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`.

[11]:
```
Tesla=yf.Ticker("TSLA")
Tesla
```

[11]:
```
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.

[10]:
```
stock_info=Tesla.history(period="max")
tesla_data=pd.DataFrame(stock_info)
tesla_data
```

[10]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
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
...	...	...	...	...	...	...	...
2023-06-08	224.220001	235.229996	223.009995	234.860001	164489700	0	0.0

2010-07-06	1.333333	1.333333	1.055333	1.074000	103003500	0	0.0
...	...	...	...	...	...	...	...
2023-06-08	224.220001	235.229996	223.009995	234.860001	164489700	0	0.0
2023-06-09	249.070007	252.419998	242.020004	244.399994	199882300	0	0.0
2023-06-12	247.940002	250.970001	244.589996	249.830002	150337900	0	0.0
2023-06-13	253.509995	259.679993	251.339996	258.709991	162384300	0	0.0
2023-06-14	260.170013	261.570007	250.500000	256.790009	169921000	0	0.0

3263 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` dataframe using the `head` function. Take a screenshot of the results and code from the beginning of Question 1 to the results below.

[8]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

[8]:

	index	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	0	2010-06-29	1.266667	1.666667	1.169333	1.592667	281494500	0	0.0
1	1	2010-06-30	1.719333	2.028000	1.553333	1.588667	257806500	0	0.0
2	2	2010-07-01	1.666667	1.728000	1.351333	1.464000	123282000	0	0.0
3	3	2010-07-02	1.533333	1.540000	1.247333	1.280000	77097000	0	0.0
4	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/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm Save the text of the response as a variable named `html_data`.

[11]:
```
url="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm"
html_data=requests.get(url)
```

Parse the html data using `beautiful_soup`.

[12]:
```
soup=BeautifulSoup(html_data.text,"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 dataframe should have columns `Date` and `Revenue`.

▶ Click here if you need help locating the table

[18]:
```
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)
tesla_revenue
```

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

[32]:
```
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.

[21]:
```
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.

[22]:
```
tesla_revenue.tail()
```

[22]:

	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`.

[43]:
```
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.

[44]:
```
stock_infos=GameStop.history(period="max")
gme_data=pd.DataFrame(stock_infos)
gme_data
```

[44]:

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	1.712707	1.716074	1.670626	1.683250	11021600	0.0	0.0
2002-02-15	1.683251	1.687459	1.658002	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
...	...	...	...	...	...	...	...

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
...	...	...	...	...	...	...	...
2023-06-09	21.780001	23.430000	21.500000	22.680000	10321200	0.0	0.0
2023-06-12	22.850000	24.299999	22.740000	24.299999	7131400	0.0	0.0
2023-06-13	26.200001	27.650000	25.030001	26.950001	17160600	0.0	0.0
2023-06-14	26.719999	27.080000	24.900000	25.700001	7243700	0.0	0.0
2023-06-15	25.410000	26.170000	24.639999	24.840000	5473100	0.0	0.0

5372 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.

[45]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

[45]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	1.620129	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	1.683251	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	1.615920	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`.

[68]:
```
link="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html"
html_data=requests.get(link)
```

Parse the html data using `beautiful_soup`.

[69]:
```
soup=BeautifulSoup(html_data.text,"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

[73]:
```
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)
gme_revenue
```

[73]:

	Date	Revenue
0	2020-04-30	$1,021
1	2020-01-31	$2,194
2	2019-10-31	$1,439

Final Assignment.ipynb

Code | Run as Pipeline | Python

4	2019-04-30	$1,548
...	...	...
57	2006-01-31	$1,667
58	2005-10-31	$534
59	2005-07-31	$416
60	2005-04-30	$475
61	2005-01-31	$709

62 rows × 2 columns

[115]:
```
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.
```

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

[113]:
```
gme_revenue.tail()
```

[113]:

	Date	Revenue
57	2006-01-31	$1,667
58	2005-10-31	$534
59	2005-07-31	$416
60	2005-04-30	$475
61	2005-01-31	$709

Final Assignment.ipynb

Code | git | Run as Pipeline | Python

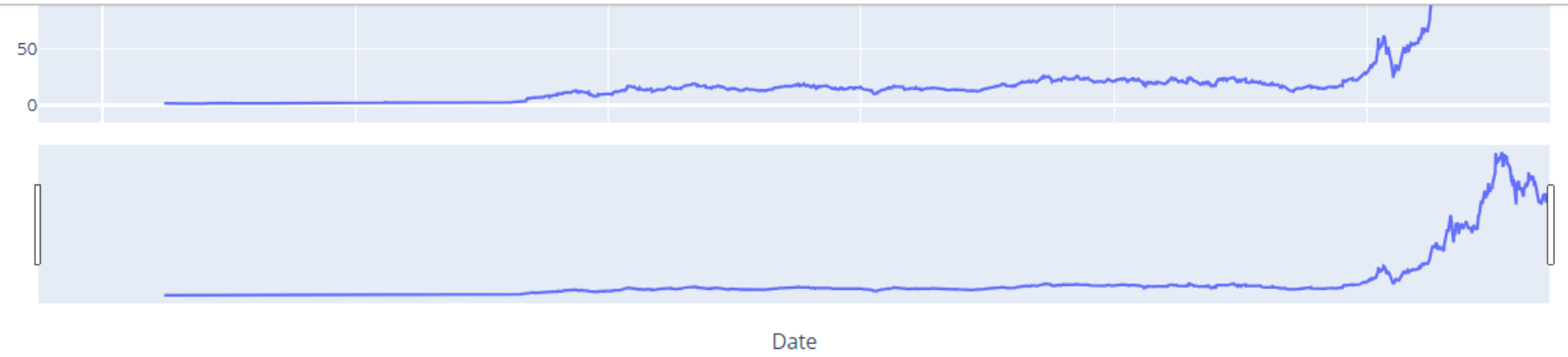
Question 5: Plot Tesla Stock Graph

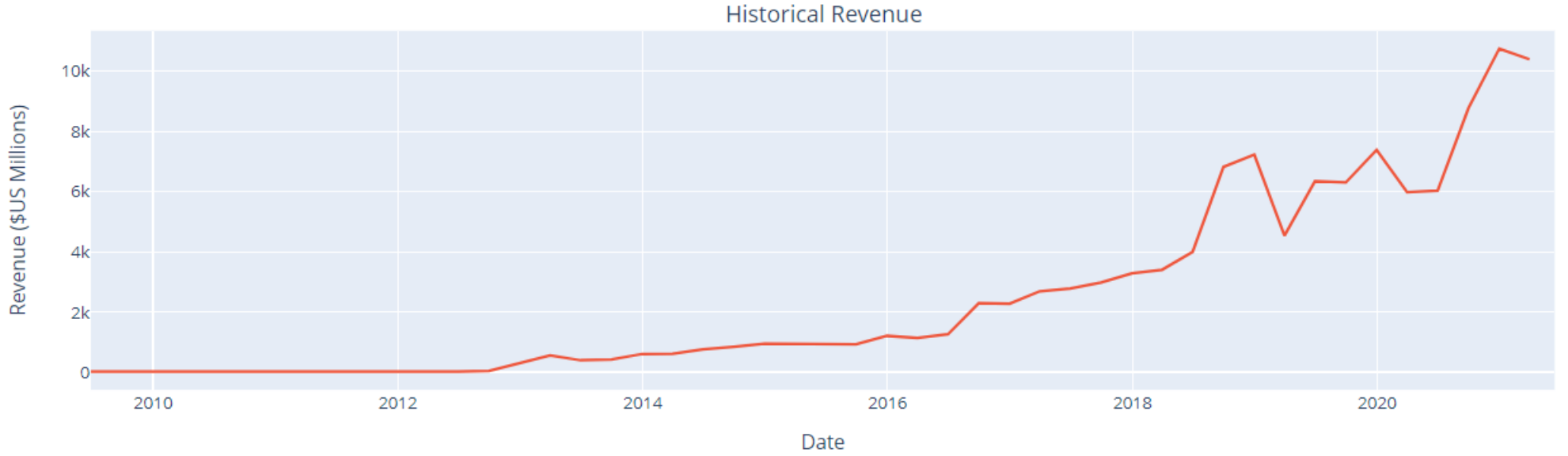
Use the `make_graph` function to graph the Tesla Stock Data, also provide a title for the graph. The structure to call the `make_graph` function is `make_graph(tesla_data, tesla_revenue, 'Tesla')`. Note the graph will only show data upto June 2021.

[103]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla')
```

Final Assignment.ipynb

Code | git | Run as Pipeline | Python





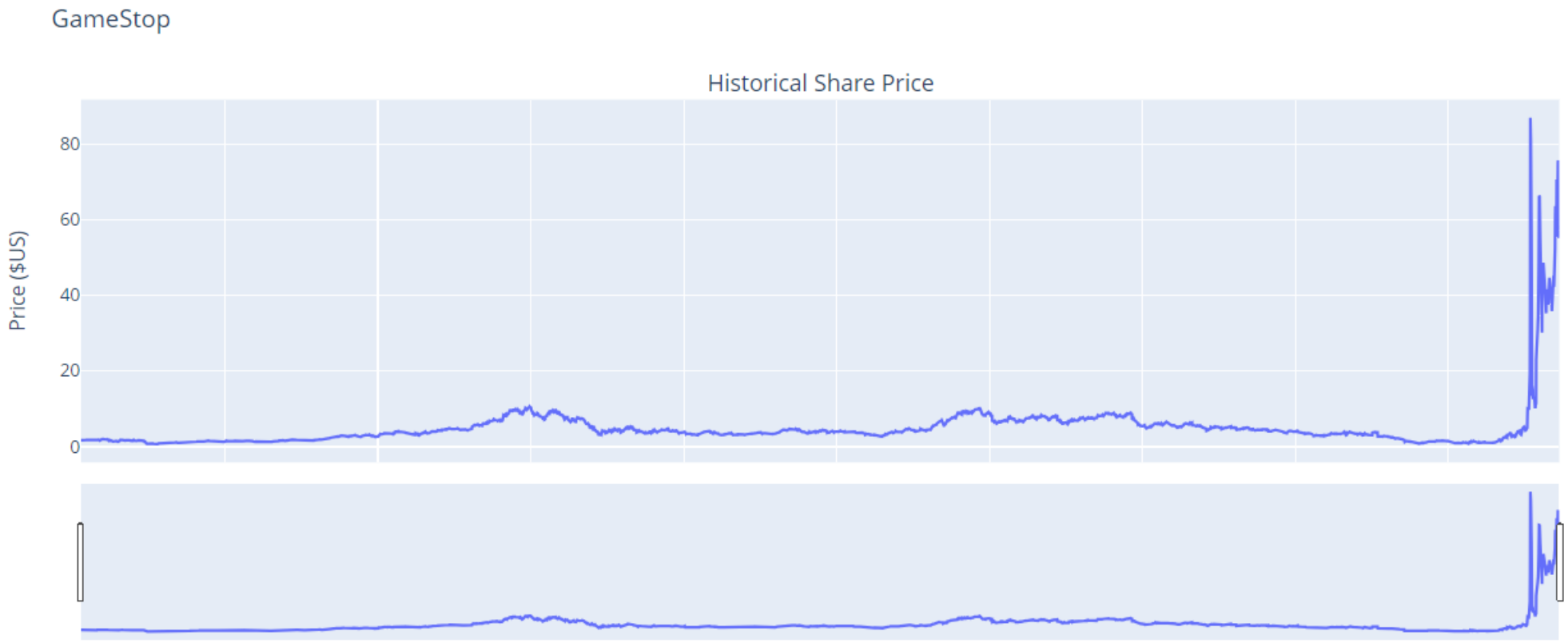
Final Assignment.ipynb

Code | git | Run as Pipeline | Python

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')
```



Final Assignment.ipynb

Code | git | Run as Pipeline | Python