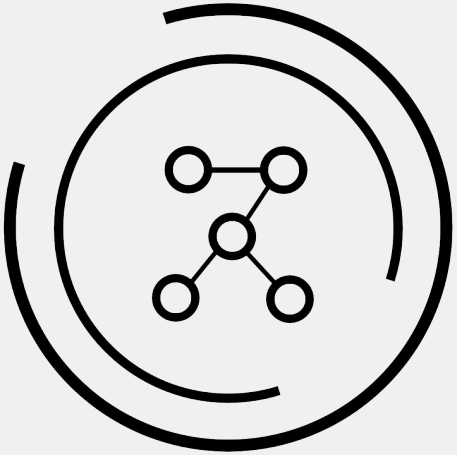


DSS
@Berkeley

Python for Data Science II

Workshop Series #2

Welcome!

Data is Everywhere!

Instructors + LA's

Created by Rockicon
from Noun Project

Instructors

Alex Nakagawa
alex.nakagawa@berkeley.edu
Sophomore
Computer Science & Cognitive Science

Neelesh Dodda
ndodda@berkeley.edu
Sophomore
Electrical Engineering & Computer Science

Lab Assistants

Jun Seo Park

Katharine Chua

Nicole Chen

Set-up Check

Set-up Check

- ➔ Anaconda
- ➔ Jupyter Notebook
- ➔ numpy
- ➔ pandas
- ➔ matplotlib
- ➔ seaborn

Question Portal

https://pollev.com/alexnakagawa209

Lesson Plan

Lesson Plan

1. Quick Review of Python I Workshop
2. Python Data Science Packages
    - 2.1. numpy (advanced)
        - 2.1.1. Creating tables
        - 2.1.2. manipulating rows
    - 2.2. pandas - data manipulation
        - 2.2.1. pd.DataFrame( )
    - 2.3. scikit-learn (machine learning)
3. Data Visualization
    - 3.1. matplotlib
    - 3.2. seaborns

Rules of the Classroom:

- Ask questions! (https://pollev.com/alexnakagawa209)
- Collaboration! (Talk to people you don't know)
- HAVE FUN!

Data Science with Python I Review

Data Types and Functions

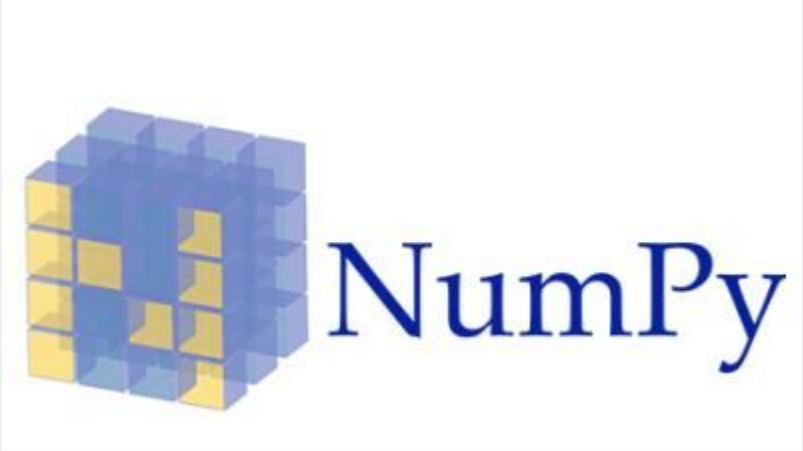
- Data Types (built-in):
  - integers
  - floats
  - strings
  - lists
  - dictionaries
- Functions:
  - user-defined functions

Data Types and Functions

- Data Types (built-in):
  - integers `a = 2`
  - floats `b = 3.1415926`
  - strings `c = 'Hello world!'`
  - lists `d = [a, b, c]`
  - dictionaries `e = {'key1':1, 'key2':2}`
- Functions:
  - user-defined functions

```
def square(x):
    return x ** 2
```

Working with arrays



Numpy

➔ General common practice to import `numpy` as `'np'`

```
In [ ]: import numpy as np
```

➔ Arithmetic

- **`np.sum()`, `np.mean()`, `np.median()`, `np.cumsum()`, `np.diff()`, `np.max()`, `np.min()`,** and more...

Numpy (cont.)

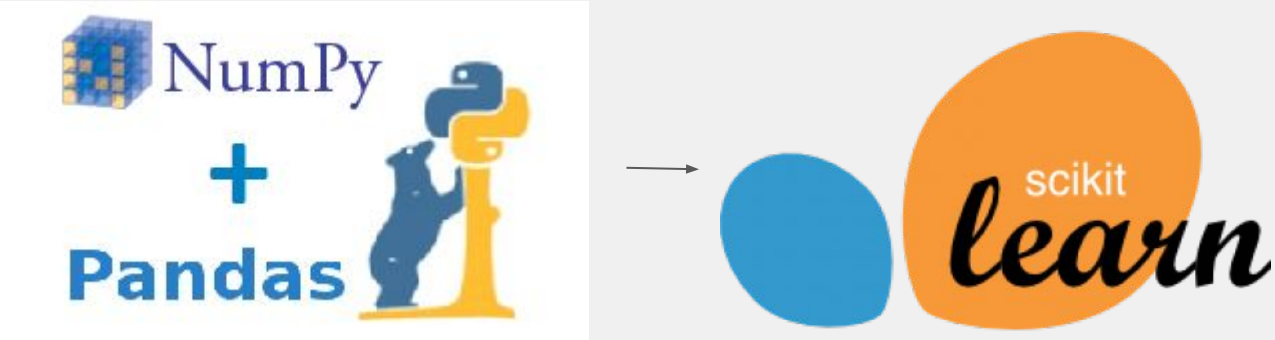
- `np.array([1,2,3], dtype=int32)`
  - Can only have one data type, has more restrictions than a Python list
- `np.zeros(N)`
  - Create a 1 x N array filled with zeroes
- `(np.array([1,2,3], dtype=int32)).shape`
  - The `.shape` attribute prints the dimensions of the numpy array
- `np.linalg`
  - A set of useful functions to use for linear algebra operations

Numpy - Sanity Check!

- ➔ The differences between an **`np.array()`** and a Python list **`[]`**
- ➔ Finding the sum, differences, mean, median, etc. of an array.
- ➔ Finding dimension of a **`np.array()`** with multiple **`np.array()`** inside of it
- ➔ Changing the data type of all the values in the array

Advanced Data Science Packages

Advanced Data Science Packages



Pandas

➔ Common practice to import `pandas` as `'pd'`

```
In [ ]: import pandas as pd
```

➔ Data management

- **`pd.read_csv()`, `pd.Dataframe()`, `pd.concat()`, `pd.Series()`, `pd.merge()`** and more...

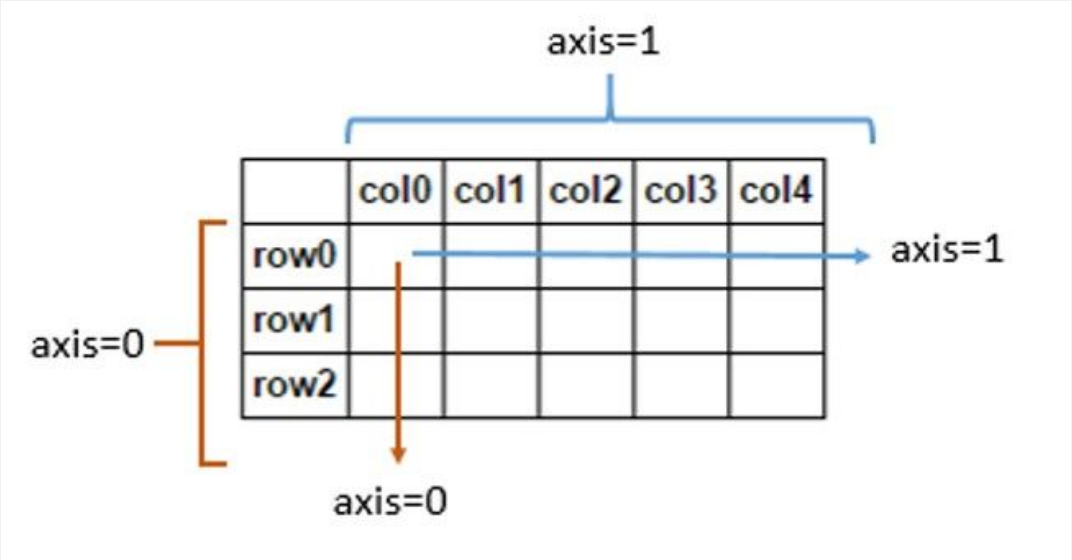
Pandas (cont.)

➔ `s = pd.Series([1, 2, 3, 4, 5])`

- Produces a 1-dimensional array, very similar to `np.array()`
- Used for creating dataframes and visualizing data
- Can specify row titles:
  - `s = pd.Series([1, 2], ['r1', 'r2'])`
  - Rows are now labelled `r1` and `r2`

Pandas (cont.)

➔ `df = pd.DataFrame()`



Pandas (cont.)

➔ `s = pd.Series([1, 2], ['r1', 'r2'])`

`df = pd.Dataframe({'c1': s, 'c2': s})`

- Produces a 2-dimensional matrix
  - In this case, the first and second row have the same values
  - **`c1`** and **`c2`** represent the column titles
  - **`r1`** and **`r2 represent the rows`**
- Used to represent tables of data and data visualization

Pandas (cont.)

- `s = pd.Series([1, 2], ['r1', 'r2'])`

  `df = pd.Dataframe({'c1': s, 'c2': s})`

  - **`df.head()`**: Quick overview of data
  - **`df.describe()`**: Basic statistical summary (i.e. max, min, mean etc.)
  - **`df['c1']`**: Retrieves the all of the entries in the **`c1`** column
  - **`df.columns`**: Displays all of the column titles
  - **`df.index:`** Displays all of the indices in an array
- And so, so much more!

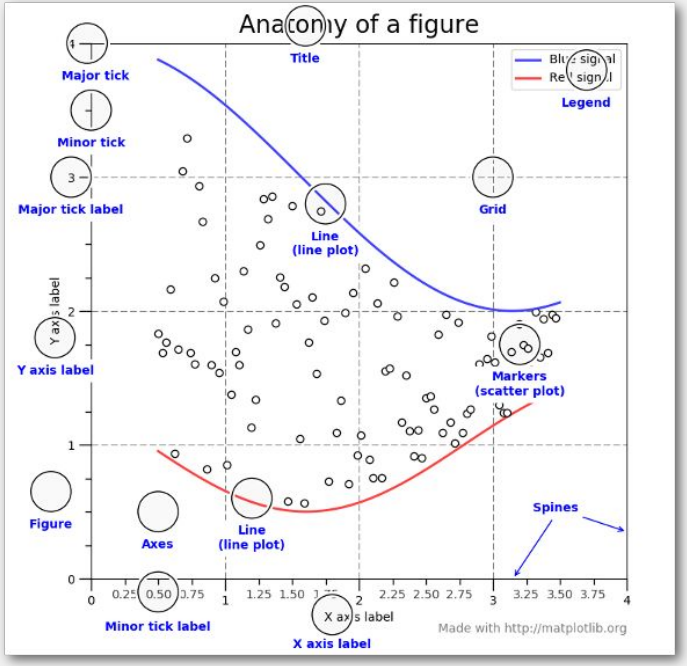
Pandas - Sanity Check!

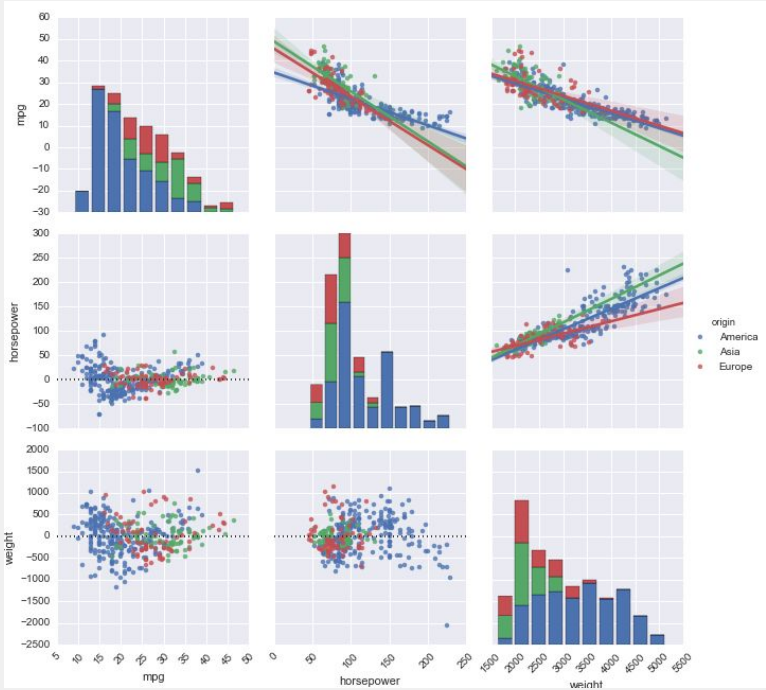
- ➔ Creating a dataframe, how to input data
- ➔ Manipulating the labels of columns, creating **`pd.Series()`**
- ➔ Access parts of the data frame, such as specific rows that satisfy a predicate
- ➔ Setting indices, finding information about each column, what is a NaN

Data Visualization

Seaborn

Examples of Graphs





Data Visualizations - matplotlib

- General common practice to import matplotlib.pyplot as plt

```
3 import matplotlib.pyplot as plt
```

```
1 # this line makes plots/graphs appear within the notebook
2 %matplotlib inline
```

- Basic 2-D plotting: histograms, line graphs, scatterplots, bar charts, and more
  - https://matplotlib.org/gallery.html
- **`plt.plot(x,y), plt.scatter(x,y), plt.hist(x),`**

  `plt.subplots(nrows,ncols), plt.style.use('ggplot')`

Data Visualizations - seaborn

- General common practice to import seaborn as sns

- More options to choose from, a more diverse color scheme, multiple graphs in the same module.
  - https://seaborn.pydata.org/api.html#api-ref
- **`sns.heatmap(), sns.barplot(), sns.swarmplot(), sns.violinplot(), sns.distplot()`**

matplotlib/seaborn - Sanity Check!

➔ Knowing the different graphs and knowing when to use them

SUMMARY!

- ★ Data Types of data science
  - `[],{},True/False`
  - `np.array()`
  - `pd.DataFrame()`
  - `pd.Series()`
- ★ Packages
  - **`numpy`**
    - used to create arrays or reformat
  - **`pandas`**
    - data frame creation, analogous to tables
  - **`matplotlib/seaborn`**
    - data visualization packages
    - helpful for visualizing trends and patterns
  - **`scikit-learn`**
    - used for machine learning models and making predictions
- ★ Tips for creating your own data
  - `np.zeros(N)`
  - check the docs if you're stuck!
  - relabeling column names
  - checking for consistent data types
- ★ Tips for reformatting data from internet
  - check data types of each column

Resources

https://data.berkeley.edu/education/faqs

Relevant Classes

Relevant Classes

- The Structure and Interpretation of Computer Programs [CS 61A]: http://cs61a.org/
- Foundations of Data Science [STAT/CS C8]: http://data8.org
- Principles and Techniques of Data Science [DS 100]: http://ds100.org
- Introduction to Probability and Statistics [STAT 20]: https://www.stat.berkeley.edu/~nolan/stat20/index.html
- Concept of Probability [STAT 134]: http://statistics.berkeley.edu/programs/undergrad/major
- Applied Data Science with Venture Applications [IEOR 135]: http://data-x.blog/
- Introduction to Machine Learning [CS 189]: https://people.eecs.berkeley.edu/~jrs/189/

Extra Practice/Cool Tools

- www.kaggle.com
- www.datacamp.com
- www.data.world
- www.codecademy.com
- www.dataquest.com
- https://books.google.com/ngrams/

Data Science Society at Berkeley

Feedback:

goo.gl/7BJKkW

Data Science Society at Berkeley

Thanks for coming!