Peer graded Assignment Analyzing Historical Stock Revenue Data and Building a Dashboard

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

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
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_spac
    fig.add_trace(go.Scatter(x=pd.to_datetime(stock_data.Date, infer_datetime_format=True), y=stock_data.Close.astype("float"), name="Sha
    fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data.Date, infer_datetime_format=True), y=revenue_data.Revenue.astype("float"), nam
    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()
```

```
Requirement already satisfied: yfinance in /usr/local/lib/python3.10/dist-packages (0.2.31)
Requirement already satisfied: pandas>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.5.3)
Requirement already satisfied: numpy>=1.16.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.23.5)
Requirement already satisfied: requests>=2.31 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.31.0)
Requirement already satisfied: multitasking>=0.0.7 in /usr/local/lib/python3.10/dist-packages (from yfinance) (0.0.11)
Requirement already satisfied: lxml>=4.9.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.9.3)
Requirement already satisfied: appdirs>=1.4.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.4.4)
Requirement already satisfied: pytz>=2022.5 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2023.3.post1)
Requirement already satisfied: frozendict>=2.3.4 in /usr/local/lib/python3.10/dist-packages (from yfinance) (2.3.8)
Requirement already satisfied: peewee>=3.16.2 in /usr/local/lib/python3.10/dist-packages (from yfinance) (3.16.3)
Requirement already satisfied: beautifulsoup4>=4.11.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (4.11.2)
Requirement already satisfied: html5lib>=1.1 in /usr/local/lib/python3.10/dist-packages (from yfinance) (1.1)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4>=4.11.1->yfinance) (2.
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (1.16.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from html5lib>=1.1->yfinance) (0.5.1)
Requirement already satisfied: python-dateutil>=2.8.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.3.0->yfinance) (2.
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2.0.6
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.31->yfinance) (2023.
Collecting bs4
  Downloading bs4-0.0.1.tar.gz (1.1 kB)
  Preparing metadata (setup.py) ... done
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from bs4) (4.11.2)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->bs4) (2.5)
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=1256 sha256=1818de50fba61b8268232279d4d2e66b861f436e11bab03d0c8c5
  Stored in directory: /root/.cache/pip/wheels/25/42/45/b773edc52acb16cd2db4cf1a0b47117e2f69bb4eb300ed0e70
Successfully built bs4
Installing collected packages: bs4
Successfully installed bs4-0.0.1
```

```
#1
tesla = yf.Ticker("TSLA")

#2
tesla_data = tesla.history(period="max")

#3
tesla_data.reset_index(inplace=True)
tesla_data.head()
```

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2010-06-29 00:00:00-04:00	1.266667	1.666667	1.169333	1.592667	281494500	0.0	0.0

```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# 1
tesla_url = "https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue"
tesla_html_data = requests.get(tesla_url).text

# 2
tesla_soup = BeautifulSoup(tesla_html_data, "html5lib")

# 3
tesla_tables = tesla_soup.find_all('table')

tesla_table_index = None  # Define tesla_table_index outside the loop

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

if tesla_table_index is not None:  # Check if tesla_table_index was found
    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)

    # 4
    tesla_revenue = tesla_revenue[tesla_revenue['Revenue'] != ""]
    # 5
    print(tesla_revenue.tail())
else:
    print("Table not found on the page.")
```

```
     Table not found on the page.
```

```
#1
gamestop = yf.Ticker("GME")

#2
gme_data = gamestop.history(period="max")

#3
gme_data.reset_index(inplace=True)
gme_data.head()
```

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-02-13 00:00:00-05:00	1.620128	1.693350	1.603296	1.691667	76216000	0.0	0.0
1	2002-02-14 00:00:00-05:00	1.712707	1.716073	1.670626	1.683250	11021600	0.0	0.0
2	2002-02-15 00:00:00-05:00	1.683250	1.687458	1.658002	1.674834	8389600	0.0	0.0
3	2002-02-19 00:00:00-05:00	1.666418	1.666418	1.578047	1.607504	7410400	0.0	0.0
4	2002-02-20 00:00:00-05:00	1.615921	1.662210	1.603296	1.662210	6892800	0.0	0.0

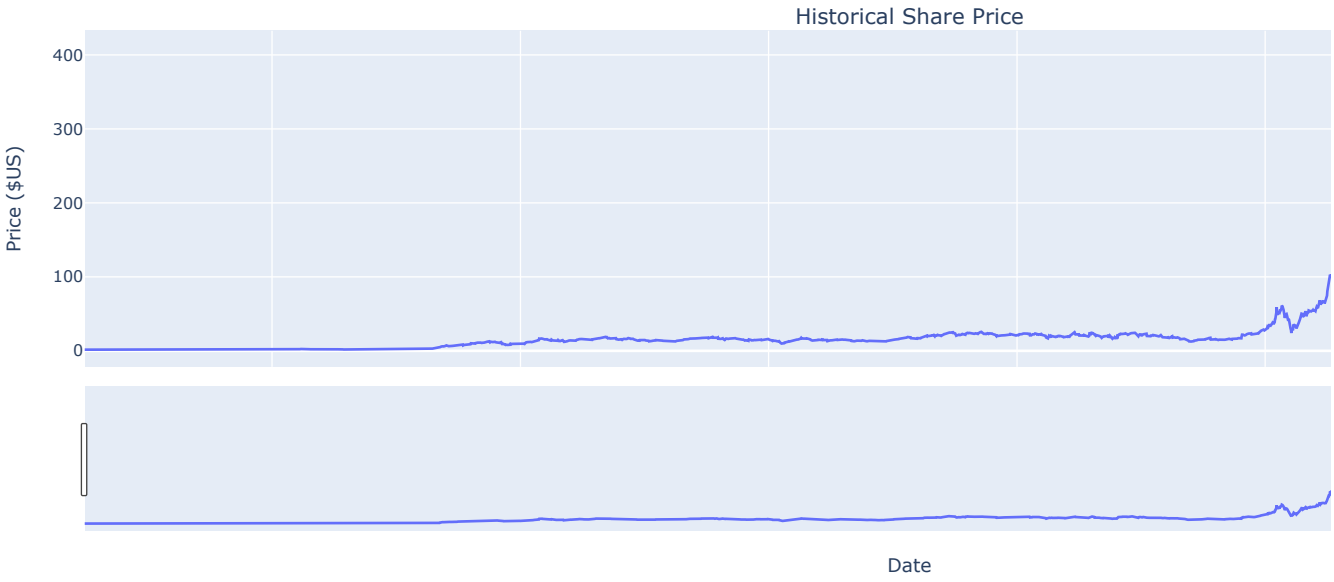
```
import requests
from bs4 import BeautifulSoup
import pandas as pd

# 1
gme_url = "https://www.macrotrends.net/stocks/charts/GME/gamestop/revenue"
gme_html_data = requests.get(gme_url).text

# 2
gme_soup = BeautifulSoup(gme_html_data, "html5lib")

# 3
gme_tables = gme_soup.find_all('table')

gme_table_index = None  # Define gme_table_index outside the loop
```

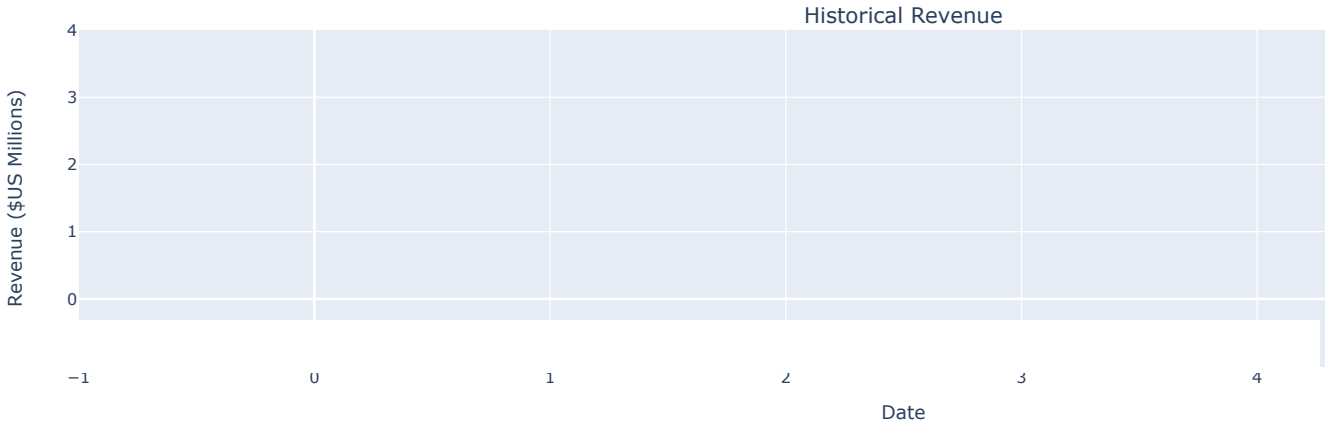
```
for index, table in enumerate(gme_tables):
    if "GameStop Quarterly Revenue" in str(table):
        gme_table_index = index

if gme_table_index is not None:  # Check if gme_table_index was found
    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)

    # 4
    print(gme_revenue.tail())
else:
    print("Table not found on the page.")
```

```
 Table not found on the page.
```

```
#1
make_graph(tesla_data, tesla_revenue, 'Tesla')

#2
make_graph(gme_data, gme_revenue, 'GameStop')
```

Tesla





GameStop