Final Project Introduction to Python

Benjamin Wilt

Project Purpose:

This project aims to demonstrate that I have learned elementary data science skills in Python by building a graph that pulls the last year of closing prices from Yahoo Finance for tech giants Amazon, Apple, Microsoft, Meta, and Google to compare their performance in several different metrics.

Import Libraries:

In [185…
```
import pandas as pd
from matplotlib import pyplot as plt
import numpy as np
from matplotlib import style
```

Import Data

In [186…
```
AMZN = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/AMZN?period1=1669745710&period2=1701281710&interval=1d&events=history&includeAdjustedClose=true')
MSFT = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/MSFT?period1=1669823012&period2=1701359012&interval=1d&events=history&includeAdjustedClose=true')
AAPL = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/AAPL?period1=1669823047&period2=1701359047&interval=1d&events=history&includeAdjustedClose=true')
GOOG = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/GOOG?period1=1669823151&period2=1701359151&interval=1d&events=history&includeAdjustedClose=true')
META = pd.read_csv('https://query1.finance.yahoo.com/v7/finance/download/META?period1=1669805181&period2=1701341181&interval=1d&events=history&includeAdjustedClose=true')
```

Let's inspect our data to determine how to approach our exploration.

In [187…
```
AMZN.head()
```

Out[187…

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-11-29	94.040001	94.410004	91.440002	92.419998	92.419998	65567300
1	2022-11-30	92.470001	96.540001	91.529999	96.540001	96.540001	102805800
2	2022-12-01	96.989998	97.230003	94.919998	95.500000	95.500000	68488000
3	2022-12-02	94.480003	95.360001	93.779999	94.129997	94.129997	72496400
4	2022-12-05	93.050003	94.059998	90.820000	91.010002	91.010002	71535500

In [188…
```
MSFT.head()
```

Out[188…

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-11-30	240.570007	255.330002	239.860001	255.139999	252.897568	47594200
1	2022-12-01	253.869995	256.119995	250.919998	254.690002	252.451538	26041500
2	2022-12-02	249.820007	256.059998	249.690002	255.020004	252.778641	21528500
3	2022-12-05	252.009995	253.820007	248.059998	250.199997	248.001007	23435300
4	2022-12-06	250.820007	251.860001	243.779999	245.119995	242.965637	22463700

In [189…
```
AAPL.head()
```

Out[189…

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-11-30	141.399994	148.720001	140.550003	148.029999	147.207184	111380900
1	2022-12-01	148.210007	149.130005	146.610001	148.309998	147.485641	71250400
2	2022-12-02	145.960007	148.000000	145.649994	147.809998	146.988419	65447400
3	2022-12-05	147.770004	150.919998	145.770004	146.630005	145.814972	68826400
4	2022-12-06	147.070007	147.300003	141.919998	142.910004	142.115646	64727200

In [190…
```
GOOG.head()
```

Out[190…

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-11-30	95.120003	101.449997	94.669998	101.449997	101.449997	39888100
1	2022-12-01	101.400002	102.589996	100.669998	101.279999	101.279999	21771500
2	2022-12-02	99.370003	101.150002	99.169998	100.830002	100.830002	18821500
3	2022-12-05	99.815002	101.750000	99.355003	99.870003	99.870003	19955500
4	2022-12-06	99.669998	100.209999	96.760002	97.309998	97.309998	20877600

In [191…
```
META.head()
```

Out[191…

	Date	Open	High	Low	Close	Adj Close	Volume
0	2022-11-30	109.510002	118.160004	109.379997	118.099998	118.099998	43348600
1	2022-12-01	119.199997	121.199997	118.400002	120.440002	120.440002	36551400
2	2022-12-02	117.830002	124.040001	117.610001	123.489998	123.489998	39950500
3	2022-12-05	121.750000	124.669998	121.349998	122.430000	122.430000	35474900
4	2022-12-06	119.910004	120.550003	113.739998	114.120003	114.120003	43689200

It seems as if the "head" of the data shows the dates from a year ago. I would like to analyze data from the last week, so I should inspect the tail of the data.

In [192…
```
AMZN.tail()
```

Out[192…

	Date	Open	High	Low	Close	Adj Close	Volume
247	2023-11-22	144.570007	147.740005	144.570007	146.710007	146.710007	45669100
248	2023-11-24	146.699997	147.199997	145.320007	146.740005	146.740005	22378400
249	2023-11-27	147.529999	149.259995	146.880005	147.729996	147.729996	53762400
250	2023-11-28	146.979996	147.600006	145.529999	147.029999	147.029999	42711700
251	2023-11-29	147.850006	148.539993	145.970001	146.320007	146.320007	40569400

In [193…
```
MSFT.tail()
```

Out[193…

	Date	Open	High	Low	Close	Adj Close	Volume
247	2023-11-24	377.329987	377.970001	375.140015	377.429993	377.429993	10176600
248	2023-11-27	376.779999	380.640015	376.200012	378.609985	378.609985	22179200
249	2023-11-28	378.350006	383.000000	378.160004	382.700012	382.700012	20453100
250	2023-11-29	383.760010	384.299988	377.440002	378.850006	378.850006	28942500
251	2023-11-30	378.489990	380.059998	375.470001	375.829987	375.829987	10409333

In [194…
```
AAPL.tail()
```

Out[194…

	Date	Open	High	Low	Close	Adj Close	Volume
247	2023-11-24	190.869995	190.899994	189.250000	189.970001	189.970001	24048300
248	2023-11-27	189.919998	190.669998	188.899994	189.789993	189.789993	40552600
249	2023-11-28	189.779999	191.080002	189.399994	190.399994	190.399994	38415400
250	2023-11-29	190.899994	192.089996	188.970001	189.369995	189.369995	42967700
251	2023-11-30	189.839996	190.320007	188.190002	188.649994	188.649994	21018283

In [195…
```
GOOG.tail()
```

Out[195…

	Date	Open	High	Low	Close	Adj Close	Volume
247	2023-11-24	139.539993	139.677002	137.470001	138.220001	138.220001	8828600
248	2023-11-27	137.570007	139.630005	137.539993	138.050003	138.050003	17886400
249	2023-11-28	137.630005	138.660004	137.039993	138.619995	138.619995	17046900
250	2023-11-29	138.985001	139.669998	136.294998	136.399994	136.399994	20994400
251	2023-11-30	136.399994	136.960007	132.789993	132.932007	132.932007	13408329

In [196…
```
META.tail()
```

Out[196…

	Date	Open	High	Low	Close	Adj Close	Volume
246	2023-11-22	339.209991	342.920013	338.579987	341.489990	341.489990	10702700
247	2023-11-24	340.130005	341.859985	336.769989	338.230011	338.230011	5467500
248	2023-11-27	336.179993	339.899994	334.200012	334.700012	334.700012	15684500
249	2023-11-28	333.399994	339.380005	333.399994	338.989990	338.989990	12637200
250	2023-11-29	339.690002	339.899994	330.779999	332.200012	332.200012	16007400

After inspecting the data it appears that META and AMZN do not have data for the current day. To avoid issues in the future, drop the last row of GOOG, APPL, and MSFT.

In [197…
```
GOOG = GOOG.drop(GOOG.index[-1])
```

In [198…
```
AAPL = AAPL.drop(AAPL.index[-1])
```

In [199…
```
MSFT = MSFT.drop(MSFT.index[-1])
```

It appears that the data can be identified by the date. To analyze the data, it would make sense to set the date as the index in each dataframe.

In [200…
```
AMZN.set_index('Date', inplace = True)
```

In [201…
```
MSFT.set_index('Date', inplace = True)
```

In [202…
```
AAPL.set_index('Date', inplace = True)
```

In [203…
```
GOOG.set_index('Date', inplace = True)
```

In [204…
```
META.set_index('Date', inplace = True)
```

Before we begin to plot let's ensure we use an appealing style

In [205…
```
style.use('ggplot')
```

Let's combine the data into one table. First we will add a column to each dataframe called 'ticker' then create a new dataframe with data from all the companies.

In [206…
```
AMZN['Ticker'] = 'AMZN';
```

In [207…
```
MSFT['Ticker'] = 'MSFT'
```

In [208…
```
AAPL['Ticker'] = 'AAPL';
```

In [209…
```
GOOG['Ticker'] = 'GOOG';
```

In [210…
```
META['Ticker'] = 'META';
```

Now create a data frame that has all of the data combined to do analysis. Now to analyze specific companies in comparison to each other, you can utilize the group_by function to do so.

In [211…
```
df = pd.concat([AMZN, MSFT, AAPL, GOOG, META])
```

Let's try to draw some conclusions utilizing descriptive statistics. We can use the standard deviation of the data as a general measure of variability.

In [228…
```
df.groupby('Ticker')['Close'].std()
```

Out[228…
```
Ticker
AAPL    18.542762
AMZN    18.927615
GOOG    17.041770
META    70.026641
MSFT    40.872540
Name: Close, dtype: float64
```

It appears that META maintained the most variable closing price, which is likely correlated with its higher share price. Let's analyze the variability of the volume.

In [227…
```
df.groupby('Ticker')['Volume'].std()
```

Out[227…
```
Ticker
AAPL    1.879505e+07
AMZN    2.140757e+07
GOOG    1.067382e+07
META    1.348139e+07
MSFT    1.076050e+07
Name: Volume, dtype: float64
```

It appears that microsoft had the highest variability in volume traded out of any of the stocks.

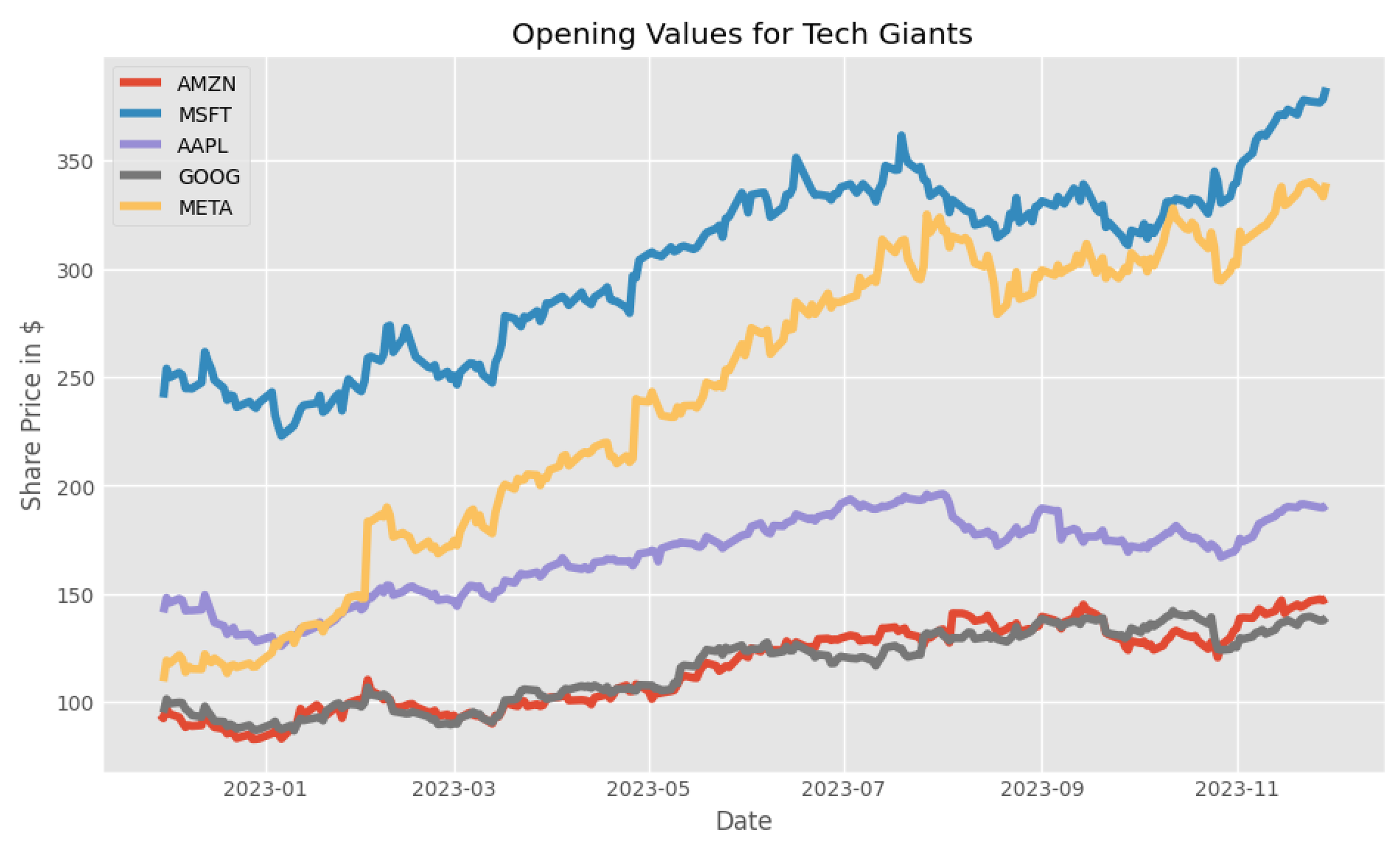
Use the new dataframe to create a plot that compares opening values of our tech giants for the past year

In [214…
```
#ensure the index is of datetime type for proper plotting
df.index = pd.to_datetime(df.index)

#set plot size
plt.figure(figsize = (10,6))

#iterate through each unique ticker and plot
for ticker in df['Ticker'].unique():
    subset = df[df['Ticker'] == ticker]
    plt.plot(subset.index, subset['Open'], label = ticker)

#Make some pretty labels
plt.title('Opening Values for Tech Giants')
plt.xlabel('Date')
plt.ylabel('Share Price in $')
plt.legend()
plt.show();
```

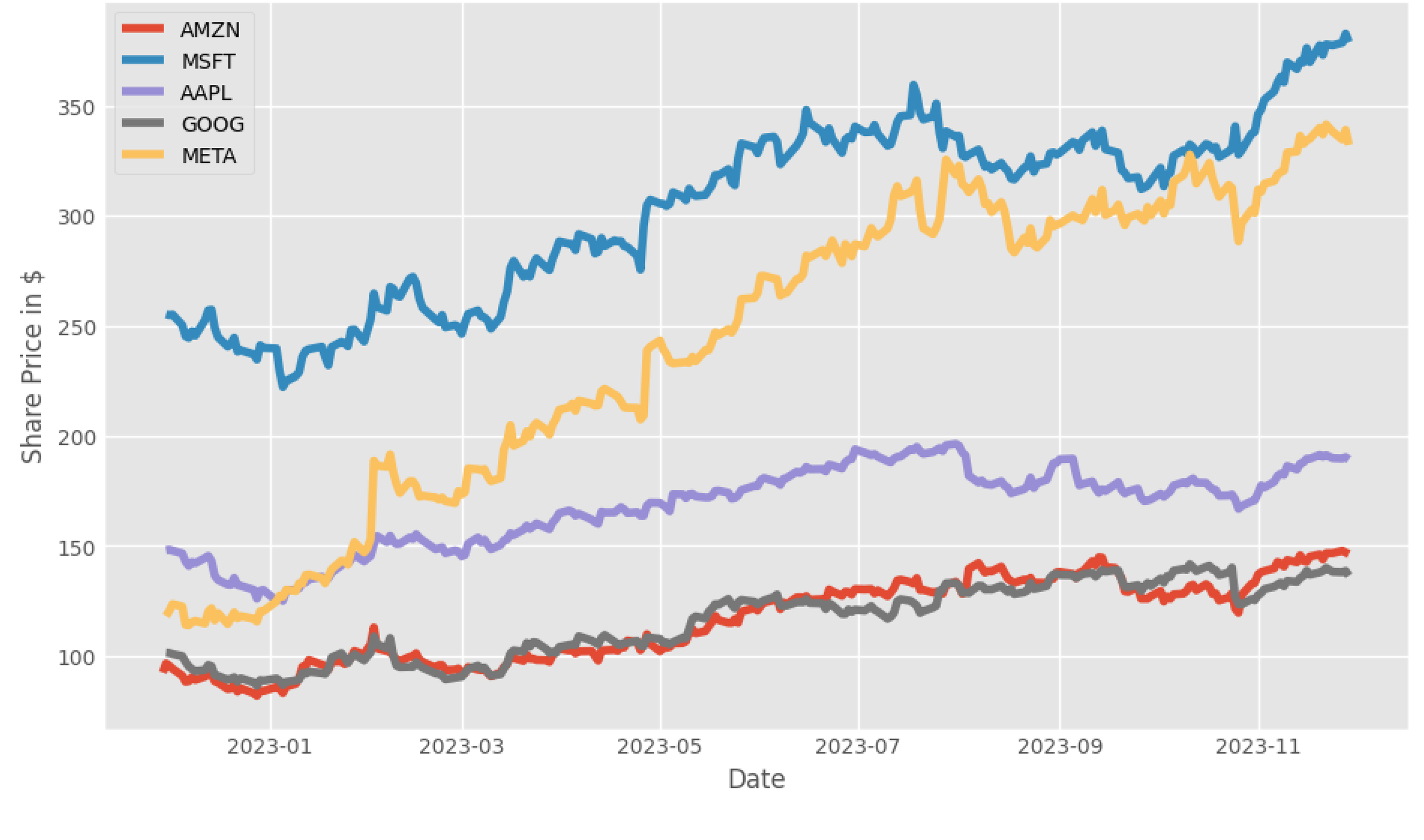


Now let's create a plot that compares the closing values of our tech giants for the last year

In [220…
```
#plot the figure
plt.figure(figsize=(10,6))

#iterate through the unique ticker and plot
for ticker in df['Ticker'].unique():
    subset = df[df['Ticker'] == ticker]
    plt.plot(subset.index, subset['Close'], label = ticker)

#add labels
plt.title('Closing Values for Tech Giants')
plt.xlabel('Date')
plt.ylabel('Share Price in $')
plt.legend()
plt.show();
```



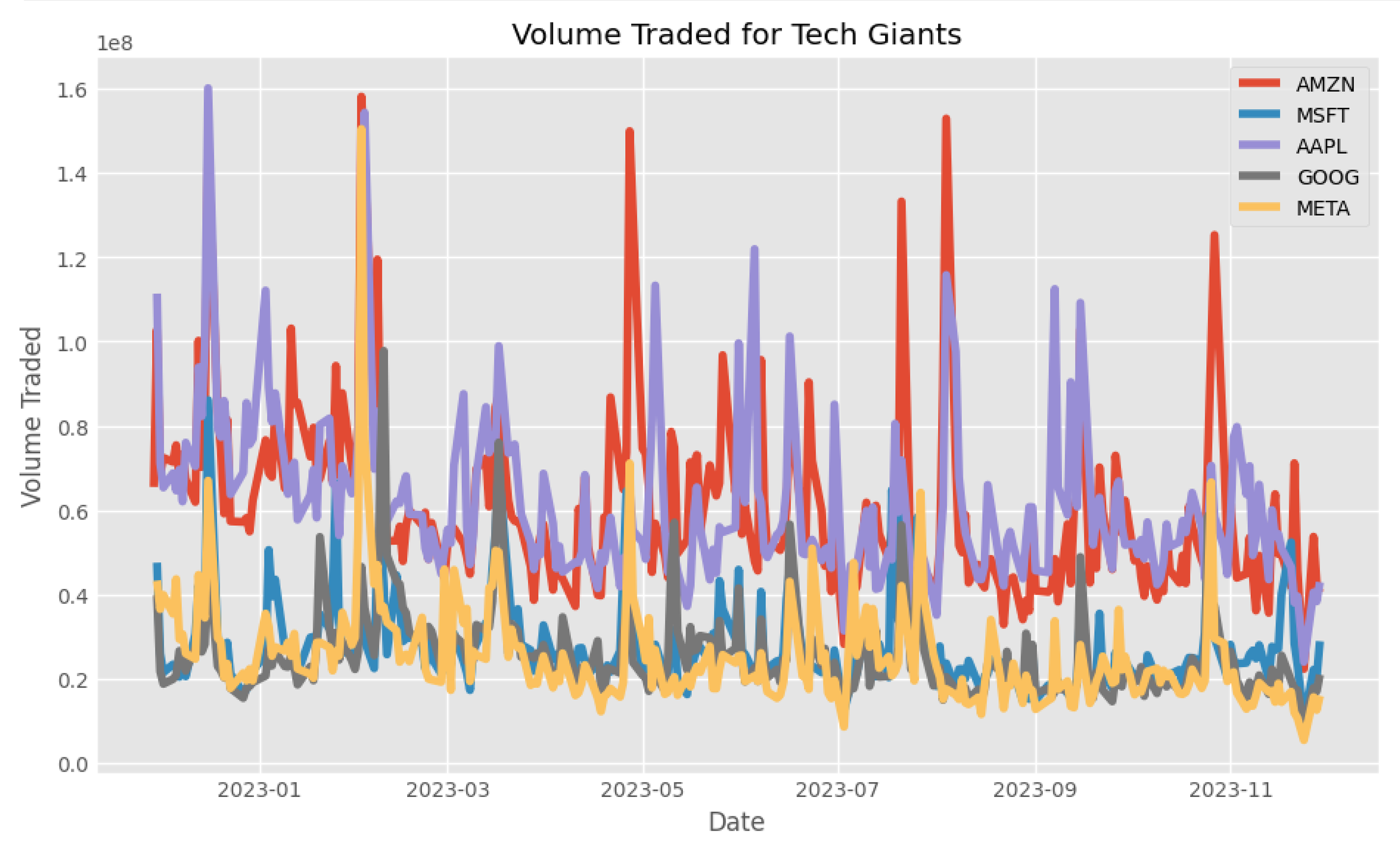
From this data we can devise that the difference in opening and closing share prices are negligible when examining share price over a period as long as a year.

Next, let's examine the volume traded over the course of a year.

In [222…
```
#set the figsize
plt.figure(figsize = (10,6))

#iterate through the unique ticker and plot
for ticker in df['Ticker'].unique():
    subset = df[df['Ticker'] == ticker]
    plt.plot(subset.index, subset['Volume'], label = ticker)

plt.title('Volume Traded for Tech Giants')
plt.xlabel('Date')
plt.ylabel('Volume Traded in Millions')
plt.legend()
plt.show();
```



From this data we can conclude that the peaks in volume traded of all tech stocks are extremely highly correlated as every single tech giant's trading volume follows a similar pattern. This suggests that within the market the share price of technology giants are very highly correlated.

This concludes my brief analysis and demonstration of my skills in for the introduction to python portion of this course. I hope that my demonstration was sufficient.