Time Series Project (Stock Market Analysis)

```
#importing the necessary libraries
import seaborn as sns
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Our first task is to

- 1. Analyse closing price of all the stocks
- 2. Analyse the total volumn of stock being trade each day

```
company_list = ['AAPL_data.csv','GOOGL_data.csv','MSFT_data.csv','AMZN_data.csv']
full_df = pd.DataFrame()
path='F:/Dataset/2-Time Series Data Analysis-20220907T063304Z-001/2-Time Series Data An

for file in company_list:
    current_file = pd.read_csv(path+ "/" + file)
    full_df= pd.concat([full_df,current_file])
```

```
full_df.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL
1	2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL
2	2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL
3	2013-02-13	66.7442	67.6628	66.1742	66.7156	118721995	AAPL
4	2013-02-14	66.3599	67.3771	66.2885	66.6556	88809154	AAPL

```
full_df.shape
```

```
(5036, 7)
```

Data types

```
full_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5036 entries, 0 to 1258
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    5036 non-null   object
 1   open    5036 non-null   float64
```

```
 2   high    5036 non-null   float64
 3   low     5036 non-null   float64
 4   close   5036 non-null   float64
 5   volume  5036 non-null   int64
 6   Name    5036 non-null   object
dtypes: float64(4), int64(1), object(2)
memory usage: 314.8+ KB
```

```
full_df['date']=pd.to_datetime(full_df['date'])
```

```
full_df.info()
```

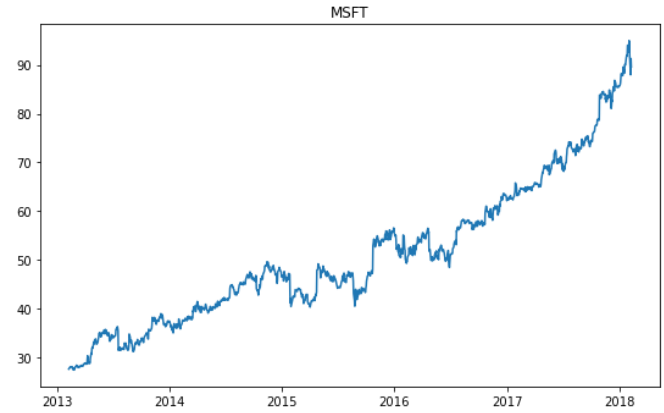
```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 5036 entries, 0 to 1258
Data columns (total 7 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   date    5036 non-null   datetime64[ns]
 1   open    5036 non-null   float64
 2   high    5036 non-null   float64
 3   low     5036 non-null   float64
 4   close   5036 non-null   float64
 5   volume  5036 non-null   int64
 6   Name    5036 non-null   object
dtypes: datetime64[ns](1), float64(4), int64(1), object(1)
memory usage: 314.8+ KB
```

```
tech_list = full_df['Name'].unique()
```

```
tech_list
```

```
array(['AAPL', 'GOOGL', 'MSFT', 'AMZN'], dtype=object)
```

```
plt.figure(figsize=(20,12))
for i,company in enumerate(tech_list,1):
    plt.subplot(2,2,i)
    df= full_df[full_df['Name']==company]
    plt.plot(df['date'],df['close'])
    plt.title(company)
```




```
import plotly.express as px
```

```
for company in tech_list:
    df = full_df[full_df['Name']==company]
    px.line(df,x='date',y='volume',title=company).show()
```

2. Analyzing Daily Returns

Problem statement

- Analyse Daily pice change in stock
- Analyse Monthly mean of close feature

```
df = pd.read_csv('F:/Dataset/2-Time Series Data Analysis-20220907T063304Z-001/2-Time Se
```

```
df.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	261.40	265.25	260.555	261.95	3879078	AMZN
1	2013-02-11	263.20	263.25	256.600	257.21	3403403	AMZN
2	2013-02-12	259.19	260.16	257.000	258.70	2938660	AMZN
3	2013-02-13	261.53	269.96	260.300	269.47	5292996	AMZN
4	2013-02-14	267.37	270.65	265.400	269.24	3462780	AMZN

As we see there is no feature which tells us Daily price change

```
df['daily_price_change']=df['close']-df['open']
```

```
df.head()
```

	date	open	high	low	close	volume	Name	daily_price_change
0	2013-02-08	261.40	265.25	260.555	261.95	3879078	AMZN	0.55
1	2013-02-11	263.20	263.25	256.600	257.21	3403403	AMZN	-5.99
2	2013-02-12	259.19	260.16	257.000	258.70	2938660	AMZN	-0.49
3	2013-02-13	261.53	269.96	260.300	269.47	5292996	AMZN	7.94
4	2013-02-14	267.37	270.65	265.400	269.24	3462780	AMZN	1.87

Lets also find the percentage return of daily

```
df['1day_%_return']=((df['close']-df['open'])/df['close'])*100
```

```
df.head()
```

	date	open	high	low	close	volume	Name	daily_price_change	1day_%_return
0	2013-02-08	261.40	265.25	260.555	261.95	3879078	AMZN	0.55	0.209964
1	2013-02-11	263.20	263.25	256.600	257.21	3403403	AMZN	-5.99	-2.328836
2	2013-02-12	259.19	260.16	257.000	258.70	2938660	AMZN	-0.49	-0.189409
3	2013-02-13	261.53	269.96	260.300	269.47	5292996	AMZN	7.94	2.946525
4	2013-02-14	267.37	270.65	265.400	269.24	3462780	AMZN	1.87	0.694548

```
px.line(data_frame=df,x='date',y='1day_%_return').show()
```

Analyse Monthly mean of close feature

```
df2 = df.copy()
```

```
df2.dtypes
```

```
date                  object
open                 float64
high                 float64
low                  float64
close                float64
volume                 int64
Name                  object
daily_price_change   float64
1day_%_return        float64
dtype: object
```

```
df2['date']=pd.to_datetime(df2['date'])
```

```
df2.set_index('date',inplace=True)
```

```
df2.head()
```

date	open	high	low	close	volume	Name	daily_price_change	1day_%_return
2013-02-08	261.40	265.25	260.555	261.95	3879078	AMZN	0.55	0.209964
2013-02-11	263.20	263.25	256.600	257.21	3403403	AMZN	-5.99	-2.328836
2013-02-12	259.19	260.16	257.000	258.70	2938660	AMZN	-0.49	-0.189409
2013-02-13	261.53	269.96	260.300	269.47	5292996	AMZN	7.94	2.946525
2013-02-14	267.37	270.65	265.400	269.24	3462780	AMZN	1.87	0.694548

```
df2['2013-02-08':'2013-03-08']
```

date	open	high	low	close	volume	Name	daily_price_change	1day_%_return
2013-02-08	261.40	265.250	260.555	261.95	3879078	AMZN	0.55	0.209964
2013-02-11	263.20	263.250	256.600	257.21	3403403	AMZN	-5.99	-2.328836

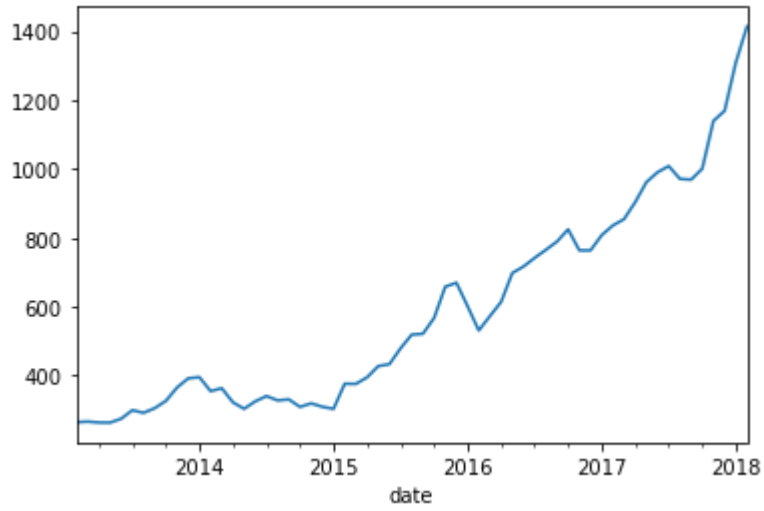
date	open	high	low	close	volume	Name	daily_price_change	1day_%_return
2013-02-12	259.19	260.160	257.000	258.70	2938660	AMZN	-0.49	-0.189409
2013-02-13	261.53	269.960	260.300	269.47	5292996	AMZN	7.94	2.946525
2013-02-14	267.37	270.650	265.400	269.24	3462780	AMZN	1.87	0.694548
2013-02-15	267.63	268.920	263.110	265.09	3979832	AMZN	-2.54	-0.958165
2013-02-19	265.91	270.110	264.500	269.75	2853752	AMZN	3.84	1.423540
2013-02-20	270.20	274.300	266.371	266.41	3528862	AMZN	-3.79	-1.422619
2013-02-21	265.12	269.480	263.250	265.94	3637396	AMZN	0.82	0.308340
2013-02-22	266.62	267.110	261.610	265.42	3123402	AMZN	-1.20	-0.452114
2013-02-25	266.94	268.694	259.650	259.87	3032109	AMZN	-7.07	-2.720591
2013-02-26	260.89	262.040	255.730	259.36	3348011	AMZN	-1.53	-0.589914
2013-02-27	259.40	265.830	256.860	263.25	2908010	AMZN	3.85	1.462488
2013-02-28	261.81	267.000	260.630	264.27	2667199	AMZN	2.46	0.930866
2013-03-01	263.27	266.600	261.040	265.74	2956724	AMZN	2.47	0.929480
2013-03-04	265.36	273.300	264.140	273.11	3452783	AMZN	7.75	2.837684
2013-03-05	274.00	276.680	269.990	275.59	3685983	AMZN	1.59	0.576944
2013-03-06	275.76	276.489	271.832	273.79	2050452	AMZN	-1.97	-0.719530
2013-03-07	274.10	274.800	271.850	273.88	1938987	AMZN	-0.22	-0.080327
2013-03-08	275.00	275.440	271.500	274.19	1879762	AMZN	-0.81	-0.295416

```
df2['close'].resample('M').mean()
```

```
date
2013-02-28     263.995000
2013-03-31     265.758400
2013-04-30     263.072364
2013-05-31     262.727727
2013-06-30     274.101900
                  ...
2017-10-31    1000.720000
2017-11-30    1139.808095
2017-12-31    1168.841500
2018-01-31    1309.010952
2018-02-28    1413.914000
Freq: M, Name: close, Length: 61, dtype: float64
```

```
df2['close'].resample('M').mean().plot()
```

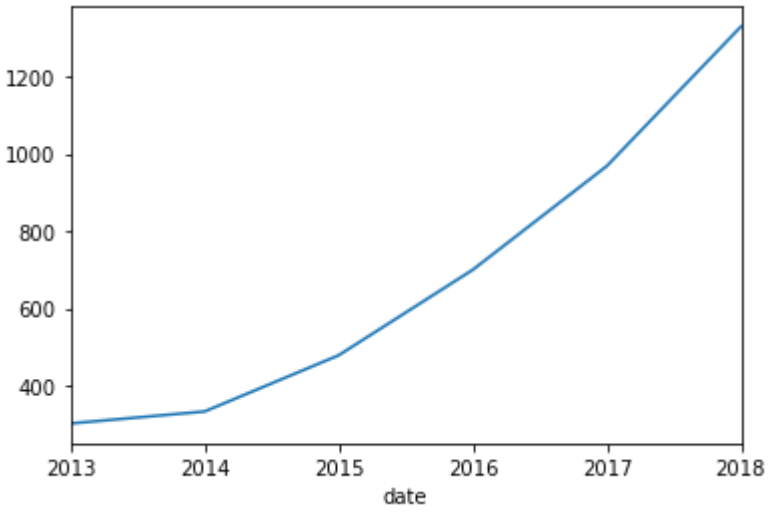
```
<AxesSubplot:xlabel='date'>
```



Resample based on year

```
df2['close'].resample('Y').mean().plot()
```

```
<AxesSubplot:xlabel='date'>
```



3. Performing Multi-variate Analysis

Problem Statment

- Analysis weather stock price of these tech companies are correlated or not
- Analyse Daily return of each stock and how they are co-related
- Value at risk analysis For Tech Companies

```
company_list
```

```
['AAPL_data.csv', 'GOOGL_data.csv', 'MSFT_data.csv', 'AMZN_data.csv']
```

```
amz = pd.read_csv('F:/Dataset/2-Time Series Data Analysis-20220907T063304Z-001/2-Time S
```

```
amz.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	261.40	265.25	260.555	261.95	3879078	AMZN
1	2013-02-11	263.20	263.25	256.600	257.21	3403403	AMZN
2	2013-02-12	259.19	260.16	257.000	258.70	2938660	AMZN
3	2013-02-13	261.53	269.96	260.300	269.47	5292996	AMZN
4	2013-02-14	267.37	270.65	265.400	269.24	3462780	AMZN

```
apple = pd.read_csv('F:/Dataset/2-Time Series Data Analysis-20220907T063304Z-001/2-Time
```

```
apple.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL
1	2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL
2	2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL
3	2013-02-13	66.7442	67.6628	66.1742	66.7156	118721995	AAPL
4	2013-02-14	66.3599	67.3771	66.2885	66.6556	88809154	AAPL

```
google = pd.read_csv('F:/Dataset/2-Time Series Data Analysis-20220907T063304Z-001/2-Tim
```

```
google.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	390.4551	393.7283	390.1698	393.0777	6031199	GOOGL
1	2013-02-11	389.5892	391.8915	387.2619	391.6012	4330781	GOOGL
2	2013-02-12	391.2659	394.3440	390.0747	390.7403	3714176	GOOGL
3	2013-02-13	390.4551	393.0677	390.3750	391.8214	2393946	GOOGL
4	2013-02-14	390.2549	394.7644	389.2739	394.3039	3466971	GOOGL

```
msft = pd.read_csv('F:/Dataset/2-Time Series Data Analysis-20220907T063304Z-001/2-Time
```

```
msft.head()
```

	date	open	high	low	close	volume	Name
0	2013-02-08	27.35	27.71	27.31	27.55	33318306	MSFT
1	2013-02-11	27.65	27.92	27.50	27.86	32247549	MSFT
2	2013-02-12	27.88	28.00	27.75	27.88	35990829	MSFT
3	2013-02-13	27.93	28.11	27.88	28.03	41715530	MSFT
4	2013-02-14	27.92	28.06	27.87	28.04	32663174	MSFT

```
close_price = pd.DataFrame()
```

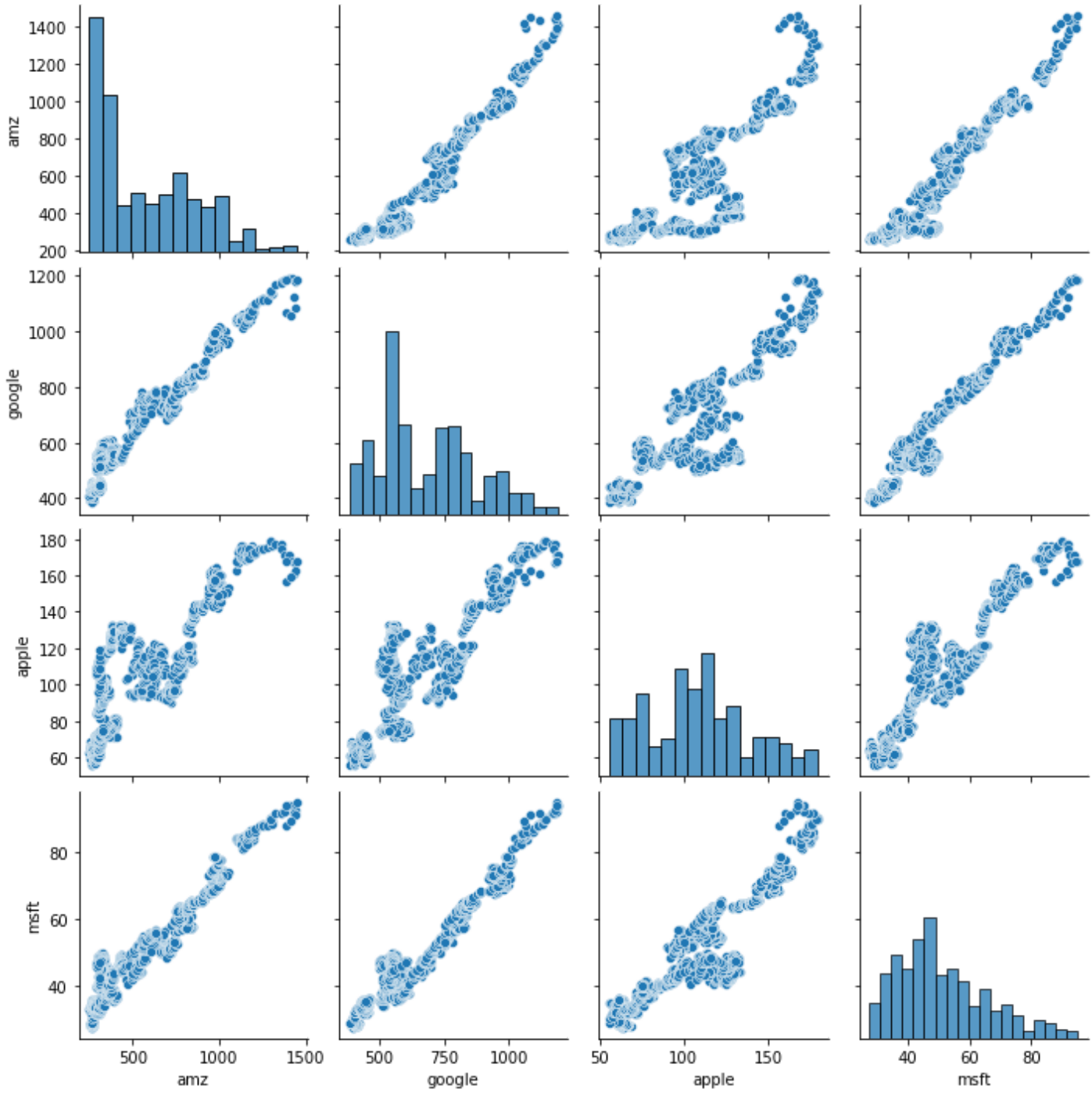
```
close_price['amz']= amz['close']
close_price['google']= google['close']
close_price['apple']= apple['close']
close_price['msft']= msft['close']
```

```
close_price.head()
```

	amz	google	apple	msft
0	261.95	393.0777	67.8542	27.55
1	257.21	391.6012	68.5614	27.86
2	258.70	390.7403	66.8428	27.88
3	269.47	391.8214	66.7156	28.03
4	269.24	394.3039	66.6556	28.04

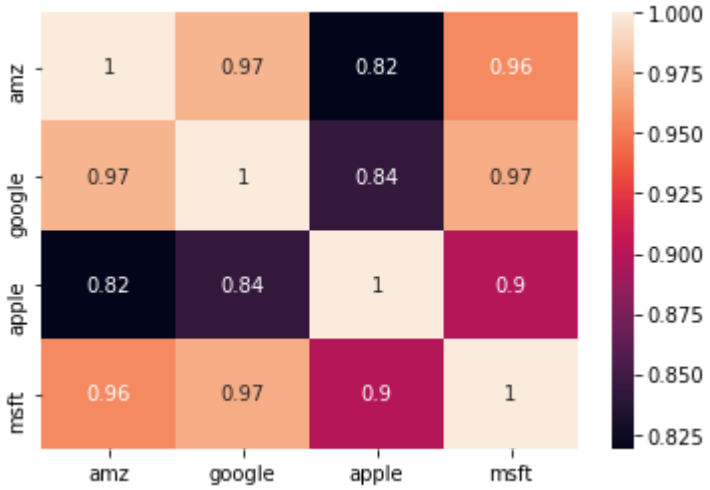
```
sns.pairplot(close_price)
```

```
<seaborn.axisgrid.PairGrid at 0x1c779b0b940>
```




```
sns.heatmap(close_price.corr(),annot=True)
```

```
<AxesSubplot:>
```



Analyse Daily return of each stock and how they are co-related.

```
company_name
```

```
['amz', 'apple', 'google', 'msft']
```

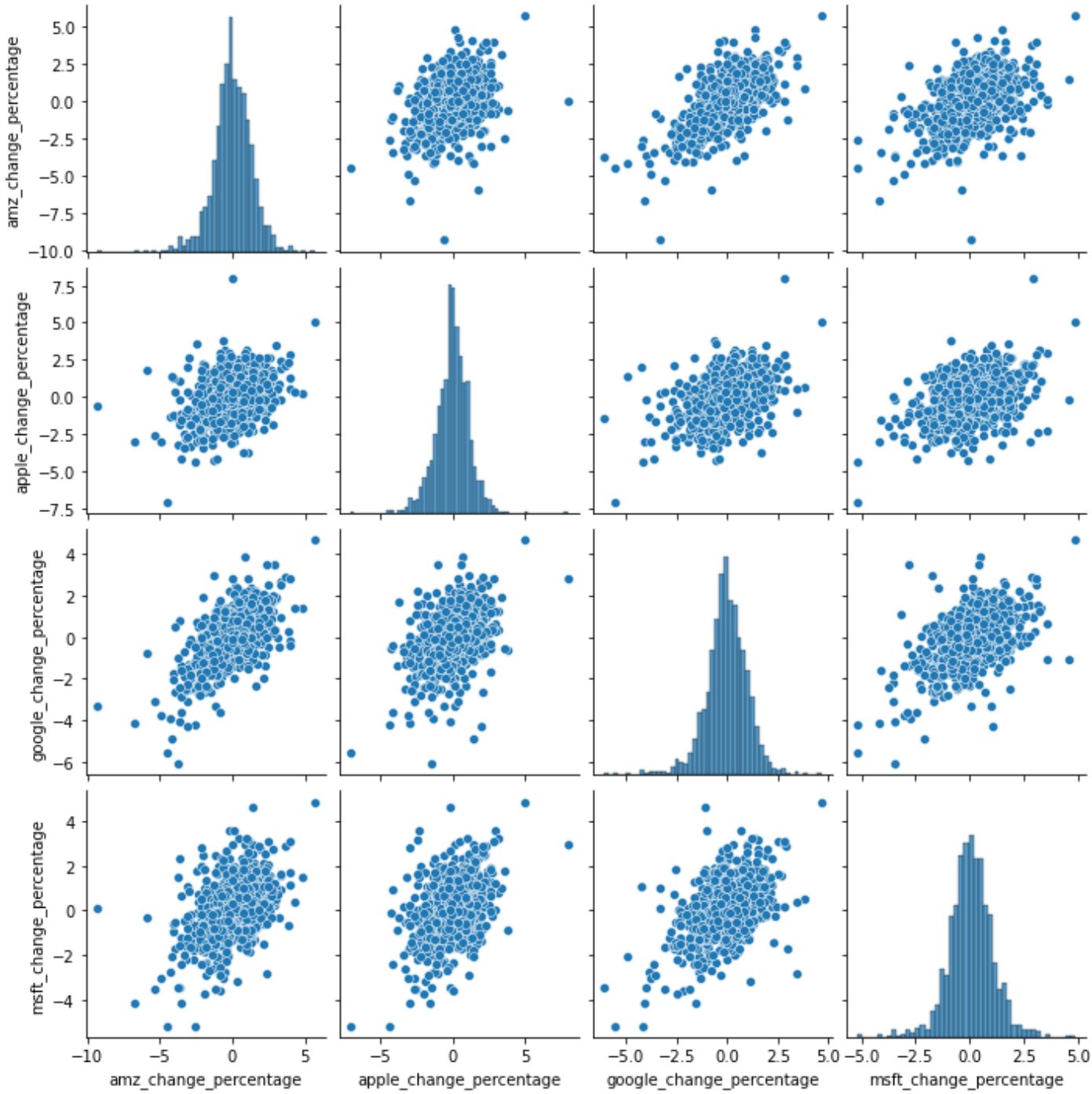
```
data = pd.DataFrame()
data['amz_change_percentage']= ((amz['close']-amz['open'])/amz['close'])*100
data['apple_change_percentage']= ((apple['close']-apple['open'])/apple['close'])*100
data['google_change_percentage']= ((google['close']-google['open'])/google['close'])*10
data['msft_change_percentage']= ((msft['close']-msft['open'])/msft['close'])*100
```

```
data.head()
```

	amz_change_percentage	apple_change_percentage	google_change_percentage	msft_change_percentage
0	0.209964	0.206325	0.667196	0.725953
1	-2.328836	0.714688	0.513788	0.753769
2	-0.189409	-2.481344	-0.134514	0.000000
3	2.946525	-0.042869	0.348705	0.356761
4	0.694548	0.443624	1.026873	0.427960

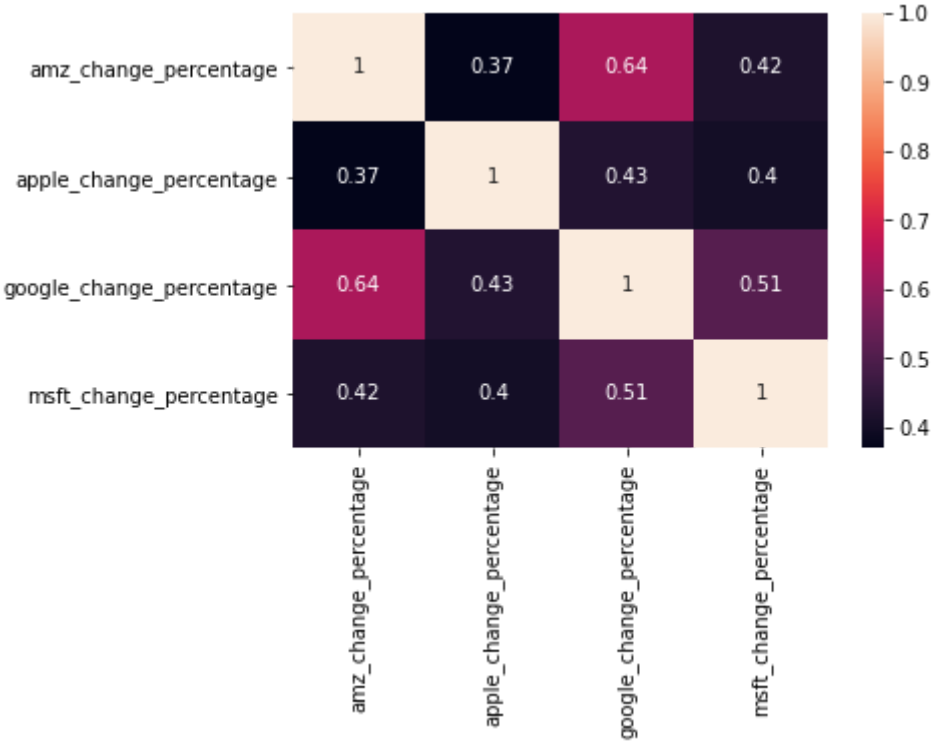
```
sns.pairplot(data=data)
```

```
<seaborn.axisgrid.PairGrid at 0x1c77abfc7f0>
```




```
sns.heatmap(data=data.corr(),annot=True)
```

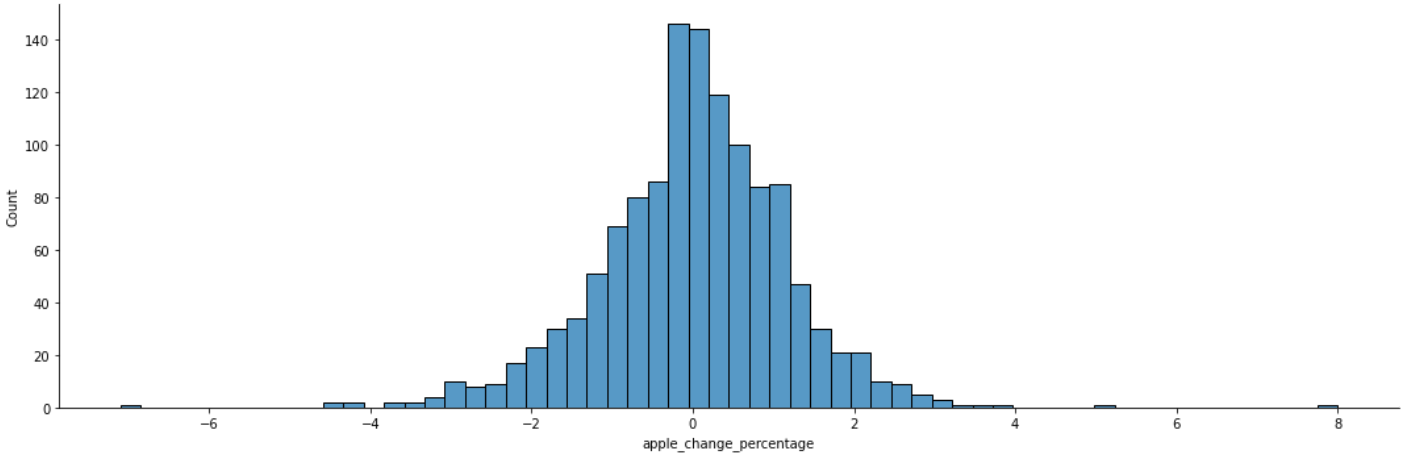
```
<AxesSubplot:>
```



Value at risk analysis For Tech Companies.

```
sns.displot(data['apple_change_percentage'],aspect=3)
```

```
<seaborn.axisgrid.FacetGrid at 0x1c778bcd0d0>
```




```
data['apple_change_percentage'].std()
# we have 68% of data in 1st std
```

```
1.1871377131421237
```

```
data['apple_change_percentage'].std()*2
#-2.37 to 2.3 we have 95% of data
```

```
2.3742754262842474
```

```
data['apple_change_percentage'].std()*3
#-3.5 to 3.5 we have 99.7% of data
```

3.561413139426371

```
data['apple_change_percentage'].quantile(0.1)
```

-1.4246644227944307

This tells us that my 90 % of time my data wont exceed -1.42 loss

```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
amz_change_percentage	1259.0	-0.000398	1.358679	-9.363077	-0.738341	-0.002623	0.852568	5.640265
apple_change_percentage	1259.0	-0.000215	1.187138	-7.104299	-0.658021	0.042230	0.715427	8.000388
google_change_percentage	1259.0	-0.028349	1.052191	-6.107290	-0.575799	-0.004508	0.624730	4.652214
msft_change_percentage	1259.0	0.076404	1.059260	-5.177618	-0.509241	0.061069	0.703264	4.861491

```

```