# Python for Data and Text Mining

Mohammed Shameer Iqbal

bit.ly/SMU-2019-2

bit.ly/smu-files-2

# Agenda

- Fetching data and clean up:
  - OS module
  - CSV module
  - Re module
- Matrix manipulation
  - NumPy and SciPy
  - PIL
- Data manipulation and Visualization:
  - Pandas
  - Matplotlib and Plotly

# OS module

Provides cross-platform libraries

- os.getcwd()
- os.listdir()
- os.walk()
- os.popen()
- os.mkdir()
- os.rmdir()

# CSV library

- ```
  import csv

  with open("numbers.csv") as f:
      r = csv.reader(f)
      for row in r:
          print row
  ```

Note: `with` is a context manager in python which automatically close the file when code block finishes
  ```

# Re module

Powerful regular expression (regex) library

- You specify a pattern by using pre-defined grammar
- Search for the pattern in text and find them

# Exercise 1

1. Use os library to run ping command and parse the logs to get Average ping time
2. Read apache log file and parse the information to get IP address and time stamp
3. Write the results to a csv file

# Non-standard libraries

- We often might need libraries beyond standard libraries
- Anyone can publish their modules as python libraries
- PyPI - Python Package Index has all third-party libraries
- We can use `pip` to install the required packages

```
pip install numpy
```

- To get specific version you can add version to install command:

```
pip install numpy==1.16.3
```

Note: Learn about virtual environments to keep your dependencies clean

# Numpy Library

- *“NumPy is the fundamental package for scientific computing with Python. It contains among other things:*
  - *a powerful N-dimensional array object*
  - *sophisticated (broadcasting) functions*
  - *tools for integrating C/C++ and Fortran code*
  - *useful linear algebra, Fourier transform, and random number capabilities”*

- Reference: https://www.numpy.org/

# Numpy - Usage

- `import numpy as np`
- You can create arrays in many ways:
  - `a = np.array([2,3,4])`
  - `a = np.zeros([3,3])`
  - `a = np.arange(15).reshape(3,5)`
- Shape property gives shape (or dimensions) of the array
- We can perform array-wise operations
  - `a = 3 * a`

# Numpy Indexing and Slicing

- Indexing and slicing is similar to list except we should be careful about the dimensions
  - ```
    aa = np.arange(15).reshape(3,5)
    print(aa[0,0])
    print(aa[0])
    print(aa[0, :])
    print(aa[:,2])
    ```
    ```

# Numpy - Shape Manipulation

- We can shape the arrays if the requested shape still contains the same amount of elements. For instance, we cannot reshape a (3,5) array into (5,2)

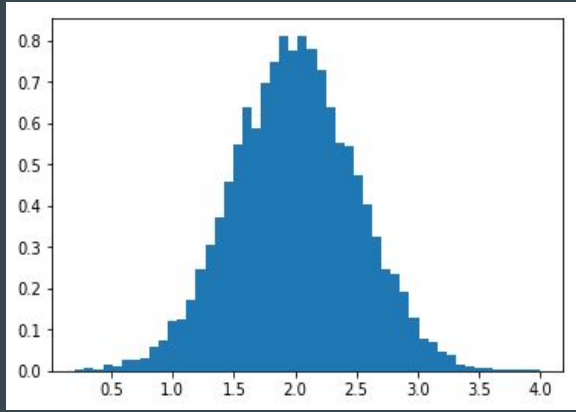
```
In [23]: aa.reshape([5,3])
Out[23]:
array([[ 0,  1,  2],
       [ 3,  4,  5],
       [ 6,  7,  8],
       [ 9, 10, 11],
       [12, 13, 14]])
```

# Numpy - Combining arrays

- vstack and hstack are used to combine two arrays along vertical and horizontal axis respectively
  - ```
    a = np.floor(10*np.random.random([2,3]))
    b = np.floor(10*np.random.random([2,3]))
    print(np.vstack((a,b)))
    print(np.hstack((a,b)))
    ```
    ```

# Numpy + Plot

```
import numpy as np
import matplotlib.pyplot as plt
# Build a vector of 10000 normal deviates with
#variance 0.5^2 and mean 2
mu, sigma = 2, 0.5
v = np.random.normal(mu,sigma,10000)
# Plot a normalized histogram with 50 bins
plt.hist(v, bins=50, density=1)
# matplotlib version (plot)
plt.show()
```



# Images

- How would you store images?
- How do you handle colours?
- How many dimensions would we need?
- Can matrices be used to store images?

# Exercise 2

1. Create a random matrix of size [5, 5]
2. Open an image from folder images and display it in console
3. Concatenate 10 images side by side and create one image
4. Combine two images using array manipulation:

# Pandas

*“pandas is an open source, BSD-licensed library providing high-performance, easy-to-use data structures and data analysis tools for the Python programming language.”*

Documentation: https://pandas.pydata.org/pandas-docs/stable/

Workshop content was adapted from this talk:

https://www.youtube.com/watch?v=5JnMutdy6Fw

https://github.com/brandon-rhodes/pycon-pandas-tutorial

# Pandas Series and Dataframe

- A Series is a one-dimensional object that can hold any data type such as integers, floats and strings.
- A DataFrame is a two dimensional object that can have columns with potential different types.

# Pandas Dataframe

- We can create data frames from different source types
  - CSV
  - Python objects
  - Databases