In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:
```
path='C:\individual_stocks_5yr'
company_list = ['AAPL_data.csv', 'GOOG_data.csv', 'MSFT_data.csv', 'AMZN_data.csv']

#blank dataframe
all_data = pd.DataFrame()

for file in company_list:
    current_df = pd.read_csv(path+"/"+file)
    all_data = pd.concat([all_data, current_df])

all_data.shape
```

Out[2]:
```
(4752, 7)
```

In [3]:
```
all_data.head()
```

Out[3]:

	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 [4]:
```
all_data.dtypes
```

Out[4]:
```
date       object
open      float64
high      float64
low       float64
close     float64
volume      int64
Name       object
dtype: object
```

In [5]:
```
all_data.dtypes
```

Out[5]:
```
date       object
open      float64
high      float64
low       float64
close     float64
volume      int64
Name       object
dtype: object
```

In [6]:

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

In [7]:
```
tech_list=all_data['Name'].unique()
```

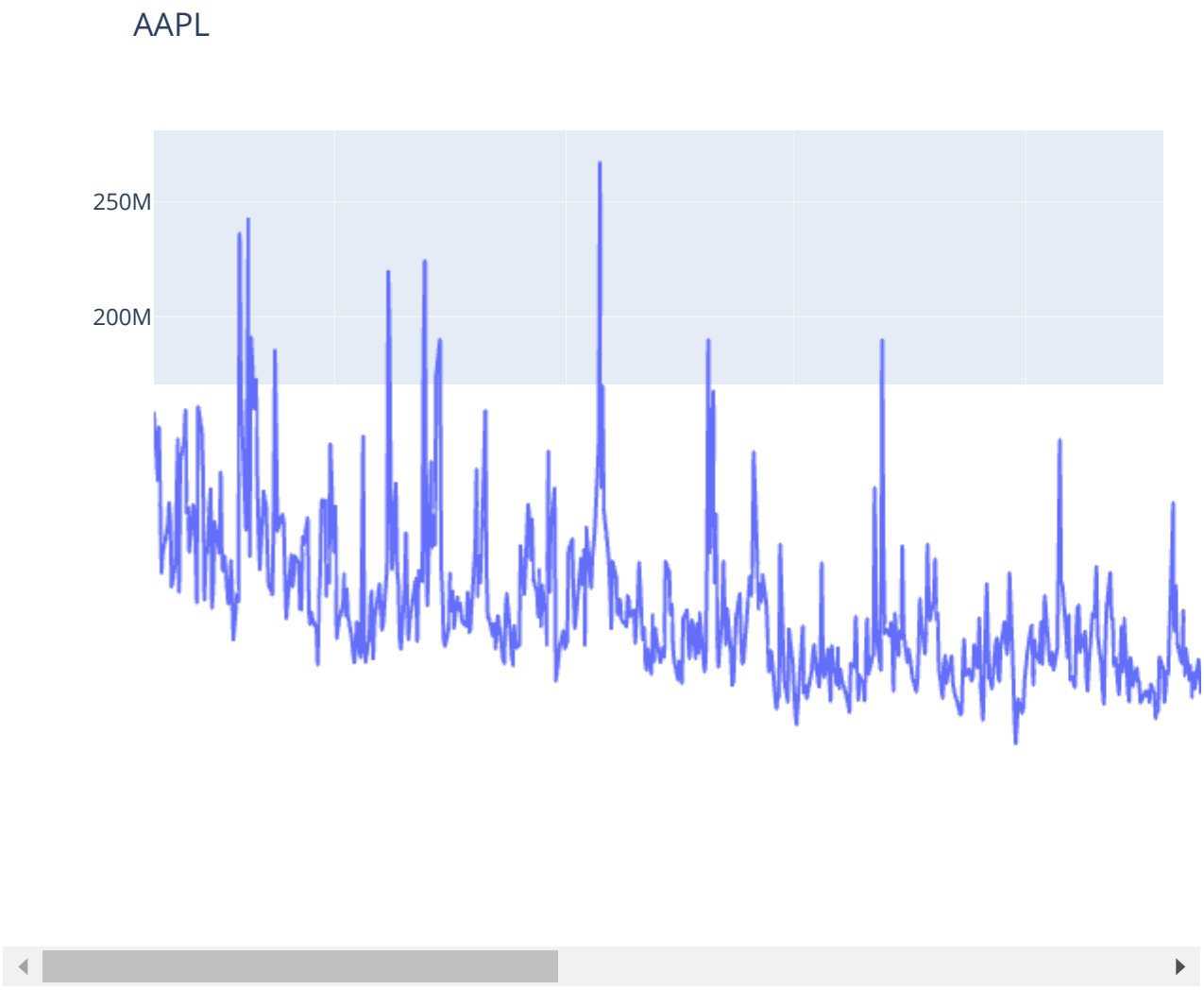
In [8]:
```
plt.figure(figsize=(20,12))
for i,company in enumerate(tech_list,1):
    plt.subplot(2,2,i)
    df=all_data[all_data['Name']==company]
    plt.plot(df['date'],df['close'])
    plt.xticks(rotation='horizontal')
    plt.title(company)
```

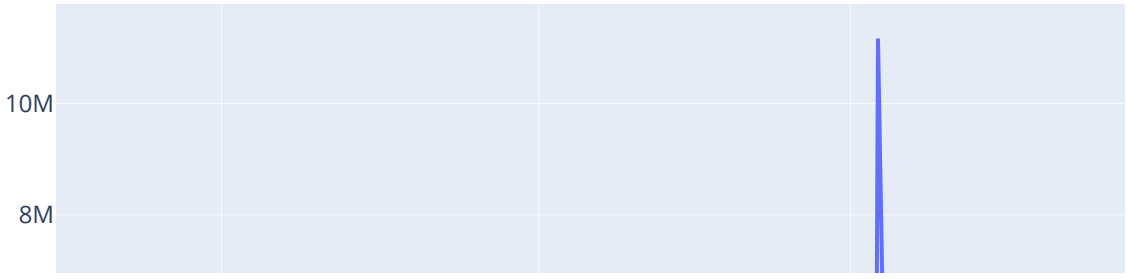
In [9]:
```
!pip install plotly
```

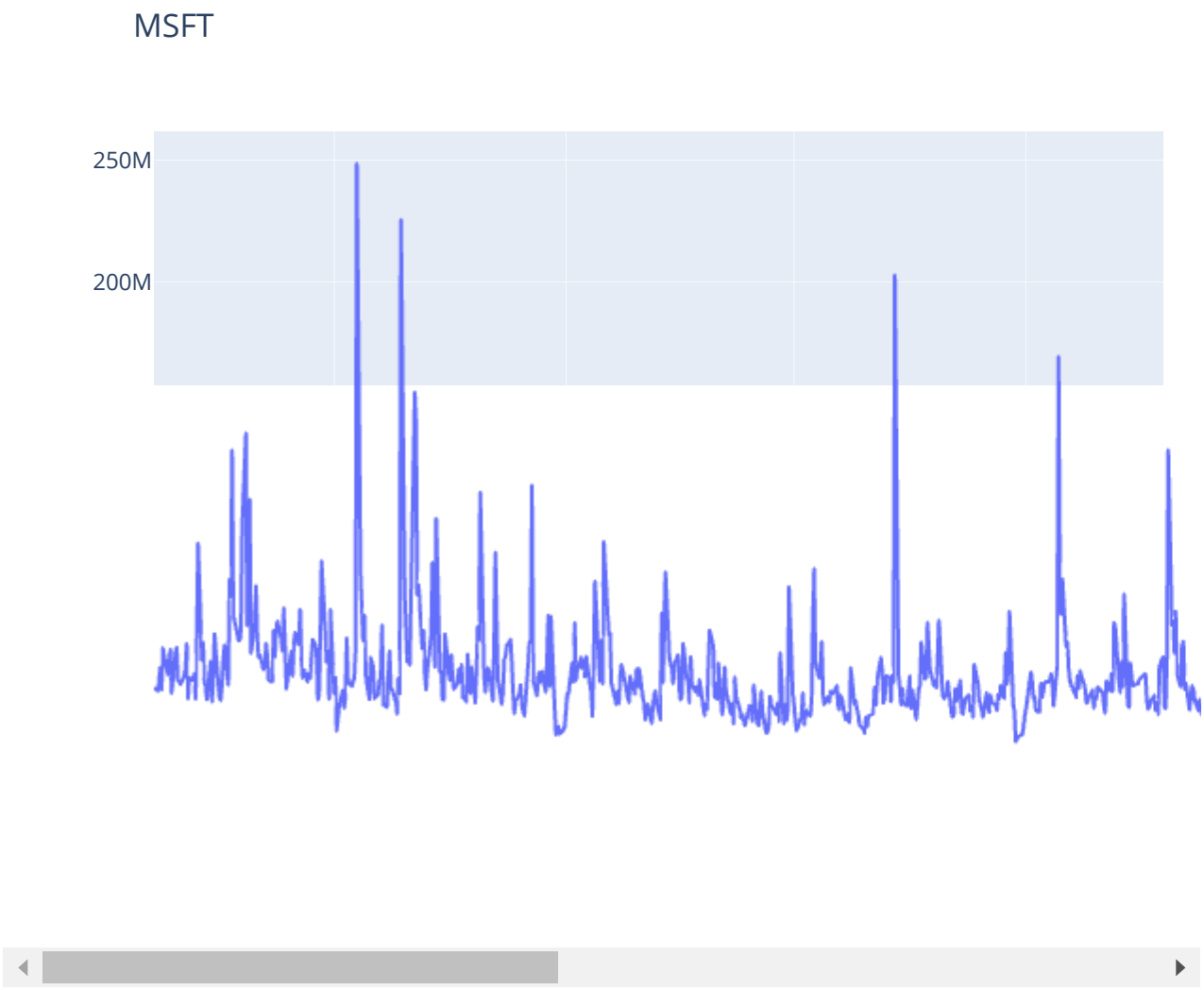
```
Requirement already satisfied: plotly in c:\users\zurie\anaconda3\lib\site-packages (5.
1.0)
Requirement already satisfied: tenacity>=6.2.0 in c:\users\zurie\anaconda3\lib\site-pack
ages (from plotly) (7.0.0)
Requirement already satisfied: six in c:\users\zurie\anaconda3\lib\site-packages (from p
lotly) (1.15.0)
```

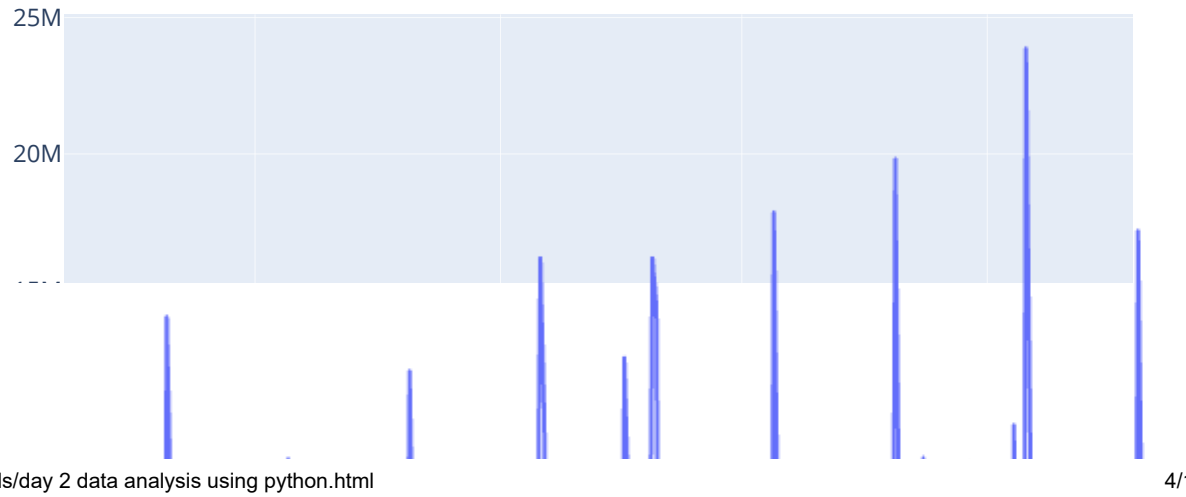
In [10]:
```
import plotly.express as px
```

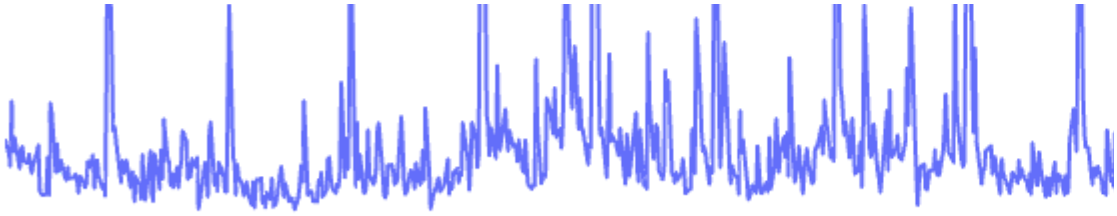
In [11]:
```
##plotly for each company
for company in tech_list:
    df=all_data[all_data['Name']==company]
    fig = px.line(df,x='date', y='volume', title=company)
    fig.show()
```


AAPL
250M
200M


GOOG
10M
8M


MSFT
250M
200M


AMZN
25M
20M



In [ ]:

```
```

In [12]:

```
df = pd.read_csv('C:\individual_stocks_5yr/AAPL_data.csv')
df.head()
```

Out[12]:

	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 [13]:

```
##analyze price change by looking for the (price)open and (price)close
df['Daily_Price_change']=df['open']-df['close']
df.head()
```

Out[13]:

	date	open	high	low	close	volume	Name	Daily_Price_change
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 [14]:

```
##analizing the daily % percentage of the price change
df['1day % return']= ((df['open']-df['close'])/df['close'])*100
```

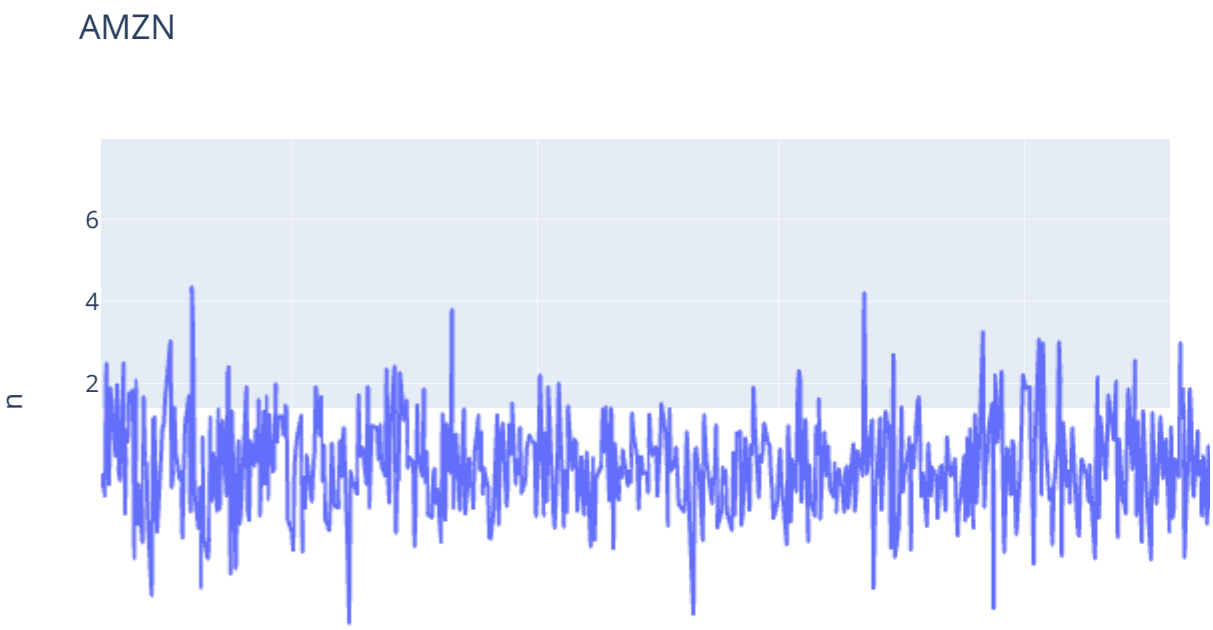
In [15]:

```
df.head()
```

Out[15]:

	date	open	high	low	close	volume	Name	Daily_Price_change	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 [27]:
```
#making plotly for 1 day percentage return
fig = px.line(df,x='date', y='1day % return', title=company)
fig.show()
```



In [ ]:

In [28]:
```
#analize monthly mean od close feature
```

In [16]:
```
df2=df.copy()
```

In [17]:
```
df2.dtypes
```

Out[17]:
```
date                  object
open                 float64
high                 float64
low                  float64
close                float64
volume                 int64
Name                  object
Daily_Price_change   float64
1day % return        float64
dtype: object
```

In [18]:
```
#because date type is a string we have to make it to datetime by..
df2['date']=pd.to_datetime(df2['date'])
```

In [19]:
```
df2.dtypes
```

Out[19]:
```
date                 datetime64[ns]
open                        float64
high                        float64
low                         float64
close                       float64
volume                        int64
Name                         object
Daily_Price_change          float64
1day % return               float64
dtype: object
```

In [20]:
```
#set yhe date as index in the chart
df2.set_index('date',inplace=True)
```

In [21]:
```
df2.head()
```

Out[21]:

date	open	high	low	close	volume	Name	Daily_Price_change	1day % return
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

date	open	high	low	close	volume	Name	Daily_Price_change	1day % return
2013-02-14	66.3599	67.3771	66.2885	66.6556	88809154	AAPL	-0.2957	-0.443624

In [22]:
```
df2['2013-02-08':'2013-02-14']
```

Out[22]:

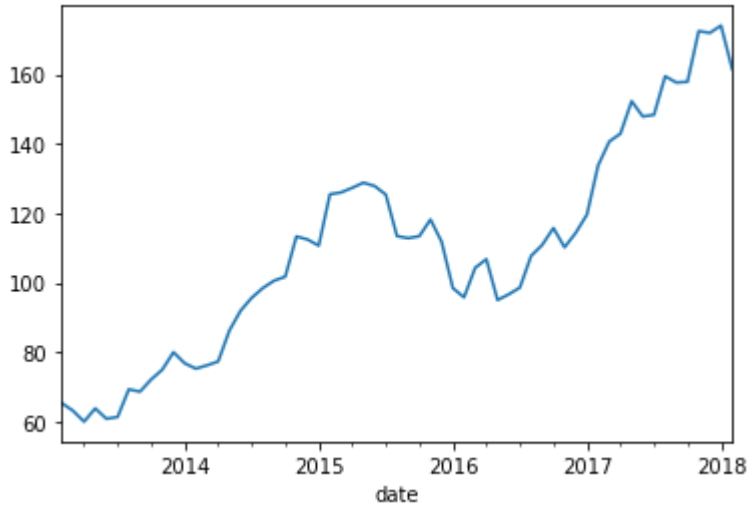
date	open	high	low	close	volume	Name	Daily_Price_change	1day % return
2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL	-0.1400	-0.206325
2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL	-0.4900	-0.714688
2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL	1.6586	2.481344
2013-02-13	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 [23]:
```
#analize monthly mean of close feature... (M) means month
df2['close'].resample('M').mean()
```

Out[23]:
```
date
2013-02-28     65.306264
2013-03-31     63.120110
2013-04-30     59.966432
2013-05-31     63.778927
2013-06-30     60.791120
                 ...
2017-10-31    157.817273
2017-11-30    172.406190
2017-12-31    171.891500
2018-01-31    174.005238
2018-02-28    161.468000
Freq: M, Name: close, Length: 61, dtype: float64
```

In [24]:
```
#make a plote for the above feature
df2['close'].resample('M').mean().plot()
```

Out[24]:
```
<AxesSubplot:xlabel='date'>
```

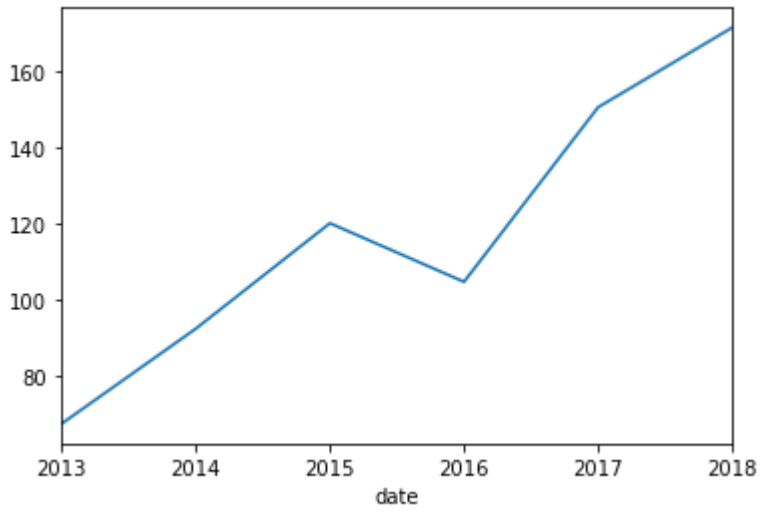


In [25]:
```
#resample data of year.. (Y) means Year
df2['close'].resample('Y').mean()
```

Out[25]:
```
date
2013-12-31     67.237839
2014-12-31     92.264531
2015-12-31    120.039861
2016-12-31    104.604008
2017-12-31    150.585080
2018-12-31    171.594231
Freq: A-DEC, Name: close, dtype: float64
```

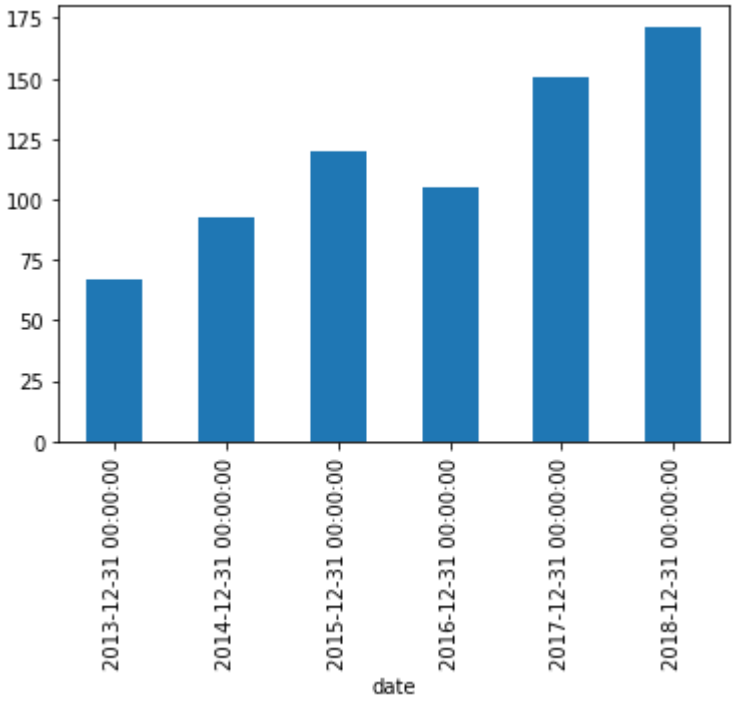
In [26]:
```
#make a plotly for the above feature
df2['close'].resample('Y').mean().plot()
```

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



In [48]:
```
#make a bar diagram for the above feature
df2['close'].resample('Y').mean().plot(kind='bar')
```

Out[48]:
```
<AxesSubplot:xlabel='date'>
```



In [58]:
```
## analize wether the stock prices of these companies are correlatrd or not
aapl=pd.read_csv('C:\individual_stocks_5yr/AAPL_data.csv')
aapl.head()
```

Out[58]:

	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 [59]:
```
amzn=pd.read_csv('C:\individual_stocks_5yr/AMZN_data.csv')
amzn.head()
```

Out[59]:

	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 [60]:
```
goog=pd.read_csv('C:\individual_stocks_5yr/GOOG_data.csv')
goog.head()
```

Out[60]:

	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 [61]:

```
msft=pd.read_csv('C:\individual_stocks_5yr/MSFT_data.csv')
msft.head()
```

Out[61]:

	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 [62]:

```
## need to get the last prices of eache company (close)
```

In [63]:

```
close=pd.DataFrame() ##making blank data frame
```

In [65]:

```
close['aapl']=aapl['close']
close['goog']=goog['close']
close['msft']=msft['close']
close['amzn']=amzn['close']
close.head() ## showing all  4 camponies prices
```

Out[65]:

	aapl	goog	msft	amzn
0	67.8542	558.46	27.55	261.95
1	68.5614	559.99	27.86	257.21
2	66.8428	556.97	27.88	258.70
3	66.7156	567.16	28.03	269.47
4	66.6556	567.00	28.04	269.24

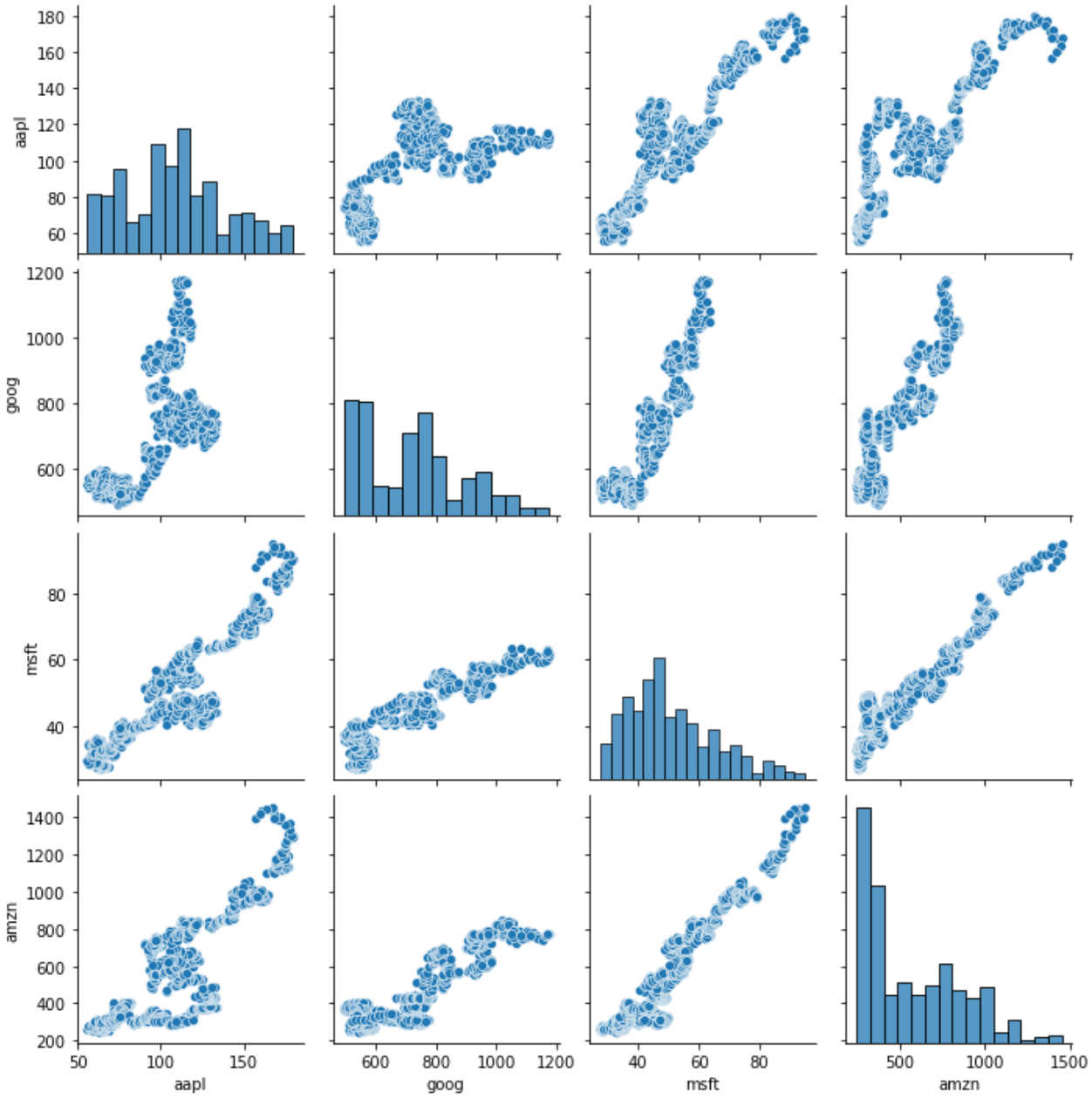
In [66]:

```
import seaborn as sns
```

In [67]:

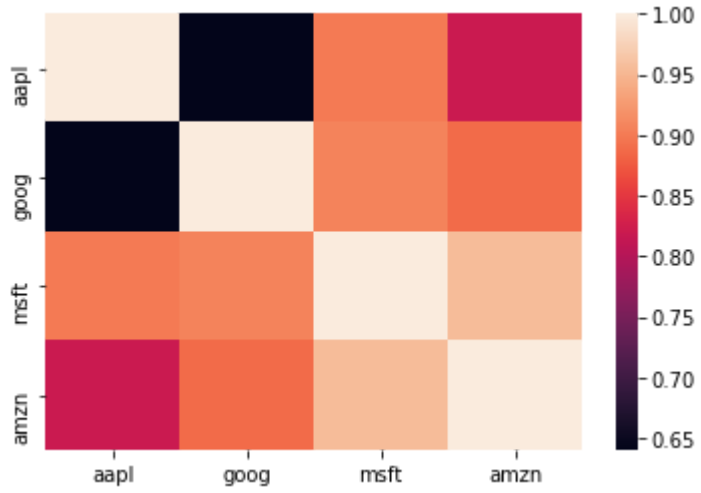
```
sns.pairplot(data=close)
```

Out[67]: <seaborn.axisgrid.PairGrid at 0x1de9a019250>



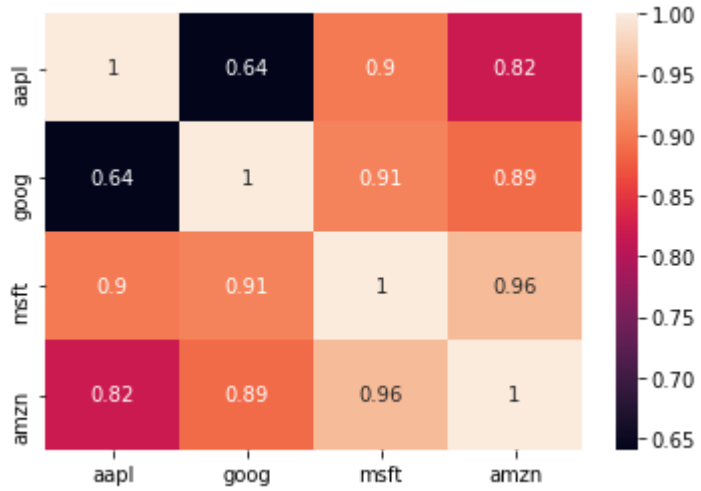
In [68]:
```
sns.heatmap(close.corr())
```

Out[68]: <AxesSubplot:>



In [69]:
```
sns.heatmap(close.corr(),annot=True)## annot mean putting valuew inside the heatmap
```

Out[69]: <AxesSubplot:>



In [70]:
```
##daily return omf each stock and how they ar correlated
##findind how prices changed by subracting open and close price
# and finding it by percentage
```

In [72]:
```
aapl.head()
```

Out[72]:

	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 [73]:
```
data=pd.DataFrame()
```

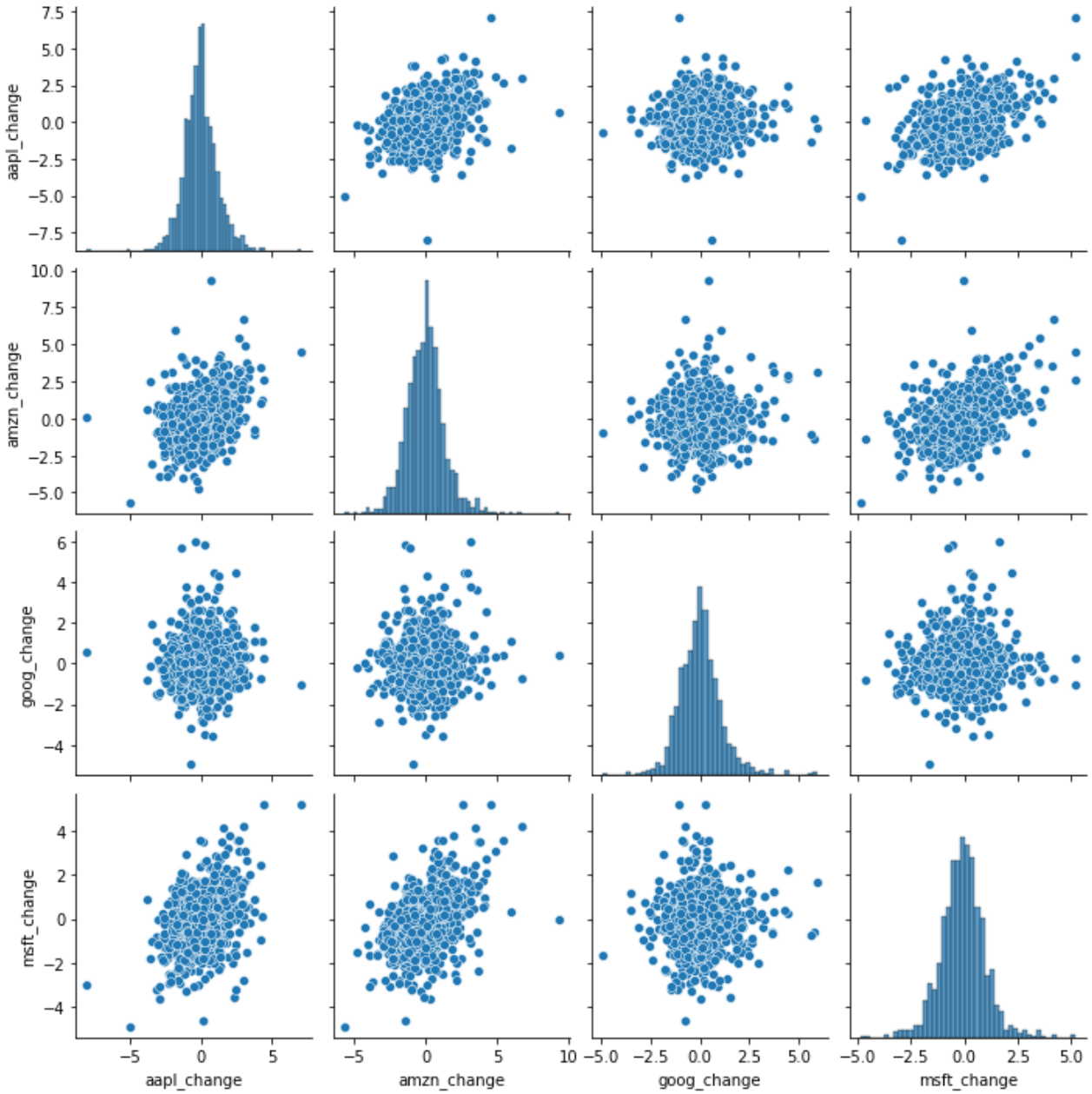
In [77]:
```
data['aapl_change']=((aapl['open']-aapl['close'])/aapl['close'])*100
data['amzn_change']=((amzn['open']-amzn['close'])/amzn['close'])*100
data['goog_change']=((goog['open']-goog['close'])/goog['close'])*100
data['msft_change']=((msft['open']-msft['close'])/msft['close'])*100
data.head()
```

Out[77]:

	aapl_change	amzn_change	goog_change	msft_change
0	-0.206325	-0.209964	1.708269	-0.725953
1	-0.714688	2.328836	0.216075	-0.753769
2	2.481344	0.189409	1.781065	0.000000
3	0.042869	-2.946525	-1.489879	-0.356761
4	-0.443624	-0.694548	-0.334039	-0.427960

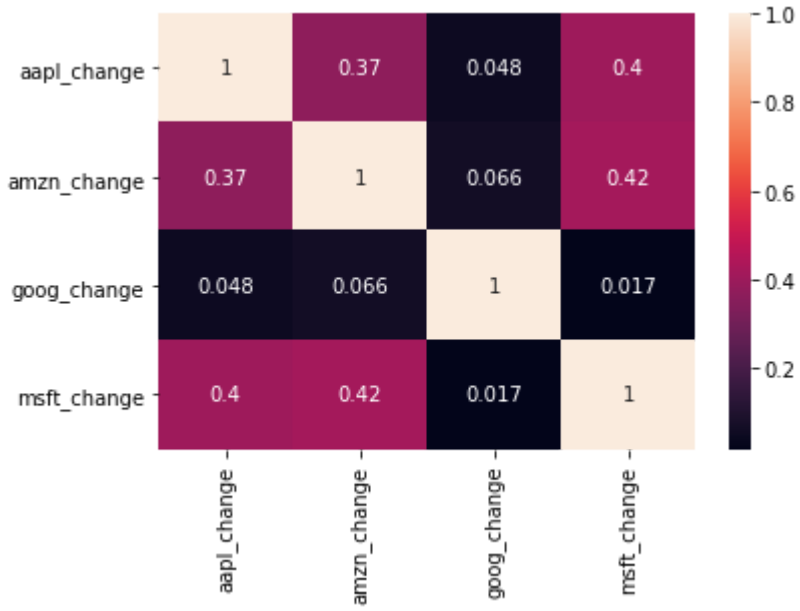
In [78]:
```
sns.pairplot(data=data)
```

Out[78]:
```
<seaborn.axisgrid.PairGrid at 0x1de9bc413a0>
```



In [79]:
```
sns.heatmap(data.corr(),annot=True)#finding correlation using heatmap
```

Out[79]: <AxesSubplot:>



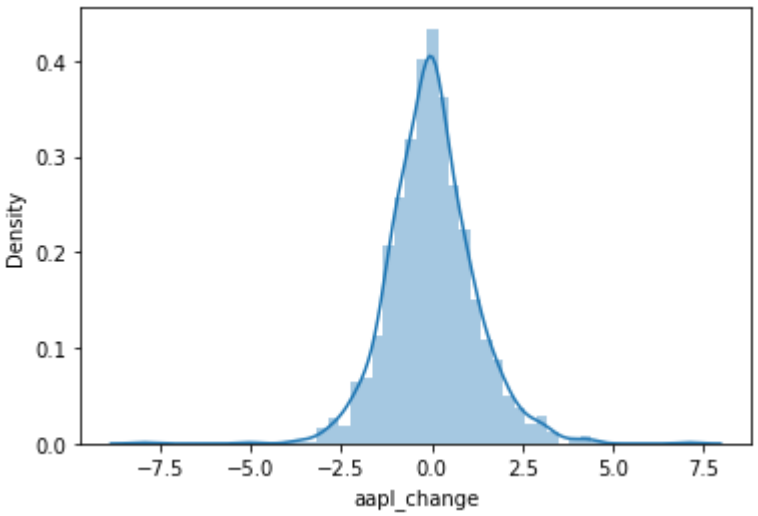
In [80]:
```
##VALUE AT RISK ANALYSIS FOR TECH COMPANIES
```

In [81]:
```
sns.distplot(data['aapl_change'])
```

```
C:\Users\zurie\anaconda3\lib\site-packages\seaborn\distributions.py:2557: 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[81]: <AxesSubplot:xlabel='aapl_change', ylabel='Density'>



In [82]:
```
data['aapl_change'].std() ## finding the standard debiation
## 68% of entire data
```

Out[82]: 1.1871377131421237

In [83]:
```
data['aapl_change'].std()*2
##95% of entire data
```

Out[83]: 2.3742754262842474

In [84]:
```
data['aapl_change'].std()*3
## 99.7% of entire data
```

Out[84]: 3.561413139426371

In [85]:
```
data.describe()
```

Out[85]:

	aapl_change	amzn_change	goog_change	msft_change
count	1259.000000	1259.000000	975.000000	1259.000000
mean	0.000215	0.000398	0.012495	-0.076404
std	1.187138	1.358679	1.092560	1.059260
min	-8.000388	-5.640265	-4.943550	-4.861491
25%	-0.715427	-0.852568	-0.672649	-0.703264
50%	-0.042230	0.002623	-0.024951	-0.061069
75%	0.658021	0.738341	0.551963	0.509241
max	7.104299	9.363077	5.952266	5.177618

In [86]:
```
data.describe().T ## to change column into raw
```

Out[86]:

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

In [ ]:
```

```