Module 2

- By Suprateek Halsana

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
sns.set(style='darkgrid')
import plotly.express as px
```

Problem [2.1] : Load Data -> Convert Date column to the index -> Plot Closing Price v/s Days

In [2]:
```
data=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\week2.csv")
data
```

Out[2]:

	Symbol	Series	Date	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliveral
0	LALPATHLAB	EQ	2017-05-15	891.15	895.0	914.15	881.00	912.00	900.60	889.35	257655	2.291453e+08	15451	1747
1	LALPATHLAB	EQ	2017-05-16	900.60	910.0	925.00	895.05	909.40	910.95	914.57	104896	9.593430e+07	13001	758
2	LALPATHLAB	EQ	2017-05-17	910.95	913.0	925.00	909.00	912.05	911.70	917.19	72460	6.645960e+07	2802	538
3	LALPATHLAB	EQ	2017-05-18	911.70	908.0	919.35	903.05	906.00	909.75	914.12	37707	3.446889e+07	2731	248
4	LALPATHLAB	EQ	2017-05-19	909.75	917.0	917.00	905.80	910.00	910.25	910.61	81082	7.383375e+07	4430	699
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
489	LALPATHLAB	EQ	2019-05-07	1024.95	1015.7	1031.95	1006.90	1013.00	1013.10	1020.66	20113	2.052851e+07	3628	115
490	LALPATHLAB	EQ	2019-05-08	1013.10	1014.0	1019.85	1001.90	1002.00	1006.10	1010.90	18048	1.824473e+07	4175	112
491	LALPATHLAB	EQ	2019-05-09	1006.10	1014.0	1014.00	977.70	979.90	982.90	987.36	60539	5.977355e+07	8169	373
492	LALPATHLAB	EQ	2019-05-10	982.90	989.9	994.30	963.60	980.00	980.65	980.71	30568	2.997840e+07	4618	190
493	LALPATHLAB	EQ	2019-05-13	980.65	970.8	996.00	960.00	975.00	976.25	974.21	37187	3.622810e+07	6798	237

494 rows × 19 columns

In [3]:
```
# Setting the Date as Index
data.set_index('Date',inplace=True)
```

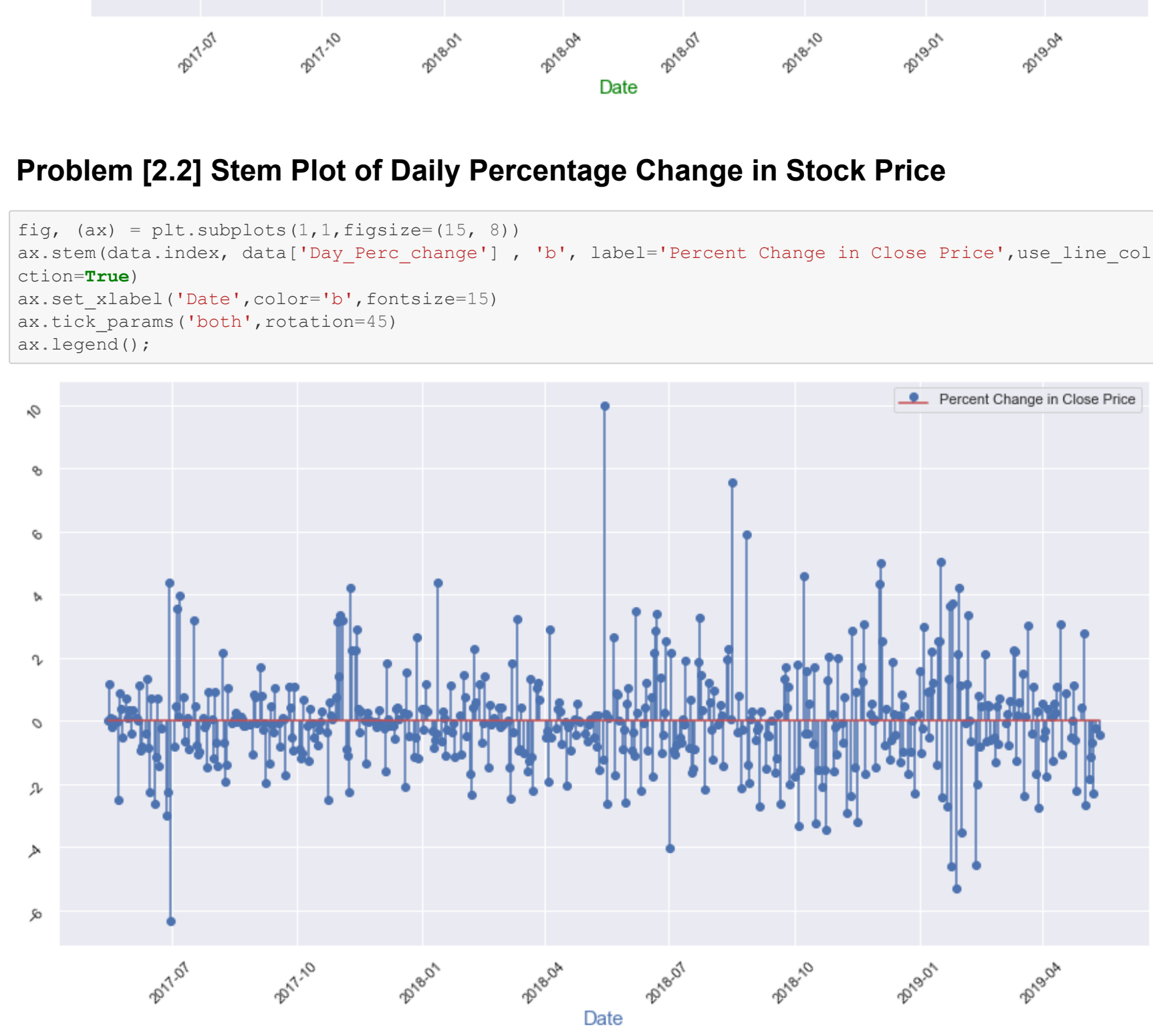
In [4]:
```
# Coverting Dates to Datetime64[ns]
data.index=data.index.astype('datetime64[ns]')
```

Close Price v/s Date

In [5]:
```
plt.figure(figsize=(15,8))
plt.plot(data.index,data['Close Price'],)
plt.xlabel('Date',fontsize=15,color='green')
plt.ylabel('close price',fontsize=15,color='green')
plt.title('Close Price v/s Date',fontsize=20,color= 'green')
plt.tick_params('both',rotation=45)
```

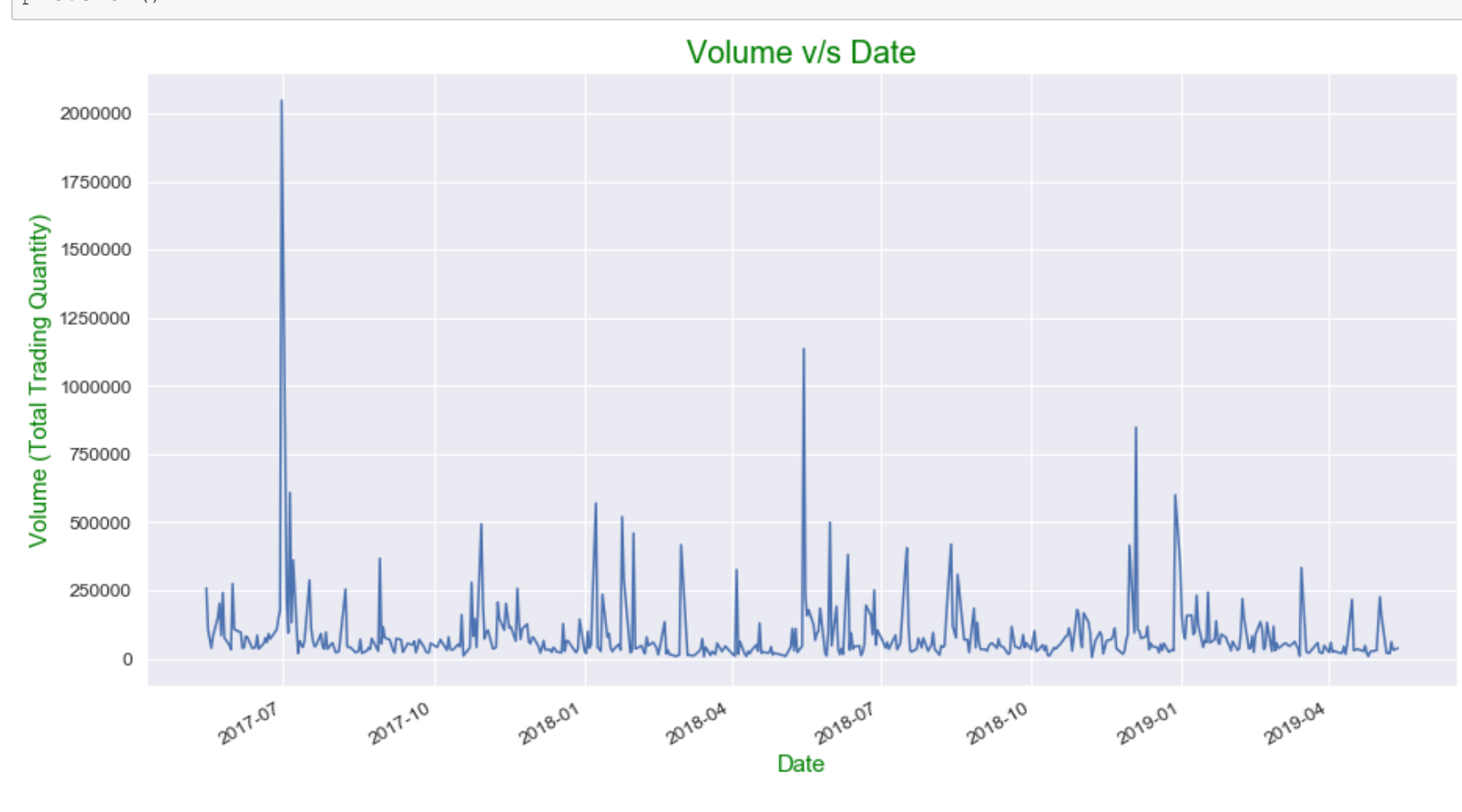
Problem [2.2] Stem Plot of Daily Percentage Change in Stock Price

In [6]:
```
fig, (ax) = plt.subplots(1,1,figsize=(15, 8))
ax.stem(data.index, data['Day_Perc_change'] , 'b', label='Percent Change in Close Price',use_line_collection=True)
ax.set_xlabel('Date',color='b',fontsize=15)
ax.tick_params('both',rotation=45)
ax.legend();
```



Problem [2.3] Daily Volume Plot -> Comparison With Perentage Stem Plot

In [7]:
```
data['Total Traded Quantity'].plot(figsize=(15,8),fontsize=12)
plt.title('Volume v/s Date',color='green',fontsize=20)
plt.xlabel('Date',color='green',fontsize=15)
plt.ylabel('Volume (Total Trading Quantity)',color='green',fontsize=15)
plt.show()
```



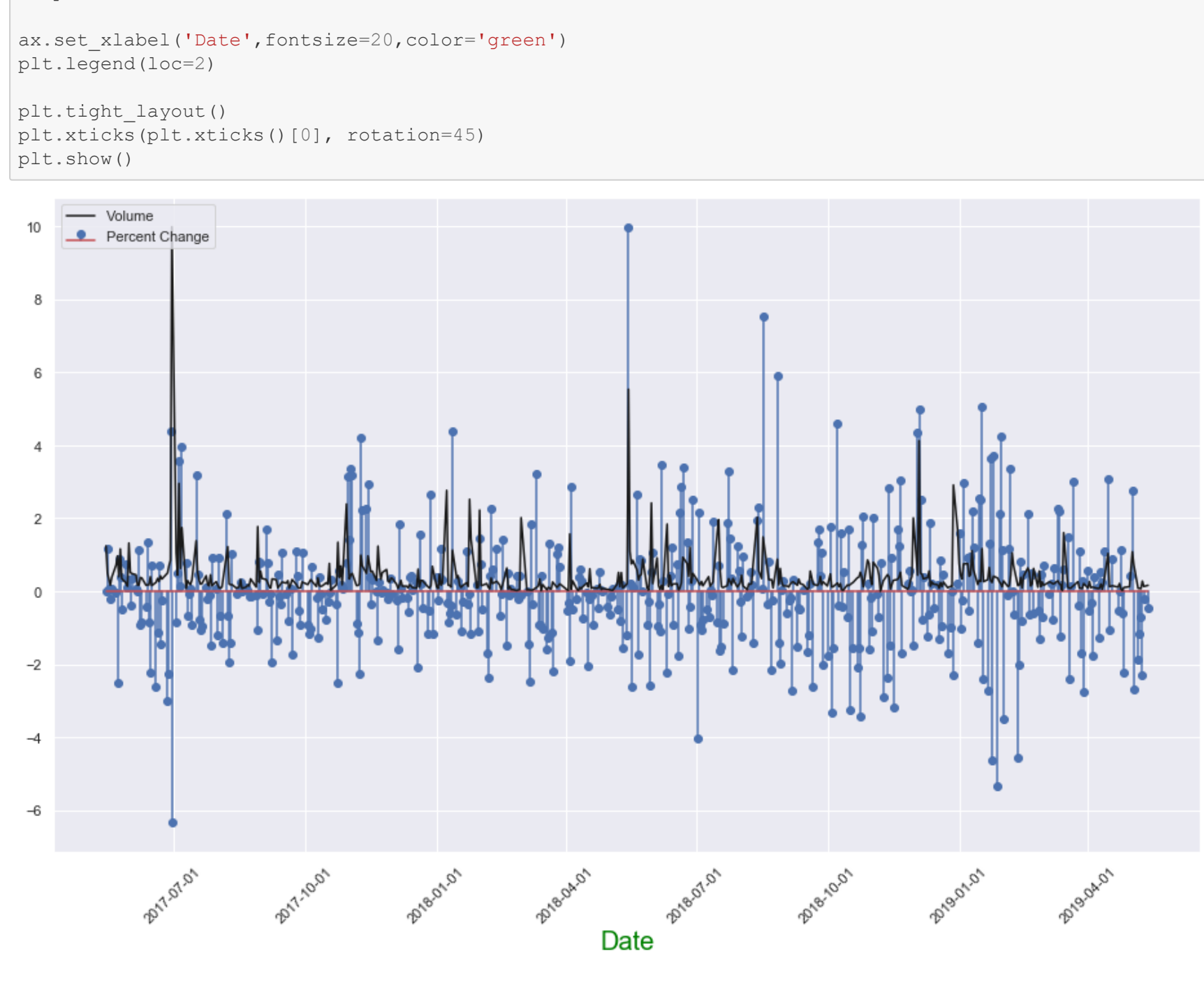
In [8]:
```
scaledvolume =  data['Total Traded Quantity'] - data["Total Traded Quantity"].min()
scaledvolume = scaledvolume/scaledvolume.max() * data.Day_Perc_change.max()

fig, ax = plt.subplots(figsize=(13, 8))

ax.stem(data.index, data.Day_Perc_change , 'b', markerfmt='bo', label='Percent Change')
ax.plot(data.index, scaledvolume, 'k', label='Volume')

ax.set_xlabel('Date',fontsize=20,color='green')
plt.legend(loc=2)

plt.tight_layout()
plt.xticks(plt.xticks()[0], rotation=45)
plt.show()
```



Comparative Study on Trading Volume and the Day_Perc_change

Based on the Above Stem plot of Daily Percentage change and plot of the Trading Volume , we can conclude that the Daily Percent proportionally changes with the change in the Trading Volume and hence, on slight increase of Trading Volume we can see increase in Daily Percentage Change at '2018-04 - 2018-07' and also an equivalent drop could be seen at certain periods at the drop of Daily Volume .

Problem [2.4] : Group By Trend -> Pie Chart of 'Trend' -> Bar Plots

In [9]:
```
# Reset the index
data=data.reset_index()
```

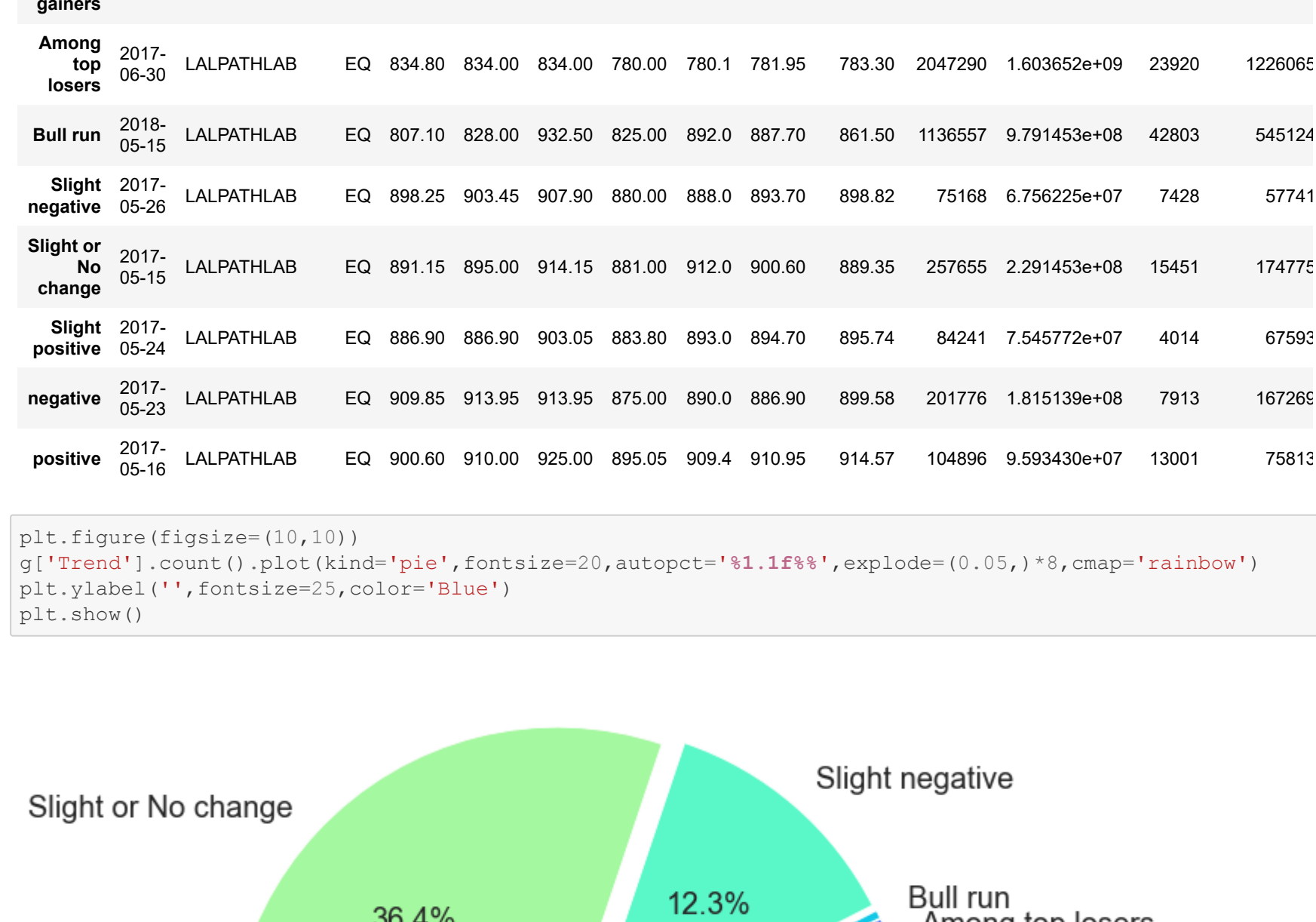
In [10]:
```
# Group by the Trend
g=data.groupby(['Trend'])
```

In [11]:
```
g.first()
```

Out[11]:

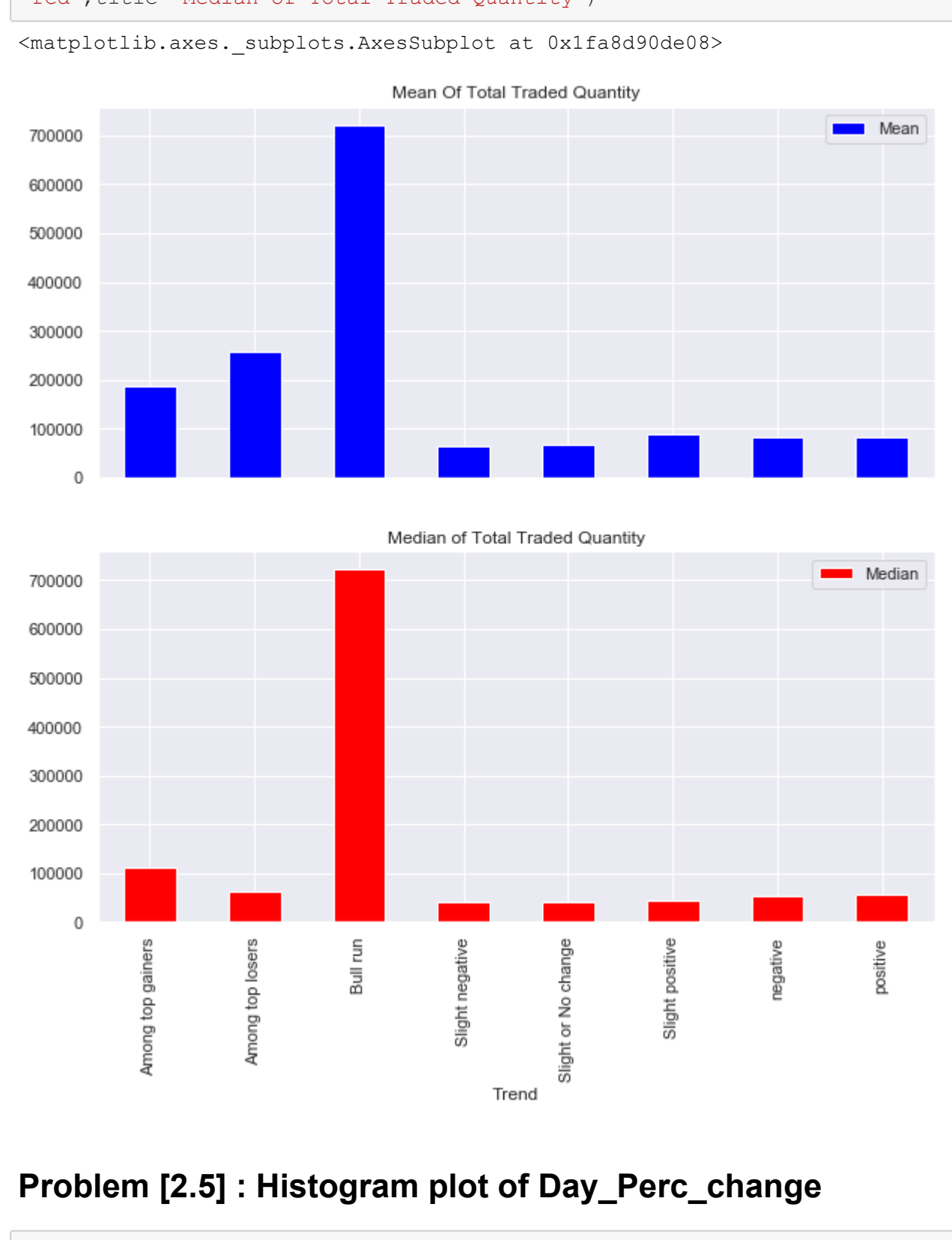
Trend	Date	Symbol	Series	Prev Close	Open Price	High Price	Low Price	Last Price	Close Price	Average Price	Total Traded Quantity	Turnover	No. of Trades	Deliverable Qty
Among top gainers	2017-06-29	LALPATHLAB	EQ	799.85	790.00	847.75	783.80	833.4	834.80	809.38	177028	1.432833e+08	16859	79503
Among top losers	2017-06-30	LALPATHLAB	EQ	834.80	834.00	834.00	780.00	780.1	781.95	783.30	2047290	1.603652e+09	23920	1226065
Bull run	2018-05-15	LALPATHLAB	EQ	807.10	828.00	932.50	825.00	892.0	887.70	861.50	1136557	9.791453e+08	42803	545124
Slight negative	2017-05-26	LALPATHLAB	EQ	898.25	903.45	907.90	880.00	888.0	893.70	898.82	75168	6.756225e+07	7428	57741
Slight or No change	2017-05-15	LALPATHLAB	EQ	891.15	895.00	914.15	881.00	912.0	900.60	889.35	257655	2.291453e+08	15451	174775
Slight positive	2017-05-24	LALPATHLAB	EQ	886.90	886.90	903.05	883.80	893.0	894.70	895.74	84241	7.545772e+07	4014	67593
negative	2017-05-23	LALPATHLAB	EQ	909.85	913.95	913.95	875.00	890.0	886.90	899.58	201776	1.815139e+08	7913	167269
positive	2017-05-16	LALPATHLAB	EQ	900.60	910.00	925.00	895.05	909.4	910.95	914.57	104896	9.593430e+07	13001	75813

In [12]:
```
plt.figure(figsize=(10,10))
g['Trend'].count().plot(kind='pie',fontsize=20,autopct='%1.1f%%',explode=(0.05,)*8,cmap='rainbow')
plt.ylabel('',fontsize=25,color='Blue')
plt.show()
```



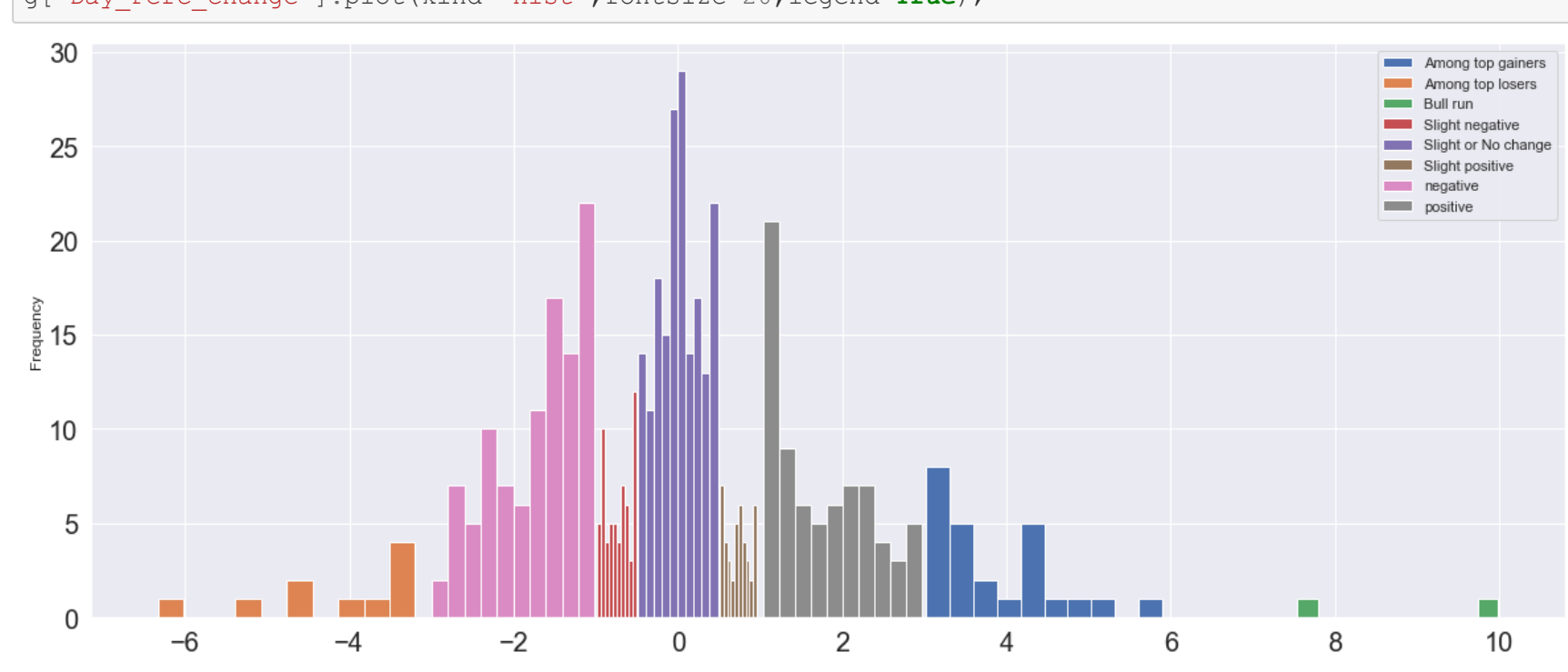
In [13]:
```
fig,(ax1,ax2)=plt.subplots(2,1,figsize=(10,10))
g['Total Traded Quantity'].mean().plot(ax=ax1,kind='bar',label='Mean',legend=True,color='blue',title='Mean Of Total Traded Quantity')
g['Total Traded Quantity'].median().plot(ax=ax2,kind='bar',label='Median',legend=True,sharex=ax1,color='red',title='Median of Total Traded Quantity')
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa8d90de08>



Problem [2.5] : Histogram plot of Day_Perc_change

In [14]:
```
plt.figure(figsize=(20,8))
g['Day_Perc_change'].plot(kind='hist',fontsize=20,legend=True);
```



Problem [2.6] : Correlation

Five Stocks of my Choice

In [15]:
```
ashoka=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\prerequisites\Small_Cap\Small_Cap\ASHOKA.csv")

bajaj=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\prerequisites\Small_Cap\Small_Cap\BAJAJELEC.csv")

centuryply=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\prerequisites\Small_Cap\Small_Cap\CENTURYPLY.csv")

fortis=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\prerequisites\Small_Cap\Small_Cap\FORTIS.csv")

pvr=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\prerequisites\Small_Cap\Small_Cap\PVR.csv")
```

In [16]:
```
# Only Working on Stocks of Series that are 'EQ'
tech_list=[ashoka,bajaj,centuryply,fortis,pvr]
for i in tech_list:
    i=i[i['Series']=='EQ']
```

Data Frame Containing only Close Price of Five Stocks

In [17]:
```
dclose=pd.DataFrame()
for i in tech_list:
    dclose[i['Symbol'][0]]=i['Close Price']
```

In [18]:
```
dclose.isnull().sum()
```

Out[18]:
```
ASHOKA        0
BAJAJELEC     0
CENTURYPLY    0
FORTIS        0
PVR           0
dtype: int64
```

Data Frame Containing only Perc_change of Five Stocks

In [19]:
```
dperc=pd.DataFrame()
for i in tech_list:
    dperc[i['Symbol'][0]]=i['Close Price'].pct_change()
    dperc[i['Symbol'][0]][0]=0
```

In [20]:
```
dperc.isnull().sum()
```

Out[20]:
```
ASHOKA        0
BAJAJELEC     0
CENTURYPLY    0
FORTIS        0
PVR           0
dtype: int64
```

In [21]:
```
dperc
```

Out[21]:

	ASHOKA	BAJAJELEC	CENTURYPLY	FORTIS	PVR
0	0.000000	0.000000	0.000000	0.000000	0.000000
1	-0.005323	0.017148	-0.002063	0.012911	0.009213
2	0.009772	0.008213	-0.004697	-0.009620	-0.005653
3	-0.033871	-0.045019	-0.016991	-0.043953	-0.009965
4	-0.016218	0.006286	-0.024198	0.034036	-0.000990
...	...	...	...	...	...
489	0.017665	-0.000179	-0.034330	-0.010163	-0.003208
490	-0.022439	-0.041775	-0.011339	-0.008434	-0.012318
491	-0.001299	-0.004865	-0.004960	-0.011834	-0.017752
492	0.018647	0.017016	0.000000	-0.004491	0.005691
493	-0.002554	0.001202	0.014642	-0.008271	-0.024482

494 rows × 5 columns

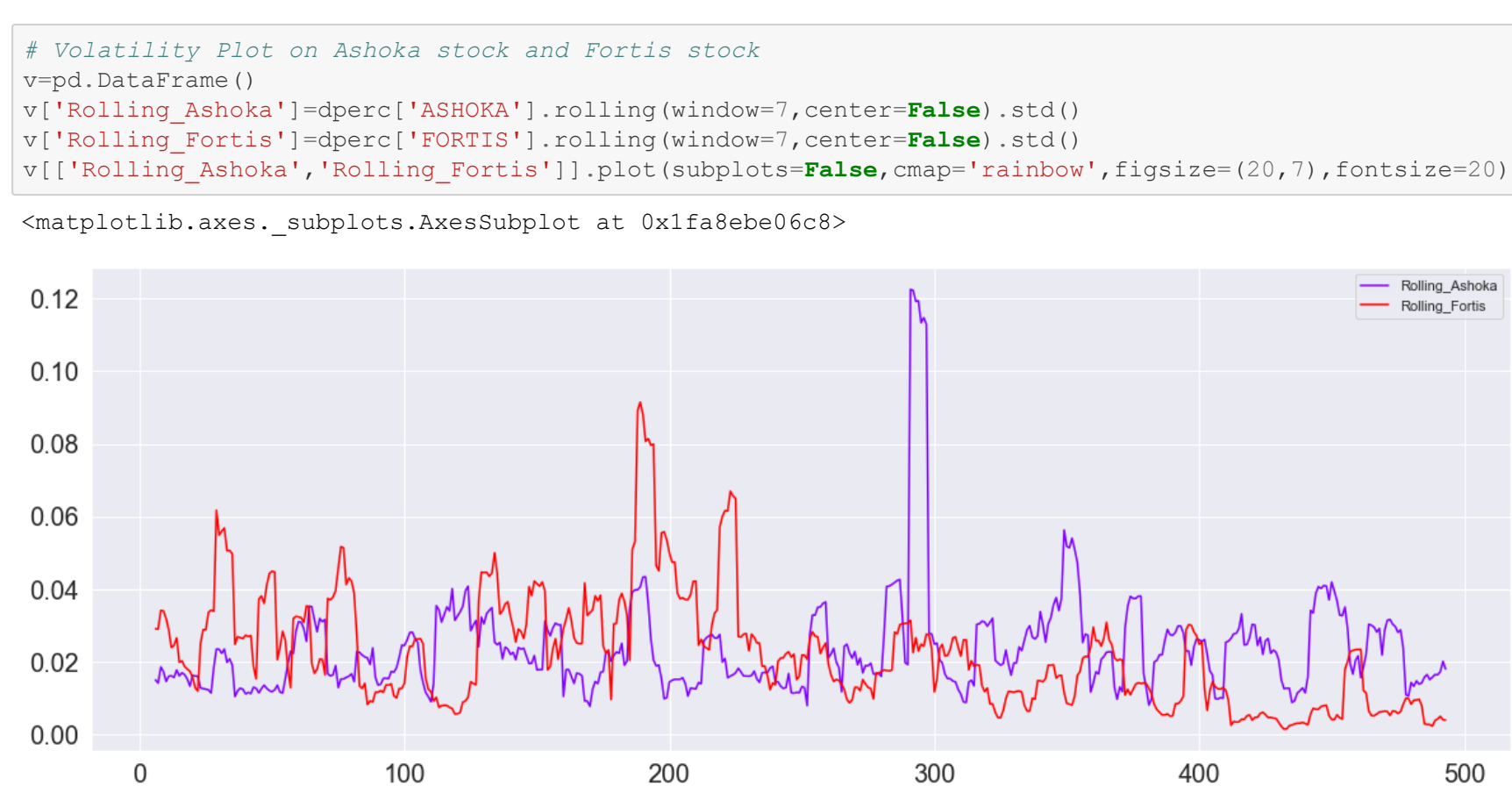
In [22]:
```
sns.pairplot(dperc)
```

Out[22]: <seaborn.axisgrid.PairGrid at 0x1fa8dbab888>

Problem [2.7] : Volatility

In [23]:
```
# Volatility Plot on Ashoka stock and Fortis stock
v=pd.DataFrame()
v['Rolling_Ashoka']=dperc['ASHOKA'].rolling(window=7,center=False).std()
v['Rolling_Fortis']=dperc['FORTIS'].rolling(window=7,center=False).std()
v[['Rolling_Ashoka','Rolling_Fortis']].plot(subplots=False,cmap='rainbow',figsize=(20,7),fontsize=20)
```

Out[23]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa8ebe06c8>



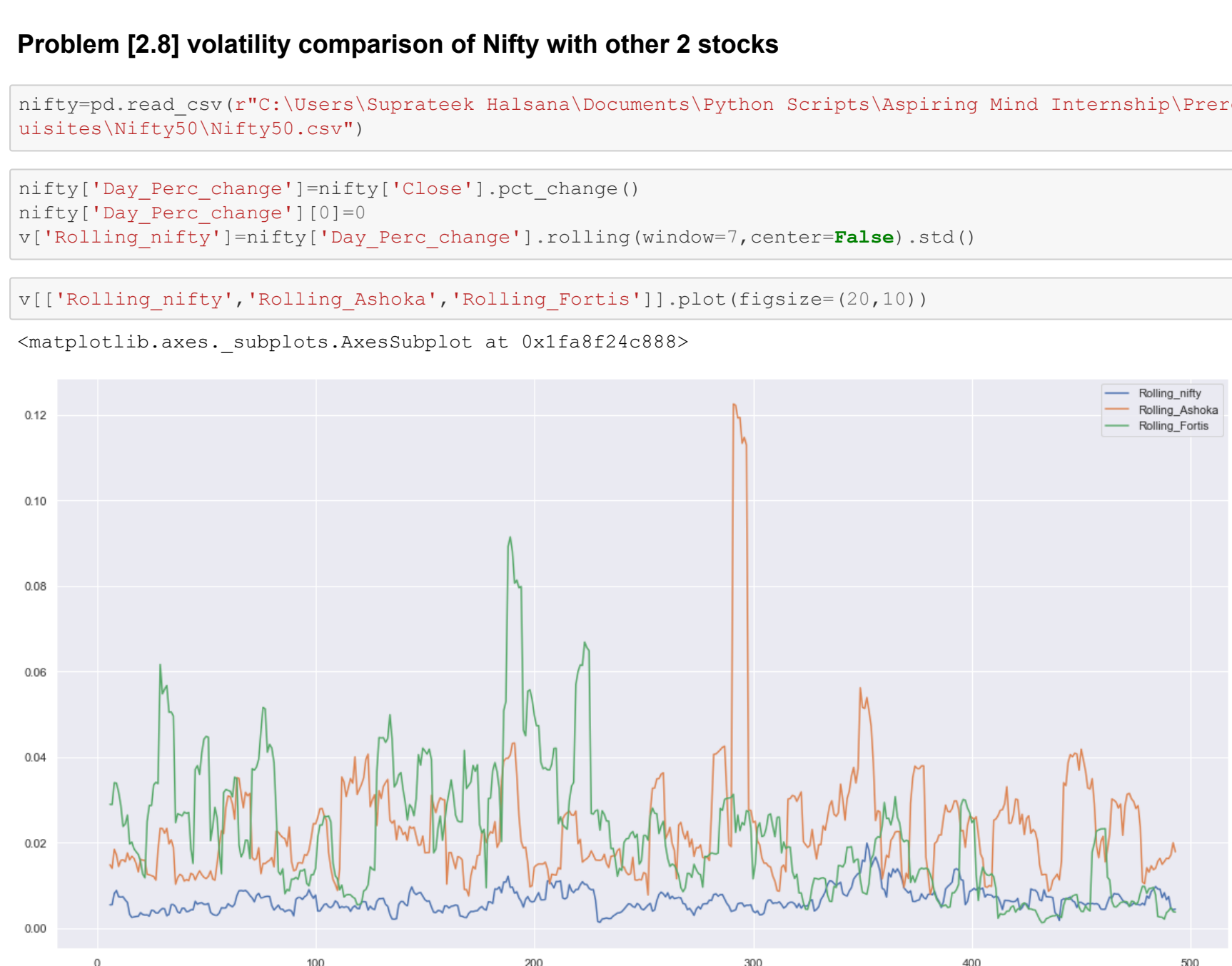
Problem [2.8] volatility comparison of Nifty with other 2 stocks

In [24]:
```
nifty=pd.read_csv(r"C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\Prerequisites\Nifty50\Nifty50.csv")
```

In [25]:
```
nifty['Day_Perc_change']=nifty['Close'].pct_change()
nifty['Day_Perc_change'][0]=0
v['Rolling_nifty']=nifty['Day_Perc_change'].rolling(window=7,center=False).std()
```

In [26]:
```
v[['Rolling_nifty','Rolling_Ashoka','Rolling_Fortis']].plot(figsize=(20,10))
```

Out[26]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa8f24c888>



In [27]:
```
#dperc[['R_Avg_ASHOKA','R_Avg_BAJAJELEC','R_Avg_CENTURYPLY']].plot(subplots=False,cmap='rainbow',figsize=(20,10))
```

Problem [2.9] Trade Calls - Using Simple Moving Average

In [28]:
```
# On LALPATHLAB STOCKS

d=data.copy()
d=d.set_index('Date')
```

In [29]:
```
d['R_Avg_21']=d['Average Price'].rolling(window=21).mean()
d['R_Avg_34']=d['Average Price'].rolling(window=34).mean()
d[['R_Avg_21','R_Avg_34','Average Price']].plot(figsize=(20,8))
```

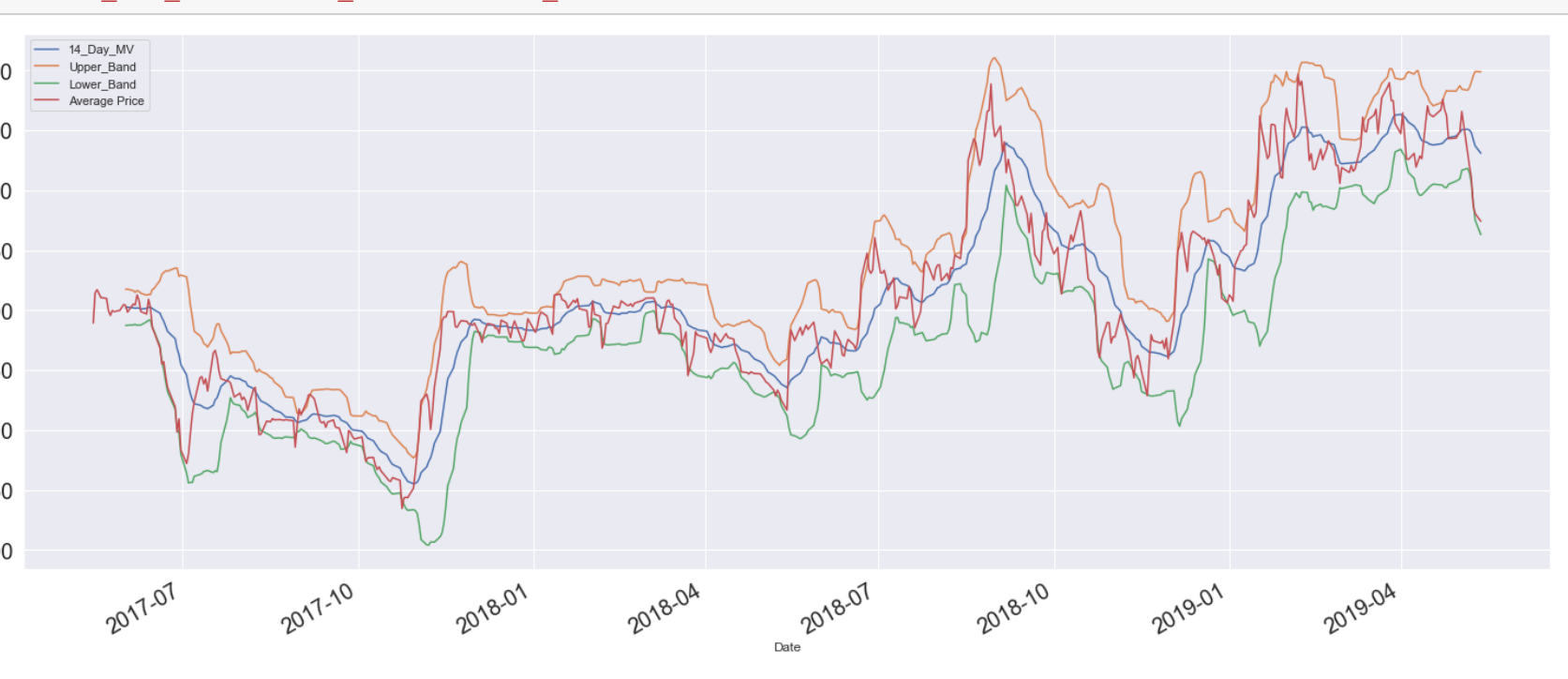
Out[29]: <matplotlib.axes._subplots.AxesSubplot at 0x1fa8efabe08>

Problem [2.10] Bollinger Bands

In [30]:

```
data['14_Day_MV']=data['Close Price'].rolling(window=14).mean()
data['14_Day_STD']=data['Close Price'].rolling(window=14).std()
data['Upper_Band']=data['14_Day_MV']+data['14_Day_STD']*2
data['Lower_Band']=data['14_Day_MV']-data['14_Day_STD']*2

data=data.set_index('Date')
data[['14_Day_MV','Upper_Band','Lower_Band','Average Price']].plot(fontsize=20,figsize=(25,10));
```



In [31]:

```
data=data.reset_index()
```

In [32]:

```
data.to_csv(r'C:\Users\Suprateek Halsana\Documents\Python Scripts\Aspiring Mind Internship\week3.csv',index=False)
```