Leseprobe aus:
Luhmann, R für Einsteiger, 5. Auflage, ISBN 978-3-621-28790-6

http://www.beltz.de/de/nc/verlagsgruppe-beltz/gesamtprogramm.html?isbn=978-3-621-28790-6

Luhmann

R für Einsteiger

Maike Luhmann

R für Einsteiger

Einführung in die Statistik-Software für die Sozialwissenschaften

Mit Online-Material

5., überarbeitete Auflage

BELTZ

Prof. Dr. Maike Luhmann
Ruhr-Universität Bochum
Fakultät für Psychologie
Psychologische Methodenlehre
Universitätsstr. 150
Raum IB E4/185 - Postfach 17
44780 Bochum
E-Mail: maike.luhmann@rub.de

Dieses Buch ist erhältlich als:
ISBN 978-3-621-28790-6 Print
ISBN 978-3-621-28791-3 E-Book (PDF)

5., überarbeitete Auflage 2020

Lektorat: Salome Fabianek
Coverbild: IconicBestiary/iStock/Getty Images Plus; R-Logo © 2016 The R Foundation, unter CC-BY-SA 4.0 modifiziert
Herstellung: Lelia Rehm
Satz: Reproduktionsfähige Vorlagen der Autorin
Gesamtherstellung: Beltz Grafische Betriebe, Bad Langensalza
Printed in Germany

Weitere Informationen zu unseren Autor_innen und Titeln finden Sie unter: www.beltz.de

Inhaltsübersicht

Inhalt

Vorwort zur fünften Auflage

In den fünf Jahren seit Erscheinen der letzten Auflage dieses Buchs ist R zu einem der beliebtesten Statistik-Programme in Lehre und Forschung und auch außerhalb der Hochschulen geworden. Auch die Ressourcen für R haben stark zugenommen. Habe ich in der letzten Auflage noch auf wenige einschlägige Webseiten zu R verweisen können, ist das Angebot an kostenlos verfügbaren Video-Tutorials, eBooks, Webinaren und Kurzanleitungen mittlerweile fast unüberschaubar.

Wozu dann noch ein klassisches Lehrbuch zu R? Ich gebe mittlerweile seit vielen Jahren Kurse zur Einführung in R. In diesen Kursen wird immer wieder deutlich, dass das große Angebot an Ressourcen für Einsteigerinnen und Einsteiger häufig überfordernd ist. Ziel dieses Buchs ist es daher, den Einstieg in R zu strukturieren und damit den Leserinnen und Lesern ein Grundverständnis von R zu vermitteln, mit dem sie anschließend die zahlreichen großartigen Online-Angebote effizient nutzen können. In diesem Buch werden die statistischen Verfahren behandelt, die in den Sozialwissenschaften am häufigsten gelehrt und angewandt werden. Für die fünfte Auflage wurden alle Kapitel aktualisiert und teilweise komplett neu geschrieben (z. B. die Kapitel zu Datenaufbereitung und Datenvisualisierung). Zudem sind einige kleinere Kapitel dazugekommen, z. B. zu Workflow und Troubleshooting.

Eine solche umfassende Überarbeitung des Buchs war nur möglich durch tatkräftige Unterstützung. Ich möchte mich besonders bei Peter Hähner, Flavio Schröder, Marie von Rogal und Merle Widlok bedanken, die intensiv die Texte und Befehle der vorherigen und der neuen Auflage geprüft und korrigiert haben. Ein großer Dank geht auch an die R Community auf Twitter, über die ich immer wieder neue Tipps und Updates zu R erhalte und die meine Fragen zu R schnell und hilfreich beantwortet haben. Zudem bedanke ich mich herzlich bei meinem Team, meinen Studierenden, Workshop-Teilnehmerinnen und -Teilnehmern, Kolleginnen und Kollegen und natürlich bei allen Leserinnen und Lesern, deren Fehlermeldungen, konstruktive Verbesserungsvorschläge und positive Rückmeldungen diese neue Auflage möglich gemacht haben.

Für alle verbleibenden Unklarheiten und Fehler bin ich natürlich allein verantwortlich. Bitte kontaktieren Sie mich, wenn Sie Fehler entdecken sollten oder Vorschläge zur Verbesserung des Buchs machen möchten.

Köln, im Sommer 2020 Maike Luhmann

1 Einleitung

In den letzten Jahren hat die Statistik-Software R (R Core Team, 2019b) deutlich an Beliebtheit gewonnen. Nicht nur an den meisten Universitäten wird R mittlerweile in Lehre und Forschung eingesetzt, sondern auch in vielen Unternehmen (Bhalla, 2016). Die Vorteile von R sind schon lange bekannt – warum also jetzt dieser rasante Aufstieg? Dafür gibt es mindestens zwei Gründe.

Zum einen leben wir in einem Daten-Zeitalter, d. h., Daten werden kontinuierlich und von in großen Mengen gesammelt. Die Nutzung dieser Daten nimmt sowohl in der Forschung als auch in der freien Wirtschaft einen immer größeren Stellenwert ein, und unter dem Stichwort »Data Science« hat sich mittlerweile eine ganze Berufsgruppe etabliert, deren Spezialität das Aufbereiten und Auswerten großer Datenmengen ist. Eines der in diesem Feld am häufigsten eingesetzten Statistik-Programme ist R (Dataflair Team, 2019).

Zum anderen findet in der sozialwissenschaftlichen Forschung, besonders in der Psychologie, derzeit ein Umbruch statt. Immer mehr Forscherinnen und Forscher, wissenschaftliche Institutionen und Fachzeitschriften verschreiben sich den Open-Science-Prinzipien (z. B. Schönbrodt et al., 2017). Zu Open Science gehört unter anderem, dass man Daten und Auswertungsskripte zur Verfügung stellt, damit andere die veröffentlichten Ergebnisse reproduzieren können. Damit wirklich alle Menschen, egal woher sie kommen oder wie viel Geld sie haben, diese Dateien nutzen können, müssen diese in nicht-proprietärer Software erstellt worden sein – also in einer Software, die nicht kommerziell vetrieben wird. Ehemals beliebte kommerzielle Statistik-Programme wie SPSS oder Stata erfüllen dieses Kriterium nicht. R schon.

Es gibt also eine Menge Gründe, R zu lernen. Allerdings ist R nicht die intuitivste Software. Das Ziel dieses Buchs ist es, Ihnen den Einstieg in R zu erleichtern. Sie werden sehen: Wenn Sie die ersten Kapitel gemeistert haben, wird es sehr viel einfacher. Ich wünsche Ihnen beim Einstieg in R viel Erfolg und hoffentlich auch etwas Spaß!

1.1 Warum R?

Es gibt viele Gründe, die für R sprechen. Hier sind die wichtigsten:

- **R kann mehr**. Die Software enthält viele Funktionen, die man bei kommerziellen Statistik-Programmen vergeblich sucht.
- **R ist aktuell**. Das Programm wird ständig weiterentwickelt, sodass viele neue statistische Methoden zuerst in R und erst Jahre später in anderen Programmen implementiert sind.
- **R ist ansprechbar**. Die Programmierer der einzelnen Pakete sind über E-Mail erreichbar und reagieren meiner Erfahrung nach innerhalb weniger Tage. So können Fragen schnell geklärt und eventuelle Programmierfehler behoben werden.
- **R schläft nicht**. R-Nutzer gibt es auf der ganzen Welt, und viele von ihnen sind in den einschlägigen Foren und in den sozialen Medien aktiv. So kann man seine Fragen rund um die Uhr klären.
- **R ist nachgefragt**. Wenn Sie R-Kenntnisse nachweisen können, stehen Ihnen viele Jobmöglichkeiten offen – in und außerhalb der Wissenschaft.
- **R ist kostenlos!**

1.2 Für wen ist dieses Buch?

Dieses Buch richtet sich sowohl an Einsteiger, die zum ersten Mal mit einer Statistik-Software arbeiten, als auch an Umsteiger, die R als eine Alternative zu anderen Statistik-Programmen ausprobieren möchten. Besondere Computer- oder gar Programmierkenntnisse brauchen Sie nicht. Da dieses Buch aber keine Statistik-Einführung ist, sollten Sie die Grundlagen der hier behandelten statistischen Tests kennen. Dafür empfehle ich das Lehrbuch von Eid et al. (2017), an dem auch die Gliederung dieses Buchs orientiert ist.

Ich bin selbst Psychologin und habe daher Datenbeispiele aus psychologischen Studien gewählt. Darüber hinaus behandle ich einige Funktionen und statistische Verfahren, die besonders für Psychologinnen und Psychologen von Interesse sind, z. B. die Berechnung von Skalenwerten (Kap. 7) oder Verfahren für die Testkonstruktion wie Itemanalysen und Faktorenanalysen (Kap. 18). Grundsätzlich richtet sich das Buch jedoch an Interessierte aus allen Fachrichtungen der Sozialwissenschaften.

In einem Einführungsbuch ist natürlich kein Platz, alle speziellen statistischen Verfahren detailliert zu besprechen. Sie werden jedoch bald in der Lage sein, sich in die Funktionen und Pakete, die Sie für solche Verfahren brauchen, selbst einzuarbeiten. Weitere Hinweise dazu finden Sie im Anhang B: Pakete.

1.3 Wie benutzt man dieses Buch?

Den Umgang mit einer Statistik-Software lernt man am besten durch Ausprobieren. Daher empfehle ich, bei der Lektüre dieses Buchs immer einen Computer mit R dabeizuhaben. So können Sie die beschriebenen Funktionen direkt ausprobieren. Die Datensätze dafür können Sie von der Webseite zu diesem Buch herunterladen: **http://www.beltz.de/r-fuer-einsteiger**. Dort finden Sie auch die Skripte für jedes Kapitel, die oftmals noch weitere Beispiele enthalten, sowie zahlreiche Zusatzmaterialien.

Hinweise für Einsteiger
Ich habe mich bemüht, dieses Buch so zu schreiben, dass auch Einsteiger ohne jegliche Erfahrung mit Statistik-Software mit R zurechtkommen können. In diesem Buch stelle ich die statistischen Verfahren in der Reihenfolge vor, in der sie typischerweise gelehrt werden. Wenn Sie dieses Buch also in Begleitung zu einer Statistik-Vorlesung in Psychologie oder anderen sozialwissenschaftlichen Fächern lesen, können Sie die Kapitel einfach von vorne nach hinten durcharbeiten.

Hinweise für Umsteiger
Als Umsteiger haben Sie bereits Erfahrung mit anderen Statistik-Programmen und interessieren sich möglicherweise nur für bestimmte Funktionen in R. In diesem Fall brauchen Sie natürlich nicht das Buch von vorne bis hinten durchzuarbeiten, sondern können sich auf die Funktionen konzentrieren, die Sie benötigen. Beginnen Sie aber auf jeden Fall mit den Kapiteln 2 bis 5, in denen die Grundlagen von R vermittelt werden.

1.4 Weiterentwicklungen und Aktualität des Buchs

R wird ständig weiterentwickelt. Sowohl die Basisversion als auch die zusätzlichen Pakete werden regelmäßig aktualisiert, und fast wöchentlich werden neue Pakete zur Verfügung gestellt. Wenn Sie regelmäßig mit R arbeiten, werden Sie vermutlich immer wieder neue Pakete entdecken, die Ihnen die statistischen Analysen noch leichter machen. Für die Erstellung dieses Buchs habe ich mich bemüht, immer die aktuellsten Versionen zu verwenden (Stand April 2020). Dieses Buch verwendet die R Version 3.6.2. Die hier vorgestellten Funktionen und Pakete sollten jedoch auch mit neueren Versionen funktionieren. Wenn Sie trotzdem veraltete Funktionen oder gar Fehler finden, lassen Sie mich dies bitte wissen. Aktualisierungen und Fehlerkorrekturen werden regelmäßig unter **http://www.beltz.de/r-fuer-einsteiger** und unter

dem Twitter-Account **@RfuerEinsteiger** veröffentlicht. Und noch ein Tipp: Die aktuellsten Hinweise zu den Funktionen und den entsprechenden Argumenten finden Sie in der dazugehörigen Hilfe-Datei (s. Abschn. 4.3.3).

1.5 Verwendete Schriftarten

Um die Übersichtlichkeit des Texts zu erhöhen, werden in diesem Buch verschiedene Schriftarten verwendet (s. Tab. 1.1).

Tabelle 1.1 Verwendung von Schriftarten im Buch

Text	Beispiel
Menübefehle	DATEI → SPEICHERN
Namen von Paketen	`psych`-Paket
Namen von Funktionen	`mean`-Funktion
Befehle und Ausgaben	`> c(1, 2, 3)`

3 Ein erster Überblick

Wenn wir R und RStudio erfolgreich installiert haben, können wir loslegen. Wir öffnen immer direkt RStudio. R muss und sollte nicht zusätzlich gestartet werden, sondern läuft automatisch im Hintergrund mit. In diesem Kapitel verschaffen wir uns einen ersten Überblick über das Programm (Abschn. 3.1) und besprechen, wie man zusätzliche Pakete installieren und laden kann (Abschn. 3.2). Wenn man mal nicht weiter weiß, kann man sich an verschiedenen Stellen Hilfe zu R holen (Abschn. 3.3).

3.1 RStudio im Überblick

Die Benutzeroberfläche von RStudio setzt sich üblicherweise aus vier nebeneinander angeordneten Fenstern zusammen (Abb. 3.1): das Skript-Fenster oben links, die Konsole unten links, das Environment-Fenster oben rechts, und ein Fenster mit einer ganzen Reihe von Reitern unten rechts. Bei der ersten Nutzung von RStudio wird das Skript-Fenster nicht immer sofort angezeigt. Wie man es öffnet, wird in Abschn. 3.1.2 beschrieben. In den folgenden Abschnitten gehen wir die Fenster einzeln durch.

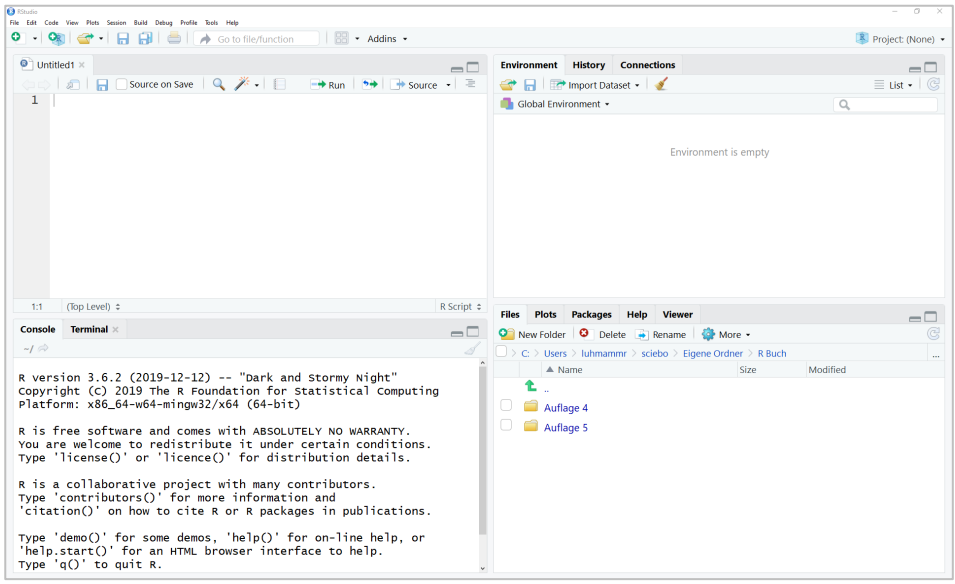

Abbildung 3.1 RStudio

3.1.1 Die Konsole

Die Konsole bietet eigentlich alles, was wir für die Arbeit mit R brauchen. Hier können wir sowohl Befehle eingeben als auch Ausgaben betrachten. Wenn wir RStudio öffnen, steht in der Konsole ein kurzer Begrüßungstext (s. Abb. 3.1), der u. a. angibt, welche R Version gerade geladen ist (hier: Version 3.6.2, veröffentlicht am 12.12.2019 unter dem Spitznamen »Dark and Stormy Night«). Unter dem Begrüßungstext erscheint das folgende Symbol:

```
>
```

Bei diesem Symbol handelt es sich um ein Eingabesymbol, d. h., dahinter können wir einen Befehl eingeben. In der R-Sprache wird dieses Symbol als »Prompt« (deutsch: Eingabeaufforderung) bezeichnet. Um einen Befehl in der Konsole auszuführen, drücken wir die Enter-Taste. Die Ausgabe erscheint dann direkt unter dem Befehl in einer anderen Farbe. Wenn alles gut gegangen ist, erwartet uns darunter schon wieder das nächste Eingabesymbol. Wir probieren das mal mit dem folgenden Befehl aus:

```
> 1+2
[1] 3
>
```

In diesem Beispiel haben wir eine kleine Rechenaufgabe lösen lassen. R kann man nämlich auch als Taschenrechner verwenden (mehr dazu in Abschn. 4.1). Das Ergebnis erscheint in der Zeile unter dem Befehl. Diesen Teil bezeichnet man auch als Ausgabe oder Output.

Die Ausgabe beginnt mit `[1]`. Das eigentliche Ergebnis steht dahinter. Das Zeichen `[1]` hat nichts mit dem Ergebnis der Rechenaufgabe zu tun, sondern wird verwendet, um bei langen Ausgaben, die über mehrere Zeilen gehen, die Elemente zu nummerieren. Das klingt jetzt noch sehr abstrakt, wird aber klarer, wenn wir später Beispiele für solche Ausgaben betrachten (s. Abschn. 5.4).

Unter der Ausgabe erscheint erneut das Eingabesymbol. Das ist ein gutes Zeichen, denn es bedeutet, dass der Befehl vollständig ausgeführt wurde und die Konsole jetzt bereit für den nächsten Befehl ist. Wurde der Befehl dagegen nicht vollständig ausgeführt (z. B. weil eine Klammer nicht geschlossen wurde), fügt die Konsole automatisch eine zweite Zeile ein, die mit einem + beginnt. Das + zeigt an, dass diese Zeile mit der vorhergehenden Zeile verknüpft ist.

Im Folgenden sind zwei Befehle dargestellt. Der erste Befehl geht über eine Zeile, daher folgt die Ausgabe direkt in der zweiten Zeile. In den zweiten Befehl wurde dagegen ein Zeilenumbruch eingefügt, sodass dieser Befehl über zwei Zeilen geht. Das

+ am Anfang der zweiten Zeile ist hier kein mathematisches Symbol, sondern es verknüpft lediglich die beiden Befehlszeilen miteinander. Die Ausgabe folgt in der dritten Zeile:

```
> (3+7+2)/3
[1] 4

> (3+7+2)/
+ 3
[1] 4
```

Tipp

Bevor man einen neuen Befehl ausführt, sollte man immer sicherstellen, dass die unterste Zeile in der Konsole mit > beginnt. Beginnt sie stattdessen mit + , so ist dies ein Zeichen, dass der vorangegangene Befehl unvollständig eingegeben und daher noch nicht beendet wurde. Das passiert z. B., wenn man vergisst, Klammern oder Anführungszeichen zu schließen. Diese und weitere typische Fehlerquellen werden in Kapitel 24 behandelt.

Konsole aufräumen

Nach einer längeren R-Sitzung ist die Konsole meistens ziemlich voll und unübersichtlich. Wenn uns das stört, können wir die Konsole mit der Tastenkombination STRG + L (Windows) bzw. CMD + L (Mac) aufräumen. Damit werden alle Inhalte der Konsole gelöscht.

Inhalt der Konsole speichern

Die Konsole enthält alle Befehle und Ausgaben, die in einer Sitzung angefordert wurden. Und damit ist wirklich alles gemeint – also auch alle Befehle mit Tippfehlern, mehrfach ausgeführte Befehle und Fehlermeldungen. Daher wird der Inhalt der Konsole üblicherweise nicht gespeichert. Stattdessen kann man bestimmte Ausgaben (z. B. Tabellen oder Diagramme) gezielt in externe Dateien zu exportieren oder eine sog. R Markdown-Datei anlegen (s. Kap. 22). Wenn man trotzdem den Inhalt der Konsole speichern möchte, kann man diesen markieren, kopieren und in ein einfaches Textdokument einfügen.

3.1.2 Das Skript-Fenster

Zwar könnte man theoretisch alle Befehle direkt in die Konsole eingeben, wie im vorherigen Abschnitt gezeigt, aber wir lassen das lieber bleiben. Der Nachteil an der Konsole ist nämlich, dass die Befehle immer wieder neu eingegeben werden müssen und nicht gut gespeichert werden können. Deshalb geben wir Befehle ab jetzt immer in das Skript-Fenster ein. Falls dieses Fenster noch nicht zu sehen ist, können wir es über FILE → NEW FILE → R SCRIPT öffnen.

Befehle im Skript-Fenster eingeben und ausführen

Im Skript-Fenster erscheint kein Eingabesymbol, sondern der Befehl wird direkt eingegeben. Ein Befehl kann über eine oder mehrere Zeilen gehen. Bei langen Befehlen mit vielen Argumenten (s. Abschn. 4.3) ist es übersichtlicher, wenn man den Befehl durch Zeilenumbrüche auf verschiedene Zeilen aufteilt. Dazu setzen wir den Cursor an die Stelle, wo der Zeilenumbruch eingefügt werden soll, und drücken die ENTER-Taste. RStudio rückt alle folgenden Zeilen etwas ein, sodass man schnell erkennen kann, wo ein Befehl anfängt und aufhört. So wird die Rechenaufgabe aus dem vorherigen Abschnitt im Skript-Fenster dargestellt, wenn wieder ein Zeilenumbruch eingefügt wird:

```
(3+7+2) /
  3
```

Wie führt man nun diesen Befehl aus? Anders als in der Konsole reicht hier die ENTER-Taste nicht aus, denn damit würde nur ein Zeilenumbruch eingefügt werden. Wir können stattdessen auf ein Symbol klicken oder eine spezielle Tastenkombination verwenden.

Zunächst müssen wir aber festlegen, welche Befehle genau ausgeführt werden sollen. Dazu gibt es mehrere Möglichkeiten: (1) Wir setzen den Cursor irgendwo in den Befehl, der ausgeführt werden soll. Es wird dann genau dieser Befehl von Anfang bis Ende ausgeführt. (2) Wir markieren den Befehl mit der Maus. Dann wird alles ausgeführt, was markiert ist. Wir können damit mehrere Befehle gleichzeitig ausführen. Aber Vorsicht: Hier wird wirklich nur der markierte Teil ausgeführt – haben wir z. B. vergessen, eine Klammer mitzumarkieren, dann wird der Befehl unvollständig ausgeführt. Diese Eigenschaft klingt zunächst etwas unpraktisch, hat aber einen wichtigen Vorteil: Wir können dadurch gezielt Teile von Befehlen ausführen, was besonders nützlich ist, wenn wir es später mit langen, geschachtelten Befehlen zu tun haben. Zudem kann man diese Eigenschaft für die Fehlersuche nutzen (s. Kap. 24).

Um den Befehl auszuführen, klicken wir entweder auf das Symbol RUN auf der rechten Seite der oberen Leiste des Skript-Fensters oder verwenden die Tastenkombination STRG + ENTER (Windows) bzw. CMD + ENTER (Mac). Dadurch erscheint die Ausgabe zusammen mit dem Befehl in der Konsole. Der Text im Skript-Fenster bleibt unverändert. Diese Trennung zwischen Eingaben und Ausgaben hat u. a. den Vorteil, dass man alte Befehle jederzeit wieder aufrufen und erneut ausführen kann, sogar mit anderen Datensätzen.

Darstellung von Befehlen in RStudio

Eben haben wir schon gesehen, dass RStudio nachfolgende Zeilen bei mehrzeiligen Befehlen einrückt. Auch bei anderen Dingen ist RStudio sehr bemüht, die Befehle so übersichtlich wie möglich darzustellen.

Ein Beispiel dafür ist die Verwendung von verschiedenen Farben. So werden Funktionen (mehr dazu in Abschn. 4.3), Texteingaben (also alles, was von Anführungszeichen umgeben ist) und sonstige Bestandteile von Befehlen in unterschiedlichen Farben dargestellt. Welche das sind, kann man selbst beeinflussen: Unter TOOLS → GLOBAL OPTIONS → APPEARANCE kann man u. a. die farbliche Darstellung der Befehle und die Hintergrundfarbe verändern.

Darstellung von Befehlen in diesem Buch

In diesem Lehrbuch stellen wir Befehle immer so dar, wie sie in der Konsole erscheinen. Das heißt, die Befehle beginnen immer mit dem Eingabesymbol, um sie besser von Ausgaben zu unterscheiden. Wenn man diese Befehle im Skript-Fenster eingibt, muss das Eingabesymbol weggelassen werden (s. Tab. 3.1).

Tabelle 3.1 Darstellung von Befehlen im Lehrbuch und Eingabe dieser Befehle im Skript-Fenster

Darstellung im Lehrbuch	Eingabe im Skript-Fenster
`> 3+4`	`3+4`
`> mean(x)`	`mean(x)`
`> vektor <- c(1, 2, 3)`	`vektor <- c(1, 2, 3)`

`> plot(y ~ x,`
`pch = 2,`
`las = 1)`
	`plot(y ~ x,`
	`pch = 2,`
	`las = 1)`

Skript speichern

Die Befehle, die wir in das Skript-Fenster schreiben, können wir als Datei speichern. Diese Datei wird Skript genannt und erhält die Dateiendung .R. Um das Skript zu

speichern, wählen wir im Menü die Option FILE → SAVE AS oder verwenden die Tastenkombination STRG + S (Windows) bzw. CMD + S (Mac).

Beim erstmaligen Speichern kann es sein, dass sich ein Fenster öffnet, in dem wir die Encodierung der Skript-Datei festlegen sollen (s. Abb. 3.2). Hier wird festgelegt, wie Sonderzeichen wie ä, ö, ü oder ß gespeichert werden. Als erste Option wird immer die Encodierung angezeigt, die auf dem aktuellen Computer als Standard eingestellt ist. Wenn man diese Encodierung wählt, kann man das Skript an allen Computern, die diese Encodierung als Standard eingestellt haben, problemlos öffnen. Öffnet man das Skript jedoch an einem Computer mit einer anderen Encodierung, werden Sonderzeichen kryptisch dargestellt, z. B. wird aus einem hübschen ä ein weniger hübsches Ã¤. Das passiert z. B. dann, wenn man eine auf Mac erstellte Skript-Datei auf einem Windows-PC öffnet, da Mac und Windows unterschiedliche Encodierungen verwenden. Wie man dieses Problem vermeidet und wie man mit einer anders encodierten Skript-Datei umgeht, wird in Kapitel 24 vertieft.

Warum sollte man überhaupt ein Skript anlegen und speichern? Der wichtigste Vorteil ist, dass man Befehle einfach modifizieren und wiederholt ausführen kann. Ein gutes Skript kann man vollständig markieren und in einem Rutsch ausführen, ohne dass man zwischendurch noch etwas per Hand machen muss. Wie ein gutes Skript aufgebaut und gestaltet ist, besprechen wir in Kapitel 23.

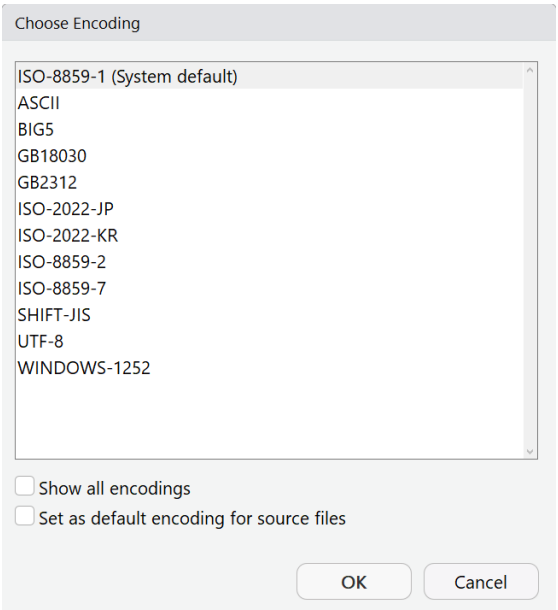

Abbildung 3.2 Auswahlfenster zur Wahl der Encodierung des Skripts

Skript öffnen
Von RStudio aus öffnen wir ein Skript über FILE → OPEN FILE. Das Skript wird dann in einem neuen Skript-Fenster angezeigt. Man kann aber eine Skript-Datei auch durch Anklicken im Explorer (Windows) bzw. im Finder (Mac) öffnen.

3.1.3 Das Environment-Fenster

Oben rechts befindet sich das Environment-Fenster. Hier werden später alle Objekte angezeigt, die wir in einer R-Sitzung erstellt oder geladen haben (s. Kap. 5). Außerdem erscheinen hier standardmäßig noch zwei weitere Reiter: HISTORY und CONNECTIONS. Im History-Fenster werden alle Befehle aufgelistet, die wir ausführen – genau in der Reihenfolge, wie wir sie aufgeführt haben, ohne Kommentare, ohne Ergebnisse und vor allem inklusive sämtlicher Tippfehler. Diese Auflistung ist selten interessant und noch seltener hilfreich, daher ignorieren wir sie in diesem Buch. Im Connections-Fenster können wir Verbindungen zu Datenquellen an anderen Orten (z. B. im Internet) aufbauen. Auch diese Funktion benötigen wir in diesem Buch nicht.

3.1.4 Files, Plots, Packages, Help, Viewer

Im Bereich unten rechts finden wir mehrere Fenster, die wir über verschiedene Reiter ansteuern können. Unter FILES befindet sich ein Bereich, in dem wir durch die Verzeichnisse auf unserem Computer navigieren und Dateien öffnen können. Wenn wir später Diagramme erstellen (s. Kap. 10), erscheinen diese im Fenster PLOTS. Der Bereich PACKAGES enthält eine Liste aller Pakete, die wir installiert haben (s. Abschn. 3.2). Wenn wir Hilfe-Dateien zu bestimmten R-Paketen oder Funktionen aufrufen (s. Abschn. 4.3.3), erscheinen diese im Fenster HELP. Der VIEWER ist vor allem relevant, wenn wir Dokumente in R Markdown erstellen (s. Abschn. 22.4).

3.1.5 RStudio anpassen

Unter TOOLS → GLOBAL OPTIONS kann man das Aussehen und einige Funktionalitäten von RStudio anpassen (s. Abb. 3.3). So können wir z. B. unter GENERAL ein bestimmtes Verzeichnis als Standard-Arbeitsverzeichnis (DEFAULT WORKING DIRECTORY) festlegen (s. Abschn. 5.5.1) oder unter CODE Einstellungen zur Darstellung und Ausführung von R-Befehlen verändern. Wir lassen erstmal alle Einstellungen so, wie sie sind. Wenn Sie aber bei der fortgeschrittenen Arbeit mit R merken, dass manche Standardeinstellungen nicht zu Ihrem Arbeitsstil passen, finden Sie in den GLOBAL OPTIONS häufig eine Lösung.

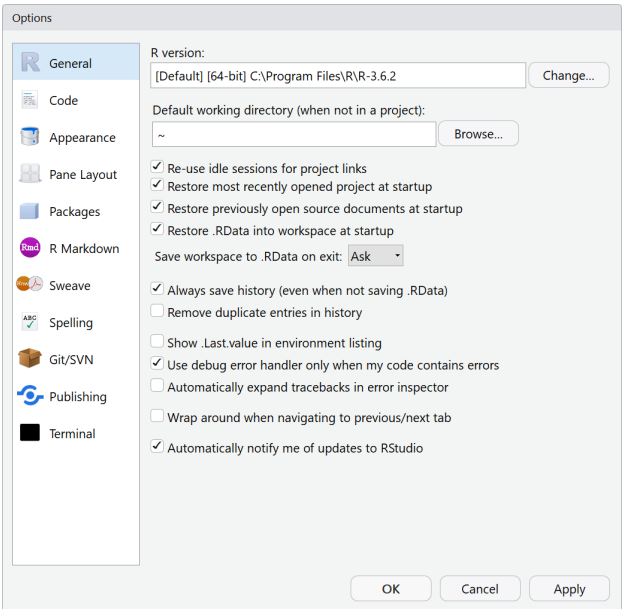

Abbildung 3.3 Global Options

3.1.6 RStudio schließen

Bevor wir RStudio schließen, sollten wir prüfen, ob alle Skripte als Skript-Dateien gespeichert sind (s. Abschn. 3.1.2). Wir schließen das Programm dann wie gewohnt, indem wir auf das Kreuz oben rechts klicken, im Menü die Option FILE → QUIT SESSION wählen, oder die Tastenkombination STRG + Q (Windows) bzw. CMD + Q (Mac) verwenden.

Dann passiert allerdings noch etwas, was für Einsteiger erstmal verwirrend ist: Wir werden gefragt, ob der Workspace bzw. das Global Environment gesichert werden soll. Den Workspace lernen wir in Kapitel 5 kennen. Er enthält sog. Objekte, z. B. Daten, mit denen wir in der R-Sitzung gearbeitet haben. Wenn wir auf JA klicken, wird der Inhalt des Workspaces in einer namenlosen Datei im Arbeitsverzeichnis gespeichert. Diese Datei wird dann jedesmal automatisch geöffnet, wenn wir R erneut starten. Dies ist fast nie wünschenswert, daher sollten wir hier immer die Option NEIN wählen. Bei den GLOBAL OPTIONS kann man außerdem einstellen, dass diese Meldung gar nicht mehr erscheint. Unter TOOLS → GLOBAL OPTIONS → GENERAL ändern wir die Einstellung bei SAVE WORKSPACE TO .RDATA ON EXIT zu NEVER.

3.2 Zusätzliche Pakete

R ist eine Open-Source-Software. Allgemein bedeutet das, dass der Quellcode für das Programm offen ist und von den Nutzerinnen und Nutzern geändert werden kann. Bei R ist das eingeschränkt der Fall: Die Basisversion, die wir mit der Installation von R erhalten, kann nicht von Ihnen oder von mir verändert werden. Das ist sicher sinnvoll, denn die Wahrscheinlichkeit, dass wir dabei etwas kaputtmachen, ist hoch (zumindest bei mir). Was tun also Menschen, die R verbessern oder erweitern möchten? Sie programmieren eigene Funktionen und stellen diese in sog. Paketen der Community zur Verfügung. Durch dieses Prinzip sind neu entwickelte statistische Methoden meistens als erstes in R verfügbar, Jahre bevor sie in kommerziellen Statistik-Programmen implementiert werden. Und die Anzahl der Pakete wächst exponentiell: Zum Zeitpunkt der vorherigen Auflage dieses Buchs (März 2015) standen ca. 6000 Pakete zur Verfügung, mittlerweile (Mai 2020) sind es bereits über 15000! Auf der Webseite https://cran.r-project.org sind alle verfügbaren Pakete aufgelistet und beschrieben.

Um Pakete zu nutzen, müssen zwei Schritte durchgeführt werden: Die Pakete müssen einmalig installiert werden (Abschn. 3.2.1) und die Pakete müssen in jeder Sitzung neu geladen werden (Abschn. 3.2.2).

3.2.1 Pakete installieren

In RStudio installieren wir Pakete, indem wir uns entweder im Menü durchklicken oder direkt einen Befehl eingeben. Wir beginnen mit der Klick-Variante, da diese für die meisten Einsteiger intuitiver ist. Fortgeschrittene bevorzugen aber meistens die Eingabe eines Befehls, da dies schneller geht.

Zur Veranschaulichung installieren wir jetzt das Paket `psych` (Revelle, 2019). Dieses Paket wurde vom Persönlichkeitspsychologen William Revelle programmiert und enthält eine Reihe nützlicher Funktionen, die wir häufig anwenden werden.

Menübefehl

In RStudio finden wir die Menüoption für das Installieren von Paketen unter TOOLS → INSTALL PACKAGES. Alternativ können wir auch im Fenster unten rechts den Reiter PACKAGES auswählen und auf INSTALL klicken.

In dem sich öffnenden Fenster (Abb. 3.4) geben wir die Namen der Pakete ein, die wir installieren möchten, bei uns also `psych`. Wenn wir mehrere Pakete auf einmal installieren möchten, können wir die Namen der Pakete einfach hintereinander eingeben, getrennt mit einem Leerzeichen oder einem Komma. Die anderen Einstellun-

gen in diesem Fenster beziehen sich darauf, von wo die Pakete heruntergeladen werden und wo sie auf dem Rechner installiert werden sollen. Ich empfehle, an diesen Einstellungen nichts zu ändern. Wir achten nun noch darauf, dass bei der Option INSTALL DEPENDENCIES ein Häkchen gesetzt ist. Was ist das? Viele Pakete nutzen Funktionen aus anderen Paketen und funktionieren daher nur dann reibungslos, wenn diese anderen Pakete ebenfalls direkt installiert werden. Wenn das Häkchen gesetzt ist, werden also alle Pakete heruntergeladen und installiert, von denen das uns eigentlich interessierende Paket abhängt.

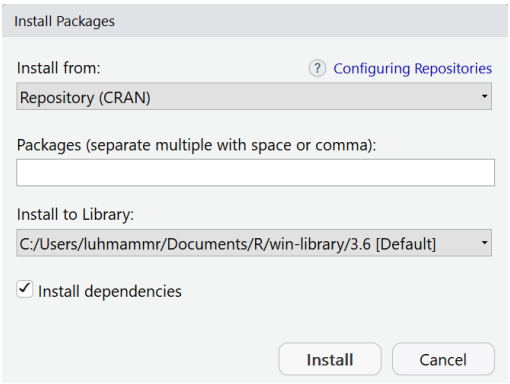

Abbildung 3.4 Eingabefenster für die Installation von Paketen

Wenn wir nun auf INSTALL klicken, wird das Paket heruntergeladen und installiert. In der Konsole erscheint dann die folgende (hier verkürzt dargestellte) Ausgabe:

```
> install.packages("psych", dependencies = TRUE)
Installing package into 'C:/Users/…/R/win-library/3.6'
(as 'lib' is unspecified)
also installing the dependencies 'abind', 'coda', 'arm',
'pbivnorm', 'numDeriv', 'matrixcalc', 'mi', 'minqa', 'nloptr',
'Rcpp', 'RcppEigen', 'psychTools', 'GPArotation', 'lavaan',
'sem', 'lme4', 'Rcsdp'

trying URL 'https://cran.rstudio.com/bin/windows/
contrib/3.6/psych_1.9.12.31.zip'
Content type 'application/zip' length 3802421 bytes (3.6 MB)
downloaded 3.6 MB
package 'psych' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
   C:\Users\...\downloaded_packages
```

Hier wurde also nicht nur das `psych`-Paket, sondern auch diverse andere Pakete installiert, von denen das `psych`-Paket abhängt. Woran erkenne ich, ob die Installation geklappt hat? Achten Sie auf die Meldung `package 'psych' successfully unpacked and MD5 sums checked`. Wenn die kommt, ist alles gut.

Funktion

Pakete können auch über die folgende Funktion installiert werden:

```
> install.packages("Name.des.Pakets", dependencies = TRUE)
```

Das Argument `dependencies` bezieht sich wiederum darauf, ob auch andere Pakete installiert werden sollen. Steht dahinter `TRUE`, werden alle Pakete, von denen unser Paket abhängt, direkt mitinstalliert. Der entsprechende Befehl für die Installation des `psych`-Pakets lautet:

```
> install.packages("psych", dependencies = TRUE)
```

Tipp

R unterscheidet zwischen Groß- und Kleinschreibung. Folgende Pakete gibt es nicht: `PSYCH`, `Psych`, `psYch` usw.

3.2.2 Pakete laden

Die zusätzlich installierten Pakete werden beim Öffnen von Rstudio nicht automatisch geladen. Das heißt, wir müssen ein Paket jedes Mal laden, wenn wir es in einer neuen Sitzung verwenden möchten. Es ist allerdings nicht notwendig, das Paket jedes Mal neu zu installieren. Woher wissen wir, ob ein Paket bereits installiert ist? In RStudio können wir im Fenster unten rechts unter dem Reiter PACKAGES eine Liste aller bereits installierten Pakete einsehen. Diese Liste erhalten wir auch mit dem folgenden Befehl:

```
> library()
```

Auch das Laden von Paketen kann wieder über das Menü oder über einen Befehl erfolgen. Wenn wir bald richtig mit R arbeiten, werden wir immer den Befehl nutzen (s. Kap. 23). Für den Einstieg schauen wir uns aber auch an, wie es über das Menü funktioniert.