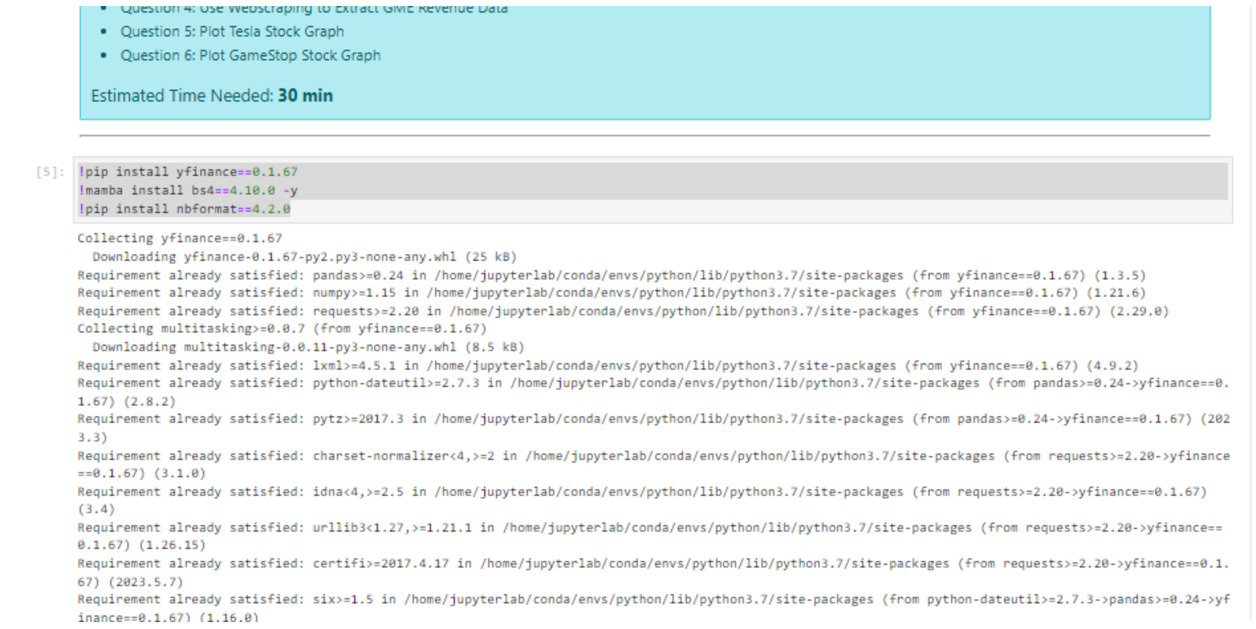
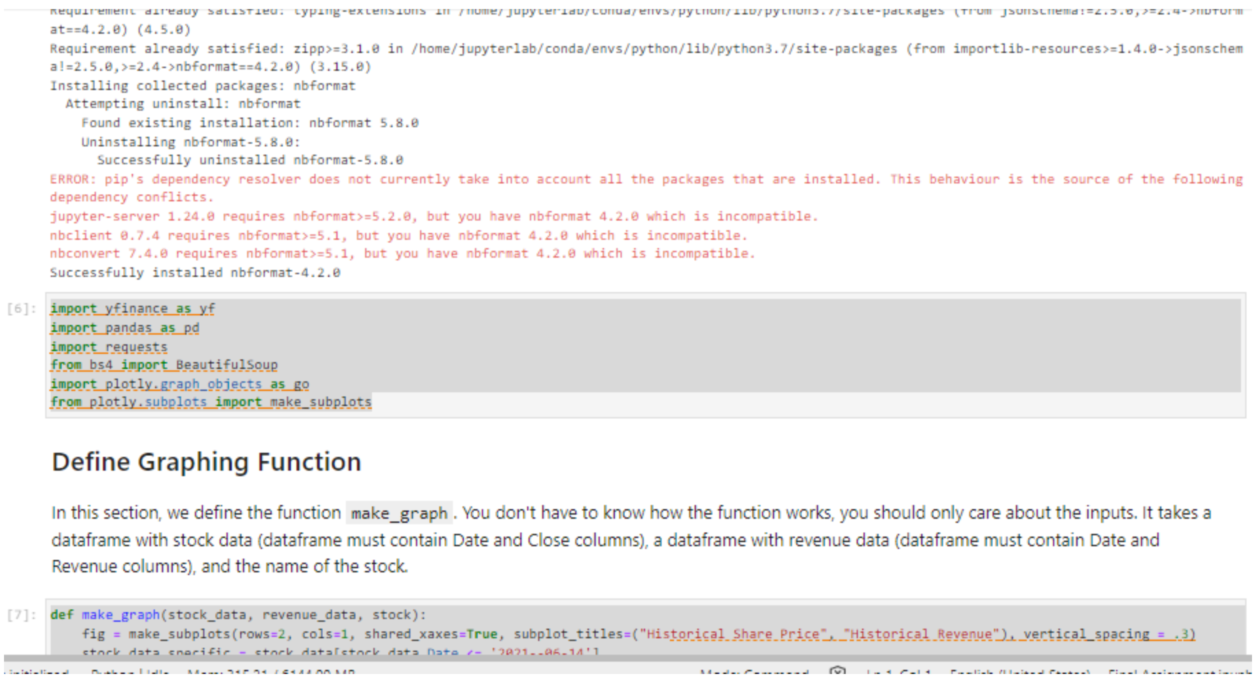


- Question 4: Use Webscraping to Extract GME Revenue Data
- Question 5: Plot Tesla Stock Graph
- Question 6: Plot GameStop Stock Graph

Estimated Time Needed: **30 min**

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

```
Collecting yfinance==0.1.67
  Downloading yfinance-0.1.67-py2.py3-none-any.whl (25 kB)
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)
Collecting multitasking>=0.0.7 (from yfinance==0.1.67)
  Downloading multitasking-0.0.11-py3-none-any.whl (8.5 kB)
Requirement already satisfied: lxml>=4.5.1 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from yfinance==0.1.67) (4.9.2)
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) (202
3.3)
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->yf
inance==0.1.67) (1.16.0)
```



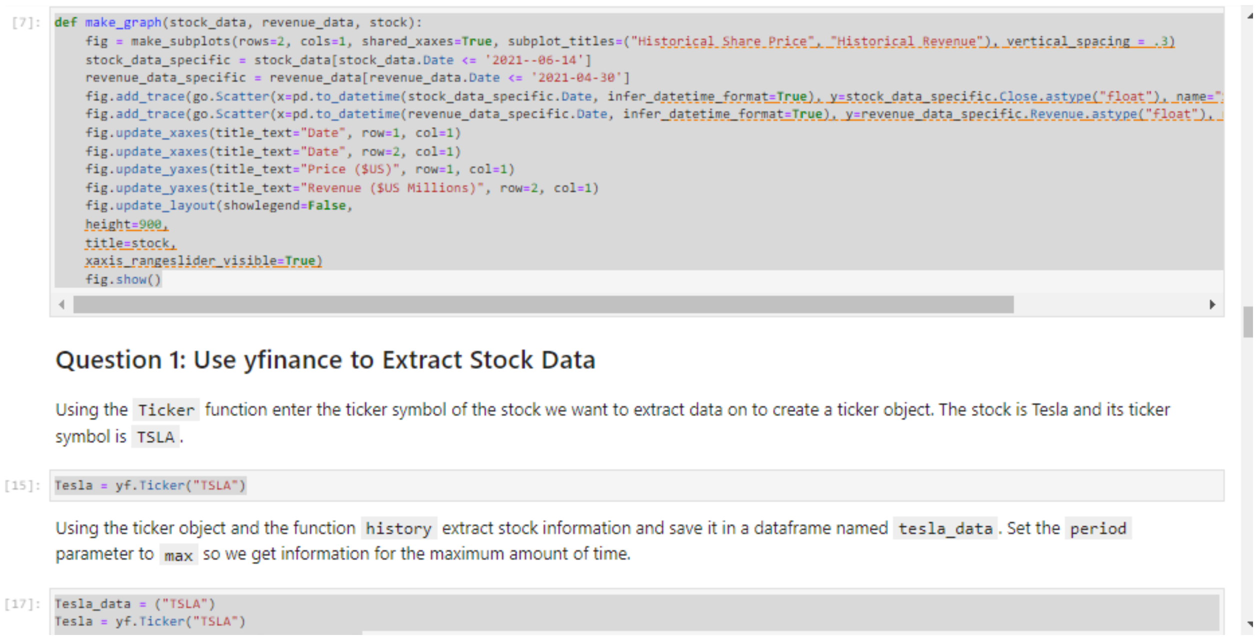
```
Requirement already satisfied: typing-extensions in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from jsonschema!=2.5.0,>=2.4->nbform
at==4.2.0) (4.5.0)
Requirement already satisfied: zipp>=3.1.0 in /home/jupyterlab/conda/envs/python/lib/python3.7/site-packages (from importlib-resources>=1.4.0->jsonschem
a!=2.5.0,>=2.4->nbformat==4.2.0) (3.15.0)
Installing collected packages: nbformat
  Attempting uninstall: nbformat
    Found existing installation: nbformat 5.8.0
    Uninstalling nbformat-5.8.0:
      Successfully uninstalled nbformat-5.8.0
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behaviour is the source of the following
dependency conflicts.
jupyter-server 1.24.0 requires nbformat>=5.2.0, but you have nbformat 4.2.0 which is incompatible.
nbclient 0.7.4 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
nbconvert 7.4.0 requires nbformat>=5.1, but you have nbformat 4.2.0 which is incompatible.
Successfully installed nbformat-4.2.0
```

```
[6]: 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.

```
[7]: 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']
```



```
[7]: 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="
         fig.add_trace(go.Scatter(x=pd.to_datetime(revenue_data_specific.Date, infer_datetime_format=True), y=revenue_data_specific.Revenue.astype("float"),
         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()
```

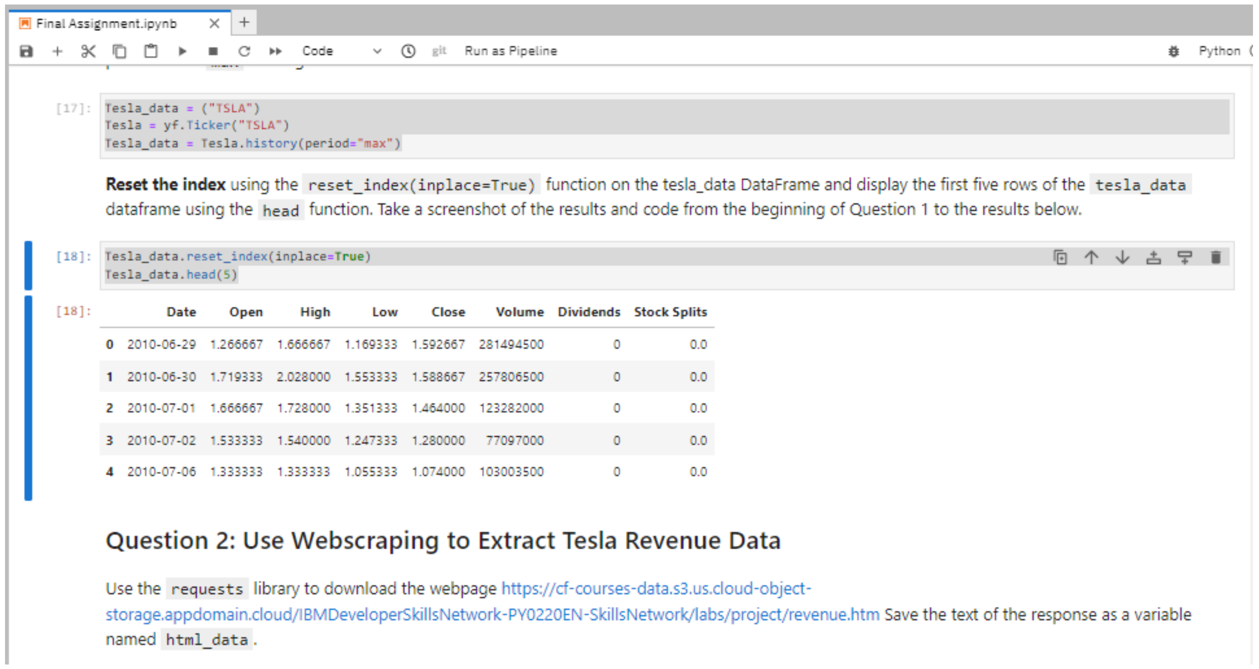
Question 1: Use yfinance to Extract Stock Data

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`.

```
[15]: 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.

```
[17]: Tesla_data = ("TSLA")
      Tesla = yf.Ticker("TSLA")
```



```
[17]: Tesla_data = ("TSLA")
      Tesla = yf.Ticker("TSLA")
      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.

```
[18]: Tesla_data.reset_index(inplace=True)
      Tesla_data.head(5)
```

```
[18]:        Date      Open      High       Low     Close     Volume  Dividends  Stock Splits
      0  2010-06-29  1.266667  1.666667  1.169333  1.592667  281494500          0           0.0
      1  2010-06-30  1.719333  2.028000  1.553333  1.588667  257806500          0           0.0
      2  2010-07-01  1.666667  1.728000  1.351333  1.464000  123282000          0           0.0
      3  2010-07-02  1.533333  1.540000  1.247333  1.280000   77097000          0           0.0
      4  2010-07-06  1.333333  1.333333  1.055333  1.074000  103003500          0           0.0
```

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`.

storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/revenue.htm Save the text of the response as a variable named `html_data`.

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

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

Parse the html data using `beautiful_soup`.

```
[33]: soup = BeautifulSoup(html_data, 'html.parser')
```

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

```
[34]: pd.read_html('https://www.macrotrends.net/stocks/charts/TSLA/tesla/revenue')[1]
```

[34]:

	Tesla Quarterly Revenue(Millions of US $)	Tesla Quarterly Revenue(Millions of US $).1
0	2023-03-31	$23,329
1	2022-12-31	$24,318
2	2022-09-30	$21,454
3	2022-06-30	$16,934
4	2022-03-31	$18,756

52	2010-03-31	$21
53	2009-12-31	NaN
54	2009-09-30	$46
55	2009-06-30	$27

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

```
[61]: tesla_revenue['Revenue'] = tesla_revenue['Revenue'].replace({'~~\$': '', ',': ''}, regex=True)~~
```

Execute the following lines to remove an null or empty strings in the Revenue column.

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

Display the last 5 row of the `tesla_revenue` dataframe using the `tail` function. Take a screenshot of the results.

```
•[87]: tesla_revenue.tail()
```

[87]:

Date	Revenue

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