Question 1: Use yfinance to Extract Stock Data

A screenshot of a computer

Description automatically generated

Title:

yfinance-to-Extract-Stock-Data-tesla-Data

Caption:

tesla = yf.Ticker("TSLA")

tesla\_data = tesla.history(period="max")

tesla\_data.reset\_index(inplace=True)

tesla\_data.head()

Question 2: Use Webscraping to Extract Tesla Revenue Data

A screenshot of a computer

Description automatically generated

Title:

Webscraping-to-Extract-Tesla-Revenue-Data-tesla-data

Caption:

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

html\_data = requests.get(url)

soup = BeautifulSoup(html\_data.content, "html.parser")

# Find all tables on the webpage

tables = soup.find\_all('table')

# Initialize DataFrame

tesla\_revenue = pd.DataFrame(columns=["Date", "Revenue"])

# Identify the relevant table

for table in tables:

if "Tesla Quarterly Revenue" in table.get\_text():

# Loop through rows in the relevant table

for row in table.find\_all("tr"):

cols = row.find\_all("td")

if len(cols) == 2: # Ensure there are 2 columns: Date and Revenue

date = cols[0].text.strip()

revenue = cols[1].text.strip()

# Clean Revenue Data

revenue = revenue.replace('$', '').replace(',', '')

# Add row to DataFrame

tesla\_revenue = pd.concat([tesla\_revenue, pd.DataFrame({

"Date": [date],

"Revenue": [revenue]

})], ignore\_index=True)

# Display the DataFrame

print(tesla\_revenue)

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

tesla\_revenue.dropna(inplace=True)

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

tesla\_revenue.tail(5)

Question 3: Use yfinance to Extract Stock Data

A screenshot of a computer

Description automatically generated

Title:

yfinance to Extract Stock Data-gme-data

Caption:

gamestop = yf.Ticker("GME”)

gme\_data = gamestop.history(period="max”)

gme\_data.reset\_index(inplace=True)

gme\_data.head()

Question 4: Use Webscraping to Extract GME Revenue Data

Title:

Webscraping to Extract GME Revenue Data-gme-revenue

Caption:

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

html\_data\_gme = requests.get(url)

soup = BeautifulSoup(html\_data\_gme.content, "html.parser”)

# Find all tables on the webpage

tables = soup.find\_all('table')

# Initialize DataFrame

gme\_revenue = pd.DataFrame(columns=["Date", "Revenue"])

# Identify the relevant table

for table in tables:

if "GameStop Quarterly Revenue" in table.get\_text():

# Loop through rows in the relevant table

for row in table.find\_all("tr"):

cols = row.find\_all("td")

if len(cols) == 2: # Ensure there are 2 columns: Date and Revenue

date = cols[0].text.strip()

revenue = cols[1].text.strip()

# Clean Revenue Data

revenue = revenue.replace('$', '').replace(',', '')

# Add row to DataFrame

gme\_revenue = pd.concat([gme\_revenue, pd.DataFrame({

"Date": [date],

"Revenue": [revenue]

})], ignore\_index=True)

# Display the DataFrame

print(gme\_revenue)

gme\_revenue.tail(5)

Question 5: Plot Tesla Stock Graph

A graph showing a price

Description automatically generated with medium confidence

A graph showing the growth of a company

Description automatically generated

Title:

Plot Tesla Stock Graph-make-graph

Caption:

make\_graph(tesla\_data, tesla\_revenue, 'Tesla')

Question 6: Plot GameStop Stock Graph

A graph showing a price

Description automatically generated with medium confidence

A graph showing a graph showing a number of years

Description automatically generated with medium confidence

Title:

Plot GameStop Stock Graph-make-graph

Caption:

make\_graph(gme\_data, gme\_revenue, 'GameStop')