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`.

```
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.

```
tesla_data = tesla.history(period="max")
```

Reset the index, save, 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.

```
tesla_data.reset_index(inplace=True)

tesla_data.head()
```

	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 - Extracting Tesla Revenue Data Using Webscraping

Question 2: Use Webscraping to Extract Tesla Revenue Data

Use the `requests` library to download the webpage https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue. Save the text of the response as a variable named `html_data`.

[7]:
```
url= 'https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue'
html_data= requests.get(url).text
```

***

[8]:
```
soup= BeautifulSoup(html_data, 'html.parser')
```

***

► Click here if you need help locating the table

[9]:
```
tables= soup.find_all('table')
len(tables)
```

[9]:
```
6
```

[10]:
```
for index, table in enumerate(tables):
    if ('Tesla Quarterly Revenue' in str(table)):
        table_index= index
print(table_index)
```

```
1
```

[11]:
```
print(tables[table_index].prettify())
```

***

[12]:
```
tesla_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
        tesla_revenue = tesla_revenue.append({'Date': Date, 'Revenue': Revenue}, ignore_index= True)
tesla_revenue
```

***

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

[13]:
```
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 an null or empty strings in the Revenue column.

[14]:
```
tesla_revenue.dropna(inplace=True)

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

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

[15]:
```
tesla_revenue.tail()
```

[15]:

	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 2 - Extracting Tesla Revenue Data Using Webscraping

	Date	Revenue
41	2010-09-30	31
42	2010-06-30	28
43	2010-03-31	21
45	2009-09-30	46
46	2009-06-30	27

Question 3 - Extracting GameStop Stock Data Using yfinance

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`.

```
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.

```
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.

```
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	1.691666	76216000	0.0	0.0
1	2002-02-14	1.712707	1.716073	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	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0

Question 3 - Extracting GameStop Stock Data Using yfinance

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`.

[16]:
```
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.

[17]:
```
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.

[18]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

[18]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620128	1.693350	1.603296	1.691666	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	1.666417	1.666417	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0

Question 4 - Extracting GameStop Revenue Data Using Webscraping

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`.

[19]:
```
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`.

[20]:
```
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

[21]:
```
tables_2= soup.find_all('table')
for index, table2 in enumerate(tables_2):
    if ('GameStop Quarterly Revenue' in str(table2)):
        table_index= index
print(table_index)
```

```
1
```

[22]:
```
gme_revenue= pd.DataFrame(columns=['Date', 'Revenue'])
for row in tables_2[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
```

***

[23]:
```
gme_revenue["Revenue"] = gme_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.
```

[24]:
```
gme_revenue.dropna(inplace=True)
gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
```

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

[25]:
```
gme_revenue.tail()
```

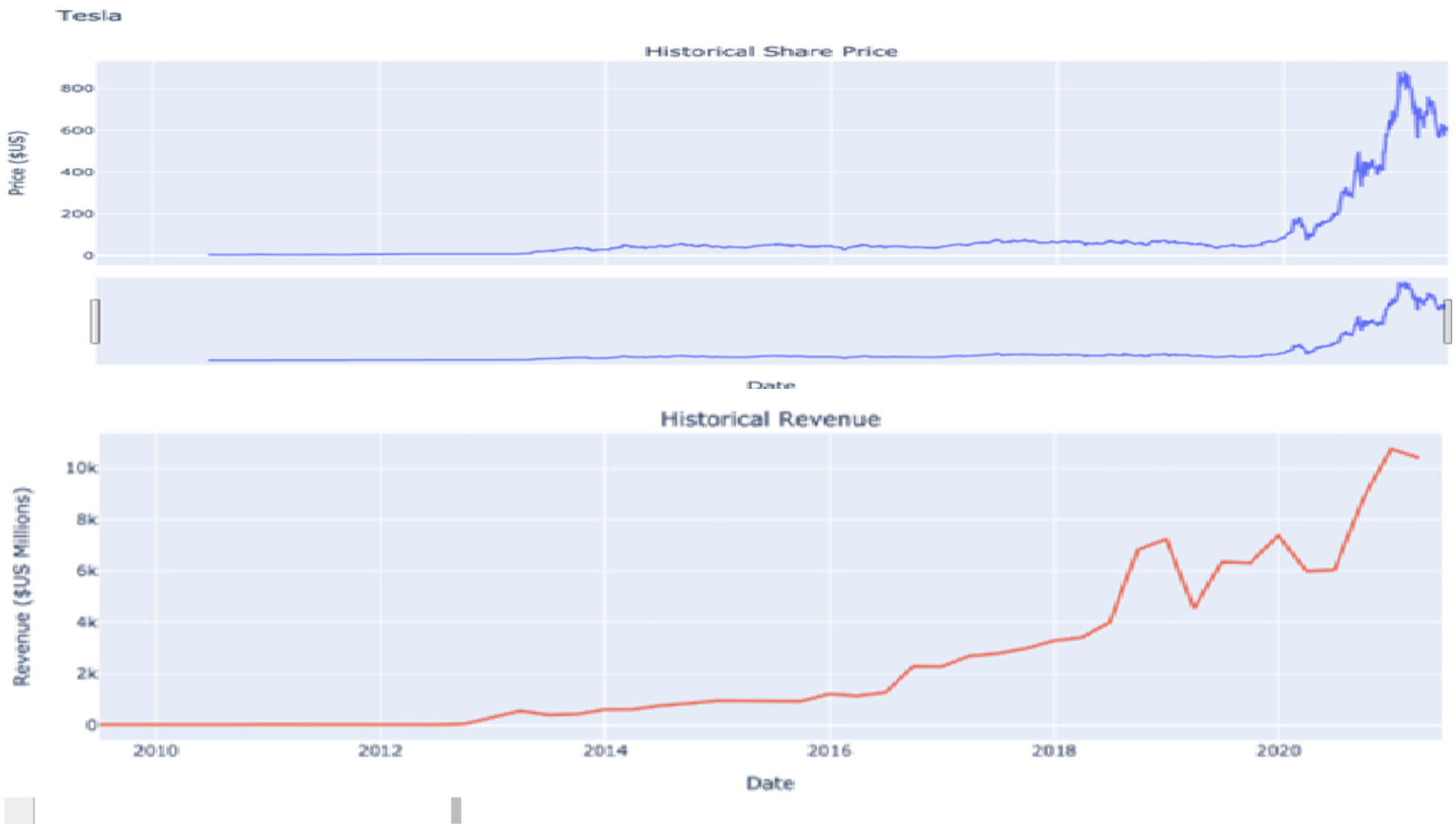
[25]:

	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 4 - Extracting GameStop Revenue Data Using Webscraping

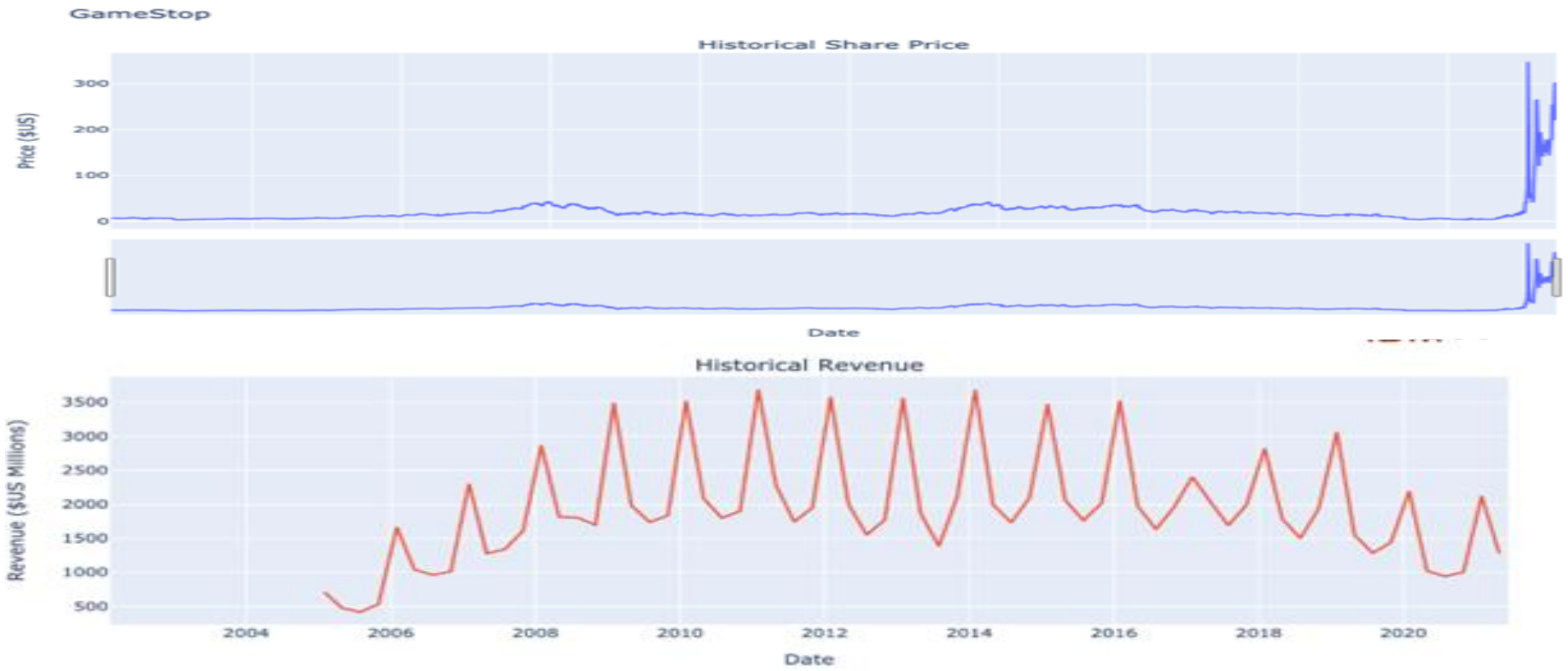
	Date	Revenue
59	2006-01-31	1667
60	2005-10-31	534
61	2005-07-31	416
62	2005-04-30	475
63	2005-01-31	709

Question 5 - Tesla Stock and Revenue Dashboard



Question 5 - Tesla Stock and Revenue Dashboard

Question 6 - GameStop Stock and Revenue Dashboard



Question 6 - GameStop Stock and Revenue Dashboard

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.

```
[27]: make_graph(gme_data, gme_revenue, '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