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
#!pip install pandas
#!pip install requests
!pip install bs4
#!pip install plotly
```

```
Collecting yfinance
  Downloading yfinance-0.1.74-py2.py3-none-any.whl (27 kB)
Requirement already satisfied: pandas>=0.24.0 in /home/jupyterlab/conda/envs/
python/lib/python3.7/site-packages (from yfinance) (1.3.5)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.7/site-packages (from yfinance) (4.9.1)
Collecting multitasking>=0.0.7
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.7/site-packages (from yfinance) (1.21.6)
Requirement already satisfied: requests>=2.26 in /home/jupyterlab/conda/envs/
python/lib/python3.7/site-packages (from yfinance) (2.28.1)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/con
da/envs/python/lib/python3.7/site-packages (from pandas>=0.24.0->yfinance)
(2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/py
thon/lib/python3.7/site-packages (from pandas>=0.24.0->yfinance) (2022.1)
Requirement already satisfied: charset-normalizer<3,>=2 in /home/jupyterlab/c
onda/envs/python/lib/python3.7/site-packages (from requests>=2.26->yfinance)
(2.1.0)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/e
nvs/python/lib/python3.7/site-packages (from requests>=2.26->yfinance) (2022.
6.15)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/cond
a/envs/python/lib/python3.7/site-packages (from requests>=2.26->yfinance) (1.
26.9)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/py
thon/lib/python3.7/site-packages (from requests>=2.26->yfinance) (3.3)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.24.0->y
finance) (1.16.0)
Installing collected packages: multitasking, yfinance
Successfully installed multitasking-0.0.11 yfinance-0.1.74
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: beautifulsoup4 in /home/jupyterlab/conda/envs/
python/lib/python3.7/site-packages (from bs4) (4.11.1)
Requirement already satisfied: soupsieve>1.2 in /home/jupyterlab/conda/envs/p
ython/lib/python3.7/site-packages (from beautifulsoup4->bs4) (2.3.1)
Building wheels for collected packages: bs4
  Building wheel for bs4 (setup.py) ... done
  Created wheel for bs4: filename=bs4-0.0.1-py3-none-any.whl size=1257 sha256
=0e8cac0286119dd5afff817a33101ebf1473c75fcac1295ad266c4493fd1e71f
  Stored in directory: /home/jupyterlab/.cache/pip/wheels/0a/9e/ba/20e5bbc1af
ef3a491f0b3bb74d508f99403aabe76eda2167ca
Successfully built bs4
Installing collected packages: bs4
Successfully installed bs4-0.0.1
```

In [59]:
```
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 [60]:
```
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, infer_datetime_format=True), 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, infer_datetime_format=True), 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()
```

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 [61]:
```
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 [62]:
```
tesla_data = tesla.history(period="max")
```

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 [63]:
```
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

Out[63]:

	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 (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-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2022-01-01). Save the text of the response as a variable named `html_data`.

In [64]:
```
url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
html_data  = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

In [65]:
```
soup = BeautifulSoup(html_data)
```

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

In [66]:
```
read_html_pandas_data = pd.read_html(url)
tesla_revenue_dataframe = read_html_pandas_data[1]
```

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

In [67]:
```
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.
```

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

In [68]:
```
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 [69]:
```
tesla_revenue_dataframe.tail()
```

Out[69]:

	Tesla Quarterly Revenue(Millions of US $)	Tesla Quarterly Revenue(Millions of US $).1
47	2010-06-30	$28
48	2010-03-31	$21
49	2009-12-31	NaN
50	2009-09-30	$46
51	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 [70]:
```
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 [71]:
```
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 [72]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

Out[72]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13	6.480515	6.770034	6.430017	6.766667	19054000	0.0	0.0
1	2002-02-14	6.850829	6.850829	6.699336	6.733001	2801400	0.0	0.0
2	2002-02-15	6.733000	6.749832	6.632005	6.699335	2097400	0.0	0.0
3	2002-02-19	6.665670	6.665670	6.312188	6.430016	1852600	0.0	0.0
4	2002-02-20	6.463683	6.648840	6.413185	6.648840	1683200	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 (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`.

In [73]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html"
html_data  = requests.get(url).text
```

Parse the html data using `beautiful_soup`.

In [74]:
```
soup = BeautifulSoup(html_data)
```

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

In [75]:
```
read_html_pandas_data = pd.read_html(url)
gme_revenue_dataframe = read_html_pandas_data[1]
gme_revenue_dataframe.head()
```

Out[75]:

	GameStop Quarterly Revenue(Millions of US $)	GameStop Quarterly Revenue(Millions of US $).1
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

In [76]:
```
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.
```

In [77]:
```
gme_revenue.dropna(inplace=True)

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

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

In [78]:
```
gme_revenue_dataframe.tail()
```

Out[78]:

	GameStop Quarterly Revenue(Millions of US $)	GameStop Quarterly Revenue(Millions of US $).1
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

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 [79]:
```
make_graph(tesla_data, tesla_revenue, 'Tesla')
```

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 [80]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

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-SkillsNetworkCoursesIBMDeveloperSkillsNetworkPY0220ENSkillsNetwork23455606-2022-01-01) 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 Hiriani

Change Log

Date (YYYY-MM-DD)	Version	Changed By	Change Description
2022-02-28	1.2	Lakshmi Holla	Changed the URL of GameStop
2020-11-10	1.1	Malika Singla	Deleted the Optional part
2020-08-27	1.0	Malika Singla	Added lab to GitLab