Question

May 6, 2022

0.1 Question 1 - Extracting Tesla Stock Data Using yfinance - 2 Points

```
[39]: !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
      !mamba install html5lib==1.1 -y
      !pip install lxml==4.6.4
      #!pip install plotly==5.3.1

      import yfinance as yf
      import pandas as pd
      from bs4 import BeautifulSoup
```
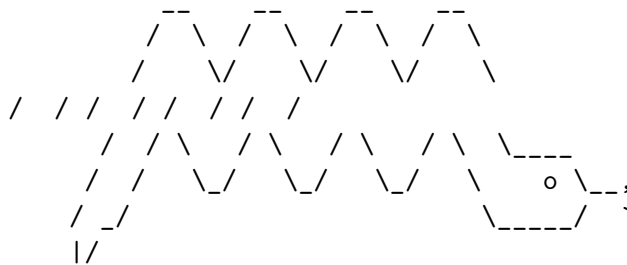
```
Requirement already satisfied: yfinance==0.1.67 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (0.1.67)
Requirement already satisfied: pandas>=0.24 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (1.3.5)
Requirement already satisfied: requests>=2.20 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (2.27.1)
Requirement already satisfied: lxml>=4.5.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (4.6.4)
Requirement already satisfied: multitasking>=0.0.7 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (0.0.10)
Requirement already satisfied: numpy>=1.15 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (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==0.1.67) (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==0.1.67) (2022.1)
Requirement already satisfied: certifi>=2017.4.17 in
```

```
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance==0.1.67) (2021.10.8)
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) (1.26.9)
Requirement already satisfied: idna<4,>=2.5 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance==0.1.67) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance==0.1.67) (2.0.12)
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==0.1.67) (1.16.0)


                      __    __    __    __
                     /  \  /  \  /  \  /  \
                    /    \/    \/    \/    \
         / /   / /   / /   /
             /   / \   / \   / \   / \  \____
            /   /    \_/    \_/    \_/    \    o \__,
           /  _/                            \____/  `
          |/

           mamba (0.22.1) supported by @QuantStack

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



Looking for: ['bs4==4.10.0']

[+] 0.0s
pkgs/main/linux-64                      0.0 B /  ??.?MB
@  ??.?MB/s  0.0s[+] 0.1s
pkgs/main/linux-64                      0.0 B /  ??.?MB
@  ??.?MB/s  0.1s
pkgs/main/noarch                        0.0 B /  ??.?MB
@  ??.?MB/s  0.1s
pkgs/r/linux-64                         0.0 B /  ??.?MB
```

```
@  ??.?MB/s  0.1s
pkgs/r/noarch                        0.0 B /  ??.?MB
@  ??.?MB/s  0.1s[+] 0.2s
pkgs/main/linux-64                   0.0 B /  ??.?MB
@  ??.?MB/s  0.2s
pkgs/main/noarch                     0.0 B /  ??.?MB
@  ??.?MB/s  0.2s
pkgs/r/linux-64                      0.0 B /  ??.?MB
@  ??.?MB/s  0.2s
pkgs/r/noarch                        0.0 B /  ??.?MB
@  ??.?MB/s  0.2spkgs/main/noarch
No change
pkgs/r/linux-64                                              No change
pkgs/r/noarch                                                No change
[+] 0.3s
pkgs/main/linux-64             382.0 B /  ??.?MB
@   1.5kB/s  0.3s[+] 0.4s
pkgs/main/linux-64             847.9kB /  ??.?MB
@   2.4MB/s  0.4s[+] 0.5s
pkgs/main/linux-64                   1.3MB
/  ??.?MB @   2.9MB/s  0.5s[+] 0.6s
pkgs/main/linux-64                   1.9MB
/  ??.?MB @   3.3MB/s  0.6s[+] 0.7s
pkgs/main/linux-64                   2.3MB
/  ??.?MB @   3.5MB/s  0.7s[+] 0.8s
pkgs/main/linux-64                   2.8MB
/  ??.?MB @   3.7MB/s  0.8s[+] 0.9s
pkgs/main/linux-64             3.3MB /  ??.?MB
@   3.8MB/s  0.9s[+] 1.0s
pkgs/main/linux-64             3.9MB /  ??.?MB
@   4.0MB/s  1.0s[+] 1.1s
pkgs/main/linux-64             4.4MB /  ??.?MB
@   4.1MB/s  1.1s[+] 1.2s
pkgs/main/linux-64             4.5MB @   4.1MB/s Finalizing
1.2s[+] 1.3s
pkgs/main/linux-64                              @   4.1MB/s
1.2s

Pinned packages:
  - python 3.7.*


Transaction

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

  All requested packages already installed
```
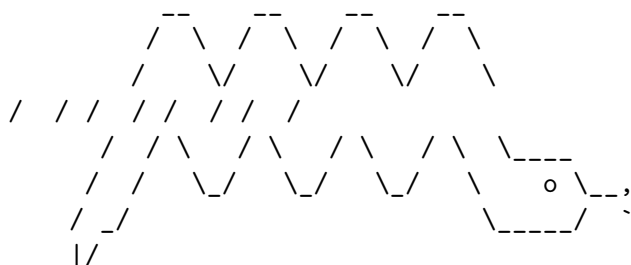
```
                __     __     __     __
               /  \   /  \   /  \   /  \
              /    \_/    \_/    \_/    \
             / / / / / / / /
                /  / \   / \   / \   / \  \____
               /  /   \_/   \_/   \_/   \    o \__,
              / _/                       \_____/  `
             |/
```

mamba (0.22.1) supported by @QuantStack

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

Looking for: ['html5lib==1.1']

pkgs/main/linux-64   Using cache
pkgs/main/noarch   Using cache
pkgs/r/linux-64   Using cache
pkgs/r/noarch   Using cache

Pinned packages:
  - python 3.7.*

Transaction

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

  All requested packages already installed

Requirement already satisfied: lxml==4.6.4 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (4.6.4)

[14]: tesla = yf.Ticker("TSLA")

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

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

```
[31]:         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
```

0.2 Question 2: Use Webscraping to Extract Tesla Revenue Data

```
[47]: pip install html5lib
```

```
Requirement already satisfied: html5lib in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (1.1)
Requirement already satisfied: webencodings in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib)
(0.5.1)
Requirement already satisfied: six>=1.9 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from html5lib)
(1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

```
[49]: from IPython.core.display import HTML
      HTML("<script>Jupyter.notebook.kernel.restart()</script>")
```

```
[49]: <IPython.core.display.HTML object>
```

```
[50]: tesla_url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
      tesla_html_data = requests.get(tesla_url).text
```

```
[53]: tesla_soup = BeautifulSoup(tesla_html_data, "lxml")
```

```
[54]: tesla_tables = tesla_soup.find_all('table')

      for index,table in enumerate(tesla_tables):
          if ("Tesla Quarterly Revenue" in str(table)):
              tesla_table_index = index

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

      for row in tesla_tables[tesla_table_index].tbody.find_all("tr"):
          col = row.find_all("td")
          if (col !=[]):
              date = col[0].text
              revenue = col[1].text.replace("$", "").replace(",", "")
```

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

[55]:
```
tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
tesla_revenue
```

[55]:
```
         Date  Revenue
0   2022-03-31   18756
1   2021-12-31   15339
2   2021-09-30   13757
3   2021-06-30   11958
4   2021-03-31   10389
5   2020-12-31    9034
6   2020-09-30    8771
7   2020-06-30    6036
8   2020-03-31    5985
9   2019-12-31    6143
10  2019-09-30    6303
11  2019-06-30    6350
12  2019-03-31    4541
13  2018-12-31    6074
14  2018-09-30    6824
15  2018-06-30    4002
16  2018-03-31    3409
17  2017-12-31    2409
18  2017-09-30    2985
19  2017-06-30    2790
20  2017-03-31    2696
21  2016-12-31    1739
22  2016-09-30    2298
23  2016-06-30    1270
24  2016-03-31    1147
25  2015-12-31    1214
26  2015-09-30     937
27  2015-06-30     955
28  2015-03-31     940
29  2014-12-31     957
30  2014-09-30     852
31  2014-06-30     769
32  2014-03-31     621
33  2013-12-31     615
34  2013-09-30     431
35  2013-06-30     405
36  2013-03-31     562
37  2012-12-31     306
38  2012-09-30      50
39  2012-06-30      27
```

```
40   2012-03-31        30
41   2011-12-31        39
42   2011-09-30        58
43   2011-06-30        58
44   2011-03-31        49
45   2010-12-31        36
46   2010-09-30        31
47   2010-06-30        28
48   2010-03-31        21
50   2009-09-30        46
51   2009-06-30        27
```

[56]: `tesla_revenue.tail()`

[56]:
```
         Date Revenue
46   2010-09-30      31
47   2010-06-30      28
48   2010-03-31      21
50   2009-09-30      46
51   2009-06-30      27
```

0.3 Question 3: Use yfinance to Extract Stock Data

[57]: `gamestop = yf.Ticker("GME")`

[58]: `gme_data = gamestop.history(period="max")`

[59]:
```
gme_data.reset_index(inplace=True)
gme_data.head()
```

[59]:
```
        Date      Open      High       Low     Close    Volume  Dividends  \
0 2002-02-13  6.480514  6.773400  6.413184  6.766667  19054000        0.0
1 2002-02-14  6.850829  6.864295  6.682504  6.733001   2755400        0.0
2 2002-02-15  6.733000  6.749832  6.632005  6.699335   2097400        0.0
3 2002-02-19  6.665671  6.665671  6.312189  6.430017   1852600        0.0
4 2002-02-20  6.463682  6.648839  6.413184  6.648839   1723200        0.0

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

0.4 Question 4: Use Webscraping to Extract GME Revenue Data

```
[60]: gme_url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
      gme_html_data = requests.get(gme_url).text
```

```
[62]: gme_soup = BeautifulSoup(gme_html_data, "lxml")
```

```
[63]: gme_tables = gme_soup.find_all('table')

      for index,table in enumerate(gme_tables):
          if ("GameStop Quarterly Revenue" in str(table)):
              gme_table_index = index


      gme_revenue = pd.DataFrame(columns=["Date", "Revenue"])

      for row in gme_tables[gme_table_index].tbody.find_all("tr"):
          col = row.find_all("td")
          if (col !=[]):
              date = col[0].text
              revenue = col[1].text.replace("$", "").replace(",", "")
              gme_revenue = gme_revenue.append({"Date" : date, "Revenue" : revenue},␣
       ↪ignore_index=True)
```

```
[64]: gme_revenue.tail()
```

```
[64]:          Date Revenue
      48  2010-01-31    3524
      49  2009-10-31    1835
      50  2009-07-31    1739
      51  2009-04-30    1981
      52  2009-01-31    3492
```

0.5 Question 5: Plot Tesla Stock Graph

```
[74]: import plotly.graph_objects as go
      from plotly.subplots import make_subplots
      import yfinance as yf
      import pandas as pd

      stock_data = yf.download("TSLA", start="2020-01-01", end="2021-09-30",␣
       ↪progress=False)
      revenue_data = yf.download("TSLA", start="2020-01-01", end="2021-09-30",␣
       ↪progress=False)
      stock_data.reset_index(inplace=True)
      revenue_data.reset_index(inplace=True)

      def make_graph(stock_data, revenue_data, stock):
```

```python
    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.Volume.astype("float"),
                            name="Volume"), 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 ($US)", row=1, col=1)
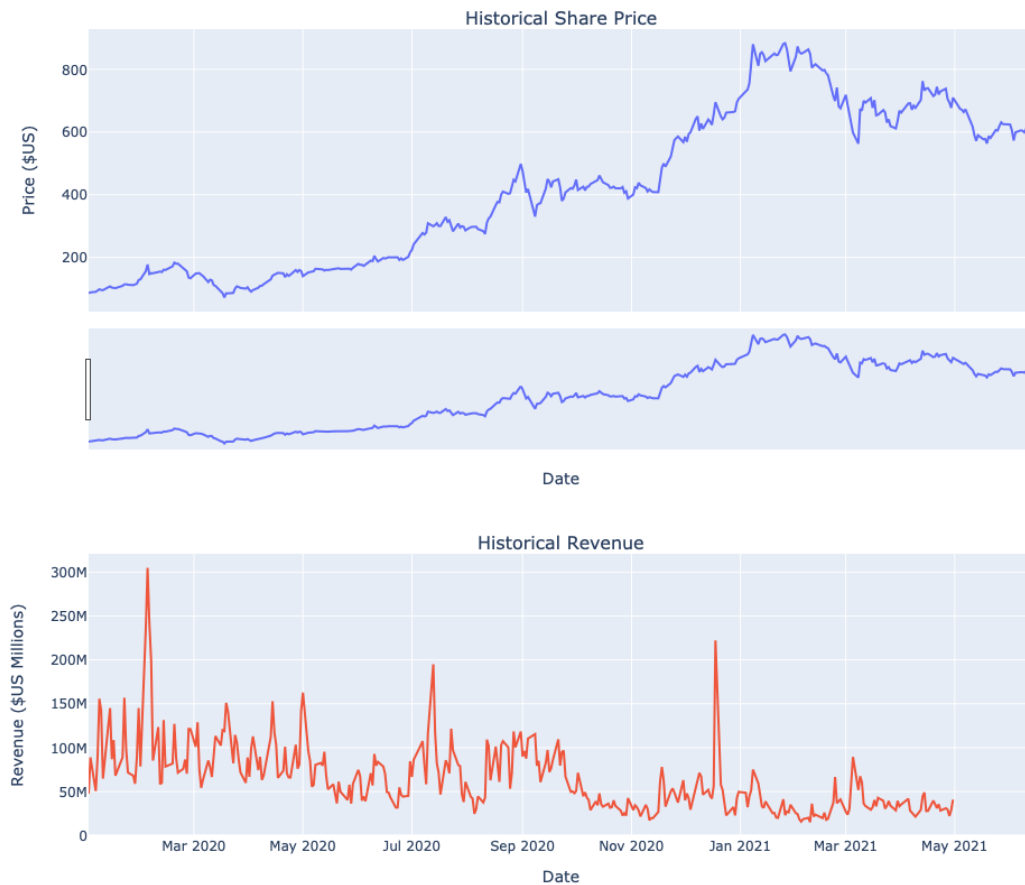    fig.update_yaxes(title_text="Revenue ($US Millions)", row=2, col=1)

    fig.update_layout(showlegend=False,
                      height=900,
                      title=stock,
                      xaxis_rangeslider_visible=True)

    fig.show()

make_graph(stock_data, revenue_data, 'TSLA')
```

TSLA

Historical Share Price

Historical Revenue

0.6   Question 6: Plot GameStop Stock Graph

```
[76]: stock_data = yf.download("GME", start="2020-01-01", end="2021-09-30",␣
      ↪progress=False)
      revenue_data = yf.download("GME", start="2020-01-01", end="2021-09-30",␣
      ↪progress=False)
      stock_data.reset_index(inplace=True)
      revenue_data.reset_index(inplace=True)

      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)
```

```python
    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.Volume.astype("float"),
                            name="Volume"), 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)
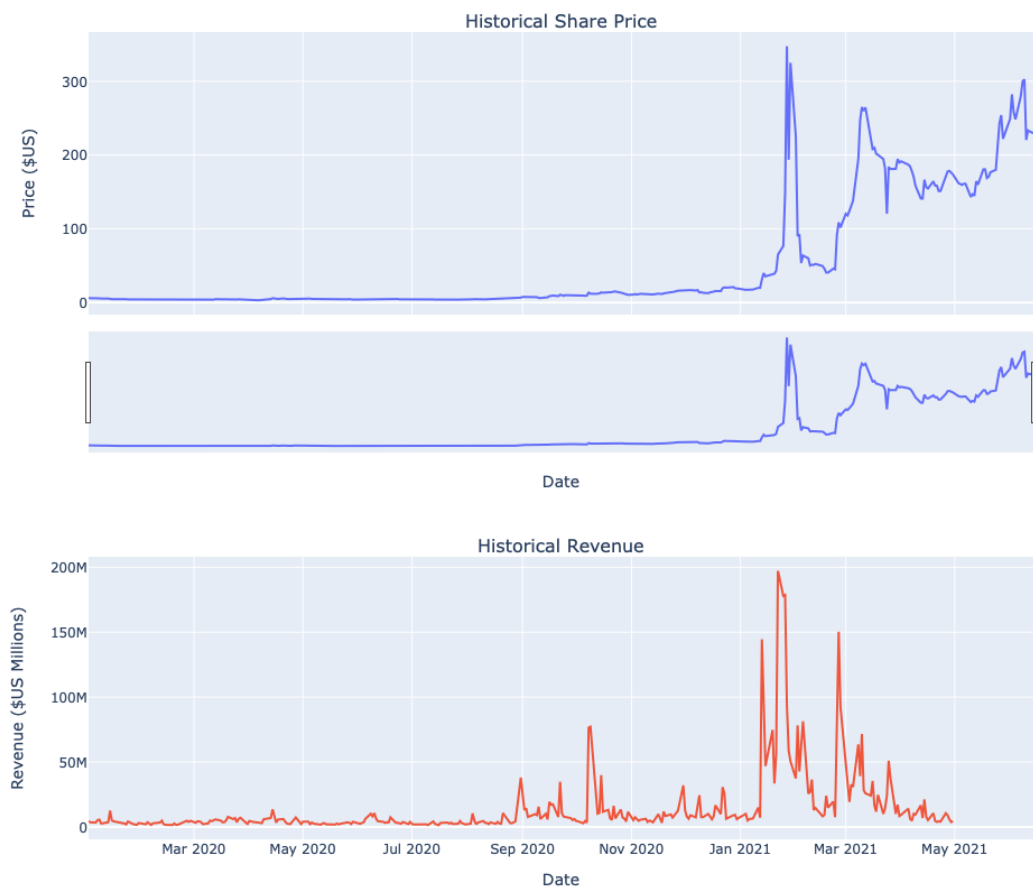
    fig.update_layout(showlegend=False,
                      height=900,
                      title=stock,
                      xaxis_rangeslider_visible=True)

    fig.show()

make_graph(stock_data, revenue_data, 'GME')
```

GME

Historical Share Price

Historical Revenue

[ ]: