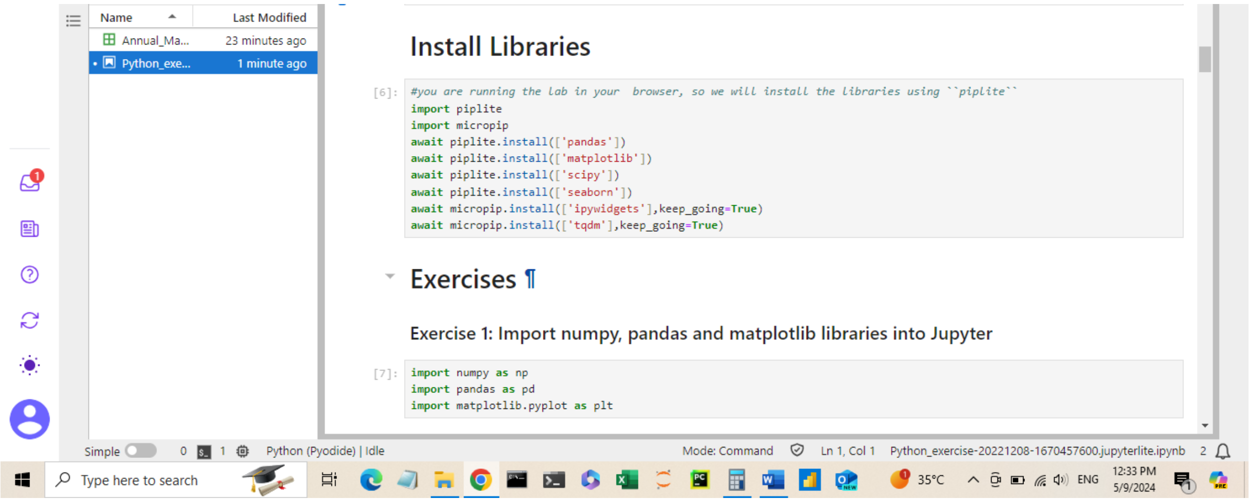
Python Exercise for Introductory Data Analysis

Exercise 1: Import numpy, pandas and matplotlib libraries into Jupyter



Install Libraries

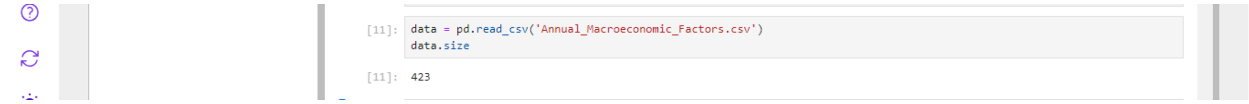
```
[6]: #you are running the lab in your  browser, so we will install the libraries using ``piplite``
import piplite
import micropip
await piplite.install(['pandas'])
await piplite.install(['matplotlib'])
await piplite.install(['scipy'])
await piplite.install(['seaborn'])
await micropip.install(['ipywidgets'],keep_going=True)
await micropip.install(['tqdm'],keep_going=True)
```

Exercises ¶

Exercise 1: Import numpy, pandas and matplotlib libraries into Jupyter

```
[7]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

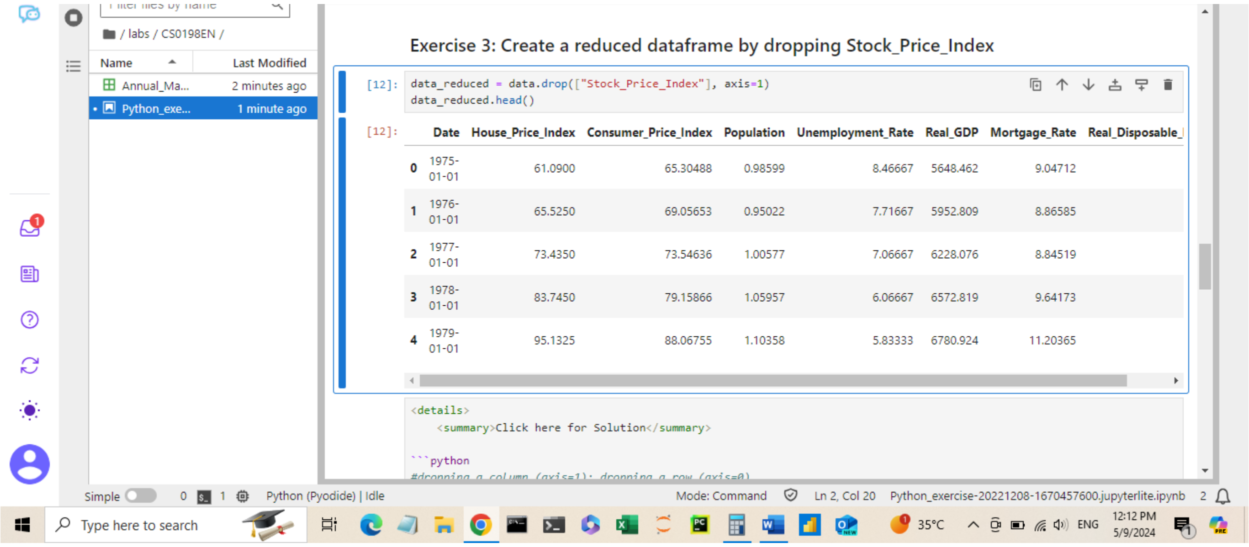
Exercise 2: Create a dataframe from a csv file and display the size of the dataframe



```
[11]: data = pd.read_csv('Annual_Macroeconomic_Factors.csv')
data.size

[11]: 423
```

Exercise 3: Create a reduced dataframe by dropping Stock_Price_Index



Exercise 3: Create a reduced dataframe by dropping Stock_Price_Index

```
[12]: data_reduced = data.drop(["Stock_Price_Index"], axis=1)
data_reduced.head()
```

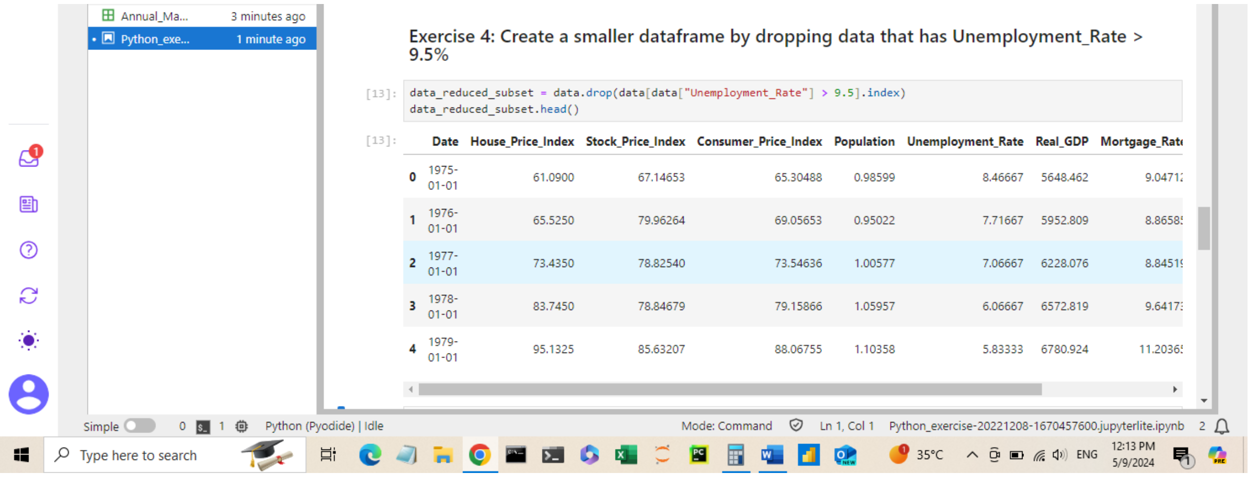
[12]:

	Date	House_Price_Index	Consumer_Price_Index	Population	Unemployment_Rate	Real_GDP	Mortgage_Rate	Real_Disposable_
0	1975-01-01	61.0900	65.30488	0.98599	8.46667	5648.462	9.04712	
1	1976-01-01	65.5250	69.05653	0.95022	7.71667	5952.809	8.86585	
2	1977-01-01	73.4350	73.54636	1.00577	7.06667	6228.076	8.84519	
3	1978-01-01	83.7450	79.15866	1.05957	6.06667	6572.819	9.64173	
4	1979-01-01	95.1325	88.06755	1.10358	5.83333	6780.924	11.20365	

~~~
<details>
    <summary>Click here for Solution</summary>

```python
#dropping a column (axis=1); dropping a row (axis=0)
~~~

Exercise 4: Create a smaller dataframe by dropping data that has Unemployment_Rate > 9.5%



Exercise 4: Create a smaller dataframe by dropping data that has Unemployment_Rate > 9.5%

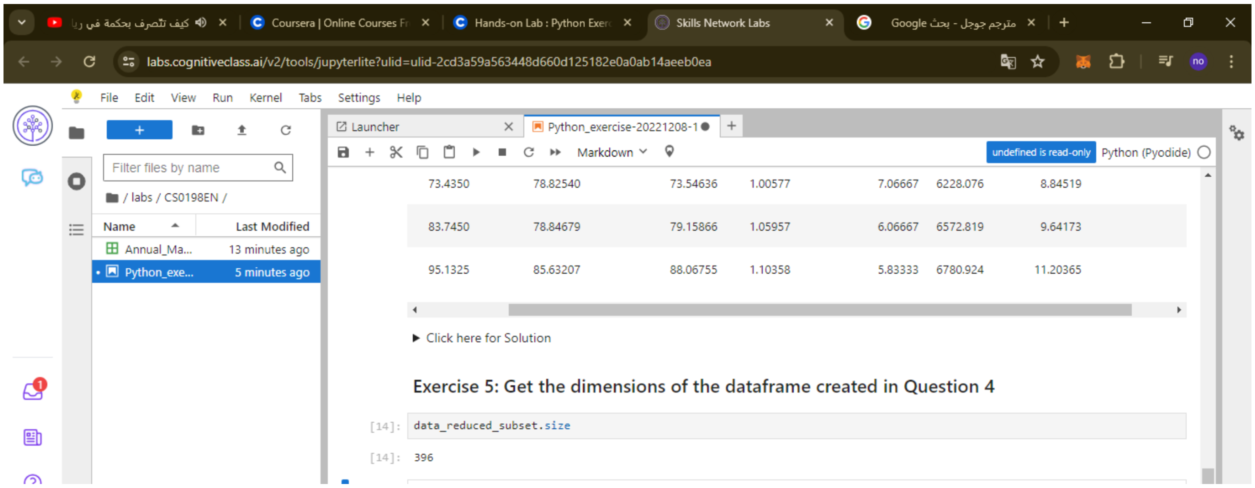
```
[13]: data_reduced_subset = data.drop(data[data["Unemployment_Rate"] > 9.5].index)
      data_reduced_subset.head()
```

[13]:

	Date	House_Price_Index	Stock_Price_Index	Consumer_Price_Index	Population	Unemployment_Rate	Real_GDP	Mortgage_Rat
0	1975-01-01	61.0900	67.14653	65.30488	0.98599	8.46667	5648.462	9.0471
1	1976-01-01	65.5250	79.96264	69.05653	0.95022	7.71667	5952.809	8.8658
2	1977-01-01	73.4350	78.82540	73.54636	1.00577	7.06667	6228.076	8.8451
3	1978-01-01	83.7450	78.84679	79.15866	1.05957	6.06667	6572.819	9.6417
4	1979-01-01	95.1325	85.63207	88.06755	1.10358	5.83333	6780.924	11.2036

Annual_Ma... 3 minutes ago
Python_exe... 1 minute ago
Simple
0
1
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Python_exercise-20221208-1670457600.jupyterlite.ipynb
2
Type here to search
35°C
ENG
12:13 PM
5/9/2024

Exercise 5: Get the dimensions of the dataframe created in Question 4



كيف تتصرف بحكمة في زر
Coursera | Online Courses Fr
Hands-on Lab : Python Exer
Skills Network Labs
Google بحث - مترجم
labs.cognitiveclass.ai/v2/tools/jupyterlite?ulid=ulid-2cd3a59a563448d660d125182e0a0ab14aeeb0ea
File
Edit
View
Run
Kernel
Tabs
Settings
Help
Filter files by name
/ labs / CS0198EN /
Name
Last Modified
Annual_Ma... 13 minutes ago
Python_exe... 5 minutes ago
Launcher
Python_exercise-20221208-1
Markdown
undefined is read-only
Python (Pyodide)

73.4350	78.82540	73.54636	1.00577	7.06667	6228.076	8.84519
83.7450	78.84679	79.15866	1.05957	6.06667	6572.819	9.64173
95.1325	85.63207	88.06755	1.10358	5.83333	6780.924	11.20365

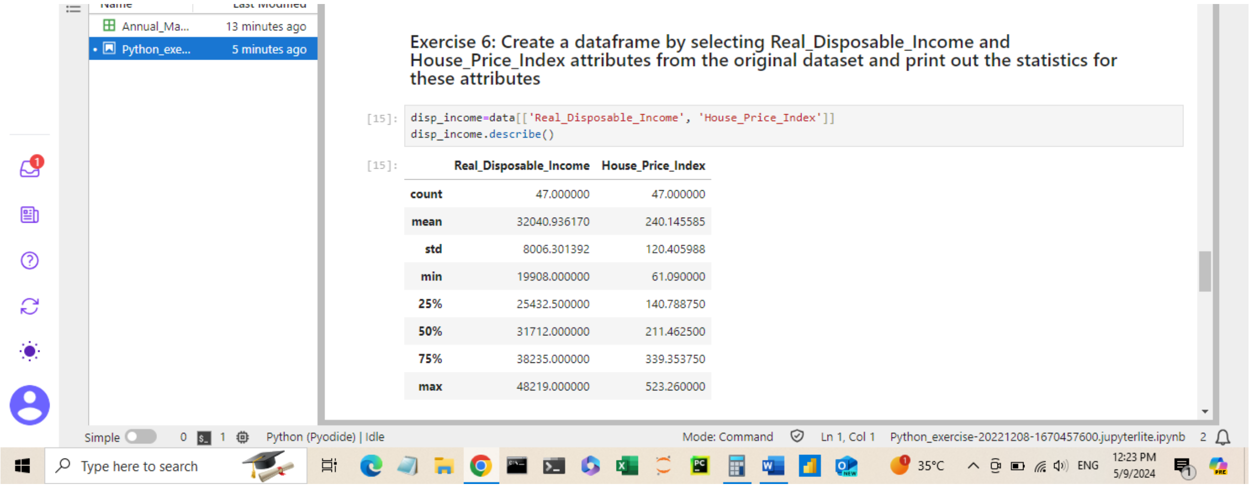
▶ Click here for Solution

Exercise 5: Get the dimensions of the dataframe created in Question 4

```
[14]: data_reduced_subset.size

[14]: 396
```

Exercise 6: Create a dataframe by selecting Real_Disposable_Income and House_Price_Index attributes from the original dataset and print out the statistics for these attributes



Annual_Ma... 13 minutes ago
Python_exe... 5 minutes ago
Simple
Python (Pyodide) | Idle
Mode: Command
Ln 1, Col 1
Python_exercise-20221208-1670457600.jupyterlite.ipynb
Type here to search
35°C
ENG
12:23 PM
5/9/2024

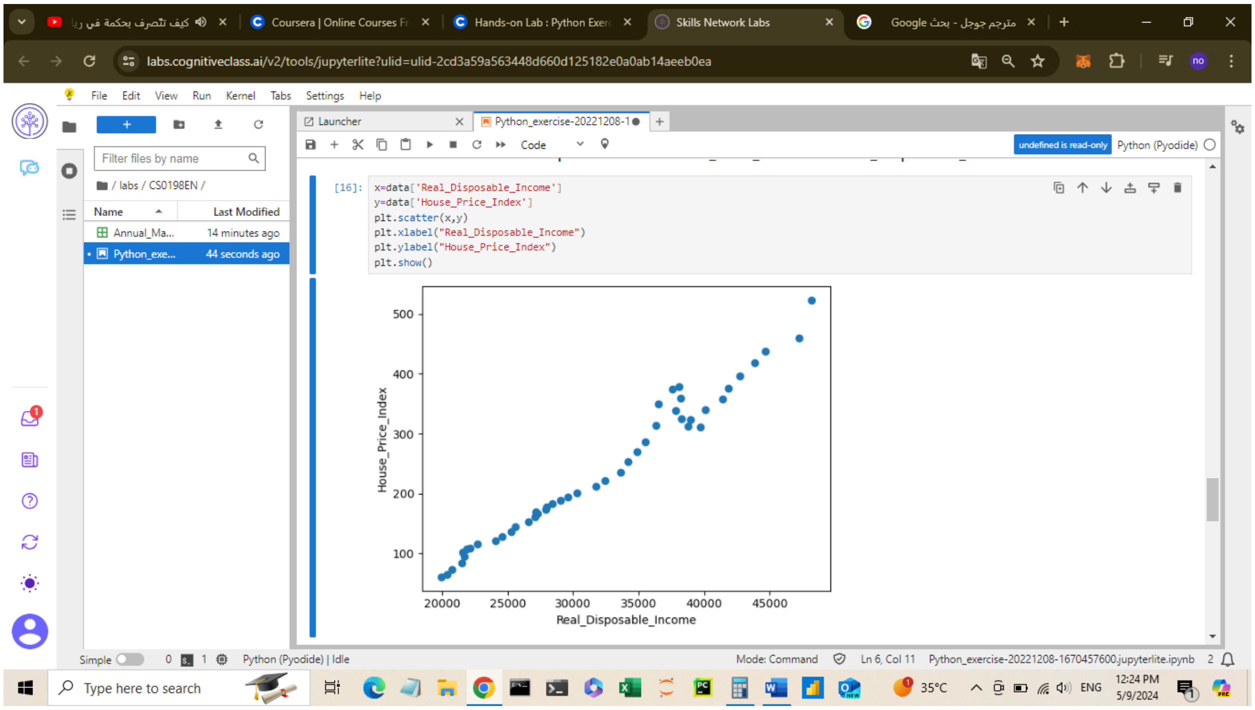
Exercise 6: Create a dataframe by selecting Real_Disposable_Income and House_Price_Index attributes from the original dataset and print out the statistics for these attributes

```
[15]: disp_income=data[['Real_Disposable_Income', 'House_Price_Index']]
      disp_income.describe()
```

[15]:

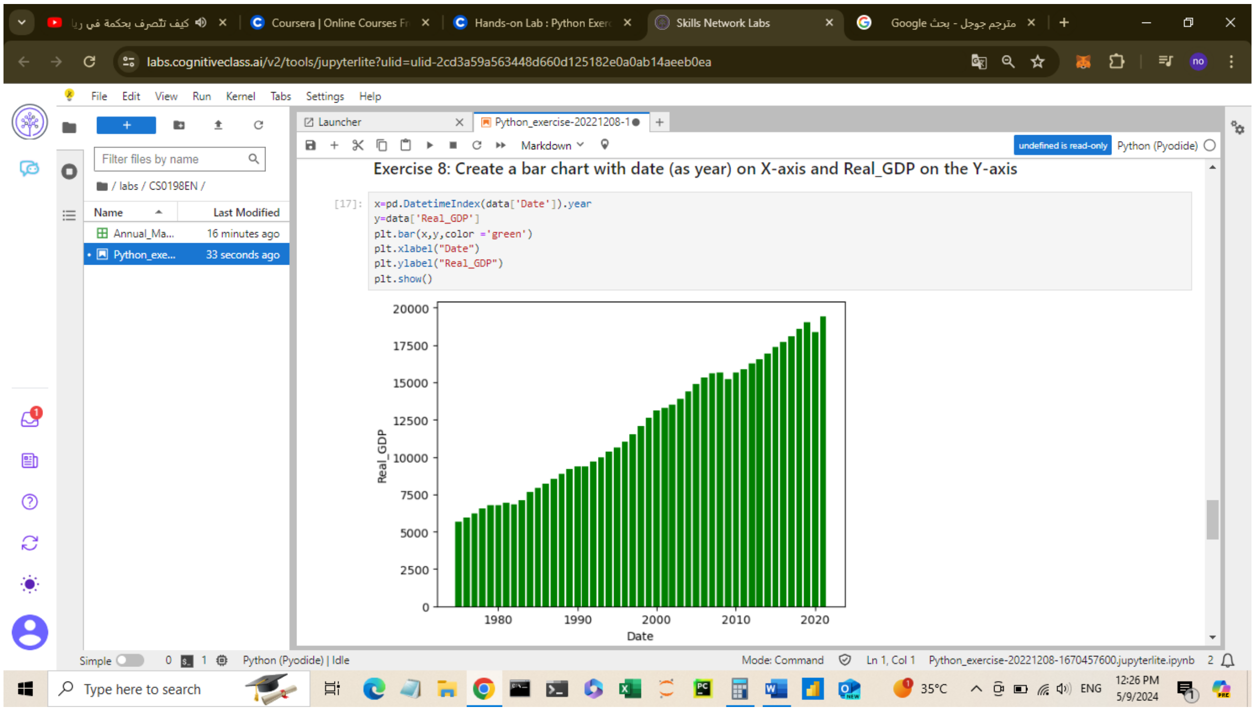
	Real_Disposable_Income	House_Price_Index
count	47.000000	47.000000
mean	32040.936170	240.145585
std	8006.301392	120.405988
min	19908.000000	61.090000
25%	25432.500000	140.788750
50%	31712.000000	211.462500
75%	38235.000000	339.353750
max	48219.000000	523.260000

Exercise 7: Plot a scatter plot between House_Price_Index and Real_Disposable_Income



```
[16]: x=data['Real_Disposable_Income']
      y=data['House_Price_Index']
      plt.scatter(x,y)
      plt.xlabel("Real_Disposable_Income")
      plt.ylabel("House_Price_Index")
      plt.show()
```

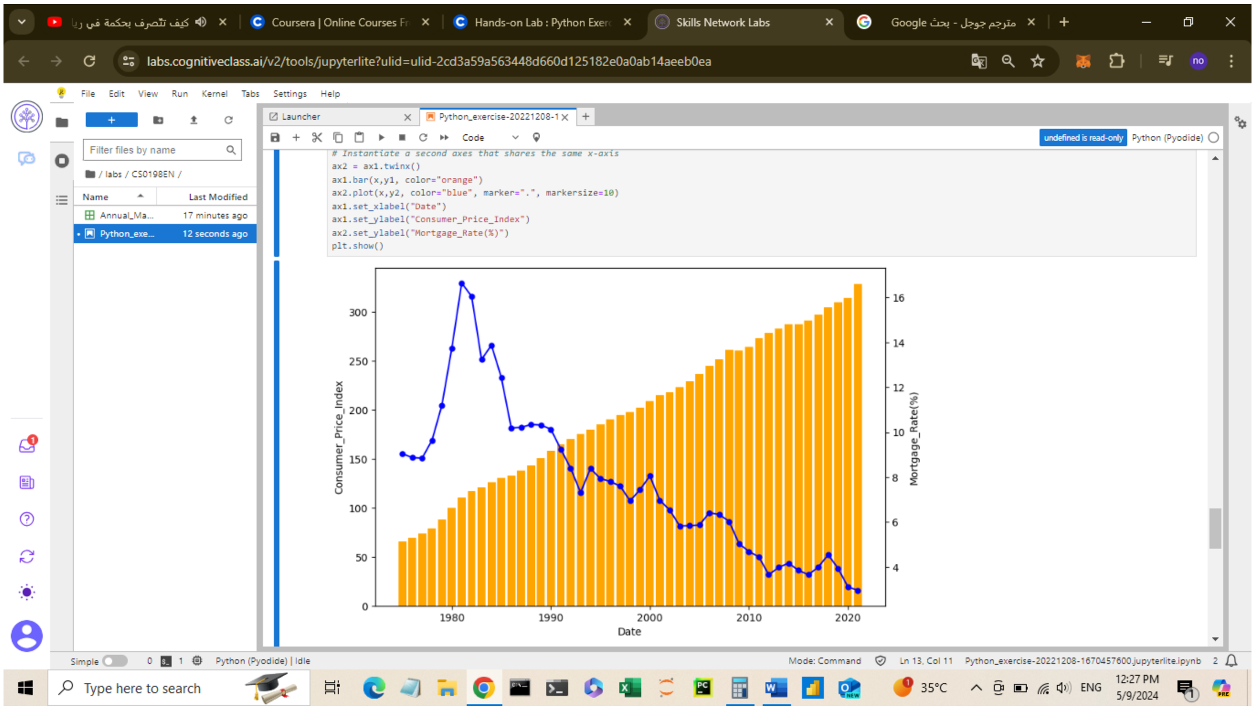
Exercise 8: Create a bar chart with date (as year) on X-axis and Real_GDP on the Y-axis



Exercise 8: Create a bar chart with date (as year) on X-axis and Real_GDP on the Y-axis

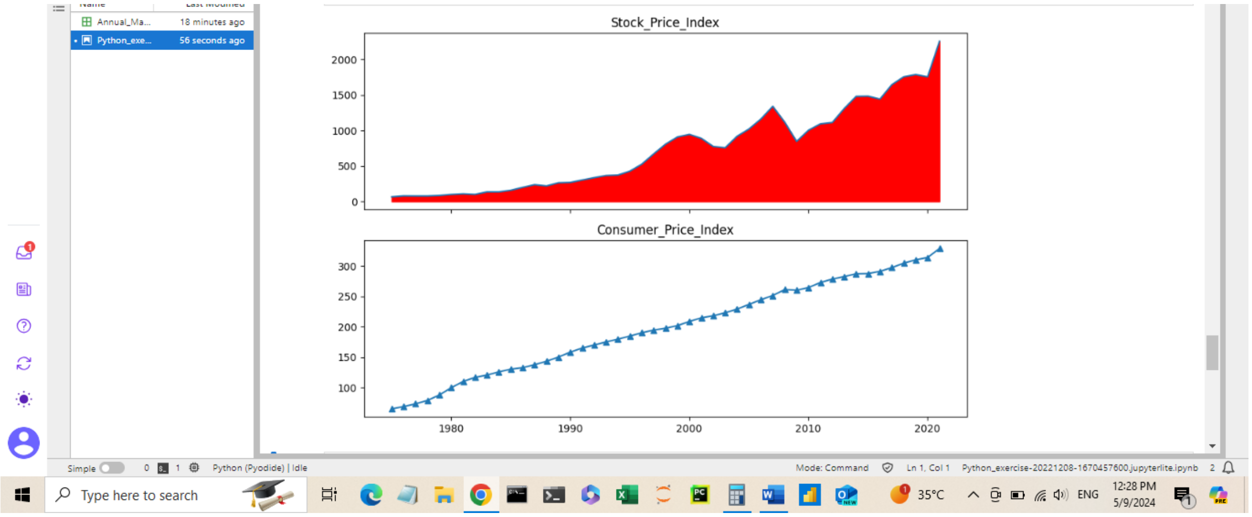
```
[17]: x=pd.DatetimeIndex(data['Date']).year
      y=data['Real_GDP']
      plt.bar(x,y,color ='green')
      plt.xlabel("Date")
      plt.ylabel("Real_GDP")
      plt.show()
```

Exercise 9: Create a combo chart with date on X-axis and Consumer_Price_Index on Y1-axis (bar) and Mortgage_Rate (line) on Y2-axis



```
# Instantiate a second axes that shares the same x-axis
ax2 = ax1.twinx()
ax1.bar(x,y1, color="orange")
ax2.plot(x,y2, color="blue", marker=".", markersize=10)
ax1.set_xlabel("Date")
ax1.set_ylabel("Consumer_Price_Index")
ax2.set_ylabel("Mortgage_Rate(%)")
plt.show()
```

Exercise 10: Create sub-plots with Date on X-axis and Stock_Price_Index on subplot 1 and Consumer_Price_Index on subplot 2



Name	Last Modified
Annual_Ma...	18 minutes ago
Python_exe...	56 seconds ago

Python_exercise-20221208-1670457600.jupyterlite.ipynb