

Introduction to Data Science with Python

MIT Political Science Methods Workshop

Soubhik Barari
Computational and Statistical Research Specialist
MIT Political Methodology Lab

February 9 2018

Pre-requisites:

- 1 Proficiency in R
- 2 Proficiency w/data analysis

Workshop materials (setup, slides, notebooks):

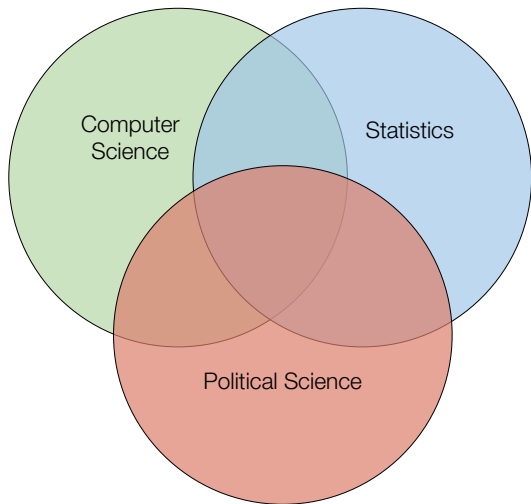
<https://github.com/soubhikbarari/MITPolMeth-PythonDataSci-02-2018>

Roadmap

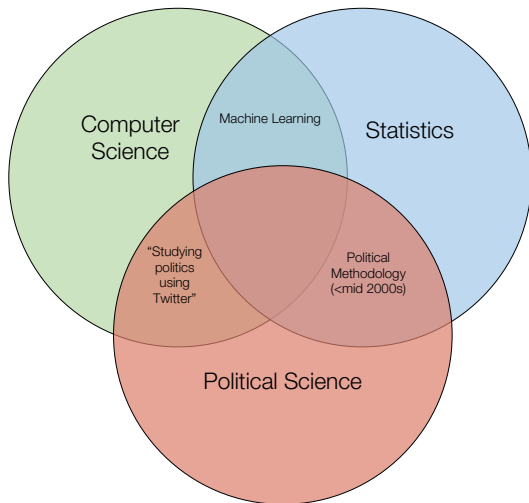
- 1 Overview [**slides**]
- 2 Python Fundamentals [**notebook**]
- 3 Python for Data Science [**notebook**]
- 4 Application: Analyzing Ideology in Congress [**script**]

Overview

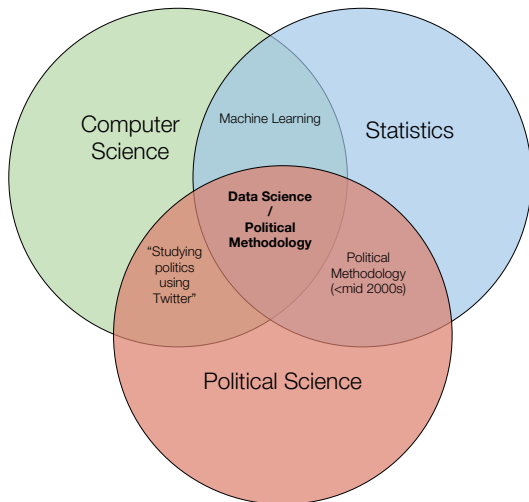
Overview of Data Science



Overview of Data Science



Overview of Data Science



***Not pictured in diagram:** your award-winning, tenure-track-faculty-position-garnering thesis dataset!

- **“Inferring Roll Call Scores from Campaign Contributions Using Supervised Machine Learning”** (Adam Bonica, 2017)
- **“Dynamic Estimation of Latent Opinion Using a Hierarchical Group-Level IRT Model”** (Devin Caughey et al., 2015)
- **“Measuring Trade Profiles with Two Billion Observations of Product Trade”** (Kosuke Imai et al., 2017)
- **“A New Automated Redistricting Simulator Using Markov Chain Monte Carlo”** (Ben Fifield et al., 2018)

Why use Python for Data Science

Because it...

- ① is easy to dig into lower levels (if you want).
- ② is able to do *many* different things.
- ③ is relatively easy to learn.
- ④ has *the best* machine learning toolkit.
- ⑤ is great for building re-usable things.

Python is a programming language that is

① **Object-oriented**

```
model = library.CreateModel()  
model.fit(data)
```

② **Functional**

```
Y = map(lambda y: y**2, filter(lambda x: x < 5, X))
```

③ **Dynamically typed**

```
myVar = getNewData()
```

Python is a programming language that is

① **Object-oriented** (organized!)

```
model = library.CreateModel()  
model.fit(data)
```

② **Functional** (clean!)

```
Y = map(lambda y: y**2, filter(lambda x: x < 5, X))
```

③ **Dynamically typed** (flexible!)

```
myVar = getNewData()
```

Python vs. R

Python relies more on packages.

R

```
df <- read.csv("data.csv")
```

Python

```
import pandas as pd  
df = pd.read_csv("data.csv")
```

Python vs. R

Python is more object-oriented.

R

```
fit <- lm(y ~ x, data=df)
predictions <- predict(fit, test)
```

Python

```
from sklearn.linear_model import LinearRegression
lm = LinearRegression()
lm.fit(train["x"], train["y"])
predictions = lm.predict(test["x"])
```

Python vs. R

...however **Python** can do anything functional that **R** can in a simpler way (e.g. `lapply`, `vapply`, `shmapply`... is just `map!`).

R

```
lapply(a_matrix, function(x) length(unique(x)))  
Filter(function(x) !is.numeric(x), a_matrix)
```

Python

```
map(lambda x: len(set(x)), a_matrix)  
filter(lambda x: type(x) != int, a_matrix)
```

Python has better support for non-statistical tasks.

R

```
# Web-scraping basketball statistics

library(rvest)
page <- read_html(url)
table <- html_nodes(page, ".stats_table")[3]
rows <- html_nodes(table, "tr")
cells <- html_nodes(rows, "td a")
teams <- html_text(cells)

extractRow <- function(rows, i){
  if(i == 1){
    return
  }
  row <- rows[i]
  tag <- "td"
  if(i == 2){
    tag <- "th"
  }
  items <- html_nodes(row, tag)
  html_text(items)
}

scrapeData <- function(team) ...
```

Python

```
# Web-scraping basketball statistics

from bs4 import BeautifulSoup
import re

soup = BeautifulSoup(data, 'html.parser')
box_scores = []
for t in soup.find_all(id=re.compile ...
    rows = []
    for i, row in enumerate(t.find_all("tr")):
        if i == 0:
            continue
        elif i == 1:
            t = "th"
        else:
            t = "td"
        rd = [item.get_text() for item in ... ]
        rows.append(rd)
    box_scores.append(rows)
```

Python vs. R: Trade-offs

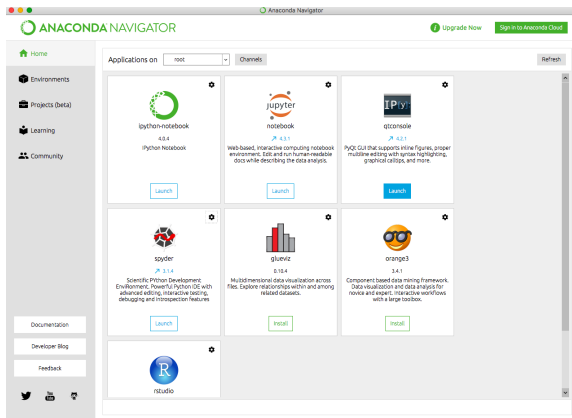
- Building tools vs. Doing analysis
- Flexibility vs. Convenience
- Speed vs. Parallelizability
- 'Computational' vs. 'Statistical'
- Great machine learning vs. Ok machine learning

Example Python Use Cases

- Build an end-to-end pipeline that automatically scrapes web data, runs analysis, and saves results.
- Write a slightly customized version of a standard machine learning algorithm using the `scikit-learn` framework.
- Work with (i.e. analyze, model, visualize) political text documents, audio data, images, or videos.

Running Python

First, install the **Anaconda** distribution of Python^{**}:



^{**}Instructions can be found on [setup.pdf](https://github.com/soubhikbarari/MITPolMeth-PythonDataSci-02-2018) at <https://github.com/soubhikbarari/MITPolMeth-PythonDataSci-02-2018>.

Running Python

A. Command line (Terminal / Command Prompt)

```
soubhikbarari@dhcp-18-189-85-156 : ipython
Python 2.7.13 |Anaconda custom (x86_64)| (default, Dec 20 2016, 23:05:08)
Type "copyright", "credits" or "license" for more information.

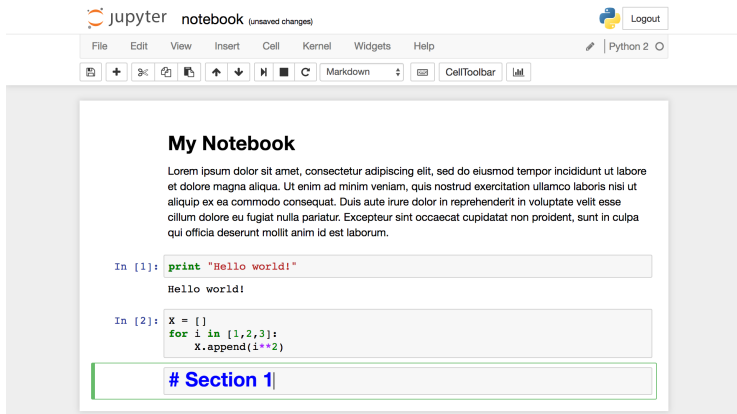
IPython 5.1.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help             -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: print "Hello world!"
Hello world!

In [2]: |
```

Running Python

B. Notebook (Jupyter)



The screenshot displays the Jupyter Notebook interface. At the top, the title bar reads "jupyter notebook (unsaved changes)" with a "Logout" button on the right. Below the title bar is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, and Help. A secondary bar shows "Python 2" and a pencil icon. The main toolbar contains icons for file operations, cell navigation, and execution. The notebook content is as follows:

My Notebook

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

```
In [1]: print "Hello world!"
```

Hello world!

```
In [2]: X = []
for i in [1,2,3]:
    X.append(i**2)
```

```
# Section 1
```

Running Python

C. IDE (Spyder)

The screenshot displays the Spyder Python IDE interface. The main window is titled "spyder" and shows a code editor on the left, a variable explorer on the right, and a Python console at the bottom.

Code Editor: The code editor shows a Python script named "feature_step_1ml.py". The code includes comments and function calls for data reading and preprocessing. Key lines include:

```
74 #
75 # ----- MEDN -----
76 #
77 #
78 #
79 #####
80 ## Read data ##
81 #####
82
83 IP = pd.read_pickle('E:\ARML\DATA\IP.pkl')
84 print "OK reading in data."
85 #####
86 ## preprocessing data ##
87 #####
88
89 for txt_name in ['url', 'link_text', 'page_title', \
90                'pdf_filename', 'pdf_reader_text']:
91     print "... preprocessing %s" % txt_name
92     DF[txt_name] = DF[txt_name].apply(clean_fm, 1)
93
94
95 for txt_name in ['tag', 'tag2']:
96     print "... preprocessing %s" % txt_name
97     DF[txt_name] = DF[txt_name].apply(lambda s: proc_fm(s), 30HC(), 1)
98
99 print "OK preprocessing data."
100
101 #####
102 ## Analyze 'distributions' ##
103 #####
104
105 #####
```

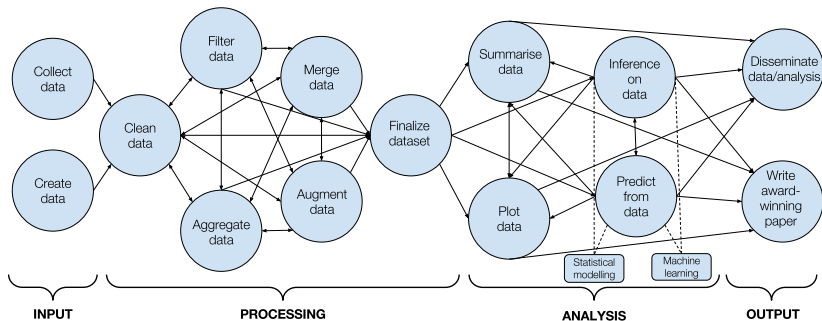
Variable Explorer: The variable explorer shows a table with columns "Type", "Size", and "Value". It is currently empty.

Python Console: The Python console displays a message: "NOTE: The Python console is going to be REMOVED in Spyder 3.2. Please start to migrate your work to the IPython console instead." Below this, it shows the Python version (2.7.13) and Anaconda version (0.84.0). It also provides instructions on how to type help, copyright, or license information, and a link to the Anaconda website. The console prompt is >>> IP >>> "test".

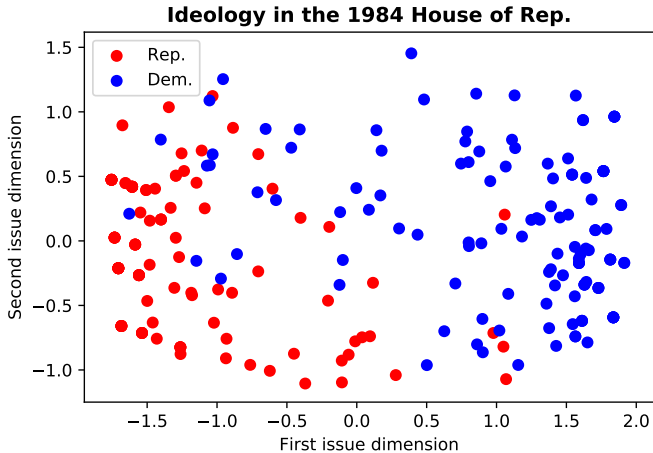
Python Fundamentals

(<https://github.com/soubhikbarari/MITPolMeth-PythonDataSci-02-2018/blob/master/fundamentals.ipynb>)

Data Science Pipeline



Ex: Discovering Political Ideology with Machine Learning



Python for Data Science

(<https://github.com/soubhikbarari/MITPolMeth-PythonDataSci-02-2018/blob/master/datasci.ipynb>)

Application: Analyzing Ideology in Congress

(<https://github.com/soubhikbarari/MITPolMeth-PythonDataSci-02-2018/blob/master/congress-analysis.py>)

Other Jupyter notebooks:

- 1 Scientific Computing with Python:
<https://github.com/jrjohansson/scientific-python-lectures>
- 2 Python Data Science Handbook:
<https://github.com/jakevdp/PythonDataScienceHandbook>

At MIT:

- 1 MIT Libraries : Data Consultation Services
- 2 Harvard-MIT Data Center Cluster
- 3 XVII : MIT Political Methodology Lab Computing Cluster
- 4 sbarari@mit.edu