Superstore- Sample** data source.

4. From the **Data** pane, drag a measure (such as **Sales**) to **Color**.

5. Click the color legend menu arrow, and select **Edit Colors**.

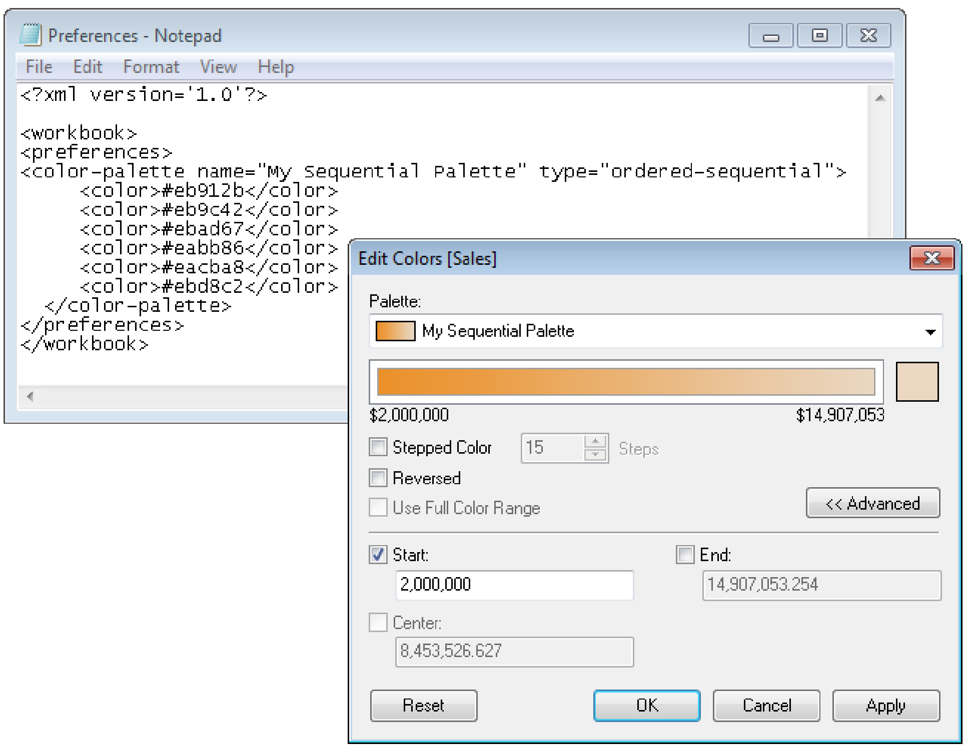
6. In the Edit Colors dialog box, from the palette drop-down list, select your custom palette.

7. If you want each color gradation to be defined within a box, select the **Stepped Color** check box, and in the **Steps** text box, type the number of color steps you want to display in the bar.

8. Click the **Advanced** button.

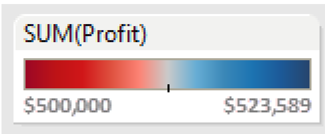
9. Select the **Start** check box, and in the text box, type the low end number you want for the continuum.

10. Click the **Apply** button to see the result, and make adjustments as needed. The color will range from high to low intensity (or the reverse) based on the order you specify in the **Preferences.tps** file. The default for sequential color palettes in Tableau is to make the high end of the continuum intense and the low end pale, though selecting the **Reversed** check box will do the opposite.



Create a custom diverging color palette

The third type of color palette is a diverging color palette. A diverging palette shows two ranges of values using color intensity to show the magnitude of the number and the actual color to show which range the number is from. Diverging palettes are most commonly used to show the difference between positive and negative numbers.



The following is an example of what to add between the `preferences` tags to add a diverging color palette. Note that the `type` attribute is specified as ordered-diverging, which identifies this palette as a diverging palette.

To create a custom diverging color palette:

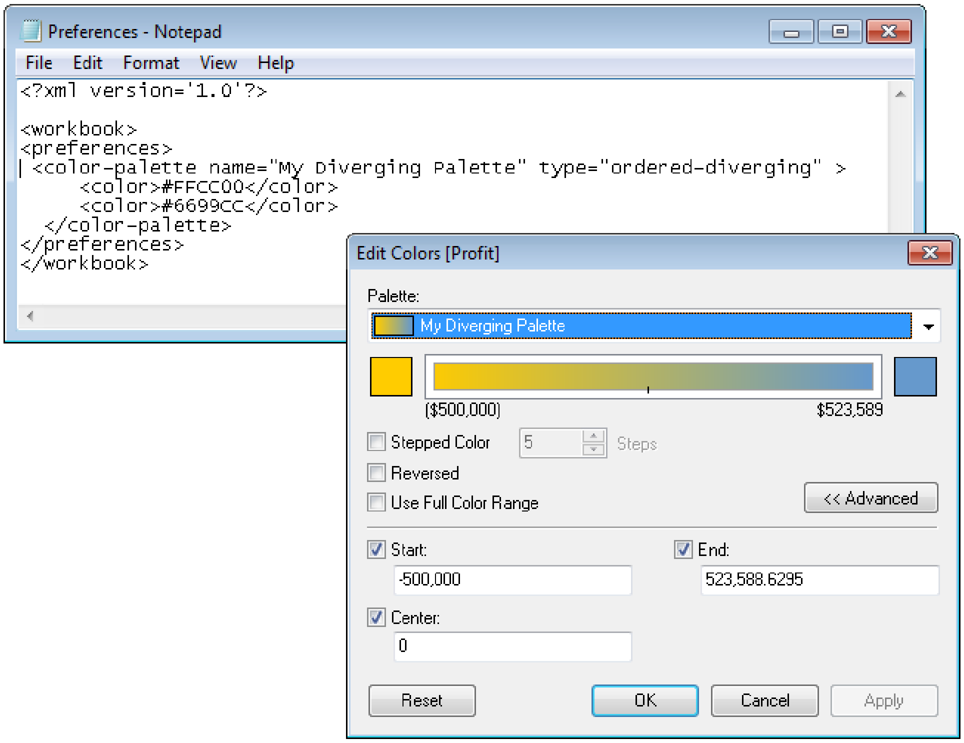
1. In the **Preferences.tps** file, between the `preferences` tags, paste the following. Be sure to use straight quotation marks, not curly quotation marks, to delimit the palette's name and type.

```
<color-palette name="My Diverging Palette" type="ordered-
diverging" >
 <color>#eb912b</color>
 <color>#59879b</color>
 </color-palette>
```

2. Save the **Preferences.tps** file and then restart Tableau Desktop.

3. Open a data source, such as the **Superstore - Sample** data source.

4. Click the **Assign Palette** button. The colors in the palette are used in the order they appear in the Preferences file.



If you add a sequential or diverging palette, remember to change the "type" attribute from "regular" to one of the following:

- ordered-sequential
- ordered-diverging

(Optional): Assign a default custom palette to dimensions and measures and publish as a data source

After you save the workbook, the custom color palette information is embedded in the workbook (for Excel and text file-based workbooks, in the .twbx) and therefore only available for that workbook. This means that colors that are in use are shown for anybody opening the that particular workbook. If they don't have the modified preferences file, they can't use the color information for any new color encoding.

To allow new color encoding using the custom color palette or to standardize a custom color palette for the Tableau workbooks in your organization, you can create the custom color palette using one of the options above, and then publish it as a Tableau Server data source.

1. On the same computer from which you modified the **Preferences.tps** file, open Tableau Desktop.
2. Open the **Superstore - Sample** data source.
3. Right-click a field in the **Data** pane, and select **Default Properties** > **Color**.
4. In the Edit Colors dialog box, associate the field values with the custom color palette, and then click **OK** when finished.
5. From the **Data** menu, select the data source, select **Publish to Server**, and then complete process to publish the data source.

After publishing the data source to Tableau Server, connect any new workbooks to this data source to use the custom color palette.

Note that after publishing the workbook with the custom color palette, that palette will not be visible or available in the **Edit Colors** dialog box. Authors will not be able to select custom palette for when using web editing in Tableau Server or Tableau Online.

Use discontinued (classic) color palettes

In version 10.0, Tableau created new color palettes, updated some existing ones (such as Tableau 10 and Tableau 20), and discontinued others. If you want to keep using a color palette that was discontinued, you can edit your **Preferences.tps** file to add the hex values for the palette. You can add as many color palettes as you like. See **Hex values for discontinued color palettes** below and **Restore a discontinued (classic) palette** on page 2991 for details.

Hex values for discontinued color palettes

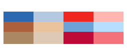
The table below lists discontinued color palettes, along with the XML code and hex values you can use in your **Preferences.tps** file to restore them. If you use the code as-is, palette names will start with **Classic** to indicate that they're from version 9.3 and earlier. In many cases, there are updated versions of color palettes that have been discontinued. See the "Details" column for specifics.

Version 9.x palette name	Details	Version 9.x HEX Values

Tableau 10 	Named **Tableau Classic 10** in version 10.0 and higher.	`<color-palette name="Classic 10" type = "regular"> <color>#17becf</color> <color>#bcbd22</color> <color>#7f7f7f</color> <color>#e377c2</color> <color>#8c564b</color> <color>#9467bd</color> <color>#d62728</color> <color>#2ca02c</color> <color>#ff7f0e</color> <color>#1f77b4</color> </color-palette>`
Tableau 10 Medium 	Named **Tableau Classic Medium** in version 10.0 and higher.	`<color-palette name="Classic 10 Medium" type = "regular"> <color>#6dccda</color> <color>#cdcc5d</color> <color>#a2a2a2</color> <color>#ed97ca</color> <color>#a8786e</color> <color>#ad8bc9</color> <color>#ed665d</color> <color>#67bf5c</color> <color>#ff9e4a</color> <color>#729ece</color> </color-palette>`
Tableau 10 Light 	Discontinued.	`<color-palette name="Classic 10 Light" type = "regular"> <color>#9edae5</color> <color>#dbdb8d</color> <color>#c7c7c7</color> <color>#f7b6d2</color> <color>#c49c94</color> <color>#c5b0d5</color> <color>#ff9896</color> <color>#98df8a</color> <color>#ffbb78</color> <color>#aec7e8</color> </color-palette>`
Tableau 20	Named **Tableau Classic 20** in	`<color-palette name="Classic 20" type = "regular"> <color>#9edae5</color> <col-`

	version 10.0 and higher.	<color>#17becf</color> <color>#dbdb8d</color> <color>#bcbd22</color> <color>#c7c7c7</color> <color>#7f7f7f</color> <color>#f7b6d2</color> <color>#e377c2</color> <color>#c49c94</color> <color>#8c564b</color> <color>#c5b0d5</color> <color>#9467bd</color> <color>#ff9896</color> <color>#d62728</color> <color>#98df8a</color> <color>#2ca02c</color> <color>#ffbb78</color> <color>#ff7f0e</color> <color>#aec7e8</color> <color>#1f77b4</color> </color-palette>
Gray 5 	Discontinued. Check out **Seattle Grays** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Gray 5" type = "regular"> <color>#cfcfcf</color> <color>#8f8782</color> <color>#414451</color> <color>#a5acaf</color> <color>#60636a</color> </color-palette>
Color Blind 10 	Discontinued. Check out **Color Blind** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Color Blind" type = "regular"> <color>#cfcfcf</color> <color>#ffbc79</color> <color>#a2c8ec</color> <color>#898989</color> <color>#c85200</color> <color>#5f9ed1</color> <color>#595959</color> <color>#ababab</color> <color>#ff800e</color> <color>#006ba4</color> </color-palette>

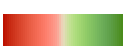
Traffic Light 	Discontinued. Check out **Traffic Light** in 10.0 and higher for the updated version of this palette.	`<color-palette name="Classic Traffic Light" type = "regular"> <color>#9fcd99</color> <color>#ffdd71</color> <color>#f26c64</color> <color>#69b764</color> <color>#ffc156</color> <color>#d82526</color> <color>#309343</color> <color>#dba13a</color> <color>#b10318</color> </color-palette>`
Purple-Gray 6 	Discontinued.	`<color-palette name="Classic Purple-Gray 6" type = "regular"> <color>#d7d5c5</color> <color>#d098ee</color> <color>#995688</color> <color>#94917b</color> <color>#dc5fbd</color> <color>#7b66d2</color> </color-palette>`
Purple-Gray 12 	Discontinued. Check out **Purple-Pink-Gray** in 10.0 and higher for the updated version of this palette.	`<color-palette name="Classic Purple-Gray 12" type = "regular"> <color>#dbd4c5</color> <color>#8b7c6e</color> <color>#d098ee</color> <color>#ab6ad5</color> <color>#d898ba</color> <color>#995688</color> <color>#b4b19b</color> <color>#5f5a41</color> <color>#ffc0da</color> <color>#dc5fbd</color> <color>#a699e8</color> <color>#7b66d2</color> </color-palette>`
Green-Orange 6 	Discontinued.	`<color-palette name="Classic Green-Orange 6" type = "regular"> <color>#b85a0d</color> <color>#39737c</color> <color>#ffd94a</color> <color>#3cb7cc</color> <color>#ff7f0f</color> <color>#32a251</color>`

		`</color-palette>`
Green-Orange 12 	Discontinued. Check out **Green-Orange-Teal** in 10.0 and higher for the updated version of this palette.	`<color-palette name="Classic Green-Orange 12" type = "regular"> <color>#ccc94d</color> <color>#82853b</color> <color>#86b4a9</color> <color>#39737c</color> <color>#ffd94a</color> <color>#b85a0d</color> <color>#98d9e4</color> <color>#3cb7cc</color> <color>#ffb977</color> <color>#ff7f0f</color> <color>#acd98d</color> <color>#32a251</color> </color-palette>`
Blue-Red 6 	Discontinued.	`<color-palette name="Classic Blue-Red 6" type = "regular"> <color>#e9c39b</color> <color>#ea6b73</color> <color>#6ba3d6</color> <color>#ac613c</color> <color>#f02720</color> <color>#2c69b0</color> </color-palette>`
Blue-Red 12 	Discontinued. Check out **Blue-Red-Brown** in 10.0 and higher for the updated version of this palette.	`<color-palette name="Classic Blue-Red 12" type = "regular"> <color>#f4737a</color> <color>#bd0a36</color> <color>#ddc9b4</color> <color>#ac8763</color> <color>#b5dffd</color> <color>#6ba3d6</color> <color>#e9c39b</color> <color>#ac613c</color> <color>#ffb6b0</color> <color>#f02720</color> <color>#b5c8e2</color> <color>#2c69b0</color> </color-palette>`
Cyclic	Discontinued. Check out	`<color-palette name="Classic Cyclic" type = "regular">`

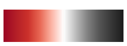
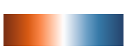
	**Hue Circle** in 10.0 and higher for the updated version of this palette.	<color>#6f63bb</color> <color>#8a60b0</color> <color>#ba43b4</color> <color>#c7519c</color> <color>#d63a3a</color> <color>#ff7f0e</color> <color>#ffaa0e</color> <color>#ffbf50</color> <color>#bcbd22</color> <color>#78a641</color> <color>#2ca030</color> <color>#12a2a8</color> <color>#1f83b4</color> </color-palette>
Green 	Discontinued. Check out **Green** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Green" type = "ordered-sequential"> <color>#09622a</color> <color>#1a7232</color> <color>#27823b</color> <color>#339444</color> <color>#69a761</color> <color>#94bb83</color> <color>#bccfb4</color> </color-palette>
Gray 	Discontinued. Check out **Gray** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Gray" type = "ordered-sequential"> <color>#1e1e1e</color> <color>#282828</color> <color>#333333</color> <color>#3f3f3f</color> <color>#4b4b4b</color> <color>#585858</color> <color>#666666</color> <color>#747474</color> <color>#838383</color> <color>#929292</color> <color>#a2a2a2</color> <color>#b2b2b2</color> <color>#c3c3c3</color> </color-palette>
Blue 	Discontinued. Check out	<color-palette name="Classic Blue" type = "ordered-sequential">

	Blue in 10.0 and higher for the updated version of this palette.	<color>#26456e</color> <color>#1c5998</color> <color>#1c73b1</color> <color>#3a87b7</color> <color>#67add4</color> <color>#7bc8e2</color> <color>#b4d4da</color> </color-palette>
Red 	Discontinued. Check out **Red** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Red" type = "ordered-sequential"> <color>#9c0824</color> <color>#b10c1d</color> <color>#c21417</color> <color>#cf1719</color> <color>#d8392c</color> <color>#e35745</color> <color>#f57667</color> <color>#f89a90</color> <color>#eac0bd</color> </color-palette>
Orange 	Discontinued. Check out **Orange** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Orange" type = "ordered-sequential"> <color>#7b3014</color> <color>#a33202</color> <color>#d74401</color> <color>#f06511</color> <color>#fd8938</color> <color>#fdab67</color> <color>#f0c294</color> </color-palette>
Area Red 	Discontinued.	<color-palette name="Classic Area Red" type = "ordered-sequential"> <color>#bd1100</color> <color>#c92b14</color> <color>#d43e25</color> <color>#e04e35</color> <color>#ea5e45</color> <color>#f46b55</color> <color>#fd7864</color> <color>#fe8b7a</color> <color>#fd9c8f</color> <color>#fbb3ab</color> <color>#f5cac7</color>

		</color-palette>
Area Green 	Discontinued.	<color-palette name="Classic Area Green" type = "ordered-sequential"> <color>#3c8200</color> <color>#4a8c1c</color> <color>#569735</color> <color>#60a24d</color> <color>#6cae59</color> <color>#7abc5f</color> <color>#8ac765</color> <color>#9ad26d</color> <color>#acdc7a</color> <color>#c3e394</color> <color>#dbe8b4</color> </color-palette>
Area Brown 	Discontinued.	<color-palette name="Classic Area-Brown" type = "ordered-sequential"> <color>#bb5137</color> <color>#bb6348</color> <color>#bb7359</color> <color>#c08262</color> <color>#cc8f63</color> <color>#d89c63</color> <color>#e4aa63</color> <color>#f0b763</color> <color>#f7c577</color> <color>#f6d29c</color> <color>#f3e0c2</color> </color-palette>
Red-Green Diverging 	Discontinued. Check out **Red-Green Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Red-Green" type = "ordered-diverging"> <color>#09622a</color> <color>#1e7735</color> <color>#2f8e41</color> <color>#69a761</color> <color>#a2c18f</color> <color>#cacaca</color> <color>#fc8375</color> <color>#df513f</color> <color>#d11719</color> <color>#bd1316</color> <color>#9c0824</color>

		</color-palette>
Red-Blue Diverging 	Discontinued. Check out **Red-Blue Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Red-Blue" type = "ordered-diverging"> <color>#26456e</color> <color>#1c5998</color> <color>#1c73b1</color> <color>#3a87b7</color> <color>#67add4</color> <color>#cacaca</color> <color>#fc8375</color> <color>#df513f</color> <color>#d11719</color> <color>#bd1316</color> <color>#9c0824</color> </color-palette>
Red-Black Diverging 	Discontinued. Check out **Red-Black Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Red-Black" type = "ordered-diverging"> <color>#1e1e1e</color> <color>#383838</color> <color>#565656</color> <color>#777777</color> <color>#9b9b9b</color> <color>#cacaca</color> <color>#fc8375</color> <color>#df513f</color> <color>#d11719</color> <color>#bd1316</color> <color>#9c0824</color> </color-palette>
Area Red-Green Diverging 	Discontinued.	<color-palette name="Classic Area Red-Green" type = "ordered-diverging"> <color>#4a8c1c</color> <color>#559633</color> <color>#5ea049</color> <color>#69aa56</color> <color>#75b65d</color> <color>#82c162</color> <color>#90cb68</color> <color>#a0d571</color> <color>#b1de7f</color> <color>#c7e298</color> <col-

		or>#e9dabe</color> <color>#fca294</color> <color>#fe8e7e</color> <color>#fd7e6b</color> <color>#f7705b</color> <color>#ef654d</color> <color>#e6583e</color> <color>#dc4930</color> <color>#d23a21</color> <color>#c82912</color> <color>#bd1100</color> </color-palette>
Orange-Blue Diverging 	Discontinued. Check out **Orange-Blue Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Orange-Blue" type = "ordered-diverging"> <color>#26456e</color> <color>#1c5998</color> <color>#1c73b1</color> <color>#3a87b7</color> <color>#67add4</color> <color>#7bc8e2</color> <color>#cacaca</color> <color>#fdab67</color> <color>#fd8938</color> <color>#f06511</color> <color>#d74401</color> <color>#a33202</color> <color>#7b3014</color> </color-palette>
Green-Blue Diverging 	Discontinued. Check out **Green-Blue Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Green-Blue" type = "ordered-diverging"> <color>#26456e</color> <color>#1c5998</color> <color>#1c73b1</color> <color>#3a87b7</color> <color>#67add4</color> <color>#cacaca</color> <color>#a2c18f</color> <color>#69a761</color> <color>#2f8e41</color> <color>#1e7735</color> <color>#09622a</color> </color-palette>
Red-White-	Discontinued.	<color-palette name="Classic Red-White-

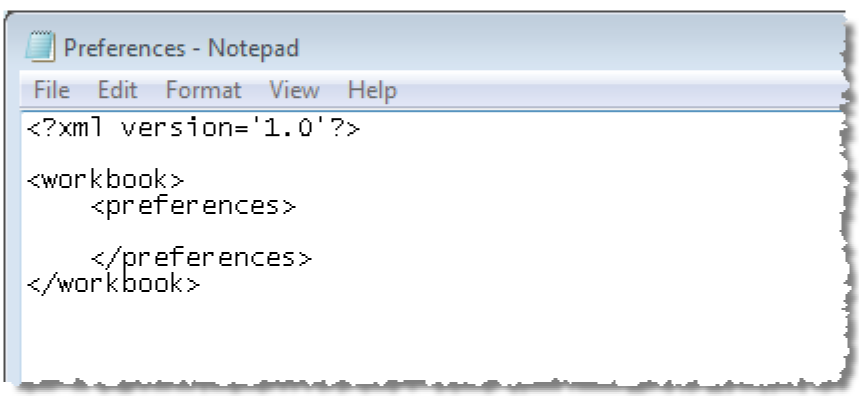
Green Diverging 	Check out **Red-Green-White Diverging** in 10.0 and higher for the updated version of this palette.	Green" type = "ordered-diverging"> <color>#09622a</color> <color>#297839</color> <color>#428f49</color> <color>#74af72</color> <color>#b9d7b7</color> <color>#ffffff</color> <color>#fcb4a5</color> <color>#e86753</color> <color>#cc312b</color> <color>#b41f27</color> <color>#9c0824</color> </color-palette>
Red-White-Black Diverging 	Discontinued. Check out **Red-Black-White Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Red-White-Black" type = "ordered-diverging"> <color>#1e1e1e</color> <color>#393939</color> <color>#575757</color> <color>#838383</color> <color>#bfbfbf</color> <color>#ffffff</color> <color>#fcb4a5</color> <color>#e86753</color> <color>#cc312b</color> <color>#b41f27</color> <color>#9c0824</color> </color-palette>
Orange-White-Blue Diverging 	Discontinued. Check out **Orange-Blue-White Diverging** in 10.0 and higher for the updated version of this palette.	<color-palette name="Classic Orange-White-Blue" type = "ordered-diverging"> <color>#26456e</color> <color>#2e5f8a</color> <color>#3679a8</color> <color>#6a9ec5</color> <color>#b7cde2</color> <color>#ffffff</color> <color>#ffc2a1</color> <color>#fb8547</color> <color>#d85a13</color> <color>#a84415</color> <color>#7b3014</color> </color-palette>

Red-White-Black Light	Discontinued.	`<color-palette name="Classic Red-White-Black Light" type = "ordered-diverging"> <color>#c6c6c6</color> <color>#d1d1d1</color> <color>#dddddd</color> <color>#e8e8e8</color> <color>#f3f3f3</color> <color>#ffffff</color> <color>#fff0f0</color> <color>#ffe0e1</color> <color>#ffd1d3</color> <color>#ffc2c5</color> </color-palette>`
Orange-White-Blue Light Diverging	Discontinued.	`<color-palette name="Classic Orange-White-Blue Light" type = "ordered-diverging"> <color>#c4d8f3</color> <color>#d0e0f6</color> <color>#dce8f8</color> <color>#e8effa</color> <color>#f3f7fd</color> <color>#ffffff</color> <color>#fff5eb</color> <color>#ffead8</color> <color>#ffe0c5</color> <color>#ffd6b1</color> <color>#ffcc9e</color> </color-palette>`
Red-White-Green Light Diverging	Discontinued.	`<color-palette name="Classic Red-White-Green Light" type = "ordered-diverging"> <color>#b7e6a7</color> <color>#c6ebb8</color> <color>#d5f0ca</color> <color>#e3f5db</color> <color>#f1faed</color> <color>#ffffff</color> <color>#fff0f0</color> <color>#ffe0e1</color> <color>#ffd1d3</color> <color>#ffc2c5</color> <color>#ffb2b6</color> </color-palette>`

Red-Green Light Diverging	Discontinued.	`<color-palette name="Classic Red-Green Light" type = "ordered-diverging"> <color>#b7e6a7</color> <color>#c1e6b4</color> <color>#cae6c0</color> <color>#d4e6cc</color> <color>#dde6d9</color> <color>#e5e5e5</color> <color>#ecdbdc</color> <color>#f2d1d2</color> <color>#f8c7c9</color> <color>#fcbdc0</color> <color>#ffb2b6</color> </color-palette>`

Restore a discontinued (classic) palette

1. Go to the My Tableau Repository folder in your Documents directory, and open the **Preferences.tps** file.
2. Between the opening and closing `workbook` tags, insert opening and closing `preferences` tags.



3. In the **Preferences.tps** file, between the "preferences" tags, paste the XML code for the discontinued palette (see **Hex values for discontinued color palettes** on page 2979). For example:

```
<color-palette name="Classic Blue Red 12" type = "regular">
<color>#f4737a</color> <color>#bd0a36</color>
<color>#ddc9b4</color> <color>#ac8763</color>
```

```
<color>#b5dffd</color> <color>#6ba3d6</color>
<color>#e9c39b</color> <color>#ac613c</color>
<color>#ffb6b0</color> <color>#f02720</color>
<color>#b5c8e2</color> <color>#2c69b0</color>
</color-palette>
```

Use straight quotation marks (as in `' '` or `" "`), not curly quotation marks, to delimit the palette's name and type.

4. Save the **Preferences.tps** file and then restart Tableau Desktop.

   When you open the Edit Colors dialog box and choose **Select Color Palette**, the color palette you added will be at the bottom of the palette list.

Optimize Workbook Performance

When we mention *performance*, we mean the speed with which you can work in Tableau. It might mean the speed of data analysis, such as working in Tableau Desktop on a large corporate database that you are accessing remotely. Or, it could refer to the speed of loading views or dashboards that you access on your desktop from Tableau Server.

If you are working with small data volumes, many of the recommendations in this section are optional for you. Your workbook performance is probably as speedy as you expect. But if you work with hundreds of millions of records, designing an efficient workbook is critically important to the speed at which you can work. We recommend that you review the topics and tips in this section before you start building your first view.

Making performance improvements almost always means making trade-offs. For example: Do you sacrifice speed to make sure you have the most current data? Is it important to include *all* the data for analysis, or will a speedier subset do? The bigger and more complicated the data, the longer it takes to interpret and render. But there are things you can do to accelerate the process. Think of fine-tuning your workbook performance as working on a puzzle with multiple pieces. Pick and choose from the topics in this section to find the pieces that solve your particular puzzle.

The topics in this section on performance discuss the big picture and then drill down to specific functionality. It starts with databases, data, and extracts, and then focuses on things that affect your data source, your workbooks, your calculations, and visualizations. The next set of topics discuss how to record and analyze workbook performance. Finally we look at how to reduce upload times to a Tableau server.

Learn more: For more in-depth and related information beyond this topic, see:
Designing Efficient Workbooks whitepaper on tableau.com
Designing Efficient Workbooks (1 hour) and Best Practices for Dashboard Performance (50 minutes) free, video presentations
The Tableau Performance Optimization Flowchart, The Tableau Performance Checklist, and the Tableau Performance Checklist blog series on the InterWorks blog.

General tips

This general list of considerations is a start in learning how to approach optimizing workbooks. It comes from the TL;DR page in the Designing Efficient Workbooks whitepaper.

- The cleaner your data is and the better it matches the structure of your questions, the faster your workbooks will run and the happier your life will be. Only connect to the data that you need for analysis.
- Extracts are a quick and easy way to make most workbooks run faster. If you don't need real-time data and aren't working over billions of rows of data, you should try them.
- The majority of slow dashboards are caused by poor design – in particular, too many charts on a single dashboard, or trying to show too much data at once. Keep it simple. Allow your users to incrementally drill down to details, rather than trying to show everything then filter (guided analysis).
- Don't work with data you don't need—both in terms of the fields you reference as well as the granularity of the records you return. Use filters, hide unused fields and aggregate.
- While reducing the data, make sure you use filters efficiently.
- Strings and dates are slow, numbers and Booleans are fast.
- There is no silver bullet for inefficient workbooks. Start by looking at the performance recorder to understand where the time is going. Long-running queries? Lots of queries? Slow calculations? Complex rendering? Use this insight to focus your efforts in the right direction.
- Upgrading to the newest version may boost performance without needing anything else.
- If it is slow in the data source, it will be slow in Tableau Desktop. If it is slow in Tableau Desktop, it will (almost always) be slow in Tableau Server.
- Performance tuning is highly individualized to the context of your environment, data, analysis, and workbook. What works in one situation may not work in another--test and iterate to see what works in your specific case.

Know Your Data at the Database Level

Understanding the impact that certain factors in the design of your database have on Tableau performance might give you information you can use to work with your database team to optimize data at the database level.

Enable support for referential integrity

Databases that support referential integrity support the Tableau Assume Referential Integrity feature, which improves the performance of inner joins. Joins cost time and resources to process on the database server. When you join multiple tables in a data source, Tableau uses functionality that is generally invisible to the user, called *join culling*. Join culling queries only the relevant tables instead of all tables defined in your join. Join culling only occurs where referential integrity is defined between tables. For more information, see **Assuming Referential Integrity for Joins** on page 899.

Make sure database permissions support creating temp tables

Does your database grant users permission to create and drop temporary tables, also known as temp tables, and does the environment have sufficient spool space for the queries being run? Tableau creates temp tables to help improve performance and add functionality. The temp tables temporarily hold information specific to a connection. If the creation of temp tables in the database is restricted, the overall performance of workbooks and views isn't as fast as it could be.

Create indexes for tables

Index the tables in your relational database. To successfully index your data set, identify the fields that you frequently filter on and add them to the index. If you have a field that you use as a context filter often, consider setting it as your primary index. If you are working with Access tables that have more than 200,000 rows of data, consider setting indexes on the tables. You can learn how to do this by searching for "index" in the Access online help. You can officially store 2 GB of data (approximately 1-2 million rows) in an Access database, but it performs poorly well below this limit.

Many database management system (DBMS) environments have management tools that will look at a query and recommend indexes that would help.

Break up your data

You can often improve performance if you partition a large database table into multiple smaller tables. For example, you can create a cluster of Access tables that addresses specific subsets of your data.

Use a database server

If you have a lot of data, you might consider storing it in a database server, such as Oracle, MySQL, or Microsoft SQL Server. The Professional Edition of Tableau can connect to these larger database servers.

Test Your Data and Use Extracts

You probably have no control over the hardware, processing power, and disk space that your database runs on. Or how many people access the database at any given time, how many services are running, or how robust the network is. But you can test the *raw performance* of the data before you begin to build a workbook, so that you understand the environment you're working in. And there are things you can do to affect the performance of the workbooks you create, starting with how you choose to connect to your data: the drivers you use, and whether your connection type is *live* or *extract*.

Test as close to the data as possible

A good way to test the raw performance of the data is to install Tableau Desktop on the computer where the data resides and to run some queries. This eliminates external factors such as network bandwidth and latency from performance results, and help you to understand the raw performance of the query in the data source.

Additionally, you can use the *localhost* name for the data instead of the DNS name to help determine if environmental factors such as slow name resolution or proxy servers are slowing performance.

Connect with native database drivers

When you connect to data with native database drivers, you will often experience significantly faster performance than when you connect to the same data with ODBC drivers. When you use native database drivers, make sure that you're using the latest version. Database vendors are always working to improve their drivers. For a list of drivers, see Drivers and Activation on the Tableau website.

Work with extracts instead of live data

Depending on your data, you can choose between a live or extract connection on the data source page. A *live* connection is a direct connection to your data. A Tableau data *extract* is a compressed snapshot of data stored locally and loaded into memory as required to render a Tableau visualization. Extracts are designed to use all parts of your computer's memory optimally.

There are several reasons to use an extract, but the main performance-related reason is if your query execution is slow. The extract data format is designed to provide a fast response to analytic queries. In this case, you can think of the extract as a query acceleration cache.

For more information about Tableau extracts, see **Extract Your Data** on page 1014. Another reference is a three-part blog post about extracts that starts with the first post, Understanding Tableau Data Extracts.

Reduce the amount of data

When you create an extract, use filters to exclude data that you don't need. Also, ask yourself if you need all of the records in a data source, or if you can limit the extract to a representative sample. For more information, see **Extract Your Data** on page 1014.

Hide unused fields

Hidden fields are not included when you create an extract. Use the **Hide All Unused Fields** option to hide unnecessary fields before you create an extract. This makes the extract smaller, which improves performance. For more information, see **Hide or Unhide Fields** on page 1155.

Optimize extracts

The **Compute Calculations Now** option materializes calculations in your extract, meaning that certain calculations are computed in advance and their values are stored in the extract. Depending on the complexity of the calculations used in your extract, this can potentially speed up future queries.

To materialize calculations, select a data source on the **Data** menu and then select **Extract** > **Compute Calculations Now**. For more information, see **Materialize Calculations in Your Extracts** on page 1047.

Use extracts for file-based data

In general it's best practice to import file-based data—text files such as CSV, Microsoft Excel spreadsheets, and Microsoft Access files, as well as statistical files like SPSS, SAS, and R—into Tableau. This makes queries perform much faster and also results in a much smaller file to store the data values.

If your data is a large text or Excel file, using an extract not only improves performance but also makes more functionality available to you. Note that if you connect Tableau to a large text file, you will be prompted to extract the data if Tableau discovers that the file is too large to perform well.

Avoid using custom SQL

In most cases, custom SQL runs slower than queries created by Tableau. Tableau cannot perform query optimizations on custom SQL. But in cases where you must use custom SQL, use an extract so that the query runs only once. For more information, see **Connect to a Custom SQL Query** on page 959.

Create Efficient Combined Data Sources

Now that you've made the initial connection to your data, consider how you set up your data source—especially how you combine data—to make it efficient.

If your analysis requires data from different databases, including different workbooks or files, you should consider the impact of relating, joining, or blending your data. Combined data sources require more processing power and take more time to execute.

A key factor in efficiency is reducing the amount of data you analyze.

Should I relate, join, or blend my data?

The default method of combining tables is with relationships. There are many aspects of relationships that make them efficient and performant. If you need to join or blend instead, make sure you truly do need to use those methods instead of a relationship.

When you consider joining tables or blending tables, think about where the data is coming from, the number of connections, and the number of records you have. If the workbook uses multiple tables from the same database, relating or joining the tables can improve performance and filtering control.

Combine tables carefully

Limit data sources to the fewest number of tables possible. In cases where you need access to many tables in a workbook, you may want to create separate data sources tailored to each analytical scenario.

Blending queries the data from both data sources at the level of the linking fields, and then merges the results of both queries together in memory. For this reason, best practice is to avoid data blending on dimensions with many unique values (for example, Order ID, Customer ID, or exact date/time).

Design for Performance While You Build a View

You’re connected to your data, and you’ve built your data source. The next step is to begin building a view in Tableau. There are a few things to keep in mind as you build your workbook so that it will perform faster while you’re building it, as well as after it is created.

Use the Describe field to get to know your data

Rather than dragging a dimension out onto the rows shelf so that you can eyeball the data—which requires Tableau to render the data as a crosstab, one of the slowest visualization types—use the **Describe field** option. Right-click the field in the **Data** pane to view a description of the data type as reported by the data source, as well as a sample of the dimension member values.

Keep workbooks a reasonable size

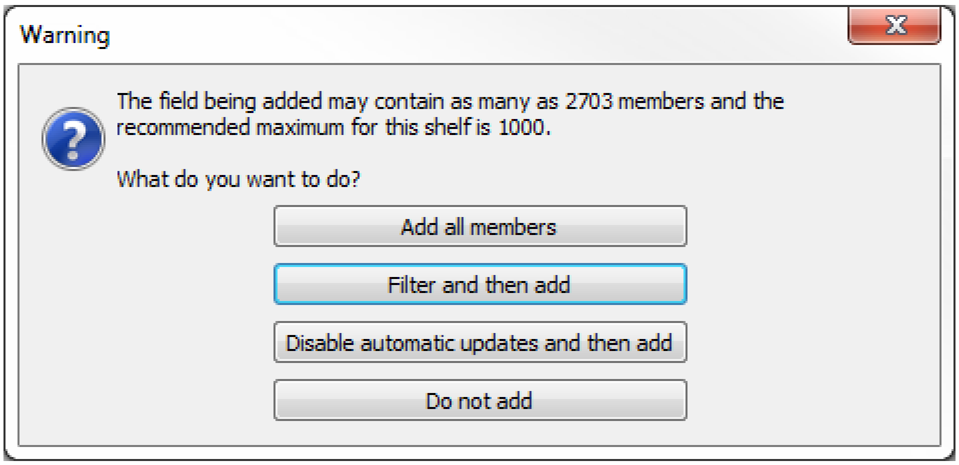
The fewer worksheets and data sources in a workbook, the faster it will perform. If you have a big topic to explore and are tempted to put everything into a single monolithic workbook, reconsider. Notice if your workbook starts to slow down with the addition of a new view and additional queries. If the performance starts to suffer as you enlarge the scope of your workbook, think about breaking your workbook into separate files.

Turn off Automatic Updates

When you place a field on a shelf, Tableau generates the view by automatically querying the data. If you are creating a dense data view, the queries might be time-consuming and significantly degrade system performance. In this case, you can turn off queries in Tableau while you build the view. You can then turn queries back on when you are ready to see the result. For more information, see **Turn off Automatic Updates to Boost Performance** on page 3014.

Look for warnings

Tableau displays a performance warning dialog box when you attempt to place a large dimension (with many members) on any shelf. The dialog box provides four choices, as shown below. If you choose to add all members, then you might experience a significant degradation in performance.



You might also see a warning when you attempt to create too many panes in a table. In this case, Tableau warns you that the requested table "contains more than the recommended maximum number of panes." It is best not to display more than the recommended number of panes, in part because you won't get a useful view.

Filter Your Data Carefully

Filters are an effective way to restrict the amount of data for analysis. The less data to analyze, the faster the query executes. In Tableau, there are a variety of ways to filter your data. You can create a filter on a data source, you can implement a filter when you create an extract, you can filter dimensions, measures, and dates. You can create a context filter that acts as an independent filter—then any other filters you set are dependent filters because they process only the data that passes through the context filter. You can also show a filter in a view that users can interact with. For information about when filters are executed, see **Tableau's Order of Operations** on page 330.

Filters provide enormous flexibility, but you should use them carefully. They can be computationally expensive, so reducing the number of filters can improve workbook performance.

Filter your data in the data source

When you create a filter on a data source, you reduce the amount of data in the data source. For systems that rely heavily on partitions or indexing, data source filters may yield tremendous control over the performance of queries issued by Tableau. For more information, see **Filter Data from Data Sources** on page 1006.

Use Keep Only filters instead of Exclude filters

Because **Exclude** filters load all of the data for a dimension, they run more slowly than **Keep Only** filters. For more information, see **Select to keep or exclude data points in your view** on page 1412.

Add filters first

If you are working with a large amount of data and have automatic updates turned off, you can inadvertently create a very slow query when you add filters to the view. Rather than building the view first, and then specifying filters, specify the filters first and then drag fields to the view. That way, when you run the update or turn on automatic updates, the filters are evaluated first.

Use context filters sparingly

Think of a context filter as an independent filter; any other filters you set process only the data that passes through the context filter. Only use context filters when they limit the size of the data set significantly. If you are setting filters that significantly reduce the data set size, and that will be used for more than several data views, you should set those filters as context filters. For more information, see **Improve View Performance with Context Filters** on page 1439. For more information about performance improvement with context filters, see **Speed up Context Filters** on page 1441.

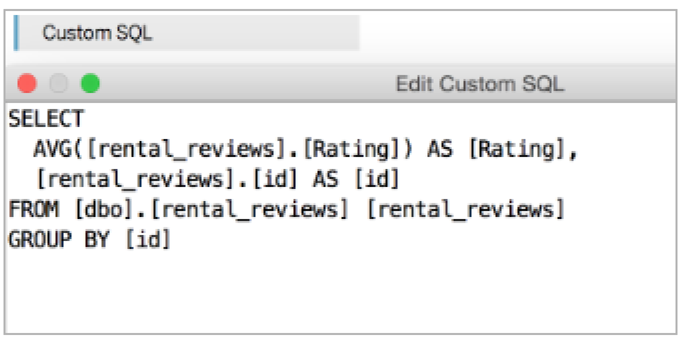
Use a set to filter data

If you want to filter a dimension to remove members based on a range of measure values, you should create a set rather than using a quantitative filter. For instance, you can create a set that only returns the Top 50 items in a dimension, rather than all of the items in a dimension.

When you create a group from a selection as described in **Group Your Data** on page 1173, make sure you include only the columns of interest. Each additional column in the set decreases performance.

Aggregate your data using custom SQL

As an alternative to filters, another effective way to restrict the amount of data for analysis is to use custom SQL to aggregate measures before you bring the data into Tableau. Suppose your reviews data is stored in a SQL Server table, and for your analysis, you're only interested in the data from the "rating" and "reviewer ID" fields. To pre-aggregate the data so that you bring into Tableau only the fields you're interested in, you might use the following custom SQL:



This custom SQL query reduces the amount of data brought into Tableau because it pre-aggregates the reviews and averages the values in the review ratings.

Filter your cube data

Here are some specific tips that apply to filtering cube data.

Filter on a specific dimension

If your cube has a single large dimension, you should set a filter directly on that dimension rather than setting a filter on another dimension or measure. For example, suppose you want to reduce the numbers of products being displayed in a view. It is much more efficient to set the filter directly on Products or to create a computed set based on Products (such as Top 10) rather to filter other fields such as Location or Profit.

Also, avoid selecting large numbers of members from a large dimension. When a dimension is large, keep the size of the filter to less than a thousand members.

Create a set

When creating a set from a selection as described in **Create Sets** on page 1186, make sure you've included only the columns of interest in the Create Set dialog box. Each additional column in the group decreases performance. For example, if you create a set that contains all regions with sales between 8,000 and 15,000, but you include a column that doesn't affect the members of the set, you might notice a performance decrease. To remove extra columns, right-click the column and select **Remove This Column** from the context menu.

Don’t apply sorts to levels

Avoid applying sorts to levels within a very large hierarchy in a cube.

Use the Exclude command

If you are working with a dimension whose root level is greater than 1000 but less than 100,000, avoid using the Filter dialog box to filter the data. Instead, drag the dimension to a shelf and use the **Exclude** command in the headers context menus to limit the data that is displayed in the view.

Create Efficient Calculations

When your data doesn’t provide all the information you need to answer your questions, you can create calculated fields to help with your analysis.

Within a calculated field you can define a hardcoded constant (such as a tax rate), do very simple mathematical operations like subtraction or multiplication (for example, revenues minus cost), use more complex mathematical formulas, perform logical tests (IF/THEN, CASE), do type conversions, send expressions to external services such as R, and much more.

There are different calculation types in Tableau:

Basic and aggregate calculations: These types of calculations are generated as part of the query to the underlying data source and are calculated in the database. In general, basic and aggregate calculations scale very well, and there are many database tuning techniques that can improve their performance.

Table calculations: These calculations are calculated by Tableau on the query result set. While this means more work for Tableau, table calculations are generally done over a much smaller set of records than are in the original data source. If table calculation performance is a problem (possibly because the result set returned to Tableau is very large) consider pushing some aspects of the calculation back to the data source layer. One way to do this is to aggregate the data and then perform the calculation on the aggregated data.

Level of detail (LOD) expressions: LOD expressions are generated as part of the query to the underlying data source and are calculated in the database. They are expressed as a *nested select*, so they are dependent on database performance. A table calculation or blending might perform better than a LOD expression, or vice versa.

If you suspect performance is slow due to a LOD expression, you might try replacing it with a table calculation or a data blend to see if performance improves. For an example, see Example 2 in **Tableau's Order of Operations** on page 330.

LOD expressions can be affected by join culling, so look at **Assuming Referential Integrity for Joins** on page 899 if your queries run slowly when you use LOD expressions.

For more information, see the Tableau whitepaper Understanding Level of Detail (LOD) Expressions.

Booleans and integers are faster

When you create calculated fields, the data type you use has a significant impact on the calculation speed. Integers and Booleans are generally much faster than strings. If your calculation produces a binary result (for example, yes/no, pass/fail, over/under), be sure to return a Boolean result rather than a string.

Use parameters for conditional calculations

A common technique in Tableau is to show a parameter control so users can select a value that determines how a calculation is performed. Typically, to give the user easy-to-understand options, it makes sense to create the parameter as a string type. But numerical calculations are much faster than string calculations, so take advantage of the **Display As** feature of parameters: that is, show text labels but use underlying integer values for the calculation logic. For more information, see **Create Parameters** on page 1218.

Convert date fields

Users often have date data that is not stored in native date formats—for example, a date might be a string or a numeric timestamp. You can use the DATEPARSE function if your data supports it—this function is available for non-legacy Microsoft Excel and text file connections, MySQL, Oracle, PostgreSQL, and Tableau data extract data sources. Otherwise, parse the field into a date string such as “2012-01-01”. ISO strings are preferred because they are not locale-specific. Then pass the value to the DATE function. If the originating data is a numeric field, converting it first to a string and then to a date is very inefficient. It is much better to keep the data as numeric and use DATEADD and date literal values to perform the calculation. The performance gains can be significant with large data sets. See **Date Functions** on page 2170 for more information about these functions.

Use ELSEIF logic statements

When working with complex logic statements, remember that ELSEIF is faster than ELSE IF, because a nested IF computes a second IF statement rather than being computed as part of the first.

Aggregate measures

If the views you create are slow, make sure you are working with aggregated measures. With disaggregated data, you might be trying to view many rows of data at once. You can reduce the number of rows by aggregating the data. To do this, select **Analysis** > **Aggregate Measures**.

Calculation tips

There are lots of little things you can do that can improve calculation performance.

- Distinct counting values is one of the slowest aggregation types in almost all data sources. Use the COUNTD aggregation sparingly.
- Using parameters with a wide scope of impact (for example, in a custom SQL statement) can affect cache performance.
- Filtering on complex calculations can potentially cause indexes to be missed in the underlying data.
- Script functions like RAWSQL and SCRIPT_* for integrating with external services can be slow, particularly if there are lots of values that need to be passed back and forth from the

DBMS/R server.

- Use NOW only if you need the time stamp level of detail. Use TODAY for date level calculations.
- Remember that all basic calculations are passed through to the underlying data—even literal calculations like label strings. If you need to create labels (for example, for column headers) and your data is very large, create a simple text/Excel file data source with just one record to hold them so they don't add overhead on the big data source.

Make Visualizations Faster

You’ve put in a lot of work to make your visualization—view, dashboard, or story—make a point or tell a story. You don’t want to lose your audience while you wait for the visualization to load. By following the guidelines in this topic, you can improve the speed of your visualizations.

Reduce the scope

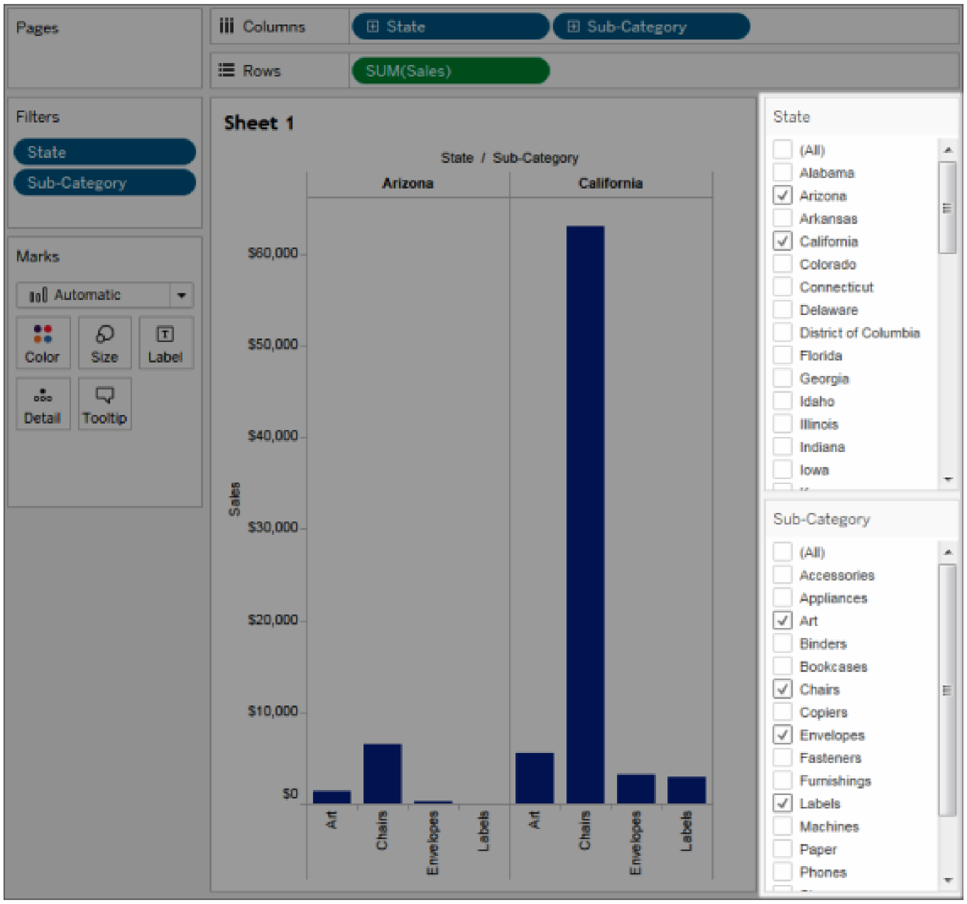
Whether you’re creating a view, dashboard, or story, it’s tempting to pack a lot of information into your visualization because it’s so easy to add more fields and calculations to the view and more sheets to the workbook. But the result can be that the visualization becomes slower and slower to render.

Remember that each worksheet runs one or more queries against the data, so the more sheets, the longer it takes to render the visualization.

Take advantage of the fact that Tableau is designed to deliver interactive visualizations to your users, and spread the data out across multiple visualizations, if you can. Be strategic when designing your visualization—the fewer sheets and data sources, the faster your visualization will perform.

Limit the number of filters you show in the view

Filters that you show in your view (formerly called quick filters) are a very powerful feature of Tableau that you can use to create rich, interactive visualizations for your users. See the filters highlighted on the right in the following screenshot:



When you add an interactive filter to a view, each filter in the view requires a query in order to populate the options. If you add a lot of interactive filters to your dashboard, it can cause the dashboard to take a long time to render.

Also, when you use “show relevant values” on a filter, it requires a query to update the shown values each time other filters are changed. Use this feature sparingly.

Reduce the number of marks on the view

While there is no hard and fast rule on what defines “too many marks,” be aware that more marks means that more processing power and memory is required to render them. You can find the number of marks by looking at the lower left of the Tableau Desktop window in the status bar. Watch out for large crosstabs and for maps with complex custom polygons. Keep in mind that too many data points on a view can also reduce the visual analytics value by causing information overload.

To avoid this problem, compile related views and connect them with action filters so that you can go from an overview to a more granular view as you explore the data. Make sure that you remove any unneeded dimensions from the **Detail** shelf. You can also try displaying your data in different types of views to see what's most effective.

Zoom without filtering

When users zoom in on a visualization with a large number of marks, Tableau doesn't filter out the marks you can't see. What changes is the view of the data, not the total number of marks. If you only need a subset of the data, filter out the data you don't need.

Record and Analyze Workbook Performance

Your workbook is done and you're wondering if its performance is taking a little longer than it should. To find out what's slowing it down, you can use a performance recording to evaluate your workbook. This is an especially good idea if you plan to share or publish the workbook.

The Performance Recording feature in Tableau records performance information about key events as you interact with a workbook. You can then view performance metrics in a workbook that Tableau creates to analyze and troubleshoot different events that are known to affect performance:

- Query execution
- Compiling query
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation
- Blending data
- Server rendering (Tableau Server only)

Tableau support may ask that you create a performance workbook as they work with you to diagnose performance issues.

Create a performance recording in Tableau Desktop

To start recording performance, follow this step:

Help > Settings and Performance > Start Performance Recording

To stop recording and view a temporary workbook containing results from the recording session, follow this step:

Help > Settings and Performance > Stop Performance Recording

You can now view the performance workbook and begin your analysis.

Click **File** > **Save** to save the workbook to the location you choose. If you're sending the recording to Tableau Support, save this workbook as a packaged workbook (.twbx) file, and then send it.

Interpret a performance recording workbook

A performance recording workbook contains two main dashboards: Performance Summary and Detailed Views. The Performance Summary dashboard provides a high-level overview of the most time-consuming events. The Detailed Views dashboard provides a lot more detail and is intended to be used by advanced users when building workbooks. The **Detailed Views** dashboard is only visible when the performance recording workbook is opened using Tableau Desktop.

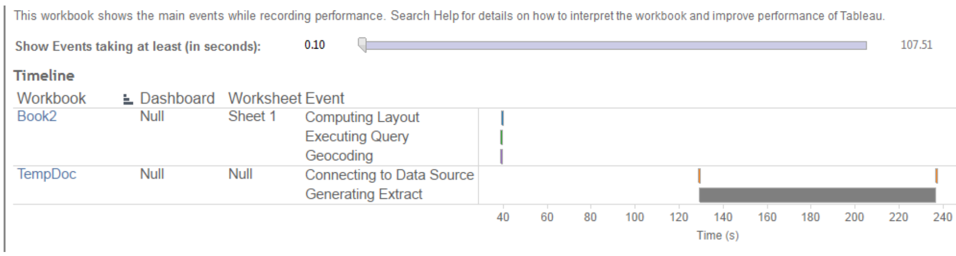
Performance Summary

The **Performance Summary** dashboard contains three views: **Timeline**, **Events**, and **Query**.

Timeline

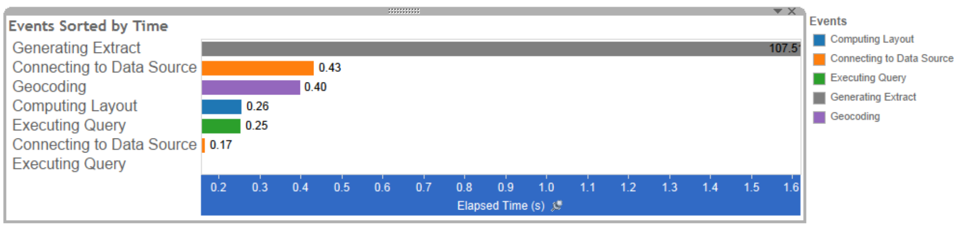
The uppermost view in the performance summary recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.

In the Timeline view, the **Workbook**, **Dashboard**, and **Worksheet** columns identify the context for events. The **Event** column identifies the nature of the event, and the final column shows each event’s duration and how it compares chronologically to other recorded events:



Events

The middle view in a performance summary dashboard shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.



Different colors indicate different types of events. The range of events that can be recorded is:

- Computing layouts

  If layouts are taking too long, consider simplifying your workbook.

- Connecting to data source

  Slow connections could be due to network issues or issues with the database server.

- Compiling query

  This event captures the amount of time spent by Tableau in generating the queries. Long compile query times indicate that the queries generated are complex. The complexity may be due to too many filters, complex calculations, or generally due to a complex workbook. Examples of complex calculations include, lengthy calculations, LOD calculations, or nested calculations. Try simplifying the workbook, using action filters or moving calculations to the underlying database.

- Executing query

  - For live connections, if queries are taking too long, it could be because the underlying data structure isn’t optimized for Tableau. Consult your database server's documentation. As an alternative, consider using an extract to speed performance.

- For extracts, if queries are taking too long, review your use of filters. If you have a lot of filters, would a context filter make more sense? If you have a dashboard that uses filters, consider using action filters, which can help with performance.

- Generating extract

  To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.

- Geocoding

  To speed up geocoding performance, try using less data or filtering out data.

- Blending data

  To speed up data blending, try using less data or filtering out data.

- Server rendering

  You can speed up server rendering by running additional VizQL Server processes on additional machines.

Query

If you click on an **Executing Query** event in either the **Timeline** or **Events** section of a performance summary dashboard , the text for that query is displayed in the Query section.

If you are connected to a published data source, the query text is displayed in XML. If you are connected to the data source directly, the query is displayed in SQL like shown below:

Query

```
SELECT "State"."ID" AS "ID",
  "StateSynonyms"."Name" AS "State_Name",
  "State"."ParentID" AS "State_ParentID"
FROM "StateSynonyms"
  INNER JOIN "State" ON (("State"."ID" = "StateSynonyms"."ParentID") AND ("State"."MapCode" = "StateSynonyms"."MapCode
```

If it makes sense, you can use the query text to work with your database team on optimizing at the database level. Sometimes the query is truncated and you'll need to look in the Tableau log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

Detailed Timeline

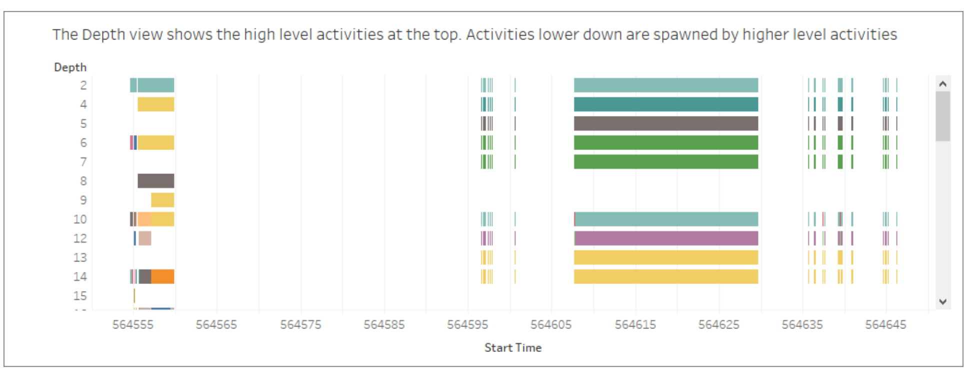
This view is the detailed version of the **Timeline** view that shows all events and separating individual items that were grouped in the **Timeline** view. It is intended to be used by advanced users during workbook designs.

Detailed Views

The **Detailed Views** dashborad contains **Depth**, **Exclusive CPU**, **Inclusive CPU**, and **Elapsed Time** views.

Depth

The **Depth** view is the uppermost view in the **Detailed Views** dashboard and provides insight into what happens when a request is made. This view is the most useful when filtered to a single user request. Examples of user requests are: loading a view, selecting a mark, or changing a filter.



Each bar on the depth view represents a single activity. An activity is a unit of work that is done as part of processing a user request. A single user request results in multiple activities. The length of each bar on the depth view is proportional to the elapsed time for the activity that the bar represents.

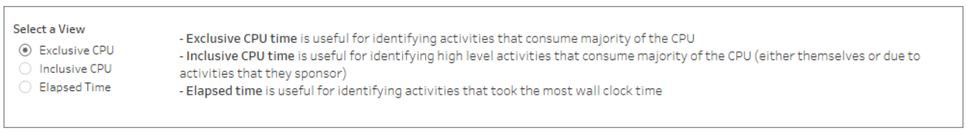
High level activities appear at the top of the view. Activities lower down are child activities generated by high level activities.

Hovering over each bar provides additional details about the activity and also highlights the corresponding row in the **CPU** and **Elapsed Time** view described in the next section.

In order to narrow down on what parts of the request took the most time, investigate long running activities at the highest levels.

CPU and Elapsed Time

The **CPU** and **Elapsed Time** views appear lowermost in the **Detailed Views** dashboard. You can toggle between **Exclusive CPU**, **Inclusive CPU** and **Elapsed Time** views by clicking on the radio buttons.



While the **Depth** view can help in quick visual identification of long running activities, it may not necessarily highlight activities that happen multiple times with each instance taking a small amount of time. The **Exclusive CPU**, **Inclusive CPU**, and the **Elapsed Time** views provide aggregate statistics for each activity. The number of times an activity took place is shown in the **Count** column and the total amount of time taken by a single activity is shown using the bar chart.

Reduce Upload Times to Tableau Server

You've created your workbook and analyzed its performance recording to make sure that it's been optimized for speed. Now you're ready to publish the workbook to Tableau Server. It isn't always possible to reduce upload times to Tableau Server, but here are some considerations to keep in mind.

Avoid publishing packaged workbooks

It is a best practice to publish workbooks (.twb files) and data sources separately, rather than publishing them together as packaged workbook (.twbx files). This may not make your initial uploads of workbooks and data sources any faster than publishing them together, but it makes republishing workbooks faster (if you don't have to also republish data sources), and it makes data sources available online for new workbooks. For data sources that are extracts, it also allows you to schedule automated refreshes.

Make extracts smaller

For Tableau extracts, you can help speed uploads by making your extract smaller. As you create extracts, consider:

- Removing unused fields from extracts.
- Making extracts smaller with sampling and filtering.
- Aggregating data for visible dimensions.

See also

Extract Your Data on page 1014

Publish a Data Source on page 3071

Turn off Automatic Updates to Boost Performance

When you place a field on a shelf, Tableau generates the resulting view by querying the data source. When you create a complex data view that involves many fields, these queries can be time-consuming. To improve performance, you can turn off automatic updates while composing a view, and then query the data source when you're done.

Automatic updates for worksheets

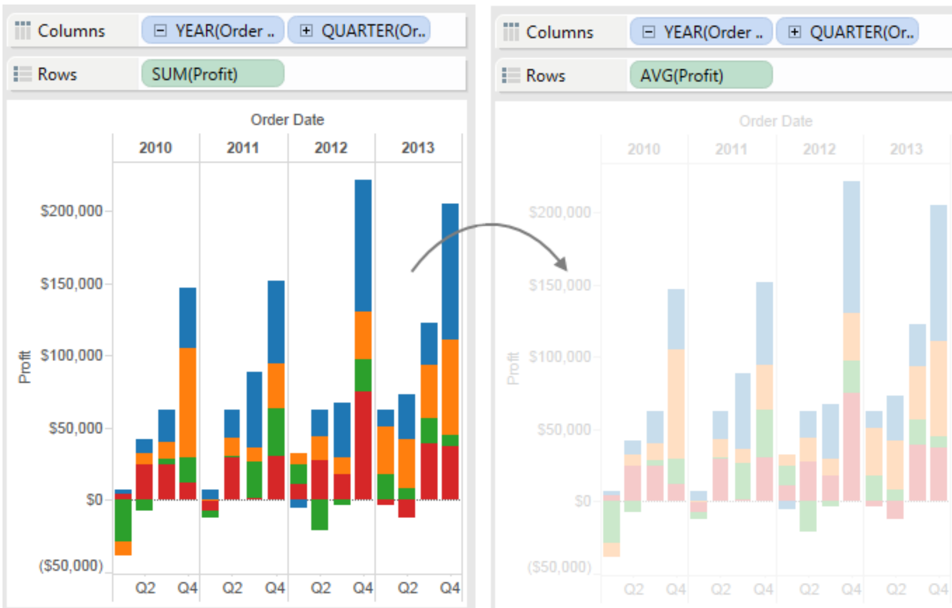
To turn off updates for worksheets, click the **Pause Auto Updates** button  on the toolbar. You can also toggle automatic updates on and off by pressing F10 (Option-Command-0 on a Mac) on your keyboard.

While automatic updates are off, you can update the view at any time by clicking the **Run Update**  button on the toolbar, or by pressing F9 (Shift-Command-0 on a Mac).

Identifying invalid states when automatic updates are off

It's possible to enter an invalid state when automatic updates are turned off. When this happens, the view is desaturated and invalid commands are disabled. The view and commands become available again when you click **Run Update** on the toolbar.

For example, the view below has automatic updates turned off. When we changed the aggregation for Profit from a summation to an average, the view became desaturated to indicate that the current view is invalid.



Automatic updates for filters

When you change a filter, even when you turn off automatic updates for worksheets, Tableau continues to query the data source to update the view. When you work with a large data set, this continuous query action can degrade performance, so Tableau also gives you an option to pause automatic updates for filters. If you pause automatic updates for filters, you can make as many changes as you need without having to wait for Tableau to update the filter each time.

To pause automatic updates for filters, click the drop-down arrow on the **Pause Auto Updates** button on the toolbar and select **Auto Update Filters** to clear the check mark .

To resume automatic updates for filters, click the drop-down arrow on the **Pause Auto Updates** button on the toolbar and select **Auto Update Filters** again. You can also click the

Run Update  button on the toolbar to manually update the view at any time.

How Automatic Updates Affect Dashboards and Stories

You configure automatic updates on a per-view basis. This means that you can have a dashboard where some views update automatically, and others don't. Similarly, you can have a story where some story points update automatically, and others don't. But when dashboards or stories are published to Tableau Server, automatic updates affect either none of the contributing views in a story or dashboard, or all of the views.

Save Your Work

You can save your work at anytime while analyzing or interacting with data in Tableau.

For Tableau Desktop

In Tableau Desktop, there are several ways for you to save your work:

- **Automatically save a workbook** - Automatically saves the workbook in the same location as the original file. In the event of a crash, a recovered version is available.
- **Save a workbook** – Saves all open worksheets.
- **Save a packaged workbook** – Saves the workbook along with all referenced local file data sources and images into a single file.
- **Save a bookmark** – Saves the current worksheet.

You can share workbooks and bookmarks with your co-workers, provided they can access the relevant data sources that the workbook uses. If your co-workers do not have access to the data sources, you can save a packaged workbook.

Custom fields such as binned measures, calculated fields, groups, and sets are saved with workbooks and bookmarks.

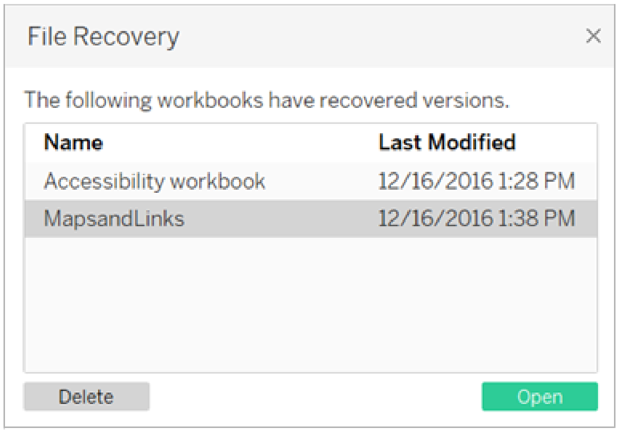
Automatically save a workbook

Tableau Desktop automatically saves your work for you every few minutes - no more losing hours of work if Tableau Desktop closes unexpectedly. This feature is enabled by default, but you can turn it off from the toolbar under **Help** >**Settings and Performance**>**Enable Autosave**.

> **Note**: If you don't see this option, your System Administrator may have disabled this feature.

If Tableau crashes, a recovered version of the workbook is automatically created with a .twbr extension and saved in the same location as the original file or in your **My Tableau Repository/Workbooks** folder. New workbooks are saved with the name "Book1" plus a numeric ID. When you reopen Tableau, a recovery dialog box shows a list of the recovered files that you can select and open to continue in your flow.

You can also delete unwanted files from this same dialog box.



For more information about how to turn this feature on or off during installation, see Before You Install in the Tableau Desktop Deployment guide.

Save a workbook

When you open Tableau Desktop, it automatically creates a new workbook. Workbooks hold the work you create and consist of one or more worksheets. Each worksheet contains a particular view of your data.

To save a Tableau workbook:

1. Select **File** > **Save**.
2. Specify the workbook file name in the **Save As** dialog box.

By default, Tableau saves the file with the .twb extension. By default, Tableau saves your workbook in the **Workbooks** folder in your My Tableau Repository. You can find this repository in your Documents folder. However, you can save Tableau workbooks to any directory you choose.

Tableau file names cannot include any of the following characters: forward slash (/), backslash (\), greater-than sign (>), less-than sign (<), asterisk (*), question mark (?), quotation mark ("), pipe symbol (|), colon (:), or semicolon (;).

To save a copy of a workbook you have open:

- Select **File** > **Save As** and save the file with a new name.

Save a packaged workbook

Packaged workbooks contain the workbook along with a copy of any local file data sources and background images. The workbook is no longer linked to the original data sources and images. These workbooks are saved with a .twbx file extension. Other users can open the packaged workbook using Tableau Desktop or Tableau Reader, and do not need access to the data sources that the workbook includes.

To learn more about how to save your workbook as a packaged workbook, see **Packaged Workbooks** on page 3022.

Save a bookmark

You can save a single worksheet as a Tableau bookmark. When you save the bookmark, Tableau creates a snapshot of the worksheet. Bookmarks can be accessed from any workbook using the Bookmarks menu. When you open a bookmarked worksheet, it adds the worksheet to your workbook in the state that it was in when it was bookmarked. It will never update or change automatically. Bookmarks are convenient when you have worksheets that you use frequently.

To save a Tableau bookmark:

1. Select **Window** > **Bookmark** > **Create Bookmark**.
2. Specify the bookmark file name and location in the Create Bookmark dialog box.

Tableau saves the file with a .tbm extension. The default location is the Bookmarks folder in the Tableau Repository. However, you can save bookmarks to any location you choose. Bookmarks that are not stored in the Tableau repository do not appear on the **Bookmark** menu.

You can organize bookmarks into folders in the same way you organize files or documents. This can be useful when you have a large number of bookmarks to manage. For example, you might organize bookmarks based on employee name, product types, or sales results. You can organize bookmarks by creating a new folder, renaming an existing folder, renaming existing bookmark files, and so on.

Delete bookmarks the same way you would delete any other file on your computer. After you delete a bookmark from the Bookmarks folder in the Tableau Repository, it is removed from the **Bookmarks** menu the next time you start Tableau.

Note: While bookmarks are generally a snapshot of the worksheet and include the data connection, formatting, etc., a bookmark does not include parameter values and the current page setting on the Pages shelf.

For web authoring

When creating, editing, and interacting with views on Tableau Server or Tableau Online, there are a couple of different ways for you to save your work:

- **Save a workbook** - saves a workbook in the project you specify.
- **Save a copy of a workbook** - saves a copy of the workbook in the project you specify.
- **Save changes as a custom view** - saves changes as a custom view, which is related to the original view and updates when the original view is updated.

Save a workbook

When you create a new workbook, or edit an existing workbook on Tableau Server or Tableau Online, you can save your work at any time.

To save a workbook:

- In web editing mode, select **File** > **Save**.

  Note: If the workbook has never been saved, you must select **File** > **Save As**.

Save a copy of an existing workbook

Sometimes you don't want to overwrite an existing view with your changes. In cases like these, you can save a copy of an existing workbook. When you do this, the existing workbook remains unchanged, and a copy of it is created for you to edit as you wish.

Note: When you save a copy of an existing workbook, it is not updated when the original workbook is updated or republished. It is also not deleted when the original view is deleted. If you would like to create a view that updates with the original, see the **Save changes as a custom view** on the facing page section

To save a copy of a workbook:

1. In web editing mode, select **File** > **Save As**.
2. In the Save Workbook dialog box that opens, do the following:
    1. For **Name**: Enter a name for the workbook.
    2. For **Project**: Select the project in which you would like to save the workbook.
    3. (Optional) Select **Show sheets as tabs** to display all worksheets, dashboards, and stories in the workbook as separate tabs.
    4. Click **Save**.

> **Note**: The person who published the data source to Tableau Server or Tableau Online may have also set up credentials for accessing the published data source included in the workbook you are about to save. These authentication options should appear in the Save Workbook dialog box. For more information, see the authentication types in the Set Credentials for Accessing Your Published Data topic.

Save changes as a custom view

If you notice you are making the same changes to a view every time you open it, you might want to consider saving the changes as a custom view. This option is not available in web editing mode, but can be accessed when you open a view to interact with it.

A custom view does not change the original, but is related to it. If the original view is updated or republished, the custom view is also updated.

You can also choose whether your custom views are visible to other users (public), or only to you (private).

For more information about custom views and how to create them, see **Use Custom Views** on page 3224.

See Also

Use Custom Views on page 3224

Packaged Workbooks

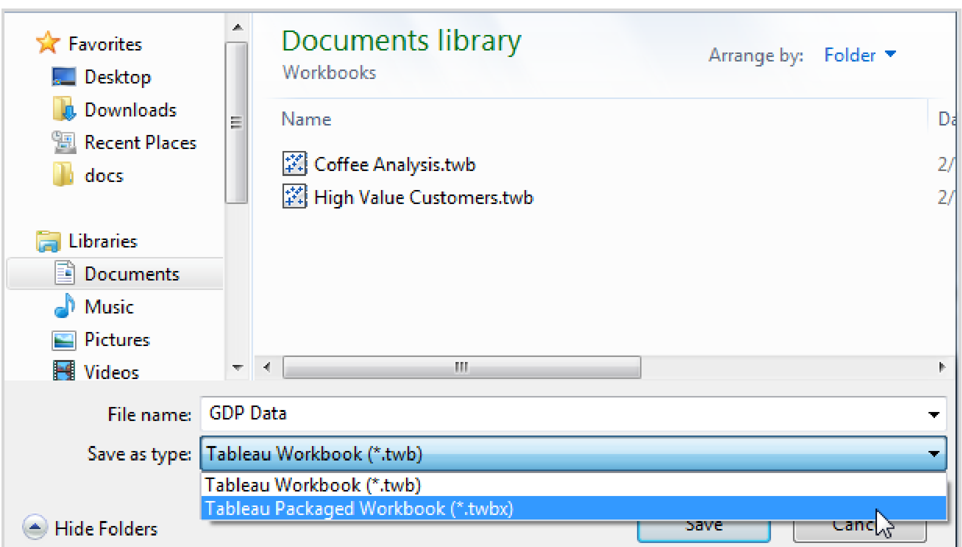
Workbooks often reference external resources. For example, workbooks might reference background images or local file data sources such as Excel files, Access files, and Tableau extract files (.hyper or .tde).

When you save a workbook, links to these resources are also saved. The next time you open the workbook, the views are automatically updated with any changes that may have occurred to the data and images. In most cases, you will want to save the workbook in this way. But if you plan to share the workbook with someone who does not have access to the referenced resources or to Tableau Server, you might want to save a packaged workbook instead.

Packaged workbooks contain the workbook along with a copy of any local file data sources and background images. The workbook is no longer linked to the original data sources and images. These workbooks are saved with a .twbx file extension. Other users can open the packaged workbook using Tableau Desktop or Tableau Reader.

Create a .twbx with file-based data sources

1. Select **File > Save As**.
2. Specify a file name for the packaged workbook in the Save As dialog box.
3. Select **Tableau Packaged Workbooks** on the **Save as type** drop-down list.



4. Click **Save**.

The default location is the **Workbooks** folder of the Tableau repository. However, you can save packaged workbooks to any directory you choose.

The following files are included in packaged workbooks:

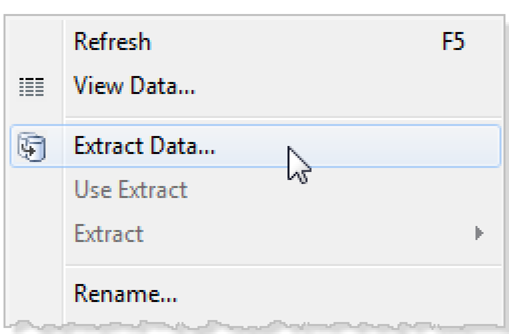
- Background images
- Custom geocoding
- Custom shapes
- Local cube files
- Microsoft Access files
- Microsoft Excel files
- Tableau extract files (.hyper or .tde)
- Text files (.csv, .txt, etc.)

> If you are sharing packaged workbooks that contain Microsoft Excel or Access 2007 data sources, the people opening the workbook must either have Microsoft Excel and Access 2007 or the Office 2007 Data Connectivity Components installed on their computers. The data connectivity components are available on the Tableau Tableau Drivers page.

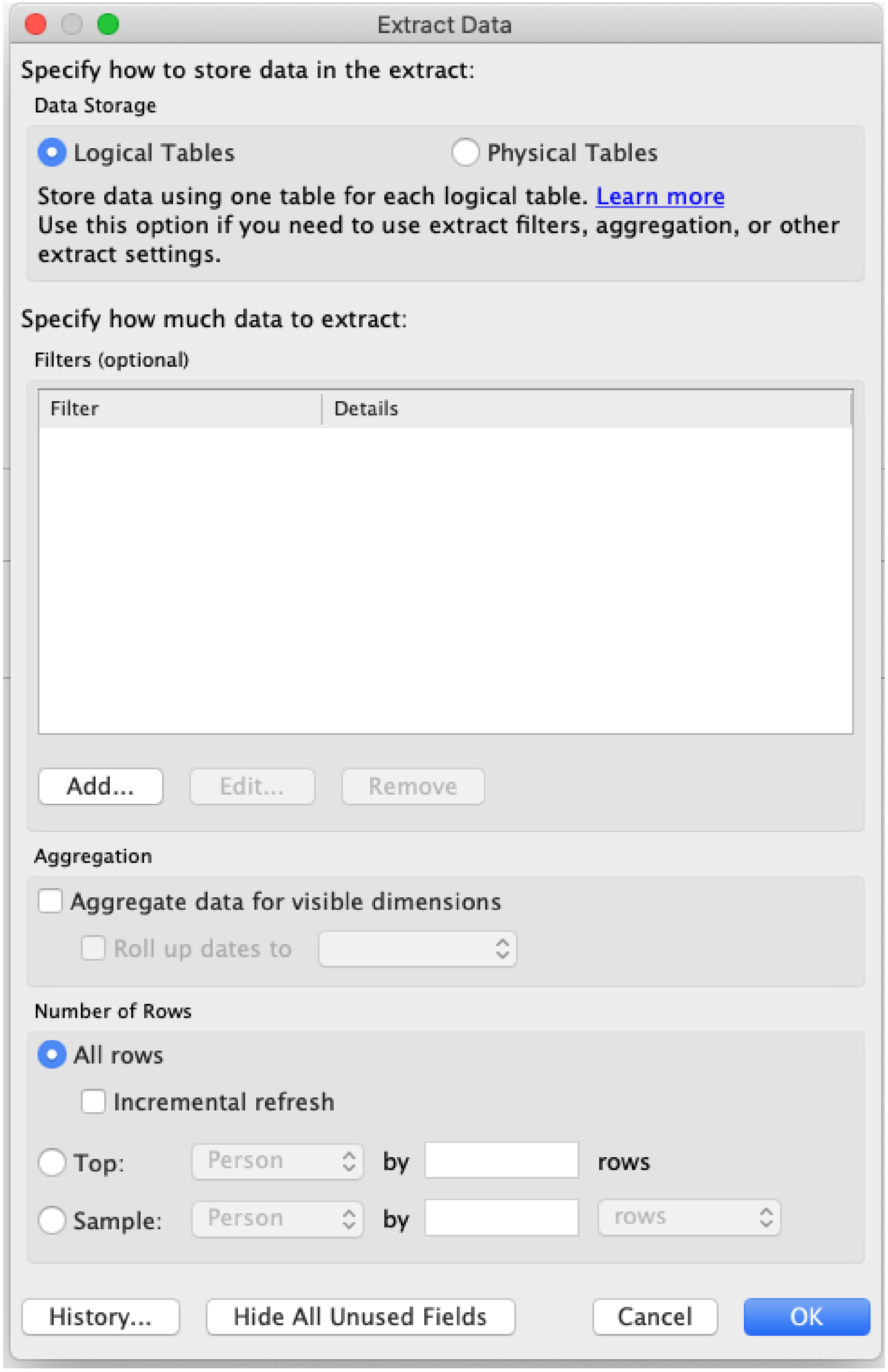
Create a .twbx with non-file-based data sources

If the workbook contains connections to enterprise data sources or other non-file-based data sources, such as Microsoft SQL, Oracle, or MySQL, the data must be extracted from the data sources for it to be included in a packaged workbook (.twbx).

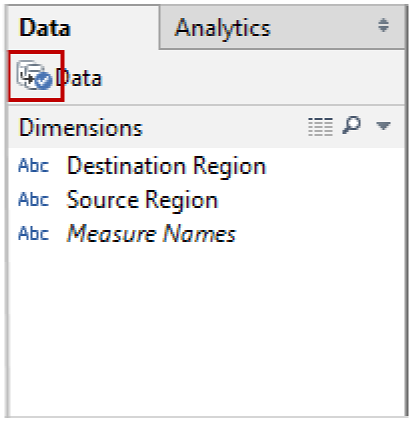
1. In the workbook, right-click the data source in the Data pane and choose **Extract Data**.



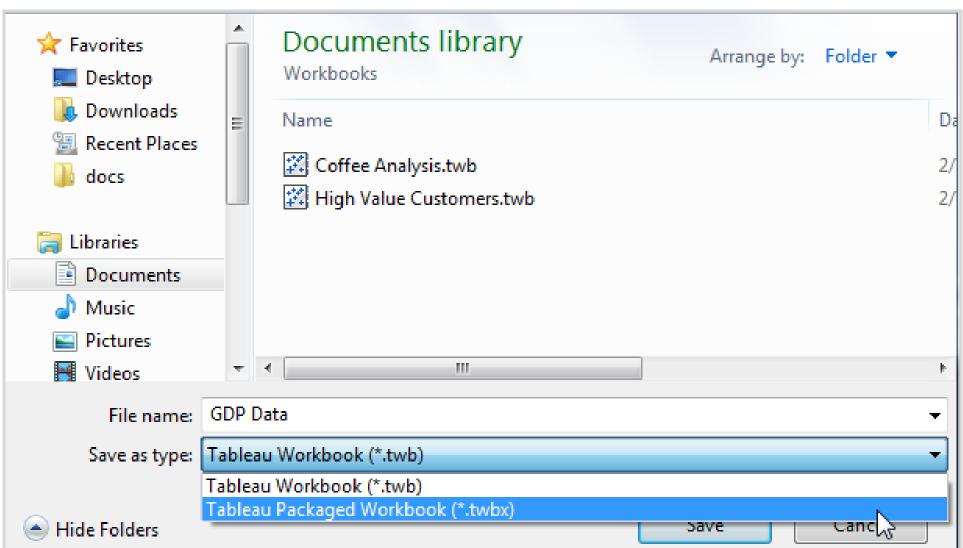
2. In the Extract Data dialog box, click the **Extract** button to extract all data from the data source.


Extract Data
Specify how to store data in the extract:
Data Storage
Logical Tables
Physical Tables
Store data using one table for each logical table. Learn more
Use this option if you need to use extract filters, aggregation, or other extract settings.
Specify how much data to extract:
Filters (optional)
Filter
Details
Add...
Edit...
Remove
Aggregation
Aggregate data for visible dimensions
Roll up dates to
Number of Rows
All rows
Incremental refresh
Top:
Person
by
rows
Sample:
Person
by
rows
History...
Hide All Unused Fields
Cancel
OK

After the extract completes, the data source icon changes to indicate that an extract is active for that data source. Instead of a single cylinder, there are two cylinders connected by an arrow.



3. **Optional:** Repeat the above steps for each data source in the workbook.
4. Select **File** > **Save As**.
5. From the **Save as type** drop-down menu, select Tableau Packaged Workbook (*.twbx).

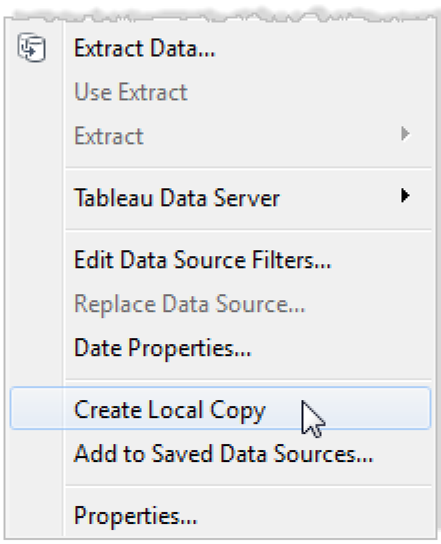


After the extracts have been created for all non-file-based data sources and the packaged workbook has been saved, you can send your workbook.

Create a .twbx with Tableau Server data sources

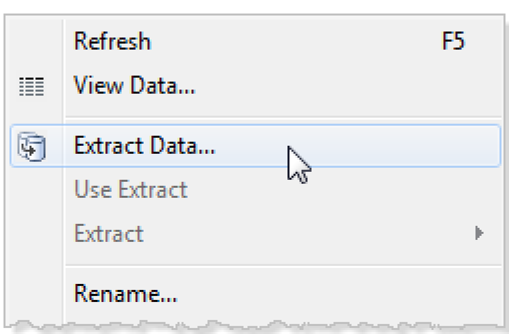
If the workbook contains connections to a published Tableau Server data source, you must download a local copy of the Tableau Server data source, take an extract of it, and then replace the connection to the local copy for it to be included in a packaged workbook (.twbx).

1. In the workbook, right-click the published data source in the Data pane, and then select **Create Local Copy**.

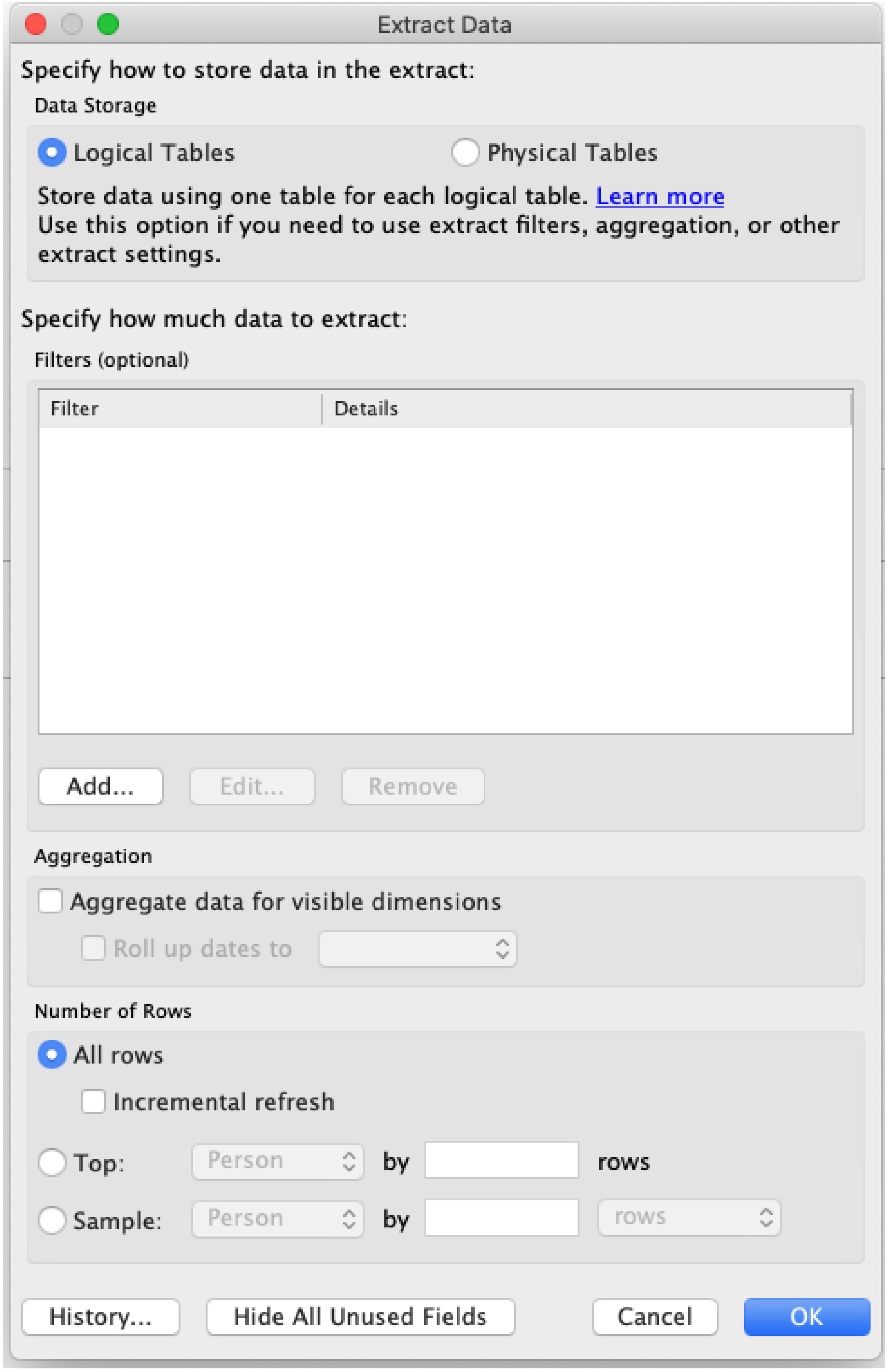
   A copy of the published data source is added to the Data pane.

2. Right-click the local copy, and select **Extract Data**.
   Note: If the local copy is a published extract you can skip this step.

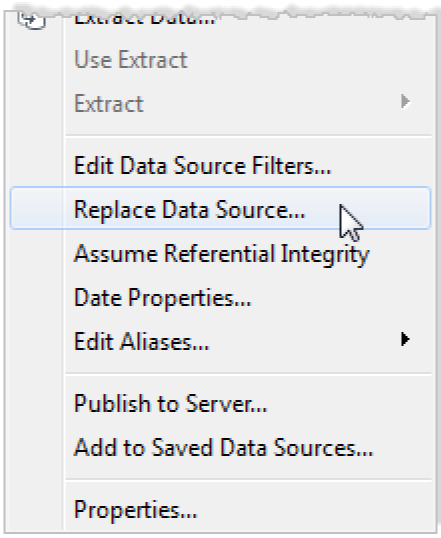
   

3. In the Extract Data dialog box, click the **Extract** button to extract all data from the data source. Creating an extract of the data source allows the person you are sharing the

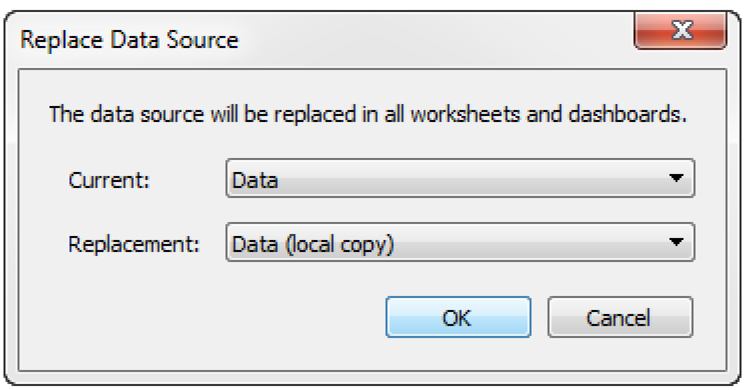
workbook with to have access to a copy of the data source.



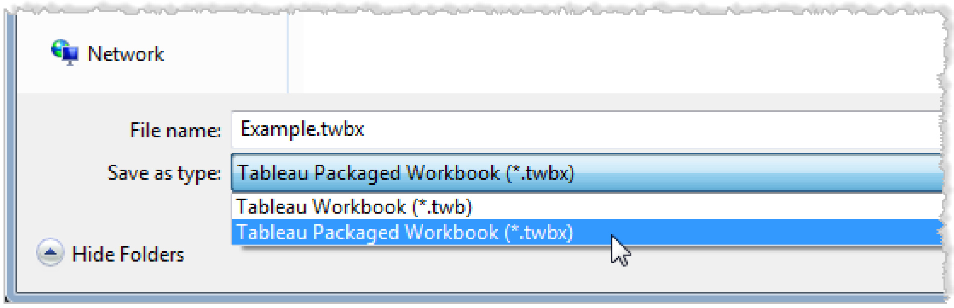
4. In the Data pane, right-click the published data source, and then select **Replace Data Source**.



5. Verify that the published data source will be replaced by the local data source, and then click **OK**.



6. Right-click the published data source, and then click **Close**.
7. Select **File** > **Save As**.
8. From the **Save as type** drop-down menu, select Tableau Packaged Workbook (*.twbx).



After the local copy and extract of the local copy is created and the packaged workbook saved, you can send your workbook.

Unpackage a .twbx

Packaged workbooks can be unpackaged.

On a Windows or macOS computer, rename the file with a .zip extension (for example, from myfile.twbx to myfile.zip) and then double-click it.

When you unpackage a workbook, you get a regular workbook file (.twb), along with a folder that contains the data sources and images that were packaged with the workbook.

Save Workbooks to Tableau Public

If you want to share your data discoveries with the world outside of your organization, you can save your workbook to Tableau Public, a free cloud service. On Tableau Public, anyone can interact with your views, or download your workbooks or data sources. For information, go to the Tableau Public website.

Save a workbook

1. With your workbook open in Tableau Desktop, select **Server** > **Tableau Public** > **Save to Tableau Public**.

   > **Note:** This option is available only if you've created a viz that contains at least one field.

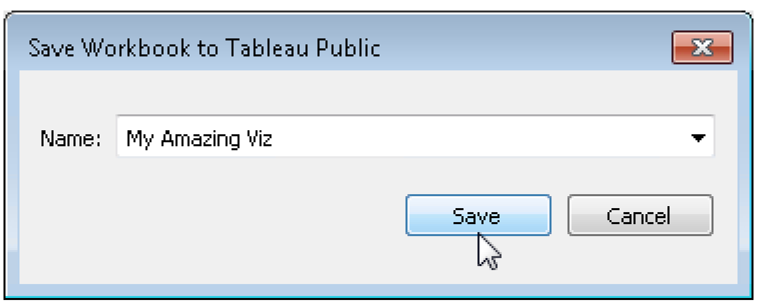
2. Sign in using your Tableau Public account.

If you don’t have an account, select the link to create a new one.

3. Type a name for the workbook and click **Save**.

   When you save a workbook to Tableau Public, the publishing process creates an extract of the data connection.

   > **Tip:** The title becomes part of your view’s metadata. Use a unique title that will help others find it when they search. (The title shown in the image is a good example of how *not* to name your workbook.)

   After the workbook is published, you are redirected to your account on the Tableau Public website.

On your profile page on Tableau Public, do any of the following to customize your profile:

- Hover the pointer over a viz to get access to actions such as selecting it as your featured viz, or hiding, downloading, or deleting it.
- Hover the pointer over a viz and then select **View** to open the viz’s home page. There you

can select **Edit Details** to customize metadata such as workbook name and description, add a permalink, and change other settings.

- To get a link to share on social media or code to embed in a web page, display a view, and then click **Share** at the bottom of the view. (You can get links and embed code for other Tableau Public users' views this way, too.)

Revert a Workbook to the Last Saved Version

You can revert a workbook at anytime to undo all of the changes you've made to it since you last saved it.

To revert to the last saved version of a workbook:

- In Tableau Desktop, select **File** > **Revert to Saved**, and then select **Revert** in the warning dialog box that opens.
- In web authoring mode, select **File** > **Revert**.

The Revert command is only available for workbooks (.twb) that do not have connections to Extract data sources.

Export Views from Tableau Desktop to Another Application

There are several ways to get views and workbooks out of Tableau Desktop and into a presentation, report, or web page.

Note: If you're using Tableau Online or Tableau Server, instead see **Download Views and Workbooks** on page 3328 and **Link to a PNG, PDF, or CSV of a View** on page 3327.

Copy a view as an image

You can quickly copy an individual view as an image and paste it into another application, such as Microsoft Word or Excel. If you're using Tableau Desktop on macOS, a TIFF (Tagged

Image File Format) image is copied to the clipboard. On Windows, a BMP (Bitmap) image is copied.

1. Select **Worksheet** > **Copy** > **Image**.
2. In the Copy Image dialog box, select the elements you want to include in the image. If the view contains a legend, under Image Options, select the legend layout.
3. Click **Copy**.
4. Open the target application and paste the image from the clipboard.

Export a view as an image file

To create an image file you can reuse, export the view rather than copy it. You can choose BMP, JPEG, or PNG format on macOS, with the additional option of EMF on Windows. Note, however, that EMF format substitutes Tableau fonts (Tableau Regular, Tableau Semibold, etc.) with a similar font.

1. Select **Worksheet** > **Export**> **Image**.
2. In the Export Image dialog box, select the elements you want to include in the image. If the view contains a legend, under Image Options, select the legend layout.
3. Click **Save**.
4. In the Save Image dialog box, specify a file location, name, and format. Then click **Save**.

Export as a PowerPoint presentation

When you export a workbook to Microsoft PowerPoint format, selected sheets become static PNG images on separate slides. If you export a story sheet, all story points export as separate slides. Any filters currently applied in Tableau are reflected in the exported presentation.

> **Tip:** To optimize a dashboard for PowerPoint, on the Dashboard tab, choose **Size** > **Fixed Size** > **PowerPoint (1600 x 900)**.

To export a workbook to PowerPoint:

1. Select **File** > **Export as PowerPoint**.
2. Select the sheets you want to include in the presentation. (Hidden sheets can also be included.)

The exported PowerPoint file reflects the file name of your workbook, and the title slide states the workbook name and the date the file was generated.

Tip: Choose **File** > **Page Setup** to show or hide titles, views, legends, and captions for an individual sheet. (These options in the Show section aren't available for dashboards.)

Export to PDF

To create a vector-based file that embeds the Tableau fonts, print to PDF. After customizing the layout of page elements using the **File** > **Page Setup** dialog, choose **File** > **Print to PDF**. For detailed instructions, see **Print Views from Tableau Desktop** on page 3044

Export Data from Tableau Desktop

You can export the data in a Tableau data source, including all or part of the records from your original data. Alternatively, you can export only the portion of data used to generate the view.

Note: If you're using Tableau Online or Tableau Server, instead see **Download Views and Workbooks** on page 3328 and **Link to a PNG, PDF, or CSV of a View** on page 3327.

Export data in the data source

After you join tables from one or more connections and make general customizations (for example, create a calculated field, pivot fields, create groups, apply data source or extract filters, etc.) to your Tableau data source, you might want to share or reuse the data in its new form. You can do this by using one of the methods listed below.

- **Export your data to .csv file** on the facing page
- **Extract your data** on the facing page
- **Export the data source** on the facing page

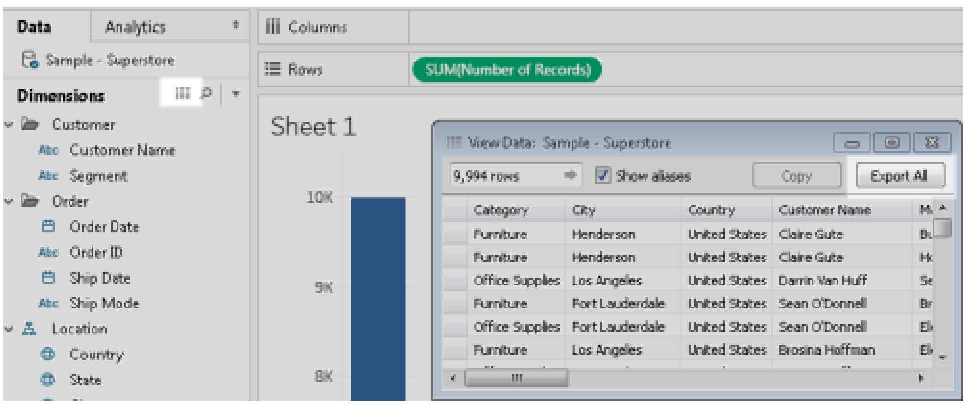
Note: The export may exclude some table calculations and level of detail expressions.

Export your data to .csv file

Because the .csv format is one of the most simple structured formats for data, it's supported by a wide range of tools, databases, and programming languages. Exporting your data in the Tableau data source using this format creates an independent data set and can be a convenient and flexible way to share your data with others.

There are two primary ways you can export your data in the data source to a .csv file in Tableau: from the Data Source page and from the view.

- **From the Data Source page:** On the Data Source page, select **Data** > **Export Data to CSV** to export all the data in your data source to .csv file.
- **From the view:** On the sheet tab, drag a field to the Columns or Rows shelf, click the View Data icon in the Data pane, and click the **Export All** button.



Extract your data

Another way to export all of your data or a subset of your data in the data source is to create an extract (.hyper) file. An extract functions as an independent data set, which you can use to connect directly from Tableau. For more information, see **Extract Your Data** on page 1014.

Export the data source

After you connect to your data, you can export and save your data source as a Tableau data source (.tds) file. Saving the data source creates a shortcut to your remote data and allows you to avoid having to create a new connection to a specific data set each time. For more information, see **Save Data Sources** on page 1075.

Export data used in the view

After you create a view, you can also export just the data used to generate that view.

The fields that are exported come from the fields on the shelves of the sheet. However, fields that function as external filters, in other words, the fields that appear only on the **Filters** shelf, are not included in the export. If you want to include other fields with the exported data without changing the baseline view, you can place those fields on the **Detail** shelf.

The various methods for exporting the data used to generate the view is listed below.

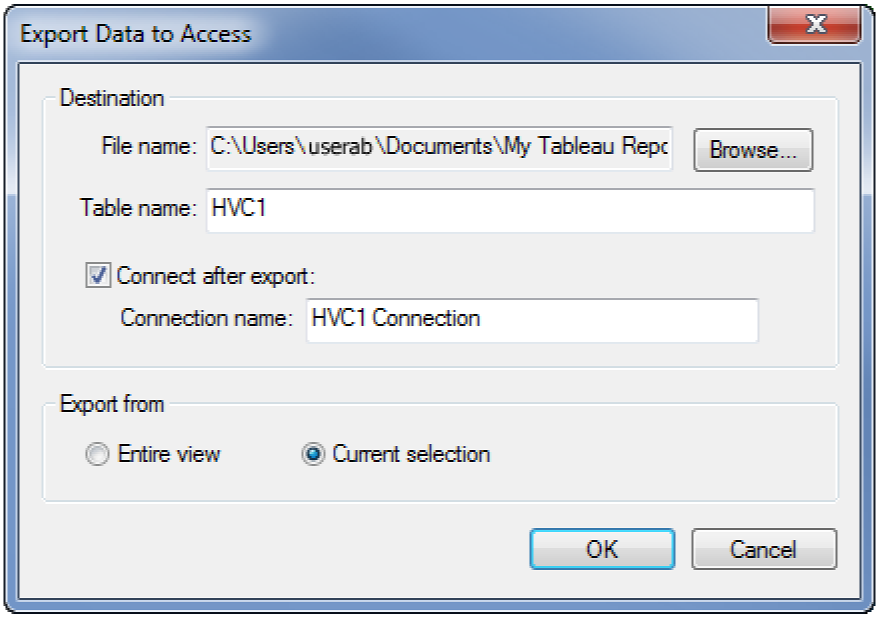
- **Export data in the view to Microsoft Access or .csv** below
- **Export crosstab of data in the view to Excel** on the facing page
- **Copy data in the view to clipboard** on the facing page
- **Copy crosstab of data in the view to clipboard** on page 3038

Export data in the view to Microsoft Access or .csv

Export the data that is used to generate the view as an Access database (Windows only) or .csv file (Mac only).

1. In Tableau Desktop, select **Worksheet** > **Export** > **Data**.
2. Select a location and type a name for your Access database or .csv file.
3. Click **Save**.

   If you're on Windows, the Export Data to Access dialog box displays to give you the option to immediately use the new Access database and continue working in Access without interrupting your work flow.



Export crosstab of data in the view to Excel

You can export directly to Excel the data used to generate the view formatted as a crosstab. When you export your view as a crosstab, Tableau automatically opens the Excel application and pastes a crosstab version of the current view into a new Excel workbook.

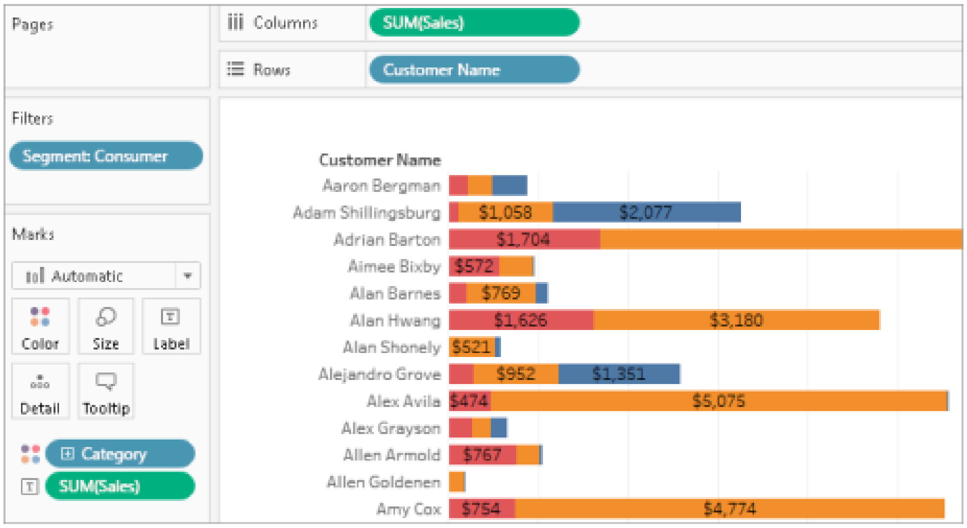
Although this option provides a direct method for exporting your data to another application, performance of the export can be affected because it is simultaneously copying and formatting the data. If the view you are exporting contains a lot of data, a dialog box opens asking whether you want to export the formatting. In this case, if you choose to exclude the formatting from the export, performance of the export might improve.

- In Tableau Desktop: select **Worksheet** > **Export** > **Crosstab to Excel**.
  If you're using a Mac, this option opens a dialog box where you can save the file. You must then manually open the file in Excel.
- In Tableau Server or Tableau Online, open a view or dashboard and select **Download** > **Crosstab**.
  Select which sheets from the workbook to export data from.

Copy data in the view to clipboard

Copy the data used to generate the view so that you can paste it into another application.

1. Create a view.



2. Select **Worksheet** > **Copy** > **Data**.

3. Open another application, such as Word, and paste the data into the document.

   In this example, the fields placed on the Columns, Rows, and Color shelves are copied into the document. However, the **Customer Segment** field is not copied because it is an external filter because it appears only on the Filters shelf.

Category	Customer Name	Sales
Furniture	Aaron Bergman $391	
Furniture	Adam Shillingsburg	$2,077
Furniture	Adrian Barton $1,280	
Furniture	Aimee Bixby $16	
Office Supplies	Aaron Bergman $274	
Office Supplies	Adam Shillingsburg	$1,058

Copy crosstab of data in the view to clipboard

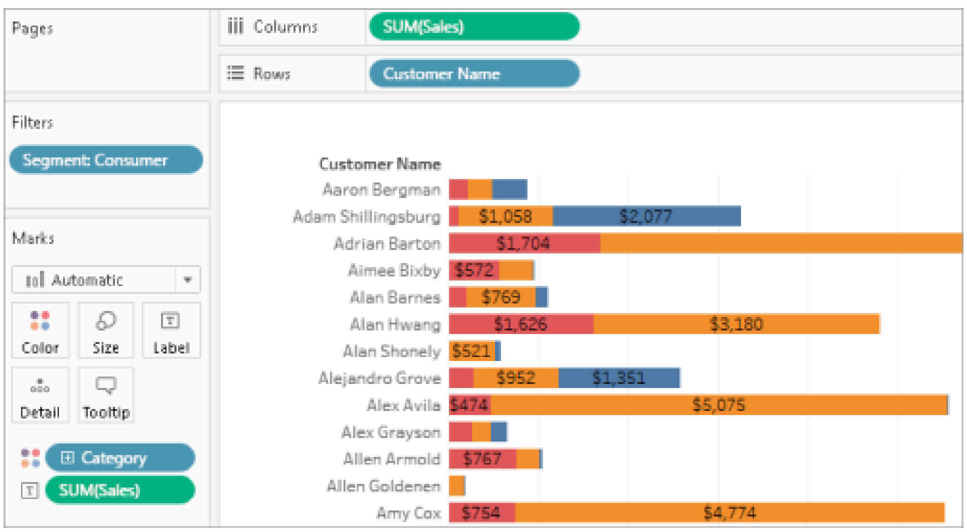
You can copy a crosstab version of a view so that you can paste or transfer the data into another application. The pasted data always appears as a crosstab, even if the initial view of the data in Tableau did not use a crosstab format.

Copying a crosstab is restricted by some general conditions:

- You must copy all records in the view. You cannot copy a subset of records.
- This option is valid for aggregated views only. It cannot be used on disaggregated views of data because a crosstab is by definition an aggregated view of data. This means the **Aggregate Measures** option on the Analysis menu must be selected in order for copying a crosstab to work properly.
- You cannot copy a crosstab if the view contains continuous dimensions such as continuous dates and times.
- Other restrictions may apply depending on the data in your view.

After the general conditions are met, copy the crosstab.

1. Create a view.



2. Select **Worksheet** > **Copy** > **Crosstab**.

3. Open another application, such as Excel, and paste the crosstab.

	A	B	C	D	E
1		Category	Category	Category	
2	Customer Name	Furniture	Office Sup	Technology	
3	Aaron Bergman	$391	$274	$222	
4	Adam Shillingsburg	$2,077	$1,058	$120	
5	Adrian Barton	$1,280	$11,489	$1,704	
6	Aimee Bixby	$16	$379	$572	
7	Alan Barnes	$131	$769	$213	

Copying Information Between Workbooks

Individual sheets, dashboards, and stories can be copied and pasted between workbooks in Tableau Desktop(although not in web authoring). This feature allows you to easily copy an analysis or port a data source into another workbook. It is also possible to import an entire workbook into your current workbook.

What gets copied or saved with selected sheets

When you copy, save, or export selected sheets, the data source or sources that are used on that sheet are also copied. This includes any calculations, parameters, sets, etc. Custom shapes and colors are also included. If the sheet being copied is a dashboard or story, all of the sheets that are used on that dashboard are also copied, whether they are hidden or not.

Custom colors and shapes will be available in the copied or imported sheet but will not be added to your Tableau Repository. If you would like these to persist for other workbooks, refer to **Create Custom Color Palettes** on page 2969 and Use custom shapes.

If you are trying to copy formatting between sheets, see Copy and paste worksheet formatting.

How Tableau handles duplicate items

When you paste or import sheets from a different workbook, some items might already exist in the destination workbook, or some items might have the same name in both places. If Tableau encounters an exact duplicate item in the Data pane, such as a calculation, it does not paste or import that item into the destination workbook. However, if an item in the Data pane has the same name but is defined differently, Tableau imports and renames it.

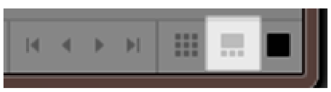
Tableau also pastes or imports sheets and data sources with duplicate names—whether only the name is identical, or their names and contents are identical—and it renames the newer copy.

Copy and paste sheets between workbooks

Copying and pasting sheets is a quick way to combine information from different workbooks or create a new workbook. It is also useful workaround to replace a data source for a single sheet rather than all sheets that use the data source. You can copy one or more sheets from the tabbed sheet view, filmstrip view or sheet sorter view. Use Shift + click or Ctrl + click to select multiple sheets.

To copy and paste a worksheet using the filmstrip view, do the following:

1. Open a workbook and click the **Filmstrip** button in the status bar.



2. Select the thumbnails of the sheets you want to copy, then right-click (Control-click on Mac) and select **Copy**.

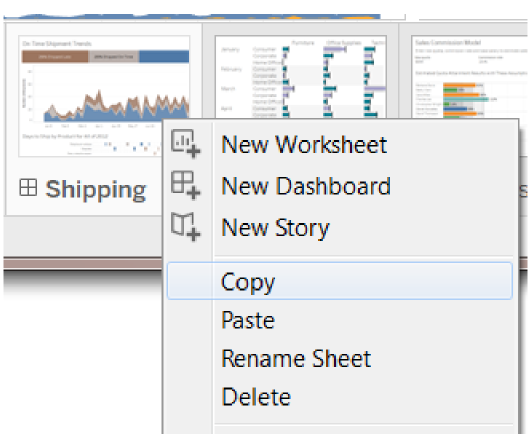


Tableau copies the information in the file format (.twb or .twbx) of the workbook.

3. Open the destination workbook, or create a new workbook. Right-click (Control-click on Mac) on the tab for any sheet, and select **Paste**.

   Pasted sheets are placed after existing worksheets, dashboards, and stories.

> **Note:** The **Paste** option is not available when the active sheet is a story.

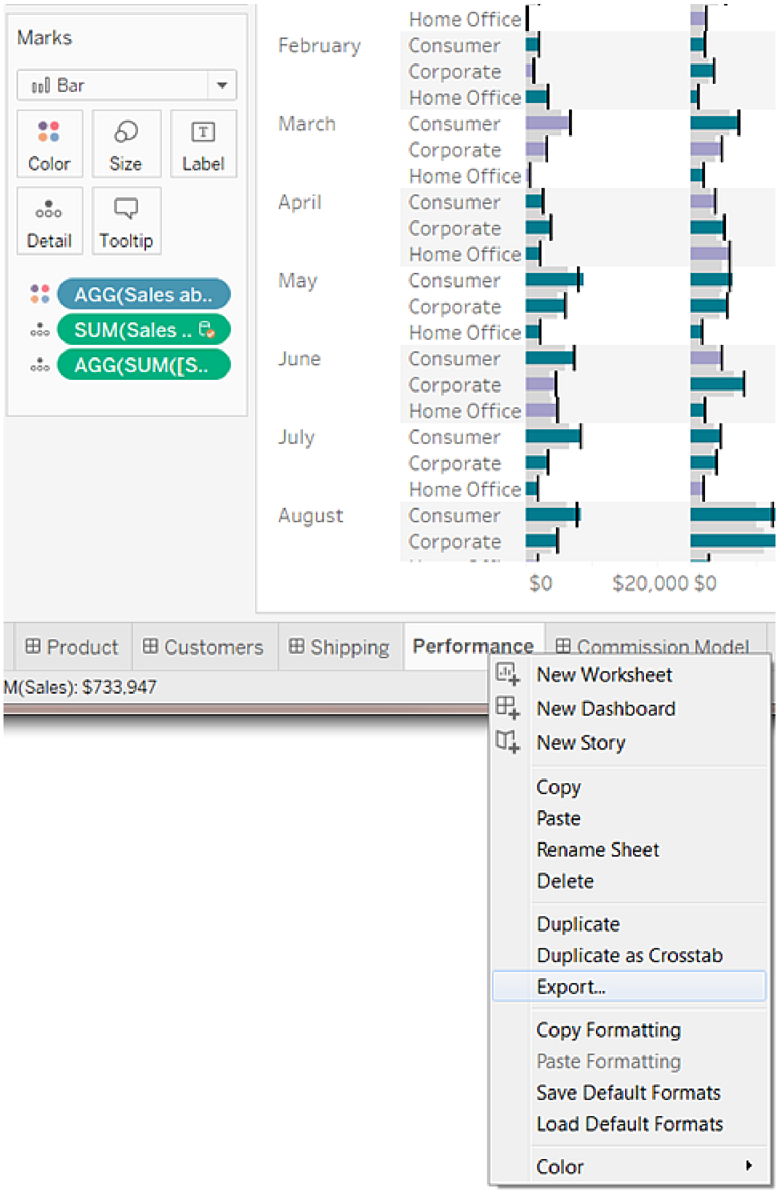
4. Save the changes.

Export and import sheets between workbooks

If you want to extract a subset of information from a larger workbook to maintain as a standalone file, you can export or save selected sheets to a new workbook. You can then import that workbook into an existing one to incorporate its sheets and other objects into the existing workbook.

> **Note:** These steps describe how to share your work among Tableau workbooks. You can also export views to use outside of Tableau. For information, see **Export Views from Tableau Desktop to Another Application** on page 3032.

1. Open the workbook that contains the sheets you want to export to a new file.
2. Using the tabbed worksheet, filmstrip view, or sheet sorter view, right-click (Control-click on Mac) the sheet tab or thumbnail view, and then select **Export** to export a single sheet. Use Shift + click or Ctrl + click to select multiple sheets.
3. In the Save As dialog box, specify the file format you want to save (.twb or .twbx), select the location for the new workbook file, give it a name, and then click **Save**.

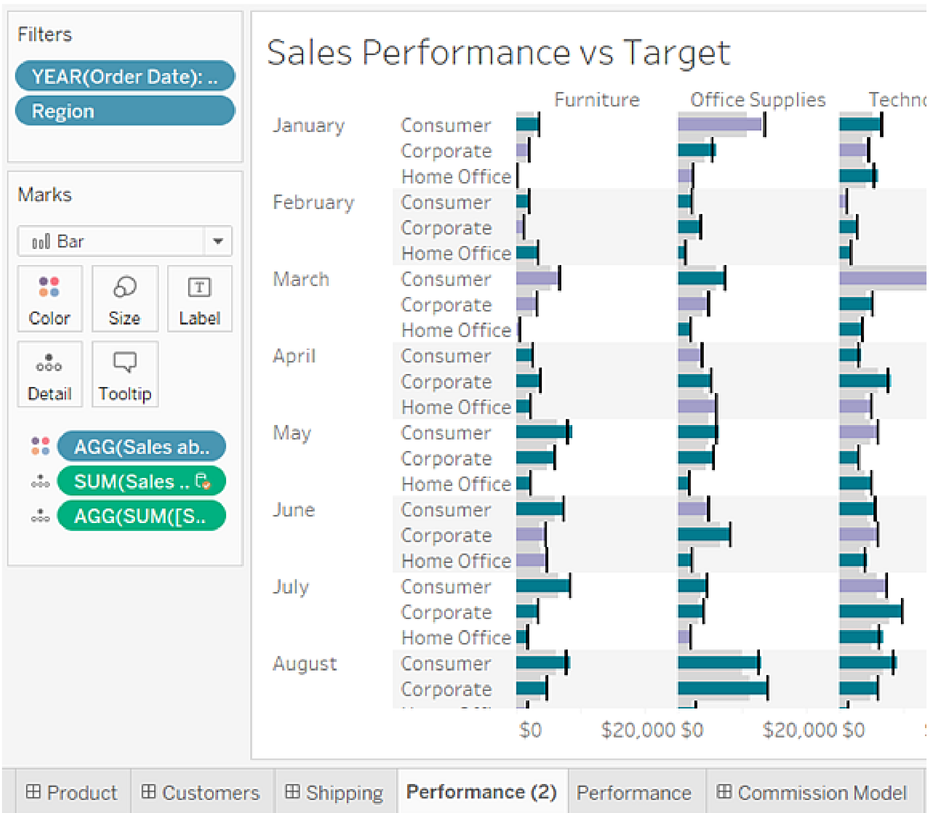


Import an entire Tableau workbook

After you save or export selected sheets to a new workbook (.twb) file, you can import the information into another workbook.

1. With the existing workbook open, select **File** > **Import Workbook**.
2. Select the workbook that contains the sheets you saved from another workbook, and click **Open**.

The following image shows the result of importing a workbook that contains a sheet with the same name as a sheet in the existing workbook. Tableau adds a number after the name of the imported sheet.



Print Views from Tableau Desktop

Before printing, specify how you want the printed page to look using the Page Setup dialog box. Then print to a printer or PDF.

Set up the page

You can apply different page setup options for each worksheet in a workbook, For example, worksheets can print with titles showing or hidden, with unique page orientation, and more.

To begin, select **File** > **Page Setup**.

General settings (available for individual sheets, not dashboards)

- **Show** - Show or hide the title, view, caption, color legend, shape legend, size legend, and map legend.
- **Headers and Breaks** - Control the appearance of these table elements.
  - Repeat headers and legends on each page - adds table row and column headers at the top of each printed page when a view breaks across several pages.
  - Break pages on pane boundaries - prevents page breaks in the middle of a table cell.
- **Pages Shelf** - If the view uses the Pages Shelf, specify whether to print the current page or all pages.

Layout settings

- **Legend Layout** - If you include one or more legends, select how you want the legends to appear on the printed page.
- **Margins** - Specify top, bottom, left, and right margins by typing values into the text boxes.
- **Centering** - Optionally, select whether to center the view horizontally or vertically—or both—on the page.

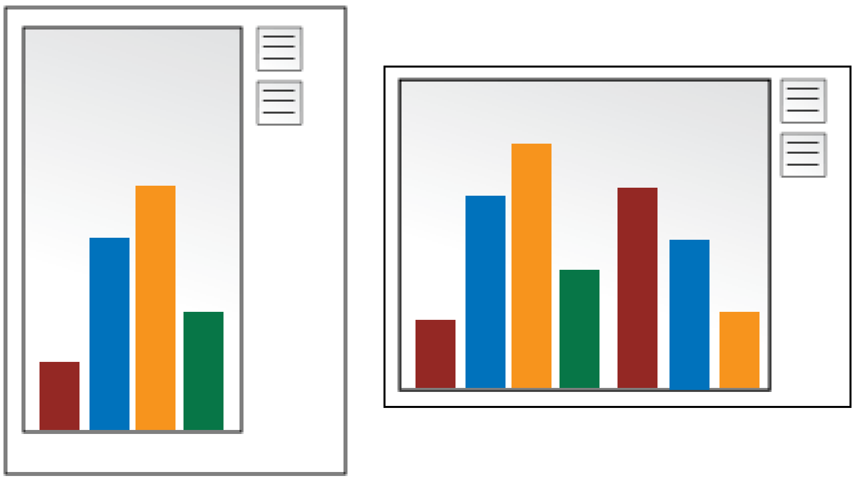
Print Scaling settings

These settings affect only printed documents, not exported images or PDFs. However, the page orientation settings are used as the default when you publish the workbook to Tableau Online or Tableau Server.

- **Print Scaling** - Scale a view to fit within a single page or print across multiple pages. Select from the following options:
  - Automatic – Scales the view automatically based on the paper size.
  - Scale to – Scales the view to the specified percentage of its original size.
  - Fit to – Scales the view to fit within the specified area. Select the number of printed pages across and down. For example, if you have a really wide view that is not very tall, you can specify three pages across by one page down.
- **Page Orientation** - Specify how you want the view oriented on the printed page. Select from the following options:

- Use Printer Setting – Use the page orientation that is already specified by the printer.
- Portrait – Presents the view so that it is oriented vertically on the printed page.
- Landscape – Presents the view so that it is oriented horizontally on the printed page.

The following diagram shows the difference between portrait and landscape page orientations.



Print a view

After you have configured the Page Setup settings, select **File** > **Print**. The following options in the Print dialog box are unique to Tableau.

Show Selections

When this option is selected any selections you've made in the views will be maintained while printing.

Change the Print Range

When you print from a workbook with multiple worksheets, each worksheet represents one or more printed pages, depending on the page setup.

Select from the following print ranges:

- **Entire Workbook** - Prints all the worksheets in the workbook.
- **Active Sheet** - Prints only the sheet currently displayed in the workbook.
- **Selected Sheets** - Prints the selected sheets.

You can select multiple worksheets in a workbook by holding down the CTRL or Shift keys (or the ⌘ key on a Mac) while clicking the worksheet tabs that you want to select.

Print to PDF

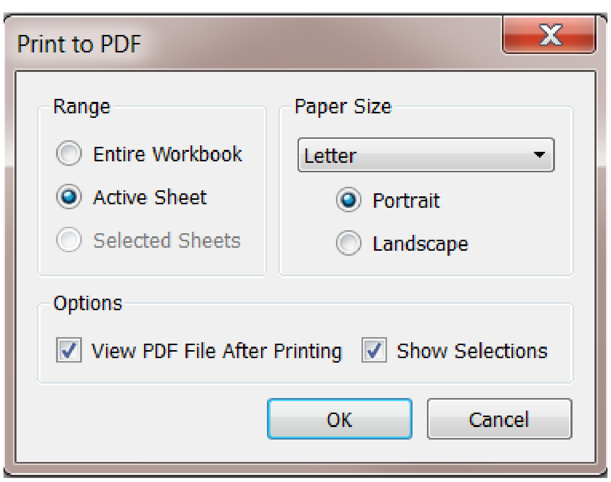
Note: If you're using Tableau Online or Tableau Server, instead see **Download Views and Workbooks** on page 3328 and **Link to a PNG, PDF, or CSV of a View** on page 3327.

In Tableau Desktop, you can save views as PDF files rather than printing them as hard copies. You do not need to have Adobe Acrobat installed on your computer.

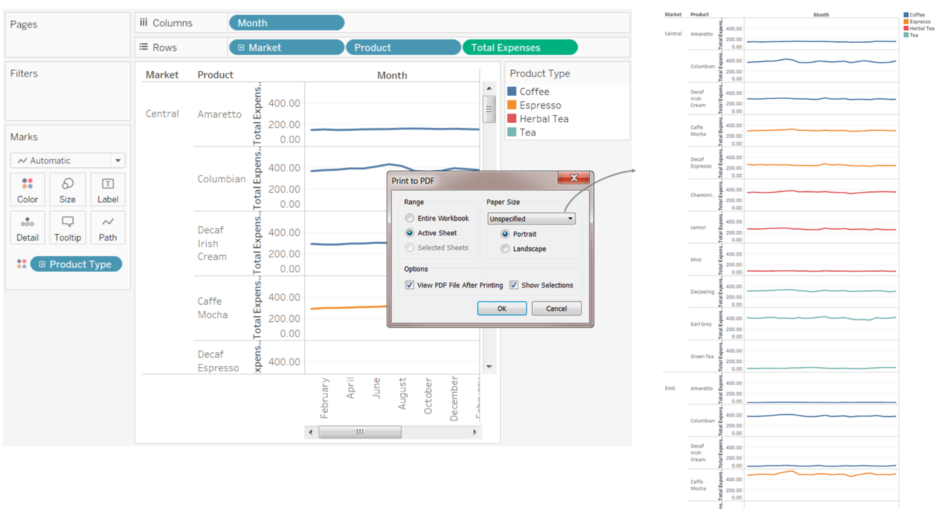
When you print an individual sheet to PDF, filters in the view are not included. To show filters, create a dashboard containing the sheet and export the dashboard to PDF.

Print to PDF using a Windows computer

1. Specify page setup options for each sheet in your workbook.
2. Select **File** > **Print to PDF**.



3. In the Print to PDF dialog box, select the print Range:
   - Entire Workbook - Publishes all the sheets in the workbook.
   - Active Sheet - Publishes only the sheet currently displayed in the workbook.
   - Selected Sheets - Publishes the selected sheets. To select multiple sheets in a Tableau workbook, hold down the Ctrl key as you select sheet tabs along the bottom of the Tableau workbook.
4. Select a **Paper Size**. If you select Unspecified, the paper size will expand to the necessary size to fit the entire view on a single page.



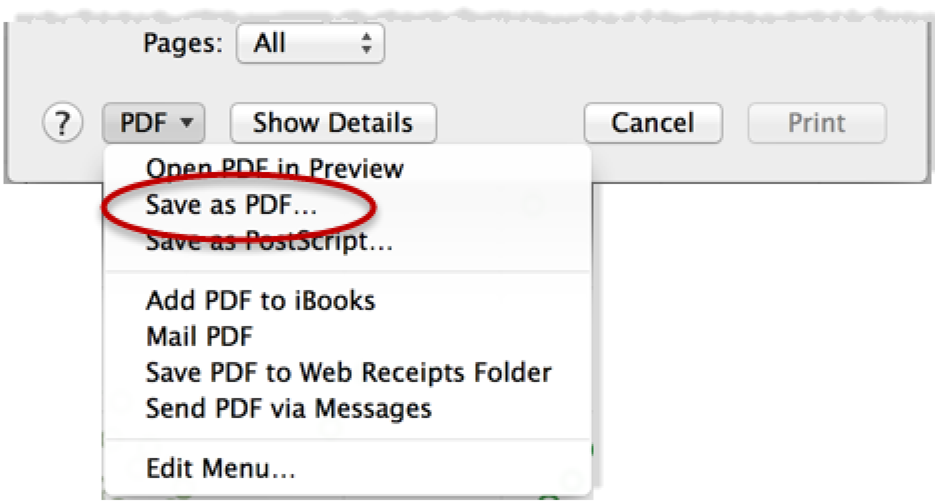
5. Select **View PDF File After Printing** if you want to automatically open the PDF after creating it. This option is only available if you have Adobe Acrobat Reader or Adobe Acrobat installed on your computer.
6. Select whether to **Show Selections**. When this option is selected the selections in the views are maintained in the PDF.
7. Click **OK** and specify where you want to save the PDF. Then click **Save**.

Print to PDF using a Mac computer

1. Specify page setup options for each sheet in your workbook.
2. Select **File** > **Print**.
3. In the Print dialog box, click **Show Details** to select a print range:

- Entire Workbook - Publishes all the sheets in the workbook.
- Active Sheet - Publishes only the sheet currently displayed in the workbook.
- Selected Sheets - Publishes the selected sheets. To select multiple sheets in a Tableau workbook, hold down the Command key as you select sheet tabs along the bottom of the Tableau workbook.

4. Click **PDF** > **Save as PDF**.



5. Specify where you want to save the PDF, then click **Save**.

Publish Data Sources and Workbooks

Suppose you create a view that exposes a new range of questions in the data you're using, and you want to share this analysis with other people using this data. Or maybe you are your team's Data Steward, in charge of building the data models approved for use by analysts, and meeting your organization's requirements for security, compliance, performance, and so on.

You can share your work with the rest of your team by publishing it to Tableau Server or Tableau Online. After it's published, you and your team can access it through your web browser or the Tableau mobile app. Publishing data sources can also help you to centralize data management.

In other resources

For the steps on how to publish, see the following topics:

- **Publish a Data Source** on page 3071
- **Comprehensive Steps to Publish a Workbook** on page 3058
- **Upload Workbooks to a Tableau Site** on page 3146

Note: If you do not have Tableau Online or Server, you can share analysis on Tableau Public, a free cloud service. As the name suggests, views published to Tableau Public are publicly accessible. For more information, see public.tableau.com.

Why publish

You can publish data sources and workbooks when you want to widen the audience for your data analysis within your organization. By publishing you can begin to do the following:

- Collaborate and share with others

  Allow people in your organization to view, interact with, download, subscribe to, share, edit, and save published views, even if they do not use Tableau Desktop. Incorporate views into blog posts or websites.

- Centralize data and database driver management

  Create and publish data models that everyone can use. Centralized data management allows for sharing a single source for your Tableau data. All workbooks connected to the published data reflect updates to it.

  In addition, when you publish and connect to data on the server, people connecting to the data from Tableau Desktop do not need to install and maintain database drivers on their own computers.

- Support mobility

  Access your data from a different computer or location, through a web browser or the Tableau Mobile iOS app. Sign in to your organization's Tableau Server from a private network offsite.

What you can publish

Content types you can publish include:

- **Data sources:** You can publish data sources that others can use to build new workbooks. A data source can contain a direct (or live) connection to your database or an extract you can refresh on a schedule.

  For information, see **Best Practices for Published Data Sources** on page 3065.

- **Workbooks:** Workbooks contain your views, dashboards, and stories, and data connection. You can include local resources, such as background images and custom geocoding, if they reside in a location that the server or other Tableau users cannot access.

Who can publish

To publish to Tableau Server or Tableau Online, your server or site administrator must grant you the following capabilities:

- A *site role* of **Creator** (formerly Publisher) on the site you're publishing to.
- **View** and **Save** capabilities set to **Allowed** on the project into which you publish.

If you use Tableau Desktop and are not sure whether you can publish to a server, or you are having trouble publishing, see your Tableau administrator. If you’re an administrator, see

Content Access and Ownership in the Tableau Server help (or the Tableau Online version) for more information about site roles and permissions.

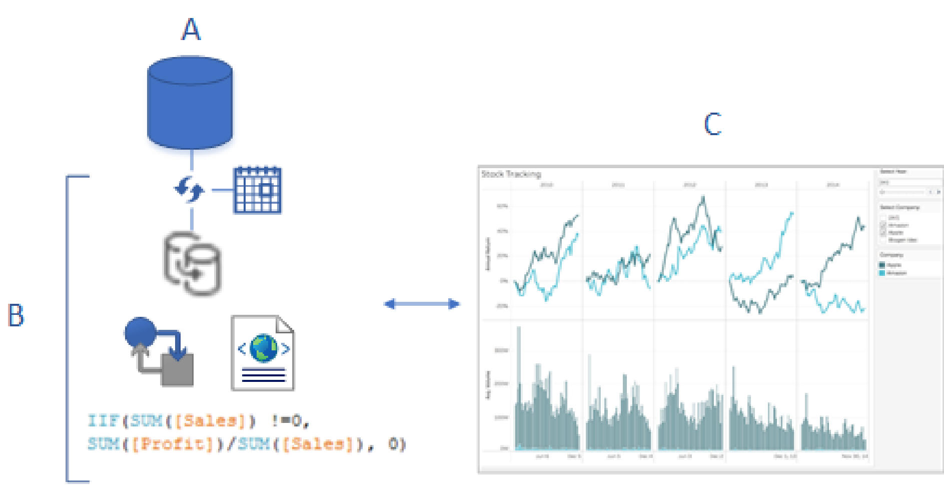
Prepare for Publishing a Workbook

When you publish a workbook, you need to make a decisions that determine how others will access the views and the data behind them. You also need to take into account the experience you and others will have of interacting with the views on the server. Before you publish your workbook, use the information in this topic to help you make these decisions and improve the workbook's performance.

> **Note:** If you have a Tableau site administrator, check with them on whether your organization already has publishing guidelines that have already answered these questions.

What makes up a typical published workbook

During the publishing process, you select settings that determine how the pieces that make up the workbook interact with each other. The following image shows a simplified overview of a workbook's structure.



A. The underlying (original) data that you connected to when you created the workbook. When you publish, you specify whether to include credentials for accessing this data, or whether to require users to enter credentials to access it.

B. The Tableau data source. This is where the action is. It contains the XML metadata that describes how to access the underlying data (A), field customizations or calculations you made in Tableau, when to refresh the extract if there is one.

   In this image, the data is embedded in the workbook, and it contains an extract connection and refresh schedule, some calculations, and so on.

C. A view showing data from B, which you want to make available for your colleagues to edit or interact with on the server. When you publish, you can select the sheets—which include views, dashboards, and stories—you want to share.

Decide how to access the data and keep it up-to-date

During the publishing steps, you need to answer the following questions about the data connection. For help with the answers, see **Best Practices for Published Data Sources** on page 3065.

- Will you publish with a live connection to the data, or will you create an extract?

  If you publish live connections to Tableau Online, see **Authorize Access to Cloud Data Published to Tableau Online** on page 3082.

- Do you want to publish the workbook's connections as separate, standalone Tableau data sources (and then connect the workbook to the published data); or embed the data into the workbook?

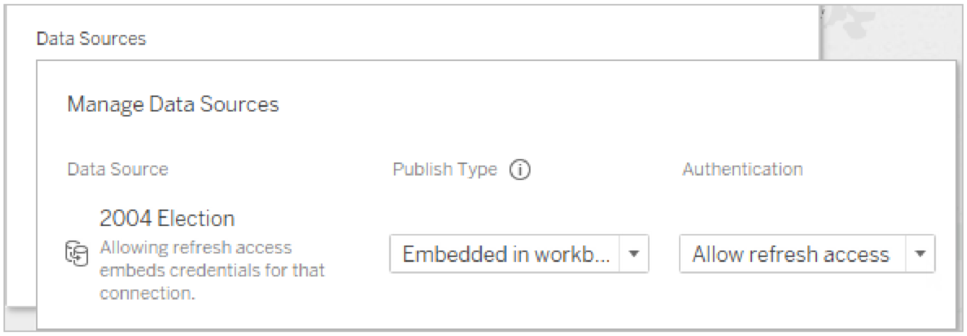
  Tip: If the workbook already connects to a Tableau data source, you keep the existing connection. That means you've just answered this question.

- Do you want the workbook to show the data when other users open the workbook, or do you want to require them to provide database credentials? (Either way, they must also have access to the site and project you publish to.)

  This can be complicated depending on the data the workbook connects to. In many cases, the simple path to data described below works great. If it doesn't work for you (or you're not sure), see **Set Credentials for Accessing Your Published Data** on page 3079.

The simple path to data authorization

When you go through the publishing steps, you specify how your workbook users will access the data the workbook connects to. You do this in the **Data Sources** section of the Publish Workbook dialog box.



Depending on the connection type, the simple path might work for your environment. You would do one of the following:

For a connection to a Tableau data source: Keep the existing connection and embed the password.

For other data connections:

1. Before you open the Publish Workbook dialog box, create an extract and include only the data you need for the workbook.
2. In the Publish Workbook dialog box, embed the extract in the workbook and select **Allow refresh access**. The latter embeds the credentials in the connection.
3. After you specify remaining settings and click Publish, set up a refresh schedule.

For the complete steps, see **Comprehensive Steps to Publish a Workbook** on page 3058.

Scenarios where data authorization requires some extra effort

The simple path doesn't work best for every situation. Some reasons you might need to divert from it include:

- Instead of embedding data into workbooks, your existing policies mandate publishing data sources separately and connecting workbooks to the Tableau published data sources.

This requires a few more steps than embedding the data; however, where there is a choice between the two, publishing data separately and managing it on the server is considered a better practice.

- You added a user filter to enforce row-level security. In this scenario, publishing live connections is more common, and other steps are required to secure the filter.
- You're publishing to Tableau Server, and you use SAP HANA or Impala single sign-on, or your workbook connects to data that can use impersonation-based authorization (SQL Server or Kerberos-enabled data).
- Other reasons that have to do with your organization's data security or authorization policies, which your IT staff can help you with.

Even if whatever you decide turns out to be not the best path, you can easily correct it by republishing. In some cases, your site administrator can change the settings on the server directly.

Assess the workbook performance and ease of use

Keep performance and interaction best practices in mind as you build workbooks you want to publish. If changes take a long time to display while you're working in Tableau Desktop, they will take as long or longer to display on the server. Simple steps you can take that can have a big impact include limiting the number of marks you add to a view, limiting the number of views you add to a dashboard, and removing unused fields from your data source.

For additional tips, see the following topics:

- **Make Visualizations Faster** on page 3006
- **Design for Performance While You Build a View** on page 2999
- Designing Efficient Workbooks, starting with "Writing Efficient Workbooks" on page 6. This Tableau white paper requires that you sign in to the Tableau website using a free Tableau ID. This is the same ID you use for the Tableau community forums or for watching our training videos.

Simple Steps to Publish a Workbook

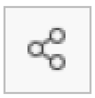
When you want to share a workbook with your colleagues, you can publish it to Tableau Server or Tableau Online with a few simple clicks. There, other people can view it, interact with it, and even edit it if their server permissions allow.

Before you publish your workbook, make sure you know the following:

- The name of the server and how you sign in to it. If your organization uses Tableau Online, you can click the Quick Connect link.
- Any publishing guidelines your Tableau administrator might have, such as the name of the project you should publish to.

Publish your workbook

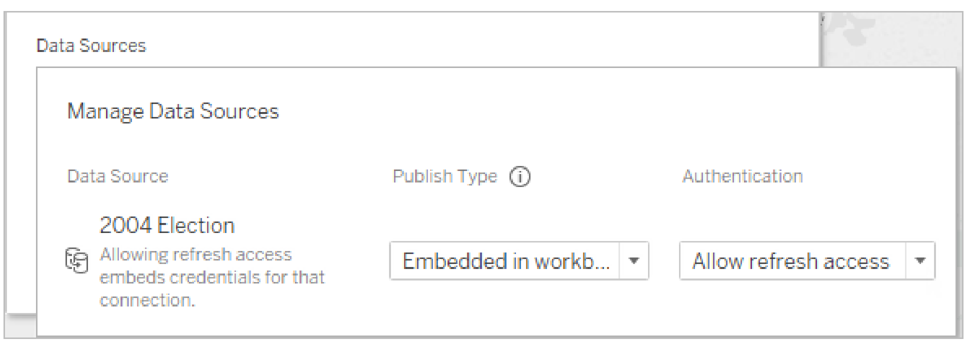
1. With the workbook open in Tableau Desktop, click the **Share** button in the toolbar.

   If you aren't already signed in to Tableau Server or Tableau Online, do so now. If you don't have a site yet, you can create one on Tableau Online.

2. In the Publish Workbook dialog box, select the project to publish to.
3. Name the workbook according to whether you're creating a new one or publishing over an existing one.
4. Under Data Sources, select **Edit**. For **Authentication**, select **Allow refresh access** or **Embed password**.

   For some data connections, only one authentication option appears. If **None** shows, leave it set to that.

4. Click **Publish**.

   If this is your first time publishing a workbook, test it on the server and work out any glitches before letting other users know the workbook is available.

For more details, see these topics:

- **Comprehensive Steps to Publish a Workbook** below
- **Publish a Data Source** on page 3071
- **Set Credentials for Accessing Your Published Data** on page 3079

Let others know the workbook is available on the server

1. Make sure your colleagues can access the site and project you published to.
2. Send them a direct link to your workbook.
3. After they sign in, the workbook opens for viewing, interacting, or editing.

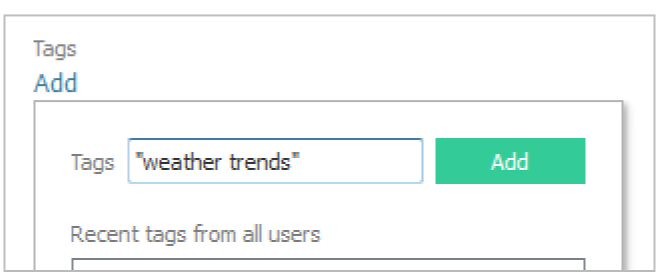
Comprehensive Steps to Publish a Workbook

1. In Tableau Desktop, open the workbook you want to publish.
2. Select **Server** > **Publish Workbook**.

   If the **Publish Workbook** option does not appear on the **Server** menu, make sure a worksheet or dashboard tab is active (not the Data Source tab).

   If necessary, sign in to a server. For Tableau Online, enter **https://online.tableau.com**. For more information, see **Sign in to Tableau Server or Tableau Online** on page 364.
3. In the **Publish Workbook** dialog box, select the project, enter a name for the workbook, and add search tags.

   Tags help users find related workbooks when they browse the server. Separate tags using a comma or space. To add a tag that contains a space, put the tag in quotation marks.



4. For **Permissions**, accept the default project settings.

   Generally, a site administrator manages permissions on the server. If you think your workbook is an exception, work with your administrator to determine the best course of action, and see **Set Permissions as You Publish a Data Source or Workbook** on page 3076.

5. For **Data Sources**, select **Edit** if you want to change whether the data is embedded in the workbook or published separately, or change how people authenticate with data sources.

   - If you're publishing an extract and want to set up a refresh schedule, you must select **Embed password** or **Allow refresh access**.
   - If your workbook connects to a Tableau data source, we recommend embedding the password. If you instead choose to prompt users, they'll need additional permissions on the data source.

   For more information, see **Set Credentials for Accessing Your Published Data** on page 3079.

6. Configure **Variable publishing options** below that are available for this workbook.
7. Click **Publish**.
8. (Optional) Set up a refresh schedule for each extract you published.

   The publishing workflow guides you through these steps. For some data types you publish to Tableau Online, the publishing process starts Tableau Bridge on your computer.

   For more information, see **Schedule Extract Refreshes as You Publish a Workbook** on page 3086.

Variable publishing options

The following options appear when they're appropriate for the workbook.

Show or hide sheets

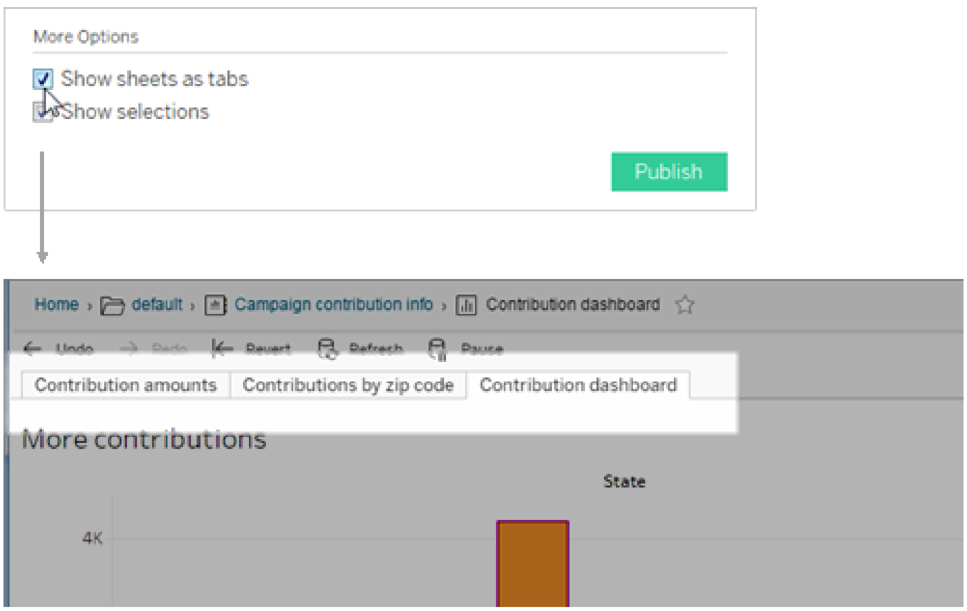
By default, Tableau Desktop publishes all sheets in a multiple-sheet workbook. In the **Sheets** section of the Publish Workbook dialog box, you can specify which sheets to include. Hiding sheets is useful when you want to publish a dashboard or story without showing the worksheets that were used to create it.

Important: Hiding sheets is not a security measure. Anyone who has the **Download/Web Save As** capability can access the hidden sheets. Other editing permissions can also allow access to hidden sheets. For more information, see Content Access and Ownership in Tableau Server Help.

Show sheets as tabs

If you select multiple sheets to show, you can specify how users navigate them.

- Select the **Show Sheets as Tabs** check box to provide tab-based navigation.



- Clear the check box to allow people to open only one view at a time.

Note that how sheet tabs are configured also impacts permissions. When a workbook shows sheets as tabs, workbook-level permission rules are applied to the sheets. When the sheets are *not* shown as tabs, any changes made to workbook permissions do not apply to the individual sheets (aka views). View-level permissions must be set independently. See the Set Content Permissions tab in the Permissions topic for more information.

Show selections

Select this to highlight a particular portion of the view when others open the workbook. Make your selections in Tableau Desktop before you start the publishing process.

Include external files

In general, if your views contain information that isn't available to the server or other users, you can use the **Include External Files** option when you publish. With this option, you can include an Excel, CSV, or other data source file local to your computer. However, the availability of this option depends on whether you are publishing to Tableau Server or Tableau Online.

For Tableau Server

If you're publishing to Tableau Server, and the workbook references data or images on a mapped drive, you can select the **Include External Files** when you publish. Alternatively, you can change the connection information so that the workbook references the UNC path to the data. For example, you could change **D:\datasource.xls** to **\\filesrv\datasource.xls**.

For Tableau Online

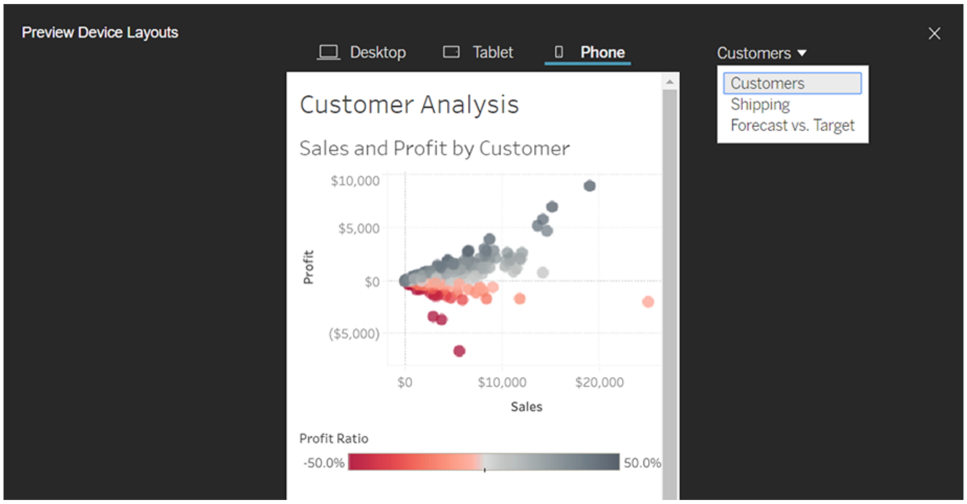
If you're publishing to Tableau Online and the workbook connects to on-premises data, like Excel or text, which Tableau Online cannot connect to directly, you will need to do *one* of the following:

- Check to see if all the connections (Supported Connectors) in the data source are supported by Tableau Bridge. If so, you can ignore the check box, but you must use Tableau Bridge to keep the data fresh. For more information, see Use Tableau Bridge to Keep Tableau Online Data Fresh.
- If one of the connections in the data source is not supported by Tableau Bridge, you'll need to select the "Include External Files" check box so that Tableau Online can refresh the data that Tableau can connect to directly. The file-based data cannot refresh and its data will remain static in this case.

Preview device layouts

To see how your designs appear on mobile devices, click **Preview device layouts** in the browser when publishing is complete. Then choose a device type above the viz, and a specific dashboard from the menu at right.

If you don't like a device preview, open the workbook in Tableau Desktop, adjust device layouts, and republish.

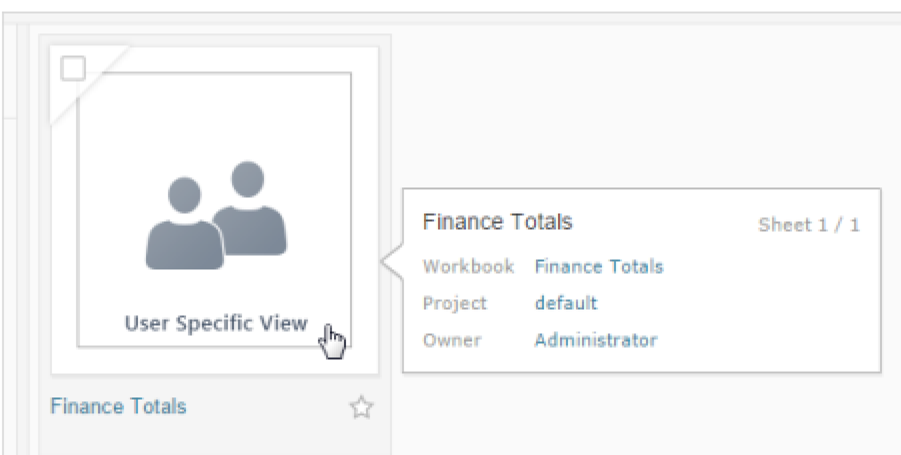


Choose how to generate thumbnails for workbooks with user filters

On the server, users can browse Tableau content with a thumbnail view. These thumbnail images are based on workbook sheets. If your workbook contains user filters, you can specify which user's filter to use when creating the thumbnails. For example, if you want the thumbnail image to show all regions of a sales forecast, you can generate thumbnails based on a user who is allowed to see all regions.

In the following scenarios, a generic image appears in place of the view thumbnail.

- The user you select does not have permission to see the data.
- The data is from a Tableau Server data source that utilizes data source filters, user calculations, impersonation, or other user references.



To learn more about user filters, see **Restrict Access at the Data Row Level** on page 3088.

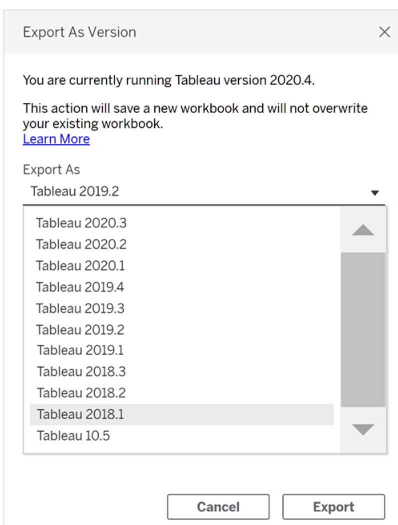
Make Workbooks Compatible with Older Versions

You can downgrade workbooks to share them with people using older versions of Tableau products.

Important: When you downgrade to an older version, any features or functionality that aren't available in that version are removed for compatibility. Additionally, any calculated fields created in the newer version that use functions unavailable in the older version will have to be manually corrected or removed upon downgrade.

Export to an earlier version from Tableau Desktop

1. In Tableau Desktop select **File** > **Export As Version** and select the version that you want to downgrade to. (The earliest version you can export is Tableau Desktop 10.5.)

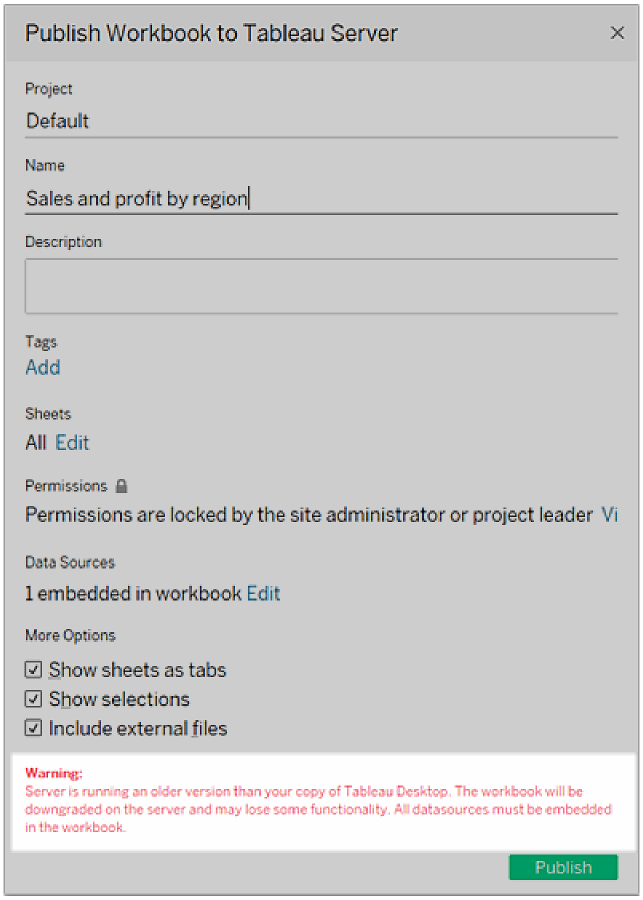


2. Save the workbook to your My Tableau Repository or a selected folder.

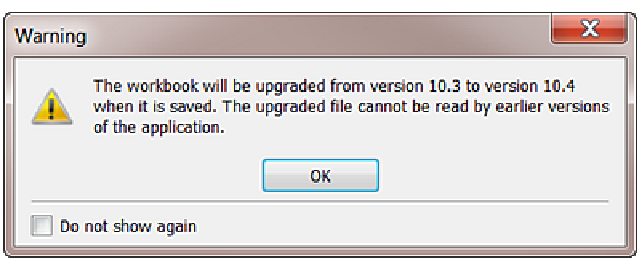
Downgrade a workbook when publishing to Tableau Server

If you're using a newer version of Tableau Desktop with an older version of Tableau Server, you can downgrade to the older Tableau Server version when you choose **Server** > **Publish**

Workbook. A warning message appears indicating that workbook will be downgraded when published:



If you later open the downgraded workbook in the newer version of Tableau Desktop, you'll see a message like the one shown below. You may then need to add back the features that were removed when the workbook was downgraded.

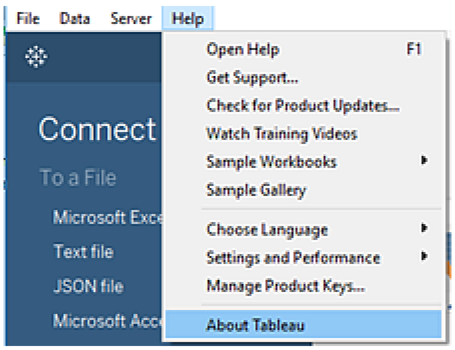


Note: Data sources are not downgraded during publishing.

Download an older version from Tableau Online or Tableau Server

If you're using Tableau Online or Tableau Server, you can download a workbook to be opened in an older version of Tableau Desktop.

1. To identify which version of Tableau Desktop you're using, select **Help** > **About Tableau** from the top menu.



2. In Tableau Server or Tableau Online, open the view you'd like to download.
3. Click the **Download** button, and then click **Tableau Workbook**.
4. Select the version that you want to export to. (The earliest version you can download is Tableau Desktop 10.5.)

For more information, see **Download Views and Workbooks** on page 3328.

Additional compatibility resources

In the Tableau Deployment Guide, check out Make Workbooks Compatible Between Versions to learn more about how different versions of Tableau products interact.

Best Practices for Published Data Sources

Publishing data sources to Tableau Online or Tableau Server is integral to maintaining a single source for your data. Publishing also enables sharing data among colleagues; including those who don’t use Tableau Desktop, but have permission to edit workbooks in the web editing environment.

Updates to a published data source flow to all connected workbooks, whether the workbooks themselves are published or not.

What makes up a published data source

A Tableau data source consists of the following:

The data connection information that describes what data you want to bring in to Tableau for analysis. When you connect to the data in Tableau Desktop, you can create joins, including joins between tables from different data types. You can rename fields on the Data Source page to be more descriptive for the people who work with your published data source.

An extract, if you decide to create one. Guidelines for when to create an extract are included below, as well as in the additional resources.

Information about how to access or refresh the data. The connection also includes access information. Examples of this type of information include:

- The path to an original Excel file.
- Embedded credentials or OAuth access tokens for accessing the data directly.
- Alternatively, no credentials, so that users are prompted to enter them when they want to access the data (whether it’s to view a workbook that connects to it, or to connect a new workbook to it).

For more information, see **Set Credentials for Accessing Your Published Data** on page 3079.

Customization and cleanup that helps you and others use the data source efficiently. When you’re working with your view, you can add calculations, sets, groups, bins, and parameters; define any custom field formatting; hide unused fields; and so on.

All of these refinements become part of the metadata contained in the data source that you publish and maintain.

Preparing a data source for publishing

When you publish a data source, consider these best practices:

- Create the connection for the information you want to bring into Tableau and do any customization and cleanup that will help you and others use the data source efficiently.
- If appropriate, create an extract of the data you want to publish. For more information,

see the following section, **When to use an extract** below.

- Develop a data source naming convention.

  After publishing a data source you cannot rename it directly. Instead, you need to publish a new copy with the new name, and then update all workbook connections. A well-considered naming convention can also help other users of the data deduce which data source to connect to.

- Consider designating the following roles among your Tableau users:
  - A data steward (or team) who creates and publishes the data sources for the Tableau community, which meet your organization's data requirements.
  - A site administrator who manages published content, extract refreshes, and permissions on the server you publish to (Tableau Server or Tableau Online).

  Central management helps to avoid data source proliferation. Authors who connect to managed data can be confident that the answers they find in it reflect the current state of the business.

When to use an extract

Under the following conditions you might be required or choose to publish an extract instead of connecting live.

Publishing data to Tableau Online that it cannot reach directly

Tableau Online in the cloud cannot reach data sources that you maintain on your local network. Depending on the connection, you might be required to publish an extract and set up a refresh schedule using Tableau Bridge.

Some cloud-hosted data sources always require extracts. These include Google Analytics, Salesforce.com, Oracle, OData, and some ODBC data sources. You can set up refresh schedules for some of these data sources directly on Tableau Online; for others you use Tableau Bridge.

Web data connector data sources always require extracts. If you connect to the data source using standard user name and password authentication, you can refresh it using Tableau Bridge. If you connect to the WDC data source using OAuth authentication, you will need to use an alternative method to refresh it.

For more about how Tableau Bridge supports both extract and live connections to data Tableau Online cannot reach directly, see Use Tableau Bridge to Expand Data Freshness Options in the Tableau Online Help.

Improving performance

Even if the server supports live connections to your data, an extract might make more sense. For example, if the database is large or the connection slow, you can extract a subset that includes only the pertinent information. The extract can be easier and faster to work with than connecting live.

In cases where you can use a live connection or an extract that you refresh on a schedule, you might want to experiment with both options to see which works best for you.

Enabling functionality the data source does not inherently support

For example, suppose you want to use the Median function with SQL Server data.

To learn more about creating data extracts, see **Extract Your Data** on page 1014.

Publishing data separately or embedded in workbooks

You can publish data sources as standalone resources that workbooks connect to, or you can publish workbooks with the data sources included within them.

When you publish a workbook, if any connection specifies anything other than a Tableau data source published to the same project, the data is published as part of the workbook (sometimes referred to as *embedded* in the workbook).

When data is embedded in a workbook:

- Access to the data source is limited to the workbook in which you published it. Neither you nor other users can connect to that data from another workbook.
- You can set up extract refresh schedules as you do for data sources that you publish separately.

Each way of publishing has its advantages. The table below shows a few common points of comparison. It is not a comprehensive list, and these are generalizations. How these and other factors apply to you are specific to your environment.

Published separately	Embedded in workbook
Publishing data sources is a step toward centralizing data management. You can create policies geared toward minimizing data source proliferation and helping people find the right data for the work they do.	Each embedded data source has a separate connection to the data. Each has the potential to show something different than the other at any given time (and data source proliferation is common).
Meant to be shared; becomes available for other Tableau users to connect to.	Data is available only inside the workbook; it is not available for other Tableau Desktop users to connect to.
Without content management and self-service guidelines, seeing a long list of data sources to connect to can be confusing to users who rely on the data to do their work, and is more difficult to manage on the server.	Users create their own connections, and they know exactly what data they're getting.
Someone who changes a shared data source might be uncertain or unaware of the effects that those changes have on connected workbooks.	Changing the data requires opening the workbook, where you can see the result of the change.
Even if effects of data source changes on connected workbooks are planned, updating those connected workbooks is cumbersome.	Same as above; however, if multiple workbooks use similar data and need to be updated, it might be worth connecting to a published data source instead.
Extracts can be refreshed on a schedule. You set up one refresh schedule for the extract, and all workbooks that connect to it always show the most current data.	Embedded extracts that aren't refreshed can be useful for showing snapshots in time. If you want to keep the data fresh, each workbook must have its own refresh schedule.

Published separately	Embedded in workbook
Generally helps you to optimize performance on the server or site.	Performance might be affected when the server contains multiple workbooks that connect to the same original data, and each workbook has its own refresh schedule.

Keeping extracts up-to-date

When you publish a data source with an extract, you can refresh it on a schedule. The way you schedule refreshes depends on the data source type and whether you're publishing to Tableau Server or Tableau Online.

For more information, see the following topics:

- Keep Data Fresh on Tableau Online
- Keep Data Fresh on Tableau Server

Additional resources

- Data Server—Training video by Tableau, with a helpful overview of data sources and publishing.
- Understanding Tableau Data Extracts

  A version-agnostic, three-part series by Gordon Rose on the Tableau blog. It includes an in-depth look at the extract's file structure, guidelines for when to use extracts, and best practices.
- O Extract, Where Art Thou? and TDE or Live? When to Use Tableau Data Extracts (or not)

  Posts by Tableau Zen Master Jonathan Drummey on his blog Drawing with Numbers. Includes tips on extracts, explains the different file types, describes different publishing scenarios. (Read the comments, too.)
- Tableau Extracts–What / Why / How etc

  From the blog maintained by The Information Lab, a Tableau Gold Partner.

Disclaimer: Although we make every effort to ensure these links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

Publish a Data Source

When you are ready to make a data source available to other users, you can publish it to Tableau Server or Tableau Online. If the data source is in a workbook that you published to Tableau Server or Tableau Online, you can make it available by saving it, provided it’s an embedded Excel or text file. For details, see **Publish a Data Source on the Web** on page 3146.

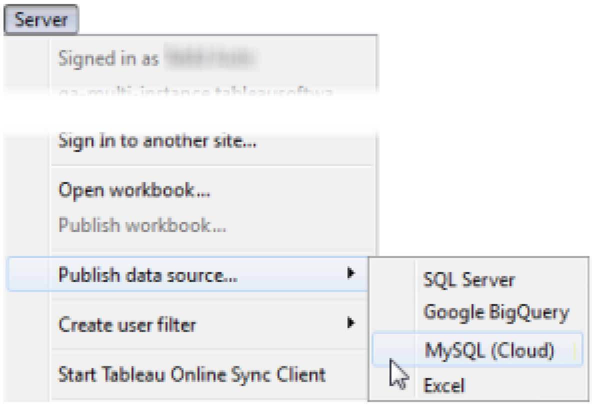
Note: If you haven't yet read about best practices for creating data sources and when to create an extract, see **Best Practices for Published Data Sources** on page 3065.

General publishing steps

The following steps give an overview of the publishing workflow you will use regardless of the type of data or the server you publish to. Below these steps you can find supplemental information for authentication types and using Tableau Bridge.

1. Select **Server** > **Publish Data Source**.

   If your workbook is connected to multiple data sources, select the one you want from the Publish Data Sources submenu.



2. If you’re not already signed in to Tableau Server or Tableau Online, sign in now.

   How you sign in depends on how your administrator has set up your environment. For information, see **Sign in to Tableau Server or Tableau Online** on page 364.

3. In the **Publish Data Source** dialog box, do the following:

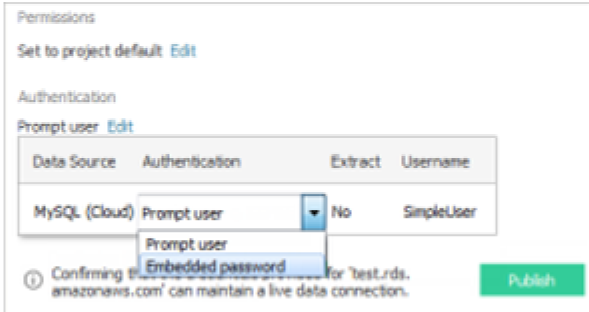
   - For **Project**, select the project you want to publish to and enter the data source name.
   - For **Description** and **Tags**, add a description and tags that will help you and other users find it.

     Separate tags using either a comma or a space. To add a tag that contains a space, put it in quotation marks (e.g., “Sales Quotes”).
   - For **Refresh Schedule**, if publishing an extract, you have the option to select a refresh schedule to ensure that your extract data stays fresh. If you choose not to select a schedule here, you can select a schedule from Tableau Server after publishing.

     Note: This option is not available when publishing from the Data Source page or to Tableau Online.
   - For **Permissions**, accept the default project settings.

     Generally a site administrator manages permissions on the server. If you think your data source is an exception, work with your administrator to determine the best course of action, and see **Set Permissions as You Publish a Data Source or Workbook** on page 3076.
   - For **Authentication**, if you need to provide credentials to access your data, you

can specify how authentication should be handled when the data is published to the server.



The options available for accessing the data source depend on the type of data you publish and whether you are publishing to Tableau Server or Tableau Online.

Information appears at the bottom of the dialog box to let you know whether you need to take further action, such as adding Tableau Online to your data provider's authorized list.

For information about the authentication types, see **Set Credentials for Accessing Your Published Data** on page 3079.

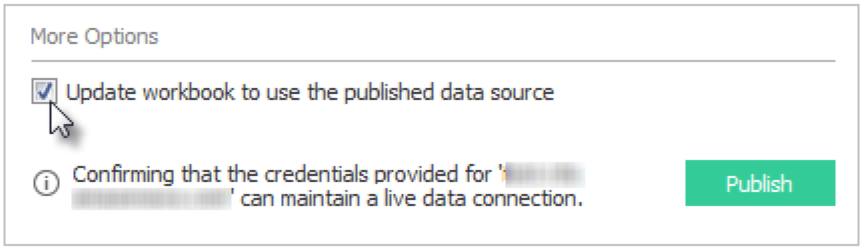
4. If you are publishing file-based data that is on a Windows mapped drive, or using images that will not be available from the server, select **Include external files**.

   When you include external files, copies of the files are put on the server as part of your data source. Copies of files are also put on the server and included as part of the data source when you publish extracts of multi-connection data sources that contain a connection to file-based data such as Excel. For more information about the implications of publishing extracts of multi-connection data sources, see **Join Your Data** on page 883

   If you do not want to publish the external files to the server, change the connection information so that the data source references a full UNC path. For example, rather than connecting to D:\datasource.xls, you would connect to \\filesrv\datasource.xls.

5. By default, during the publishing process, Tableau updates the workbook connection to use the new published data source. It also closes the local data source.

   To continue using the local data source instead, clear the **Update workbook to use the published data source** check box.



> **Note:** If you select **Undo** after publishing the data source, Tableau will revert to using the local data source, but the data source will remain published. In addition, Tableau does not replace a local data source when you publish a cube (multidimensional) data source to Tableau Server. (Tableau Online does not support publishing cube data sources.)

6. Click **Publish**.

   After publishing is complete, your web browser opens Ask Data for the data source, where you can ask questions to automatically create vizzes. For more information, see **Automatically Build Views with Ask Data** on page 1109.

7. (Optional) Set up a refresh schedule on the server. For more information, see the following topics:

   - Schedule Refreshes on Tableau Server
   - Schedule Refreshes on Tableau Online
   - Schedule Refreshes Using Tableau Bridge - see the section below.

Publish on-premises data (Tableau Online only)

To keep data sources that connect to on-premises data fresh after publishing to Tableau Online, Tableau Bridge is required. Tableau Online relies on Bridge to facilitate the connection between it and data accessible only from inside a private network.

As part of the publishing process, Tableau Online will automatically detect if Bridge is needed. If Bridge is needed, your publishing workflow might differ from the publishing process described above.

Refer to one of the following topics depending on the type of data source you're publishing:

- If publishing a data source that uses an extract connection, see Set up a Bridge Refresh Schedule in the Tableau Online Help. If publishing a data source that connects to a private cloud database, see Set up schedules for private cloud-based data sources.
- If publishing a data source that uses an live connection, see Publish a Bridge Data Source with a Live Connection in the Tableau Online Help. If publishing a data source that connects to a private cloud database, see Publish private cloud-based data sources to ensure Bridge is used to facilitate the data freshness tasks.

Note: Bridge keeps data fresh for published data sources only, i.e., data sources published separately from workbooks. Bridge can't keep data fresh in data sources embedded in workbooks.

Publish with a Web Data Connector

To publish a web data connector data source, you need to *import* the web data connector to the server before you can set up a refresh schedule. You can do this only on Tableau Server.

You can refresh some web data connector data sources on Tableau Online, using Tableau Bridge.

For information, see Web Data Connectors in Tableau Server in the Tableau Server Help or Connectivity with Tableau Bridge in the Tableau Online Help.

Hidden fields in workbooks

Workbooks connected to a published data source respect the state of hidden fields in the published the data source.

- If you create a new workbook that uses a published data source with hidden fields, those fields remain hidden in the workbook and cannot be used in calculations, sets, groups, and other object creation.
- If you work with an existing workbook that uses a published data source with hidden fields, those hidden fields are displayed in red in the workbook to indicate that the fields, and therefore the views and calculations that use those fields, are invalid.

You can address this issue in one of the following ways:

- Show (unhide) the relevant fields in the data source, and then republish the data source.
- Update the relevant workbooks to exclude the hidden fields.

For information, see **Hide or Unhide Fields** on page 1155.

See also

- Keep Data Fresh (Tableau Online)
- Data Sources (Tableau Server)

Set Permissions as You Publish a Data Source or Workbook

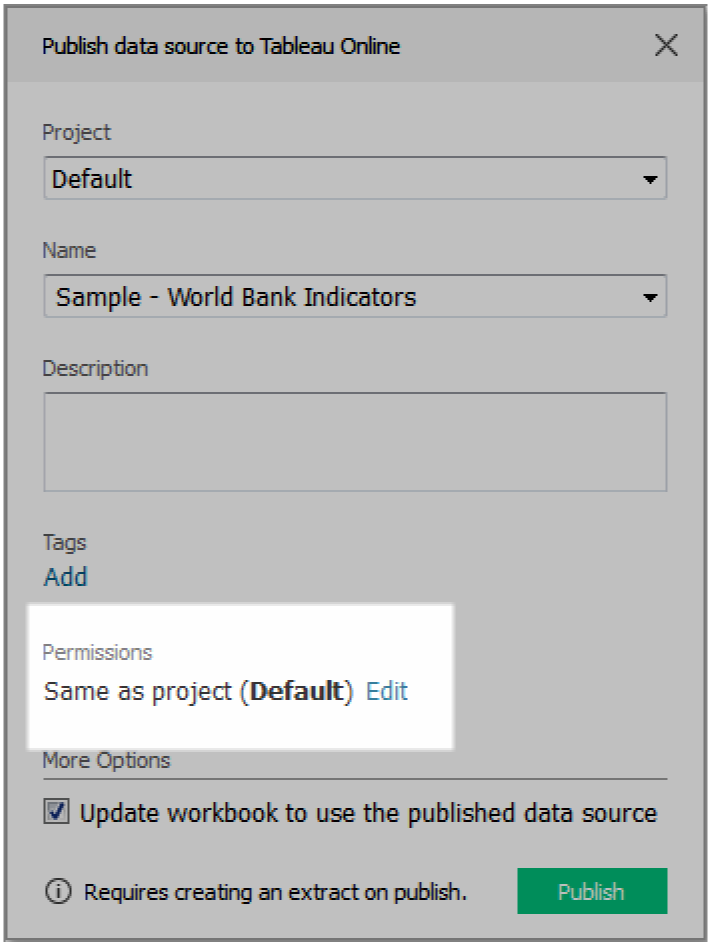
As the publisher of a workbook or data source, you may be able to set permissions as part of the publishing process. Permissions allow or deny other users access to published content on Tableau Server or Tableau Online. For example, who can interact with views in a workbook, download a copy of a data source, and so on.

Note that permissions are different than access to the data source. Accessing some data types requires signing in using a database name and password or embedding database credentials into the connection. For information, see **Set Credentials for Accessing Your Published Data** on page 3079.

When possible, it is best to use the default permission rules for the project where you are publishing your content. If you are publishing to a locked project, you won't be able to modify the permissions. But if you are publishing content into a customizable project and there is a reason your content should have unique permissions, you can set permission rules during publishing.

About setting permissions during publishing

When you start the publishing process, the dialog box shows the permissions that will be applied. By default, the content you publish follows the permission rules of project you are publishing to.



When you change permissions in the publishing dialog box, you are setting unique permission rules for the content you're publishing. This means changes to the permission rules for the project will not impact your content. Depending on your environment, this might be as you intend, or it might conflict with the guidelines your administrator has set and have unintended consequences.

Tips for deciding whether to set permissions on content

- **Learn your organization's practices**

  Consult with your Tableau administrator to learn the guidelines for your organization. It's common (and recommended) practice for an administrator to lock permissions to the project. If you work in such an environment, you will not be able to set unique permissions for your content. And be aware, even if you set permissions during publishing, the person who manages permissions on the server might change these settings afterward.

- **Know the consequences of setting explicit permissions**

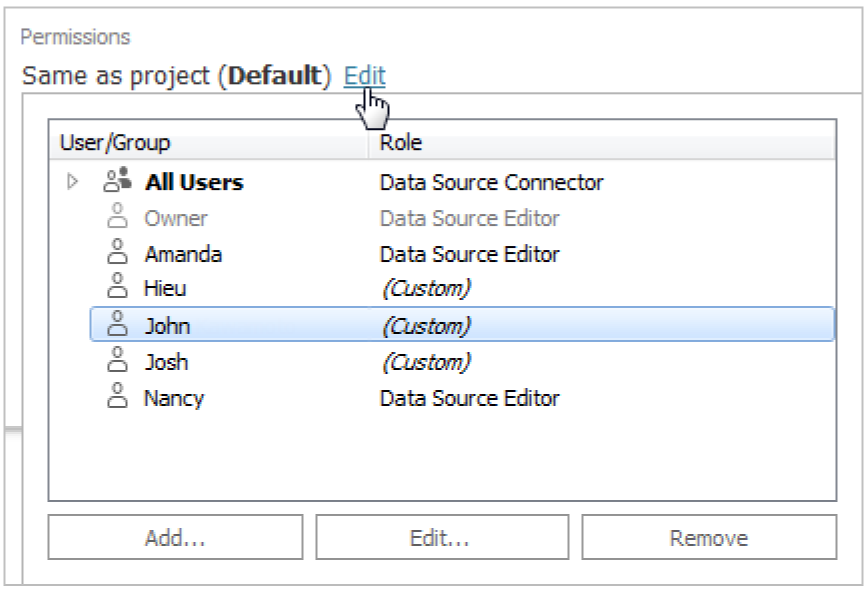
In addition to the potential conflicts described earlier, unique permissions on some content requires extra maintenance to keep track of which content has exceptions, and which exceptions are applied.

- **Publish quickly by accepting the default permissions settings**

  If necessary, you or your administrator can update permissions on the server afterward, where you have a more comprehensive view into the effects of your changes.

How to set permissions during publishing

1. In the publishing dialog box, next to the summary that indicates the current settings, click **Edit**.



2. In the popup that appears, do one of the following:

   - To set custom capabilities or assign a role explicitly, select an existing user or group and click **Edit**, or click **Add**.

     In the **Add/Edit Permissions** dialog box, make your changes.

     Click **Apply** to save changes and keep the dialog box open to configure another user or group. Click **OK** to close the dialog box.

   - To remove a permission rule, select the user or group, and then click **Remove**.

Assign permission templates

When you publish content, you can assign any of the following templates to a selected user or group:

- **View**: Allows the user basic access to the content, such as filtering a workbook or connecting to a data source.
- **Explore**: Allows the user all capabilities from the View template as well as additional functionality, such as web editing a workbook or downloading a data source.
- **Publish**: Allows the user the ability to overwrite the content (such as via web editing and saving over the original). This will also give them ownership of the content.
- **Administer**: Allows the user the ability to manage the content, such as deleting it and setting permissions.

Set Credentials for Accessing Your Published Data

When you publish a workbook to Tableau Online or Tableau Server, you can publish the data source it connects to as part of the workbook (*embedded* into the workbook), or as a separate, standalone data source. In addition, if the data source you're publishing requires authentication, you can customize how credentials are obtained.

The type of authentication to your data source is independent of how people sign in to your Tableau Online or Tableau Server site. For example, to give people direct access to the data in a workbook, you would embed a database user's credentials into the data source's connection. But anyone viewing the workbook would still need to be able to sign in to the site on Tableau Online or Tableau Server to open your workbook.

This topic describes how to set authentication on data connections as part of the publishing process.

Note: This topic does not apply to connections to that do not require authentication, such as text files or Excel files.

Set the authentication type

For many types of connection you can embed a database user's name and password, or use single sign on (SSO). Specific exceptions are described later in this topic.

The following steps describe how to set authentication as part of publishing a data source or workbook. You can do this for each connection in the data source.

1. In the Publish Workbook dialog box, go to the **Data Sources** area, which lists the workbook's connections, and select **Edit**.
2. In the **Manage Data Sources** popup, after you decide whether to publish the data source separately or as part of the workbook, select an authentication type for each connection in the data source.

   The available authentication types depend on the connection type, and they can include one or more of the following:

   - **Prompt user**: Users must enter their own database credentials to access the published data when the view or workbook loads.
   - **Embedded password**: The credentials you used to connect to the data will be saved with the connection and used by everyone who accesses the data source or workbook you publish.
   - **Server run as account**: A single Kerberos service account will be used to authenticate the user. On Windows this is the account that Tableau Server runs as. On Linux it can be any Kerberos account.
   - **Viewer credentials**: The viewer's credentials are passed through to the database using SSO (usually Kerberos).
   - **Impersonate with embedded account** or **Impersonate with server Run As service account**: Impersonation using embedded credentials connects with the embedded credentials and then switches to the viewer's identity (only for databases that support this). Impersonation using the Run As service account is similar but first, connects with the Kerberos service account before switching to the viewer's identity.
   - **Refresh not enabled** or **Allow refresh access**: These options appear when you publish an extract of cloud data such as from Salesforce, and database credentials are needed to access the underlying data. **Allow refresh access** embeds the credentials in the connection, so that you can set up refreshes of that

extract on a regular schedule. Setting **Refresh not enabled** prompts users when they open the workbook.

Important: How you want to keep extracted data fresh is also a factor:

- If you want to set up an automatic refresh schedule, you must embed the password in the connection.
- If you're publishing a cloud data connection to Tableau Online, the publishing steps will alert you if you need to add Tableau Online to the data provider's authorized list.
- You can't publish an extract that's created from a Kerberos-delegated, row-level-secure data source.

Dropbox, OneDrive connections

For Dropbox and OneDrive, when you publish a data source or workbook and select **Embedded password**, Tableau creates a saved credential and embeds it in the data source or workbook.

Workbook connections to Tableau data sources

When you publish a workbook that connects to a Tableau Online or Tableau Server data source, rather than setting the credentials to access the underlying data, you set whether the workbook can access the published data source it connects to. Regardless of the original data type, the choice for server data sources is always **Embedded password** or **Prompt users**.

If you select to prompt users, a user who opens the workbook must have **View** and **Connect** permissions on the data source to see the data. If you select embed password, users can see the information in the workbook even if they don't have View or Connect permissions.

See also

- If you publish to Tableau Server, see Edit Connections in the Tableau Server Help.
- If you publish to Tableau Online and the workbook connects to Salesforce, Google Analytics, Google Sheets, Google BigQuery, OneDrive, Dropbox, and QuickBooks Online data, see Refresh Data Using Saved Credentials in the Tableau Online Help.
- If you are a Tableau Server administrator looking for more information about authentication, see the Tableau Server help topics, "Authentication" (Windows | Linux) and "Data Connection Authentication" (Windows | Linux).

Authorize Access to Cloud Data Published to Tableau Online

The information in this topic applies to you if you publish workbooks or data sources to Tableau Online that contain live connections to cloud data—for example, Amazon, Google, Salesforce.com.

As a security measure, cloud data providers might require you to supply a list of authorized IP addresses from which external applications request access to your data. A request from an IP address that is not explicitly approved could be rejected. To make sure live connections you publish to Tableau Online remain uninterrupted, add Tableau Online to your data provider’s allowlist (safe list).

The table lists IP address ranges Tableau Online uses, depending on your site location. You can see its location in the URL that appears after you sign in to Tableau Online.

Host Name (Instance)	Site Location	IP Address or Range
10ax.online.tableau.com	US West - Oregon	34.208.207.197 52.39.159.250
10ay.online.tableau.com	US West - Oregon	34.218.129.202 52.40.235.24
10az.online.tableau.com	US West - Oregon	34.218.83.207 52.37.252.60
us-west-2b.online.tableau.com	US West - Oregon	34.214.85.34 34.214.85.244
us-east-1.online.tableau.com	US East - Virginia	50.17.26.34 52.206.162.101
prod-useast-a.online.tableau.com	US East - Virginia	3.219.176.16/28
prod-useast-b.online.tableau.com	US East - Virginia	3.219.176.16/28

Host Name (Instance)	Site Location	IP Address or Range
dub01.online.tableau.com	EU West - Ireland	34.246.74.86 52.215.158.213
eu-west-1a.online.tableau.com	EU West - Ireland	34.246.62.141 34.246.62.203
prod-apnortheast-a.online.tableau.com	Asia Pacific - Japan	18.176.203.96/28
prod-apsoutheast-a.online.tableau.com	Asia Pacific - Australia	3.25.37.32/28
prod-uk-a.online.tableau.com	EU West - UK	18.134.84.240/28
prod-ca-a.online.tableau.com	Canada - Quebec	3.98.24.208/28

These addresses are dedicated to and controlled by Tableau.

Note: In addition to enabling communication over the Tableau Online IP range, you might need to enable access over the appropriate database port (for example, 80 or 443) depending on the communication type (HTTP or HTTPS).

Find authorization steps for your data provider

The following links take you to the steps on common data providers' websites for authorizing external applications on their platforms.

Amazon:

- Redshift
- RDS
- EC2

Microsoft Azure

Google Cloud Platform

Disclaimer: The links in the list above take you outside of Tableau.com. Although we make every effort to ensure links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

Tableau Bridge connections to Tableau Online

To facilitate connections between on-premises data and Tableau Online, Tableau Bridge uses common ports (80 and 443) to make outbound requests to Tableau Online. For more information, see Network access section in the Install Tableau Bridge topic.

See also

Keep Data Fresh (Tableau Online Help)

Use Bridge to Keep Data Fresh

For data sources that Tableau Online can't reach directly, you can use Tableau Bridge to keep data fresh. For example, use Bridge when your data source connects to data hosted behind a firewall.

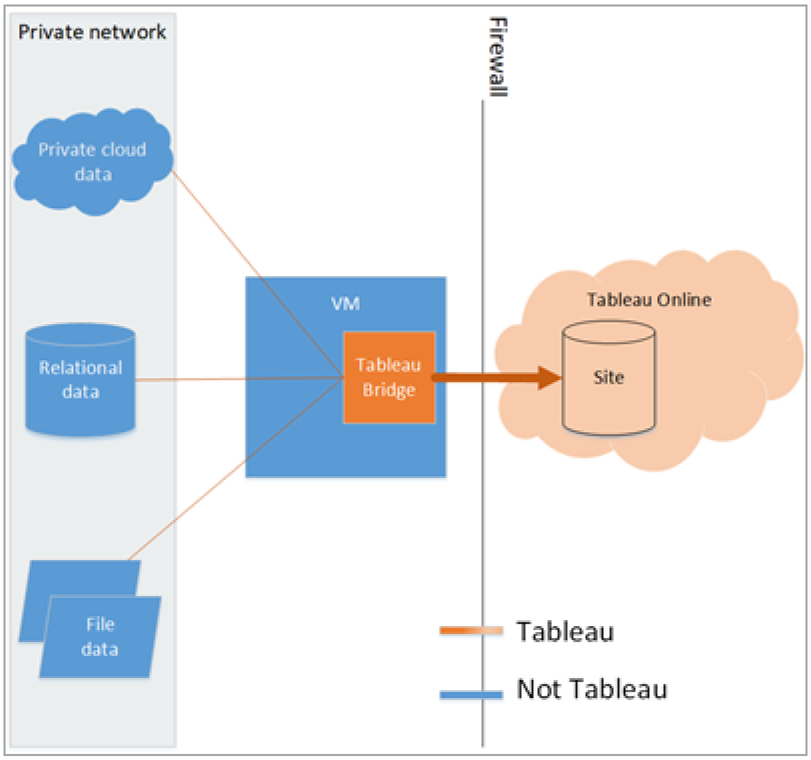
Note: If a data source connects to underlying data hosted in the cloud accessible from the public internet, live and extract connections run from Tableau Online directly.

What is Bridge

Tableau Bridge is client software that runs on a machine in your network. The client works in conjunction with Tableau Online to keep data sources that connect to private network data, which Tableau Online can't reach directly, up to date. Private network data includes on-premises data and virtual cloud data.

How does it work

Tableau Bridge functions like a conduit between private network data, such as Excel files and SQL Server data, and Tableau Online. The client communicates with Tableau Online through an outbound encrypted connection to enable connectivity between data behind a firewall and your Tableau Online site.



For more information about how Bridge communicates with Tableau Online, see Tableau Bridge Security.

Who can use it

Although any authorized user of Tableau Online can use Bridge, Bridge is optimized for users that perform the following functions in an organization: site admins and data source owners.

Site admins, or users who have the Site Administrator or Site Administrator Creator role on Tableau Online, install and manage Bridge clients. For more information, see Plan Your Bridge Deployment.

Data source owners, or users who have the Creator or Explorer (can publish) role on Tableau Online, typically publish and manage their own data sources. Data source owners use Bridge to facilitate the live and extract connections between Tableau Online and private network data.

- For **live connections**, Bridge is detected automatically as part of the data source publishing process.

  After the site admin enables support for maintaining live connections (through pooling), users see the option to publish the data source with a live connection during the publishing process. This option is available when live connections are supported for relational or cloud databases accessible only from inside the network.

  After the user publishes the data source, an available client in the pool facilitates the live queries. That's all there is to it.

  To get started, users publish a data source to Tableau Online, and select the option to maintain a live connection. Or, publish a workbook, and select the option to publish the data source separately, and then specify a live connection. For more information about publishing data sources, see Publish a Bridge Data Source with a Live Connection.

  Note: If publishing a data source that connects to a private cloud database, follow the steps described at Publish private cloud-based data sources to ensure Bridge is used to facilitate the data freshness tasks.

- For **extract connections**, users can set up refresh schedules. Users need to publish the data source separately, whether he or she goes through the publishing data source or workbook process. For more information, see Set Up a Bridge Refresh Schedule.

  Note: If publishing a data source that connects to a private cloud database, follow the steps described at Set up schedules for private cloud-based data sources to use Bridge facilitated refresh schedules.

Schedule Extract Refreshes as You Publish a Workbook

When you publish workbooks that connect to extracts, you can set up a schedule for updates (refreshes) for those extracts, so the views in those workbooks stay current.

The ways you can set up and manage extract refresh schedules depends on where you publish and on the original data type.

- When you publish to **Tableau Server**, the schedule runs and is managed on the server.
- When you publish to **Tableau Online**:
  - If extracts are from cloud data (for example, Google Analytics or Salesforce), schedules run and are managed on Tableau Online.
  - If extracts are from on-premises or web data connector (WDC) data, you set up and manage refresh schedules using Tableau Bridge.

Note: In many organizations a server content manager or administrator manages all refresh schedules. We recommend that, before you publish, you find out from your administrator whether they have guidelines for these schedules.

Set up a schedule as you publish content

During the publishing process, after you click the **Publish** button, the scheduling options appropriate for your data types and publishing destination appear.

If you are publishing a multi-connection data source, you need to set a refresh schedule for each extract connection in the data source.

To use refresh schedules with connections to a database that requires you to sign in to it, you must store (embed) a database user's credentials with the connection. For more information, see **Set Credentials for Accessing Your Published Data** on page 3079.

Publish and refresh on Tableau Online

For the subtleties around refreshing each type of data you publish to Tableau Online, see Keep Data Fresh. Here are examples of things you need to know:

- If your workbook connects to cloud data, add Tableau Online to your cloud data provider's authorized list (allowlist). The IP address range for your site location is listed in **Authorize Access to Cloud Data Published to Tableau Online** on page 3082.
- Tableau Bridge starts during the publishing process if your data source or workbook connects to on-premises data.

  Tableau Bridge supports standard database authorization and cannot refresh data you connect to through OAuth. To learn more, see Use Tableau Bridge to Expand Data Freshness Options.
- When you publish a multi-connection data source to Tableau Online, if any one

connection requires using Tableau Bridge, you must use Bridge to refresh *all* connections in the data source.

For example, say you publish a data source with an extract connection to a MySQL data hosted in the cloud, and a live connection to an on-premises SQL Server database. Although Tableau Online supports refreshing hosted MySQL data, in this case, you would need to use the Bridge client to keep both the SQL Server and MySQL connections fresh.

For information about using both live and extract connections in a data source, see About working with multi-connection data sources.

Refreshing web data connector extracts

When you publish a workbook with a web data connector (WDC) data source, you must *import* the web data connector to Tableau Server before you can set up a refresh schedule. You can do this only on Tableau Server. For information, see Web Data Connectors in Tableau Server in the Tableau Server Help.

For information about refreshing web data connector data sources published to Tableau Online, see Use Tableau Bridge to Expand Data Freshness Options in the Tableau Online Help.

Restrict Access at the Data Row Level

When you share workbooks with others by publishing them to Tableau Server or Tableau Online, by default, all users who have access to the workbooks can see all of the data shown in the views. You can override this behavior by applying a type of filter that allows you to specify which data “rows” any given person signed in to the server can see in the view.

This approach for securing data at the row level applies to data sources with live connections and extract data sources whose tables are stored as multiple tables. For more information about storing extract data using multiple tables, see **Decide how the extract data should be stored** on page 1022.

> **Note**: For related information, see the whitepaper Best Practices for Row Level Security with Entitlement Tables or How to Set Up Your Database for Row Level Security in Tableau on the blog *Tableau and Behold*.

How user-based filtering works

Suppose you created a quarterly sales report for a set of products over several years, in different geographic regions.

When you publish this report, you want to allow each regional manager to see only the data relevant to his or her region. Rather than creating a separate view for each manager, you can apply a *user filter* that restricts access to the data based on users' characteristics, such as their role.

Restricting access to data in this way is referred to as *row-level security* (RLS). Tableau offers the following approaches to row-level security:

- **Create a user filter and map users to values manually** on page 3092.

  This method is convenient but high maintenance, and security can be tentative. It must be done per-workbook, and you must update the filter and republish the data source as your user base changes.

- **Create a dynamic filter using a security field in the data** on page 3095.

  Using this method, you create a calculated field that automates the process of mapping users to data values. This method requires that the underlying data include the security information you want to use for filtering.

The most common way to do this is to use a reference (“look-up,” "entitlements," or "security") table that contains this information. For example, if you want to filter a view so that only supervisors can see it, the underlying data must be set up to include user names and specify each user’s role.

Because filtering is defined at the data level and automated by the calculated field, this method is more secure than mapping users to data values manually.

Adding user filters to data sources

The two methods in the previous section describe ways to add filters to data embedded in workbooks. If multiple workbooks connect to the same data, instead of wrangling filters on each workbook, you can filter the data source, and then connect the workbooks to the data source after you publish it.

Workbooks that connect to your filtered data source expose only the data the user signed in to the server is allowed to see. In addition, all connected workbooks show data refreshes as they occur.

Extracts vs. live connections with user filters

Generally, when using one of the methods described above, RLS with extracts are faster to create and have better performance than RLS with data sources that use live connections.

Requirements for RLS with extract data sources

As mentioned earlier, the first requirement to using RLS with extracts is that the data in the extract should be stored using multiple physical tables. You can configure your extract to have its data stored using multiple physical tables by following **Decide how the extract data should be stored** on page 1022.

In addition to the above requirement, there are some additional considerations to make if you plan to use RLS with your extract. Because extract data stored using multiple tables do not support extract filters and some other functionality that help reduce the amount of data in the extract, you might consider using one of the following suggestions:

- Connect to data using custom SQL
- Connect to a database view that already has the appropriate level of filtering

For more information about these suggestions, see **Alternative filtering suggestions when using the Physical Tables option** on page 1030.

Recommended practices for RLS with extract data sources

To effectively perform RLS with extracts, Tableau recommends keeping the number of tables (or database views or custom SQL queries) in your extracts to two. In other words, Tableau recommends that the tables in your extract be comprised of the following types of tables:

- A data table—this is the "object" table that contains all the data you want to show.
- A reference table—this is the "look-up" or "entitlements" table that contains the user information and the security groups the users belong to.

By minimizing the tables in your extract to these two, you ensure that the only join that Tableau has to perform is between these two tables and thus avoid any duplication of data or "join explosion."

About RLS and previous versions of Tableau

Previously, Tableau was unable to support RLS workflows with extracts because of complications around row duplication and performance. Ultimately, these complications derived from the extract whose data could only be stored and queried as a single table. However, beginning in Tableau 2018.3, you can choose to store the data in your extract using multiple tables and thus enabling a workflow for RLS with extracts as you might have previously done with data sources with live connections.

For a comprehensive discussion about RLS with extracts in Tableau, read the blog maintained by a Tableau Sales Consultant who has extensive experience in this area.

- Multiple Table (Normalized) Hyper Extracts
- Defusing Row Level Security…Part 1
- Defusing Row Level Security…Part 2

Disclaimer: Clicking these links will take you away from Tableau.com. Although we make every effort to ensure links to external websites are accurate and relevant, Tableau cannot take responsibility or provide support for the external content.

See also

- Data Security in the Tableau Server Help

Create a User Filter and Secure it for Publishing

Prerequisites

If you want to implement RLS with a data source that uses a live connection, you can proceed with the steps below.

If you plan to implement RLS with an extract data source, there are some extra steps and considerations you should make before proceeding with the steps below. For more information see **Requirements for RLS with extract data sources** on page 3090.

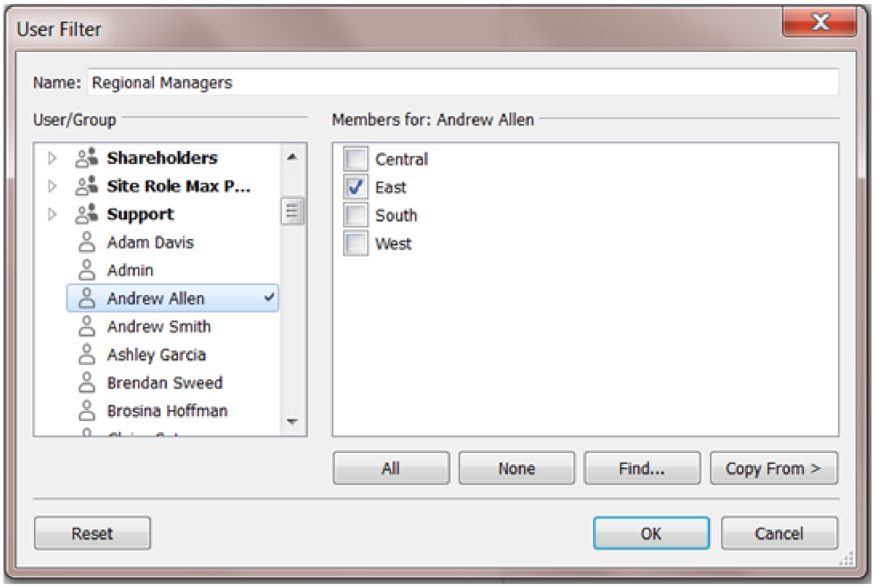
Create a user filter and map users to values manually

The steps in this section describe the simplest way to incorporate user-based filtering to help secure your data source or workbook using RLS. This procedure can be sufficient if you have a small and fairly static set of users or groups to manage, and only a small number of workbooks that need user filters. Completing the steps is also a low-cost way to get more familiar with user filtering.

1. In Tableau Desktop, open the workbook, or create a new one, and set up the connection to the data you want to filter.
2. Navigate to the worksheet that you want to apply a filter to.
3. Select **Server** > **Create User Filter**. Then select the field you want to use for filtering the view. This example uses **Region**.
4. If prompted, sign in to your server or site. For information, see **Sign in to Tableau Server or Tableau Online** on page 364.
5. In the **User Filter** dialog box, type a name for the set of rules you are creating.

   In this example, we'll use **Regional Managers**.
6. In the list on the left, select a user or group. On the right, select the individual members of the field you selected earlier, that you want the selected users to be able to see.

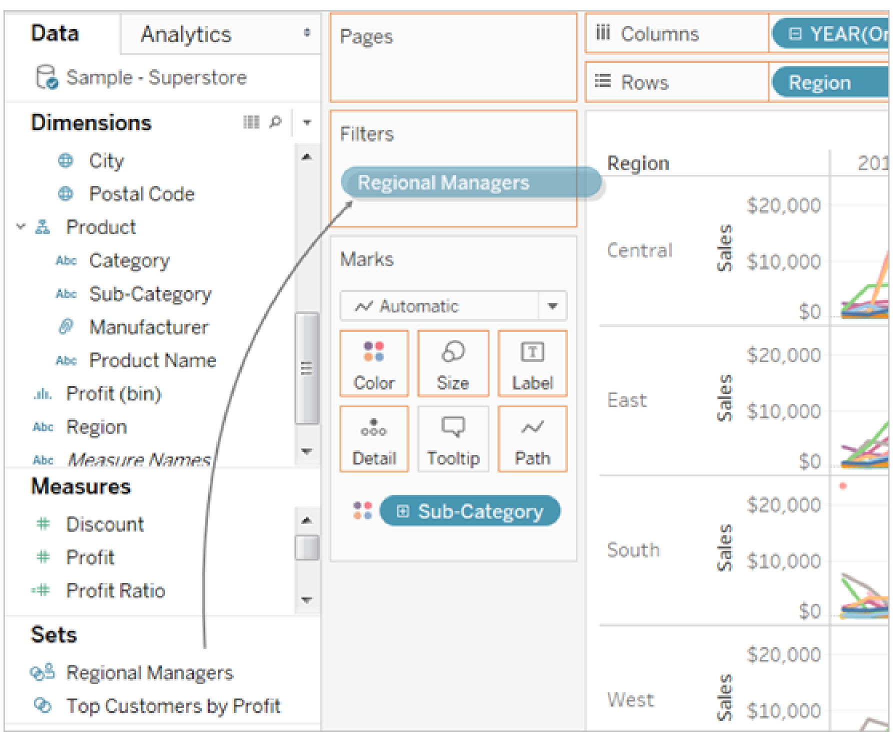
   For this example, selected user Andrew Allen is the manager of the Eastern region, so in the field member list, you would select **East**.



Repeat this process for each user or group, and click **OK** when you're done mapping users to values.

After you create the user filter, it appears in the **Sets** area of the Data pane.

7. Drag the user filter to the **Filters** shelf.



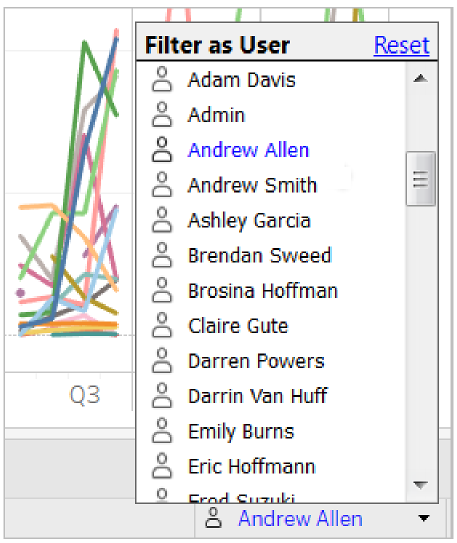
The filter becomes a context filter, and the view adjusts to show data that you are allowed to see.

8. Do any of the following to test or fine-tune the filter:

   - If the view appears as a blank canvas, you need to allow yourself or a group you are a member of to see a region.

     In the **Sets** area of the **Data** pane, open the drop-down menu on the user filter, and then select **Edit Set**.

   - To preview how the filter works in the published view, in the lower-right corner of the workbook, open the **Filter as User** menu, and select the user or group from the list.



     Note: Previewing is not available if the workbook connects to a Tableau Server data source.

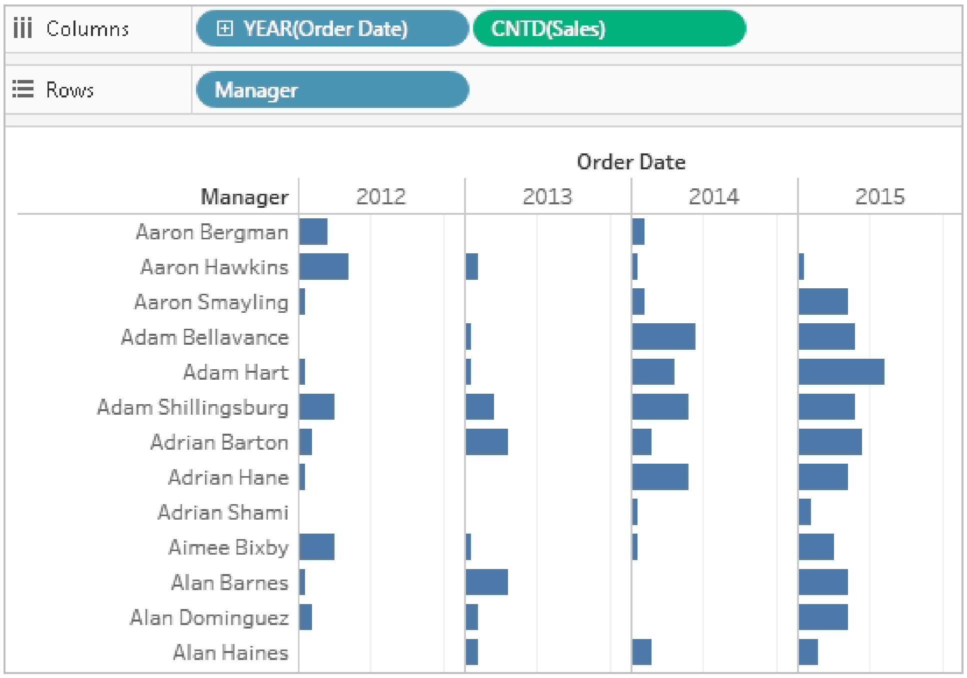
   - To return to viewing the workbook as yourself, in the top right corner of the **Filter as User** menu, select **Reset**.

   - To copy the mapping selections you set on one user or group to another (rather than manually mapping the same settings), see **Copy selected field values from one user to another** on page 3099.

When you publish the workbook, you need to take additional steps to make sure users cannot edit the workbook and remove the filter. For information, see **Secure user filters on published content** on page 3099.

Create a dynamic filter using a security field in the data

The steps below are based on the following view, which shows annual sales performance for a list of regional managers.



As described in **Restrict Access at the Data Row Level** on page 3088, if you want to take this approach, the database must include the field you want to use for filtering.

For this example, the data includes a reference table called **People**, which contains two columns: Region and Manager. Names in the Manager field match Tableau Server or Online user names, and we'll use this field for filtering.

You can follow along using the Superstore data that comes with Tableau Desktop, although the fields and values do not match exactly.

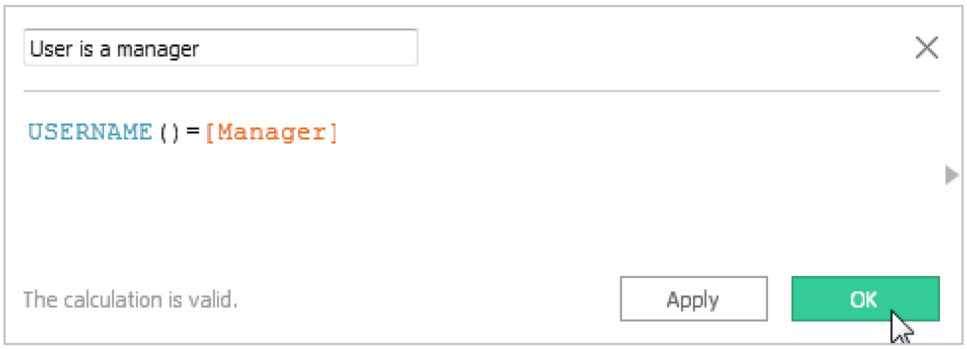
Connect to the data and set up the user filter

1. In Tableau Desktop, open the workbook you want to add user filtering to, or create a new one, connecting to the data you want to filter.

In this example, we use a table called **Orders**.

2. On the **Data Source** page, add the reference table, creating a left join. Here we add the **People** table and create a left join on the **Region** field.

3. Go to the worksheet, select **Analysis** > **Create Calculated Field**, and create the following field:
   - Name: User is a manager
   - Formula: `USERNAME() = [Manager]`



This new true/false field appears in the Dimensions pane. The formula returns TRUE if the user name of the person signed in to the server exists in the manager column.

4. Add the **User is a manager** field to the **Filters** shelf.

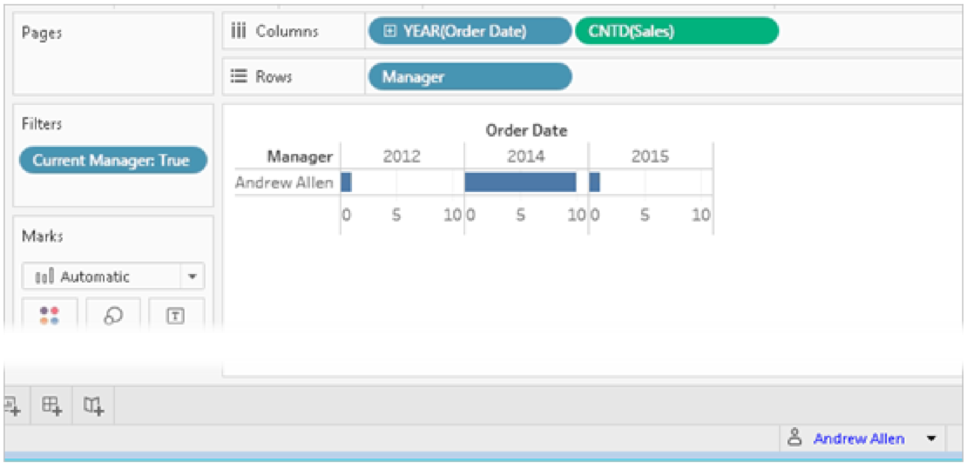
5. In the **Filter** dialog box, select **True**, and then click **OK**.

   This sets the filter so that only people who are managers can see the data in the view.

   If you are not listed in the Manager field, your view might appear as a blank canvas.

6. See how the view looks to a particular person: in the lower-right, open the **Filter as User** menu, and select someone you know is a manager.

   The following image shows how the view shown earlier would look if Andrew Allen were signed in.



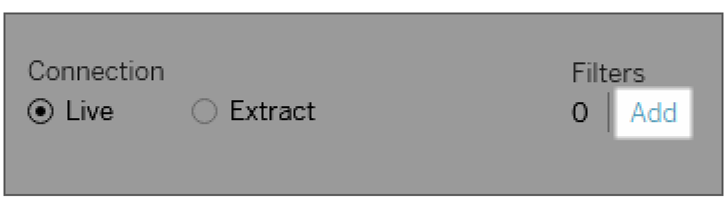
As with a manual user filter, you need to take steps to **Secure user filters on published content** on page 3099.

Filter a data source

Rather than maintain user filters and special permissions on each published workbook, you can filter a data source, and then publish the data source to make it available as a shared, one-to-many resource for anyone who uses that data.

This procedure builds from the dynamic-filter approach described in this topic.

1. Complete the steps in **Create a dynamic filter using a security field in the data** on page 3095.
2. In the lower-left area of Tableau Desktop, select the **Data Source** tab.
3. In the upper-right area of the **Data Source** page, under **Filter**, click **Add**.



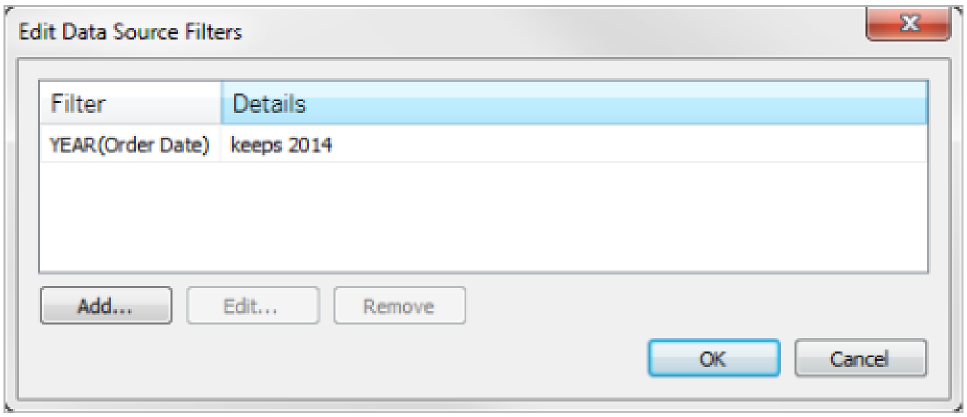
4. In the **Edit Data Source Filters** dialog box, click **Add**, add the calculated field you created for the dynamic filter, and set the filter to **True**.

   In our examples, this is the **User is a manager** field.

   Click **OK** until you get back to the **Data Source** page.

Global filters and data source filters

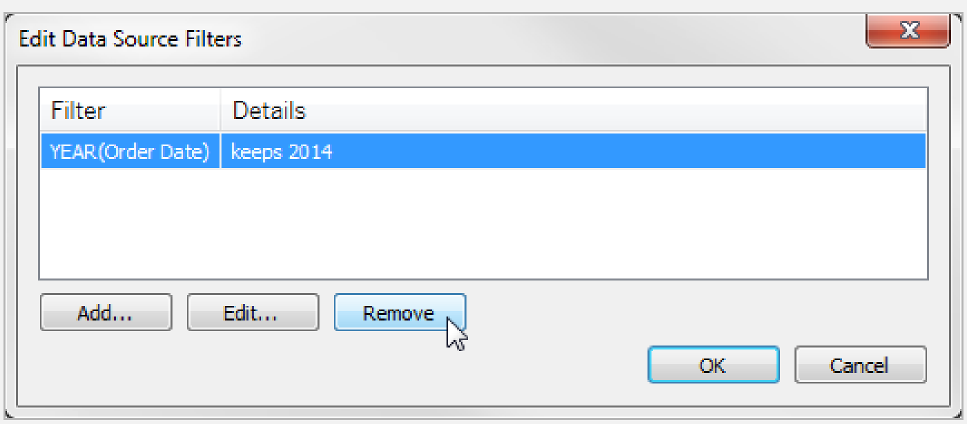
When you create a data source filter, any global filters that use that data source are displayed automatically in the **Edit Data Source Filters** dialog box to make it easy for you to promote a global filter to be a data source filter. To promote the global filter to be a data source filter, click **OK**.



If you promote a global filter to be a data source filter, that global filter will no longer be visible in worksheets of the workbook (because it becomes a data source filter).

Important: Be aware that you do not need to select a global filter in the **Edit Data Source Filters** dialog box to promote it. When you click **OK**, all global filters in the list will be promoted.

To prevent a global filter from being promoted to a data source filter, select the global filter in the **Edit Data Source Filters** dialog box, and then click **Remove**.



Secure user filters on published content

When you publish a workbook or data source with user filters, you need to set permissions to make sure that users opening your workbook or connecting to your Tableau Server data source cannot remove the filter, thereby gaining access to all of the data.

Notes

- Before publishing, we recommend that you consult with your Tableau administrator regarding any existing policies in your organization, particularly for setting permissions. See also **Set Permissions as You Publish a Data Source or Workbook** on page 3076.
- This information applies to publishing data sources with live connections and extracts whose data is stored as multiple tables.

Permissions settings

To secure user filters, the following capabilities must be set to **Deny**, either during publishing, or afterward on the server.

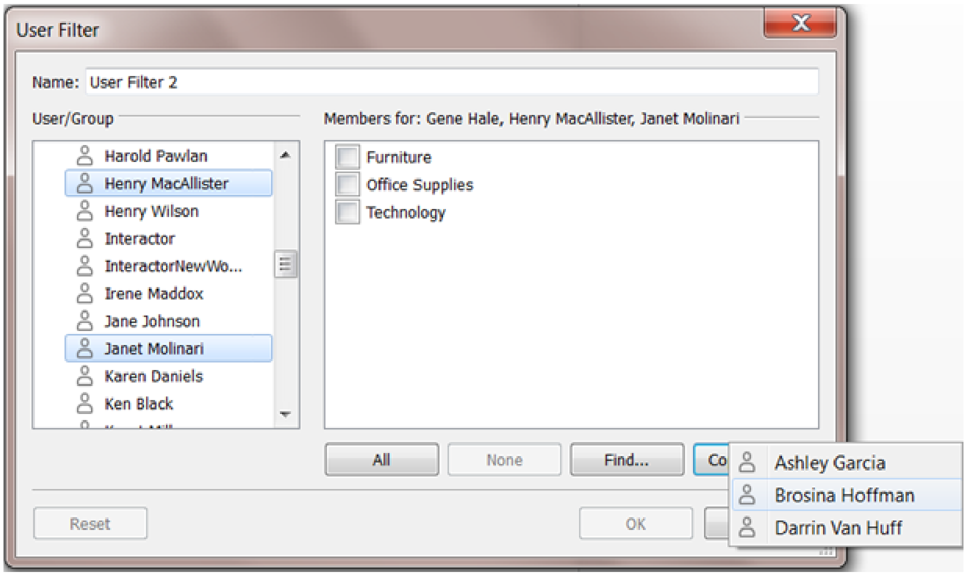
Workbooks	Data sources
• Web Edit • Download Workbook/ Save As • Set Permissions	• Save • Download/ Save As • Set Permissions

Copy selected field values from one user to another

When you're creating a user filter manually, after you map a user or group to data values (members), you might want to map another user or group in the same way. You can do this by copying and pasting the settings.

1. In the **Data** pane, under **Sets**, select a user filter. Click the drop-down arrow and select **Edit Set**.
2. In the **User Filter** dialog box, select the user or group on which you want to paste settings from another user or group.

3. Click **Copy From**, and select the user or group whose settings you want to copy.



See also

Data Security with User Filters, from the Tableau video learning library.

To watch the video, you need to sign in to Tableau.com (or create a free account).

Use Tableau on the Web

Get started

Explore and manage content

Author web views

Share and collaborate

Save your work

If this is the first time you've ever seen Tableau Server or Tableau Online and you're not sure what you can do with it, watch this five-minute video.

To view more training and introductory videos, go to Free Training Videos on the Tableau website.

What Can I Do with a Tableau Site?

Your Tableau site is a private workspace on the web where you can share data insights and collaborate with your colleagues. With certain access levels, you can connect to data and create workbooks right on the web, or connect to data sources and workbooks built in Tableau Desktop. Your site role sets what you’re able to do in your Tableau site.

What is a site?

A Tableau site is a place for your team to publish data sources and workbooks to share with each other.

In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from other sites (collections of users, groups, and content).

As a site user, you might be able to sign in to one site, or you might be able to sign into multiple sites, depending on the access your administrator gives you. After you sign in, you can see projects and content that you can access.

How does my site work with Tableau Desktop?

Tableau Desktop and your Tableau site are built to work together. In Tableau Desktop, you can connect to data and build workbooks, stories, and dashboards. When you want to share insights with others, you can publish workbooks and data sources to your Tableau site.

In your Tableau site, you can view, interact with, and edit those published views. You can also use your Tableau site to find and edit workbooks in Tableau Desktop.

What can I do on the web?

Your site role and content permissions determine what you can see and do with Tableau on the web. Starting in 2018.1, your Tableau site role can be Creator, Explorer, or Viewer. For in-depth details on each site role, see the capability grid on Tableau Pricing.

Not sure what your site role is? To check, select the icon in the top right corner of the site with your initials or profile picture and select **My Account Settings**. **Site Role** tells you if you are a a Creator, Explorer, or Viewer.

In general:

Creators can:

- Make new connections to data in the browser
- Upload workbooks (.twb and .twbx files)
- Build and publish data sources and workbooks from the browser or Tableau Desktop
- Open and edit published workbooks in the browser or Tableau Desktop
- Create new data flows and edit an existing flow
- Use Dashboard Starters (Tableau Online only)
- Everything else that Explorers and Viewers can do

Explorers* can:

- Connect to published data sources on the site to create new workbooks
- Edit and analyze data in published workbooks (Save and Save As options vary based on permissions)
- Create data alerts
- Create subscriptions for others
- Everything that Viewers can do

*The user level Explorer (can publish) is an Explorer who can publish new content from Tableau Desktop to a site and create and publish new content from an existing published data source, but cannot connect to external data or create new data sources.

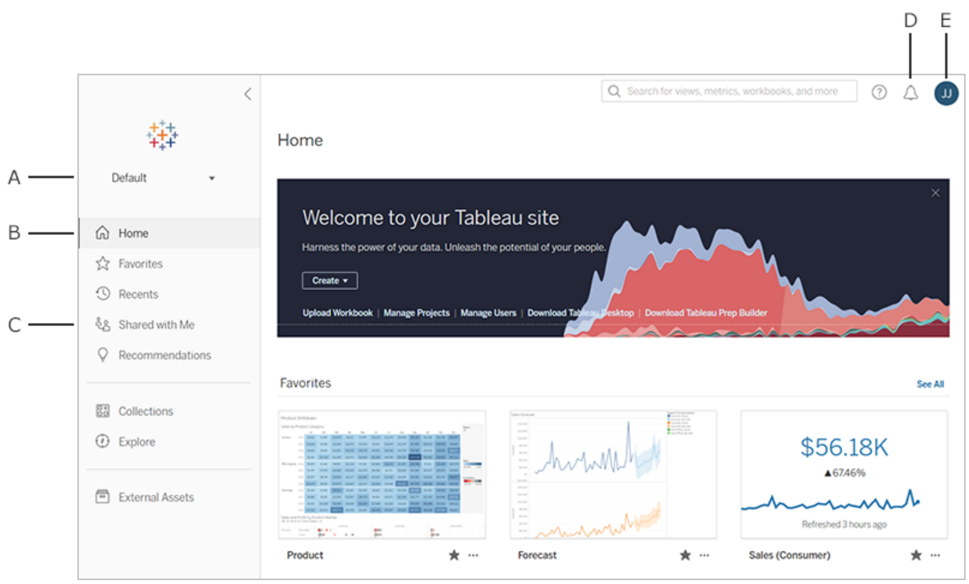
Viewers can:

- Create custom views and see published and custom views others have created
- Explore the data in a view using filters and legends, sorting, and tooltips
- Share, comment on, and download content (options vary based on permissions)

For more details, see What Can I Do with a Tableau Web View?

Tour Your Tableau Site

When you sign in to Tableau Server or Tableau Online, the first page you see looks similar to the following example.

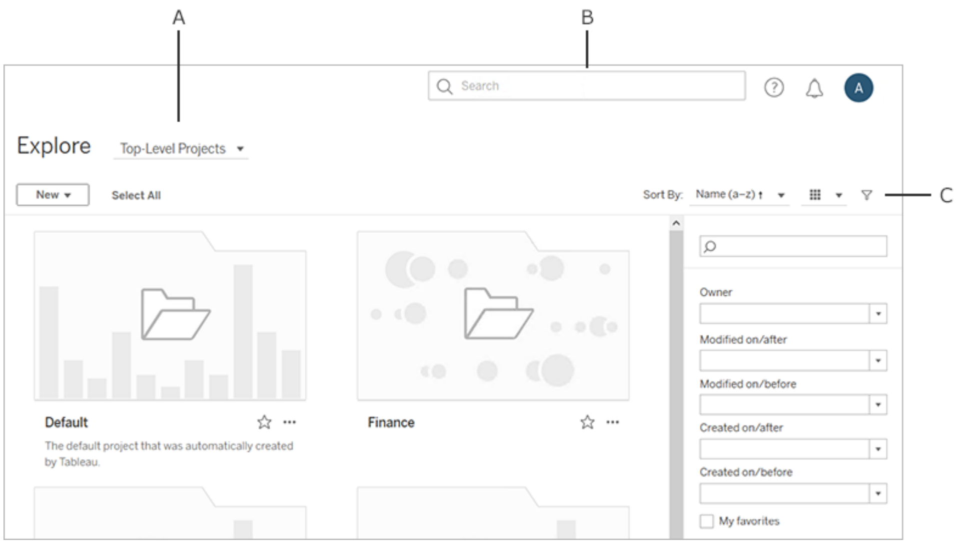


- **A** - Site picker (if you have access to more than one site)
- **B** - Home
- **C** - Navigation menu
- **D** - Notifications
- **E** - Your content and account settings

Note: You might see additional features if your site has the Data Management Add-on, which includes Tableau Catalog and Tableau Prep Conductor. When Tableau Catalog is enabled, you can see an External Assets page, view data lineage information, and more. To learn more, see Use the Data Management Add-on.

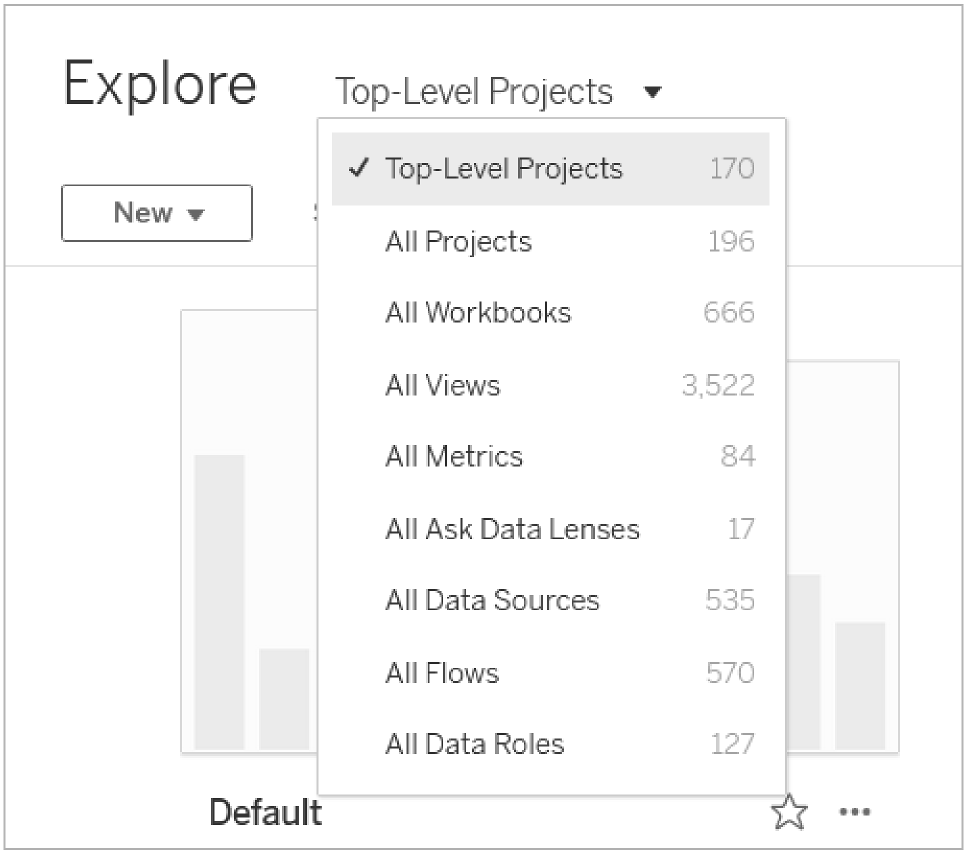
Browse and find content

To see all of the content you have access to across your Tableau site, from the navigation menu, click **Explore**. Your site role and permissions determine the content that you can see and the options that are available.



- **A** - Content type menu
- **B** - Quick search
- **C** - Your account and settings

By default, Explore shows **Top-Level Projects**. To see other types of content, select them from the drop-down content type menu.

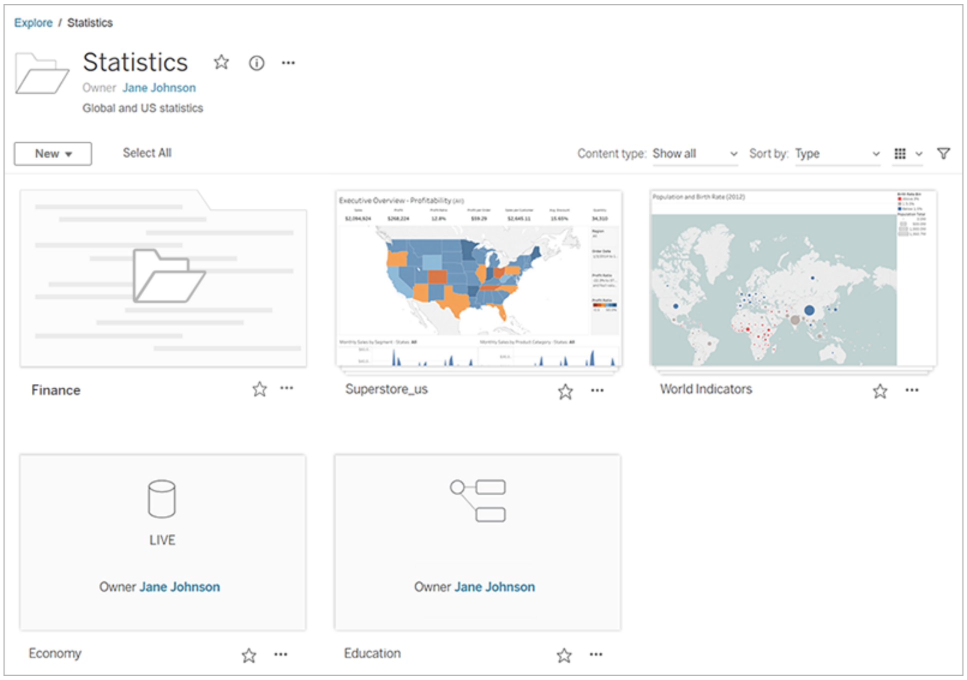


Note: When Tableau Prep Conductor is enabled with the Data Management Add-on, flows from Tableau Prep Builder can be published to your site. To learn more, see Use the Data Management Add-on.

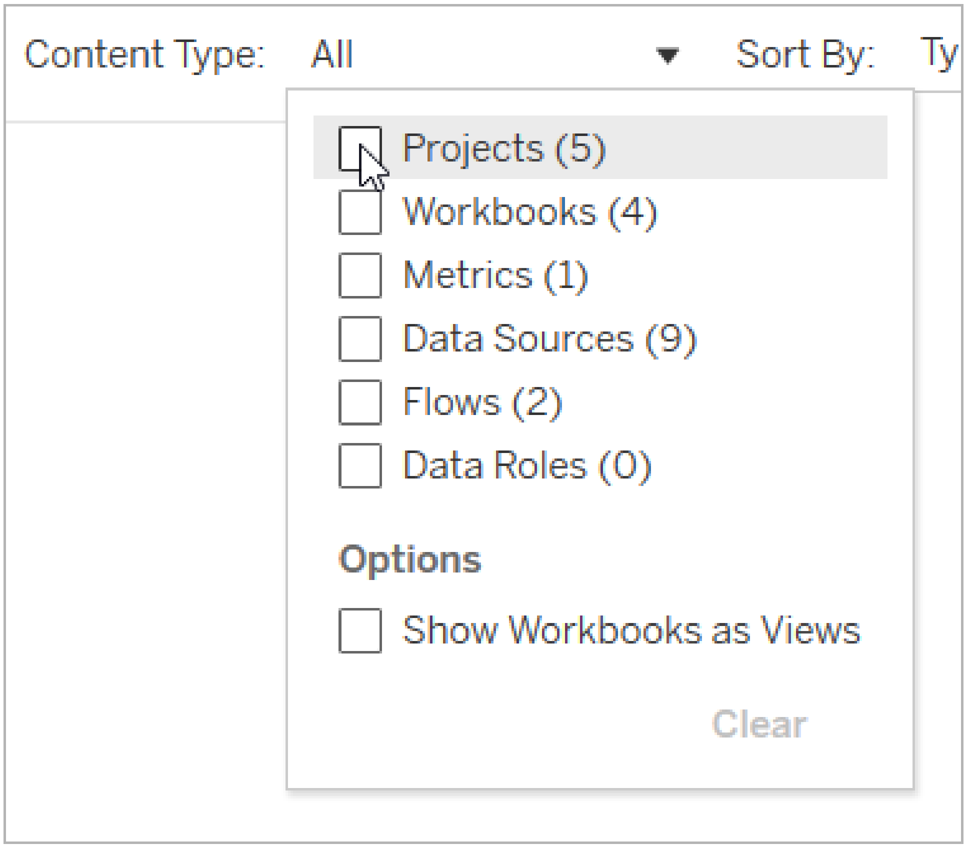
Projects

Projects provide a way to organize the content on your site. Top-level projects can contain other (nested) projects, forming a hierarchy that you navigate like the file system on your computer.

When you open a project, you see all of the content that project contains on a single page, including any nested projects.

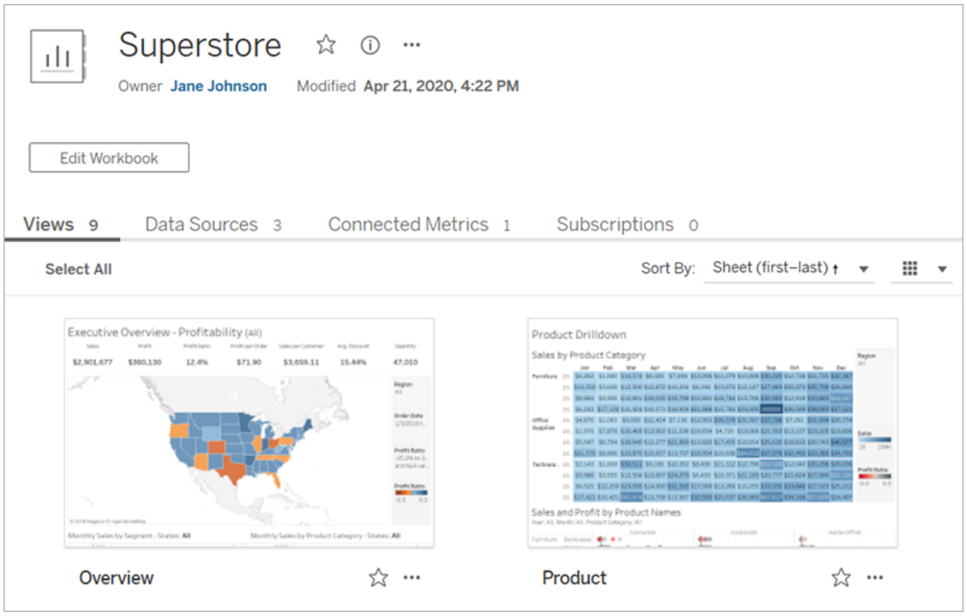


To see only specific content types, click the **Content Type** drop-down menu and select the content types you want to see. You can select the **Show Workbooks as Views** option to see all of the individual views within a workbook displayed on the project page.



Workbooks

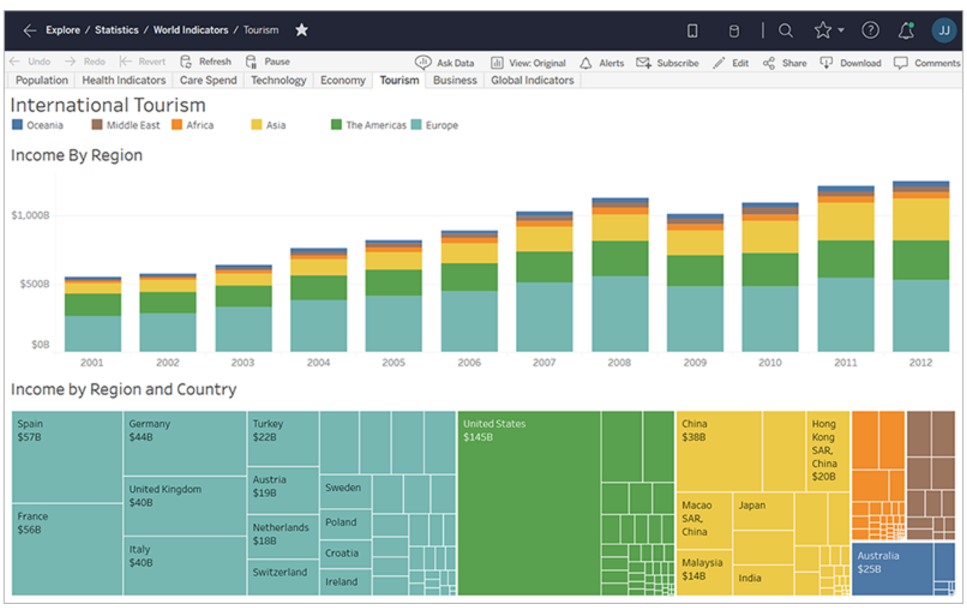
A workbook page shows the views included in the workbook, the data sources used by the workbook, and any metrics created from the views in the workbook. It also provides a list of user subscriptions to either the workbook or to individual views.



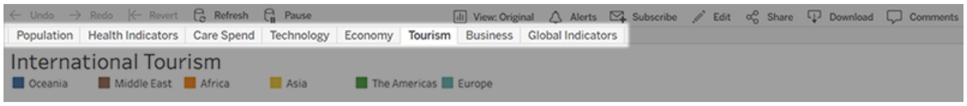
For more information about creating workbooks, see **Creators: Get Started with Web Authoring** on page 3119.

Views

A view page displays options for interacting with the view, including sharing, subscribing, and editing. The options available in the toolbar vary depending on the site configuration and your permissions.



When a workbook is published from Tableau Desktop with the **Show sheets as tabs** option selected, you can see each sheet by clicking the tabs below the view toolbar.

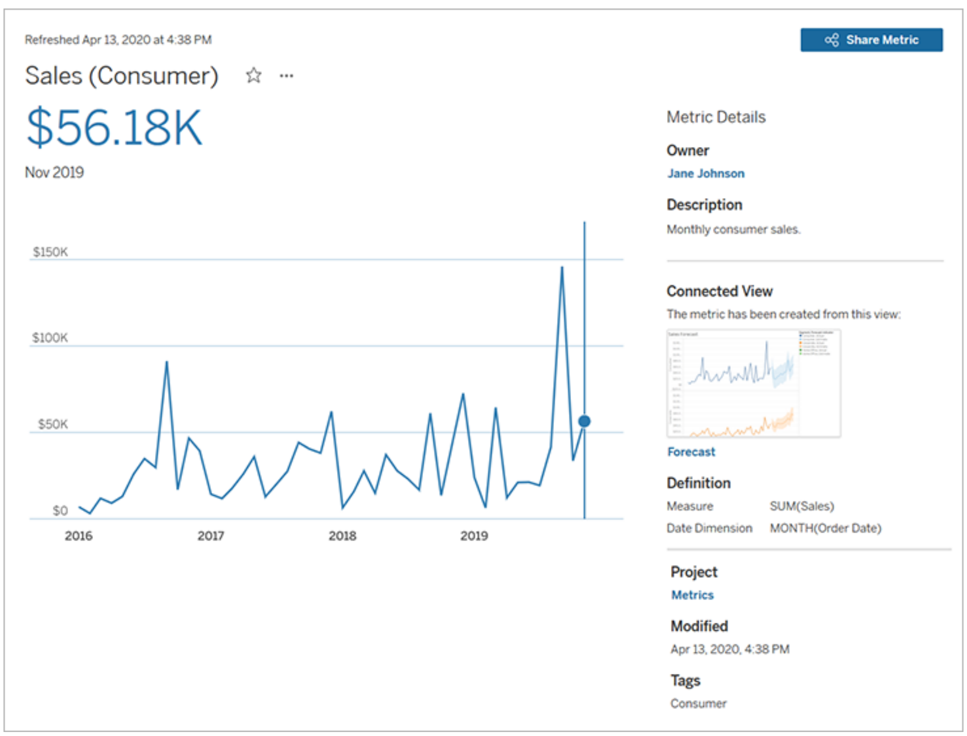


Tip Views you have recently visited appear on your **Recents** page.

For more information about working with views, see **What Can I Do with a Tableau Web View?** on page 3173.

Metrics

A metric page shows the current value for the metric, the time the metric last refreshed, and details about the metric, such as the owner, description, and definition. If the metric has a date dimension as part of its definition, it will show a timeline that displays historical values for the metric.

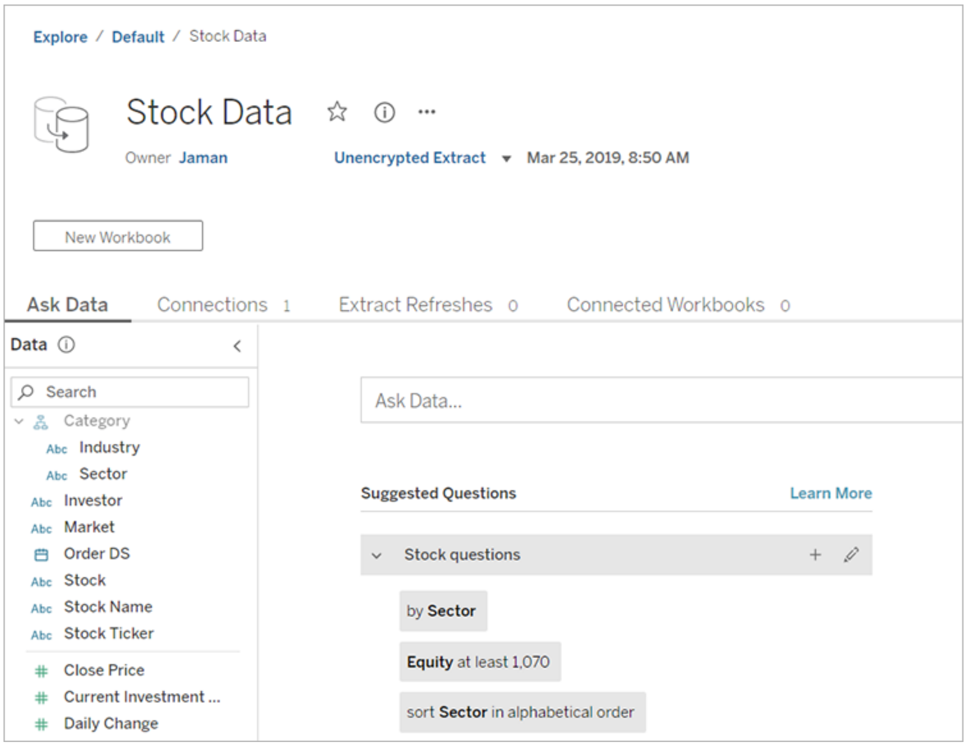


For more information about metrics, see **Create and Troubleshoot Metrics** on page 3152.

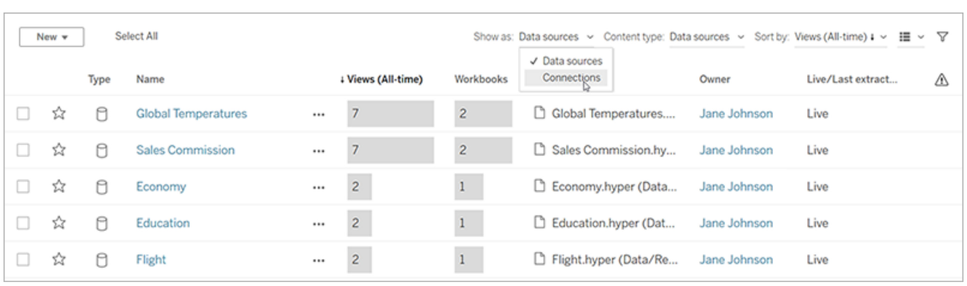
Data sources

A data source page shows connections to databases or file-based data, as well as the workbooks connected to the data source. If the data source is an extract, the page shows scheduled extract refreshes.

You can use **Ask Data** to investigate your data right from the data source page. Type a question and instantly get a visualization as an answer. For more information, see **Automatically Build Views with Ask Data** on page 1109.

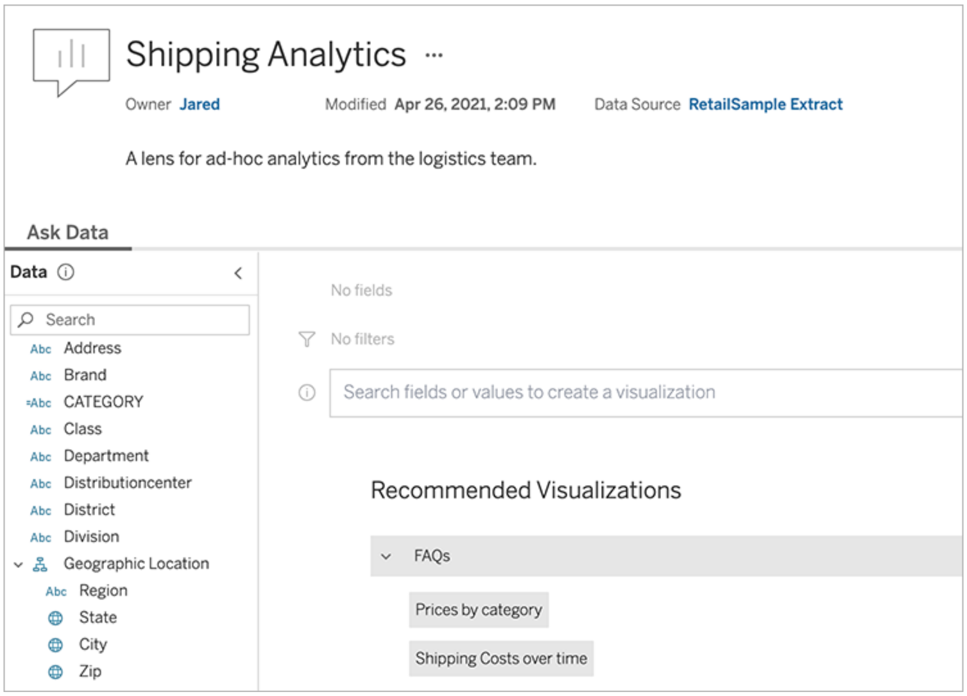


When viewing a list of data sources, to see connection and authentication details, from the **Show As** menu, select **Connections**.



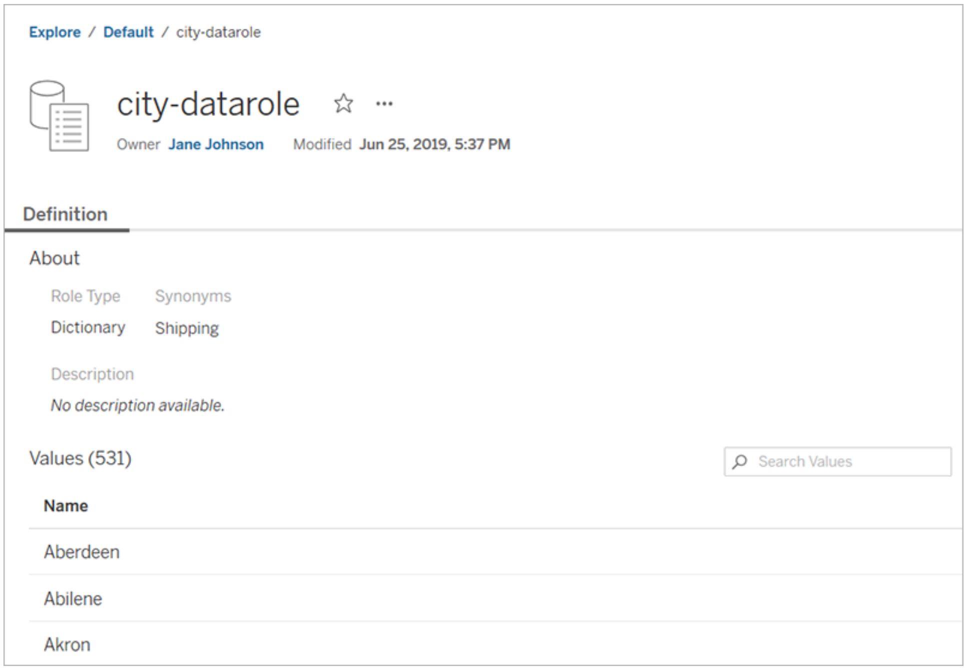
Ask Data lenses

An Ask Data lens provides a curated subset of data from a data source, making it easier for you to find relevant answers to your questions. For more information, see **Create Lenses that Focus Ask Data for Specific Audiences** on page 1126.



Data roles

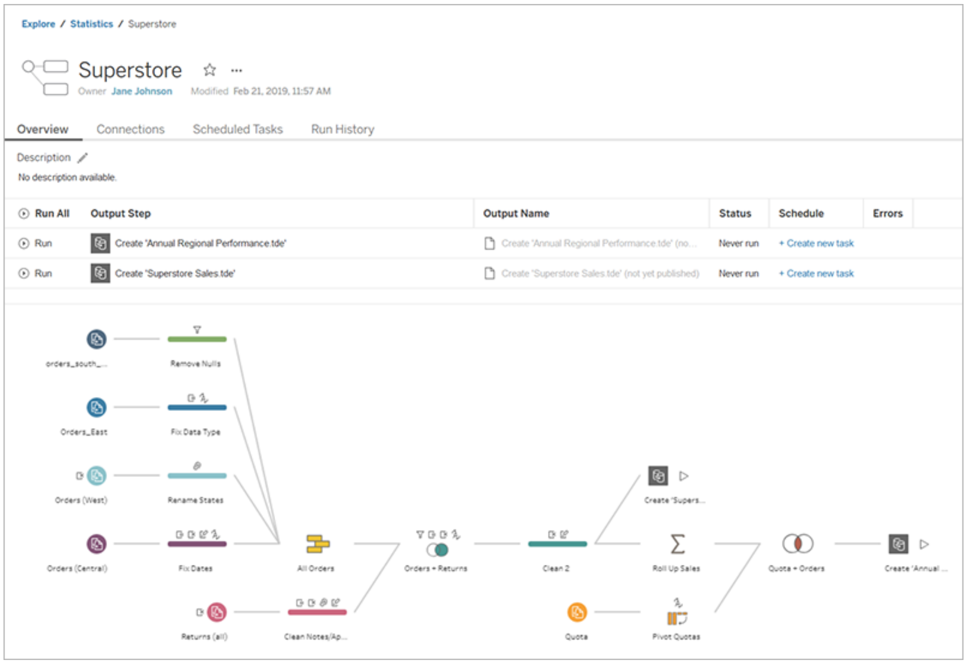
A data role page shows custom data roles published to your site or server. You can create custom data roles in Tableau Prep Builder using the field values in your data set and publish them to Tableau Server or Tableau Online to share with others. You can use data roles to validate field values when cleaning your data.



For more information about creating and using data roles to clean data, see Use Data Roles to Validate Your Data in the Tableau Prep online help.

Flows

A flow page allows you to schedule flows to run automatically to update flow output data. To see and work with flows, you must have the Data Management Add-on, and Tableau Prep Conductor must be enabled.



For more information about working with flows, see Keep Flow Data Fresh in the Tableau Prep help.

Access important content quickly

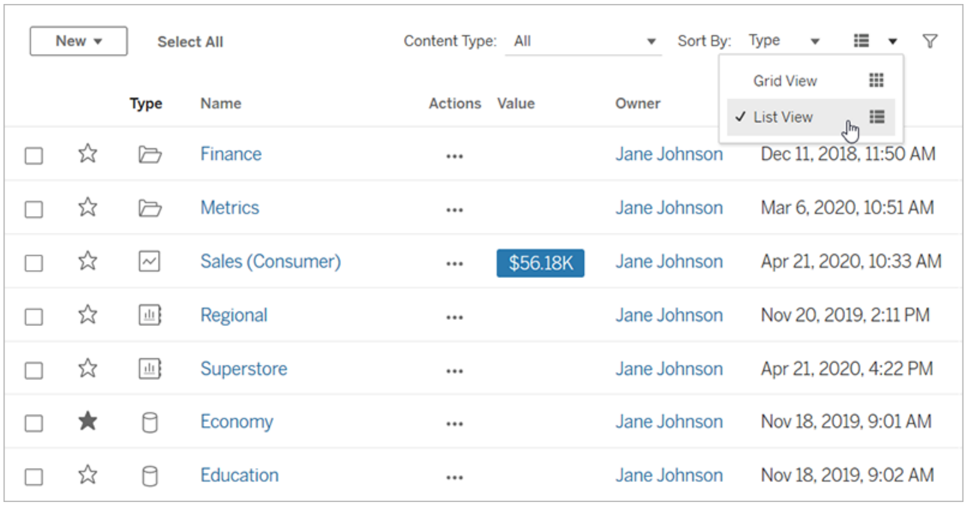
In addition to browsing content in projects, take advantage of the following areas of your site to keep track of the items that are important to you:

- **Favorites** - Add any type of content to your favorites.
- **Recents** - See the 24 views and metrics you have most recently accessed.
- **Shared with Me** - Find items that other users have sent you.
- **Collections** - Create your own collection of related items, or access collections made by other users. For more information, see **Organize Items in a Collection** on page 3199.

View content as list or grid

You can display content as a list or as a grid of thumbnails by clicking the **View As** icons.

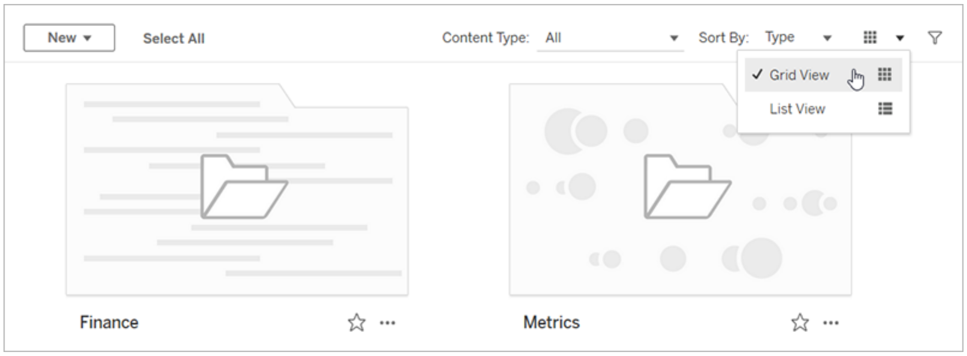
Here's the list view:



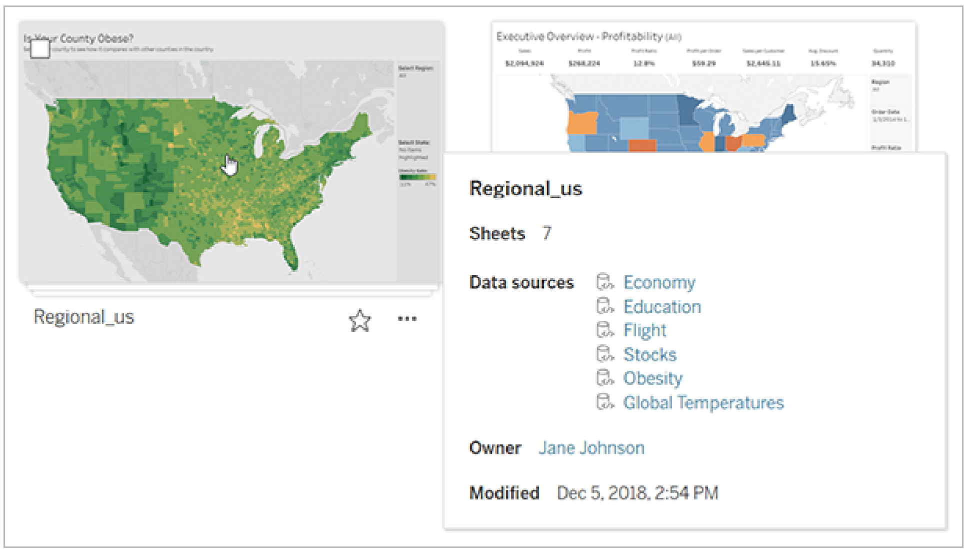
In list view:

- indicates a project
- indicates a workbook
- indicates a view
- indicates a metric
- indicates an Ask Data lens
- indicates a data source
- indicates a data role
- indicates a flow

And here's the grid view:



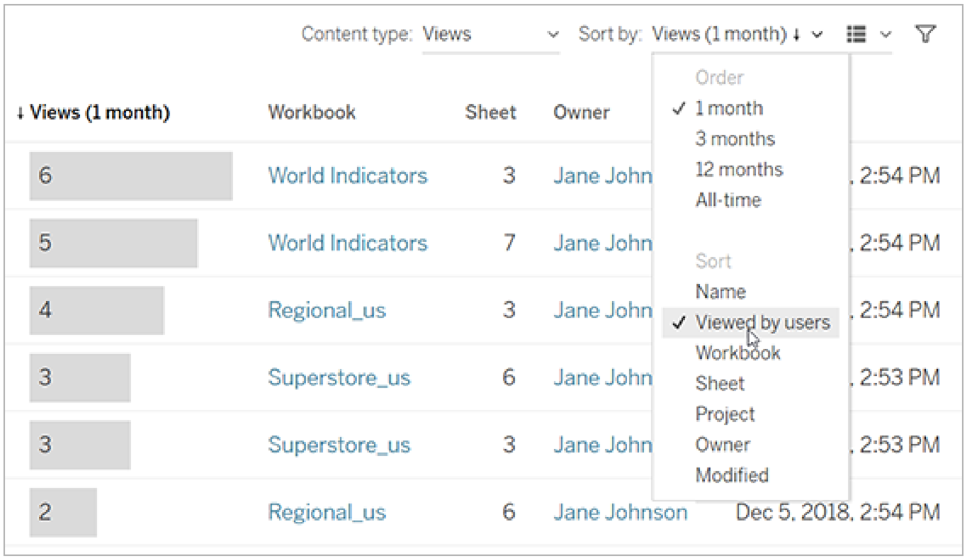
In grid view, you can hover over thumbnails to see details about the content.



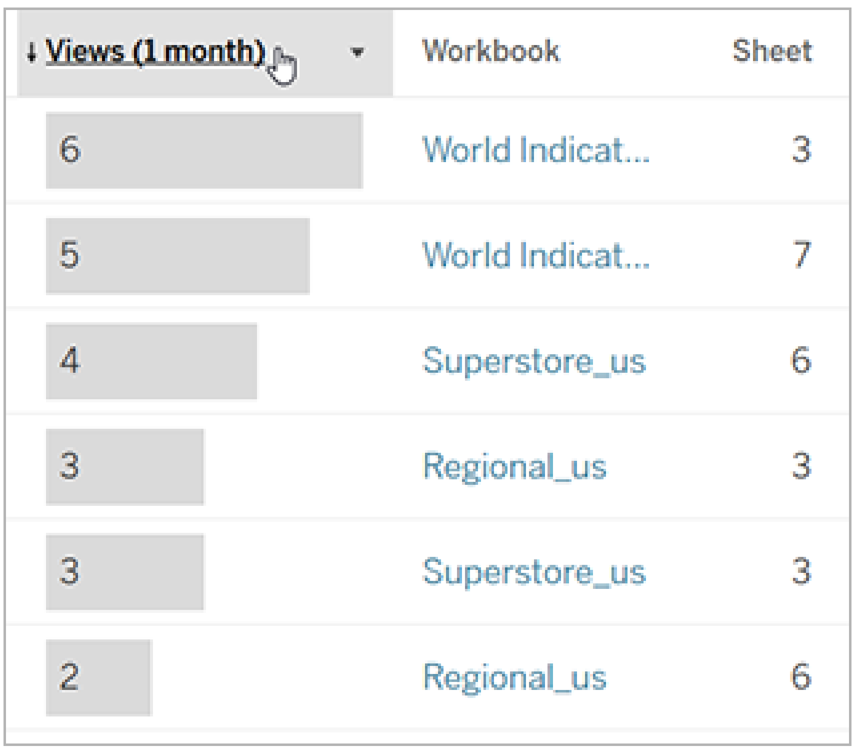
Sort content

To sort items, click the **Sort By** drop-down menu. You can sort by characteristics such as name, owner, number of views, and creation date.

This example shows the sort options for number of page views.



You can also sort a list by clicking the column heading. An arrow indicates the order: ascending (up arrow) or descending (down arrow).



Identify and navigate sites

Each site on Tableau Server and Tableau Online has a name and an ID.

On Tableau Server, if only one site exists, that site is named **Default**. When you're signed in to the **Default** site, the browser URL looks something like this:

https://server-name/#/home

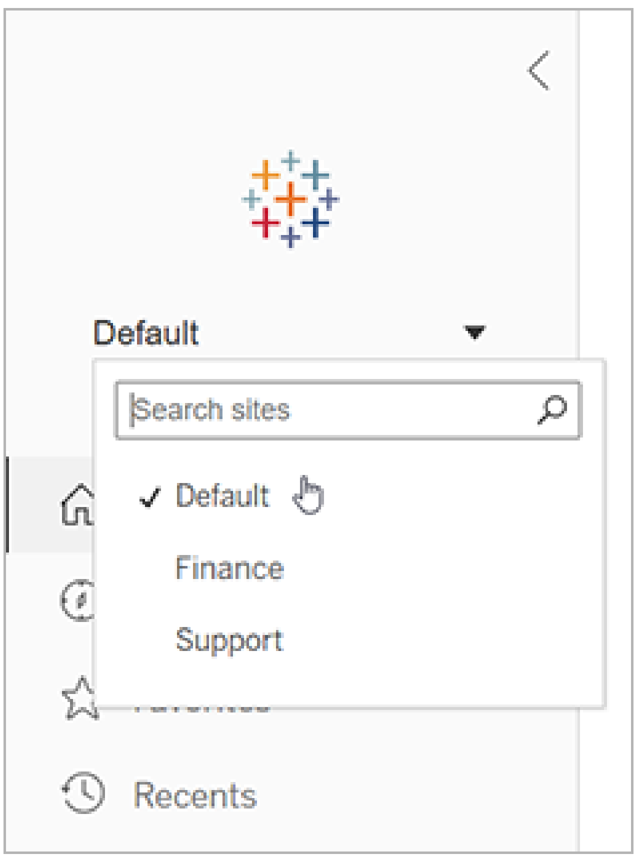
If you have access to multiple sites, you select one when you sign in.

On Tableau Online or an instance of Tableau Server running multiple sites, the browser URL includes **#/site/** followed by the site ID. If the URL doesn't include **/site**, then you're signed in to the **Default** site.

The following URL shows an on-premises Tableau Server site whose site ID is **finance**:

https://localhost/#/site/finance/home

Once signed in, you can navigate to another site by selecting it from the site picker. If you have access to only one site, you will not see the site picker.



Creators: Get Started with Web Authoring

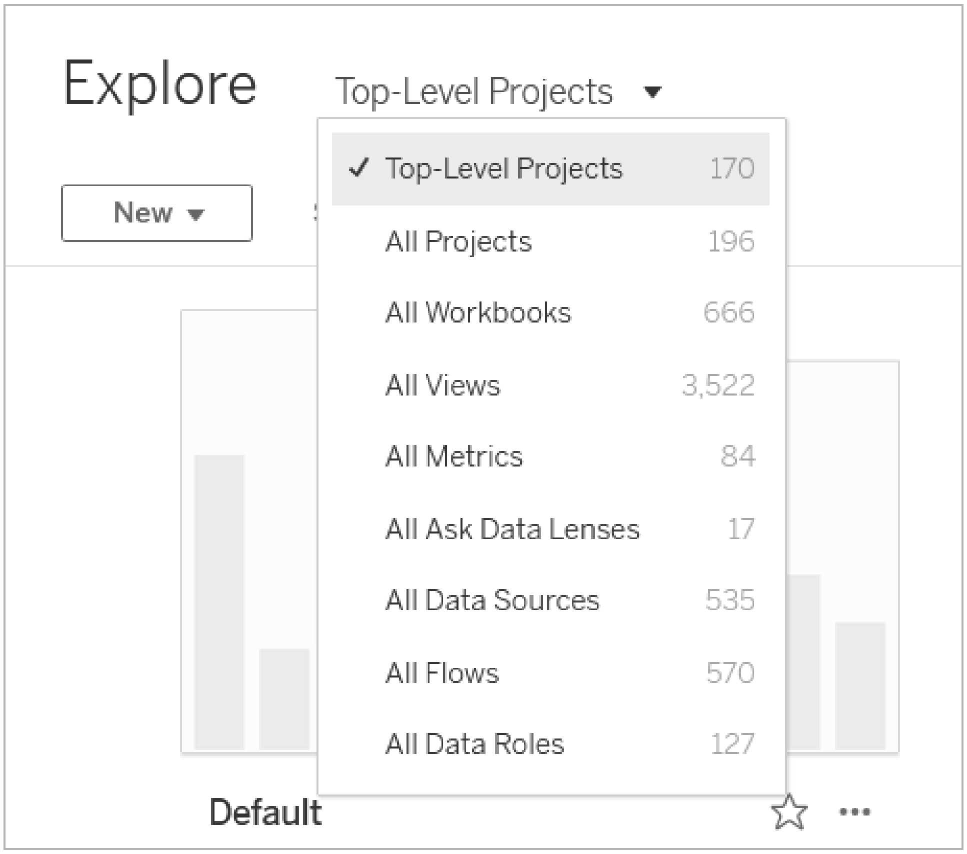
There's a lot you can do with Tableau on the web. As a Creator, you can connect to data directly to power your analytics, create stunning data visualizations, and share those insights with others. In this guide, we'll cover the Tableau Online environment, connecting to and preparing your data for analysis, creating a workbook and sharing those insights with others.

Not sure what your user level is? To check, select the icon in the top right corner of the site with your initials or profile picture and select **My Account Settings**. **Site Role** tells you if you are a a Creator, Explorer, or Viewer.

Step 1: Sign in and explore projects

To get started, enter your user name and password to sign in to your Tableau site. Then, from the navigation pane, click **Explore** to see all the content you have access to across the site.

Initially, you will see **Top-Level Projects**. You can browse these projects or select other content types.



Workbooks, data sources, data roles, and flows belong to projects, and projects can also contain nested projects. Much like folders on your desktop, projects are a way to sort and manage content in your site. If you've been invited to an existing site, you may see projects created by your teammates.

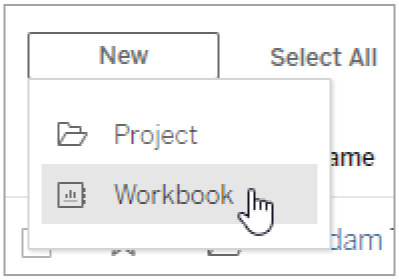
Your site comes with a default project, and you can use this project for your explorations if additional projects are not available.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to navigating and connecting to data from Explore, you can navigate and connect to more kinds of data, like databases and tables, from External Assets. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

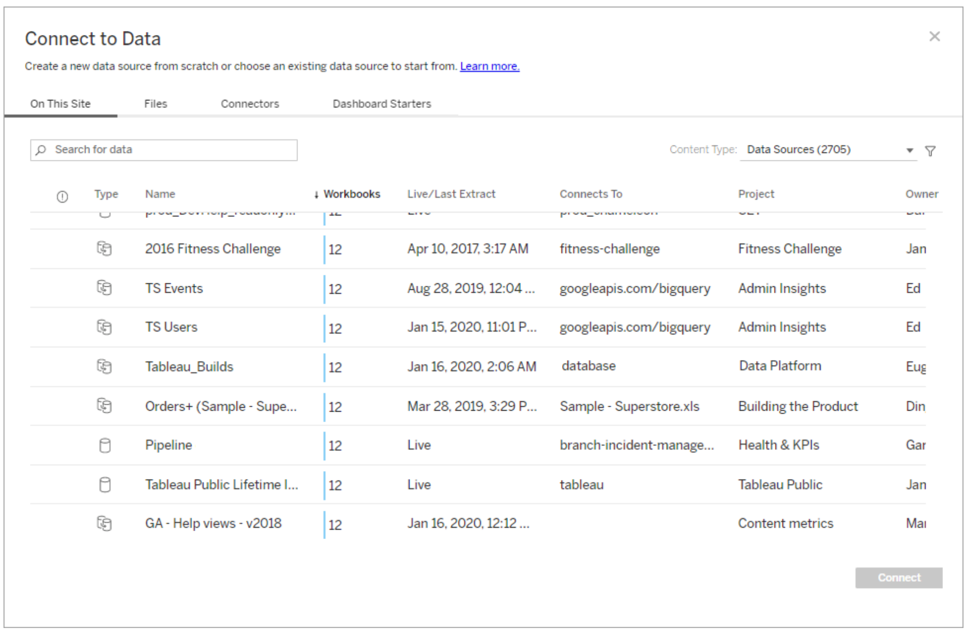
Step 2: Create a new workbook and connect to data

In some way, you'll need to connect to data in order to analyze it and build a workbook. As a Creator, Tableau lets you connect to data sources directly on the web, or you can connect to data sources published to Tableau Online through Tableau Desktop.

To begin, click the **New** button and select **Workbook**.



The **Connect to Data** window appears.



You can connect to the data that will power your workbook in several ways:

- Connect to published data sources from the **On This Site** tab.

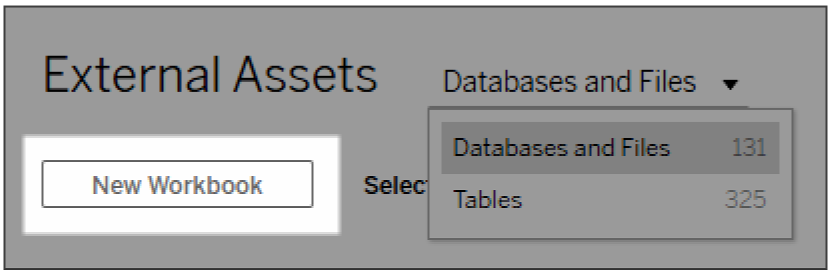
  When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On This Site** to connect to databases and tables, as well as data sources.
- Upload Excel or text-based data sources from the **Files** tab.
- Connect to server or cloud data sources from the **Connectors** tab.
- On Tableau Online, quickly author and analyze data from LinkedIn Sales Navigator, Oracle Eloqua, Salesforce, ServiceNow ITSM, and Marketo from the **Dashboard**

Starters tab.

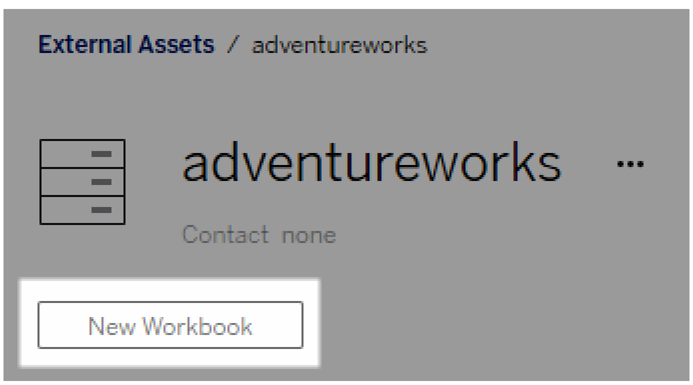
To learn more about connecting to data on the web, see Creators: Connect to Data on the Web.

Tableau Catalog users create workbooks from external assets

When you have the Data Management Add-on with Tableau Catalog enabled, you can create a new workbook when you're making a Catalog-supported connection from the External Assets page:



And from database and table pages:

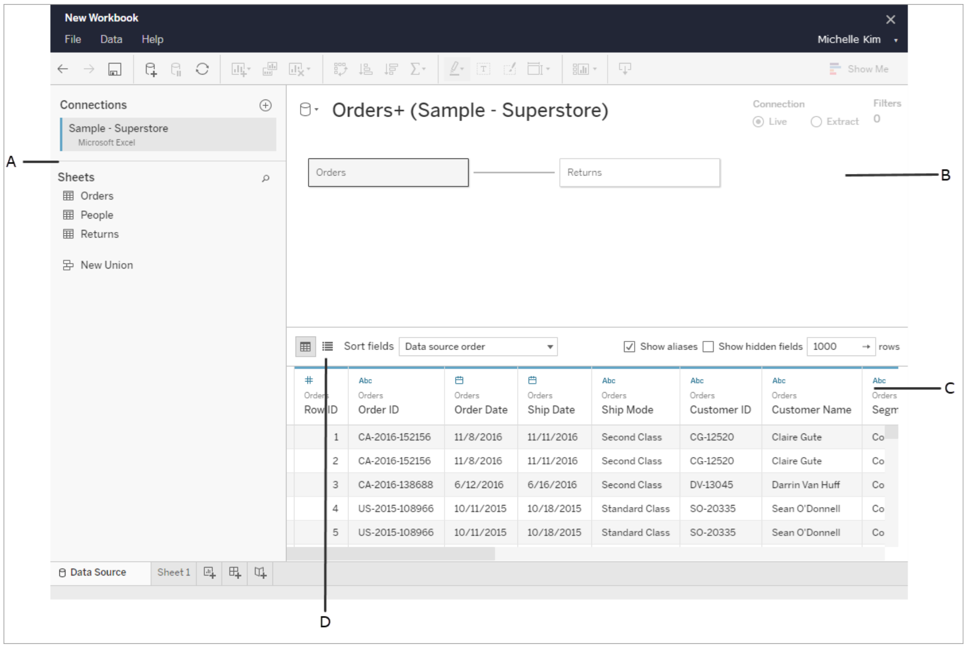


If Catalog doesn't support the connection, **New Workbook** doesn't appear.

Step 3: Prepare your data

After you have connected to data, use the data source page to prepare your data for analysis.

Parts of the data source page



There are four areas of the data source page:

A. **Left pane**: Displays details about your data. For file-based data, the left pane might display the file name and the worksheets in the file. For relational-based data, the left pane might display the server, the database or schema, and the tables in the database.

B. **Canvas**: When connected to most relational and file-based data, you can drag one or more tables to the canvas area to set up your Tableau data source for analysis.

C. **Data grid**: You can use the data grid to review the fields and the first 1,000 rows of the data contained in the Tableau data source.

D. **Metadata grid**: The button takes you to the metadata grid, which displays the fields in your data source as rows so you can look at its structure.

You can prepare your data for analysis in the data source page by:

- Renaming your data source
- Cleaning your data using the Data Interpreter
- Setting text file options

- Relating your data
- Editing data columns

To learn more about preparing your data for analysis on the web, see Creators: Prepare your Data on the Web.

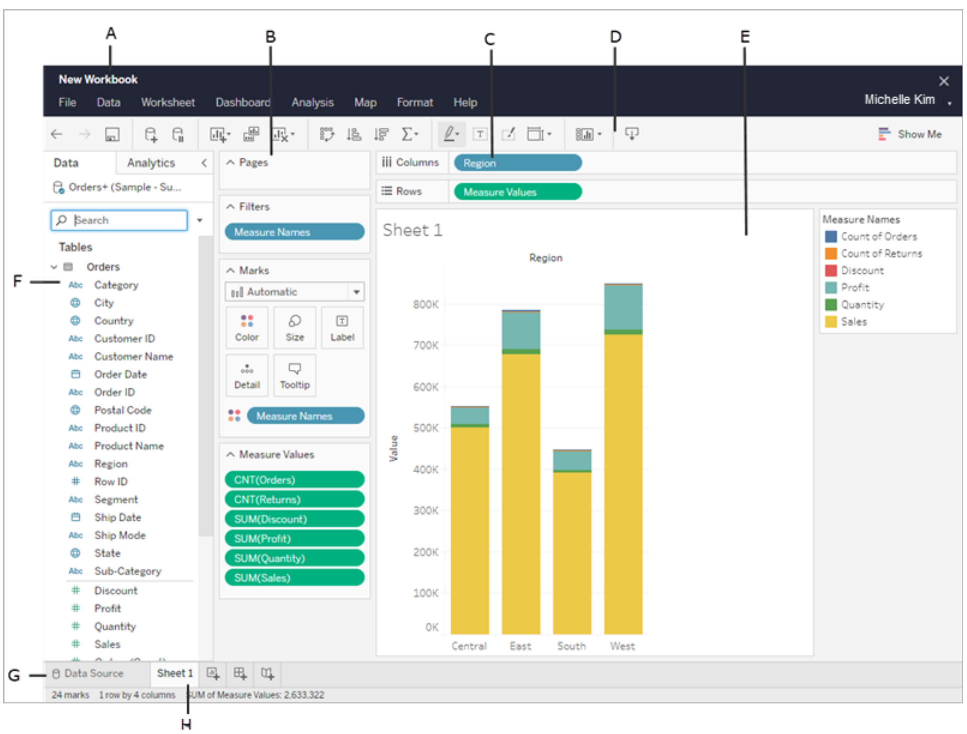
Once your data is ready for analysis, click on **Sheet 1** at the bottom of the screen to be taken to the Tableau workspace to build a view.

Step 4: Author a view

You've connected to data, you've set it up for analysis, and now you're ready to create a visual analysis of your data. Tableau calls this a **view**. If you've used Tableau Desktop before, the workspace area is very similar.

Workspace area

This is the layout of the Tableau workspace:



A. **Workbook name**. A workbook contains sheets. A sheet can be a worksheet, a dashboard, or a story. For more information, see Workbooks and Sheets.

B. **Pages shelf, Filters shelf, Marks card** - Drag fields to the cards and shelves in the workspace to add data to your view. For more information, see Shelves and Cards.

C. **Columns and Rows shelf** - Drag fields to the cards and shelves in the workspace to add data to your view. For more information, see Shelves and Cards.

D. **Toolbar** - Use the toolbar to access commands and analysis and navigation tools. See the Tableau Toolbar Button reference.

E. **View** - This is the workspace where you create your data visualizations.

F. **Side Bar** - In a worksheet, the side bar area contains the Data pane and the Analytics pane.

G. Go to the **data source** page. For more information, see Creators: Preparing Data on the Web.

H. **Sheet tabs** - Tabs represent each sheet in your workbook. This can include worksheets, dashboards, and stories.

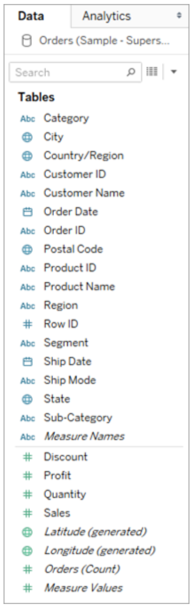
Starting in Tableau 2020.2: For each table or folder in the Data pane, a gray line separates dimensions (above line) and measures (below line). The **Dimensions** and **Measures** labels are no longer used in the Data pane.

Different ways to start building a view

Every time you drag a field into the view or onto a shelf, you are asking a question about that data. The question will vary depending on where you drag various fields, the types of fields, and the order in which you drag fields into the view.

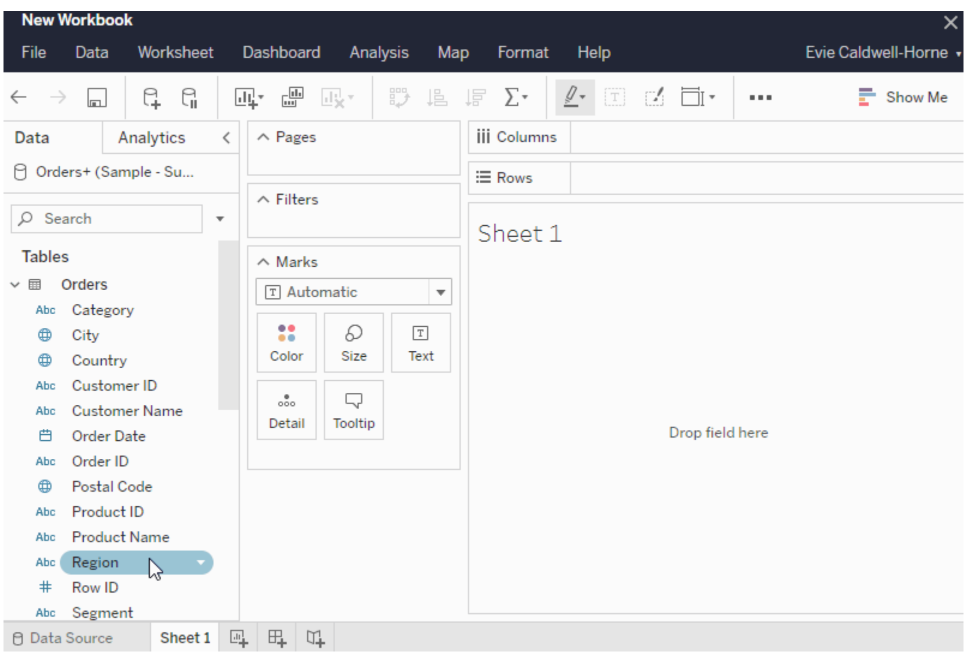
For every question you ask, the view changes to represent the answer visually - with marks (shapes, text, hierarchies, table structures, axes, color).

In the worksheet, the columns from your data source are shown as fields on the left side in the **Data** pane. The **Data** pane contains a variety of different kinds of fields, organized by table.

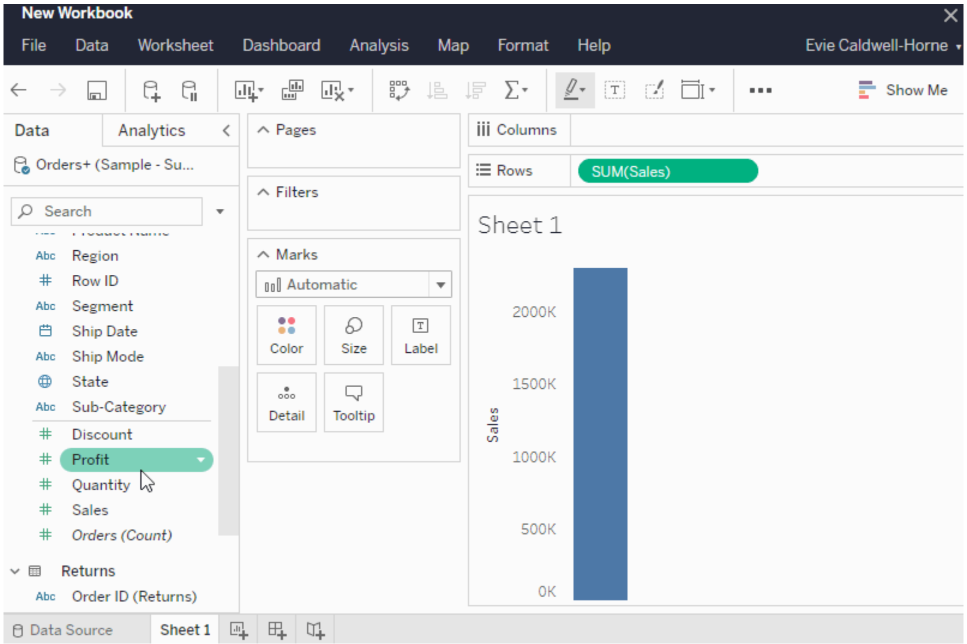


When you build a view, you add fields from the **Data** pane. You can do this in different ways. For example:

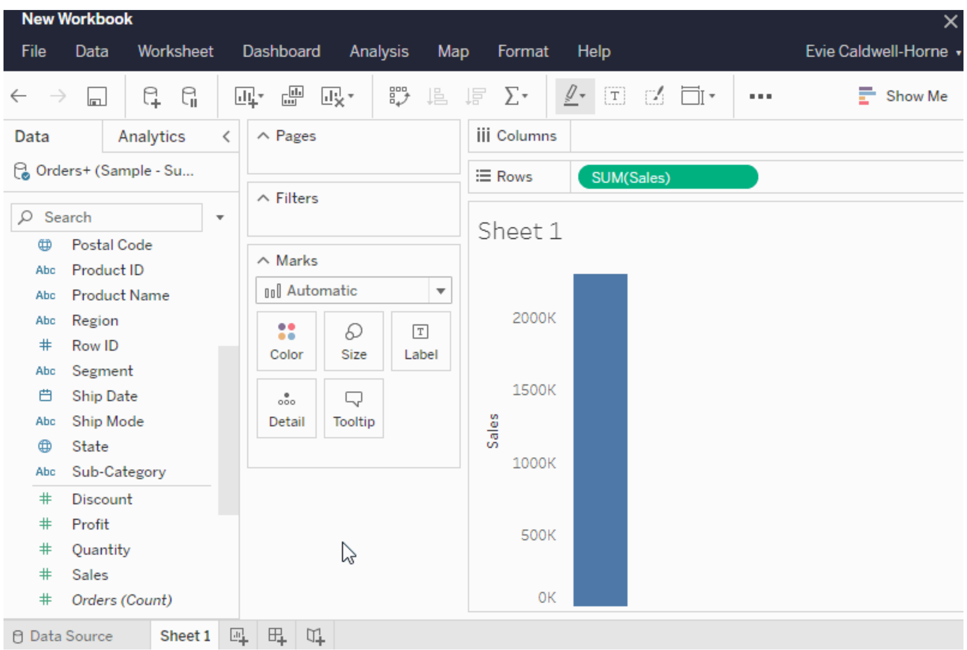
- Drag fields from the **Data** pane and drop them onto the cards and shelves that are part of every Tableau worksheet.



- Double-click one or more fields in the **Data** pane.



- Select one or more fields in the **Data** pane and then choose a chart type from **Show Me**, which identifies the chart types that are appropriate for the fields you selected. For more information, see Show Me.



Filtering data

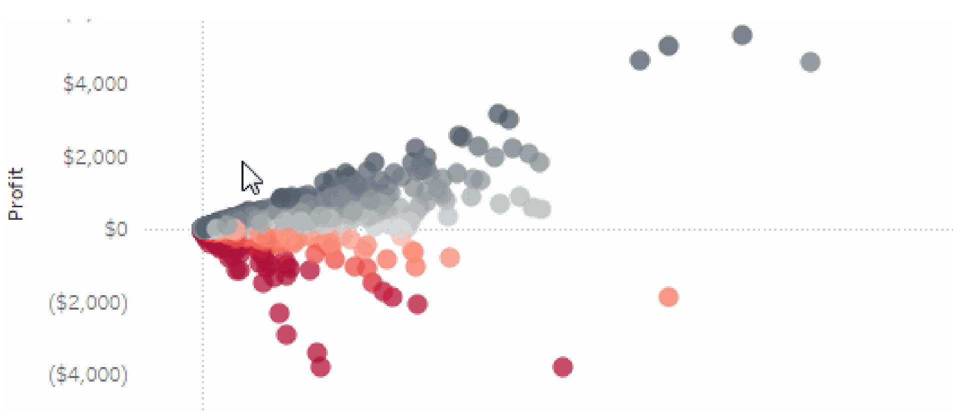
You might want to filter your data to only display a certain amount in your view. You can filter your data in a number of ways, including:

When you add fields to the filter Shelf, an interactive filter appears in the view. You can then select the items you want to include or exclude in the view.

- To create a filter in Tableau on the web, drag a dimension, measure, or date field to the filter shelf.

  When you add a field to the filters shelf, an interactive filter appears in the view. You can select items you want to include or exclude from the view, or select a range of values.

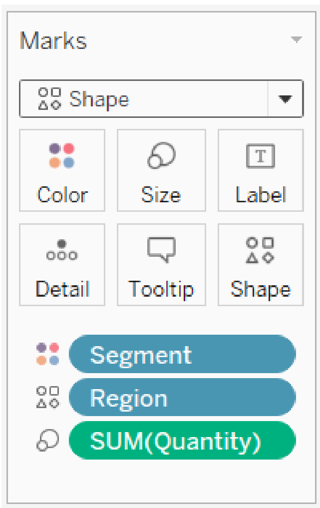
- You can also interact with a published view by selecting a single mark (data point) or click and drag in the view to select several marks. On the tooltip that appears, you can select Keep Only to keep only the selected marks in the view, or select Exclude to remove the selected marks from the view.



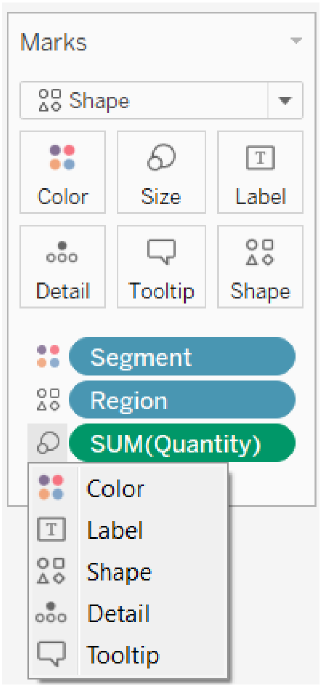
For more information about filtering in Tableau, including Tableau's order of operations, see Filter Data from Your Views.

Using the Marks card

You can use the Marks card to add context and details to the marks in your view. You use the Marks card to set the mark type, and to encode your data with color, size, shape, detail, and text.



After you add a field to the Marks card, you can click the icon next to the field to change the property it is using. You can also click the property buttons in the Marks card to change those settings.



Many properties can have multiple fields. For example, you can add multiple fields to Label, Detail, Tooltip, and Color. Size and Shape can only have one field at a time. For more information, see Mark Properties Reference.

Undo your work

Tableau is extremely flexible and very forgiving. As you build a view, if you ever take a path that isn't answering your question, you can always undo to a previous point in your exploration.

- To undo or redo, click undo ← or redo → on the toolbar.

You can undo all the way back to the last time you opened the workbook. You can undo or redo an unlimited number of times.

Build and edit views on the web

There are many ways to explore data in Tableau. For instructions on building specific chart types on the web, see **Edit Tableau Views on the Web** on page 3149, or for more features available to help you build a visualization in Tableau, see Build and Explore Data Views.

For a 6-minute walkthrough on building data views, see the Getting Started with Visual Analytics free training video. Use your tableau.com account to sign in.

Step 5: Present your work

There are several ways to tell a story or persuade others with the data insights you've found in your views.

Format your work

You can format almost everything you see on a workbook, including parts of a view, your workbook's fonts, and graph lines.

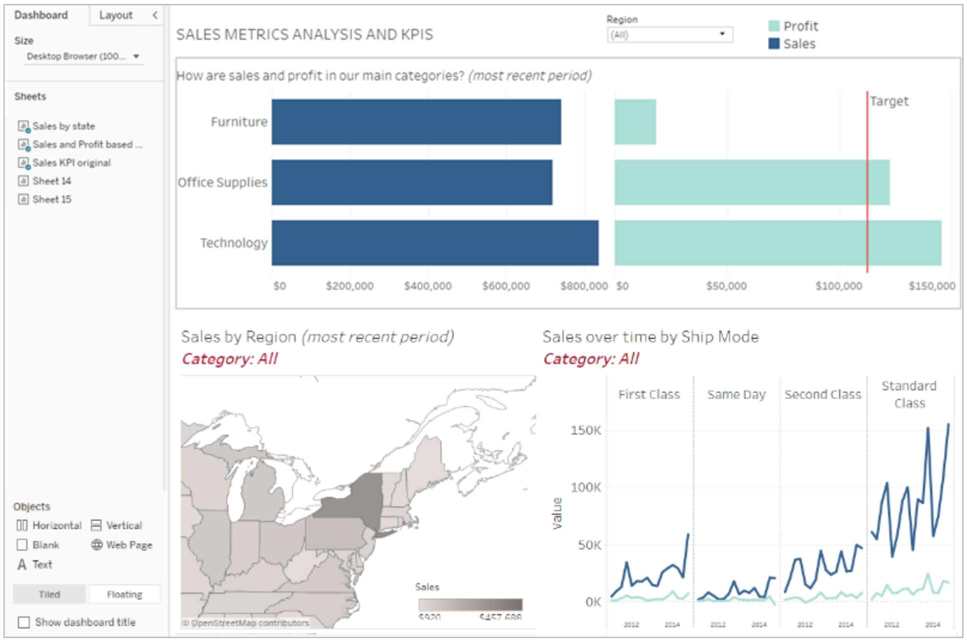
- Learn about visual best practices
- Format parts of a view
- Create Custom Color Palettes
- Format Text and Numbers
- Format at the workbook level

Dashboards

A **dashboard** is a collection of several worksheets and supporting information shown in a single space so you can compare and monitor different data simultaneously.

To open a new dashboard sheet and start creating a dashboard, click the New Dashboard icon at the bottom of the workbook:

The Dashboard area appears on the left and lists the sheets in your workbook. You can add one or more views to a dashboard, add objects like web pages, blank space, and layout containers, or interactivity to associate different views on your sheet.

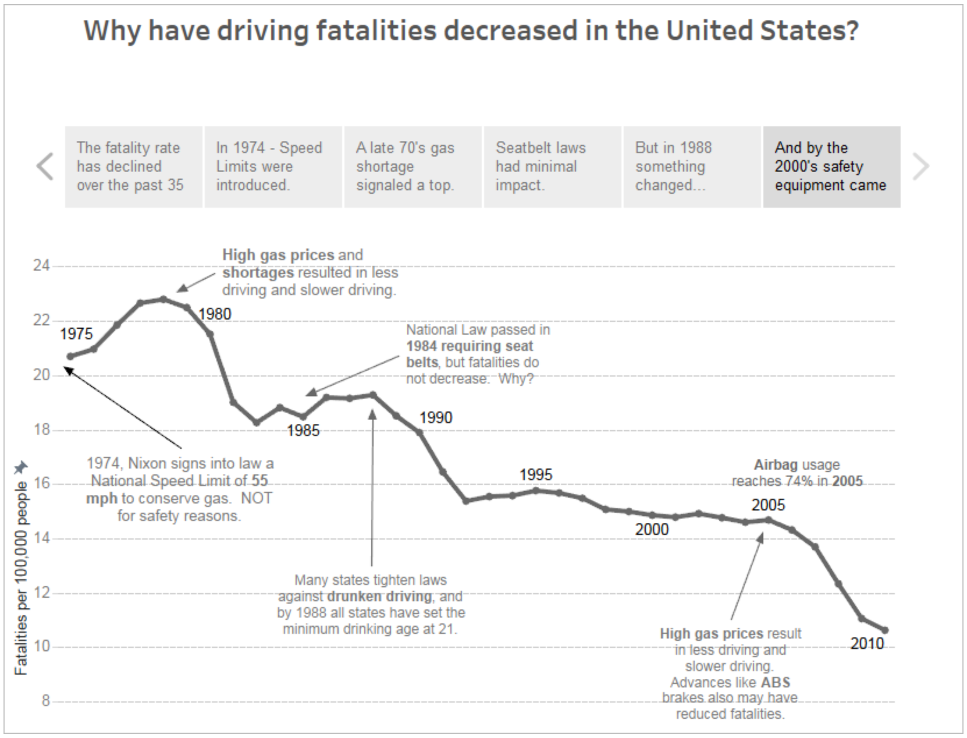


- Learn more about dashboards
- Watch a 6-minute walkthrough on building a dashboard
- See stunning dashboard examples in Tableau Public
- Learn about tips and tricks from dashboard experts

Stories

A **story** is a sheet than contains a sequence of worksheets and dashboards that work together to convey information. You can use stories to make a compelling case by showing how facts are connected, and how decisions create stories to tell a data narrative, provide context, demonstrate how decisions relate to outcomes, or make a compelling case with your data.

Create a story or learn more about stories.

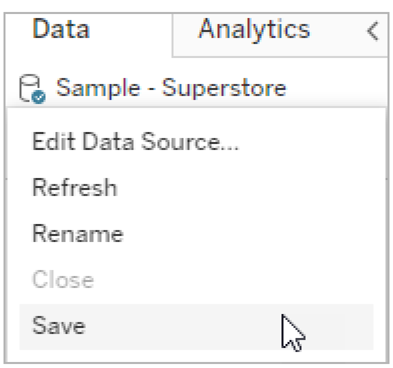


Step 6: Save your work

You can save your workbook at any time by selecting **File > Save As**. Your workbook will save into the folder it was created.

You can also save your data source to create another workbook later, or enable members of your team to use that data for their own analysis.

To save a data source, hover over the data source name in the Data tab in the workspace until an arrow appears. Select the arrow and select **Save**.



> **Note**: When you save a data source on the web, it will appear as a published data source in the **Connect to Data** window as well as in the **Data Sources** section.

When you're finished, select **File > Close** to exit the workspace.

Step 7: Collaborate and share your findings

There are a number of ways to share your work and stay up-to-date.

Collaborate

Other Creators or Explorers in your organization can edit or update your workbooks and views if they have been given access to your projects. Access is controlled by your site's administrator, who can set project permissions, move content between projects, and change a user's role. If you've saved your data source, those users can connect to your data and use it to create new workbooks.

Share Views and Insights

Every view or workbook saved to your site can be shared via links in email and other applications, or can be embedded directly into web pages, wikis, and web applications.

To see a view, users must have permission to access it in Tableau Server or Tableau Online.

Learn how to:

- Share web views (link to a view or embed a view).
- Subscribe to Views to receive a snapshot of it by email at regular intervals.
- Create Data-Driven Alerts to receive email notifications when data reaches important thresholds.
- Comment on views to share a conversation with other users.
- Embed views and dashboards into webpages.
- Download views or workbooks.

Creators: Connect to data on the web

Before you can create a new workbook and build a view on the web to analyze your data on the web, you need to connect to your data. Tableau supports connecting to data sources on the web published through Tableau Desktop, or, connecting to data directly through Tableau Online, Tableau Server, or Tableau Public.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to navigating and connecting to data from Explore, you can navigate and connect to more kinds of data, like databases and tables, from Tableau Catalog. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Note: Data connections created in Tableau Server or Tableau Online are **live connections** only. If you need to use an extract for web authoring, you can publish your data source through Tableau Desktop. To publish through Tableau Desktop, see Publish Data Sources and Workbooks. However, data connections in Tableau Public are extracts only, and there is no way to publish a data source.

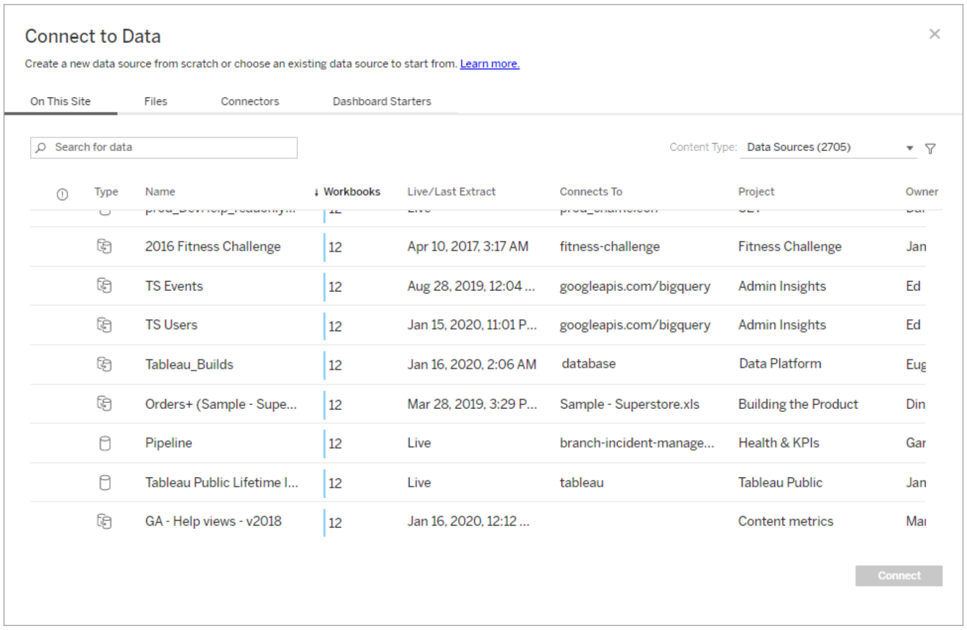
Open the Connect to Data page

On the web, you use the Connect to Data page to access data to connect to. After you sign in to Tableau Server or Tableau Online, you can open this page two ways:

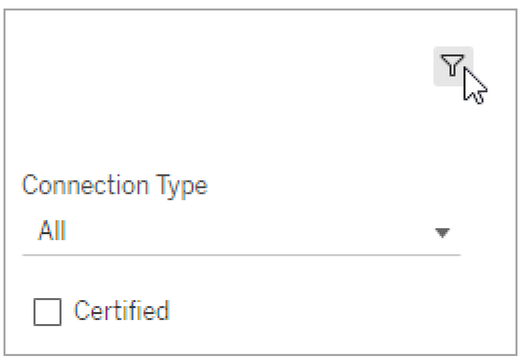
- **Home** > **New** > **Workbook**
- **Explore** > **New** > **Workbook**

If you're on Tableau Public, you can open this page from your author profile:

- **My Profile** > **Create a Viz**

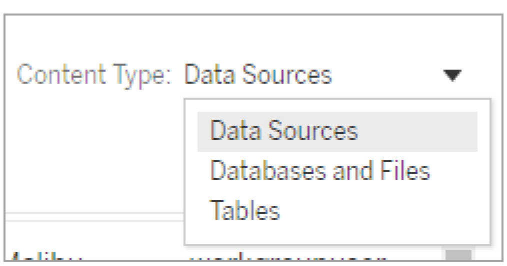


Select the filter icon to filter by Connection Type and by Certified.

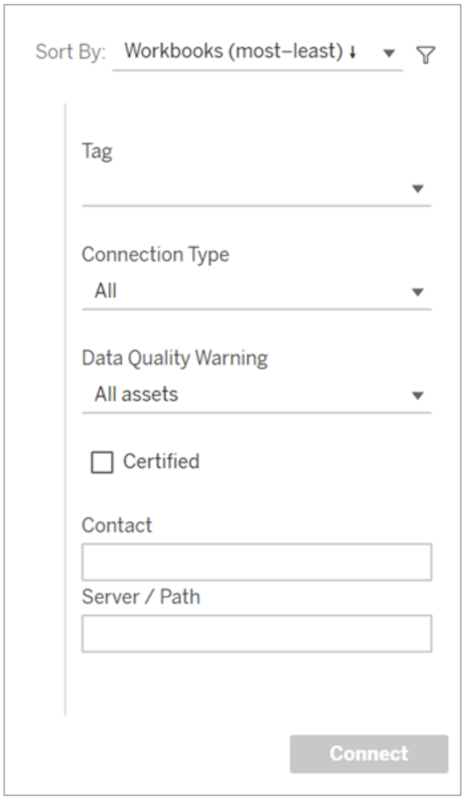


When you have Catalog enabled in your environment:

- You can select from **Data Sources**, **Databases and Files**, or **Tables** to search for data.



- When you select **Databases and Files** or **Tables**, you can filter assets by Data Quality Warning: All assets, Assets without warnings, and Assets with warnings.



On the Connect to Data page, the tabs you see depend on the product you have.

Tableau Server

On Tableau Server, select from the following tabs to connect to data: On this site, Files, and Connectors.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button.

> **Note:** When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On this site** to connect to databases and tables, as well as data sources.

Connect to files

Tableau supports uploading Excel, text-based data sources (.xlsx, .csv, .tsv), and spatial file formats that only require one file (.kml, .geojson, .topojson, .json, and Esri shapefiles and Esri File Geodatabases packaged in a .zip) directly in your browser. In the **Files** tab of the **Connect to Data** window, connect to a file by dragging and dropping it into the field or clicking **Upload from Computer**.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

Supported Connectors has information on how to connect Tableau to each of these connector types to set up your data source. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to Publish a Data Source in Tableau Desktop.

When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

Tableau Server connectors

Actian Matrix*

Alibaba AnalyticDB for MySQL

Alibaba Data Lake Analytics

Alibaba MaxCompute

Amazon Athena

Amazon Aurora for MySQL

Amazon Redshift

Apache Drill

Google Cloud SQL

Google Drive

Kognito*

Kyvos

Hortonworks Hadoop Hive

IBM BigInsights

IBM DB2

IBM PDA (Netezza)*

MapR Hadoop Hive

OData

OneDrive

Oracle

Pivotal Greenplum Database

PostgreSQL

Progress OpenEdge*

Qubole Presto

SAP Sybase ASE*

SAP Sybase IQ*

Aster Database*

Box

Cloudera Hadoop

Databricks

Denodo

Dropbox

Exasol

Google BigQuery**

MariaDB

MarkLogic*

Microsoft Azure Synapse Analytics

Microsoft SQL Server

MonetDB*

MongoDB BI Connector

MySQL

Salesforce

SharePoint Lists

SingleStore (formerly MemSQL)

Snowflake

Spark SQL

Teradata***

Vertica

*Not available on Linux servers.

**Google BigQuery requires OAuth when creating data sources from the web. Learn more about how server administrators can Set up OAuth for Google.

***Teradata web authoring currently does not support query banding functionality. See Teradata for more details.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of the data connectors that Tableau Server supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Server. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

Tableau Online

On Tableau Online, select from the following tabs to connect to data: On this site, Files, Connectors, and Dashboard Starters.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button

> **Note:** When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On this site** to connect to databases and tables, as well as data sources.

Connect to files

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data window, connect to an Excel or text file by dragging and dropping it into the field or clicking **Upload from Computer**.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

Supported Connectors has information on how to connect Tableau to your data using connectors. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to Publish a Data Source in Tableau Desktop.

> **Note**: If you’re unable to connect to your data from Tableau Online, check to see if the database is publicly accessible. Tableau Online can only connect to data that's accessible from the public internet. If your data is behind a private network, you can connect using Tableau Bridge. To learn more, see Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh.

Tableau Online Connectors

Amazon Athena

Amazon Aurora for MySQL

Amazon EMR Hadoop Hive

Amazon Redshift

Box

Cloudera Hadoop

Databricks

Google BigQuery*

Google Cloud SQL

Google Drive

Hortonworks Hadoop Hive

MapR Hadoop Hive

Microsoft Azure Synapse Analytics

Microsoft SQL Server

OneDrive*

Oracle

PostgreSQL

SharePoint Lists

SingleStore (formerly MemSQL)

Snowflake

Spark SQL

Denodo	MongoDB BI Connector	Teradata**
Dropbox*	MySQL	Vertica
Exasol	OData	

*For more information about using OAuth 2.0 standard for Google BigQuery, OneDrive, and Dropbox connections in Tableau Online, see OAuth Connections.

**Teradata web authoring currently does not support query banding functionality. See Teradata for more details.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of data connectors that Tableau Online supports. If a data source, database, file, or table is grayed out, you can't connect from Tableau Online. You can, however, connect from the Tableau Desktop **Connect** pane, if you have the correct permissions.

Use Dashboard Starters

On Tableau Online, you can author and analyze data from LinkedIn Sales Navigator, Oracle Eloqua, Salesforce, ServiceNow ITSM, and QuickBooks Online using Dashboard Starters. On the **Dashboard Starter** tab, from the list of pre-built designs, select an option and click **Use Dashboard**. For more information, see Dashboard Starters for Cloud-based Data Sources.

Tableau Public

On Tableau Public, you can connect to data by uploading a supported file.

Connect to files

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data window, connect to an Excel or text file by dragging and dropping it into the field or clicking **Upload from Computer**. If you don’t have a data set, check out the free sample data sets on the Tableau Public website.

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply your sign-in information.

Supported Connectors has information on how to connect Tableau to your data using connectors. If the connector you need doesn't appear in the Connectors tab, you can connect to data through Tableau Desktop and create an extract.

> **Note**: If you’re unable to connect to your data from Tableau Public, check to see if the database is publicly accessible. Tableau Public can only connect to data that's accessible from the public internet.

Tableau Public Connectors

Google Drive

OData

After you connect

When Tableau connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

Keep data fresh in web authoring

Update uploaded files in Tableau Online or Tableau Server: If you manually upload a file (Excel or text) for web authoring, Tableau can’t refresh the file automatically. To update your data, select “Edit Connection” to upload a new version of the file.

In Tableau Public, go to your viz and click **Request Update**. You can also keep your data fresh automatically by selecting “Keep this data in sync” in Tableau Desktop Public Edition.

Update file-based published data sources in Tableau Online: If you have a published data source in Tableau Online (published through Tableau Desktop) that uses file-based data, you can keep it fresh using Tableau Bridge. For more information, see Expand Data Freshness Options by Using Tableau Bridge.

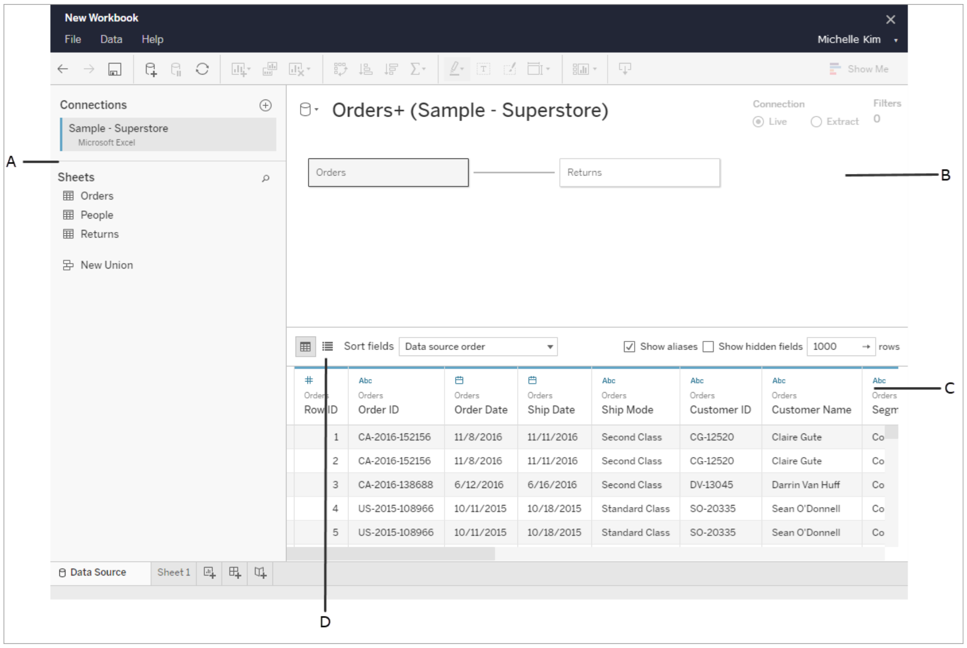
Prepare Data on the Web

Once you've connect to data in Tableau Online or Tableau Server, you can use the Data Source page to set up your data source and prepare your data for analysis. Having data that is formatted in a way Tableau likes is crucial to building a view or doing analysis in Tableau, and communicating the right information to the right people.

This article describes preparing your data in the browser with Tableau Online or Tableau Server. Although Tableau supports many of the same functionality for preparing your data for analysis in the browser as Tableau Desktop, for specific differences between the two, see **Web Authoring and Tableau Desktop Feature Comparison** on page 3330.

> **Note on keeping your data fresh**: If you connect to a published **flat file** (Excel or text), that data will not be refreshed even if it’s modified. If your data is in an **on-premises server** and is published to the web through Tableau Desktop, it will be rendered as an extract and won’t be refreshed. If you need to keep data published through Tableau Desktop fresh on the web, you can use Tableau Bridge. To learn more about these requirements, see Keep Data Fresh and **Use Bridge to Keep Data Fresh** on page 3084

Tour the Data Source page



There are four parts of the data source page:

A: Left pane

Displays details about your data connection. For example, for Excel-based data, the left pane shows your Excel workbook name, and sheets contained in that workbook. For databases, the left pane displays the server, the database or schema, and tables in the database.

B: Canvas

When connected to most relational and file-based data, you can drag one or more tables to the canvas area to join tables for analysis.

In the canvas, you can also join tables from the same or multiple data sources

C: Data grid

You can use the data grid to review or rename fields, create calculations, preview your changes, and more.

D: Metadata grid

Select the metadata grid button to see the fields in your data source as rows. This view is helpful in understanding the structure of your data source.

Set up data sources

After you connect to your data, use the Data Source page to set up the data source and prepare your data for analysis. There are many optional configurations that you can make before you begin your analysis. The configurations that you make on the Data Source page creates the data source that Tableau uses to interpret and interact with your data.

Plan the Data Source on page 741

Relate Your Data on page 766

Connect to a Custom SQL Query on page 959

Join Your Data on page 883

Union Your Data on page 942

Convert a Field to a Date Field on page 980

Pivot Data from Columns to Rows on page 988

Clean Data from Excel, CSV, PDF, and Google Sheets with Data Interpreter on page 993

Edit and sort fields in the data grid

You can make changes to your fields in the data grid. You can rename your fields, group them, create aliases, create calculated fields, change data type, and convert to a measure.

Understand Field Type Detection and Naming Improvements

Get Started with Calculations in Tableau

Convert a Field to a Date Field

Data Types

Field types: Dimensions and Measures, Blue and Green

Manage data sources

Edit Data Sources on page 1060

Refresh Data Sources on page 1071

Replace Data Sources on page 1073

Close Data Sources on page 1084

Save Data Sources on page 1075

Run Initial SQL on page 736

Publish a Data Source on the Web

You can save data sources to share or reuse from your published workbooks on Tableau Server. To do this, you'll need to have the specific permissions to edit the workbook and publish on Tableau Server. For more about publishing permissions, see Who can publish content in the Tableau Server Help.

If your data source uses a large extract, this publish step might take a few minutes.

1. On Tableau Server, open a workbook in web authoring.
2. Click the tab for any worksheet.
3. From the **Data** pane, click the drop-down menu for the data source you want to share, and click **Save as Published Data Source**.

> **Note**: After the data source is published to the web, you will not be able to edit or view the data model.

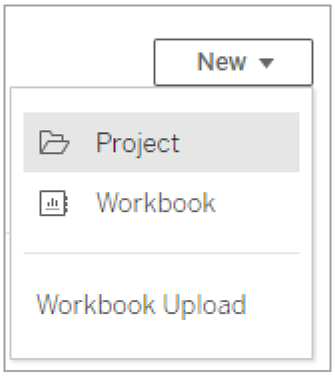
Upload Workbooks to a Tableau Site

Most workbooks you encounter in a Tableau Online or Tableau Server site are created and published to the web using Tableau Desktop, or are created directly on the web. When Desktop is unavailable, you can upload Tableau workbooks and Tableau packaged workbooks directly to your Tableau Online or Tableau Server site if you have a Creator and Explorer (can publish) site role.

Upload a new workbook to a Tableau site

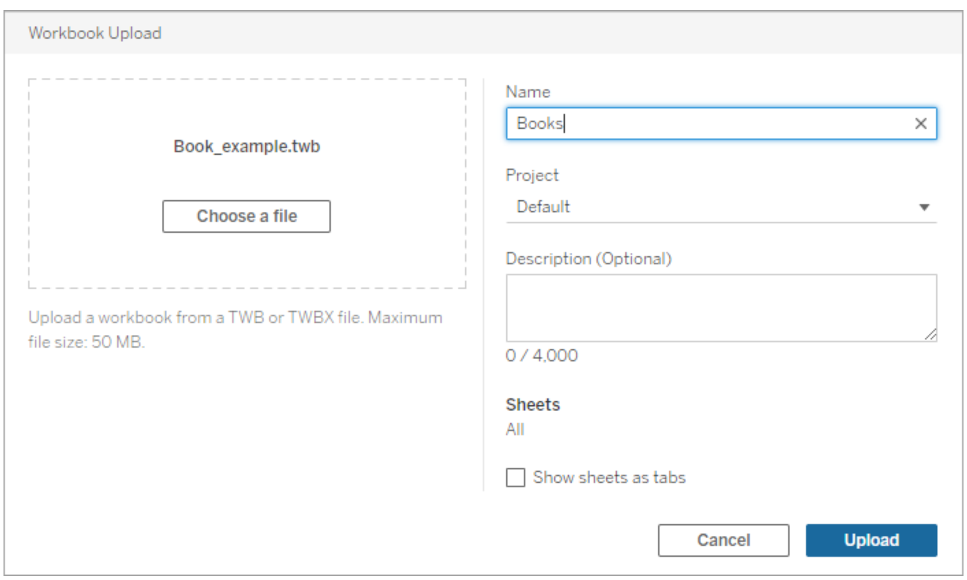
To upload a workbook:

1. Sign in to a site on Tableau Online or Tableau Server.
2. From the Home or Explore pages, select **New** > **Workbook Upload.**



3. In the dialog that opens, do either of the following:
   - Select **Choose a file** to select a .twb or .twbx file from your computer
   - Drag and drop a file into the upload area of the dialog.

   Files must be 50 MB or smaller to be uploaded.



4. In the Name field, enter a name for your workbook. By default, the workbook will retain the name of the file.

5. Choose a project where your workbook will be published, or leave as is to publish the workbook to the Default project.

   > **Note**: You'll see a warning if the project already contains a workbook with the same name. You can override an existing workbook, or update the **Name** field to something else.

6. Optionally, add a workbook description or select **Show sheets as tabs** to provide tab-based navigation.

   How sheet tabs are configured also impacts permissions. When a workbook shows sheets as tabs, workbook-level permission rules are applied to the sheets. When the sheets are not shown as tabs, any changes made to workbook permissions do not apply to the individual sheets (aka views). View-level permissions must be set independently. For more infomation, see the Set Content Permissions tab in the Permissions topic for more information.
7. Select **Upload**.

   If the workbook is overwriting an existing workbook, confirm that you'd like to overwrite, or select **Cancel** to rename.
8. When the workbook uploads successfully, you're directed to the first sheet of the workbook.

> **Note**: Uploading publishes every sheet in the workbook. To limit publishing to specific sheets, use Tableau Desktop to upload.

Permissions and data source connections

To upload a workbook, you must have permission to publish to the site and the project.

If your workbook uses an embedded data source that requires a password, you can embed credentials after you publish the workbook.

To edit the data source connection after uploading and publishing a workbook:

- Open the workbook for editing, select the Data Source tab, and select Edit Connection.
- From the Data Sources tab, select the Actions menu (...) and select Edit Connection.

Edit Tableau Views on the Web

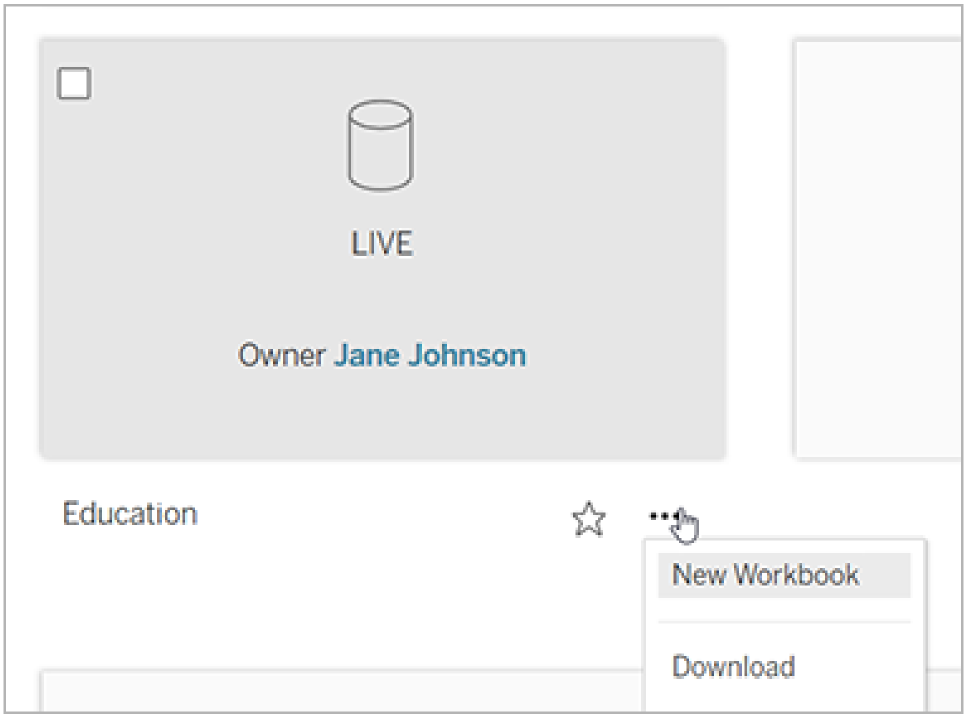
If you can see the **Edit** button when you are looking at a Tableau view, that means you can make changes to it. Depending on your access level and permissions, you can:

- Edit an existing published workbook and add worksheets for views, dashboards, and stories.
- Create and edit a new workbook based on a published data source.
- Edit an existing workbook and add worksheets in the browser, or by opening the workbook in Tableau Desktop.
- Connect to different published data sources while editing. For details, see Connect to Published Data Sources while Web Editing.

For more information about how to build different types of charts and views in Tableau, see **Build Charts and Analyze Data** on page 1109.

Create a new workbook on the web

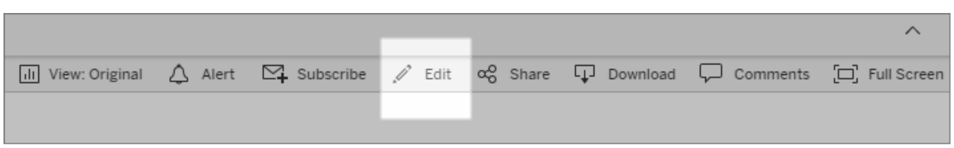
1. Sign in to a site on Tableau Online or Tableau Server.
2. Either open the project that contains the data source you want to use, or show **All Data Sources**.
3. Click the **…** next to the data source name to open the actions menu, then click **New Workbook**.



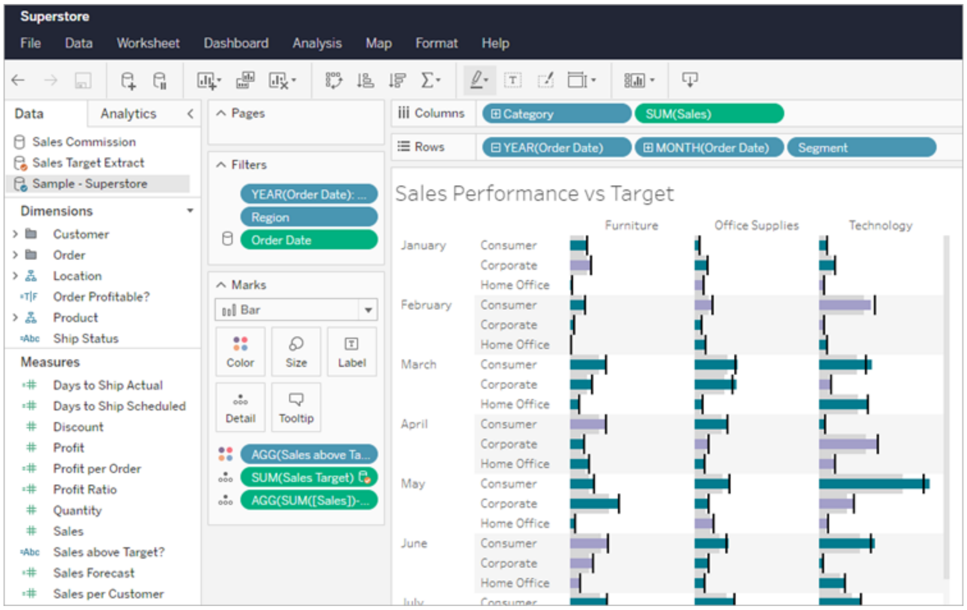
A new, blank workbook opens in the Tableau Server web editing environment.

Edit a published view

1. Sign in to a site, then either open the workbook that contains the view you want to edit, or show **All Views**.
2. Open the view.
3. Click **Edit** in the view toolbar.



When you click **Edit**, the view opens in web authoring mode.



Now you can edit the view.

> **Note:** If you If you have Tableau Desktop version 2021.2 or later, you can also open the workbook in Tableau Desktop and edit there. Select the Edit in in Desktop icon in the authoring toolbar to open, then publish the workbook to the site when you're finished.

To learn how to build a chart or dashboard, see get started, see **Step 4** in Creators: Get Started with Web Authoring and Explore and Analyze Data in a View.

4. The **Save** options available to you will vary depending on your permissions set by your Tableau site administrator.

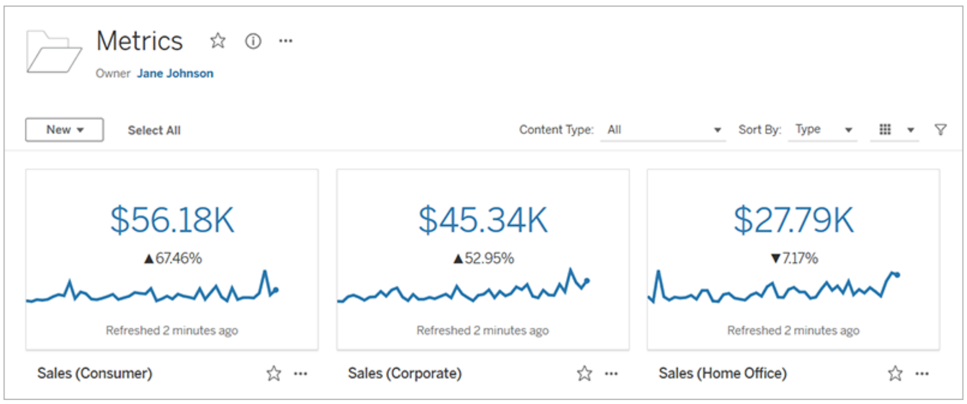
   - To close the view without saving your changes, select **File** > **Close**.
   - To save your work in the current workbook, select **File** > **Save**.
   - To save your work as a new workbook, select **File** > **Save As**.

> **Note:** if you edited a workbook in Tableau Desktop and want to publish your changes, select Server > Publish Workbook and select Publish. The workbook's location and name will be populated in the Publish dialog.

Create and Troubleshoot Metrics

Metrics help you track changes to your data. You can quickly monitor key numbers, because metrics update automatically and display their current value on the grid and list view for your content. This means that you don't need to open a metric to see your data.

If you have multiple dashboards you frequently check, create metrics for the numbers you want to monitor, then track them in one place by adding them to your favorites or creating them in the same project.



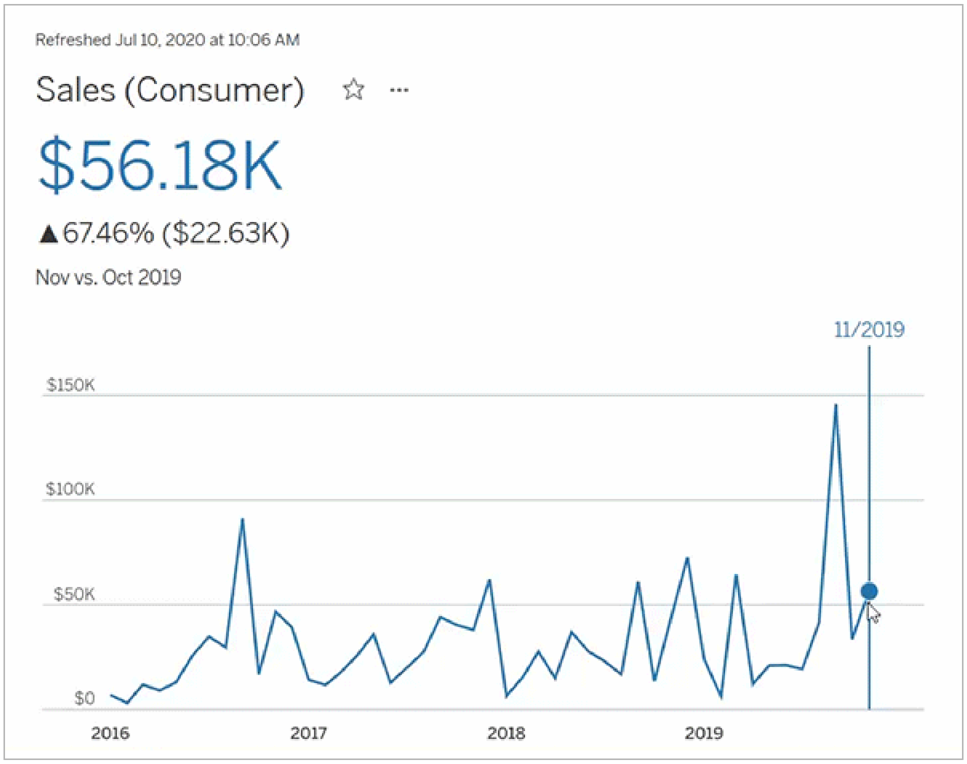
How metrics are defined

A metric's value is defined by an aggregate measure, such as sum of sales, and an optional date dimension, such as order date. For information about dimensions and measures, see Dimensions and Measures, Blue and Green. Any filters applied to the measure before the metric was created will apply to the metric value, for example, filters for segment or region.

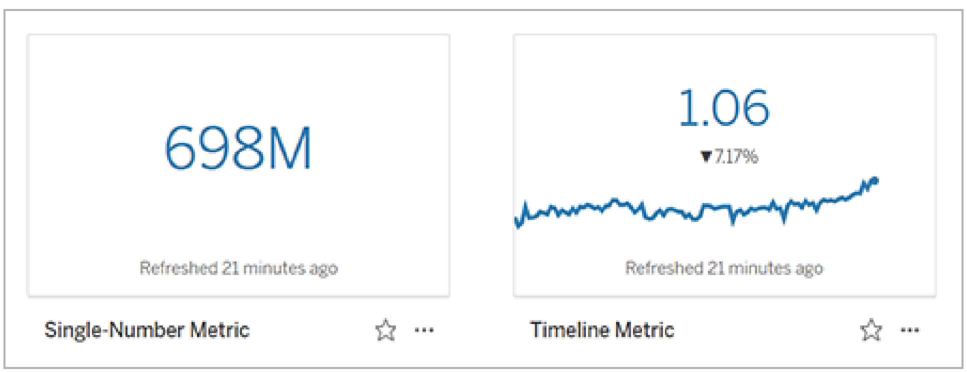
Timeline metrics and single-number metrics

A metric that has a date dimension as part of its definition will have a timeline. Because metrics are defined based on the mark you select when you create a metric, you include a timeline by selecting a mark on a chart with a date axis or header.

When you open a metric, the timeline shows the value of the measure based on the granularity of the date dimension, for example, daily sales or monthly users. Move your mouse along the timeline to see historical values.

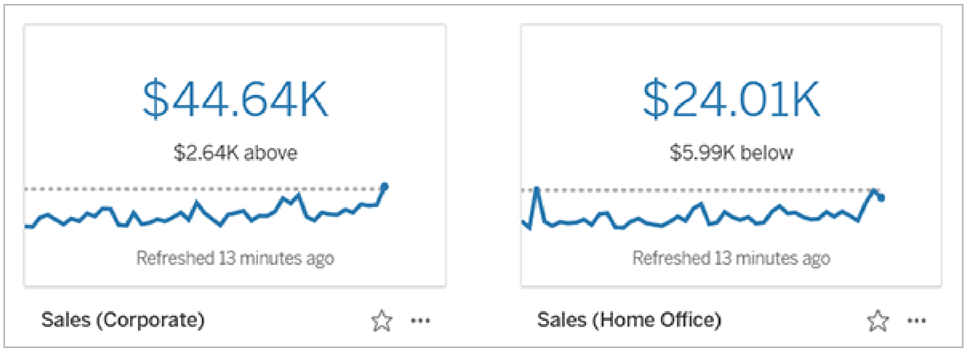


If the mark you select doesn't have a date dimension associated with it, your metric will be a single-number metric. Single-number metrics still refresh, and their values update when the data changes, but they don't show previous values on a timeline.

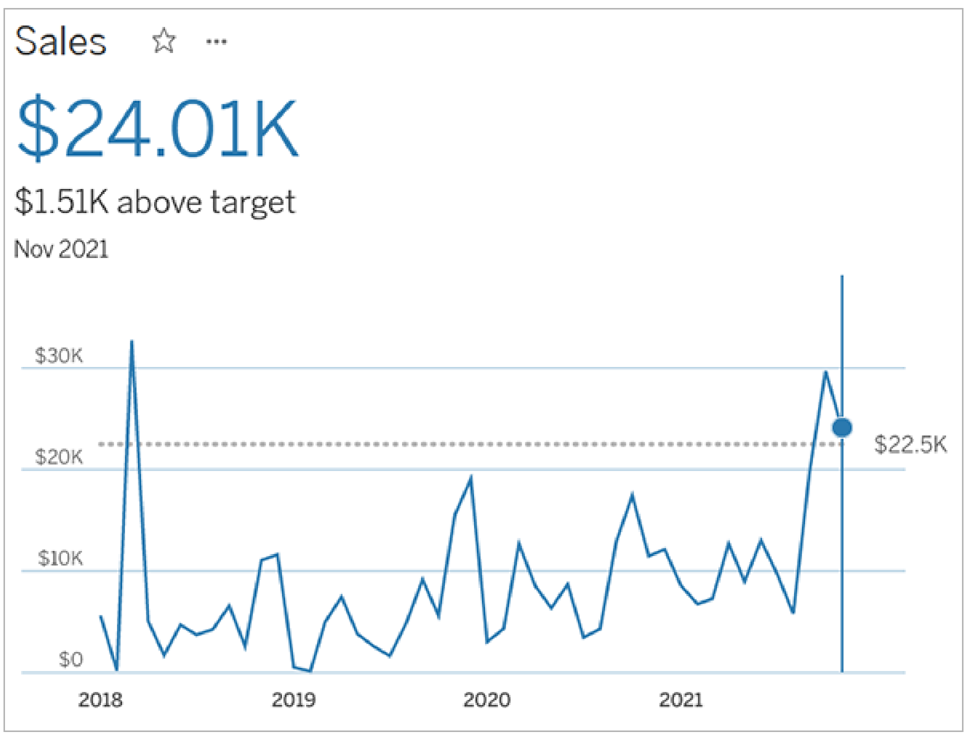


Metrics with target values

If a metric has a target value assigned to it, you will see a comparison to the target value below the current metric value, both on the metric card and on the metric details. The creator of a metric might include a target value to represent a threshold to stay above, for example, if you need to maintain a 90% on-time delivery rate. Or a target value might represent a cumulative goal you are working toward, for example, a monthly sales target.

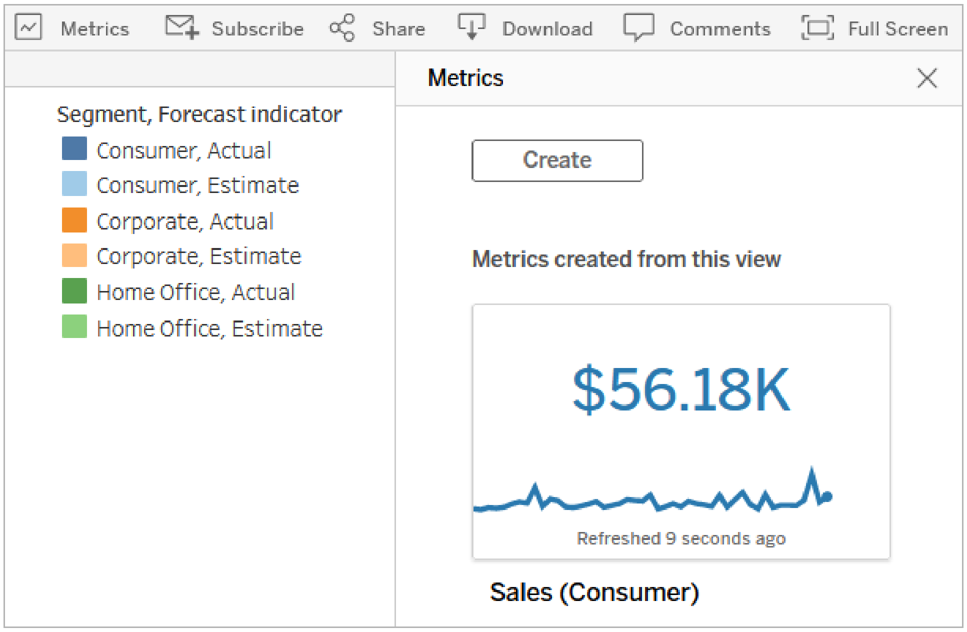


Timeline metrics display the target value as a reference line on the timeline. When a timeline metric includes a target value, the comparisons shown for each point on the timeline are to the target value, rather than to the previous point in time.

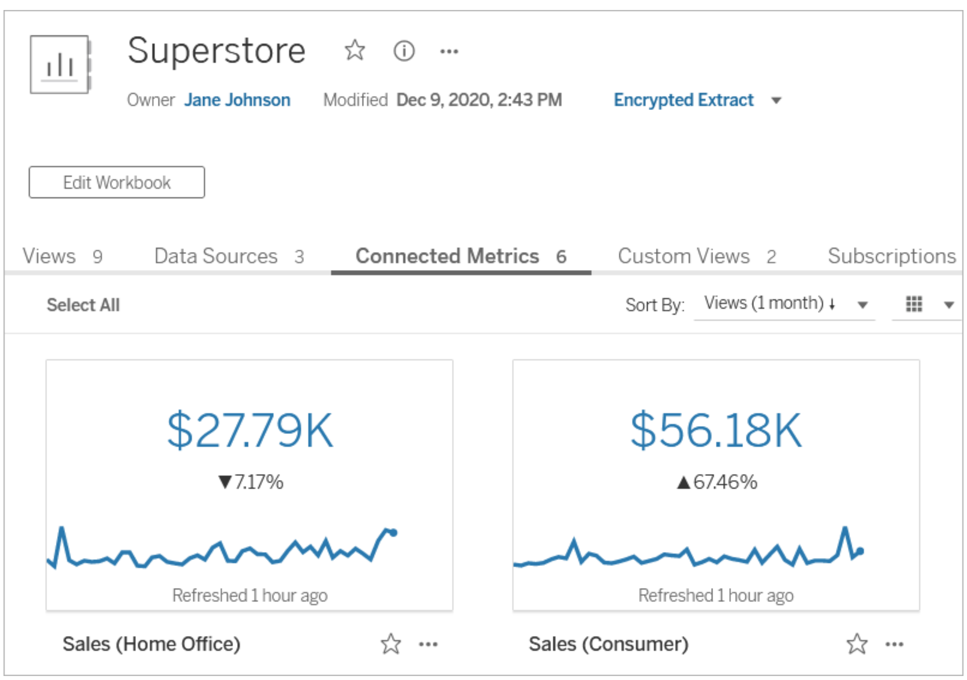


See connected metrics

Connected metrics are the metrics that were created from a particular view or workbook. To see connected metrics for a view, click the **Metrics** button in the view toolbar. The metrics displayed are ordered from the newest creation date to the oldest. Users with all site roles can see connected metrics for a view. However, the Create button will appear only if you have a site role of Creator or Explorer (can publish). For more information, see Create a metric.



To see connected metrics for a workbook, click the **Connected Metrics** tab. You can sort these metrics using the Sort By menu.

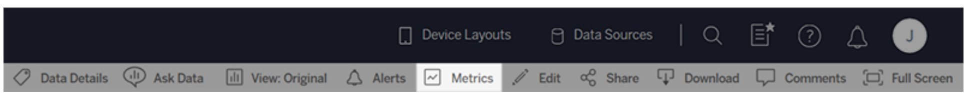


Before creating your own metric from a view, it's a good idea to check the connected metrics. Another user might have already created the metric you want. Instead of creating a duplicate metric, open the existing metric and add it as a favorite.

Create a metric

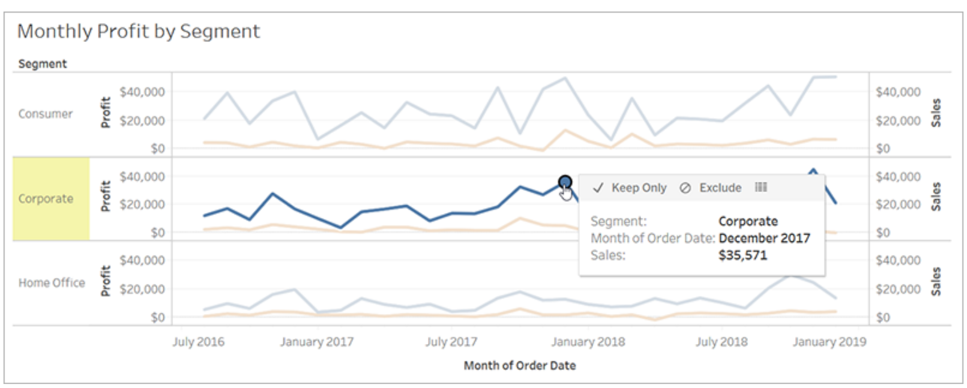
If you have a site role of Creator or Explorer (can publish), you can create metrics on Tableau Online or Tableau Server. Metrics are formatted automatically and optimized for viewing on both mobile and desktop devices.

1. Navigate to the view that you want to create a metric from.
2. On the view toolbar, click the **Metrics** button.

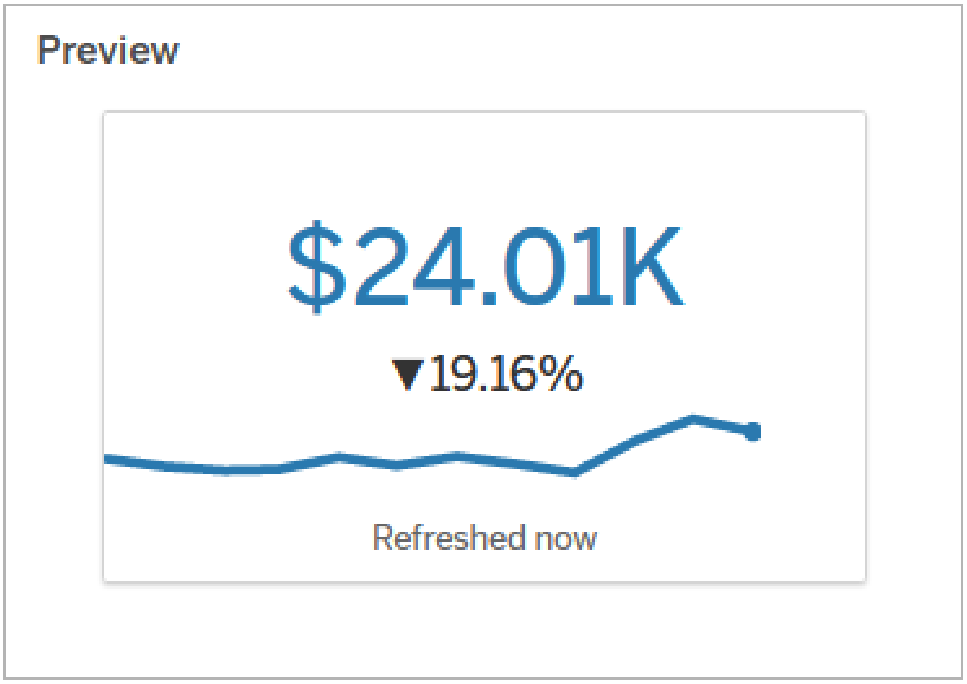
   

   The metrics pane opens.
3. If the pane shows connected metrics, select the **Create** button to enter authoring mode.
4. Select a mark. If you encounter an error, see **When you can't create a metric** on page 3158.

   The measure, and optional date dimension, associated with this mark will define your metric. Any filters you apply to this mark will be applied to your metric.

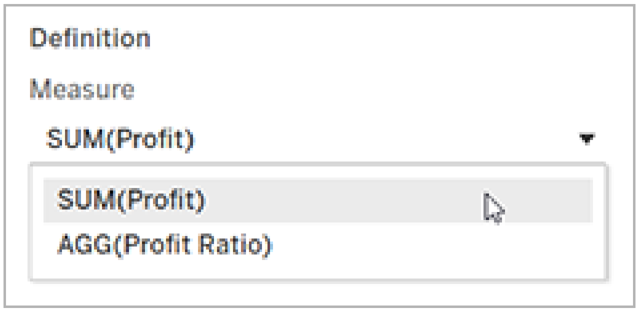
   

   The metrics pane shows a preview of your metric. The value in the preview is the most recent value for the metric, which may differ from the value of the mark you selected in the case of metrics with a date dimension.



5. The **Name** field is pre-populated based on your selection. You can give the metric a different name. A metric must have a unique name within the project it belongs to.
6. Under **Description**, enter an optional message to help others understand your metric. For example, describe filters applied to the metric or indicate the data source used by the metric.
7. Under **Target value**, enter an optional number to compare the metric against. Don't include commas or symbols in this field. If your target is a percentage, simply type the number without the percent sign, for example, enter 25 instead of 0.25 for a target of 25%.

   When you enter a valid target value, the preview updates to show how far above or below the target the current value is. Make sure to note what your target value represents in the metric description.
8. Under **Definition** > **Measure**, select the measure to use for your definition from the drop-down. This option appears only if the mark you selected has more than one measure associated with it.



9. Under **Project**, click **Change Location** to choose a different project for the metric. By default, your metric will be added to the same project that the view belongs to.
10. Click the **Create** button.

    A message appears with a link to the metric in the project you added it to.

Though metrics are created from a view, they exist independent of that view. This means you manage metrics similar to how you manage workbooks. For example, you can set permissions for a metric or move it to another project. For information about managing content on your Tableau site, see Manage Web Items.

Overwrite a metric

1. Create a metric with the same name in the same project as the metric you want to overwrite.

   The Overwrite Metric dialog appears.
2. Click the **Overwrite** button.

When you overwrite a metric, the metric continues to appear for those who have added it to their favorites, and any changes made to permissions for the previous metric will apply to the new metric.

When you can't create a metric

If you select a mark on a chart that doesn't support metrics, you will get an error message explaining why you can't create a metric. The table below summarizes these scenarios.

Reason	Scenarios
You don’t have access to the full data for the chart.	• Row level security or user filters limit the data you can see. For more information, see Restrict Access at the Data Row Level. • You don’t have permission to see the full data. For more information, see View Underlying Data.
The password for the workbook’s data source is not embedded or is no longer valid. For more information, see Set Credentials for Accessing Your Published Data.	• The workbook prompts for a password. • Credentials are passed using single sign-on (SSO).
The data isn’t at the correct level of granularity.	• The data in the chart isn't aggregated. Metrics use aggregations, such as sum or average. For more information, see Data Aggregation in Tableau. • There are multiple values per cell of data—a result of data blending. For more information, see Troubleshoot Data Blending.
The date dimension is not supported.	• The chart includes both date parts and date values. For more information, see Change Date Levels. • The date dimension uses the ISO 8601 calendar rather than the standard Gregorian calendar. For more information, see ISO-8601 Week-Based Calendar. • The date dimension is aggregated at the custom level of Month / Year or Month / Day / Year. For more information, see Custom Dates.

How metrics refresh

When a metric refreshes, it checks the connected view (the view the metric was created from) for new data. A refresh doesn't necessarily update the value of a metric, because there may be no changes to the data.

Metrics refresh at a frequency either based on an extract's refresh schedule or, for live data, every 60 minutes. The time of the last refresh is displayed on the metric.

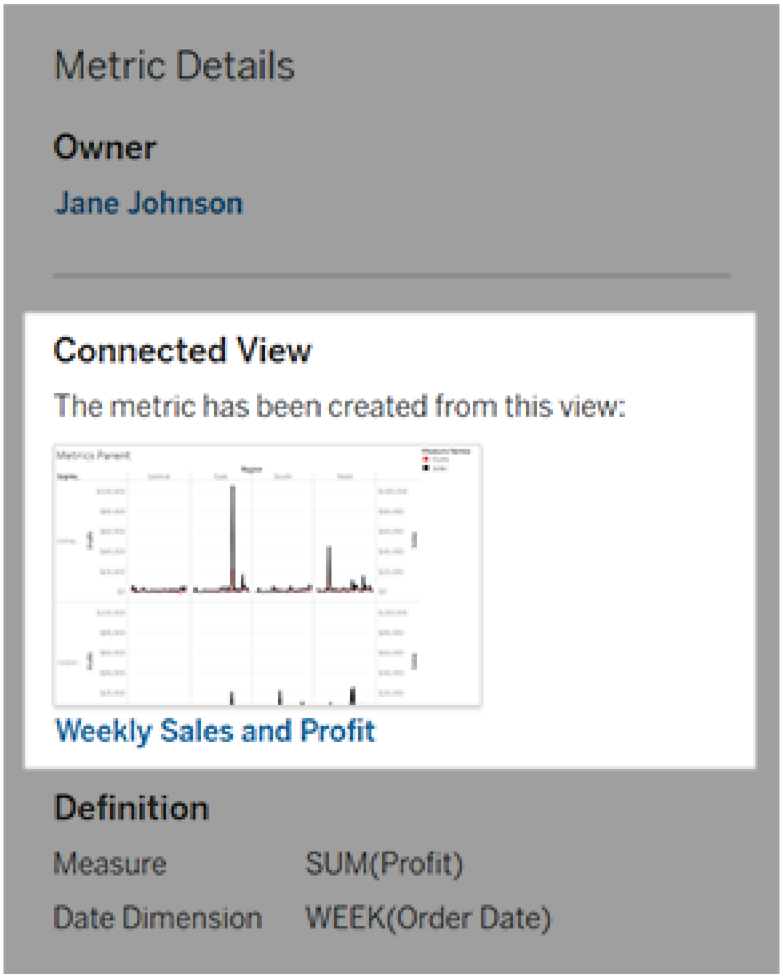
Fix failing refreshes

If a metric isn't able to access the connected view or its underlying data, the refresh will fail. If the refresh for your metric fails, you'll receive an email notification, which notes the time of the failure and the affected metric.

Metric refreshes may fail for one of the following reasons.

- The connected view was deleted or modified.
- Permissions changed for the connected view.
- The password for the data source is no longer embedded or is no longer valid.
- The metric owner doesn't have the required site role to refresh the metric. A site role of Creator or Explorer (can publish) is required.
- There was a temporary connectivity issue, which will resolve itself.

To identify the cause of the failure, look at the metric details. Make sure that the metric owner has the required site role to refresh the metric. Then inspect the **Connected View**.



If the connected view is still listed

Open the view to investigate the cause of the failure.

Evidence	Cause	Solution
The workbook prompts for a password or displays an error when connecting to the data source.	The password for the data source is not embedded or is no longer valid.	The content owner or a Tableau administrator can edit the data source connection to embed the password. For more information, see Edit Connections.
You can't view the full data, following the instructions on View	You no longer have permission to access to the full data.	The content owner or a Tableau administrator can change the permissions for the view. For more information, see Per-

Evidence	Cause	Solution
Underlying Data.		missions.
The measure or date dimension that defines the metric is no longer present.	The view has been modified so the metric can't connect to the data needed to refresh.	The content owner or a Tableau administrator can check the revision history and restore previous versions. For more information, see View Revision History

If there is no connected view listed

The view was deleted or you no longer have permission to access the view. Contact your Tableau administrator for assistance.

Resume suspended refreshes

If a refresh fails enough times, the refresh is suspended. You'll receive an email notification if the refresh for your metric is suspended.

When a metric refresh is suspended, Tableau no longer attempts to get new data for the metric. Metrics with suspended refreshes continue to present historical data.

If the cause of the failure is fixed, you can resume the refresh.

1. Open the affected metric.
2. On the warning message, click **Resume refresh**.

Tableau attempts to perform the refresh. If this attempt succeeds, you'll receive a notification, and the refresh will resume on schedule. If the attempt doesn't succeed, your refresh remains suspended.

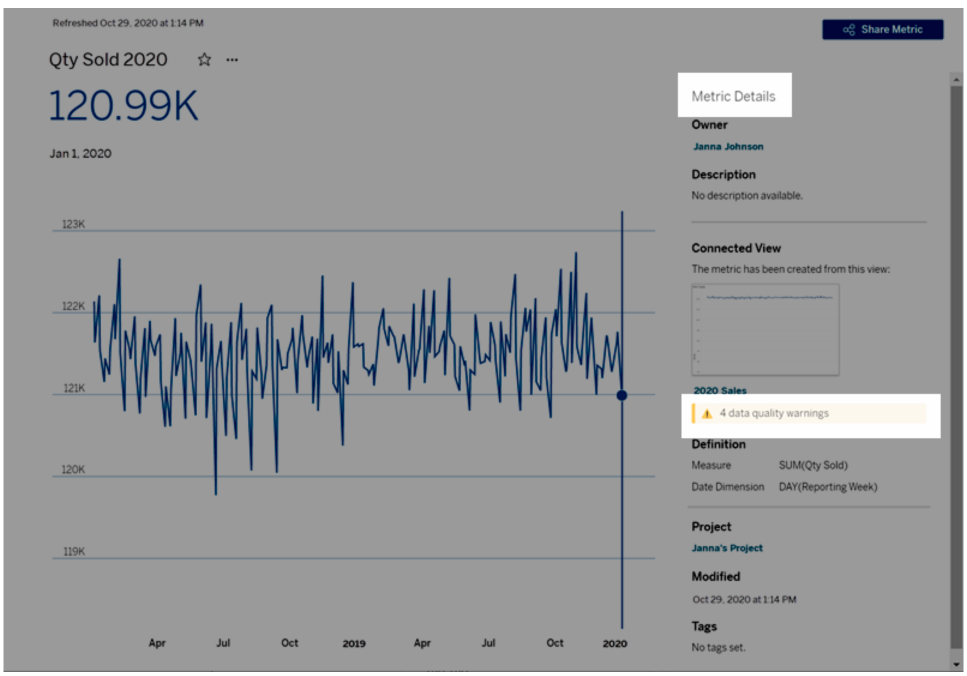
Try overwriting the metric if the connected view is still available. For more information, see Overwrite a metric. Otherwise, you can keep the metric to reference past data or delete the metric.

> **Note:** If the metric refresh is suspended because you don't have the required site role for it to refresh, you won't be able to resume the refresh or delete the metric.

Metrics Appear in Tableau Catalog

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When the Tableau Catalog is enabled in your environment, you can see metrics in the Catalog lineage tool, and metrics affected by data quality warnings display those warnings. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

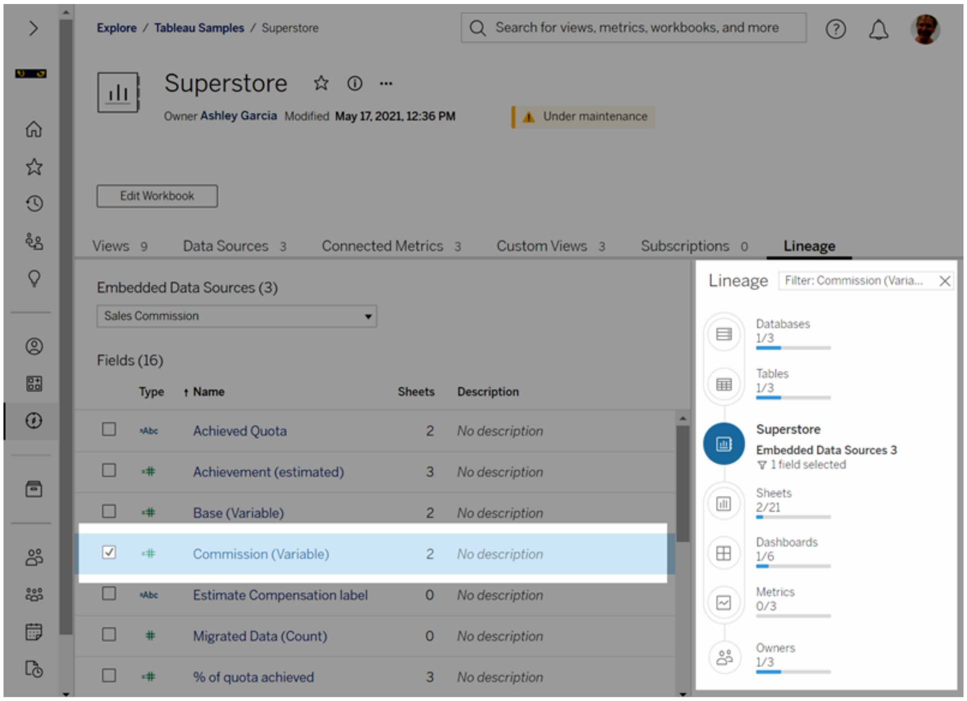
When you have metrics defined for the numbers you want to monitor, it's important to know if the data that the metrics rely on are impacted in any way. You can use Catalog in a couple of ways to know this. First, data quality warnings set on the data your metric is based on show on the metric. These warnings appear when you open a metric in Tableau Mobile, and in Tableau Server and Tableau Online when you hover over a metric in grid view and on the metric details page, as shown below:



For more information, see "Set a Data Quality Warning" in the Tableau Server or Tableau Online Help.

You can also use the lineage tool in Tableau Catalog to see the upstream sources that your metric relies on. When doing an impact analysis, you can see which metrics are affected if a certain column or a table is changed or deprecated, or if certain workbooks are removed.

Including metrics in the lineage means that Catalog gives a full picture of the impact changes can have on the assets in your environment.



For more information, see "Use Lineage for Impact Analysis" in the Tableau Server or Tableau Online Help.

Keyboard Accessibility for Tableau on the Web

Interact with views

Users sent a link to a WCAG-conformant view or workbook in Tableau Online or Tableau Server can interact and explore using commonly supported WAI-ARIA standards for keyboard navigation. The following keyboard navigation also applies to embedded views.

All components

What to do	Keystrokes
Move focus to next navigation component	**Tab**
Move focus to previous navigation component	**Shift+Tab**
Button activation	**Space** or **Enter**

Navigation panel and toolbar

To skip the navigation panel and toolbar, select **Skip to Content**.

What to do	Keystrokes
Move focus to next component in navigation panel.	**Tab**
Move focus to next item in toolbar.	**Right**
Move focus to previous item in toolbar.	**Left**
Activate the selected toolbar button.	**Enter**
Escape toolbar button window.	**Esc**
In a toolbar button window, move focus between components.	**Tab**
In a toolbar button window, move focus for drop-down menus.	**Arrow keys**
Choose a new menu item from drop-down menus.	**Enter**
Move focus between people or groups in a user selection box.	**Up/Down**
Delete or remove additions in a selected toolbar window.	**Delete**

Custom Views toolbar window

What to do	Keystrokes
Rename custom view	**Ctrl+R**
Set default custom view	**Ctrl+D**

What to do	Keystrokes
Make visible to others toggle on/off	**Ctrl+P**
Delete selected custom view	Windows: **Delete** Mac OS: **Fn+Delete**

Comments pane

What to do	Keystrokes
Apply comment snapshot to the view	**Enter** or **Space**
Delete comment	**Delete** or **Backspace**

Sheet tabs

What to do	Keystrokes
Move focus to previous	**Left/Up**
Move focus to next	**Right/Down**
Navigate to the focused tab	**Enter**

Legends

What to do	Keystrokes
Move focus to previous	**Up**
Move focus to next	**Down**
(Single column) move focus to previous, (Multi-column) move focus to left	**Left**
(Single column) move focus to next, (Multi-column) move focus to right	**Right**
Move focus to first item	**Home**
Move focus to last item	**End**
Toggle item selection	**Space**

What to do	Keystrokes
Toggle highlighting on and off **Note**: This is the keyboard equivalent of **Highlight Selected Items**.	**Enter**
Clear all item selections **Note**: To view legend item text with more contrast, select individual legend items, and then toggle highlighting on and off.	**Esc**

Filters

What to do	Keystrokes
Move focus to previous item	**Left/Up**
Move focus to next item	**Right/Down**
Move focus to first item	**Home**
Move focus to last item	**End**
Toggle selection (check/clear checkboxes, select/ clear radio buttons)	**Space**
Apply changes, if **Cancel** or **Apply** buttons available	**Enter**
Revert changes to default settings.	**Esc**

Filter controls and search

What to do	Keystrokes
Move focus from filter item to filter control buttons (search, Single Value (List)/Multiple Values (List), Single Value (Dropdown)/Multiple Values (Drop-down), Include/Exclude Values). Press **Space** to click a button.	**Shift+Tab**
Move focus to in the drop-down menu	**Arrow keys**

What to do	Keystrokes
Choose a menu item	**Enter**
Press on search button and enter search terms	**Space**
Press to clear search input. Press **Esc** again to dismiss Search.	**Esc**

Parameter controls

What to do	Keystrokes
Single Value (list)	
Move focus to previous item	**Left/Up**
Move focus to next item	**Right/Down**
Commit selection and update parameter value	**Space**
Single Value (dropdown)	
Dropdown list collapsed: Move focus to previous item and commit to value, Dropdown list open: Move focus to previous item and commit to value; doesn't commit to value	**Up**
Dropdown list collapsed: Move focus to next item and commit to value, Dropdown list open: Move focus to next item and commit to value; doesn't commit to value	**Down**
Dropdown list collapsed: Open or close domain list, Dropdown list open: Commit selection of focused item	**Space**
Dropdown list open: Collapse domain list without making a selection	**Esc**
Slider	
See previous item in domain and commit	**Left/Up**

What to do	Keystrokes
See next item in domain and commit	**Right/Down**

View Data Window

What to do	Keystrokes
Clear mark selections	**Esc**
Show the View Data window when the view has the focus **Note**: The screen reader you are using might require some combination of modifier keys (**Shift**, **Ctrl**, **Alt**, **Cmd**) in conjunction with **Enter**. See the documentation for your screen reader.	**Enter**, or press **Ctrl** + **Shift** + **Enter**
Move within table (when screen reader is on)	**Arrow keys**
Move focus to download data link in the window (when screen reader is on). Press **Enter** to navigate to link.	**Tab**

Example: Navigate a view in Tableau Online or Tableau Server

This example shows how you can navigate and interact with a view using keyboard navigation.

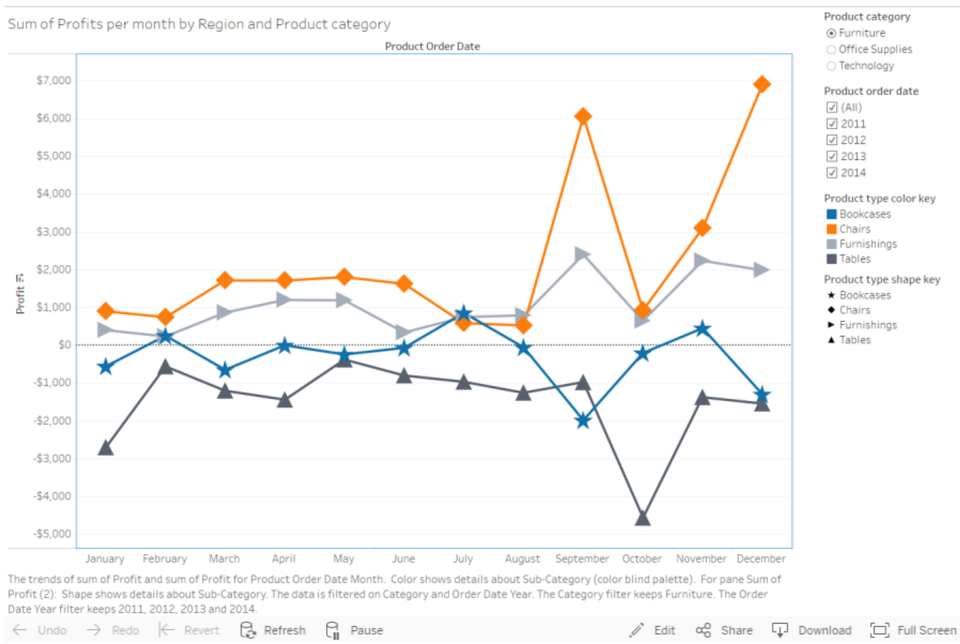
1. When you first open a link to a view in Tableau Online or Tableau Server, press **Tab** to move the focus to view page.

   Press **Enter** to select **Skip to Content**. This lets you skip over the navigation panel and toolbar right to the view page.

   In a workbook that has multiple sheet tabs, press the arrow keys to move focus between the tabs. Then press **Enter** or the spacebar to open a specific view.

1. Press **Tab** to move focus to the visualization area.



   This image shows the visualization area in a sheet. A blue border indicates the focus.

   When the visualization has the focus, screen readers will treat the visualization area as a graphic. Screen readers will identify the workbook that contains the view, and read the view's title and captions.

2. Navigate to filters and legends in the view by pressing **Tab**.

   To change a filter setting, press **Tab** to move focus to the filter, and then press the arrow keys to move between items.

   - Press **Space** to toggle item selections. Press **Enter** to apply filter changes. Press **Esc** to revert changes to the default settings.
   - Press **Enter** to apply filter changes. Press **Esc** to revert changes to the default settings.

   To highlight legend items, press **Tab** to move focus to the legend, and then press the arrow keys to move between items.

- Press **Space** to toggle item selections.

> **Note**: To view legend item text with more contrast, select individual legend items, and then toggle highlighting on and off.

- Press **Enter** to highlight the selected legend items in the view. Press **Enter** again to turn off highlighting.
- Press **Esc** to clear all item selections.

See the table of keystrokes at the beginning of this article for the list of keystroke options for legends and filters.

3. To access underlying data that can be read by your screen reader, tab the focus to the the visualization area and then press **Enter** or press **Ctrl** + **Shift** + **Enter**.

> **Note**: Your screen reader may require some combination of modifier keys (**Shift**, **Ctrl**, **Alt**, **Cmd**) in conjunction with **Enter**. See the documentation for your screen reader for details.

Summary | Full data

Showing first 96 rows.
Download all rows as a text file

MONTH(Product Order Date)	Sub-Category	Sub-Category (color blind palette)	AGG(Profit per Order)	AGG(Profit Ratio)	AVG(Discount)	SUM(Profit)
January	Null	Tables	($207.59)	-25%	27%	-$2,699
February	Null	Tables	($69.42)	-13%	30%	-$555
March	Null	Tables	($54.53)	-7%	25%	-$1,200
April	Null	Tables	($84.57)	-15%	27%	-$1,438
May	Null	Tables	($22.09)	-4%	24%	-$375
June	Null	Tables	($31.84)	-5%	24%	-$796
July	Null	Tables	($50.71)	-9%	27%	-$964
August	Null	Tables	($43.23)	-7%	28%	-$1,254
September	Null	Tables	($29.56)	-5%	26%	-$976
October	Null	Tables	($123.26)	-23%	29%	-$4,561
November	Null	Tables	($34.27)	-4%	23%	-$1,371
December	Null	Tables	($32.73)	-4%	26%	-$1,538
January	Null	Furnishings	$11.23	10%	10%	$404
February	Null	Furnishings	$7.58	10%	17%	$235
March	Null	Furnishings	$18.07	17%	13%	$868
April	Null	Furnishings	$21.09	17%	10%	$1,202
May	Null	Furnishings	$16.32	16%	15%	$1,192
June	Null	Furnishings	$6.10	6%	18%	$341
July	Null	Furnishings	$11.70	10%	13%	$748
August	Null	Furnishings	$15.55	18%	13%	$793
September	Null	Furnishings	$20.17	20%	12%	$2,400
October	Null	Furnishings	$9.40	12%	15%	$649
November	Null	Furnishings	$16.33	13%	14%	$2,237
December	Null	Furnishings	$14.63	14%	15%	$1,990
January	Null	Chairs	$60.11	8%	17%	$902
February	Null	Chairs	$43.72	10%	17%	$743
March	Null	Chairs	$47.73	8%	19%	$1,718
April	Null	Chairs	$47.62	9%	18%	$1,714

An example of the table shown in the Summary tab of the View Data window

The **View Data** window opens in a new browser window.

The **View Data** window includes a link to download the raw data as a text file. The **Summary** tab lists the data that is currently selected in the view, or all data in the view if no data is selected. To clear selected data in the view, press **Esc**.

Use keyboard navigation for table items to move through the data.

4. To access buttons in the toolbar, tab the focus to the toolbar area and then use the right arrow to move to the next toolbar button.

   Press **Enter** to activate the Comments button on the toolbar.

   To add a comment, tab to the Add a comment field and type in your comment. To add a snapshot to your comment, use tab to move the focus to the snapshot button. Select **Enter** to add a snapshot, and tab to the Post button. Press **Enter** to add the comment.

   Select **Tab** to move through options in the button's window, including exiting the window.

Manage content

Use the following keyboard interactions to manage content on Tableau Server or Tableau Online.

Organize favorites

On your Favorites page, you can move items when content is displayed in grid view. After you reorder your favorites, switch between this custom order and other orders like name or date by using the Sort By menu. For more information, see **Mark Favorites** on page 3195.

What to do	Keystrokes
Pick up item	Windows: **Alt+Up** Mac OS: **Option+UP**
Move item left	**Left**
Move item right	**Right**
Undo move	**Esc** or **Tab**
Drop item	Windows: **Alt+Down** Mac OS: **Option+Down**

Explore and Manage Web Content

What Can I Do with a Tableau Web View?

You’ve seen hundreds of presentations, emails, and documents with data charts. Those things are static and sad. First off, the data is frozen in time. Second, if you have any questions about it, you’ve got to ask whoever created the chart . . . if you can find them.

Tableau is different. When someone sends you a link to a web view, you’re often seeing the freshest data available, and more importantly, you can *interact* with it. Want to dig deeper and answer questions the minute they occur to you? Simply click the view right in your web browser; it’s that simple.

What can you do with a Tableau web view? A lot!

> **Note:** We’re covering the most popular ways you can interact with views. Sometimes, view creators enable only a subset of these features.

Will I hurt the data?

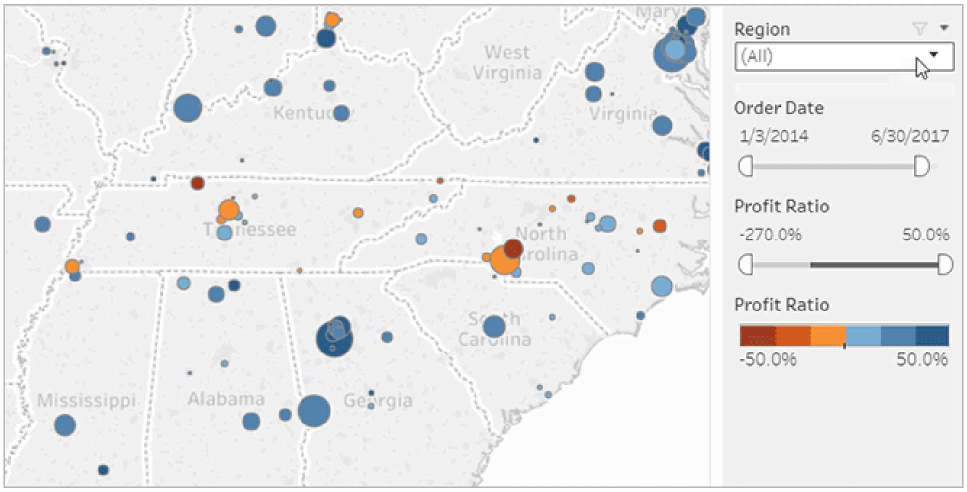
Not at all. When you interact with a view, you’re simply changing how it *looks* for a moment. Next time you or your colleagues open the view, it will look exactly as it originally appeared. The underlying data always remains safe and sound, so feel free to explore!

Filter and sort data

Want to trim or rearrange the visible data so you can focus on stuff you’re interested in? You’ve got a couple easy options:

Filter data

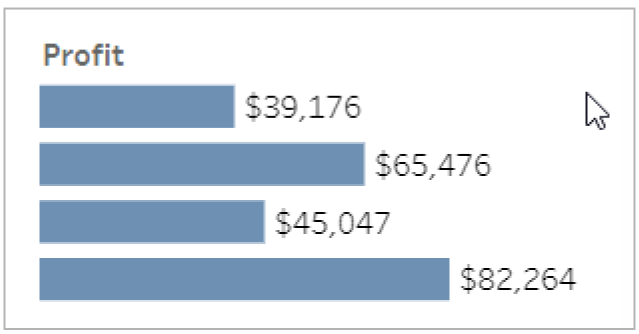
Most view creators add filters that let you limit visible data to specific date ranges, regions, and categories. When you see these, feel free to click or drag them as needed.



Click the image above to replay the animation.

Sort data

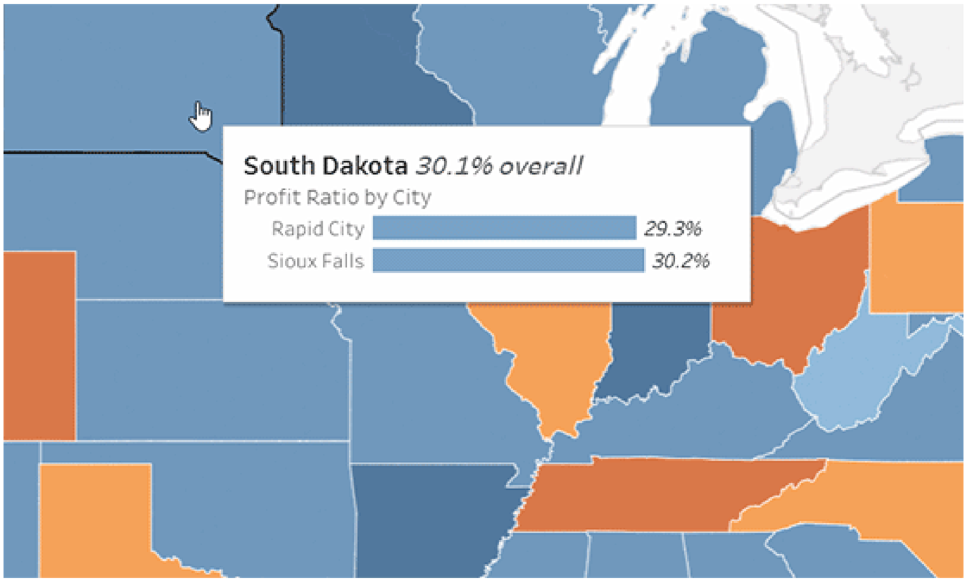
If you're looking at a table of data and want to sort it alphabetically or numerically, just hover over a column header and click the sort icon.



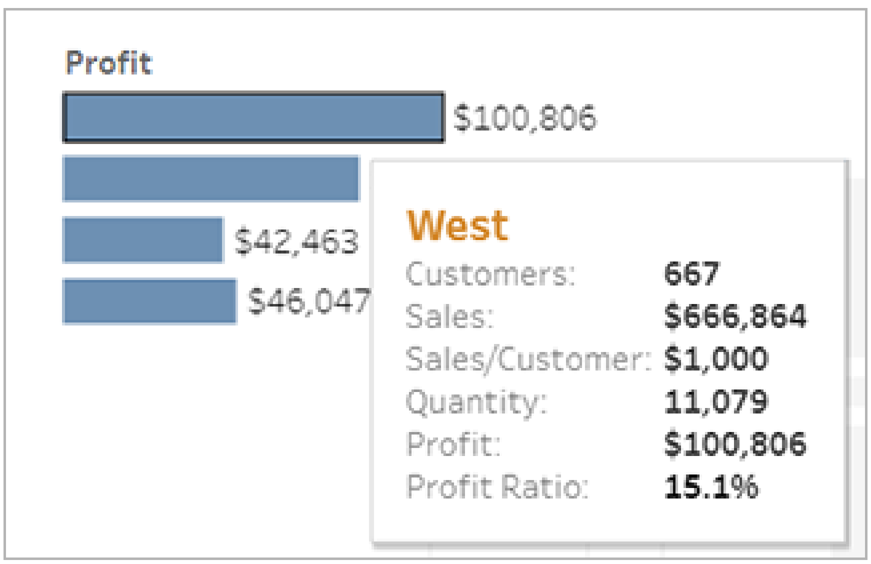
Click the image above to replay the animation.

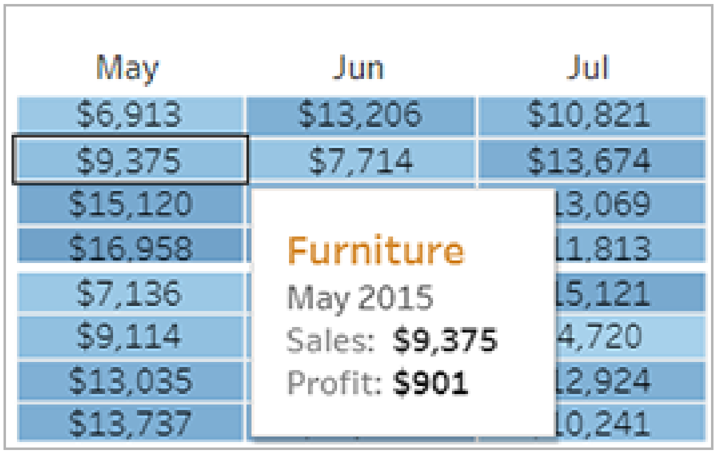
See details about specific data points

As you move the mouse across a view, you'll often see tooltips that reveal details about each data point, or *mark*. Marks can take many forms; here's how they look in maps, bar charts, and tables:



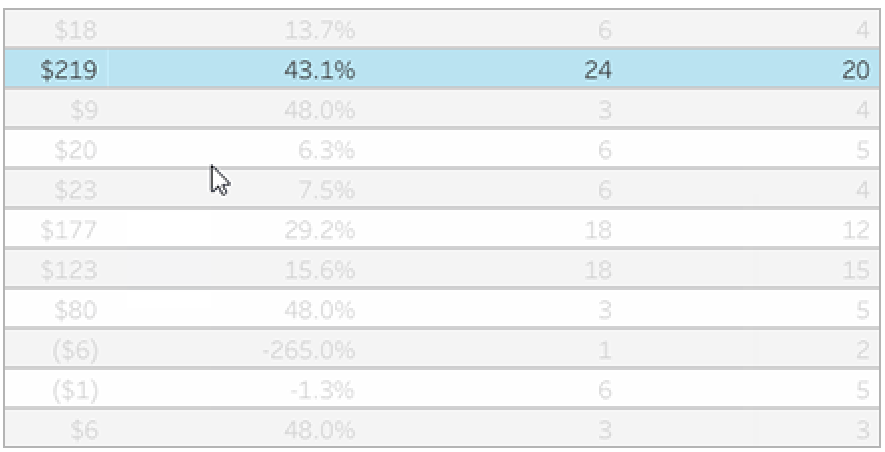
Click the image above to replay the animation.





View underlying data

If you're the inquisitive type, you might want to see summarized source data to get a better sense of the numbers behind the visuals. If the view creator has given you permission, you can click any mark in the view, and then click the **View Data** icon.



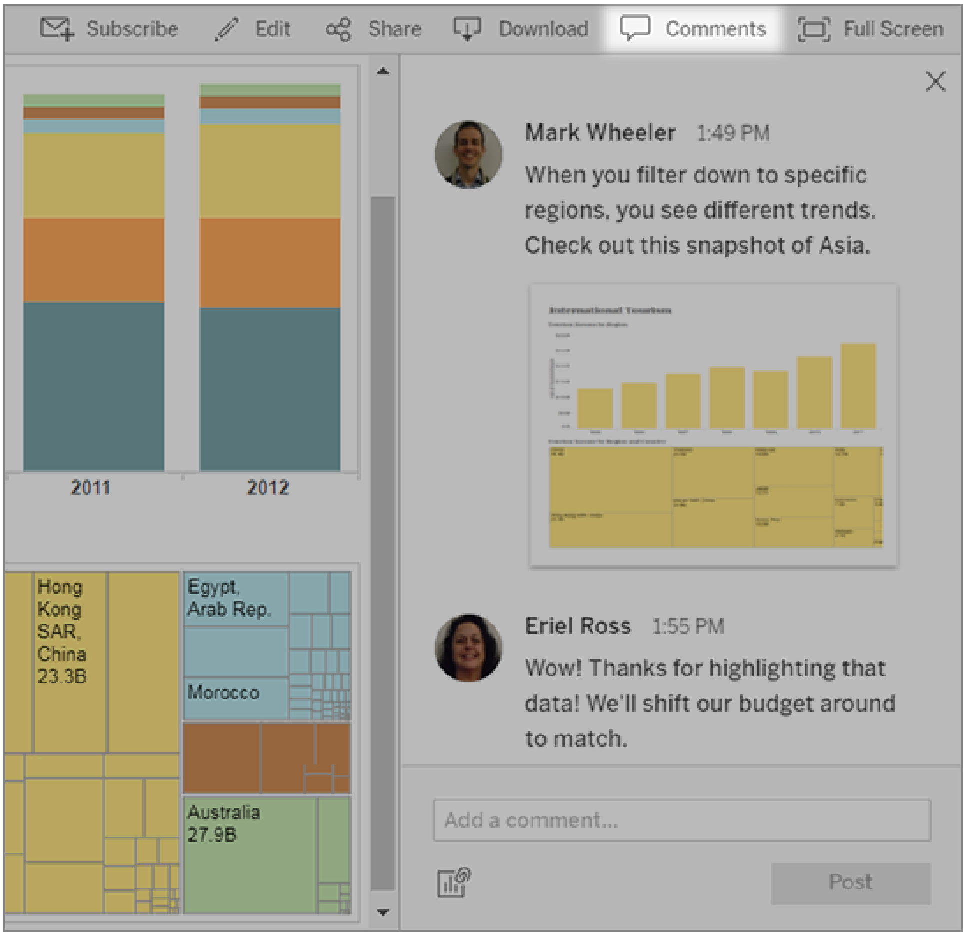
Click the image above to replay the animation.

Collaborate on data

Data becomes more meaningful when you explore it with other people.

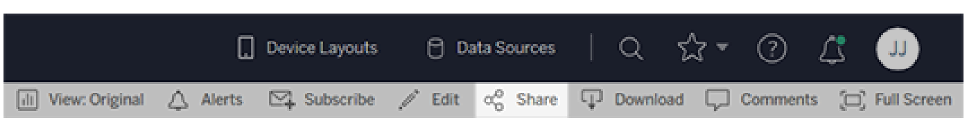
Add comments

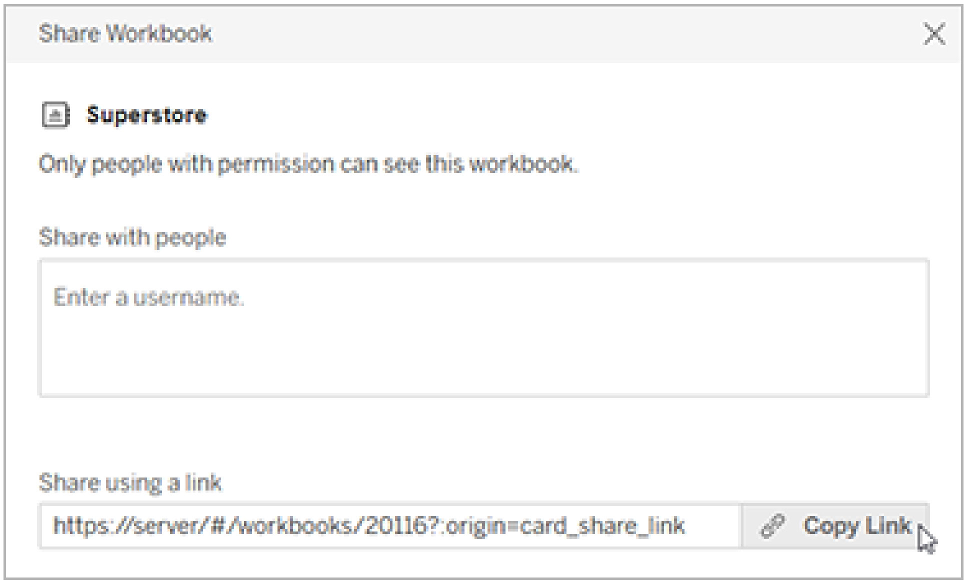
To quickly ask questions or share insights, click **Comment** in the toolbar. You can pair your remarks with a snapshot of the view that reflects any filters or other changes you made.



Share links

Want to share the excitement of directly engaging with data? Click **Share** in the toolbar, copy the link to the view, and then email or instant-message it to anyone with a Tableau account at your organization.





Get views emailed to you on a schedule

To automatically receive emails of a view on a regular schedule (before weekly meetings, for example), click **Subscribe** in the toolbar.

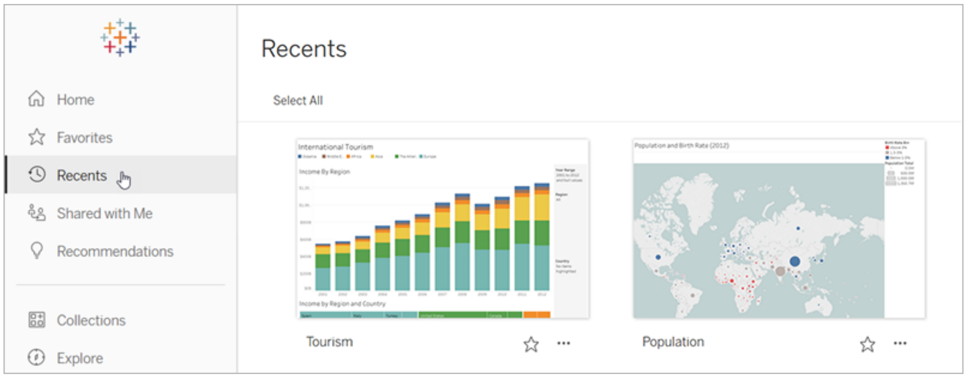
> **Note:** You might also receive emails to views that other people have arranged for you, including data-driven alerts that notify users when data crosses key thresholds for your business.

Download a view

You've seen the power of interactive Tableau web views. But there are times when you want to show off that data outside the browser. If the view creator has given you permission, click **Download** in the toolbar to create a static .png, .pdf, or .pptx file that reflects the view as it looks right now.

Find a recently accessed view

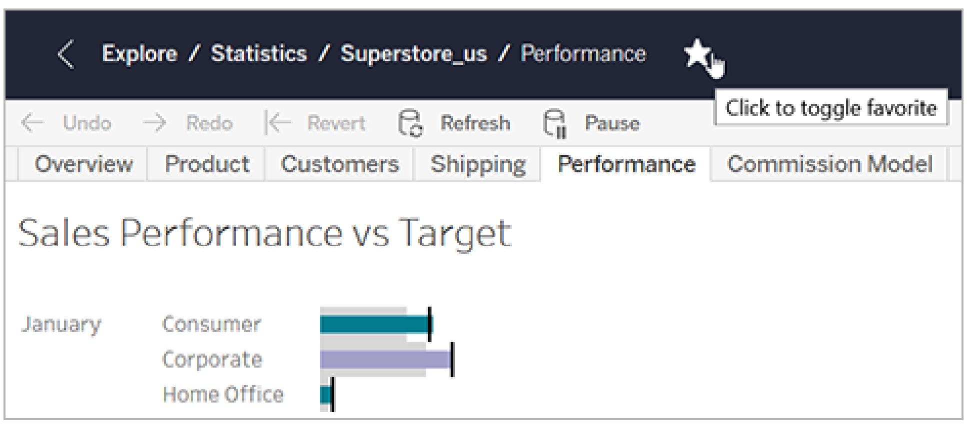
If you come across an interesting view but later forget where you found it, don't worry. Tableau keeps track for you. You can find your 12 most recently accessed views on the Recents page, which you can open from the navigation panel.



You can also find a smaller selection of your most recently accessed views on the Home page.

Add a view to favorites

As the Tableau content in your organization grows, you'll want to easily find data important to you. To add a view to your favorites, click the star icon in the header.



To return to a favorite, open the Favorites page from the navigation panel.

Favorites you choose in Tableau Server or Tableau Online also appear in the Tableau Mobile app, and vice versa, giving you instant access to key data wherever you go.

Congratulations, data rockstar!

See how easy that was? Now that you're familiar with web views, start exploring all the Tableau content available at your company. Answer questions. Share insights. You've got this!

See also

Tour Your Tableau Site on page 3104

Share and Collaborate on the Web on page 3243

Zoom and Pan Views, and Select Marks on page 2719

Use Tags

Tags are keywords that you add to items on Tableau Server and Tableau Online to categorize them. Tags help users find and filter Tableau content (workbooks, views, metrics, data sources, and flows) and, with Tableau Catalog, external assets (databases, files, and tables).

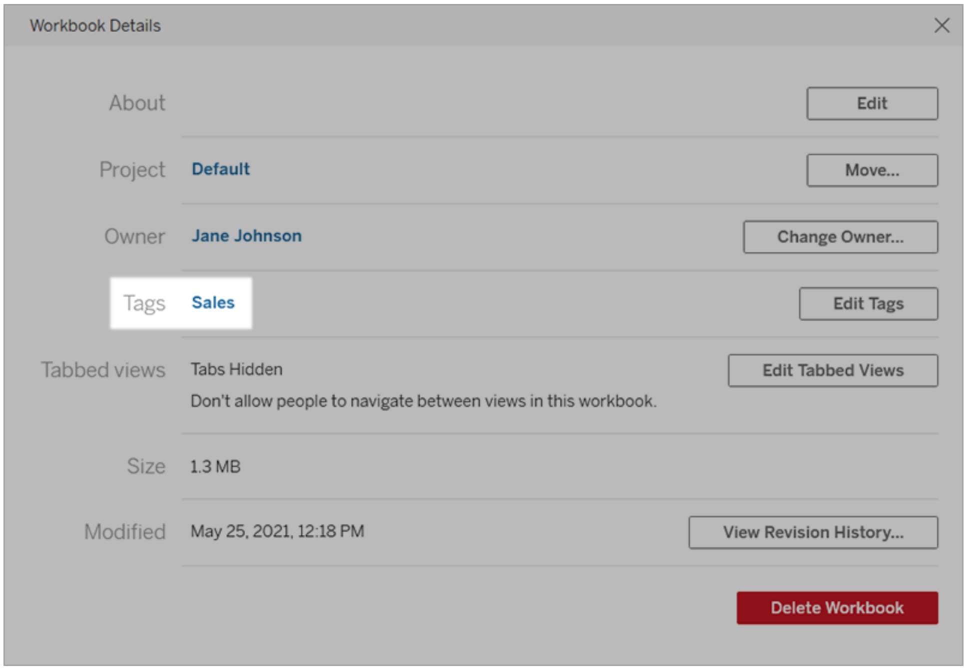
If you have a site role of Explorer or Creator, you can add tags and delete the tags you have added. Authors can also tag content when they publish from Tableau Desktop or Tableau Prep to a Tableau site.

> **Note:** You can't tag projects, data roles, or Ask Data lenses.

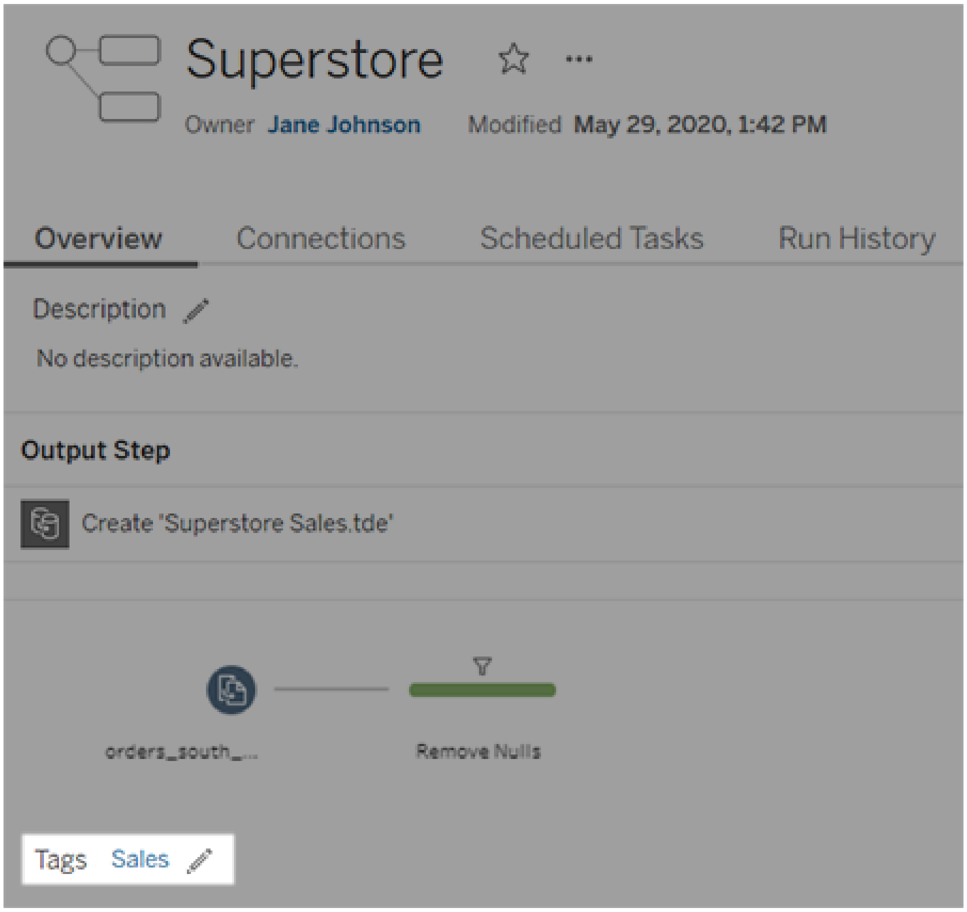
Tableau Catalog, part of the Data Management Add-on to Tableau Server and Tableau Online, is required to be able to tag external assets (databases, files, tables, and columns). For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

See tags for an item

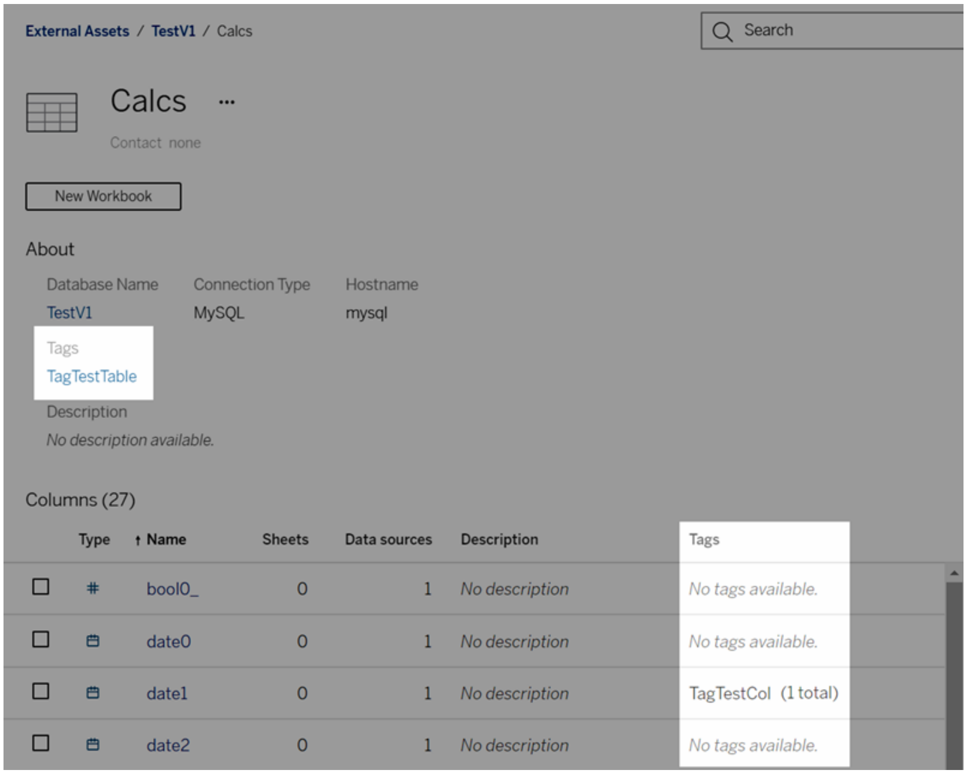
Tags appear in the details for workbooks, metrics, and data sources. Click on a tag to see a list of all items of that type with that tag.



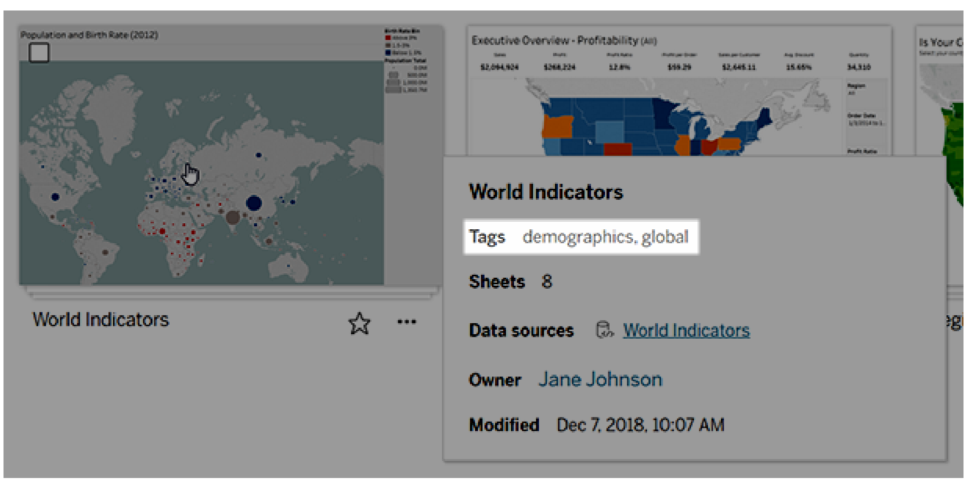
For flows, tags appear at the bottom of the Overview page.



When Tableau Catalog is enabled, tags for external assets appear in the About section. Tables also include a Tags column that lists the tags for each column in the table.

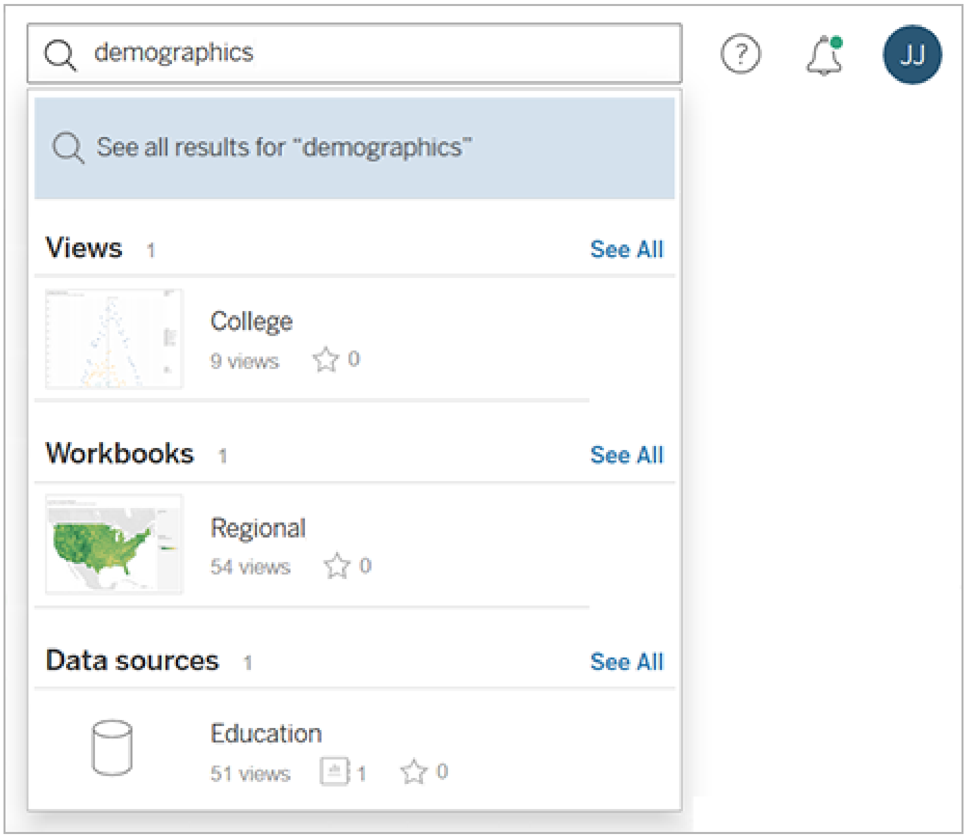


When you browse content in grid view, tags appear in the tooltips shown when you hover over thumbnails. If an item has no tags, the tags section doesn't appear. There is no grid view for external assets.

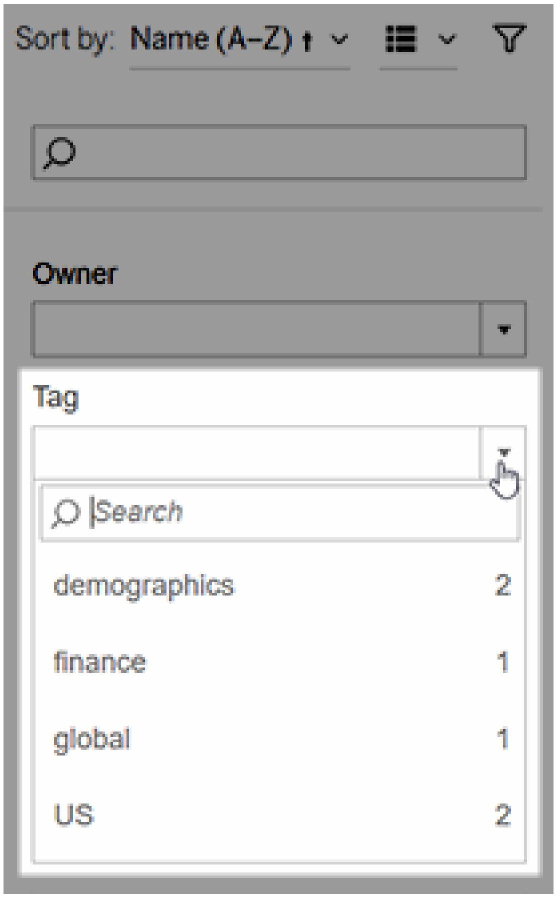


Find tagged content

Use tags to find content with quick search.



Or use filters to search the list of tags.



Note: The tag filter doesn't appear if you are looking at multiple types of content in a project. To access the tag filter from within a project, first use the content type selector to display a single type of content that can be tagged, such as workbooks.

When to use and when not to use tags

The purpose of tags is to help users filter content in order to find relevant items. Because any user can add tags, they aren't recommended for scenarios when you need tight control over the categorization of content. For example, don't use the existence of a tag to make decisions around whether to grant access to sensitive content.

Examples of when to use tags:

- Add the tag "Sales" to content located across different projects that is relevant to salespeople, so they can easily find it.

- Add a tag to content created during a particular initiative, so it can be viewed together, regardless of which projects it lives in.

Examples of when not to use tags:

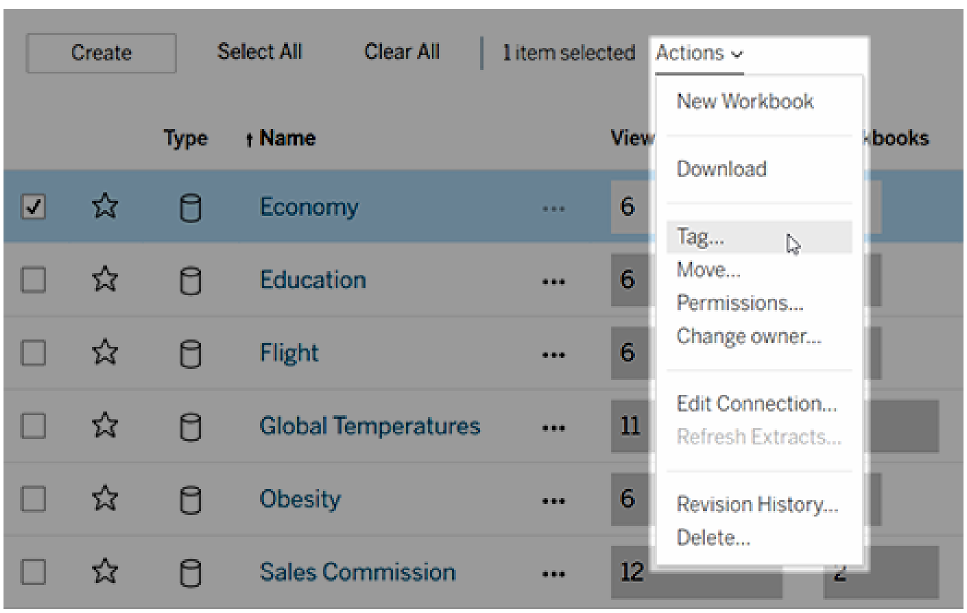
- Don't add tags with the intention of using them to decide how to set permissions for content. Anyone can add these tags to content.
- Don't add a tag about whether data is approved to share with a certain people. Another user can add this tag without approval.

Add tags

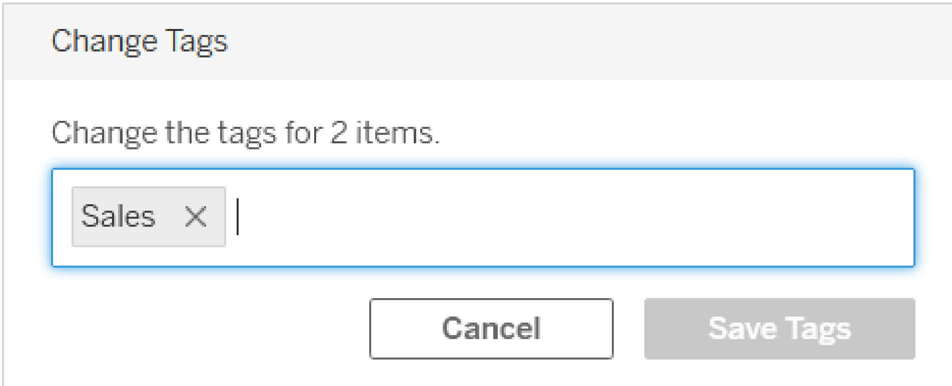
If you have a site role of Explorer or Creator, you can add tags to any items that you have permission to access. You don't need to be the owner of an item to tag it. (To add a tag to a column, see **Use Tags** on page 3180.)

1. Navigate to a list of items.

2. Select one or more items you want to tag, and then select **Actions** > **Tag**.



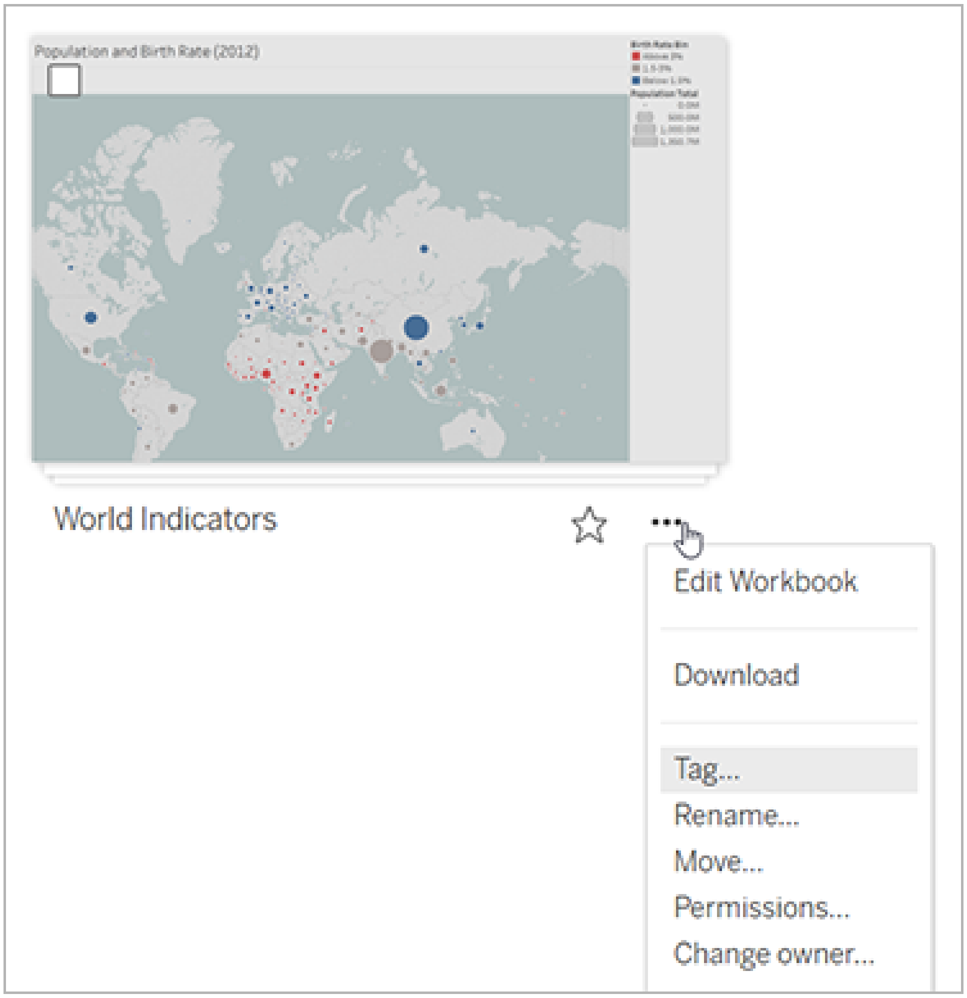
3. Enter one or more tags in the text box, and then click **Save Tags**.



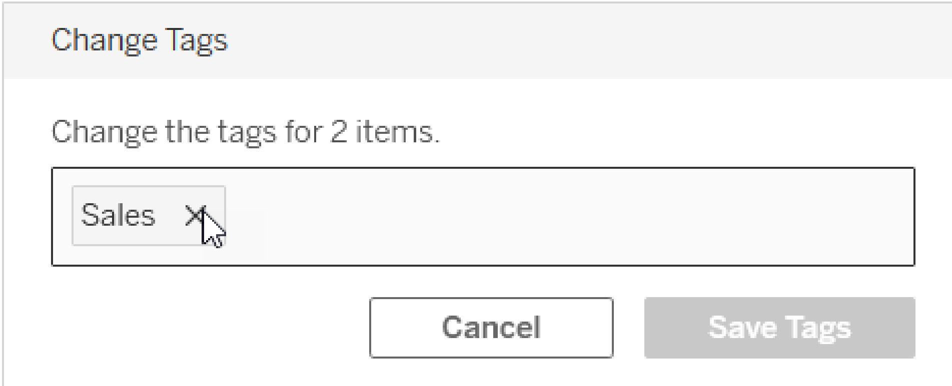
Delete tags

You can delete tags that you have added.

1. Click the Actions menu (**...**) for an item, then select **Tag**.



2. Click the '**x**' next to the tag you want to remove, and then click **Save Tags**.



Other places tag information is surfaced

Tags you add are visible to others via auto-complete when users add their own tags.

In addition to seeing tag information in the product (**See tags for an item** on page 3180), you can also surface tag information using the REST API or Metadata API to query lineage-related tasks. For more information about these APIs, see Tableau Metadata API and Tableau REST API - Metadata Methods.

Tag item count mismatch

When you search tags, the item count shown for a tag represents the items you have permission to view. Depending on how your site is configured, when you look at lineage data, the item count shown for a tag might be the total number of items regardless of your permission to view those items. However, all sensitive information about items you don’t have permission to view will be obfuscated. For information about sensitive lineage data, see "Access lineage information" in the Tableau Server or Tableau Online Help.

Search Your Tableau Site

Tableau Server and Tableau Online provide two ways to search: quick search and filtered search.

Quick search, accessed at the top of the page, searches the entire site for matching items of any type. Filtered search, accessed from the filters pane, finds matches on the page you are viewing.

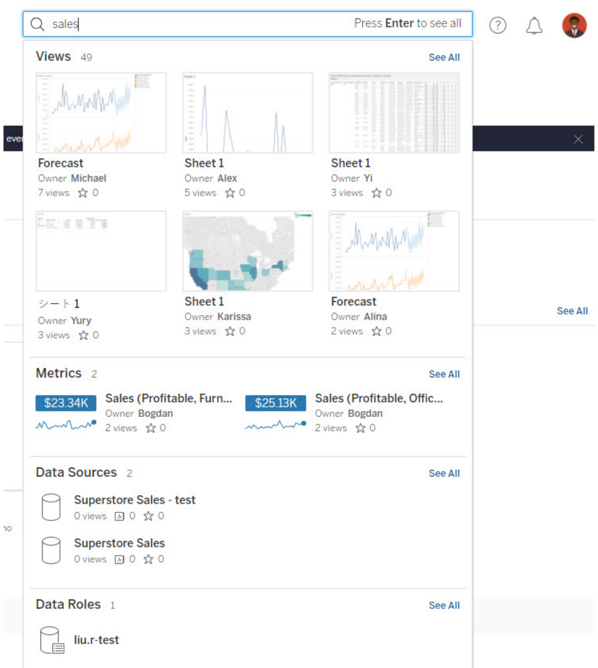
Both quick search and filtered search support attributes and operators that help you set the scope of the search. For example, you can include an attribute to limit a search to just titles of views or use operators such as `and` and `not`.

Note: If Tableau Catalog is enabled on your site, you can use quick search to find databases and tables (also called external assets) that are not embedded. Starting in 2019.3, Tableau Catalog is available in the Data Management to Tableau Server and Tableau Online. For more information, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

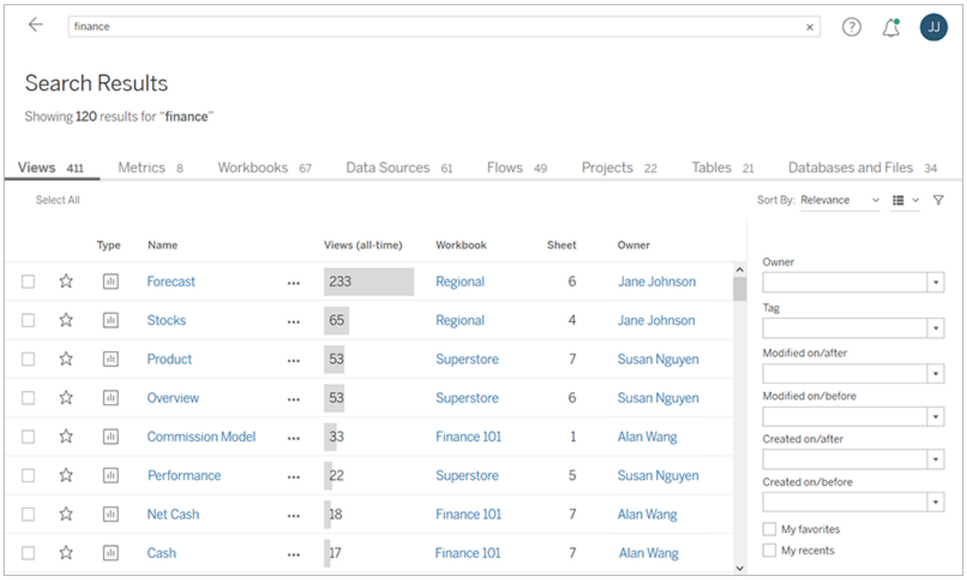
Quick search

Use the quick search field at the top of the page to find items across your Tableau site. Quick search looks for matching text in item names, metadata, owners, tags, captions, comments, and other information.

As you type, the top items that match your search text appear. These brief results show limited details, such as the number of page views for each item.



Press Enter key to see the full page of search results. To display more details, select **View As** > **List**. The details provided about each result, such as the date it was last modified and the number of times it has been viewed, help you find the freshest and most popular items.



Click the tabs at the top of the page to show results for each type. Use the **Sort By** options to order items by general properties, like name, and type-specific properties, like certification status for data sources. Further refine your search with the filters on the right side of the page.

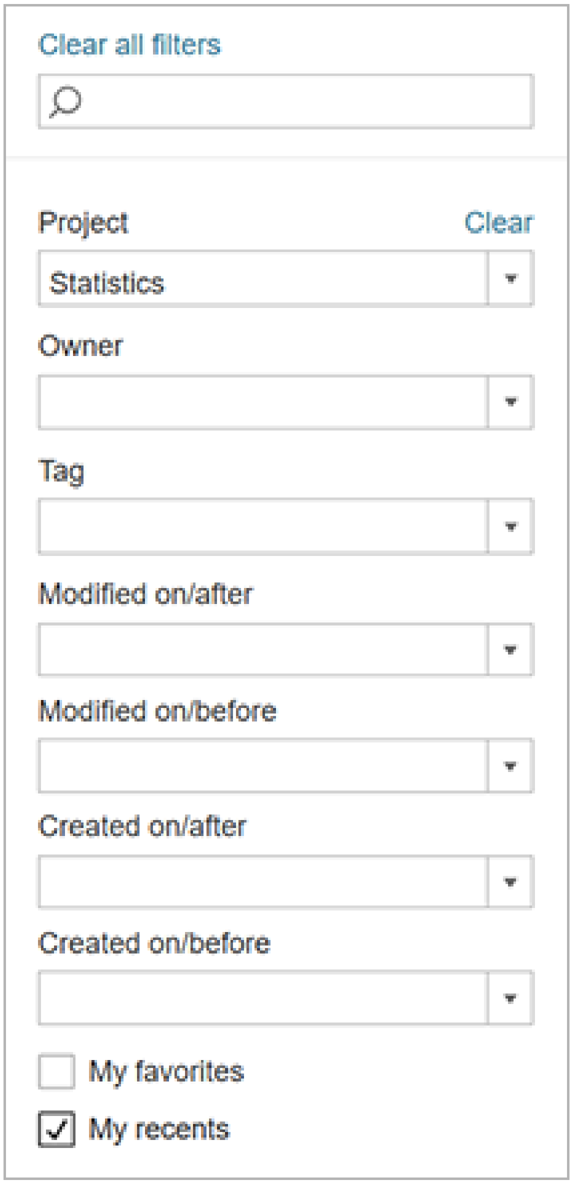
> **Note**: Quick search searches published data sources only. To search on embedded data sources (workbooks that use a data connection only), use filtered search.

Tableau Catalog users might see search results for non-embedded databases and tables before ingestion to Catalog is finished. When this happens, if you click this entry from the quick search results, a "Showing partial results..." warning message displays. If you review the search results page, the entry is grayed out until ingestion to Catalog is complete.

Filtered search

With filtered search, you can use a combination of search text and filters to find items. In addition to general filters, some items have filters unique to their type. For example, you can filter to show workbooks with alerts or data sources that are encrypted.

Filter options for views are shown below.

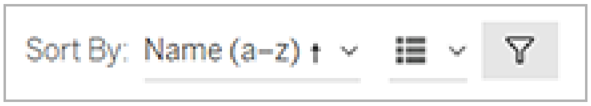


A filtered search applies to the items on the page you are viewing. It does not search elsewhere on the site.

Note: If you are looking at a page that shows more than one type of items, you will see only the general filter options. To see filters specific to, for example, workbooks, you must first filter to show only workbooks on that page.

To conduct a filtered search:

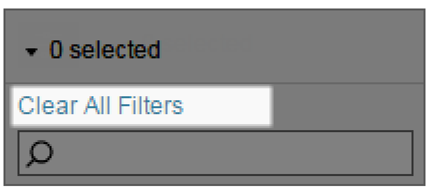
1. Navigate to the page you want to search, then click the Filters (funnel) button to open the pane.



2. To search, enter your text in the top field of the Filters pane, then press the Enter key.
3. To filter, select the filter, then enter a value for the filter or select from the available options.

To remove a filter, click **Clear** next to the filter name.

To remove all filters, click **Clear All Filters** at the top of the Filters pane.



Search attributes

Tip: Search is not case sensitive.

In addition to a general search, you can limit search on Tableau Server and Tableau Online to a specific attribute such as name, workbook, data source, and so on.

To include a search attribute, use the following syntax in the search box:

`<attribute-name>:`*`search-text`* (with no spaces on either side of the colon).

For example, to return only items whose names contain the words *sales*, type the following:

`name:`*`sales`*

You can include multiple attributes to further limit a search. For example, to find all dashboards that are owned by Smith, type the following:

`sheettype:`*`dashboard`*` owner:`*`smith`*

A complete list of attributes is shown below.

This attribute...	Followed by...	Returns
`name:`	search term	Items with names that match the search term
`title:`	search term	Views with titles that match the search term
`caption:`	search term	Applies to views with captions.
`owner:`	user name	Items that are owned (published) by the specified users. **Note:** Prior to 8.2, owners were listed as publishers in Tableau Server. The **publisher** search attribute is still supported and returns the same results as the **owner** attribute.
`publisher:`	user name	(See owner above)
`project:`	search term	Items that are part of a project whose name matches the search term
`comment:`	search term	Views whose comments match the search term
`tag:`	search term	Items whose tags match the search term
`field:`	search term	Views with matching fields on the rows, columns, level of detail, pages, or encoding shelves
`sheettype:`	view, dash-board, or story	Views that are of the matching sheet type
`class:`	type of data source (e.g., mysql)	Views and data sources that are associated with the matching type of data source
`dbname:`	name of data-base	Published data sources that are associated with the matching data source
`nviews:`	number	Workbooks that contain the specified number of views

Note: Quick search restricts the searches of `dbname` to published data sources only. To search on embedded data sources (workbooks that use a data connection only), use filtered search.

Search operators

You can use `and`, `or`, `not`, and `*` with search phrases, including combining them with attributes, to build search expressions. For example, if you want to search for all items that do not match a specific phrase, or to match one phrase or another but not necessarily both.

Operator	Definition	Examples
`and`	Returns items that match both search terms.	sales `and` marketing; pens `and` paper
`or`	Returns items that match either search term.	west `or` east; soccer `and` football
`not`	Excludes items that match the search term following this operator	`not` sheettype:dashboard
`*`	Acts as a substitute for any character or word following or as part of the search term. This operator can be used by itself or at the beginning or end of the search term. This operator is useful when you don't know the exact term you are searching for.	dev`*` sales`*`

Special characters

If your search phrase includes spaces, punctuation, or reserved terms such as `and`, `or`, or `not`, enclose your search phrase in double quotation marks, for example:

`name:"pens and paper"`

Mark Favorites

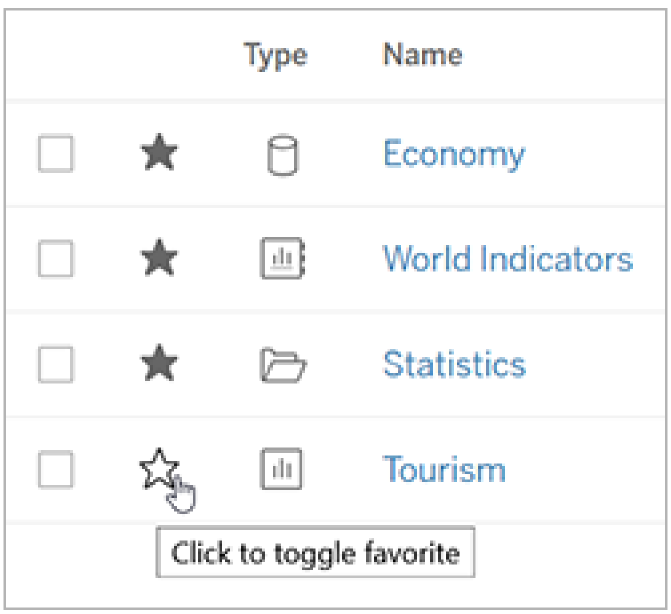
To make it easy to find your most-used content, you can mark it as a favorite. You can add any type of content to your favorites, except for embedded data sources and Ask Data lenses.

Mark content as a favorite

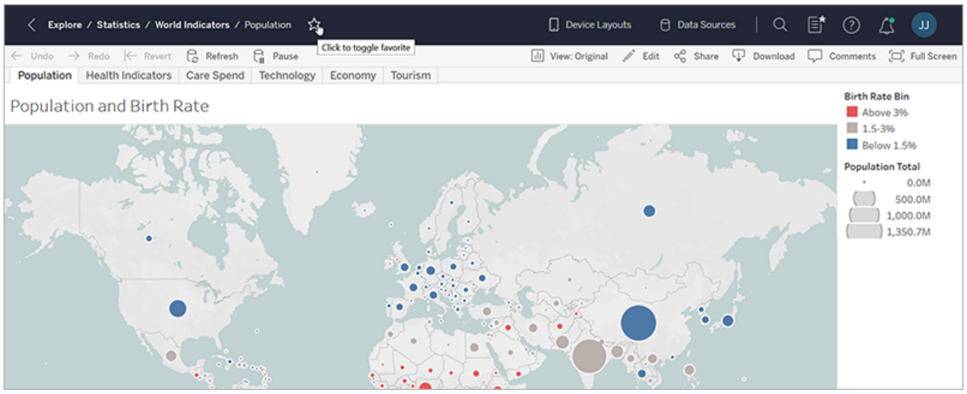
Click the favorites star near the content name.

In grid view:

Or in list view:



Alternatively, when you have a view open, click the star in the header.



Remove an item from favorites

Click the star that you used to mark the favorite.

Find favorites

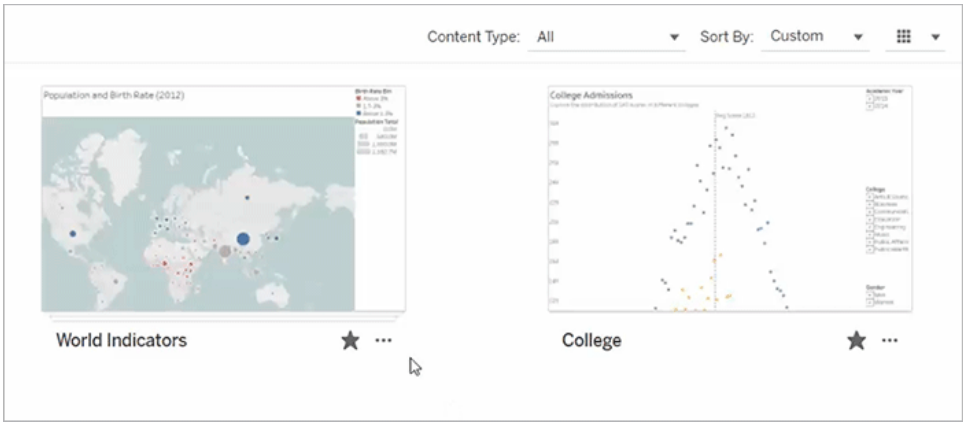
To access your Favorites page, from the navigation panel, click **Favorites**.

You can filter the content on your Favorites page to find specific items. Filters are helpful if you accumulate a large number of favorites. For more information, see Filtered search.

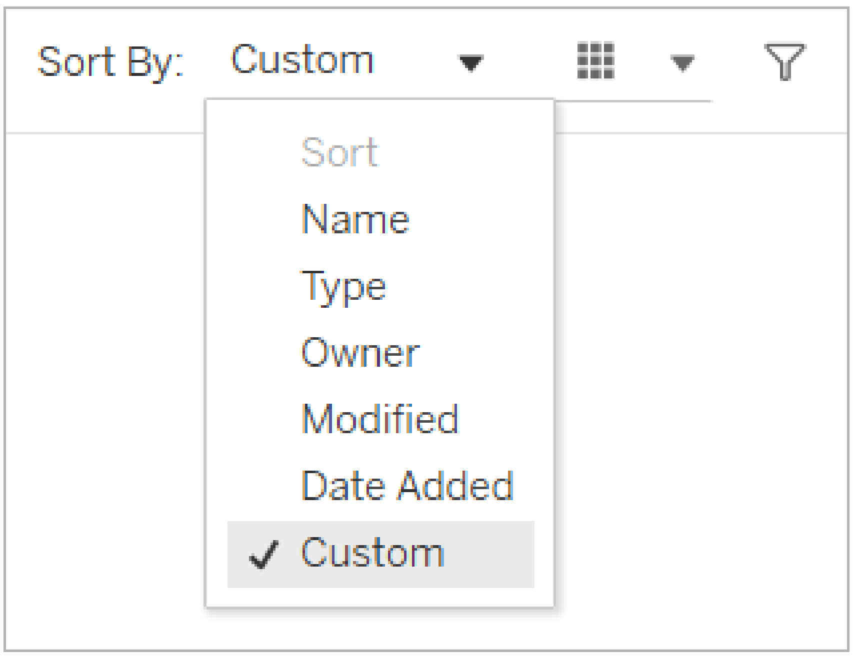
> **Note:** Your recently added favorites also appear on the Home page.

Organize your favorites

In grid view, drag and drop cards to reorder your favorites.



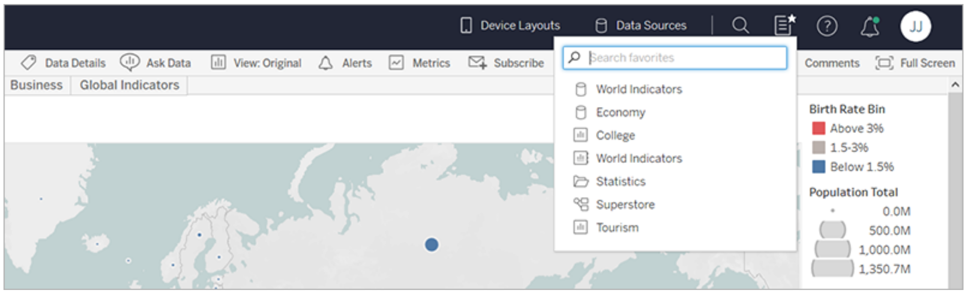
Switch between the custom sorting order you create and other orders like date or name using the **Sort By** menu.



Note: For information about how to organize favorites using a keyboard, see **Keyboard Accessibility for Tableau on the Web** on page 3164.

Access favorites from a view

When you open a view, the navigation panel is not displayed. To access your favorites while looking at a view, click the list icon on the right side of the header. Enter keywords into the search box to find specific items.



Organize Items in a Collection

Collections let you gather related items into an easy to access list. With a collection, you can organize items in a way that is meaningful to you, without worrying about who owns an item or which project it lives in on your Tableau site.

You can keep a collection private or share it with others. When you give others access to a collection, there's no risk of exposing data, because collections don't affect the permissions for the items they contain. In a collection, users see only the items that they have permission to access.

You might want to build a collection to:

- Onboard new employees with data appropriate for their roles
- Organize the data you frequently use for a project in one place
- Promote certified data and approved dashboards in your organization

Find collections

From the navigation pane, click Collections to see all the collections you have permission to view. The collections that you are the owner of appear under the My Collections tab. Collections aren't included in the Explore section of your site or when you search, but they do appear in Favorites and Shared with Me, if applicable.

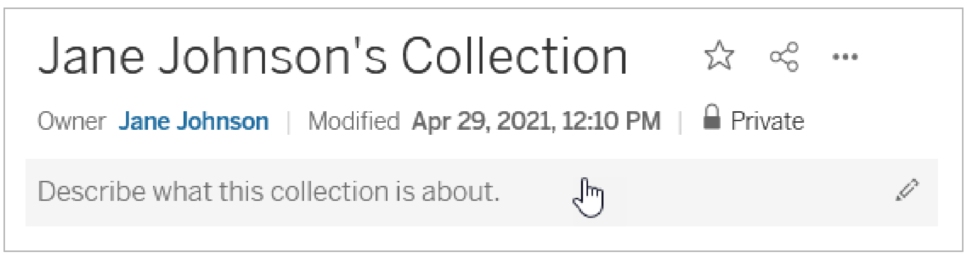
When you open a collection, you'll see the items you have permission to view. If there are many items in the collection, filter and sort to find specific items (some filter and sort options are limited in collections). The number of items displayed in a collection may differ from the total number of items listed in the collection details, depending on the permissions for the individual items.

Create a collection

To add items to a collection, you first need to create one. Anyone can create a collection. You don’t need a specific role or permission.

1. From the navigation pane, click **Collections**.
2. Click the **New Collection** button.

   The collection is created with a default name.
3. To change the name from the default, mouse over the name and click the highlighted text. Enter a new name, and then click **Save**.
4. To add a description, mouse over the description area and click the highlighted text. Enter a description, and then click **Save**.

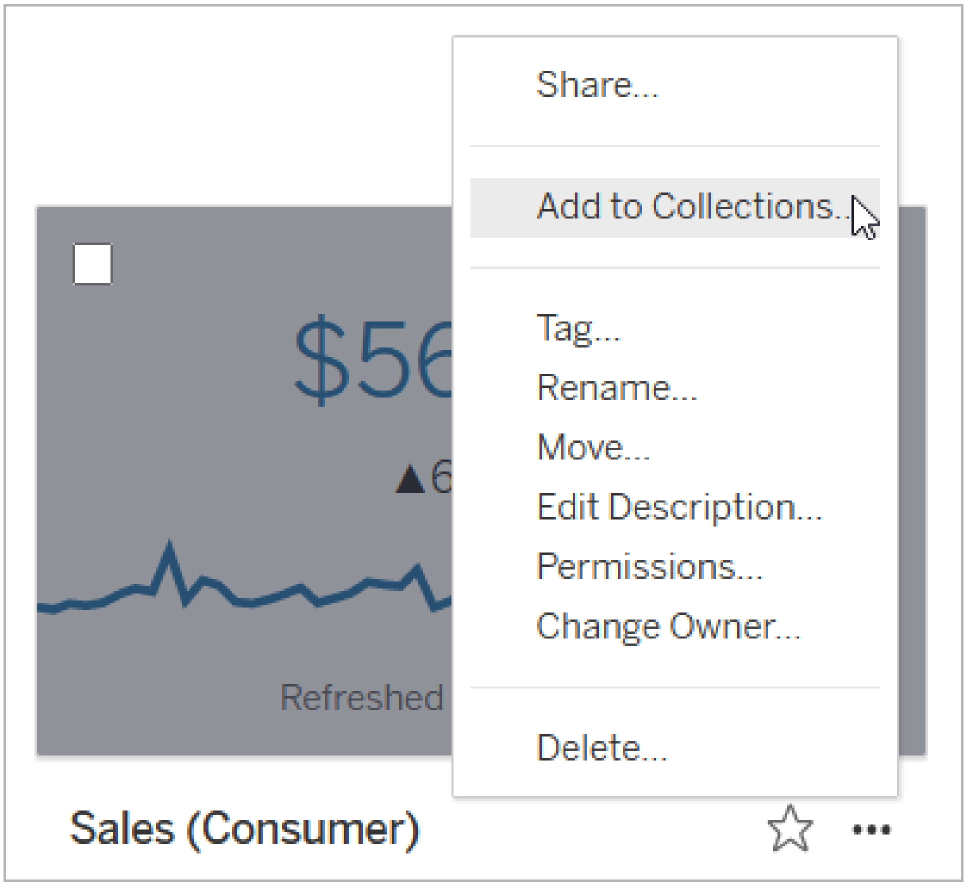


You can also create a collection from the Add to Collections dialog, shown below.

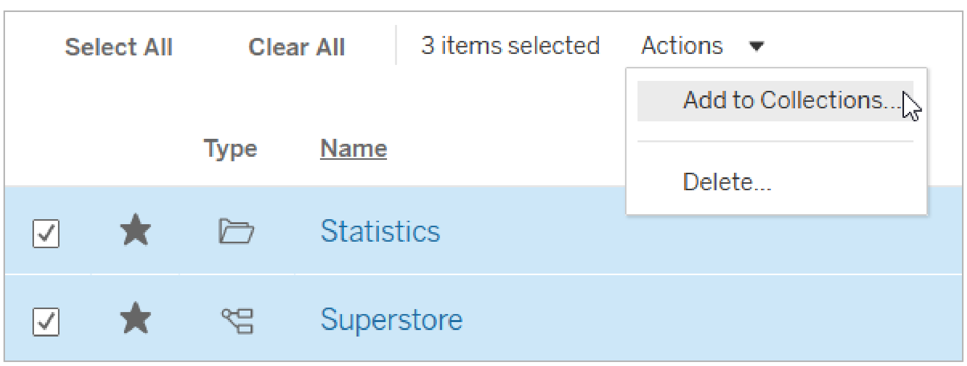
Add items to a collection

If you can view an item on your site, you can add it to your collection, with the exception of external assets (databases, files, and tables), lenses, and other collections. You must own a collection to be able to add items to it. Though the items you collect might belong to different projects, you can add them to a collection without needing to move or copy them.

1. Navigate to the item you want to add to a collection.
2. Click the Actions (**...**) menu, and then click **Add to Collections**.

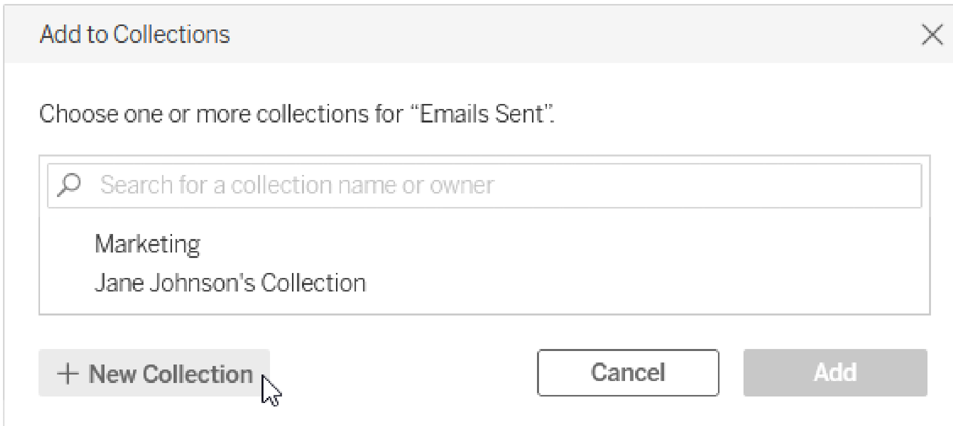


To add multiple items to a collection, select checkboxes for the items, and then click **Actions** > **Add to Collections**.



3. On the Add to Collections dialog, select the collection you want to add the item or items to. You can select multiple collections. If a collection name isn't selectable, that means that the collection doesn't have enough space to accommodate the items. A collection can contain a maximum of 1000 items.

(Optional) If you want to make a new collection for these items, create one by clicking the **New Collection** button. Once you create it, the new collection will be pre-selected in the dialog.

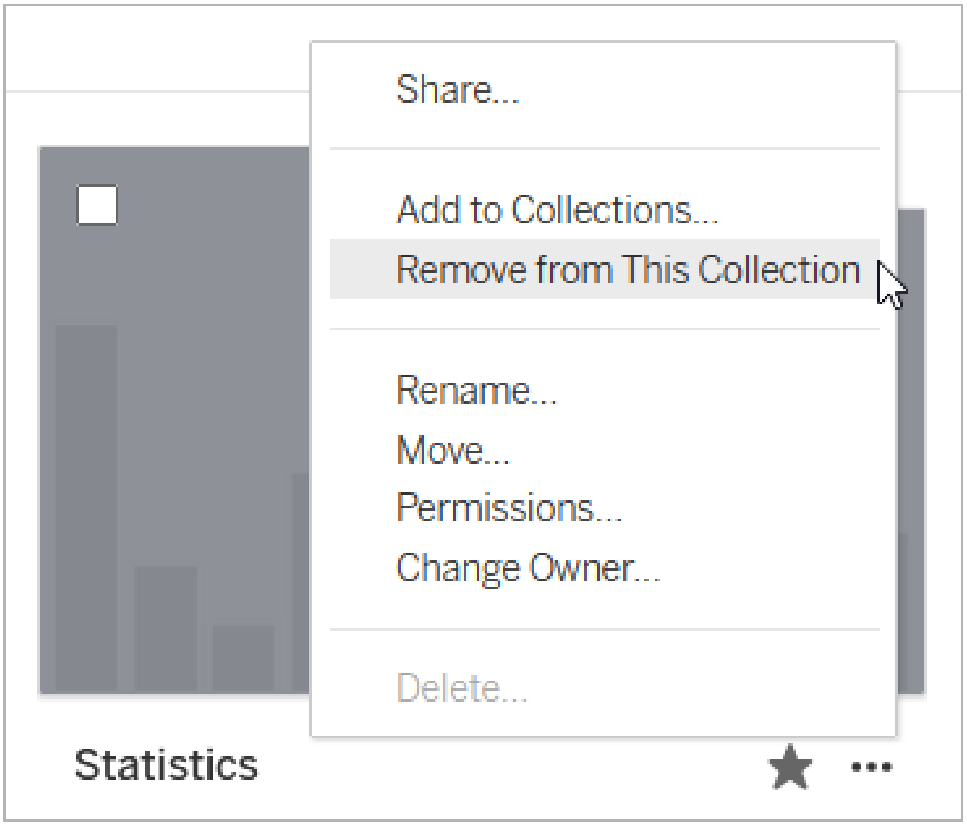


4. Click **Add**.

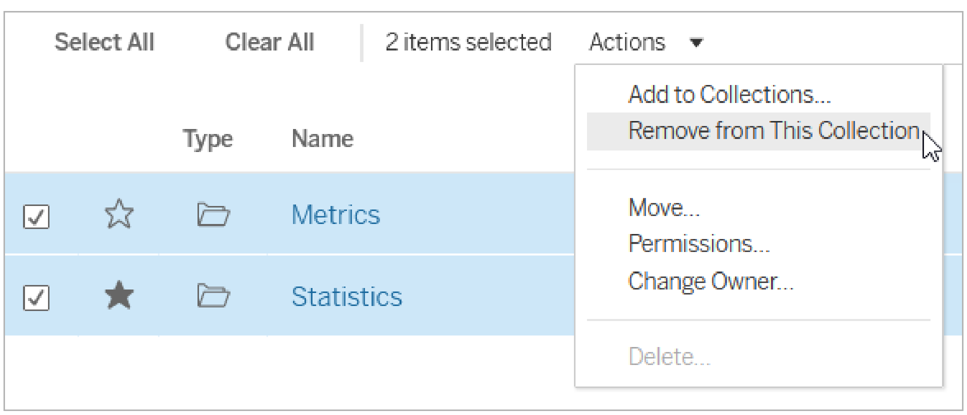
Remove items from a collection

You can remove items from collections that you own.

1. From the navigation pane, click **Collections**.
2. Click the **My Collections** tab.
3. Open the collection you want to remove items from.
4. To remove a single item, click the Actions (**...**) menu for that item, and then click **Remove from This Collection**.



To remove multiple items, select checkboxes for the items, and then click **Actions** > **Remove from This Collection**.



If you accidentally remove an item, don't worry. You can simply navigate back to where it lives and re-add it.

Deleting a collection or removing items from a collection won't delete the actual items, because the link between the collection and the items it contains is like a shortcut. This is different from deleting a project or an item in a project, where you are managing the actual items. The Delete action is disabled from within a collection, so there is no risk of accidentally deleting an item while managing items in a collection.

Manage collection permissions

Collections are private by default. You can give other users access to a collection either by configuring the permissions for the collection directly or by granting access after sharing the collection.

Be aware that permissions for a collection do not affect the items in the collection. This means that some users might see fewer items in your collection than you do, because they don’t have permission to access all of the items. Contact the content owner or you site administrator for help with permissions for restricted items.

Configure permissions

If you are the owner of a collection, you can configure permissions for the collection. There is only one capability available for collections: View. Giving users the View capability will allow them to see the collection, but they won’t be able to add or remove items from the collection. The View capability for a collection doesn’t give users access to items in the collection that they don’t have permission to. For more information about how permissions work in Tableau, see Permissions.

1. Click the Actions menu (**...**) for the collection, then click **Permissions**.
2. Under Permission Rules, click **Add Group/User Rule**.
3. Enter a user or group name.
4. Under Template, click **View**.
5. Click **Save**.

The collection will now appear on the Collections page for the users you added. These users won’t be notified about the change in permissions, so if you want to make them aware of the collection, share it with them.

Grant access by sharing

The option to grant access appears after you share a collection if you are the collection owner.

1. Click the Share ( ) button for the collection.
2. Enter the name of the user or users you want to share with.

3. Click **Share**.
4. On the Grant Access dialog that appears, click **Grant Access**.

If the people you shared with already have access to the collection, either because you previously configured permissions or because their site role gives them access, you won’t see the grant access dialog.

Manage Web Content

Use the options detailed in this topic to edit item descriptions or perform actions like moving or renaming items on your Tableau site. The options available to you are based on your site role and permissions.

Note: If Tableau Catalog is enabled on your site, you can add descriptions to fields on a published data source page. You can also access external assets (databases and tables). Some of the options for external assets differ from the options for Tableau content (project, workbooks, views, data sources, data roles, and flows).

Starting in 2019.3, Tableau Catalog is available in the Data Management to Tableau Server and Tableau Online. For more information, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Access options

The top of the page for each item displays the name of the item and the owner or contact. Some pages might show additional information, such as a description of the item or the date the item was last modified.

You can use the icons next to the item name to:

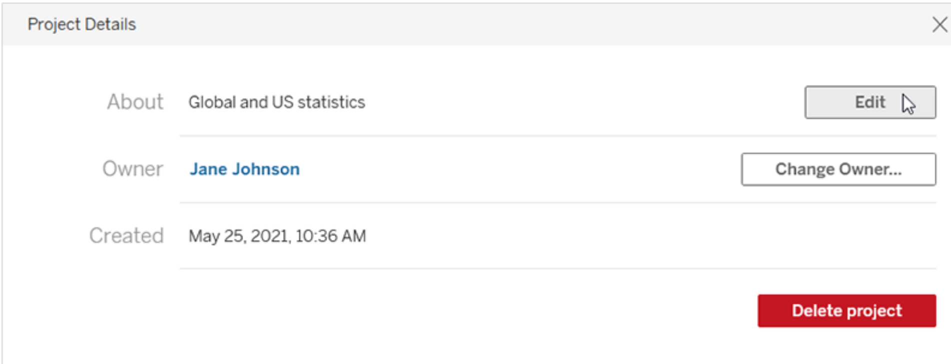
- **A** - Add the item to your favorites.
- **B** - View the item’s details.
- **C** - Perform actions on the item.

Add or edit descriptions

Descriptions make it easier for everyone in your organization to find the right items. If you own an item or have the appropriate permissions, you can edit the item’s description.

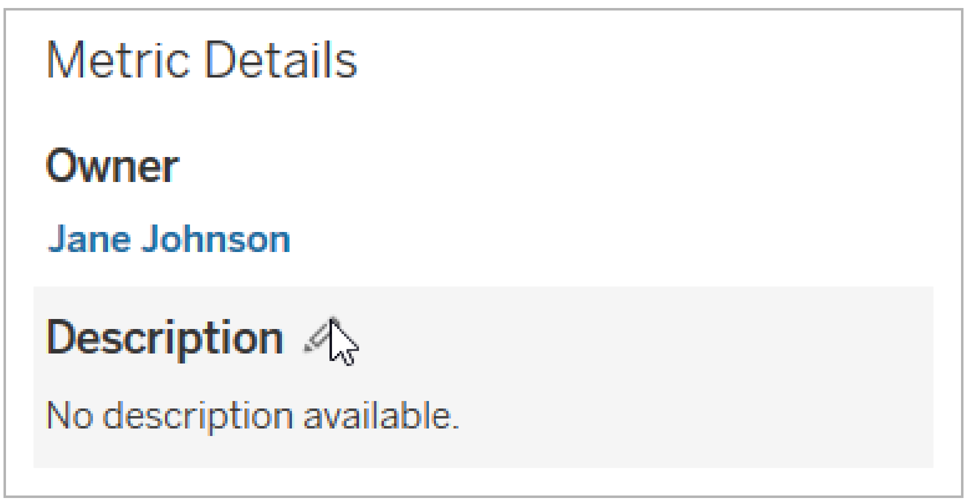
Describe projects, workbooks, and data sources

1. From the navigation pane, select **Explore**.
2. Navigate to the project, workbook, or data source you want to describe.
3. Click the details icon ⓘ.
4. To the right of **About**, click **Edit**.
5. Enter a description in the **About** field, then click **Save**.

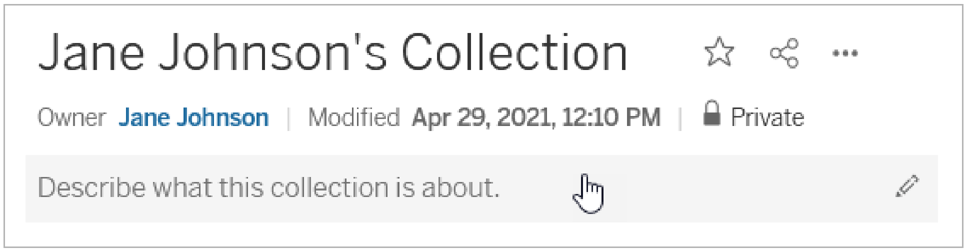


Describe metrics, data roles, Ask Data lenses, and collections

1. From the navigation pane, select **Explore**.
2. Navigate to the content you want to describe.
3. For metrics and data roles, mouse over the **Description** header and click anywhere on the gray background that appears.



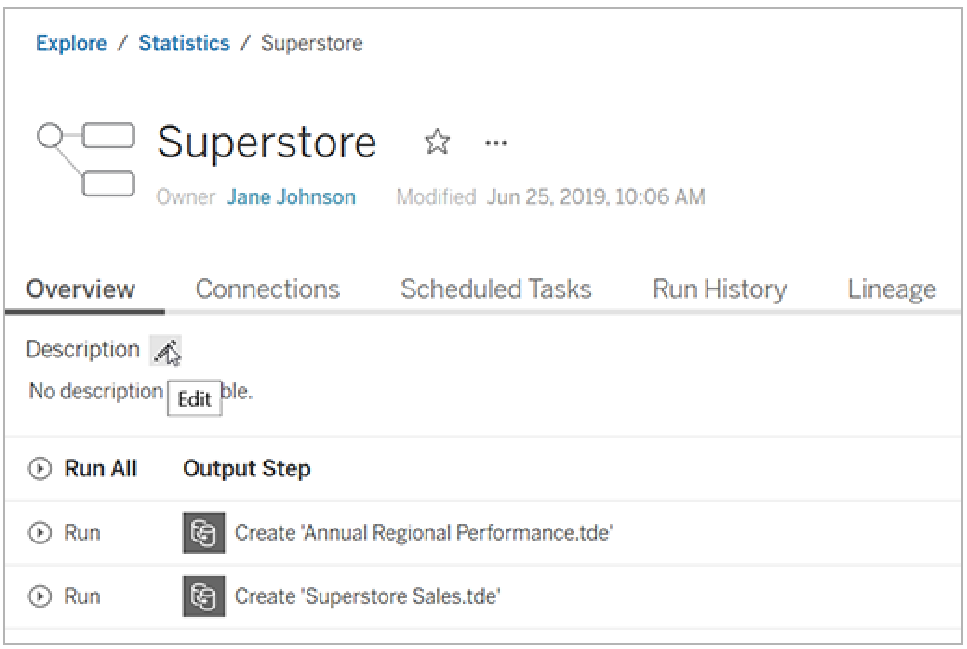
For lenses and collections, mouse over the description text and click anywhere on the gray background appears.



4. Enter a description, then click **Save**.

Describe flows

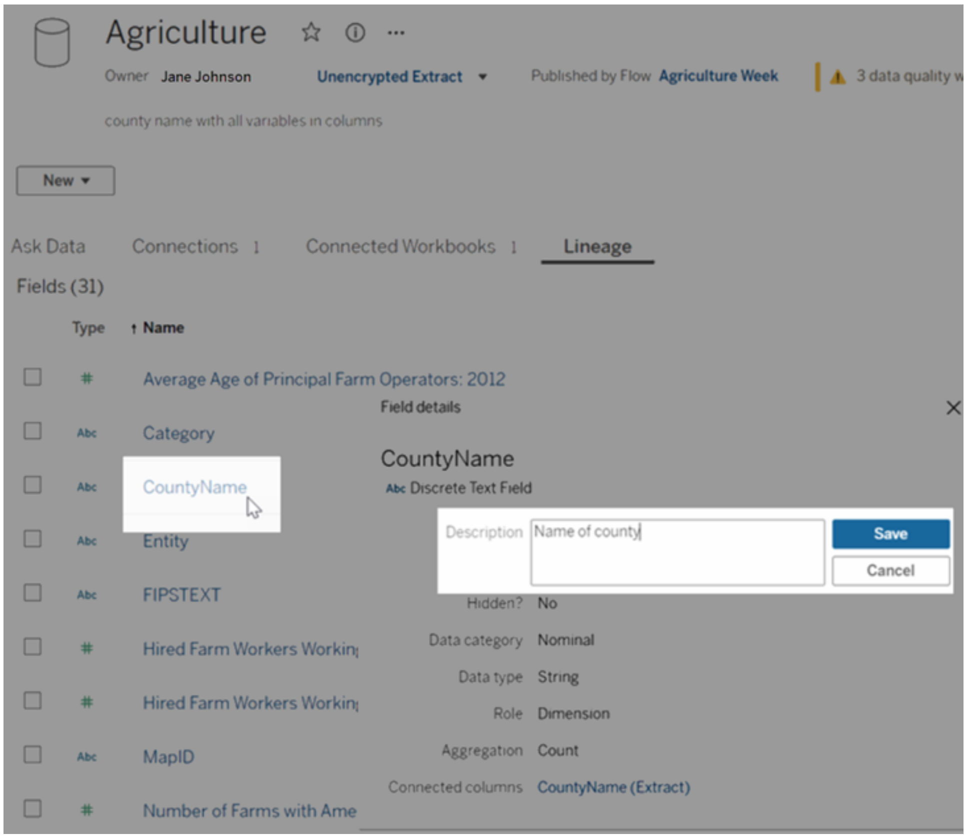
1. From the navigation pane, select **Explore**.
2. Select **All Flows** from the drop-down menu.
3. In the **Name** column, select the flow you want to describe.
4. On the **Overview** tab, click the edit icon next to **Description**.
5. Enter a description, then click **Save**.



Describe fields in a published data source

1. From the navigation pane, select **Explore**.
2. Select **All Data Sources** from the drop-down menu.
3. Click the data source name to open the data source page.

4. On the **Lineage** tab (available if Tableau Catalog is enabled), select the field you want to describe.
5. In the **Description** field of the Field details dialog box, enter a description, then click **Save**.



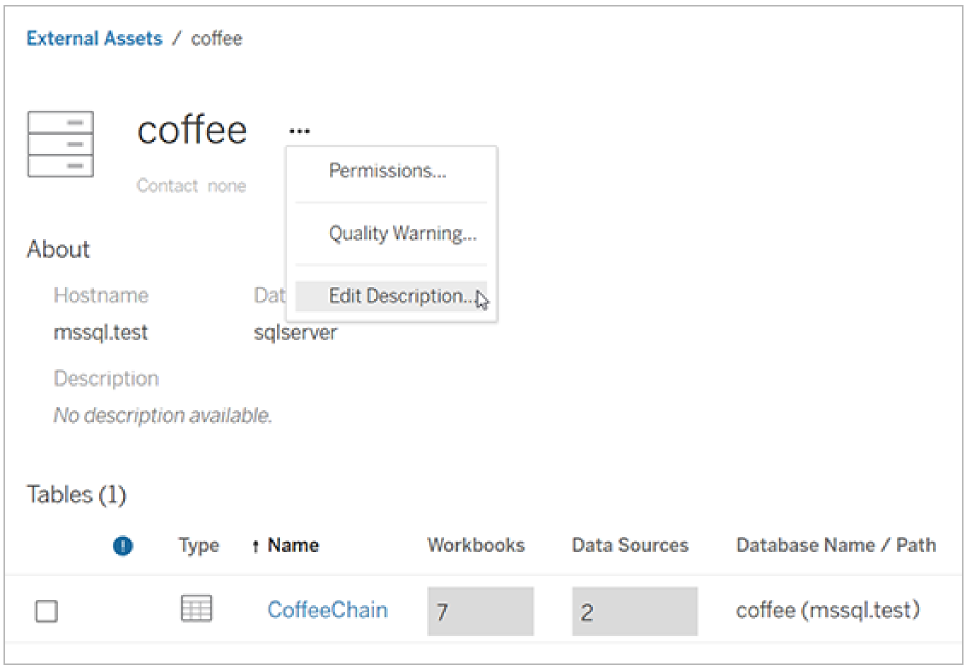
When you add the description, there might be a delay before you can see your change. After the description has been applied, you can see it in the **Description** column on the data source page **Lineage** tab. When you build a visualization, you can see the description on the **Data** tab by hovering your mouse over the field name.

Inherited descriptions appear in Desktop

If a description for a field exists upstream from the field, it appears in Tableau Desktop with information about where the description is inherited from. Inherited descriptions are not supported if your data source or workbook has more than 500 fields or is so large that the inheritance query times out.

Describe databases and tables

1. From the navigation pane, select **External Assets** (available if Tableau Catalog is enabled).
2. Select **Databases and Files** or **Tables**.
3. In the **Name** column, select the asset you want to describe.
4. From the Actions (**...**) drop-down menu to the right of the asset name, select **Edit Description**.

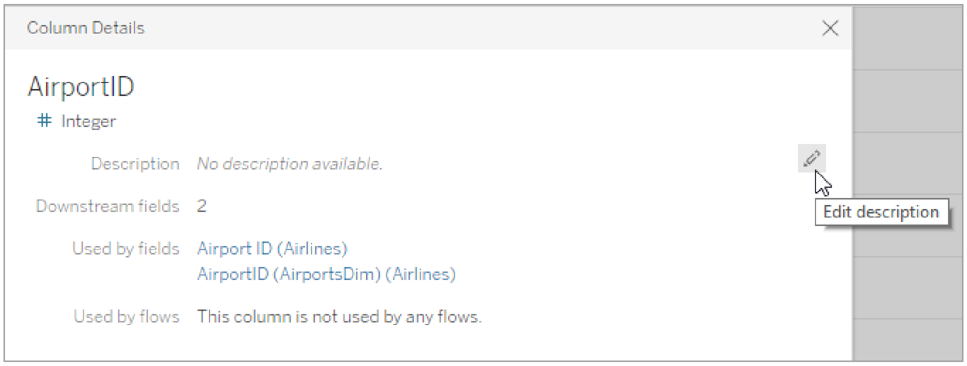


5. In the **About** field of the Edit Description dialog box, enter a description, then click **Save**.

Describe table columns

1. From the navigation pane, select **External Assets** (available if Tableau Catalog is enabled).
2. Select **Databases and Files** or **Tables**.
3. From the list, select the name of the asset with the column you want to describe.
4. On the table page, under **Columns**, select the name of the column.

5. In the Column Details dialog box, select the edit icon to the right of **Description**.

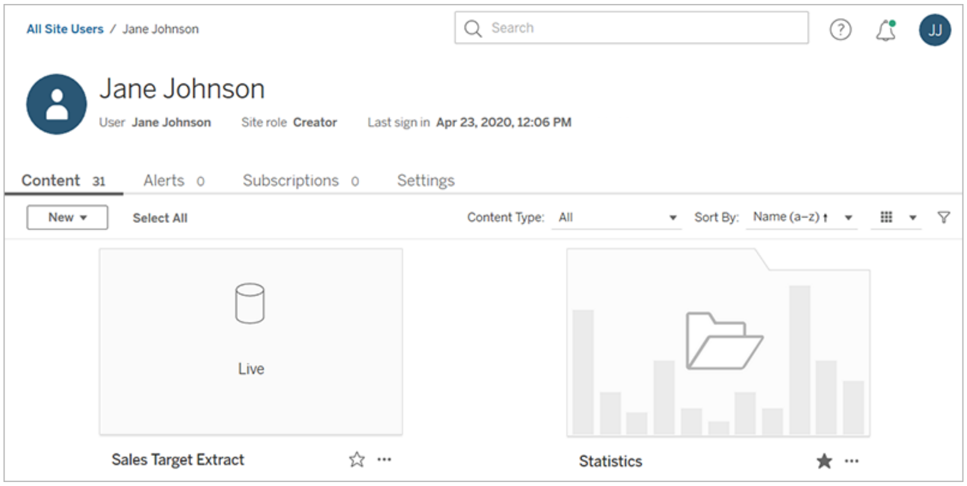


6. Enter a description in the **Description** field, then click **Save**.

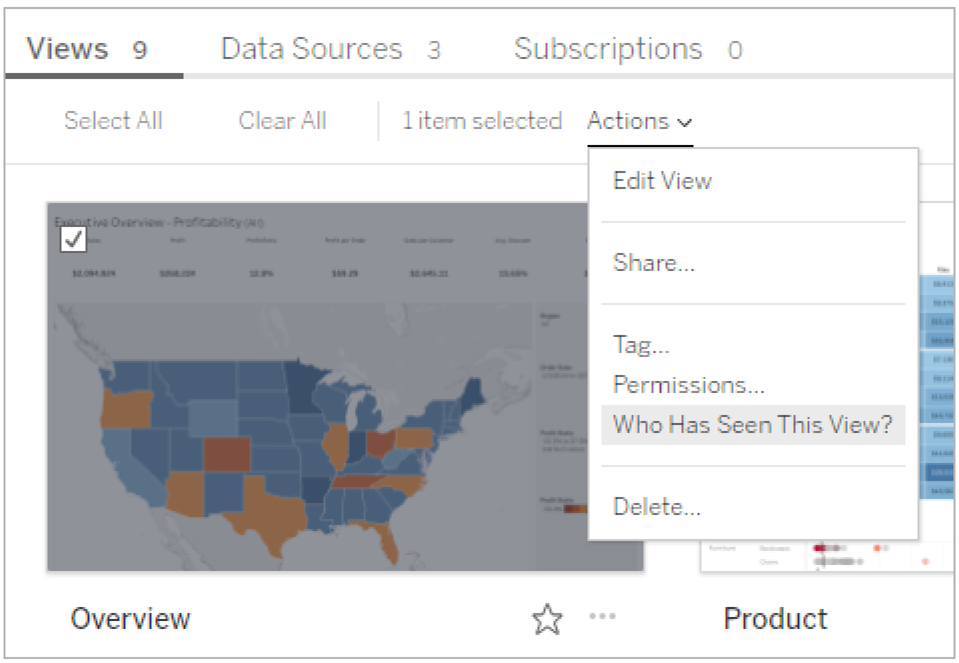
Access your content

To see the content you own, click your profile image or initials at the top of the page, and then click **My Content**.

Your content page displays your user information and the content that you own. From this page, you can also access your alerts, subscriptions, and account settings. To learn more, see **Manage Your Account Settings** on page 3233.



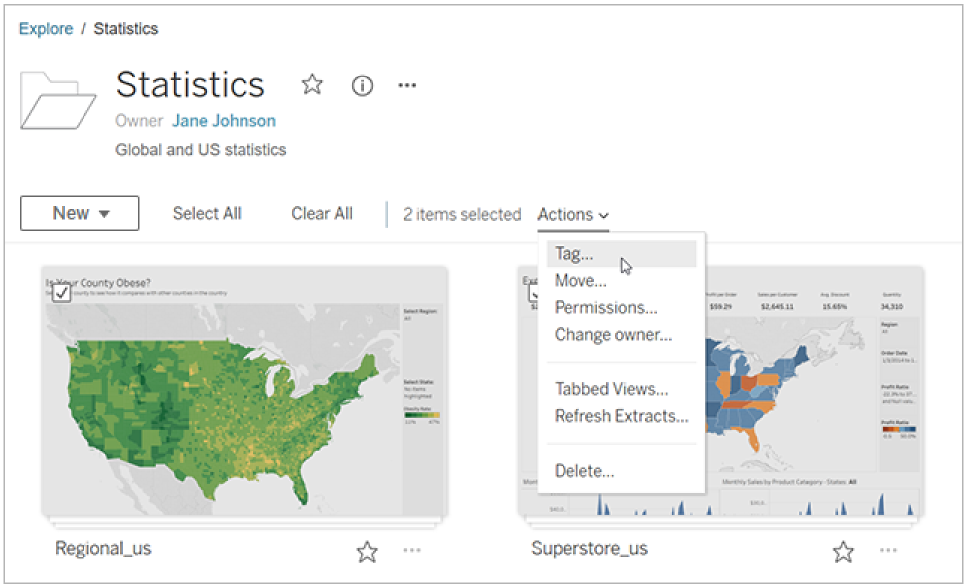
For the views you own, you can see which users have looked at a particular view by selecting **Who Has Seen This View?**.



Perform actions

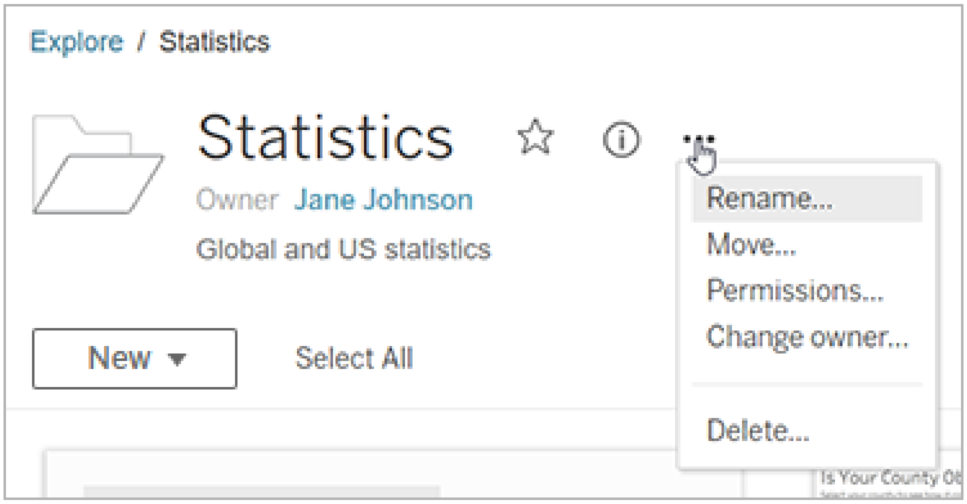
Click check boxes to select items, then click the **Actions** drop-down menu to access commands available for those items.

In you don't have permission to perform an action on the item selected, the action will be disabled. If you select multiple items, and you don't have permission to perform an action on one of the items selected, you will get an error when you try to perform the action on that item.

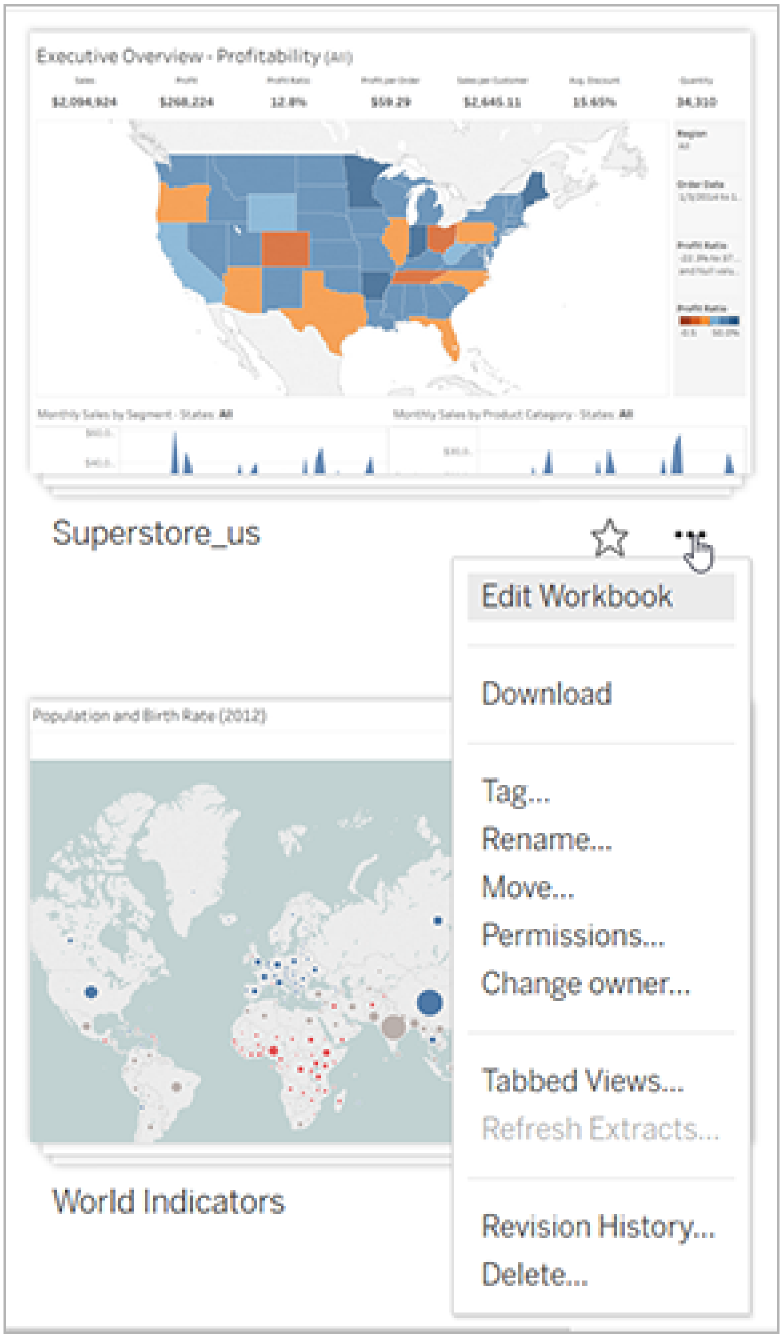


> **Note:** Different types of items have different actions available. If you select more than one type, for example, selecting a workbook and a data source, the actions you see will be limited to those in common between the types.

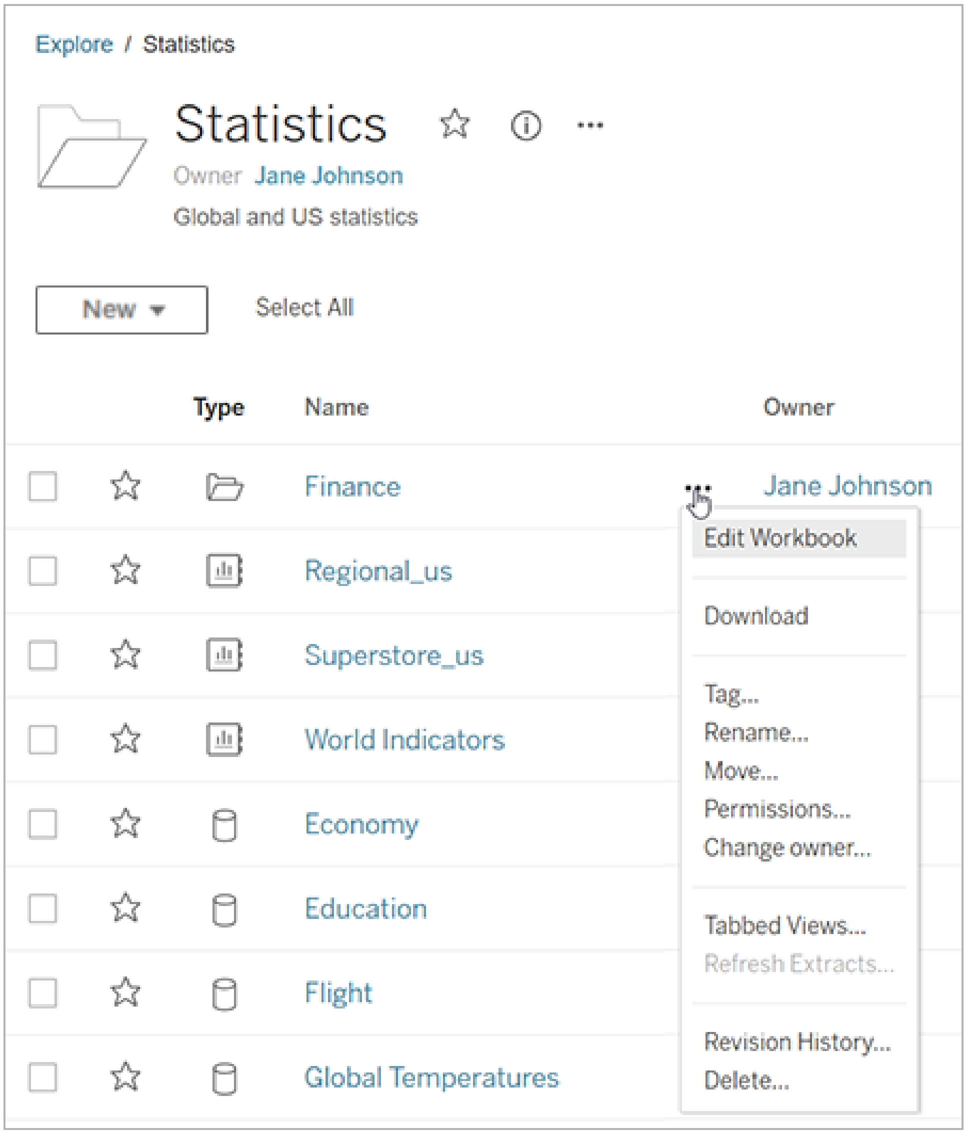
Access the actions menu for the item you currently have open by clicking the **...** at the top of the page. This actions menu is disabled if you have check boxes selected for other items on the page; instead, use the actions menu shown above to modify checked items.



In grid view, access the actions menu from the lower-right corner of a thumbnail.



In list view, access the actions menu to the right of the item's name.



Delete content

In order to delete content, you must meet one of the following criteria:

- Have a site role of Explorer (can publish) or Creator AND have the Delete capability for the content
- Be the content owner
- Be the project owner or project leader where the content is located

For more information about how permissions and site roles interact, see Effective Permissions.

Tip: Be careful when deleting content from your site. Deleting is a permanent action for all types of content except for views.

To delete content:

1. From the navigation pane, select **Explore**.
2. Navigate to the content you want to delete.
3. From the Actions (**...**) drop-down menu, select **Delete**.

Some areas of your site, like collections, recommendations, and external assets, provide links or references to items. These links or references do not affect the items themselves. You can remove an item from a collection or hide a recommendation without worrying about deleting the original item.

Consequences of deleting content

Type of content	Consequence
Workbooks, metrics, flows, and data roles	The content is permanently deleted. You can't undo this action.
Views	The view is removed from the site; however, it is still accessible if you edit or download the workbook. The effect is the same as if the view had been hidden upon publishing. For more information, see **Show or hide sheets** on page 3059.
Projects	The project, as well as any content inside the project, is permanently deleted.
Collections	The collection is permanently deleted, but the items added to the collection are not deleted.

Work with Content Revisions

When you publish a workbook or data source, a version is saved in the revision history for Tableau Server and Tableau Online. You can revert to a previous version at any time.

Required permissions for revision history

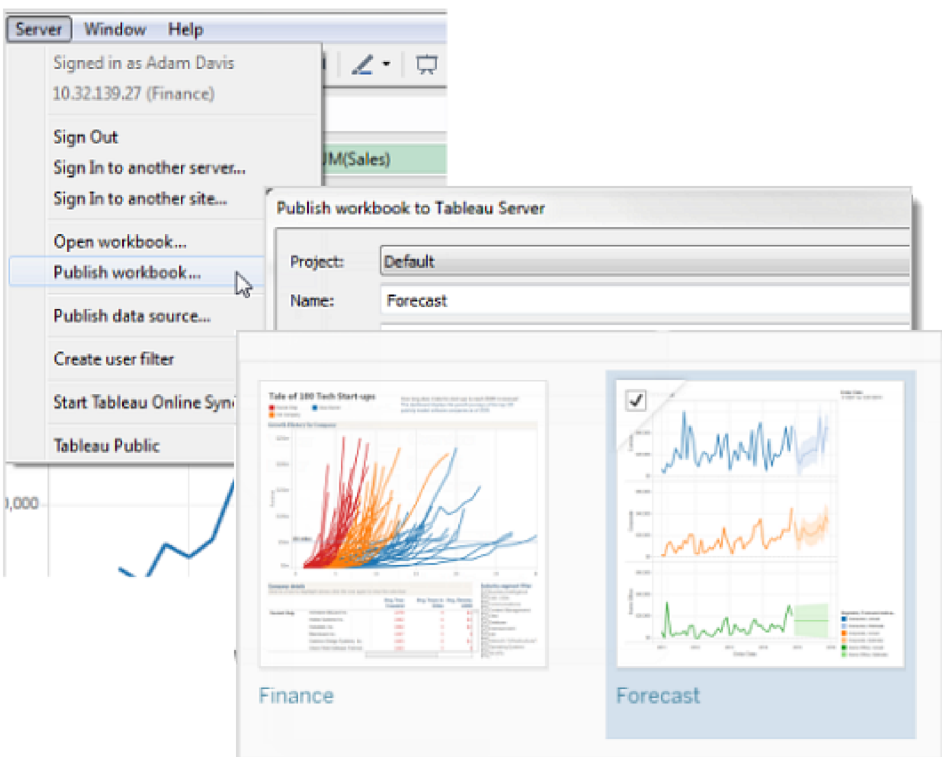
To access revision history, you must have a site role of **Creator** or **Explorer (Can Publish)**, plus the following permissions, depending on the content type:

- Project: **View** and **Save**
- Workbooks in the project: **View**, **Save**, and **Download Workbook/Save As**
- Data sources in the project: **View**, **Save**, and **Download Data Source**

Publish your content

In Tableau Desktop, click **Server** > **Publish Workbook** or **Server** > **Publish Data Source**. Make changes to the workbook or data source, and then publish it again to the same project, with the same name. (You'll need to confirm that you want to overwrite the existing content.)

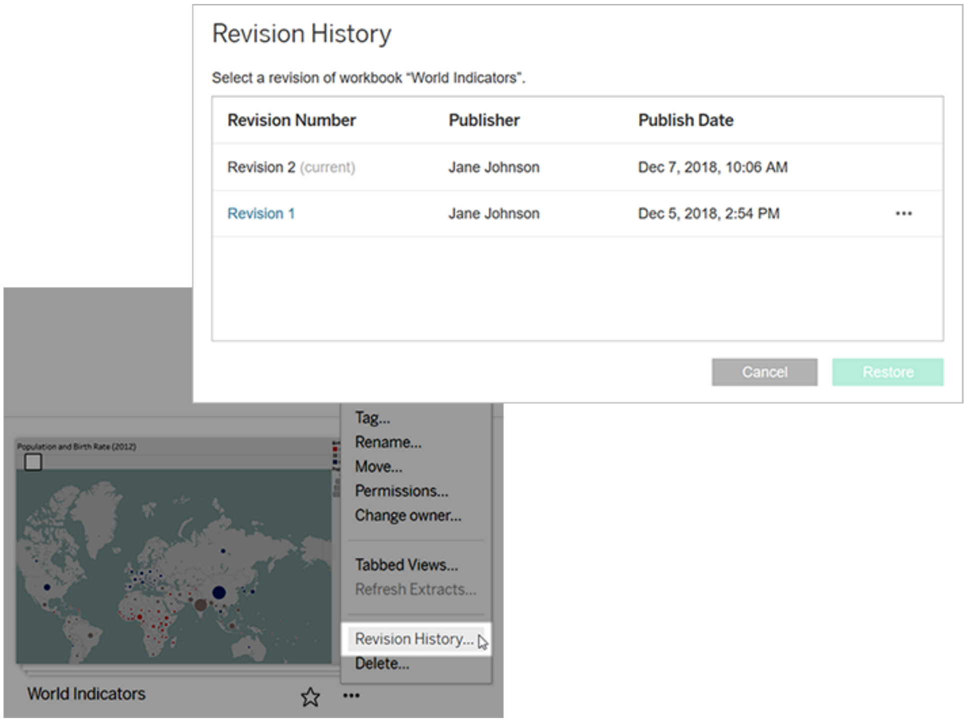
You can also save workbook revisions by editing and saving a workbook in the web-authoring interface of Tableau Server and Tableau Online.



View revision history

- Click the actions menu (**. . .**) for a workbook or data source, then click **Revision History**.

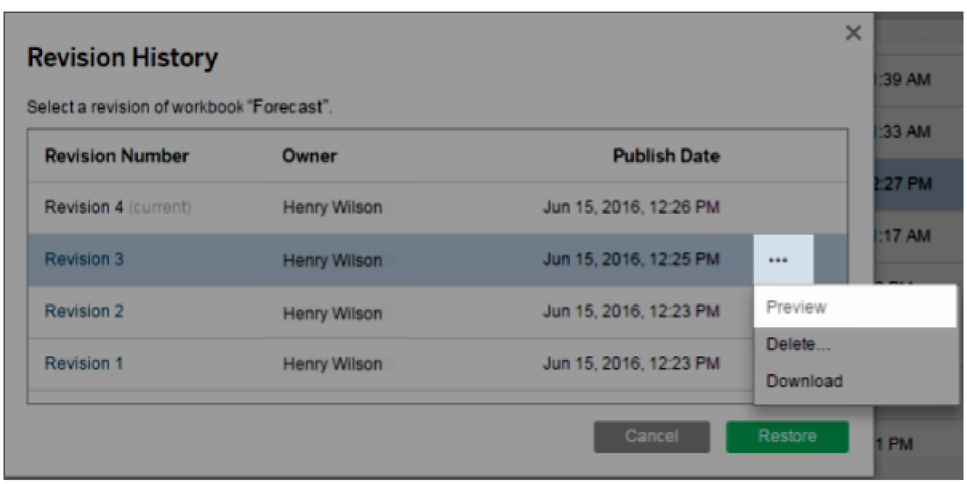
The following image shows a workbook's revision history.



Manage revisions

Preview a workbook revision

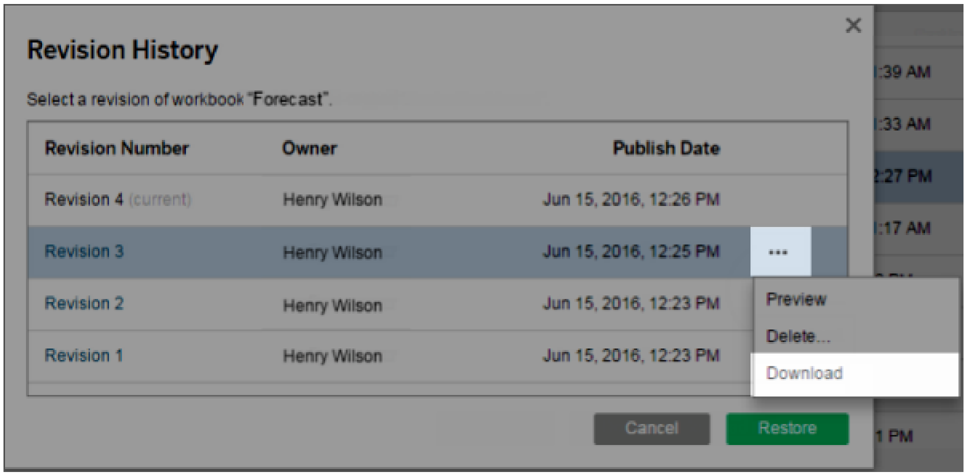
- From a revision’s actions menu (**. . .**), click **Preview**.



If a preview is available, it opens in a new tab in the browser. For workbooks that you can't preview on the server, such as those with OAuth data connections, you can download the revision and open it in Tableau Desktop.

Download a workbook or data source revision

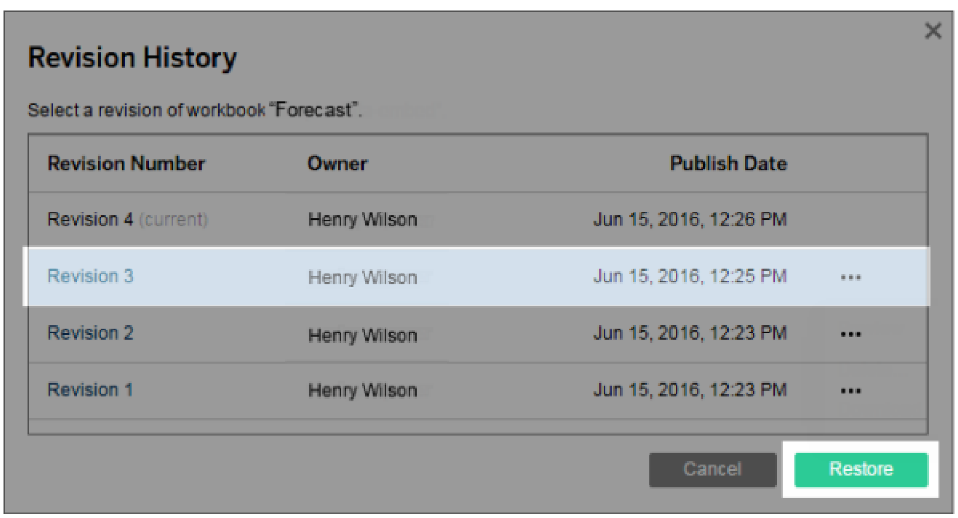
1. From a revision’s actions menu (**. . .**), click **Download**.



2. Open the downloaded file in Tableau Desktop.

Restore a workbook revision

- In the revision history, select a revision, and then click **Restore**.



The restored version becomes the current version.

> **Note**: If a revision can't be restored directly on the server, download the revision, open it in Tableau Desktop, and republish it.

Restoring workbooks that require database credentials

If a workbook uses a live data connection that prompts for a user name and password, you have the option to embed credentials for the connection. If the workbook uses a data source with multiple connections, you might need to provide credentials for each one.

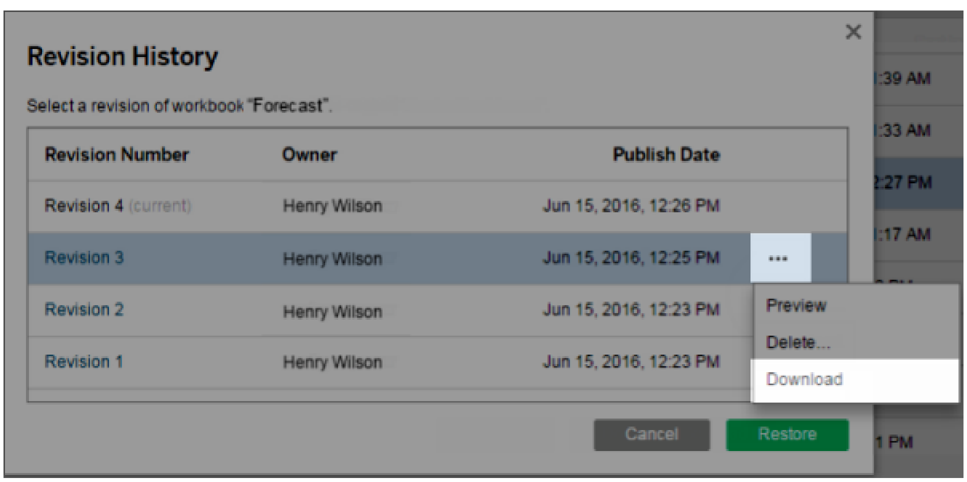
If a workbook uses data extracts with scheduled refreshes and embedded credentials, you'll need to edit the data connection to provide the credentials.

Restore a data source revision

For a variety of reasons—for example, extracts commonly are on refresh schedules—a previous revision of a data source will not include the extract as it was published at the time. To restore the extract, you can download the previous revision of the data source (in .tdsx format), open it in Tableau Desktop, and re-publish the extract.

For more information, see **Data changes and deletions** on the next page under **Potential revision history issues** on the next page.

1. Sign in to the Tableau Server or Tableau Online web editing environment, navigate to the data source, and open its revision history.
2. From a revision's actions menu (**. . .**), click **Download** .



3. Open the downloaded file in Tableau Desktop, and then republish it with the same name,

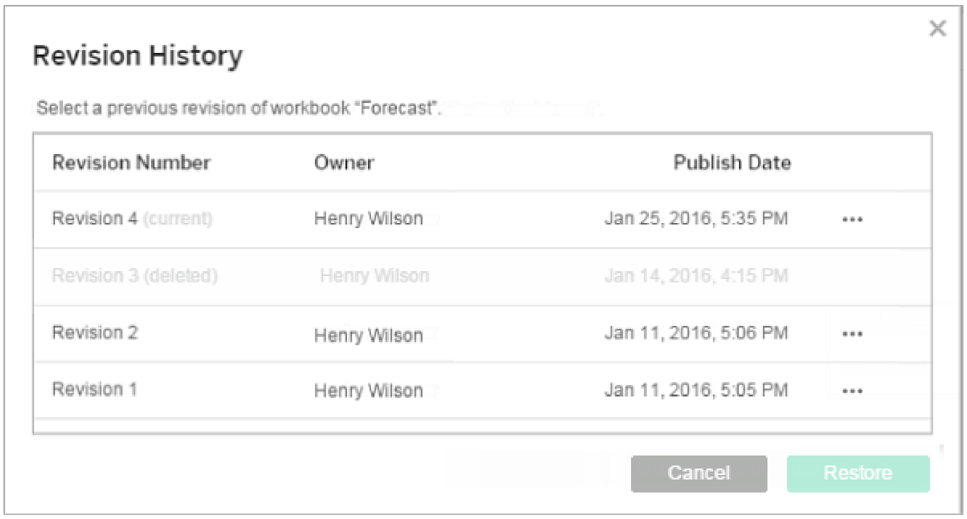
to the same location on your Tableau Server or Tableau Online site (confirming that you want to overwrite the existing version).

The uploaded version becomes the most current version.

Delete a workbook or data source revision from history

1. From a revision’s actions menu (**. . .**), click **Delete**.

   The revision history list updates to indicate a revision has been deleted.



Potential revision history issues

Overwriting content with the same name

If a different author publishes over a workbook or data source with the same name, the most recent author becomes the owner of the content and can see its entire revision history.

Data changes and deletions

- Workbooks and data sources are downloaded with the latest configuration of their extract or data connection. If the data model or data connection has changed between revisions, you might need to update the downloaded workbook or data source.
- Revisions of workbooks and data sources that use .xls or .csv files are saved with an extract of that data.
- Revisions are saved for .TDE files or hyper files with a direct connection. Revisions are

not saved for extracts.

- If a workbook or data source is deleted from a site, all revisions are also deleted.

Turning revision history on and off

On Tableau Server, server administrators can disable revision history for specific sites.

If revision history is turned on and then off, saved revisions are retained, and new versions overwrite the latest version. If revision history is then turned on again, version numbering starts from the last saved revision.

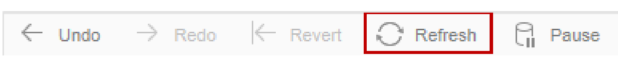
If you’re a Tableau Server administrator, see Enable Users to Save Revision History for more about revision history settings.

Refresh Data or Pause Automatic Updates

Tableau Server and Tableau Online let you control how views interact with data sources so you can keep data fresh and improve performance.

Refresh Data

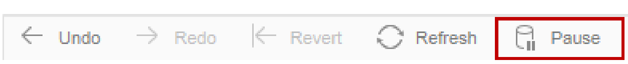
If a data source is changed (for example, with new fields, field names, or data values), the view reflects those changes the next time you load the page. To manually update data while you interact with a view, click **Refresh** on the toolbar.



When you refresh data, Tableau Server and Tableau Online clear any cache and retrieve the latest information from the data source. If you're working with a complex view or a very large data source, refreshing data can take a long time.

Pause Automatic Updates

As you interact with a view, the server sometimes has to query the data source to update the view. To temporarily stop updates so you can more quickly interact with a view, click **Pause** on the toolbar.



Click **Resume** to again automatically query the data source as needed.

Use Custom Views

A custom view is a saved version of the view with your selections and filters applied. Since views are interactive, custom views let you save your filters, sorts, or selections, without making changes to the underlying view.

You might create a custom view if you find yourself changing a Tableau Server or Tableau Online view every time you open it. For example, if you apply a filter to include only data relevant to you, or you sort data differently, you can save separate custom views for each scenario.

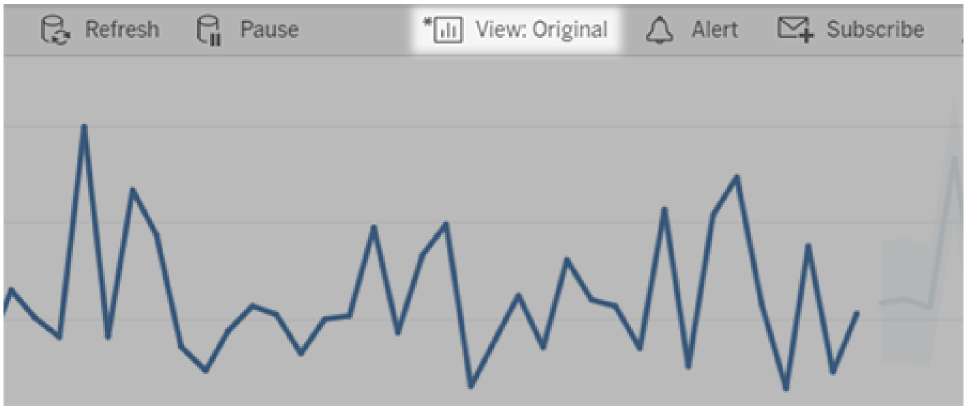
> **Tip:** A custom view doesn't change the original view, but it remains related to it. If the original view is updated or republished (with different filters, for example), your custom view is updated too. Likewise, if the original view is deleted from the server, your custom view is deleted.

To make changes to the underlying workbook, see **Edit Tableau Views on the Web** on page 3149

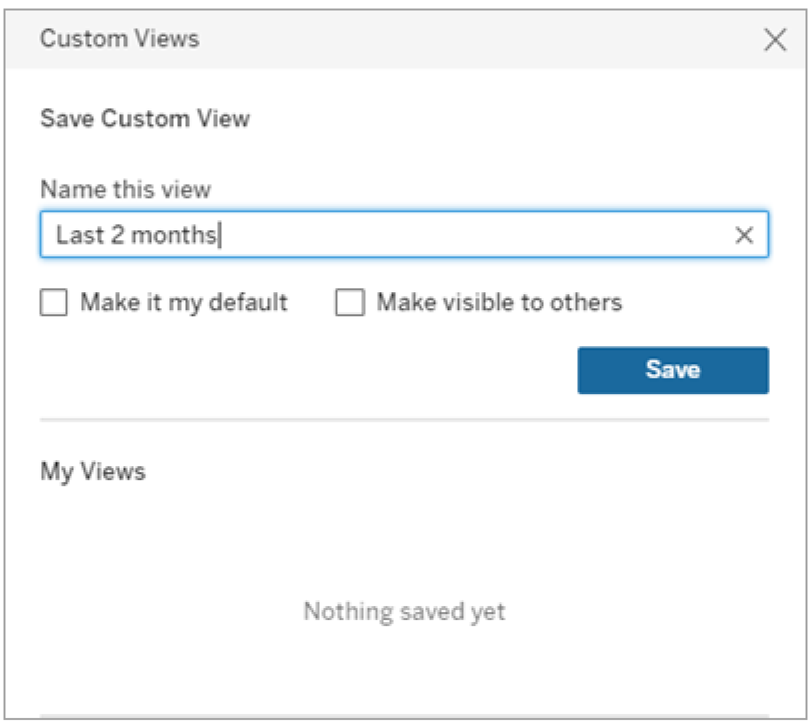
Create a custom view

1. Open the individual view that you want to customize.
2. Filter the data, change sorting, select specific marks, zoom in or out, or make any other modifications.

   In the toolbar, the View option now includes an asterisk, indicating that you made changes to the view.



3. Click **View** in the toolbar.
4. In the Custom Views dialog, enter a name for the custom view. Then select whether you want it to be the default view you see, and if it should be visible to others who view the workbook.



5. Click **Save**.

With your custom view open, click the **Share** button in the view's toolbar to share it directly with other users. To add it to the list of public custom views shown on the workbook, see Make a custom view public or private.

See custom views

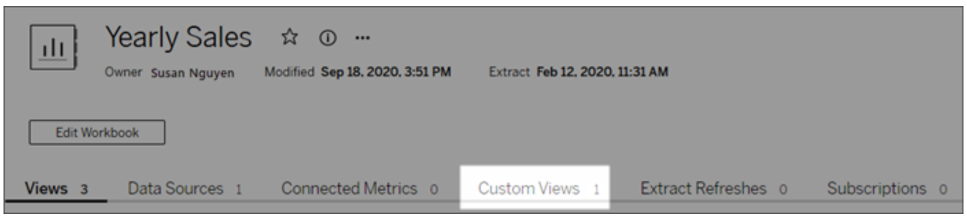
In Tableau Online and Tableau Server, you can see custom views created on a workbook or on individual views

See and manage custom views on a workbook

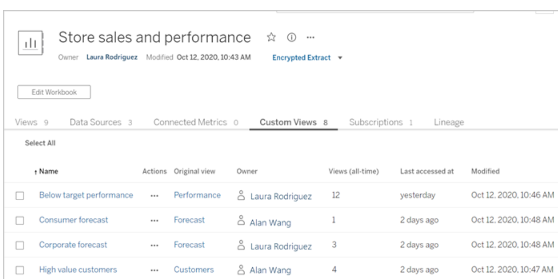
You can see all the custom views you created on a workbook in the custom views menu by selecting View from the toolbar, or on the custom views tab of the workbook page.

To see a workbook's custom views:

1. Select the workbook you'd like to view.
2. Click the Custom Views tab to see available custom views.



Select the custom view you'd like to see or filter available views my name, owner, original title, and more.



If you have publish permission on the workbook, you can see all custom views created for the workbook. Site and server administrators can delete other user's custom views or change the owner.

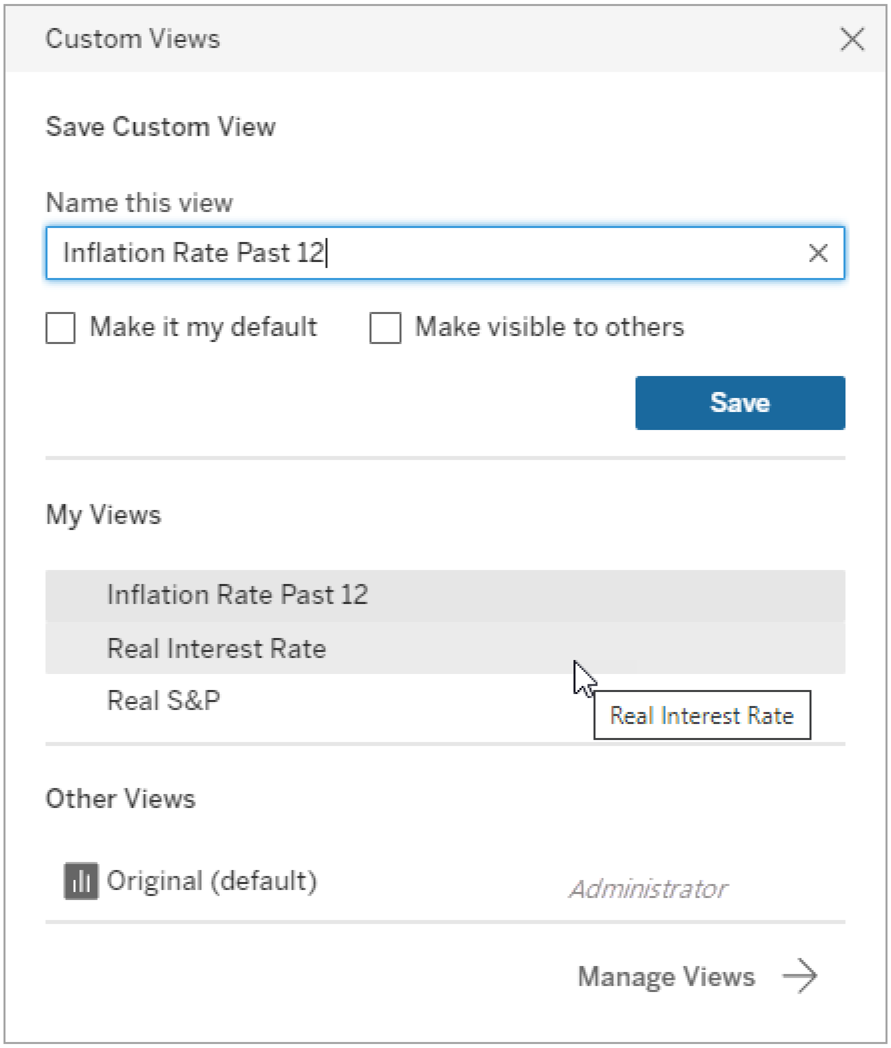
See saved custom views

With a view open, do the following:

1. Click **View** in the toolbar.

   The currently selected view appears with a gray background.

2. Click another view to display it.

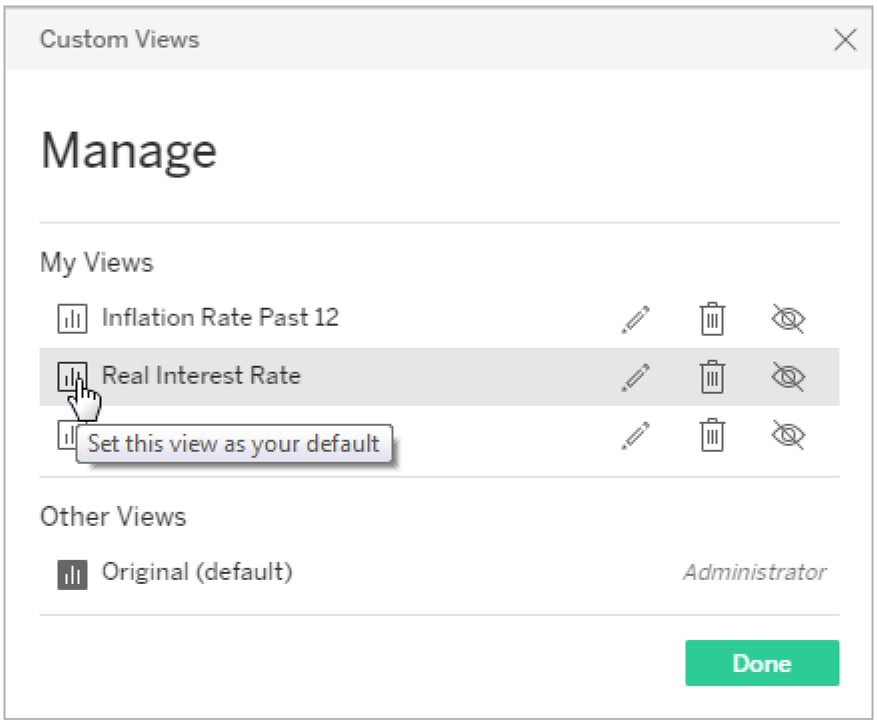


   Custom views that you have saved appear under **My Views**. The original view and views created by others appear under **Other Views**.

Change your default view

1. Click **View** in the toolbar, and then click **Manage** in the lower right corner of the **Custom Views** dialog box.

2. Click next to the custom view name to make the view the default view. The icon changes to to indicate that the view is now the default.



3. Click **Done**.

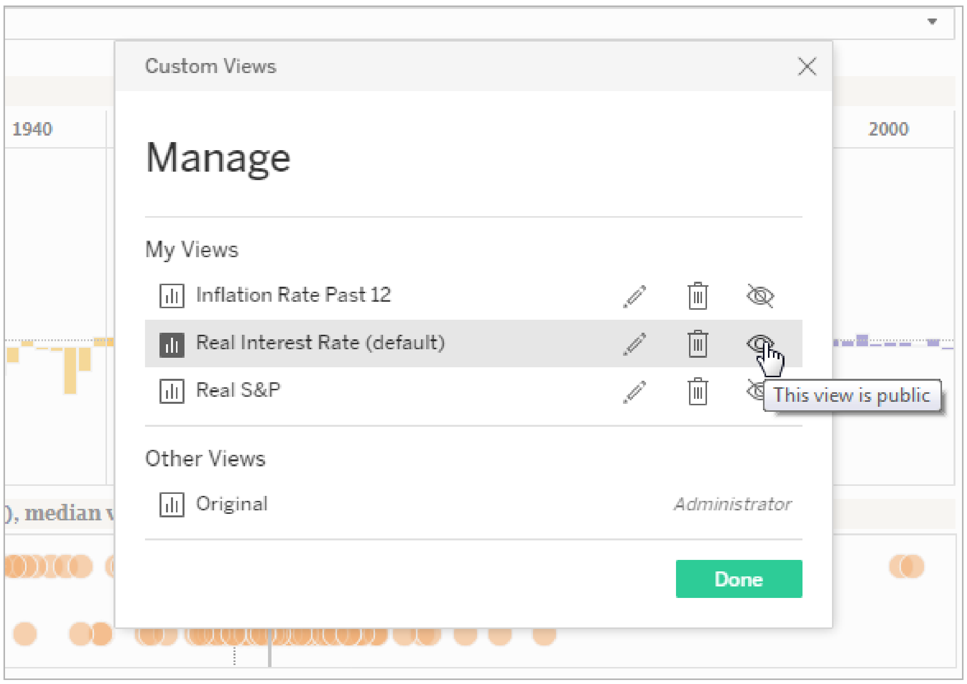
Make a custom view public or private

By default, a custom view is private so only you see it in the Custom Views dialog box. But if your site role is Explorer or Creator, and if you have the correct permissions on the workbook, you can make a view public so anyone with access to the original view will see your custom version in the “Other Views” list.

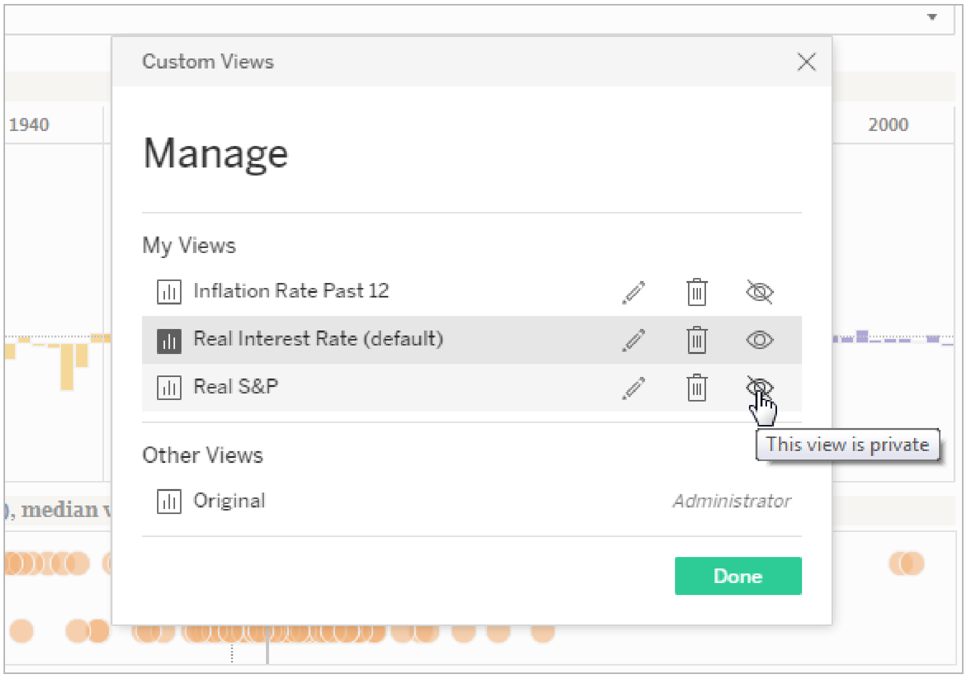
Tip: Private custom views are hidden only from the “Other Views” list. You can share the URL with anyone who has permission to access the original view.

1. Click **View** in the toolbar, and then click **Manage** in the lower right corner of the **Custom Views** dialog box.
2. Click the public or private icon to change the view's status.

   - The public icon indicates that the view is visible to others in the Custom Views dialog box:



- The private icon indicates that the view is visible only to you in the dialog box:



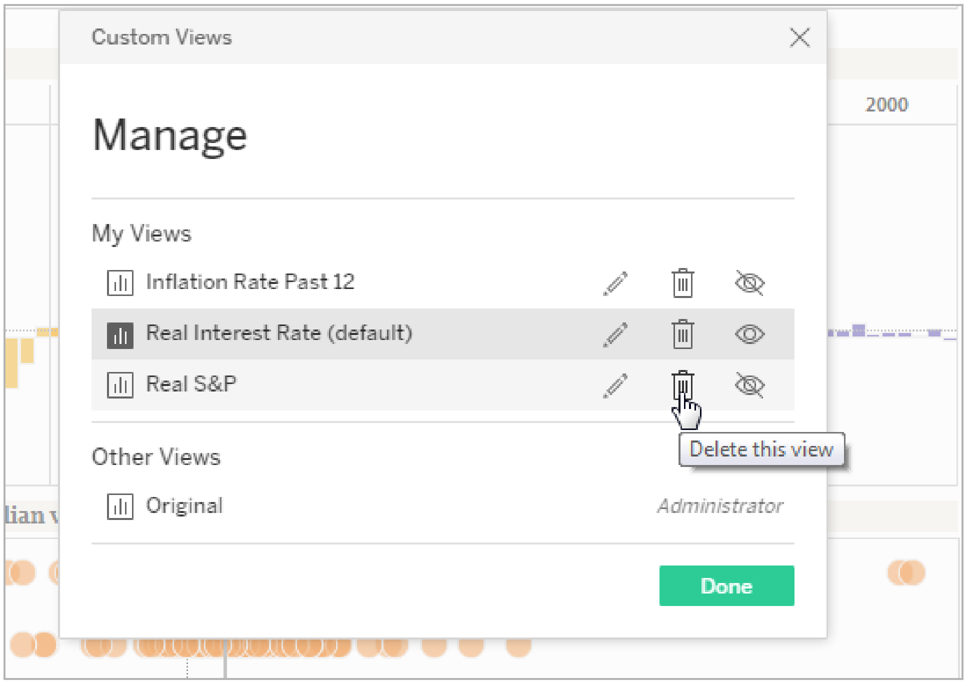
3. Click **Done**.

Delete a custom view

You can delete a custom view you created at any time. Removing your custom view does not affect the original view.

1. Click **View** in the toolbar, and then click **Manage** in the lower right corner of the **Custom Views** dialog box.

2. Click the delete icon next to the custom view name.

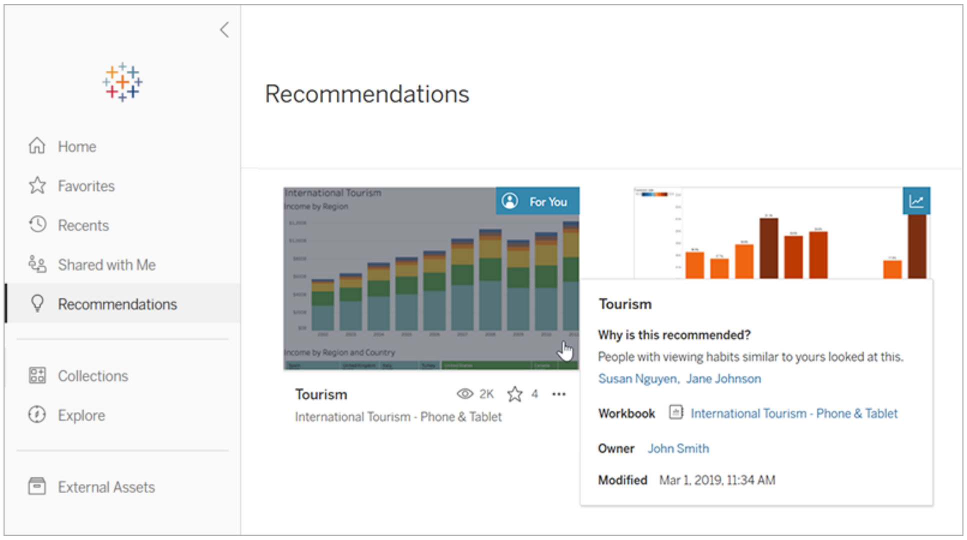


3. Click **Done**.

Note: Deleting a custom view also deletes any subscriptions based on that custom view.

Discover Recommended Views on Your Tableau Site

Recommendations make it easier to discover relevant content on your Tableau site. Recommendations for views appear both on the Home page and on the Recommendations page, accessed from the left navigation pane.



Why are these views recommended?

The model used to recommend content looks at the viewing habits of users and trends in popularity of content on your site. Recommendations that are For You are personalized based on the content you have looked at. Recommendations that are Trending have been popular across your site in the past week.

Tip: If a recommendation isn't useful, you can hide it by selecting the actions menu (**...**), and then selecting **Hide**.

Whose names are shown?

On recommendations that are For You, you might see the names of other users who have looked at that content. These are users on your site who have viewing habits similar to yours. The names give you an additional indicator of why you might want to investigate a view, because a colleague with similar interests looked at it.

The visibility of names on recommendations is controlled by a site-wide setting. If you don't see names on a recommendation For You, the setting might have been turned off for your site.

Which views appear as recommendations?

You will see recommendations for particular views only if you have the correct permissions to see those views. You can't exclude a view from being recommended to others on your site; however, you or your site administrator can set permissions so that only users who should have access to a workbook or view are able to see it. That way, the view will not show up in recommendations for the wrong users.

Manage Your Account Settings

In Tableau Server and Tableau Online, your Account Settings page lets you manage your credentials, email settings, and user interface options in one central place.

Go to your Account Settings page

At the top of a page, click your profile image or initials, and then select **My Account Settings**.

To change your profile image, click the current image or initials on your account settings page.

Manage your credentials and passwords

When you access a workbook or data source that has a live connection to data and requires you to sign in, Tableau offers to save your password for you. If you accept, it stores your credentials in a cookie or an access token, depending on the data type. You can remove these credentials if you no longer use the data, or you have exceeded the maximum number of saved credentials and want to make room for a new one.

Under **Saved Credentials**, do either of the following:

- Select the **Delete** link next to an individual access token.
- Select **Clear All Saved Credentials**.

When you clear all credentials, the following items are removed from your user account:

- Passwords you used to access published data sources or workbooks that connect to them.
- Access tokens for OAuth data connections, such as to Google or Salesforce.com data.

  Caution: Removing an access token is effectively like “changing the locks.” If the token is stored with workbooks or data sources you published, deleting the token also removes access to the data from those workbooks and data sources. If the token is embedded in an extract connection, and the extract is refreshed on a schedule, the refreshes will not be able to complete until you embed the new credentials or access token into the connection.

Manage multi-factor authentication verification methods

After Tableau with MFA has been enabled and you've register your verification method, you use your TableauID credentials and verification method each time you sign in to Tableau Online.

To add or remove verification methods, you can click the **Manage My Verification Methods** link to do the following:

- Add additional verification methods for backup purposes
- Delete existing verification methods if you no longer need them

For more information, see Manage verification methods in Tableau Online Help.

Create and revoke personal access tokens

Create personal access tokens (PATs) to authenticate automated tasks with REST API. We recommend creating a token for each automation task that requires authentication. By creating multiple tokens in this way, managing many automated tasks is simplified in the event you need to remove a task. To remove a task immediately, you can just revoke the personal access token associated with the task.

Note: If multi-factor authentication (MFA) is enabled with Tableau authentication, PATs are required. You must use a PAT, instead of user name and password, to make a REST API sign in request to Tableau Online. For more information, see Tableau with MFA in Tableau Online Help.

Creating personal access token

This procedure requires you to copy a secret token to a file. The secret token is the string that you will include in your automation scripts and will be used to authenticate to Tableau Server. Treat the secret token as you would a password: protect the secret token and do not share it with others.

1. Under **Personal Access Tokens**, enter a descriptive name for your token in the **Token Name** field, and then click **Create new token**.
2. In the resulting window, click **Copy to clipboard** and then close the window.
3. Paste the token secret to a file. Store the file in a safe location.

Revoking a personal access token

1. Under **Personal Access Tokens**, identify the token name that you want to revoke.
2. Click the **Revoke** link next to token name.
3. In the resulting window, click **Delete** to revoke the token.

Remove connected clients

The first time you sign in to Tableau Server or Tableau Online from a Tableau *client* like Tableau Desktop or Tableau Mobile, a secure token is created is stored in your account. This token allows you to access the site directly, without having to sign in.

You can delete a connected client (token) if you no longer use it, or if you want to add a new client but get an error that says you're already using the maximum number of connected clients. After you remove a connected client from your account, you will need to provide your credentials the next time you access Tableau Server or Tableau Online from that client.

- In the **Connected clients** section, next to the client you want to remove, select **Delete**.

Change notification settings

Notifications are messages that let you know when something happens to content you own or work on. You can receive email notification or see them on your Tableau site by selecting the bell icon in the upper right corner of your browser.

For comment mentions, shares, flow runs, extract jobs, and webhooks notifications, under Notifications, select **Email**, **On Tableau**, or both.

Change subscription settings

1. Under **Subscription time zone**, select the time zone for schedules you create.
2. To modify subscriptions, click **Subscriptions** at the top of the page, and then select a

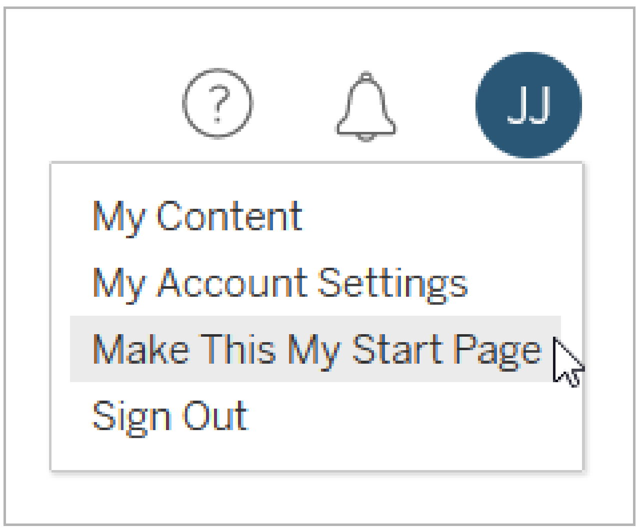
workbook or view.

3. From the **Actions** drop-down menu, select **Change Schedule**, **Change Subject**, **Change Empty View Mode**, or **Unsubscribe**.

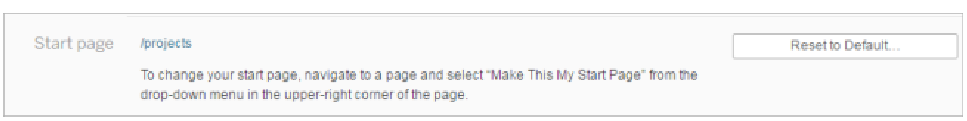
   (The empty-view option sends subscription emails only when data exists in a view. It’s a good choice for high-priority alerts.)

Change your start page

To change the start page that appears when you sign in, navigate to the page you want, click your name in the upper right area of the page, and then click **Make This My Start Page**. Updates to your start page take effect the next time the page loads completely, or after you sign out, and then sign in again.



To return to the default start page, click your user name, and then click **My Account Settings**. In the start page section, click **Reset to Default.** (The URL for your current start page is displayed here too; click the link to go to the page.)



Change language and locale

The **Language** setting determines the language you see for user interface options. **Locale** affects views, such as how numbers are formatted, or which currency is used.

Change the **Language** and **Locale**, and then click **Save Changes**. The language and locale update immediately.

If you use Tableau Server and want to select a language that isn't currently supported, contact your administrator.

Enable or disable extract refresh failure emails

If you own published data sources that contain extracts on a refresh schedule, you can receive email notification when the scheduled refresh does not complete successfully.

- Under **Refresh Failure Notification**, select the **Send email when extract refreshes fail** check box to opt in, or clear it to opt out.

The email contains the following information:

- Extract or workbook name.
- The last successful refresh time.
- The number of consecutive times the refresh has failed.

  After five consecutive failures, refreshes are suspended until you take an action to resolve the cause of the failure.
- A suggested action to take to address the cause of the failure—such as updating embedded credentials or a path to the original data file—and a link to the server to take the action.

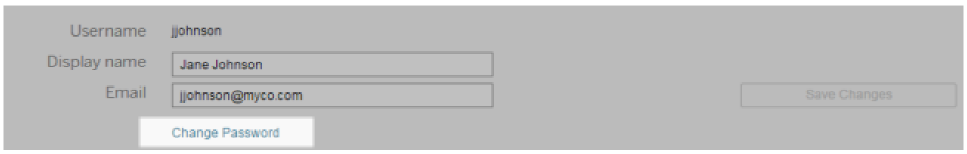
Change your display name, password, or email address for Tableau Server

Change your display name

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your display name. Select the display name text and enter the new display name, and then click **Save Changes**.

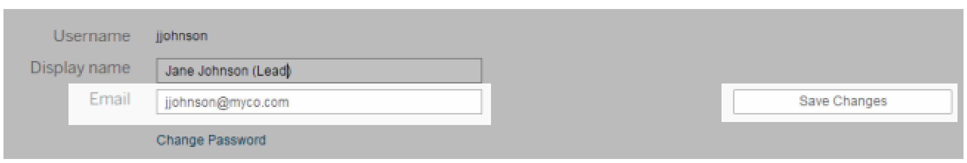
Change your password

If the server is configured to use the internal user management system (Local Authentication) instead of Active Directory, you can change your password by clicking **Change Password**. Click **Save Password** to save your changes.



Change your email address

If you subscribe to views or receive data-driven alerts, the related email account appears on the Account Settings page. Enter the new email address in the **Email** text box, and then click **Save Changes**.



Change your display name or password for Tableau Online

If your site is not set up for single sign-on, your Tableau Online display name and password are based on your TableauID account. Your TableauID allows access to Tableau Online, the Tableau website, the Customer Portal, and other resources.

If you forgot your password

To reset your password, go to **https://online.tableau.com**, click **Forgot password**, and enter the email address you use to sign in to Tableau Online. Then follow the instructions in the email you receive.

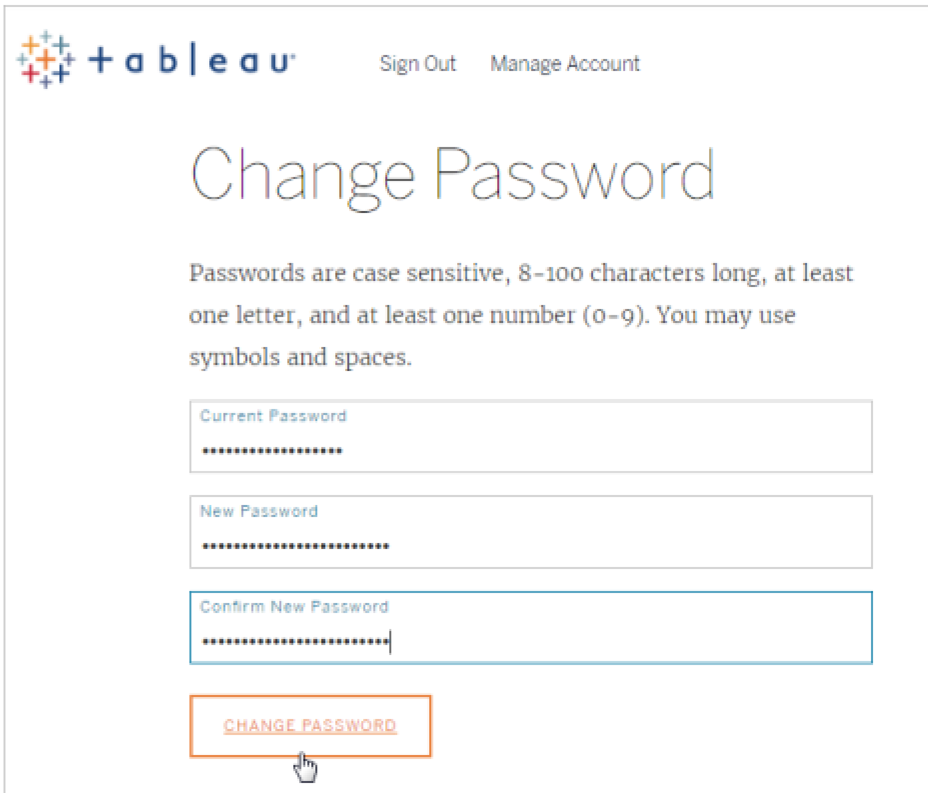
If you’re signed in to Tableau Online

1. Open your **Account Settings** page, and click **Change Password**.

   This redirects you to the Tableau website.

2. If prompted, sign in using your Tableau Online credentials, and then select the **Change Password** link at the top.

   - In the fields provided, type your current and new passwords, confirm the new password, and click **Change Password**.



   - To change your display name, click **Manage Account**.

On the User Profile page, change your first or last name, update any other information that might be out of date, and click **Update**.

Note: When you change your display name or password, you're directed outside of Tableau Online to your TableauID account profile. You can also access your TableauID profile by navigating to the Tableau website and clicking **Sign In**.

Manage Saved Credentials for Data Connections

Saved credentials enable you to connect to a data source without being prompted for your credentials. The credentials saved for your connection can be OAuth access tokens, or other credentials, such as user name and password.

Note: If you do not see the Saved Credentials section, consult with your about allowing saving credentials.

Remove saved credentials

To remove Tableau access to data, delete the associated saved credentials for that data from your account.

After you delete the credentials, you will need to sign in to the data the next time you access it. This will create new saved credentials.

Your administrator might choose for all users to use the same shared credentials for connecting to a data source. If this is the case, the saved credential is associated with the data connection for all users, and it does not appear under Saved Credentials on your Account Settings page.

Note: If you're a Tableau Server user and can't delete saved credentials, ask your administrator if they've cleared the **Allow users to save data source access tokens** option in the server settings.

Test connections using saved credentials

If the connector supports test functionality, you can test the connection using saved credentials.

1. While you're signed in to Tableau Server or Tableau Online, display your Account Settings page.
2. In the Saved Credentials section, click the **Test** link next to the stored connection that you want to test.

This test confirms that Tableau Online or Tableau Server can access your account using this corresponding saved credential. If the test succeeds, but you cannot access your data through this managed connection, confirm that the credentials you provided for this connection can access your data.

For example, if you accidentally created the connection using your personal Gmail account, but you use a different account to access a Google Analytics database, you will need to delete the saved credentials and sign in to the data using the appropriate Gmail account.

Update saved credentials

To help ensure uninterrupted data access from existing Tableau content after a custom OAuth client has been configured for your site, we encourage you to update your saved credentials. To update saved credentials, you can delete the previous saved credentials for a particular connector and then add it again.

When you add saved credentials again, both new and existing Tableau content will access the data using the custom OAuth client configured by your .

1. Sign in to Tableau Server and navigate to your **My Account Settings** page.
2. Under **Saved Credentials for Data Sources**, do the following:
   1. Click **Delete** next to the saved credentials for a connector.
   2. Next to the same connector, click **Add** and follow the prompts to 1) connect to the custom OAuth client that your site admin notified you about and 2) save the latest credentials.

Clear all saved credentials

When you select **Clear All Saved Credentials**, the following items are removed from your user account:

- All saved credentials for connections that are stored in your account.

  > **Caution:** If any of these saved credentials are stored with published workbooks or data sources, deleting them also removes access to the data source from those locations. Effectively, this is like "changing the locks" anywhere the affected saved credentials are used.

- Passwords you have used to access published data extracts or workbooks that connect to them.

Share and Collaborate on the Web

Share Web Content

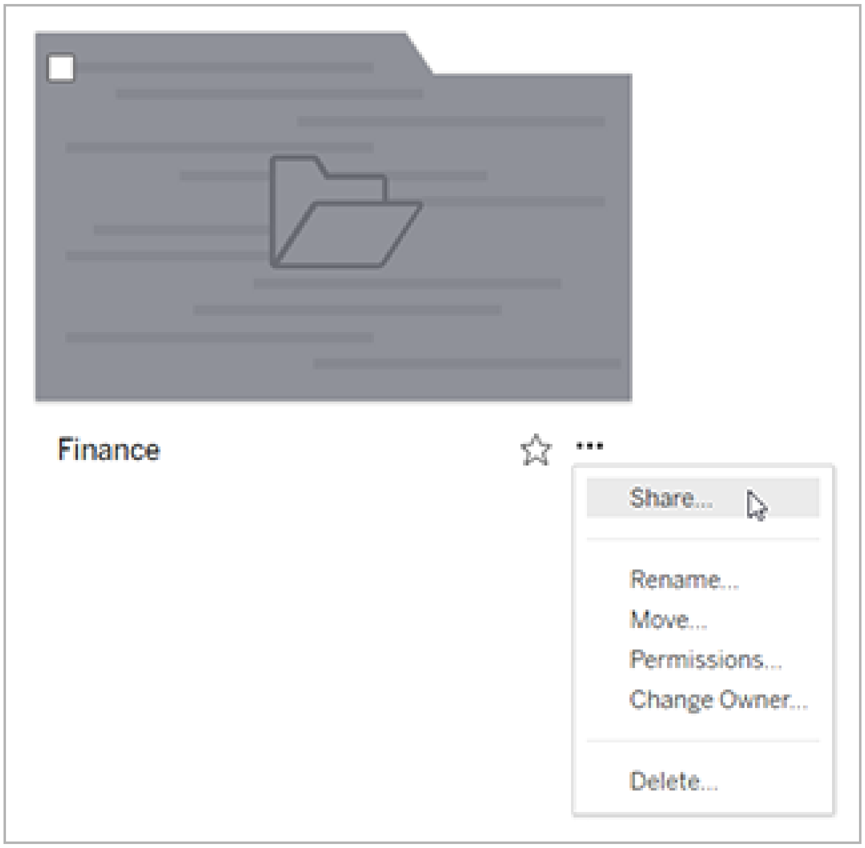
Share projects, workbooks, views, data sources, and flows either directly with other users or by copying a link to the content. For views, you can also copy code to embed the view in a web page.

To access the content you share with them, users must have permission to view that content in Tableau Server or Tableau Online. If users don’t have permission, they can request access to projects, workbooks, and views upon opening the link to the shared content.

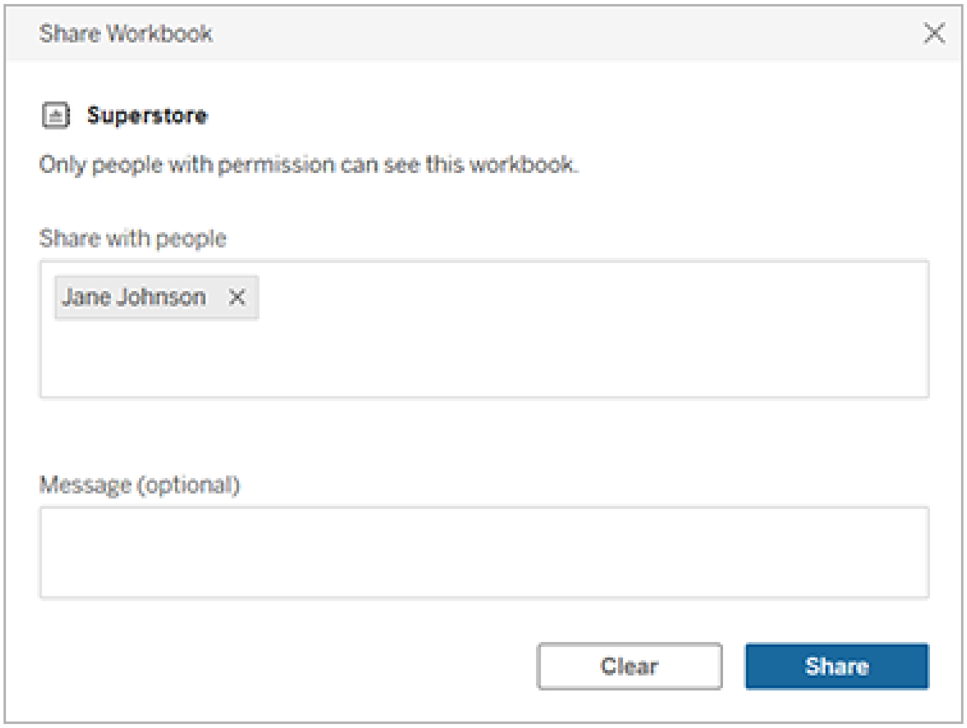
Share directly with other users

When you share content directly with other users on your site, they receive an email with a link to that content. The content is also added to their Shared with Me page, so they can easily find it later.

1. Open the Actions (**…**) menu for the content you want to share.
2. Select **Share** from the menu.



3. Under **Share with people**, enter at least one user name. When you type, names that match your text appear.



4. Under **Message**, enter an optional note to your recipients.

5. Click the **Share** button.

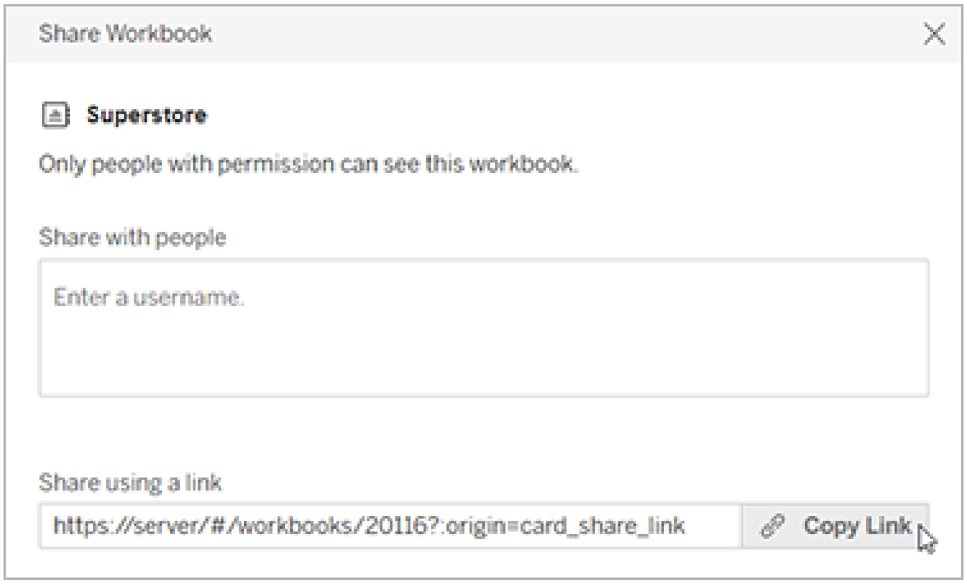
Copy a link to share

Copy the link generated for a piece of content to share without sending an email to specific users.

1. Open the Actions (**...**) menu for the content you want to share.

2. Select **Share** from the menu.

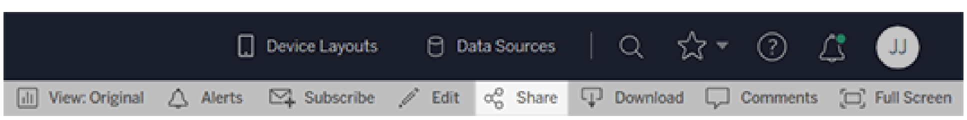
3. Click the **Copy Link** button, then paste the link into an application to share it with others.



Tip: If you don't see the **Sharing using a link** option, clear any names from under **Share with people**.

Share and embed views

In addition to the sharing options listed above, views have unique options available when you click the **Share** button on the toolbar for a view. You can share a view that you have modified, and you can get the embed code for a view.



Share modified or custom views

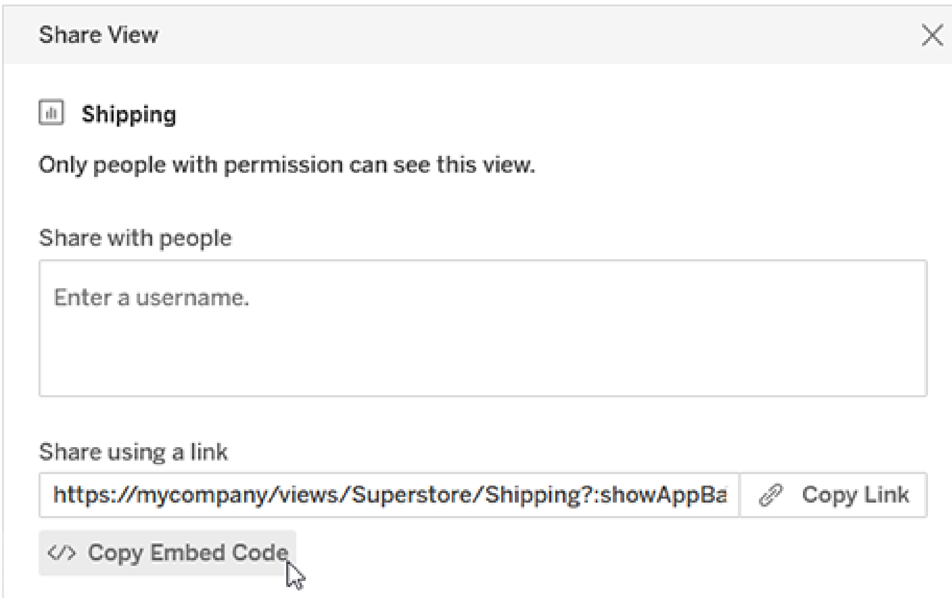
If you modify a view, by selecting marks or filters, Tableau captures these modifications by generating a unique link when you click the Share button. This unique link replaces the link to the original view for all sharing and embedding options. That way, the people you share with will see exactly what you were seeing when you shared.

The link generated for a modified view expires two years after it was last accessed, or two years after it was created if never accessed.

To share a custom view that you have saved, simply open the custom view and use any sharing option.

Embed views

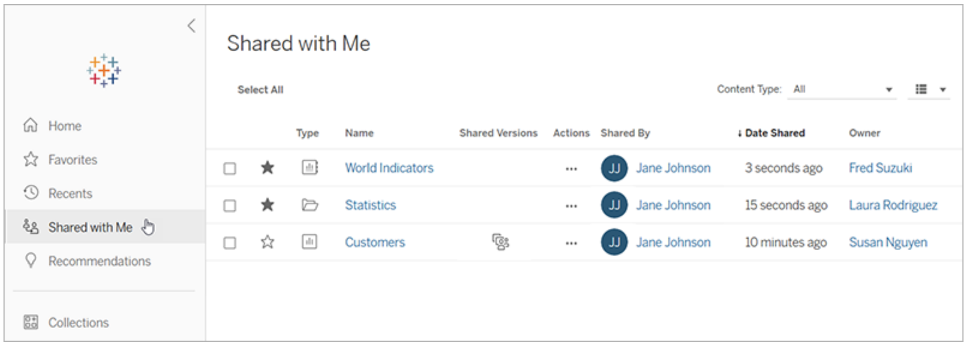
1. Click **Share** in the toolbar for the view you want to embed.
2. Click **Copy Embed Code**, then paste the embed code into the source code for the page where you want to embed the view.



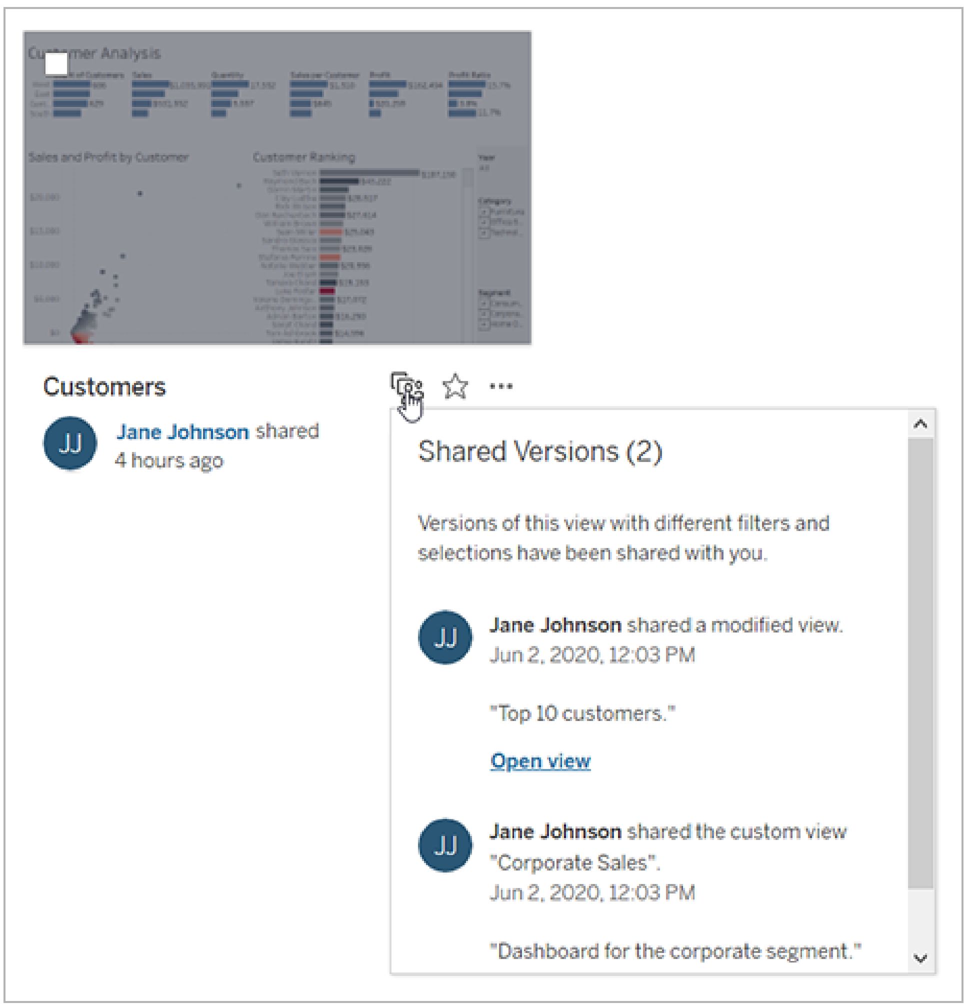
For more information, see **Embed Views into Webpages** on page 3263.

Find content shared with you

The Shared with Me page, accessed from the navigation pane, stores items that have been shared with you. The most recently shared items also appear in the Shared with Me channel on the Home page. Each item shows who it was shared by, the date is was shared, and any message added by the sharer.



If an item has been shared with you multiple times, the information shown is for the most recent share. Views include additional information if multiple versions of the view have been shared with you. Custom views, modified views, and the original view all appear with unique details and links under **Shared Versions**.



Create a Subscription to a View or Workbook

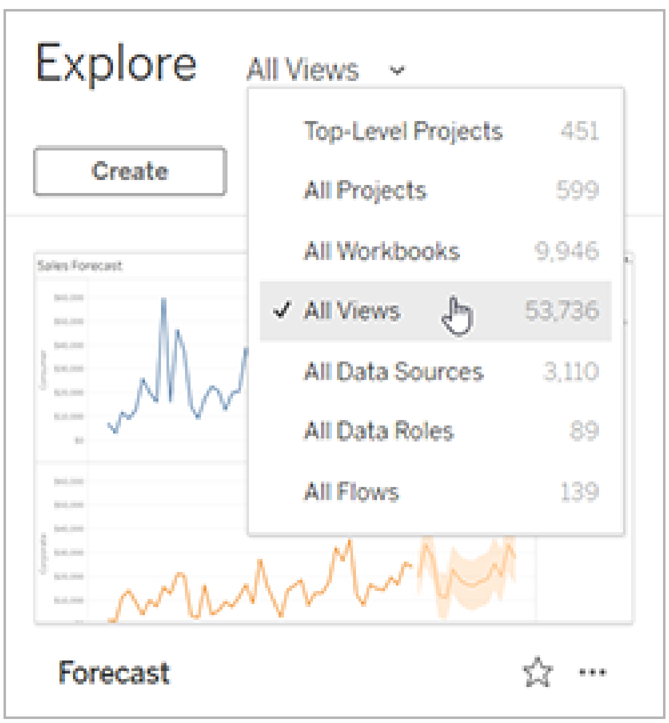
Subscriptions email you an image or PDF snapshot of a view or workbook at regular intervals—without requiring you to sign in to Tableau Server or Tableau Online.

> **Note**: In Tableau Server, administrators determine whether subscriptions are enabled for a site.

Set up a subscription for yourself or others

When you open a view in Tableau Server or Tableau Online, if you see a subscription icon ( ) in the toolbar, you can subscribe to that view or to the entire workbook. You can subscribe other users who have permission to view the content if you own a workbook, if you are a project leader with an appropriate site role, or if you are an administrator.

1. From the Explore section of your site, select **All Workbooks** or **All Views**, or open the project that contains the view you want to subscribe to.



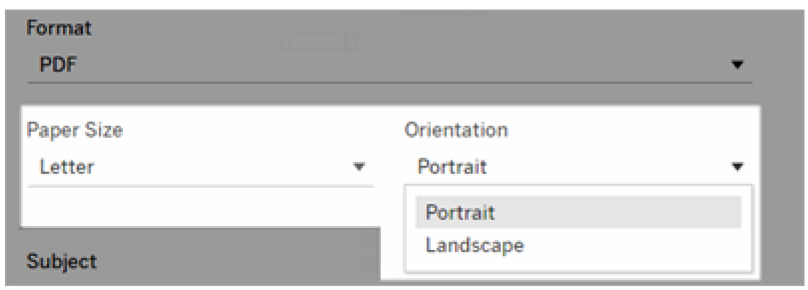
2. Open a view either directly, or after opening the containing workbook.

3. In the toolbar above the view, click **Subscribe**.

4. Add the Tableau users or groups you want to receive the subscription. To receive a subscription, users must have the **View** and **Download Image/PDF** permissions. If they use Tableau Server, their accounts must also have email addresses.

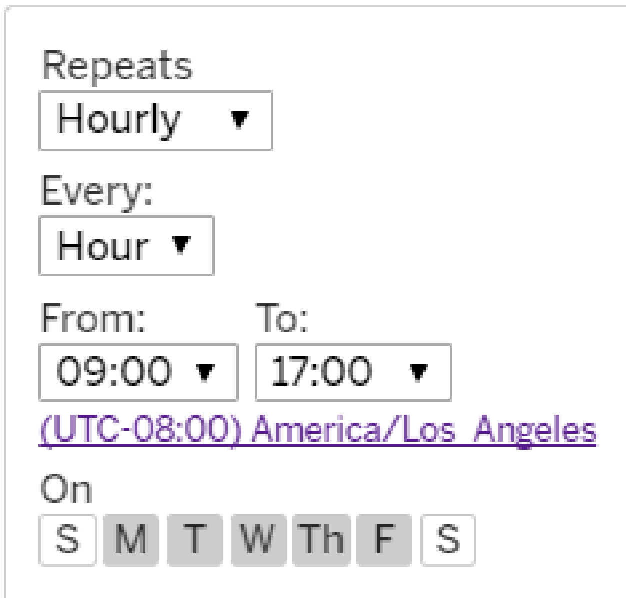
   Note: When you subscribe a group, each user is added individually at the time the subscription is created. If more users are added to the group later, you must re-subscribe the group for those new users to receive the subscription. Likewise, users later removed from the group will not have their subscriptions removed automatically unless their permissions to the subscribed view are removed.

5. Choose whether subscription emails include the current view or the entire workbook.
6. Choose the format for your snapshot: as a PNG image, a PDF attachment, or both.
   - If PDFs, choose the paper size and orientation you'd like to receive.



7. When the workbook uses one data extract on a published connection, you can pick a frequency:
   - **When Data Refreshes**: sends only when data in the view or workbook is refreshed by running refresh schedules.
   - **On Selected Schedule**: Pick a schedule for the subscription.
8. If frequency is not set to When Data Refreshes, pick a schedule:
   - For Tableau Server, choose from subscription schedules established by your administrator.
   - For Tableau Online and Tableau Server with custom schedules enabled, click the drop-down arrow to the right of the current settings.

Then specify a custom schedule that sends subscription emails whenever you wish. (The precise delivery time may vary if server load is high.)



To change the time zone, click the Time Zone link it to go to your account settings page.

9. To clarify subscription emails, customize the subject line, and add a message.

10. If the view contains data only when high-priority information exists, select **Don't send if view is empty**.

11. If you own the workbook, select **Subscribe me**.

12. Click **Subscribe**.

When you receive a subscription email, you can select the image (or the link in the message body for PDF subscriptions) to be taken to the view or workbook in Tableau Online or Tableau Server.

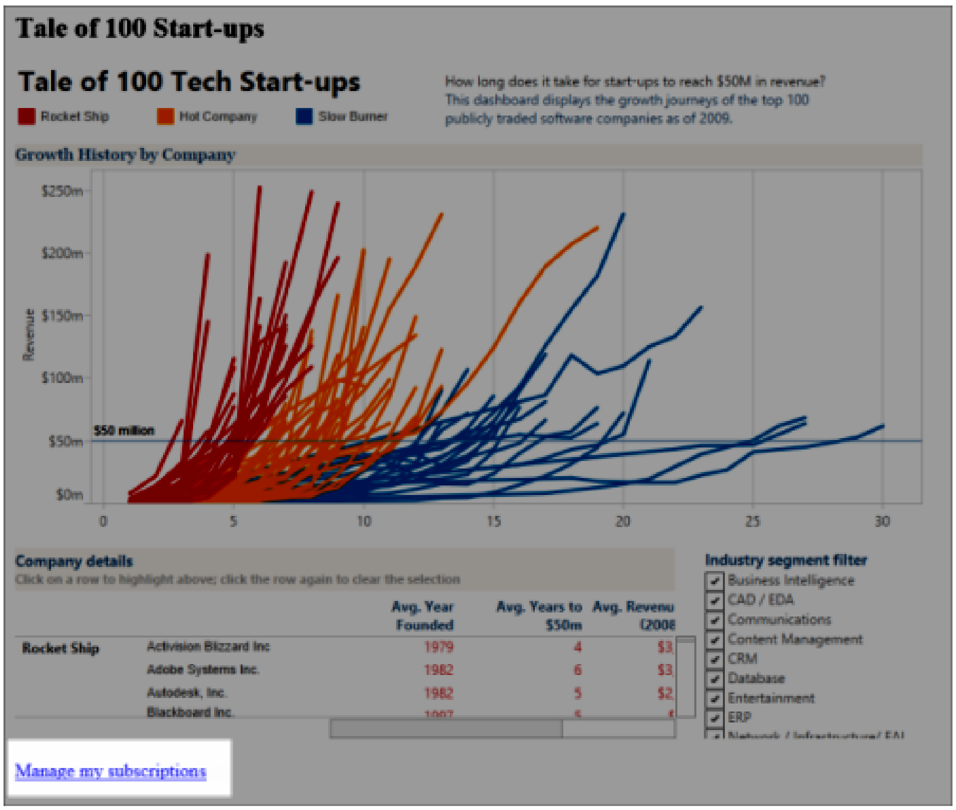
Update subscription settings

Update or unsubscribe from a subscription

You can unsubscribe from an existing subscription, or make changes to a subscription's format, schedule, subject, or empty view mode.

1. Access your Tableau Server or Tableau Online account settings by doing one of the following:

   - Click **Manage my subscriptions** at the bottom of a subscription email.



   - Sign in to Tableau Server or Tableau Online. At the top of the page, select your user icon, and then select **My Content**.

2. Click **Subscriptions**.

3. Select the check box next to the view you want to unsubscribe from, click **Actions**, and then click **Unsubscribe**, or select the subscription option you'd like to change.

Resume or delete suspended subscriptions

Sometimes, subscriptions fail because of an issue with the workbook or a problem loading the view. If a subscription fails more than five times, you'll receive a notification email that your subscription has been suspended. There are a few ways to resume a suspended subscription if you're a subscription owner or administrator:

- From the My Content area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. Select **...** > **Resume Subscription** to resume.
- From the Subscriptions tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. Select **...** > **Resume Subscription** to resume.

You'll receive an email notification when the subscription is working again.

See also

Change Subscription Settings in the Tableau Desktop and Web Authoring Help.

Project-level administration in the Tableau Online Help, to learn which site roles allow full Project Leader capabilities.

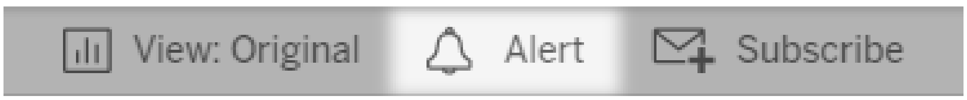
Send Data-Driven Alerts from Tableau Online or Tableau Server

When data reaches important thresholds for your business, data-driven alerts automatically send email notifications to key people you specify. You can set data-driven alerts on dashboards and views, but not story points.

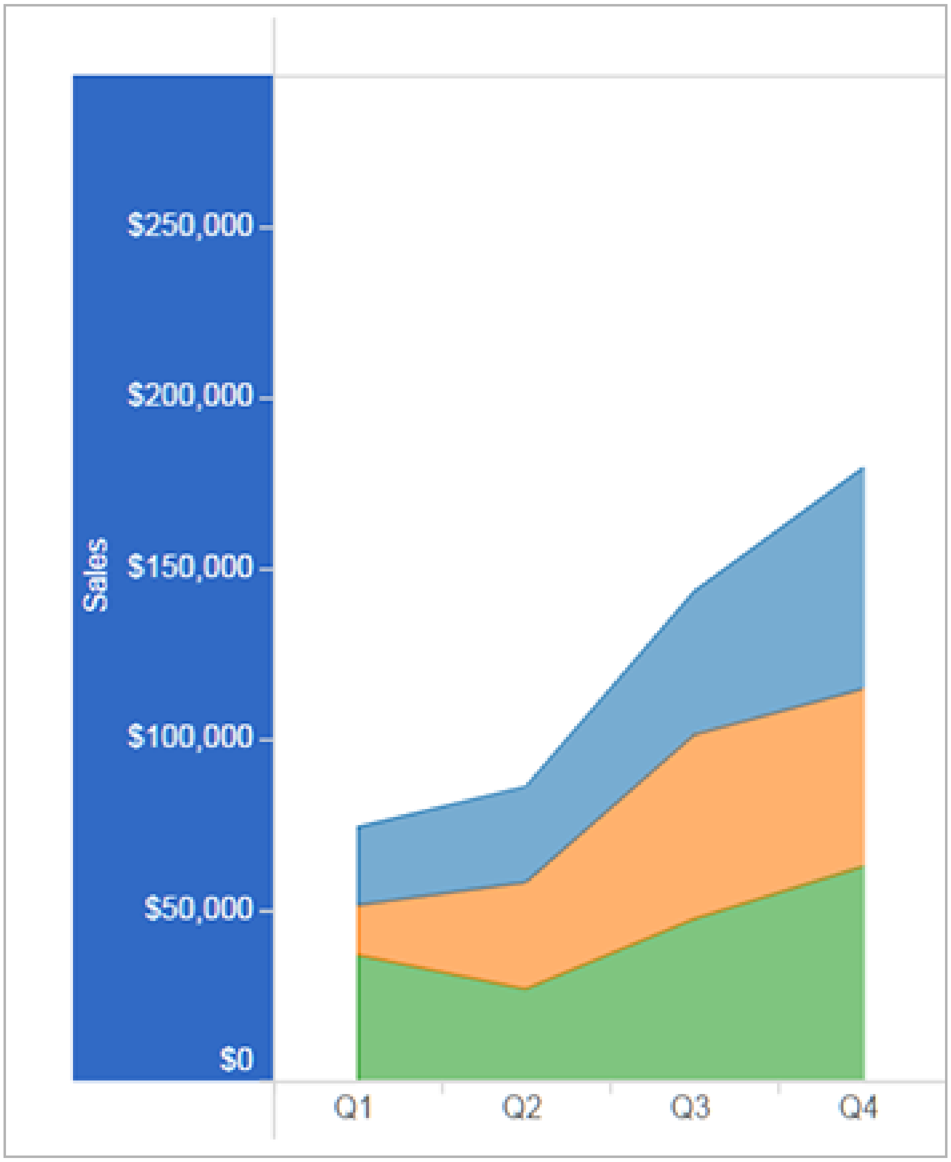
For time-based charts, use relative date filters so people automatically receive alerts as new data appears. If you don't own the content, ask the author to make this change.

Create a data-driven alert

When you select **Alerts** in the toolbar of your view or dashboard, you can create a data-driven alert or add yourself to an alert that someone else set up.

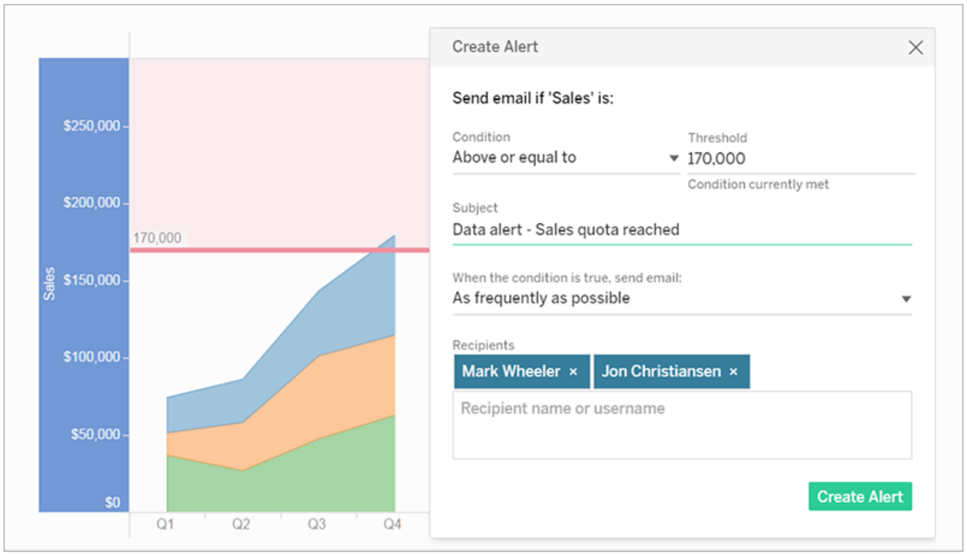


1. In the toolbar on a dashboard or view, select **Alert** to display the Alerts side panel. If there are existing alerts created on the view, those alerts will display here.
2. Select a continuous numeric axis of any chart other than a Gantt chart or map. (Numeric bins and discrete numeric axes aren't supported.)



3. Select **Create** from the Alerts side panel.
4. In the Create Alert pop-up window, set the data condition and threshold value that triggers alert emails.

   In the view, a red line shows where the threshold falls relative to your current data.



5. Specify the email subject line, schedule, and recipients. Select **Make visible to others** if you'd like others users to see your alert and add themselves to it For more information, see **Add yourself to an existing data-driven alert** below.

   For live data sources, alerts are checked every 60 minutes for Tableau Online, or on a frequency set by administrators for Tableau Server. Extracts are checked each time they're refreshed.

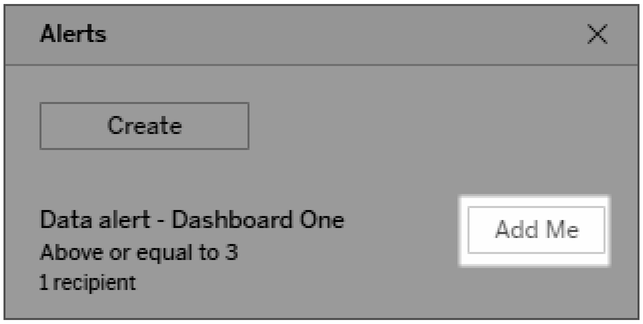
   > **Note:** Alerts created prior to 2019.2 are not visible to others by default, but can be edited to be visible.

6. Click **Create Alert**.

   > **Note:** Alerts are checked every 60 minutes for Tableau Online, or on a frequency set by administrators for Tableau Server. Alerts with extract connections on Tableau Server are also checked each time they're refreshed.

Add yourself to an existing data-driven alert

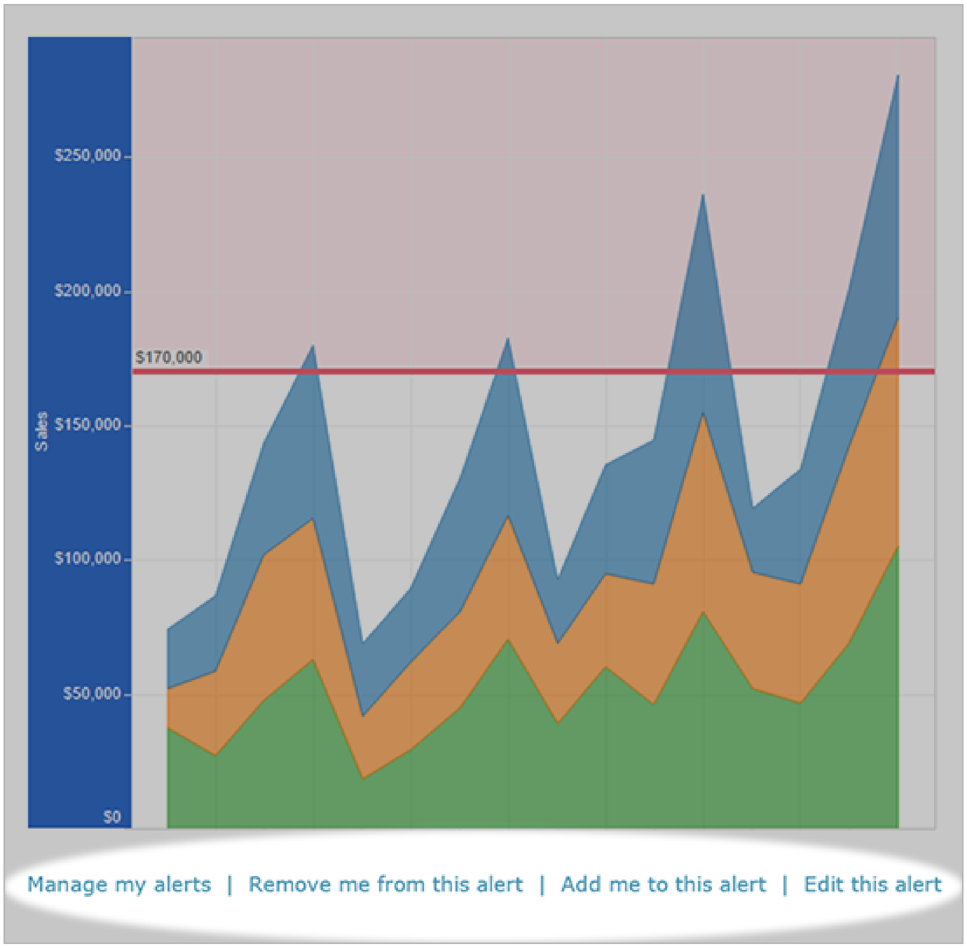
Anyone with access to a view can see alerts created by others. Select the alert to see its details, including its threshold, schedule, and notification frequency. Add yourself to the alert by selecting **Add Me**.



Manage your data-driven alerts

You can manage alerts from the My Content area of Tableau web pages, but doing so directly from alert emails can be quicker. Click the links below the image to manage all your alerts, or add or remove yourself from the emailed alert. If you're the alert owner, click **Edit this alert** to change alert settings such as threshold, schedule, and recipients.

Tip: To add recipients to an alert you don't own, you can forward an alert email and ask people to click **Add me to this alert**, or ask them to select **Add Me** from the Alert side panel.



If you are an owner of an alert, you also can manage that alert through the **Actions** menu from the Alerts side panel of the dashboard or view.

Fix failing data-driven alerts

Alerts can sometimes fail, usually for these reasons:

- Temporary connectivity issues. In this case, the alert will fix itself.
- A data source has been removed
- Credentials to data have expired
- The workbook or sheet that the alert was created on has been removed.

If an alert fails, you'll receive a notification email that tells you which alert failed and when, with a link to your site to fix the problem.

There are a few ways to fix a failing alert if you're the alert owner:

- Open the workbook in Tableau Online or Tableau Server. A warning will appear if credentials to your data have expired or if the data for your workbook has been removed.
- Click **Edit this alert** to change alert settings such as threshold, schedule, and recipients. A warning will appear to tell you if the workbook or sheet that your alert was created on has been removed.

You'll receive an email notification when the alert is working again.

Resume or delete suspended data-driven alerts

If an alert fails enough times, you'll receive a notification email that your alert has been suspended. There are a few ways to resume a suspended alert if you're an alert owner or administrator:

- From the My Content area of Tableau web pages, an icon appears in the Last checked column to indicate that the alert is suspended. Select **...** > **Resume Alert** to resume the alert.
- Click **Resume Alert** in the notification email to resume the alert. A notification will either allow you to resume the alert, or indicate that the view has changed and the alert should be deleted.

You'll receive an email notification when the alert is working again.

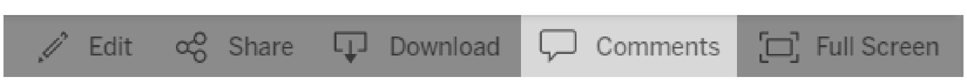
Comment on Views

Comments let you share a conversation about data discoveries with other Tableau Server or Tableau Online users.

Tip: To keep comments when you republish a workbook, be sure to use the same workbook and view names.

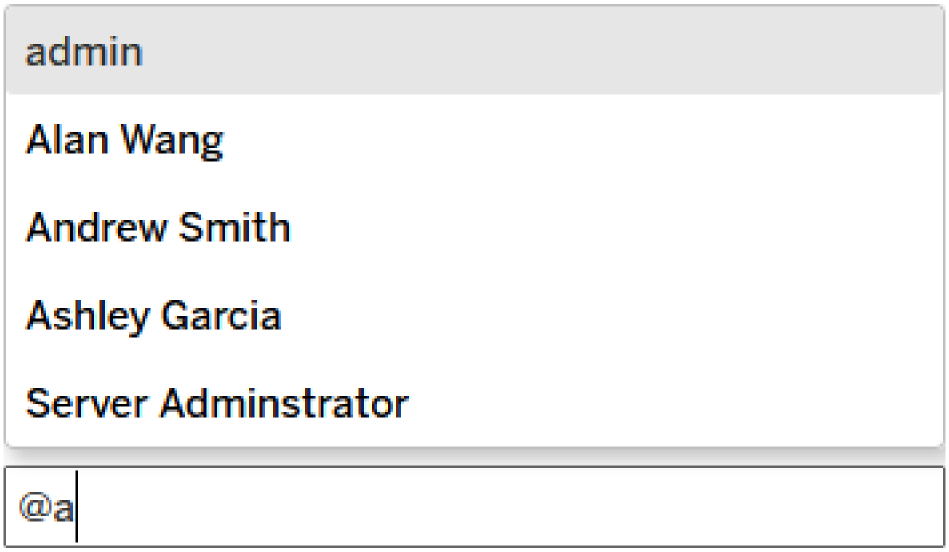
Add comments

1. In the toolbar above a view, click **Comments**.



2. In the Comments pane at right, enter your remarks.

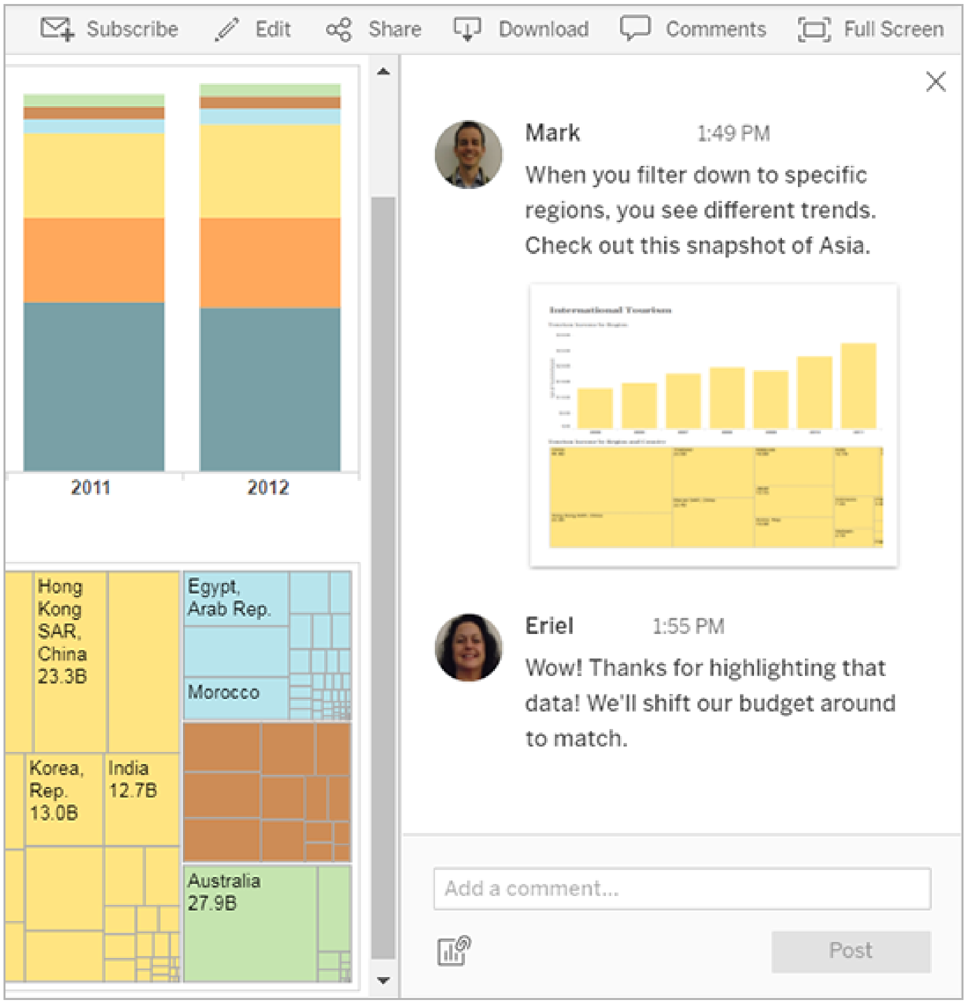
3. If you'd like to notify others about the comment, include an @mention. The @mention search shows a maximum of ten users, in alphabetical order. If you have trouble locating someone, try searching for a username instead of a display name, e.g., jjohnson instead of Jane Johnson.



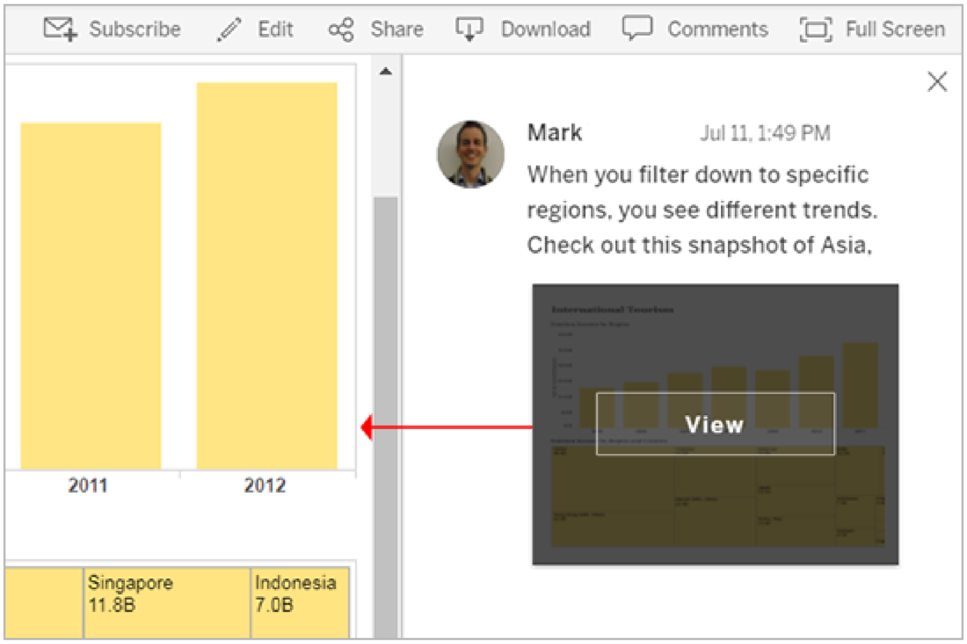
4. If you've filtered the view, click the snapshot icon  to share an interactive image that highlights the data you're describing.

> **Note:** The snapshot icon doesn't appear if a view is user-filtered, protecting secure data.

5. Click **Post**.

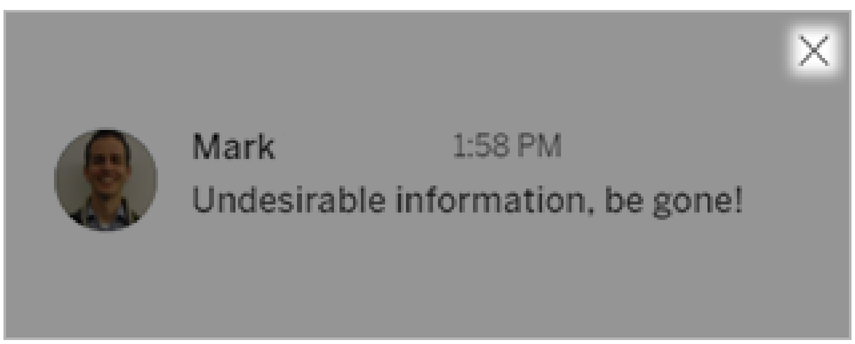


6. To filter a view and see a data snapshot in detail, hover over the thumbnail image, and click **View**.



Delete comments

If a comment is unnecessary or inaccurate, you can quickly delete it. Just click the **X** in the upper-right corner.



You can delete a comment if you created it, are the content owner, are a project leader with an appropriate site role, or are an administrator. To learn which site roles are required for full project leader access, see Project-level administration in the Tableau Online Help.

Embed Views and Dashboards

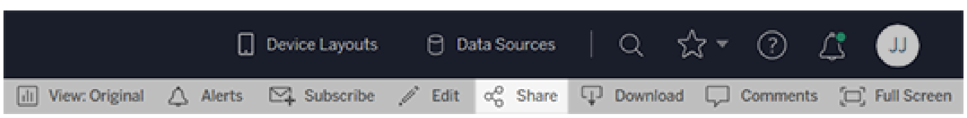
Embed Views into Webpages

You can embed interactive Tableau views and dashboards into web pages, blogs, wiki pages, web applications, and intranet portals. Embedded views update as the underlying data changes, or as their workbooks are updated on Tableau Server or Tableau Online. Embedded views follow the same licensing and permission restrictions used on Tableau Server and Tableau Online. That is, to see a Tableau view that's embedded in a web page, the person accessing the view must also have an account on Tableau Server or Tableau Online.

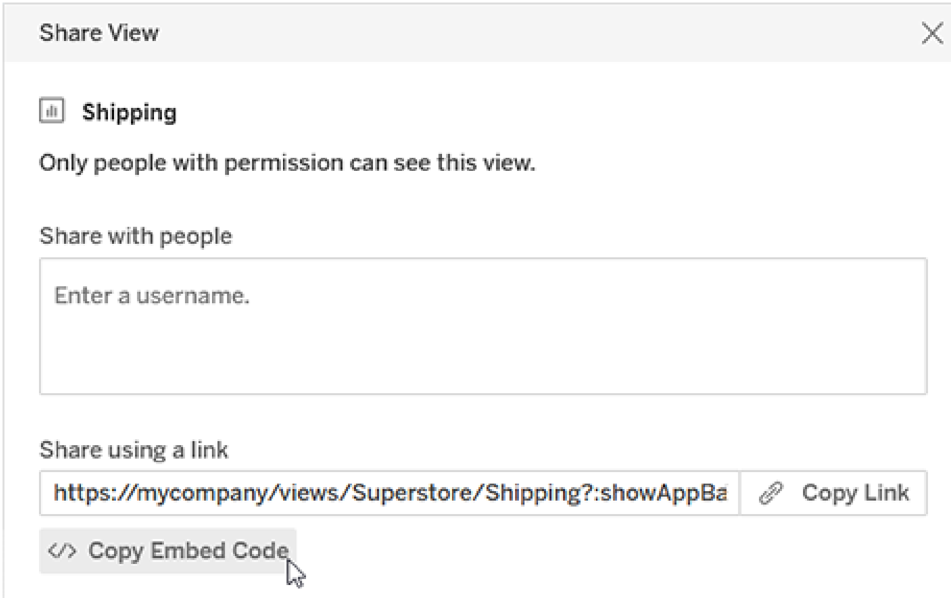
Alternatively, if your organization uses a core-based license on Tableau Server, a Guest account is available. This allows people in your organization to view and interact with Tableau views embedded in web pages without having to sign in to the server. Contact your server or site administrator to find out if the Guest user is enabled for the site you publish to.

You can do the following to embed views and adjust their default appearance:

- **Get the embed code provided with a view:**
  1. Click the **Share** button at the top of a view. (The **Share** button doesn't appear in embedded views if you change the `showShareOptions` parameter to `false` in the code.)

2. Click **Copy Embed Code**, then paste the code into your webpage.



- **Customize the embed code:** You can customize the embed code using parameters that control the toolbar, tabs, and more. For more information, see **Parameters for Embed Code** on page 3267.
- **Use the Tableau JavaScript API:** Web developers can use Tableau JavaScript objects in web applications. To get access to the API, documentation, code examples, and the Tableau developer community, see the Tableau Developer Portal.

Note: For users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

Writing Embed Code

If you're writing your own embed code, you can take one of two approaches:

- **Use Tableau JavaScript:** This is the preferred approach. Use the embed code that Tableau generates as the starting point for your own code, adding or editing object parameters that control the toolbar, tabs, and more. The default embed code, which relies on a Tableau JavaScript file, is also the only way to control the load order of multiple embedded views.
- **Specify the View URL:** Embed a view using an iframe or image tag, where the source

is the URL you get when you click **Share** on a view and then click **Copy Link**. You may want to do this if you can't use JavaScript on your website. There may also be situations when all you can specify is a URL.

When you embed a view, you should define a width and height that the view will be displayed in. Otherwise, the client browser will arbitrarily pick a width and height.

> **Note:** When you need to specify the server name for Tableau Online in embed code (for example, to point to the location of the JavaScript API), use the URL **https://online.tableau.com**.

Use Tableau JavaScript

The following code shows an example of embed code that is generated when you click **Share** on a published view. Special characters in the `host_url` parameter are URL encoded, and those in the `site_root` and `name` parameters are notated as HTML numeric character references.

Tableau Server example:

```
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width:800; height:600;'>
<object class='tableauViz' width='800' height='600' style-
='display:none;'>
   <param name='host_url' value='http%3A%2F%2Fmyserver%2F' />
   <param name='site_root' value=&#47;t&#47;Sales' />
   <param name='name' value='MyCoSales&#47;SalesScoreCard&#47;'
/>
   <param name='tabs' value='yes' />
   <param name='toolbar' value='yes' /></object></div>
```

Tableau Online example:

```
<script type='text/javascript' src-
='https://online.tableau.com/javascripts/api/viz_v1.js'></script>
<div class='tableauPlaceholder' style='width:800; height:600;'>
<object class='tableauViz' width='800' height='600'
```

```
style='display:none;'>
   <param name='host_url' value-
='https%3A%2F%2Fonline.tableau.com%2F' />
   <param name='site_root' value='&#47;t&#47;Sales' />
   <param name='name' value='MyCoSales&#47;SalesScoreCard&#47;'
/>
   <param name='tabs' value='yes' />
   <param name='toolbar' value='yes' />
</object>
</div>
```

The source for the `<script>` tag is the URL for the Tableau Server and Tableau Online JavaScript file, **viz_v1.js**. The JavaScript file handles assembling the full URL of the view that's displayed for your users. The `name` and `site_root` object parameters are the only required parameters; all other parameters are optional.

Specify the View URL

Here's an example of embedding the same view using an iframe, where the source is the URL you get when you click **Share** on a view and then click **Copy Link**.

Tableau Server example:

```
<iframe src-
c="http://myserver/t/Sales/views/MyCoSales/SalesScoreCard
?:embed=yes&:tabs=yes&:toolbar=yes" width="800"
height="600"></iframe>
```

Tableau Online example:

```
<iframe src-
="h-
ttps://online.tableau.com/t/Sales/views/MyCoSales/SalesScoreCard
?:embed=yes&:tabs=yes&:toolbar=yes" width="800"
height="600"></iframe>
```

The `embed` URL parameter is required, and you can optionally include parameters that control the toolbar and revert options, among others. You can also add filters to the URL that control the specific data that shows when a view is loaded.

Tip: When using this method, add the HTML attribute allowfullscreen="true" to the iframe element to enable the Full Screen button on the view.

Parameters for Embed Code

You can configure an embedded view using either of these methods:

- **Object parameters for JavaScript tags** below
- **URL parameters for iframe tags** on page 3283

Object parameters for JavaScript tags

Object Parameter	Values	Description	Examples
dataDetails	no	Hides the Data Details button in the toolbar, which lets users see details about the data used in that view.	<param name='dataDetails' value='no' />
alerts	no	Hides the	<param name='alerts' value='no'/>

Object Parameter	Values	Description	Examples
		Alerts button in the toolbar, which lets users create data-driven alerts.	
`customViews`	`no`	Hides the View button in the toolbar, which lets users save custom views.	`<param name='customViews' value='no'/>`
`device`	`desktop; ta-`	If a dashboard	`<param name='device' value='phone'/>`

Object Parameter	Values	Description	Examples
	ble-t; phone	has layouts for mobile devices, displays a specific layout, regardless of screen size. If this parameter isn't set, Tableau Server or Tableau Online detects screen size and	

Object Parameter	Values	Description	Examples
		onding layout. See **Embed Dashboards** on page - 3306 for examples.	
`filter`	string	Filters the data displayed when the view opens. You can also filter using URL parameters.	`<param name='filter' value='Team=Blue'/>`

Object Parameter	Values	Description	Examples
`host_url`	string	Required. The server name as it appears in the URL.	`<param name='host_url' value='http://myserver.exampleco.com/'>` `<param name="host_url" value="http://localhost/">`
`linktarget`	string	The target window name for external hyperlinks.	`<param name="linktarget" value="_blank"/>`
`load-order`	number	When multiple views are embedded, determines the order	`<param name="load-order" value="2"/>`

Object Parameter	Values	Description	Examples
		in which they load on the page. Negative numbers are allowed.	
mobile	yes	Displays touch-optimized quick filters for mobile devices.	`<param name='mobile' value='yes'/>`
name	string	Required object parameter, with this	`<param name='name' value='ExampleCoSales/Sales'/>` `<param name='name' value='ExampleCoSales/Sales/jsmith@example.com/EastRegionSales'/>`

Object Parameter	Values	Description	Examples
		structure: [workbook name]/[sheet name]/[optional custom view name] Format optional custom view names like this: username@domain/[custo	

Object Parameter	Values	Description	Examples
		m view name] If you refer to the Tableau Server or Tableau Online URL to confirm the value of `name`, exclude the session ID (`:iid =<n>`) at the end of the URL.	

Object Parameter	Values	Description	Examples
:original_view	yes	If the name parameter refers to a workbook or sheet URL (and does not explicitly refer to a custom view) including this parameter displays the view as the original	`<param name='filter' value=':original_view=yes'/>`

Object Parameter	Values	Description	Examples
		view.	
`path` (Tableau Server only. Not applicable to Tableau Online.)	string	For trusted authentication only, cannot be used with the `ticket` parameter. Overrides value of the `name` parameter and is used as the URL. For more inform-	`<param name='path' value='trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4'/>` `http://tableauserver/trusted/Etdpsm_Ew6rJY-9kRrALjauU/views/workbookQ4/SalesQ4?:embed=yes&:tabs=yes`

Object Parameter	Values	Description	Examples
		ation, see Display the View with the Ticket in the Tableau Server Administrator help.	
`showShareOptions`	`true; false`	Controls whether the Share options are displayed in an embedded view.	`<param name='showShareOptions' value='true' />`

Object Parameter	Values	Description	Examples
`site_root`	string	Required. The site name. The Default site value is null (`value=''`). If your server is multi-site and you want to use trusted authentication, see Display the	`<param name='site_root' value='/#/Sales'/>` `<param name='site_root' value=''/>`

Object Parameter	Values	Description	Examples
		View with the Ticket in the Tableau Server help.	
`subscriptions`	`no`	Hides the Subscriptions button in the toolbar, which lets users set up email subscriptions to receive snapshots of views	`<param name='subscriptions' value='no'/>`

Object Parameter	Values	Description	Examples
		at regular intervals.	
tabs	yes; no	Displays or hides tabs.	`<param name='tabs' value='yes'/>`
ticket (Tableau Server only. Not applicable to Tableau Online.)	string	For trusted authentication only, cannot be used with the path object parameter. Must be used with name	`<param name='ticket' value='9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5'/>` `http://tableauserver/trusted/9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5/views/workbookQ4/SalesQ4?:embed=yes&:tabs=yes`

Object Parameter	Values	Description	Examples
		object to construct the trusted ticket redemption URL. For more information, see Display the View with the Ticket in the Tableau Server help.	
`toolbar`	`yes`; `no`; `to-`	The toolbar is	`<param name='toolbar' value='top'/>`

Object Parameter	Values	Description	Examples
	p	displayed on the bottom by default. The toolbar is placed above the view when you set this parameter to `top` and excluded from the embedded view when you set it to `no`.	

Object Parameter	Values	Description	Examples
`tooltip`	`yes; no`	Tooltips are displayed by default. If you set this parameter to `no`, however, tooltips are excluded from the embedded view.	`<param name='tooltip' value='no'/>`

URL parameters for iframe tags

> Note: Before you add URL parameters, remove `:iid=[#]` at the end of the URL. This is a temporary view counter for your current browser session.

URL Parameter	Values	Description	Examples
`:alerts`	`no`	Hides the Alerts button in an embedded view.	`http://tabserver/views/DateTime/DateCalcs?:embed=yes&:alerts=no`
`:customViews`	`no`	**Remember my changes** option.	`http://tabserver/views/DateTime/DateCalcs?:embed=yes&:customViews=no`
`:device`	`desktop; tablet; phone`	If a dashboard has layouts for mobile devices, displays a specific layout, regardless of screen size. If this parameter isn't set, Tableau Server or Tableau Online detects screen size and loads a corresponding layout. See **Embed**	`http://tabserver/views/sales/sales_dashboard?:device=tablet`

URL Parameter	Values	Description	Examples
		Dashboards on page 3306 for examples.	
:display_count (Tableau Public only)	yes	Shows the total number of pageviews when you click the Tableau logo at lower left.	https://public.tableau.com/en-us/s/gallery/what-emoji-say-about-music?:display_count=yes
:embed	yes	Required URL parameter. Hides the top navigation area, better blending the view into your web page.	http://tabserver/views/Date-Time/DateCalcs?:embed=yes
:format	pdf; png	Displays a view as a PDF or .png file.	http://tabserver/views/Sales/Q2?:format=pdf
:highdpi	false	Renders a view using	http://tableau-

URL Parameter	Values	Description	Examples
		standard DPI (dots per inch) for high-resolution displays and devices.	`server/views/Sales/Q2?:highdpi=false`
`:linktarget`	string	The target window name for external hyperlinks.	`http://tabserver/views/DateTime/DateCalcs?:embed=yes&:linktarget=_blank`
`:openid_hd`	string	Used when authenticating with OpenID Connect. Specifies the hosted domain to which a user belongs. For example, you might use this option if you do not want to render the domain	`http://tabserver/views/Sales/Q2?:openid_hd=example.com`

URL Parameter	Values	Description	Examples
		selection pane. Used in conjunction with `:openid_login_hint` to specify the full user identifier.	
`:openid_login_hint`	string	Used when authenticating with OpenID Connect. Specifies the full user identifier (email address). Used in conjunction with `:openid_hd` to specify the hosted domain to which the user belongs.	`http://tabserver/views/Sales/Q2?:openid_login_hint=tara@example.com`

URL Parameter	Values	Description	Examples
:record_performance (Tableau Server only. Not applicable to Tableau Online.)	yes	Starts a performance recording for a view. Add this at the end of the URL, immediately before the session ID (:iid=<*n*>). For more information, see Create a Performance Recording in Tableau Server help.	http://tabserver/views/Sales2013/Regions?:recordperformance=yes
:refresh	yes	Renders the view using the latest data from Tableau Server or Tableau Online.	http://tabserver/views/DateTime/DateCalcs?:embed=yes&:refresh=yes

URL Parameter	Values	Description	Examples
		Tip: To continually refresh a view, in the `<head>` section of the web page, add `<meta http-equiv="refresh" content="#`	

URL Parameter	Values	Description	Examples
		">, where # is the number of seconds between refreshes.	
:render	true; false; number	If client-side rendering is enabled (the default), setting this to false forces server-side rendering for the session. Setting this to true forces client-side rendering for the ses-	http://tabserver/views/Date-Time/DateCalcs?:render=false

URL Parameter	Values	Description	Examples
		sion. A number from 1 to 100 can be entered to set the complexity threshold above which views are rendered by the server. For more information, see About Client-Side Rendering in Tableau Server help.	
`:revert`	`all; filters; sorts; axes; shel-`	Returns the item to its original state.	`http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:revert=all`

URL Parameter	Values	Description	Examples
	yes		
:showAppBanner	false	Hides the “Open in Tableau Mobile” banner in mobile web browsers.	http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:showAppBanner=false
:showAskData	false	Hides the Ask Data button in the toolbar in an embedded view. By default the Ask Data button displays.	http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:showAskData=false
:showShareOptions	true; false	Controls whether the Share options are displayed in an embedded view.	http://tabserver/views/Date-Time/DateCalcs?:embed=yes&:showShareOptions=true
:showVizHome (Tableau Public only)	yes; no	Shows or hides the content navigation interface of	https://public.tableau.com/en-us/s/gallery/what-emoji-say-about-music?:showVizHome=no

URL Parameter	Values	Description	Examples
		Tableau Public. Set to no when embedding vizzes in other web pages.	
`:subscriptions`	`no`	Hides the Subscribe button in an embedded view.	`http://tabserver/views/DateTime/DateCalcs?:embed=yes&:subscriptions=no`
`:tabs`	`yes; no`	Displays or hides tabs.	`http://tabserver/views/DateTime/DateCalcs?:embed=yes&:tabs=no`
`:toolbar`	`yes; no; top`	The toolbar is displayed by default on the bottom when this parameter is not set. When `no` the toolbar is excluded from the embedded view. When `top`, the toolbar is placed above the	`http://tabserver/views/DateTime/DateCalcs?:embed=yes&:toolbar=no`

URL Parameter	Values	Description	Examples
		view.	
`:tooltip`	`yes; no`	Tooltips are displayed by default in a view when this parameter is not set. If set to `no`, tooltips are excluded from the embedded view.	`http://tabserver/views/workbookQ4/SalesQ4?:embedded=yes&:tooltip=no`

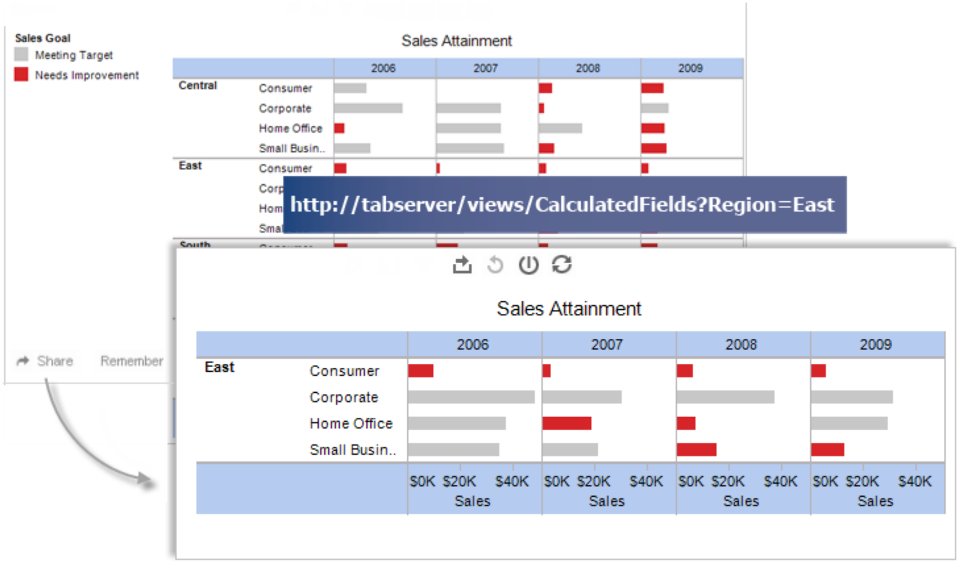
Add Filters to Embed Code

You can include filter values in embedded views so they highlight just the data you want. For example, you may want to include a hyperlink from another part of your web application to an embedded sales performance view that shows only a specific region.

> **Note:** Stories don't support embedded filters.

Filter on one field

Passing a filter on one field is a quick way to focus an embedded view on specific data.



Script tag example:

```
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='800' height='600' style-
='display:none;'>
   <param name='host_url' value='http://myserver/' />
   <param name='site_root' value='' />
   <param name='name' value='Superstore/Product' />
   <param name='filter' value='Region=East' />
</object>
```

To pass through multiple filter values, just separate each with a comma. For example:

```
<param name='filter' value='Region=East,West' />
```

Iframe tag examples:

```
<iframe src-
="h-
ttp://myserver-
/views/Superstore/Product?:embed=y&Region=East,West"
width="800" height="600"></iframe>
```

Filter on multiple fields

You can pass filters on as many fields as you want, including fields that are not in visible the original view.

Script tag example:

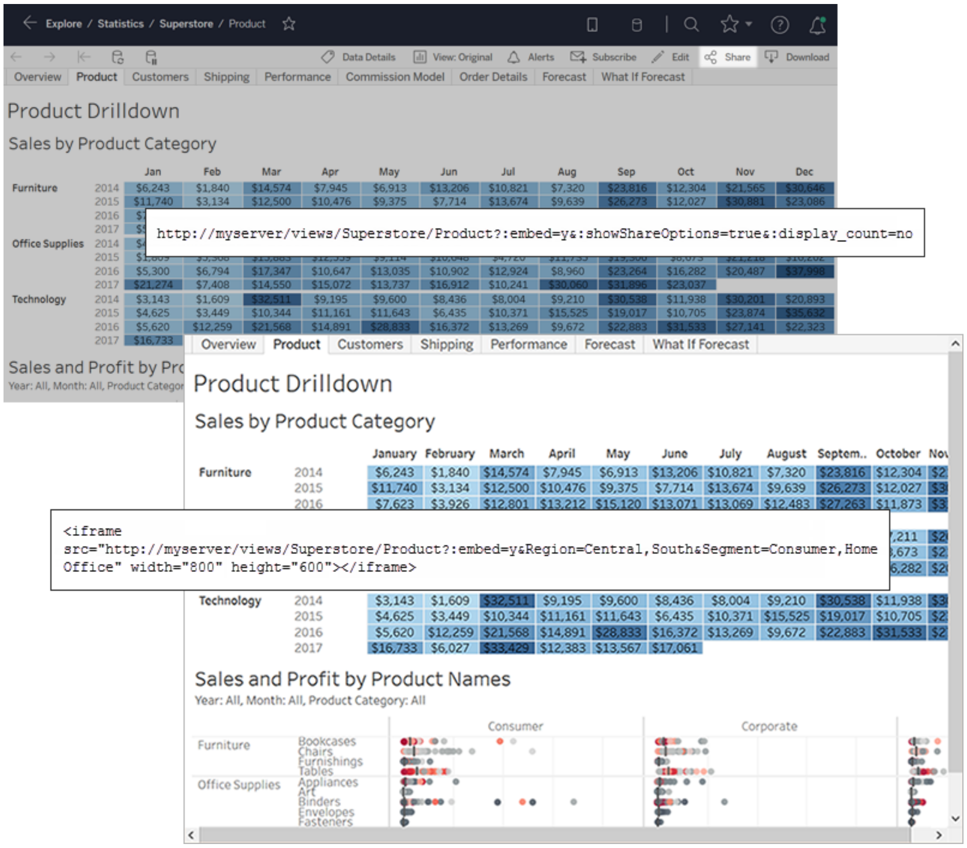
```
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='800' height='600' style-
='display:none;'>
   <param name='host_url' value='http://myserver/' />
   <param name='site_root' value='' />
   <param name='name' value='Superstore/Product' />
   <param name='filter' value='Region=Central,South&Customer Seg-
ment=Consumer,Home Office' />
</object>
```

Iframe tag example:

```
<iframe src-
="h-
ttp://myserver-
/views/Su-
per-
store/Product?:embed=y&Region=Central,South&Segment=Consumer,Home
Office"
width="800" height="600"></iframe>
```

The first image below shows an example of the URL you get when you click **Share** on a view and then click **Copy Link**.

The second image shows how you might modify the URL and add it to an iframe by deleting the `showShareOptions` and `display_count` parameters, adding filter parameters for Region and Segment, and adding width and height parameters, to create an embed link that displays only Consumer and Home Office products from the Central and South regions.



> **Note**: If a filter value contains a special character, such as a comma, replace the character with the URL encoding sequence for `\` (backslash, `%5c`) followed by the URL encoding sequence for the special character. The backslash is needed to escape the special character. For example, the URL encoding sequence for `\,` (backslash, comma) is `%5c%2c`.

Filter dates and times

If you want to filter on a Date/Time field, include the value using the default Tableau format shown below:

```
yyyy-mm-dd hh:mm:ss
```

The time part uses a 24-hour clock. Many databases store all date values as Datetime fields, so you may need to pass a time value along with your date.

Script tag example:

```
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'></script>
<object class='tableauViz' width='800' height='600' style-
='display:none;'>
   <param name='host_url' value='http://myserver/' />
   <param name='site_root' value='' />
   <param name='name' value='Sales/Sales-Performance' />
   <param name='filter' value='Date=2012-12-01' />
</object>
```

This example filters on both a date field and a datetime field:

```
<param name='filter' value='2012-12-01%2022:18:00' />
```

Iframe tag example:

```
<iframe src="http://myserver/Sales/SalesPer-
formance?:embed=yes&Date=2008-12-01%2022:18:00" width="800"
height="600"></iframe>
```

To filter multiple dates, separate each date with a comma.

Filter measures

You can filter measures by including one or more values. There is no support for greater than, less than, or ranges. The example below filters to show only $100 and $200 sales.

Script tag example:

```
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='800' height='600'  style-
='display:none;'>
   <param name='host_url' value='http://myserver/' />
   <param name='site_root' value='' />
   <param name='name' value='Sales/Sales-Performance' />
   <param name='filter' value='Profit=100, 200' />
</object>
```

Iframe tag example:

```
<iframe src="http://myserver/views/Sales/Sales-Per-
formance?:embed=yes&Profit=100,200"
width="800" height="600"></iframe>
```

How View URLs Are Structured

To understand how Tableau structures view URLs, let's take a close look at an example in the browser address bar.

The structure of the base URL

- In Tableau Server or Tableau Online, the basic URL pattern for a Tableau view is:
  http://<servername>/#/views/<workbook>/<sheet>
- In a multi-site environment, for views saved to sites other than Default, the site ID is also included in the URL:
  http://<servername>/#**/site/<sitename>**/views/<-
  workbook>/<sheet>

Adding parameters to the base URL

Appending a query string

To create a filtering URL, you append a *query string* to the base URL. In a query string, the view filters are the parameters. To construct a query string:

- Begin the query string with a question mark (?).
- Separate the parameters using ampersands (&).
- Separate multiple values per parameter using commas (**,**).

For example:
http://<servername>/#/views/<workbook>/sheet
?param1=value1,value2&param2=value

In addition, any characters in your field or sheet names that cannot appear in URLs are URL-encoded. For example, a space is converted to %20.

> **Note:** Before you add URL parameters, remove **:**iid=[#] at the end of the URL. This is a temporary view counter for your current browser session.

Reflecting the capitalization of field names

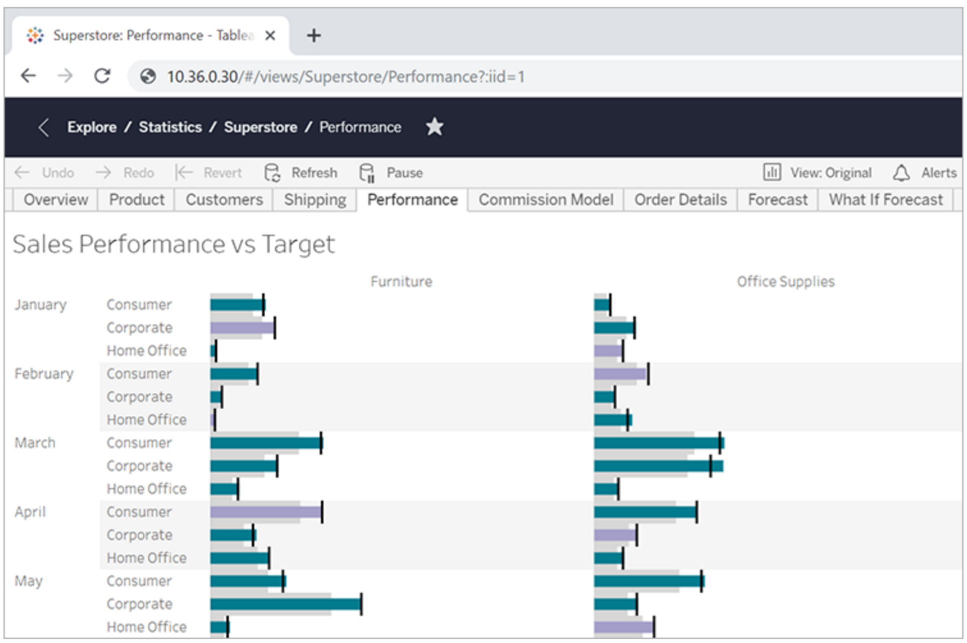
URL parameters are case-sensitive. When you append query strings, be sure to precisely reflect the capitalization of field names in your source data.

How query strings affect dashboards

Unlike visible filters in a dashboard, which sometimes affect only one sheet, URL parameters apply to all sheets. Before appending URL parameters, become familiar with the fields shown in each sheet to avoid accidentally filtering out data you'd like to retain.

Select an example view to work with

The examples in this article use the Performance view, in the Superstore sample data that comes with Tableau Server.



The example URL reflects the following attributes:

- The sheet name is **Performance**
- The workbook name is **Superstore**
- The server name is indicated using a placeholder IP address, where your on-premise server name or `online.tableau.com` would appear.

To follow along with the examples, you can look at a similar view from Superstore, or you can adjust the example URLs to use the field, sheet, and other names from your own environment. If you are using your own view, make sure you first publish it to the server.

Filter the view

For the example view used in this article, the base URL is:

`http:<servername>/#/views/Superstore/Performance`

Furniture only

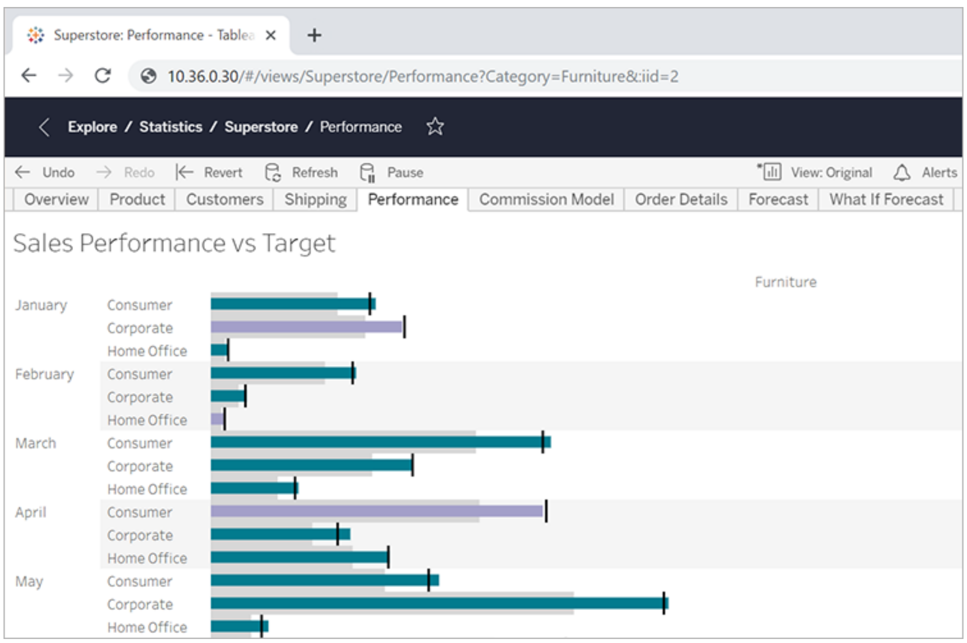
To display only sales of furniture, at the end of the URL, add the following query string:

`?Category=Furniture`

For example:

`http:<servername>/#/views/Superstore/Performance`

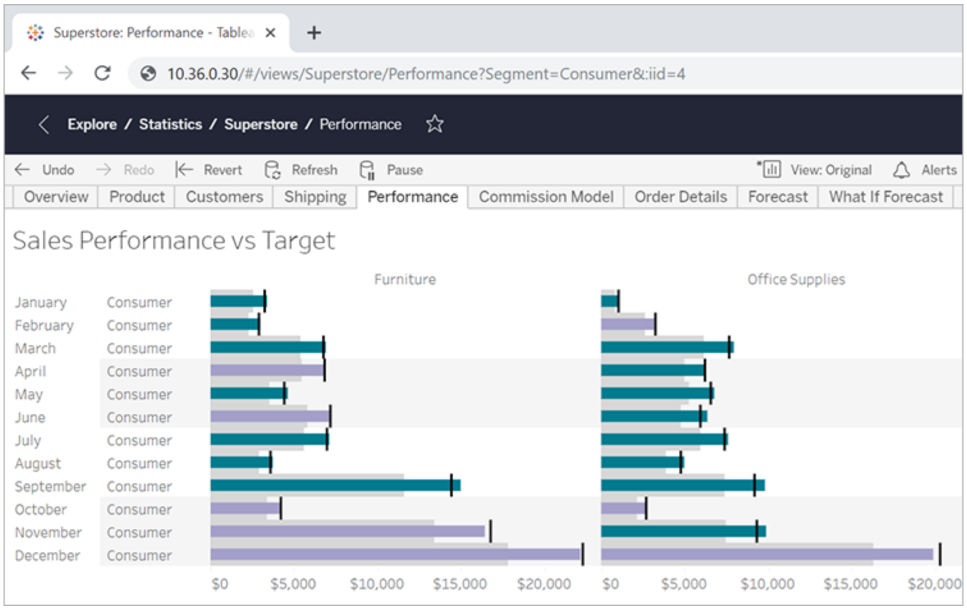
`?Category=Furniture`



Consumer only

To limit the view to Consumer sales, change the URL to:

`http:<servername>/#/views/Superstore/Performance`**`?Segment=Consumer`**



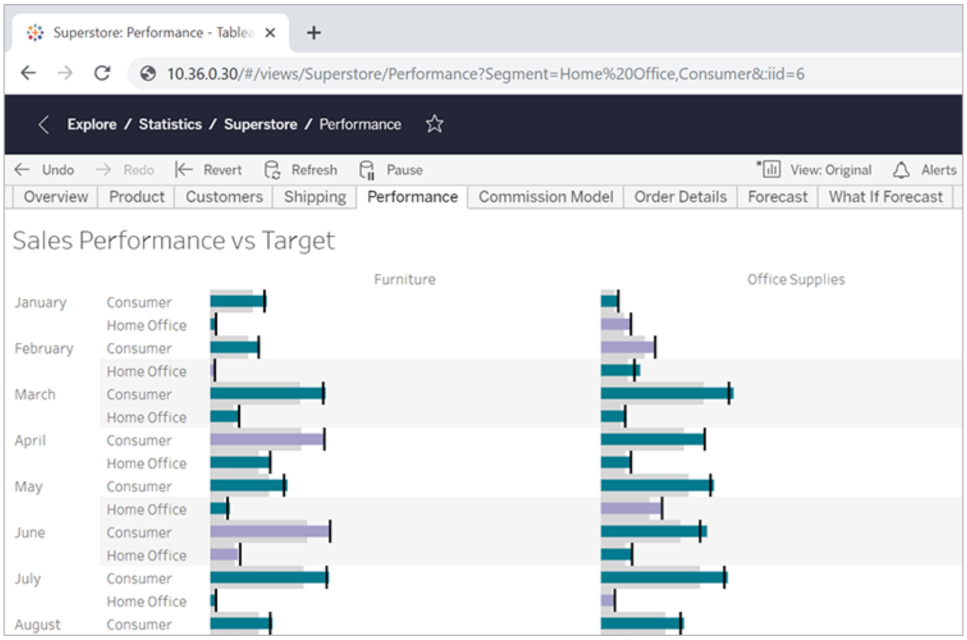
Home Office and Consumer

To show both home office and consumer sales, change the URL to:

`http:<servername>/#/views/Superstore/Performance`
`?Segment=Home%20Office,Consumer`

Notice that in the field name "Home Office," %20 represents the URL-encoded space character, as described in **Adding parameters to the base URL** on page 3299.

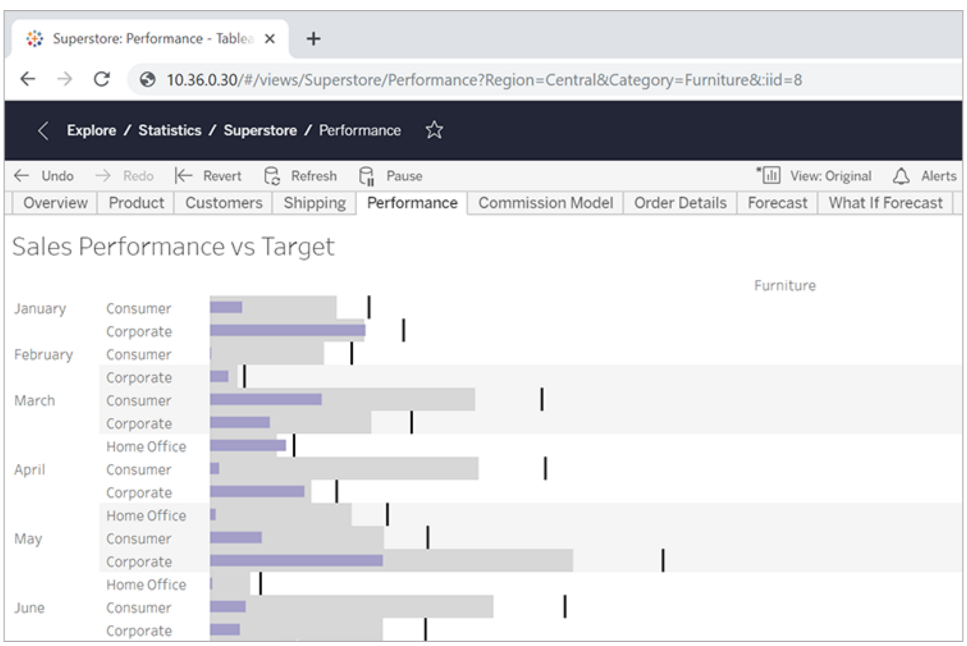
Here multiple values per parameter are separated with a comma, but not a space.



Furniture in the Central

To show multiple field/value pairs, separate them with an ampersand:

`http:<servername>/#/views/Superstore/Performance`
`?Region=Central&Category=Furniture`



Exclusive filtering

So far, the examples shown in this article display all values for the fields in the resulting views. For more flexibility, you can specify multiple values for a specific filter, and include an additional parameter that displays only particular intersections of those values.

The following URL describes the base, unfiltered view used in this example, showing department sales by region:
`http://<servername>/#/views/Superstore/Performance`

Next, to show sales for only the Furniture and Technology departments in the Central and West regions, you would add the following query string to the base URL:
`?Region=Central,West&Category=Furniture,Technology`

To show only Furniture sales in the Central region and Technology sales in the West region, add the `~s0` parameter to the query string:
`?Region`**`~s0`**`=Central,West&Category`**`~s0`**`=Furniture,Technology`

The final URL looks like this:
`http://<servername>/#/views/Superstore/Performance?Region~s0=Central,West&Category~s0=Furniture,Technology`

Special considerations for date filtering

When you want to filter date fields, take into account how dates are formatted and behave in a database environment.

- Date (and time) values passed via URL parameter need to match the following Tableau default format:
  `yyyy-mm-dd hh:mm:ss`
- Many databases store dates as datetime values, so you may need to include a time part in the value you provide in the parameter. The time part is based on a 24 hour clock, so 10:18 pm is specified as 22:18:00.

Example date parameters

The following example query strings use a date field called **Order Date**. As in the previous examples in this article, you would add these to the base URL of your view.

- If the Order Date field type includes only the date (with no time of day), and you want to show data only for July 8, 2018, the query string would look something like this:
  `?Order%20Date=2018-07-08`

- If Order Date includes the time part, to filter on July 8, 2018 at 10:18 pm, the query string might look like this:
  `?Order%20Date=2018-07-08%2022:18:00`
- If Order Date is only the date, and you want to filter on multiple dates, you would use commas, as described earlier in this article. For example:
  `?Order%20Date=2018-07-08,2018-07-09,2018-07-10,2018-07-11`

Parameters as DATEPART filters

To filter by date part, use the same nomenclature as in the default Tableau Desktop date hierarchy. For more information, see Date Functions in the Tableau Help.

year(Order%20Date)	Integer
quarter(Order%20Date)	Integer between 1 and 4
month(Order%20Date)	Integer between 1 and 12
day(Order%20Date)	Integer between 1 and 31
hour(Order%20Date)	Integer 0–23
minute(Order%20Date)	Integer 0–59
second(Order%20Date)	Integer 0–59
week(Order%20Date)	Integer 1–53
my(Order%20Date)	Six-digit integer: YYYYMM
mdy(Order%20Date)	Eight-digit integer: YYYYMMDD

Control Load Order for Multiple Embedded Views

You can control the order in which multiple views load for the people working with your views. This feature can be accessed only using embed code that relies on the Tableau JavaScript file.

In the following example, two views are embedded. The second view loads first, followed by the top view. If you embed multiple views and give them all the same load order value, or if you don't specify load order parameters, they are loaded in the order in which they appear on the page.

Script Tag Example

```
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='600' height='400' style-
='display:none;'>
   <param name='host_url' value='http://myserver/' />
   <param name='site_root' value='' />
   <param name='name' value='MyCoSales/TopPerformers' />
   <param name='tabs' value='yes' />
   <param name='toolbar' value='yes' />
   <param name='filter' value='Salesperson=Top 5' />
   <param name='load-order' value='0' />
</object>
<script type='text/javascript' src-
='http://myserver/javascripts/api/viz_v1.js'>
</script>
<object class='tableauViz' width='600' height='400' style-
='display:none;'>
   <param name='host_url' value='http://myserver/' />
   <param name='site_root' value='' />
   <param name='name' value='MyCoSales/SalesScoreCard' />
   <param name='tabs' value='yes' />
   <param name='toolbar' value='yes' />
   <param name='load-order' value='-1' />
</object>
```

Embed Dashboards

You can embed a Tableau dashboard in your own web site or wiki page and you can have it display in different layouts based on the size of the iframe.

- To automatically make the right layout appear for your users, regardless of which device they are using, create device-specific layouts for your dashboard in Tableau Desktop. When you do, Tableau Server and Tableau Online automatically display the correct layout based on iframe size, provided you use 100% for the iframe's width and height instead of exact pixel values (see below).

- To always make a certain layout appear, regardless of iframe size, use the `device` parameter in your embed code.

In the following example, the embed code displays a dashboard that's 800 pixels wide by 600 pixels high. Exact width and height values are part of the default embed code you get when you click the **Share** button at the top of a view or dashboard:

```
<script type='text/javascript' src-
='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
  <div class='tableauPlaceholder' style='width: 800px; height:
600px;' >
  <object class='tableauViz' width='800' height='600' style-
='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
  <param name='name' value='ProfitAnalysis/Sales_Dashboard' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
  <param name='filter' value=':original_view=yes' /></ob-
ject></div>
```

In this example, where the dashboard uses device-specific layouts, the style attributes for the div class have been removed and the object class values for width and height have been replaced with 100%. In most cases, the correct layout displays; that is, if width and height aren't also being controlled elsewhere in your CSS.

```
<script type='text/javascript' src-
='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
  <div class='tableauPlaceholder' >
  <object class='tableauViz' width='100%' height='100%' style-
='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
  <param name='name' value='ProfitAnalysis/Sales_Dashboard' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
```

```
  <param name='filter' value=':original_view=yes' /></ob-
ject></div>
```

If the correct layout doesn't display, it may be because the HTML page you're using for embedding has a `<!DOCTYPE html>` tag, and that tag is preventing items in the body of the page from resizing to 100% (see details on Stack Overflow). A workaround is to add the following lines to the body of your HTML page:

```
<style>
  html, body { height: 100% }
</style>
```

The following example assumes that the embedded dashboard has device-specific layouts. The `device` parameter is set to `phone`. This means that, no matter which device displays the dashboard, the layout created for phones will be the one that's displayed.

```
<script type='text/javascript' src-
='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
  <div class='tableauPlaceholder'>
  <object class='tableauViz' width='100%' height='100%' style-
='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
  <param name='name' value='ProfitAnalysis/Sales_Dashboard' />
  <param name='device' value='phone' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
  <param name='filter' value=':original_view=yes' /></ob-
ject></div>
```

Iframe dimensions and device layouts

The dashboard layout a device displays is based on the smallest dimension (height or width) of the iframe in which the Tableau view appears. Sometimes Desktop, Tablet, or Phone layouts may appear on other types of devices. For example, a Tablet layout may appear on a desktop computer if the display or browser window is small.

If the smallest iframe dimension is ...	This device layout appears ...

500 pixels or less	Phone
Between 501 and 800 pixels	Tablet
Greater than 800 pixels	Desktop

For details on how to create a dashboard that uses device-specific layouts, see **Create Dashboard Layouts for Different Device Types** on page 2801.

Embed Code for Custom Views

When you embed a custom view of a workbook or sheet, the default view is determined by these factors:

- If the embed code URL specifically refers to a custom view, that view is displayed by default.
- If the embed code URL does not refer to a custom view, the Default custom view is displayed by default.
- If no Default custom view has been defined, the original view is displayed by default.

> **Note:** To ensure the original view will be displayed by default in an embedded view, make sure the embed code URL for the name parameter does not explicitly refer to a custom view, and include the following `filter` parameter in the embed code: `<param name='filter' value=':original_view=yes'/>`.

In the following example, the embed code will always display the original view of the Profit Analysis sheet in the Profit Analysis workbook, because the `filter` parameter is set to `:original_yes`, and the `name` parameter does not refer to a specific custom view in the URL for the sheet.

```
<script type='text/javascript' src-
='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
  <div class='tableauPlaceholder' style='width: 1496px; height:
749px;'>
  <object class='tableauViz' width='1496' height='749' style-
='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
```

```
  <param name='name' value='ProfitAnalysis/ProfitAnalysis' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' />
  <param name='filter' value=':original_view=yes' /></ob-
ject></div>
```

In this example, the setting for the `name`parameter in this example specifically refers to the URL for a custom view named Furniture (in the Profit Analysis sheet in the Profit Analysis workbook).

```
<script type='text/javascript' src-
='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
  <div class='tableauPlaceholder' style='width: 1496px; height:
749px;'>
  <object class='tableauViz' width='1496' height='749' style-
='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
  <param name='name' value-
='ProfitAnalysis/ProfitAnalysis/Furniture' />
  <param name='tabs' value='yes' />
  <param name='toolbar' value='yes' /></object></div>
```

In this example, the `name` parameter does not refer to a specific custom view in the URL for the sheet, and the `original_view` parameter has not been specified. The embed code here will display the custom view that has been set to Default in the Profit Analysis sheet in the Profit Analysis workbook. However, if the original view is still the Default (no other custom view has been set to Default), then the original view will be displayed as the default view.

```
<script type='text/javascript' src-
='http://mysite.myserver.com/javascripts/api/viz_v1.js'></script>
  <div class='tableauPlaceholder' style='width: 1496px; height:
749px;'>
  <object class='tableauViz' width='1496' height='749' style-
='display:none;'>
  <param name='host_url' value='http://mysite.myserver.com' />
  <param name='site_root' value='' />
```

```
<param name='name' value='ProfitAnalysis/ProfitAnalysis' />
<param name='tabs' value='yes' />
<param name='toolbar' value='yes' /></object></div>
```

Embed Views into Wikis

You can easily embed a view into a wiki or other web page simply by putting the view inside an `<iframe>` tag.

1. Navigate to the wiki page you want to embed a view into.
2. Edit the page and add an `<iframe>` where the source is the URL you get when you click **Share** on a view and then click **Copy Link**. For example:

   ```
   <iframe src="http://myserver/views/Date-Time/DateCalc-
   s?:embed=yes&:toolbar=no"
   width="800" height="600"></iframe>
   ```

3. Save your changes.

   If you use Tableau Server and both it and the wiki are configured to use Active Directory to automatically authenticate users, they will immediately see the view. Otherwise, users will be asked to sign in before they can see the view.

Embed Images of Tableau Server Views

In addition to embedding a view into a `<script>` or `<iframe>` tag you can also embed the view as an image. When you embed an image the view is not interactive; however, it is updated every time the page fully reloads, showing the latest data.

> **Note:** This approach works only if users accessing the embedded image have an active web browser session with Tableau Server and are automatically signed in using Active Directory.

1. Navigate to the page where you want to embed the image.
2. Edit the page and add an `<img>` tag where the source is the URL you get when you click **Share** on a view and then click **Copy Link**. For example:

```
<img src="http://tableauserver/views/Date-Time/DateCalc-
s.png"
width="900" height="700">
```

Embed Tableau Server Views into SharePoint (Active Directory Authentication)

If your Tableau Server is configured to use Active Directory and automatically authenticate users, they can see views you embed in SharePoint pages with the Tableau web part.

If your Tableau Server instead uses Local Authentication to authenticate users, see **Embed Tableau Server Views into SharePoint (Local Authentication)** on page 3315. If you don't know which type of authentication your Tableau Server uses, ask your administrator.

Requirements

- To embed views, you need SharePoint 2013 or later, and an administrator must deploy the TableauEmbeddedView web part to your SharePoint server. For sample SharePoint code and instructions, look in this folder:

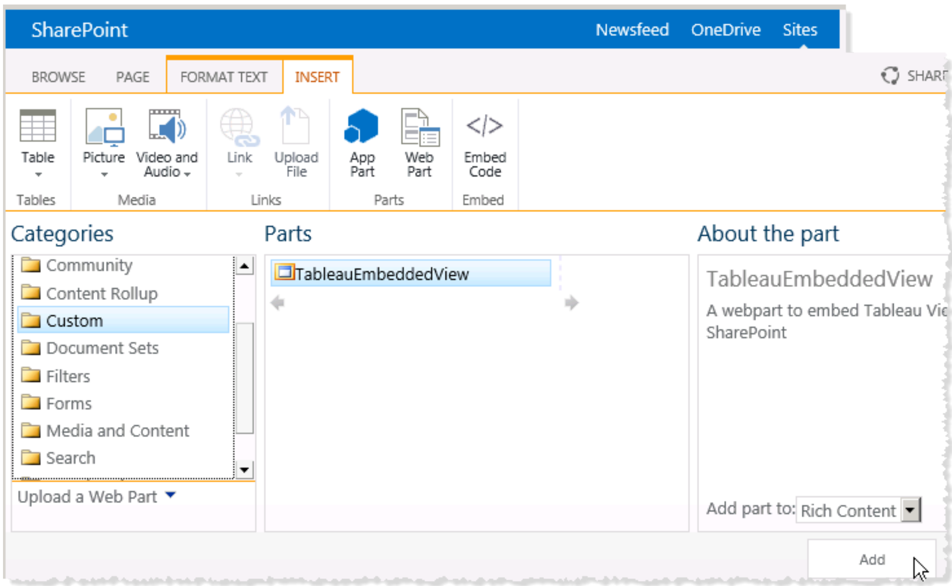
  ```
  C:\Program Files\Tableau\Tableau
  Server\packages\extras.<version_code>\embedding\sharepoint\
  ```

  The sample code is provided as an example and may require modification to work in your SharePoint deployment.
- To see embedded views, users must have a Tableau Server license and the same user name on SharePoint.

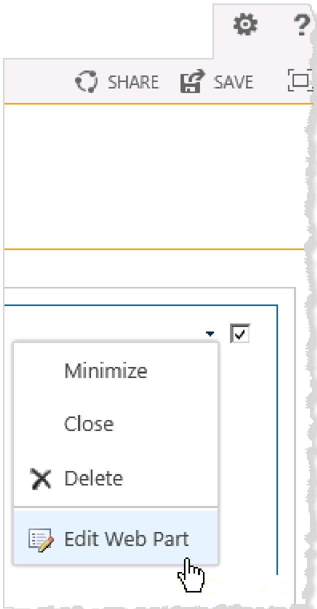
Embedding a View into SharePoint

You can embed the Tableau web part in a new or existing SharePoint page.

1. Open the page where you want to embed a view and switch to edit mode.
2. In the section of the page where you want to embed the view, on the **Insert** tab, click **Web Part**.
3. Under Categories, in the **Custom** (or **Miscellaneous**) folder, select **TableauEmbeddedView**, and then click **Add** in the lower-right corner.

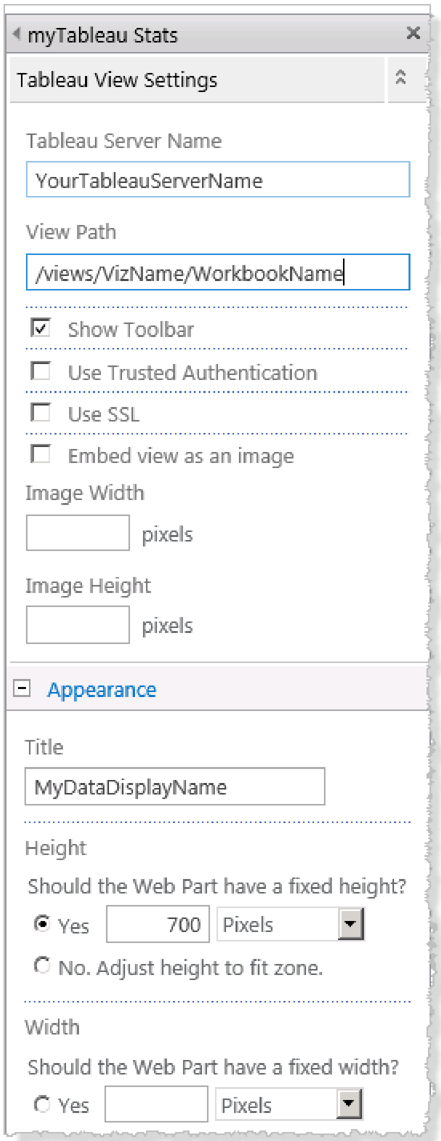


4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select **Edit Web Part**.



5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
   - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.
   - In **View Path**, enter the path to the view you want to embed.
   - Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.
   - In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed

height (for example, 700 Pixels) and adjust the width to fit the zone.



6. Click **OK** to apply the changes and exit edit mode.

The view will be embedded into the web part that you just created. To see the view, your users will be automatically authenticated using Active Directory.

Embed Tableau Server Views into SharePoint (Local Authentication)

If your Tableau Server uses Local Authentication to authenticate users, there are some extra steps you need to take so they can see views embedded in SharePoint pages.

If your Tableau Server instead uses Active Directory to authenticate users, see **Embed Tableau Server Views into SharePoint (Active Directory Authentication)** on page 3312. If you don't know which type of authentication your Tableau Server uses, ask your administrator.

Requirements

- To embed views, you need SharePoint 2013 or later.
- To see embedded views, users must have a Tableau Server license and the same user name on SharePoint.

Edit Security Permissions for TableauEmbeddedView.dll

Edit the security permissions for TableauEmbeddedView.dll so that all users of the operating system can use it.

1. Locate the TableauEmbeddedView.dll and TableauEmbeddedView.wsp files that install with Tableau Server. If Tableau Server is installed on drive C, the files will be in the following directory:

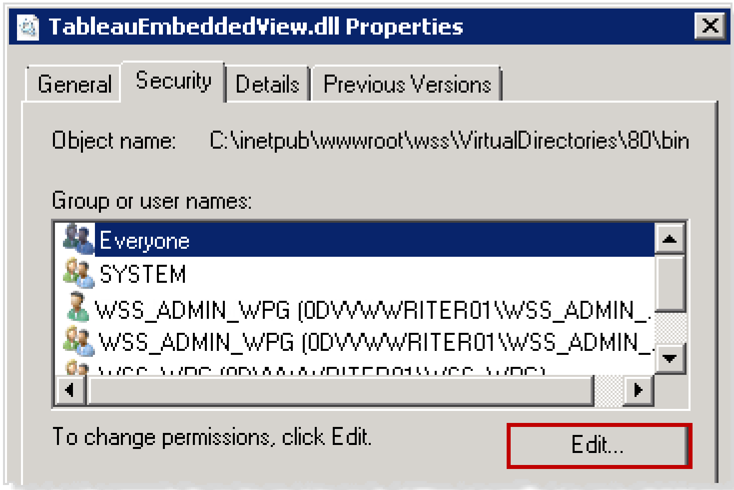
   ```
   C:\Program Files\Tableau\Tableau
   Server\packages\extras.<version_code>\embedding\sharepoint\
   ```

2. Copy the files to the root directory of your SharePoint server. The root directory is usually located at `C:\Inetpub\wwwroot\wss\VirtualDirectories\<port>\bin`, for example:

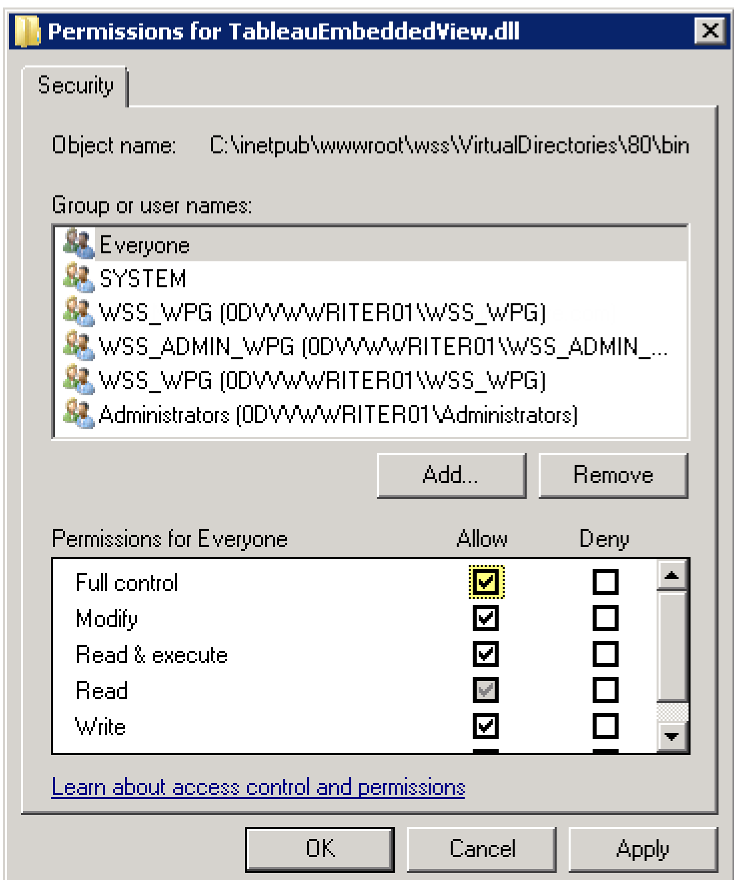
   ```
   C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin
   ```

3. To edit the security permissions on TableauEmbeddedView.dll, right-click **TableauEmbedded.dll** and then select **Properties** > **Security**.

4. Under **Group or user names**, select **Everyone**, and then click **Edit.**



5. Under **Permissions for Everyone**, for the **Full control** permission, select **Allow**.



6. Click **OK**.

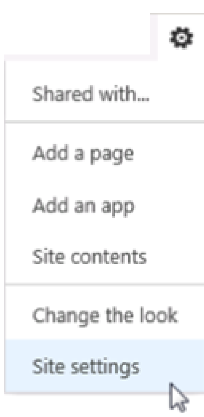
Install and Deploy TableauEmbeddedView.wsp

The TableauEmbeddedView.wsp file gives SharePoint more information about what to do with the .dll file. You copied the TableauEmbeddedView.wsp file to the SharePoint root directory in the previous procedure. To install and deploy the .wsp file, follow these steps:

1. Open SharePoint 2013 Management Shell and enter the following command:

   ```
   Add-SPSolution -LiteralPath
   "C:\Inetpub\wwwroot\wss\VirtualDirectories\80\bin\TableauEmb
   eddedView.wsp"
   ```

2. On the SharePoint Central Administration home page, click **System Settings**.
3. In the **Farm Management** section, click **Manage farm solutions**.
4. On the Solution Management page, click the solution that you want to deploy.
5. On the Solution Properties page, click **Deploy Solution**.
6. On the Deploy Solution page, in the **Deploy When** section, select one of the following options:
   - **Now**
   - **At a specified time**. Specify a time by using the date and time boxes.
7. In the **Deploy To?** section, in the **A specific web application** list, click **All web applications** or select a specific Web application, and then click **OK**.
8. Open your SharePoint site. Click the settings icon, and then select **Site settings**.

9. Under Site Collection Administration, click **Site collection features**.

10. Scroll to the TableauEmbeddedView feature and then click **Activate** to activate the feature.

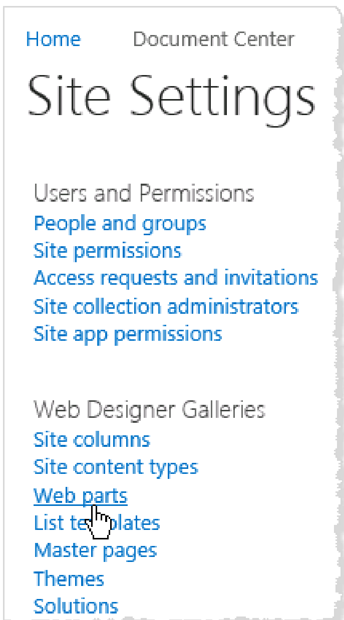
Verify the Web Part's Deployment

In the following procedure, you will verify that the Tableau web part is installed.

1. Open your SharePoint site in a web browser.
   It may take a few moments for the site to appear.

2. Click the settings icon, and then select **Site settings**.

3. Under **Web Designer Galleries**, click **Web parts**.



4. Confirm that **TableauEmbeddedView.webpart** is listed.

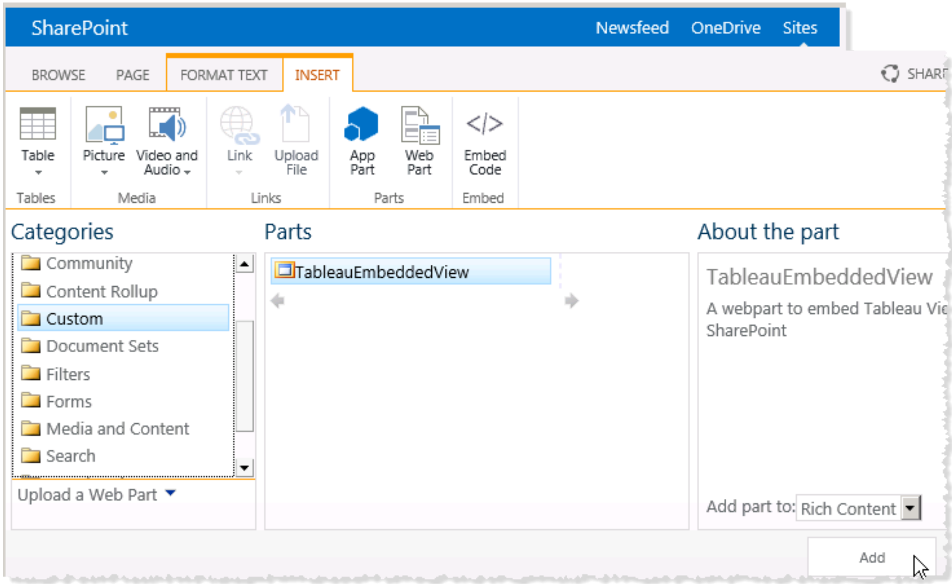
Embed a View Using the Tableau Web Part

You can embed the Tableau web part in a new or existing SharePoint page.

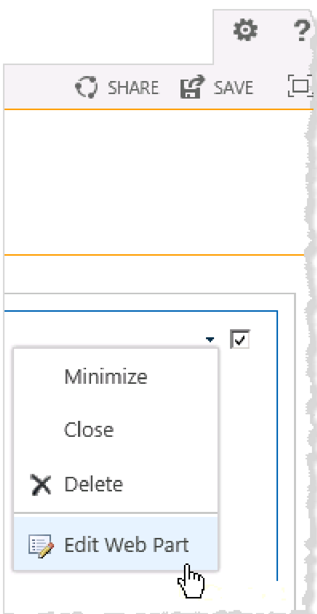
1. Open the page where you want to embed a view and switch to edit mode.

2. In the section of the page where you want to embed the view, on the **Insert** tab, click **Web Part**.

3. Under Categories, in the **Custom** (or **Miscellaneous**) folder, select **TableauEmbeddedView**, and then click **Add** in the lower-right corner.

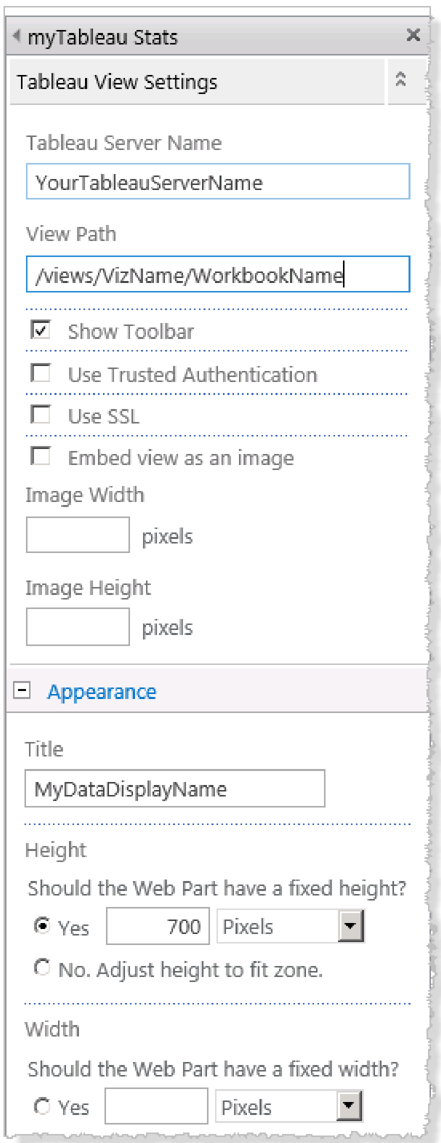


4. Select the TableauEmbeddedView web part, click the drop-down arrow, and then select **Edit Web Part**.



5. On the right side of the page, you can specify the attributes of the TableauEmbeddedView web part.
   - In **Tableau Server Name**, enter the name of your Tableau Server. You do not need to enter "http://" before the Tableau Server name.

- In **View Path**, enter the path to the view you want to embed.
- Specify whether you want to show the toolbar, use Trusted Authentication, use SSL, or if you want to embed the view as an image instead of as an interactive view.
- In the **Appearance** section you can specify a **Title** for the web part, the **Height**, **Width**, **Chrome State**, and **Chrome Type**. In general you should specify a fixed height (for example, 700 Pixels) and adjust the width to fit the zone.



6. Click **OK** to apply the changes and exit edit mode.

Now the view is embedded in the page and users who access it will be automatically signed in based on their user name and password for SharePoint.

> This is an example of embedding views into SharePoint using the provided .dll file. You can also embed views into other types of web applications. For more information, see the JavaScript API on the Tableau Developer Portal.

Embed Tableau Views into Salesforce

You can easily embed Tableau views into Salesforce Lightning pages using the Tableau Viz Lightning web component that is available from the Salesforce AppExchange. When you edit a Lightning page, you simply drag and drop the Tableau Viz Lightning web component onto the page and then provide the URL for the Tableau view. You can embed Tableau views from Tableau Server, Tableau Online, or Tableau Public. The Tableau Viz Lightning web component supports all versions of Tableau. Using the Tableau Viz Lightning web component, Salesforce administrators and developers can quickly integrate Tableau dashboard into Salesforce pages.

You can set up Tableau Server and Tableau Online for single sign-on (SSO), so that users of the Tableau Lightning web component won't have to sign in to Tableau every time they visit the Salesforce page.

The Tableau Lightning web component includes built-in support for filtering the view based upon the Lightning record page you are embedding the view in. The component lets you configure how the view is displayed in Salesforce, and provides additional filtering capabilities based upon Tableau and Salesforce fields you select.

Requirements and Permissions

- The Tableau Viz Lightning web component is designed to run on the Salesforce Lightning Experience. Lightning Web Components are supported in many Salesforce experiences and tools such as Lightning Communities and Salesforce App. The Tableau Viz Lightning web component does not support Salesforce Classic.
- To install and configure the Tableau Viz Lightning component, you need to have administrator privileges for the Salesforce org.
- Context and advanced filtering of the Tableau view is only available on Lightning Experience record pages.

The Tableau Viz Lightning web component only supports the Salesforce Lightning Experience. If you want to embed a Tableau view into Salesforce Classic, you have other options. You could use the Salesforce Canvas Adapter for Tableau, or create a custom solution using Apex code and Visualforce pages. For more information, see Tableau Viz Lightning Web Component Does Not Work on Salesforce Classic.

Configure single sign-on (SSO)

If you are embedding a Tableau view from Tableau Server or Tableau Online, you might want to configure SSO with Salesforce. That way when users sign into Salesforce and open a page that contains an embedded Tableau view they won't need to re-authenticate with Tableau. The Tableau Viz Lightning web component only supports SAML as the SSO method. The SAML IdP used for Tableau authentication must be either the Salesforce IdP or the same IdP that is used for your Salesforce instance. Configuring Tableau Server or Tableau Online requires Tableau administrator permissions.

Note: Users in Tableau Online who are configured with Salesforce Authentication will need to reauthenticate to view embedded visualizations in Tableau Online.

- For information about setting up SSO with Tableau Online, see Configure SAML with Salesforce and Configure SAML for Tableau Viz Lightning Web Component.
- For information about setting up SSO with Tableau Server, see Configure SAML for Tableau Viz Lightning Web Component.

Install the Tableau Lightning web component

The Tableau Lightning web component is available from the Salesforce AppExchange.

1. Get the component from the Salesforce AppExchange.
2. Install the **Tableau Viz Lightning Web Component** in your Salesforce org.

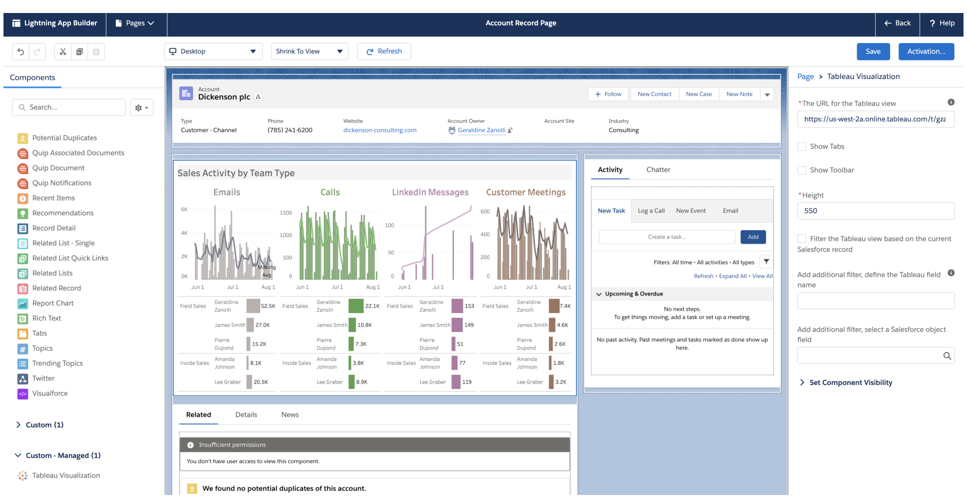
Add the Tableau view to the Lightning page

After you have successfully installed the Tableau component for your Salesforce org, you can use the component to add a Tableau view to Lightning Experience pages.

1. From the App Launcher ( ), find and select a page to embed the Tableau visualization. (For example, select **Sales** or any other App that provides a home or record page where you can embed the Tableau Lightning web component).
2. Click the Setup gear ( ) and then select **Edit Page**.
3. Drag the **Tableau Visualization** component from the Custom - Managed area of the Lightning Components list to the top of the Page Canvas.
4. Select a Tableau view to display by providing the URL for the view in the **Enter the Viz URL** text box.
5. To find the URL for a view on Tableau Server or Tableau Online, click **Share** on the toolbar and select **Copy Link** from the Share View dialog box. For Tableau Public, copy the URL for the view from the address bar in your browser.
6. In Salesforce, use the options for the **Tableau Visualization** component to control the size of the view and whether you want to show the Tableau toolbar or any tabs for the view.

   You can also control when the Tableau Viz Lightning web component appears by selecting component visibility filters. For more information about the **Set Component Visibility** options, see Dynamic Lightning Pages in the Salesforce Help.

For the best experience, you might want to configure Tableau and Salesforce to use SSO. See **Configure single sign-on (SSO)** on the previous page.



Filter the view based upon the page context

The Tableau Lightning component supports two ways of filtering the Tableau visualization. These methods of filtering only work on Lightning record pages. Filtering is not available for Home pages or App pages. For record pages, you can automatically filter the Tableau visualization based upon the page it is embedded in (*context filtering*), or you can specify the fields to use for filtering in Tableau and in Salesforce to create more sophisticated views.

To have the view filtered based upon the page it is embedded in, the Tableau view needs to have a field that corresponds to the record page ID. For example, if you want to embed a view for sales on a user's page that shows just the sales for that user, the Tableau view should include a field or parameter that holds the record IDs for users.

1. In your Salesforce org, select a record page where you want to embed the viz (for example, Users or Opportunities).
2. Edit the record page and configure the component by entering the URL of the Tableau view that contains the data you want to display. The Tableau visualization must include a field that holds the record IDs (for example, user or account IDs).
3. Select **Filter visualization based on the page** and save your changes to the page.

   When the record ID of the page matches the corresponding value in a field in the Tableau view, the Tableau Lightning component applies that filter. For example, if you want to embed a Tableau view on the Account page and filter for the specific account, the Tableau view needs to have a field named **Account ID**. The Tableau Viz Lightning web component uses the default field name when a user connects to Salesforce using Tableau.

Apply filters based upon on Tableau and Salesforce fields

If you want more control over filtering the Tableau visualization on a records page, you can map a field or parameter in Tableau to a specific field in Salesforce. The Tableau field or parameter needs to be on the view that you are embedding. The Salesforce field must be a qualified field on the page where you are embedding the view. When the values of these two fields match, the Tableau Visualization component will automatically filter the Tableau view.

To apply the advanced filtering, add the **Tableau Visualization** component to a record page. For example, if you have a Tableau view that contains Salesforce data that is related to sales information, you can add that view to your opportunity record pages.

1. In your Salesforce org, click the App Launcher ( ), find and select **Sales** (or any other App that provides a record page where you can embed the Tableau Lightning component). For example, the click Accounts tab. Select an account, for example, Burlington Textiles from the All Accounts list.

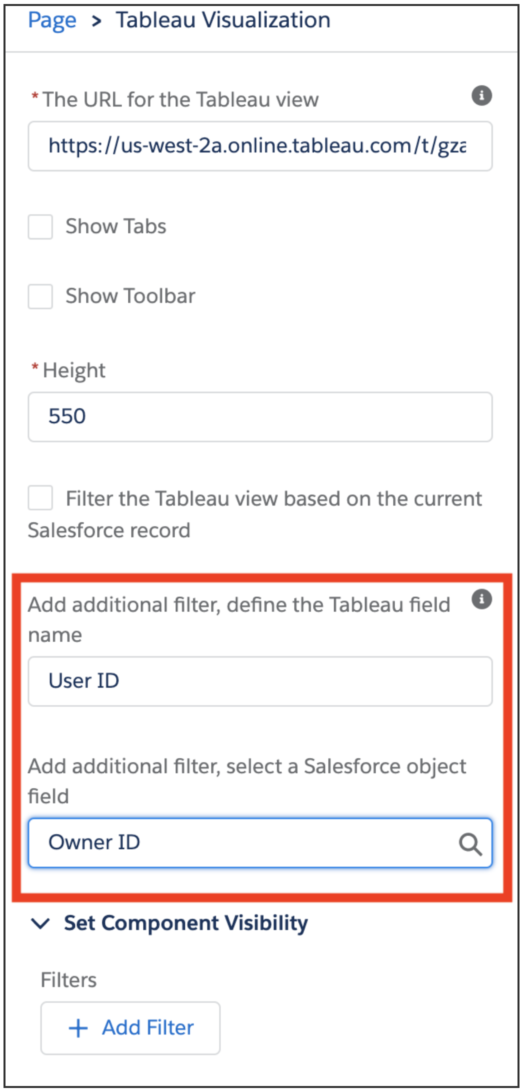
2. Click the Setup gear ( ) and then select **Edit Page**.

3. Drag the **Tableau Visualization** component from the Custom area of the Lightning Components list to the top of the Page Canvas.

4. Configure the component.

   - Enter the URL of the Tableau view that contains the sales data you want to display.
   - Provide the name of the Tableau field to use for filtering. The Tableau field needs to be the name of a field in the view that you are embedding. For example, the view might contain a field for the **User ID**.
   - Provide the name of the Salesforce field to use in filtering. You can select the qualified field names for the page from the drop-down list. For example, on the account record page, you could select **Owner ID**.

   When the values from these two fields match, the Tableau Visualization component automatically filters the view.



5. Save your changes to the page.

Link to a PNG, PDF, or CSV of a View

Note: To download to these formats rather than link to them, instead see **Download Views and Workbooks** on the next page Or, if you're using Tableau Desktop, see Export Views and Export Data.

If you work with a website or application that doesn't support embedded, interactive Tableau views, you can link to PNG or PDF versions of them.

If you're familiar with scripting, you can also use links to automatically convert multiple views into PNGs, PDFs, or even CSVs. The resulting files can be shared with people who lack Tableau Online or Tableau Server accounts, incorporated into presentations, or archived for future reference.

When you link to any of these formats, they always load the latest data available on the server. But be aware that CSVs of dashboards link to only one sheet—the one whose title is first in alphabetical order.

1. With the help of a Tableau content owner or site administrator, ensure that your audience has permission to access the content. (If you're using a script to process multiple files, only you need access.)
   - For links to PNG and PDF files, users need the Download Image/PDF permission.
   - For links to CSVs, users need the Download Full Data permission.
2. Replace the end of a view's browser URL with the appropriate file extension.

   For example, change `http://<servername>/#/views/<workbook>/sheet?:iid=7` to `http://<servername>/#/views/<workbook>/sheet.png`
3. If you want to filter a view, add a question mark after the file extension, followed by URL parameters that reflect the view's data structure.

   For example, change `http://<servername>/#/views/<workbook>/sheet.png` to `http://<servername>/#/views/<workbook>/sheet.png?Region=South&Department=Sales`

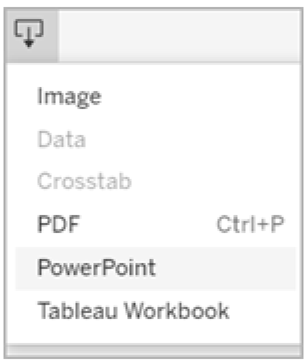
Download Views and Workbooks

> **Note:** To link to views in other formats rather than download them, instead see **Link to a PNG, PDF, or CSV of a View** on the previous page. Or, if you're using Tableau Desktop, see Export Views and Export Data.

1. At the top of a view in Tableau Online or Tableau Server, click **Download**. Or, click the download button wherever it appears on the page.

2. Select a download format:

Note: The download formats available to you depend on permissions granted by Tableau content owners and site administrators.



- **Image**: Downloads an image of the view in .png format. Any filters, parameters, or selections currently applied in Tableau are reflected in the downloaded image.
- **Data**: Opens a new tab in the browser window and displays the view's data in summary and detail. You can then download the data as a comma-separated value (.csv) file.

  When downloading from a dashboard, first click the specific sheet with data you want.
- **Crosstab**: Opens a dialog window to select download options. If viewing a dashboard, select a sheet from the dashboard to download. Under Select Format, select .csv or Microsoft Excel .xlsx.

  For dashboards, all sheets will be listed, including hidden sheets.

  Any filters, parameters, or selections currently applied in Tableau are reflected in the downloaded crosstab.

- **PDF**: Opens a dialog window to select download options. Under **Include**, select the the part of the workbook you want to download. Select this view, specific sheets from a workbook or dashboard, or select all. Select Scaling to control the image's appearance on the PDF. Select Paper Size and Orientation.

  If you're downloading a dashboard to PDF format, web page objects aren't included.

- **PowerPoint**: Downloads selected sheets as images on individual slides in a PowerPoint presentation. For untabbed views, you can select only currently visible content. However, hidden sheets visible in a dashboard can be selected if you choose **Specific sheets from this workbook**. If you select a story sheet, all story points export as separate slides.

  Any filters, parameters, or selections currently applied in Tableau are reflected in the exported presentation. The generated PowerPoint file includes a title slide with the name of your workbook and the date the file was generated. The title is a hyperlink that opens the workbook in Tableau Online or Tableau Server.

- **Tableau Workbook**: Downloads a workbook you can open with Tableau Desktop. (As an alternative, you can select **Server** > **Open Workbook** in Tableau Desktop.)

  You have the option of downloading the workbook to different versions of Tableau or keeping the workbook in its current version. For example, if the workbook needs to be opened in Tableau Desktop 2020.4, select Tableau 2020.4. For information about version compatibility, see **Make Workbooks Compatible with Older Versions** on page 3063.

> **Note:** Downloading extremely large amounts of data can affect server performance and might not complete successfully. If you encounter these issues, try exporting the data directly from the underlying data source.

Web Authoring and Tableau Desktop Feature Comparison

For anyone familiar with Tableau Desktop and new to the web authoring environment in Tableau Server and Tableau Online, this topic provides a summary of the web features that you use similarly to the way you do in Tableau Desktop. It also lists some fundamental differences between the two environments.

Note: This topic summarizes core authoring functionality and does not cover every difference between the desktop and web environments.

Features listed by version

For a list of the latest web editing features to be added to each release, see the web authoring sections in What's New in Tableau and What’s New in Tableau Online.

For a list of features compared by version, see the Tableau Desktop vs Tableau Web Editor viz on Tableau Public. This information is curated and maintained by Andrew Pick of the Information Lab, a Tableau Gold partner in the UK.

Disclaimer: Clicking this link will take you away from the Tableau website. Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact The Information Lab if you have questions regarding its content.

General differences in web authoring

- Your authoring capabilities are determined by your license level. For an overview of what you can do with each license level, see What can I do with a Tableau site?
- You can access right-click menu actions on fields in the view, but not on every item in the workspace.
- Keyboard shortcuts for web authoring and Tableau Desktop are not the same. For a list of web authoring keyboard shortcuts, see Shortcuts for web authoring.

Web authoring capabilities

In the web environment, you can connect to data and create workbooks from those data sources, or data published through Tableau Desktop. You can edit views created on the web or published from Tableau Desktop.

Administrators can set at the site level what web authoring abilities users can have. Explorers can edit workbooks, create new workbooks from published data sources, connect to published data sources, and create and edit views, dashboards, and stories. Creators have those same capabilities, but can also create new workbooks, connect to data on the web, and use Ask Data and (in Tableau Online) Dashboard Starters to quickly dive into analysis.

Data management

- **Creators**: Connect to data sources, upload files (text, excel, and Tableau workbooks), or use pre-built Dashboard Starter templates for certain data sources. For more information, see Creators: Connect to Data on the Web.
- **Creators**: Prepare data on the web in the Data Source page. For more information, see Creators: Prepare Data on the Web.

  > **Note**: There is a limitation on the number of rows that can be viewed in the Data Source page when authoring data on the web, determined by browser:
  > - Internet Explorer: 10,000 rows
  > - Other browsers: 100,000 rows
  >
  > Independent of browser, the total number of records (rows by columns) that can be viewed in the Data Source page on the web is **3 million**.

  - Run Initial SQL when connecting with some data sources
  - Relate your data to combine data from multiple tables
  - Join data from different tables in the same data source or from different databases using a multi-connection data source
  - Add a join calculation
  - Union data
  - Pivot data
  - Copy values in a grid (Ctrl+C, or Command-C on a Mac)
  - Edit Data Sources
  - Clean data using the Data Interpreter
  - Create a Custom SQL query
- **Explorers**: Connect to published data sources.

- Blend published data sources.
- Save a data source (embedded in a published workbook) as a separate, published data source.
- Change aggregation of measures in the view. Change the default aggregation of measures in the Data pane.
- Search for fields in the schema.
- Duplicate, hide, or rename fields.
- Change the data type of fields.
- Convert measures to dimensions or vice versa.
- Convert a discrete field to continuous and vice versa. This option is available for measures and date dimensions.
- Assign a geographic role for a field.
- Create aliases for members of dimensions.
- Create and edit groups.
- Create and edit sets (conditioned sets are not available)
- Create, edit, and remove parameters. Formatting numbers and dates, and adding comments for the parameter are not supported on the web.

Analytics

- Create, edit, rename, duplicate, and clear sheets (views, dashboards, and stories) in a workbook.
- Use Ask Data to automatically create views.
- Use Explain Data to automatically create views.
- Search for fields in the Data pane with schema search
- Drag fields to the view, Rows, Columns, and different mark types in the **Marks** card.
- Use **Show Me** to create views. Also, from the Data pane, select and drag a field of interest to the view area to automatically create a "Show Me" view.
- View underlying data (via tooltips).
- Viz in Tooltip works in web views, but must be configured in Tableau Desktop. Viz in

Tooltip worksheets can be hidden, the same way you would hide worksheets used in stories or dashboards.

- Actions work in web views, but must be configured in Tableau Desktop.
- Create and edit calculated fields.
- Create bins from continuous measures, and edit bins.
- Create and edit table calculations, and use quick table calculations.
- Create sets and show set controls
- Use the **Analytics** pane to drag reference lines, trend lines, and other objects into the view. Edit reference lines, trend lines, and bands. Create and configure reference distributions on a continuous axis. Adding a cluster, and adding and modifying a forecast, are not supported for the web.
- Create groups by selecting marks in the view and then clicking Group Members (paperclip) in the tooltip for that selection. You can also edit existing groups in the Data pane.
- Create hierarchies by dragging one dimension onto another in the Data pane. **Note**: You can't create hierarchies when fields are already grouped within a folder.
- Change options for interacting with maps, including enabling or disabling pan and zoom, or showing map search, the view toolbar, or map scale. Users can also map units.
- Drill up and down a continuous hierarchy in the view. In a view with a continuous hierarchy, hover near the headers on a continuous axis to display the + and - controls. Click to drill down or up.
- Show labels, totals, and subtotals.
- Show, hide, and format mark labels
- Show and hide titles and captions.
- Show and hide cards for filters and highlighters.
- Show, hide, and resize headers in the view.
- Swap X and Y axes. Resize axes in the view.
- Change the view size.
- Show and hide the View Toolbar for any view or dashboard.
- Duplicate a sheet as a crosstab view.

Filtering and sorting

- Use data highlighting.
- Add, edit, and remove filters, and edit filter control layouts. **Wildcard** filter not available.

> **Note**: There is a limitation on the number of results that can be filtered when authoring data on Tableau Online or Tableau Server. Only the first 100 results are returned to limit the performance impact one user has when loading a large domain on the server.

- Filter across published data sources.
- Apply filters to multiple sheets
- Create context filters (**Add to Context** option in the Filters shelf) and dependent filters (**Only Relevant Values** option in a filter control shown in a view).
- Apply table calculation filters to totals in the view.
- Show hidden fields, and exclude or remove fields from the view.
- Sort fields in the view in ascending or descending order. Access the **Sort** dialog box by right-clicking a dimension on the Rows or Columns shelf. Nested sorting on dimension values within the context of each pane.
- Drag and drop headers to create a custom sort order within a view.

Formatting

- Resize the width of row headers and the height of column headers.
- Edit workbook formatting, including formatting lines.
- Edit worksheet and dashboard titles.
- Edit axes (double-click an axis in the view). Other options available: **Synchronize dual axes**, clearing the axis range (**Reset**), and tick mark settings. Enable or disable **Dual axis** in a field context menu (right-click a measure field on Rows or Columns shelf). Logarithmic scales can be positive or symmetric (includes 0 and negative values).

- Edit number formatting (decimal places, percentage, thousands separator, units, and currency).
- Create, edit, move, and resize point, mark, and area annotations.
- Add and edit dashboards objects, including: horizontal and vertical layout containers, text, images, navigation buttons, web page links, and dashboard extensions.
- Create transparent worksheet backgrounds (set background color to **None**). Combine transparent worksheets with transparent filters, highlighters, and parameters.
- Change the color palette. For categorical fields you can assign specific colors and custom colors (using a hex code) to data items. For continuous fields, you can set custom colors for start and end colors (using a hex code).
- Create, rearrange, and preview device-specific dashboard layouts
- Set a dashboard item's exact size, position, and spacing.
- Add padding, borders, and background colors around items in dashboards.
- Select a background map in map views.
- Legends per measure. If you create separate color legends for the measures in your view, Tableau assigns the default color palette to each new color legend. To change the color legend for each measure, click the drop-down arrow on the color legend to open the **Edit Colors**dialog box and select the palette that you want to use. For more details, see Legends per measure.

Related topics

What can I do with a Tableau site?

Set a Site’s Web Authoring Access

Getting Started with Web Authoring

Creators: Connect to Data on the Web

Creators: Prepare your Data on the Web

Build Views on the Web

Grant Web Edit, Save, and Download Permissions

Install or Upgrade Tableau Desktop

For comprehensive installation information, see Install or Upgrade Tableau Desktop in the Tableau Desktop Deployment Guide.

When you install or upgrade you have the option to share usage data with us to help us continually improve our product to better meet your needs. For more information about the data we collect, see Tableau Product Usage Data. For information about how to opt out of sharing usage data see Turn off usage reporting in the Tableau Desktop and Tableau Prep Deployment Guide.

Accessing the repository from a previous version

When you upgrade to Tableau 2021.2 from previous versions, your repository is upgraded. Any bookmarks, workbooks, and data sources that you had in your old repository will still be accessible by the application. In addition, the new sample data sources and workbooks will replace the old samples unless you have modified them and saved them as your own.

> **Note**: Your recent workbooks are automatically imported and restored to your Start page along with your recent connections list and any organization-specific settings on the Discover pane.

Accessing the repository from a beta version

If you have participated in the Tableau Software Beta program, you also have a beta repository. While this folder will still exist after you install Tableau 2021.2, the application will no longer access it. To make your beta workbooks accessible in Tableau 2021.2, copy the workbooks from the beta repository to your new 2021.2 repository.

Turn Product Updates Off or On

To ensure that you always have the most up-to-date features, security resolutions, and corrected issues, Tableau Desktop includes a product update feature. When you start Tableau

Desktop, product updates prompts you to download an updated maintenance version of Tableau Desktop, if one exists. The update downloads immediately and then installs when you exit Tableau.

You can also choose to postpone or skip the update. If you do this, you can always check for product updates at any time by selecting **Help** > **Check for Product Updates**.

For more information about how to turn product updates off or on, see the Control Product Updates article in the Tableau Desktop and Tableau Prep Deployment Guide.

Note the following:

- The product update installers are downloaded to the Downloads/TableauAutoUpdate folder. If the Downloads folder doesn't exist, the installers are downloaded to the TEMP/TableauAutoUpdate folder.
- Updates are not downloaded and installed on your computer if your Product Maintenance has expired. For more information, see the Renewing your Licenses with Tableau.
- You may not be prompted for product updates. There are a number of reasons why this may be so. For details, see the **Troubleshoot maintenance updates** section in the Troubleshoot Tableau Desktop Installation article in the Tableau Desktop and Tableau Prep Deployment Guide.

Turn off product updates

Product updates is on by default. You can turn off product updates from the Help menu, or on Windows, by running the installer.

Use the Help menu

Select **Help** > **Settings and Performance** > **Enable Automatic Product Updates** and clear the check box.

Run the installer (Windows only)

To turn off product updates on Windows, run the Tableau Desktop installer, and then click **Customize**. In the Custom setup dialog box,

1. Clear the **Check for Tableau product updates** check box.
2. Click **Install**.

To turn product updates on again, run the installer, click **Customize**, and select the check box.

Administrators control product updates

As an administrator, you can turn product updates off or on for your users. You can also determine the Tableau Desktop version that your users will update to. Rather than having users update to the version of their choice (or choose not to update), you can make sure that your users update to the version you choose. For more information, see the Control Product Updates article in the Tableau Desktop and Tableau Prep Deployment Guide.

Maintain Licenses for Tableau Desktop and Tableau Prep

Tableau Desktop and Tableau Prep can be licensed under a term license model. Term licenses, also known as subscription licenses, allow you to use and update Tableau Desktop and Tableau Prep for a specified period of time.

Term licenses must be renewed and the product key refreshed to continue providing uninterrupted service. You can continuously renew the term license as each specified period expires. If you don't renew your term license and the term expires, Tableau will stop working and you will no longer have access to the software. For more information about renewing your license, see How to Renew your Tableau Licenses.

> **Note**: Trial licenses for Tableau Desktop or Tableau Prep expire after a set period of time, usually 14 days. After the trial period expires, you'll need to purchase a license to continue using the product.

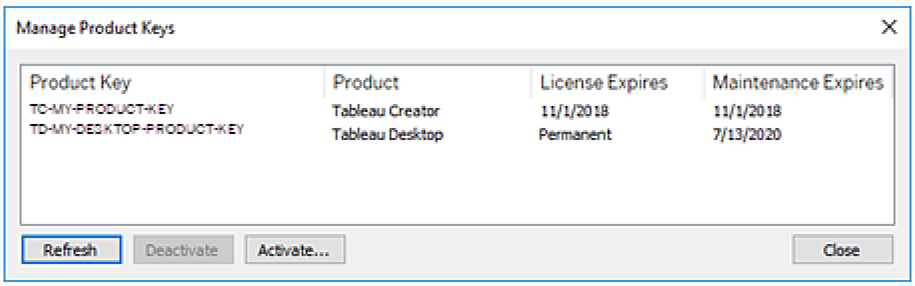
View data about your license

After you install Tableau Desktop or Tableau Prep open the application and then navigate to **Help** > **Manage Product Keys** from the top menu to see information about the type of license you have and when it expires.

You can also activate or deactivate a product key or refresh a maintenance product key from this dialog.

Note: Tableau offers term licenses that provide a range of capabilities. The type of license that you have is displayed in the **Product** field. For more information about the different type of user-based licenses that are available, see User-based licenses in the Tableau Server help.

Existing Tableau Desktop users may have a perpetual (permanent) license. Perpetual licenses don't expire. However, to get access to product updates and technical support you must purchase Support and Maintenance services. These services must be renewed to continue receiving the service. Perpetual (permanent) licenses are no longer available for Tableau Desktop.



Use the following buttons to take action on your product key:

- **Refresh** (Tableau Desktop only): Click the **Refresh** button to refresh a maintenance license that is expiring, then close and restart Tableau Desktop. If the **Maintenance Expires** date doesn't update, check with your license administrator as the key or maintenance agreement may have changed.

  A product key whose License Expires value is listed as "Permanent," as shown in the Manage Product Keys dialog box above, is a legacy product key. You can refresh a Permanent product key at any time as long as the maintenance end date listed in the Tableau Customer Portal is higher than the date reflected in the Desktop Manage Product Keys dialog box.

  If the product key has reached its expiration date, you cannot refresh the product key. Visit the Tableau Customer Portal to obtain an updated subscription product key and perform a new activation. If the product key has not reached its expiration date, you can refresh the product key. When you refresh a product key that has not yet expired, only the "License Expires" value will change and not the product key. The product key will change when it reaches its expiration date.

To refresh a maintenance key from the command line see Refresh the product key in the Tableau Desktop and Tableau Prep Deployment guide.

> **Note:** You cannot refresh the product key if Tableau Desktop is offline. If you are activating Tableau Desktop in offline mode, you must obtain and activate a new key from the Tableau Customer Portal.

- **Deactivate**: Select a product key in the list then click **Deactivate** to deactivate the product key. Deactivate a product key if you need to move the product key to another computer or when you no longer need the product key on this computer.

  For more information about deactivating a product key, see Move or Deactivate Product Keys in the Tableau Desktop and Tableau Prep Deployment guide.

- **Activate**: After Tableau Desktop or Tableau Prep is installed, click **Activate** to open the activation dialog and enter your product key. If you get an error and can't activate Tableau Desktop or Tableau Prep using your product key, contact Tableau Support.

  For more information about activating a product key, see Activate and Register your Product in the Tableau Desktop and Tableau Prep Deployment guide.

Automatically refresh product keys using zero downtime licensing

Beginning in Tableau version 2021.1, internet-connected Tableau Desktop and Tableau Prep Builder users no longer have to manually refresh product keys. Licenses are automatically refreshed without requiring any action as their subscription approaches expiration.

Tableau Desktop and Tableau Prep Builder will attempt to silently refresh an active product key and will warn users 14 days before their license is set to expire if the silent refresh was unsuccessful. Tableau will attempt to refresh a product key twice (at 14 days and 1 day before license expiration) to reflect license end date extensions as a result of your subscription renewal.

If the license end date for Tableau Desktop or Tableau Prep Builder has been extended prior to a silent refresh attempt, the updated license term is updated automatically and downloaded into the trusted storage of the computer where Tableau Desktop or Tableau Prep Builder is running. This allows users to continue using Tableau Desktop or Tableau Prep Builder without interruption. However, if no updated license end dates are found before the active license expires, Tableau Desktop or Tableau Prep Builder will become unlicensed and stop working.

Users will need to activate a new or updated product key to start working with Tableau Desktop or Tableau Prep Builder again.

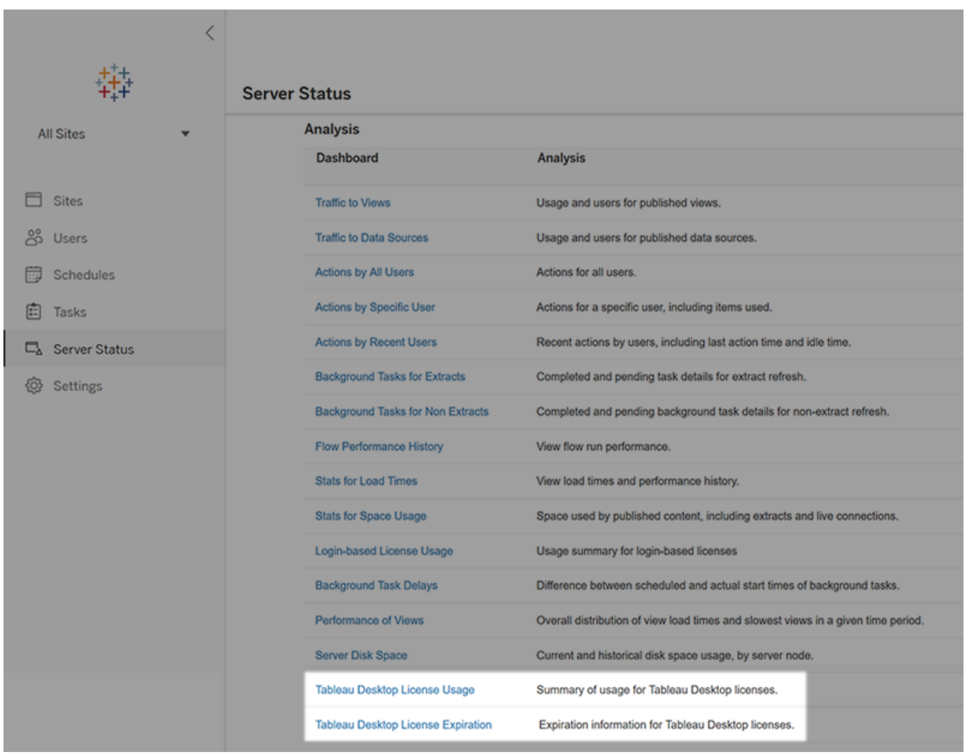
Track Tableau Desktop license usage and expiration data

If you want to track and view license usage and expiration data for Tableau Desktop in Tableau Server you must configure Tableau Desktop to send license data to Tableau Server on a set interval, and then enable reporting on Tableau Server.

This enables server administrators to access two reports:

- **Desktop License Usage**: This report lets server administrators see usage data for Tableau Desktop licenses in your organization.
- **Desktop License Expiration**: This report gives server administrators information about which Tableau Desktop licenses in your organization have expired or need maintenance renewal.

If Tableau Desktop and Tableau Server are configured for license reporting, when signed in to Tableau Server as an Administrator, you will see these two reports listed on the **Server Status** page in the **Analysis** section.



If you don't see these reports listed, then Tableau Desktop and Tableau Server may not be configured for Tableau Desktop usage reporting.

For information about how to configure Tableau Desktop and Tableau Server for usage reporting, see Manage Tableau Desktop License Usage in the Tableau Desktop and Tableau Prep Deployment guide.

Additional resources

For more information about managing your license refer to the following topics:

- To find your product key and activate Tableau Desktop or Tableau Prep Builder, see Where's my product key.
- To deactivate a product key or move it to another computer, see Move or Deactivate Tableau Desktop.
- To learn more about product keys for non-persistent virtual desktops or for computers that are regularly re-imaged, see Configure Virtual Desktop Support.
- To learn more about product key management for Tableau Server or Tableau Online, see Licensing Overview (Linux | Windows)
- To learn more about the license renewal process or to renew a license, see How to Renew your Tableau Licenses.

Keyboard Shortcuts

This is a list of keyboard shortcuts for working in Tableau. For information on navigating a view using a keyboard, see **Keyboard Accessibility for Tableau on the Web** on page 3164.

Shortcuts for managing workbooks, sheets, and files (Tableau Desktop)

Description	Windows shortcut	Mac shortcut
New workbook	Ctrl+N	Command+N
New worksheet	Ctrl+M	Command+T
Describe sheet	Ctrl+E	Command+E
Cycle forward through open worksheets	Ctrl+Tab, Ctrl+F6	Shift+Command+right bracket
Cycle backward through open worksheets	Ctrl+Shift+Tab, Ctrl+Shift+F6	Shift+Command+left bracket
Switch in and out of Presentation Mode	F7 , Ctrl+H	Option+Return
Switch in and out of Full Screen mode		Control+Command+F
Open file	Ctrl+O	Command+O
Save file	Ctrl+S	Command+S
Save file as	Ctrl+Shift+S	
Revert workbook to last saved state	F12	Option+Command+E

Close the current workbook	Alt+F4	Command+W
Print	Ctrl+P	Command+P
Open Help	F1	Control+Command+question mark

Shortcuts for managing workbooks, sheets, and files (Tableau Server, Tableau Online)

Description	**Windows shortcut**	**Mac shortcut**
New worksheet	Ctrl+Alt+T	Command+Alt+T
Switch in and out of Full Screen mode	F11	Control+Command+F
Save file	Ctrl+S	Command+S
Save workbook as	Ctrl+Shift+S	Shift+Command+S
Close the current workbook	Alt+F4	Alt+Q
Print (or export as PDF)	Ctrl+P	Command+P
Open Help	F1	Control+Command+question mark

Shortcuts for data (Tableau Desktop)

Description	**Windows shortcut**	**Mac shortcut**
Connect to data source	Ctrl+D	Command+D
Activate the find command in the Data pane	Ctrl+F	Command+F

Refresh data source	F5	Command+R
Run data updates on a view	F9	Shift+Command+0
Toggle automatic data updates on and off	F10	Option+Command+0

Shortcuts for data (Tableau Server, Tableau Online)

Description	Windows shortcut	Mac shortcut
Connect to data source	Ctrl+D	Command+D
Refresh data source	Alt+F5	Option+F5
Rename data source	Alt+F2	Option+F2
New data source	Ctrl+Alt+D	Control+D
Edit data source	Ctrl+Alt+Shift+D	Control+Shift+D

Shortcuts for authoring views (Tableau Desktop)

Description	Windows shortcut	Mac shortcut
Show Me!	Ctrl+1 , Ctrl+Shift+1	Command+1
Add the selected field to the sheet. (Only works with a single field.)	Enter or double-click	Return or double-click
Place selected field on Columns shelf	Alt+Shift+C	Option+Shift+C
Place selected field on	Alt+Shift+F	Option+Shift+F

Filters shelf		
Place selected field on Size	Alt+Shift+I	Option+Shift+I
Place selected field on Detail	Alt+Shift+L	Option+Shift+L
Place selected field on Color	Alt+Shift+O	Option+Shift+O
Place selected field on Pages shelf	Alt+Shift+P	Option+Shift+P
Place selected field on Rows shelf	Alt+Shift+R	Option+Shift+R
Place selected field on Shape	Alt+Shift+S	Option+Shift+S
Place selected field on Text/Label	Alt+Shift+T	Option+Shift+T
Place selected field on Rows shelf	Alt+Shift+X	Option+Shift+X
Place selected field on Columns shelf	Alt+Shift+Y	Option+Shift+Y
Open the Drop Field menu	Right-click+drag to shelf	Option+drag to shelf
Copy a field in the view and place it on another shelf or card	Ctrl+drag	Command+drag
Swap rows and columns	Ctrl+W	Control+Command+W
Flip orientation of column labels at bot-	Ctrl+L	

tom of view		
Toggle dashboard grid on and off	G	G
Toggle between Dashboard and Layout tabs	T	T
Cut text selection (in captions, titles, formulas, etc.)	Ctrl+X	Command+X
Paste clipboard	Ctrl+V	Command+V
Undo	Ctrl+Z	Command+Z
Redo	Ctrl+Y	Command+Shift+Z
Clear the current worksheet	Alt+Shift+Backspace	Option+Shift+Delete
Move floating dashboard object	Arrow key moves 1 pixel, Shift+arrow moves 10 pixels	Arrow key moves 1 pixel, Shift+arrow moves 10 pixels
Resize floating dashboard object	Alt+arrow key resizes in 1-pixel increments, Shift+Alt+arrow resizes in 10-pixel increments	Option+arrow key resizes in 1-pixel increments, Shift+Alt+arrow resizes in 10-pixel increments

Shortcuts for authoring views (Tableau Server, Tableau Online)

Description	**Windows shortcut**	**Mac shortcut**
Add the selected field to the sheet. (Only works with a single field.)	Double-click	Double-click
Copy a field in the view and place it on another	Ctrl+drag	Command+drag

shelf or card		
Toggle dashboard grid on and off	G	G
Toggle between Dashboard and Layout tabs	T	T
Cut text selection (in captions, titles, formulas, etc.)	Ctrl+X	Command+X
Paste clipboard	Ctrl+V	Command+V
Undo	Ctrl+Z	Command+Z
Redo	Ctrl+Y	Command+Shift+Z
Open current workbook in Tableau Desktop	Ctrl+O	Command+alt+O
Clear the current worksheet	Alt+Shift+Backspace	Option+Shift+Delete
Move floating dashboard object	Arrow key moves 1 pixel, Shift+arrow moves 10 pixels	Arrow key moves 1 pixel, Shift+arrow moves 10 pixels
Resize floating dashboard object	Alt+arrow key resizes in 1-pixel increments, Shift+Alt+arrow resizes in 10-pixel increments	Option+arrow key resizes in 1-pixel increments, Shift+Alt+arrow resizes in 10-pixel increments
Rename column field (when field is selected in the data schema pane)	F2	F2
Rename calculated field (when calculated field editor is open)	F2	F2

Shortcuts for resizing rows and columns (Tableau Desktop)

Smaller cell size	Ctrl+B	Command+B
Bigger cell size	Ctrl+Shift+B	Command+Shift+B
Make rows narrower	Ctrl+left arrow	Control+Command+left arrow
Make rows wider	Ctrl+right arrow	Control+Command+right arrow
Make columns shorter	Ctrl+down arrow	Control+Command+down arrow
Make columns taller	Ctrl+up arrow	Control+Command+up arrow

Shortcuts for pages (Tableau Desktop)

Description	**Windows short-cut**	**Mac shortcut**
Start or stop forward playback on the Pages shelf	F4	F4
Start or stop backward playback on the Pages shelf	Shift+F4	Shift+F4
Skip forward one page	Ctrl+period	Command+period
Skip backward one page	Ctrl+comma	Command+comma

Shortcuts for selecting and navigating marks

Description	**Windows shortcut**	**Mac shortcut**
Select a mark	Click	Click
Select a group of marks	Drag	Drag

Add individual marks to the selection	Ctrl+click	Command+click
Add a group of marks to the selection	Ctrl+drag	Command+drag
Use Rectangular Selection tool	A	A
Use Lasso Selection Tool	D	D
Use Radial Selection Tool	S	S
Select all data	Ctrl+A	Command+A
Copy selected data	Ctrl+C	Command+C
Clear the mark selection	Esc	Esc
Pan around the view	Shift+drag	Shift+drag
Zoom in to a point in the view (if not map, requires zoom mode)	Double-click , Ctrl+Shift+click	Double-click, Shift+Command+click
Zoom out from a point on a map (if not map, requires zoom mode)	Ctrl+Shift+Alt+click	Shift+Option+Command+click
Zoom out	Shift+Double-click	Shift+Double+click
Zoom in to an area in the view (requires zoom mode if not map)	Ctrl+Shift+drag	Shift+Command+drag
Zoom in and out on a map	Scroll	Scroll
Drag a row and scroll through a long list simultaneously	Click+drag to bottom of pane+hold	Click+scroll, Command+hold

Copyright