BINGHAMTON
UNIVERSITY
STATE UNIVERSITY OF NEW YORK

Political Analysis with Python

Öner Yigit

Week 1

July 9, 2023

Binghamton University (SUNY)

Announcements

A few notes before we start!

Section 1

Python

- A **general-purpose** programming language
- by Guido van Rossum in 1991
- Python 3 released in 2008
- the goal was designing an easy-to-read programming language.
- base language and libraries

Popularity

TIOBE
(the software quality company)

About us | News | Coding Standards | TIOBE Index | Contact

Products | Quality Models | Markets | Schedule a demo

Jun 2023	Jun 2022	Change	Programming Language	Ratings	Change
1	1		Python	12.46%	+0.26%
2	2		C	12.37%	+0.46%
3	**4**	↑	**C++**	**11.36%**	**+1.73%**
4	3	↓	Java	11.28%	+0.81%
5	5		C#	6.71%	+0.59%
6	6		Visual Basic	3.34%	-2.08%
7	7		JavaScript	2.82%	+0.73%
8	13	⇈	PHP	1.74%	+0.49%
9	8	↓	SQL	1.47%	-0.47%
10	9	↓	Assembly language	1.29%	-0.56%
11	12	↑	Delphi/Object Pascal	1.26%	-0.07%
12	24	⇈	MATLAB	1.11%	+0.48%
13	25	⇈	Scratch	1.02%	+0.43%
14	15	↑	Go	1.00%	-0.02%
15	26	⇈	Fortran	0.99%	+0.44%
16	11	⇊	Classic Visual Basic	0.96%	-0.36%
17	16	↓	R	0.94%	-0.04%

Python is free

- No need to pay for software like Stata, SPSS or SAS.
- Open-source
- Supportive community
- New packages are being developed constantly for free.

Python as English

- Python syntax is easy to read compared to other languages.
- That is why many people start with Python

Anything!!!

Essentially, you can do anything with Python!

Python allows people to use it for to a wide variety of tasks.

Here are some common applications:

- web scraping
- automation
- machine learning
- web development
- statistical modeling
- programming
- visualization

Data

Python can handle various data types.

- numeric
- text
- video
- image
- audio
- geospatial
- many more...

Once you understand basics

You will begin to notice what you can do using Python for your own projects!

Example: I had to do a transcription of audio files that were 1 hour long. Python did it for me!

To begin

So for beginners, installing Python can be challenging. In fact, it can be intimidating.

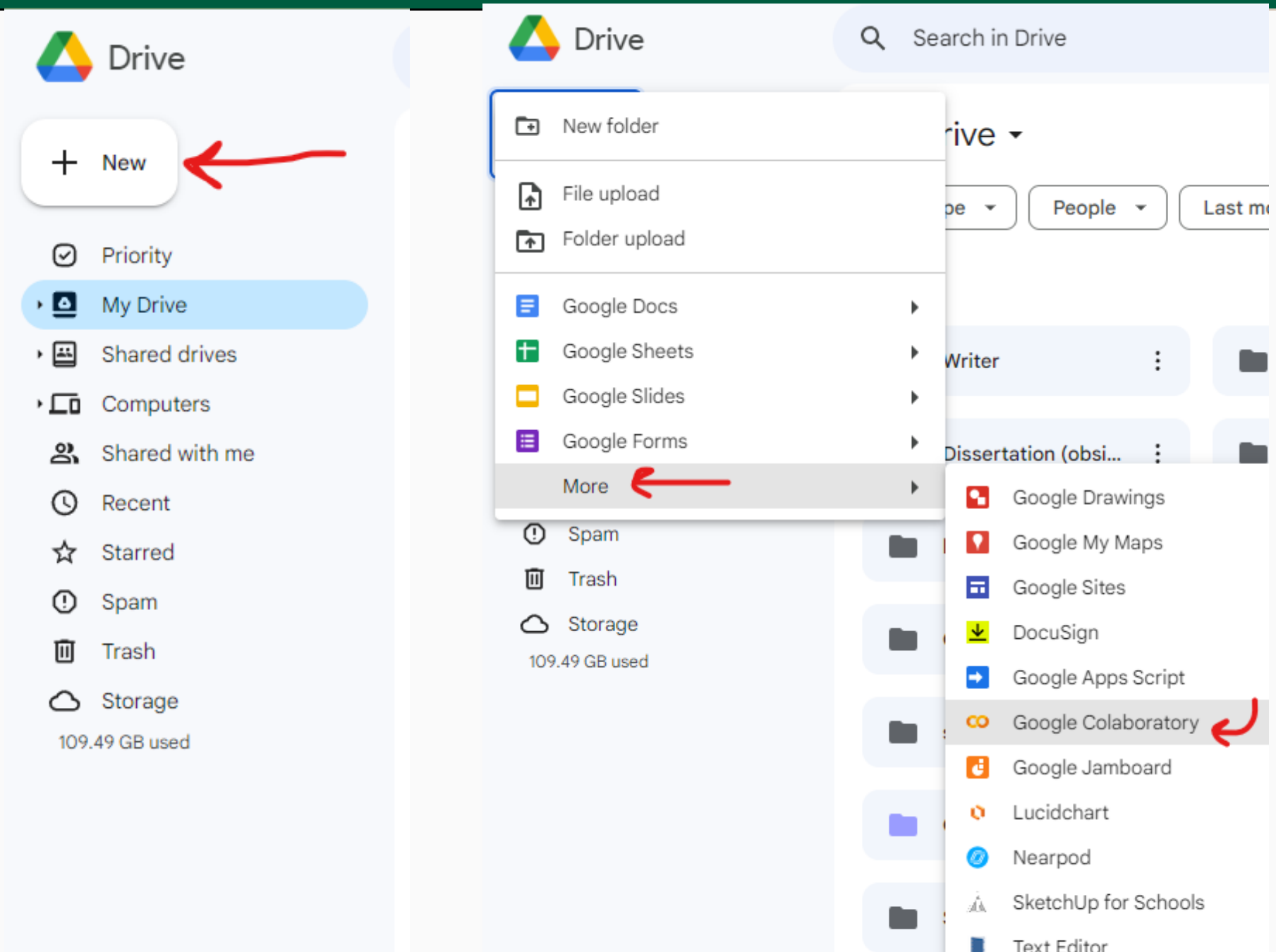
At this time, I do not recommend beginners to deal with Python installation in their computer.

Google has invested in developing software for using Python without dealing with all the potential issues when installing Python.

We will use Google Colaboratory (known as Colab) through our Google Drive.

Let's have a look at Colab.

Colab


Drive
New
Priority
My Drive
Shared drives
Computers
Shared with me
Recent
Starred
Spam
Trash
Storage
109.49 GB used
Drive
Search in Drive
New folder
File upload
Folder upload
Google Docs
Google Sheets
Google Slides
Google Forms
More
Spam
Trash
Storage
109.49 GB used
People
Writer
Dissertation (obsi...
Google Drawings
Google My Maps
Google Sites
DocuSign
Google Apps Script
Google Colaboratory
Google Jamboard
Lucidchart
Nearpod
SketchUp for Schools
Text Editor

File extentions

Python files end with **.py** or **.ipynb**.

.py is great and can be accessed more easily using simple Notepad or TextEdit apps. However, we will be mostly using .ipynb for learning purposes.

Markdown

It is important to be familiar with markdown essentials. Let's jump into Google Colab to see some examples.

Section 2

Hello World!

Let's start coding :)

Of course, we start with `print()` function.

```
print("hello world")
```

```
hello world
```

Basic Data Types

Python manipulates data objects, and objects have types.

Four main object types.

```
#str: string(text, characters)
type("hello world")
#int: integers(Whole numbers)
type(5)
#float: float(Numbers with a decimal point)
type(5.2)
#bool: boolean(Logical value True or False)
type(4+2==9-3)
```

We can look type of Python object with `type()`

Basic Math

```
print(1+1)
2

print(3*2)
6

print(((2 + 3) * (5 + 5))
50

print((4+2==9-3)
True

print((1 == 2) or (2 == 3) or (4 == 4))
True
```

Operators

Operator	Name	Example
$+$	Addition	$\mathrm{x} + \mathrm{y}$
$-$	Subtraction	$\mathrm{x} - \mathrm{y}$
$*$	Multiplication	$\mathrm{x} * \mathrm{y}$
$/$	Division	x/y
%	Modulus	$x\%\mathrm{y}$
$**$	Exponentiation	$\mathrm{x} * *\mathrm{y}$
$//$	Floor division	$\mathrm{x}//\mathrm{y}$

Variables

```
#We assign values to variables with =
num1 = 1.5
num2 = 6.3
# Add two numbers
sum = num1 + num2
# Display the sum
print(sum)
7.8
```

Example

```
pi = 3.14159
radius = 5
print(pi)
print(radius)
3.14159
5
area = pi * (radius ** 2)
print(area)
78.53975
circumference = 2 * pi * radius
print(circumference)
31.4159
```

In Python, = means assignment not equal to.

```
x=10
print(x)
10
x=x+1
print(x)
11
```

Storing data

Base Python can store data objects in lists, dictionaries, tuples and sets.

```
list :[10, "hello", 200.3]
dictionary: {"favcolor": "green", "name": "Oner"}
tuples: (1,"hi",90.3)
sets: {"x","y"}
```

Notice the parentheses.

lists

```
# Creating a list
fruits = ['apple', 'banana', 'orange', 'grape']
 my_list = ['a','b','c']
 my_list.append('d')
 print(my_list)
 ['a', 'b', 'c', 'd']

 print(my_list[0])
 a
```

Lists are very useful. We can modify it.

dictionary

```
  # dictionary
d = {'key1':'item1','key2':'item2'}

# Creating a dictionary
student = {
    'name': 'John Smith',
    'age': 20,
    'major': 'Computer Science',
    'university': 'ABC University'}

print(student['major'])
Computer Science
```

tuples

```
# Creating a tuple
student = ('John Smith', 20, 'Computer Science', 'ABC Unive

t = (1,2,3)
print(t[0])
1

#this will give error
# because we cannot change tuples
t[0] = 'NEW'
TypeError:
```

sets

```
    # Creating a set
fruits = {'apple', 'banana', 'orange', 'grape'}

my_set={1,2,3,1,2,1,2,3,3,3,3,2,2,2,1,1,2}
print(my_set)
{1,2,3}
```

Python Objects

	Mutable	Ordered	Indexing / Slicing	Duplicate Elements
List	✓	✓	✓	✓
Tuple	✗	✓	✓	✓
Set	✓	✗	✗	✗

Source: link

Next Week

Next week, we will look at logical statements, loops and functions.
It will still be the base Python.