Python for Public Policy

PADM-GP 4506

Who am I?

- NYU Wagner Master of Urban Planning
- **Current:** Product Manager & Web Developer
- **Former:**
  - Data analyst / strategist focused on NYC's capital planning and land use
  - Environmental engineer
- **Passion:** Using data analytics and civic tech to help cities operate more efficiently, sustainably, and equitably
- **Mindset:** Technology = a means to an end

Why learn Python?

- Much faster manipulation of large datasets
- Easy to change variables in an analysis methodology and then quickly rerun
- Can automate tasks, like data cleanup or producing charts
- Documents steps and assumptions
- Reproducibility
- Many resources online - rapidly growing community of data analysts, especially data scientists

What will this course provide?

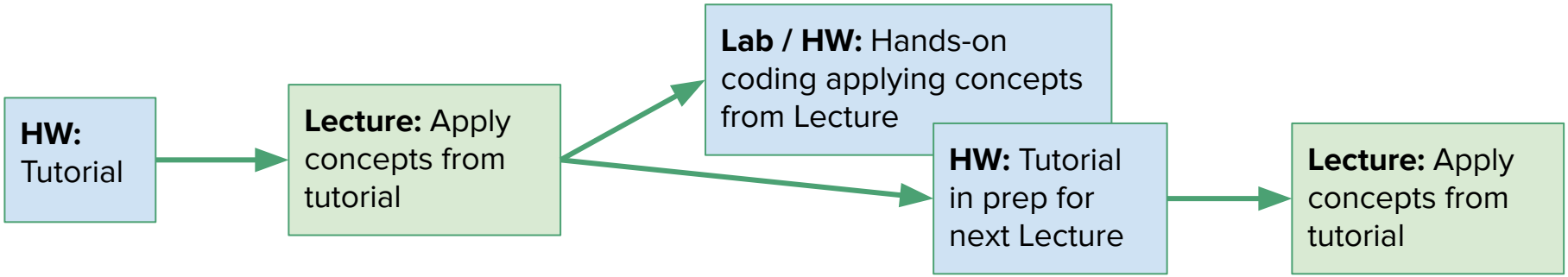
- Relevant, concrete applications of Python toward policy research questions
- Structured, in-person support from a professor
- Community of fellow students to work with

Class format

- **Lecture:** For the first ~40 minutes of class, the professor will work through examples using Python and Python packages to analyze data focused on public policy questions
    - It's better if you **don't** code along during this portion
- **Lab:** For the remainder of the class period, students will work on their coding homework assignments, either individually or in groups

Homework assignments

- **Online tutorials:** In advance of classes, online tutorials will be assigned as homework. The following lecture will focus on applying those concepts
- **Coding focused on data exploration, analysis, and visualization:** Students will complete Python coding exercises that apply new concepts covered in lecture



Course schedule

Date	Topics	Homework
10/24	Intro to Python, Google Colab Notebook, & GitHub. Opening data and exploring the contents	HW0 due (survey & GitHub); HW1 assigned
10/31	Python data structures and Pandas. Data cleanup, iterating with "for" loops, basic summary stats	HW1 due; HW2 assigned
11/07	Manipulating and combining Pandas DataFrames, writing functions	HW2 due; HW3 assigned
11/14	Data visualization	HW3 due; HW4 assigned
11/21	Linear regression and inferential statistics	HW4 due; HW5 assigned
11/28	No class: Thanksgiving holiday	
12/05	Working with dates and time series analysis	HW5 due; HW6 assigned
12/12	Review course content and topics of student choice	HW6 due; HW7 assigned - **Due 12/19**

Grade

Participation: 10%

8 Assignments: 90%

- HW0: 0%
- HW1: 12.5%
- HW2: 12.5%
- HW3: 12.5%
- HW4: 12.5%
- HW5: 12.5%
- HW6: 12.5%
- HW7: 15%

What is Python? Let’s demystify it

- Python is a language; it’s just text that your computer understands
- A Python script is just a text file
- Popular for data analysis and data science
- Packages aka Libraries
    - Developers have created “packages” for Python with advanced functionality so you don’t have to create it yourself
    - Packages are like free software plug-ins that you “import”
    - Packages we’ll use: pandas, numpy, matplotlib, sklearn, and more

Python can be run in many environments

On your computer

- Text editor (.py) + command line
- Jupyter notebook (.ipynb)

Installation package

Notebook

In notebooks on the cloud

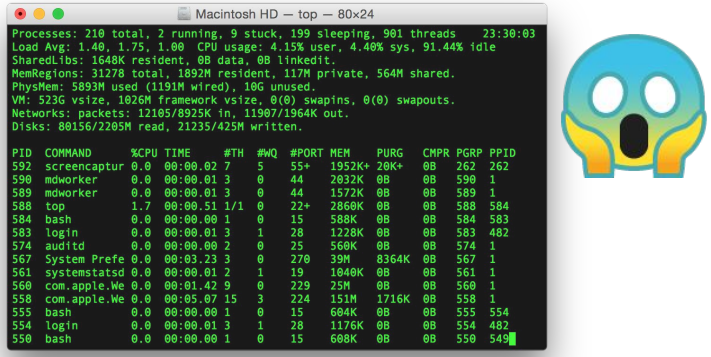
- Google Colab (used in this class)
- Mode (we'll use some of their tutorials)
- Many more!

Included in Google Suite

Better fit for enterprise analytics teams

Google Colab Notebooks

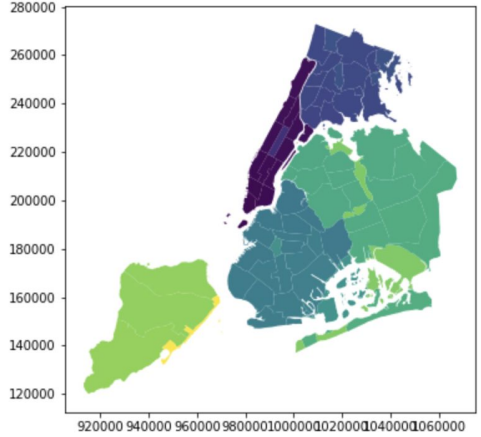
- A web based programming environment for Python
- Nicely displays output of your Python code so you can check and share the results
- Connects with Google Drive
- Avoids using the command line and problems with installing packages on different computer operating systems



vs.

```
[ ]  import matplotlib.pyplot as plt
     from matplotlib.pyplot import figure
     fig, ax = plt.subplots(figsize=(14, 6))
     gdf.plot(column='BoroCD',
                      ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fa40b72e908>
```



Demo

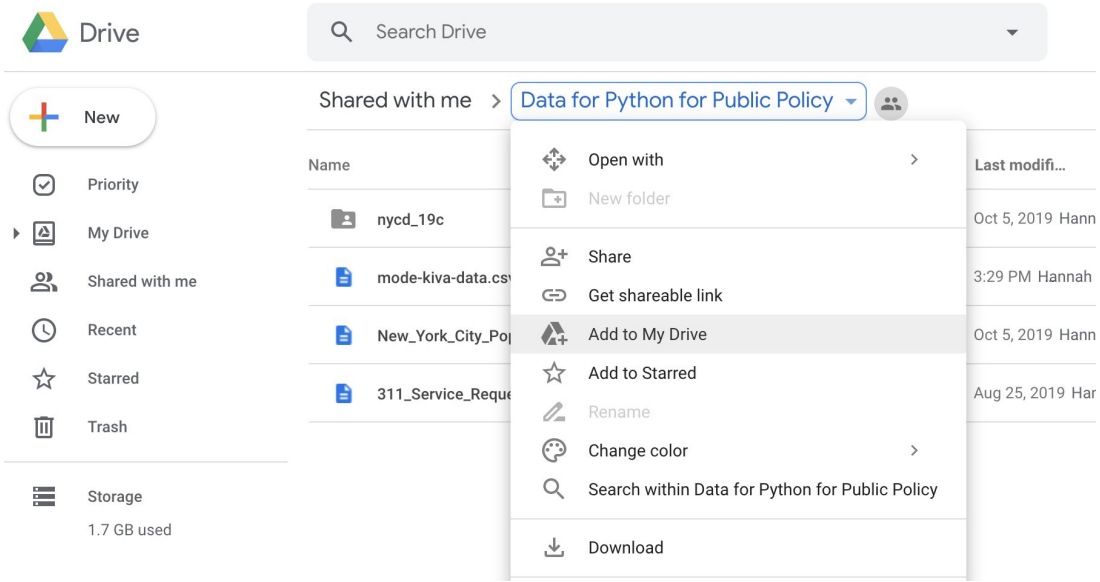
Creating and running a simple python script in the command line

vs.

Running same python code in Google Colab notebook

Setup: Google Drive

1. Login to Google Drive using your NYU account
2. Go to “Shared with me”
3. Move the “**Data for Python**” folder into “My Drive”



Setup: Colab Notebook + Data from Drive

1. Go to **colab.research.google.com**
2. Click on “New Python 3 Notebook”
3. Use the code below to load a dataset from the “Data for Python” folder:

```
import pandas as pd
from google.colab import drive
drive.mount('/content/drive')

# follow the link it generates, choose your NYU account, and then paste in the authorization code it provides

df = pd.read_csv('/content/drive/My Drive/Data for Python/zoo.csv', header='infer')
```

Example notebook : https://colab.research.google.com/drive/1nP_4NfBpHfGbguAosam7ECPqrB5Qq9ZL

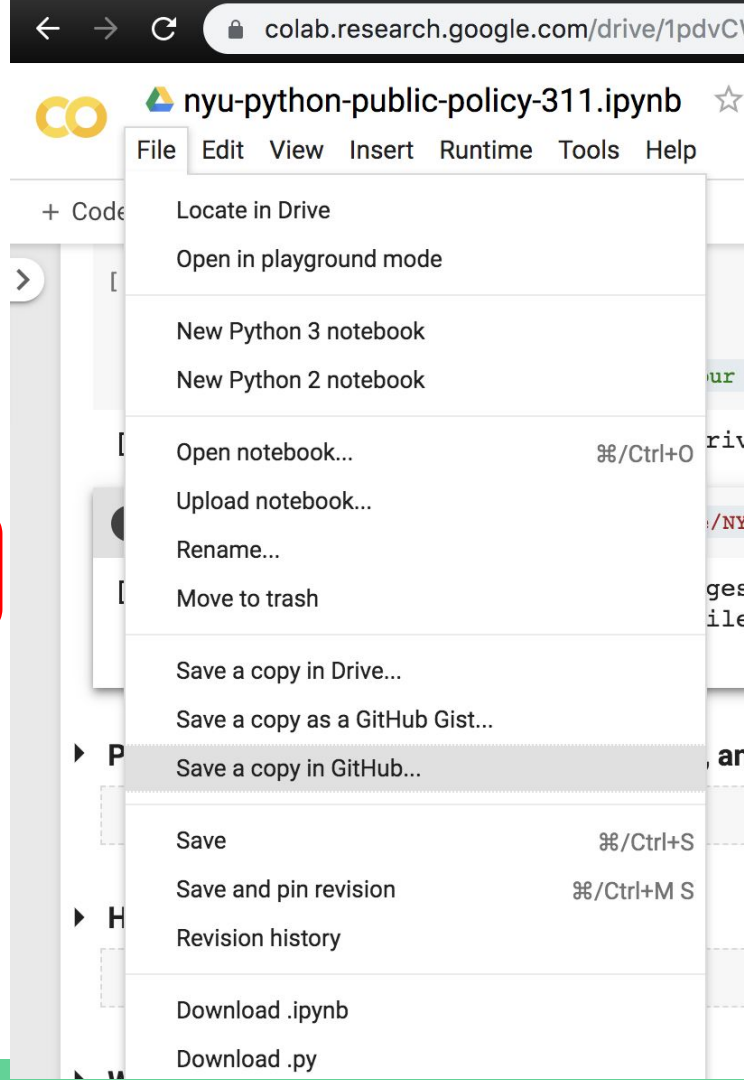
GitHub

- Website for sharing and managing code
- Why?
  - Transparency! Enables people to reproduce your analysis
  - Version control
  - A public portfolio of your work

Setup: GitHub repo + Colab

1. Login to GitHub.com
2. Create a new public repository called “**nyu-python-public-policy**”

3. Go back you your Colab notebook
4. Save a copy in your “**nyu-python-public-policy**” repo
   a. Note: This is how you will submit your homework
5. Refresh GitHub.com to see that a .ipynb file has been added to your repo



Basics: Data Types

- **“String”** text
- **“Boolean”** True/False
- **“Integer”** number without decimal places. Ex: 43
- **“Float”** number with decimal places. Ex: 43.672
- **“List”** ordered list of objects, denoted with square brackets [ ]
  - Ex: [‘Harry’, ‘Fred’, ‘George’]
- **“Dictionary”** ordered list of key: value objects, denoted with curly brackets { }
  - Ex: {name: ‘Harry’, age: 28, house: ‘Gryffindor’}
- **“Tuple”** ordered list of objects that cannot be changed, denoted with ( )
  - Ex: (‘40.7128° N’, ‘74.0060° W’)

Basics: Indexes

An index is a number value that represents the location of an item in a list, string, or table, etc.

Indexes start at 0. Ex: a list containing 5 items has indexes [0, 1, 2, 3, 4]

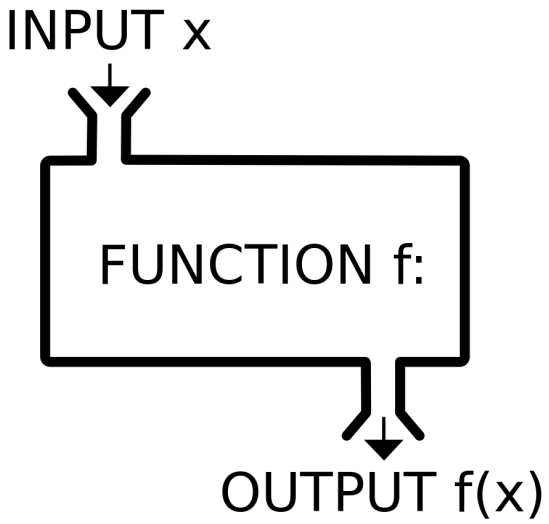
Example: [‘hey’, ‘how’, ‘are’, ‘you’]

The index of ‘how’ is 1.

Basics: Functions

Think back to algebra....

A function takes in a variable, applies calculations or logic to that variable, and provides an output



Python 2 vs. Python 3

Python 2 (used in Mode tutorials)

print thing_i_want_to_print

vs.

Python 3 (used in my examples)

print(thing_i_want_to_print)