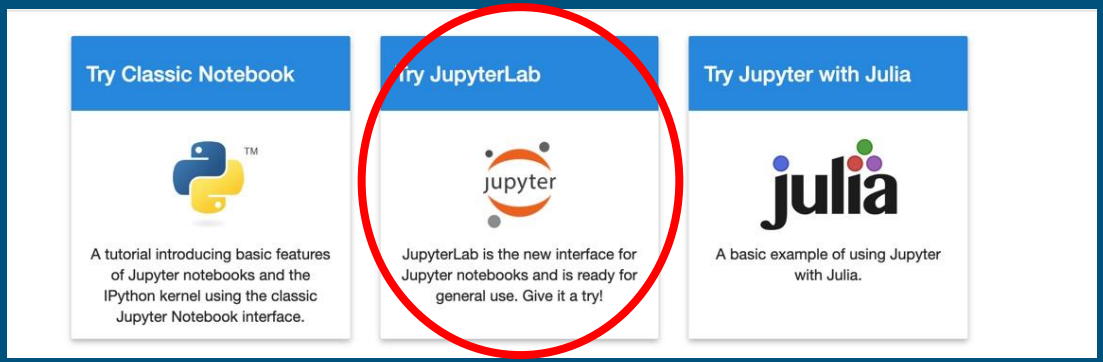
Use Python to create basic visualizations with data!

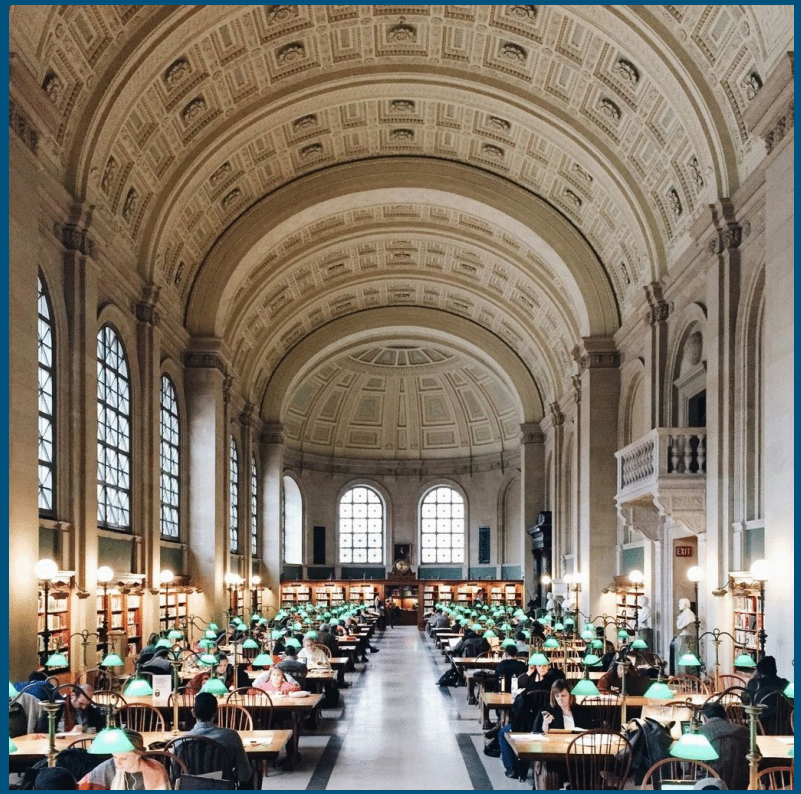
What you will need:

On your device go to https://jupyter.org/try and click “Try JupyterLab”.



Libraries Upload:

First, we need to upload the libraries we will be using for this project. Every programming language has different "classes" that you need to use to access specific variables. The same thing occurs in Python, but instead they are known as libraries.



Libraries Upload Continued:

In this workshop we will be using: pandas, Numpy, and matplotlib.

Pandas: This is not the furry animal you are thinking of, pandas is a efficient and easy way to use open source data analysis.

Matplotlib: This is a plotting library using a numerical mathematics extension Numpy.

Numpy: This is a python library that is very good with hard mathematical functions and arrays.

Let’s Make a Simple Line Plot!

Line Plot:

```
#Example for Simple Line Plot

#importing libraries
import pandas as pd
import matplotlib.pyplot as plt

#creating x and y axis
x=[x*2 for x in range (100)]
y=[y*2 for y in range (100)]

#plotting function
plt.plot(x,y)

#showing graph
plt.show()
```

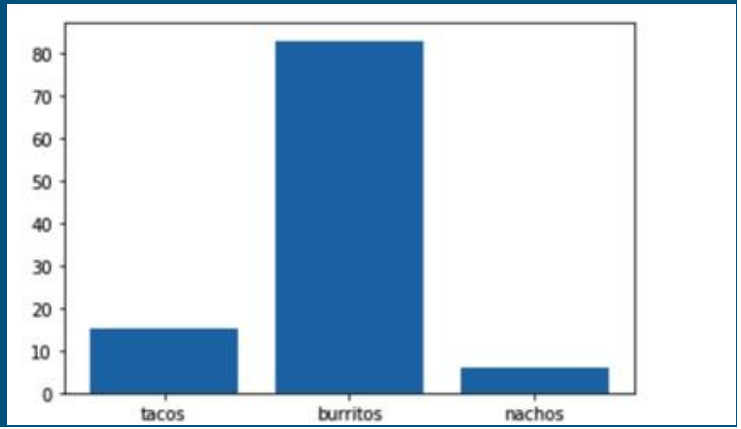
Customize the x and y how you want!

This line shows the data + graph

Let’s Make a Simple Bar Graph!

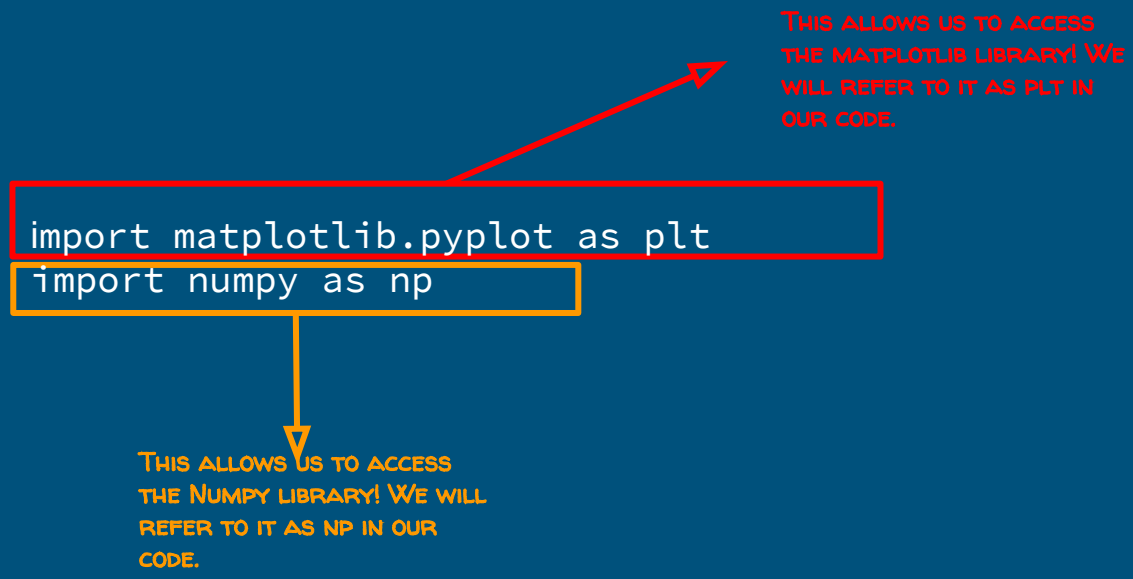
Bar Graph:

```
import pandas as pd
import matplotlib.pyplot as plt
import random

x=['Tacos','Burritos','Churros']

y=[random.randint(0,30),
random.randint(0,90),random.randint(0,10)]

plt.bar(x,y)

plt.show()
```



Project Creation!
1. Importing libraries :)

What does this look like?



Creating the Graph!

To create the graph we will be using the matplotlib library! It will...

- Create outline
- Create axes
- Create axes ranges
- Allows us to plot points

Look @ the Comments in the Code! Also, feel free to customize your graph, this Code is just a sample!

What does this look like?

```
#creates graph/plot
fig,ax= plt.subplots()

#plots points
ax.plot([1,2,3,4],[1,4,2,3])

plt.show()
```

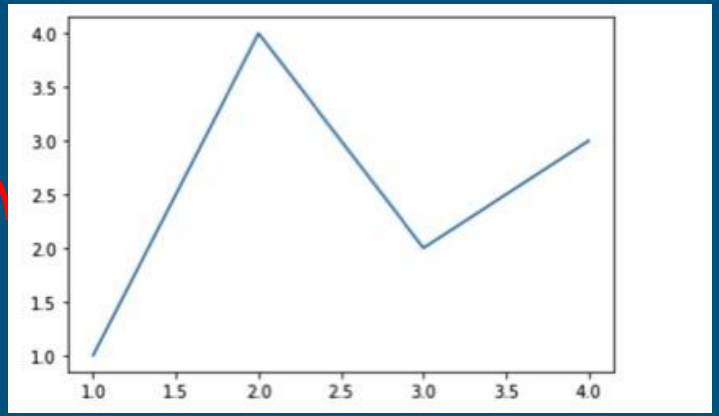


Figure (Fig).

In the code above, and anytime you want to create a graph you will be using ***fig.***

- For example in our code so far we have used...
- This created a figure with a single axes.
- Here are a few different ways you can use fig...

```
fig,ax= plt.subplots()
```

```
fig=plt.figure() #creates empty graph returns with num. figues

fig,ax =plt.subplots(2,2) #creates 2  graph figues
```

Let’s make it a little more advanced! :)

Let's create our own graphs...

1. Set up the spacing...
2. Example...

```
x=np.linspace(0,2,100)
```

**Make sure your libraries are imported!

Create a figure!

Try to do this on your own, but click the slide to see an example if needed!

Hint: fig,ax= plt._____()

```
fig,ax= plt.subplots()
```

Plots some lines!

This is an example of three simple line plots…

```
ax.plot(x,x,label='Sleep')
ax.plot(x,x**2,label='Worry if sky is
falling...')
ax.plot(x,x**3,label='Be a icon')
```

Plots simple cubed functions. Feel free to change it up!

Customize these labels!

Setting up the Table (Finishing touches!)

```
ax.set_xlabel('Time') #adds X axis label

ax.set_ylabel('Years')#add Y axis label

ax.set_title("Chicken Little's Plot") #adds a title

ax.legend() #adds a key of the graph
```

Names your axes, and your plot!

Example Plot w/ Code!

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np

x=np.linspace(0,2,100)

fig,ax= plt.subplots()


ax.plot(x,x,label='Sleep')

ax.plot(x,x**2,label='Worry if sky is falling...')

ax.plot(x,x**3,label='Be a icon')

ax.set_xlabel('Time') #adds X axis label

ax.set_ylabel('Years')#add Y axis lable

ax.set_title("Chicken Little's Plot") #adds a title

ax.legend() #adds a key of the graph
```

PS. If you have never watched Chicken Little

You must do that asap, you are missing out on a animated masterpiece.

Add on!

- Look up how to link in raw data from the web into your graphs
  - Hint: start off w/ *pd.read_csv()*
- Heavily personalize your graph!
- Google code for different kinds of graphs (dot plot, pie charts, etc.)
- Make multiple different graphs (the set of four/set them up with figure.
- Continue to play around!
- Google!

  - If any one wants a challenge please let me know or need more time...

Challenge!

Visualizing Covid-19 Data

Nic Piepenbreier

Click to Follow along, or follow along with slides.

Section One:

Importing Libraries

```
#Source: Nik Piepenbreier

# Section 1 - Loading our Libraries
import pandas as pd
import matplotlib.pyplot as plt
from matplotlib.dates import DateFormatter
import matplotlib.ticker as ticker
%matplotlib inline

# Section 2 - Loading and Selecting Data

df = pd.read_csv('https://raw.githubusercontent.com/datasets/covid-19/master/data/countries-aggregated.csv'
parse_dates=['Date'])

countries = ['Canada', 'Germany', 'United Kingdom', 'US', 'France', 'China']
df = df[df['Country'].isin(countries)]

# Section 3 - Creating a Summary Column
df['Cases'] = df[['Confirmed', 'Recovered', 'Deaths']].sum(axis=1)
```

Importing CSV File, the raw data. From Github

Section Two:

Pivoting data frame,
and creating columns.

```
#Source: Nik Piepenbreier
# Section 4 - Restructuring our Data
df = df.pivot(index='Date', columns='Country', values='Cases')
countries = list(df.columns)
covid = df.reset_index('Date')
covid.set_index(['Date'], inplace=True)
covid.columns = countries

# Section 5 - Calculating Rates per 100,000
populations = {'Canada':37664517, 'Germany': 83721496 , 'United Kingdom': 67802690 ,
'US': 330548815, 'France': 65239883, 'China':1438027228}

percapita = covid.copy()
for country in list(percapita.columns):
    percapita[country] = percapita[country]/populations[country]*100000
```

Section 3:

Plotting

```
#Source: Nik Piepenbreier
# Section 6 - Generating Colours and Style
colors = {'Canada':'#045275', 'China':'#089099', 'France':'#7CCBA2',
'Germany':'#FCDE9C', 'US':'#DC3977', 'United Kingdom':'#7C1D6F'}
plt.style.use('fivethirtyeight')

# Section 7 - Creating the Visualization
plot = covid.plot(figsize=(12,8), color=list(colors.values()), linewidth=5, legend=False)
plot.yaxis.set_major_formatter(ticker.StrMethodFormatter('{x:,.0f}'))
plot.grid(color='#d4d4d4')
plot.set_xlabel('Date')
plot.set_ylabel('# of Cases')

# Section 8 - Assigning Colour
for country in list(colors.keys()):
    plot.text(x = covid.index[-1], y = covid[country].max(), color = colors[country], s = country, weight = 'bold')

# Section 9 - Adding Labels
plot.text(x = covid.index[1], y = int(covid.max().max())+45000, s = "COVID-19 Cases by Country",
fontsize = 23, weight = 'bold', alpha = .75)

plot.text(x = covid.index[1], y = int(covid.max().max())+15000, s =
"For the USA, China, Germany, France, United Kingdom, and Canada\nIncludes Current Cases, Recoveries, and Deaths",
fontsize = 16, alpha = .75)

plot.text(x = percapita.index[1],
y = -100000,s = 'datagy.io Source: https://github.com/datasets/covid-19/blob/master/data/countries-aggregated.csv',
fontsize = 10)
```

Adding details and labels

Section 4 (visualization and color):

```
#Source: Nik Piepenbreier
percapitaplot = percapita.plot(figsize=(12,8), color=list(colors.values()), linewidth=5, legend=False)
percapitaplot.grid(color='#d4d4d4')
percapitaplot.set_xlabel('Date')
percapitaplot.set_ylabel('# of Cases per 100,000 People')
for country in list(colors.keys()):
    percapitaplot.text(x = percapita.index[-1], y = percapita[country].max(),
                       color = colors[country], s = country, weight = 'bold')

percapitaplot.text(x = percapita.index[1], y = percapita.max().max()+25,
                   s = "Per Capita COVID-19 Cases by Country", fontsize = 23, weight = 'bold', alpha = .75)

percapitaplot.text(x = percapita.index[1], y = percapita.max().max()+10,
                   s = "For the USA, China, Germany, France, United Kingdom, and Canada\nIncludes Current Cases,
                   "Recoveries, and Deaths", fontsize = 16, alpha = .75)

percapitaplot.text(x = percapita.index[1],
    y = -55,s = 'datagy.io Source: https://github.com/datasets/covid-19/blob/master/data/countries-aggregated.csv',
                   fontsize = 10)
```

Problems that may occur:

Learn More:

1. Learning pandas-https://www.learnpython.org/en/Pandas_Basics
2. Learning Matplotlib-https://realpython.com/python-matplotlib-guide/
3. Codeacademy- https://www.codecademy.com/learn/paths/visualize-data-with-python

Pic. Sites:

https://media.tenor.com/images/614ba4f8feab2c798cd83d931c4b0f4b1/tenor.gif

https://images.adsttc.com/media/images/57d9/afd1/e58e/ce72/2b00/02eb/original/output_OnCjCW.gif?1473884108

https://media1.giphy.com/media/GA2Opryocs0tg/giphy.gif

https://impeccabletablemanners.files.wordpress.com/2016/05/monkey-puppet-omg-shock-gif.gif

https://s2.thingpic.com/images/Gi/PUEchrYS16bZFgwU4Lsh6Hib.jpeg

https://img2.pngio.com/graph-cartoon-png-2-png-image-cartoon-graph-png-1920_1080.png

https://vignette.wikia.nocookie.net/chickenlittle/images/0/04/Chicken_Little.jpg/revision/latest/scale-to-width-down/340?cb=20170302012156

https://lumiere-a.akamaihd.net/v1/images/open-uri20150422-12561-1kv7u8t_994307c5.jpeg