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

Collecting Data

In [2]:
```
import glob
```

In [3]:
```
glob.glob(r'C:\Users\Vaishali\Desktop\Data Analysis with Python\S&P\individual_stocks_5yr\*csv')
```

Out[3]:
```
['C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AAL_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AAPL_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AAP_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ABBV_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ABC_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ABT_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ACN_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ADBE_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ADI_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ADM_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ADP_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ADSK_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\ADS_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AEE_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AEP_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AES_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AET_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AFL_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AGN_data.csv',
```

In [4]:
```
len(glob.glob(r'C:\Users\Vaishali\Desktop\Data Analysis with Python\S&P\individual_stocks_5yr\*csv'))
```

Out[4]:
```
505
```

In [5]:
```
company_list=[
    'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AAPL_data.csv',
    'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AMZN_data.csv',
    'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\GOOGL_data.csv',
    'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\MSFT_data.csv',
]
```

In [6]:
```
all_data=pd.DataFrame()

for file in company_list:
    current_df=pd.read_csv(file)
    all_data=pd.concat([all_data,current_df],ignore_index=True)
```

In [7]:
```
all_data.shape
```

Out[7]:
```
(5036, 7)
```

In [8]:
```
all_data.head(5)
```

Out[8]:

	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 [9]:
```
all_data['Name'].unique()
```

Out[9]:
```
array(['AAPL', 'AMZN', 'GOOGL', 'MSFT'], dtype=object)
```

Analysing change in the stock prices overtime

In [10]:
```
all_data.isnull().sum()
```

Out[10]:
```
date      0
open      0
high      0
low       0
close     0
volume    0
Name      0
dtype: int64
```

In [11]:
```
all_data.dtypes
```

Out[11]:
```
date       object
open      float64
high      float64
low       float64
close     float64
volume      int64
Name       object
dtype: object
```

In [12]:
```
all_data['date']=pd.to_datetime(all_data['date'])
```

In [13]:
```
tech_list= all_data['Name'].unique()
```

In [14]:
```
tech_list
```

Out[14]:
```
array(['AAPL', 'AMZN', 'GOOGL', 'MSFT'], dtype=object)
```

In [15]:
```
plt.figure(figsize=(20,12))
for index,company in enumerate(tech_list,1):
    plt.subplot(2,2,index)
    filter1=all_data['Name']==company
    df=all_data[filter1]
    plt.plot(df['date'],df['close'])
    plt.title(company)
```

Analyzing moving average of the various stocks

In [16]:
```
all_data['close'].rolling(window=10).mean().head(14)
```

Out[16]:
```
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
5          NaN
6          NaN
7          NaN
8          NaN
9     66.03251
10    65.57280
11    65.13051
12    64.79722
13    64.43137
Name: close, dtype: float64
```

In [17]:
```
new_data=all_data.copy()
```

In [18]:
```
ma_day=[10,20,50]
for ma in ma_day:
    new_data['close_'+str(ma)]=new_data['close'].rolling(ma).mean()
```

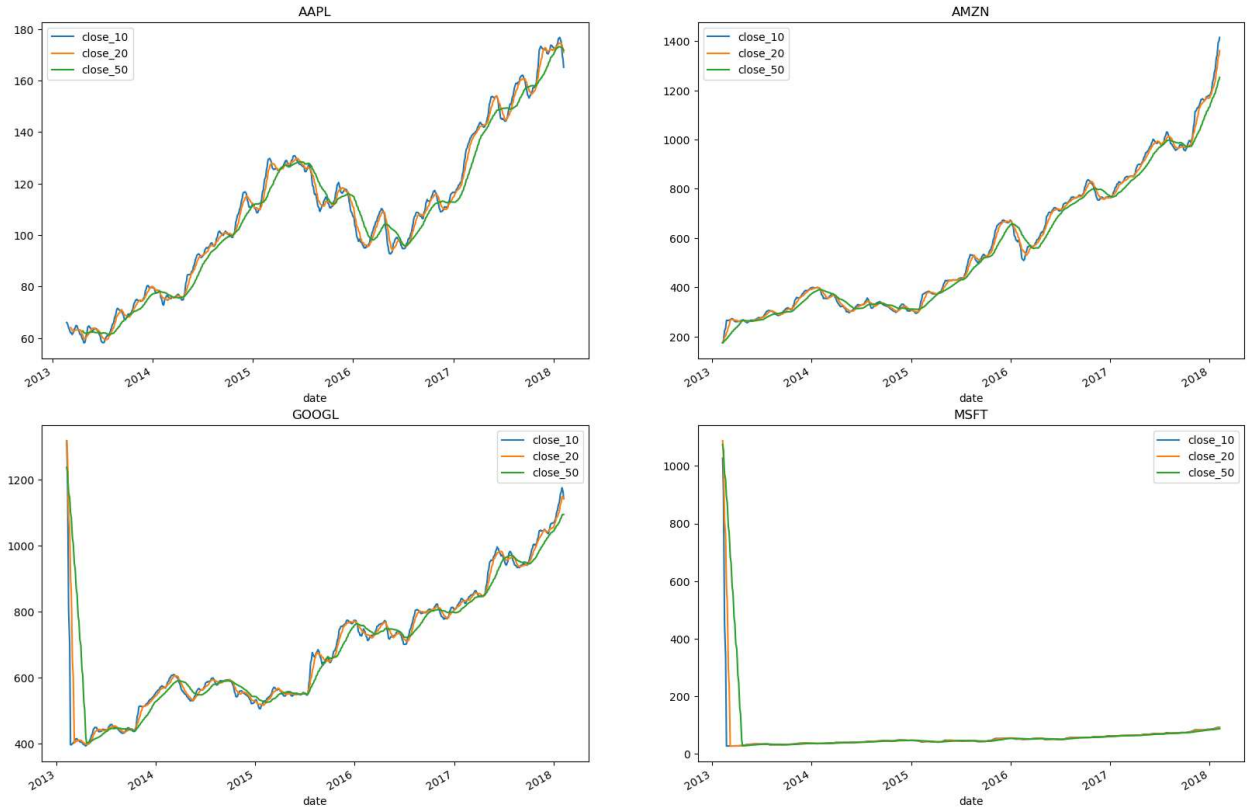
In [19]:
```
new_data.tail(7)
```

Out[19]:

	date	open	high	low	close	volume	Name	close_10	close_20	close_50
5029	2018-01-30	93.30	93.660	92.1000	92.74	38635053	MSFT	91.862	89.8285	86.5244
5030	2018-01-31	93.75	95.400	93.5100	95.01	48756338	MSFT	92.349	90.2815	86.7606
5031	2018-02-01	94.79	96.070	93.5813	94.26	47227882	MSFT	92.765	90.6770	86.9978
5032	2018-02-02	93.64	93.970	91.5000	91.78	47867753	MSFT	92.943	90.9105	87.1828
5033	2018-02-05	90.56	93.240	88.0000	88.00	51031465	MSFT	92.582	90.9010	87.2684
5034	2018-02-06	86.89	91.475	85.2500	91.33	67998564	MSFT	92.525	91.0535	87.4328
5035	2018-02-07	90.49	91.770	89.2000	89.61	41107592	MSFT	92.304	91.1230	87.5598

In [20]:
```
new_data.set_index('date',inplace=True)
```

In [21]:
```
plt.figure(figsize=(20,14))
for index,company in enumerate(tech_list,1):
    plt.subplot(2,2,index)
    filter1=new_data['Name']==company
    df=new_data[filter1]
    df[['close_10','close_20','close_50']].plot(ax=plt.gca())
    plt.title(company)
```



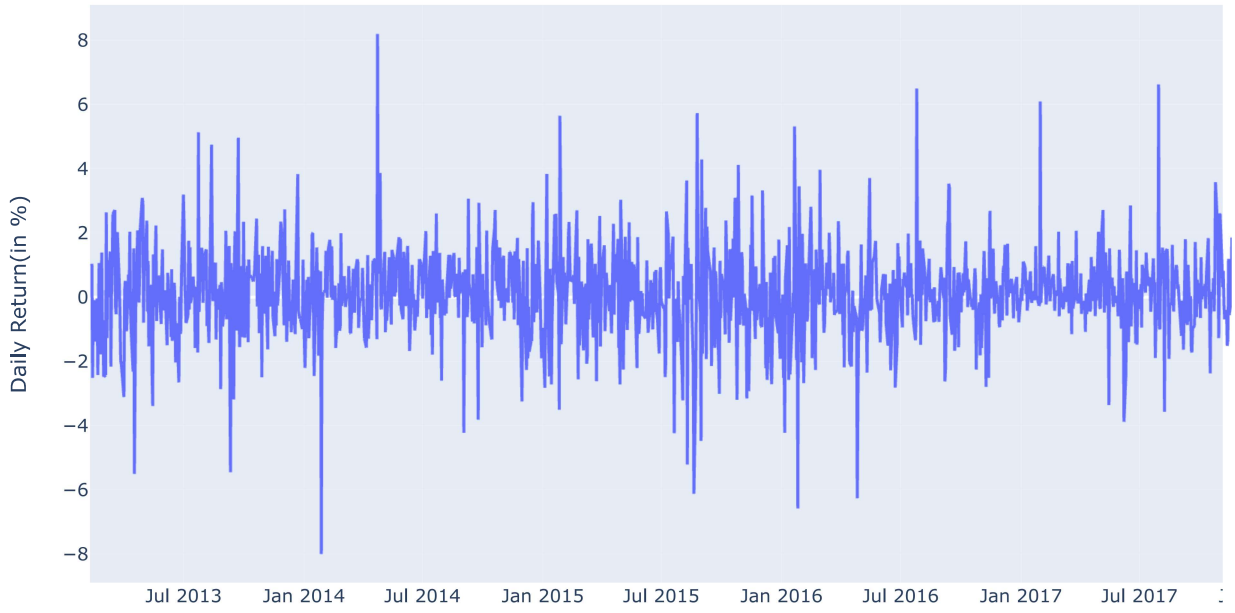
Observing closing price change in Apple stock

In [22]:
```
apple=pd.read_csv('C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AAPL_d
```

In [23]:
```
apple['Daily Return(in %)']=apple['close'].pct_change()*100
```

In [24]:
```
import plotly.express as px
```

In [25]:
```
px.line(apple,x='date',y='Daily Return(in %)')
```



Perform resampling of closing price

In [29]:
```
apple.dtypes
```

Out[29]:
```
date                  object
open                 float64
high                 float64
low                  float64
close                float64
volume                 int64
Name                  object
Daily Return(in %)   float64
dtype: object
```

In [31]:
```
apple['date']=pd.to_datetime(apple['date'])
```

In [32]:
```
apple.dtypes
```

Out[32]:
```
date                 datetime64[ns]
open                        float64
high                        float64
low                         float64
close                       float64
volume                        int64
Name                         object
Daily Return(in %)          float64
dtype: object
```

In [34]:
```
apple.set_index('date',inplace=True)
```

In [35]:
```
apple.head(4)
```

Out[35]:

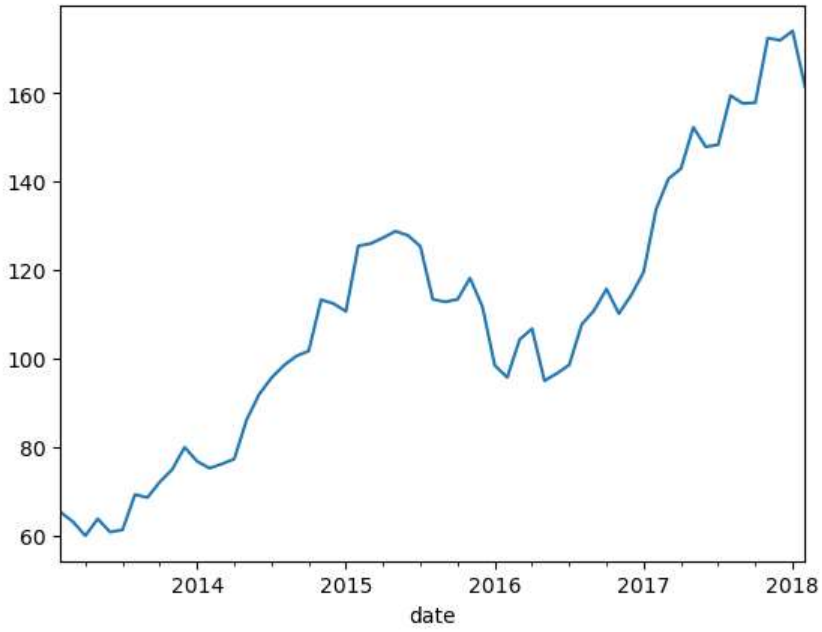
date	open	high	low	close	volume	Name	Daily Return(in %)
2013-02-08	67.7142	68.4014	66.8928	67.8542	158168416	AAPL	NaN
2013-02-11	68.0714	69.2771	67.6071	68.5614	129029425	AAPL	1.042235
2013-02-12	68.5014	68.9114	66.8205	66.8428	151829363	AAPL	-2.506658
2013-02-13	66.7442	67.6628	66.1742	66.7156	118721995	AAPL	-0.190297

In [36]:
```
apple['close'].resample('M').mean()
```

Out[36]:
```
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 [37]:
```
apple['close'].resample('M').mean().plot()
```

Out[37]:
```
<Axes: xlabel='date'>
```

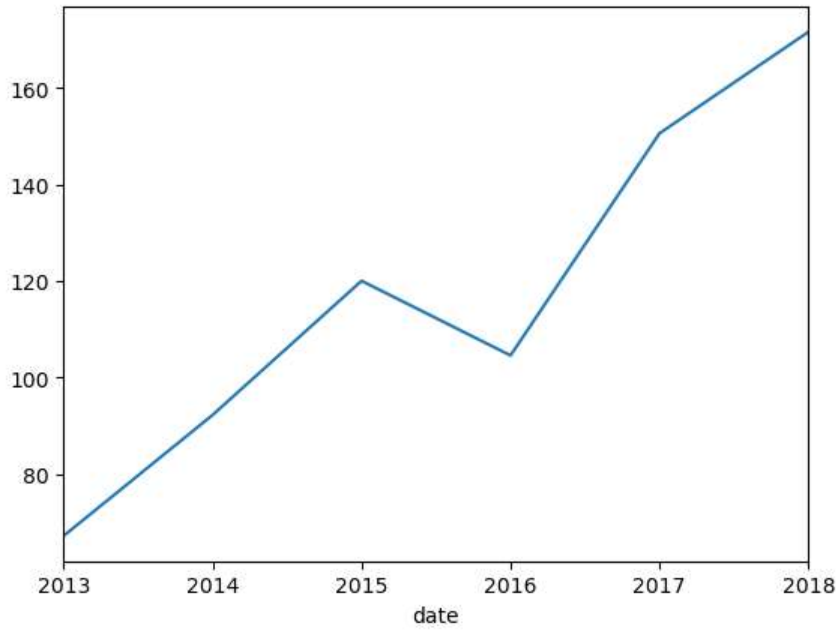


In [38]:
```
apple['close'].resample('Y').mean()
```

Out[38]:
```
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 [39]:
```
apple['close'].resample('Y').mean().plot()
```

Out[39]: <Axes: xlabel='date'>

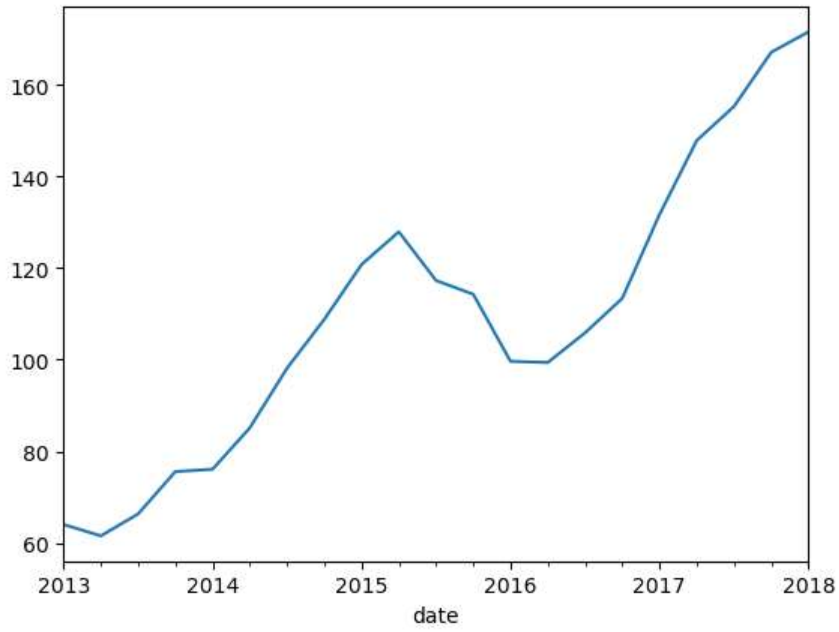


In [40]:
```
apple['close'].resample('Q').mean()
```

Out[40]:
```
date
2013-03-31     64.020291
2013-06-30     61.534692
2013-09-30     66.320670
2013-12-31     75.567478
2014-03-31     76.086293
2014-06-30     85.117475
2014-09-30     98.163311
2014-12-31    108.821016
2015-03-31    120.776721
2015-06-30    127.937937
2015-09-30    117.303438
2015-12-31    114.299297
2016-03-31     99.655082
2016-06-30     99.401250
2016-09-30    105.866094
2016-12-31    113.399048
2017-03-31    131.712500
2017-06-30    147.875397
2017-09-30    155.304603
2017-12-31    167.148254
2018-03-31    171.594231
Freq: Q-DEC, Name: close, dtype: float64
```

In [41]:
```
apple['close'].resample('Q').mean().plot()
```

Out[41]: <Axes: xlabel='date'>



Multivariate Analysis (closing price of various companies are correlated or not)

In [42]:
```
company_list
```

Out[42]:
```
['C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AAPL_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\AMZN_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\GOOGL_data.csv',
 'C:\\Users\\Vaishali\\Desktop\\Data Analysis with Python\\S&P\\individual_stocks_5yr\\MSFT_data.csv']
```

In [47]:
```
app=pd.read_csv(company_list[0])
amzn=pd.read_csv(company_list[1])
google=pd.read_csv(company_list[2])
msft=pd.read_csv(company_list[3])
```

In [48]:
```
closing_price=pd.DataFrame()
```

In [49]:
```
closing_price['apple_close']=app['close']
closing_price['amazon_close']=amzn['close']
closing_price['google_close']=google['close']
closing_price['msft_google']=msft['close']
```

In [50]:
```
closing_price
```

Out[50]:

	apple_close	amazon_close	google_close	msft_google
0	67.8542	261.95	393.0777	27.55
1	68.5614	257.21	391.6012	27.86
2	66.8428	258.70	390.7403	27.88
3	66.7156	269.47	391.8214	28.03
4	66.6556	269.24	394.3039	28.04
...	...	...	...	...
1254	167.7800	1390.00	1181.5900	94.26
1255	160.5000	1429.95	1119.2000	91.78
1256	156.4900	1390.00	1062.3900	88.00
1257	163.0300	1442.84	1084.4300	91.33
1258	159.5400	1416.78	1055.4100	89.61

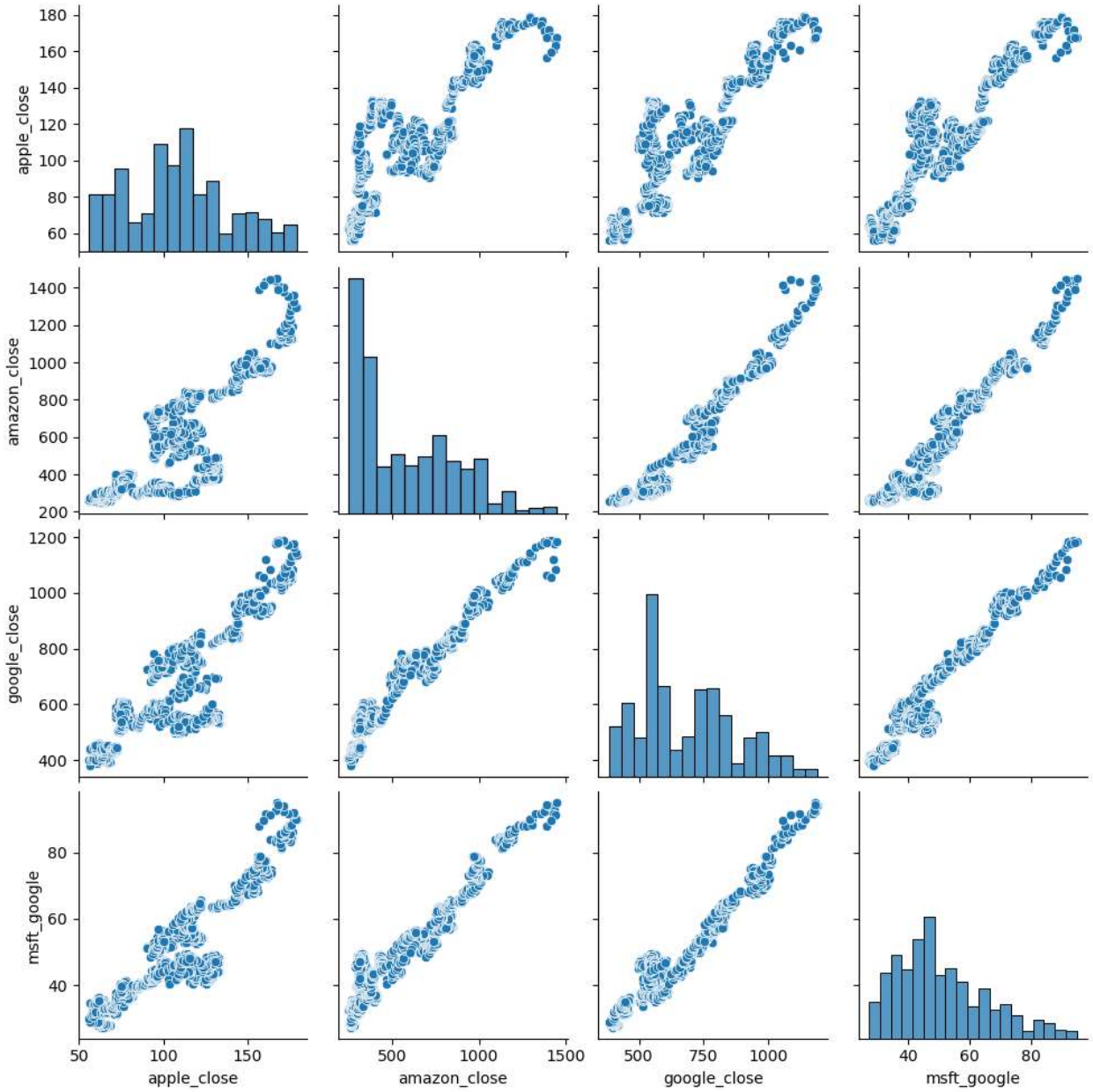
1259 rows × 4 columns

In [51]:
```
sns.pairplot(closing_price)
```

```
C:\Users\Vaishali\anaconda3\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning:

The figure layout has changed to tight
```

Out[51]:
```
<seaborn.axisgrid.PairGrid at 0x1d4099f2e50>
```



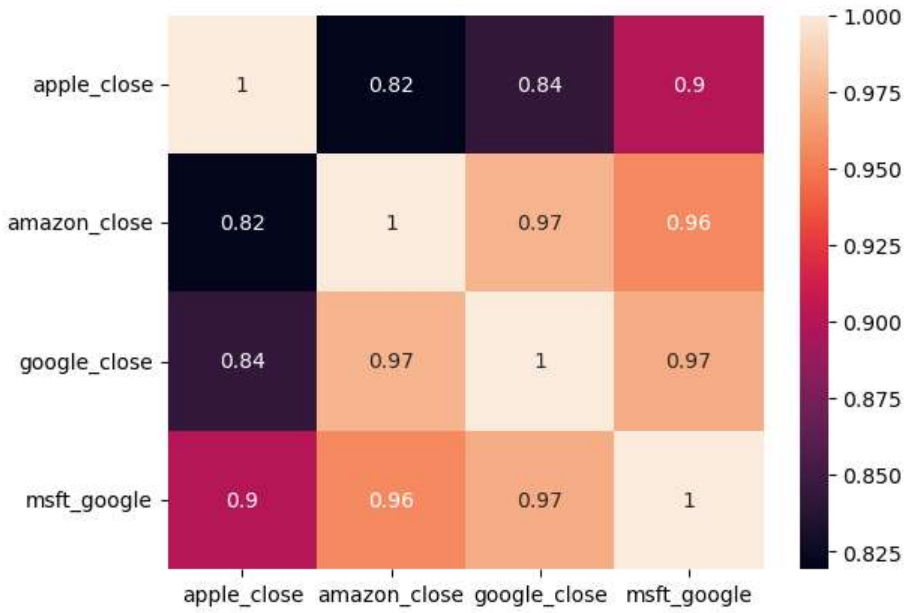
In [52]:
```
closing_price.corr()
```

Out[52]:

	apple_close	amazon_close	google_close	msft_google
apple_close	1.000000	0.819078	0.843736	0.899689
amazon_close	0.819078	1.000000	0.973988	0.955977
google_close	0.843736	0.973988	1.000000	0.970779
msft_google	0.899689	0.955977	0.970779	1.000000

In [54]:
```
sns.heatmap(closing_price.corr(),annot=True)
```

Out[54]: <Axes: >



Correlation Analysis

In [55]:
```
(closing_price['apple_close']-closing_price['apple_close'].shift(1))/closing_price['apple_close'].shift(1)*100
```

Out[55]:
```
0            NaN
1       1.042235
2      -2.506658
3      -0.190297
4      -0.089934
          ...
1254    0.209043
1255   -4.339015
1256   -2.498442
1257    4.179181
1258   -2.140710
Name: apple_close, Length: 1259, dtype: float64
```

In [58]:
```
for col in closing_price.columns:
    closing_price[col +'_pct_change']=(closing_price[col]-closing_price[col].shift(1))/closing_price[col].shi
```

In [59]:
```
closing_price
```

Out[59]:

	apple_close	amazon_close	google_close	msft_google	apple_close_pct_change	amazon_close_pct_change	google_close_pct_cha
0	67.8542	261.95	393.0777	27.55	NaN	NaN	N
1	68.5614	257.21	391.6012	27.86	1.042235	-1.809506	-0.375
2	66.8428	258.70	390.7403	27.88	-2.506658	0.579293	-0.219
3	66.7156	269.47	391.8214	28.03	-0.190297	4.163123	0.276
4	66.6556	269.24	394.3039	28.04	-0.089934	-0.085353	0.633
...	...	...	...	...	...	...	
1254	167.7800	1390.00	1181.5900	94.26	0.209043	-4.196734	-0.053
1255	160.5000	1429.95	1119.2000	91.78	-4.339015	2.874101	-5.280
1256	156.4900	1390.00	1062.3900	88.00	-2.498442	-2.793804	-5.075
1257	163.0300	1442.84	1084.4300	91.33	4.179181	3.801439	2.074
1258	159.5400	1416.78	1055.4100	89.61	-2.140710	-1.806160	-2.676

1259 rows × 8 columns

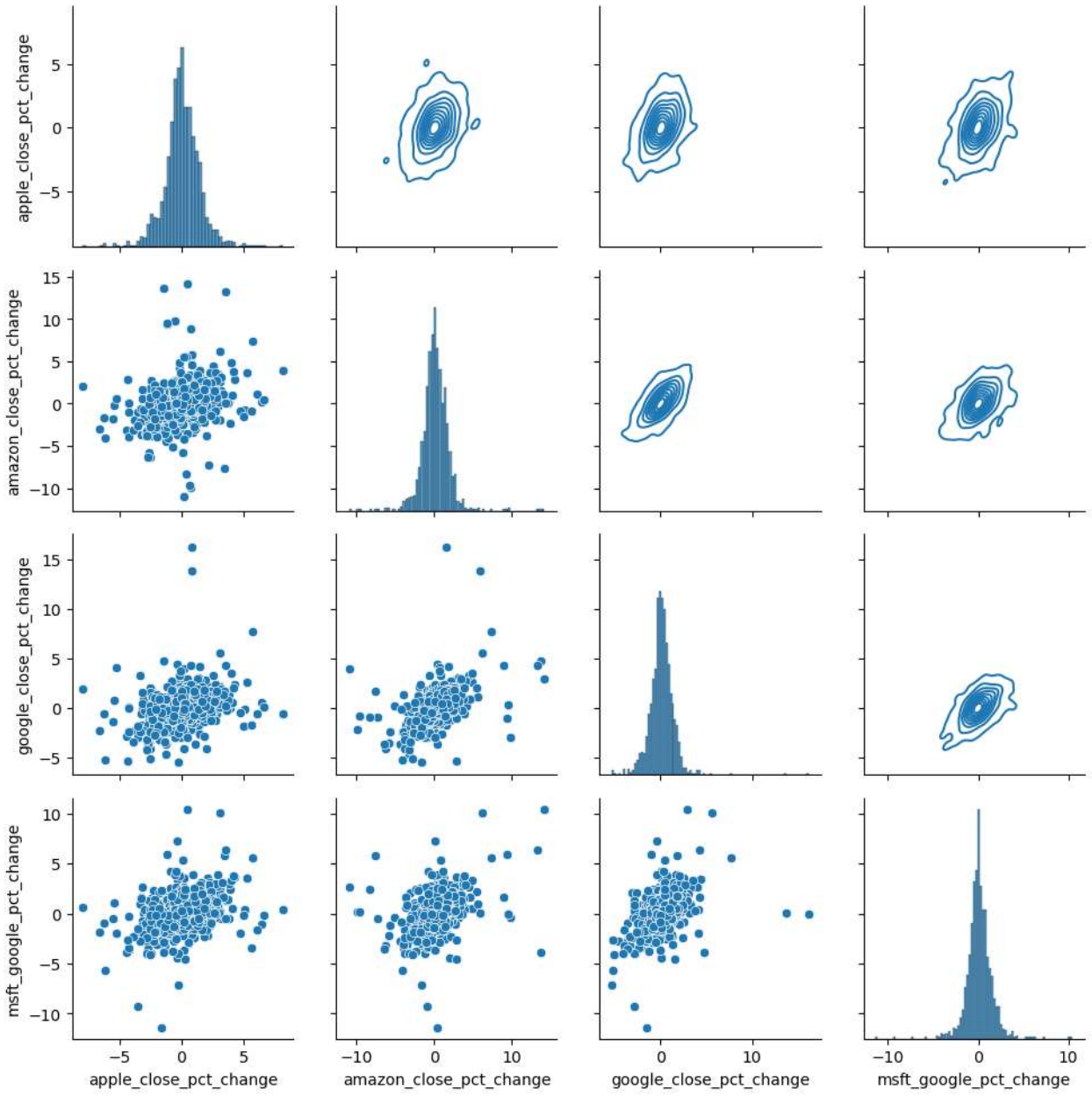
In [60]:
```
closing_price.columns
```

Out[60]:
```
Index(['apple_close', 'amazon_close', 'google_close', 'msft_google',
       'apple_close_pct_change', 'amazon_close_pct_change',
       'google_close_pct_change', 'msft_google_pct_change'],
      dtype='object')
```

In [63]:
```
closing_pct=closing_price[['apple_close_pct_change', 'amazon_close_pct_change',
       'google_close_pct_change', 'msft_google_pct_change']]
```

In [68]:
```
g=sns.PairGrid(closing_pct)
g.map_diag(sns.histplot)
g.map_lower(sns.scatterplot)
g.map_upper(sns.kdeplot)
```

Out[68]: <seaborn.axisgrid.PairGrid at 0x1d414cbe750>



In [69]:
```
closing_pct.corr()
```

Out[69]:

	apple_close_pct_change	amazon_close_pct_change	google_close_pct_change	msft_google_pct_change
apple_close_pct_change	1.000000	0.287659	0.348858	0.366598
amazon_close_pct_change	0.287659	1.000000	0.548423	0.402678
google_close_pct_change	0.348858	0.548423	1.000000	0.488373
msft_google_pct_change	0.366598	0.402678	0.488373	1.000000

In [ ]: