Mini Project 1-Stock Market Analysis Using Python

In [1]:
```
# Importing libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
# Read the Data from Yahoo finance website directly.
# we downloaded the data
```

In [3]:
```
amazon = pd.read_csv(r"C:\Users\umang\Desktop\edureka\mini project -1\amazon.csv")
print(amazon.head())
```

```
         Date       Open       High        Low      Close  Adj Close    Volume
0  2022-12-27  84.970001  85.349998  83.000000  83.040001  83.040001  57284000
1  2022-12-28  82.800003  83.480003  81.690002  81.820000  81.820000  58228600
2  2022-12-29  82.870003  84.550003  82.550003  84.180000  84.180000  54995900
3  2022-12-30  83.120003  84.050003  82.470001  84.000000  84.000000  62401200
4  2023-01-03  85.459999  86.959999  84.209999  85.820000  85.820000  76706000
```

In [4]:
```
microsoft = pd.read_csv(r"C:\Users\umang\Desktop\edureka\mini project -1\microsoft.csv")
print(microsoft.head())
```

```
         Date        Open        High         Low       Close   Adj Close  \
0  2022-12-27  238.699997  238.929993  235.830002  236.960007  234.877350
1  2022-12-28  236.889999  239.720001  234.169998  234.529999  232.468719
2  2022-12-29  235.649994  241.919998  235.649994  241.009995  238.891769
3  2022-12-30  238.210007  239.960007  236.660004  239.820007  237.712234
4  2023-01-03  243.080002  245.750000  237.399994  239.580002  237.474350

     Volume
0  16688600
1  17457100
2  19770700
3  21938500
4  25740000
```

In [5]:
```
google = pd.read_csv(r"C:\Users\umang\Desktop\edureka\mini project -1\google.csv")
print(google.head())
```

```
         Date       Open       High        Low      Close  Adj Close    Volume
0  2022-12-27  89.309998  89.500000  87.535004  87.930000  87.930000  15470900
1  2022-12-28  87.500000  88.519997  86.370003  86.459999  86.459999  17879600
2  2022-12-29  87.029999  89.364998  86.989998  88.949997  88.949997  18280700
3  2022-12-30  87.364998  88.830002  87.029999  88.730003  88.730003  19190300
4  2023-01-03  89.830002  91.550003  89.019997  89.699997  89.699997  20738500
```

In [6]:
```
apple = pd.read_csv(r"C:\Users\umang\Desktop\edureka\mini project -1\apple.csv")
print(apple.head())
```

```
         Date        Open        High         Low       Close   Adj Close  \
0  2022-12-27  131.380005  131.410004  128.720001  130.029999  129.307251
1  2022-12-28  129.669998  131.029999  125.870003  126.040001  125.339409
2  2022-12-29  127.989998  130.479996  127.730003  129.610001  128.889587
3  2022-12-30  128.410004  129.949997  127.430000  129.929993  129.207779
4  2023-01-03  130.279999  130.899994  124.169998  125.070000  124.374802

      Volume
0   69007800
1   85438400
2   75703700
3   77034200
4  112117500
```

In [7]:
```
# Perform cleaning.
```

In [8]:
```
# checking for null values
amazon.isnull().sum()
microsoft.isnull().sum()
google.isnull().sum()
apple.isnull().sum()
```

Out[8]:
```
Date         0
Open         0
High         0
Low          0
Close        0
Adj Close    0
Volume       0
dtype: int64
```

In [9]:
```
# checking for duplicate values
amazon.duplicated().sum()
microsoft.duplicated().sum()
google.duplicated().sum()
apple.duplicated().sum()
```

Out[9]:
```
0
```

In [10]:
```
#creating a new column in dataframe for month
amazon['Date'] = pd.to_datetime(amazon['Date'])
amazon['Month'] = amazon['Date'].dt.month
amazon.head()
```

Out[10]:

	Date	Open	High	Low	Close	Adj Close	Volume	Month
0	2022-12-27	84.970001	85.349998	83.000000	83.040001	83.040001	57284000	12
1	2022-12-28	82.800003	83.480003	81.690002	81.820000	81.820000	58228600	12
2	2022-12-29	82.870003	84.550003	82.550003	84.180000	84.180000	54995900	12
3	2022-12-30	83.120003	84.050003	82.470001	84.000000	84.000000	62401200	12
4	2023-01-03	85.459999	86.959999	84.209999	85.820000	85.820000	76706000	1

In [11]:
```
#creating a new column in dataframe for month
microsoft['Date'] = pd.to_datetime(microsoft['Date'])
microsoft['Month'] = microsoft['Date'].dt.month
microsoft.head()
```

Out[11]:

	Date	Open	High	Low	Close	Adj Close	Volume	Month
0	2022-12-27	238.699997	238.929993	235.830002	236.960007	234.877350	16688600	12
1	2022-12-28	236.889999	239.720001	234.169998	234.529999	232.468719	17457100	12
2	2022-12-29	235.649994	241.919998	235.649994	241.009995	238.891769	19770700	12
3	2022-12-30	238.210007	239.960007	236.660004	239.820007	237.712234	21938500	12
4	2023-01-03	243.080002	245.750000	237.399994	239.580002	237.474350	25740000	1

In [12]:
```
#creating a new column in dataframe for month
google['Date'] = pd.to_datetime(google['Date'])
google['Month'] = google['Date'].dt.month
google.head()
```

Out[12]:

	Date	Open	High	Low	Close	Adj Close	Volume	Month
0	2022-12-27	89.309998	89.500000	87.535004	87.930000	87.930000	15470900	12
1	2022-12-28	87.500000	88.519997	86.370003	86.459999	86.459999	17879600	12
2	2022-12-29	87.029999	89.364998	86.989998	88.949997	88.949997	18280700	12
3	2022-12-30	87.364998	88.830002	87.029999	88.730003	88.730003	19190300	12
4	2023-01-03	89.830002	91.550003	89.019997	89.699997	89.699997	20738500	1

In [13]:
```
#creating a new column in dataframe for month
apple['Date'] = pd.to_datetime(apple['Date'])
apple['Month'] = apple['Date'].dt.month
apple.head()
```

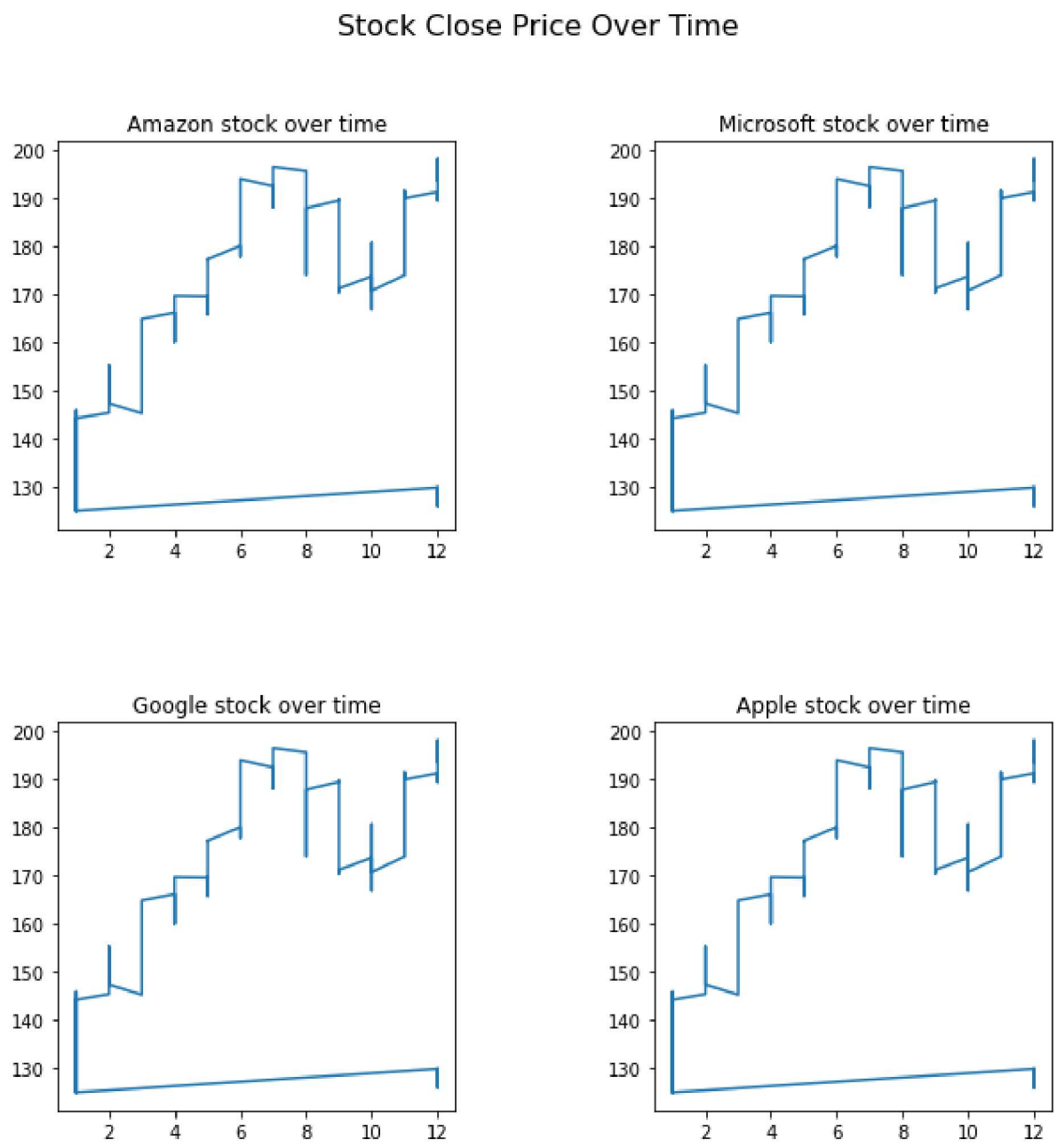
Out[13]:

	Date	Open	High	Low	Close	Adj Close	Volume	Month
0	2022-12-27	131.380005	131.410004	128.720001	130.029999	129.307251	69007800	12
1	2022-12-28	129.669998	131.029999	125.870003	126.040001	125.339409	85438400	12
2	2022-12-29	127.989998	130.479996	127.730003	129.610001	128.889587	75703700	12
3	2022-12-30	128.410004	129.949997	127.430000	129.929993	129.207779	77034200	12
4	2023-01-03	130.279999	130.899994	124.169998	125.070000	124.374802	112117500	1

In [14]:
```
# What was the change in stock price over time?
```

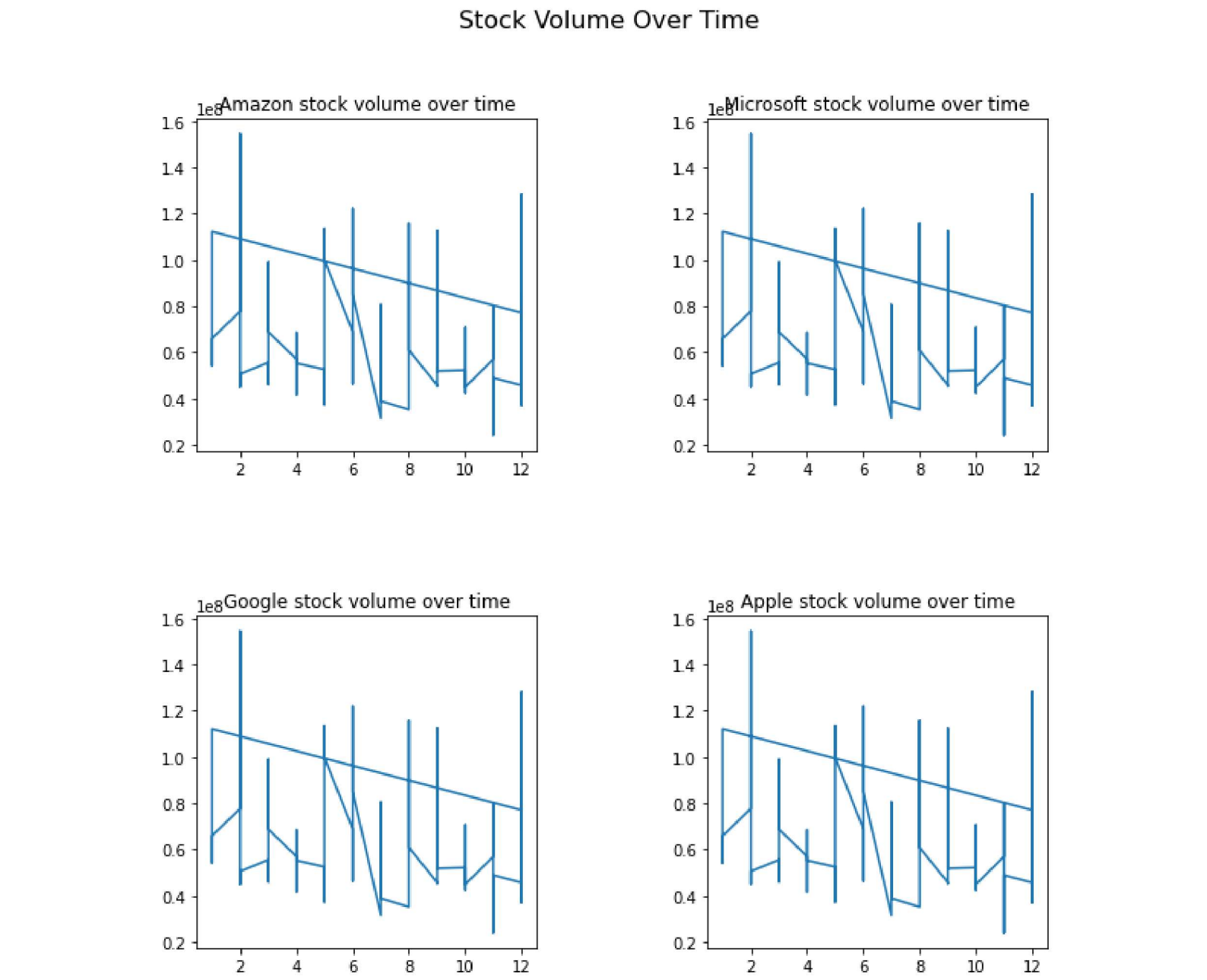
In [15]:
```
plt.figure(figsize=(10,10))
plt.subplots_adjust(wspace=0.5,hspace=0.5)

plt.subplot(2,2,1)
plt.plot(amazon['Month'],apple['Close'])
plt.title("Amazon stock over time")

plt.subplot(2,2,2)
plt.plot(microsoft['Month'],apple['Close'])
plt.title("Microsoft stock over time")

plt.subplot(2,2,3)
plt.plot(google['Month'],apple['Close'])
plt.title("Google stock over time")

plt.subplot(2,2,4)
plt.plot(apple['Month'],apple['Close'])
plt.title("Apple stock over time")
```

```
plt.suptitle("Stock Close Price Over Time", fontsize=16)
plt.show()
```



In [16]:
```
# Visualize the change in a stock's volume being traded, over time
```

In [17]:
```
plt.figure(figsize=(10,10))
plt.subplots_adjust(wspace=0.5,hspace=0.5)

plt.subplot(2,2,1)
plt.plot(amazon['Month'],apple['Volume'])
plt.title("Amazon stock volume over time")

plt.subplot(2,2,2)
plt.plot(microsoft['Month'],apple['Volume'])
plt.title("Microsoft stock volume over time")

plt.subplot(2,2,3)
plt.plot(google['Month'],apple['Volume'])
plt.title("Google stock volume over time")

plt.subplot(2,2,4)
plt.plot(apple['Month'],apple['Volume'])
plt.title("Apple stock volume over time")

plt.suptitle("Stock Volume Over Time", fontsize=16)
plt.show()
```

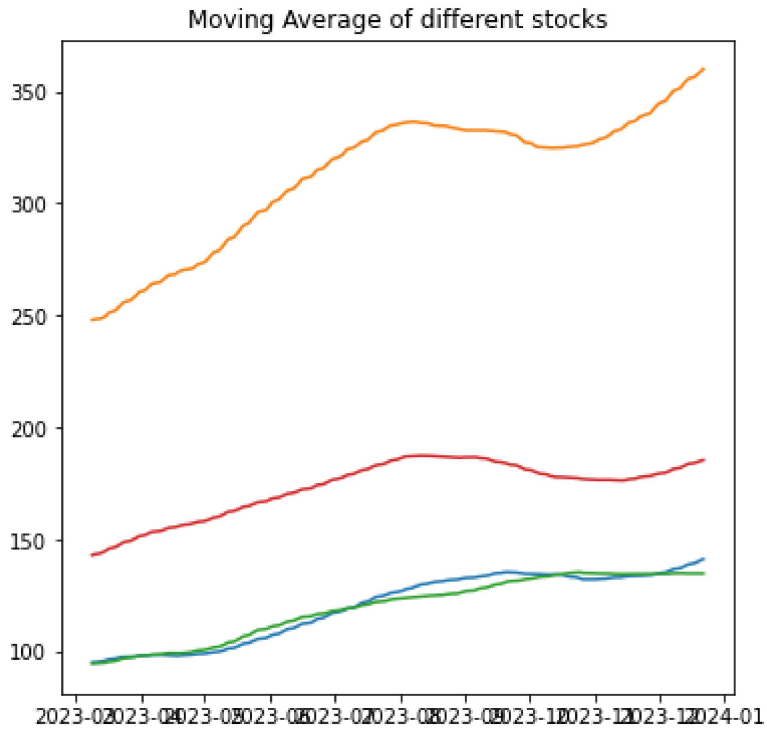


In [18]:
```
# What was the moving average of various stocks?
```

In [19]:
```
amazon.set_index('Date', inplace=True)
microsoft.set_index('Date', inplace=True)
google.set_index('Date', inplace=True)
apple.set_index('Date', inplace=True)
```

In [20]:
```
amazon['MA50'] = amazon['Close'].rolling(window=50).mean()

microsoft['MA50'] = microsoft['Close'].rolling(window=50).mean()

google['MA50'] = google['Close'].rolling(window=50).mean()

apple['MA50'] = apple['Close'].rolling(window=50).mean()
```

In [21]:
```
plt.figure(figsize=(6,6))
plt.plot(amazon.index, amazon['MA50'], label='Amazon')
plt.plot(microsoft.index, microsoft['MA50'], label='Microsoft')
plt.plot(google.index, google['MA50'], label='Google')
plt.plot(apple.index, apple['MA50'], label='Apple')
plt.title("Moving Average of different stocks")
plt.show()
```



In [22]:
```
# What was the daily return average of a stock?
```

In [23]:
```
amazon['Daily_Return'] = amazon['Close'].pct_change()
microsoft['Daily_Return'] = microsoft['Close'].pct_change()
google['Daily_Return'] = google['Close'].pct_change()
apple['Daily_Return'] = apple['Close'].pct_change()
```

In [24]:
```
avg_daily_return_amazon = amazon['Daily_Return'].mean()
print("Avg daily return for Amazon is",avg_daily_return_amazon)

avg_daily_return_microsoft = microsoft['Daily_Return'].mean()
print("Avg daily return for Microsoft is",avg_daily_return_microsoft)

avg_daily_return_google = google['Daily_Return'].mean()
print("Avg daily return for Google is",avg_daily_return_google)

avg_daily_return_apple = apple['Daily_Return'].mean()
print("Avg daily return for Apple is",avg_daily_return_apple)
```

```
Avg daily return for Amazon is 0.0026855084576662283
Avg daily return for Microsoft is 0.0019665339438624794
Avg daily return for Google is 0.002134746885412589
Avg daily return for Apple is 0.001684834070008406
```

In [25]:
```
# Add a new column ‘Trend’ whose values are based on the 'Daily Return'
```

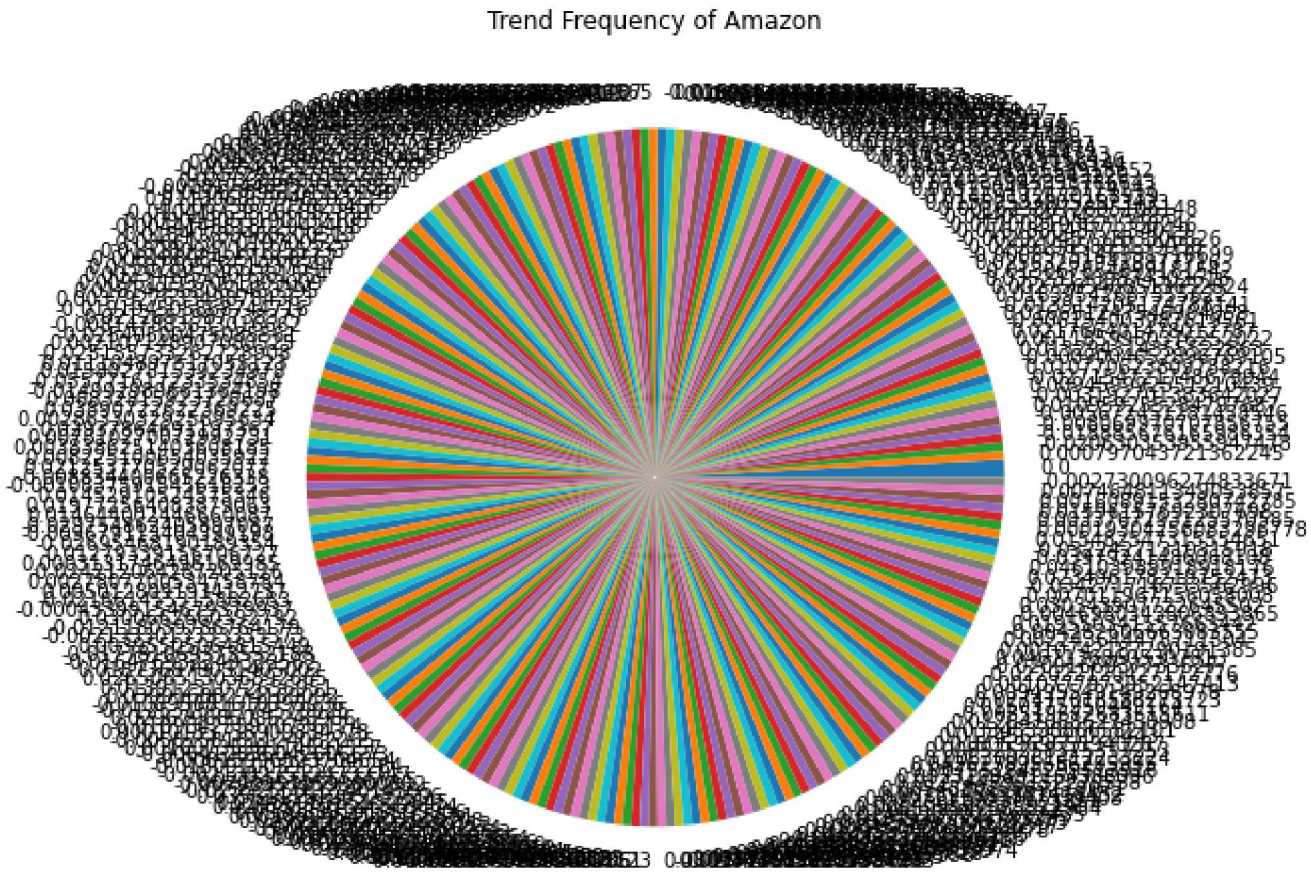
In [26]:
```
amazon['Trend'] = amazon['Daily_Return']
microsoft ['Trend'] = microsoft['Daily_Return']
google['Trend'] = google['Daily_Return']
apple['Trend'] = apple['Daily_Return']
```

In [27]:
```
# Visualize trend frequency through a Pie Chart.
```

In [28]:
```
amazon_trend = amazon['Trend'].value_counts()
print(amazon_trend)
```

```
 0.000000    2
 0.000797    1
-0.020630    1
-0.018886    1
-0.008070    1
             ..
 0.003377    1
 0.015962    1
 0.001609    1
 0.007465    1
-0.002730    1
Name: Trend, Length: 248, dtype: int64
```

In [29]:
```
plt.figure(figsize=(8, 8))
plt.pie(amazon_trend, labels=amazon_trend.index)
plt.title('Trend Frequency of Amazon')
plt.show()
```



In [30]:
```
# What was the correlation between the daily returns of different stocks?
```

In [31]:
```
# creating a new table
merged_table = pd.DataFrame()
```

In [32]:
```
merged_table['amazon_daily_return'] = amazon['Daily_Return']
merged_table['microsoft_daily_return'] = microsoft['Daily_Return']
merged_table['google_daily_return'] = google['Daily_Return']
merged_table['apple_daily_return'] = apple['Daily_Return']
```

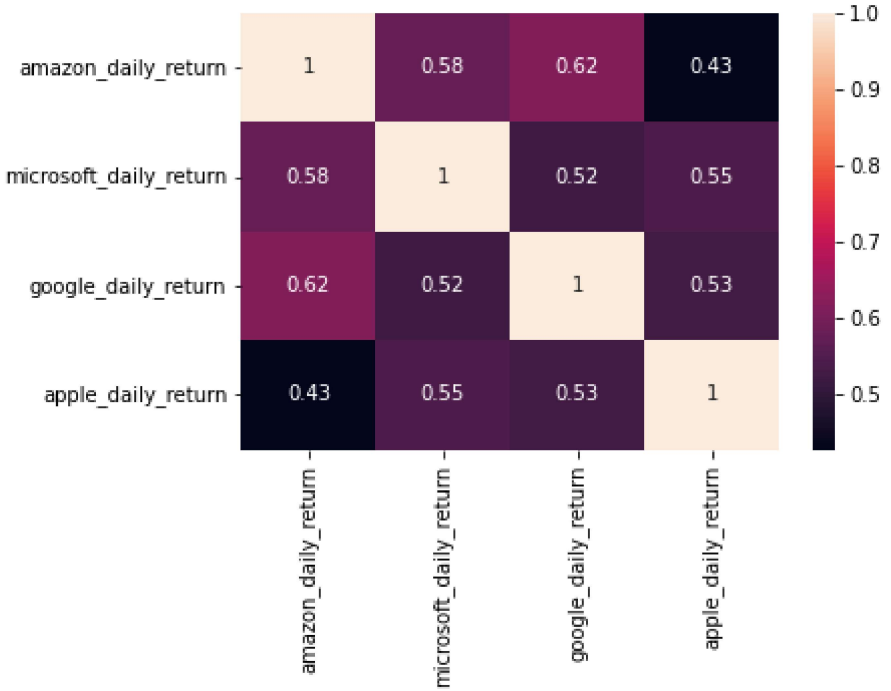
In [33]:
```
print(merged_table.head())
```

```
            amazon_daily_return  microsoft_daily_return  google_daily_return  \
Date
2022-12-27                  NaN                     NaN                  NaN
2022-12-28            -0.014692               -0.010255            -0.016718
2022-12-29             0.028844                0.027630             0.028799
2022-12-30            -0.002138               -0.004938            -0.002473
2023-01-03             0.021667               -0.001001             0.010932

            apple_daily_return
Date
2022-12-27                 NaN
2022-12-28           -0.030685
2022-12-29            0.028324
2022-12-30            0.002469
2023-01-03           -0.037405
```

In [34]:
```
correlation_matrix = merged_table.corr()
print(correlation_matrix)
```

```
                        amazon_daily_return  microsoft_daily_return  \
amazon_daily_return                1.000000                0.579102
microsoft_daily_return             0.579102                1.000000
google_daily_return                0.616524                0.522540
apple_daily_return                 0.428669                0.545366

                        google_daily_return  apple_daily_return
amazon_daily_return                0.616524           0.428669
microsoft_daily_return             0.522540           0.545366
google_daily_return                1.000000           0.527188
apple_daily_return                 0.527188           1.000000
```

In [35]:
```
#creating a correlation heatmap
sns.heatmap(correlation_matrix, xticklabels = correlation_matrix.columns,
            yticklabels = correlation_matrix.columns, annot = True)
plt.show()
```



In [36]:
```
# this mini project is submitted to Edureka
# Submitted by Umang
```