Extracting and Visualizing Stock Data

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, I will extract some stock data, then display this data in a graph.

Define a Function that Makes a Graph Question 1: Use yfinance to Extract Stock Data Question 2: Use Webscraping to Extract Tesla Revenue Data Question 3: Use yfinance to Extract Stock Data Question 4: Use Webscraping to Extract GME Revenue Data Question 5: Plot Tesla Stock Graph Question 6: Plot GameStop Stock Graph

In [2]:
```
!pip install yfinance==0.1.67
#!pip install pandas==1.3.3
#!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 c:\users\user\anaconda3\lib\site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in c:\users\user\anaconda3\lib\site-packages (from yfinance==0.1.67)
(1.2.4)
Requirement already satisfied: numpy>=1.15 in c:\users\user\anaconda3\lib\site-packages (from yfinance==0.1.67) (
1.20.1)
Requirement already satisfied: requests>=2.20 in c:\users\user\anaconda3\lib\site-packages (from yfinance==0.1.67
) (2.25.1)
Requirement already satisfied: multitasking>=0.0.7 in c:\users\user\anaconda3\lib\site-packages (from yfinance==0
.1.67) (0.0.10)
Requirement already satisfied: lxml>=4.5.1 in c:\users\user\anaconda3\lib\site-packages (from yfinance==0.1.67) (
4.6.3)
Requirement already satisfied: pytz>=2017.3 in c:\users\user\anaconda3\lib\site-packages (from pandas>=0.24->yfin
ance==0.1.67) (2021.1)
Requirement already satisfied: python-dateutil>=2.7.3 in c:\users\user\anaconda3\lib\site-packages (from pandas>=
0.24->yfinance==0.1.67) (2.8.1)
Requirement already satisfied: six>=1.5 in c:\users\user\anaconda3\lib\site-packages (from python-dateutil>=2.7.3
->pandas>=0.24->yfinance==0.1.67) (1.15.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\user\anaconda3\lib\site-packages (from requests>
=2.20->yfinance==0.1.67) (1.26.4)
Requirement already satisfied: idna<3,>=2.5 in c:\users\user\anaconda3\lib\site-packages (from requests>=2.20->yf
inance==0.1.67) (2.10)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\user\anaconda3\lib\site-packages (from requests>=2.2
0->yfinance==0.1.67) (4.0.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\user\anaconda3\lib\site-packages (from requests>=2.
20->yfinance==0.1.67) (2020.12.5)
```

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

In [6]:
```
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
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
<ipython-input-6-2804ab81232d> in <module>
      3 import requests
      4 from bs4 import BeautifulSoup
----> 5 import plotly.graph_objects as go
      6 from plotly.subplots import make_subplots
      7 import matplotlib

ModuleNotFoundError: No module named 'plotly'
```

In [17]:
```
import matplotlib
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

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

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 [11]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Historical Share Price", "Historical
    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
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_
    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.

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.

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 [15]:
```
tesla = yf.Ticker("TSLA")
tesla_data = tesla.history(period="max")
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

Out[15]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29	3.800	5.000	3.508	4.778	93831500	0	0.0
1	2010-06-30	5.158	6.084	4.660	4.766	85935500	0	0.0
2	2010-07-01	5.000	5.184	4.054	4.392	41094000	0	0.0
3	2010-07-02	4.600	4.620	3.742	3.840	25699000	0	0.0
4	2010-07-06	4.000	4.000	3.166	3.222	34334500	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?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.

Parse the html data using beautiful_soup. # 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. # remove the comma and dollar sign from the Revenue column. # remove an null or empty strings in the Revenue column. # Display the last 5 row of the tesla_revenue dataframe using the tail function. Take a screenshot of the results

In [26]:
```
import pandas as pd
import requests
from bs4 import BeautifulSoup
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue?utm_medium=Exinfluencer&utm_source=Exinfluenc

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

In [27]:
```
soup = BeautifulSoup(html_data, 'html5lib')
```

In [23]:
```
tables = pd.read_html(url)
tesla_revenue = tables[1].rename(columns = {"Tesla Quarterly Revenue(Millions of US $)": "Date", "Tesla Quarterly
```

In [24]:
```
tesla_revenue.dropna(inplace=True)
```

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

In [25]:
```
tesla_revenue.tail()
```

Out[25]:

	Date	Revenue
44	2010-09-30	$31
45	2010-06-30	$28
46	2010-03-31	$21
48	2009-09-30	$46
49	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.

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.

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 [40]:
```
GameStop = yf.Ticker("GME")
GameStop_data = gme_data
```

In [41]:
```
gme_data = GameStop.history(period="max")
```

In [42]:
```
GameStop_data.reset_index(inplace=True)
GameStop_data = gme_data
gme_data.head()
```

Out[42]:

Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
2002-02-13	6.480512	6.773398	6.413182	6.766665	19054000	0.0	0.0
2002-02-14	6.850828	6.864294	6.682503	6.733001	2755400	0.0	0.0
2002-02-15	6.733001	6.749833	6.632006	6.699336	2097400	0.0	0.0
2002-02-19	6.665672	6.665672	6.312189	6.430017	1852600	0.0	0.0
2002-02-20	6.463682	6.648839	6.413184	6.648839	1723200	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. Save the text of the response as a variable named html_data.

In [ ]:
```
# url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue?utm_medium=Exinfluencer&utm_source=Exinfl

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

Parse the html data using *beaut if _ soup*.

In [46]:
```
soup = BeautifulSoup(html_data, 'html5lib')
```

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.

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

In [47]:
```
gme_tables = pd.read_html(url)
gme_revenue = gme_tables[1].rename(columns = {"GameStop Quarterly Revenue(Millions of US $)":"Date", "GameStop Qu
```

In [48]:
```
gme_revenue.tail()
```

Out[48]:

	Date	Revenue
63	2006-01-31	$1,667
64	2005-10-31	$534
65	2005-07-31	$416
66	2005-04-30	$475
67	2005-01-31	$709

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 [5]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-5-2f887f7e5449> in <module>
----> 1 make_graph(tesla_data, tesla_revenue, 'Tesla')

NameError: name 'make_graph' is not defined
```

In [2]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-2-eff003b2809f> in <module>
----> 1 make_graph(gme_data, gme_revenue, 'GameStop')

NameError: name 'make_graph' is not defined
```