Visualizing Top 4 Tech Stocks

August 3, 2025

0.1 Visualizing Top 4 Tech Stocks

Project based in analyizing and visualizing the top 4 highest valued technology stocks, as of the end of the first half of the year 2025 :

1. Microsoft (NASDAQ:MSFT)
2. Amazon (NASDAQ:AMZN)
3. Apple (NASDAQ:AAPL)
4. Alphabet (NASDAQ:GOOG)

Using Pandas, yfinance and Matplotlib, will take a visual look into the similarities and differences between these stocks during the six month period from January through June 2025.The analysis covers :

1. Visualize the Stock Prices using Matplotlib
2. Calculate and Visualize the Daily Simple Rate of Return
3. Calculate and Visualize the Mean Rates of Return
4. Calculate and Visualize the Variances of the Returns
5. Calculate and Visualize the Standard Deviations of the Returns
6. Correlations between the Tech Stocks

0.1.1 1. Visualize the Stock Prices using Matplotlib

Step (I): Import Data Manipulation Packages Import the pandas and numpy module as `pd` and `np`, respectively

[2]:

```
import pandas as pd
import numpy as np
import warnings
from pandas.errors import SettingWithCopyWarning
warnings.simplefilter(action="ignore", category=FutureWarning)
warnings.simplefilter(action="ignore", category=SettingWithCopyWarning)
```

Step (II): Import Financial Package Import the Library yahoo finance as `yf`

[5]:

```
import yfinance as yf
```

Step (III): Import Visualization Package Import the matplotlib `pyplot` module as `plt`.

[3]:

```
import matplotlib.pyplot as plt
%matplotlib inline
```

Step (IV): Define Stocks Create a list named `Tech` containing the symbols for the top 4 tech stocks.

Step (V): Create Dates Create a datetime object representing January 1st, 2025 named `start_date` and a datetime object representing July 1st, 2025 named `end_date`.

Step (VI): Retrieve Data Call the function `download` with arguments `Tech`, `start_date` and `end_date` and save the result to `stock_data`.

Step (VII): View Data View both `stock_data` and `stock_data['Close']`. What information is stored in these DataFrames?

[6]:

```
# Defining Stocks
Tech = ['AAPL', 'AMZN', 'GOOG', 'MSFT']

#Creating dates
start_date ='2025-01-01'
end_date ='2025-07-01'

#Retriving data

stock_data = yf.download("AAPL AMZN GOOG␣
 ↪MSFT",start="2025-01-01",end="2025-07-01",progress=False)

#Viewing data

stock_data
```

[6]:

```
Price            Close                                               High  \
Ticker            AAPL        AMZN        GOOG        MSFT        AAPL
Date
2025-01-02  243.263199  220.220001  190.184464  416.976868  248.500565
2025-01-03  242.774368  224.190002  192.678635  421.728607  243.592387
2025-01-06  244.410416  227.610001  197.497345  426.211365  246.734810
2025-01-07  241.627136  222.110001  196.250259  420.752350  244.959095
2025-01-08  242.115952  222.130005  194.933350  422.933990  243.123531
…                    …           …           …           …           …
2025-06-24  200.300003  212.770004  167.740005  490.109985  203.440002
2025-06-25  201.559998  211.990005  171.490005  492.269989  203.669998
2025-06-26  201.000000  217.119995  174.429993  497.450012  202.639999
2025-06-27  201.080002  223.300003  178.270004  495.940002  203.220001
2025-06-30  205.169998  219.389999  177.389999  497.410004  207.389999

Price                                                  Low              \
Ticker            AMZN        GOOG        MSFT        AAPL        AMZN
Date
2025-01-02  225.149994  192.748449  424.438202  241.238085  218.190002
2025-01-03  225.360001  194.045428  422.405996  241.307905  221.619995
2025-01-06  228.839996  199.093596  432.656586  242.614744  224.839996
2025-01-07  228.380005  201.667561  429.000637  240.769205  221.460007
2025-01-08  223.520004  197.178091  425.334764  239.472335  220.199997
…                  …          …          …          …          …
2025-06-24  214.339996  169.250000  491.850006  200.199997  211.050003
2025-06-25  216.029999  173.360001  494.559998  200.619995  211.110001
2025-06-26  218.039993  174.649994  498.040009  199.460007  212.009995
2025-06-27  223.300003  178.839996  499.299988  200.000000  216.740005
2025-06-30  223.820007  181.580002  500.760010  199.259995  219.119995

Price                                 Open                           \
Ticker            GOOG        MSFT        AAPL        AMZN        GOOG
Date
2025-01-02  188.268953  413.261173  248.330961  222.029999  191.037461
2025-01-03  190.902796  417.933202  242.774368  222.509995  192.274582
2025-01-06  194.604114  423.850447  243.722074  226.779999  194.693900
2025-01-07  195.482055  419.188356  242.395272  227.899994  197.806611
2025-01-08  193.297183  419.925568  241.337830  223.190002  193.496713
…                  …          …          …          …          …
2025-06-24  166.910004  486.799988  202.589996  212.139999  167.684998
2025-06-25  168.561005  489.390015  201.449997  214.619995  168.649994
2025-06-26  170.860001  492.809998  201.429993  213.119995  173.384995
2025-06-27  172.710007  493.029999  201.889999  219.919998  174.630005
2025-06-30  175.500000  495.329987  202.009995  223.520004  180.785004

Price                     Volume
Ticker            MSFT      AAPL       AMZN      GOOG      MSFT
Date
2025-01-02  423.900262  55740700   33956600  17545200  16896500
2025-01-03  419.467282  40244100   27515600  12875000  16662900
2025-01-06  426.360784  45045600   31849800  19483300  20573600
2025-01-07  427.356962  40856000   28084200  16966800  18139100
2025-01-08  421.838197  37628900   25033300  14335300  15054600
…                  …        …         …        …        …
2025-06-24  488.950012  54064000   38378800  27310300  22305600
2025-06-25  492.040009  39525700   31755700  23627400  17495100
2025-06-26  492.980011  50799100   50480800  25909100  21578900
2025-06-27  497.549988  73188600  119217100  74053100  34539200
2025-06-30  497.040009  91912800   58887800  43964600  28369000

[122 rows x 20 columns]
```

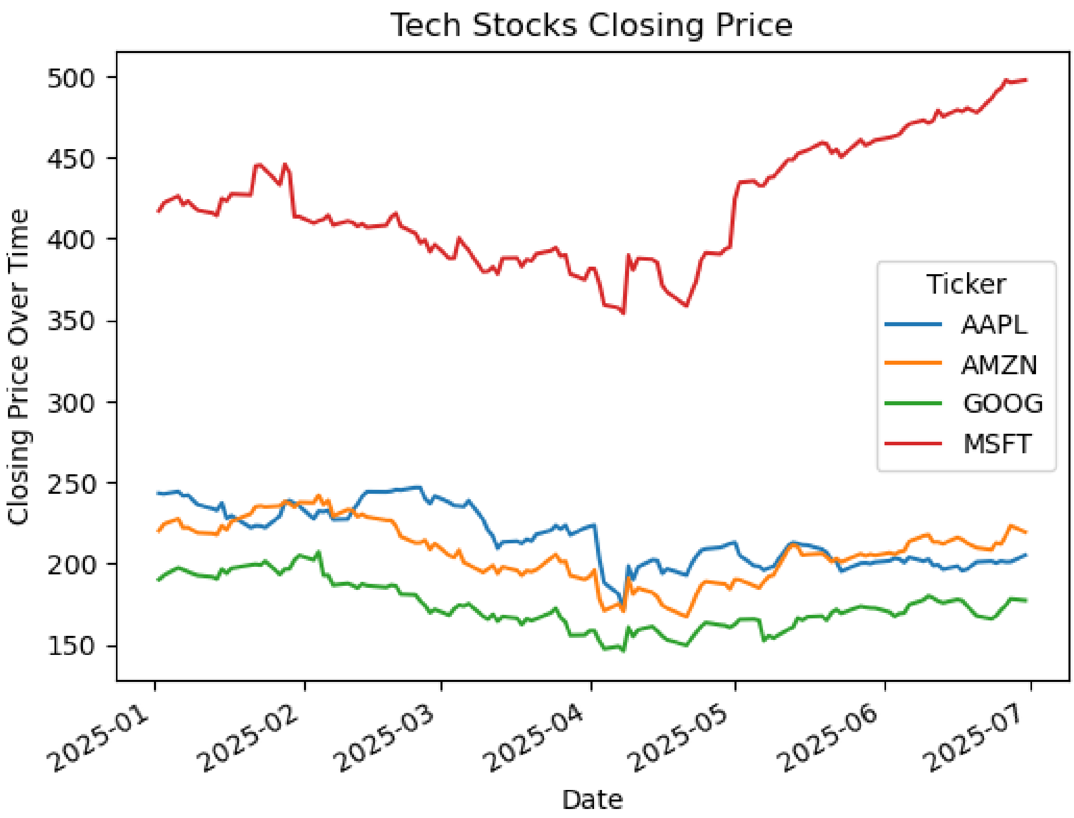
[11]:

```
stock_data['Close']
```

```
[11]: Ticker            AAPL        AMZN        GOOG        MSFT
      Date
      2025-01-02  243.263199  220.220001  190.184464  416.976868
      2025-01-03  242.774368  224.190002  192.678635  421.728607
      2025-01-06  244.410416  227.610001  197.497345  426.211365
      2025-01-07  241.627136  222.110001  196.250259  420.752350
      2025-01-08  242.115952  222.130005  194.933350  422.933990
      …                  …         …         …         …
      2025-06-24  200.300003  212.770004  167.740005  490.109985
      2025-06-25  201.559998  211.990005  171.490005  492.269989
      2025-06-26  201.000000  217.119995  174.429993  497.450012
      2025-06-27  201.080002  223.300003  178.270004  495.940002
      2025-06-30  205.169998  219.389999  177.389999  497.410004

      [122 rows x 4 columns]
```

Step (VIII): Plot the adjusted closing prices over time. Create a plot with matplotlib that shows the adjusted closing prices of each stock over time. Set the x label to `"Date"`. Set the y label to `"Closing Price Over Time"`. Set the graph title to `"Tech Stocks Closing Price"`.

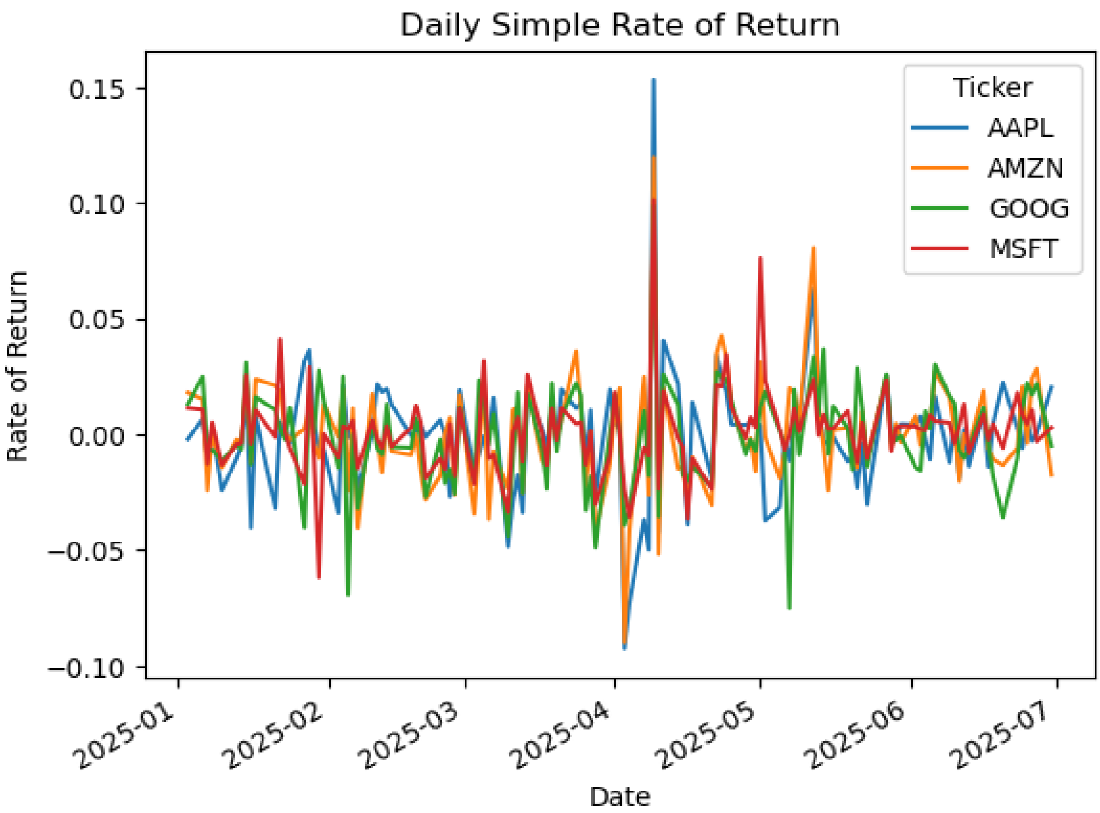
```
[7]: adj_date = stock_data["Close"]
     adj_date.plot()
     plt.title("Tech Stocks Closing Price")
     plt.xlabel("Date")
     plt.ylabel("Closing Price Over Time")
     plt.show()
```



0.1.2 2. Calculate and Visualize the Daily Simple Rate of Return

Create a plot with matplotlib that shows the daily simple rate of return for each tech stock over time. Label the graph appropriately.

[8]:
```
daily_return = adj_date.pct_change()

daily_return.plot()
plt.title("Daily Simple Rate of Return")
plt.xlabel("Date")
plt.ylabel("Rate of Return")
plt.show()
```



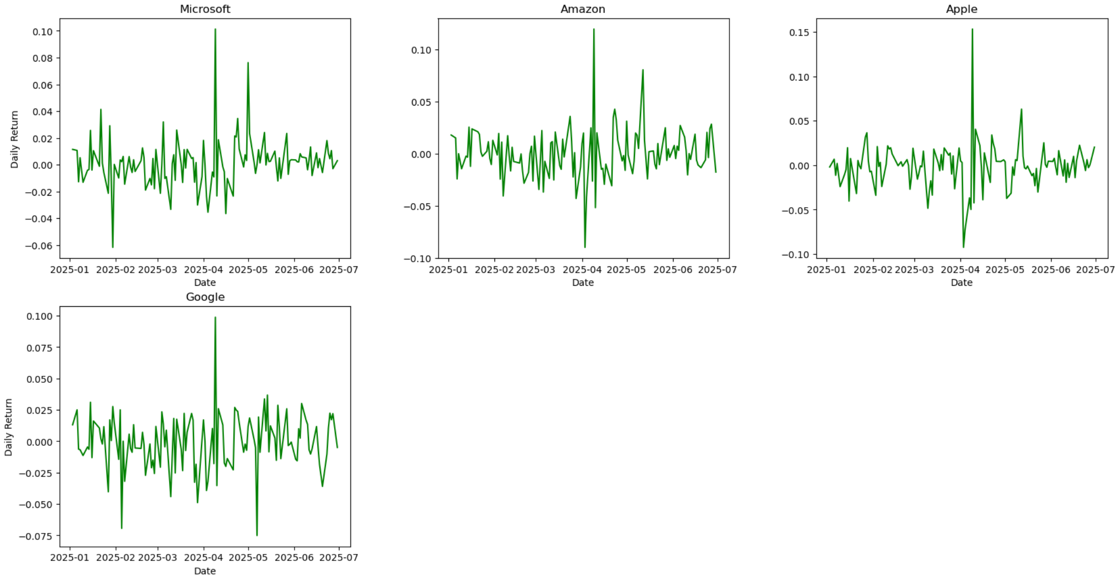
```
[9]: print(daily_return.head())
```

```
Ticker          AAPL      AMZN      GOOG      MSFT
Date
2025-01-02       NaN       NaN       NaN       NaN
2025-01-03 -0.002009  0.018027  0.013114  0.011396
2025-01-06  0.006739  0.015255  0.025009  0.010629
2025-01-07 -0.011388 -0.024164 -0.006314 -0.012808
2025-01-08  0.002023  0.000090 -0.006710  0.005185
```

Create subplots of daily simple rate of return. In order to better visualize the daily returns, create a subplot for each tech stock.

```
[10]: fig = plt.figure(figsize=(20, 10))


#Microsoft
ax1 = plt.subplot(2, 3, 1)
plt.plot(daily_return['MSFT'], color='green')
plt.title('Microsoft')
plt.xlabel('Date')
```

```python
plt.ylabel('Daily Return')

#Amazon
ax2 = plt.subplot(2, 3, 2)
plt.plot(daily_return['AMZN'], color='green')
plt.title('Amazon')
plt.xlabel('Date')


#Apple
ax3 = plt.subplot(2, 3, 3)
plt.plot(daily_return['AAPL'], color='green')
plt.title('Apple')
plt.xlabel('Date')


#Google
ax4 = plt.subplot(2, 3, 4)
plt.plot(daily_return['GOOG'], color='green')
plt.title('Google')
plt.xlabel('Date')
plt.ylabel('Daily Return')

plt.subplots_adjust(wspace=0.3, bottom=0.1)

plt.show()
```



0.1.3 3. Calculate and Visualize the Mean Rates of Return

Step (I): Calculate mean rate of return For each stock, calculate the mean daily simple rate of return.

[11]:

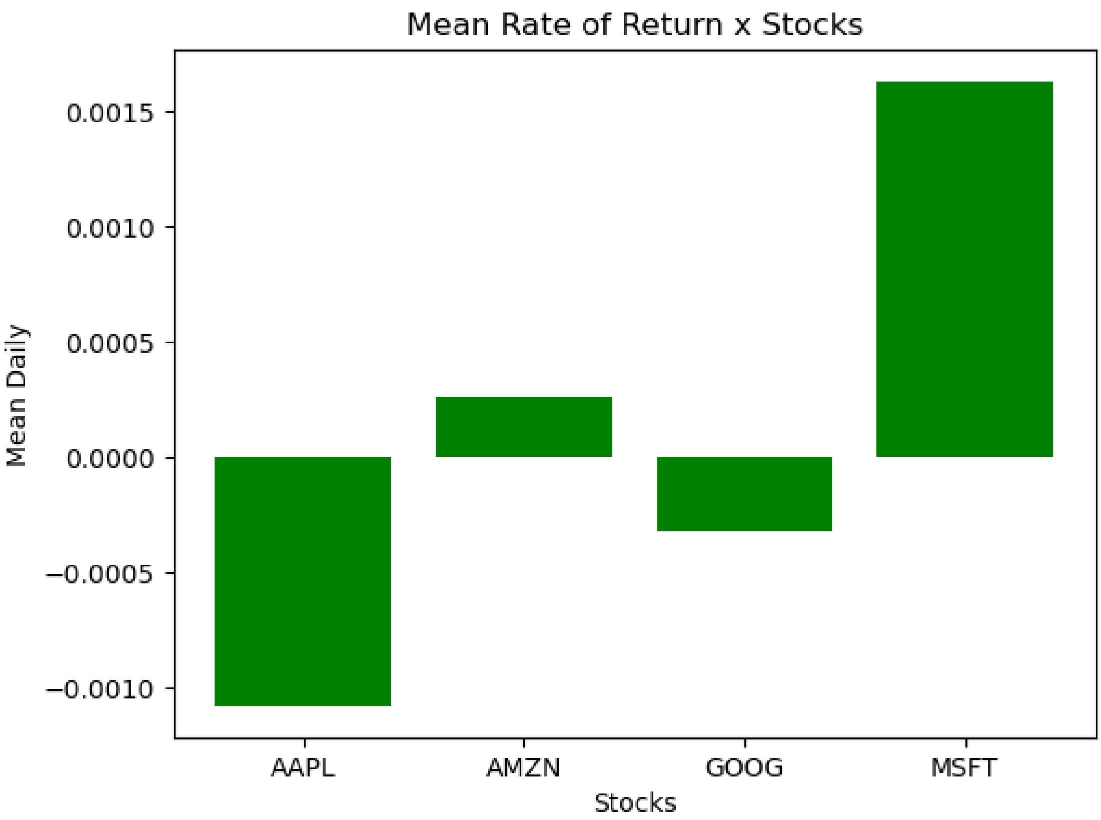
```
#Calculating the mean rate of return

mean_daily_return = daily_return.mean()
print(mean_daily_return)
```

```
Ticker
AAPL   -0.001081
AMZN    0.000257
GOOG   -0.000323
MSFT    0.001630
dtype: float64
```

Step (II): Plot bar chart

[12]:

```
#Plotting bar chart

ax7 = plt.subplot()
ax7.set_xticks(range(len(Tech)))
ax7.set_xticklabels(Tech)

plt.bar(range(len(Tech)), mean_daily_return, color = 'green')

plt.xlabel('Stocks')
plt.ylabel('Mean Daily')
plt.title('Mean Rate of Return x Stocks')

plt.show()
```



Step (III): Analyse the mean Q : Based on the mean daily rate of return of stocks, which stock would be the best/worst stock to invest in ?

Microsoft stock has the highest mean return rate and is the best option for investment. Conversely, Apple has the lowest mean simple rate of return and wouldn't be the best choice of investment based on the mean over this period of time.

0.1.4 4. Calculate and Visualize the Variance.

Step (I): Calculate the variance For each stock, calculate the variance of the mean daily simple rate of return.

[13]:
```
#Calculating the variance

variance_daily_return = daily_return.var()
print(variance_daily_return)
```

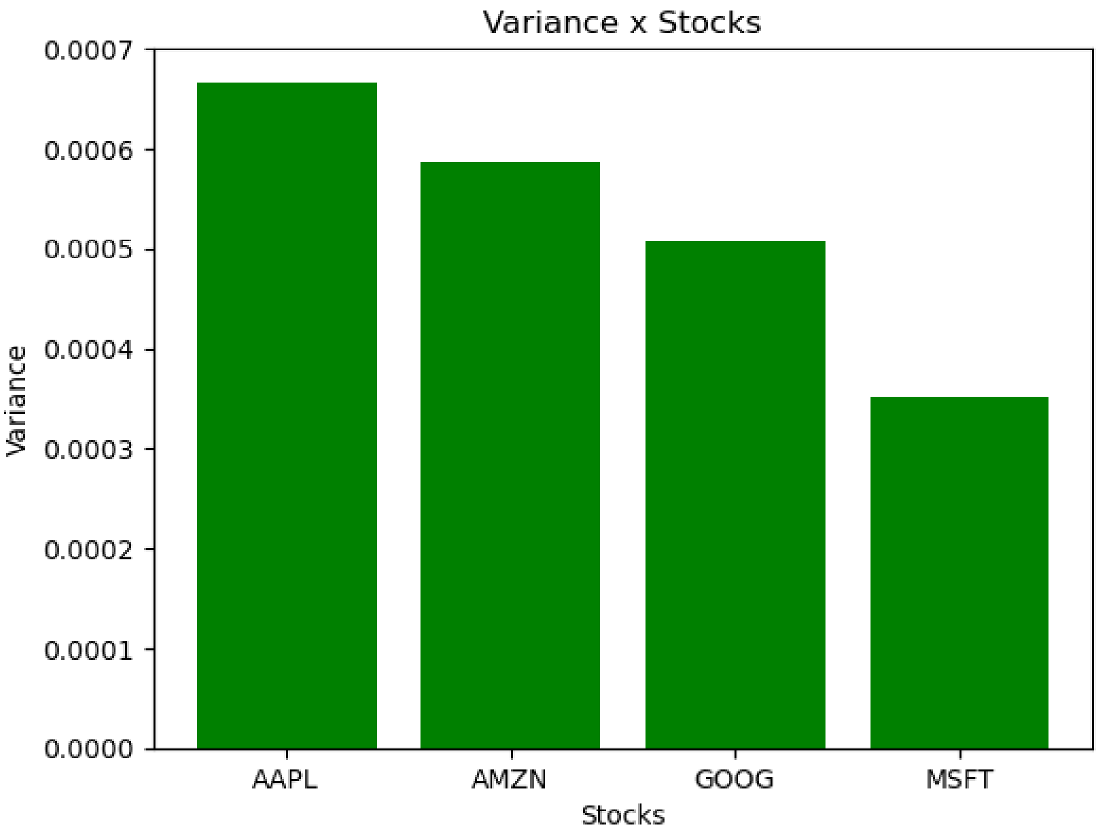
```
Ticker
AAPL    0.000667
AMZN    0.000587
GOOG    0.000508
```

```
MSFT    0.000352
dtype: float64
```

Step (II): Plot bar chart Use matplotlib to create a bar chart comparing the variance for each stock. Label the chart appropriately

[14]:

```
#Plotting bar chart
ax8 = plt.subplot()
ax8.set_xticks(range(len(Tech)))
ax8.set_xticklabels(Tech)

plt.bar(range(len(Tech)), variance_daily_return, color = 'green')

plt.xlabel('Stocks')
plt.ylabel('Variance')
plt.title('Variance x Stocks')

plt.show()
```



Step (III): Analyse the variance Q : Based on the variance, which stock would be the riskiest to invest in?

Apple has the highest variance and thus is the riskest stock in this time period.

0.1.5 5. Calculate and Visualize the Standard Deviation

Step (I): Calculate the standard deviation For each stock, calculate the standard deviation of the mean daily simple rate of return.

[26]:

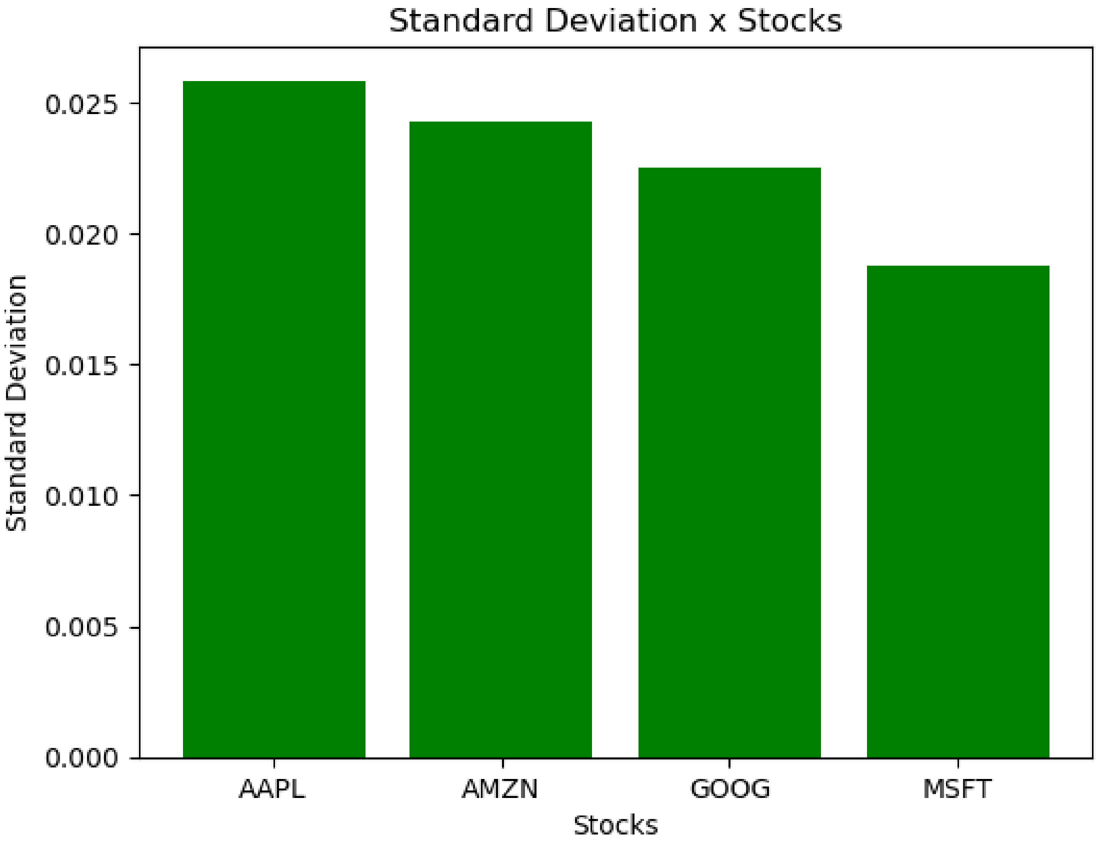
```python
#Calculating Standard Deviation

sd_daily_return = daily_return.std()
print(sd_daily_return)
```

```
Ticker
AAPL    0.025821
AMZN    0.024233
GOOG    0.022530
MSFT    0.018767
dtype: float64
```

Step (II): Plot the bar chart Use matplotlib to create a bar chart comparing the standard deviation of the mean daily simple rate of return of each stock. Label the chart appropriately

[25]:

```python
#Plotting bar chart
ax9 = plt.subplot()
ax9.set_xticks(range(len(Tech)))
ax9.set_xticklabels(Tech)

plt.bar(range(len(Tech)), sd_daily_return, color = 'green')

plt.xlabel('Stocks')
plt.ylabel('Standard Deviation')
plt.title('Standard Deviation x Stocks')

plt.show()
```



Step (III): Analyze the standard deviation Q : Based on the standard deviation of the rates of return, which stock would you choose to invest in and why?

I would choose to invest in Microsoft stock since it has lowest variance and standard variance. Market currently is violatile with geopolitical dynamics adversely affecting stocks i.e the international “Trade War” of tariffs has resulted in uncertainity in global markets that is exhibited in by violatility in market prices. Can see from the data that Apple is mostly affected since it’s business model is highly integrated in global trade with China .China and US trade deal recently saw the highest tariff raise which has negatively affected stock prices of Apple as it imports most of it’s manufacturing components for products from China.

0.1.6 6. Calculate the Correlations

Step (I) Calculate the correlations between each of the stocks.

[15]:
```
#Calculating Correlation

correlation= daily_return.corr()
print(correlation)
```

```
Ticker      AAPL      AMZN      GOOG      MSFT
Ticker
```

```
AAPL    1.000000  0.697985  0.561548  0.619242
AMZN    0.697985  1.000000  0.710832  0.738278
GOOG    0.561548  0.710832  1.000000  0.599448
MSFT    0.619242  0.738278  0.599448  1.000000
```

Q : Which stocks are positively correlated? Which are negatively correlated? Which have highest correlation?

All stockes are positvely correlated. No stock is negatively correlated. Microsoft and Amazon having the highest correlation at 0.738278