



Extracting and Visualizing Stock Data

Description

Extracting essential data from a dataset and displaying it is a necessary part of data science; therefore individuals can make correct decisions based on the data. In this assignment, you will extract some stock data, you will then display this data in a graph.

Table of Contents

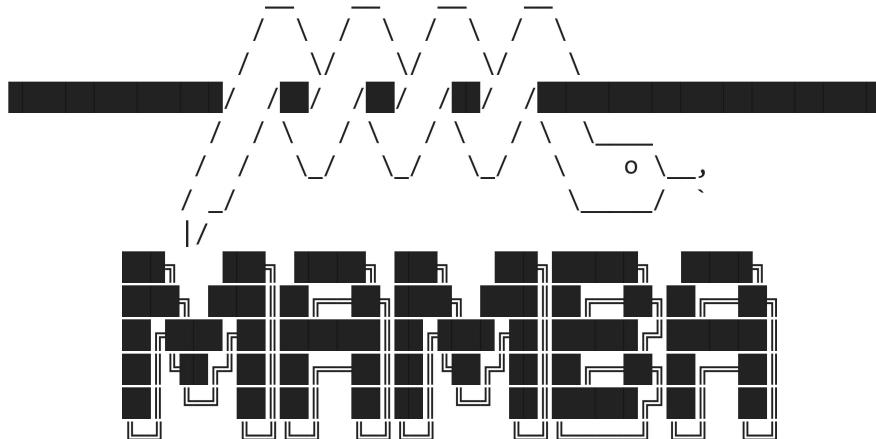
- Define a Function that Makes a Graph
- Question 1: Use yfinance to Extract Stock Data
- Question 2: Use Webscraping to Extract Tesla Revenue Data
- Question 3: Use yfinance to Extract Stock Data
- Question 4: Use Webscraping to Extract GME Revenue Data
- Question 5: Plot Tesla Stock Graph
- Question 6: Plot GameStop Stock Graph

Estimated Time Needed: **30 min**

***Note*:-** If you are working in IBM Cloud Watson Studio, please replace the command for installing nbformat from `!pip install nbformat==4.2.0` to simply `!pip install nbformat`

```
In [2]: !pip install yfinance==0.1.67  
!mamba install bs4==4.10.0 -y  
!pip install nbformat==4.2.0
```

```
Collecting yfinance==0.1.67
  Downloading yfinance-0.1.67-py2.py3-none-any.whl (25 kB)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.3.5)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.21.6)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (2.29.0)
Collecting multitasking>=0.0.7 (from yfinance==0.1.67)
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (4.9.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (2023.5.7)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance==0.1.67) (1.16.0)
Installing collected packages: multitasking, yfinance
Successfully installed multitasking-0.0.11 yfinance-0.1.67
```



mamba (1.4.2) supported by @QuantStack

GitHub: <https://github.com/mamba-org/mamba>
Twitter: <https://twitter.com/QuantStack>

Looking for: ['bs4==4.10.0']

```

[+] 0.0s
[+] 0.1s
pkgs/main/linux-64 ━━━━━━ 0.0 B / ???.?MB @ ???.?MB/s 0.1s
pkgs/main/noarch ━━━━ 0.0 B / ???.?MB @ ???.?MB/s 0.1s
pkgs/r/linux-64 ━━━━ 0.0 B / ???.?MB @ ???.?MB/s 0.1s
pkgs/r/noarch ━━━━ 0.0 B / ???.?MB @ ???.?MB/s 0.1s[+]
0.2s
pkgs/main/linux-64 ━━━━ 28.7kB / ???.?MB @ 186.3kB/s 0.2s
pkgs/main/noarch ━━━━ 0.0 B / ???.?MB @ ???.?MB/s 0.2s
pkgs/r/linux-64 ━━ 73.7kB / ???.?MB @ 480.2kB/s 0.2s
pkgs/r/noarch ━━ 41.0kB / ???.?MB @ 265.4kB/s 0.2s[+]
0.3s
pkgs/main/linux-64 ━━ 553.0kB / ???.?MB @ 2.2MB/s 0.3s
pkgs/main/noarch ━━ 376.8kB / ???.?MB @ 1.5MB/s 0.3s
pkgs/r/linux-64 ━━ 581.6kB / ???.?MB @ 2.3MB/s 0.3s
pkgs/r/noarch ━━ 577.5kB / ???.?MB @ 2.3MB/s 0.3spkg
s/main/noarch
859.8kB @ 2.5MB/s 0.4s
[+] 0.4s
pkgs/main/linux-64 ━━ 1.3MB / ???.?MB @ 3.4MB/s 0.4s
pkgs/r/linux-64 ━━ 1.3MB / ???.?MB @ 3.3MB/s 0.4s
pkgs/r/noarch ━━ 1.3MB / ???.?MB @ 3.3MB/s 0.4s[+]
0.5s
pkgs/main/linux-64 ━━ 1.9MB / ???.?MB @ 4.0MB/s 0.5s
pkgs/r/linux-64 ━━ 1.8MB / ???.?MB @ 3.7MB/s 0.5s
pkgs/r/noarch ━━ 1.8MB / ???.?MB @ 3.7MB/s 0.5spkg
s/r/linux-64
1.9MB @ 3.8MB/s 0.5s
[+] 0.6s
pkgs/main/linux-64 ━━ 2.4MB @ 4.1MB/s 0.6s
pkgs/r/noarch ━━ 2.3MB @ 3.9MB/s Finalizing 0.6spkg
s/r/noarch
@ 3.9MB/s 0.6s
[+] 0.7s
pkgs/main/linux-64 ━━ 2.9MB / ???.?MB @ 4.3MB/s 0.7s[+]
0.8s
pkgs/main/linux-64 ━━ 3.5MB / ???.?MB @ 4.4MB/s 0.8s[+]
0.9s
pkgs/main/linux-64 ━━ 4.1MB / ???.?MB @ 4.6MB/s 0.9s[+]
1.0s
pkgs/main/linux-64 ━━ 4.6MB / ???.?MB @ 4.6MB/s 1.0s[+]
1.1s
pkgs/main/linux-64 ━━ 4.9MB / ???.?MB @ 4.6MB/s 1.1s[+]
1.2s
pkgs/main/linux-64 ━━ 5.4MB / ???.?MB @ 4.7MB/s 1.2s[+]
1.3s
pkgs/main/linux-64 ━━ 6.1MB / ???.?MB @ 4.8MB/s 1.3s[+]
1.4s
pkgs/main/linux-64 ━━ 6.1MB / ???.?MB @ 4.8MB/s 1.4s[+]
1.5s
pkgs/main/linux-64 ━━ 6.6MB @ 4.7MB/s Finalizing 1.5spkg
s/main/linux-64
@ 4.7MB/s 1.5s
[+] 1.6s

```

Pinned packages:
- python 3.7.*

Transaction

Prefix: /home/jupyterlab/conda/envs/python

Updating specs:

- bs4==4.10.0
- ca-certificates
- certifi
- openssl

Package	Version	Build	Channel	Size
<hr/>				
Install:				
<hr/>				
+ bs4	4.10.0	hd3eb1b0_0	pkgs/main/noarch	10kB
<hr/>				
Upgrade:				
<hr/>				
- ca-certificates	2023.5.7	hbcca054_0	conda-forge	
+ ca-certificates	2023.12.12	h06a4308_0	pkgs/main/linux-64	129kB
- openssl	1.1.1t	h0b41bf4_0	conda-forge	
+ openssl	1.1.1w	h7f8727e_0	pkgs/main/linux-64	4MB
<hr/>				
Downgrade:				
<hr/>				
- beautifulsoup4	4.11.1	pyha770c72_0	conda-forge	
+ beautifulsoup4	4.10.0	pyh06a4308_0	pkgs/main/noarch	87kB
<hr/>				
Summary:				
<hr/>				
Install: 1 packages				
Upgrade: 2 packages				
Downgrade: 1 packages				
<hr/>				
Total download: 4MB				
<hr/>				

[+]	0.0s			
Downloading	<div style="width: 100%; height: 10px; background-color: black;"></div>	0.0 B		0.0s
Extracting	<div style="width: 100%; height: 10px; background-color: black;"></div>	0		0.0s[+]
0.1s				
Downloading (4)	<div style="width: 100%; height: 10px; background-color: yellow;"></div>	0.0 B	beautifulsoup4	0.0s
Extracting	<div style="width: 100%; height: 10px; background-color: black;"></div>	0		0.0sbs4
10.2kB @ 76.1kB/s	0.1s			
ca-certificates		128.7kB @ 908.0kB/s	0.1s	
beautifulsoup4		86.6kB @ 537.4kB/s	0.2s	
openssl		3.9MB @ 19.8MB/s	0.2s	
[+]	0.2s			
Downloading	<div style="width: 100%; height: 10px; background-color: black;"></div>	4.1MB		0.1s
Extracting (3)	<div style="width: 100%; height: 10px; background-color: yellow;"></div>	0	beautifulsoup4	0.0s[+]

0.3s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 beautifulsoup4	0.1s[+]
0.4s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 beautifulsoup4	0.2s[+]
0.5s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 beautifulsoup4	0.3s[+]
0.6s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 bs4	0.4s[+]
0.7s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 bs4	0.5s[+]
0.8s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 bs4	0.6s[+]
0.9s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 bs4	0.7s[+]
1.0s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 ca-certificates	0.8s[+]
1.1s			
Downloading	(4)	4.1MB	0.1s
Extracting	(4)	0 ca-certificates	0.9s[+]
1.2s			
Downloading	(4)	4.1MB	0.1s
Extracting	(3)	1 ca-certificates	1.0s[+]
1.3s			
Downloading	(3)	4.1MB	0.1s
Extracting	(3)	1 ca-certificates	1.1s[+]
1.4s			
Downloading	(2)	4.1MB	0.1s
Extracting	(2)	2 openssl	1.2s[+]
1.5s			
Downloading	(2)	4.1MB	0.1s
Extracting	(2)	2 openssl	1.3s[+]
1.6s			
Downloading	(1)	4.1MB	0.1s
Extracting	(1)	3 openssl	1.4s
Downloading and Extracting Packages			

Preparing transaction: done

Verifying transaction: done

Executing transaction: done

Collecting nbformat==4.2.0

 Downloading nbformat-4.2.0-py2.py3-none-any.whl (153 kB)

 ————— 153.3/153.3 kB 19.8 MB/s eta 0:00:00

Requirement already satisfied: ipython-genutils in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (0.2.0)

Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (4.17.3)

Requirement already satisfied: jupyter-core in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (4.12.0)

```

Requirement already satisfied: traitlets>=4.1 in /home/jupyterlab/conda/envs/python/
lib/python3.7/site-packages (from nbformat==4.2.0) (5.9.0)
Requirement already satisfied: attrs>=17.4.0 in /home/jupyterlab/conda/envs/python/l
ib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (23.1.0)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (4.1
1.4)
Requirement already satisfied: importlib-resources>=1.4.0 in /home/jupyterlab/conda/
envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.
2.0) (5.12.0)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10 in /home/jupyterlab/cond
a/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==
4.2.0) (1.3.10)
Requirement already satisfied: pyrsistent!=0.17.0,!0.17.1,!0.17.2,>=0.14.0 in /hom
e/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>
=2.4->nbformat==4.2.0) (0.19.3)
Requirement already satisfied: typing-extensions in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (4.5.
0)
Requirement already satisfied: zipp>=3.1.0 in /home/jupyterlab/conda/envs/python/li
b/python3.7/site-packages (from importlib-resources>=1.4.0->jsonschema!=2.5.0,>=2.4-
>nbformat==4.2.0) (3.15.0)
Installing collected packages: nbformat
  Attempting uninstall: nbformat
    Found existing installation: nbformat 5.8.0
    Uninstalling nbformat-5.8.0:
      Successfully uninstalled nbformat-5.8.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following dependency conflicts.
jupyter-server 1.24.0 requires nbformat>=5.2.0, but you have nbformat 4.2.0 which is incompatible.
nbclient 0.7.4 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
nbconvert 7.4.0 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
Successfully installed nbformat-4.2.0

```

```
In [19]: import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
```

In Python, you can ignore warnings using the `warnings` module. You can use the `filterwarnings` function to filter or ignore specific warning messages or categories.

Define Graphing Function

In this section, we define the function `make_graph`. You don't have to know how the function works, you should only care about the inputs. It takes a dataframe with stock data

(dataframe must contain Date and Close columns), a dataframe with revenue data (dataframe must contain Date and Revenue columns), and the name of the stock.

```
In [ ]: def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Stock Data", "Revenue Data"))
    stock_data_specific = stock_data[stock_data.Date <= '2021-06-14']
    revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date, infer_datetime_format=True), y=stock_data_specific.Close, name="Stock Price"))
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_specific.Revenue, name="Revenue"))
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
                      height=900,
                      title=stock,
                      xaxis_rangeslider_visible=True)
    fig.show()
```

Question 1: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is Tesla and its ticker symbol is `TSLA`.

```
In [4]: tesla = yf.Ticker("TSLA")
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `tesla_data`. Set the `period` parameter to `max` so we get information for the maximum amount of time.

```
In [15]: tesla_data = tesla.history(period="max")
tesla_data.head()
```

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0

Reset the index using the `reset_index(inplace=True)` function on the `tesla_data` DataFrame and display the first five rows of the `tesla_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

```
In [16]: # reset the index of the dataframe
tesla_data.reset_index(inplace=True)

# display the first 5 rows of the dataframe
tesla_data.head()
```

Out[16]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
1	2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2	2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
3	2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
4	2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0

```
In [6]: pip install html5lib
```

```
Collecting html5lib
  Downloading html5lib-1.1-py2.py3-none-any.whl (112 kB)
    112.2/112.2 kB 16.7 MB/s eta 0:00:00
Requirement already satisfied: six>=1.9 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (1.16.0)
Requirement already satisfied: webencodings in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (0.5.1)
Installing collected packages: html5lib
Successfully installed html5lib-1.1
Note: you may need to restart the kernel to use updated packages.
```

```
In [62]: from bs4 import BeautifulSoup
```

Question 2: Use Webscraping to Extract Tesla Revenue Data

Use the `requests` library to download the webpage <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm>. Save the text of the response as a variable named `html_data`.

```
In [20]: pip install --upgrade pip
```

```
Requirement already satisfied: pip in /home/jupyterlab/conda/envs/python/lib/python  
3.7/site-packages (23.1.2)  
Collecting pip  
  Downloading pip-24.0-py3-none-any.whl (2.1 MB)  
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 51.5 MB/s eta 0:00:00 00:01  
Installing collected packages: pip  
Attempting uninstall: pip  
  Found existing installation: pip 23.1.2  
  Uninstalling pip-23.1.2:  
    Successfully uninstalled pip-23.1.2  
Successfully installed pip-24.0  
Note: you may need to restart the kernel to use updated packages.
```

In [21]: `pip install lxml`

```
Requirement already satisfied: lxml in /home/jupyterlab/conda/envs/python/lib/python  
3.7/site-packages (4.9.2)  
Note: you may need to restart the kernel to use updated packages.
```

In [22]: `pip install html5lib`

```
Requirement already satisfied: html5lib in /home/jupyterlab/conda/envs/python/lib/py  
thon3.7/site-packages (1.1)  
Requirement already satisfied: six>=1.9 in /home/jupyterlab/conda/envs/python/lib/py  
thon3.7/site-packages (from html5lib) (1.16.0)  
Requirement already satisfied: webencodings in /home/jupyterlab/conda/envs/python/li  
b/python3.7/site-packages (from html5lib) (0.5.1)  
Note: you may need to restart the kernel to use updated packages.
```

In [17]: `url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDelv
html_data = requests.get(url).text`

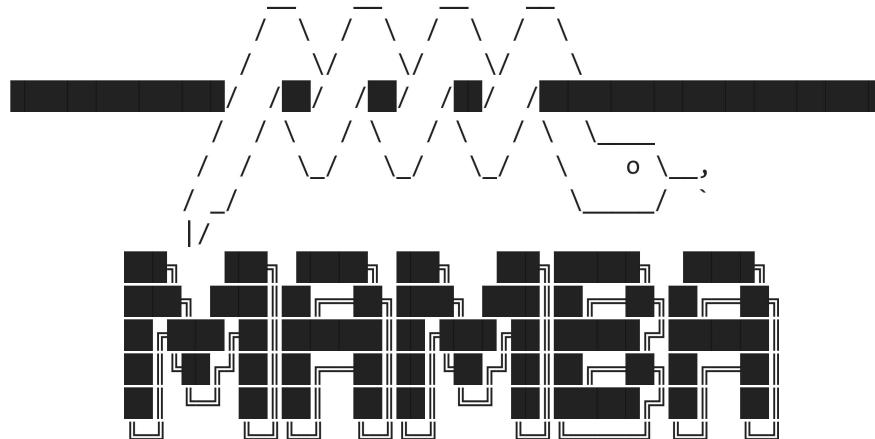
Parse the html data using `beautiful_soup`.

In [13]: `!pip install yfinance==0.1.67
!mamba install bs4==4.10.0 -y
!pip install nbformat==4.2.0
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots
!pip install html5lib
tesla = yf.Ticker("TSLA")
tesla_data = tesla.history(period="max")
reset the index of the dataframe
tesla_data.reset_index(inplace=True)
display the first 5 rows of the dataframe
tesla_data.head()

url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDelv
html_data = requests.get(url).text
soup = BeautifulSoup(html_data, 'html.parser')
tesla_revenue = pd.DataFrame(columns=['Date', 'Revenue'])`

```
tesla_table = soup.find_all('table')[1]
for row in tesla_table.find("tbody").find_all("tr"):
    col = row.find_all('td')
    date = col[0].string
    revenue = col[1].string
tesla_revenue = tesla_revenue.append({'Date':date, 'Revenue':revenue}, ignore_index=True)
print(tesla_revenue)
```

```
Requirement already satisfied: yfinance==0.1.67 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.3.5)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.21.6)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (2.29.0)
Requirement already satisfied: multitasking>=0.0.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (0.0.11)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (4.9.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (2023.5.7)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance==0.1.67) (1.16.0)
```



mamba (1.4.2) supported by @QuantStack

GitHub: <https://github.com/mamba-org/mamba>
Twitter: <https://twitter.com/QuantStack>

Looking for: ['bs4==4.10.0']

[+] 0.0s
[+] 0.1s

pkgs/main/linux-64		0.0 B / ???.?MB @ ???.?MB/s 0.1s
pkgs/main/noarch		0.0 B / ???.?MB @ ???.?MB/s 0.1s
pkgs/r/linux-64		0.0 B / ???.?MB @ ???.?MB/s 0.1s
pkgs/r/noarch		0.0 B / ???.?MB @ ???.?MB/s 0.1s
s/main/noarch		No change
pkgs/r/noarch		No change
pkgs/main/linux-64		No change
pkgs/r/linux-64		No change

Pinned packages:

- python 3.7.*

Transaction

Prefix: /home/jupyterlab/conda/envs/python

All requested packages already installed

Requirement already satisfied: nbformat==4.2.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (4.2.0)
Requirement already satisfied: ipython-genutils in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (4.17.3)
Requirement already satisfied: jupyter-core in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (4.12.0)
Requirement already satisfied: traitlets>=4.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (5.9.0)
Requirement already satisfied: attrs>=17.4.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (23.1.0)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (4.1.4)
Requirement already satisfied: importlib-resources>=1.4.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (5.12.0)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (1.3.10)
Requirement already satisfied: pyrsistent!=0.17.0,!0.17.1,!0.17.2,>=0.14.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (0.19.3)
Requirement already satisfied: typing-extensions in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (4.5.0)
Requirement already satisfied: zipp>=3.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from importlib-resources>=1.4.0->jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (3.15.0)
Requirement already satisfied: html5lib in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (1.1)
Requirement already satisfied: six>=1.9 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (1.16.0)
Requirement already satisfied: webencodings in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (0.5.1)

```
Date Revenue
0 2009-06-30      $27
```

```
In [14]: # URL of the webpage to scrape
url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDevelo
```

```
In [15]: # Fetch HTML content of the webpage
html_data = requests.get(url).text
```

Using `BeautifulSoup` or the `read_html` function extract the table with `Tesla Revenue` and store it into a dataframe named `tesla_revenue`. The dataframe should have columns `Date` and `Revenue`.

► Click here if you need help locating the table

```
In [16]: import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots

# Install html5lib package if not already installed
!pip install html5lib

# Fetch Tesla stock data using yfinance
tesla = yf.Ticker("TSLA")
tesla_data = tesla.history(period="max")

# Reset the index of the dataframe
tesla_data.reset_index(inplace=True)

# Display the first 5 rows of the dataframe
print(tesla_data.head())

# URL of the webpage to scrape
url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDevelo

# Fetch HTML content of the webpage
html_data = requests.get(url).text

# Parse HTML content using BeautifulSoup
soup = BeautifulSoup(html_data, 'html.parser')

# Find the desired table containing revenue data
tesla_table = soup.find_all('table')[1] # Adjust the index if necessary

# Initialize an empty DataFrame to store revenue data
tesla_revenue = pd.DataFrame(columns=['Date', 'Revenue'])

# Extract data from each row of the table and append it to the DataFrame
for row in tesla_table.find("tbody").find_all("tr"):
    col = row.find_all('td')
    date = col[0].string
```

```
revenue = col[1].string
tesla_revenue = tesla_revenue.append({'Date': date, 'Revenue': revenue}, ignore_index=True)

# Display the extracted revenue data
print(tesla_revenue)
```

Requirement already satisfied: html5lib in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (1.1)

Requirement already satisfied: six>=1.9 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (1.16.0)

Requirement already satisfied: webencodings in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (0.5.1)

	Date	Open	High	Low	Close	Volume	Dividends	\
0	2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	
1	2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	
2	2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	
3	2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	
4	2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	

Stock Splits

0	0.0
1	0.0
2	0.0
3	0.0
4	0.0

Date Revenue

0	2022-09-30	\$21,454
1	2022-06-30	\$16,934
2	2022-03-31	\$18,756
3	2021-12-31	\$17,719
4	2021-09-30	\$13,757
5	2021-06-30	\$11,958
6	2021-03-31	\$10,389
7	2020-12-31	\$10,744
8	2020-09-30	\$8,771
9	2020-06-30	\$6,036
10	2020-03-31	\$5,985
11	2019-12-31	\$7,384
12	2019-09-30	\$6,303
13	2019-06-30	\$6,350
14	2019-03-31	\$4,541
15	2018-12-31	\$7,226
16	2018-09-30	\$6,824
17	2018-06-30	\$4,002
18	2018-03-31	\$3,409
19	2017-12-31	\$3,288
20	2017-09-30	\$2,985
21	2017-06-30	\$2,790
22	2017-03-31	\$2,696
23	2016-12-31	\$2,285
24	2016-09-30	\$2,298
25	2016-06-30	\$1,270
26	2016-03-31	\$1,147
27	2015-12-31	\$1,214
28	2015-09-30	\$937
29	2015-06-30	\$955
30	2015-03-31	\$940
31	2014-12-31	\$957
32	2014-09-30	\$852
33	2014-06-30	\$769
34	2014-03-31	\$621
35	2013-12-31	\$615

36	2013-09-30	\$431
37	2013-06-30	\$405
38	2013-03-31	\$562
39	2012-12-31	\$306
40	2012-09-30	\$50
41	2012-06-30	\$27
42	2012-03-31	\$30
43	2011-12-31	\$39
44	2011-09-30	\$58
45	2011-06-30	\$58
46	2011-03-31	\$49
47	2010-12-31	\$36
48	2010-09-30	\$31
49	2010-06-30	\$28
50	2010-03-31	\$21
51	2009-12-31	None
52	2009-09-30	\$46
53	2009-06-30	\$27

Execute the following line to remove the comma and dollar sign from the `Revenue` column.

```
In [17]: tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',', '$', '')
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: The default value of regex will change from True to False in a future version.
    """Entry point for launching an IPython kernel.
```

Execute the following lines to remove all null or empty strings in the Revenue column.

```
In [18]: tesla_revenue.dropna(inplace=True)

tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

Display the last 5 rows of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
In [19]: tesla_revenue.tail()
```

	Date	Revenue
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
52	2009-09-30	46
53	2009-06-30	27

Question 3: Use yfinance to Extract Stock Data

Using the `Ticker` function enter the ticker symbol of the stock we want to extract data on to create a ticker object. The stock is GameStop and its ticker symbol is `GME`.

```
In [11]: gamestop = yf.Ticker('GME')
```

Using the ticker object and the function `history` extract stock information and save it in a dataframe named `gme_data`. Set the `period` parameter to `'max'` so we get information for the maximum amount of time.

```
In [12]: gme_data = gamestop.history(period = 'max')
```

Reset the index using the `reset_index(inplace=True)` function on the `gme_data` DataFrame and display the first five rows of the `gme_data` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 3 to the results below.

```
In [13]: gme_data.reset_index(inplace=True)
gme_data.head()
```

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620128	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14	1.712707	1.716074	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
3	2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0

Question 4: Use Webscraping to Extract GME Revenue Data

Use the `requests` library to download the webpage <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDriverSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html>. Save the text of the response as a variable named `html_data`.

```
In [19]: !pip install yfinance==0.1.67
!pip install bs4==4.10.0
!pip install html5lib
```

```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup

url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperData/PyData/quarterly_revenue.html"
html_data = requests.get(url).text

soup = BeautifulSoup(html_data, "html.parser")

tables = soup.find_all('table')
for index,table in enumerate(tables):
    if("GameStop Quarterly Revenue" in str(table)):
        table_index = index

gme_revenue = pd.DataFrame(columns=['Date','Revenue'])
for row in tables[table_index].tbody.find_all('tr'):
    col = row.find_all('td')
    if(col != []):
        date = col[0].text
        revenue = col[1].text
        gme_revenue = gme_revenue.append({'Date':date, 'Revenue':revenue}, ignore_index=True)

gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',', '$')
gme_revenue.dropna(inplace=True)
gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
gme_revenue
```

```
Requirement already satisfied: yfinance==0.1.67 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.3.5)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (1.21.6)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (2.29.0)
Requirement already satisfied: multitasking>=0.0.7 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (0.0.11)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (4.9.2)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.1.0)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (1.26.15)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (2023.5.7)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance==0.1.67) (1.16.0)
ERROR: Could not find a version that satisfies the requirement bs4==4.10.0 (from versions: 0.0.0, 0.0.1, 0.0.2)
ERROR: No matching distribution found for bs4==4.10.0
Requirement already satisfied: html5lib in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (1.1)
Requirement already satisfied: six>=1.9 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (1.16.0)
Requirement already satisfied: webencodings in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib) (0.5.1)

/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:28: FutureWarning: The default value of regex will change from True to False in a future version.
```

Out[19]:

	Date	Revenue
0	2020-04-30	1021
1	2020-01-31	2194
2	2019-10-31	1439
3	2019-07-31	1286
4	2019-04-30	1548
...
57	2006-01-31	1667
58	2005-10-31	534
59	2005-07-31	416
60	2005-04-30	475
61	2005-01-31	709

62 rows × 2 columns

Parse the html data using `beautiful_soup`.In [5]: `soup = BeautifulSoup(html_data, "html.parser")`

```
NameError                                 Traceback (most recent call last)
/tmp/ipykernel_69/194777951.py in <module>
----> 1 soup = BeautifulSoup(html_data, "html.parser")

NameError: name 'BeautifulSoup' is not defined
```

Using `BeautifulSoup` or the `read_html` function extract the table with `GameStop Revenue` and store it into a dataframe named `gme_revenue`. The dataframe should have columns `Date` and `Revenue`. Make sure the comma and dollar sign is removed from the `Revenue` column using a method similar to what you did in Question 2.

► Click here if you need help locating the table

Display the last five rows of the `gme_revenue` dataframe using the `tail` function. Take a screenshot of the results.

In [22]:

```
gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',', '$', '')
gme_revenue.dropna(inplace=True)
gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
gme_revenue.tail()
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:1: FutureWarning: The default value of regex will change from True to False in a future version.
    """Entry point for launching an IPython kernel.
```

Out[22]:

	Date	Revenue
57	2006-01-31	1667
58	2005-10-31	534
59	2005-07-31	416
60	2005-04-30	475
61	2005-01-31	709

Question 5: Plot Tesla Stock Graph

Use the `make_graph` function to graph the Tesla Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(tesla_data, tesla_revenue, 'Tesla')`. Note the graph will only show data upto June 2021.

In [26]:

```
import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots

def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Stock Data", "Historical Revenue"))
    stock_data_specific = stock_data[stock_data.Date <= '2021-06-14']
    revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date, infer_datetime=True), y=stock_data_specific.Close, name='Stock Price'), row=1, col=1)
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime=True), y=revenue_data_specific.Revenue, name='Revenue'), row=2, col=1)
    fig.update_xaxes(title_text="Date", row=1, col=1)
    fig.update_xaxes(title_text="Date", row=2, col=1)
    fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
    fig.update_layout(showlegend=False,
                      height=900,
                      title=stock,
                      xaxis_rangeslider_visible=True)
    fig.show()

tesla = yf.Ticker("TSLA")
tesla_data = tesla.history(period="max")
# reset the index of the dataframe
tesla_data.reset_index(inplace=True)
# display the first 5 rows of the dataframe
tesla_data.head()

url = ' https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloper/
```

```
html_data = requests.get(url).text
soup = BeautifulSoup(html_data, 'html5lib')
tesla_revenue = pd.DataFrame(columns=['Date', 'Revenue'])
tesla_table = soup.find_all('table')[1]
for row in tesla_table.find("tbody").find_all("tr"):
    col = row.find_all('td')
    date = col[0].text
    revenue = col[1].text
    revenue = col[1].text.replace("$","",).replace(",","",)
    tesla_revenue = tesla_revenue.append({'Date':date, 'Revenue':revenue}, ignore_index=True)
tesla_revenue.dropna(inplace=True)
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]

print(tesla_revenue.tail())

make_graph(tesla_data, tesla_revenue, 'Tesla')
```

	Date	Revenue
48	2010-09-30	31
49	2010-06-30	28
50	2010-03-31	21
52	2009-09-30	46
53	2009-06-30	27

Question 6: Plot GameStop Stock Graph

```
In [57]: gme_revenue=gme_revenue.rename(columns = {'GameStop Quarterly Revenue(Millions of U
gme_revenue["Revenue"] = gme_revenue["Revenue"].str.replace(",","",).str.replace("$"
gme_revenue.head()
gme_revenue.dropna(inplace=True)
gme_revenue.tail(2021)
```

/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launcher.py:2: FutureWarning:

The default value of regex will change from True to False in a future version. In addition, single character regular expressions will *not* be treated as literal strings when regex=True.

Out[57]:

	Date	Revenue
0	2020-04-30	1021
1	2020-01-31	2194
2	2019-10-31	1439
3	2019-07-31	1286
4	2019-04-30	1548
...
57	2006-01-31	1667
58	2005-10-31	534
59	2005-07-31	416
60	2005-04-30	475
61	2005-01-31	709

62 rows × 2 columns

Use the `make_graph` function to graph the GameStop Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(gme_data, gme_revenue, 'GameStop')`. Note the graph will only show data upto June 2021.

In []:

```
In [21]: import yfinance as yf
import pandas as pd
import requests
from bs4 import BeautifulSoup
import plotly.graph_objects as go
from plotly.subplots import make_subplots

def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=(
```

```

stock_data_specific = stock_data[stock_data.Date <= '2021-06-14']
revenue_data_specific = revenue_data[revenue_data.Date <= '2021-04-30']
fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date, infer_datetime_format=True), y=stock_data_specific['Price ($US)'], mode='lines'))
fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_specific['Revenue ($US Millions)'], mode='lines'))
fig.update_xaxes(title_text="Date", row=1, col=1)
fig.update_xaxes(title_text="Date", row=2, col=1)
fig.update_yaxes(title_text="Price ($US)", row=1, col=1)
fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)
fig.update_layout(showlegend=False,
height=900,
title=stock,
xaxis_rangeslider_visible=True)
fig.show()

gamestop = yf.Ticker('GME')
gme_data = gamestop.history(period = 'max')
gme_data.reset_index(inplace=True)

url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0101EN-SkillsNetwork/labs/Module%204/PyCharm/HTMLTableToPandas.ipynb"
html_data = requests.get(url).text

soup = BeautifulSoup(html_data, "html.parser")
gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])
gme_table = soup.find_all('table')[1]

for row in gme_table.find("tbody").find_all('tr'):
    col = row.find_all("td")
    date = col[0].text
    revenue = col[1].text
    revenue = col[1].text.replace("$","",).replace(",","",")
    gme_revenue = gme_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)
gme_revenue.dropna(inplace=True)
gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
print (gme_revenue.tail())

make_graph(gme_data, gme_revenue, 'GameStop')

```

	Date	Revenue
57	2006-01-31	1667
58	2005-10-31	534
59	2005-07-31	416
60	2005-04-30	475
61	2005-01-31	709

About the Authors:

[Joseph Santarcangelo](#) has a PhD in Electrical Engineering, his research focused on using machine learning, signal processing, and computer vision to determine how videos impact human cognition. Joseph has been working for IBM since he completed his PhD.

Azim Hirjani

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-02-28	1.2	Lakshmi Holla	Changed the URL of GameStop
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab

© IBM Corporation 2020. All rights reserved.