GR5067
Natural Language Processing
Quantitative Methods – Social Sciences (QMSS)

Professor: Patrick Houlihan

Introduction

- Professor: Patrick Houlihan, PhD
  - Email: pjh2144@columbia.edu
  - Office Hours: Monday 06:30pm-7:45pm
    - Please contact me in advance

- Class:
  - When: Mon 8:10pm-10:00pm EST
  - Location: Kent Hall 413

- TAs:
  - Lin Zhu lz2808@columbia.edu
  - Sibo Geng sg4010@columbia.edu
  - Zeya Ahmad za2291@columbia.edu

Introduction

- Expectations
- My Background
- TA Introduction
- Syllabus Review
- Project Review
- Required Installations
- What is Natural Language Processing?
- Python
- Natural Language Toolkit (NLTK)

Expectations

- Familiar with a programming language, i.e. R, Java, Matlab
- Previous experience with Python a plus, though NOT a show stopper
- Lots of data wrangling as text can be quite messy
- Professor and TAs are here to help
- Effort and Grade are directly correlated
  - You get out of the class what you put in

Professor Background

- Spent 15+ years in semiconductors
  - Altera
  - Nvidia

- Last 10 years been surrounded by big data
  - Dissertation:
    - Forecasting Asset Price Direction Through Sentiment
  - Publications:
    - Risk Premium of Social Media Sentiment
    - Can Sentiment Analysis and Options Volume Anticipate Future Returns?
    - Leveraging a call-put ratio as a trading signal
    - Leveraging Social Media to Predict Continuation and Reversal in Asset prices

- Founded financial data analytics company – SentiQuant

- Currently Senior Vice President, Data Science at Publicis Media

Syllabus Review

- HW 50%
  - 4 HWs
  - Equal Weighted
- Project 50%
  - Paper 30%
  - Presentation 20%

Textbooks

- https://web.stanford.edu/~jurafsky/

- Natural Language Processing with Python (https://www.nltk.org/book/)

Project Review

- Team based project – 50% total grade
  - Presentation – 20%
  - Paper – 30%
- Illustrate knowledge of Python and NLP techniques

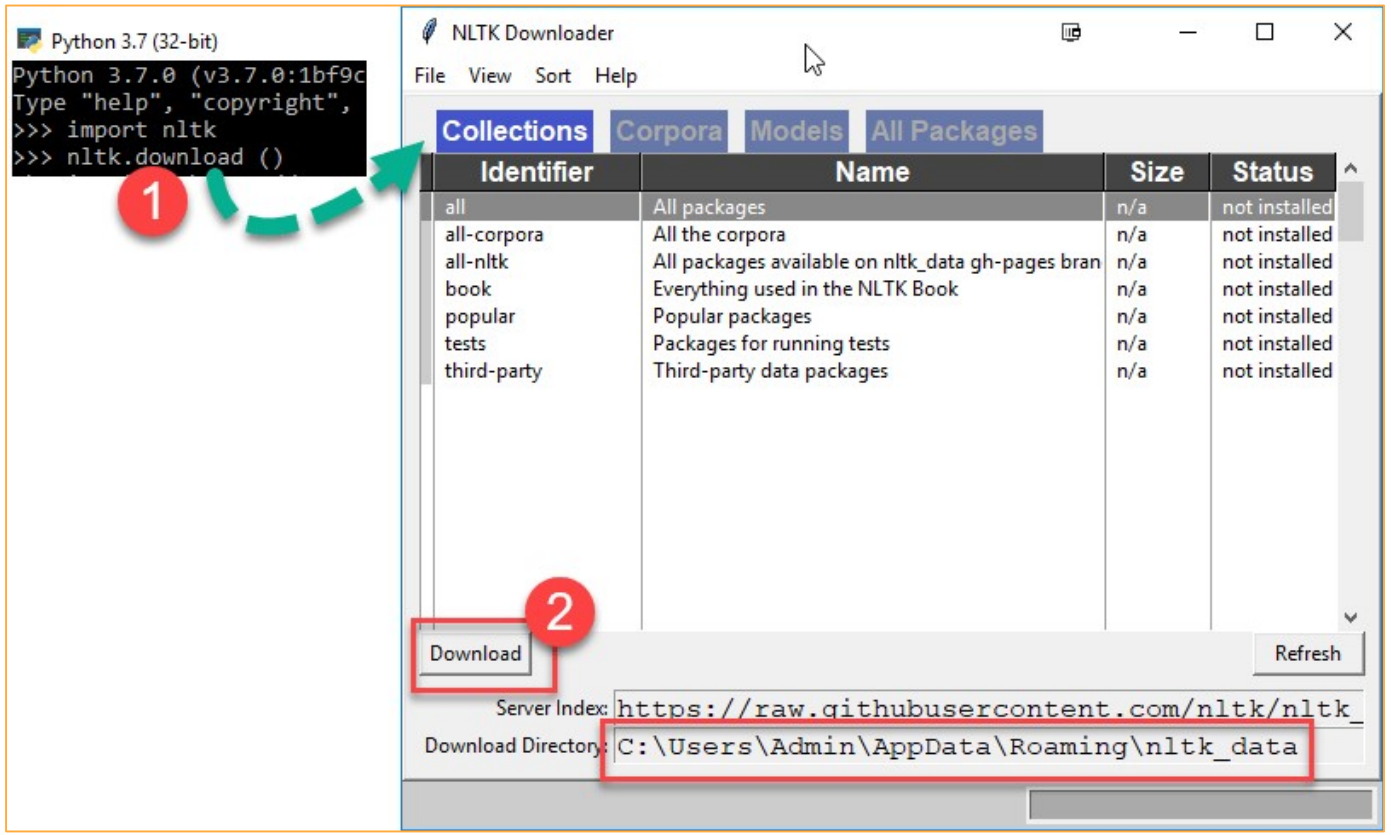
Nice to Know

- Optional but highly recommended
- Create GitHub accounts:
  - https://github.com/

Required Installations

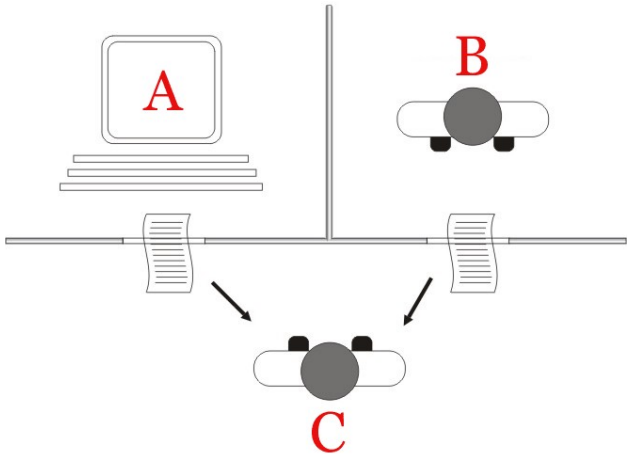
- Python 3.X version
  - https://www.anaconda.com/download/

- NLTK download and Install



What is Natural Language Processing?

- Interaction between computers and human (natural) languages
  - Understanding – Process meaning of spoken/typed words
  - Generation – Expression into natural (human) language i.e. English
- Processing of vast amounts of natural language data (text)
- Brief History
  - Started in 50's
  - Alan Turing
    - Turing test
      - Test of intelligent behavior



Applications of Natural Language Processing

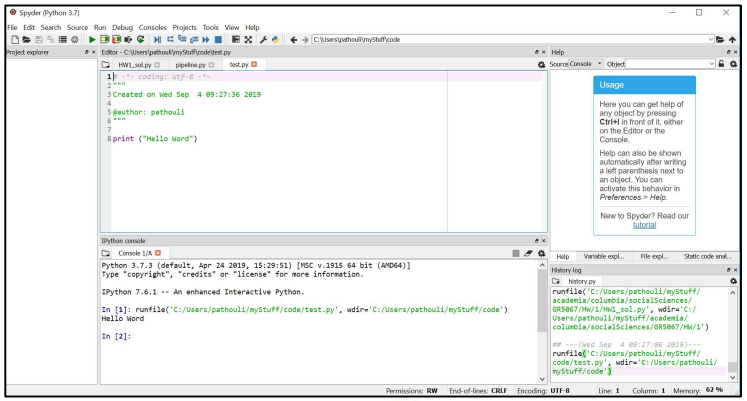
- Language Modeling
  - Predict next word based on previous words
- Speech Recognition
  - Mapping acoustical signals to a natural language
- Word Associations
  - Determine synonyms and related words for a word or phrase
- Sentiment Analysis
  - Determine tone or mood of author or entire society
- Text Classification
  - Predict categorical associations of text
- Topical Extraction
  - Determine main topic/theme of a body of text

Python

- Big emphasis on Python (not of Serpentes Suborder) programming language

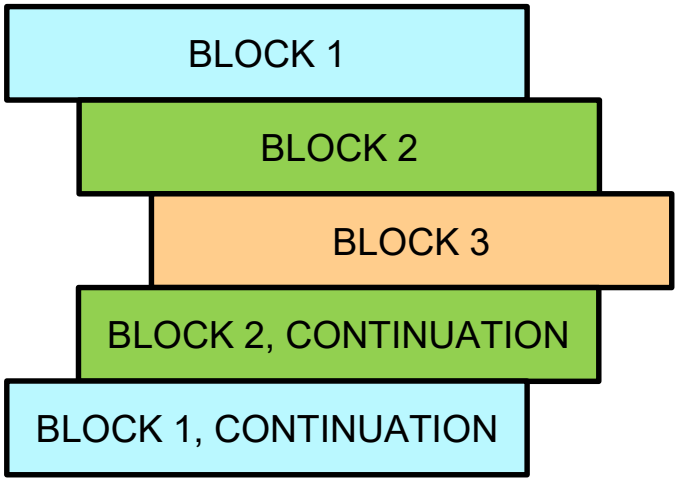
- Python Programming Language
  - Code – instructions in a program
  - Syntax – valid structures and commands
  - Output – Messages printed by program
  - Shell – Interpreter
  - Integrated Development Environment (IDE) – Software to write and test software

```
(base) C:\Users\pathouli\myStuff\academia\columbia\socialSciences\GR5067\HW\1>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```



Python Syntax

- Python uses indentation



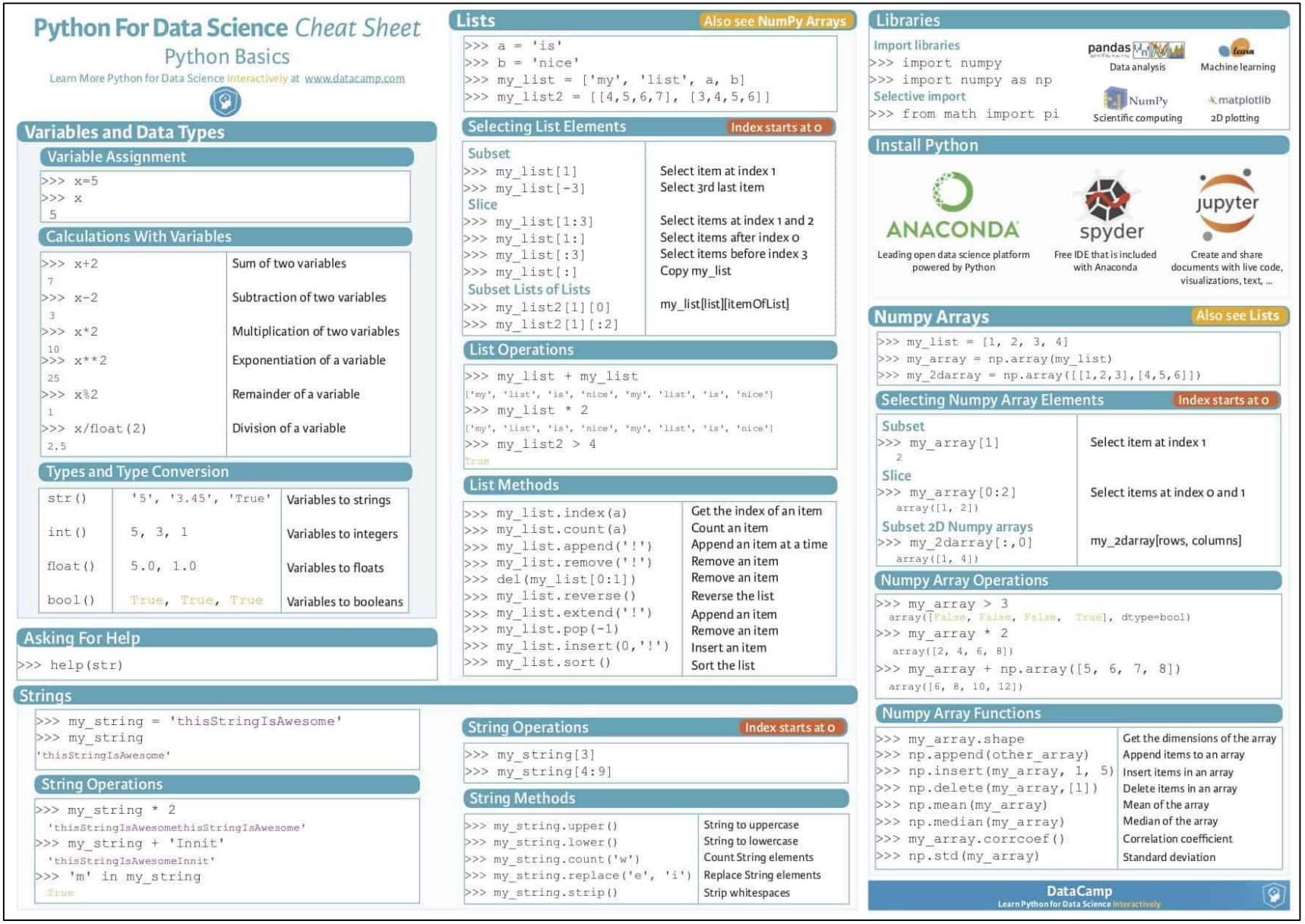
- Variables
  - Variables are case sensitive, myVar is different than myvar
  - Do:
    - Start variable name with a letter
  - Don't:
    - No whitespace
    - Begin with a number
    - Use operator symbols like – between words

Python Math Commands

Command name	Description
`abs(`***value***`)`	absolute value
`ceil(`***value***`)`	rounds up
`cos(`***value***`)`	cosine, in radians
`floor(`***value***`)`	rounds down
`log(`***value***`)`	logarithm, base *e*
`log10(`***value***`)`	logarithm, base 10
`max(`***value1***`, `***value2***`)`	larger of two values
`min(`***value1***`, `***value2***`)`	smaller of two values
`round(`***value***`)`	nearest whole number
`sin(`***value***`)`	sine, in radians
`sqrt(`***value***`)`	square root

To use some of these you need to import the math library → from math import *

Python



Python For Data Science *Cheat Sheet*

Python Basics

Learn More Python for Data Science Interactively at www.datacamp.com

Variables and Data Types

Variable Assignment

```
>>> x=5
>>> x
5
```

Calculations With Variables

```
>>> x+2
7
>>> x-2
3
>>> x*2
10
>>> x**2
25
>>> x%2
1
>>> x/float(2)
2.5
```

Sum of two variables
Subtraction of two variables
Multiplication of two variables
Exponentiation of a variable
Remainder of a variable
Division of a variable

Types and Type Conversion

`str()`	`'5', '3.45', 'True'`	Variables to strings
`int()`	`5, 3, 1`	Variables to integers
`float()`	`5.0, 1.0`	Variables to floats
`bool()`	`True, True, True`	Variables to booleans

Asking For Help

```
>>> help(str)
```

Strings

```
>>> my_string = 'thisStringIsAwesome'
>>> my_string
'thisStringIsAwesome'
```

String Operations

```
>>> my_string * 2
'thisStringIsAwesomethisStringIsAwesome'
>>> my_string + 'Innit'
'thisStringIsAwesomeInnit'
>>> 'm' in my_string
True
```

Lists — Also see NumPy Arrays

```
>>> a = 'is'
>>> b = 'nice'
>>> my_list = ['my', 'list', a, b]
>>> my_list2 = [[4,5,6,7], [3,4,5,6]]
```

Selecting List Elements — Index starts at 0

Subset

```
>>> my_list[1]
>>> my_list[-3]
```

Select item at index 1
Select 3rd last item

Slice

```
>>> my_list[1:3]
>>> my_list[1:]
>>> my_list[:3]
>>> my_list[:]
```

Select items at index 1 and 2
Select items after index 0
Select items before index 3
Copy my_list

Subset Lists of Lists

```
>>> my_list2[1][0]
>>> my_list2[1][:2]
```

my_list[list][itemOfList]

List Operations

```
>>> my_list + my_list
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list * 2
['my', 'list', 'is', 'nice', 'my', 'list', 'is', 'nice']
>>> my_list2 > 4
True
```

List Methods

`>>> my_list.index(a)`	Get the index of an item
`>>> my_list.count(a)`	Count an item
`>>> my_list.append('!')`	Append an item at a time
`>>> my_list.remove('!')`	Remove an item
`>>> del(my_list[0:1])`	Remove an item
`>>> my_list.reverse()`	Reverse the list
`>>> my_list.extend('!')`	Append an item
`>>> my_list.pop(-1)`	Remove an item
`>>> my_list.insert(0,'!')`	Insert an item
`>>> my_list.sort()`	Sort the list

String Operations — Index starts at 0

```
>>> my_string[3]
>>> my_string[4:9]
```

String Methods

`>>> my_string.upper()`	String to uppercase
`>>> my_string.lower()`	String to lowercase
`>>> my_string.count('w')`	Count String elements
`>>> my_string.replace('e', 'i')`	Replace String elements
`>>> my_string.strip()`	Strip whitespaces

Libraries

Import libraries

```
>>> import numpy
>>> import numpy as np
```

Selective import

```
>>> from math import pi
```

pandas — Data analysis
learn — Machine learning
NumPy — Scientific computing
matplotlib — 2D plotting

Install Python

ANACONDA — Leading open data science platform powered by Python
spyder — Free IDE that is included with Anaconda
jupyter — Create and share documents with live code, visualizations, text, ...

Numpy Arrays — Also see Lists

```
>>> my_list = [1, 2, 3, 4]
>>> my_array = np.array(my_list)
>>> my_2darray = np.array([[1,2,3],[4,5,6]])
```

Selecting Numpy Array Elements — Index starts at 0

Subset

```
>>> my_array[1]
2
```

Select item at index 1

Slice

```
>>> my_array[0:2]
array([1, 2])
```

Select items at index 0 and 1

Subset 2D Numpy arrays

```
>>> my_2darray[:,0]
array([1, 4])
```

my_2darray[rows, columns]

Numpy Array Operations

```
>>> my_array > 3
array([False, False, False,  True], dtype=bool)
>>> my_array * 2
array([2, 4, 6, 8])
>>> my_array + np.array([5, 6, 7, 8])
array([6, 8, 10, 12])
```

Numpy Array Functions

`>>> my_array.shape`	Get the dimensions of the array
`>>> np.append(other_array)`	Append items to an array
`>>> np.insert(my_array, 1, 5)`	Insert items in an array
`>>> np.delete(my_array,[1])`	Delete items in an array
`>>> np.mean(my_array)`	Mean of the array
`>>> np.median(my_array)`	Median of the array
`>>> my_array.corrcoef()`	Correlation coefficient
`>>> np.std(my_array)`	Standard deviation

DataCamp
Learn Python for Data Science Interactively

COLUMBIA UNIVERSITY