**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**

* Define a Function that Makes a Graph
* Question 1: Use yfinance to Extract Stock Data
* Question 2: Use Webscraping to Extract Tesla Revenue Data
* Question 3: Use yfinance to Extract Stock Data
* 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**

[ ]:



**!**pip install yfinance**==**0.1.67

**!**mamba install bs4**==**4.10.0 **-**y

**!**pip install nbformat**==**4.2.0

[ ]:



**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.

[ ]:



**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()

**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.

[ ]:



​

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.

[ ]:



​

**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.

[ ]:



​

**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.

[ ]:



​

Parse the html data using beautiful\_soup.

[ ]:



​

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

[ ]:



​

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

[ ]:



tesla\_revenue["Revenue"] **=** tesla\_revenue['Revenue'].str.replace(',|\$',"")

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

[ ]:



tesla\_revenue.dropna(inplace**=True**)

​

tesla\_revenue **=** tesla\_revenue[tesla\_revenue['Revenue'] **!=** ""]

Display the last 5 row of the tesla\_revenue dataframe using the tail function. Take a screenshot of the results.

[ ]:



​

**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 ticker symbol is GME.

[ ]:



​

Using the ticker object and the function history extract stock information and save it in a dataframe named gme\_data. Set the period parameter to max so we get information for the maximum amount of time.

[ ]:



​

**Reset the index** using the reset\_index(inplace=True) function on the gme\_data DataFrame and display the first five rows of the gme\_data dataframe using the head function. Take a screenshot of the results and code from the beginning of Question 3 to the results below.

[ ]:



​

**Question 4: Use Webscraping to Extract GME 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/stock.html>. Save the text of the response as a variable named html\_data.

[ ]:



​

Parse the html data using beautiful\_soup.

[ ]:



​

Using BeautifulSoup or the read\_html function extract the table with GameStop Quarterly Revenue and store it into a dataframe named gme\_revenue. The dataframe should have columns Date and Revenue. Make sure the comma and dollar sign is removed from the Revenue column using a method similar to what you did in Question 2.

Click here if you need help locating the table

[ ]:



​

Display the last five rows of the gme\_revenue dataframe using the tail function. Take a screenshot of the results.

[ ]:



​

**Question 5: Plot Tesla Stock Graph**

Use the make\_graph function to graph the Tesla Stock Data, also provide a title for the graph. The structure to call the make\_graph function is make\_graph(tesla\_data, tesla\_revenue, 'Tesla'). Note the graph will only show data upto June 2021.

[ ]:



​

**Question 6: Plot GameStop Stock Graph**

Use the make\_graph function to graph the GameStop Stock Data, also provide a title for the graph. The structure to call the make\_graph function is make\_graph(gme\_data, gme\_revenue, 'GameStop'). Note the graph will only show data upto June 2021.