A company's stock share is a piece of the company more precisely:

**A stock (also known as equity) is a security that represents the ownership of a fraction of a corporation. This entitles the owner of the stock to a proportion of the corporation's assets and profits equal to how much stock they own. Units of stock are called "shares." [1]**

An investor can buy a stock and sell it later. If the stock price increases, the investor profits, If it decreases,the investor with incur a loss.  Determining the stock price is complex; it depends on the number of outstanding shares, the size of the company's future profits, and much more. People trade stocks throughout the day the stock ticker is a report of the price of a certain stock, updated continuously throughout the trading session by the various stock market exchanges.

You are a data scientist working for a hedge fund; it's your job to determine any suspicious stock activity. In this lab you will extract stock data using a Python library. We will use the yfinance library, it allows us to extract data for stocks returning data in a pandas dataframe. You will use the lab to extract.

STEP1

!pip install yfinance==0.1.67

#!pip install pandas==1.3.3

STEP2

import yfinance as yf

import pandas as pd

STEP3

Using the Ticker module we can create an object that will allow us to access functions to extract data. To do this we need to provide the ticker symbol for the stock, here the company is Apple and the ticker symbol is AAPL.

apple = yf.Ticker("AAPL")

STEP4

Using the attribute info we can extract information about the stock as a Python dictionary

apple\_info=apple.info

apple\_info

STEP5

We can get the 'country' using the key country

apple\_info['country']

STEP6

A share is the single smallest part of a company's stock that you can buy, the prices of these shares fluctuate over time. Using the <code>history()</code> method we can get the share price of the stock over a certain period of time. Using the `period` parameter we can set how far back from the present to get data. The options for `period` are 1 day (1d), 5d, 1 month (1mo) , 3mo, 6mo, 1 year (1y), 2y, 5y, 10y, ytd, and max.

apple\_share\_price\_data = apple.history(period="max")

apple\_share\_price\_data.head()

apple\_share\_price\_data.reset\_index(inplace=True)

apple\_share\_price\_data.plot(x="Date", y="Open")

apple.dividends

apple.dividends.plot()

**Extracting Stock Data Using a Web Scraping**

STEP1: INSTALL LIBRARIES

#!pip install pandas==1.3.3

#!pip install requests==2.26.0

!mamba install bs4==4.10.0 -y

!mamba install html5lib==1.1 -y

!pip install lxml==4.6.4

#!pip install plotly==5.

STEP 2

IMPORT LIB

import pandas as pd

import requests

from bs4 import BeautifulSoup

STEP 3

First we must use the `request` library to downlaod the webpage, and extract the text. We will extract Netflix stock data <https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/netflix\_data\_webpage.html>.

url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/netflix\_data\_webpage.html"

url = "https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/netflix\_data\_webpage.html"

data = requests.get(url).text

STEP4

parsing text into HTML with beautiful soup

soup = BeautifulSoup(data, 'html5lib')

Step 5 convert html table to pandas dataframe

netflix\_data = pd.DataFrame(columns=["Date", "Open", "High", "Low", "Close", "Volume"])

# First we isolate the body of the table which contains all the information

# Then we loop through each row and find all the column values for each row

for row in soup.find("tbody").find\_all('tr'):

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

date = col[0].text

Open = col[1].text

high = col[2].text

low = col[3].text

close = col[4].text

adj\_close = col[5].text

volume = col[6].text

# Hands-on Lab: Analyzing Historical Stock/Revenue Data and Building a Dashboard

This environment is to do the Peer Graded Assignment.

In case you need to download the lab notebook (.ipynb file) click [HERE](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-PY0220EN-SkillsNetwork/labs/project/Final%20Assignment.ipynb) to download the lab notebook.

Once you **complete the lab** you can download the notebook as follows:

* Select the notebook which is displayed in the left **Navigation pane.**
* Right click on it and click on the **Download** option.
* Your notebook gets downloaded.

Later upload to GitHub by following the instructions in the reading [Getting Started with GitHub .](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DS0105EN-SkillsNetwork/labs/Module2/GitHub1_Getting_Started.md.html?origin=www.coursera.org)

Further in the **MySubmission tab** of the assignment paste the GitHub link of the notebook in the **URL textbox** instead of the Watson studio link .

This course uses a third-party tool, Hands-on Lab: Analyzing Historical Stock/Revenue Data and Building a Dashboard, to enhance your learning experience. The tool will reference basic information like your name, email, and Coursera ID.

**Extracting and Visualizing Stock Data**

**<li>Define a Function that Makes a Graph</li>**

**<li>Question 1: Use yfinance to Extract Stock Data</li>**

**<li>Question 2: Use Webscraping to Extract Tesla Revenue**

**<li>Question 3: Use yfinance to Extract Stock Data</li>**

**<li>Question 4: Use Webscraping to Extract GME Revenue**

**<li>Question 5: Plot Tesla Stock Graph</li>**

**<li>Question 6: Plot GameStop Stock Graph</li>**

Step1 installation of libraries

!pip install yfinance==0.1.67

!mamba install bs4==4.10.0 -y

!pip install nbformat==4.2.0

Step2

Import the libraries

* 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

step 3

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\_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()

trial activation link

42259ca124fed0bfd927f3a143bbefe6