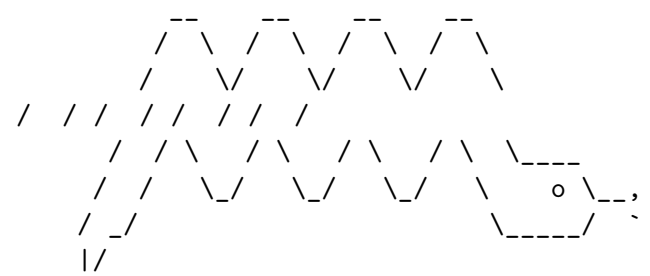
PYTHONASSIGNMENT

April 9, 2024

[1]:
```
!pip install yfinance==0.1.67
!mamba install bs4==4.10.0 -y
!pip install nbformat
```

```
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: 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: requests>=2.20 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (2.29.0)
Requirement already satisfied: multitasking>=0.0.7 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (0.0.11)
Requirement already satisfied: lxml>=4.5.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
yfinance==0.1.67) (5.2.1)
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) (2023.3)
Requirement already satisfied: charset-normalizer<4,>=2 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
requests>=2.20->yfinance==0.1.67) (3.1.0)
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) (3.4)
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) (1.26.15)
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) (2023.5.7)
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)
```



```
        mamba (1.4.2) 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  0.1spkgs/r/linux-64
No change
pkgs/main/noarch                                              No change
pkgs/main/linux-64                                            No change
pkgs/r/noarch                                                 No change

Pinned packages:
  - python 3.7.*
```

```
Transaction

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

  All requested packages already installed

Requirement already satisfied: nbformat in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (4.2.0)
Requirement already satisfied: ipython-genutils in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat)
(0.2.0)
Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat)
(4.17.3)
Requirement already satisfied: jupyter-core in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat)
(4.12.0)
Requirement already satisfied: traitlets>=4.1 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from nbformat)
(5.9.0)
Requirement already satisfied: attrs>=17.4.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat) (23.1.0)
Requirement already satisfied: importlib-metadata in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat) (4.11.4)
Requirement already satisfied: importlib-resources>=1.4.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat) (5.12.0)
Requirement already satisfied: pkgutil-resolve-name>=1.3.10 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat) (1.3.10)
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
jsonschema!=2.5.0,>=2.4->nbformat) (0.19.3)
Requirement already satisfied: typing-extensions in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from
jsonschema!=2.5.0,>=2.4->nbformat) (4.5.0)
Requirement already satisfied: zipp>=3.1.0 in
/home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from importlib-
resources>=1.4.0->jsonschema!=2.5.0,>=2.4->nbformat) (3.15.0)
```

[48]:
```
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
```

[49]:
```
import warnings
# Ignore all warnings
warnings.filterwarnings("ignore", category=FutureWarning)
```

[50]:
```
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()
```

[51]:
```
import yfinance as yf
```

[52]:
```
ticker=yf.Ticker("TSLA")
```

[53]:
```
tesla_data=ticker.history(period="max")
```

[54]:
```
tesla_data.reset_index(inplace=True)
```

[57]:
```
tesla_data.head()
```

[57]:

	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
```

[56]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

[13]:
```
url="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
 ↪IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm"
```

[17]:
```
html_data=requests.get(url).text
```

[18]:
```
soup = BeautifulSoup(html_data, "html.parser")
```

[19]:
```
tables= soup.find_all("table")
tesla_revenue=pd.read_html(str(tables[1]))[0]
```

[20]:
```
tesla_revenue.columns=["Date","Revenue"]
```

[21]:
```
tesla_revenue["Revenue"] = tesla_revenue['Revenue'].str.replace(',|\$',"")
```

[22]:
```
tesla_revenue.dropna(inplace=True)

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

[23]:
```
tesla_revenue.tail()
```

[23]:
```
          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
```

[24]:
```
import yfinance as yf
```

[25]:
```
ticker=yf.Ticker("GME")
gme_data=ticker.history(period="max")
```

[26]:
```
gme_data.reset_index(inplace=True)
```

[27]:
```
gme_data.head()
```

[27]:
```
        Date      Open      High       Low     Close    Volume  Dividends  \
0 2002-02-13  1.620129  1.693350  1.603296  1.691667  76216000        0.0
1 2002-02-14  1.712707  1.716074  1.670626  1.683251  11021600        0.0
2 2002-02-15  1.683250  1.687458  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
```

[33]:
```
import requests
from bs4 import BeautifulSoup
import pandas as pd
```

[34]:
```
url="https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/
 ↪IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/stock.html"
```

[35]:
```
html_data=requests.get(url).text
```

[36]:
```
soup = BeautifulSoup(html_data, "html.parser")
```

[37]:
```
tables=soup.find_all("table")
gme_revenue=pd.read_html(str(tables[1]))[0]
```

[38]:
```
gme_revenue.columns=["Date","Revenue"]
```

[39]:
```
gme_revenue["Revenue"]=gme_revenue["Revenue"].str.replace(',',"").str.
 ↪replace('$',"")
```

[ ]:
```
gme_revenue.dropna(inplace=True)
gme_revenue[gme_revenue["Revenue"] !=""]
```

[43]:
```
gme_revenue.tail()
```

[43]:
```
          Date Revenue
57  2006-01-31    1667
58  2005-10-31     534
59  2005-07-31     416
60  2005-04-30     475
61  2005-01-31     709
```

[44]:
```
import matplotlib as plt
```

[70]:
```
def make_graph(stock_data, revenue_data, stock_name):
    fig, ax = plt.subplots(figsize=(10, 6))
    ax.plot(stock_data["Date"], stock_data["Close"], label="Stock Price",␣
 ↪color="blue")
    ax.set_xlabel("Date")
    ax.set_ylabel("Stock Price")
    ax2 = ax.twinx()
    ax2.plot(revenue_data["Date"], revenue_data["Revenue"], label="Revenue",␣
 ↪color="red")
    ax2.set_ylabel("Revenue")
    plt.title(f"{stock_name} Stock Price and Revenue")
    plt.show()
```

[75]:
```
make_graph(tesla_data, tesla_revenue, "TESLA")
```

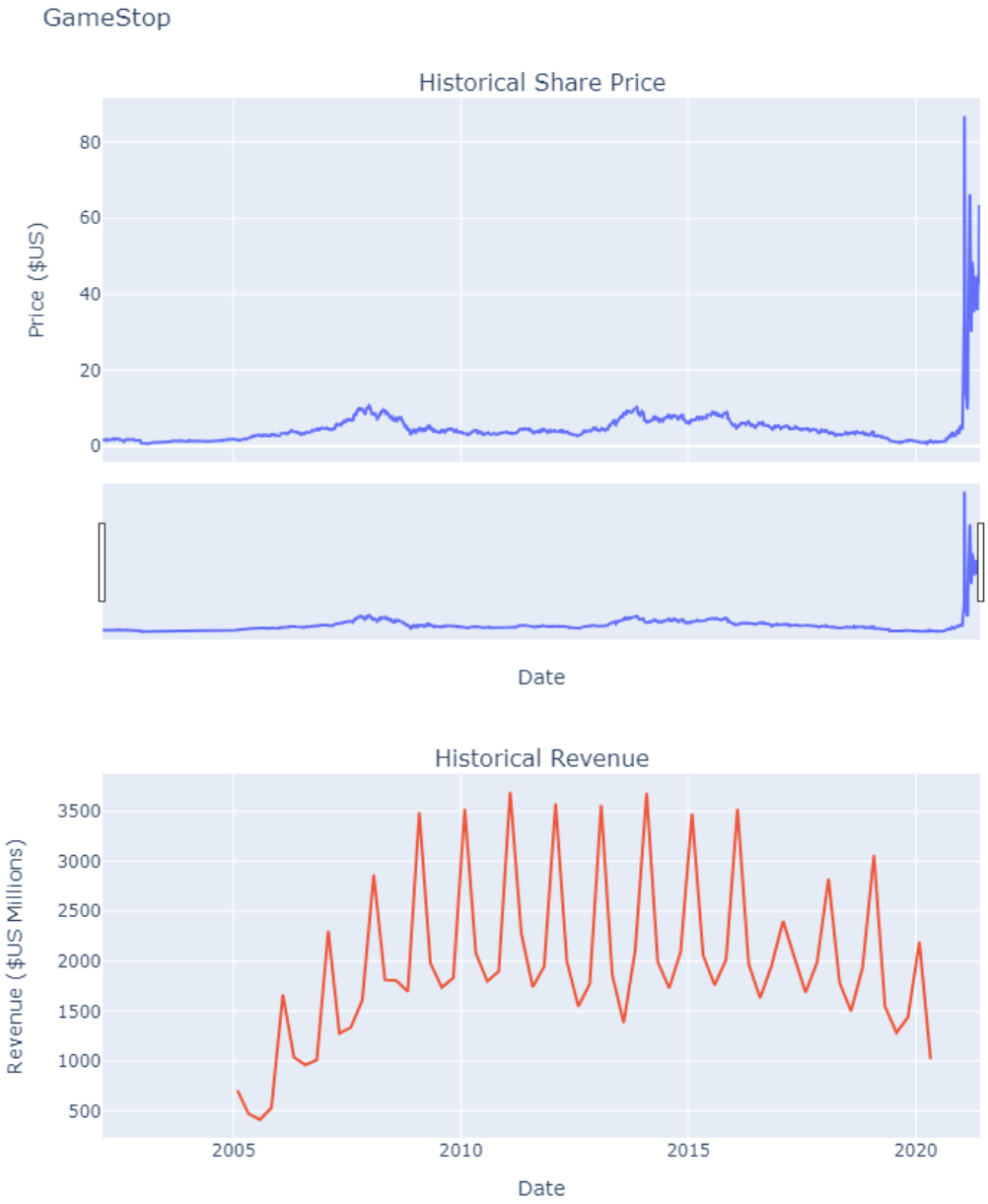
```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
/tmp/ipykernel_3135/4174722495.py in <module>
----> 1 make_graph(tesla_data, tesla_revenue, "TESLA")

/tmp/ipykernel_3135/4146753501.py in make_graph(stock_data, revenue_data,␣
 ↪stock_name)
      1 def make_graph(stock_data, revenue_data, stock_name):
----> 2     fig, ax = plt.subplots(figsize=(10, 6))
      3     ax.plot(stock_data["Date"], stock_data["Close"], label="Stock␣
 ↪Price", color="blue")
      4     ax.set_xlabel("Date")
      5     ax.set_ylabel("Stock Price")

~/conda/envs/python/lib/python3.7/site-packages/matplotlib/_api/__init__.py in␣
 ↪__getattr__(name)
    221             return props[name].__get__(instance)
    222         raise AttributeError(
--> 223             f"module {cls.__module__!r} has no attribute {name!r}")
    224
    225     return __getattr__

AttributeError: module 'matplotlib' has no attribute 'subplots'
```

[76]:
```
import matplotlib.pyplot as plt

def make_graph(stock_data, revenue_data, stock_name):
    fig, ax = plt.subplots(figsize=(10, 6))
    ax.plot(stock_data["Date"], stock_data["Close"], label="Stock Price",␣
 ↪color="blue")
```

```
    ax.set_xlabel("Date")
    ax.set_ylabel("Stock Price")
    ax2 = ax.twinx()
    ax2.plot(revenue_data["Date"], revenue_data["Revenue"], label="Revenue",␣
 ↪color="red")
    ax2.set_ylabel("Revenue")
    plt.title(f"{stock_name} Stock Price and Revenue")
    plt.xlim([None, "2021-06-01"])  # Limit x-axis to June 2021
    plt.show()

# Call the make_graph function to plot the GameStop Stock Data
make_graph(gme_data, gme_revenue, 'GameStop')
```

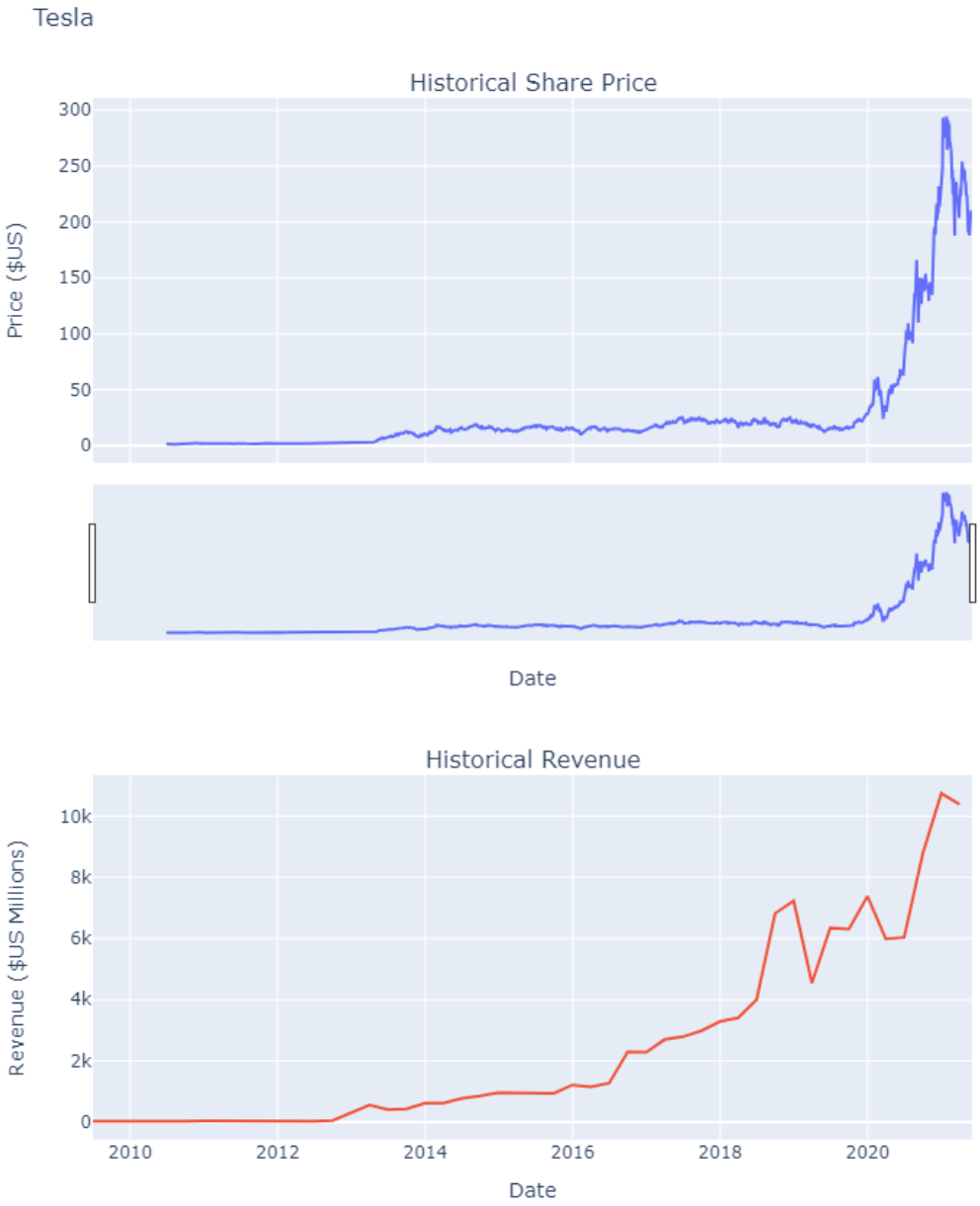
[78]:
```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import pandas as pd

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=0.3)

    stock_data_specific = stock_data[stock_data["Date"] <= '2021-06-01']
    revenue_data_specific = revenue_data[revenue_data["Date"] <= '2021-06-01']
```

```
    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()

# Call the make_graph function to plot the GameStop Stock Data
make_graph(gme_data, gme_revenue, 'GameStop')
```



[79]:
```
from plotly.subplots import make_subplots
import plotly.graph_objects as go
import pandas as pd

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=0.3)

    stock_data_specific = stock_data[stock_data["Date"] <= '2021-06-01']
    revenue_data_specific = revenue_data[revenue_data["Date"] <= '2021-06-01']

    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()

# Call the make_graph function to plot the Tesla Stock Data
make_graph(tesla_data, tesla_revenue, 'Tesla')
```


Tesla
Historical Share Price
Price ($US)
300
250
200
150
100
50
0
Date
Historical Revenue
Revenue ($US Millions)
10k
8k
6k
4k
2k
0
2010
2012
2014
2016
2018
2020
Date

[ ]: