In [2]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:

```
path = r"X:\Data Science\Udemy Projects\S&P_500\individual_stocks_5yr"
company_list = ['AAPL_data.csv','GOOG_data.csv','MSFT_data.csv','AMZN_data.csv']
all_data = pd.DataFrame()
for file in company_list:
    current_df = pd.read_csv(path+'/'+file)
    all_data = pd.concat([all_data,current_df])
```

In [4]:

```
all_data.head()
```

Out[4]:

	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

In [5]:

```
all_data.shape
```

Out[5]:

```
(4752, 7)
```

Analyse closing price of all stocks

In [6]:

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

In [7]:

```
all_data.dtypes # date was object type
```

Out[7]:

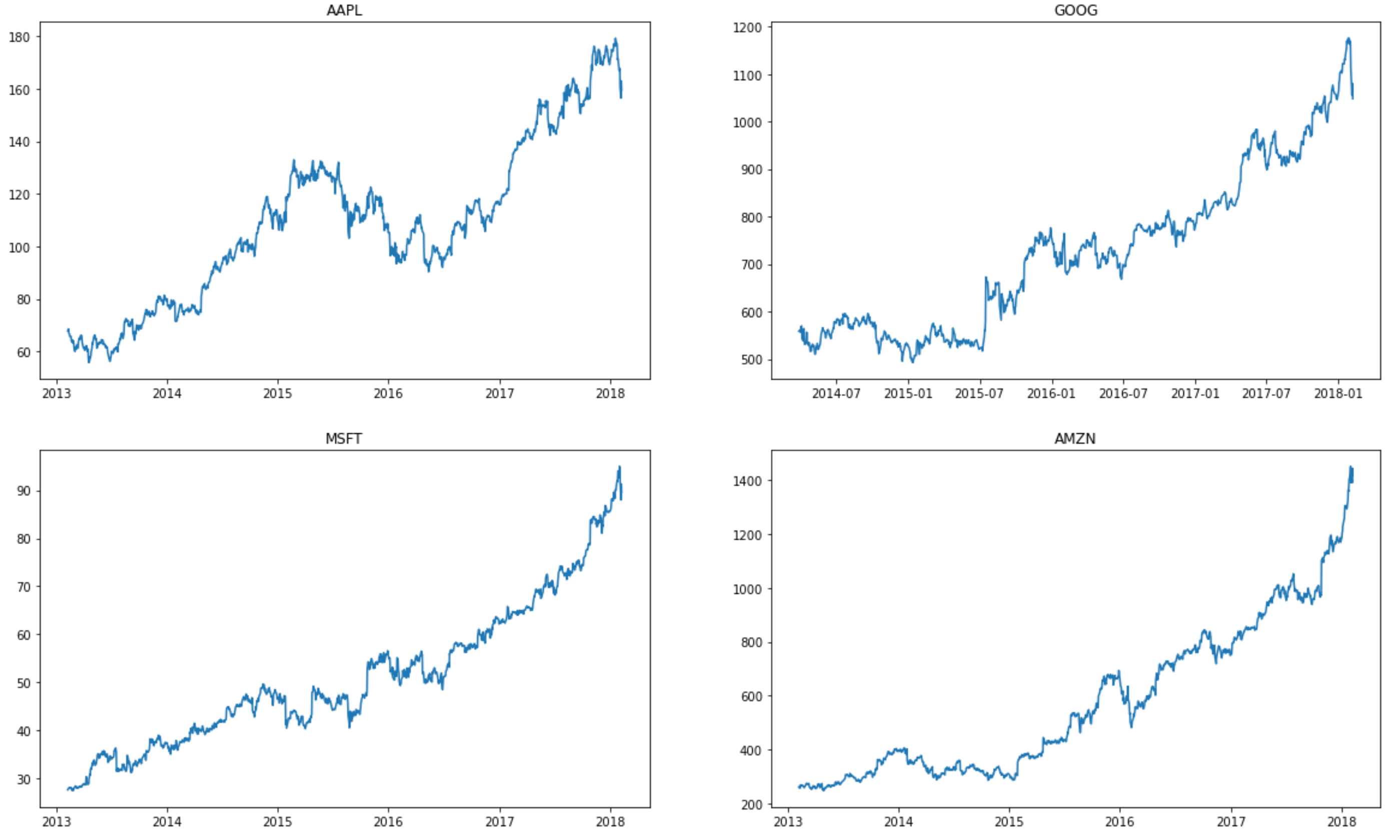
```
date        object
open       float64
high       float64
low        float64
close      float64
volume       int64
Name        object
dtype: object
```

In [8]:

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

In [9]:

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



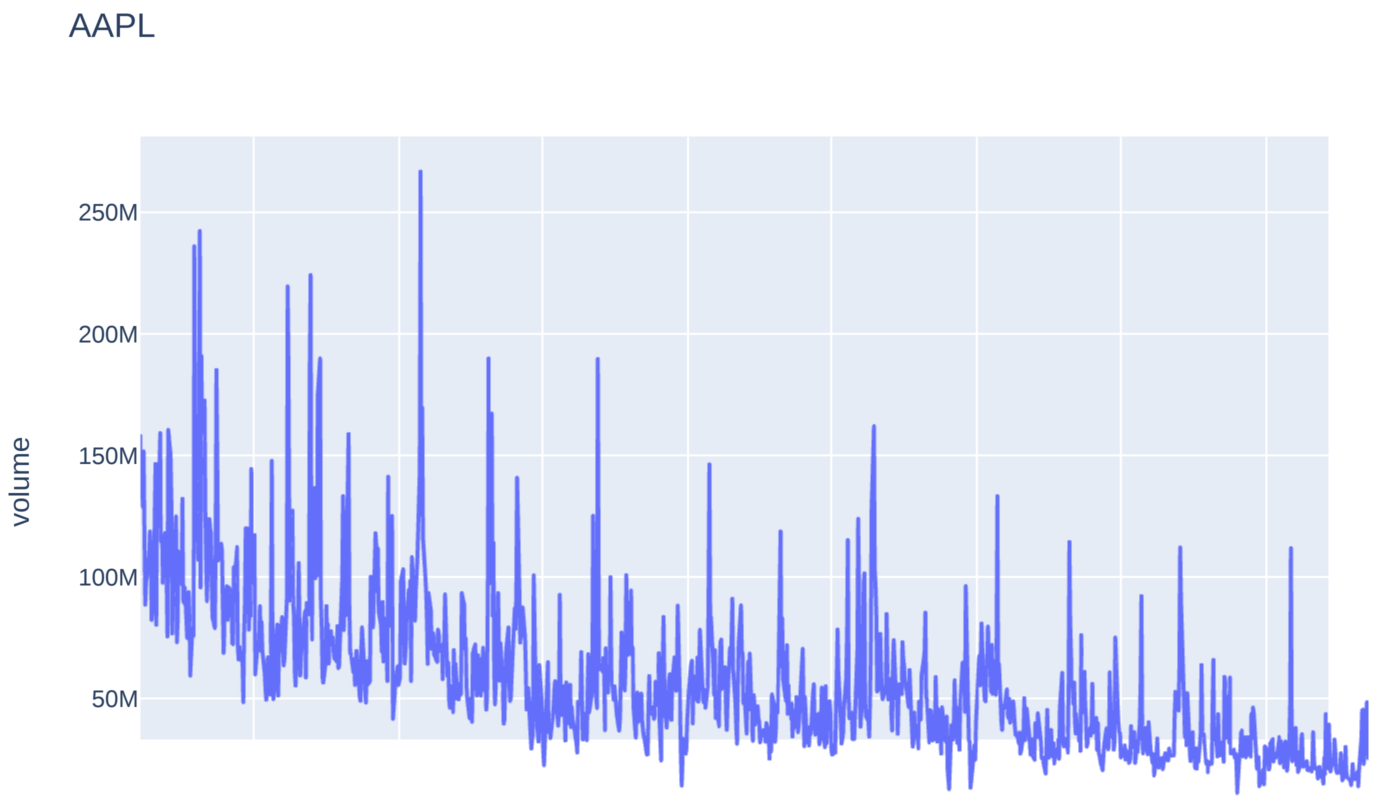
Analyse the total volume of stock being traded each day

In [10]:

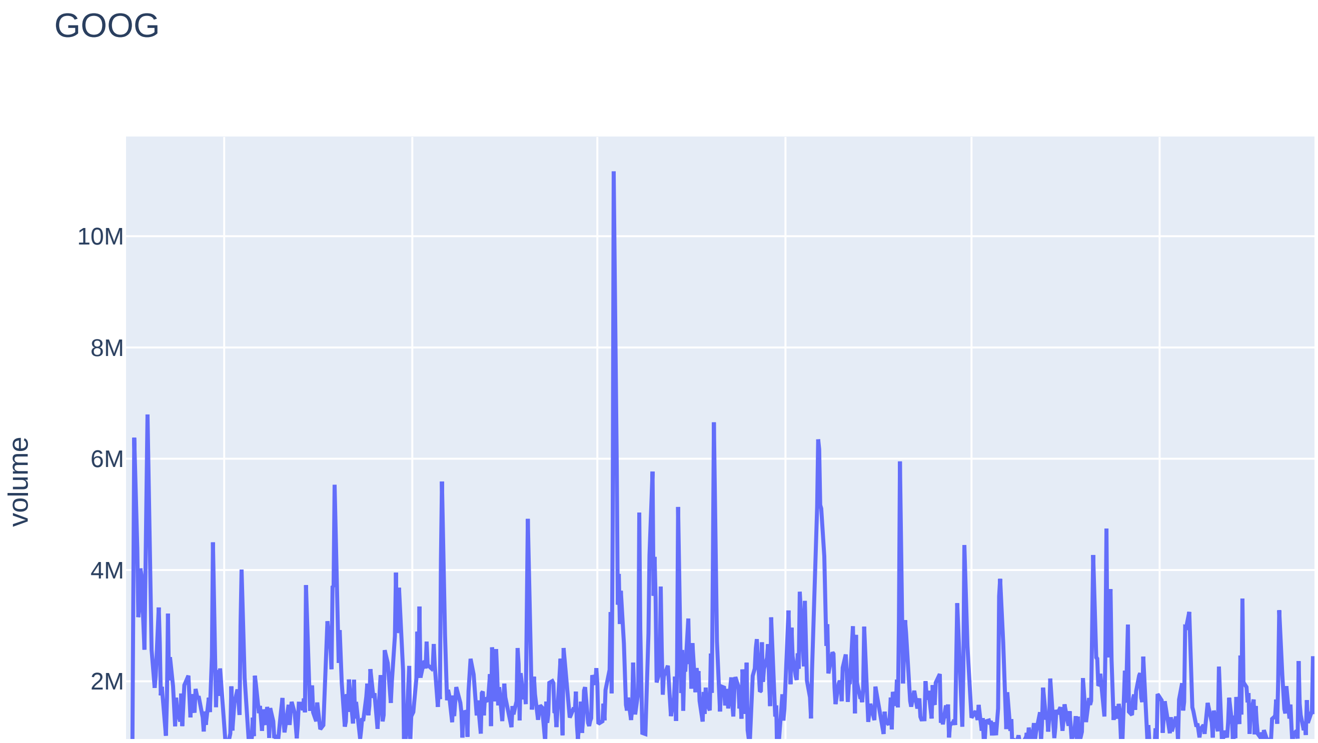
```
import plotly.express as px
```

In [11]:

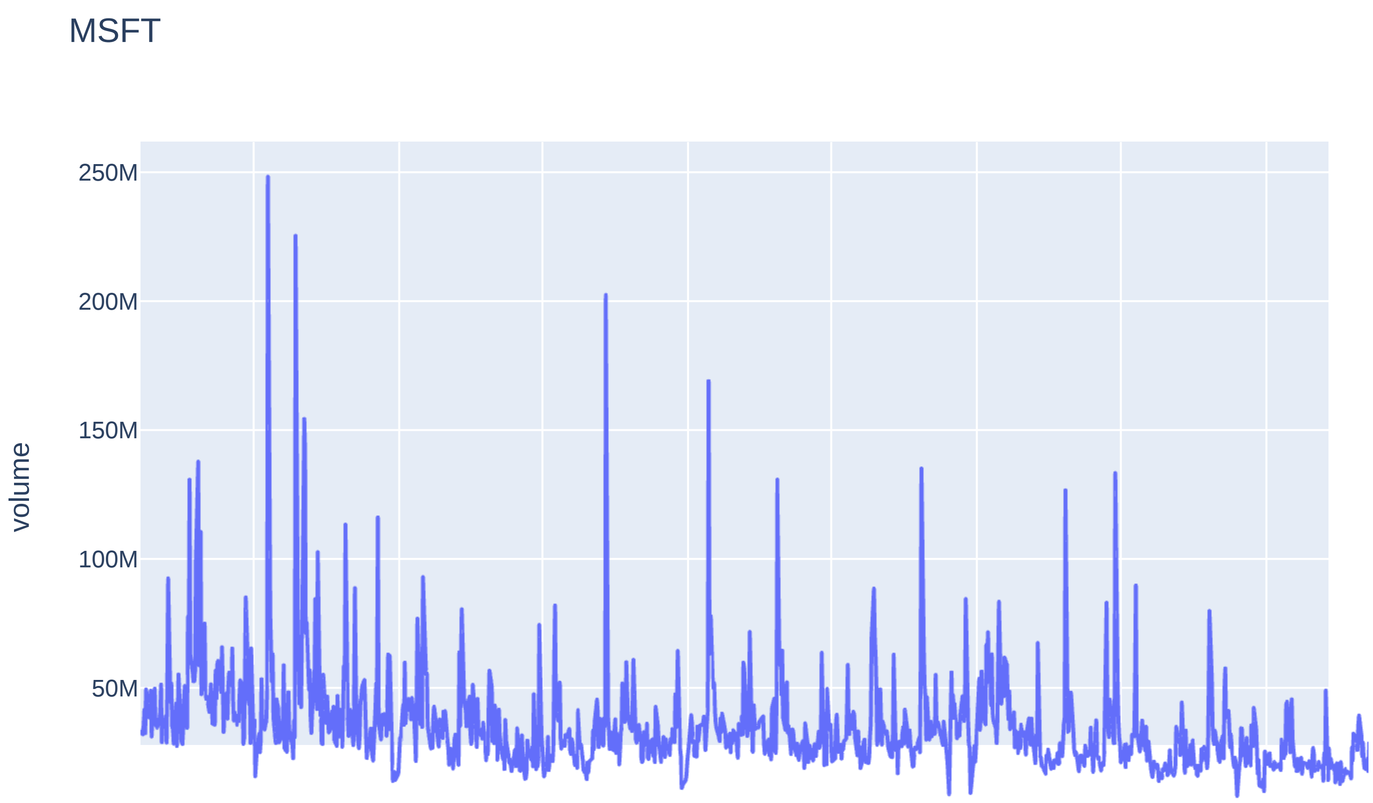
```
for company in tech_list:
    df = all_data[all_data['Name']==company]
    fig = px.line(df,x='date',y='volume',title= company)
    fig.show()
```



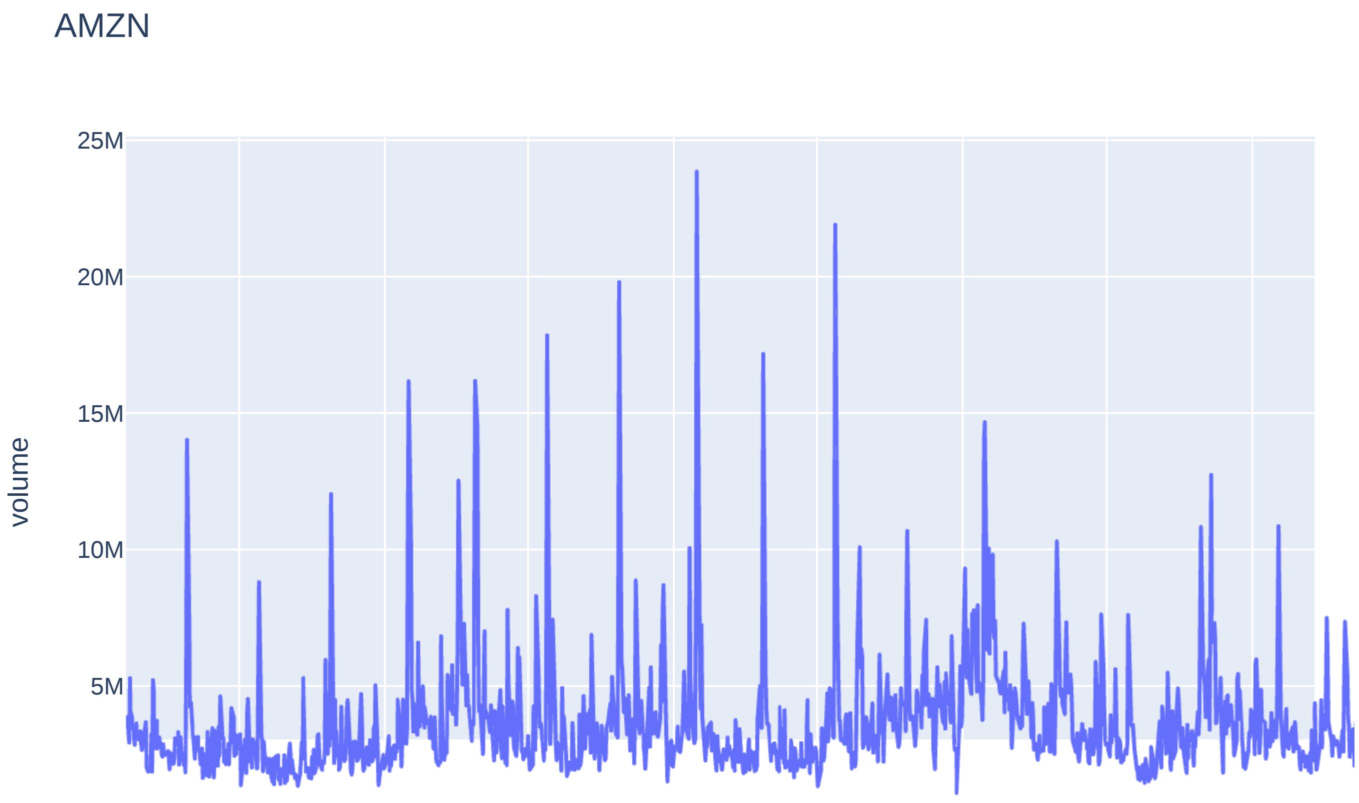
Loading [MathJax]/extensions/Safe.js



Loading [MathJax]/extensions/Safe.js



Loading [MathJax]/extensions/Safe.js



Analyse Daily price change in stock

```
In [12]: df = pd.read_csv(r"X:\Data Science\Udemy Projects\S&P_500\individual_stocks_5yr/AAPL_dat
```

```
In [13]: df.head()
```

Out[13]:

	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

```
In [14]: df['Daily_Price_Chage'] = df['close']-df['open']
```

```
In [15]: df.head()
```

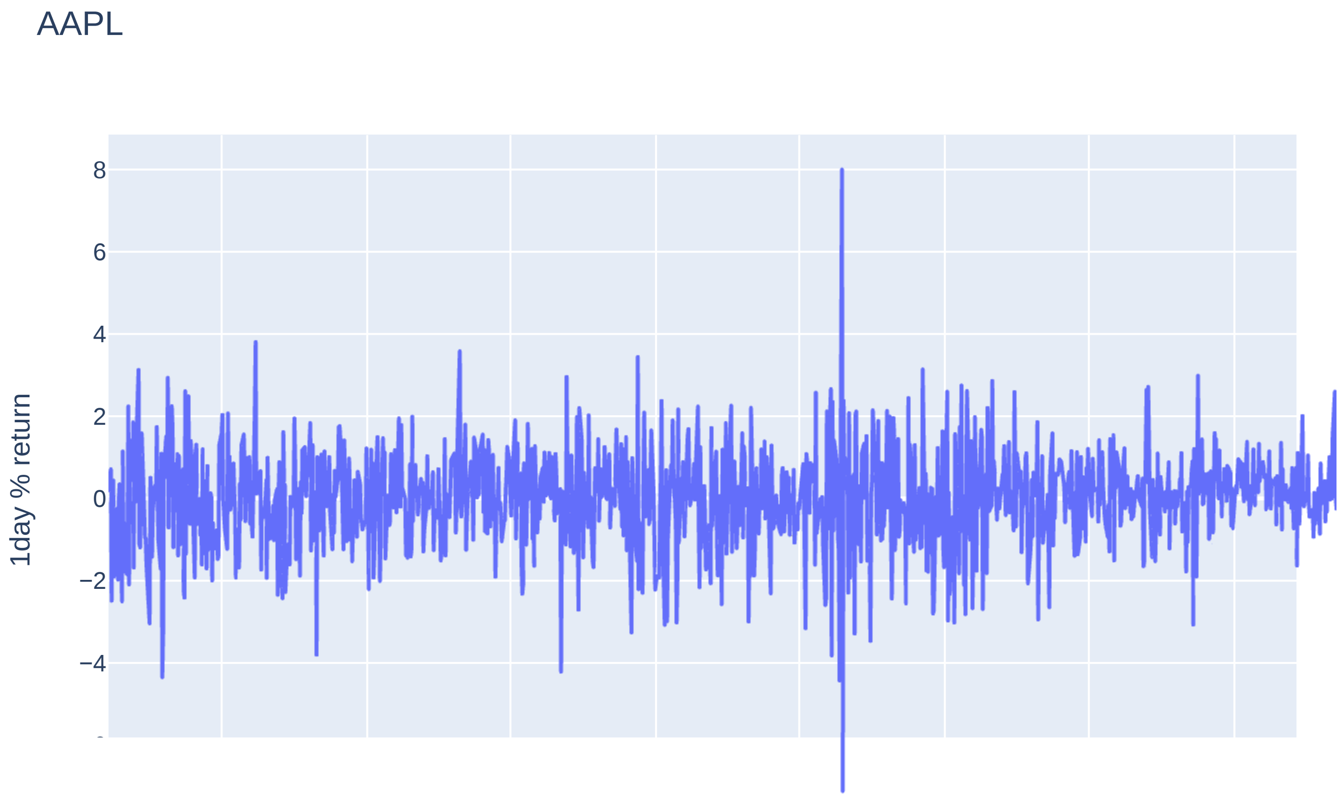
```
Out[15]:
         date     open     high      low    close     volume  Name  Daily_Price_Chage
   0  2013-02-08  67.7142  68.4014  66.8928  67.8542  158168416  AAPL             0.1400
   1  2013-02-11  68.0714  69.2771  67.6071  68.5614  129029425  AAPL             0.4900
   2  2013-02-12  68.5014  68.9114  66.8205  66.8428  151829363  AAPL            -1.6586
   3  2013-02-13  66.7442  67.6628  66.1742  66.7156  118721995  AAPL            -0.0286
   4  2013-02-14  66.3599  67.3771  66.2885  66.6556   88809154  AAPL             0.2957
```

```
In [16]: df['1day % return'] = ((df['close']-df['open'])/df['close'])*100
```

```
In [17]: df.head()
```

```
Out[17]:
         date     open     high      low    close     volume  Name  Daily_Price_Chage  1day % return
   0  2013-02-08  67.7142  68.4014  66.8928  67.8542  158168416  AAPL             0.1400       0.206325
   1  2013-02-11  68.0714  69.2771  67.6071  68.5614  129029425  AAPL             0.4900       0.714688
   2  2013-02-12  68.5014  68.9114  66.8205  66.8428  151829363  AAPL            -1.6586      -2.481344
   3  2013-02-13  66.7442  67.6628  66.1742  66.7156  118721995  AAPL            -0.0286      -0.042869
   4  2013-02-14  66.3599  67.3771  66.2885  66.6556   88809154  AAPL             0.2957       0.443624
```

```
In [18]: fig = px.line(df,x = 'date', y= '1day % return', title='AAPL')
         fig.show()
```



Loading [MathJax]/extensions/Safe.js

Analyse monthly mean of close feature

In [19]:

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

In [20]:

```
df2.dtypes
```

Out[20]:

```
date                     object
open                    float64
high                    float64
low                     float64
close                   float64
volume                    int64
Name                     object
Daily_Price_Chage       float64
1day % return           float64
dtype: object
```

In [21]:

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

In [22]:

```
df2.dtypes
```

Out[22]:

```
date                     datetime64[ns]
open                            float64
high                            float64
low                             float64
close                           float64
volume                            int64
Name                             object
Daily_Price_Chage               float64
1day % return                   float64
dtype: object
```

In [23]:

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

In [24]:

```
df2.head()
```

Out[24]:

	open	high	low	close	volume	Name	Daily_Price_Chage	1day % return
date								
2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL	0.1400	0.206325
2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL	0.4900	0.714688
2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL	-1.6586	-2.481344
2013-02-13	66.7442	67.6628	66.1742	66.7156	118721995	AAPL	-0.0286	-0.042869
2013-02-14	66.3599	67.3771	66.2885	66.6556	88809154	AAPL	0.2957	0.443624

In [25]:

```
df2['2013-02-08':'2013-02-13']
```

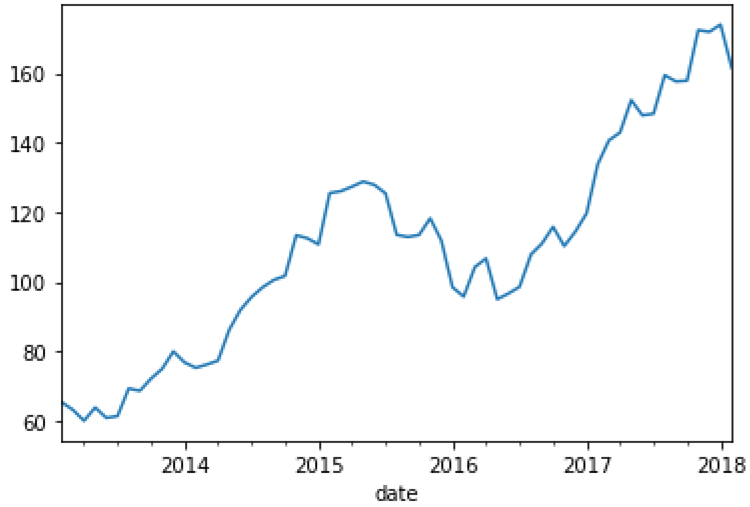
Out[25]:

	open	high	low	close	volume	Name	Daily_Price_Chage	1day % return
date								
2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL	0.1400	0.206325
2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL	0.4900	0.714688
2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL	-1.6586	-2.481344
2013-02-13	66.7442	67.6628	66.1742	66.7156	118721995	AAPL	-0.0286	-0.042869

Loading [MathJax]/extensions/Safe.js

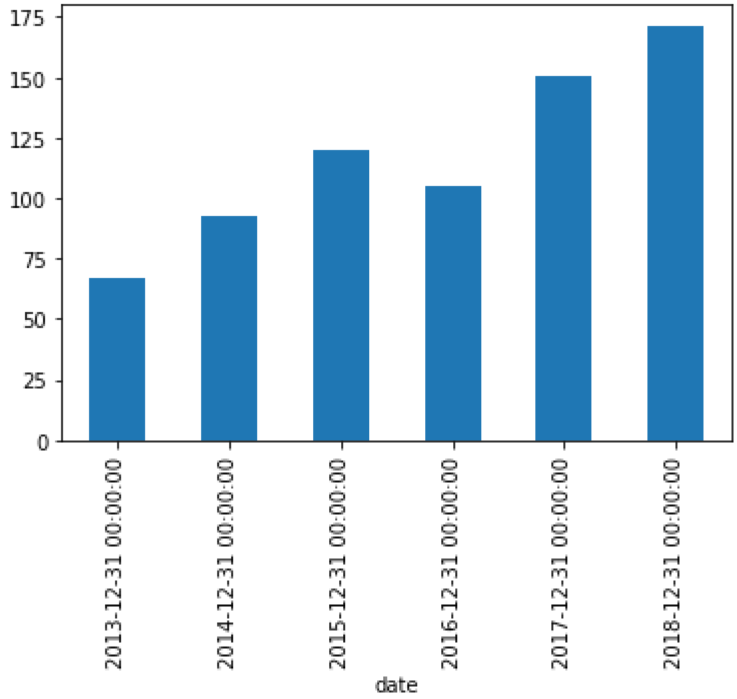
In [26]:
```
df2['close'].resample('M').mean().plot()
```

Out[26]:
```
<AxesSubplot:xlabel='date'>
```



In [27]:
```
df2['close'].resample('Y').mean().plot(kind='bar')
```

Out[27]:
```
<AxesSubplot:xlabel='date'>
```



Weather the stock prices of Amazon, Apple,Google and Microsoft are correlated or not

In [28]:
```
apple = pd.read_csv(r"X:\Data Science\Udemy Projects\S&P_500\individual_stocks_5yr/AAPL_
```

In [29]:
```
apple.head()
```

Loading [MathJax]/extensions/Safe.js

Out[29]:

```
         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
```

In [30]:

```
amazon = pd.read_csv(r"X:\Data Science\Udemy Projects\S&P_500\individual_stocks_5yr/AMZN
amazon.head()
```

Out[30]:

```
         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
```

In [31]:

```
google = pd.read_csv(r"X:\Data Science\Udemy Projects\S&P_500\individual_stocks_5yr/GOOG
google.head()
```

Out[31]:

```
         date     open    high     low   close  volume  Name
0  2014-03-27  568.000  568.00  552.92  558.46   13052  GOOG
1  2014-03-28  561.200  566.43  558.67  559.99   41003  GOOG
2  2014-03-31  566.890  567.00  556.93  556.97   10772  GOOG
3  2014-04-01  558.710  568.45  558.71  567.16    7932  GOOG
4  2014-04-02  565.106  604.83  562.19  567.00  146697  GOOG
```

In [32]:

```
microsoft = pd.read_csv(r"X:\Data Science\Udemy Projects\S&P_500\individual_stocks_5yr/M
microsoft.head()
```

Out[32]:

```
         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
```

In [33]:

```
close = pd.DataFrame()
```

In [34]:

```
close['apple'] = apple['close']
close['google']= google['close']
close['amazon'] = amazon['close']
close['microsoft'] = microsoft['close']
```

In [35]:

```
close.head()
```

Loading [MathJax]/extensions/Safe.js

Out[35]:

	apple	google	amazon	microsoft
0	67.8542	558.46	261.95	27.55
1	68.5614	559.99	257.21	27.86
2	66.8428	556.97	258.70	27.88
3	66.7156	567.16	269.47	28.03
4	66.6556	567.00	269.24	28.04

In [36]:

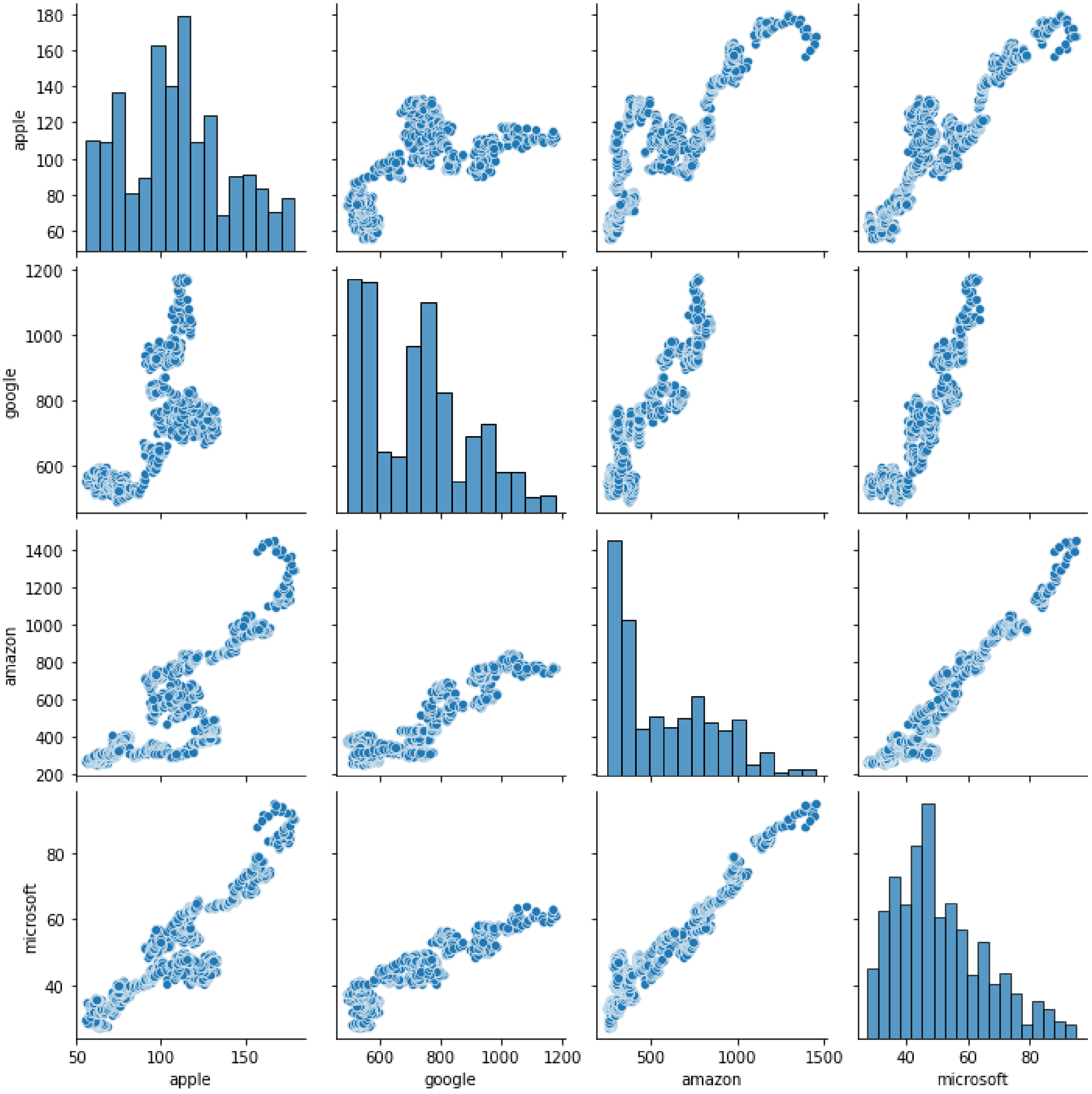
```
import seaborn as sns
```

In [37]:

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

Out[37]:

```
<seaborn.axisgrid.PairGrid at 0x244384862e0>
```



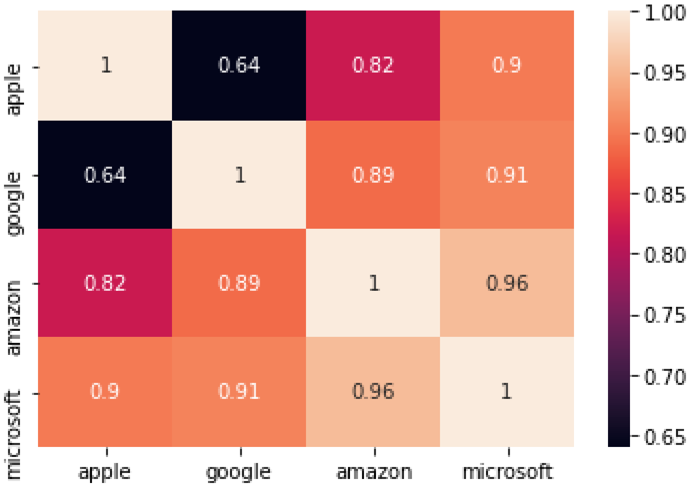
In [38]:

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

Out[38]:

```
<AxesSubplot:>
```

Loading [MathJax]/extensions/Safe.js



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

In [39]:

```
apple.head()
```

Out[39]:

	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

In [40]:

```
data = pd.DataFrame()
```

In [41]:

```
data['apple_change']= ((apple['close']-apple['open'])/apple['close'])*100
data['google_change']= ((google['close']-google['open'])/google['close'])*100
data['amazon_change']= ((amazon['close']-amazon['open'])/amazon['close'])*100
data['microsoft_change']=((microsoft['close']-microsoft['open'])/microsoft['close'])*100
```

In [42]:

```
data.head()
```

Out[42]:

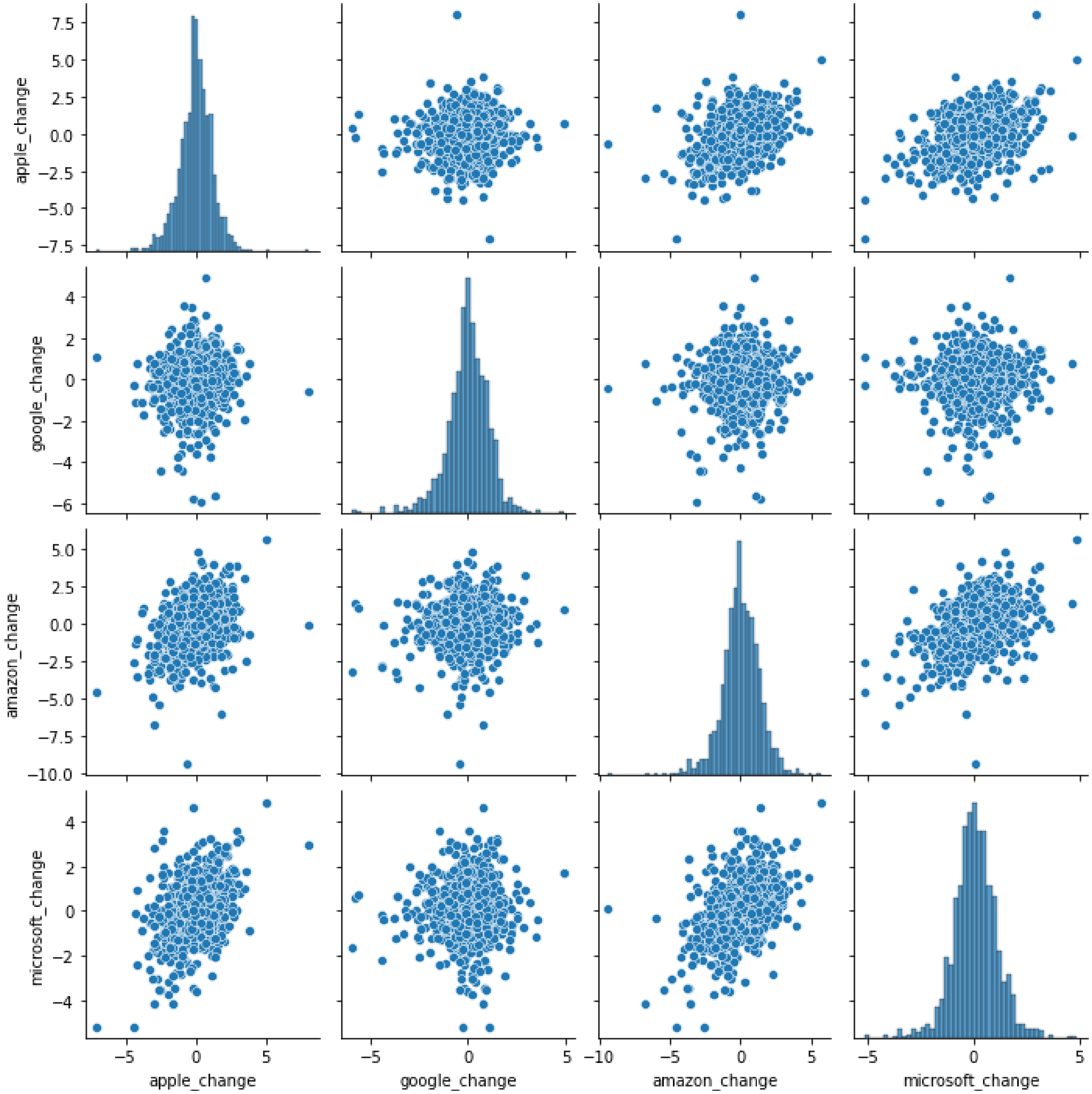
	apple_change	google_change	amazon_change	microsoft_change
0	0.206325	-1.708269	0.209964	0.725953
1	0.714688	-0.216075	-2.328836	0.753769
2	-2.481344	-1.781065	-0.189409	0.000000
3	-0.042869	1.489879	2.946525	0.356761
4	0.443624	0.334039	0.694548	0.427960

In [43]:

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

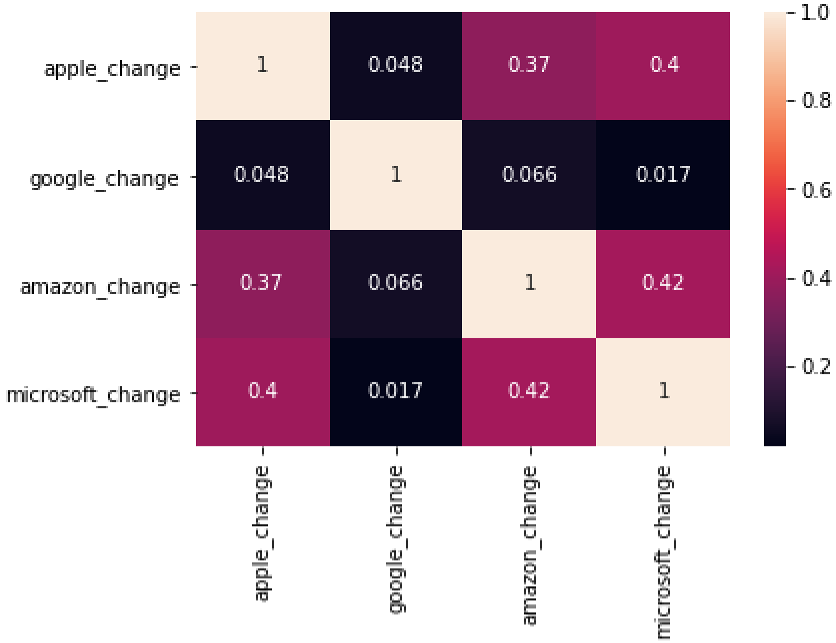
Out[43]:

```
<seaborn.axisgrid.PairGrid at 0x2443ad8db50>
```



```
In [44]: sns.heatmap(data.corr(),annot= True)
```

```
Out[44]: <AxesSubplot:>
```



Value at Risk Analysis for Tech Companies

In [45]:

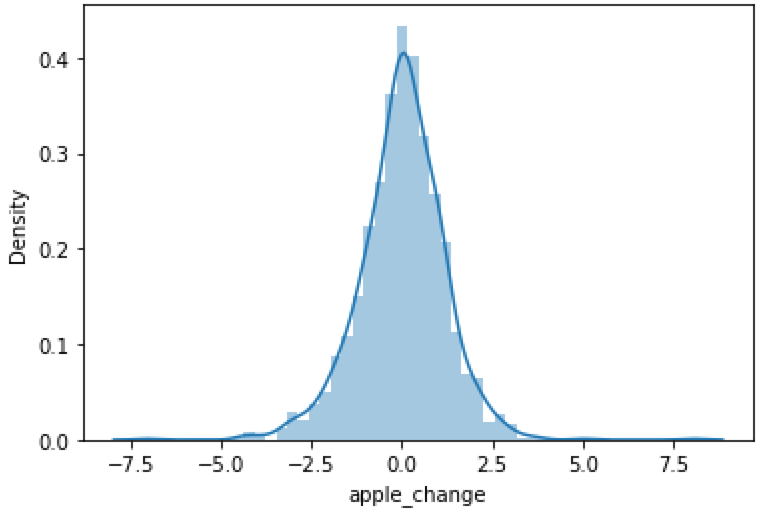
```
sns.distplot(data['apple_change'])
```

```
C:\Users\Amir\AppData\Roaming\Python\Python39\site-packages\seaborn\distributions.py:255
1: FutureWarning:

`distplot` is a deprecated function and will be removed in a future version. Please adap
t your code to use either `displot` (a figure-level function with similar flexibility) o
r `histplot` (an axes-level function for histograms).
```

Out[45]:

```
<AxesSubplot:xlabel='apple_change', ylabel='Density'>
```



In [46]:

```
data['apple_change'].std()
```

Out[46]:

```
1.1871377131421237
```

In [47]:

```
data['apple_change'].std()*2
```

Out[47]:

```
2.3742754262842474
```

In [48]:

```
data['apple_change'].std()*3
```

Out[48]:

```
3.561413139426371
```

In [50]:

```python
data['apple_change'].quantile(0.1)
```

Out[50]:

```
-1.4246644227944307
```

In [55]:

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

Out[55]:

	count	mean	std	min	25%	50%	75%	max
apple_change	1259.0	-0.000215	1.187138	-7.104299	-0.658021	0.042230	0.715427	8.000388
google_change	975.0	-0.012495	1.092560	-5.952266	-0.551963	0.024951	0.672649	4.943550
amazon_change	1259.0	-0.000398	1.358679	-9.363077	-0.738341	-0.002623	0.852568	5.640265
microsoft_change	1259.0	0.076404	1.059260	-5.177618	-0.509241	0.061069	0.703264	4.861491

In [ ]:

In [ ]:

In [ ]:

Loading [MathJax]/extensions/Safe.js