A Simple Statistical Analysis of the Top 5 Tech Stocks MAMAA

Here are the current top 5 tech stocks of the first half of 2022 (In no particular order)

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

The 6 month period we are focused on is January-June 2022, here are the objectives of this project

1. Visualize the tech stocks using matplotlib
   - matplotlib = graph plotting library in Python to create interactive visualizations
2. Calculate and visualize the daily simple rate of return (ROR)
3. Calculate and visualize the mean ROR
4. Calculate and visualize the variance of the returns
5. Calculate and visualize the correlation of the returns

First, we must import some packages into Python for some data

In [ ]:

```
import pandas_datareader as web
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

Retrieving the Adjusting Closings for the Top 5 Tech Stocks

Define the stocks, create dates, and retrieve and view the data within

In [ ]:

```
symbols = ['MSFT', 'AMZN', 'AAPL', 'GOOG', 'META']

#Create dates
start_date = '2022-01-01'
end_date = '2022-07-01'

#Retrieve the data
stock_data = web.get_data_yahoo(symbols, start_date, end_date)

#Viewing the data
stock_data
```

Out[ ]:

Attributes	Adj Close							
Symbols	**MSFT**	**AMZN**	**AAPL**	**GOOG**	**META**	**MSFT**	**AMZN**	
Date								
2022-01-03	333.283020	170.404495	181.511703	145.074493	338.540009	334.750000	170.404495	182.0
2022-01-04	327.568176	167.522003	179.208038	144.416504	336.529999	329.010010	167.522003	179.6
2022-01-05	314.993530	164.356995	174.441116	137.653503	324.170013	316.380005	164.356995	174.9
2022-01-06	312.504486	163.253998	171.529129	137.550995	332.459991	313.880005	163.253998	172.0
2022-01-07	312.663788	162.554001	171.698654	137.004501	331.790009	314.040009	162.554001	172.1
...	...	...	...	...	...	...	...	
2022-06-27	264.890015	113.220001	141.660004	116.622498	169.490005	264.890015	113.220001	141.6
2022-06-28	256.480011	107.400002	137.440002	112.571503	160.679993	256.480011	107.400002	137.4
2022-06-29	260.260010	108.919998	139.229996	112.256500	163.940002	260.260010	108.919998	139.2
2022-06-30	256.829987	106.209999	136.720001	109.372498	161.250000	256.829987	106.209999	136.7
2022-07-01	259.579987	109.559998	138.929993	109.081001	160.029999	259.579987	109.559998	138.9

125 rows × 30 columns

Woa, thats a lot of data. Let's simplify it by finding just the adjusted closing prices

In [ ]:

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

Out[ ]:

Symbols	MSFT	AMZN	AAPL	GOOG	META
Date					
2022-01-03	333.283020	170.404495	181.511703	145.074493	338.540009
2022-01-04	327.568176	167.522003	179.208038	144.416504	336.529999
2022-01-05	314.993530	164.356995	174.441116	137.653503	324.170013
2022-01-06	312.504486	163.253998	171.529129	137.550995	332.459991
2022-01-07	312.663788	162.554001	171.698654	137.004501	331.790009
...	...	...	...	...	...
2022-06-27	264.890015	113.220001	141.660004	116.622498	169.490005
2022-06-28	256.480011	107.400002	137.440002	112.571503	160.679993
2022-06-29	260.260010	108.919998	139.229996	112.256500	163.940002
2022-06-30	256.829987	106.209999	136.720001	109.372498	161.250000
2022-07-01	259.579987	109.559998	138.929993	109.081001	160.029999

125 rows × 5 columns

1. Visualize the tech stocks using matplotlib

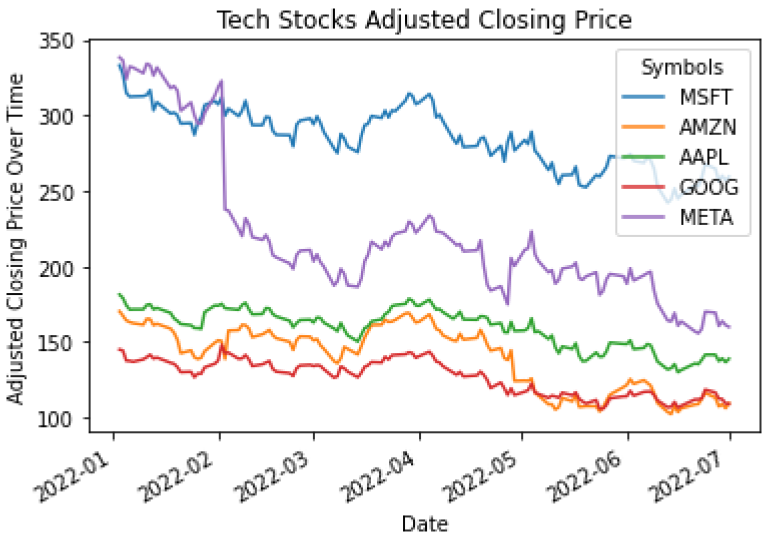
Add some visualization of the data that was just collected

In [ ]:

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

Out[ ]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



2. Calculate and visualize the daily simple ROR

Plot the daily simple ROR for each of the tech stocks over time

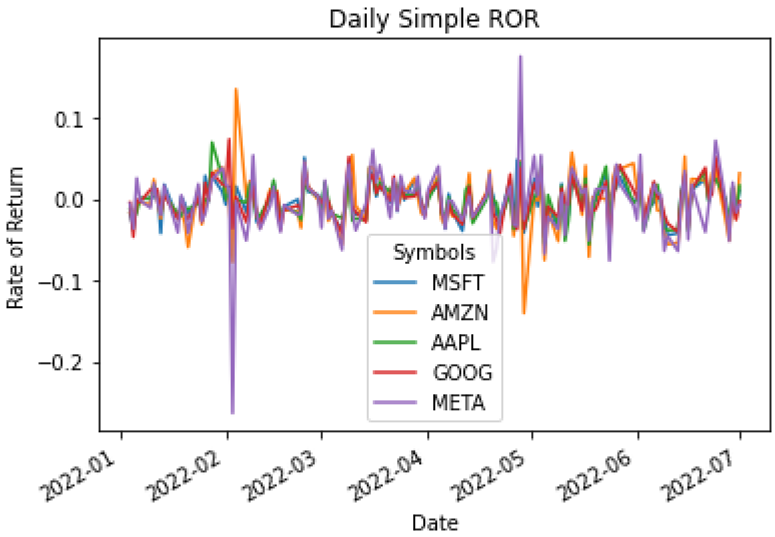
In [ ]:

```
daily_return = adj_date.pct_change()

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

Out[ ]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



This graph is a little difficult to accurately read, bring it back to table form

In [ ]:

```
daily_return.head()
```

Out[ ]:

Symbols	MSFT	AMZN	AAPL	GOOG	META
Date					
2022-01-03	NaN	NaN	NaN	NaN	NaN
2022-01-04	-0.017147	-0.016916	-0.012692	-0.004536	-0.005937
2022-01-05	-0.038388	-0.018893	-0.026600	-0.046830	-0.036728
2022-01-06	-0.007902	-0.006711	-0.016693	-0.000745	0.025573
2022-01-07	0.000510	-0.004288	0.000988	-0.003973	-0.002015

Seperating the graph also helps with accurate readings

In [ ]:

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

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

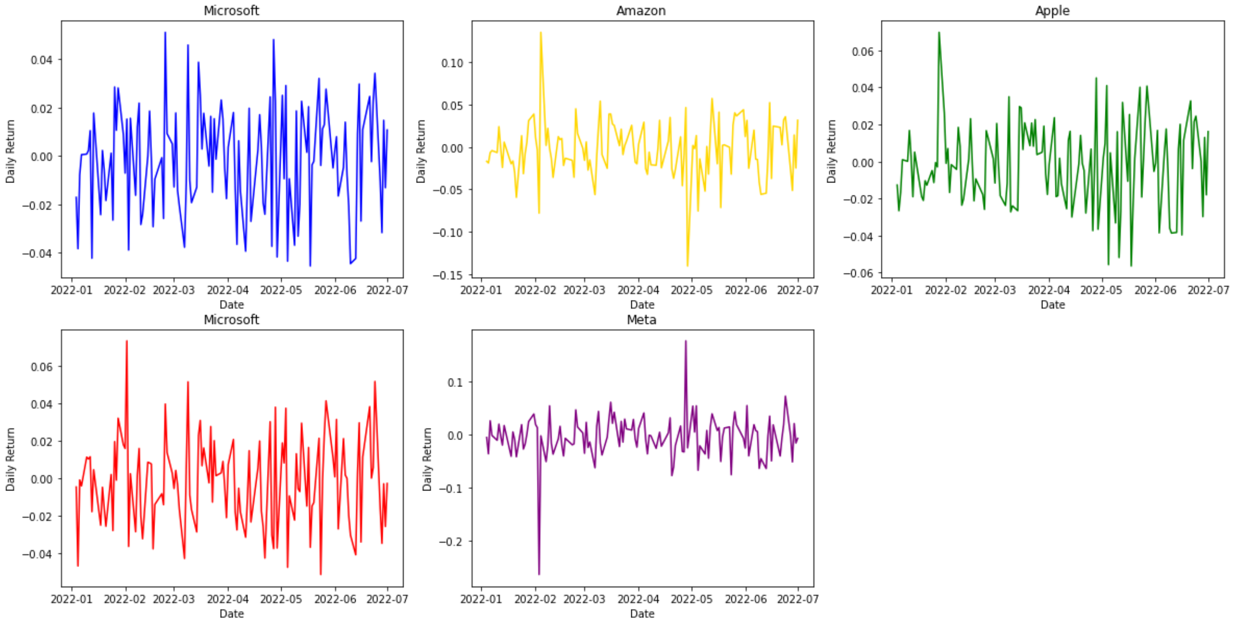
```
#Amazon graph
ax1 = plt.subplot(2, 3, 2)
plt.plot(daily_return['AMZN'], color='gold')
plt.title('Amazon')
plt.xlabel('Date')
plt.ylabel('Daily Return')

#Apple graph
ax1 = plt.subplot(2, 3, 3)
plt.plot(daily_return['AAPL'], color='green')
plt.title('Apple')
plt.xlabel('Date')
plt.ylabel('Daily Return')

#Google graph
ax1 = plt.subplot(2, 3, 4)
plt.plot(daily_return['GOOG'], color='red')
plt.title('Microsoft')
plt.xlabel('Date')
plt.ylabel('Daily Return')

#Meta graph
ax1 = plt.subplot(2, 3, 5)
plt.plot(daily_return['META'], color='purple')
plt.title('Meta')
plt.xlabel('Date')
plt.ylabel('Daily Return')
```

```
Out[ ]: Text(0, 0.5, 'Daily Return')
```



3. Calculate and visualize the mean ROR

Create a bar chart to compare the mean daily simple ROR for each stock

```
In [ ]: mean_daily_return = daily_return.mean()
        print(mean_daily_return)
```

```
Symbols
MSFT   -0.001767
AMZN   -0.002986
AAPL   -0.001902
GOOG   -0.002015
META   -0.005092
dtype: float64
```

In [ ]:

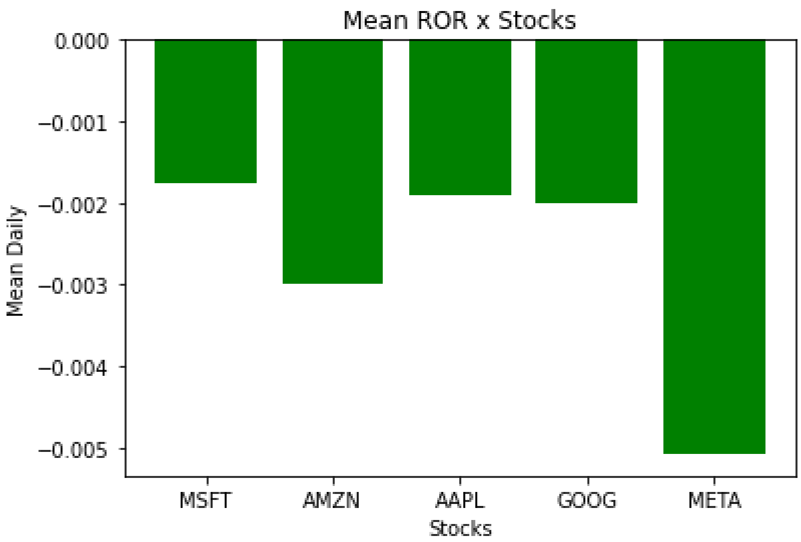
```
#Plotting the bar chart

ax5 = plt.subplot()
ax5.set_xticks(range(len(symbols)))
ax5.set_xticklabels(symbols)

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

plt.xlabel('Stocks')
plt.ylabel('Mean Daily')
plt.title('Mean ROR x Stocks')

plt.show()
```



As you can see from the table and chart above, Microsoft has the highest mean daily simple ROR. This would mean Microsoft would be the best investment choice over the period of time. Alternatively, Meta has the lowest mean simple ROR, so this would be the worst investment choice over the period of time.

4. Calculate and visualize the variance of the returns

In [ ]:

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

```
Symbols
MSFT    0.000496
AMZN    0.001138
AAPL    0.000506
GOOG    0.000570
META    0.001781
dtype: float64
```

In [ ]:

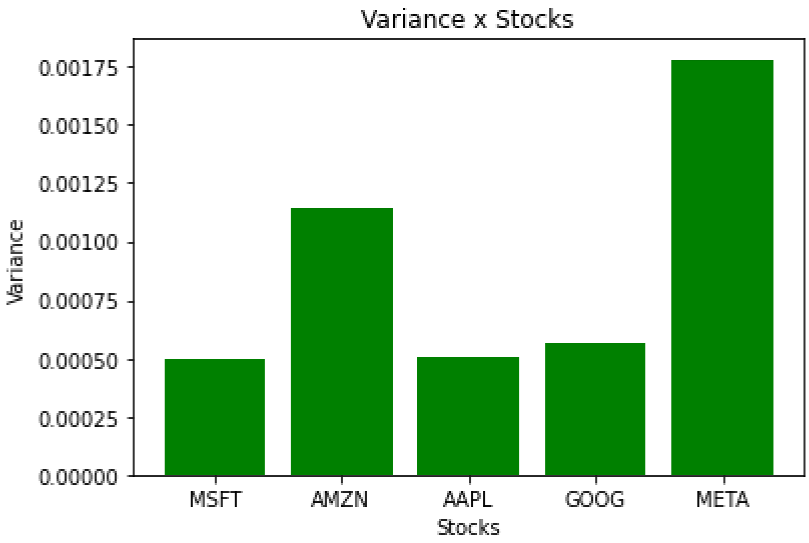
```
#Plotting the bar chart

ax6 = plt.subplot()
ax6.set_xticks(range(len(symbols)))
ax6.set_xticklabels(symbols)

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

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

plt.show()
```



As per the table and the graph above, Meta shows the highest variance of all the stocks. Microsoft shows the lowest variance, indicating that the returns are more predictable.

5. Calculate and visualize the correlation of the returns

In [ ]:

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

```
Symbols      MSFT      AMZN      AAPL      GOOG      META
Symbols
MSFT     1.000000  0.730538  0.838292  0.799962  0.637945
AMZN     0.730538  1.000000  0.694366  0.707870  0.628862
AAPL     0.838292  0.694366  1.000000  0.799650  0.615430
GOOG     0.799962  0.707870  0.799650  1.000000  0.701772
META     0.637945  0.628862  0.615430  0.701772  1.000000
```

None of the stocks are negatively correlated. Microsoft and Apple are highly correlated, while Meta and Apple show the lowest correlation