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.

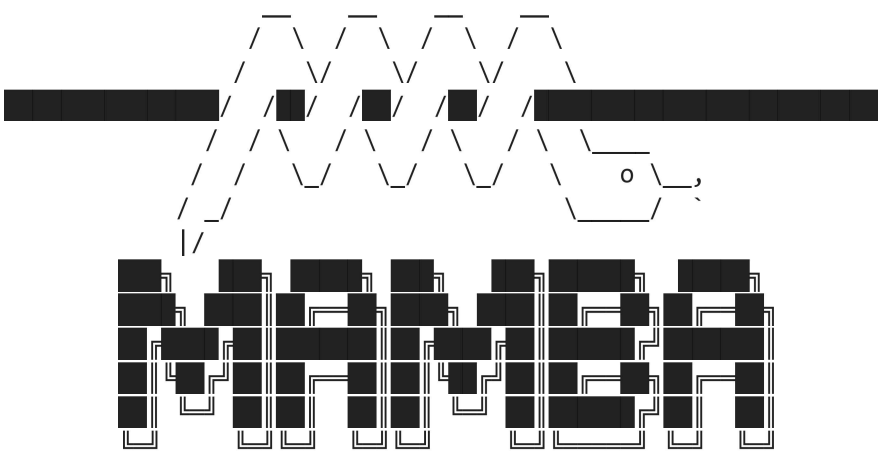
Table of Contents

- 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

Estimated Time Needed: **30 min**

In [2]:
```
!pip install yfinance==0.1.67
!mamba install bs4==4.10.0 -y
!pip install nbformat==4.2.0
```

```
Requirement already satisfied: yfinance==0.1.67 in /home/jupyterlab/conda/envs/
python/lib/python3.7/site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from yfinance==0.1.67) (1.3.5)
Requirement already satisfied: requests>=2.20 in /home/jupyterlab/conda/envs/py
thon/lib/python3.7/site-packages (from yfinance==0.1.67) (2.28.1)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.7/site-packages (from yfinance==0.1.67) (4.6.4)
Requirement already satisfied: multitasking>=0.0.7 in /home/jupyterlab/conda/en
vs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (0.0.11)
Requirement already satisfied: numpy>=1.15 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.7/site-packages (from yfinance==0.1.67) (1.21.6)
Requirement already satisfied: python-dateutil>=2.7.3 in /home/jupyterlab/cond
a/envs/python/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67)
(2.8.2)
Requirement already satisfied: pytz>=2017.3 in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from pandas>=0.24->yfinance==0.1.67) (2022.6)
Requirement already satisfied: charset-normalizer<3,>=2 in /home/jupyterlab/con
da/envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.
67) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in /home/jupyterlab/conda/env
s/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (2
022.9.24)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in /home/jupyterlab/conda/
envs/python/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67)
(1.26.13)
Requirement already satisfied: idna<4,>=2.5 in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from requests>=2.20->yfinance==0.1.67) (3.4)
Requirement already satisfied: six>=1.5 in /home/jupyterlab/conda/envs/python/l
ib/python3.7/site-packages (from python-dateutil>=2.7.3->pandas>=0.24->yfinance
==0.1.67) (1.16.0)
```



```
mamba (0.15.3) supported by @QuantStack

GitHub:  https://github.com/mamba-org/mamba
Twitter: https://twitter.com/QuantStack
```

```
Looking for: ['bs4==4.10.0']

pkgs/main/linux-64       [>                   ] (--:--) No change
pkgs/main/linux-64       [====================] (00m:00s) No change
pkgs/r/noarch            [>                   ] (--:--) No change
```

```
pkgs/r/noarch           [====================] (00m:00s) No change
pkgs/main/noarch        [>                   ] (--:--) No change
pkgs/main/noarch        [====================] (00m:00s) No change
pkgs/r/linux-64         [>                   ] (--:--) No change
pkgs/r/linux-64         [====================] (00m:00s) No change

Pinned packages:
  - python 3.7.*


Transaction

  Prefix: /home/jupyterlab/conda/envs/python

  All requested packages already installed

Collecting nbformat==4.2.0
  Downloading nbformat-4.2.0-py2.py3-none-any.whl (153 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 153.3/153.3 kB 18.8 MB/s eta 0:00:0
0
Requirement already satisfied: jupyter-core in /home/jupyterlab/conda/envs/pyth
on/lib/python3.7/site-packages (from nbformat==4.2.0) (4.12.0)
Requirement already satisfied: traitlets>=4.1 in /home/jupyterlab/conda/envs/py
thon/lib/python3.7/site-packages (from nbformat==4.2.0) (5.6.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /home/jupyterlab/cond
a/envs/python/lib/python3.7/site-packages (from nbformat==4.2.0) (4.17.3)
Requirement already satisfied: ipython-genutils in /home/jupyterlab/conda/envs/
python/lib/python3.7/site-packages (from nbformat==4.2.0) (0.2.0)
Requirement already satisfied: importlib-resources>=1.4.0 in /home/jupyterlab/c
onda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbf
ormat==4.2.0) (5.10.1)
Requirement already satisfied: attrs>=17.4.0 in /home/jupyterlab/conda/envs/pyt
hon/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==4.2.0)
(22.1.0)
Requirement already satisfied: typing-extensions in /home/jupyterlab/conda/env
s/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==
4.2.0) (4.4.0)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10 in /home/jupyterla
b/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->
nbformat==4.2.0) (1.3.10)
Requirement already satisfied: importlib-metadata in /home/jupyterlab/conda/env
s/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbformat==
4.2.0) (4.11.4)
Requirement already satisfied: pyrsistent!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschem
a!=2.5.0,>=2.4->nbformat==4.2.0) (0.19.2)
Requirement already satisfied: zipp>=3.1.0 in /home/jupyterlab/conda/envs/pytho
n/lib/python3.7/site-packages (from importlib-resources>=1.4.0->jsonschema!=2.
5.0,>=2.4->nbformat==4.2.0) (3.11.0)
Installing collected packages: nbformat
  Attempting uninstall: nbformat
    Found existing installation: nbformat 5.7.0
    Uninstalling nbformat-5.7.0:
      Successfully uninstalled nbformat-5.7.0
ERROR: pip's dependency resolver does not currently take into account all the p
ackages that are installed. This behaviour is the source of the following depen
dency conflicts.
nbconvert 7.2.6 requires nbformat>=5.1, but you have nbformat 4.2.0 which is in
compatible.
nbclient 0.7.2 requires nbformat>=5.1, but you have nbformat 4.2.0 which is inc
```

```
ompatible.
jupyter-server 1.23.3 requires nbformat>=5.2.0, but you have nbformat 4.2.0 whi
ch is incompatible.
Successfully installed nbformat-4.2.0
```

In [28]:
```
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 [29]:
```
def make_graph(stock_data, revenue_data, stock):
    fig = make_subplots(rows=2, cols=1, shared_xaxes=True, subplot_titles=("Hist
    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_da
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_
    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 [30]:
```
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 [31]:
```
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 [32]:
```
tesla_data.reset_index(inplace = True)
tesla_data.head
```

Out[32]:
```
<bound method NDFrame.head of            Date        Open        High         L
ow       Close     Volume  \
0    2010-06-29    1.266667    1.666667    1.169333    1.592667  281494500
1    2010-06-30    1.719333    2.028000    1.553333    1.588667  257806500
2    2010-07-01    1.666667    1.728000    1.351333    1.464000  123282000
3    2010-07-02    1.533333    1.540000    1.247333    1.280000   77097000
4    2010-07-06    1.333333    1.333333    1.055333    1.074000  103003500
...         ...         ...         ...         ...         ...         ...
3132 2022-12-06  181.220001  183.649994  175.330002  179.820007   92150800
3133 2022-12-07  175.029999  179.380005  172.220001  174.039993   84213300
3134 2022-12-08  172.199997  175.199997  169.059998  173.440002   97624500
3135 2022-12-09  173.839996  182.500000  173.360001  179.050003  104746600
3136 2022-12-12  176.100006  177.359894  170.080002  171.669998   54220426

      Dividends  Stock Splits
0             0           0.0
1             0           0.0
2             0           0.0
3             0           0.0
4             0           0.0
...         ...           ...
3132          0           0.0
3133          0           0.0
3134          0           0.0
3135          0           0.0
3136          0           0.0

[3137 rows x 8 columns]>
```

Question 2: Use Webscraping to Extract Tesla 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/revenue.htm Save the text of the response as a variable named `html_data`.

In [33]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDev

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

Parse the html data using `beautiful_soup`.

In [34]:
```
soup = BeautifulSoup(html_data, 'html5lib')
```

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 [35]:
```
tesla_revenue = pd.DataFrame(columns=["Date", "Revenue"])

for row in soup.find("tbody").find_all("tr"):
    col = row.find_all("td")
    date = col[0].text
    Revenue = col[1].text

    tesla_revenue = tesla_revenue.append({"Date":date, "Revenue":Revenue}, ignor
```

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

In [36]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launch
er.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 [37]:
```
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 [38]:
```
tesla_revenue.tail
```

Out[38]:
```
<bound method NDFrame.tail of     Date Revenue
0   2021   53823
1   2020   31536
2   2019   24578
3   2018   21461
4   2017   11759
5   2016    7000
6   2015    4046
7   2014    3198
8   2013    2013
9   2012     413
10  2011     204
11  2010     117
12  2009     112>
```

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 [39]:
```
gme = 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 [40]:
```
gme_data = gme.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 [41]:
```
gme_data.reset_index(inplace = True)
gme_data.head
```

Out[41]:
```
<bound method NDFrame.head of            Date       Open       High        Low
Close    Volume  \
0    2002-02-13   1.620129   1.693350   1.603296   1.691667  76216000
1    2002-02-14   1.712707   1.716074   1.670626   1.683250  11021600
2    2002-02-15   1.683250   1.687458   1.658001   1.674834   8389600
3    2002-02-19   1.666418   1.666418   1.578047   1.607504   7410400
4    2002-02-20   1.615921   1.662210   1.603296   1.662210   6892800
...         ...        ...        ...        ...        ...       ...
5240 2022-12-06  25.410000  25.580000  23.110001  23.389999   7699300
5241 2022-12-07  23.400000  23.610001  21.969999  22.260000  10078200
5242 2022-12-08  22.000000  25.000000  21.969999  24.790001  10252000
5243 2022-12-09  24.590000  24.590000  22.590000  22.629999   5312900
5244 2022-12-12  22.660000  23.135000  22.180000  22.959999   2284213

      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
...         ...           ...
5240        0.0           0.0
5241        0.0           0.0
5242        0.0           0.0
5243        0.0           0.0
5244        0.0           0.0

[5245 rows x 8 columns]>
```

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. Save the text of the response as a variable named `html_data`.

In [42]:
```
url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDev

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

Parse the html data using `beautiful_soup`.

In [43]:
```
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.

▶ Click here if you need help locating the table

In [44]:
```
gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])

for row in soup.find("tbody").find_all("tr"):
    col = row.find_all("td")
    date = col[0].text
    Revenue = col[1].text

    gme_revenue = gme_revenue.append({"Date":date, "Revenue":Revenue}, ignore_ir
gme_revenue["Revenue"] = gme_revenue['Revenue'].str.replace(',|\$',"")
gme_revenue.dropna(inplace=True)

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

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages/ipykernel_launch
er.py:9: FutureWarning:

The default value of regex will change from True to False in a future version.
```

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

In [45]:
```
gme_revenue.tail
```

Out[45]:
```
<bound method NDFrame.tail of     Date Revenue
0   2020    6466
1   2019    8285
2   2018    8547
3   2017    7965
4   2016    9364
5   2015    9296
6   2014    9040
7   2013    8887
8   2012    9551
9   2011    9474
10  2010    9078
11  2009    8806
12  2008    7094
13  2007    5319
14  2006    3092
15  2005    1843>
```

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 [46]:
```
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 [47]:
```
make_graph(gme_data, gme_revenue, 'GameStop')
```

About the Authors:

Joseph Santarcangelo 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
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

In [ ]: