# Data Collection

In [ ]:

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

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

In [3]:
```
glob.glob(r'C:\Users\lenovo\Downloads\S&P_resources\individual_stocks_5yr/*c
```

Out[3]:
```
['C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AA
L_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AA
PL_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AA
P_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AB
BV_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AB
C_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AB
T_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AC
N_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AD
BE_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AD
I_data.csv',
 'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AD
```

In [4]:
```
len (glob.glob(r'C:\Users\lenovo\Downloads\S&P_resources\individual_stocks_5
```

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

In [ ]:

In [5]:
```
company_list= [
    r'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AA
    r'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\AM
    r'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\GO
    r'C:\\Users\\lenovo\\Downloads\\S&P_resources\\individual_stocks_5yr\\MS
]
```

In [6]:
```
import warnings
from warnings import filterwarnings
filterwarnings('ignore')
```

In [7]:
```
all_data = pd.DataFrame()

for file in company_list:

    current_df = pd.read_csv(file)

    all_data =  current_df.append(all_data , ignore_index=True)
    ##full_df = pd.concat([full_df , current_df] , ignore_index=True)
```

In [8]:
```
all_data.shape
```

Out[8]:
```
(4752, 7)
```

In [9]:
```
all_data.head(6)
```

Out[9]:

| | date | open | high | low | close | volume | Name |
|---|---|---|---|---|---|---|---|
| **0** | 2013-02-08 | 27.35 | 27.71 | 27.310 | 27.55 | 33318306 | MSFT |
| **1** | 2013-02-11 | 27.65 | 27.92 | 27.500 | 27.86 | 32247549 | MSFT |
| **2** | 2013-02-12 | 27.88 | 28.00 | 27.750 | 27.88 | 35990829 | MSFT |
| **3** | 2013-02-13 | 27.93 | 28.11 | 27.880 | 28.03 | 41715530 | MSFT |
| **4** | 2013-02-14 | 27.92 | 28.06 | 27.870 | 28.04 | 32663174 | MSFT |
| **5** | 2013-02-15 | 28.04 | 28.16 | 27.875 | 28.01 | 49650538 | MSFT |

In [ ]:

In [10]:
```
all_data['Name'].unique()
```

Out[10]:
```
array(['MSFT', 'GOOG', 'AMZN', 'AAPL'], dtype=object)
```

In [ ]:

# What was the change in Price of the Stock overtime

In [ ]:

In [11]:
```
all_data.isnull().sum()
```

Out[11]:
```
date      0
open      0
high      0
low       0
close     0
volume    0
Name      0
dtype: int64
```

In [12]:
```
all_data.dtypes
```

Out[12]:
```
date       object
open      float64
high      float64
low       float64
close     float64
volume      int64
Name       object
dtype: object
```

In [ ]:

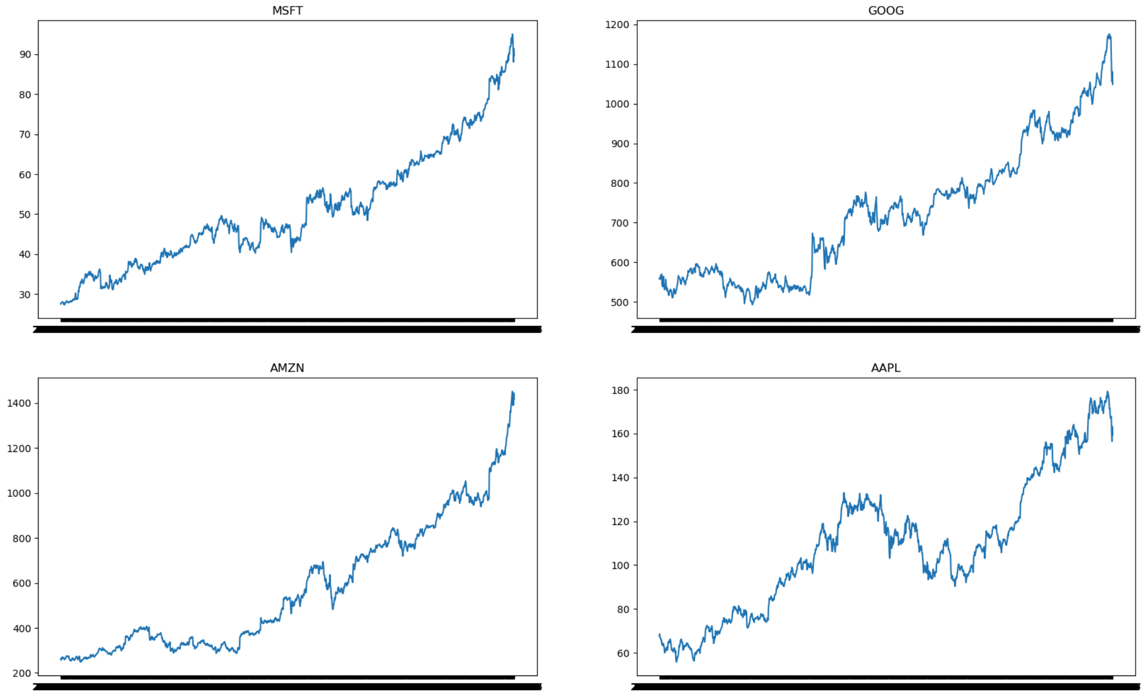
In [13]:
```
tech_list= all_data['Name'].unique()
```

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

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

In [ ]:

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)
```




In [ ]:

In [ ]:

# What was the moving average of the various Stocks

In [ ]:

In [16]:
```
all_data.head(15)
```

Out[16]:

| | date | open | high | low | close | volume | Name |
|---|---|---|---|---|---|---|---|
| **0** | 2013-02-08 | 27.3500 | 27.71 | 27.310 | 27.550 | 33318306 | MSFT |
| **1** | 2013-02-11 | 27.6500 | 27.92 | 27.500 | 27.860 | 32247549 | MSFT |
| **2** | 2013-02-12 | 27.8800 | 28.00 | 27.750 | 27.880 | 35990829 | MSFT |
| **3** | 2013-02-13 | 27.9300 | 28.11 | 27.880 | 28.030 | 41715530 | MSFT |
| **4** | 2013-02-14 | 27.9200 | 28.06 | 27.870 | 28.040 | 32663174 | MSFT |
| **5** | 2013-02-15 | 28.0400 | 28.16 | 27.875 | 28.010 | 49650538 | MSFT |
| **6** | 2013-02-19 | 27.8801 | 28.09 | 27.800 | 28.045 | 38804616 | MSFT |
| **7** | 2013-02-20 | 28.1300 | 28.20 | 27.830 | 27.870 | 44109412 | MSFT |
| **8** | 2013-02-21 | 27.7400 | 27.74 | 27.230 | 27.490 | 49078338 | MSFT |
| **9** | 2013-02-22 | 27.6800 | 27.76 | 27.480 | 27.760 | 31425726 | MSFT |
| **10** | 2013-02-25 | 27.9700 | 28.05 | 27.370 | 27.370 | 48011248 | MSFT |
| **11** | 2013-02-26 | 27.3800 | 27.60 | 27.340 | 27.370 | 49917353 | MSFT |
| **12** | 2013-02-27 | 27.4200 | 28.00 | 27.330 | 27.810 | 36390889 | MSFT |
| **13** | 2013-02-28 | 27.8800 | 27.97 | 27.740 | 27.800 | 35836861 | MSFT |
| **14** | 2013-03-01 | 27.7200 | 27.98 | 27.520 | 27.950 | 34849287 | MSFT |

In [ ]:

In [17]:
```
all_data['close'].rolling(window=10).mean().head(14)
```

Out[17]:
```
0          NaN
1          NaN
2          NaN
3          NaN
4          NaN
5          NaN
6          NaN
7          NaN
8          NaN
9      27.8535
10     27.8355
11     27.7865
12     27.7795
13     27.7565
Name: close, dtype: float64
```

In [18]:
```
new_data = all_data.copy()
```

In [19]:
```
ma_day= [10, 20, 50]

for ma in ma_day:
    new_data['close_'+str(ma)] = new_data['close'].rolling(ma).mean()
```

In [20]:
```
new_data.tail(7)
```

Out[20]:

| | date | open | high | low | close | volume | Name | close_10 | close_20 | close |
|---|---|---|---|---|---|---|---|---|---|---|
| **4745** | 2018-01-30 | 165.525 | 167.3700 | 164.7000 | 166.97 | 46048185 | AAPL | 174.263 | 174.3340 | 172.9 |
| **4746** | 2018-01-31 | 166.870 | 168.4417 | 166.5000 | 167.43 | 32478930 | AAPL | 173.096 | 174.0925 | 172.8 |
| **4747** | 2018-02-01 | 167.165 | 168.6200 | 166.7600 | 167.78 | 47230787 | AAPL | 171.948 | 173.8700 | 172.8 |
| **4748** | 2018-02-02 | 166.000 | 166.8000 | 160.1000 | 160.50 | 86593825 | AAPL | 170.152 | 173.2435 | 172.6 |
| **4749** | 2018-02-05 | 159.100 | 163.8800 | 156.0000 | 156.49 | 72738522 | AAPL | 168.101 | 172.3180 | 172.3 |
| **4750** | 2018-02-06 | 154.830 | 163.7200 | 154.0000 | 163.03 | 68243838 | AAPL | 166.700 | 171.7520 | 172.0 |
| **4751** | 2018-02-07 | 163.085 | 163.4000 | 159.0685 | 159.54 | 51608580 | AAPL | 165.232 | 171.0125 | 171.7 |

In [ ]:

In [21]:
```
new_data.set_index('date', inplace=True)
```

In [22]:
```
new_data.columns
```

Out[22]:
```
Index(['open', 'high', 'low', 'close', 'volume', 'Name', 'close_10',
       'close_20', 'close_50'],
      dtype='object')
```

In [23]:
```
plt.figure(figsize=(20,12))
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)
```


In [ ]:

In [ ]:

# Analyse closing price change in apple stock

In [ ]:

In [24]:
```
company_list
```

Out[24]:
```
['C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\AAPL_data.csv',
 'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\AMZN_data.csv',
 'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\GOOG_data.csv',
 'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\MSFT_data.csv']
```

In [25]:
```
apple = pd.read_csv(r'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\
```

In [26]:
```
apple.head(4)
```

Out[26]:

| | date | open | high | low | close | volume | Name |
|---|---|---|---|---|---|---|---|
| **0** | 2013-02-08 | 15.07 | 15.12 | 14.63 | 14.75 | 8407500 | AAL |
| **1** | 2013-02-11 | 14.89 | 15.01 | 14.26 | 14.46 | 8882000 | AAL |
| **2** | 2013-02-12 | 14.45 | 14.51 | 14.10 | 14.27 | 8126000 | AAL |
| **3** | 2013-02-13 | 14.30 | 14.94 | 14.25 | 14.66 | 10259500 | AAL |

In [27]:
```
apple['close']
```

Out[27]:
```
0       14.75
1       14.46
2       14.27
3       14.66
4       13.99
        ...
1254    53.88
1255    52.10
1256    49.76
1257    51.18
1258    51.40
Name: close, Length: 1259, dtype: float64
```

In [ ]:

In [28]:
```
apple['daily retuen(in%)'] = apple['close'].pct_change() * 100
```

In [29]:
```
apple.head(4)
```

Out[29]:

| | date | open | high | low | close | volume | Name | daily retuen(in%) |
|---|---|---|---|---|---|---|---|---|
| **0** | 2013-02-08 | 15.07 | 15.12 | 14.63 | 14.75 | 8407500 | AAL | NaN |
| **1** | 2013-02-11 | 14.89 | 15.01 | 14.26 | 14.46 | 8882000 | AAL | -1.966102 |
| **2** | 2013-02-12 | 14.45 | 14.51 | 14.10 | 14.27 | 8126000 | AAL | -1.313970 |
| **3** | 2013-02-13 | 14.30 | 14.94 | 14.25 | 14.66 | 10259500 | AAL | 2.733006 |

In [ ]:

In [30]:
```
import plotly.express as px
```

In [31]:
```
px.line(apple , x='date' , y='daily retuen(in%)')
```

In [ ]:

In [ ]:

# Performing resampling analysis of closing price..

In [ ]:

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

Out[32]:
```
date                   object
open                  float64
high                  float64
low                   float64
close                 float64
volume                  int64
Name                   object
daily retuen(in%)     float64
dtype: object
```

In [33]:
```
apple['date'] = pd.to_datetime(apple['date'])
```

In [34]:
```
apple.dtypes
```

Out[34]:
```
date                 datetime64[ns]
open                        float64
high                        float64
low                         float64
close                       float64
volume                        int64
Name                         object
daily retuen(in%)           float64
dtype: object
```

In [ ]:

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

Out[35]:

| | date | open | high | low | close | volume | Name | daily retuen(in%) |
|---|---|---|---|---|---|---|---|---|
| **0** | 2013-02-08 | 15.07 | 15.12 | 14.63 | 14.75 | 8407500 | AAL | NaN |
| **1** | 2013-02-11 | 14.89 | 15.01 | 14.26 | 14.46 | 8882000 | AAL | -1.966102 |
| **2** | 2013-02-12 | 14.45 | 14.51 | 14.10 | 14.27 | 8126000 | AAL | -1.313970 |
| **3** | 2013-02-13 | 14.30 | 14.94 | 14.25 | 14.66 | 10259500 | AAL | 2.733006 |

In [36]:
```
apple.set_index('date', inplace=True)
```

In [37]:
```
apple.head(4)
```

Out[37]:

| | open | high | low | close | volume | Name | daily retuen(in%) |
|---|---|---|---|---|---|---|---|
| **date** | | | | | | | |
| **2013-02-08** | 15.07 | 15.12 | 14.63 | 14.75 | 8407500 | AAL | NaN |
| **2013-02-11** | 14.89 | 15.01 | 14.26 | 14.46 | 8882000 | AAL | -1.966102 |
| **2013-02-12** | 14.45 | 14.51 | 14.10 | 14.27 | 8126000 | AAL | -1.313970 |
| **2013-02-13** | 14.30 | 14.94 | 14.25 | 14.66 | 10259500 | AAL | 2.733006 |

In [ ]:

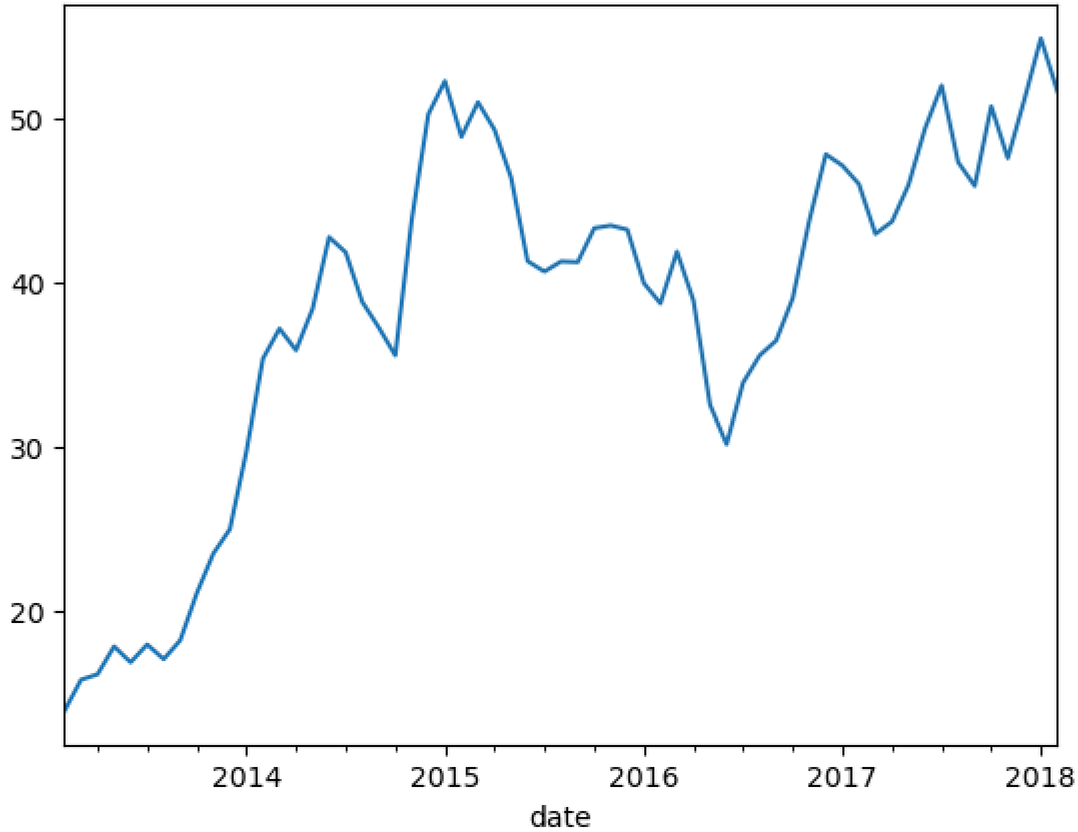
In [38]:
```
apple['close'].resample('M').mean()
```

Out[38]:
```
date
2013-02-28    13.877143
2013-03-31    15.776500
2013-04-30    16.108636
2013-05-31    17.810909
2013-06-30    16.839000
                ...
2017-10-31    50.756364
2017-11-30    47.587143
2017-12-31    51.150500
2018-01-31    54.902857
2018-02-28    51.664000
Freq: M, Name: close, Length: 61, dtype: float64
```

In [ ]:

In [39]:
```
apple['close'].resample('M').mean().plot()
```

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




In [ ]:

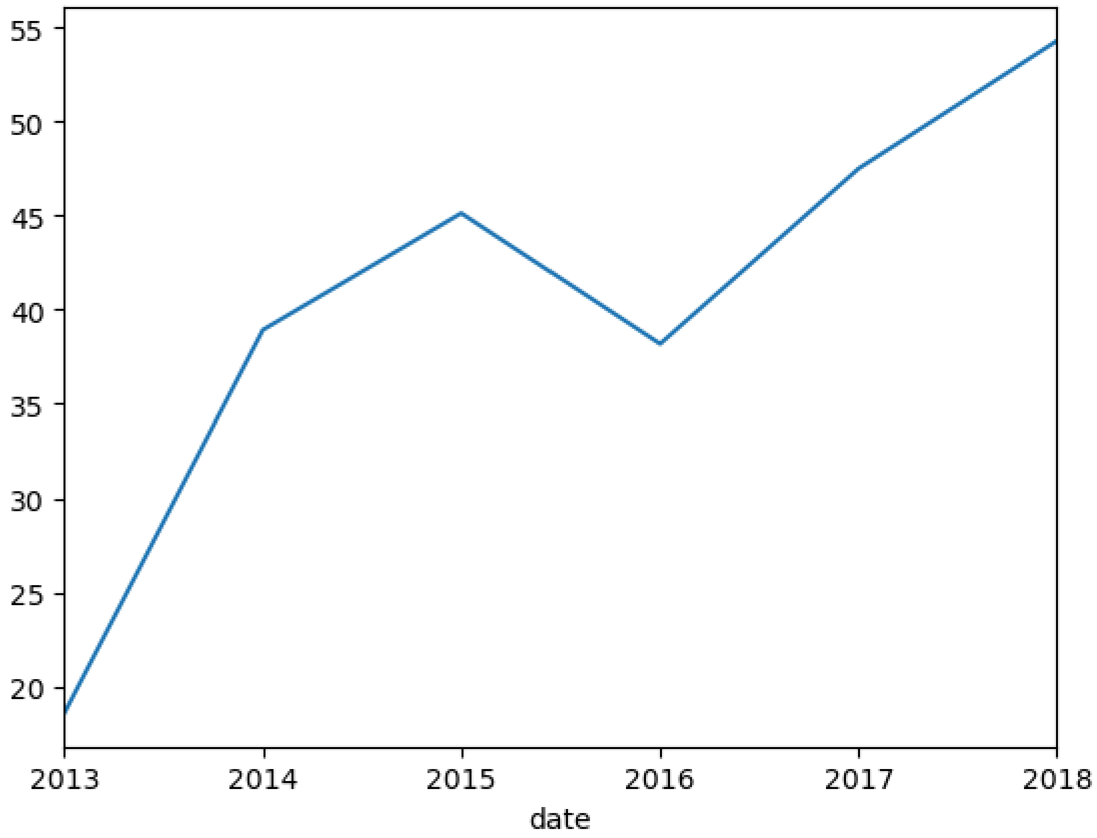
In [40]:
```
apple['close'].resample('Y').mean()
```

Out[40]:
```
date
2013-12-31    18.601549
2014-12-31    38.923550
2015-12-31    45.121607
2016-12-31    38.183829
2017-12-31    47.490717
2018-12-31    54.280000
Freq: A-DEC, Name: close, dtype: float64
```

In [ ]:

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

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




In [ ]:

In [ ]:

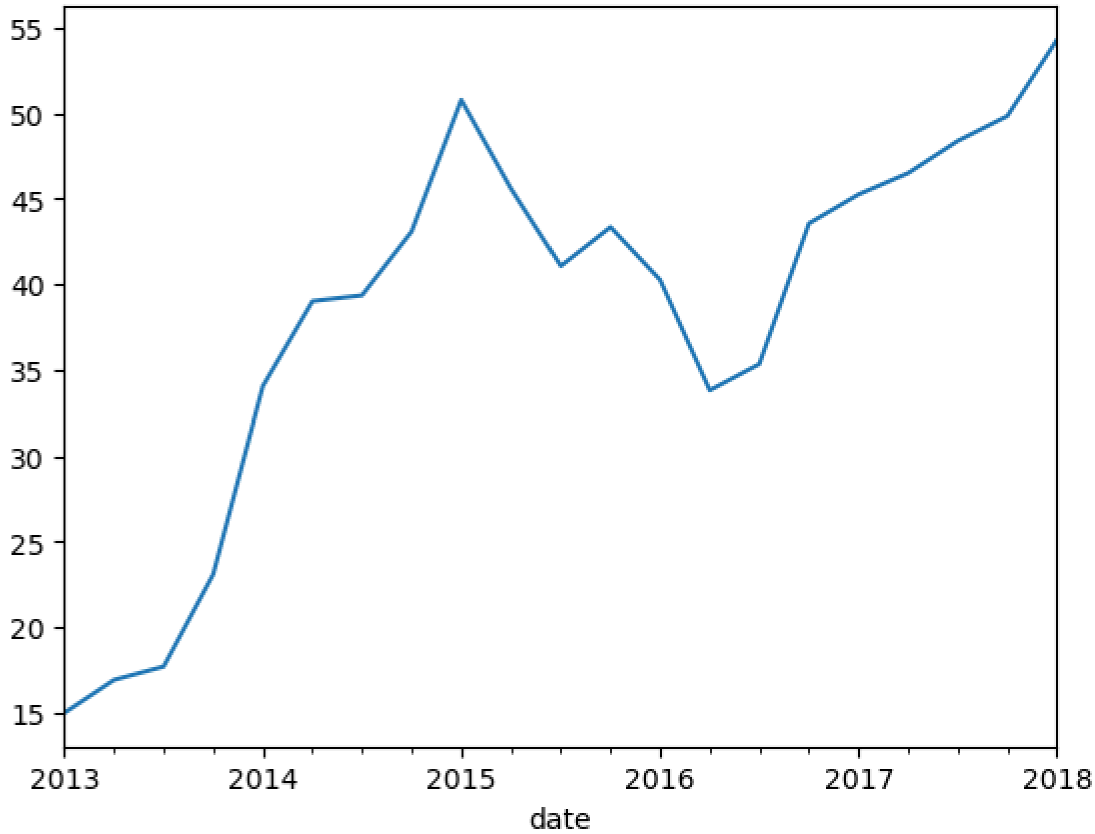
In [42]:
```
apple['close'].resample('Q').mean()
```

Out[42]:
```
date
2013-03-31    14.994412
2013-06-30    16.922031
2013-09-30    17.700625
2013-12-31    23.098281
2014-03-31    34.039343
2014-06-30    39.005935
2014-09-30    39.344778
2014-12-31    43.076484
2015-03-31    50.769672
2015-06-30    45.595397
2015-09-30    41.059453
2015-12-31    43.334063
2016-03-31    40.246230
2016-06-30    33.796016
2016-09-30    35.325781
2016-12-31    43.547778
2017-03-31    45.243226
2017-06-30    46.488730
2017-09-30    48.370159
2017-12-31    49.825079
2018-03-31    54.280000
Freq: Q-DEC, Name: close, dtype: float64
```

In [ ]:

In [43]:
```
apple['close'].resample('Q').mean().plot()
```

Out[43]: <AxesSubplot:xlabel='date'>




In [ ]:

In [ ]:

# Multi-Variate Analysis to understand co-relation

# Checking if the closing prices of these tech companies (Amazon, Apple, Microsoft, Google) are correlated or not

In [44]:
```
company_list
```

Out[44]:
```
['C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\AAPL_data.csv',
 'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\AMZN_data.csv',
 'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\GOOG_data.csv',
 'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks
_5yr\\\\MSFT_data.csv']
```

In [45]:
```
company_list[0]
```

Out[45]:
```
'C:\\\\Users\\\\lenovo\\\\Downloads\\\\S&P_resources\\\\individual_stocks_5yr\\\\AAPL_data.csv'
```

In [46]:
```
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 [47]:
```
closing_price = pd.DataFrame()
```

In [48]:
```
closing_price['apple_close'] = app['close']
closing_price['amzn_close'] = amzn['close']
closing_price['google_close'] = google['close']
closing_price['msft_close'] = msft['close']
```

In [49]:
```
closing_price
```

Out[49]:

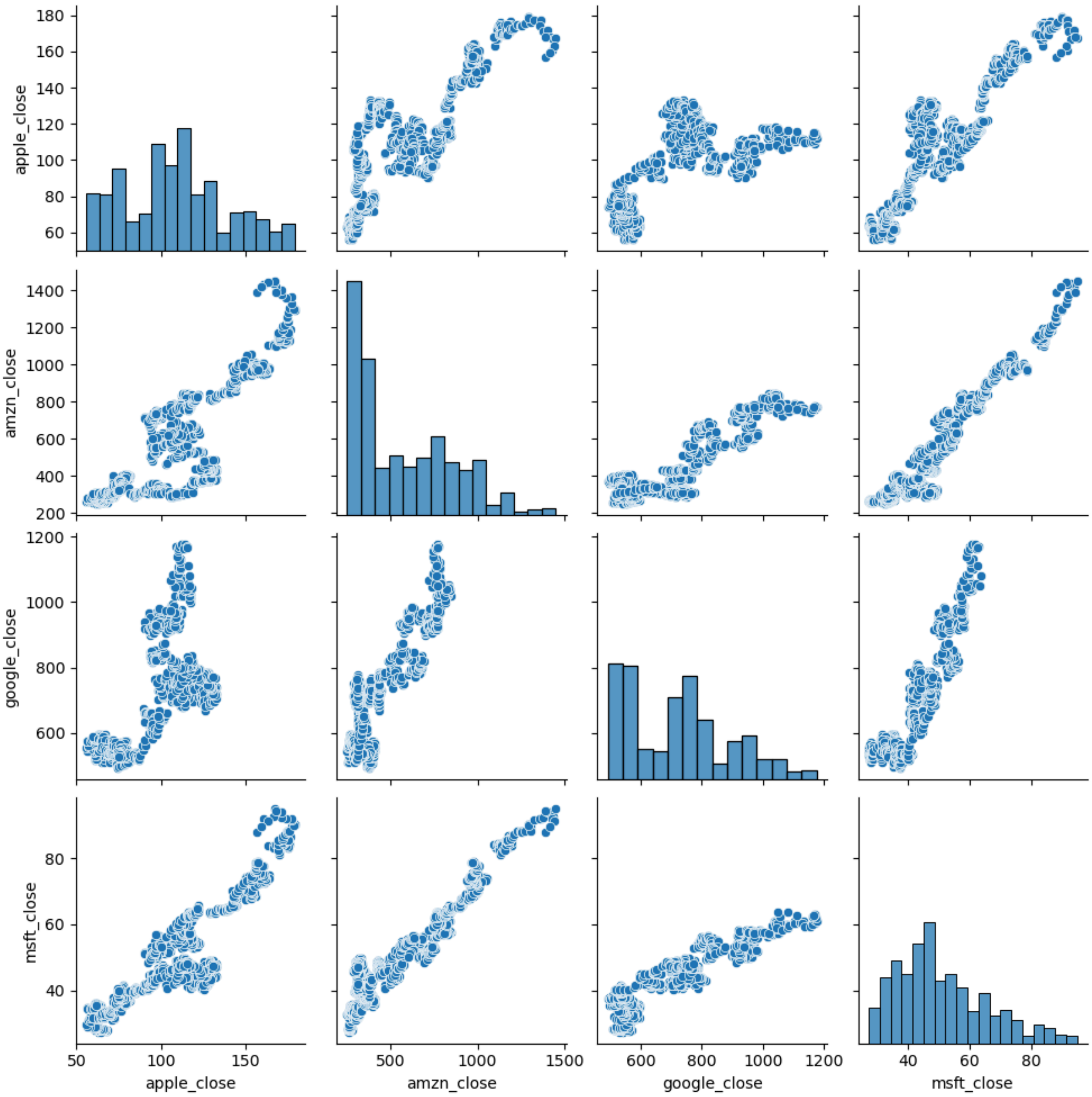
| | apple_close | amzn_close | google_close | msft_close |
|---|---|---|---|---|
| **0** | 67.8542 | 261.95 | 558.46 | 27.55 |
| **1** | 68.5614 | 257.21 | 559.99 | 27.86 |
| **2** | 66.8428 | 258.70 | 556.97 | 27.88 |
| **3** | 66.7156 | 269.47 | 567.16 | 28.03 |
| **4** | 66.6556 | 269.24 | 567.00 | 28.04 |
| **...** | ... | ... | ... | ... |
| **1254** | 167.7800 | 1390.00 | NaN | 94.26 |
| **1255** | 160.5000 | 1429.95 | NaN | 91.78 |
| **1256** | 156.4900 | 1390.00 | NaN | 88.00 |
| **1257** | 163.0300 | 1442.84 | NaN | 91.33 |
| **1258** | 159.5400 | 1416.78 | NaN | 89.61 |

1259 rows × 4 columns

In [ ]:

In [50]:
```
sns.pairplot(closing_price)
```

Out[50]: <seaborn.axisgrid.PairGrid at 0x1d6f76aedc0>




In [ ]:

In [51]:
```
closing_price.corr()
```

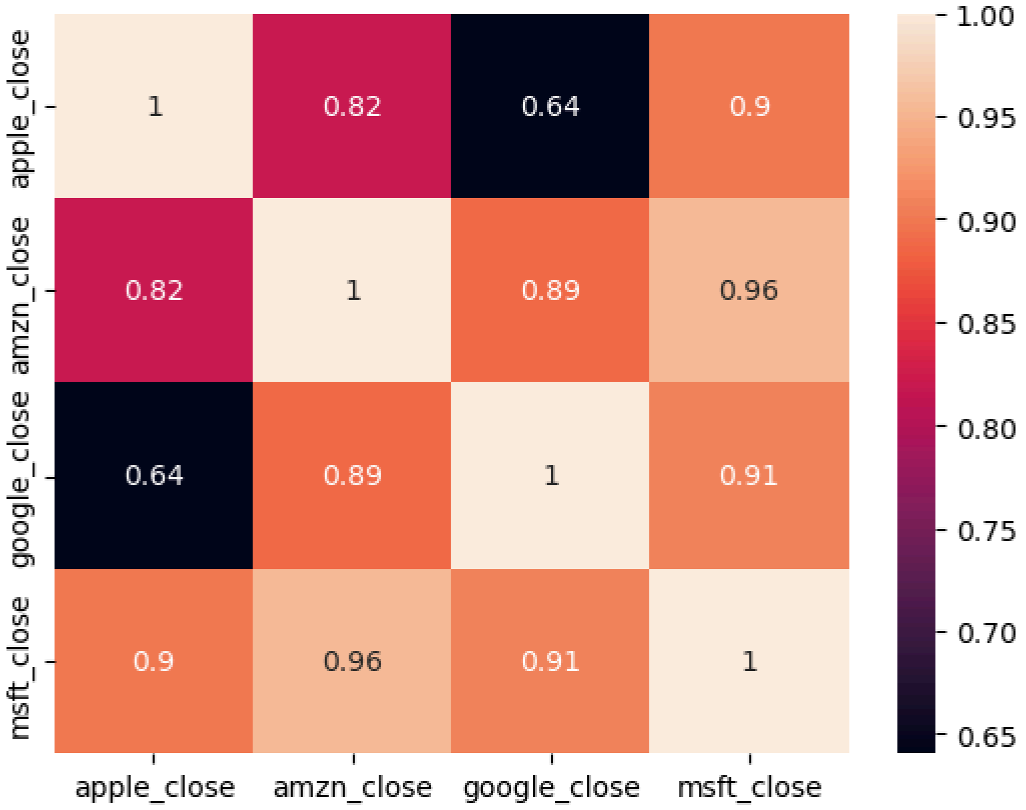
Out[51]:

| | apple_close | amzn_close | google_close | msft_close |
|---|---|---|---|---|
| **apple_close** | 1.000000 | 0.819078 | 0.640522 | 0.899689 |
| **amzn_close** | 0.819078 | 1.000000 | 0.888456 | 0.955977 |
| **google_close** | 0.640522 | 0.888456 | 1.000000 | 0.907011 |
| **msft_close** | 0.899689 | 0.955977 | 0.907011 | 1.000000 |

In [ ]:

In [52]:
```
sns.heatmap(closing_price.corr(), annot=True)
```

Out[52]: <AxesSubplot:>




In [ ]:

# Analysis whether daily changes in closing price of stocks or daily returns in stock are co-related or not

In [ ]:

In [ ]:

In [53]:
```
closing_price
```

Out[53]:

| | apple_close | amzn_close | google_close | msft_close |
|---|---|---|---|---|
| **0** | 67.8542 | 261.95 | 558.46 | 27.55 |
| **1** | 68.5614 | 257.21 | 559.99 | 27.86 |
| **2** | 66.8428 | 258.70 | 556.97 | 27.88 |
| **3** | 66.7156 | 269.47 | 567.16 | 28.03 |
| **4** | 66.6556 | 269.24 | 567.00 | 28.04 |
| **...** | ... | ... | ... | ... |
| **1254** | 167.7800 | 1390.00 | NaN | 94.26 |
| **1255** | 160.5000 | 1429.95 | NaN | 91.78 |
| **1256** | 156.4900 | 1390.00 | NaN | 88.00 |
| **1257** | 163.0300 | 1442.84 | NaN | 91.33 |
| **1258** | 159.5400 | 1416.78 | NaN | 89.61 |

1259 rows × 4 columns

In [ ]:

In [54]:
```
closing_price['apple_close']
```

Out[54]:
```
0        67.8542
1        68.5614
2        66.8428
3        66.7156
4        66.6556
          ...
1254    167.7800
1255    160.5000
1256    156.4900
1257    163.0300
1258    159.5400
Name: apple_close, Length: 1259, dtype: float64
```

In [55]:
```
closing_price['apple_close'].shift(1)
```

Out[55]:
```
0            NaN
1        67.8542
2        68.5614
3        66.8428
4        66.7156
          ...
1254    167.4300
1255    167.7800
1256    160.5000
1257    156.4900
1258    163.0300
Name: apple_close, Length: 1259, dtype: float64
```

In [ ]:

In [56]:
```
(closing_price['apple_close'] - closing_price['apple_close'].shift(1))/clos
```

Out[56]:
```
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 [ ]:

In [57]:
```
for col in closing_price.columns:
    closing_price[col + '_pct_change'] = (closing_price[col] - closing_price
```

In [58]:
```
closing_price
```

Out[58]:

| | apple_close | amzn_close | google_close | msft_close | apple_close_pct_change | amzn_clos |
|---|---|---|---|---|---|---|
| **0** | 67.8542 | 261.95 | 558.46 | 27.55 | NaN | |
| **1** | 68.5614 | 257.21 | 559.99 | 27.86 | 1.042235 | |
| **2** | 66.8428 | 258.70 | 556.97 | 27.88 | -2.506658 | |
| **3** | 66.7156 | 269.47 | 567.16 | 28.03 | -0.190297 | |
| **4** | 66.6556 | 269.24 | 567.00 | 28.04 | -0.089934 | |
| **...** | ... | ... | ... | ... | ... | |
| **1254** | 167.7800 | 1390.00 | NaN | 94.26 | 0.209043 | |
| **1255** | 160.5000 | 1429.95 | NaN | 91.78 | -4.339015 | |
| **1256** | 156.4900 | 1390.00 | NaN | 88.00 | -2.498442 | |
| **1257** | 163.0300 | 1442.84 | NaN | 91.33 | 4.179181 | |
| **1258** | 159.5400 | 1416.78 | NaN | 89.61 | -2.140710 | |

1259 rows × 8 columns

In [59]:
```
closing_price.columns
```

Out[59]:
```
Index(['apple_close', 'amzn_close', 'google_close', 'msft_close',
       'apple_close_pct_change', 'amzn_close_pct_change',
       'google_close_pct_change', 'msft_close_pct_change'],
      dtype='object')
```

In [60]:
```
closing_price[['apple_close_pct_change', 'amzn_close_pct_change',
        'google_close_pct_change', 'msft_close_pct_change']]
```

Out[60]:

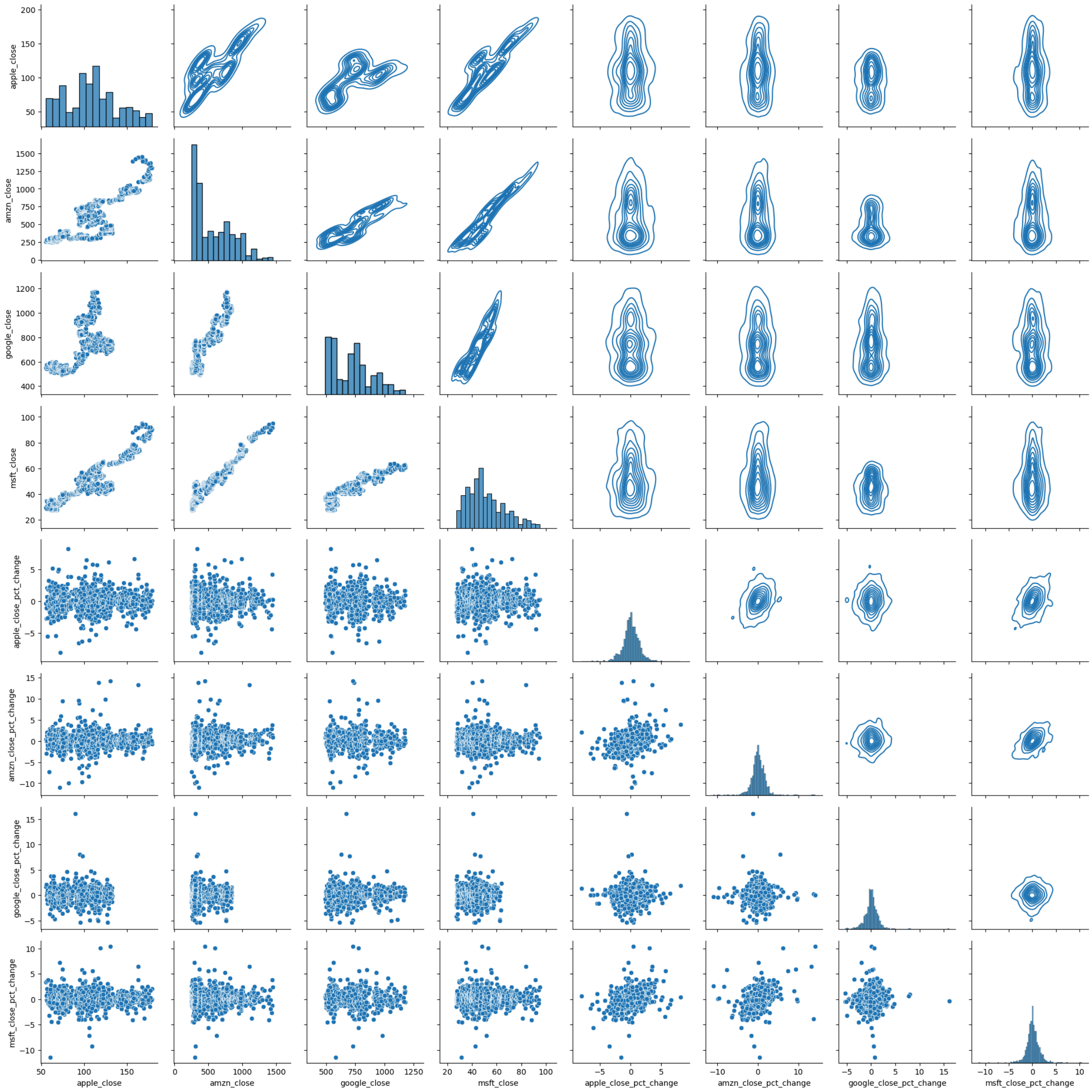
| | apple_close_pct_change | amzn_close_pct_change | google_close_pct_change | msft_close_ |
|---|---|---|---|---|
| 0 | NaN | NaN | NaN | |
| 1 | 1.042235 | -1.809506 | 0.273968 | |
| 2 | -2.506658 | 0.579293 | -0.539295 | |
| 3 | -0.190297 | 4.163123 | 1.829542 | |
| 4 | -0.089934 | -0.085353 | -0.028211 | |
| ... | ... | ... | ... | |
| 1254 | 0.209043 | -4.196734 | NaN | |
| 1255 | -4.339015 | 2.874101 | NaN | |
| 1256 | -2.498442 | -2.793804 | NaN | |
| 1257 | 4.179181 | 3.801439 | NaN | |
| 1258 | -2.140710 | -1.806160 | NaN | |

1259 rows × 4 columns

In [ ]:

In [61]:
```
g = sns.PairGrid(data = closing_price)
g.map_diag(sns.histplot)
g.map_lower(sns.scatterplot)
g.map_upper(sns.kdeplot)
```

Out[61]: <seaborn.axisgrid.PairGrid at 0x1d6f766fc70>




In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: