Python에서 EMR데이터 (생존)분석 따라하기

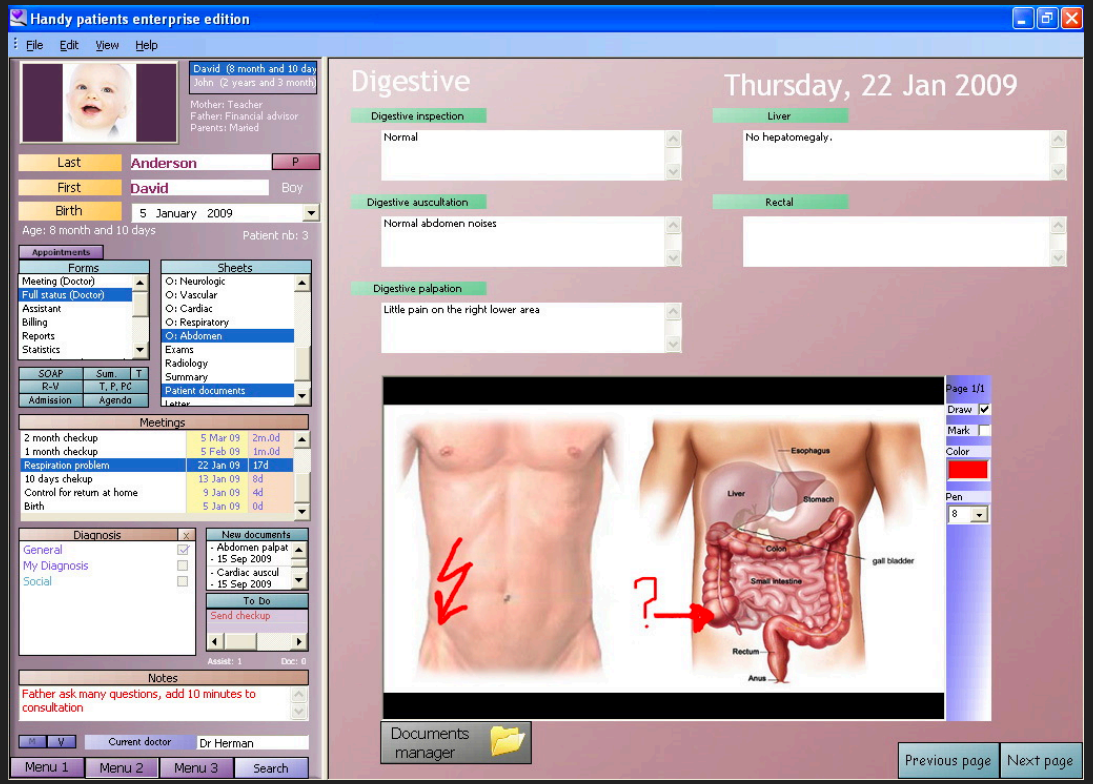
Soo-Heang Eo, Lead Data Scientist
HuToM

Python에서 EMR데이터 ~~(생존)~~분석 따라하기

Soo-Heang Eo, Lead Data Scientist
HuToM

Background

EMR (Electronic Medical Records) Data

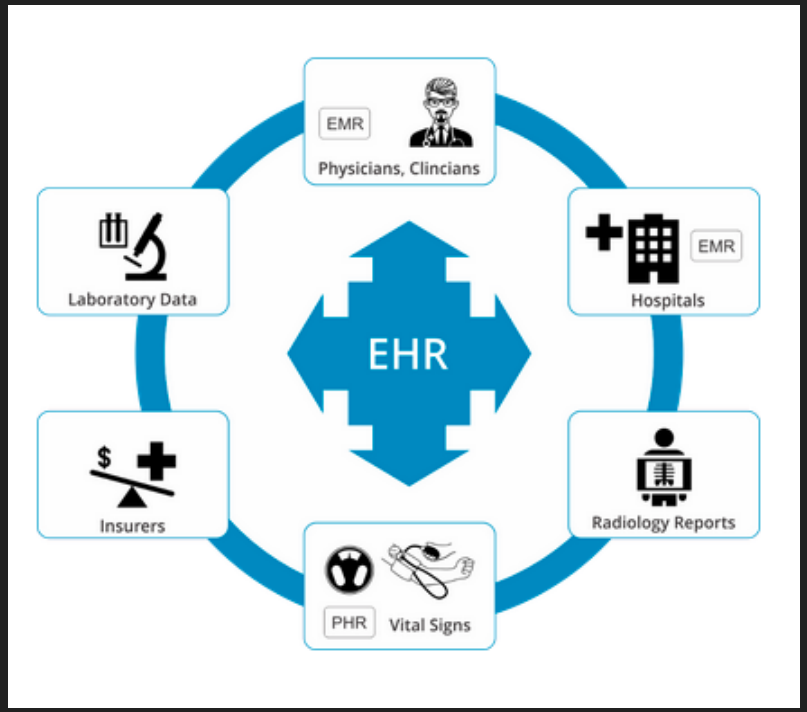


Systematized collection of patient and population electronically-stored health information in a digital format including

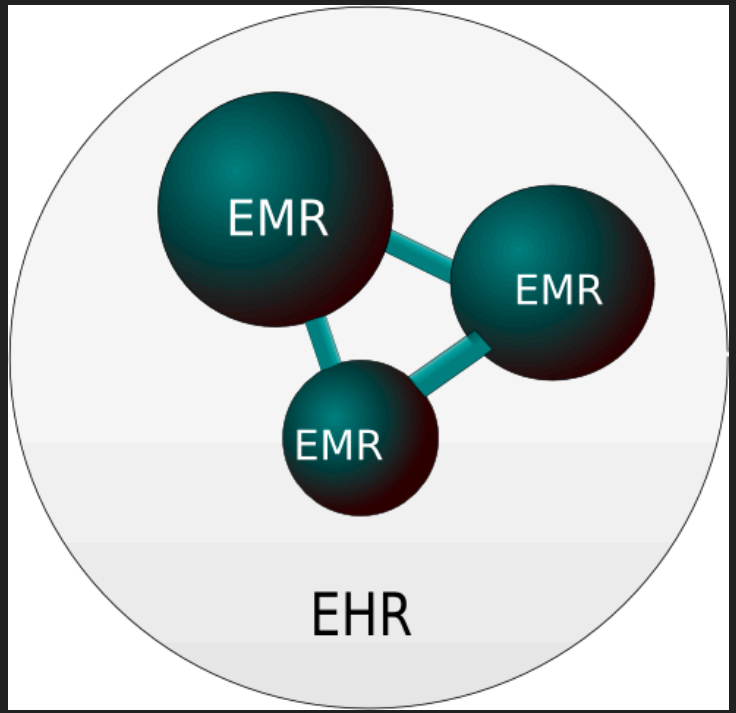
- Demographics
- medical history
- medication and allergies
- immunization status
- laboratory test results
- radiology images
- vital signs
- personal statistics like age and weight
- billing information

https://en.wikipedia.org/wiki/Electronic_health_record

EMR vs. EHR?



https://doctors.practo.com/emr-vs-ehr-whats-difference/



https://sooyongshin.wordpress.com/2017/05/14/healthcare-data-data-data-2-clinical-data-data-in-emr/

Admission Date: [**2952-11-3**] Discharge Date: [**2952-11-9**]

Date of Birth: [**2887-7-23**] Sex: F

Service: MEDICINE

Allergies:
No Known Allergies / Adverse Drug Reactions

Attending:[**First Name3 (LF) 3925**]
Chief Complaint:
Sepsis, respiratory distress

Major Surgical or Invasive Procedure:
None

History of Present Illness:
F w/ h/o metastatic breast cancer to breast and lungs
currently receiving CMT, brought to the ED by
rehab for abnormal labs. She was found to be neutropenic, anemia
and thrombocytopenic. At the rehab, vitals were reportedly T
100.4, HR 107, BP 92/42. There is also a concern for possible...

https://github.com/MIT-LCP/bhi-bsn-challenge/tree/master/slides

Publicly Available EMR Dataset

3. Data derived from Electronic Health Records (EHRs)

Building the graph of medicine from millions of clinical narratives
Co-occurence statistics for medical terms extracted from 14 million clinical notes and 260,000 patients.
Paper: http://www.nature.com/articles/sdata201432
Data: http://datadryad.org/resource/doi:10.5061/dryad.jp917

Learning Low-Dimensional Representations of Medical Concept
Low-dimensional embeddings of medical concepts constructed using claims data. Note that this paper utilizes data from *Building the graph of medicine from millions of clinical narratives*
Paper: http://cs.nyu.edu/~dsontag/papers/ChoiChiuSontag_AMIA_CRI16.pdf
Data: https://github.com/clinicalml/embeddings

MIMIC-III, a freely accessible critical care database
Anonymized critical care EHR database on 38,597 patients and 53,423 ICU admissions. **Requires registration**.
Paper: http://www.nature.com/articles/sdata201635
Data: http://physionet.org/physiobank/database/mimic3cdb/

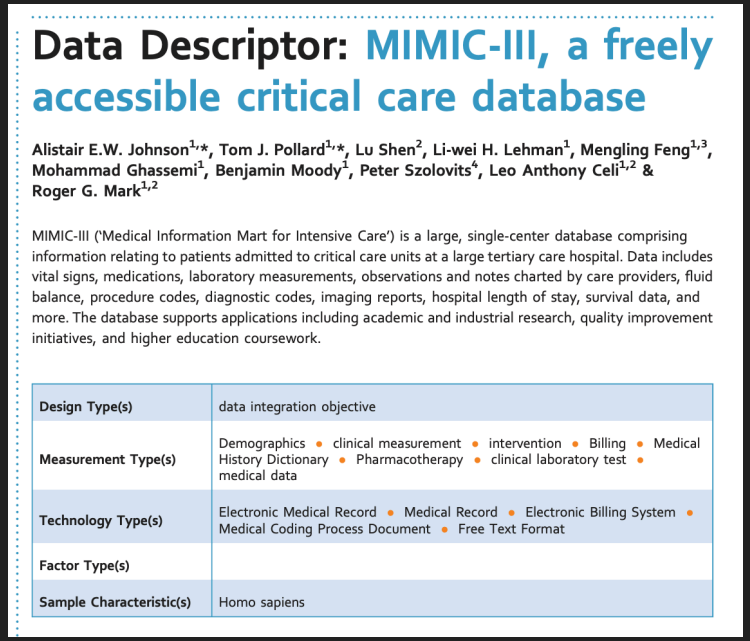
Clinical Concept Embeddings Learned from Massive Sources of Medical Data
Embeddings for 108,477 medical concepts learned from 60 million patients, 1.7 million journal articles, and clinical notes of 20 million patients
Paper: https://arxiv.org/abs/1804.01486
Embeddings: https://figshare.com/s/00d69861786cd0156d81
Interactive tool: http://cui2vec.dbmi.hms.harvard.edu

---

https://github.com/beamandrew/medical-data

Publicly Available EMR Dataset

MIMIC Critical Care Data



Data Descriptor: MIMIC-III, a freely accessible critical care database

Alistair E.W. Johnson[1,*], Tom J. Pollard[1,*], Lu Shen[2], Li-wei H. Lehman[1], Mengling Feng[1,3], Mohammad Ghassemi[1], Benjamin Moody[1], Peter Szolovits[4], Leo Anthony Celi[1,2] & Roger G. Mark[1,2]

MIMIC-III ('Medical Information Mart for Intensive Care') is a large, single-center database comprising information relating to patients admitted to critical care units at a large tertiary care hospital. Data includes vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and more. The database supports applications including academic and industrial research, quality improvement initiatives, and higher education coursework.

Design Type(s)	data integration objective
Measurement Type(s)	Demographics • clinical measurement • intervention • Billing • Medical History Dictionary • Pharmacotherapy • clinical laboratory test • medical data
Technology Type(s)	Electronic Medical Record • Medical Record • Electronic Billing System • Medical Coding Process Document • Free Text Format
Factor Type(s)	
Sample Characteristic(s)	Homo sapiens

https://www.nature.com/articles/sdata201635.pdf

I2B2 Clinical Notes (NLP) Data

i2b2

Informatics for Integrating Biology & the Bedside

About Us | Software | NLP Data Sets | i2b2 tranSMART Foundation |

NLP Research Data Sets

Data Sets

- Home
- *User*
  - Register
  - Login
- *Data Use Agreement*
  - Academic
  - Commercial

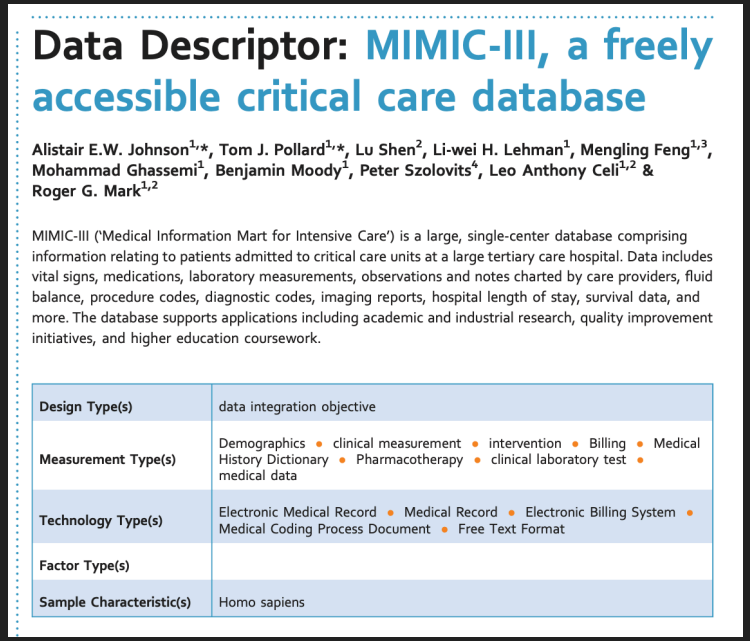
i2b2 is a passionate advocate for the potential of exist
directly impact healthcare improvement. In our many
become increasingly obvious that the value locked in
mission. In order to enhance the ability of natural lang
fine grained information from clinical records, i2b2 has
notes from the Research Patient Data Repository at P
organized by Dr. Ozlem Uzuner. We are pleased to n
for general research purposes. At this time we are rele
Challenges as i2b2 NLP Research Data Sets. A simila
Challenge will be released on the one year anniversa
enabled hundreds of journal and conference articles b

To access these notes, please use the Registration lir
your proposal and, if acceptable, will ask you to sign a
before releasing the notes to you. Given the goal of le
you share your annotations back to us following public
access, whichever comes first.

https://www.i2b2.org/NLP/DataSets/Main.php

Publicly Available EMR Dataset

MIMIC Critical Care Data



Data Descriptor: MIMIC-III, a freely accessible critical care database

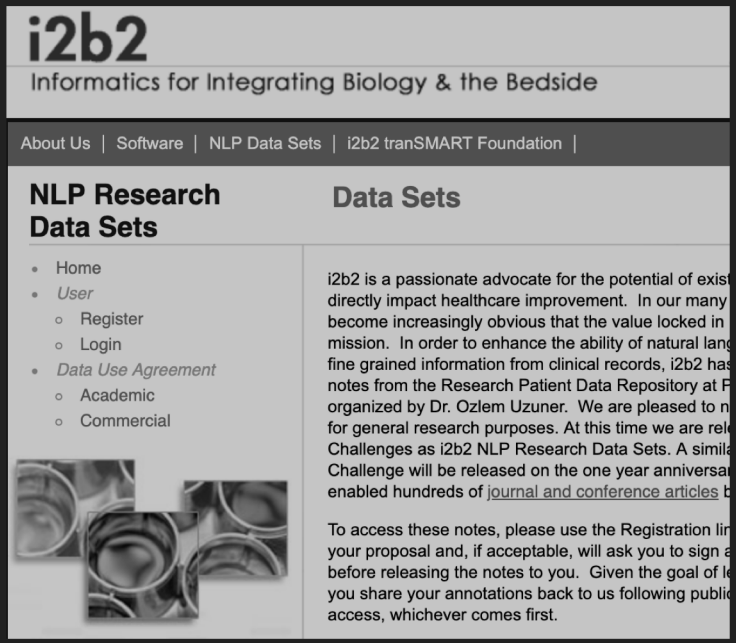
Alistair E.W. Johnson[1,*], Tom J. Pollard[1,*], Lu Shen[2], Li-wei H. Lehman[1], Mengling Feng[1,3], Mohammad Ghassemi[1], Benjamin Moody[1], Peter Szolovits[4], Leo Anthony Celi[1,2] & Roger G. Mark[1,2]

MIMIC-III ('Medical Information Mart for Intensive Care') is a large, single-center database comprising information relating to patients admitted to critical care units at a large tertiary care hospital. Data includes vital signs, medications, laboratory measurements, observations and notes charted by care providers, fluid balance, procedure codes, diagnostic codes, imaging reports, hospital length of stay, survival data, and more. The database supports applications including academic and industrial research, quality improvement initiatives, and higher education coursework.

Design Type(s)	data integration objective
Measurement Type(s)	Demographics • clinical measurement • intervention • Billing • Medical History Dictionary • Pharmacotherapy • clinical laboratory test • medical data
Technology Type(s)	Electronic Medical Record • Medical Record • Electronic Billing System • Medical Coding Process Document • Free Text Format
Factor Type(s)	
Sample Characteristic(s)	Homo sapiens

https://www.nature.com/articles/sdata201635.pdf

I2B2 Clinical Notes Data



i2b2

Informatics for Integrating Biology & the Bedside

About Us | Software | NLP Data Sets | i2b2 tranSMART Foundation |

NLP Research Data Sets

Data Sets

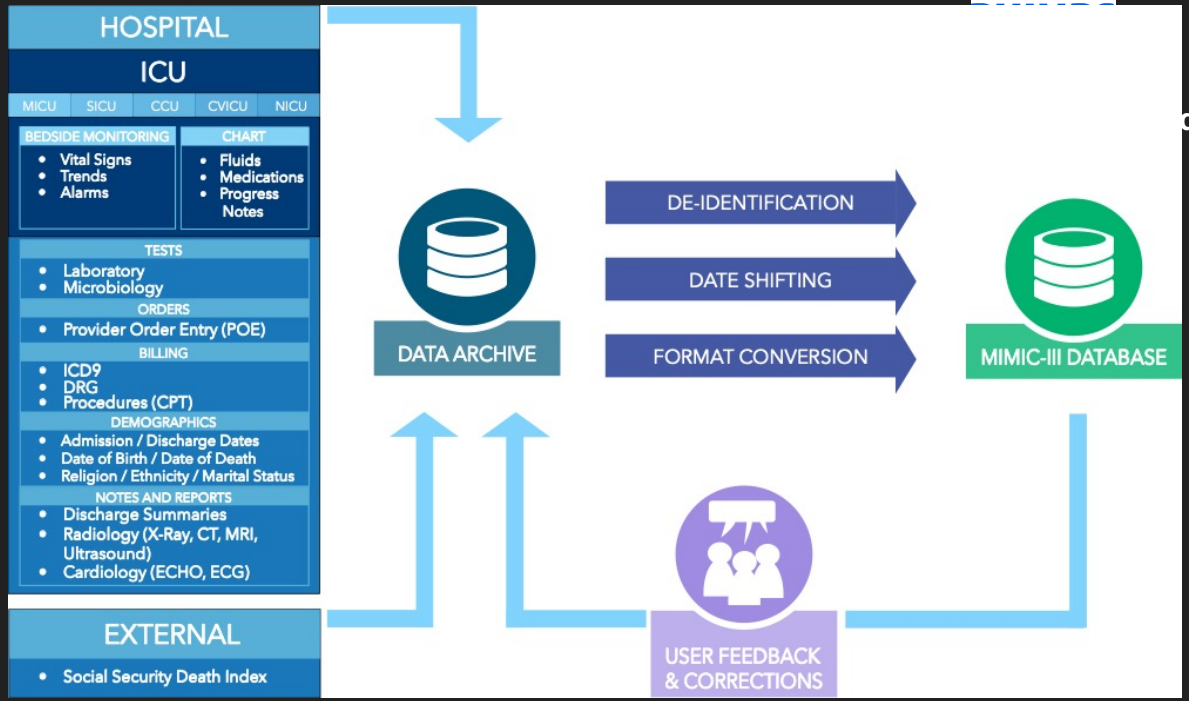
- Home
- *User*
  - Register
  - Login
- *Data Use Agreement*
  - Academic
  - Commercial

i2b2 is a passionate advocate for the potential of exist
directly impact healthcare improvement. In our many
become increasingly obvious that the value locked in
mission. In order to enhance the ability of natural lang
fine grained information from clinical records, i2b2 has
notes from the Research Patient Data Repository at P
organized by Dr. Ozlem Uzuner. We are pleased to n
for general research purposes. At this time we are rele
Challenges as i2b2 NLP Research Data Sets. A simila
Challenge will be released on the one year anniversa
enabled hundreds of journal and conference articles b

To access these notes, please use the Registration lir
your proposal and, if acceptable, will ask you to sign a
before releasing the notes to you. Given the goal of le
you share your annotations back to us following public
access, whichever comes first.

https://www.i2b2.org/NLP/DataSets/Main.php

MIMIC III Dataset



dical Information Mart for Intensive Care

Single Centre: Beth Israel Deaconess Medical Centre

- U.S. Based (Boston, MA)
- Has MICU, SICU, CCU, CSRU, TSICU, ...

Detailed in-ICU information derived from: electronic medical records, critical care information systems, lab system,...

Limited out-of-ICU information (social security death masterfile)

>60,000 ICU stays,
>40,000 patients (2002-2012)

https://github.com/MIT-LCP/mimic-workshop/tree/master/intro_to_mimic

MIMIC

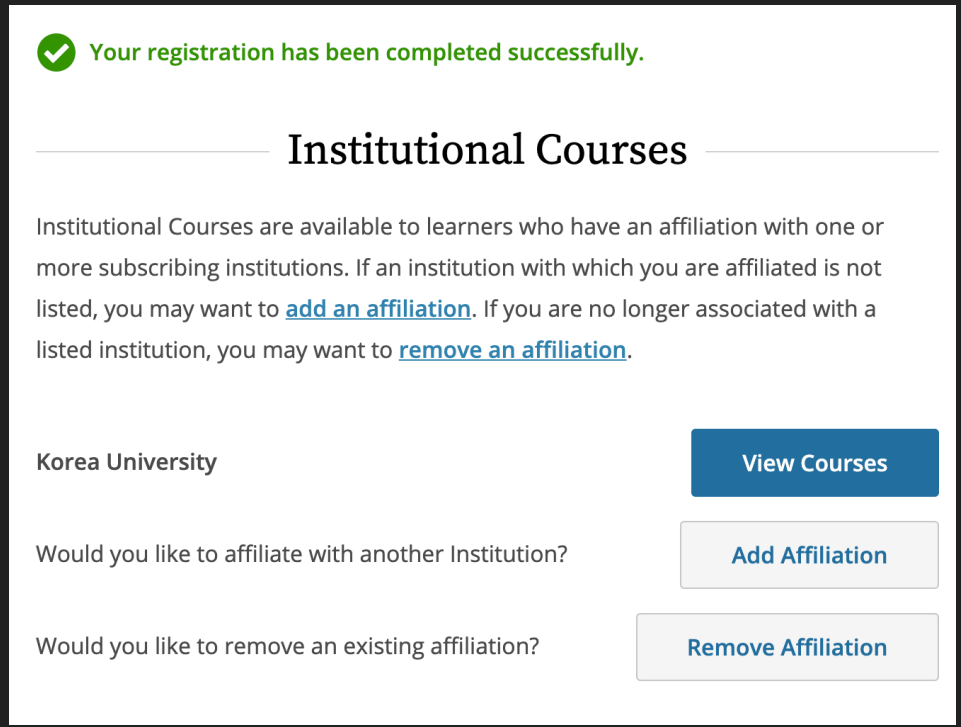
Data Analysis

ACESS MIMIC Dataset

Complete the required training course

Prior to requesting access to MIMIC, you will need to complete the CITI "Data or Specimens Only Research" course:

- First register on the CITI program website, selecting "Massachusetts Institute of Technology Affiliates" as your affiliation (***not*** *"independent learner"*): https://www.citiprogram.org/index.cfm?pageID=154&icat=0&ac=0
- Follow the links to add a Massachusetts Institute of Technology Affiliates course. In the Human Subjects training category, select the "Data or Specimens Only Research" course
- Complete the course and save a copy of your completion report. The completion report lists all modules completed, with dates and scores.



https://mimic.physionet.org/gettingstarted/access/

ACESS MIMIC Dataset

Request access to MIMIC-III:

- Create an account on PhysioNet using the following link: https://physionet.org/pnw/login. If you already have a PhysioNetWorks account, log in to it.
- Follow the instructions on PhysioNet to apply for access to the MIMIC-III project, remembering to provide your CITI completion report: https://physionet.org/works/MIMICIIIClinicalDatabase/access.sht
- When your application has been approved you will receive emails containing instructions for downloading the database from PhysioNetWorks. Approval may take several business days, and will be delayed if your request is missing any required information.

https://mimic.physionet.org/gettingstarted/access/

PhysioNet Clinical Database Restricted Data Use Agreement

If I am granted access to PhysioNet Clinical Databases (MIMIC-II, MIMIC-III, eICU Collaborative Research Database, Deidentified Medical Text, MIMIC-CXR), I agree to the terms and conditions below:

1. I will not attempt to identify any individual or institution referenced in PhysioNet restricted data.
2. I will exercise all reasonable and prudent care to avoid disclosure of the identity of any individual or institution referenced in PhysioNet restricted data in any publication or other communication.
3. I will not share access to PhysioNet restricted data with anyone else.
4. I will exercise all reasonable and prudent care to maintain the physical and electronic security of PhysioNet restricted data.
5. If I find information within PhysioNet restricted data that I believe might permit identification of any individual or institution, I will report the location of this information promptly by email to PHI-report@physionet.org, citing the location of the specific information in question so that it can be investigated and removed if necessary.
6. I have requested access to PhysioNet restricted data for the sole purpose of lawful use in scientific research, and I will use my privilege of access, if it is granted, for this purpose and no other.
7. I have completed a training program in human research subject protections and HIPAA regulations, and I am submitting proof of having done so.
8. I will indicate the general purpose for which I intend to use the database in my application.
9. If I openly disseminate my results, I will also contribute the code used to produce those results to a repository that is open to the research community.
10. This agreement may be terminated by either party at any time, but my obligations with respect to restricted data from PhysioNet shall continue after termination.

I agree | I do not agree

DOWNLOAD MIMIC dataset

MIMIC3py

A Python library to load and analyze the MIMIC III Critical Care Database

"MIMIC-III (**M**edical **I**nformation **M**art for **I**ntensive **C**are III) is a large, freely-available database comprising deidentified health-related data associated with over forty thousand patients who stayed in critical care units of the Beth Israel Deaconess Medical Center between 2001 and 2012." It includes demographics, vital signs, laboratory tests, medications, clinical notes, and more.

More details are availible at:

- https://mimic.physionet.org/about/mimic/
- http://www.nature.com/articles/sdata201635

This repository contains Python 3 code that will help you:

- Download the data.
- Load the data into Pandas data frames.

The package is hosted on pypi at: https://pypi.python.org/pypi/MIMIC3py/0.11

You can install it using `pip install MIMIC3py`

https://github.com/SpiroGanas/MIMIC3py/tree/master/MIMIC3py

config.py

```
# This is the folder where the files will located
local_save_folder = "/data/ehr/mimic3"

# ENTER YOUR PHYSIONET USERNAME AND PASSWORD HERE ############################
physionet_USERNAME = "eo.sooheang@gmail.com"
physionet_PASSWORD = "******************"
##############################################################################

# You may need to update some of these variables if MIMIC III gets updated
physionet_BASE_URL = "https://physionet.org/works/MIMICIIIClinicalDatabase/files/ve

# Comment out any file you don't want to download.
# Note that some of the files are very, very big.
physionet_FILENAMES = [
    "ADMISSIONS.csv.gz",  # 12 MB
    "CALLOUT.csv.gz",  # 6.1 MB
    "CAREGIVERS.csv.gz",  # 199 KB
    "CHARTEVENTS.csv.gz",  # 33 GB ------BIG!!!
    "CPTEVENTS.csv.gz",  # 56 MB
    "DATETIMEEVENTS.csv.gz",  # 502 MB
    "D_CPT.csv.gz",  # 14 KB
    "DIAGNOSES_ICD.csv.gz",  # 19 MB
    "D_ICD_DIAGNOSES.csv.gz",  # 1.4 KB
    "D_ICD_PROCEDURES.csv.gz",  # 305 KB
    "D_ITEMS.csv.gz",  # 933 KB
    "D_LABITEMS.csv.gz",  # 43 KB
    "DRGCODES.csv.gz",  # 11 MB
    "ICUSTAYS.csv.gz",  # 6.1 MB
    "INPUTEVENTS_CV.csv.gz",  # 2.3 GB ------BIG!!!
    "INPUTEVENTS_MV.csv.gz",  # 931 MB
    "LABEVENTS.csv.gz",  # 1.8GB ------BIG!!!
    "MICROBIOLOGYEVENTS.csv.gz",  # 70 MB
    "NOTEEVENTS.csv.gz",  # 3.8 GB  ------BIG!!!
    "OUTPUTEVENTS.csv.gz",  # 379 MB
    "PATIENTS.csv.gz",  # 2.6 MB
    "PRESCRIPTIONS.csv.gz",  # 735 MB
    "PROCEDUREEVENTS_MV.csv.gz",  # 47 MB
    "PROCEDURES_ICD.csv.gz",  # 6.5 MB
    "SERVICES.csv.gz",  # 3.4 MB
```

Connect to the MIMIC

```
# Data processing libraries
import pandas as pd
import numpy as np
import itertools

# Database libraries
import psycopg2

# Stats libraries
from tableone import TableOne
import statsmodels.api as sm
import statsmodels.formula.api as smf
import scipy.stats

# Image libraries
# https://jakevdp.github.io/pdvega/
# jupyter nbextension enable vega3 --py --sys-prefix
import matplotlib.pyplot as plt
import pdvega
%matplotlib inline
```

```
# Create a database connection
user = 'XXX'
password = 'XXX'
host = 'localhost'
dbname = 'mimic'
schema = 'public, mimiciii_demo'
```

```
# Connect to the database
con = psycopg2.connect(dbname=dbname, user=user, host=host,
                       password=password)
cur = con.cursor()
cur.execute('SET search_path to {}'.format(schema))
```

```
Password:·····
```

https://github.com/MIT-LCP/bhi-bsn-challenge

Connect to the MIMIC Dataset (EASY WAY)

MIMIC-III Clinical Database (Demo)

Our new website is in development. Try it out at: https://alpha.physionet.org and give us feedback

The MIMIC-III (Medical Information Mart for Intensive Care) Clinical Database contains comprehensive clinical data from tens of thousands of Intensive Care Unit (ICU) patients. This project is a demo version which has only 100 patients and excludes the noteevents table.

Tom Pollard
Massachusetts Institute of Technology
Institute for Medical Engineering & Science
Cambridge, Massachusetts
United States of America

If you wish to obtain the demo database, you must agree to the following conditions:

1. You agree not to attempt to identify any of the individual subjects of the file.
2. If you find information within the file that you believe might permit identification of any of the individual subjects of the file, you agree to report this promptly by email to phi-support@physionet.org, citing the specific information in question so that it can be investigated and removed if necessary.
3. You agree to exercise all reasonable and prudent care to avoid disclosure of the identity of any of the individual subjects of the file in any publication or other communication of your analysis of the file.
4. If you agree to all of these terms and conditions, access to this file will be granted to you as an individual. You may not share it with anyone else. Your colleagues can obtain access to this file as individuals via the same procedure you are following.

For access to this project, indicate your agreement with the terms and conditions above by clicking on the words "I agree" below:

I agree

Downloads:

The MIMIC-III demo dataset (v1.4) can be downloaded either as 25 comma-separated-value (CSV) files or as a single Postgres database backup file (Postgres 9.5). The Postgres backup file can be downloaded by clicking on the link below:

- mimiciii_demo-postgres_9_5.backup

On a unix based system, the CSV files can be downloaded in a shell using the following command:

```
wget --user YOURUSERNAME --ask-password -A csv.gz -m -p -E -k -K -np
https://physionet.org/works/MIMICIIIClinicalDatabaseDemo/files/
```

After download, the files can be decompressed using the command-line tool gzip ("`gzip -d *.gz`"). Alternatively, files can be kept compressed and imported directly into a database.

https://physionet.org/works/MIMICIIIClinicalDatabaseDemo/

Tools for MIMIC

MIMIC Code Repository

build passing | DOI 10.5281/zenodo.821872 | chat on gitter

This is a repository of code shared by the research community. The repository is intended to be a central hub for sharing, refining, and reusing code used for analysis of the MIMIC critical care database. To find out more about MIMIC, please see: https://mimic.physionet.org

You can read more about the code repository in the following open access paper: The MIMIC Code Repository: enabling reproducibility in critical care research.

Brief introduction

The repository is organized as follows:

- benchmark - Various speed tests for indices
- buildmimic - Scripts to build MIMIC-III in a relational database management system (RDMS), in particular postgres is our RDMS of choice
- concepts - Useful views/summaries of the data in MIMIC-III, e.g. demographics, organ failure scores, severity of illness scores, durations of treatment, easier to analyze views, etc. The paper above describes these in detail.
- notebooks - A collection of R markdown and Jupyter notebooks which give examples of how to extract and analyze data
- notebooks/aline - An entire study reproduced in the MIMIC-III database - from cohort generation to hypothesis testing
- tests - You should always have tests!
- tutorials - Similar to the notebooks folder, but focuses on explaining concepts to new users

MIMIC Critical Care Datathon

These are training materials for the MIMIC Critical Care Database. The package includes:

- a demo version of MIMIC which can be quickly installed in the Firefox web browser with the SQLite Plugin.
- some sample SQL queries which can be used to query the MIMIC data
- an IPython Notebook which connects to the demo MIMIC database and allows analysis to be carried out using Python.

What is MIMIC-III?

MIMIC-III is a widely-used, freely available dataset developed by the MIT Lab for Computational Physiology, comprising deidentified health data associated with >40,000 critical care patients. It includes demographics, vital signs, laboratory tests, medications, and more. Details are available on the MIMIC website: https://mimic.physionet.org/

Workshop overview

During the workshop, you will:

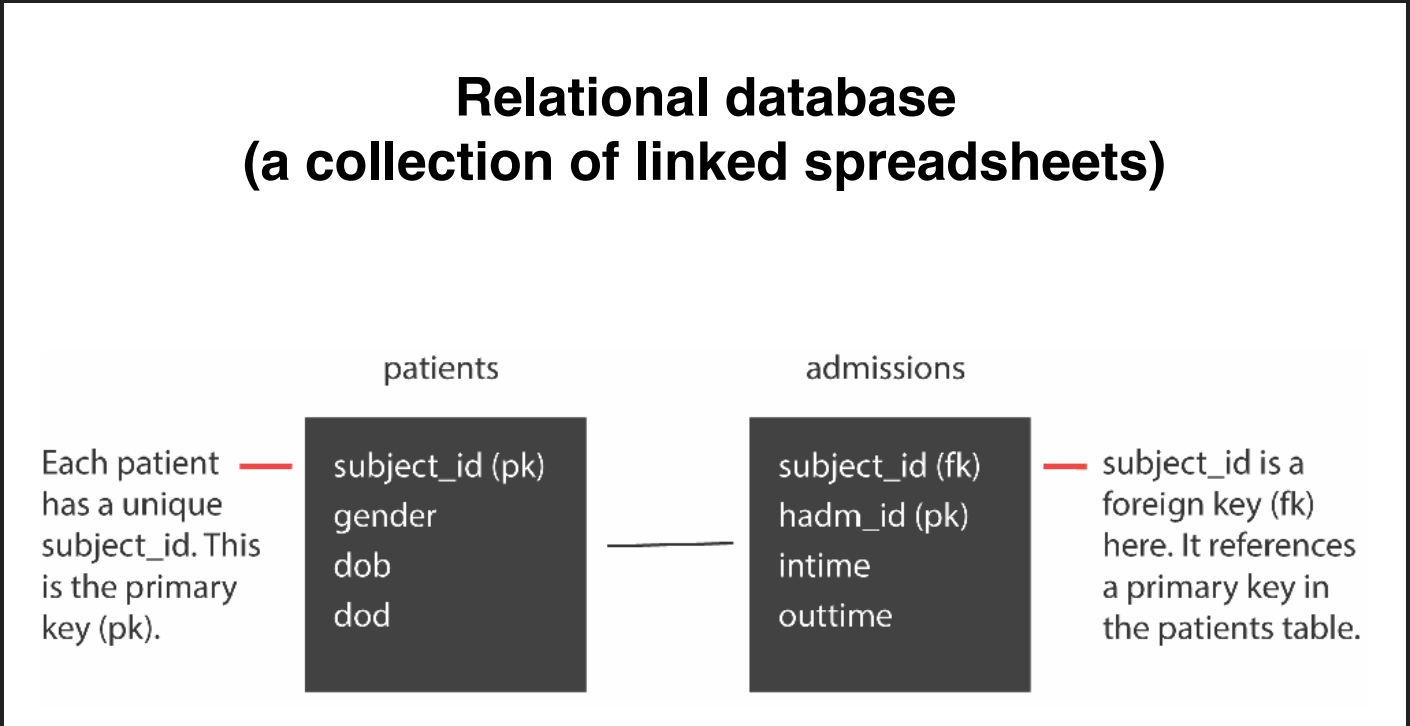
- Learn about MIMIC-III, the publicly accessible critical care database
- Create a local version of MIMIC-III with a small sample of patients using the Firefox SQLite Plugin
- Explore the patient data using SQL
- Plot and analyse the data using Python
- Get inspiration for future research projects

Downloading the materials

If you are familiar with git, please clone this repository. If not, click the 'Download ZIP' button on the right and then unzip the materials onto your computer.

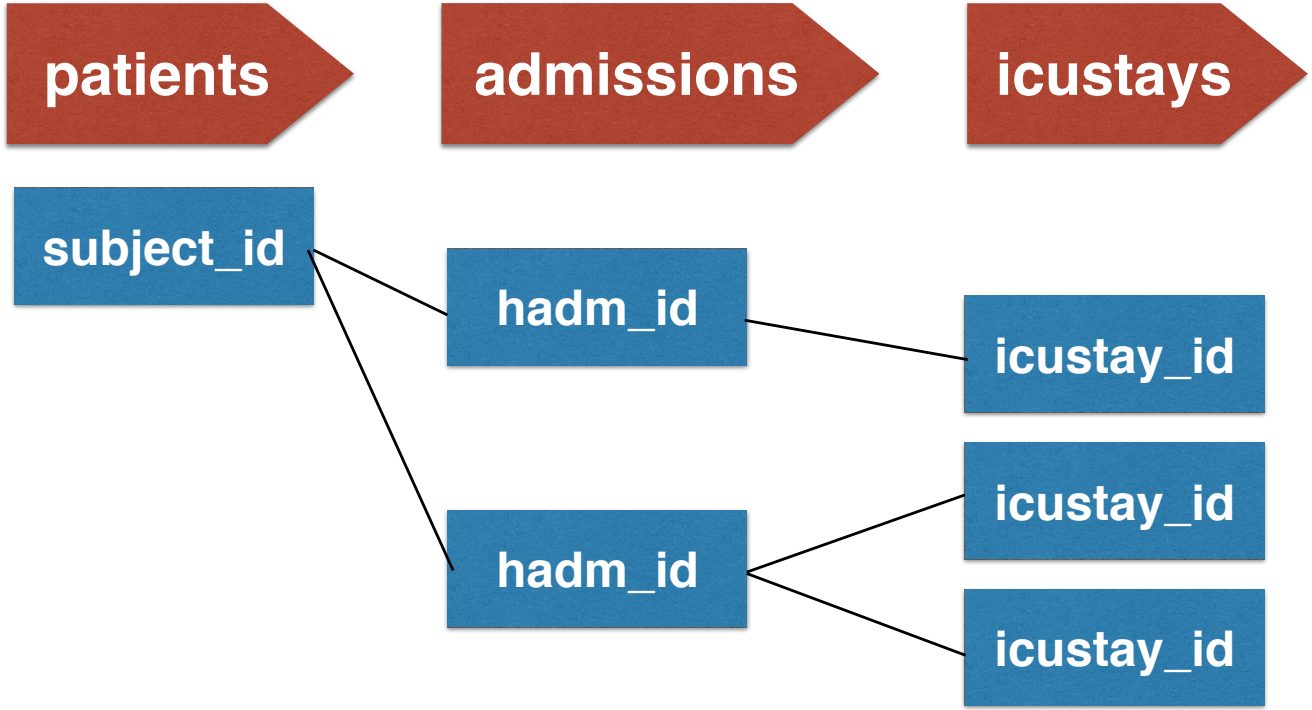
https://github.com/MIT-LCP/mimic-workshop

Navigating MIMIC



https://github.com/MIT-LCP/bhi-bsn-challenge/tree/master/slides

Patient tracking tables



https://github.com/MIT-LCP/bhi-bsn-challenge/tree/master/slides

Navigating MIMIC

Events tables

chartevents	Charted observations for a patient
labevents	Lab measurements both within hospital and outpatient clinics
inputevents	Input fluids (e.g. intravenous medications)
microbiology events	Microbiology measurements and sensitivities
noteevents	Deidentified patient notes

13

Other data tables

diagnoses_icd	Hospital assigned diagnosis codes
procedures_icd	Hospital assigned procedure codes
caregivers	Caregivers who have recorded data
prescriptions	Medications ordered for a patient

14

https://github.com/MIT-LCP/bhi-bsn-challenge/tree/master/slides

Navigating MIMIC

```
[1]:
# Data processing libraries
import pandas as pd
import numpy as np
import itertools

# Database libraries
import psycopg2
import sqlite3

# Stats libraries
from tableone import TableOne
import statsmodels.api as sm
import statsmodels.formula.api as smf
import scipy.stats

# Image libraries
# https://jakevdp.github.io/pdvega/
# jupyter nbextension enable vega3 --py --sys-prefix
import matplotlib.pyplot as plt
import pdvega
%matplotlib inline
```

https://github.com/MIT-LCP/mimic-workshop/tree/master/temp

Connect to the database

- We can use the sqlite3 library to connect to the MIMIC database
- Once the connection is established, we'll run a simple SQL query.

```
In [2]:
# Connect to the MIMIC database
conn = sqlite3.connect('../data/mimicdata.sqlite')
```

```
In [3]:
# Create our test query
test_query = """
SELECT subject_id, hadm_id, admittime, dischtime, admission_type, diagnosis
FROM admissions
LIMIT 10;
"""
```

```
In [4]:
# Run the query and assign the results to a variable
test = pd.read_sql_query(test_query,conn)
```

```
In [5]:
# Display the first few rows
test.head()
```

Out[5]:

	SUBJECT_ID	HADM_ID	ADMITTIME	DISCHTIME	ADMISSION_TYPE	DIAGNOSIS
0	40036	198489	2141-08-01 23:46:00	2141-08-09 19:15:00	EMERGENCY	SEPSIS
1	40080	162107	2106-05-31 16:43:00	2106-06-05 01:18:00	EMERGENCY	CONGESTIVE HEART FAILURE
2	40084	195762	2173-01-31 22:11:00	2173-02-05 01:31:00	EMERGENCY	INTRACRANIAL HEMORRHAGE;OPEN FX
3	40116	157106	2150-02-19 00:12:00	2150-03-11 13:58:00	EMERGENCY	GASTROINTESTINAL BLEED
4	40120	146466	2120-01-27 20:41:00	2120-02-12 17:14:00	EMERGENCY	CONGESTIVE HEART FAILURE

Navigating MIMIC

```
# Run query and assign the results to a Pandas DataFrame
# Requires the icustay_detail view from:
# https://github.com/MIT-LCP/mimic-code/tree/master/concepts/demographics
# And the OASIS score from:
# https://github.com/MIT-LCP/mimic-code/tree/master/concepts/severityscores
query = \
"""
WITH first_icu AS (
    SELECT i.subject_id, i.hadm_id, i.icustay_id, i.gender, i.admittime admittime_hospital,
      i.dischtime dischtime_hospital, i.los_hospital, i.age, i.admission_type,
      i.hospital_expire_flag, i.intime intime_icu, i.outtime outtime_icu, i.los_icu,
      s.first_careunit
    FROM icustay_detail i
    LEFT JOIN icustays s
    ON i.icustay_id = s.icustay_id
    WHERE i.hospstay_seq = 1
      AND i.icustay_seq = 1
      AND i.age >= 16
)
SELECT f.*, o.icustay_expire_flag, o.oasis, o.oasis_prob
FROM first_icu f
LEFT JOIN oasis o
ON f.icustay_id = o.icustay_id;
"""

data = pd.read_sql_query(query,con)
```

https://github.com/MIT-LCP/bhi-bsn-challenge/blob/master/challenge-demo.ipynb

Navigating MIMIC

In [8]:
```
data['admitday_hospital'] = data.admittime_hospital.dt.weekday_name
data['dischday_hospital'] = data.dischtime_hospital.dt.weekday_name
data['inday_icu'] = data.intime_icu.dt.weekday_name
data['inday_icu_seq'] = data.intime_icu.dt.weekday
data['outday_icu'] = data.outtime_icu.dt.weekday_name
data.head()
```

Out[8]:

reunit	icustay_expire_flag	oasis	oasis_prob	admitday_hospital	dischday_hospital	inday_icu	inday_icu_seq	outday_icu
	0	35	0.152892	Friday	Thursday	Friday	4	Thursday
	0	26	0.054187	Sunday	Wednesday	Sunday	6	Monday
	0	56	0.724202	Monday	Saturday	Monday	0	Wednesday
	1	47	0.454600	Wednesday	Saturday	Wednesday	2	Saturday
	0	35	0.152892	Thursday	Sunday	Thursday	3	Sunday

https://github.com/MIT-LCP/bhi-bsn-challenge/blob/master/challenge-demo.ipynb

Navigating MIMIC

```
[12]: columns = ['gender', 'los_hospital', 'age', 'admission_type', 'hospital_expire_flag',
                 'los_icu','icustay_expire_flag', 'oasis', 'oasis_prob', 'first_careunit',
                 'inday_icu_wkd']

      groupby = 'inday_icu'

      pval = False

      categorical = ['gender','admission_type','hospital_expire_flag','icustay_expire_flag',
                     'first_careunit','inday_icu_wkd']

      t = TableOne(data, columns=columns, categorical=categorical, groupby=groupby, pval=pval)
      t.tableone
```

ADMISSION_TYPE

ADMISSION_TYPE describes the type of the admission: 'ELECTIVE', 'URGENT', 'NEWBORN' or 'EMERGENCY'. Emergency/urgent indicate unplanned medical care, and are often collapsed into a single category in studies. Elective indicates a previously planned hospital admission. Newborn indicates that the HADM_ID pertains to the patient's birth.

https://mimic.physionet.org/mimictables/admissions/

		Grouped by inday_icu							
		Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday	isnull
variable	level								
n		6263	6097	4235	3960	5876	6141	5985	
admission_type	ELECTIVE	1016 (16.22)	1265 (20.75)	162 (3.83)	101 (2.55)	999 (17.0)	1292 (21.04)	1243 (20.77)	0
	EMERGENCY	5118 (81.72)	4687 (76.87)	3852 (90.96)	3681 (92.95)	4746 (80.77)	4704 (76.6)	4600 (76.86)	
	URGENT	129 (2.06)	145 (2.38)	221 (5.22)	178 (4.49)	131 (2.23)	145 (2.36)	142 (2.37)	
age		74.56 (53.22)	73.13 (51.37)	73.92 (58.58)	75.26 (60.66)	75.51 (55.46)	75.48 (55.22)	74.16 (53.88)	0
first_careunit	CCU	838 (13.38)	918 (15.06)	695 (16.41)	621 (15.68)	850 (14.47)	919 (14.96)	851 (14.22)	0
	CSRU	1416 (22.61)	1632 (26.77)	237 (5.6)	194 (4.9)	1282 (21.82)	1575 (25.65)	1268 (21.19)	
	MICU	2139 (34.15)	1940 (31.82)	1765 (41.68)	1706 (43.08)	2020 (34.38)	2019 (32.88)	2020 (33.75)	
	SICU	1044 (16.67)	865 (14.19)	743 (17.54)	743 (18.76)	996 (16.95)	933 (15.19)	1038 (17.34)	
	TSICU	826 (13.19)	742 (12.17)	795 (18.77)	696 (17.58)	728 (12.39)	695 (11.32)	808 (13.5)	
gender	F	2662 (42.5)	2559 (41.97)	1857 (43.85)	1736 (43.84)	2603 (44.3)	2671 (43.49)	2636 (44.04)	0
	M	3601 (57.5)	3538 (58.03)	2378 (56.15)	2224 (56.16)	3273 (55.7)	3470 (56.51)	3349 (55.96)	

https://github.com/MIT-LCP/bhi-bsn-challenge/blob/master/challenge-demo.ipynb

Navigating MIMIC

In [6]:
```
query = """
SELECT de.icustay_id
  , (strftime('%s',de.charttime)-strftime('%s',ie.intime))/60.0/60.0 as HO
  , di.label
  , de.value
  , de.valuenum
  , de.uom
FROM chartevents de
INNER join d_items di
ON de.itemid = di.itemid
INNER join icustays ie
ON de.icustay_id = ie.icustay_id
WHERE de.subject_id = 40084
ORDER BY charttime;
"""

ce = pd.read_sql_query(query,conn)


# OPTION 2: load chartevents from a CSV file
# ce = pd.read_csv('data/example_chartevents.csv', index_col='HOURSSINCEAD
```

In [7]:
```
# Preview the data
# Use 'head' to limit the number of rows returned
ce.head()
```

Out[7]:

	ICUSTAY_ID	HOURS	LABEL	VALUE	VALUENUM	UOM
0	264630	0.201667	PH (dipstick)	5	5.00	units
1	264630	0.201667	Specific Gravity (urine)	1.02	1.02	
2	264630	1.801667	Temperature Fahrenheit	94.3	94.30	¿F
3	264630	2.668333	Heart Rate	84	84.00	bpm
4	264630	2.668333	Non Invasive Blood Pressure systolic	106	106.00	mmHg

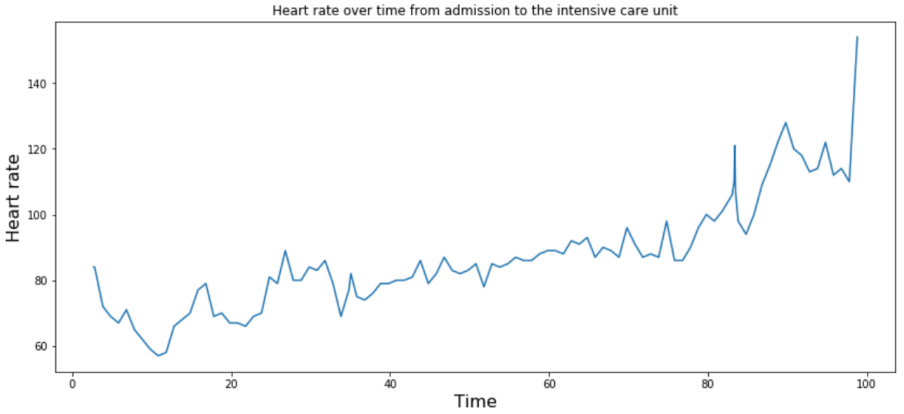
Plot 1: How did the patients heart rate change over time?

- Using the methods described above to select our data of interest, we can create our x and y axis values to create a time series plot of heart rate.

In [11]:
```
# Which time stamps have a corresponding heart rate measurement?
print(ce.index[ce.LABEL=='Heart Rate'])
```

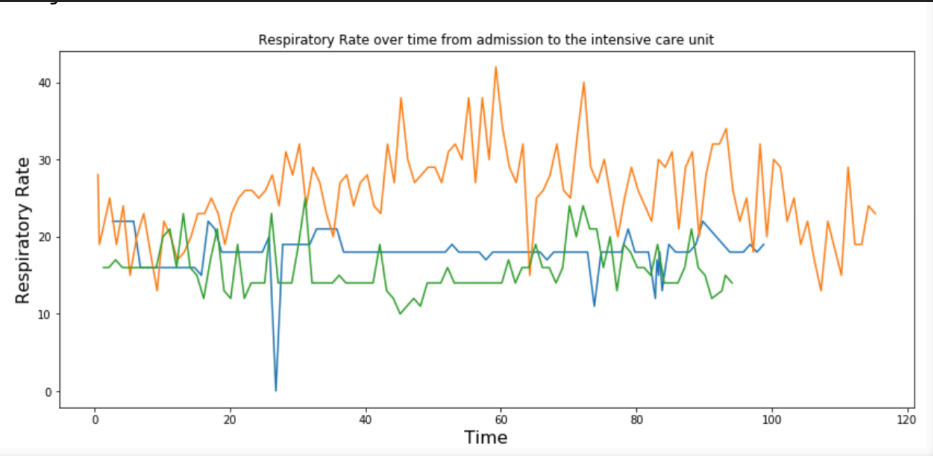
```
Int64Index([   3,   18,   97,  101,  118,  161,  190,  198,  230,  285,
            ...
            2241, 2276, 2284, 2293, 2303, 2323, 2332, 2358, 2365, 2371],
           dtype='int64', length=103)
```

```
Text(0.5, 1.0, 'Heart rate over time from admission to the intensive care unit')
```



https://github.com/MIT-LCP/mimic-workshop/tree/master/temp

Navigating MIMIC




```
data = []
for subject_id in [40084, 40080, 40004]:
    query = """
    SELECT de.icustay_id
      , (strftime('%s',de.charttime)-strftime('%s',ie.intime))/60.0/60.0 a
      , di.label
      , de.value
      , de.valuenum
      , de.uom
    FROM chartevents de
    INNER join d_items di
    ON de.itemid = di.itemid
    INNER join icustays ie
    ON de.icustay_id = ie.icustay_id
    WHERE de.subject_id = """ + str(subject_id) + """
    ORDER BY charttime;
    """

    ce = pd.read_sql_query(query,conn)

    valueName = "Respiratory Rate";

    # Set x equal to the times
    x_hr = ce.HOURS[ce.LABEL==valueName]

    # Set y equal to the heart rates
    y_hr = ce.VALUENUM[ce.LABEL==valueName]

    # Plot time against heart rate
    plt.figure(figsize=(14, 6))
    data.append([x_hr,y_hr]);

for patient in data:
    plt.plot(patient[0], patient[1]);

plt.xlabel('Time',fontsize=16)
plt.ylabel(valueName,fontsize=16)
plt.title(valueName + ' over time from admission to the intensive care uni
```

https://github.com/MIT-LCP/mimic-workshop/tree/master/temp

Navigating MIMIC

Select data on the first hospital stay

In [4]:

```
# Run query and assign the results to a Pandas DataFrame
# Get first admission for each patient
query = \
"""
WITH admit AS (
    SELECT p.gender,
        ROUND( (CAST(EXTRACT(epoch FROM a.dischtime - a.admittime)/(60*60*24) AS numeric)), 4) AS
 los_hospital,
        ROUND( (CAST(EXTRACT(epoch FROM a.admittime - p.dob)/(60*60*24*365.242) AS numeric)), 4) A
S age,
        DENSE_RANK() OVER (PARTITION BY a.subject_id ORDER BY a.admittime) AS admission_seq,
        a.ethnicity, a.admission_type, a.insurance, a.religion, a.marital_status, a.hospital_expir
e_flag
    FROM patients p
    LEFT JOIN admissions a
    ON p.subject_id = a.subject_id)
SELECT *
FROM admit
WHERE admission_seq = 1;
"""

data = pd.read_sql_query(query,con)
```

https://github.com/MIT-LCP/mimic-code/blob/master/notebooks/tableone-demo.ipynb

Navigating MIMIC

```
# Columns to include in the summary table
columns = ['gender', 'los_hospital', 'age', 'ethnicity','admission_type', 'insurance',
           'religion', 'marital_status','hospital_expire_flag']

# List of categorical variables
categorical = ['gender', 'ethnicity', 'insurance', 'religion', 'marital_status',
           'hospital_expire_flag']

# Group the data
groupby = 'admission_type'

# Display the top n number of categorical variables
limit = 3

# Compute p values
pval = False

# Display a count of null values
isnull = False

t = TableOne(data, columns=columns, categorical=categorical,
         groupby=groupby, limit=limit, pval=pval, isnull=isnull)

t.tableone
```

		Grouped by admission_type		
		ELECTIVE	EMERGENCY	URGENT
variable	level			
n		8	90	2
age		74.23 (11.05)	90.01 (68.12)	75.66 (4.29)
ethnicity	WHITE	8 (100.0)	65 (72.22)	1 (50.0)
	UNKNOWN/NOT SPECIFIED		9 (10.0)	1 (50.0)
	BLACK/AFRICAN AMERICAN		6 (6.67)	
gender	F	5 (62.5)	48 (53.33)	2 (100.0)
	M	3 (37.5)	42 (46.67)	
hospital_expire_flag	0	8 (100.0)	58 (64.44)	1 (50.0)
	1		32 (35.56)	1 (50.0)
insurance	Medicare	5 (62.5)	70 (77.78)	1 (50.0)
	Private	3 (37.5)	15 (16.67)	1 (50.0)
	Medicaid		4 (4.44)	
los_hospital		11.67 (11.79)	9.86 (14.43)	6.26 (0.81)
marital_status	MARRIED	4 (50.0)	36 (48.0)	1 (50.0)
	SINGLE	2 (25.0)	18 (24.0)	
	WIDOWED	1 (12.5)	13 (17.33)	
religion	CATHOLIC	6 (75.0)	33 (37.08)	
	UNOBTAINABLE		16 (17.98)	
	NOT SPECIFIED	1 (12.5)	14 (15.73)	

Navigating MIMIC (GLM)

In [18]:
```
# R style syntax
simple_glm = smf.glm('hospital_expire_flag ~ C(inday_icu_wkd)',
                     data=data, family=sm.families.Binomial()).fit()
simple_glm.summary2()

# Alternative syntax
# y = data.hospital_expire_flag
# X = sm.tools.add_constant(data.inday_icu_wkd.factorize()[0])
# simple_glm = sm.GLM(y, X, family=sm.families.Binomial()).fit()
# simple_glm.summary2()
```

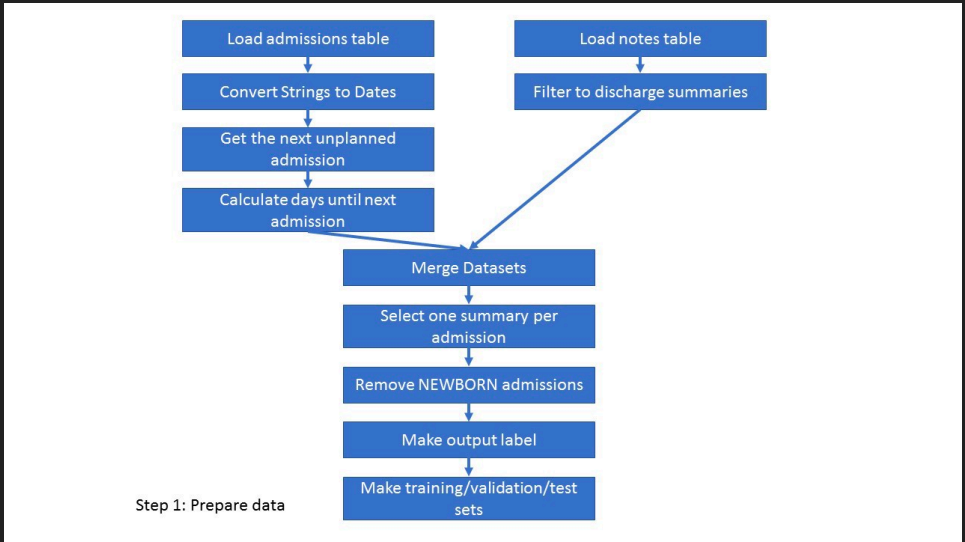
Out[18]:

Model:	GLM	AIC:	27416.8526
Link Function:	logit	BIC:	-379723.8199
Dependent Variable:	hospital_expire_flag	Log-Likelihood:	-13706.
Date:	2018-03-02 10:45	LL-Null:	-13738.
No. Observations:	38557	Deviance:	27413.
Df Model:	1	Pearson chi2:	3.86e+04
Df Residuals:	38555	Scale:	1.0000
Method:	IRLS		

	Coef.	Std.Err.	z	P>|z|	[0.025	0.975]
Intercept	-2.1127	0.0185	-114.2000	0.0000	-2.1490	-2.0765
C(inday_icu_wkd)[T.weekend]	0.2992	0.0368	8.1288	0.0000	0.2270	0.3713

https://github.com/MIT-LCP/bhi-bsn-challenge/blob/master/challenge-demo.ipynb

Navigating MIMIC (Clinical Text)



NOTEEVENTS — contains all notes for each hospitalization (links with HADM_ID)

https://towardsdatascience.com/introduction-to-clinical-natural-language-processing-predicting-hospital-readmission-with-1736d52bc709

Navigating MIMIC (Clinical Text)

NOTEEVENTS — contains all notes for each hospitalization (links with HADM_ID)

```
[13]: print('Number of notes:',len(df_notes))

      Number of notes: 2083180

      There can be multiple notes per hospitalizations so don't worry that the number of notes is higher than the number of hospitalizations.

[14]: df_notes.CATEGORY.unique()

[14]: array(['Discharge summary', 'Echo', 'ECG', 'Nursing', 'Physician ',
             'Rehab Services', 'Case Management ', 'Respiratory ', 'Nutrition',
             'General', 'Social Work', 'Pharmacy', 'Consult', 'Radiology',
             'Nursing/other'], dtype=object)

[ ]: df_notes.head()

[ ]: # look at the first note
     df_notes.TEXT.iloc[0]

     We can see that the dates and PHI have been converted for confidentiality. There are '\n' characters, numbers and punctuation.

     At this point, we have to make a choice on what notes to use. For simplicity, let's use the discharge summary, but we could use all the notes by concatenating them.
```

https://towardsdatascience.com/introduction-to-clinical-natural-language-processing-predicting-hospital-readmission-with-1736d52bc709

Navigating MIMIC (Clinical Text)

Prepare a label

I like to create a specific column in the dataframe as OUTPUT_LABEL that has exactly what we are trying to predict. Here we want if the patient was re-admitted within 30 days

```
df_adm_notes_clean['OUTPUT_LABEL'] = (df_adm_notes_clean.DAYS_NEXT_ADMIT < 30).astype('int')
```

```
print('Number of positive samples:', (df_adm_notes_clean.OUTPUT_LABEL == 1).sum())
print('Number of negative samples:',  (df_adm_notes_clean.OUTPUT_LABEL == 0).sum())
print('Total samples:', len(df_adm_notes_clean))
```

```
Number of positive samples: 3004
Number of negative samples: 48109
Total samples: 51113
```

```
my_stop_words = ['the','and','to','of','was','with','a','on','in','for','name',
                 'is','patient','s','he','at','as','or','one','she','his','her','am',
                 'were','you','pt','pm','by','be','had','your','this','date',
                'from','there','an','that','p','are','have','has','h','but','o',
                'namepattern','which','every','also']
```

```
from sklearn.feature_extraction.text import CountVectorizer
vect = CountVectorizer(max_features = 3000,
                       tokenizer = tokenizer_better,
                       stop_words = my_stop_words)
# this could take a while
vect.fit(df_train.TEXT.values)
```

https://towardsdatascience.com/introduction-to-clinical-natural-language-processing-predicting-hospital-readmission-with-1736d52bc709

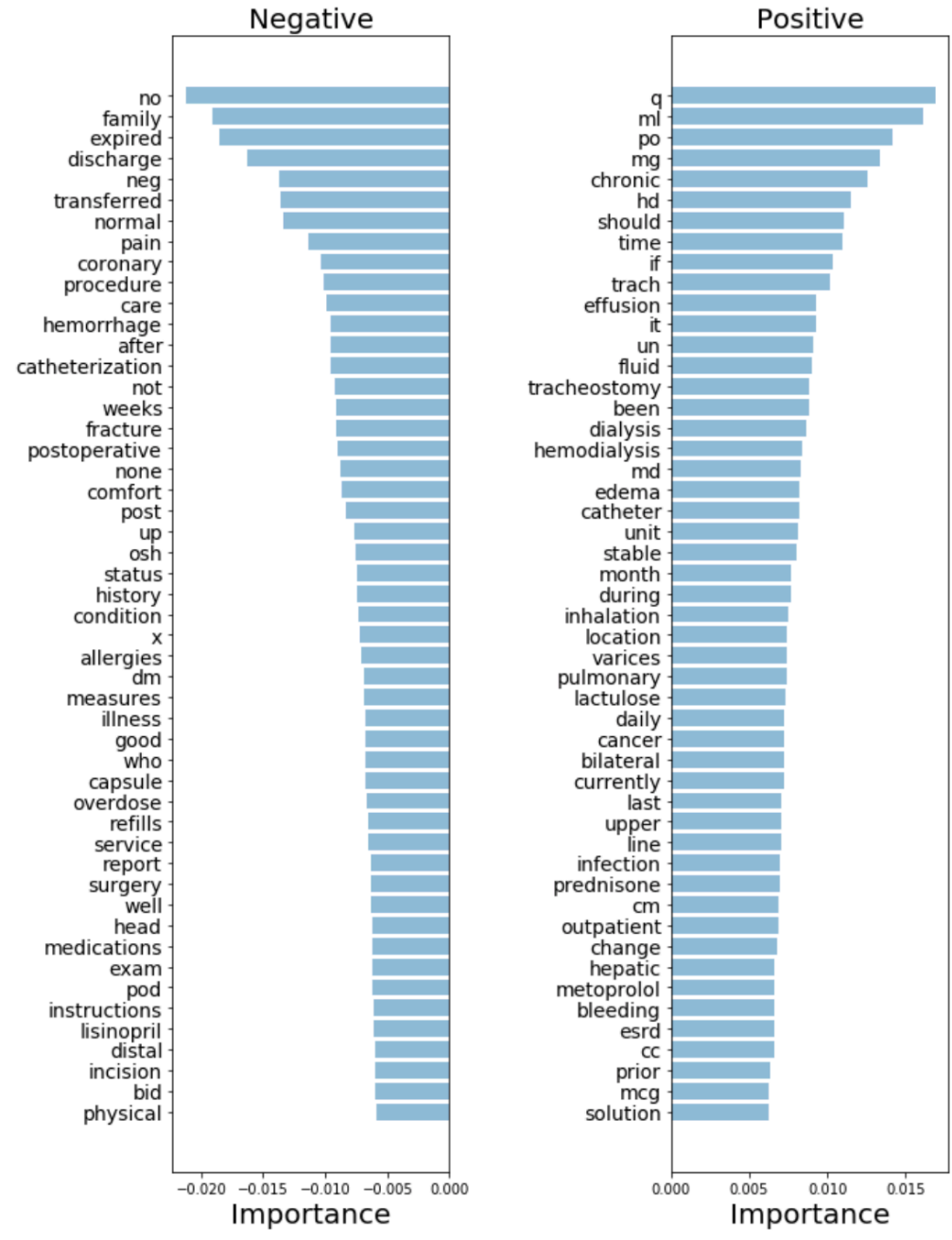
Navigating MIMIC (Clinical Text)

```
# logistic regression
from sklearn.linear_model import LogisticRegression
clf=LogisticRegression(C = 0.0001, penalty = 'l2', random_state = 42)
clf.fit(X_train_tf, y_train)
```

```
def get_most_important_features(vectorizer, model, n=5):
    index_to_word = {v:k for k,v in vectorizer.vocabulary_.items()}

    # loop for each class
    classes ={}
    for class_index in range(model.coef_.shape[0]):
        word_importances = [(el, index_to_word[i]) for i,el in enumerate(model.coef_[class_index
])]
        sorted_coeff = sorted(word_importances, key = lambda x : x[0], reverse=True)
        tops = sorted(sorted_coeff[:n], key = lambda x : x[0])
        bottom = sorted_coeff[-n:]
        classes[class_index] = {
            'tops':tops,
            'bottom':bottom
        }
    return classes
```

https://towardsdatascience.com/introduction-to-clinical-natural-language-processing-predicting-hospital-readmission-with-1736d52bc709

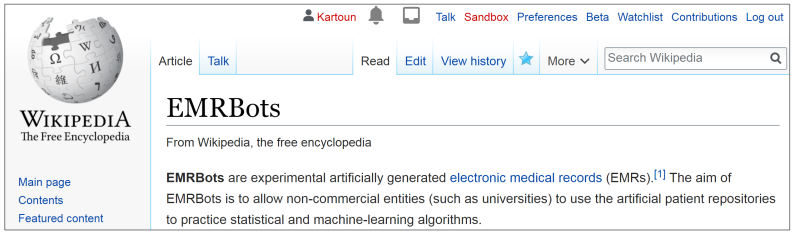

Negative
no
family
expired
discharge
neg
transferred
normal
pain
coronary
procedure
care
hemorrhage
after
catheterization
not
weeks
fracture
postoperative
none
comfort
post
up
osh
status
history
condition
x
allergies
dm
measures
illness
good
who
capsule
overdose
refills
service
report
surgery
well
head
medications
exam
pod
instructions
lisinopril
distal
incision
bid
physical
−0.020
−0.015
−0.010
−0.005
0.000
Importance
Positive
q
ml
po
mg
chronic
hd
should
time
if
trach
effusion
it
un
fluid
tracheostomy
been
dialysis
hemodialysis
md
edema
catheter
unit
stable
month
during
inhalation
location
varices
pulmonary
lactulose
daily
cancer
bilateral
currently
last
upper
line
infection
prednisone
cm
outpatient
change
hepatic
metoprolol
bleeding
esrd
cc
prior
mcg
solution
0.000
0.005
0.010
0.015
Importance

Discussion

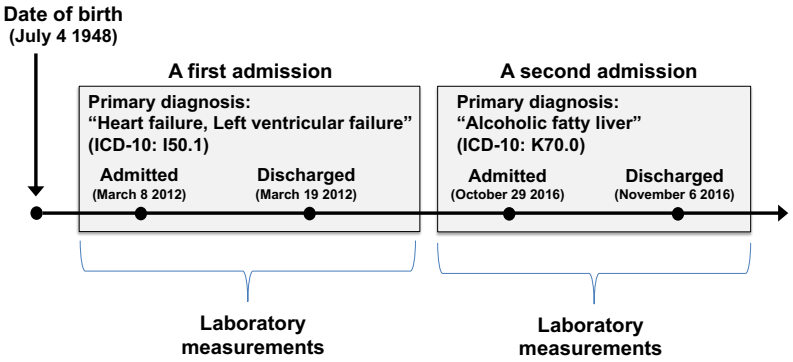
Synthetic Data

What are EMRBots?

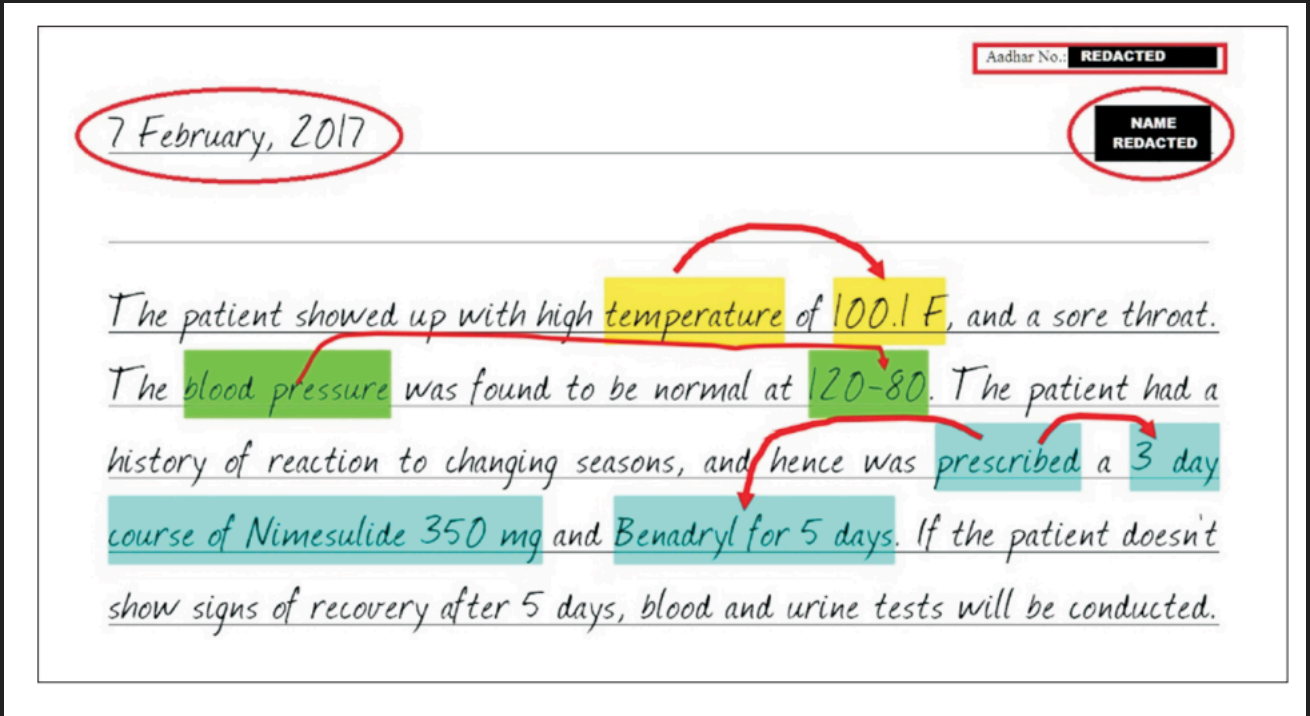
Because EMRs are subject to confidentiality requirements, accessing and analyzing EMR databases is a privilege given to only a small number of individuals.



How bots are created?



Clinical Text



Aggarwal et al. (2018). HEDEA: A Python Tool for Extracting and Analysing Semi-structured Information from Medical Records

Clinical Text Mining using Deep Learning

Publicly Available Clinical BERT Embeddings

Emily Alsentzer
Harvard-MIT
Cambridge, MA
emilya@mit.edu

John R. Murphy
MIT CSAIL
Cambridge, MA
jrmurphy@mit.edu

Willie Boag
MIT CSAIL
Cambridge, MA
wboag@mit.edu

Wei-Hung Weng
MIT CSAIL
Cambridge, MA
ckbjimmy@mit.edu

Di Jin
MIT CSAIL
Cambridge, MA
jindi15@mit.edu

Tristan Naumann
Microsoft Research
Redmond, WA
tristan@microsoft.com

Matthew B. A. McDermott
MIT CSAIL
Cambridge, MA
mmd@mit.edu

Model	MedNLI	i2b2 2006	i2b2 2010	i2b2 2012	i2b2 2014
BERT	77.6%	93.9	83.5	75.9	92.8
BioBERT	80.8%	**94.8**	86.5	78.9	**93.0**
Clinical BERT	80.8%	91.5	86.4	78.5	92.6
Discharge Summary BERT	80.6%	91.9	86.4	78.4	92.8
Bio+Clinical BERT	**82.7%**	94.7	87.2	**78.9**	92.5
Bio+Discharge Summary BERT	**82.7%**	94.8	**87.8**	78.9	92.7

Table 2: Accuracy (MedNLI) and Exact F1 score (i2b2) across various clinical NLP tasks.

Model	Disease			Operations			Generic		
	Glucose	Seizure	Pneumonia	Transfer	Admitted	Discharge	Beach	Newspaper	Table
BioBERT	insulin	episode	vaccine	drainage	admission	admission	coast	news	tables
	exhaustion	appetite	infection	division	sinking	wave	rock	official	row
	dioxide	attack	plague	transplant	hospital	sight	reef	industry	dinner
Clinical	potassium	headache	consolidation	transferred	admission	disposition	shore	publication	scenario
	sodium	stroke	tuberculosis	admitted	transferred	transfer	ocean	organization	compilation
	sugar	agitation	infection	arrival	admit	transferred	land	publicity	technology

https://arxiv.org/pdf/1904.03323.pdf

Clinical Text Mining using Deep Learning

clinicalBERT

Repository for Publicly Available Clinical BERT Embeddings Paper (NAACL Clinical NLP Workshop 2019)

We are in the process of submitting Clinical BERT to PhysioNet.

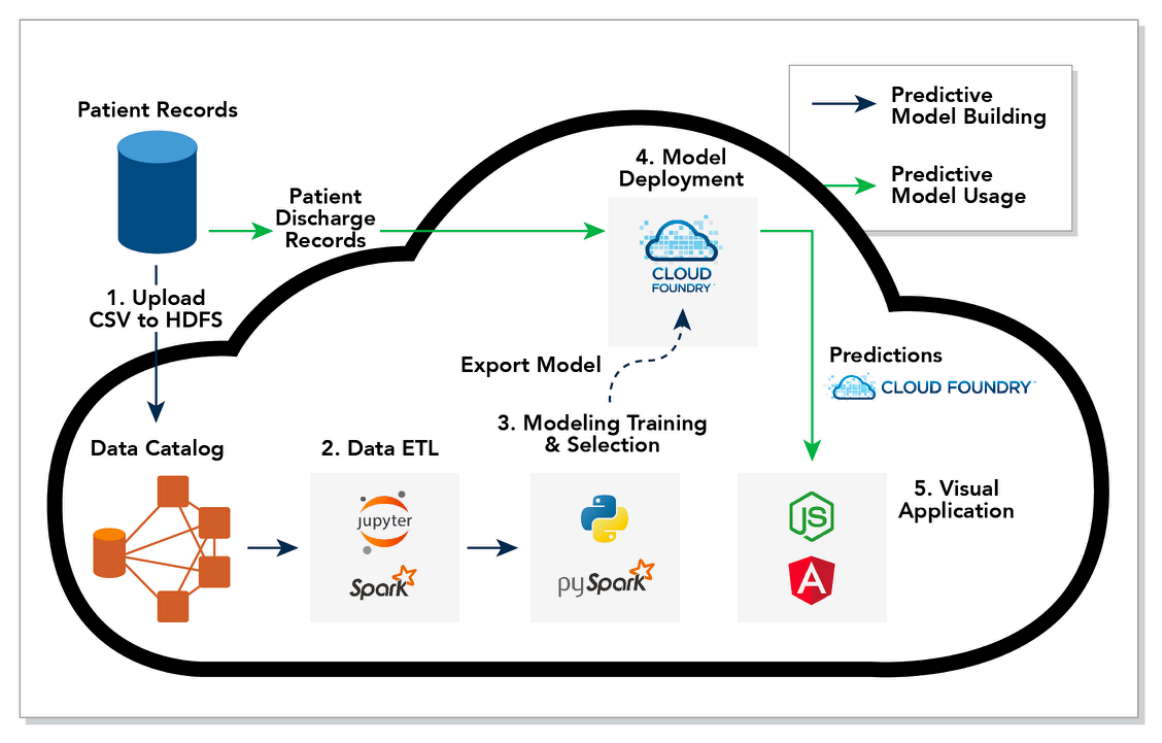
In the interim, the models can be downloaded here, or via

```
wget -O pretrained_bert_tf.tar.gz https://www.dropbox.com/s/8armk04fu16algz/pretrained_bert_tf.tar.gz?dl=1
```

All models are finetuned from the Cased BERT-Base model.

https://github.com/EmilyAlsentzer/clinicalBERT

We need SPEED! Large-scale data processing



https://github.com/ProKarma-Inc/TAP-readmissions-reduction

Large-scale data processing

Simple use-case to run ICU Mortality Prediction

A simplistic example of a spark-submit command to run the program to construct/store features, training the SVM using 70% of input data and generating AUC/ROC for testing set (30% of the input) in a local computer with Baseline (age, sex without SAPS generated from MIMIC codebase), Time-varying events (Lab, Diag, Drugs) and Notes (without Comorbidities generated from MIMIC codebase) using standard MIMIC files (keeping the MIMIC3 files name and format unchanged but can have small set of data)-

Pre-requiste: Apache Spark 2.0.0 is installed in the local machine.

Step-1: Compile the source code and create the fat .jar by running the following sbt commands -

```
a. command => cd [sourcecode]     #changing the directory to the source-code root location
b. command => sbt assembly        #Create the jar

The mortality_prediction-assembly-1.0.jar will be created at [sourcecode]/target/scala-2.11/mortality_pred
```

Step-2: Run the following command -

```
spark-submit --driver-memory 4G --executor-memory 4G \
--master local \
--class com.datalogs.main.Main [sourcecode]/target/scala-2.11/mortality_prediction-assembly-1.0.jar \
--csv-dir "[MIMIC-input-file-location]" \
--feature-dir "To-be-created-feature-location" \
--output-dir "Output-location" \
--stop-word-file "[sourcecode-location]/src/main/resources/stopWords.txt" \
--no-saps-data \
--no-comorbidities \
--pipeline-stage 0
```

The explanation of each command line parameters and output are available in the detailed section below.

https://github.com/joychak/mortality-prediction

In-Hospital Mortality Predictions with Scala & Spark on MIMIC-III

Introduction

This project is a collection of scripts to reproduce a reserach project for Georgia Tech's Online Masters in Computer Science (OMSCS) course Big Data For Health Infomatics. The scripts are intended to run in the spark-shell on an Elastic Map-Reduce(EMR) cluster hosted in Amazon Web Services (AWS) with the Medical Info-Mart Intensive Care (MIMIC) III data hosted in an AWS S3 Bucket.

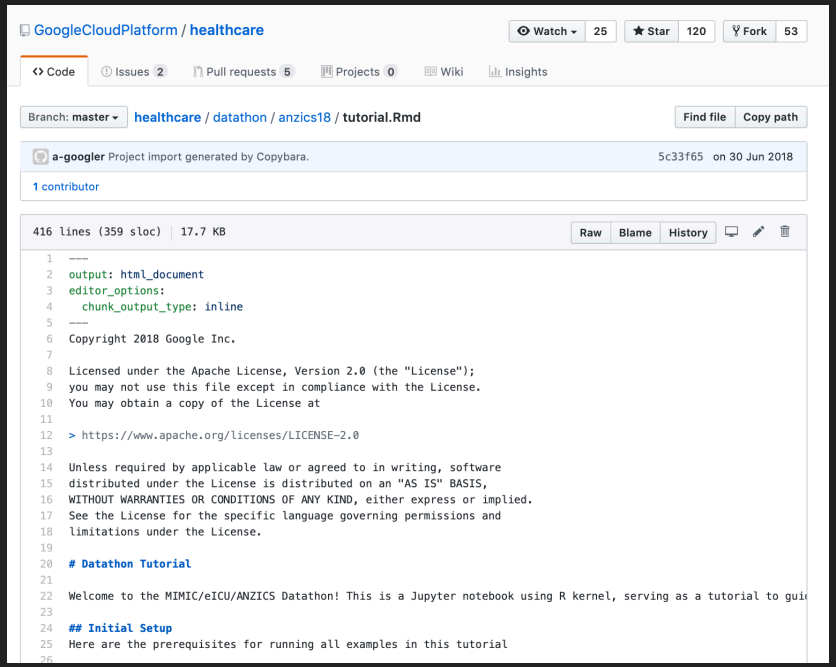
Elastic Map-Reduce on Amazon Web Services

This code was successull run several times with the MIMIC III v1.3 hosted, uncompressed, in an S3 bucket. The cluster that I used to run are scripts consisted of 1 r3.xlarge master node and 6 r3.xlarge worker nodes. The following AWS Command Line Script is what I used to start the cluster each time I successfully ran the scripts.

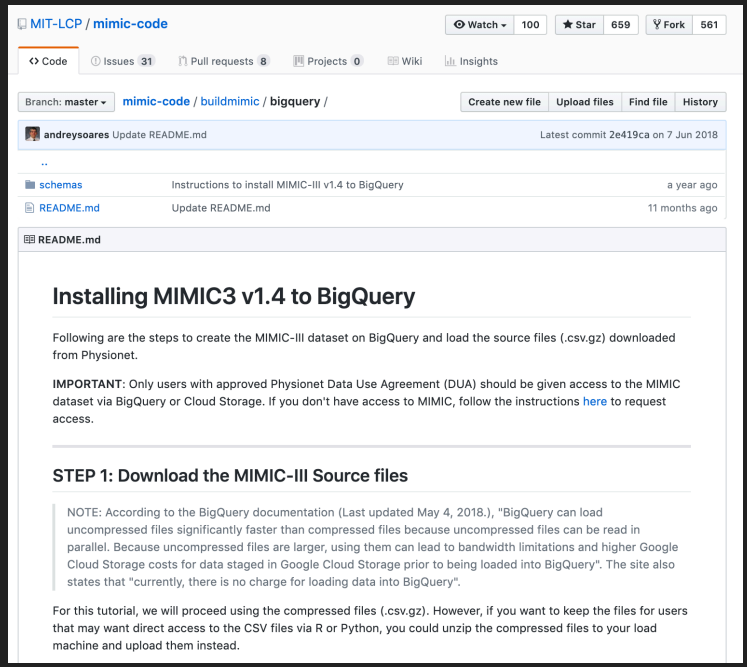
```
aws emr create-cluster --termination-protected \
--applications Name=Hadoop Name=Hive Name=Pig Name=Hue Name=Spark Name=Zeppelin-Sandbox \
--ec2-attributes '{"KeyName":"<-your-keypair-name>","InstanceProfile":"EMR_EC2_DefaultRole","EmrManagedSla
--service-role EMR_DefaultRole \
--enable-debugging \
--release-label emr-4.5.0 \
--log-uri 's3n://aws-logs-924441742886-us-east-1/elasticmapreduce/' \
--name 'MIMIC 3 Cluster' \
--instance-groups '[{"InstanceCount":1,"InstanceGroupType":"MASTER","InstanceType":"r3.xlarge","Name":"Mas
--region us-east-1
```

https://github.com/bryantravissmith/In-Hospital-Mortality-Predictions-With-Scala-on-MIMIC-III

Large-scale data processing

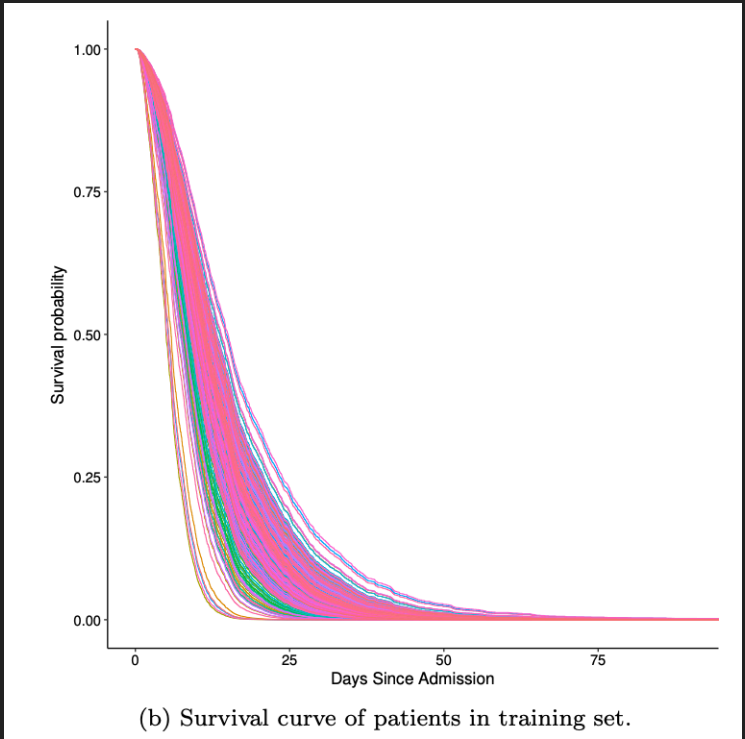


https://github.com/GoogleCloudPlatform/healthcare/blob/master/datathon/anzics18/tutorial.Rmd



https://github.com/MIT-LCP/mimic-code/tree/master/buildmimic/bigquery

Advanced Analysis (Survival Analysis)



(b) Survival curve of patients in training set.

Tanigawa and Pfhol (2017)

scikit-survival

scikit-survival is a Python module for survival analysis built on top of scikit-learn. It allows doing survival analysis while utilizing the power of scikit-learn, e.g., for pre-processing or doing cross-validation.

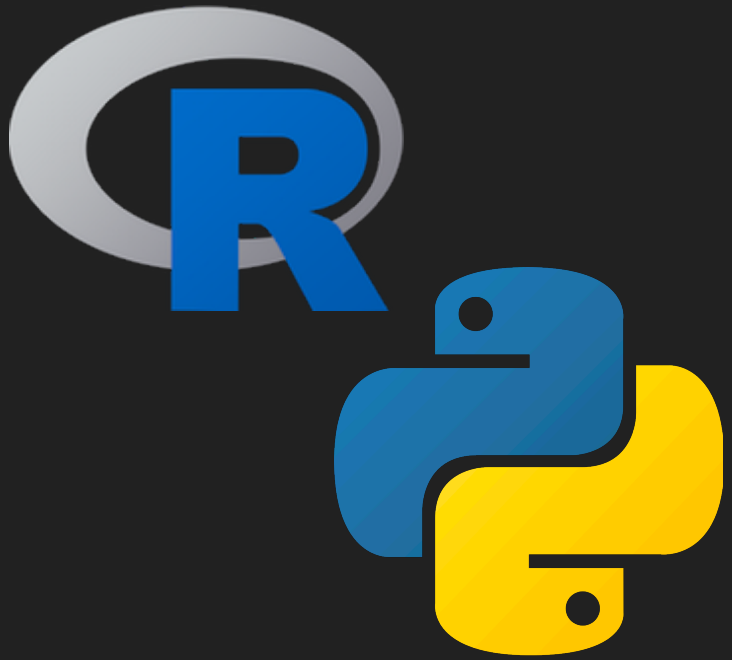
https://github.com/sebp/scikit-survival

LIFELINES

https://github.com/CamDavidsonPilon/lifelines

Conclusion

R vs. Python



MIT-LCP / **mimic-code**

Code | Issues 31 | Pull requests 8 | Projects 0 | Wiki | Insigh

Branch: master ▾ **mimic-code** / **tutorials** /

corybrunson and **alistairewj** dplyr frontend tutorial folder + intro tutorial + gitignore R project... ···

..

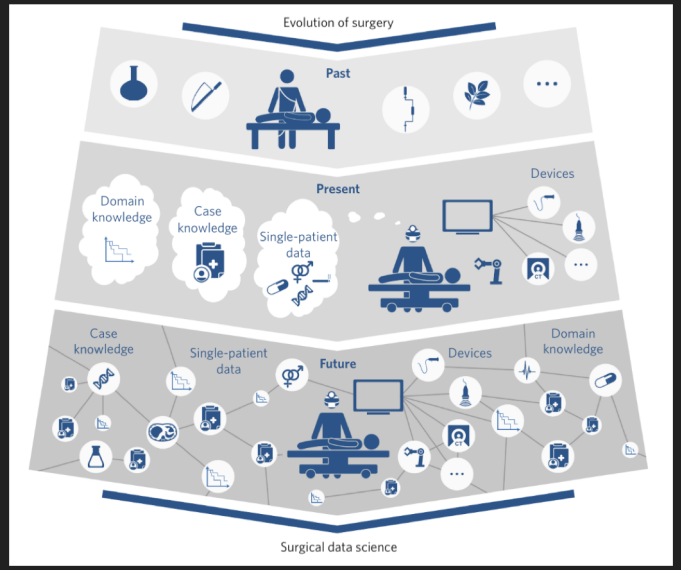
dplyr-frontend	dplyr frontend tutorial folder + intro tutorial + gitignore
README.md	add brief readme
cohort-selection.ipynb	hadm_id written as icustay_id
explore-items.Rmd	fix typo
sql-crosstab.md	hard-code columns
sql-intro.md	fix syntax errors in examples/solutions
using_r_with_jupyter.ipynb	renamed tutorial on using R+Jupyter+MIMIC-III

README.md

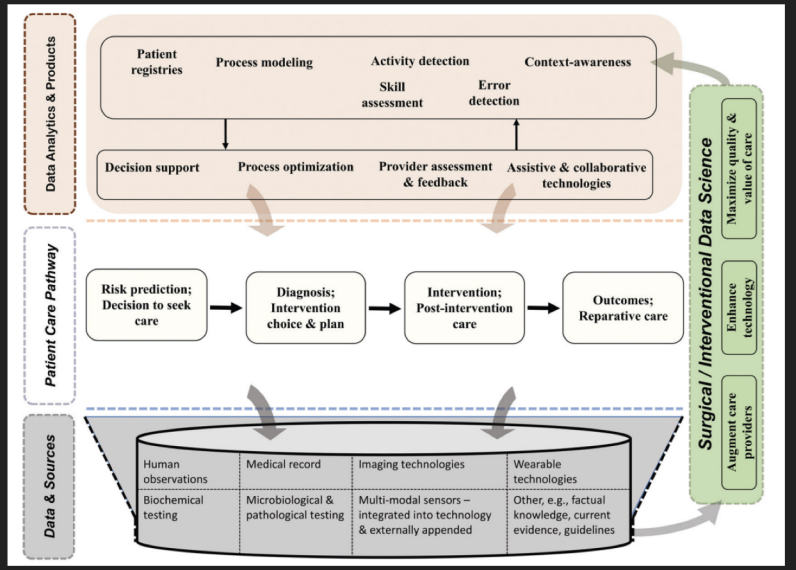
Tutorials

Surgical Data Science

The age of computer integrated surgery (CIS) with patient specific data



https://www.nature.com/articles/s41551-017-0132-7



Vedula and Hager (2017, Innov Surg Sci)