FE 511 – Introduction to Bloomberg & Thomson Reuters
Final Project

Yashkumar Baldevbhai Chauhan

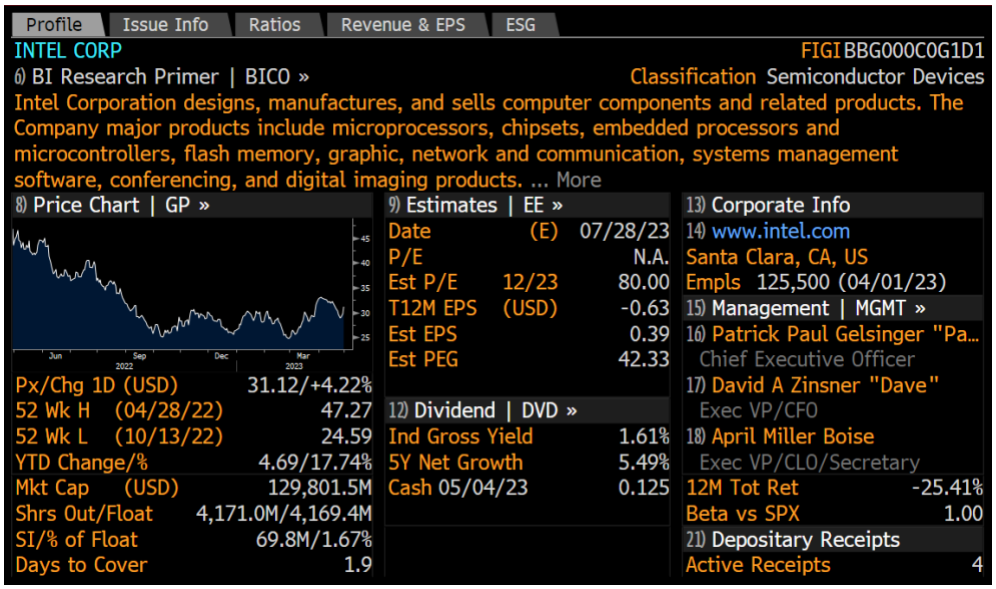
Stock Analysis & Time Series Analysis of Intel

Intel Company Description & Background:

Intel Corporation is a global technology firm based in the United States that develops and manufactures computer processors, other semiconductor chips, and related technologies. The headquarters of the corporation are in California.

Intel is well-known for its microprocessors, which power a wide range of personal computers, servers, and mobile devices. Its processors are utilized in a wide range of applications, from laptops and desktop computers to data centers and cloud computing systems.

Intel manufactures memory and storage solutions, networking and communications equipment, and programmable solutions such as field-programmable gate arrays (FPGAs) and structured ASICs in addition to microprocessors. Intel is also interested in a variety of other technologies, including autonomous driving, artificial intelligence (AI), and the Internet of Things (IoT). The firm is dedicated to improving technology and pushing innovation in all facets of modern life.



Intel Corp's Basic Info (Use of DES)

With total revenues of $77.9 billion in 2021, Intel Corporation is one of the world's leading technological businesses. The selling of CPUs, memory and storage products, and other associated technologies generates the majority of the company's income.

Intel competes with a number of other semiconductor and technology firms, including Advanced Micro Devices (AMD), Qualcomm, NVIDIA, Samsung, and TSMC, among others. These firms compete with Intel in a variety of areas, including the development of faster and more powerful CPUs, advances in AI and machine learning, and the manufacture of more efficient and cost-effective semiconductor chips.

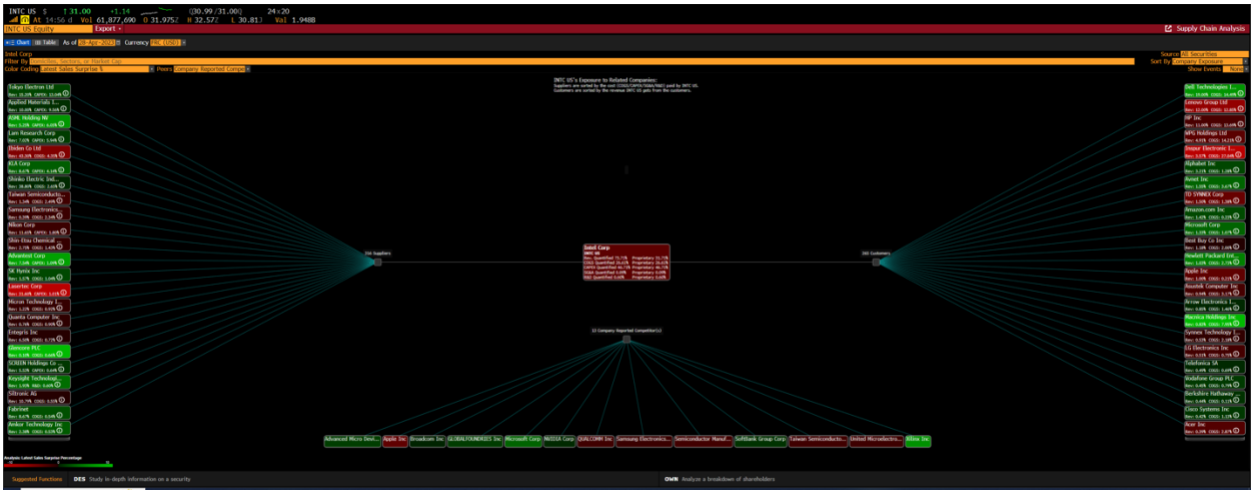
Despite competition, Intel is a semiconductor industry leader, with a significant presence in both consumer and corporate industries. To preserve its competitive advantage, the firm is always investing in research and development, and it has a history of developing and launching new technologies that propel the industry ahead.

I have used several functions in Bloomberg Terminal to deep dive into Intel's Business.

(Function used - CCB)

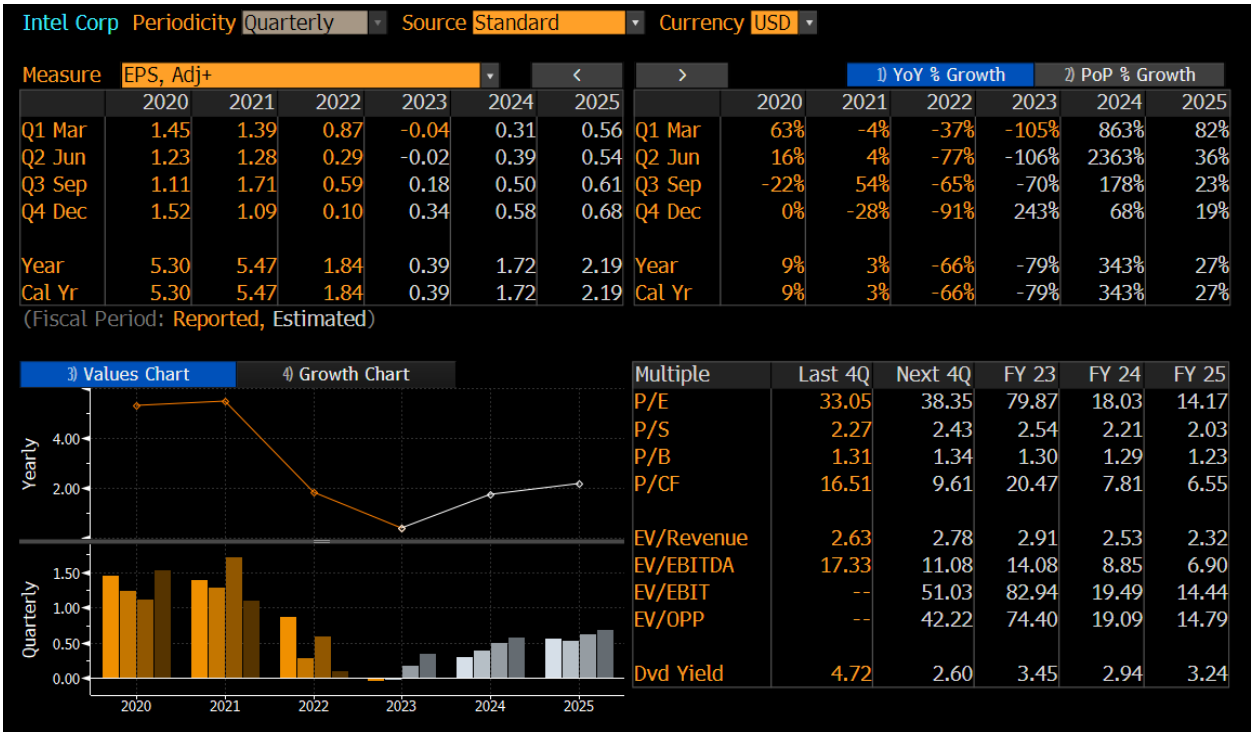
Here, Company classification on BICS from CCB gives descriptive analysis of Intel. It can be clearly seen that majority of revenues of Intel comes from Semiconductor devices. Intel has Market Cap of 121.17B as of April 2023.

Intel is a vast corporation and has a lot of suppliers as well as customers. I have used SPLC function in the terminal to analyze its data.



Use of SPLC function

Intel has a lot of suppliers and some of big suppliers include Tokyo Electron Ltd, ASML Holding NV, TSMC etc. Intel's biggest customers are Dell, Lenovo and HP amongst the likes of Alphabet Inc and Amazon Corp.



	2020	2021	2022	2023	2024	2025
Q1 Mar	1.45	1.39	0.87	-0.04	0.31	0.56
Q2 Jun	1.23	1.28	0.29	-0.02	0.39	0.54
Q3 Sep	1.11	1.71	0.59	0.18	0.50	0.61
Q4 Dec	1.52	1.09	0.10	0.34	0.58	0.68
Year	5.30	5.47	1.84	0.39	1.72	2.19
Cal Yr	5.30	5.47	1.84	0.39	1.72	2.19

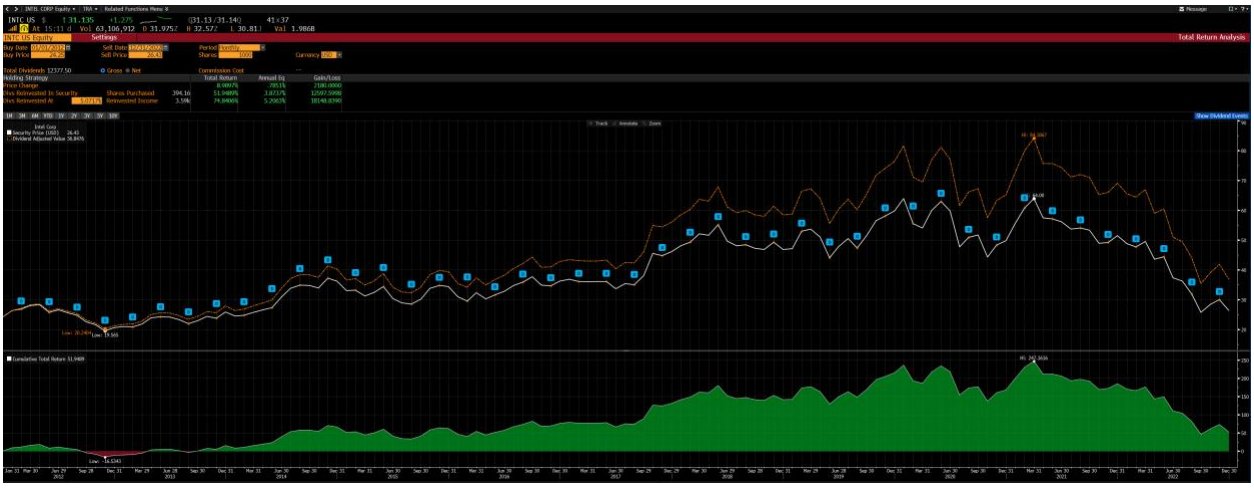
	2020	2021	2022	2023	2024	2025
Q1 Mar	63%	-4%	-37%	-105%	863%	82%
Q2 Jun	16%	4%	-77%	-106%	2363%	36%
Q3 Sep	-22%	54%	-65%	-70%	178%	23%
Q4 Dec	0%	-28%	-91%	243%	68%	19%
Year	9%	3%	-66%	-79%	343%	27%
Cal Yr	9%	3%	-66%	-79%	343%	27%

Multiple	Last 4Q	Next 4Q	FY 23	FY 24	FY 25
P/E	33.05	38.35	79.87	18.03	14.17
P/S	2.27	2.43	2.54	2.21	2.03
P/B	1.31	1.34	1.30	1.29	1.23
P/CF	16.51	9.61	20.47	7.81	6.55
EV/Revenue	2.63	2.78	2.91	2.53	2.32
EV/EBITDA	17.33	11.08	14.08	8.85	6.90
EV/EBIT	--	51.03	82.94	19.49	14.44
EV/OPP	--	42.22	74.40	19.09	14.79
Dvd Yield	4.72	2.60	3.45	2.94	3.24

Earning & Revenue Projections (Use of EM)

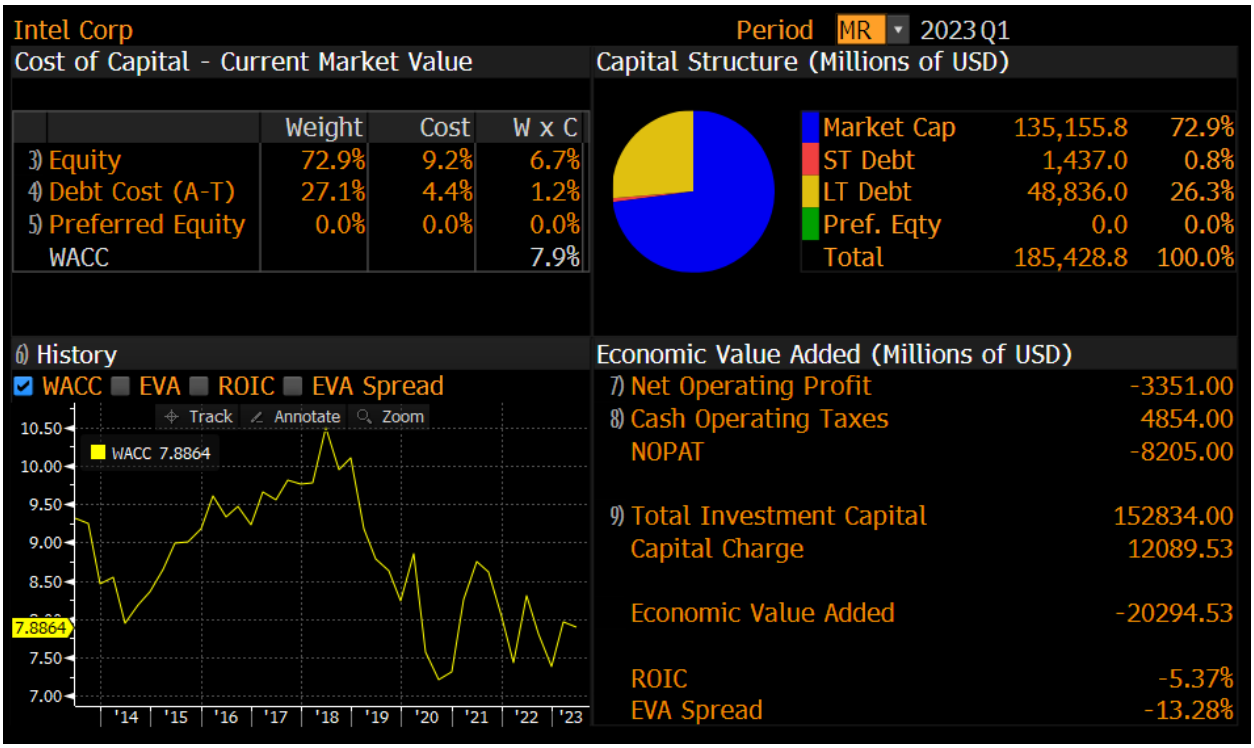
Here I used EM function and set measures for EPS. In the screenshot quarterly periodicity is shown from 2020 to 2025. Yellow numbers are historic data and white numbers are projections. EPS for Q1 March 2024 is in plus from -0.04 of Q1 March 2023.

After this analysis, I used Total Return Analysis (TRA) Function to see further data.



Use of TRA

From TRA we can see that Cumulative Total Return is 51.9489% which is a great looking magnitude of Intel. Monthly Volumes are a bit low at 394 shares traded. Security price is 26.43 and dividend adjusted value is at 36.84 which is higher. Dividends are reinvested at 5.07% for Intel Corp.



Use of WACC

I used Weighted Average Cost of Capital for Intel Corp and this is the representation of it in Bloomberg terminal. Equity is weighted at 72.9% and debt cost is 27.1%. We can see that net operating profit is -3351 million USD which is bad.

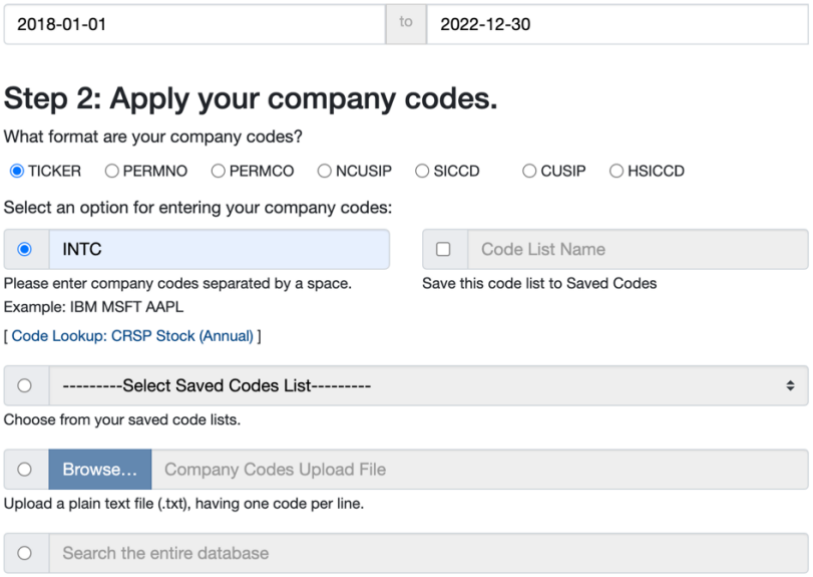
I had used GF function for Graph fundamentals of Intel and its competitor in semi-conductor industry Qualcomm against S&P 500.

Graph fundamentals -GF

The white line represents Qualcomm and green one represents Intel along with brown line representing S&P 500 index. We can see that S&P 500 index shows a rising trend from September 2020 in both revenues and basic earnings per share while from June 2022 onwards there is a downfall for both Intel and Qualcomm for basic earnings per share and revenues. All the representation in graphs are shown quarterly.

Time Series Analysis with Python

For stock analysis of Intel, I retrieved data from Wharton Research Data Services (WRDS). I used CRPS Daily Stock function for data retrieval.



I selected data from 1st January 2018 to 30th December 2020. I selected ticker as INTC which is official ticker for Intel Corporation and selected High, Low, Volume, Close and Open price of stock for each date. This gave me a CSV file containing all information requested for python analysis.

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

In [5]: data_ret = pd.read_csv("INTC 2.csv")
        data_ret
```

Out[5]:

	Date	Low	High	Vol	Close	Open
0	01/02/2018	46.2100	46.900	23370791	46.83	46.380
1	01/03/2018	43.6500	46.210	116804036	45.24	45.470
2	01/04/2018	42.6900	44.650	89209521	44.43	43.520
3	01/05/2018	43.9000	45.150	41824006	44.69	44.430
4	01/08/2018	43.9600	44.840	33733769	44.71	44.270
...	...	...	...	...	...	...
1254	12/23/2022	25.6835	26.190	23540515	26.09	25.920
1255	12/27/2022	25.6500	26.100	29334593	25.95	25.845
1256	12/28/2022	25.5000	26.115	26059030	25.53	25.840
1257	12/29/2022	25.7500	26.290	30766030	26.19	25.770
1258	12/30/2022	25.8000	26.460	30775951	26.42	25.902

1259 rows × 6 columns

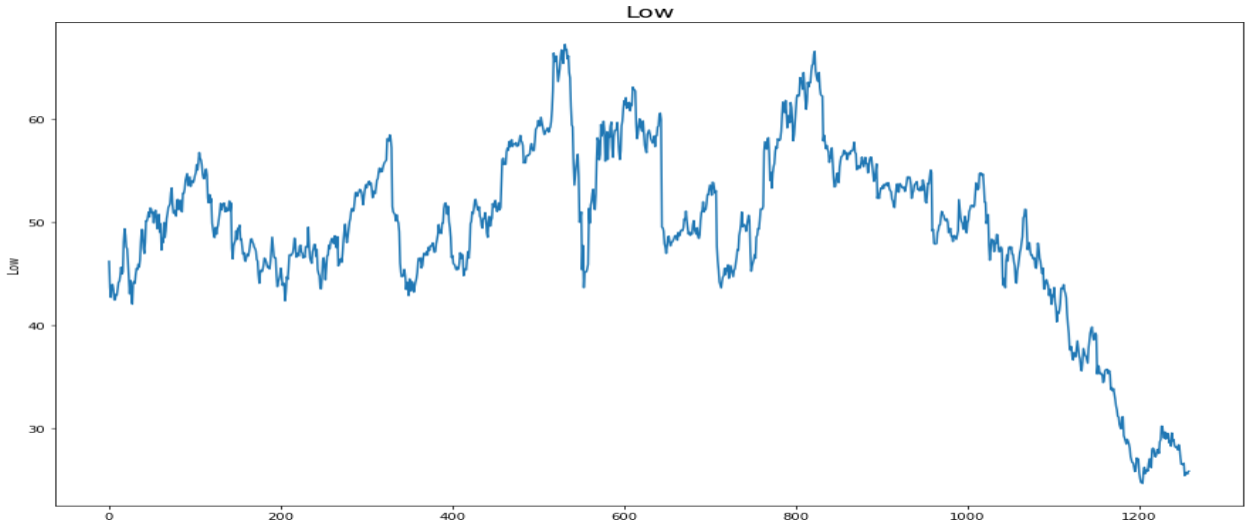
I imported data for pre-processing and using pandas, I read the data CSV file. There are 1259 rows and 6 columns of data for analysis.

```
In [6]: data_ret.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1259 entries, 0 to 1258
Data columns (total 6 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    1259 non-null   object 
 1   Low     1259 non-null   float64
 2   High    1259 non-null   float64
 3   Vol     1259 non-null   int64  
 4   Close   1259 non-null   float64
 5   Open    1259 non-null   float64
dtypes: float64(4), int64(1), object(1)
memory usage: 59.1+ KB
```

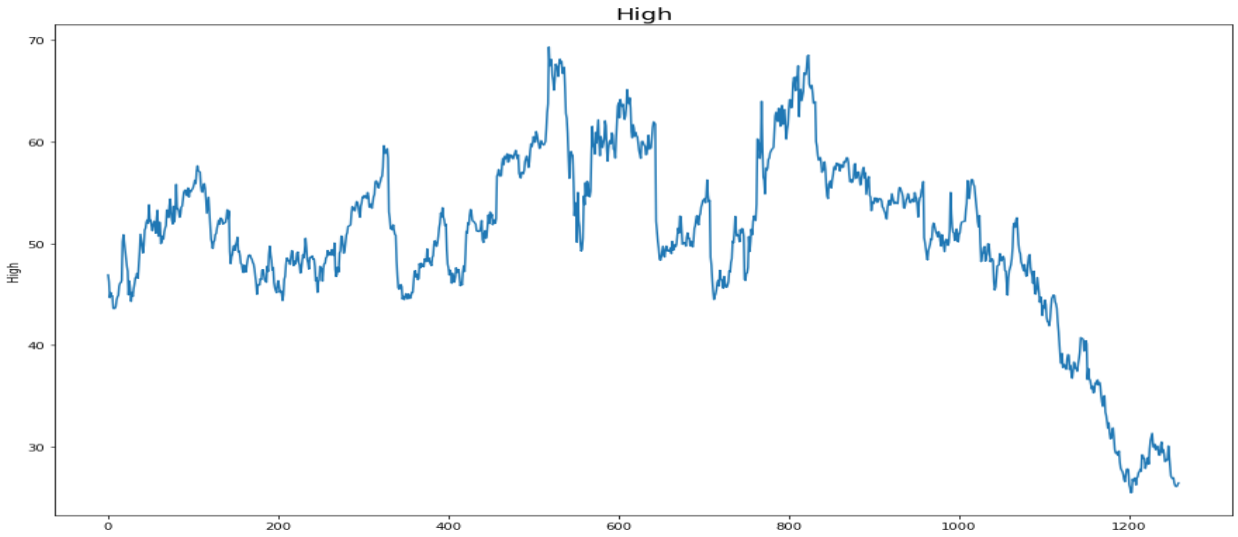
We can see that only Volume is in int datatype and other columns are of float datatype and date is in Object.

```
In [11]: for column in data_ret.columns:
             plt.figure(figsize=(15,10))
             sns.lineplot(x = data_ret.index, y = data_ret[column], data = data_ret)
             plt.title(column, fontsize = 18)
             plt.show()
```

For visualizing the data, I used matplotlib library in python and following code to generate the graphs for each column. This is what I achieved from this operation.



[INTC LOW]



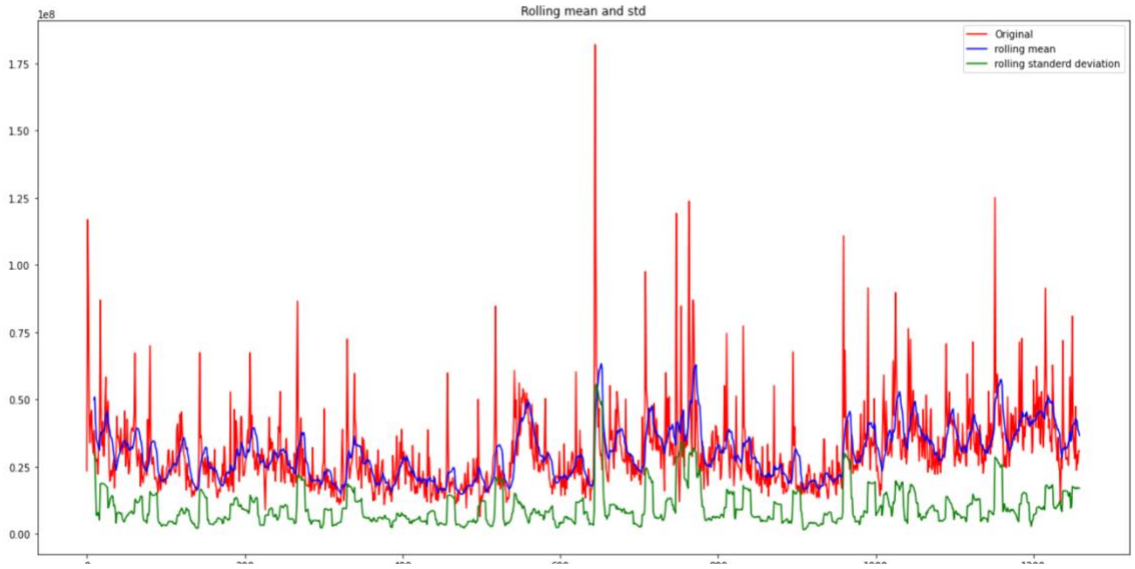
[INTC High]

These graphs show Low and High prices of Intel stock from 1st January 2018 to 31st December 2022. Just like these two graphs other graphs for volume of stock, open and close price of stock were generated with the code.

```
In [14]: from statsmodels.tsa.stattools import adfuller
def test_stat(timeseries):
    MA = timeseries.rolling(window=10).mean() #MEAN
    MSTD = timeseries.rolling(window=10).std() #Standerd Deviation

    plt.figure(figsize=(20,10))
    org = plt.plot(timeseries, color = 'red', label = 'Original')
    mean = plt.plot(MA, color = 'blue', label = 'rolling mean')
    std = plt.plot(MSTD, color = 'green', label = 'rolling standerd deviation')
    plt.title('Rolling mean and std')
    plt.legend(loc = 'best')
    plt.show()

    print('Results of Dicky-fuller Test:')
    dftest = adfuller(timeseries, analog = 'AIC')
    dfout = pd.Series(dftest[0:4], index =['Test Statistic', 'p-value', '#Lags used', 'No. of observations'])
    for key,value in dftest[4].items():
        dfout['Critical value (%s)'%key] = value
    print(dfout)
```

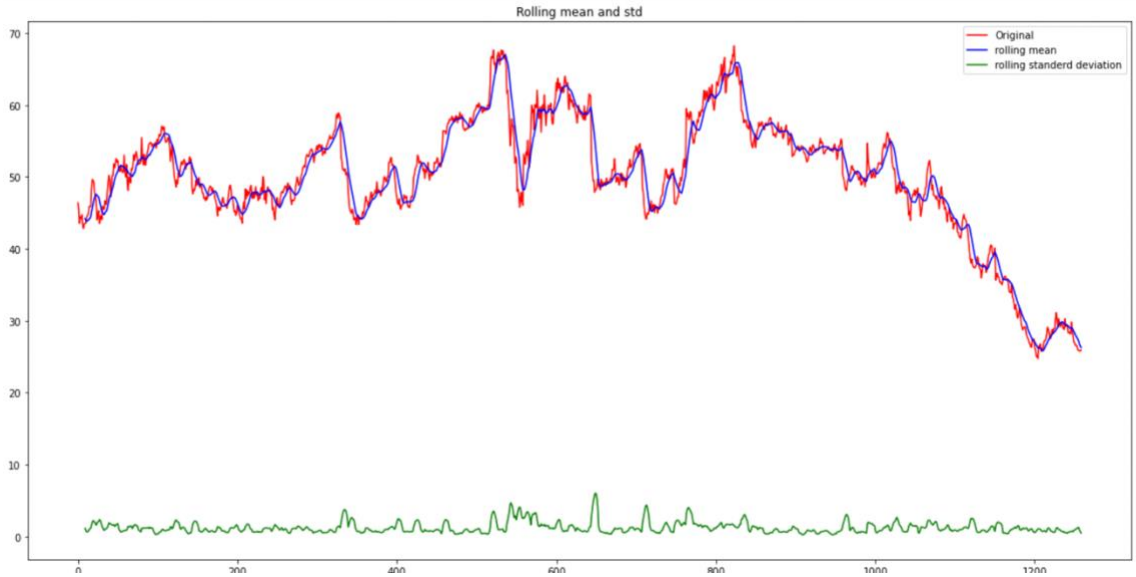
This is the code I ran for Dickey-Fuller test. The Dickey-Fuller test in statistics examines the null hypothesis that a unit root exists in an autoregressive (AR) time series model. The alternative hypothesis varies based on the version of the test employed, but it is often stationarity or trend-stationarity.

```
In [16]: test_stat(data_ret['Vol'])
```



This is the rolling mean and standard deviation graph generated for Volume in the dataset. This graph can be generated for every column in the data like “Open” price of stock like below.

```
In [17]: test_stat(data_ret['Open'])
```



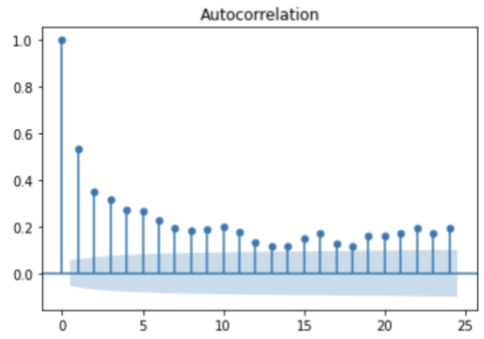
From this graph we can see that mean as well as standard deviation is decreasing. This is the result of signed values being compared to absolute values.

Let’s move now to the modeling part. There are 2 popular plotting techniques.

Autocorrelation Function (ACF):

The autocorrelation function (ACF) describes how data points in a time series are connected to each other on average. In other words, it assesses the signal's self-similarity over multiple delay durations.

```
from statsmodels.graphics import tsaplots

tsaplots.plot_acf(data_ret['Vol'], lags = 24)
plt.figure(figsize = (20,10))
plt.show()
```



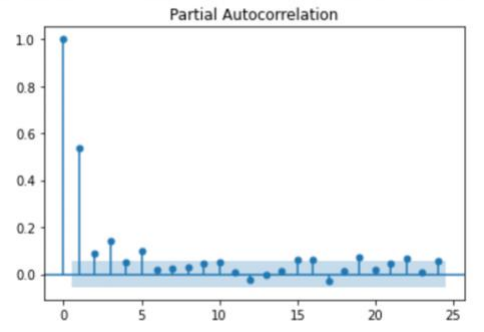

```
<Figure size 1440x720 with 0 Axes>
```

Here, bar represents the size and direction of correlation.

Partial Autocorrelation Function (PACF):

The partial autocorrelation function (PACF) computes the partial correlation of a stationary time series with its own lagged values, regressing the time series values at all shorter lags.

```
tsaplots.plot_pacf(data_ret['Vol'], lags = 24
plt.figure(figsize=(20,10))
plt.show()
```



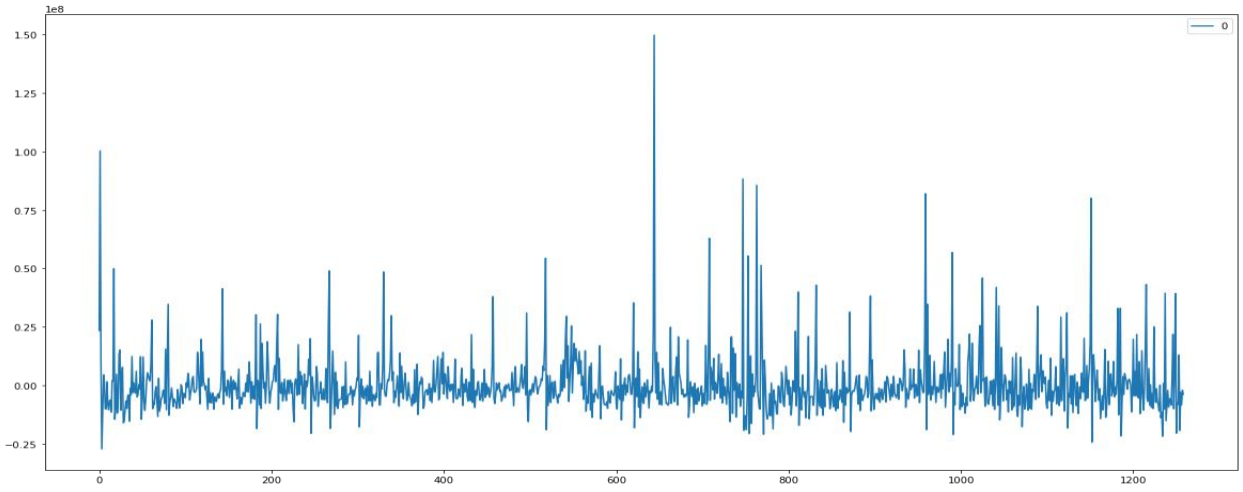

```
<Figure size 1440x720 with 0 Axes>
```

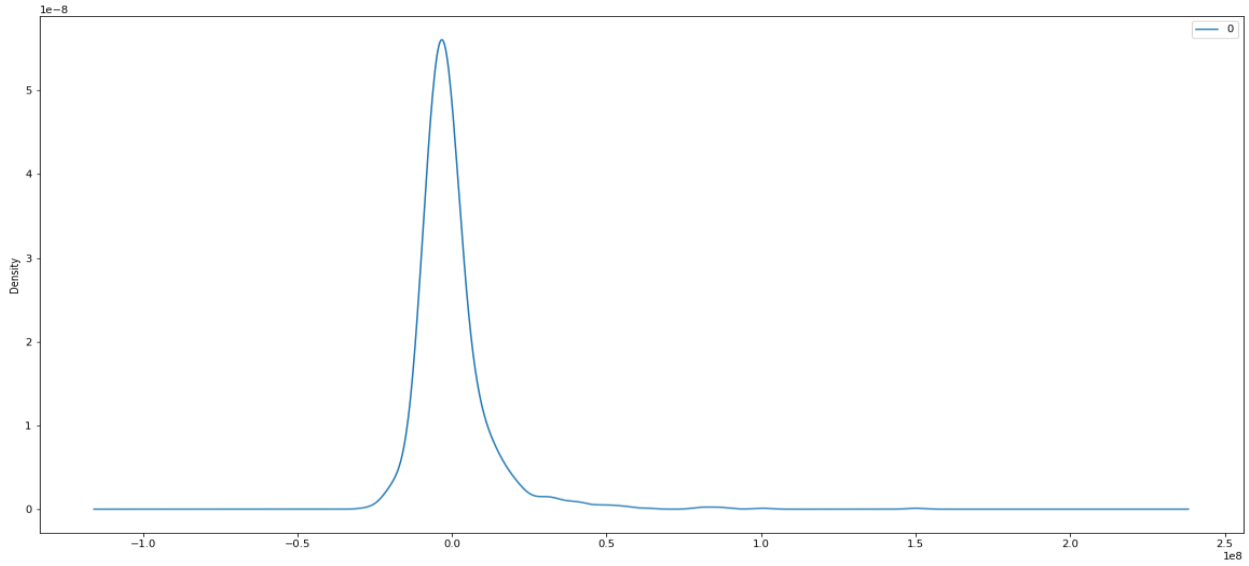
ARIMA (Auto Regressive Integrated Moving Average) Model:

An autoregressive integrated moving average model is a type of regression analysis that measures the strength of one dependent variable in relation to other variables that change. The purpose of the model is to forecast future securities or financial market movements by studying the discrepancies between values in the series rather than actual values.

```
from statsmodels.tsa.arima.model import ARIMA

model = ARIMA(data_ret['Vol'], order = (1,1,2))
model_fit = model.fit()
#print(model.fit.summary())

residuals = pd.DataFrame(model_fit.resid)
residuals.plot(figsize = (20,10))
plt.show()

residuals.plot(kind = 'kde', figsize = (20,10))
plt.show()
print(residuals.describe())
```

After running the code, I got a graph plotted which is given below. ARIMA is really helpful for stock analysis.






```
count   1.259000e+03
mean    3.421959e+04
std     1.243518e+07
min    -2.729686e+07
25%    -6.298010e+06
50%    -2.255446e+06
75%     2.627126e+06
max     1.497164e+08
```

At the end data for Intel (INTC) was useful for Time Series Analysis for a period. The analysis was successful, and report is generated. From this analysis, we learned Bloomberg Terminal functions as well as behavior of intel data over a period and basic of stock evaluation.