Stock Price Analysis - Tesla & GameStop

Author: Satyndra Kumar Gautam

Email: gautam.ishu5@gmail.com

For this assignment, as a data scientist working for an investment firm, the main goal is to extract the revenue data for Tesla and GameStop and build a dashboard to compare the price of the stock vs the revenue. It includes basic descriptive analysis.

◉ Table of Contents

◉ 1.0 Introduction

• Overview

Determining the price of a stock is complex; it depends on the number of outstanding shares, the size of the company's future profits, and much more. An essential factor is the company's profit and growth of profits; if the company's profit is increasing, the stock price should increase. If you suspect the company's profit increases, you should buy the stock as the stock should increase, But what happens if you think the stock price will decrease.

A few years ago, many short sellers *(Short sellers are those people who borrow a stock, sells the stock, and then repurchases it to return it to the lender)* targeted **Tesla**. Then, Tesla started becoming profitable, and profits were increasing; thus, the company stock went up. This was based on the companies performance, so the stock should continue to rise, and the short seller should sell the stock.

While for **GameStop** scenario, many individuals from a community named WallStreetBets started buying into shares and then, the influx of demand caused GameStop shares to soar. All this produced billions of dollars in losses for hedge funds who had sold the stock short.[Test](#section-two)

● About Company

Tesla: Tesla, Inc. is an American multinational automotive and clean energy company headquartered in Austin, Texas. Tesla designs and manufactures electric vehicles, stationary battery energy storage devices from home to grid-scale, solar panels and solar roof tiles, and related products and services. Click here for more details

GameStop: GameStop Corp. is an American video game, consumer electronics, and gaming merchandise retailer. The company is headquartered in Grapevine, Texas (a suburb of Dallas), and is the largest video game retailer worldwide.Click here for more details

● Purpose

The purpose of this project is to simply extract data through webscraping and yfinance for Tesla and GameStop and compare their stock price and revenue and also, look through historical data to find out patterns and insights (descriptive) which could be a alarming for these companies.

◉ 2.0 Objectives of this assignment

● Installed Required Packages and Libraries

In order to extract the data for both companies according to the tasks, we need to install `yfinance` & `BeautifulSoup`

In [1]:

```
#libraries needed
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 a Function that Makes a Graph

Given details with this assignment, we must define a function which will produce graph and then, Date, Close and Revenue and name of stock must be included in the dataframes.

In [2]:

```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Histori
    stock_data_specific = stock_data[stock_data.Date <= '2019--01-01']
    revenue_data_specific = revenue_data[revenue_data.Date <= '2019-01-01']
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data_specific.Date), y=stock_da
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date), y=revenu
    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=700,
    title=stock,
    xaxis_rangeslider_visible=True)
    fig.show()
```

● Use yfinance to Extract Stock Data for Tesla

Here, we are going to extract the Tesla's stock price data by using the yfinance library which has `Ticker` function focused on extracting the historical data of a particular company.

In [3]:

```
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 1000y or more, as per your convinience but not `max' since it is not working in my system or notebook. No idea, so we get information for the maximum amount of time.

In [4]:

```
tesla_data = tesla.history(period = '90y')
tesla_data.head(5)
```

Out[4]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits
Date							
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

Furthermore, let's use `reset_index(inplace = True)` function to reset the above dataframe.

In [5]:

```
tesla_data.reset_index(inplace = True)
```

According to the assignment, let's show the first 5 rows of the tesla_data by using `head`.

In [6]:

```
tesla_data.head(5)
```

Out[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

• Use Webscraping to Extract Tesla Revenue Data

Here, we are going to extract the Tesla's revenue by using the webscraping methods such as `request` & `BeautifulSoup` and then, we will display the results as per assignment questions.

In [7]:
```
# create df url
url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDevelo

# now send a request to the webpage and extract the html data
html_data = requests.get(url).text

# parse the html data by using BeautifulSoup and html.parser
soup = BeautifulSoup(html_data, 'html.parser')

#soup.title.string

# find out how many tables are there

tables = soup.find_all('table')
len(tables)
```

Out[7]:
```
6
```

Per assignment description, 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`.

In [8]:

```python
# create for loop in order to find the index of Tesla Quarterly Revenue

for index,table in enumerate(tables):
    if ("Tesla Quarterly Revenue" in str(table)):
        table_index = index
        print(table_index)
```

```
1
```

Now, let's create a dummy dataframe `tesla_revenue` with `Date` and `Revenue`.

In [9]:

```python
# create a dummy df
tesla_revenue = pd.DataFrame(columns = ["Date", "Revenue"])

# create a for loop to assign all the values
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}, ign
```

As per assignment details, we need to remove **$, comma, null, empty strings** etc. from the dataframe. In order to do so, we are going to use `replace` function.

In [10]:

```python
#print(tesla_revenue.isna().sum())
tesla_revenue.dropna(inplace = True)
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
tesla_revenue["Revenue"] = tesla_revenue["Revenue"].str.replace('$','').str.replace
tesla_revenue.head(5)
```

Out[10]:

	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

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

In [11]:

```
tesla_revenue.tail(5)
```

Out[11]:

	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

• Use yfinance to Extract Stock Data for GameStop

Here, we are going to extract the GameStop's stock price data by using the yfinance library which has `Ticker` function focused on extracting the historical data of a particular company.

Please note: All the process for `GME` are as same as previous process for Tesla.

In [12]:

```
# creating object for GME
gme_ticker = yf.Ticker('GME')

# Exract historical data by using history()
gme_data = gme_ticker.history(period = '100y')

#reset the index of the dataframe
gme_data.reset_index(inplace = True)
gme_data.head(5)
```

Out[12]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691666	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712708	1.716074	1.670626	1.683251	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615920	1.662209	1.603296	1.662209	6892800	0.0	0.0

According to the assignment, let's show the first 5 rows of the gme_data by using `head`.

In [13]:

```
gme_data.head(5)
```

Out[13]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691666	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712708	1.716074	1.670626	1.683251	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683250	1.687458	1.658001	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615920	1.662209	1.603296	1.662209	6892800	0.0	0.0

• Use Webscraping to Extract GME Revenue Data

Here, we are going to extract the **GME's** revenue by using the webscraping methods such as `request` & `BeautifulSoup` and then, we will display the results as per assignment questions.

In [14]:
```
# create url from the webpage
gme_url = 'https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDe

# send requests to obtain the html data
gme_html_data = requests.get(gme_url).text

# parse the html data by using BeautifulSoup
gme_soup = BeautifulSoup(gme_html_data, 'html.parser')
```

Let's find out how many tables this data contains

In [15]:
```
tables = gme_soup.find_all('table')
len(tables)
```

Out[15]:
```
6
```

Let's find the index of the table we require to do the further analysis as per assignment descriptions provided. Extract the table with **GameStop Quarterly Revenue**

In [16]:
```
# create a for loop
for index, table in enumerate(tables):
    if("Game Quarterly Revenue" in str(table)):
        table_index = index

table_index
```

Out[16]:
```
1
```

As we can see from the above result, the second table is what we need. Now, we gotta extract the data accordingly. First we need to create dummy df and then store the data in the dummy data.

In [17]:
```
# create a dummy df
gme_revenue = pd.DataFrame(columns = ["Date", "Revenue"])

#create for loop to map and store the data we require
for row in tables[table_index].tbody.find_all('tr'):
    col = row.find_all('td')
    if(col != []):
        col1 = col[0].text
        col2 = col[1].text
        gme_revenue = gme_revenue._append({"Date":col1, "Revenue":col2}, ignore_ind

gme_revenue.head(5)
```

Out[17]:

	Date	Revenue
0	2020-04-30	$1,021
1	2020-01-31	$2,194
2	2019-10-31	$1,439
3	2019-07-31	$1,286
4	2019-04-30	$1,548

Now, let's remove **$, comma, NA, Empty Strings** etc from the columns if exists.

In [18]:

```
gme_revenue.dropna()

gme_revenue = gme_revenue[gme_revenue["Revenue"] != '']

gme_revenue ["Revenue"] = gme_revenue["Revenue"].str.replace('$', '').str.replace('

gme_revenue.head(5)
```

Out[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

As per assignment question: Display the last 5 rows of `gme_revenue`.

In [19]:

```
gme_revenue.tail(5)
```

Out[19]:

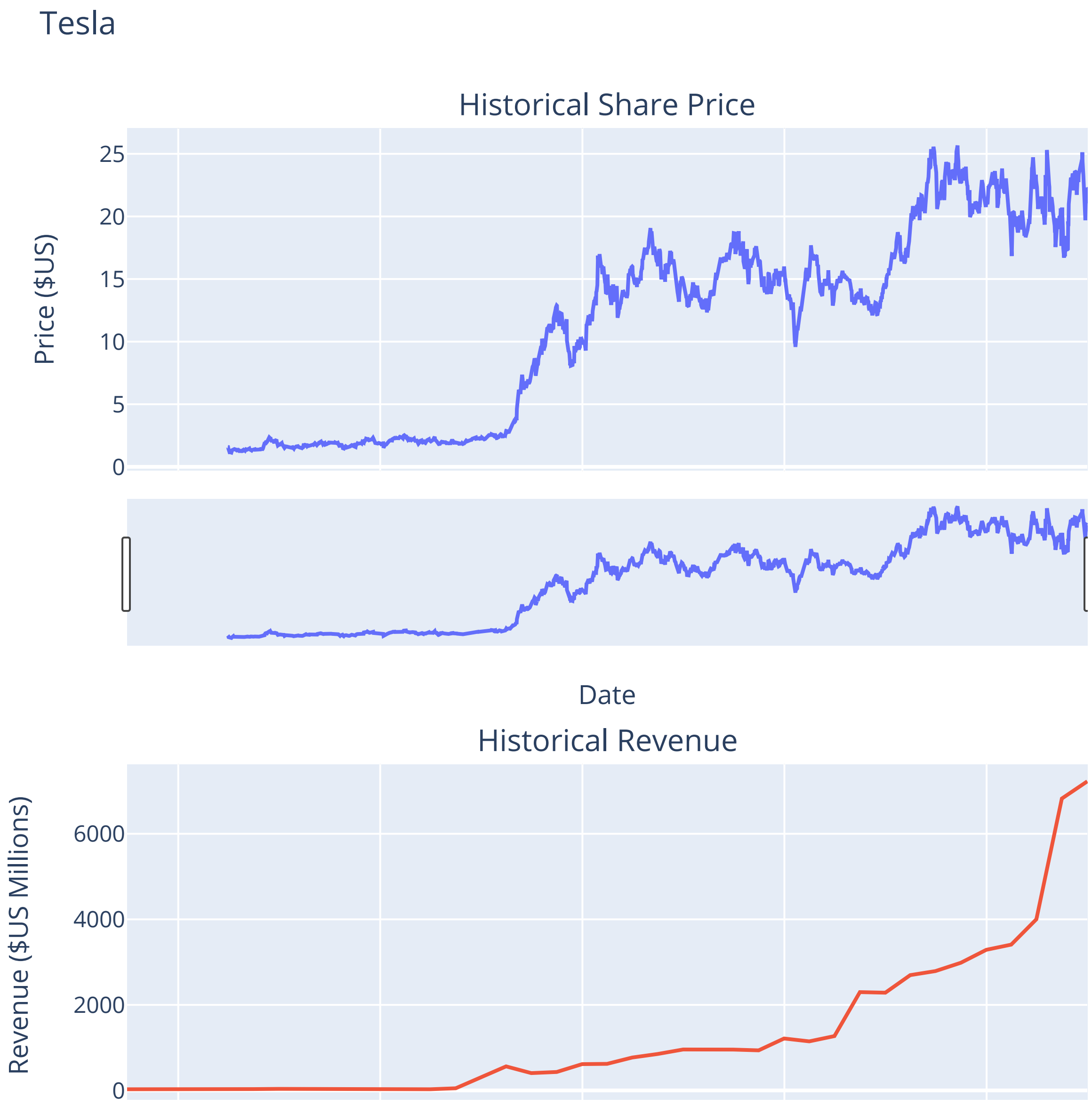
	Date	Revenue
57	2006-01-31	1667
58	2005-10-31	534
59	2005-07-31	416
60	2005-04-30	475
61	2005-01-31	709

• 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 2015. However, you can modify the defined function which we created at first hand on the top.

In [20]:

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

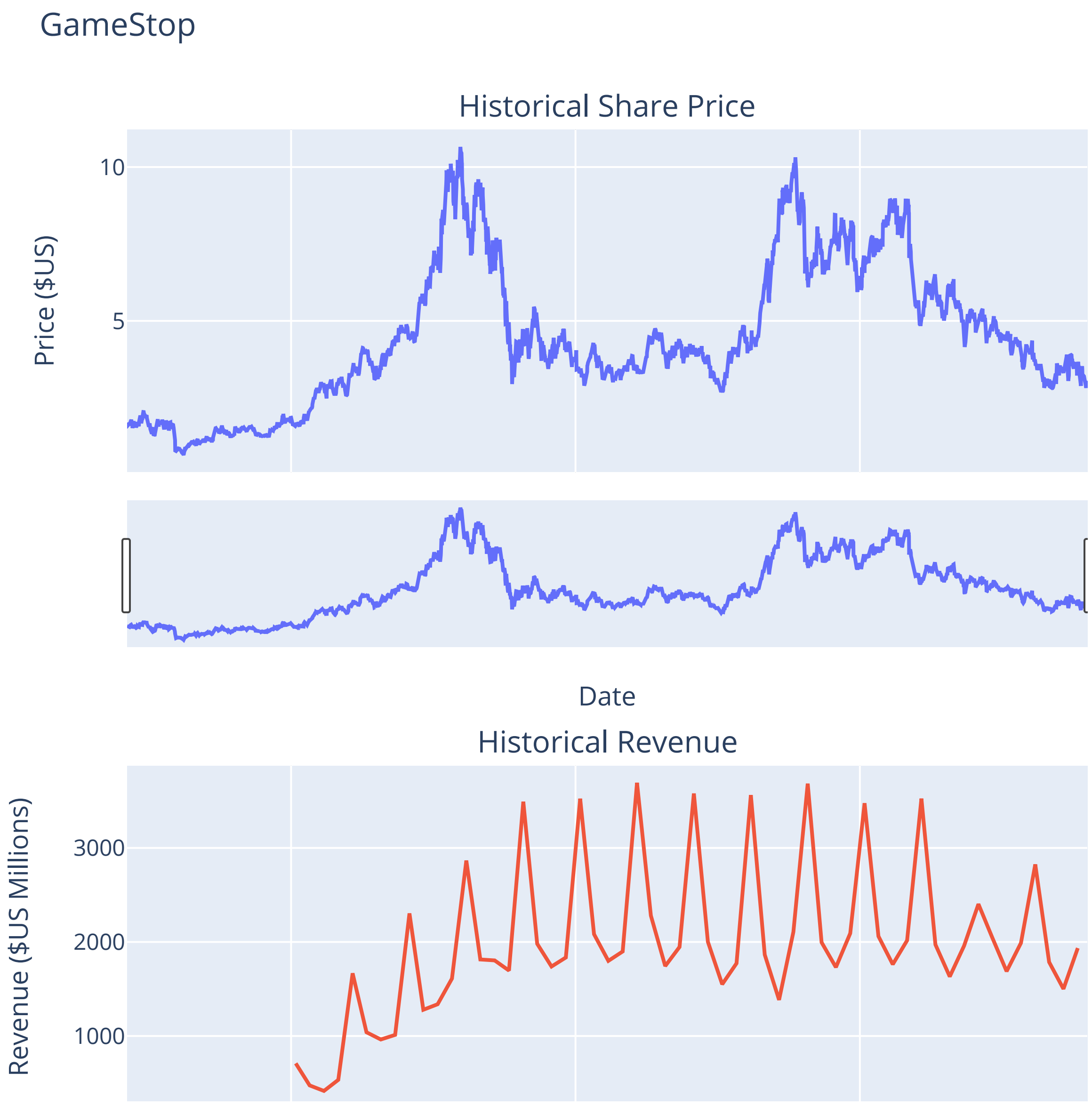


• Plot GameStop Stock Graph

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 2015. However, you can modify the defined function which we created at first hand on the top.

In [21]:

```
make_graph(gme_data, gme_revenue, "GameStop")
```


GameStop
Historical Share Price
Price ($US)
Date
Historical Revenue
Revenue ($US Millions)