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

Estimated Time Needed: **30 min**

In [1]:
```
!pip install yfinance==0.1.67
!pip install pandas==1.3.5
!pip install requests==2.26.0
!mamba install bs4==4.10.0 -y
!pip install plotly==5.3.1
```

```
Requirement already satisfied: yfinance==0.1.67 in d:\anaconda2023\lib\site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in d:\anaconda2023\lib\site-packages (from yfinance==0.1.67) (1.3.1)
Requirement already satisfied: multitasking>=0.0.7 in d:\anaconda2023\lib\site-packages (from yfinance==0.1.67) (0.
0.11)
Requirement already satisfied: requests>=2.20 in d:\anaconda2023\lib\site-packages (from yfinance==0.1.67) (2.26.0)
Requirement already satisfied: numpy>=1.15 in d:\anaconda2023\lib\site-packages (from yfinance==0.1.67) (1.21.5)
Requirement already satisfied: lxml>=4.5.1 in d:\anaconda2023\lib\site-packages (from yfinance==0.1.67) (4.9.1)
Requirement already satisfied: python-dateutil>=2.7.3 in d:\anaconda2023\lib\site-packages (from pandas>=0.24->yfina
nce==0.1.67) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in d:\anaconda2023\lib\site-packages (from pandas>=0.24->yfinance==0.1.6
7) (2022.1)
Requirement already satisfied: idna<4,>=2.5 in d:\anaconda2023\lib\site-packages (from requests>=2.20->yfinance==0.
1.67) (3.3)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda2023\lib\site-packages (from requests>=2.20->yfinanc
e==0.1.67) (2022.9.14)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in d:\anaconda2023\lib\site-packages (from requests>=2.20->yfin
ance==0.1.67) (1.26.11)
Requirement already satisfied: charset-normalizer~=2.0.0 in d:\anaconda2023\lib\site-packages (from requests>=2.20->
yfinance==0.1.67) (2.0.4)
Requirement already satisfied: six>=1.5 in d:\anaconda2023\lib\site-packages (from python-dateutil>=2.7.3->pandas>=
0.24->yfinance==0.1.67) (1.16.0)
Collecting pandas==1.3.5
  Using cached pandas-1.3.5-cp39-cp39-win_amd64.whl (10.2 MB)
Requirement already satisfied: python-dateutil>=2.7.3 in d:\anaconda2023\lib\site-packages (from pandas==1.3.5) (2.
8.2)
Requirement already satisfied: numpy>=1.17.3 in d:\anaconda2023\lib\site-packages (from pandas==1.3.5) (1.21.5)
Requirement already satisfied: pytz>=2017.3 in d:\anaconda2023\lib\site-packages (from pandas==1.3.5) (2022.1)
Requirement already satisfied: six>=1.5 in d:\anaconda2023\lib\site-packages (from python-dateutil>=2.7.3->pandas==
1.3.5) (1.16.0)
Installing collected packages: pandas
  Attempting uninstall: pandas
    Found existing installation: pandas 1.3.1
    Uninstalling pandas-1.3.1:
      Successfully uninstalled pandas-1.3.1
Successfully installed pandas-1.3.5
Requirement already satisfied: requests==2.26.0 in d:\anaconda2023\lib\site-packages (2.26.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in d:\anaconda2023\lib\site-packages (from requests==2.26.0)
(1.26.11)
Requirement already satisfied: charset-normalizer~=2.0.0 in d:\anaconda2023\lib\site-packages (from requests==2.26.
0) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in d:\anaconda2023\lib\site-packages (from requests==2.26.0) (202
```

```
2.9.14)
Requirement already satisfied: idna<4,>=2.5 in d:\anaconda2023\lib\site-packages (from requests==2.26.0) (3.3)

'mamba' is not recognized as an internal or external command,
operable program or batch file.

Requirement already satisfied: plotly==5.3.1 in d:\anaconda2023\lib\site-packages (5.3.1)
Requirement already satisfied: tenacity>=6.2.0 in d:\anaconda2023\lib\site-packages (from plotly==5.3.1) (8.0.1)
Requirement already satisfied: six in d:\anaconda2023\lib\site-packages (from plotly==5.3.1) (1.16.0)
```

In [2]:
```
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.

In [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 Rever
    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_spec
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_
    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()
```

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`.

In [4]:
```
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.

In [5]:
```
tesla_data = Tesla.history('max')
```

In [ ]:
```

```

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.

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

Out[6]:

	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://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue (https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2021-01-01). Save the text of the response as a variable named `html_data`.

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

Parse the html data using `beautiful_soup`.

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

Out[8]:
```
[<title>Tesla Revenue 2010-2023 | TSLA | MacroTrends</title>]
```

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

```
Below is the code to isolate the table, you will now need to loop through the rows and columns like in the pr
evious lab

soup.find_all("tbody")[1]

If you want to use the read_html function the table is located at index 1
```

In [14]:
```
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.replace("$", "").replace(",", "")

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

In [ ]:
```

```

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

In [15]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

```
C:\Users\kates\AppData\Local\Temp\ipykernel_15004\349343550.py:1: FutureWarning: The default value of regex will cha
nge from True to False in a future version.
  tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

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

In [16]:
```
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.

In [17]:
```
tesla_revenue.tail()
```

Out[17]:

	Date	Revenue
50	2010-09-30	31
51	2010-06-30	28
52	2010-03-31	21
54	2009-09-30	46
55	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` .

In [18]:
```
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.

In [19]:
```
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.

In [20]:
```
gme_data.reset_index(inplace = True)
gme_data.head()
```

Out[20]:

	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	1.712708	1.716074	1.670626	1.683251	11021600	0.0	0.0
2	2002-02-15	1.683250	1.687458	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://www.macrotrends.net/stocks/charts/GME/gamestop/revenue (https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2021-01-01). Save the text of the response as a variable named `html_data`.

In [21]:
```
url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
html_data = requests.get(url).text
```

Parse the html data using `beautiful_soup` .

In [22]:
```
soup = BeautifulSoup(html_data, "html.parser")
soup.find_all('title')
```

Out[22]:
```
[<title>GameStop Revenue 2010-2023 | GME | MacroTrends</title>]
```

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

```
Below is the code to isolate the table, you will now need to loop through the rows and columns like in the pr
evious lab

soup.find_all("tbody")[1]

If you want to use the read_html function the table is located at index 1
```

In [23]:
```
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.replace("$", "").replace(",", "")

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

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

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

Out[24]:

	Date	Revenue
52	2010-01-31	3524
53	2009-10-31	1835
54	2009-07-31	1739
55	2009-04-30	1981
56	2009-01-31	3492

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.

In [25]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla')
```

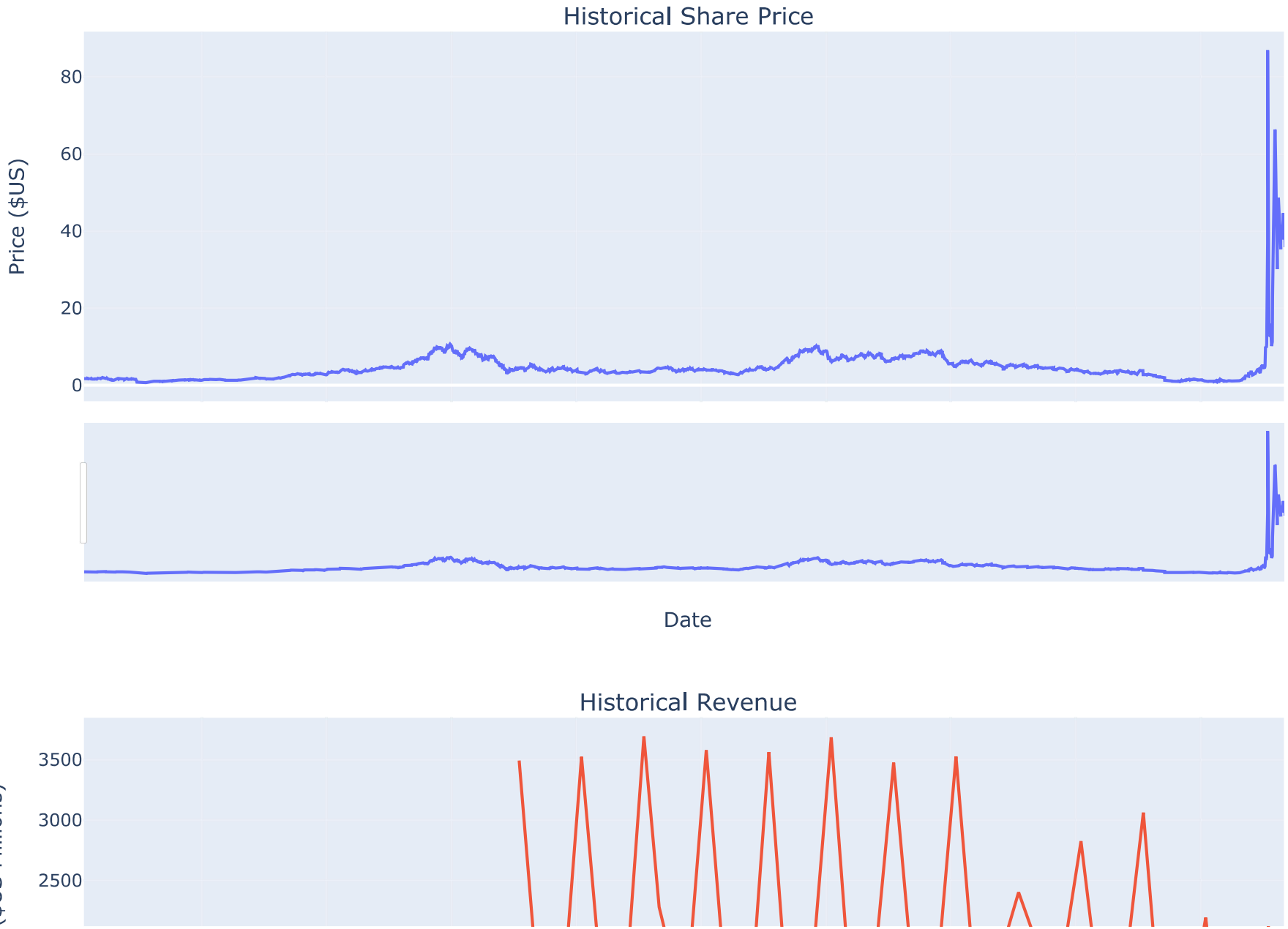

Tesla
Historical Share Price
Price ($US)
300
250
200
150
100
50
0
Date
Historical Revenue
($US Millions)
10k
8k
6k

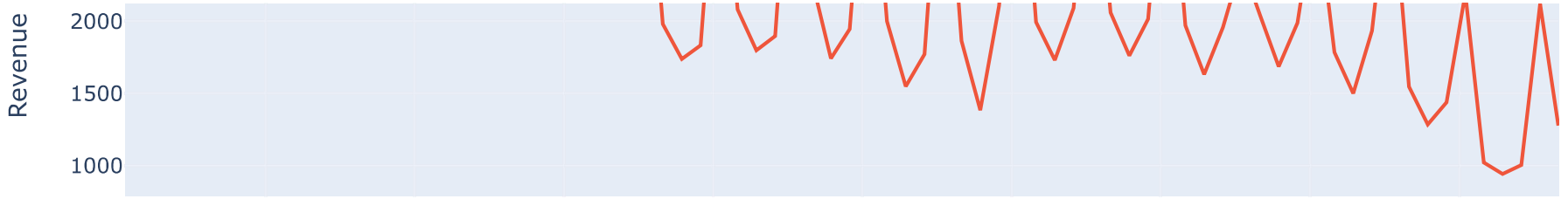
Question 6: Plot GameStop Stock Graph

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.

In [26]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

GameStop


Historical Share Price
Price ($US)
80
60
40
20
0
Date
Historical Revenue
($US Millions)
3500
3000
2500



About the Authors:

Joseph Santarcangelo (https://www.linkedin.com/in/joseph-s-50398b136/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2021-01-01) 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
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab

In [ ]:

In [ ]: