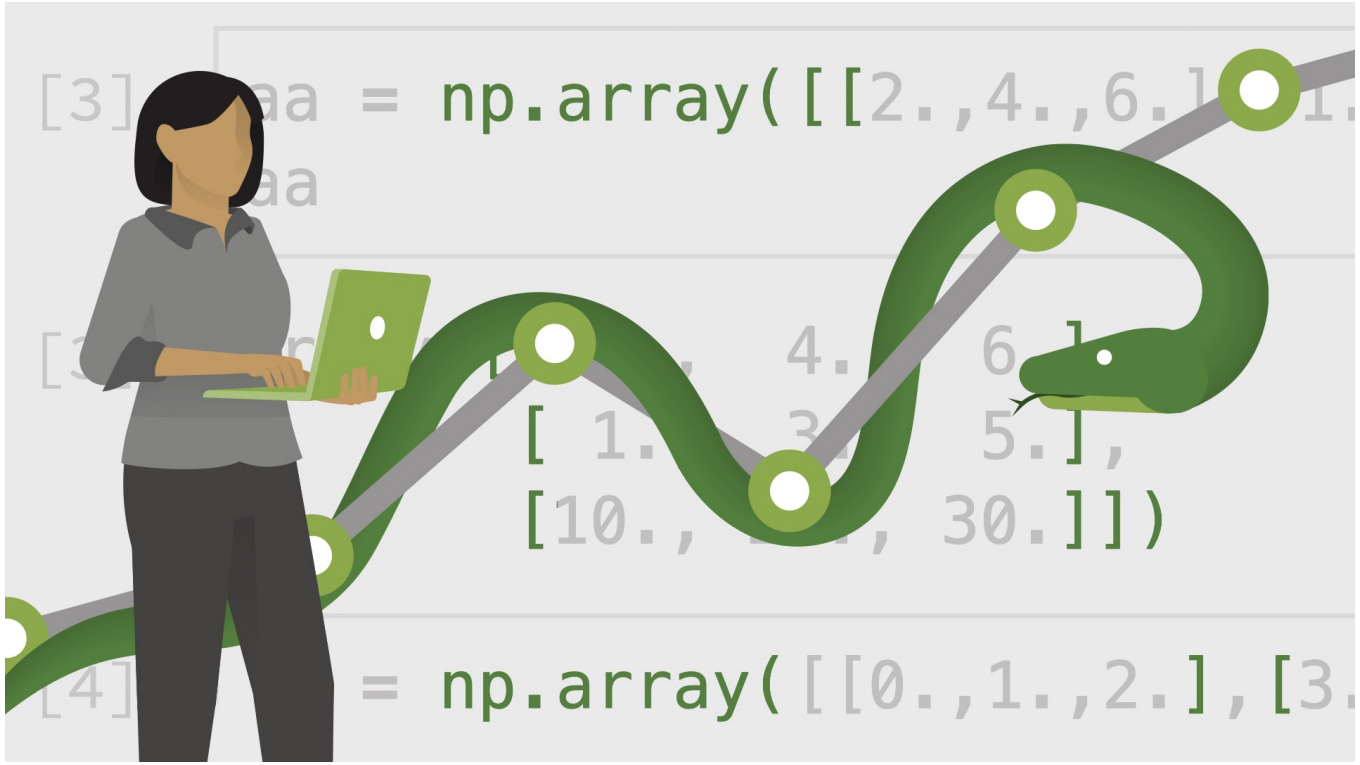
Teaching Data Science using Python II



The Python Data Science Ecosystem

Teaching Data Science using Python

Sandbox option: Berkeley's Data 8 course

- Uses the datascience package

Real world option uses a set of Python packages:

- Standard Python libraries
- NumPy
- Pandas
- Matplotlib
- Also: seaborn, statsmodels, scikitlearn

Python Data Science packages

Going to give a basic overview of some of the main Python Data Science packages

Will redo the avocado analyses using some of these packages

NumPy is a library that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays.

- i.e., it is similar to MATLAB

The core data structure of NumPy is its "ndarray".

Ndarrays are similar to Python lists, except that all elements in an ndarray must of the same type

- E.g., all elements are numbers, or all elements are strings, etc.


```
import numpy as np
x = np.array([1, 2, 3])
2 * x

#  the numbers 0 to 9
x = np.arange(10)

#  3 x 3 matrix
M = np.array([[1, 2, 3], [3, 4, 6.7], [5, 9.0, 5]])
```

SciPy contains modules for optimization, linear algebra, integration, interpolation, FFT, signal and image processing, etc.

- Uses ndarrays as main data structure

pandas is a library for data manipulation and analysis that has two main data structures:

1. Series: One-dimensional ndarray with an index for each value

- Similar to a named vector in R

2. DataFrame: Two-dimensional, size-mutable, potentially heterogeneous tabular data.

- Similar to an R data frame
- (or multiple Series of the same length with the same index)


```
import pandas as pd
avocado = pd.read_csv("avocado.csv")
avocado.head(3)     # show the first 3 rows

avocado["AveragePrice"]  # returns a series

# Get the average value for all numerical
columns separately for each type of avocado
avocado.groupby("type").mean().reset_index()
```

Matplotlib is a plotting library. Each plot has a figure and a number of different subplots (axes).

- somewhat similar to base R graphics

It has two interfaces for plotting:

1. A "pylab" procedural interface based on a state machine that closely resembles MATLAB
   - Updates are made to the most recent axis plotted on

2. An object-oriented API
   - Updates are made to the axis that is selected

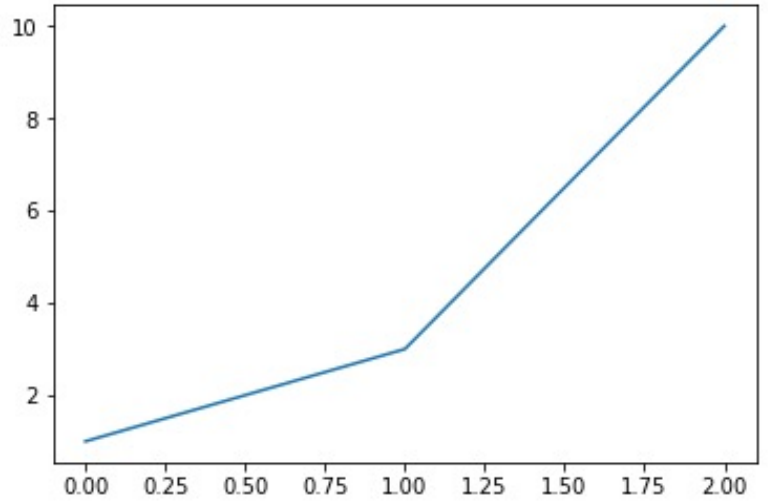
The objected oriented interface is preferred (not a big difference)

matplotlib

```
import matplotlib.pyplot as plt

# pylab interface (like matlab)
plt.plot([1,3,10]);

# object oriented interface
fig, ax = plt.subplots()
ax.plot([1,3,10]);
```



seaborn

seaborn is a visualization library built off Matplotlib, but it provides a higher level interface that uses Pandas DataFrames

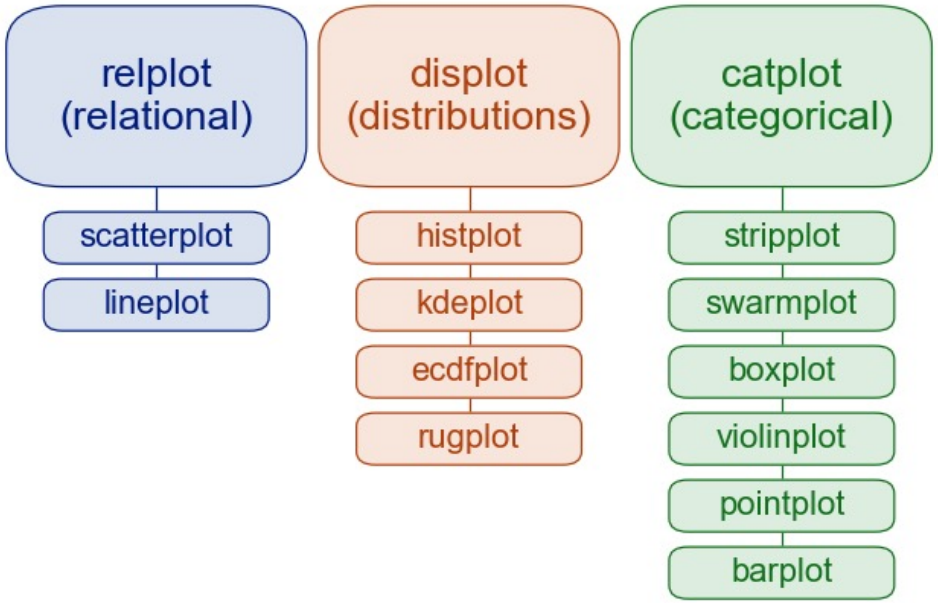
- somewhat similar to ggplot

There are "axes-level" functions that plot on a single axis and "figure-level" functions that plot across multiple axes

Figure level plots are grouped based on the types of variables being plotted

- E.g., a single quantitative variable, two quantitative variables, etc.

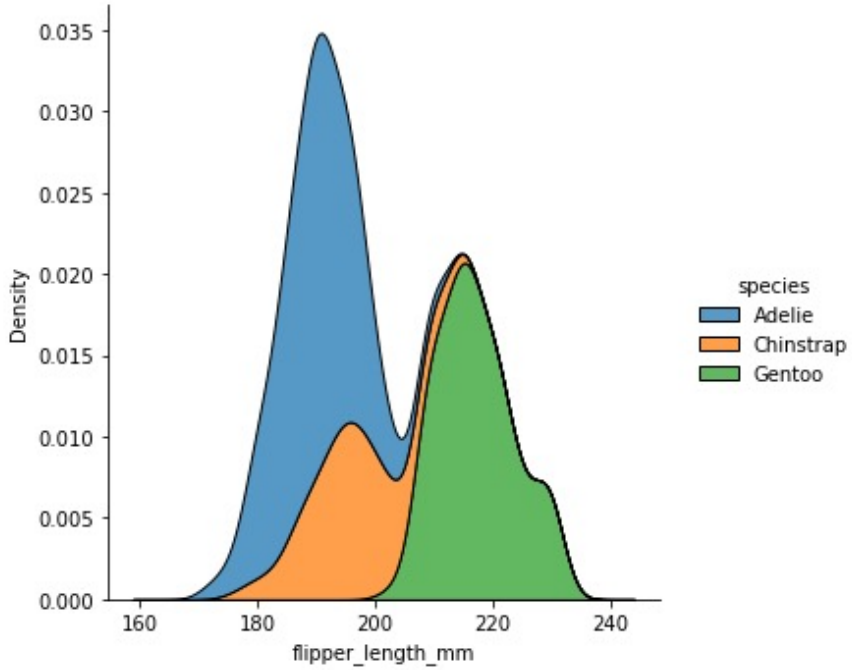
Figure level plots



seaborn

```
import seaborn as sns
penguins = sns.load_dataset("penguins")

# figure-level plot
sns.displot(data=penguins,
                x="flipper_length_mm",
                hue="species",
                multiple="stack",
                kind="kde");
```



Translation between Tables and DataFrames

Translation between datascience Tables and pandas DataFrames

Translation between datascience Tables and babypandas DataFrames

Let’s try it ourselves!