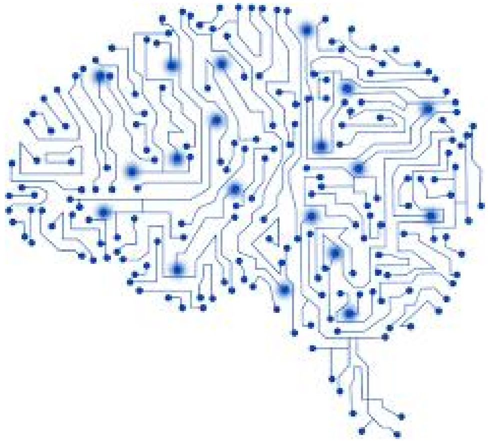
University of Minho
School of Engineering



Dados e Aprendizagem Automática

Intro to Data Science & Python/Scikit-learn

DAA @ MEI-1º/MiEI-4º – 1º Semestre

César Analide, Bruno Fernandes, Filipa Ferraz, Filipe Gonçalves, Victor Alves

Part I

Contents

- Data Types
- Mean, Median & Mode
- Standard Deviation & Variance
- Probability Density Functions
- Percentiles
- Covariance & Correlation
- Virtual Environment
- Environment Setup
- Hands On

Data Types

Data Types

- Major Types of Data:
  - Numerical
  - Categorical
  - Ordinal

Data Types

Numerical

- Represents some sort of quantitative measurement
  - Heights of people, page load times, stock prices, etc.

- Discrete Data
  - Integer based; often counts of some event.
    - How many purchases did a customer make in a year?
    - How many times did I flip “heads”?

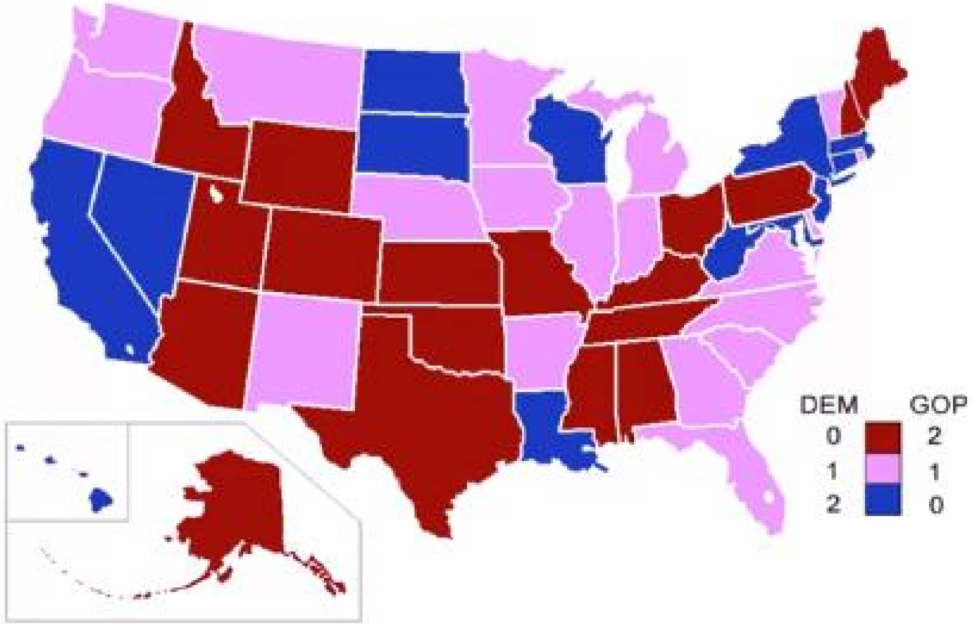
- Continuous Data
  - Has an infinite number of possible values
    - How much time did it take for a user to check out?
    - How much rain fell on a given day?

Data Types

Categorical

- Qualitative data that has no inherent mathematical meaning
  - Gender, Yes/No (Binary Data), Race, State of Residence, Product Category, Political Party, etc.

- You can assign numbers to categories in order to represent them more compactly, but the numbers don't have mathematical meaning



Data Types

Ordinal

- A mixture of numerical and categorical
- Categorical data that has mathematical meaning
- Example: movie ratings on a 1-5 scale.
  - Ratings must be 1,2,3,4 or 5
  - These values have mathematical meaning; 1 means it's a worse movie than a 2.

Data Types

Quick Quiz:

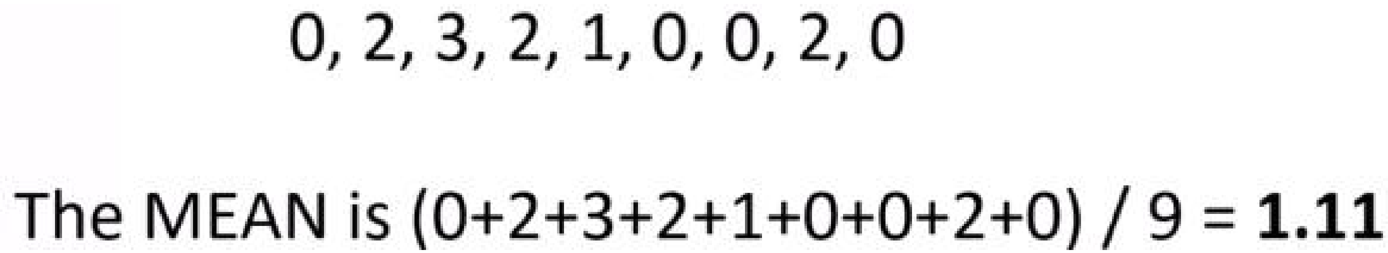
- Are the following types of data numerical, categorical, or ordinal?
  - How much gas is in your gas tank?
  - A rating of your overall health where the choices are 1,2,3 or 4, corresponding to “poor”, “moderate”, “good” and “excellent”
  - The nationalities of your classmates
  - Ages in years
  - Money spent in a store

Mean, Median & Mode

Mean, Median & Mode

Mean

- □ AKA Average
- □ Sum / number of samples
- □ Example:
  - ○ Number of children in each house on my street:



0, 2, 3, 2, 1, 0, 0, 2, 0

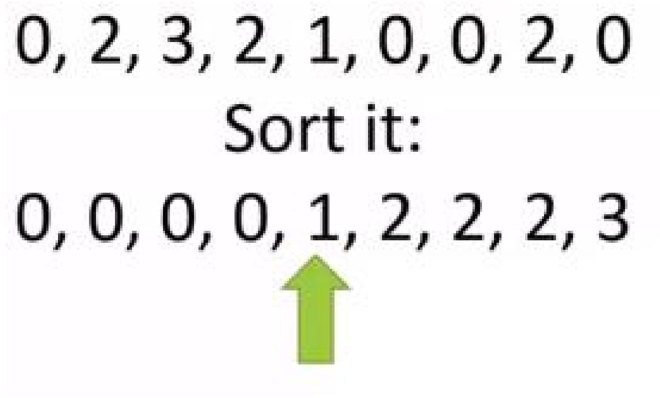
The MEAN is (0+2+3+2+1+0+0+2+0) / 9 = **1.11**

Mean, Median & Mode

Median

□ Sort the values, and take the value at the midpoint.

□ Example:



0, 2, 3, 2, 1, 0, 0, 2, 0

Sort it:

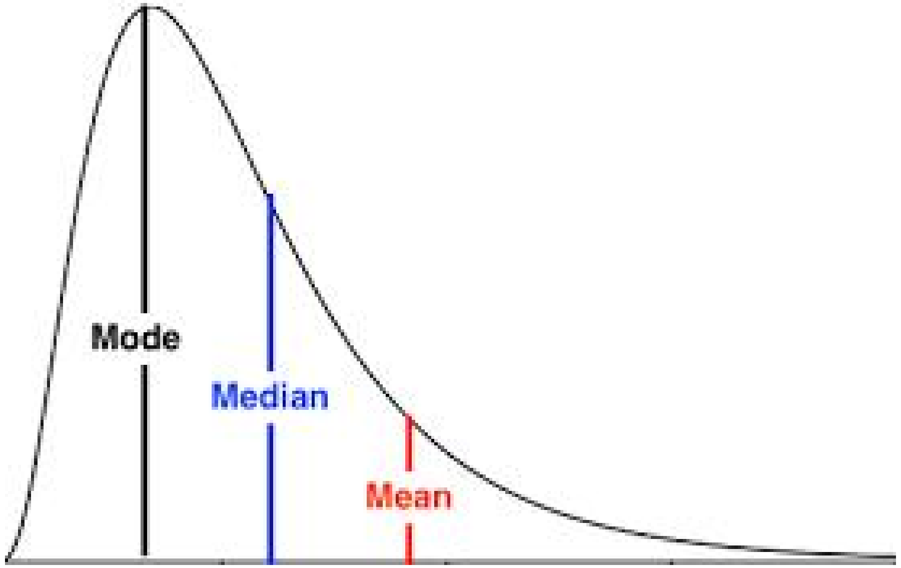
0, 0, 0, 0, 1, 2, 2, 2, 3

□ If you have an even number of samples, take the average of the two in the middle.

Mean, Median & Mode

Median

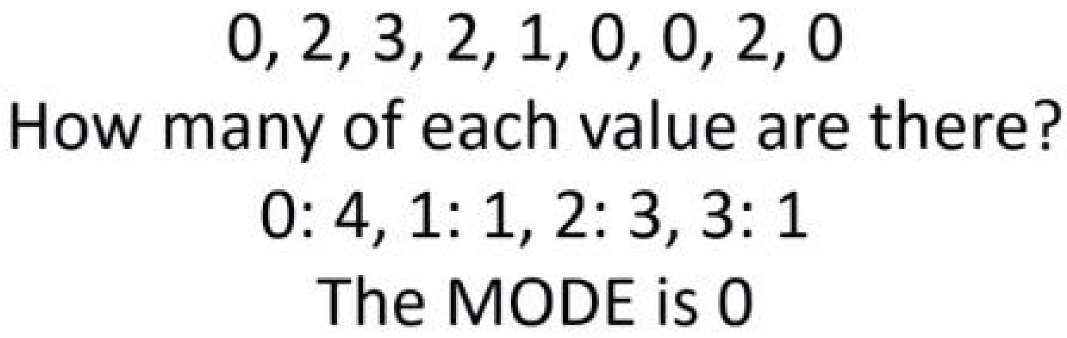
- Median is less susceptible to outliers than the mean
  - Example: mean household income in the USA is $72,641, but the median is only $51,939 – because the mean is skewed by a handful of billionaires.
  - Median represents better the “typical” American in this example.



Mean, Median & Mode

Mode

- The most common value in a dataset
  - Not relevant to continuous numerical data
- Number of kids in each house example:



0, 2, 3, 2, 1, 0, 0, 2, 0

How many of each value are there?

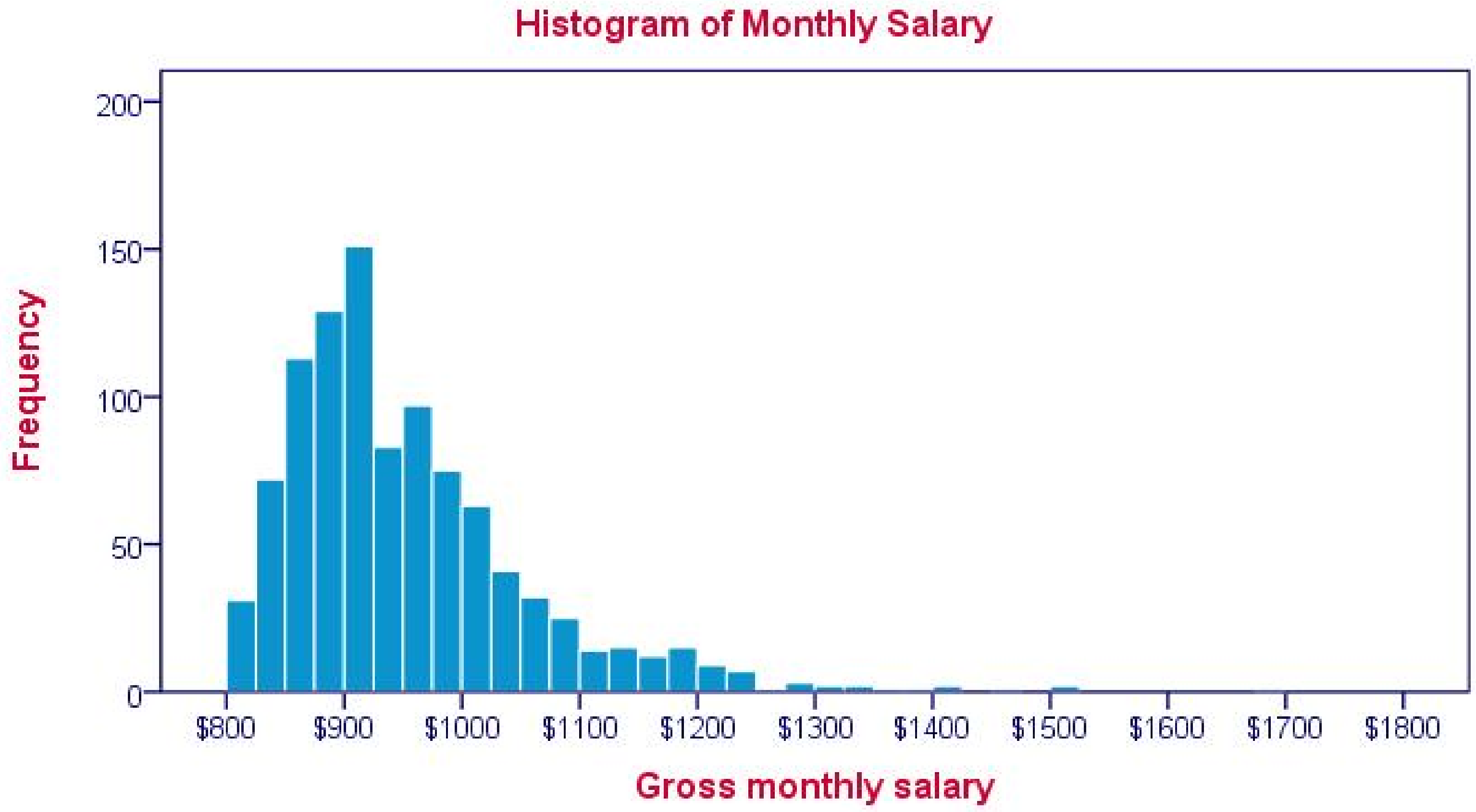
0: 4, 1: 1, 2: 3, 3: 1

The MODE is 0

Standard Deviation & Variance

Standard Deviation & Variance

An example of a histogram…



Standard Deviation & Variance

Variance measures how "spread-out" the data is.

- Variance (δ^2) is simply the average of the squared differences from the mean

- Example: What is the variance of the data set (1, 4, 5, 4, 8)?
  - First find the mean: (1+4+5+4+8) / 5 = 4.4
  - Now find the difference from the mean: (-3.4, -0.4, 0.6, -0.4, 3.6)
  - Find the squared differences: (11.56, 0.16, 0.36, 0.16, 12.96)
  - Find the average of the squared differences:
  - $\delta^2 = (11.56+0.16+0.36+0.16+12.96) / 5 = 5.04$

Standard Deviation & Variance

Standard Deviation δ is the square root of the variance.

□ Case Study = (1,4,5,4,8)

□ Mean = 4.4

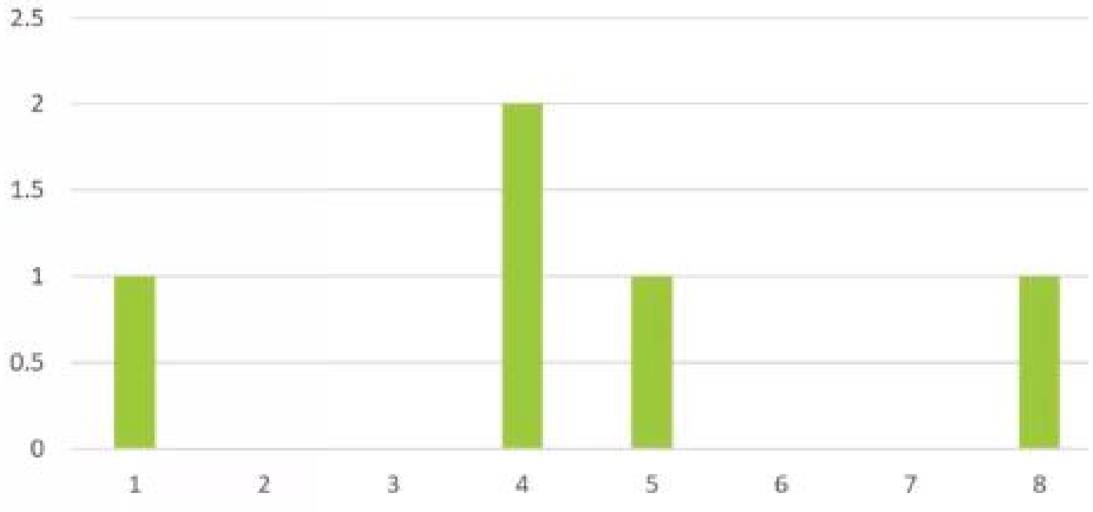
□ $\delta^2 = 5.04$

□ $\delta = 2.24$

□ Stand. Dev. Is usually used as a way to identify outliers.

□ Data points that lie more than one standard deviation from the mean can be considered unusual.

□ You can talk about how extreme a data point is by talking about "how many sigmas" away from the mean it is.

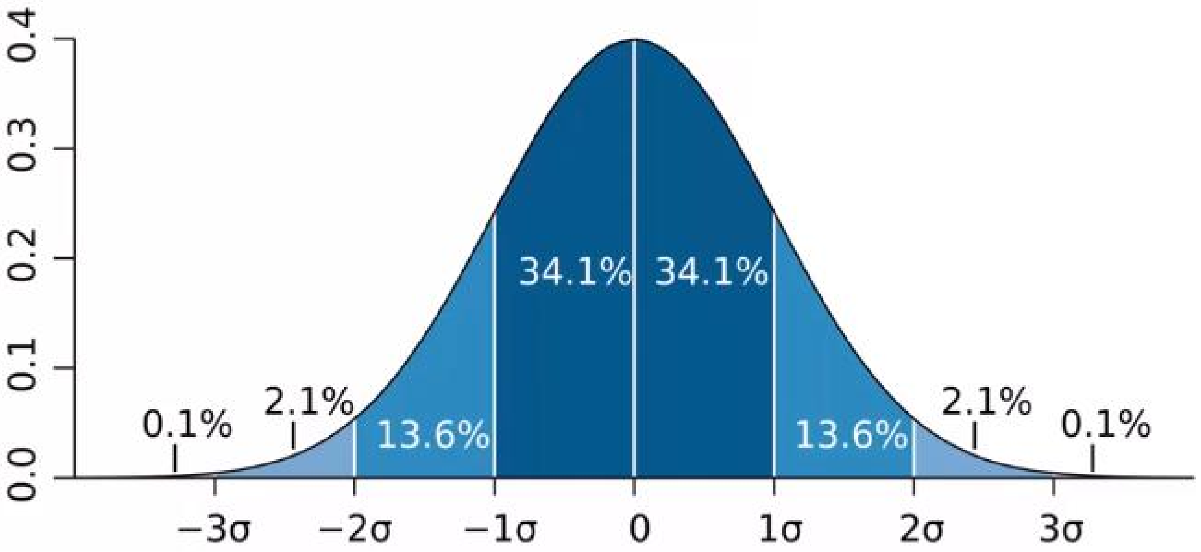


Probability Density Functions

Probability Density Functions

Example: a "normal distribution"

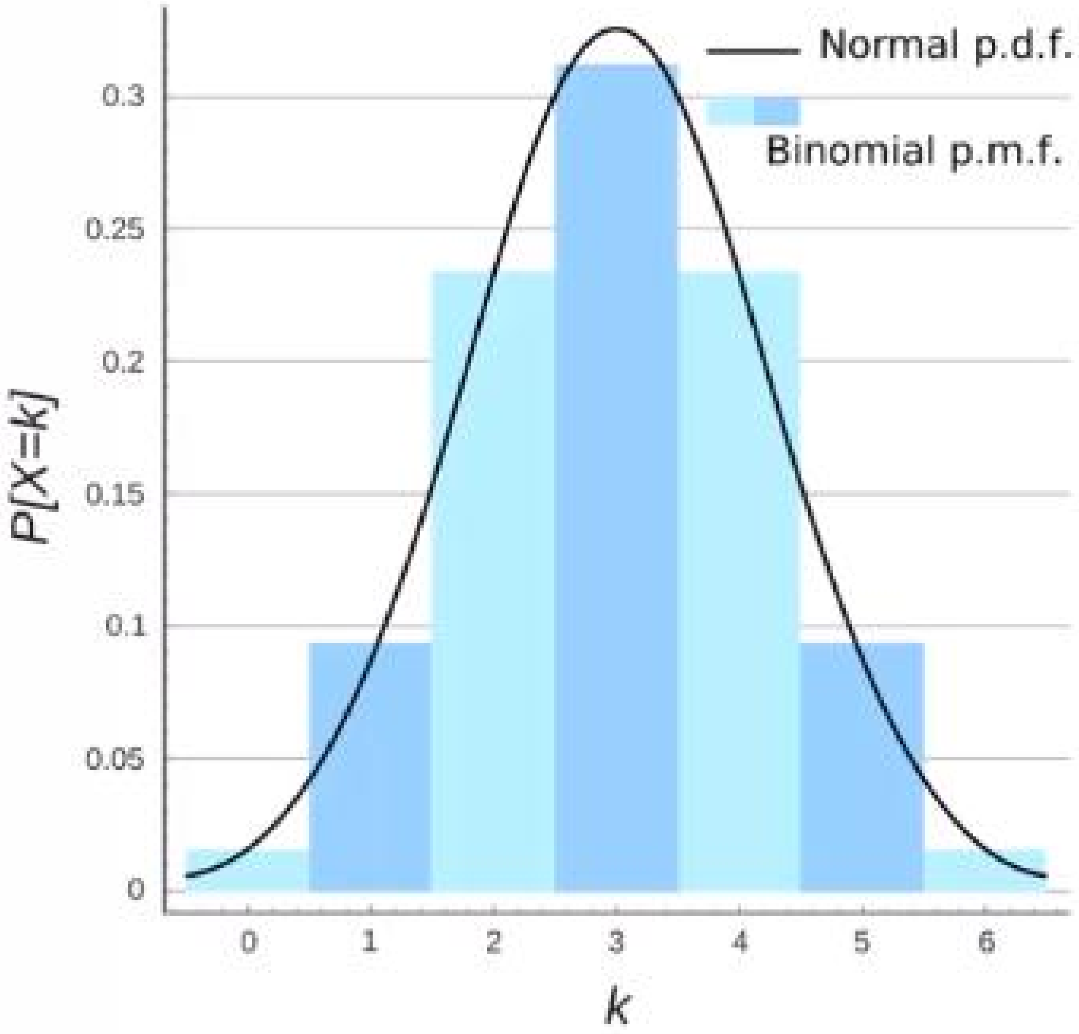
- Gives you the probability of a data point falling within some given range of a given value
- Based on histogram values, a normal probability density function can be calculated



Probability Density Functions

Example: Probability Mass Function

- Used for discrete data
- Based on histogram values, a normal probability density function can be calculated

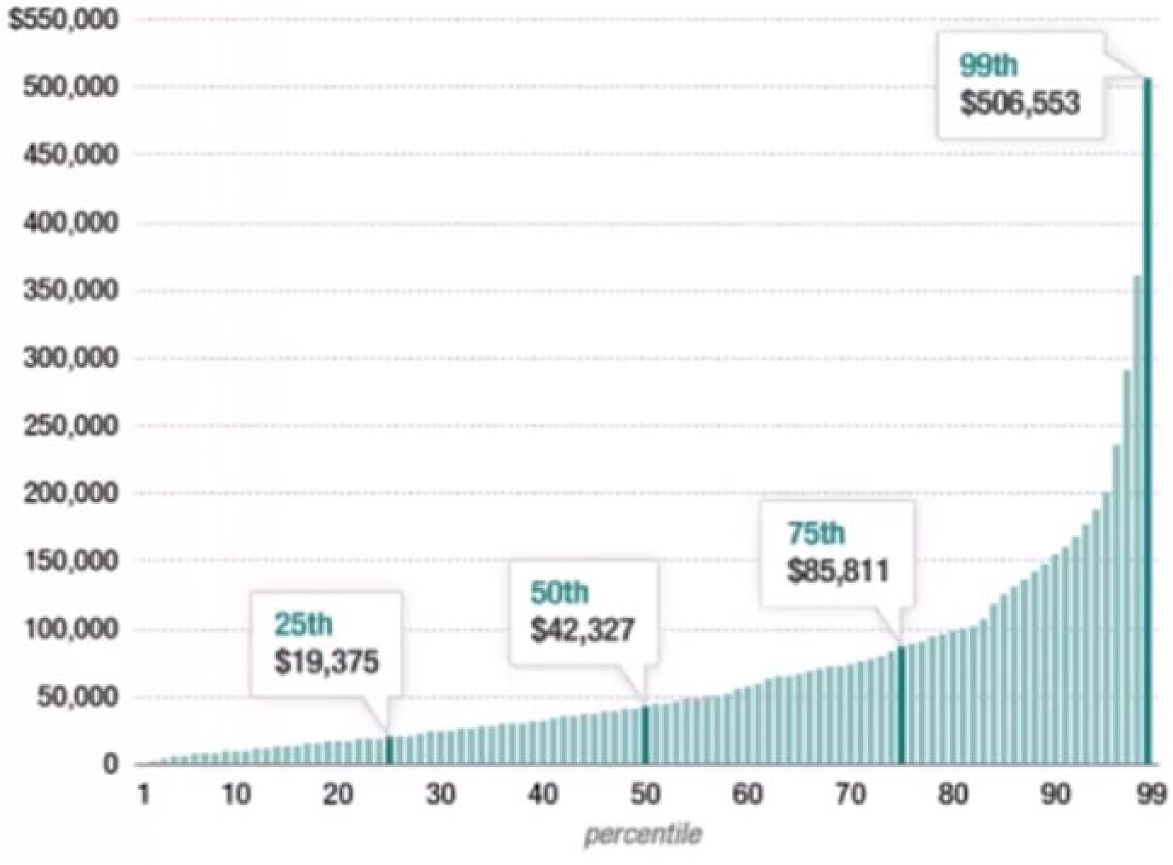


Percentiles

Percentiles

Percentiles

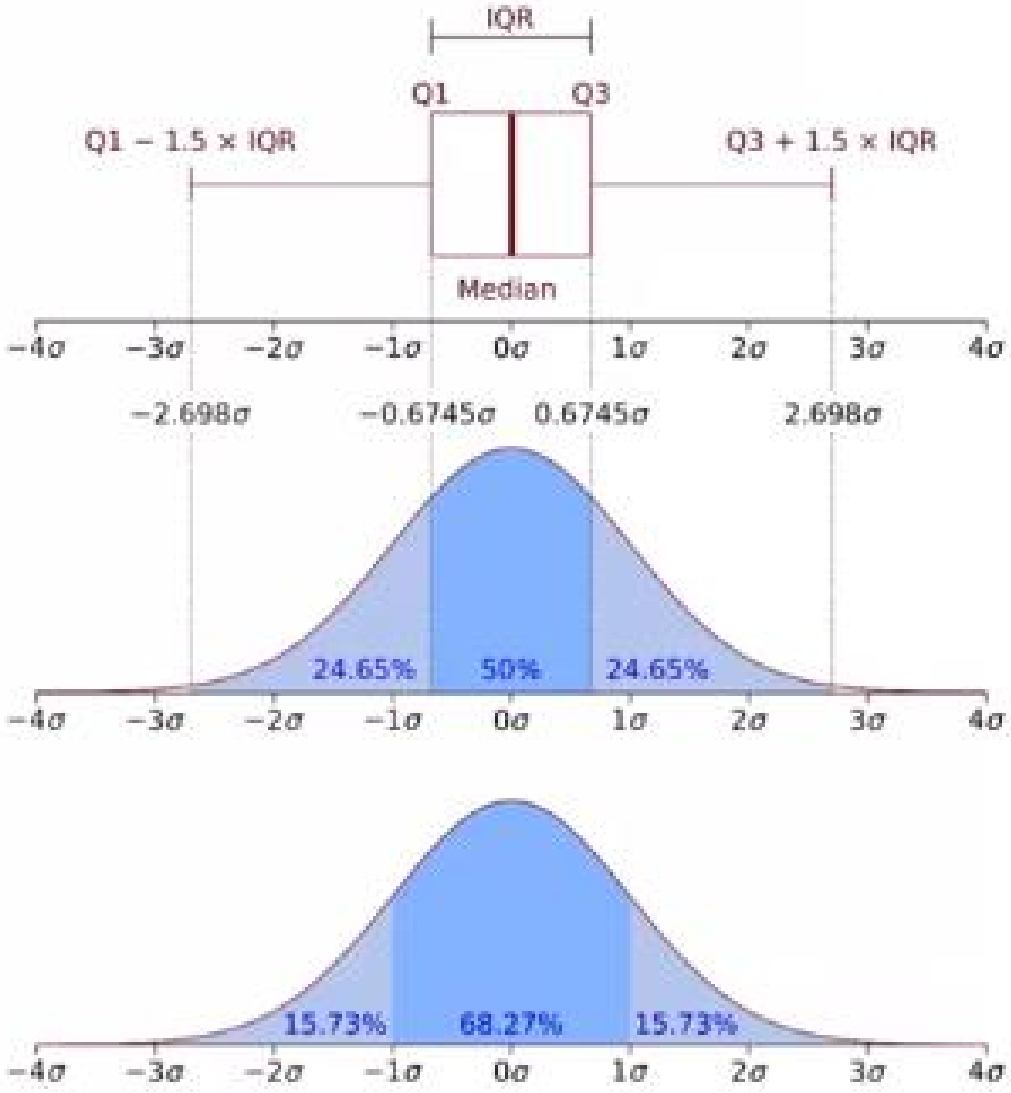
- In a dataset, what's the point at which X% of the values are less than that value?
- Example: income distribution
  - Take all incomes from a country's population and sort them
  - 99th percentile represents the income amount in which 99% of the population gains less then that value (i.e., $506,553)



Percentiles

Percentiles in a normal distribution

- □ Between Quartil 1 & Quartil 3 represents 50% of the data distribution
- □ IQR (Inter-Quartil Range) represents the area in the middle of the distribution (where data is more focused)

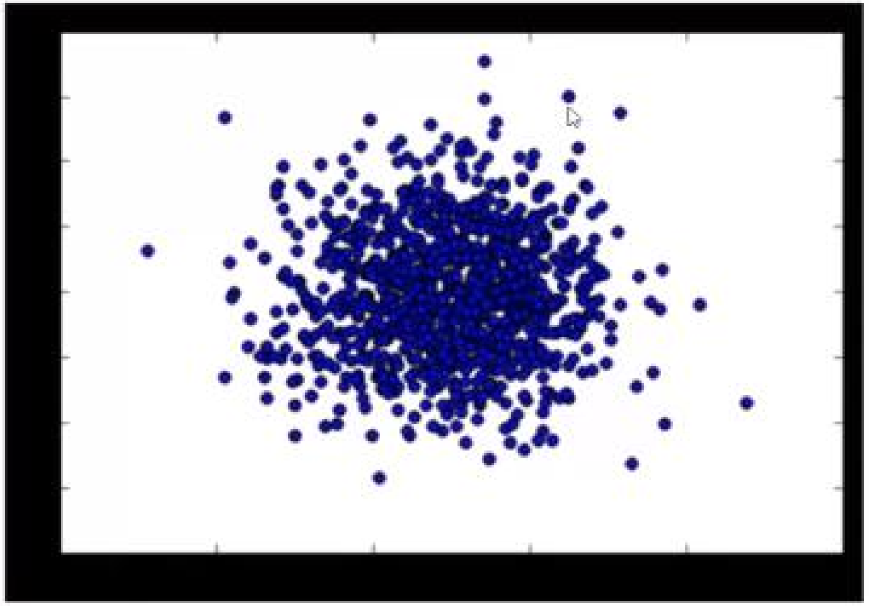


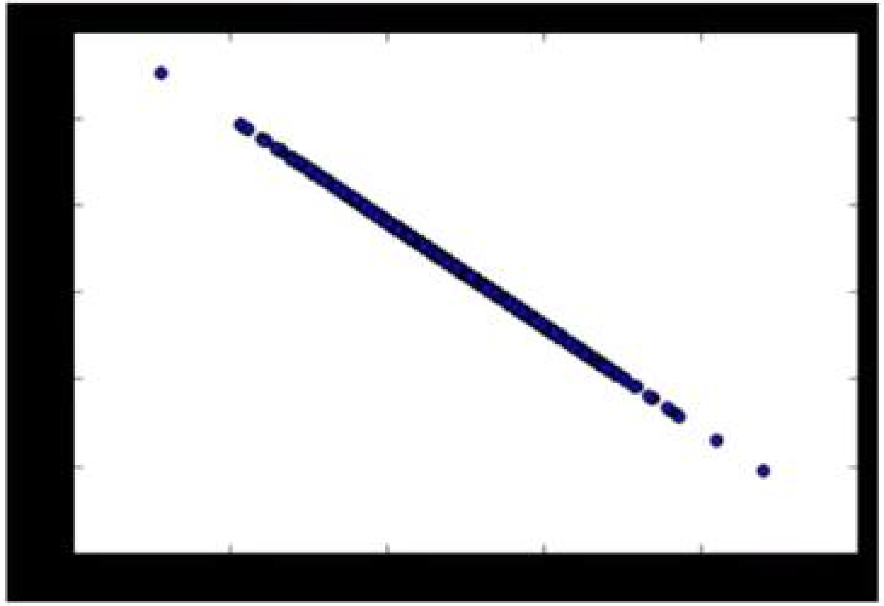
Covariance & Correlation

Covariance & Correlation

Covariance

- □ Measures how two variables vary in tandem from their means.
- □ i.e. how 2 attributes depend on each other (left plot – low covariance / right plot – high covariance)

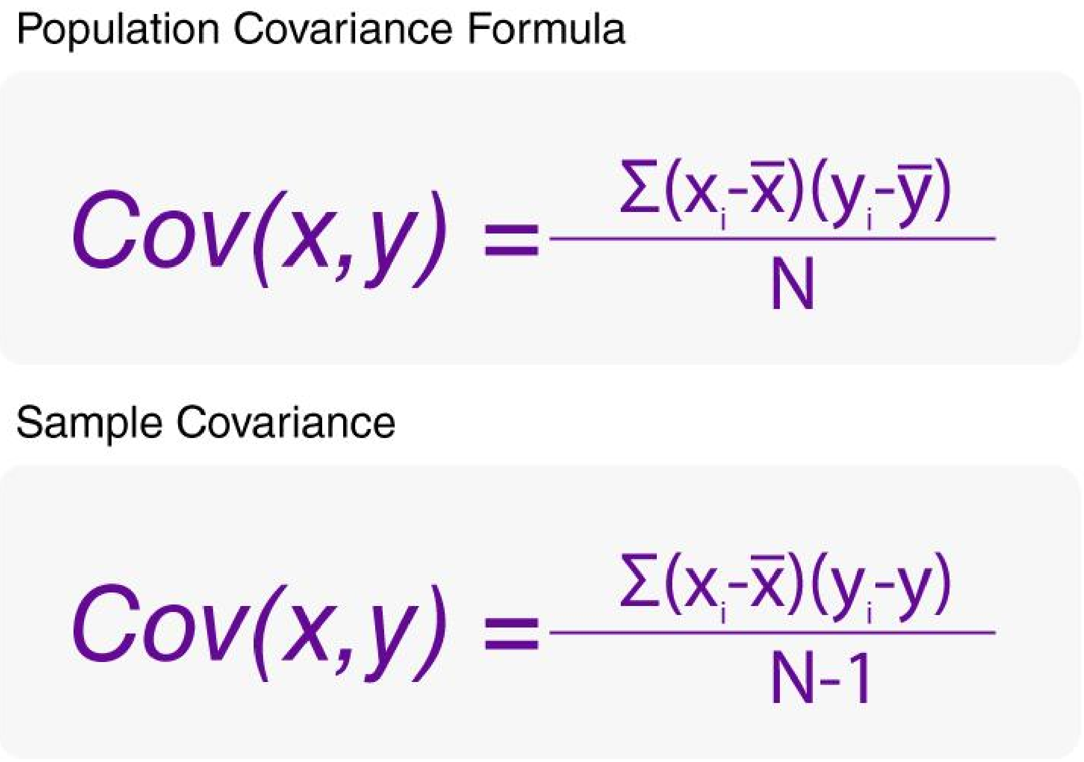




Covariance & Correlation

Measuring covariance

- □ Think of the datasets for the two variables as high-dimensional vectors
- □ Convert these to vectors of variances from the mean
- □ Take the dot product (cosine of the angle between them) of the two vectors
- □ Divide by the population size



Population Covariance Formula

$$Cov(x,y) = \frac{\Sigma(x_i-\bar{x})(y_i-\bar{y})}{N}$$

Sample Covariance

$$Cov(x,y) = \frac{\Sigma(x_i-\bar{x})(y_i-y)}{N-1}$$

Covariance & Correlation

Interpreting covariance is hard

- □ Small covariance (close to 0) means there isn't much correlation between the two variables
- □ Large covariance (far from 0 – can be negative for inverse relationships) means that there is a correlation

Interpreting correlation is easier

- □ Normalization value of covariance divided by the standard deviations of both variables
  - ○ Correlation of -1: perfect inverse correlation
  - ○ Correlation of 0: no correlation
  - ○ Correlation of 1: perfect correlation

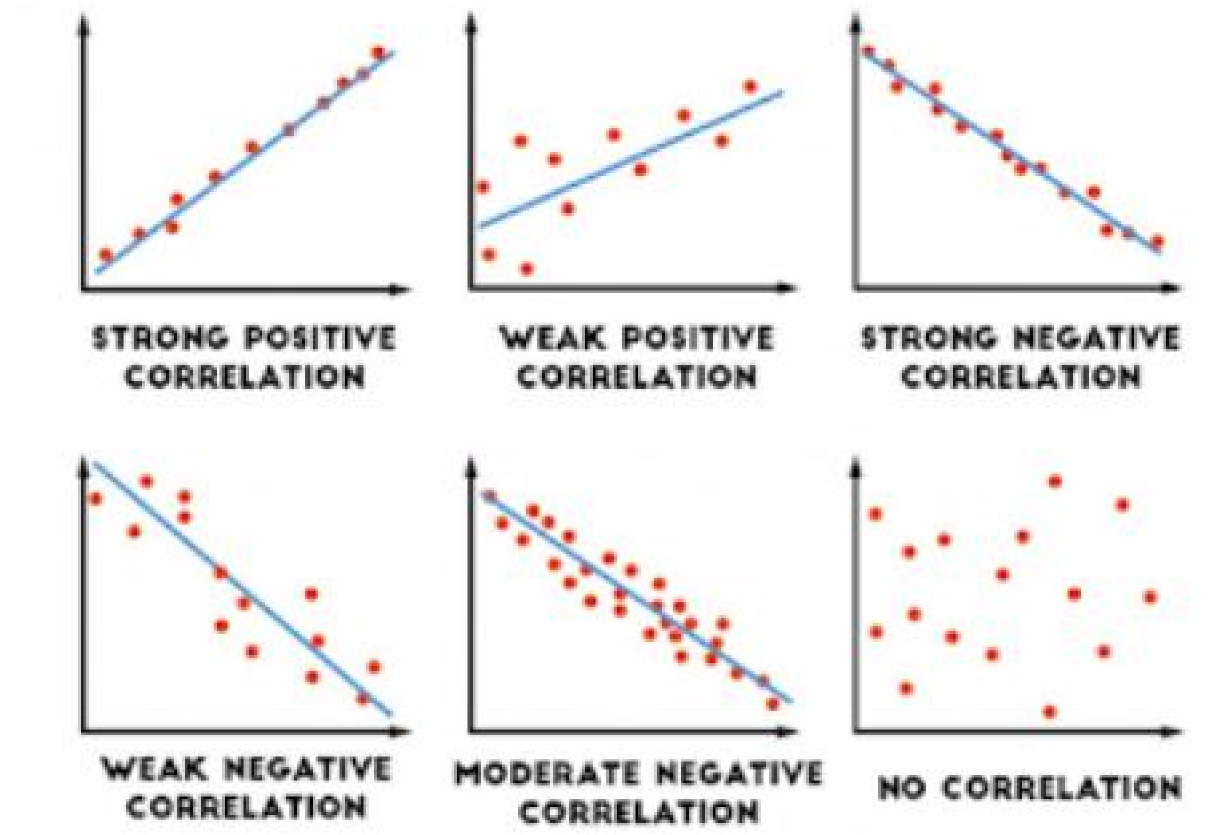
Covariance & Correlation

Correlation does not imply causation!

□ Only a controlled, randomized experiment can give you insights on causation.

□ Use correlation to decide what experiments to conduct!

Covariance & Correlation



Virtual Environments

Virtual Environments

- Virtual Environments allow you to set up virtual installations of Python and libraries on your computer
- You can have multiple versions of Python or libraries and easily activate or deactivate these environments
- Let’s see some examples of why you may want to do this

Virtual Environments

- Sometimes you’ll want to program in different versions of a library
- For example:
  - You develop a program with SciKit-Learn 0.17
  - SciKit-Learn 0.18 is released
  - You want to explore 0.18 but don’t want your old code to break

- Sometimes you’ll want to make sure your library installations are in the correct location

- For example:
  - You want multiple versions of Python on your computer
  - You want one environment with Python 2.7 and another with Python 3.6

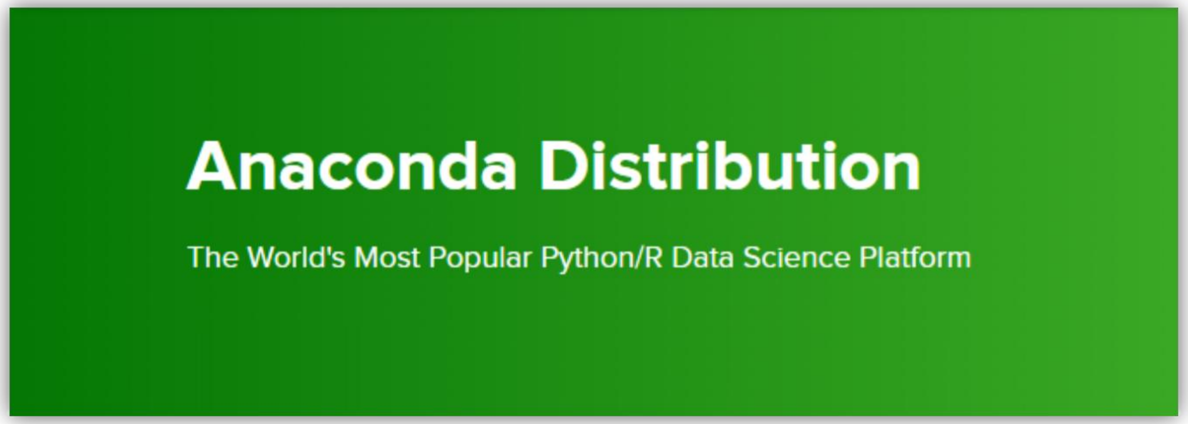
Virtual Environments

- Anaconda has a built-in virtual environment manager that makes the whole process really easy
- Check out the resource link for the official documentation:
    - http://conda.pydata.org/docs/using/envs.html

- Command Prompt Example (create env. and activate it):

```
conda create --name mypython3version python=3.6 numpy
conda info --evns
activate mypython3version
python
import numpy as np
import pandas as pd       -> Error
quit()
conda install pandas
deactivate
```

ANACONDA®



Anaconda Distribution

The World's Most Popular Python/R Data Science Platform

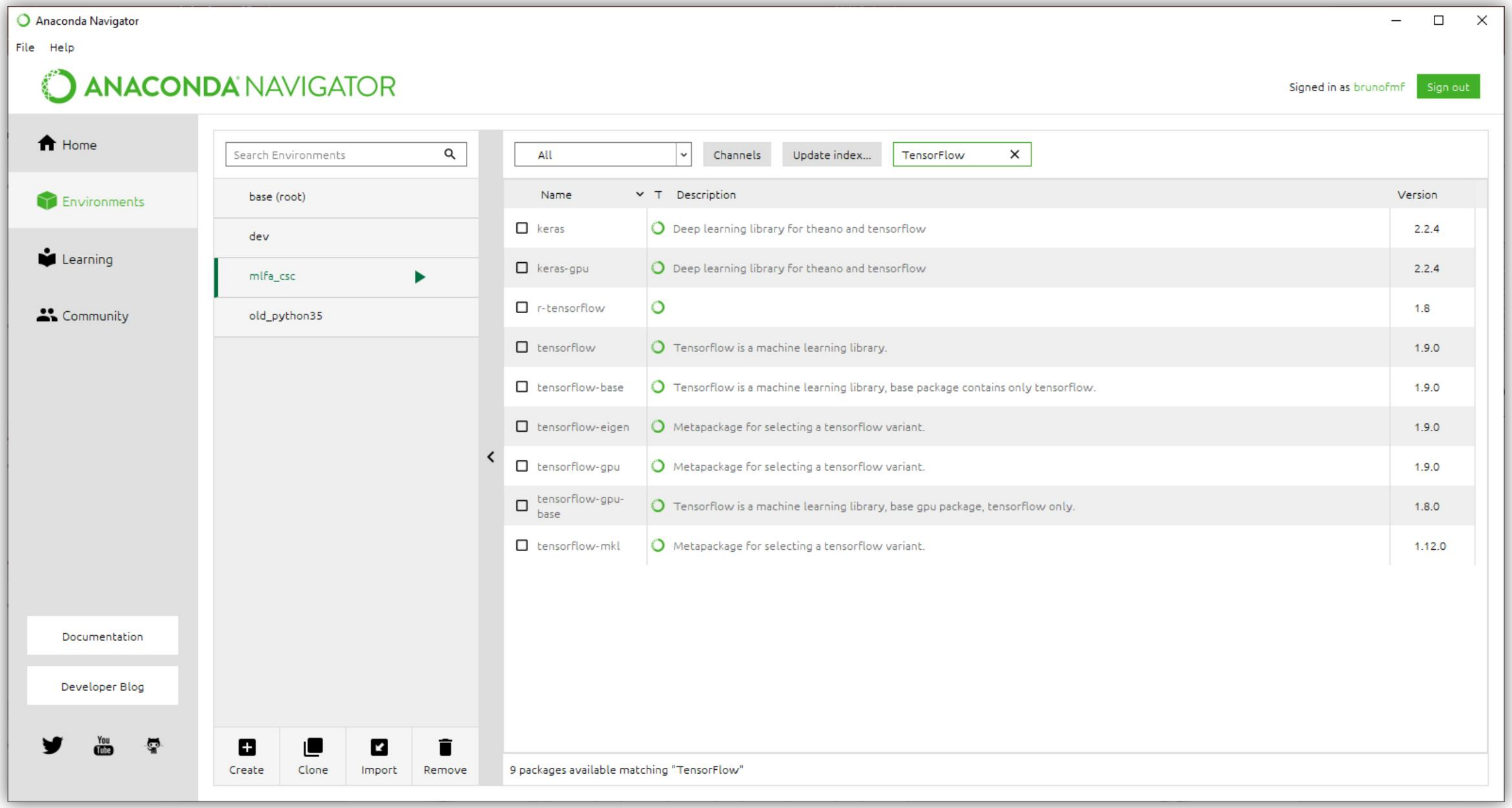
jupyter
spyder
NumPy
SciPy
Numba
pandas
DASK
Bokeh
HoloViews
Datashader
matplotlib
scikit learn
H2O.ai
TensorFlow
CONDA

ANACONDA

- FOSS
- Share, collaborate on, and reproduce projects
- Highly supported by the community
- Conda, a package, dependency and environment manager
  - Easily create, save, load and switch between environments
  - Easily install, update and run any package (and its dependencies… automatically!)

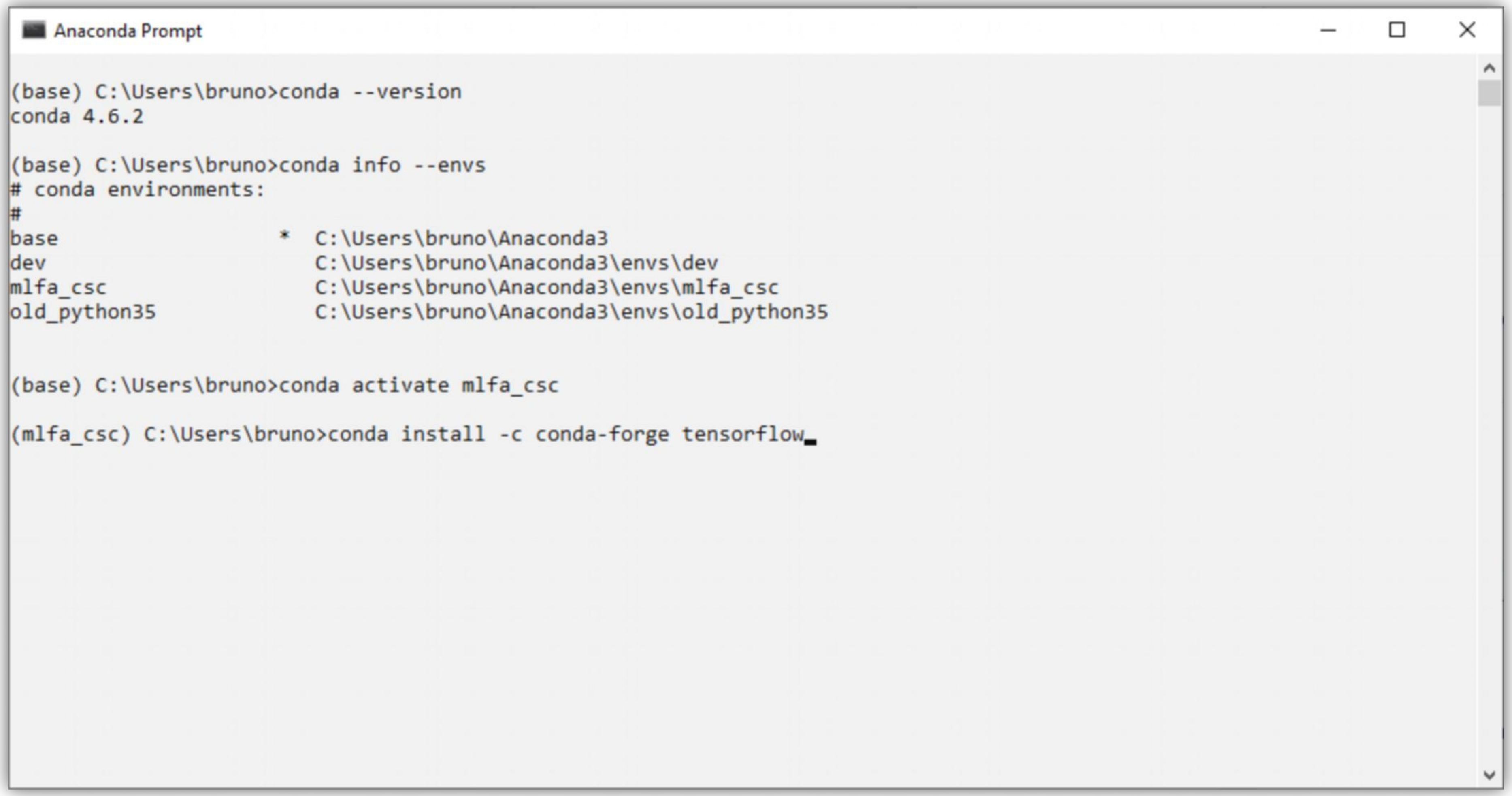
- Anaconda provides two user clients
  - Anaconda Navigator
  - Anaconda Prompt (or the terminal on Linux and macOS)

ANACONDA®



	Name	T	Description	Version
□	keras	○	Deep learning library for theano and tensorflow	2.2.4
□	keras-gpu	○	Deep learning library for theano and tensorflow	2.2.4
□	r-tensorflow	○		1.8
□	tensorflow	○	Tensorflow is a machine learning library.	1.9.0
□	tensorflow-base	○	Tensorflow is a machine learning library, base package contains only tensorflow.	1.9.0
□	tensorflow-eigen	○	Metapackage for selecting a tensorflow variant.	1.9.0
□	tensorflow-gpu	○	Metapackage for selecting a tensorflow variant.	1.9.0
□	tensorflow-gpu-base	○	Tensorflow is a machine learning library, base gpu package, tensorflow only.	1.8.0
□	tensorflow-mkl	○	Metapackage for selecting a tensorflow variant.	1.12.0

9 packages available matching "TensorFlow"



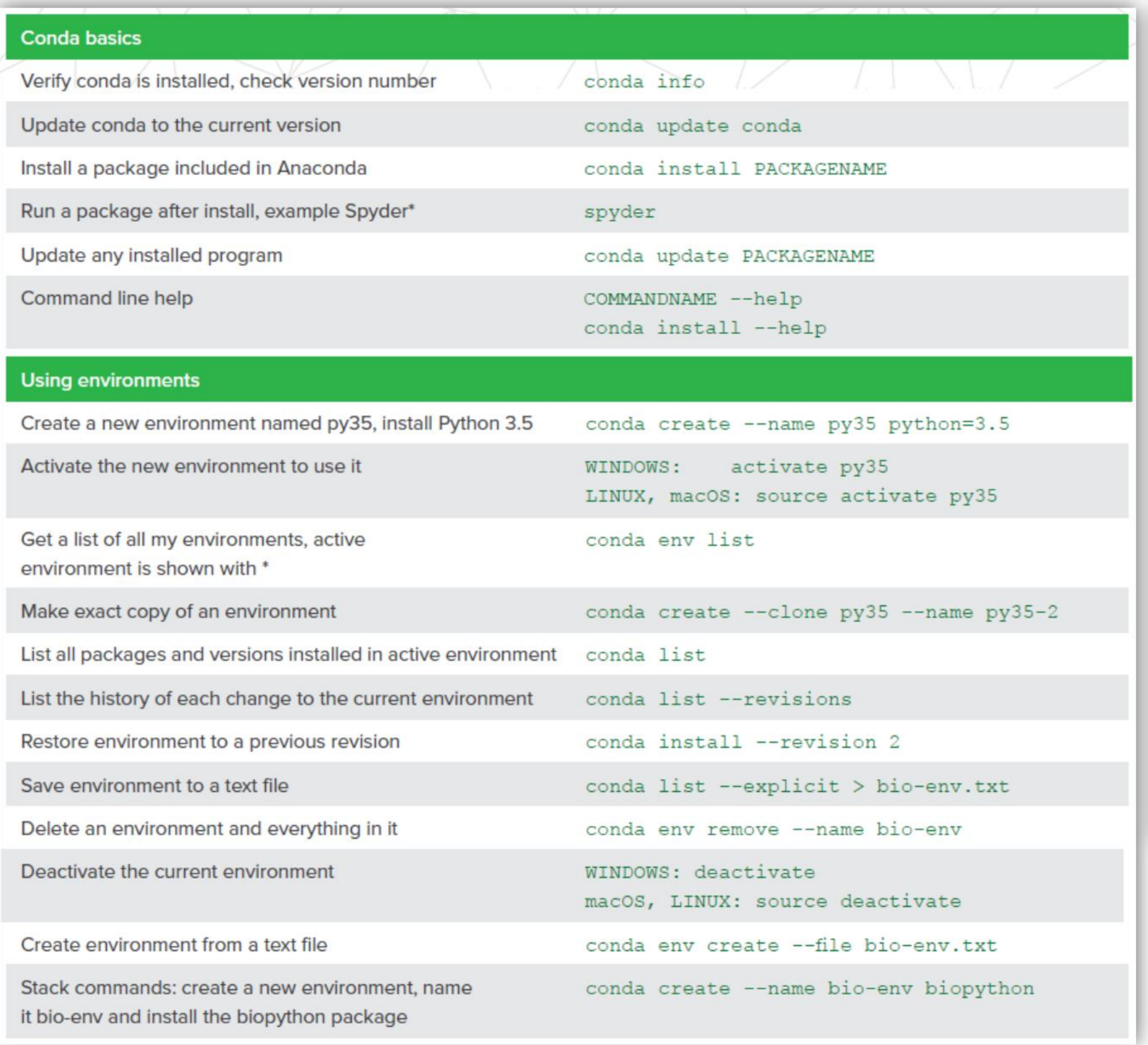
```
(base) C:\Users\bruno>conda --version
conda 4.6.2

(base) C:\Users\bruno>conda info --envs
# conda environments:
#
base                  *  C:\Users\bruno\Anaconda3
dev                      C:\Users\bruno\Anaconda3\envs\dev
mlfa_csc                 C:\Users\bruno\Anaconda3\envs\mlfa_csc
old_python35             C:\Users\bruno\Anaconda3\envs\old_python35


(base) C:\Users\bruno>conda activate mlfa_csc

(mlfa_csc) C:\Users\bruno>conda install -c conda-forge tensorflow
```

ANACONDA



Conda basics	
Verify conda is installed, check version number	`conda info`
Update conda to the current version	`conda update conda`
Install a package included in Anaconda	`conda install PACKAGENAME`
Run a package after install, example Spyder*	`spyder`
Update any installed program	`conda update PACKAGENAME`

Command line help	`COMMANDNAME --help`
	`conda install --help`

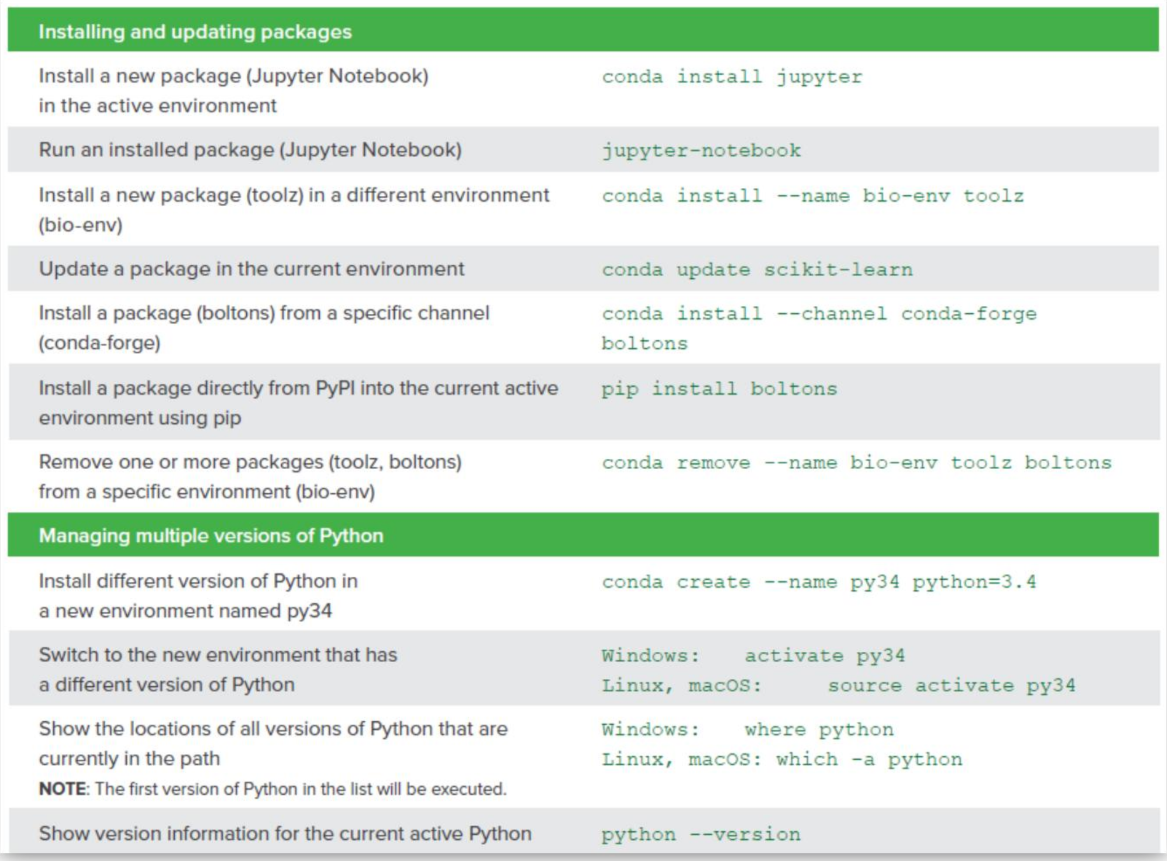
Using environments	
Create a new environment named py35, install Python 3.5	`conda create --name py35 python=3.5`

Activate the new environment to use it	`WINDOWS:    activate py35`
	`LINUX, macOS: source activate py35`

Get a list of all my environments, active environment is shown with *	`conda env list`
Make exact copy of an environment	`conda create --clone py35 --name py35-2`
List all packages and versions installed in active environment	`conda list`
List the history of each change to the current environment	`conda list --revisions`
Restore environment to a previous revision	`conda install --revision 2`
Save environment to a text file	`conda list --explicit > bio-env.txt`
Delete an environment and everything in it	`conda env remove --name bio-env`

Deactivate the current environment	`WINDOWS: deactivate`
	`macOS, LINUX: source deactivate`

Create environment from a text file	`conda env create --file bio-env.txt`
Stack commands: create a new environment, name it bio-env and install the biopython package	`conda create --name bio-env biopython`



Installing and updating packages	
Install a new package (Jupyter Notebook) in the active environment	`conda install jupyter`
Run an installed package (Jupyter Notebook)	`jupyter-notebook`
Install a new package (toolz) in a different environment (bio-env)	`conda install --name bio-env toolz`
Update a package in the current environment	`conda update scikit-learn`
Install a package (boltons) from a specific channel (conda-forge)	`conda install --channel conda-forge boltons`
Install a package directly from PyPI into the current active environment using pip	`pip install boltons`
Remove one or more packages (toolz, boltons) from a specific environment (bio-env)	`conda remove --name bio-env toolz boltons`

Managing multiple versions of Python	
Install different version of Python in a new environment named py34	`conda create --name py34 python=3.4`

Switch to the new environment that has a different version of Python	`Windows:   activate py34`
	`Linux, macOS:    source activate py34`

Show the locations of all versions of Python that are currently in the path
NOTE: The first version of Python in the list will be executed.
	`Windows:   where python`
	`Linux, macOS: which -a python`

Show version information for the current active Python	`python --version`

Environment Setup

Environment Setup

- This course will use Jupyter Notebooks/spyder for teaching and to provide notes
  - **Note**: you are free to use **whatever development environment you prefer (e.g., Spyder, PyCharm, ..)**

- We will be using the Python 3.6 for this course through the Anaconda Distribution
- Now let’s go over your installation options for Jupyter Notebook!

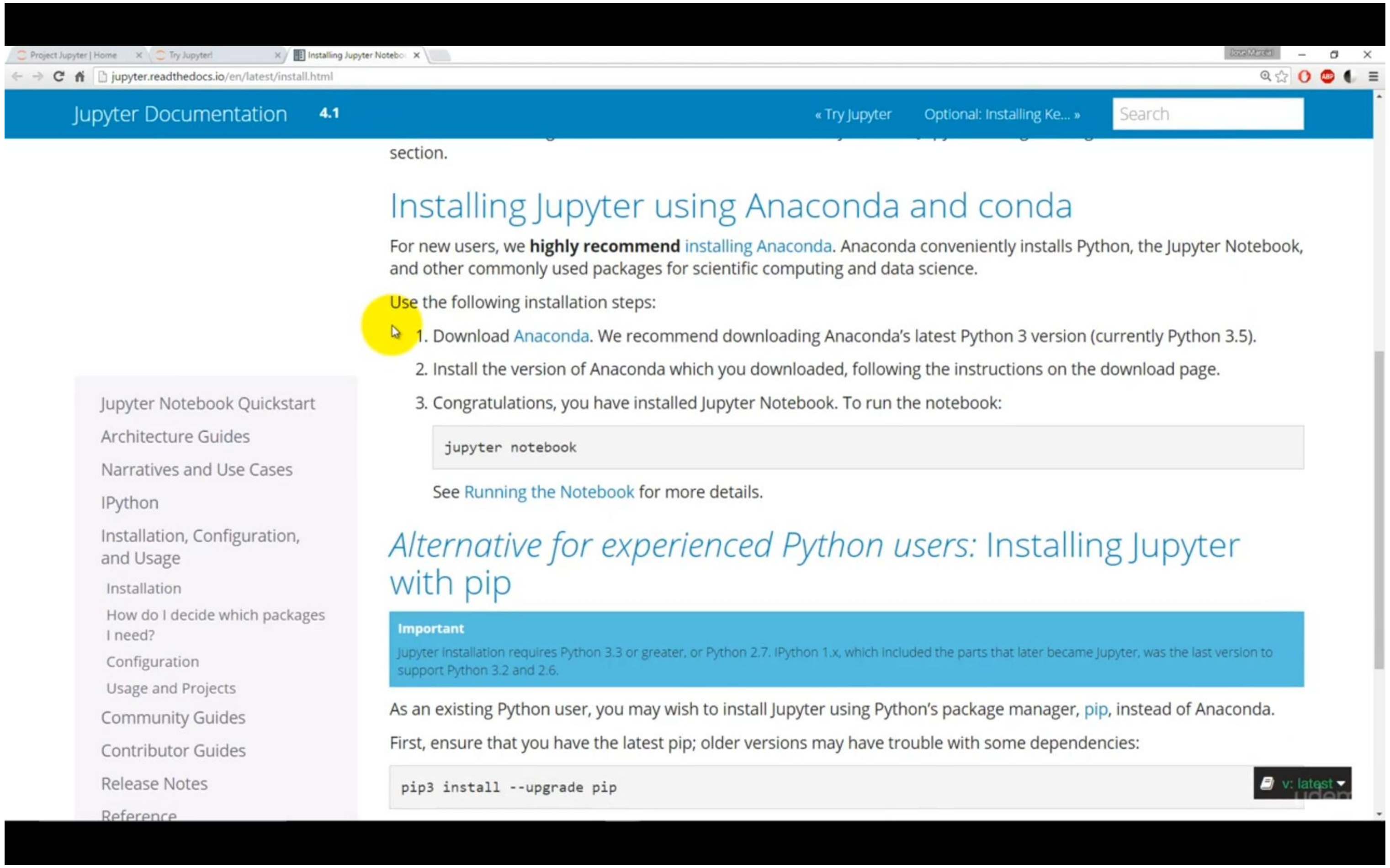
Environment Setup

- For experienced users who already have Python
  - As an existing Python user, you may wish to install Jupyter and required APIs using Python package manager pip, instead of Anaconda
  - Just go to your command prompt or terminal and use:

```
pip install jupyter
```

- For new users, we highly recommend installing Anaconda
  - Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science
  - Let’s go to www.jupyter.org to walkthrough the installation steps!

Environment Setup


Project Jupyter | Home
Try Jupyter!
Installing Jupyter Notebo
jupyter.readthedocs.io/en/latest/install.html
Jupyter Documentation 4.1
« Try Jupyter
Optional: Installing Ke... »
Search
Jupyter Notebook Quickstart
Architecture Guides
Narratives and Use Cases
IPython
Installation, Configuration, and Usage
Installation
How do I decide which packages I need?
Configuration
Usage and Projects
Community Guides
Contributor Guides
Release Notes
Reference
v: latest

section.

Installing Jupyter using Anaconda and conda

For new users, we **highly recommend** installing Anaconda. Anaconda conveniently installs Python, the Jupyter Notebook, and other commonly used packages for scientific computing and data science.

Use the following installation steps:

1. Download Anaconda. We recommend downloading Anaconda's latest Python 3 version (currently Python 3.5).
2. Install the version of Anaconda which you downloaded, following the instructions on the download page.
3. Congratulations, you have installed Jupyter Notebook. To run the notebook:

```
jupyter notebook
```

See Running the Notebook for more details.

Alternative for experienced Python users: Installing Jupyter with pip

Important

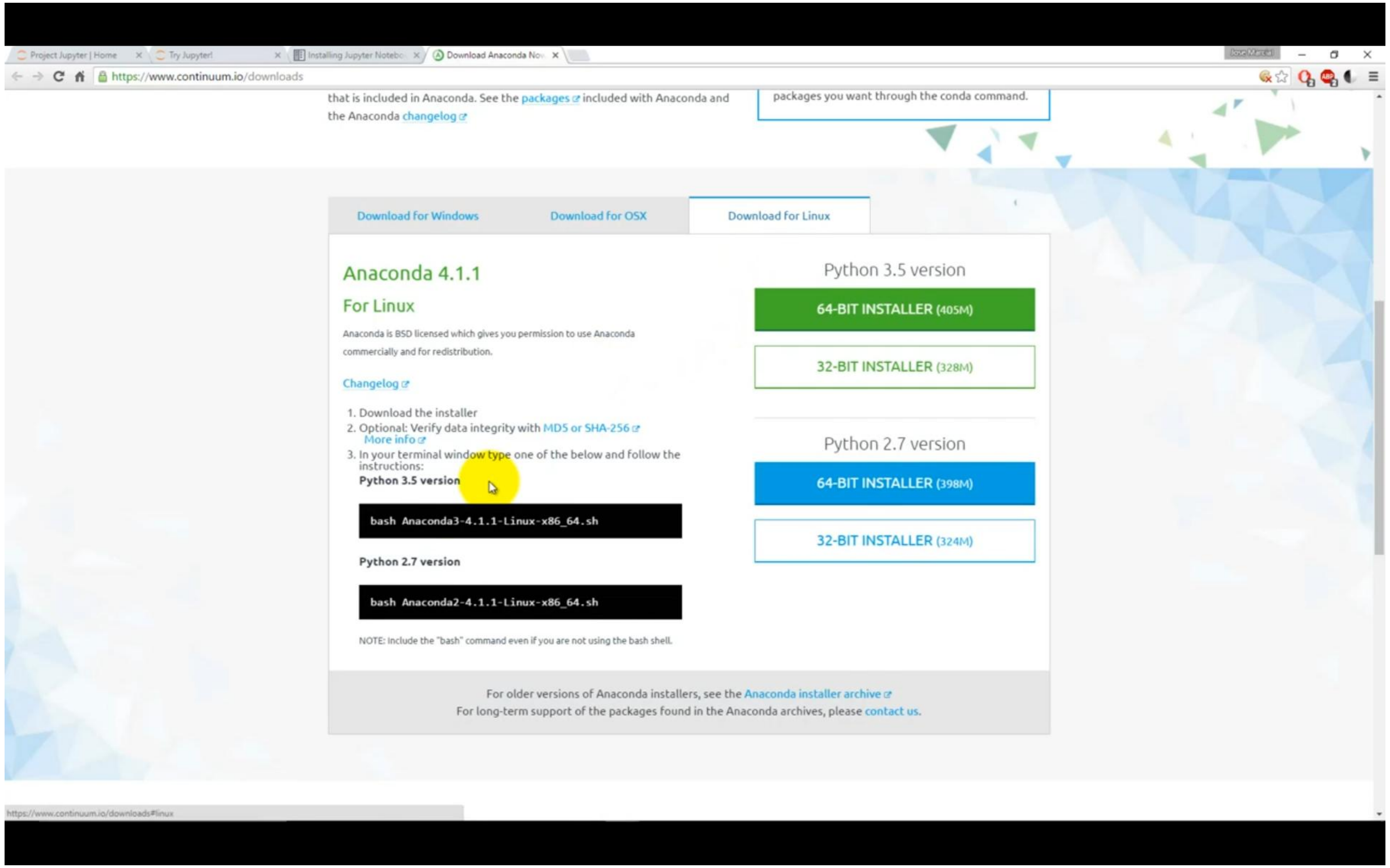
Jupyter installation requires Python 3.3 or greater, or Python 2.7. IPython 1.x, which included the parts that later became Jupyter, was the last version to support Python 3.2 and 2.6.

As an existing Python user, you may wish to install Jupyter using Python's package manager, pip, instead of Anaconda.

First, ensure that you have the latest pip; older versions may have trouble with some dependencies:

```
pip3 install --upgrade pip
```

Environment Setup



that is included in Anaconda. See the packages included with Anaconda and the Anaconda changelog

packages you want through the conda command.

Anaconda 4.1.1

For Linux

Anaconda is BSD licensed which gives you permission to use Anaconda commercially and for redistribution.

Changelog

1. Download the installer
2. Optional: Verify data integrity with MD5 or SHA-256
   More info
3. In your terminal window type one of the below and follow the instructions:

   Python 3.5 version

   ```
   bash Anaconda3-4.1.1-Linux-x86_64.sh
   ```

   Python 2.7 version

   ```
   bash Anaconda2-4.1.1-Linux-x86_64.sh
   ```

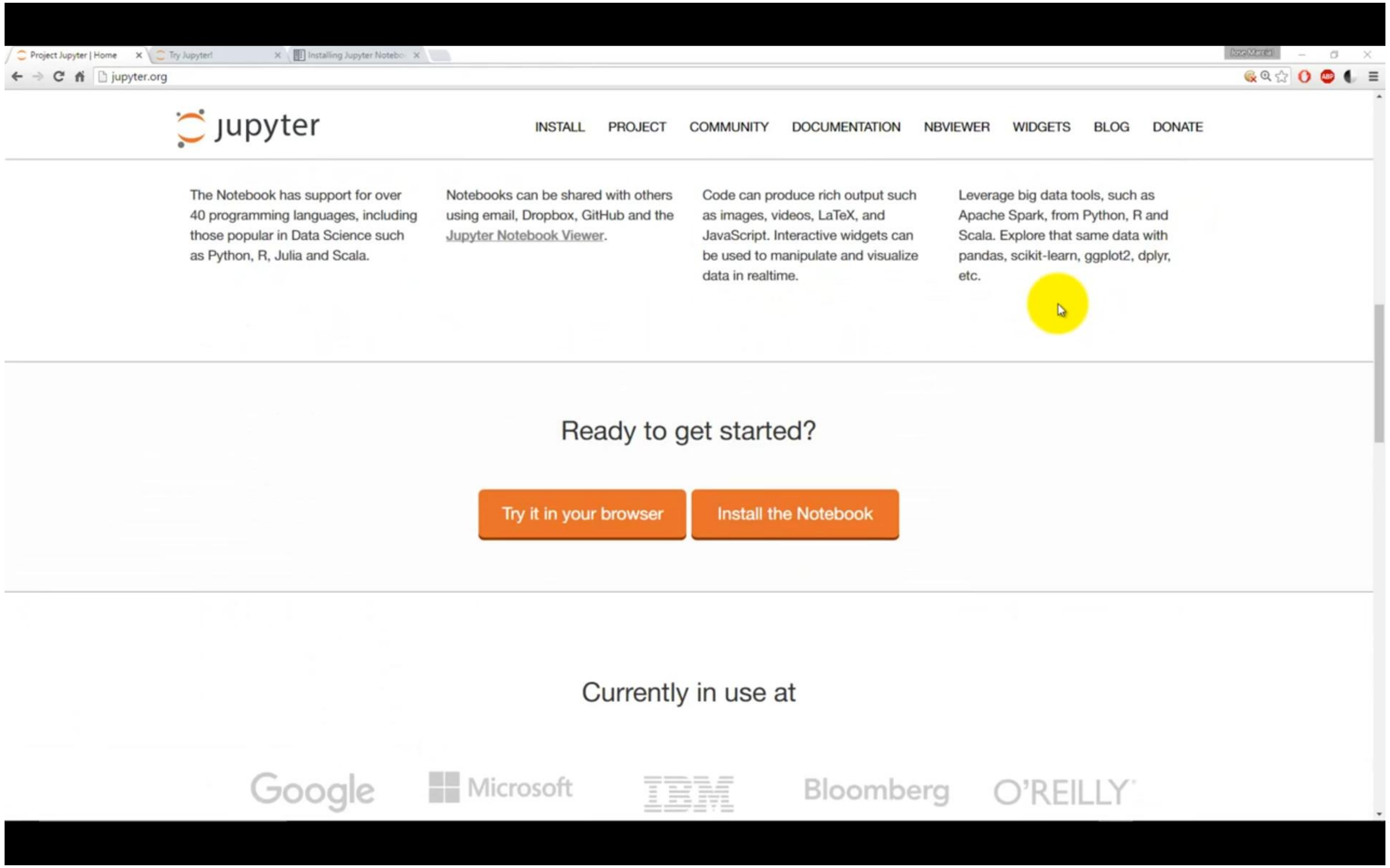
NOTE: Include the "bash" command even if you are not using the bash shell.

Python 3.5 version

Python 2.7 version

For older versions of Anaconda installers, see the Anaconda installer archive
For long-term support of the packages found in the Anaconda archives, please contact us.

Environment Setup



The Notebook has support for over 40 programming languages, including those popular in Data Science such as Python, R, Julia and Scala.

Notebooks can be shared with others using email, Dropbox, GitHub and the Jupyter Notebook Viewer.

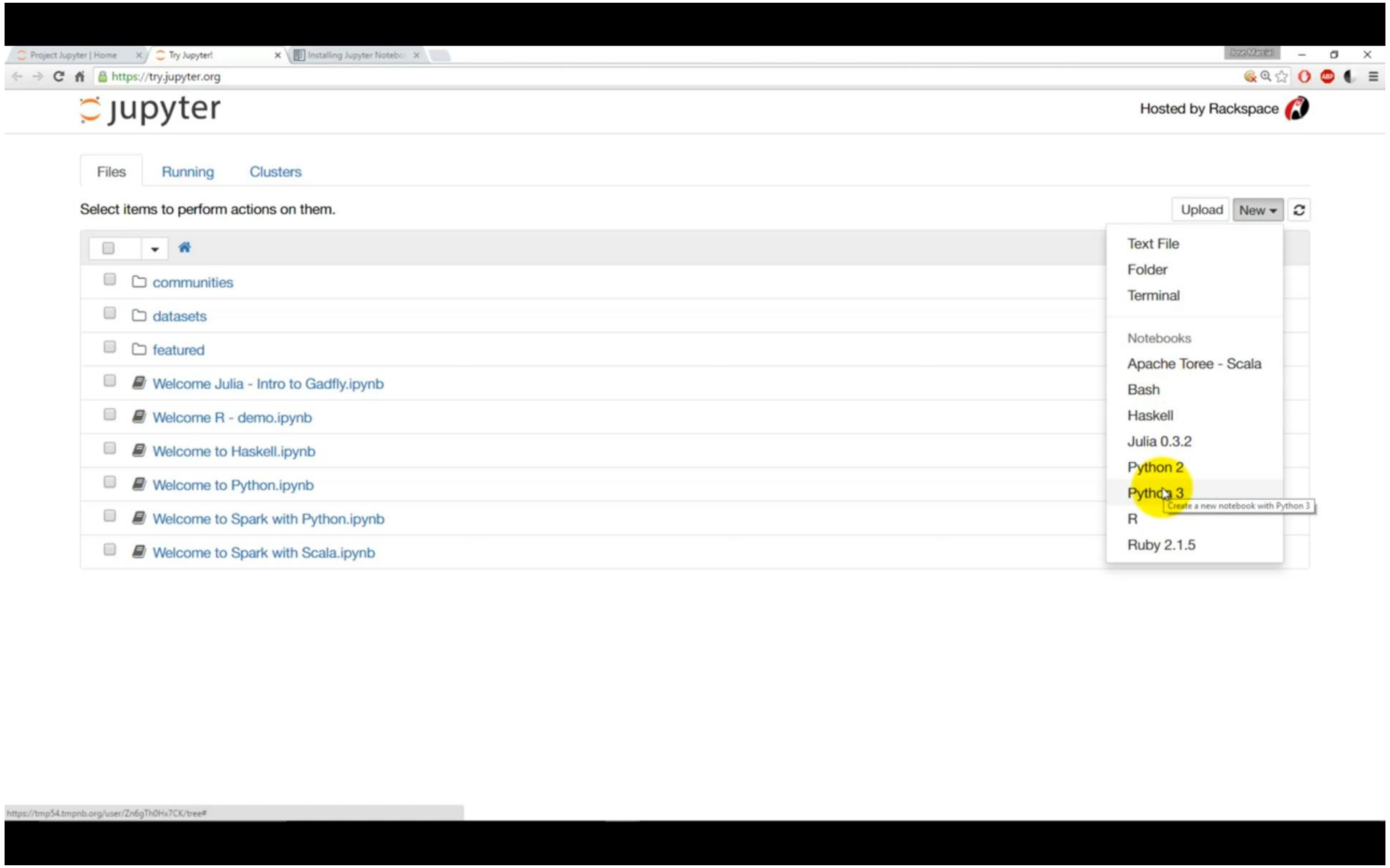
Code can produce rich output such as images, videos, LaTeX, and JavaScript. Interactive widgets can be used to manipulate and visualize data in realtime.

Leverage big data tools, such as Apache Spark, from Python, R and Scala. Explore that same data with pandas, scikit-learn, ggplot2, dplyr, etc.

Ready to get started?

Currently in use at

Environment Setup

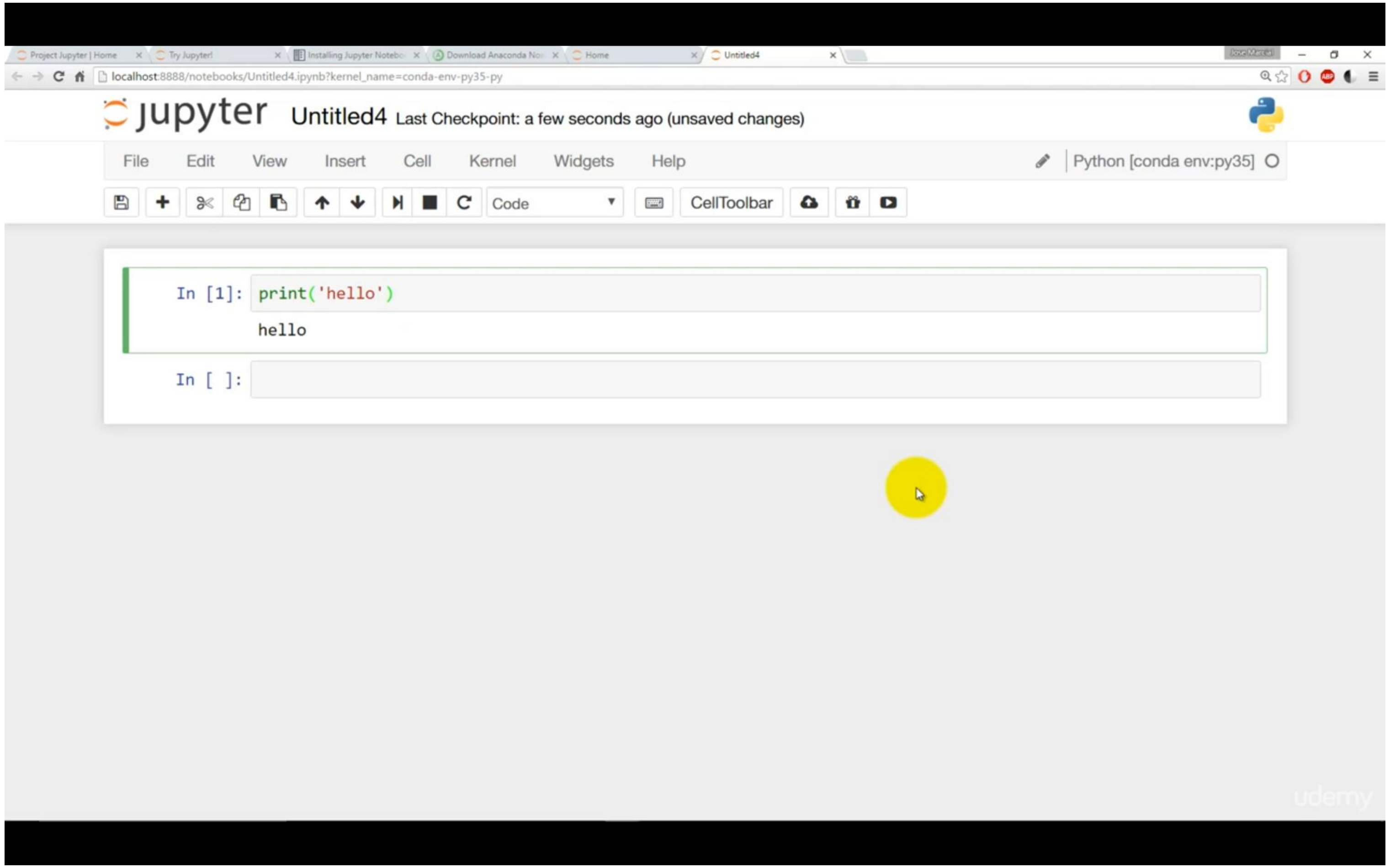


Select items to perform actions on them.

- communities
- datasets
- featured
- Welcome Julia - Intro to Gadfly.ipynb
- Welcome R - demo.ipynb
- Welcome to Haskell.ipynb
- Welcome to Python.ipynb
- Welcome to Spark with Python.ipynb
- Welcome to Spark with Scala.ipynb

https://tmp54.tmpnb.org/user/Zn6gThOHx7CK/tree#

Environment Setup


Project Jupyter | Home
Try Jupyter!
Installing Jupyter Notebo
Download Anaconda No
Home
Untitled4
localhost:8888/notebooks/Untitled4.ipynb?kernel_name=conda-env-py35-py
jupyter
Untitled4 Last Checkpoint: a few seconds ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Python [conda env:py35]
Code
CellToolbar
udemy

```
In [1]: print('hello')
        hello

In [ ]:
```

Hands On

Hands On

T1

- We will use scikit-learn/sklearn (Anaconda – Python Management Environment). Download and install the Anaconda Python package for your respective platform (Windows, Mac OS, Linux). The platform is available at https://www.anaconda.com/
  - Anaconda – Python 3.6
  - Deep Learning Libraries **not** required (Theano, Tensorflow, Keras)
  - Required to install Python IDE
  - https://machinelearningmastery.com/setup-python-environment-machine-learning-deep-learning-anaconda/

Hands On

T2

- Start Anaconda and create a virtual Python3.6 environment:
  - Open Terminal & Execute:
    - *conda create --name env python==3.6 numpy pandas xlrd xlwt matplotlib seaborn scikit-learn jupyterlab*
  - To install packages, enter the env. and execute: *conda install PACKAGENAME*
  - To work inside the python environment, execute: *conda activate env*
  - To exit python environment, execute: *conda deactivate*

Hands On

T2

- In this environment, the following libraries must be installed:

  a. Numpy

  b. Pandas

  c. Xlrd

  d. Xlwt

  e. Matplotlib

  f. Seaborn

  g. Scikit-learn

  h. Jupyterlab

Hands On

T3

- Activate the created virtual environment and check the installed libraries; validate the installation of the set of libraries presented in T2

T4

- Briefly check the documentation for each library mentioned in question T2. Identify its relevance in the context of Machine Learning algorithm development.