Final Assignment

January 21, 2023

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

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

Estimated Time Needed: 30 min

[1]:

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

```
Requirement already satisfied: yfinance 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/python/lib/python3.7/site-packages (from yfinance)
(1.3.5)
Requirement already satisfied: requests>=2.20 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance)
(2.28.1)
Requirement already satisfied: lxml>=4.5.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance)
(4.6.4)
Requirement already satisfied: multitasking>=0.0.7 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance)
```

```
(0.0.11)
Requirement already satisfied: numpy>=1.15 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance)
(1.21.6)
Requirement already satisfied: python-dateutil>=2.7.3 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
pandas>=0.24->yfinance) (2.8.2)
Requirement already satisfied: pytz>=2017.3 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
pandas>=0.24->yfinance) (2022.6)
Requirement already satisfied: charset-normalizer<3,>=2 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance) (2.1.1)
Requirement already satisfied: certifi>=2017.4.17 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance) (2022.12.7)
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) (1.26.13)
Requirement already satisfied: idna<4,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance) (3.4)
Requirement already satisfied: six>=1.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from python-
dateutil>=2.7.3->pandas>=0.24->yfinance) (1.16.0)
Requirement already satisfied: bs4 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.0.1)
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/python/lib/python3.7/site-packages (from
beautifulsoup4->bs4) (2.3.2.post1)
```

[2]:

```python
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

[3]:

```python
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)
```

```
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date,␣
↪infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Share␣
↪Price"), row=1, col=1)
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date,␣
↪infer_datetime_format=True), y=revenue_data.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

[4]:
```
Tesla = yf.Ticker('TSLA')
```

[5]:
```
tesla_data = Tesla.history(period ="max")
```

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

[6]:
```
        Date      Open      High       Low     Close     Volume  Dividends  \
0 2010-06-29  1.266667  1.666667  1.169333  1.592667  281494500          0
1 2010-06-30  1.719333  2.028000  1.553333  1.588667  257806500          0
2 2010-07-01  1.666667  1.728000  1.351333  1.464000  123282000          0
3 2010-07-02  1.533333  1.540000  1.247333  1.280000   77097000          0
4 2010-07-06  1.333333  1.333333  1.055333  1.074000  103003500          0

   Stock Splits
0           0.0
1           0.0
2           0.0
3           0.0
4           0.0
```

Question 2: Use Webscraping to Extract Tesla Revenue Data

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

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

```
[8]: [<title>Tesla Revenue 2010-2022 | TSLA | MacroTrends</title>]
```

```
[9]: 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)
```

```
[10]: tesla_revenue.dropna(inplace=True)
      tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
```

```
[11]: tesla_revenue.tail()
```

```
[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
```

Question 3: Use yfinance to Extract Stock Data

```
[12]: GameStop = yf.Ticker("GME")
```

```
[13]: gme_data = GameStop.history(period = 'max')
```

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

```
[14]:         Date      Open      High       Low     Close    Volume  Dividends  \
      0 2002-02-13  1.620128  1.693350  1.603296  1.691666  76216000        0.0
      1 2002-02-14  1.712707  1.716074  1.670626  1.683250  11021600        0.0
      2 2002-02-15  1.683251  1.687459  1.658002  1.674834   8389600        0.0
      3 2002-02-19  1.666418  1.666418  1.578047  1.607504   7410400        0.0
      4 2002-02-20  1.615920  1.662210  1.603296  1.662210   6892800        0.0

         Stock Splits
      0           0.0
      1           0.0
      2           0.0
      3           0.0
      4           0.0
```

Question 4: Use Webscraping to Extract tesla Revenue Data

[29]:

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

[21]:

```
soup = BeautifulSoup(html_data, "html.parser")
soup.find_all('title')
```

[21]:

```
[<title>GameStop Revenue 2006-2020 | GME | MacroTrends</title>]
```

[22]:

```
me_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)
```

[32]:

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

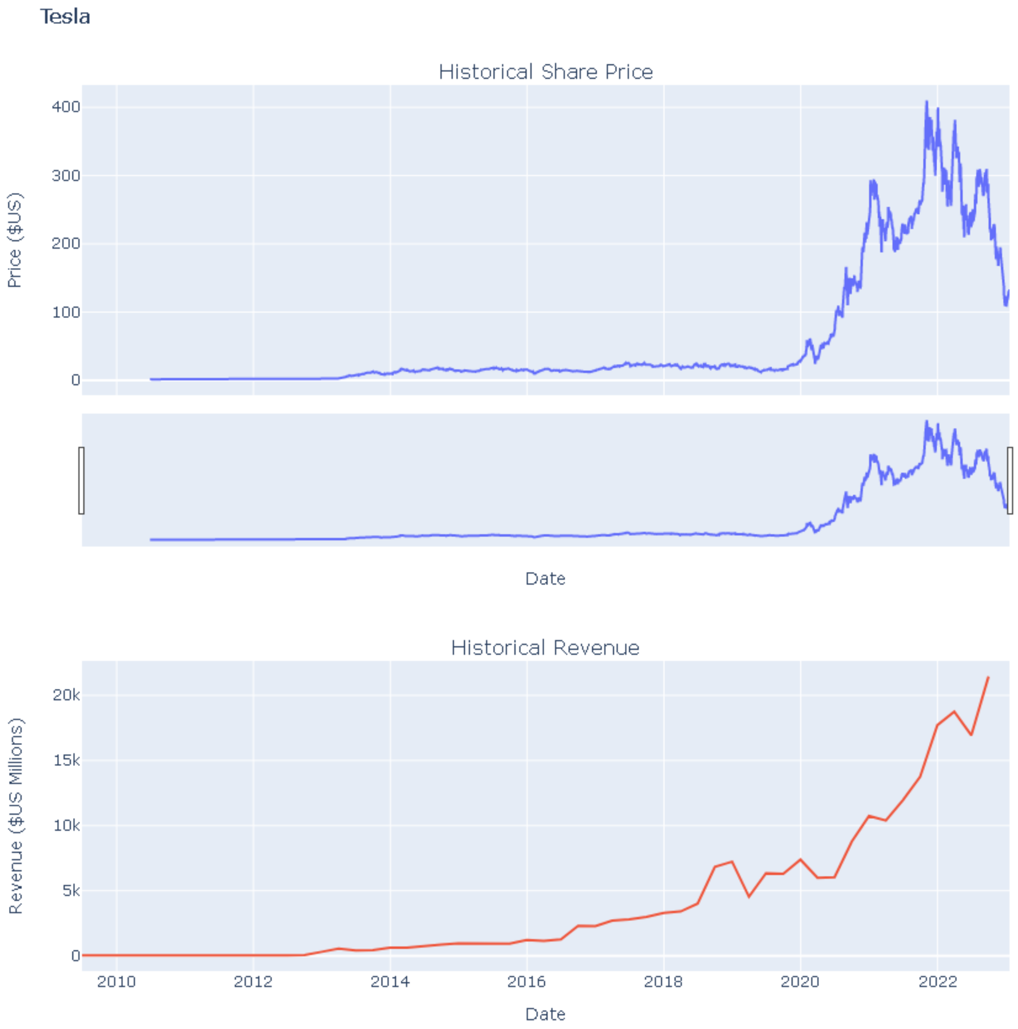
[32]:

```
           Date Revenue
113  2006-01-31    1667
114  2005-10-31     534
115  2005-07-31     416
116  2005-04-30     475
117  2005-01-31     709
```

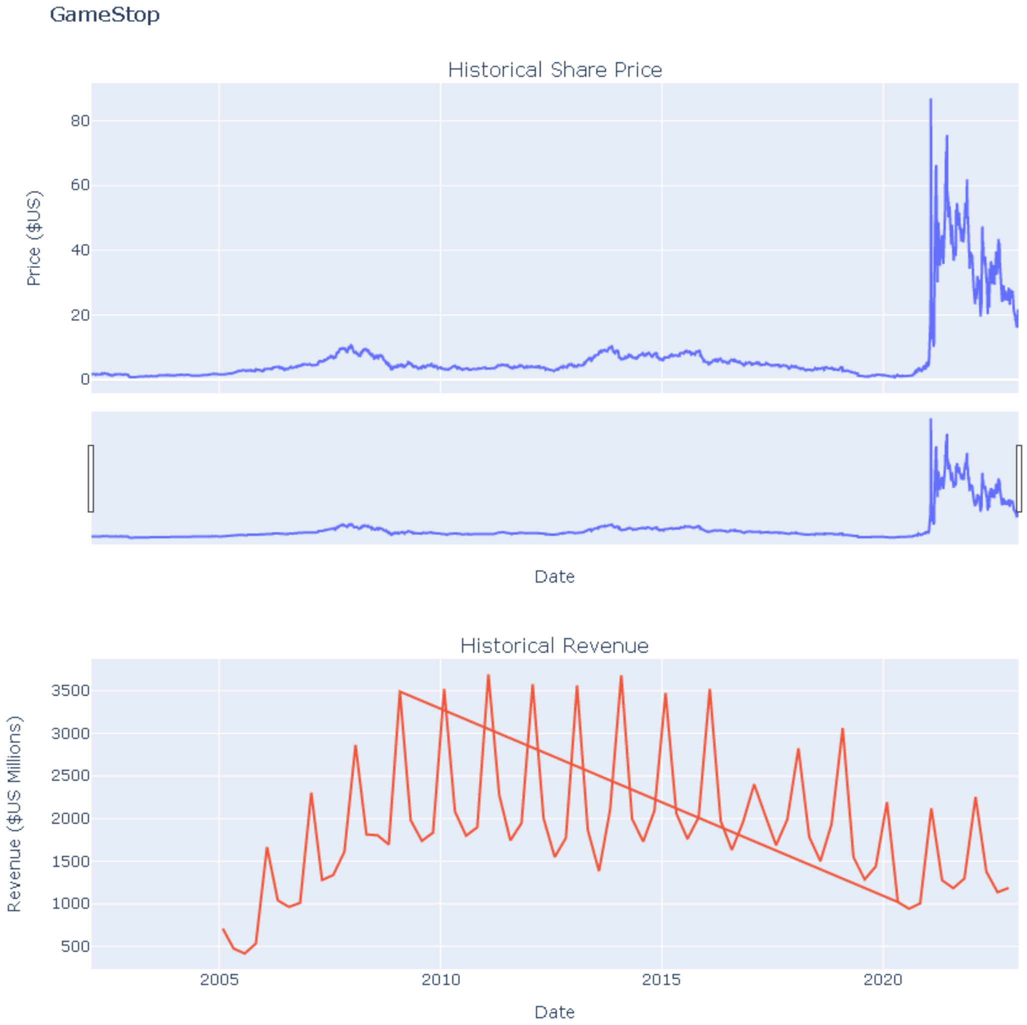
plat tesla stock graph

[24]:

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



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


GameStop
Historical Share Price
Price ($US)
80
60
40
20
0
Date
Historical Revenue
Revenue ($US Millions)
3500
3000
2500
2000
1500
1000
500
2005
2010
2015
2020
Date