Tools and Libraries

Jupyter

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.

https://jupyter.org/

- Origin: iPython, iPython Notebook
- Open source, BSD license
- Started in 2014 by Fernando Pérez, assistant professor in the Department of Statistics at UC Berkeley
- Supported by Microsoft, Google and several foundations
- Very popular in the data analysis / data science / machine learning space
- Supports ~50 languages: Python, R, Julia, Scala, ...

Using Jupyter

Run Cells

- Run and stay at current cell: `Ctrl+Enter`
- Run and advance to next cell: `Shift+Enter`
- Run all cells in a notebook -> Menu

Manage Cells

- Switch between command and edit mode: `Enter`, `ESC`/`Ctrl+M`
- In command mode:
    - Delete cell: `dd`
    - Add cell before `a` or after `b` current cell
    - Copy cell: `c` + `v`
    - Change cell type: markdown `m`, code `y`, raw `r`

Pandas

- Python library (can be used independent of Jupyter)
- Data structues and tools for data analysis (in-memory)
- Tabluar data and time series
- Homepage: https://pandas.pydata.org/ (https://pandas.pydata.org/)
- Documentation: https://pandas.pydata.org/pandas-docs/stable/ (https://pandas.pydata.org/pandas-docs/stable/)

Pandas Data Structure: DataFrame

- Two-dimensional, like a spreadsheet or SQL table
- Rows have an *index*
- Columns have a *label* and a *data type*

In [10]:
```
df = pd.DataFrame({'somecat' : ['a', 'b', 'b', 'a'],
  'somedate': [pd.datetime.utcnow(), pd.datetime(2000,1,1), None, None],
  'somefloat' : [1.4, 1.23456789e-5, None, np.pi],
  'someint' : [-1, 0, 42, 1000],
  'sometext': ['foo', 'bar', 'baz', None]})
df
```

Out[10]:

	somecat	somedate	somefloat	someint	sometext
0	a	2019-06-24 14:03:21.035563	1.400000	-1	foo
1	b	2000-01-01 00:00:00.000000	0.000012	0	bar
2	b	NaT	NaN	42	baz
3	a	NaT	3.141593	1000	None

DataFrame: head, tail, sample

In [11]:
```
df.head(2)
```

Out[11]:

	somecat	somedate	somefloat	someint	sometext
0	a	2019-06-24 14:03:21.035563	1.400000	-1	foo
1	b	2000-01-01 00:00:00.000000	0.000012	0	bar

In [12]:
```
df.tail(2)
```

Out[12]:

	somecat	somedate	somefloat	someint	sometext
2	b	NaT	NaN	42	baz
3	a	NaT	3.141593	1000	None

In [13]:
```
df.sample(2)
```

Out[13]:

	somecat	somedate	somefloat	someint	sometext
1	b	2000-01-01 00:00:00.000000	0.000012	0	bar
0	a	2019-06-24 14:03:21.035563	1.400000	-1	foo

Data Import

- Pandas has many `read_*` functions for importing data into a DataFrame
    - CSV, SQL (query or table), JSON, XML
- CSV:
    - Tries to automagically detect: separators, column names, data types
    - Fails sometimes, but can be defined explicitely
    - Parsing of dates needs to be defined explicitely
- SQL:
    - Requires connection to database

In [ ]:
```
df = pd.read_csv('data/demo.csv')
```

scikit-learn

- Tools for data mining and statistical machine learning
- Built on NumPy, SciPy, and matplotlib
- Open source https://scikit-learn.org/stable/index.html (https://scikit-learn.org/stable/index.html)

Important classification functions in sklearn

- `fit(x_train, y_train)` : trains the model on the data
- `predict(x_test)` : shows the predicted outcome
- `predict_proba(x_test)` : shows the probability for the predicted outcome
- `score(x_test, y_test)` : gives us a value max 1.0 (which is the perfect score) for the performance of our model

Other libraries

There are a lot of useful python libraries for data analysis and machine learning. We will work with:

- **numpy**: library adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays
- **matplotlib** and **seaborn**: plotting libraries

Default imports

In [8]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import sklearn as sklearn
```

Datasets we are working with

- Iris flower Dataset
    - Part of scikit-learn
    - 3 classes of flowers
    - 4 dimensions: sepal length (cm), sepal width (cm), petal length (cm), petal width (cm)
- Heart Disease Data Set
    - From the UC Irvine Machine Learning Repository
    - Details: https://archive.ics.uci.edu/ml/datasets/heart+Disease (https://archive.ics.uci.edu/ml/datasets/heart+Disease)

Attribute Information Heart Disease Dataset

1. *age*
2. *sex*
3. *chest pain type (4 values)*
4. *resting blood pressure*
5. *serum cholestoral in mg/dl*
6. *fasting blood sugar > 120 mg/dl*
7. *resting electrocardiographic results (values 0,1,2)*
8. *maximum heart rate achieved*
9. *exercise induced angina*
10. *oldpeak = ST depression induced by exercise relative to rest*
11. *the slope of the peak exercise ST segment*
12. *number of major vessels (0-3) colored by flourosopy*
13. *thal: 3 = normal; 6 = fixed defect; 7 = reversable defect*

Exercise Jupyter

Goals:

- Getting familiar with Jupyter

Tasks:

- Create a notebook file, create some code cells, write some Python code, and execute it
- Create a markdown cell
- Try some shortcuts:
    - Execute a cell: `Ctrl+Enter` and `Shift+Enter`
    - Create a cell before `a` or after `b`
    - Copy `c` and paste `v` a cell
- Try code completion
    - `Tab`

Exercise Pandas (1)

Goals:

- Get an idea what the dataset is about
- Getting familiar with pandas

Tasks:

- Import `data/heart.csv` as pandas dataframe
- How many tupels and how many attributes do we have?
- Set meaningful column names (See https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html (https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.html))
- Look at the top lines of the data table
- A useful function for a statistical overview is `describe()`. Try it for your dataframe
- Plot the outcome with Seaborn (See: https://seaborn.pydata.org/generated/seaborn.countplot.html (https://seaborn.pydata.org/generated/seaborn.countplot.html))

Exercise Pandas (2)

Goals:

- Preparing data for working with scikit-learn

Tasks:

- Separate input data and output, common variables are x for input, y for output.