New - Tesla and GME Share Price and Revenue Data

July 10, 2024

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.

[1]:
```
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.

[2]:
```
import warnings
# Ignore all warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

0.1 Define Graphing Function

[3]:
```
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),␣
 ↪y=stock_data_specific.Close.astype("float"), name="Share Price"), row=1,␣
 ↪col=1)
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date),␣
 ↪y=revenue_data_specific.Revenue.astype("float"), 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()
```

0.2 Use yfinance to Extract Stock Data of Tesla

[4]:
```
tesla = yf.Ticker('TSLA')
print(tesla)
```

```
yfinance.Ticker object <TSLA>
```

[5]:
```
tesla_data = tesla.history(period='max')
tesla_data
```

[5]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0
…	…	…	…	…	…	…	…
2024-07-03 00:00:00-04:00	234.559998	248.350006	234.250000	246.389999	166561500	0.0	0.0
2024-07-05 00:00:00-04:00	249.809998	252.369995	242.460007	251.520004	154501200	0.0	0.0
2024-07-08 00:00:00-04:00	247.710007	259.440002	244.570007	252.940002	157219600	0.0	0.0
2024-07-09 00:00:00-04:00	251.000000	265.609985	250.300003	262.329987	160210900	0.0	0.0
2024-07-10 00:00:00-04:00	262.829987	265.720001	257.859985	265.387604	89013656	0.0	0.0

```
[3531 rows x 7 columns]
```

[6]:
```
tesla_data.reset_index(inplace=True)

tesla_data.head()
```

[6]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0
1	2010-06-30 00:00:00-04:00	1.719333	2.028000	1.553333	1.588667	257806500	0.0	0.0
2	2010-07-01 00:00:00-04:00	1.666667	1.728000	1.351333	1.464000	123282000	0.0	0.0
3	2010-07-02 00:00:00-04:00	1.533333	1.540000	1.247333	1.280000	77097000	0.0	0.0
4	2010-07-06 00:00:00-04:00	1.333333	1.333333	1.055333	1.074000	103003500	0.0	0.0

0.3 Use Webscraping to Extract Tesla Revenue Data

Site URL- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm

[7]:
```
url = " https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
↪IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm"

html_data = requests.get(url).text
```

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

[9]:
```
read_html_pandas_data = pd.read_html(str(soup))
tesla_revenue = read_html_pandas_data[1]
tesla_revenue.columns = ['Date', 'Revenue']
tesla_revenue
```

[9]:

	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	NaN
52	2009-09-30	$46
53	2009-06-30	$27

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

[10]:
```
tesla_revenue['Revenue'] = tesla_revenue['Revenue'].replace({'\$': '', ',':␣
↪''}, regex=True)
```

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

[11]:
```
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.

[12]:
```
last_5_rows = tesla_revenue.tail(5)
last_5_rows
```

[12]:

	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

0.4 Use yfinance to Extract Stock Data of GME.

[13]:
```
gme = yf.Ticker("GME")
gme
```

[13]:
```
yfinance.Ticker object <GME>
```

[14]:
```
gme_data = gme.history(period="max")
gme_data
```

[14]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2002-02-13 00:00:00-05:00	1.620129	1.693350	1.603296	1.691667	76216000	0.0	0.0
2002-02-14 00:00:00-05:00	1.712707	1.716074	1.670626	1.683250	11021600	0.0	0.0
2002-02-15 00:00:00-05:00	1.683251	1.687459	1.658002	1.674834	8389600	0.0	0.0
2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
2002-02-20 00:00:00-05:00	1.615920	1.662210	1.603296	1.662210	6892800	0.0	0.0
…	…	…	…	…	…	…	…
2024-07-03 00:00:00-04:00	24.030001	24.889999	23.650000	24.370001			
2024-07-05 00:00:00-04:00	24.180000	25.080000	23.820000	24.180000			
2024-07-08 00:00:00-04:00	24.120001	25.139999	23.850000	24.450001			
2024-07-09 00:00:00-04:00	24.600000	25.180000	24.000000	24.600000			
2024-07-10 00:00:00-04:00	25.000000	26.450001	24.938101	25.409901			

```
…                                 …         …             …
2024-07-03 00:00:00-04:00  11829500         0.0           0.0
2024-07-05 00:00:00-04:00  11782100         0.0           0.0
2024-07-08 00:00:00-04:00  11815500         0.0           0.0
2024-07-09 00:00:00-04:00   9419800         0.0           0.0
2024-07-10 00:00:00-04:00  17806571         0.0           0.0

[5639 rows x 7 columns]
```

[15]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

[15]:

	Date	Open	High	Low	Close	Volume	\
0	2002-02-13 00:00:00-05:00	1.620129	1.693350	1.603296	1.691667	76216000	
1	2002-02-14 00:00:00-05:00	1.712707	1.716074	1.670626	1.683250	11021600	
2	2002-02-15 00:00:00-05:00	1.683251	1.687459	1.658002	1.674834	8389600	
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	
4	2002-02-20 00:00:00-05:00	1.615920	1.662210	1.603296	1.662210	6892800	

	Dividends	Stock Splits
0	0.0	0.0
1	0.0	0.0
2	0.0	0.0
3	0.0	0.0
4	0.0	0.0

0.5 Use Webscraping to Extract GME Revenue Data

Site URL - https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html.

[16]:
```
url = " https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
↪IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html"

html_data_2 = requests.get(url).text
```

[17]:
```
soup = BeautifulSoup(html_data_2, 'html.parser')
```

[18]:
```
read_html_pandas_data_2 = pd.read_html(str(soup))
gme_revenue = read_html_pandas_data_2[1]
gme_revenue.columns = ['Date', 'Revenue']
gme_revenue['Revenue'] = gme_revenue['Revenue'].replace({'\$': '', ',': ''},␣
↪regex=True)
gme_revenue.dropna(inplace=True)
gme_revenue = gme_revenue[gme_revenue['Revenue'] != ""]
gme_revenue
```

[18]:

	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 x 2 columns]

[19]:
```
last_5_rows_GME = gme_revenue.tail(5)
last_5_rows_GME
```

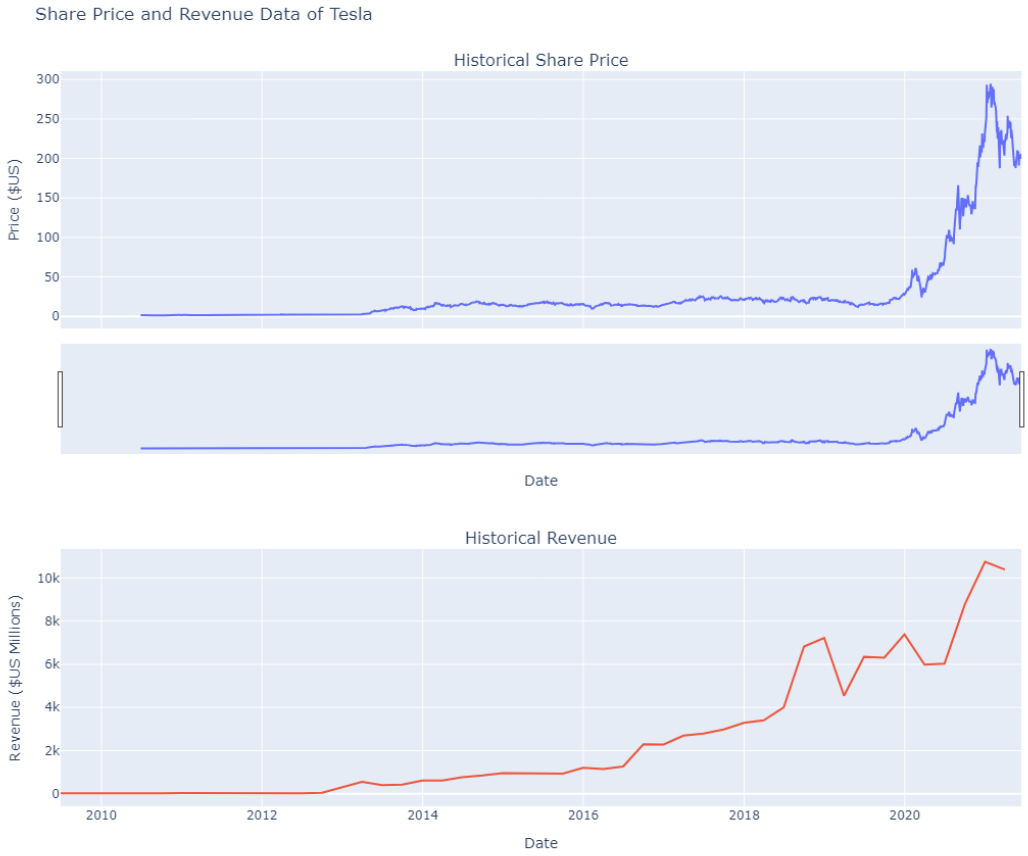
[19]:

	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

0.6 Plot Tesla Stock Graph

0.6.1 Note: The graph will only show data upto June 2021.

[20]:
```
make_graph(tesla_data, tesla_revenue, 'Share Price and Revenue Data of Tesla')
```



0.7 Plot GameStop Stock Graph

[21]:
```
make_graph(gme_data, gme_revenue, 'Share Price and Revenue Data of GME')
```


Share Price and Revenue Data of GME
Historical Share Price
Price ($US)
80
60
40
20
0
Date
Historical Revenue
Revenue ($US Millions)
3500
3000
2500
2000
1500
1000
500
2004
2006
2008
2010
2012
2014
2016
2018
2020
Date