



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

```
[6]: !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: requests>=2.20 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (2.28.1)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (4.9.1)
Collecting multitasking<0.7
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from multitasking<0.7) (1.21.6)
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) (202.6)
Requirement already satisfied: charset-normalizer<3,>=2 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (2.1.1)
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) (2022.9.24)
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) (1.26.13)
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: 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 (0.15.3) supported by @QuantStack

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

Looking for: ['bs4==4.10.0']

```
pkgs/main/noarch      [=>]          [ (00m:00s)
pkgs/main/noarch      [>]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/noarch      [>]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/r/noarch         [<=]          [ (00m:00s)
pkgs/main/noarch      [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/r/noarch         [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/noarch      [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/r/noarch         [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/linux-64    [<=]          [ (00m:00s)
pkgs/main/noarch      [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/r/noarch         [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/linux-64    [=]          [ (00m:00s) 764 KB / ?? (2.45 MB/s)
pkgs/main/noarch      [<=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/r/noarch         [<=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/linux-64    [=]          [ (00m:00s) 764 KB / ?? (2.45 MB/s)
pkgs/main/noarch      [=]          [ (00m:00s) 776 KB / ?? (2.49 MB/s)
pkgs/r/noarch         [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/linux-64    [=]          [ (00m:00s) 764 KB / ?? (2.45 MB/s)
pkgs/main/noarch      [=]          [ (00m:00s) 776 KB / ?? (2.49 MB/s)
pkgs/r/noarch         [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
pkgs/main/linux-64    [=]          [ (00m:00s) 776 KB / ?? (2.49 MB/s)
pkgs/main/noarch      [=]          [ (00m:00s) 776 KB / ?? (2.49 MB/s)
pkgs/r/noarch         [=]          [ (00m:00s) 496 B / ?? (3.16 KB/s)
```

Pinned packages:

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	10 KB
<hr/>				
Upgrade:				
<hr/>				
- ca-certificates	2022.9.24	ha878542_0	installed	
+ ca-certificates	2023.01.10	h06a4308_0	pkgs/main/linux-64	120 KB
- certifi	2022.9.24	pyhd8ed1ab_0	installed	
+ certifi	2022.12.7	py37hb6a4308_0	pkgs/main/linux-64	150 KB
- openssl	1.1.1s	hb041bf4_1	installed	
+ openssl	1.1.1t	h7f8727e_0	pkgs/main/linux-64	4 MB
<hr/>				
Downgrade:				
<hr/>				
- beautifulsoup4	4.11.1	pyth70c72_0	installed	
+ beautifulsoup4	4.10.0	h16c4298_0	pkgs/main/noarch	95 KB

```
Install: 1 packages  
Upgrade: 3 packages  
Reinstall: 1 package
```

Table 3.1.1-4. M

```
  Downloading [>] ] (00m:00s) 3.43 KB/s
  Extracting [>] ] (-:--)
  Downloading [>] ] (00m:00s) 3.43 KB/s
  Extracting [>] ] (-:--)
  Downloading [>] ] (00m:00s) 67.46 KB/s
  Extracting [>] ] (-:--)
Finished bs4 (00m:00s) 10 KB 64 KB/s
  Downloading [>] ] (00m:00s) 67.46 KB/s
  Extracting [>] 1 (-:--)
```

```

Downloading [>] (00m:00s) 6/.46 KB/s
Extracting [>] (--:-)
Downloading [>] (00m:00s) 67.46 KB/s
Extracting [>] (--:-)
Downloading [>] (00m:00s) 1.00 MB/s
Extracting [>] (--:-)
Finished certifi (00m:00s) 150 KB 960 KB/s
Downloading [>] (00m:00s) 1.00 MB/s
Extracting [>] (--:-)
Downloading [>] (00m:00s) 1.00 MB/s
Extracting [>] (--:-)
Downloading [>] (00m:00s) 1.51 MB/s
Extracting [>] (--:-)
Finished beautifulsoup4 (00m:00s) 85 KB 532 KB/s
Downloading [>] (00m:00s) 1.51 MB/s
Extracting [>] (--:-)
Downloading [>] (00m:00s) 1.51 MB/s
Extracting [>] (--:-)
Downloading [>] (00m:00s) 1.51 MB/s
Extracting [>] (--:-)
Extracting [=====] (00m:00s) 1 / 5
Downloading [>] (00m:00s) 1.51 MB/s
Extracting [=====] (00m:00s) 1 / 5
Downloading [>] (00m:00s) 2.13 MB/s
Extracting [=====] (00m:00s) 1 / 5
Finished ca-certificates (00m:00s) 120 KB 718 KB/s
Downloading [>] (00m:00s) 2.13 MB/s
Extracting [=====] (00m:00s) 1 / 5
Downloading [>] (00m:00s) 2.13 MB/s
Extracting [=====] (00m:00s) 1 / 5
Downloading [>] (00m:00s) 2.13 MB/s
Extracting [=====] (00m:00s) 1 / 5
Downloading [>] (00m:00s) 2.13 MB/s
Extracting [=====] (00m:00s) 2 / 5
Downloading [>] (00m:00s) 2.13 MB/s
Extracting [=====] (00m:00s) 2 / 5
Downloading [>] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 2 / 5
Downloading [>] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 3 / 5
Downloading [>] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 3 / 5
Downloading [>] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 4 / 5
Finished openssl (00m:00s) 4 MB 19 MB/s
Downloading [=====] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 4 / 5
Downloading [=====] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 4 / 5
Downloading [=====] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 4 / 5
Downloading [=====] (00m:00s) 20.27 MB/s
Extracting [=====] (00m:00s) 4 / 5
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 20.5 MB/s eta 0:00:00
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) (5.6.0)
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: ipython-genutils in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (0.2.0)
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) (5.10.1)
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) (22.1.0)
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) (4.4.0)
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: importlib-metadata in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0) (4.11.4)
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) (0.19.2)
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) (3.11.0)
Installing collected packages: nbformat
  Attempting uninstall: nbformat
    Found existing installation: nbformat 5.7.0
      Uninstalling nbformat-5.7.0:
        Successfully uninstalled nbformat-5.7.0
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.
nbconvert 7.2.0 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
nbclient 0.7.2 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
jupyter-server 1.23.3 requires nbformat>=5.2.0, but you have nbformat 4.2.0 which is incompatible.
Successfully installed nbformat-4.2.0

```

```
[7]: 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
```

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.

```
[8]: def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Historical Revenue"), vertical_spacing=.3)
    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.astype("float"), name="Stock"))
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_specific.Revenue.astype("float"), 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,
    axis_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`.

```
[60]: tsla = yf.Ticker("TSLA")
print(tsla)

yfinance.Ticker object <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.

```
[62]: tesla_data = tsla.history(period="max")
tesla_data
```

```
[62]:      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
...
2023-03-03  194.800003 200.479996 192.880005 197.789993 153800400      0      0.0
2023-03-06  198.539993 198.600006 192.300003 193.809998 128100100      0      0.0
2023-03-07  191.380005 194.199997 186.100006 187.710007 148125800      0      0.0
2023-03-08  185.039993 186.500000 180.000000 182.000000 151897800      0      0.0
2023-03-09  180.250000 185.179993 172.509995 172.919998 168837500      0      0.0
```

3196 rows × 7 columns

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` datafram using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

```
[11]: tesla_data.reset_index(inplace=True)
tesla_data.head()
```

```
[11]:      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
```

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/IBMDveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm> Save the text of the response as a variable named `html_data`.

```
[63]: url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm"
html_data = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

```
[64]: soup = BeautifulSoup(html_data, 'html.parser')
```

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

► Click here if you need help locating the table

```
[66]: tesla_revenue = pd.DataFrame(columns=["Date", "Revenue"])
for row in soup.find_all("tbody")[1].find_all("tr"):
    col = row.find_all("td")
    date = col[0].text
    revenue = col[1].text
    tesla_revenue = tesla_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)
```

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

```
[67]: tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',', '').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.

```
[67]:      Date   Revenue
0  2022-09-30     21454
1  2022-06-30     16934
2  2022-03-31     18756
3  2021-12-31     17719
4  2021-09-30     13757
```

5	2021-06-30	11958
6	2021-03-31	10389
7	2020-12-31	10744
8	2020-09-30	8771
9	2020-06-30	6036
10	2020-03-31	5985
11	2019-12-31	7384
12	2019-09-30	6303
13	2019-06-30	6350
14	2019-03-31	4541
15	2018-12-31	7226
16	2018-09-30	6824
17	2018-06-30	4002
18	2018-03-31	3409
19	2017-12-31	3288
20	2017-09-30	2985
21	2017-06-30	2790
22	2017-03-31	2696
23	2016-12-31	2285
24	2016-09-30	2298
25	2016-06-30	1270
26	2016-03-31	1147
27	2015-12-31	1214
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	
52	2009-09-30	46
53	2009-06-30	27

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

```
[68]: tesla_revenue.dropna(inplace=True)
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

	Date	Revenue
0	2022-09-30	21454
1	2022-06-30	16934
2	2022-03-31	18756
3	2021-12-31	17719
4	2021-09-30	13757
5	2021-06-30	11958
6	2021-03-31	10389
7	2020-12-31	10744
8	2020-09-30	8771
9	2020-06-30	6036
10	2020-03-31	5985

```

11 2019-12-31    7384
12 2019-09-30    6303
13 2019-06-30    6350
14 2019-03-31    4541
15 2018-12-31    7226
16 2018-09-30    6824
17 2018-06-30    4002
18 2018-03-31    3409
19 2017-12-31    3288
20 2017-09-30    2985
21 2017-06-30    2790
22 2017-03-31    2696
23 2016-12-31    2285
24 2016-09-30    2298
25 2016-06-30    1270
26 2016-03-31    1147
27 2015-12-31    1214
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
52 2009-09-30    46
53 2009-06-30    27

```

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

```
[69]: 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`.

```
[71]: GameStop = yf.Ticker("GME")
print(GameStop)
```

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.

```
[73]: gme_data = GameStop.history(period="max")
gme_data
```

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2002-02-13	1.620129	1.693350	1.603296	1.691667	76216000	0.0	0.0
2002-02-14	1.620129	1.693350	1.603296	1.691667	110216000	0.0	0.0

2002-02-14	1.712707	1.716074	1.693350	1.683250	11021600	0.0	0.0
2002-02-15	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
2002-02-19	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
2002-02-20	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0
...
2023-03-03	18.190001	19.100000	18.100000	18.700001	2502000	0.0	0.0
2023-03-06	18.520000	19.500000	18.520000	18.650000	2815300	0.0	0.0
2023-03-07	18.540001	18.840000	18.059999	18.059999	2375300	0.0	0.0
2023-03-08	18.049999	18.379999	17.629999	18.170000	3277800	0.0	0.0
2023-03-09	18.070000	18.379999	17.480000	17.520000	2644900	0.0	0.0

5304 rows × 7 columns

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.

[38]:	<code>gme_data.reset_index(inplace=True) gme_data.head()</code>																																																						
[38]:	<table border="1"><thead><tr><th></th><th>Date</th><th>Open</th><th>High</th><th>Low</th><th>Close</th><th>Volume</th><th>Dividends</th><th>Stock Splits</th></tr></thead><tbody><tr><td>0</td><td>2002-02-13</td><td>1.620128</td><td>1.693350</td><td>1.603296</td><td>1.691666</td><td>7621600</td><td>0.0</td><td>0.0</td></tr><tr><td>1</td><td>2002-02-14</td><td>1.712707</td><td>1.716074</td><td>1.670626</td><td>1.683250</td><td>11021600</td><td>0.0</td><td>0.0</td></tr><tr><td>2</td><td>2002-02-15</td><td>1.683251</td><td>1.687459</td><td>1.658002</td><td>1.674834</td><td>8389600</td><td>0.0</td><td>0.0</td></tr><tr><td>3</td><td>2002-02-19</td><td>1.666418</td><td>1.666418</td><td>1.578047</td><td>1.607504</td><td>7410400</td><td>0.0</td><td>0.0</td></tr><tr><td>4</td><td>2002-02-20</td><td>1.615921</td><td>1.662210</td><td>1.603296</td><td>1.662210</td><td>6892800</td><td>0.0</td><td>0.0</td></tr></tbody></table>		Date	Open	High	Low	Close	Volume	Dividends	Stock Splits	0	2002-02-13	1.620128	1.693350	1.603296	1.691666	7621600	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	1.683251	1.687459	1.658002	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
	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits																																															
0	2002-02-13	1.620128	1.693350	1.603296	1.691666	7621600	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	1.683251	1.687459	1.658002	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/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html>. Save the text of the response as a variable named `html_data`.

```
[57]: url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html"
html_data = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

```
[58]: soup = BeautifulSoup(html_data, 'html.parser')
```

Using `BeautifulSoup` or the `read_html` function extract the table with `GameStop Quarterly 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

```
[74]: gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])

for row in soup.find_all("tbody")[1].find_all("tr"):
    col = row.find_all("td")
    date = col[0].text
    revenue = col[1].text

    gme_revenue = gme_revenue.append({"Date":date, "Revenue":revenue}, ignore_index=True)
```

```
[75]: gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',', '').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.
```

```
[78]: gme_revenue.dropna(inplace=True)

gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
print(gme_revenue)
```

	Date	Revenue
0	2022-09-30	21454
1	2022-06-30	16934
2	2022-03-31	18756
3	2021-12-31	17719
4	2021-09-30	13757
5	2021-06-30	11958
6	2021-03-31	10389
7	2020-12-31	10744
8	2020-09-30	8771
9	2020-06-30	6036
10	2020-03-31	5985
11	2019-12-31	7384
12	2019-09-30	6303
13	2019-06-30	6350
14	2019-03-31	4541
15	2018-12-31	7226
16	2018-09-30	6824
17	2018-06-30	4002
18	2018-03-31	3409
19	2017-12-31	3288
20	2017-09-30	2985
21	2017-06-30	2790
22	2017-03-31	2696
23	2016-12-31	2285
24	2016-09-30	2298
25	2016-06-30	1270
26	2016-03-31	1147
27	2015-12-31	1214
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
52 2009-09-30      46
53 2009-06-30      27

```

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

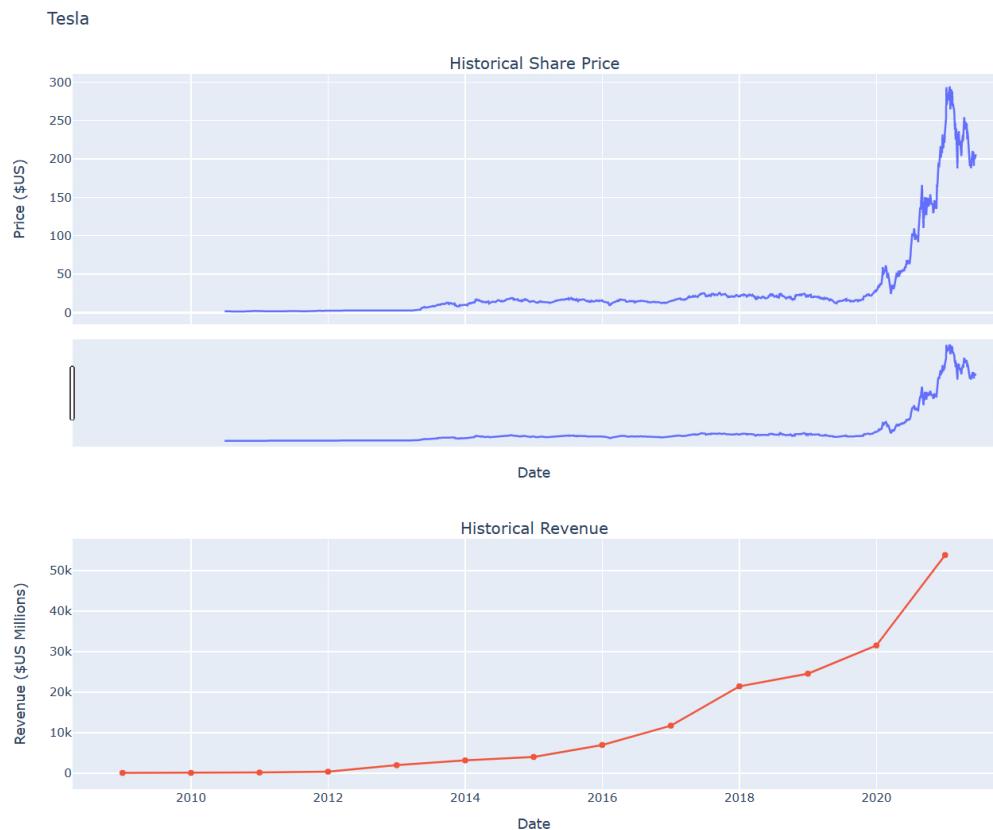
```
[77]: gme_revenue.tail()
```

	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 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.

```
[115]: make_graph(tesla_data, tesla_revenue, 'Tesla')
```

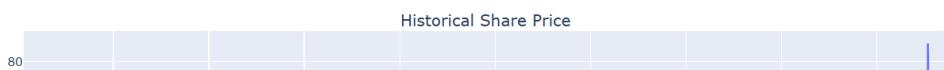


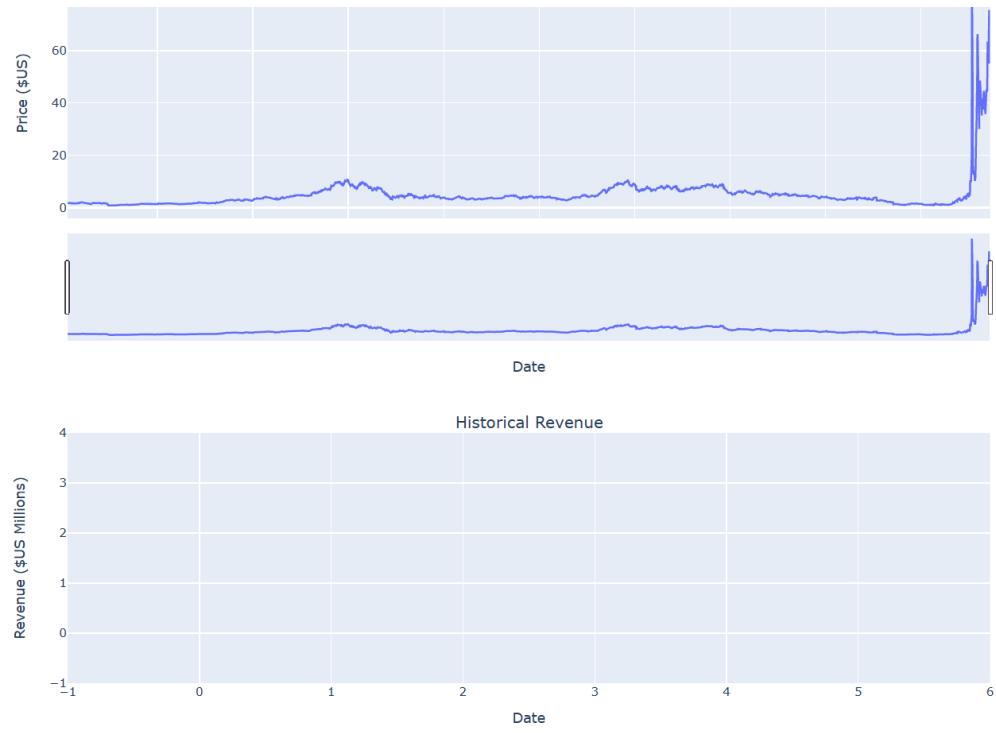
Question 6: Plot GameStop Stock Graph

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.

```
[116]: make_graph(gme_data, gme_revenue, 'GameStop')
```

GameStop





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.