R.V. COLLEGE OF ENGINEERING,
BENGALURU-560059
(Autonomous Institution Affiliated to VTU, Belagavi)

VISUALISATION AND COMPARISON OF NIFTY50 STOCK DATASETS

Exploratory Data Analysis Of Datasets

Python PLC Theory EL Report

Submitted By :-

SNEHIL VUKKUSILA --- RVCE22BAI006

PARTH SHUKLA---RVCE22BAI031

Department of Artifial Intelligence And Machine Learning,
RV COLLEGE OF ENGINEERING®, BENGALURU – 560059

Outline

- Introduction

- Problem Statement

- Objectives

- Methodology

- Results

- Conclusion

Introduction and Summary

Problem Statement

The Tata Group is one of the largest conglomerates in India with businesses ranging from steel, automobiles, and IT services to hotels, airlines, and consumer goods. Through analysis our task is performing an exploratory data analysis (EDA) on the stock prices of Tata Group companies over the last 10 years. We want to see how the stocks of a particular sector within a huge conglomerate as Tata, have grown along with the group as a whole, and the individual stock performances over a long period of growth. The goal of this project is to identify trends and patterns in the stock prices and the range of volatility of different Tata Group companies, and to provide insights that can help inform investment decisions, through visualisation and comparison of Volume, Price, Moving Averages, Market Capitalization.

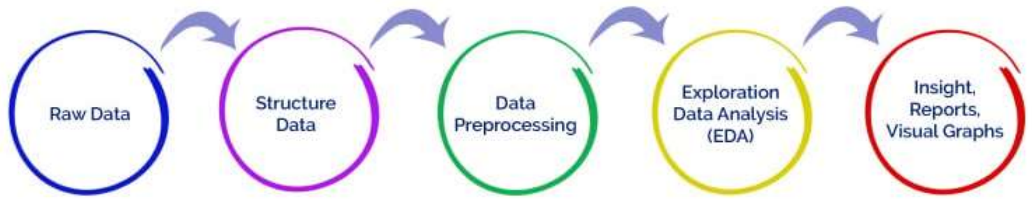
We will be utilizing past data of stocks present in the Nifty 50 Index, and by visualization and comparison we will be able to determine the risk , and future prospects for investment into those stock

Overview

The objective of this Exploratory Data Analysis is to apply a complete data-driven analytic approach to understand the trend in the price of a stock, and create comparisons with other stocks in same sector or group of companies.

It involves data collection, data wrangling, EDA with data visualization, Building a Dashboard with Plotly and matplotlib, Seaborn.

Utilising Data analysis libraries and modules such as NumPy , Pandas Scikitlearn . Visualization libraries and modules like matplotlib , Seaborn


Raw Data
Structure Data
Data Preprocessing
Exploration Data Analysis (EDA)
Insight, Reports, Visual Graphs

Understanding the Problem Statement

Broadly, stock market analysis is divided into two parts – Fundamental Analysis and Technical Analysis.

● Fundamental Analysis involves analyzing the company's future profitability on the basis of its current business environment and financial performance.

● Technical Analysis, on the other hand, includes reading the charts and using statistical figures to identify the trends in the stock market.

To understand the risk associated with it there must be a proper analysis of stock before buying it.

Objectives to find answers:

1)Utilising previous data predict of price, volume, volatility, market capitalization, we will try to analyse the patterns and the similarities of the stock price movement over a period of 15 years worth of data.

2)We will visualize the various parameters that influence stock analysis, which will affect the decision of an investor.

3)Plot and show how stocks have different volatility values and ROI , and why some stocks a more risky and some are consolidated in nature.

4)Analyse the moving averages of 50 day and 200 day price values, and plot these with respect to the daily values, and see how we can predict the future movements.

Section 1

Methodology

Methodology

•Data collection methodology:

•The data was collected by sending get request to Kaggle API and by web scraping NSE INDIA and Nifty 50 data sets Records.

•Perform data wrangling on the csv files which have been generated.

•Dealing with missing values, creating new columns, dropping irrelevant columns and visualizing through Panda's data frames

•Perform exploratory data analysis (EDA) using Pandas and Numpy

•Perform visual analytics Plotly , Matplotlib

•Perform comparative analysis using plots and graphs

Including, Price Analysis, Volume Comparison, 50-day and 200-day Moving Average comparison, Volatility Histogram

•Comparison analysis

Data Collection

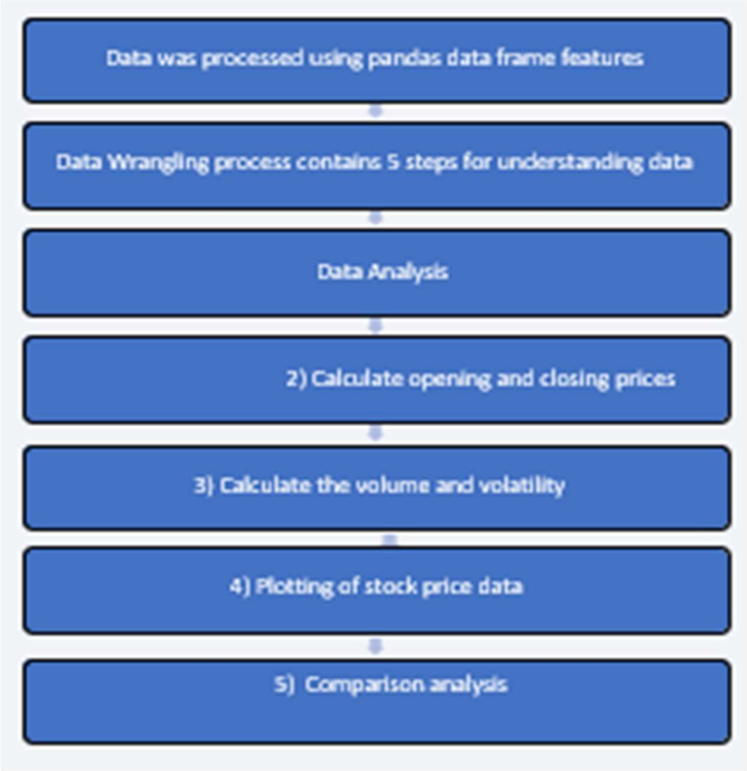
How was the data collected?

•The data is collected from the Kaggle dataset, and the API will give us data in form of a zipped file containing csv files of various stock data.

•We also use NSEIndia website to extract csv files.

•We extract the data set into the form of rows and columns which can be easily operated upon, by using pandas, which can easily access and parse through CSV files.

Data Wrangling



Data was processed using pandas data frame features

Data Wrangling process contains 5 steps for understanding data

Data Analysis

2) Calculate opening and closing prices

3) Calculate the volume and volatility

4) Plotting of stock price data

5) Comparison analysis

Data description:

•**Open:** The price of the stock when the market opens in the morning.

•**Close:** The price of the stock when the market closed in the evening.

•**High:** Highest price the stock reached during that day.

•**Low:** Lowest price the stock is traded on that day.

•**Volume:** The total amount of stocks traded on that day.

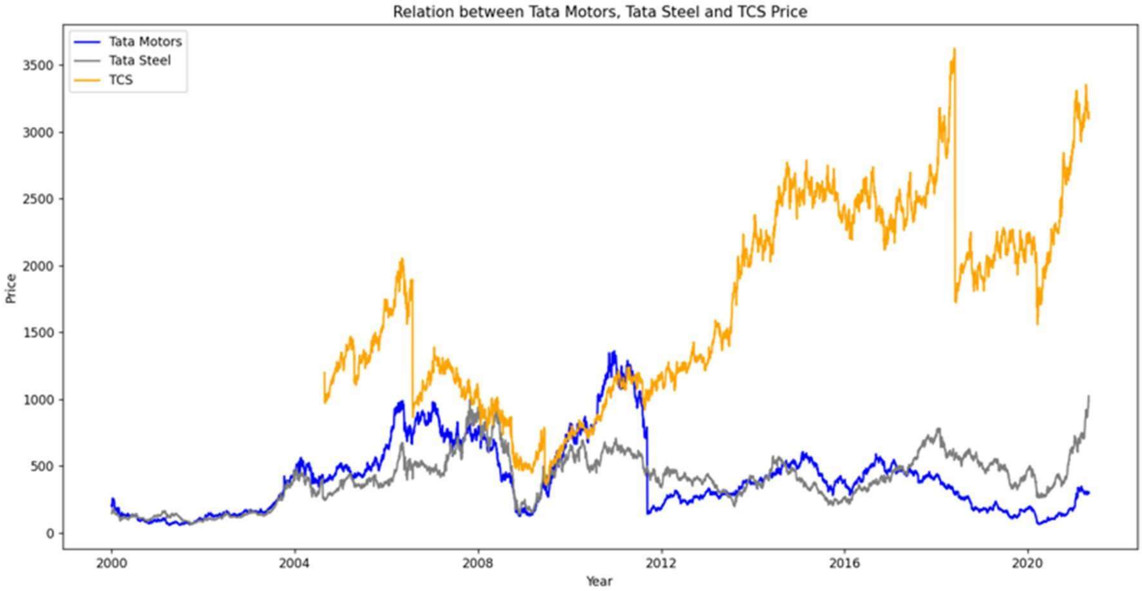
What is Stock Market Analysis?

Stock market analysis enables investors to identify the intrinsic worth of a security even before investing in it. All stock market tips are formulated after thorough research by experts. Stock analysts try to find out activity of an instrument/sector/market in future. By using stock analysis, investors and traders arrive at equity buying and selling decisions. Studying and evaluating past and current data helps investors and traders to gain an edge in the markets to make informed decisions. Fundamental Research and Technical Research are two types of research used to first analyse and then value a security.

Section 2

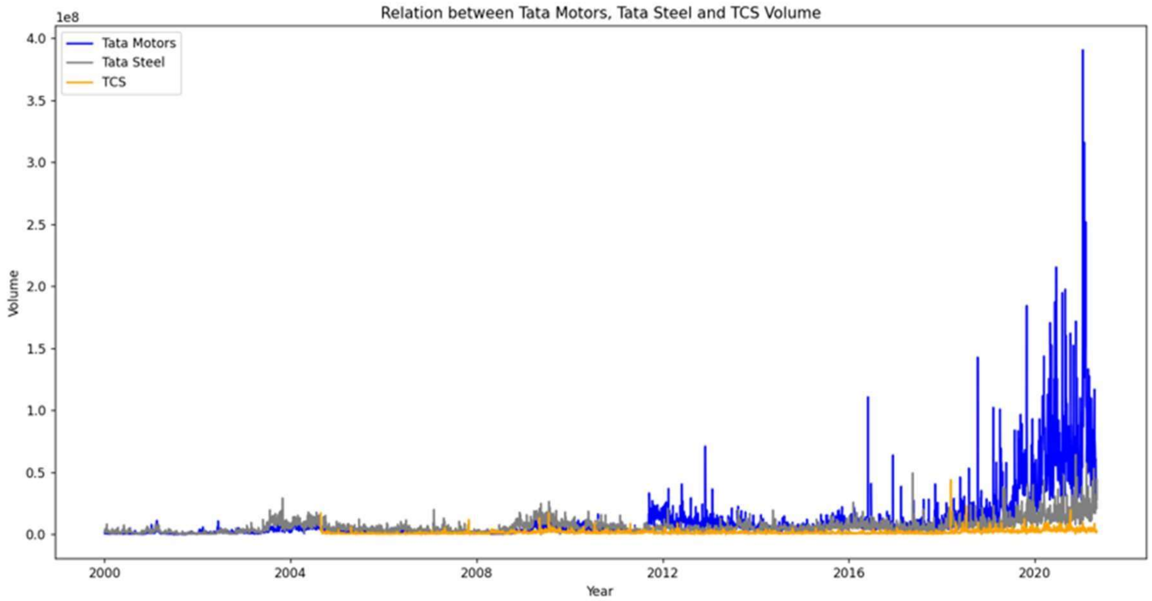
Insights drawn from EDA

Comparison of Price



According to the graph above, the price of TCS has skyrocketed significantly higher than that of Tata Steel and Tata Motors. TCS's pricing trajectory has been generally upward from its beginning, whereas Tata Steel and Tata Motors have been more on a consolidation trend, this shows the innovative IT industry as a whole has developed vastly, as compared to a legacy generalised industreis such as steel ore and motor vehicles are more slow growth, but stable and guaranteed to be used for many generations.

Comparison of Volume

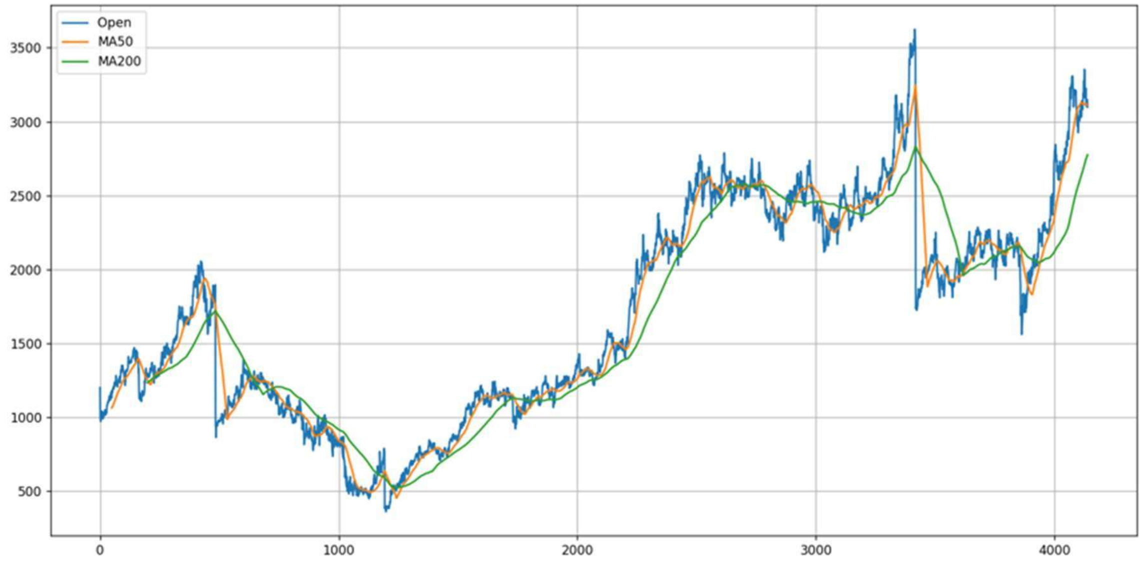


Though the price of TCS has risen more significantly as compared to Tata Steel and Tata Motors, we can notice from the above graph that TCS has the least volume signifying that the stock has been traded comparatively less as compared to Tata Steel and Tata Motors and is lesser liquid.

Tata Motors on the other hand has been traded the most signifying higher liquidity, and better order execution.

We can see that traders and investors who may be of a retail or FII level investment firms, have more confidence on a stable industry such as Tata Steel and Tata Motors and are wary about the IT industry as it is still growing at such a rapid pace without a proven track record.

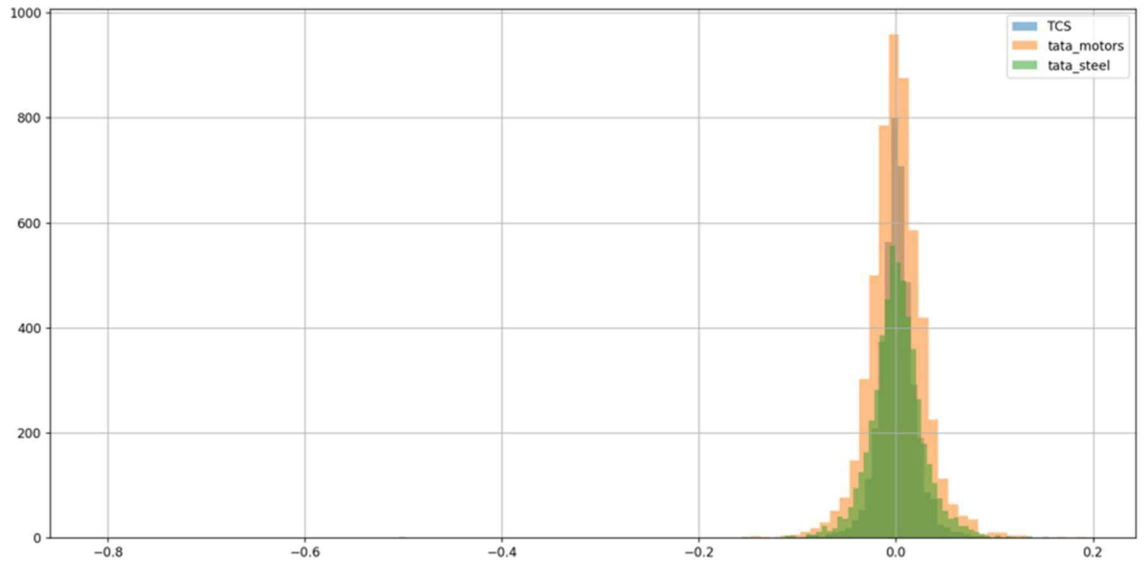
Moving Averages



The above shows the price, along with the 50 day and 200 day moving average plots. As we know the stock prices are highly volatile and prices change quickly with time. To observe any trend or pattern we can take the help of a

50-day 200-day average

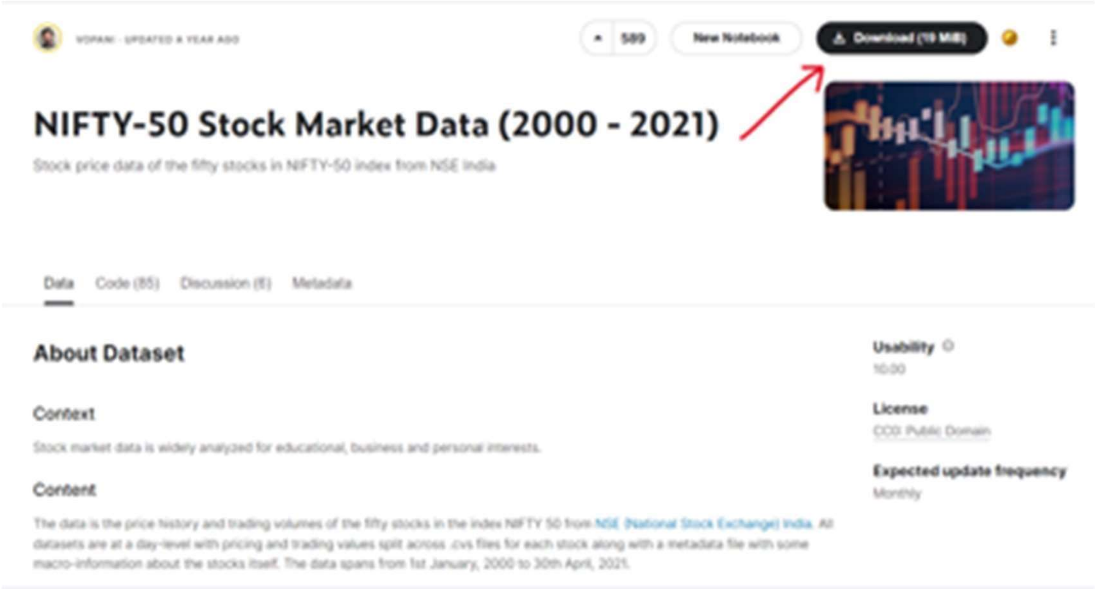
Volatility Analysis



It is clear from the graph that the percentage increase in stock price histogram for TCS is the widest which indicates the stock of TCS is the most volatile among the three companies compared.

The width of the histogram shape is related to the standard deviation of the distribution. The higher it is the more variants we have, the variants of a certain distribution are directly related to the standard deviation. It’s actually the square value of the standard deviation, the higher the standard deviation is, the higher the variance is. The more volatility we have in terms of stock analysis.

Data Collection and Wrangling



NIFTY-50 Stock Market Data (2000 - 2021)

Stock price data of the fifty stocks in NIFTY-50 index from NSE India

About Dataset

Context

Stock market data is widely analyzed for educational, business and personal interests.

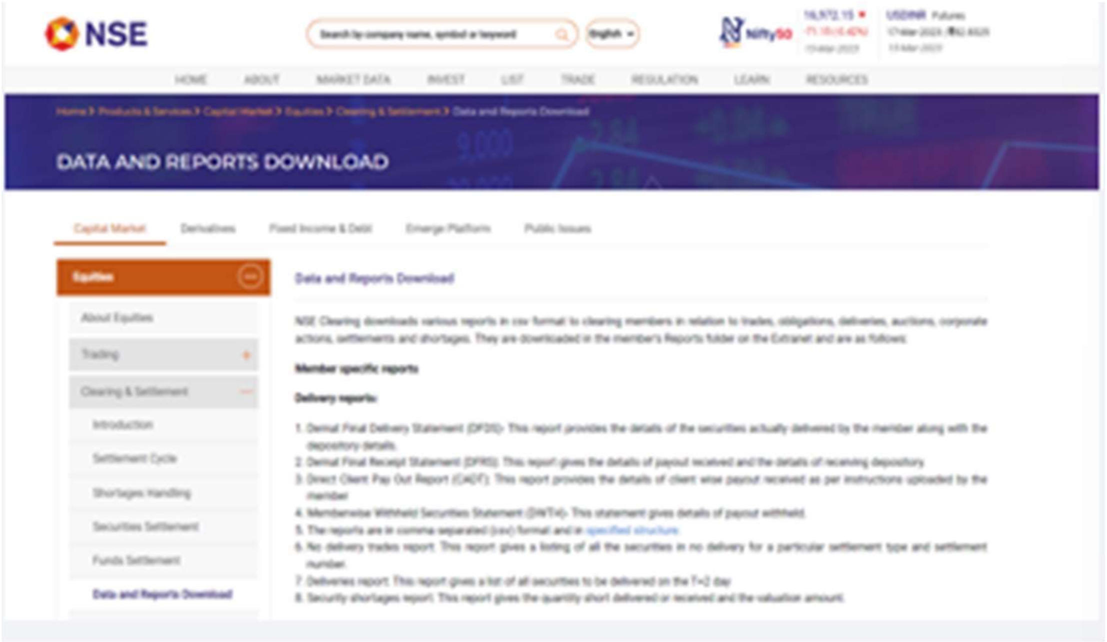
Content

The data is the price history and trading volumes of the fifty stocks in the index NIFTY 50 from NSE (National Stock Exchange) India. All datasets are at a day-level with pricing and trading values split across .cvs files for each stock along with a metadata file with some macro-information about the stocks itself. The data spans from 1st January, 2000 to 30th April, 2021.

Usability: 10.00

License: CC0: Public Domain

Expected update frequency: Monthly



DATA AND REPORTS DOWNLOAD

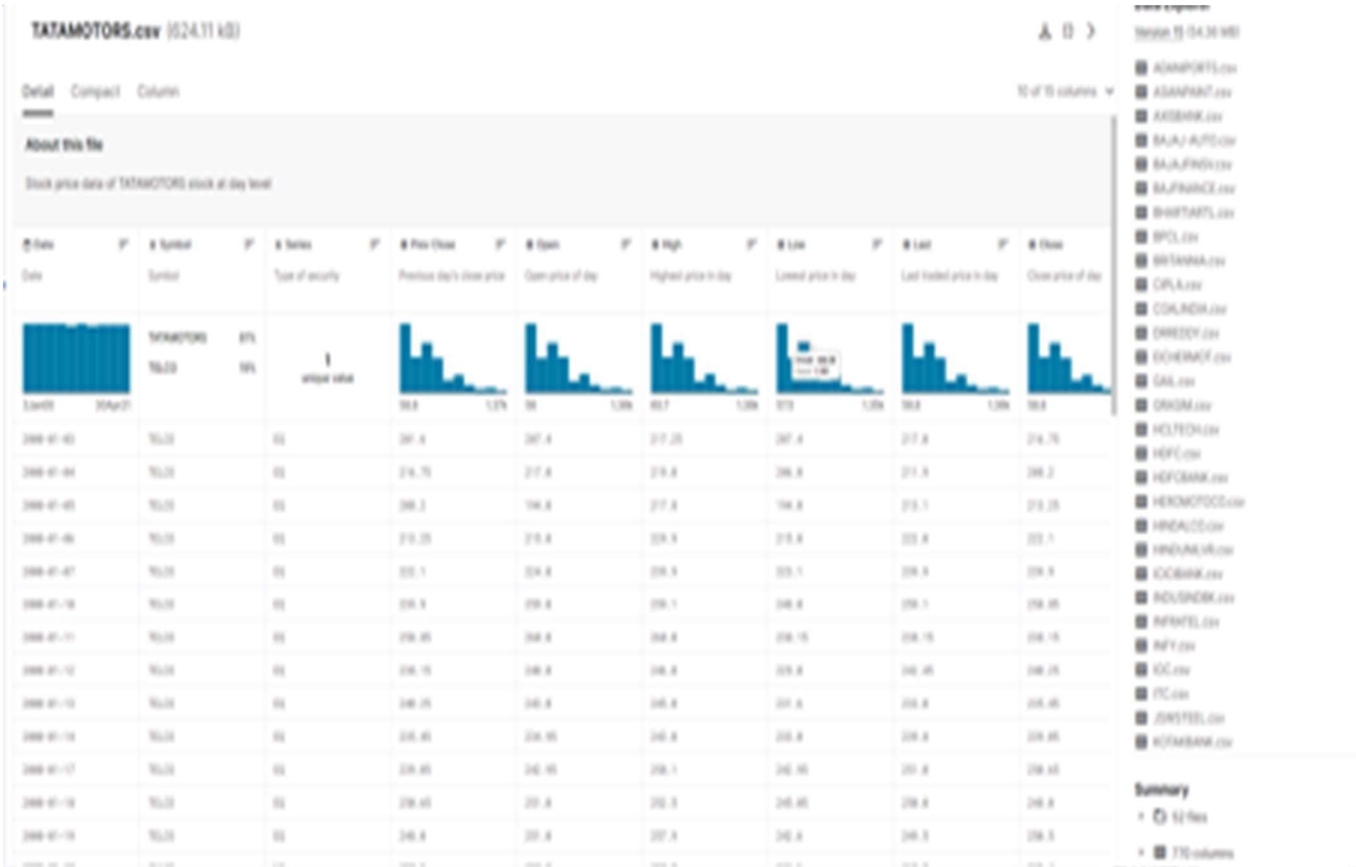
Data and Reports Download

NSE Clearing downloads various reports in csv format to clearing members in relation to trades, obligations, deliveries, auctions, corporate actions, settlements and shortages. They are downloaded in the member's Reports folder on the Extranet and are as follows:

Member specific reports

Delivery reports:

1. Demat Final Delivery Statement (DFDS)- This report provides the details of the securities actually delivered by the member along with the depository details.
2. Demat Final Receipt Statement (DFRS): This report gives the details of payout received and the details of receiving depository.
3. Direct Client Pay Out Report (CADT): This report provides the details of client wise payout received as per instructions uploaded by the member
4. Memberwise Withheld Securities Statement (DWTH)- This statement gives details of payout withheld.
5. The reports are in comma separated (csv) format and in specified structure.
6. No delivery trades report. This report gives a listing of all the securities in no delivery for a particular settlement type and settlement number.
7. Deliveries report. This report gives a list of all securities to be delivered on the T+2 day
8. Security shortages report. This report gives the quantity short delivered or received and the valuation amount.



TATAMOTORS.csv (624.11 kB)

Detail Compact Column

10 of 15 columns

About this file

Stock price data of TATAMOTORS stock at day level

Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close
Date	Symbol	Type of security	Previous day's close price	Open price of day	Highest price in day	Lowest price in day	Last traded price in day	Close price of day
3Jan00 – 30Apr21	TATAMOTORS 81% TELCO 19%	1 unique value	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-03	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-04	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-05	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-06	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-07	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-10	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-11	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-12	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-13	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-14	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-17	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-18	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2000-01-19	TELCO	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

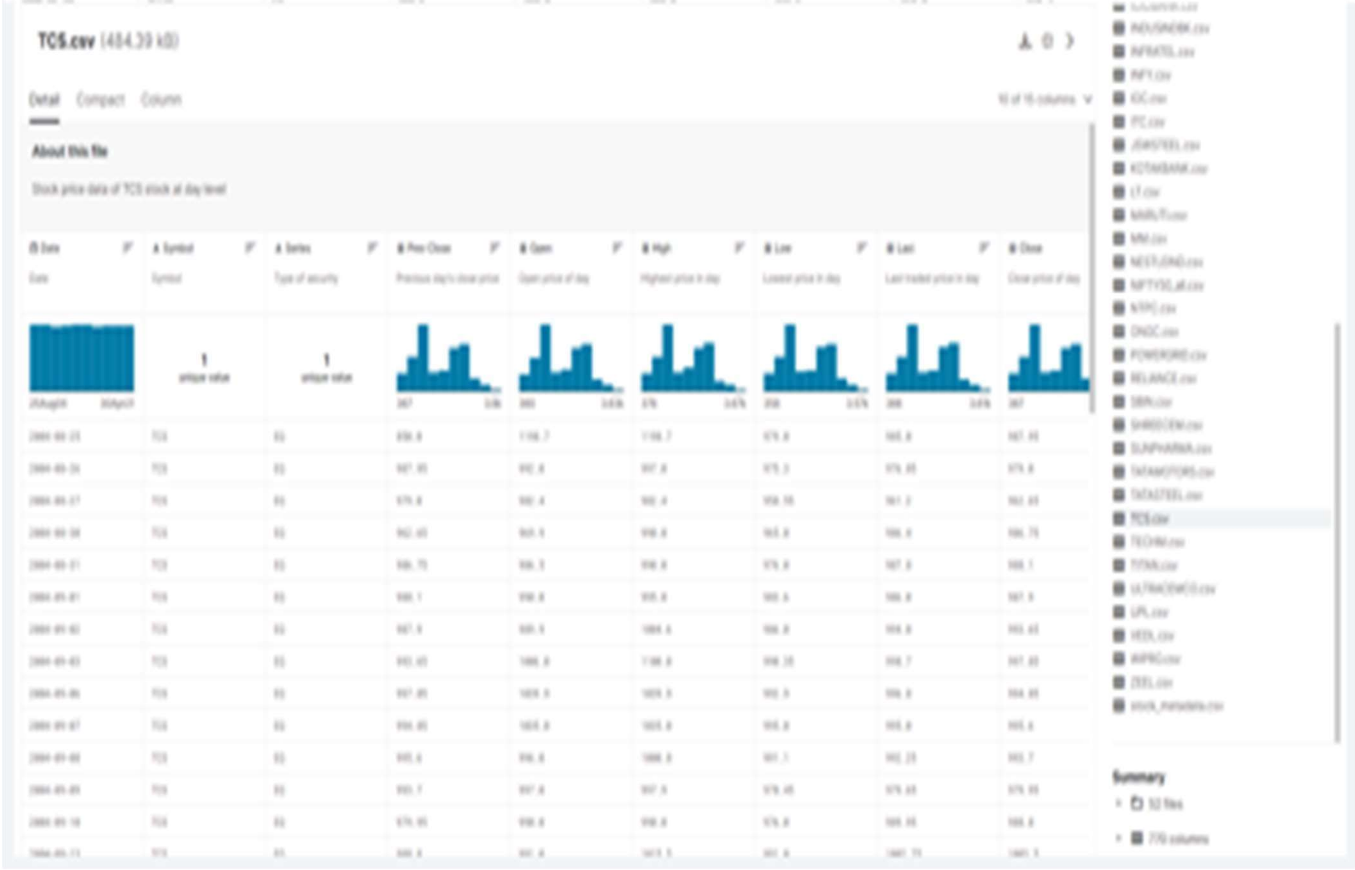
Data Explorer

Version 15 (54.36 MB)

- ADANIPORTS.csv
- ASIANPAINT.csv
- AXISBANK.csv
- BAJAJ-AUTO.csv
- BAJAJFINSV.csv
- BAJFINANCE.csv
- BHARTIARTL.csv
- BPCL.csv
- BRITANNIA.csv
- CIPLA.csv
- COALINDIA.csv
- DRREDDY.csv
- EICHERMOT.csv
- GAIL.csv
- GRASIM.csv
- HCLTECH.csv
- HDFC.csv
- HDFCBANK.csv
- HEROMOTOCO.csv
- HINDALCO.csv
- HINDUNILVR.csv
- ICICIBANK.csv
- INDUSINDBK.csv
- INFRATEL.csv
- INFY.csv
- IOC.csv
- ITC.csv
- JSWSTEEL.csv
- KOTAKBANK.csv

Summary

- 52 files
- 770 columns



TCS.csv (484.39 kB)

Detail Compact Column

10 of 15 columns

About this file

Stock price data of TCS stock at day level

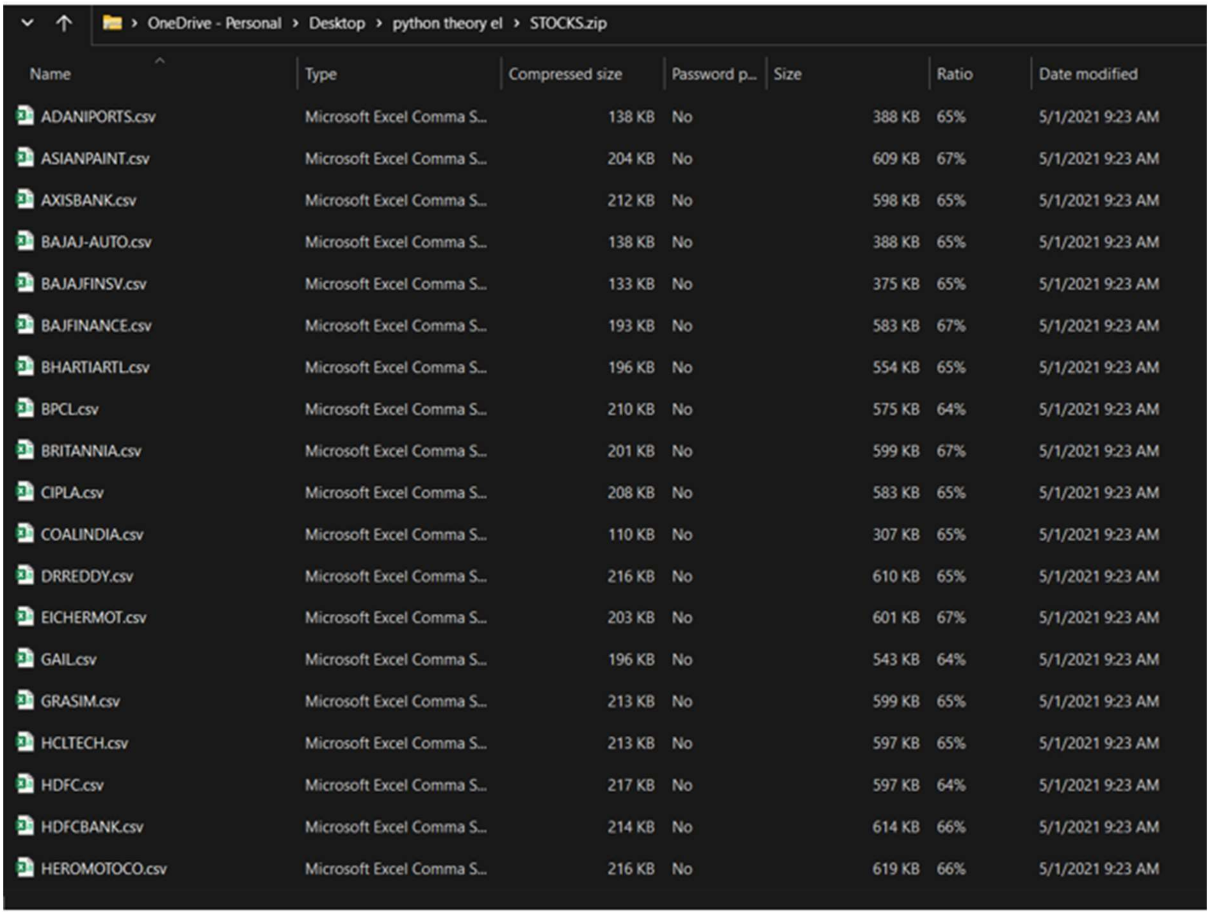
Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close
Date	Symbol	Type of security	Previous day's close price	Open price of day	Highest price in day	Lowest price in day	Last traded price in day	Close price of day
25Aug04 – 30Apr21	1 unique value	1 unique value	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-08-25	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-08-26	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-08-27	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-08-30	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-08-31	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-01	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-02	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-03	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-06	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-07	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-08	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-09	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
2004-09-10	TCS	EQ	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

- INDUSINDBK.csv
- INFRATEL.csv
- INFY.csv
- IOC.csv
- ITC.csv
- JSWSTEEL.csv
- KOTAKBANK.csv
- LT.csv
- MARUTI.csv
- MM.csv
- NESTLEIND.csv
- NIFTY50_all.csv
- NTPC.csv
- ONGC.csv
- POWERGRID.csv
- RELIANCE.csv
- SBIN.csv
- SHREECEM.csv
- SUNPHARMA.csv
- TATAMOTORS.csv
- TATASTEEL.csv
- TCS.csv
- TECHM.csv
- TITAN.csv
- ULTRACEMCO.csv
- UPL.csv
- VEDL.csv
- WIPRO.csv
- ZEEL.csv
- stock_metadata.csv

Summary

- 52 files
- 770 columns

Data Collection and Wrangling

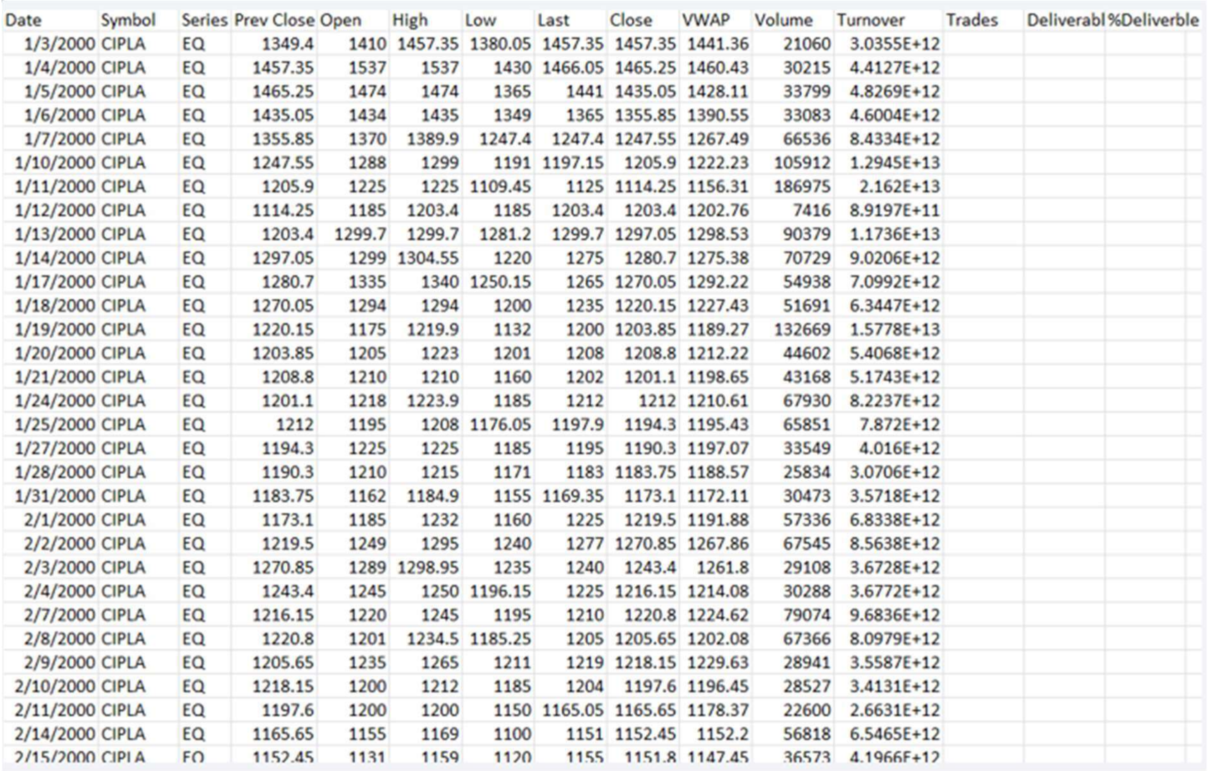


OneDrive - Personal › Desktop › python theory el › STOCKS.zip

Name	Type	Compressed size	Password p...	Size	Ratio	Date modified
ADANIPORTS.csv	Microsoft Excel Comma S...	138 KB	No	388 KB	65%	5/1/2021 9:23 AM
ASIANPAINT.csv	Microsoft Excel Comma S...	204 KB	No	609 KB	67%	5/1/2021 9:23 AM
AXISBANK.csv	Microsoft Excel Comma S...	212 KB	No	598 KB	65%	5/1/2021 9:23 AM
BAJAJ-AUTO.csv	Microsoft Excel Comma S...	138 KB	No	388 KB	65%	5/1/2021 9:23 AM
BAJAJFINSV.csv	Microsoft Excel Comma S...	133 KB	No	375 KB	65%	5/1/2021 9:23 AM
BAJFINANCE.csv	Microsoft Excel Comma S...	193 KB	No	583 KB	67%	5/1/2021 9:23 AM
BHARTIARTL.csv	Microsoft Excel Comma S...	196 KB	No	554 KB	65%	5/1/2021 9:23 AM
BPCL.csv	Microsoft Excel Comma S...	210 KB	No	575 KB	64%	5/1/2021 9:23 AM
BRITANNIA.csv	Microsoft Excel Comma S...	201 KB	No	599 KB	67%	5/1/2021 9:23 AM
CIPLA.csv	Microsoft Excel Comma S...	208 KB	No	583 KB	65%	5/1/2021 9:23 AM
COALINDIA.csv	Microsoft Excel Comma S...	110 KB	No	307 KB	65%	5/1/2021 9:23 AM
DRREDDY.csv	Microsoft Excel Comma S...	216 KB	No	610 KB	65%	5/1/2021 9:23 AM
EICHERMOT.csv	Microsoft Excel Comma S...	203 KB	No	601 KB	67%	5/1/2021 9:23 AM
GAIL.csv	Microsoft Excel Comma S...	196 KB	No	543 KB	64%	5/1/2021 9:23 AM
GRASIM.csv	Microsoft Excel Comma S...	213 KB	No	599 KB	65%	5/1/2021 9:23 AM
HCLTECH.csv	Microsoft Excel Comma S...	213 KB	No	597 KB	65%	5/1/2021 9:23 AM
HDFC.csv	Microsoft Excel Comma S...	217 KB	No	597 KB	64%	5/1/2021 9:23 AM
HDFCBANK.csv	Microsoft Excel Comma S...	214 KB	No	614 KB	66%	5/1/2021 9:23 AM
HEROMOTOCO.csv	Microsoft Excel Comma S...	216 KB	No	619 KB	66%	5/1/2021 9:23 AM

Once downloaded, extract the zip file.

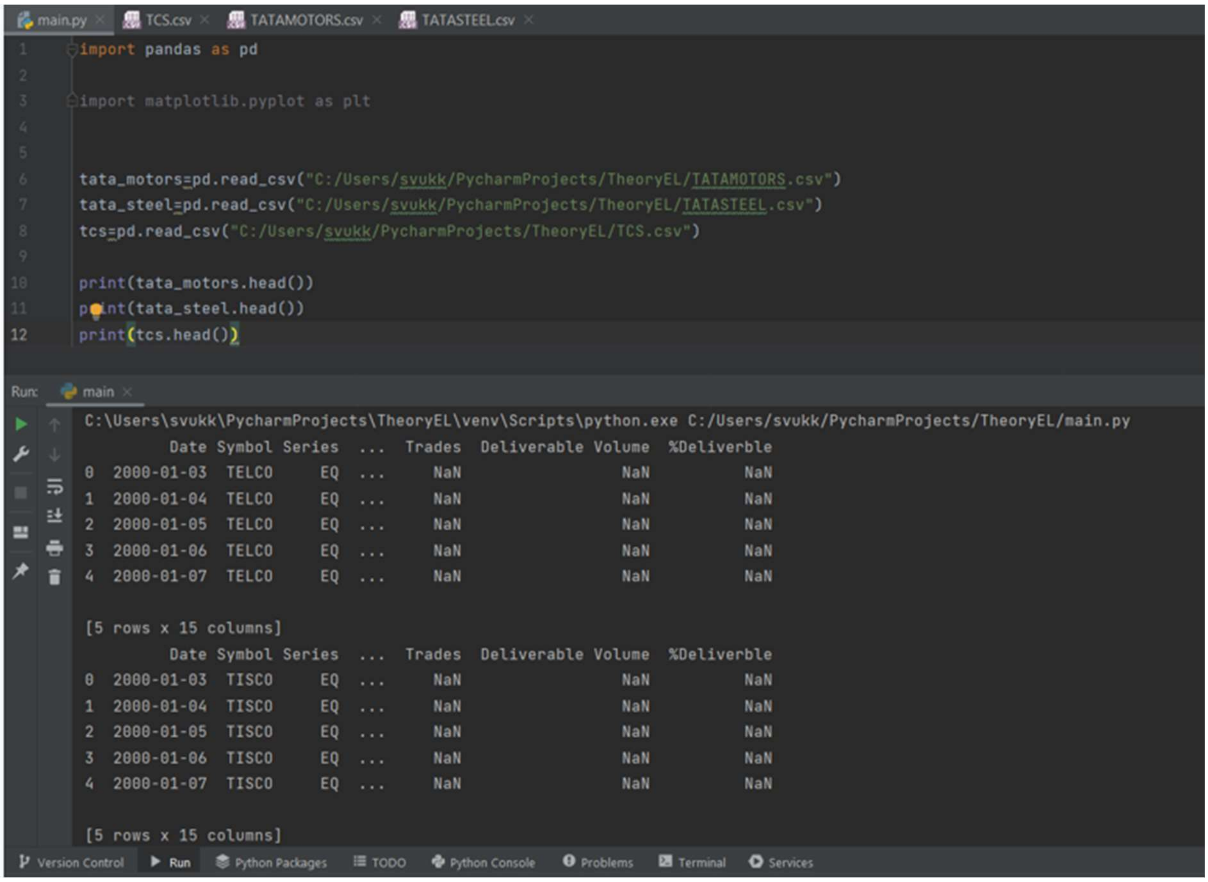
This data set consists of a number of companies’ stock data from 2000-2021 including Adani Ports, Bajaj Finance, Wipro, Infosys, and many more. But for this project, we will be analyzing three Tata stocks – Tata Motors, Tata Steel, and Tata Consultancy Services (TCS).



Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverabl	%Deliverble
1/3/2000	CIPLA	EQ	1349.4	1410	1457.35	1380.05	1457.35	1457.35	1441.36	21060	3.0355E+12			
1/4/2000	CIPLA	EQ	1457.35	1537	1537	1430	1466.05	1465.25	1460.43	30215	4.4127E+12			
1/5/2000	CIPLA	EQ	1465.25	1474	1474	1365	1441	1435.05	1428.11	33799	4.8269E+12			
1/6/2000	CIPLA	EQ	1435.05	1434	1435	1349	1365	1355.85	1390.55	33083	4.6004E+12			
1/7/2000	CIPLA	EQ	1355.85	1370	1389.9	1247.4	1247.4	1247.55	1267.49	66536	8.4334E+12			
1/10/2000	CIPLA	EQ	1247.55	1288	1299	1191	1197.15	1205.9	1222.23	105912	1.2945E+13			
1/11/2000	CIPLA	EQ	1205.9	1225	1225	1109.45	1125	1114.25	1156.31	186975	2.162E+13			
1/12/2000	CIPLA	EQ	1114.25	1185	1203.4	1185	1203.4	1203.4	1202.76	7416	8.9197E+11			
1/13/2000	CIPLA	EQ	1203.4	1299.7	1299.7	1281.2	1299.7	1297.05	1298.53	90379	1.1736E+13			
1/14/2000	CIPLA	EQ	1297.05	1299	1304.55	1220	1275	1280.7	1275.38	70729	9.0206E+12			
1/17/2000	CIPLA	EQ	1280.7	1335	1340	1250.15	1265	1270.05	1292.22	54938	7.0992E+12			
1/18/2000	CIPLA	EQ	1270.05	1294	1294	1200	1235	1220.15	1227.43	51691	6.3447E+12			
1/19/2000	CIPLA	EQ	1220.15	1175	1219.9	1132	1200	1203.85	1189.27	132669	1.5778E+13			
1/20/2000	CIPLA	EQ	1203.85	1205	1223	1201	1208	1208.8	1212.22	44602	5.4068E+12			
1/21/2000	CIPLA	EQ	1208.8	1210	1210	1160	1202	1201.1	1198.65	43168	5.1743E+12			
1/24/2000	CIPLA	EQ	1201.1	1218	1223.9	1185	1212	1212	1210.61	67930	8.2237E+12			
1/25/2000	CIPLA	EQ	1212	1195	1208	1176.05	1197.9	1194.3	1195.43	65851	7.872E+12			
1/27/2000	CIPLA	EQ	1194.3	1225	1225	1185	1195	1190.3	1197.07	33549	4.016E+12			
1/28/2000	CIPLA	EQ	1190.3	1210	1215	1171	1183	1183.75	1188.57	25834	3.0706E+12			
1/31/2000	CIPLA	EQ	1183.75	1162	1184.9	1155	1169.35	1173.1	1172.11	30473	3.5718E+12			
2/1/2000	CIPLA	EQ	1173.1	1185	1232	1160	1225	1219.5	1191.88	57336	6.8338E+12			
2/2/2000	CIPLA	EQ	1219.5	1249	1295	1240	1277	1270.85	1267.86	67545	8.5638E+12			
2/3/2000	CIPLA	EQ	1270.85	1289	1298.95	1235	1240	1243.4	1261.8	29108	3.6728E+12			
2/4/2000	CIPLA	EQ	1243.4	1245	1250	1196.15	1225	1216.15	1214.08	30288	3.6772E+12			
2/7/2000	CIPLA	EQ	1216.15	1220	1245	1195	1210	1220.8	1224.62	79074	9.6836E+12			
2/8/2000	CIPLA	EQ	1220.8	1201	1234.5	1185.25	1205	1205.65	1202.08	67366	8.0979E+12			
2/9/2000	CIPLA	EQ	1205.65	1235	1265	1211	1219	1218.15	1229.63	28941	3.5587E+12			
2/10/2000	CIPLA	EQ	1218.15	1200	1212	1185	1204	1197.6	1196.45	28527	3.4131E+12			
2/11/2000	CIPLA	EQ	1197.6	1200	1200	1150	1165.05	1165.65	1178.37	22600	2.6631E+12			
2/14/2000	CIPLA	EQ	1165.65	1155	1169	1100	1151	1152.45	1152.2	56818	6.5465E+12			
2/15/2000	CIPLA	EQ	1152.45	1131	1159	1120	1155	1151.8	1147.45	36573	4.1966E+12			

The data in the data set of the csv file opened using excel consists of Date, Symbol, Prev Close, Open, High, Low, Last, Close, VWAP, Turnover, Trades, Deliverable Volume, and % Deliverable. We will be utilizing the Date, Open, and Volume.

Data viewing



```
import pandas as pd

import matplotlib.pyplot as plt


tata_motors=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TATAMOTORS.csv")
tata_steel=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TATASTEEL.csv")
tcs=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TCS.csv")

print(tata_motors.head())
print(tata_steel.head())
print(tcs.head())
```

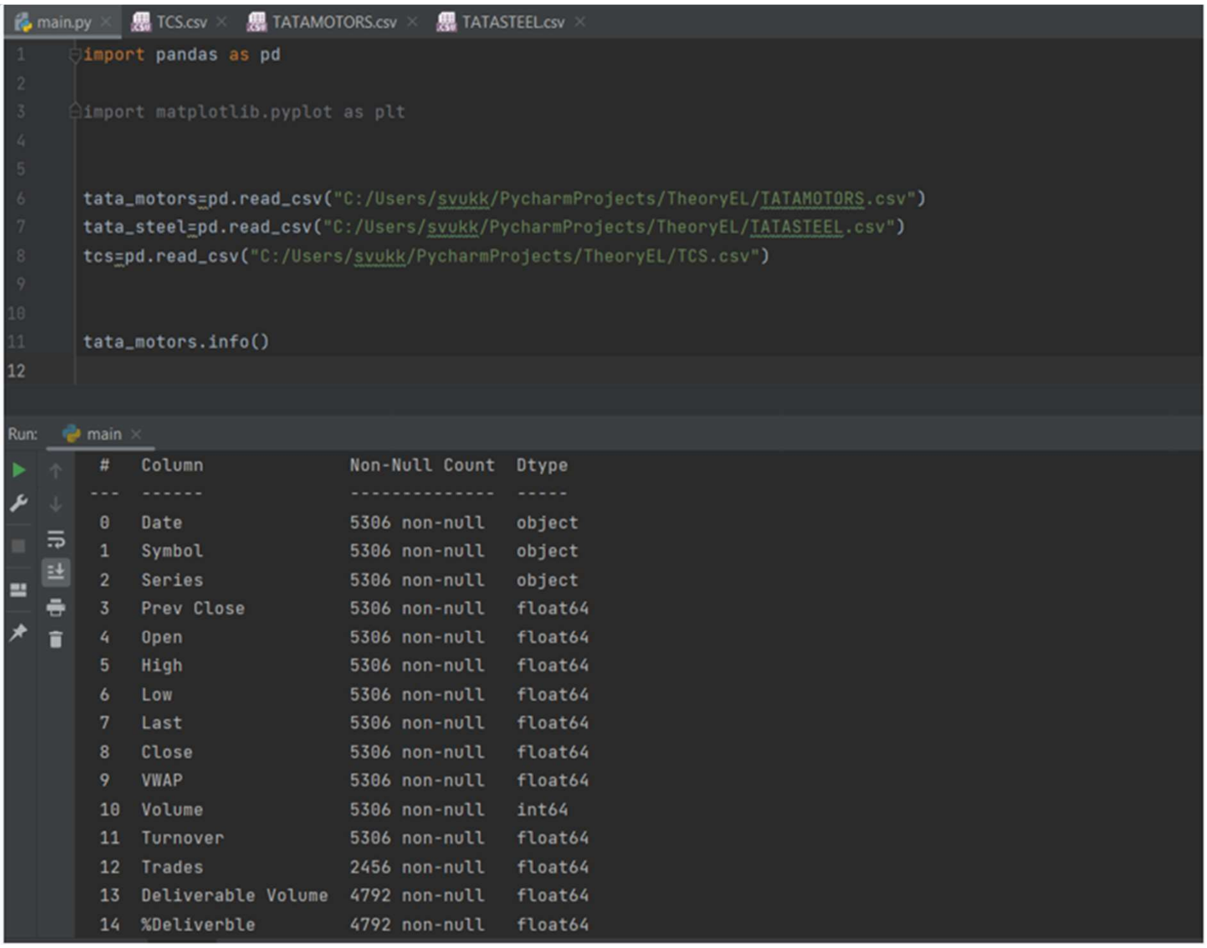
```
C:\Users\svukk\PycharmProjects\TheoryEL\venv\Scripts\python.exe C:/Users/svukk/PycharmProjects/TheoryEL/main.py
         Date Symbol Series  ...  Trades  Deliverable Volume  %Deliverble
0  2000-01-03  TELCO     EQ  ...     NaN                 NaN          NaN
1  2000-01-04  TELCO     EQ  ...     NaN                 NaN          NaN
2  2000-01-05  TELCO     EQ  ...     NaN                 NaN          NaN
3  2000-01-06  TELCO     EQ  ...     NaN                 NaN          NaN
4  2000-01-07  TELCO     EQ  ...     NaN                 NaN          NaN

[5 rows x 15 columns]
         Date Symbol Series  ...  Trades  Deliverable Volume  %Deliverble
0  2000-01-03  TISCO     EQ  ...     NaN                 NaN          NaN
1  2000-01-04  TISCO     EQ  ...     NaN                 NaN          NaN
2  2000-01-05  TISCO     EQ  ...     NaN                 NaN          NaN
3  2000-01-06  TISCO     EQ  ...     NaN                 NaN          NaN
4  2000-01-07  TISCO     EQ  ...     NaN                 NaN          NaN

[5 rows x 15 columns]
```

Here, we can notice the data type of “Date” is an ‘object’ in the Tata Motors dataset, hence we need to convert it into the ‘date’ datatype (Which we will do in the “Working on Data” section).

You will see similar results for the datatypes for Tata Steel and TCS datasets after executing the tata_steel.info() and tcs.info() functions respectively.



```python
import pandas as pd

import matplotlib.pyplot as plt


tata_motors=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TATAMOTORS.csv")
tata_steel=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TATASTEEL.csv")
tcs=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TCS.csv")


tata_motors.info()
```

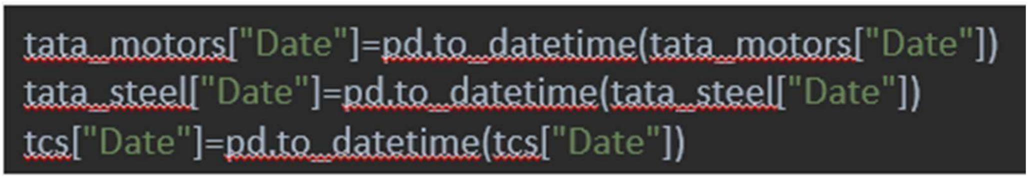
```
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   Date                5306 non-null   object
 1   Symbol              5306 non-null   object
 2   Series              5306 non-null   object
 3   Prev Close          5306 non-null   float64
 4   Open                5306 non-null   float64
 5   High                5306 non-null   float64
 6   Low                 5306 non-null   float64
 7   Last                5306 non-null   float64
 8   Close               5306 non-null   float64
 9   VWAP                5306 non-null   float64
 10  Volume              5306 non-null   int64
 11  Turnover            5306 non-null   float64
 12  Trades              2456 non-null   float64
 13  Deliverable Volume  4792 non-null   float64
 14  %Deliverble         4792 non-null   float64
```

From the above table, we can view the first 5 rows of the Tata Motors dataset and get a brief overview of the data present.You will see the results of the dataset for Tata Steel and TCS by executing the tata_steel.head() and tcs.head() functions respectively.

Working on Data

Converting the “Date” column dtype from object to date

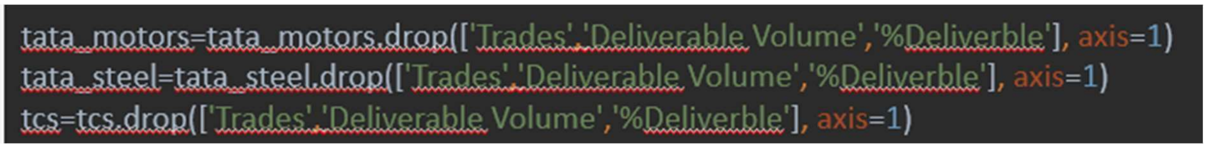
Once this code is executed, if you try executing the .info() function on any of the datasets, you will notice the datatype of the ‘Date’ column changed from ‘object’ to
‘datetime64[ns]’ for all 3 datasets



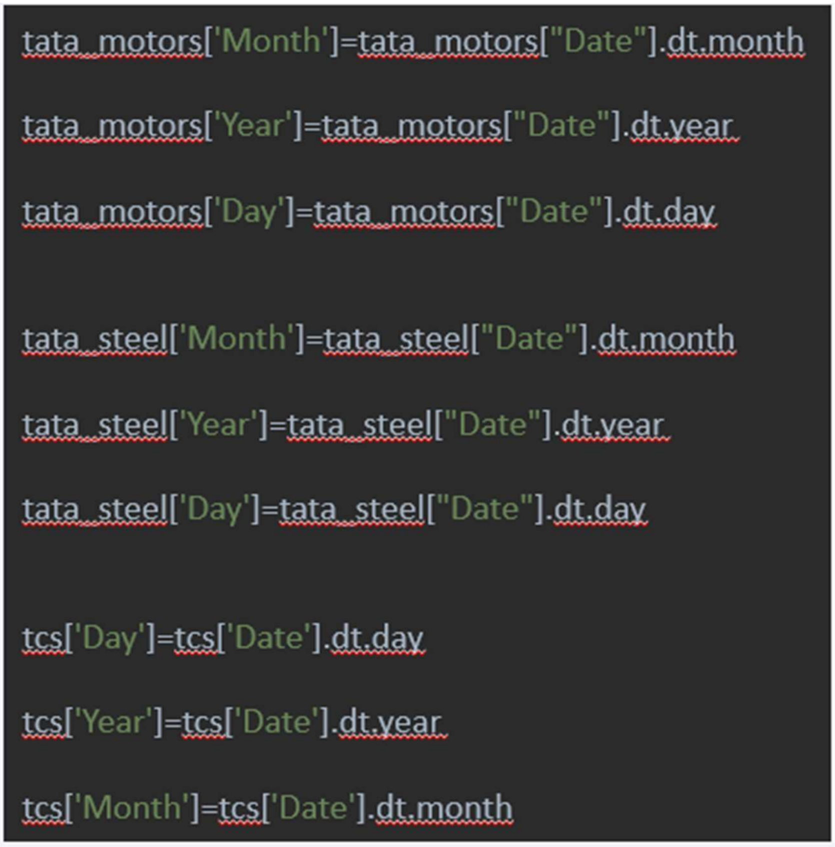
```
tata_motors["Date"]=pd.to_datetime(tata_motors["Date"])
tata_steel["Date"]=pd.to_datetime(tata_steel["Date"])
tcs["Date"]=pd.to_datetime(tcs["Date"])
```

Dropping columns Trades, Deliverable Volume, and %Deliverable

Once this code is executed, if you try running the .head() or .tail() function on any of the datasets, you will notice all the 3 columns Trades, Deliverable Volume, and %Deliverable not present.



```
tata_motors=tata_motors.drop(['Trades','Deliverable Volume','%Deliverble'], axis=1)
tata_steel=tata_steel.drop(['Trades','Deliverable Volume','%Deliverble'], axis=1)
tcs=tcs.drop(['Trades','Deliverable Volume','%Deliverble'], axis=1)
```



```
tata_motors['Month']=tata_motors["Date"].dt.month

tata_motors['Year']=tata_motors["Date"].dt.year

tata_motors['Day']=tata_motors["Date"].dt.day


tata_steel['Month']=tata_steel["Date"].dt.month

tata_steel['Year']=tata_steel["Date"].dt.year

tata_steel['Day']=tata_steel["Date"].dt.day


tcs['Day']=tcs['Date'].dt.day

tcs['Year']=tcs['Date'].dt.year

tcs['Month']=tcs['Date'].dt.month
```

Once this code is executed, if you try running the .head() or .tail() function on any of the datasets, you will notice 3 new columns ‘Day’, ‘Month’ and ‘Year’ present. We will be using the ‘Day’ column for our analysis

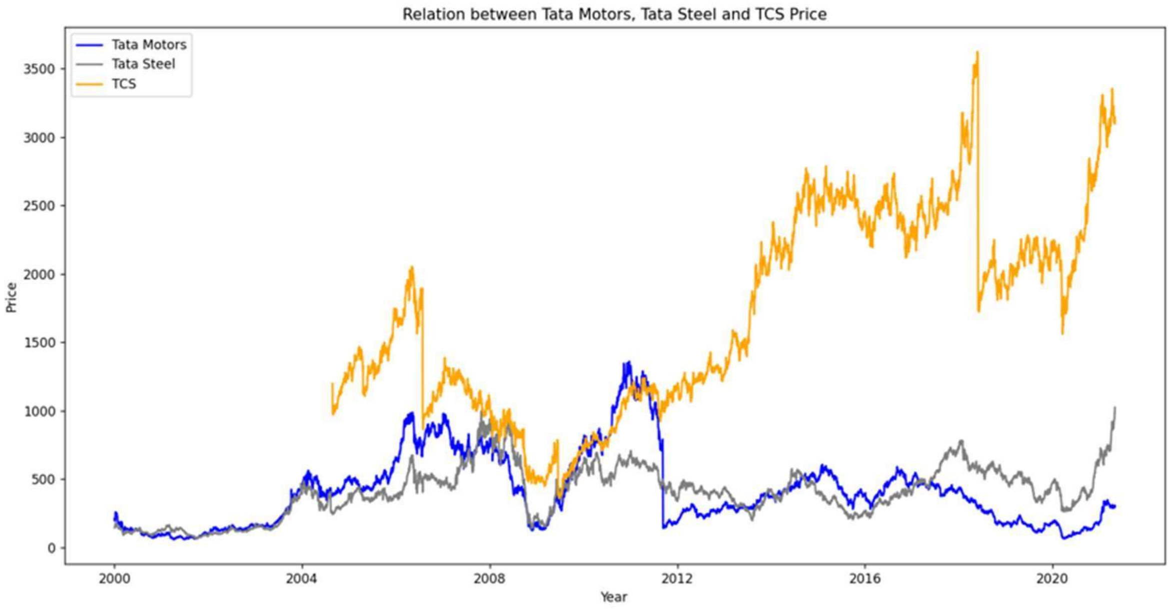
EDA with Data Visualization

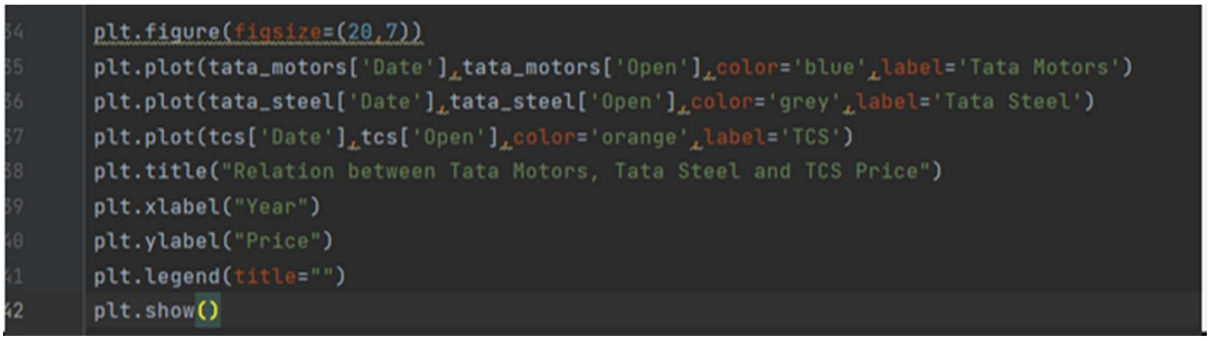
Visualization charts and reason for the choice of the chart

•Bar Graph makes it easy to compare sets of data between different groups immediately. The graph represents categories on one axis and a discrete value in the other. The goal is to show the relationship between the two axes. Bar charts can also show big changes in data over time. Mean vs. Orbit

•Line Graphs are useful in that they show data variables and trends very clearly and can help to make predictions about the results of data not yet recorded

Price Comparision

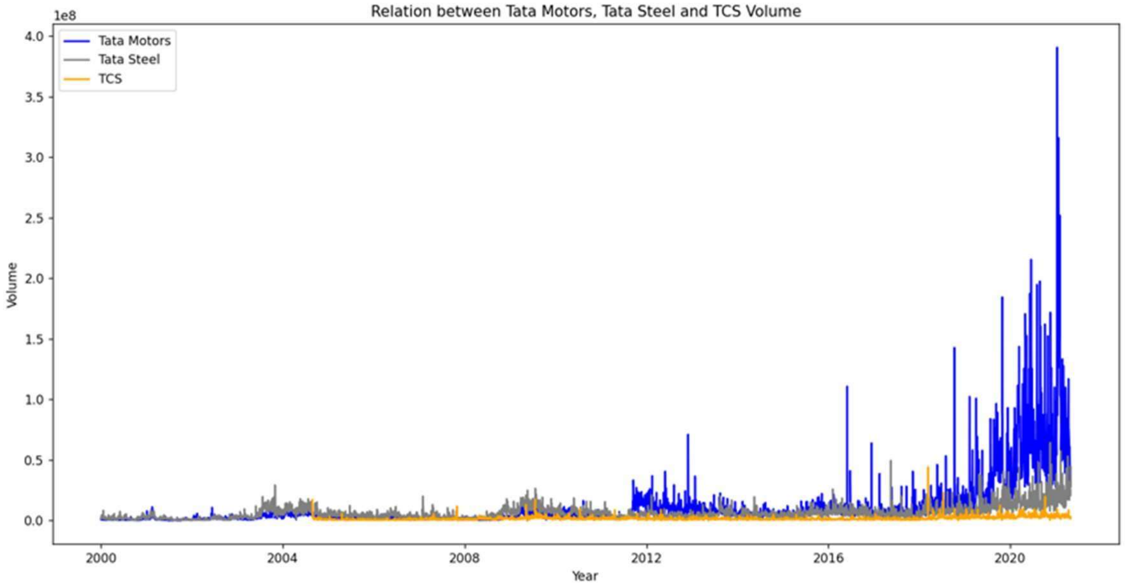


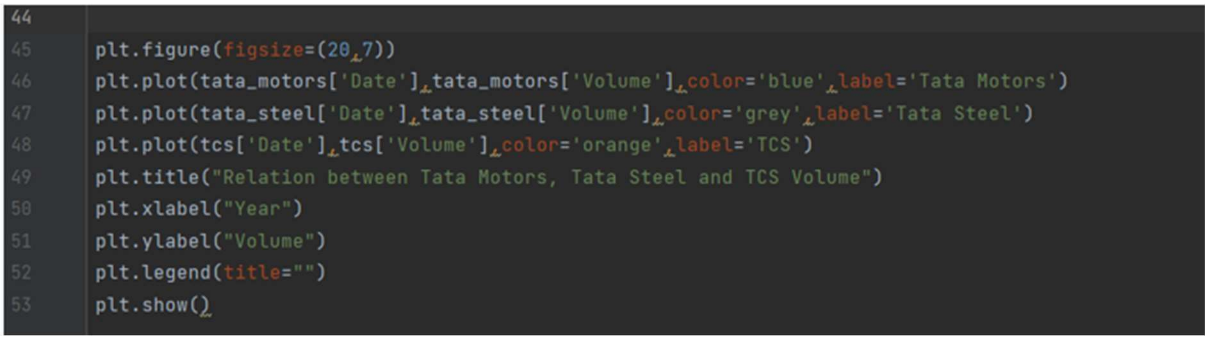


```
plt.figure(figsize=(20,7))
plt.plot(tata_motors['Date'],tata_motors['Open'],color='blue',label='Tata Motors')
plt.plot(tata_steel['Date'],tata_steel['Open'],color='grey',label='Tata Steel')
plt.plot(tcs['Date'],tcs['Open'],color='orange',label='TCS')
plt.title("Relation between Tata Motors, Tata Steel and TCS Price")
plt.xlabel("Year")
plt.ylabel("Price")
plt.legend(title="")
plt.show()
```

The next segment of code creates a plot showing the relationship between the opening price of each company's stock over time. It plots the 'Open' column against the 'Date' column for each company, sets the title and axes labels, and adds a legend for clarity.

Volume Comparision

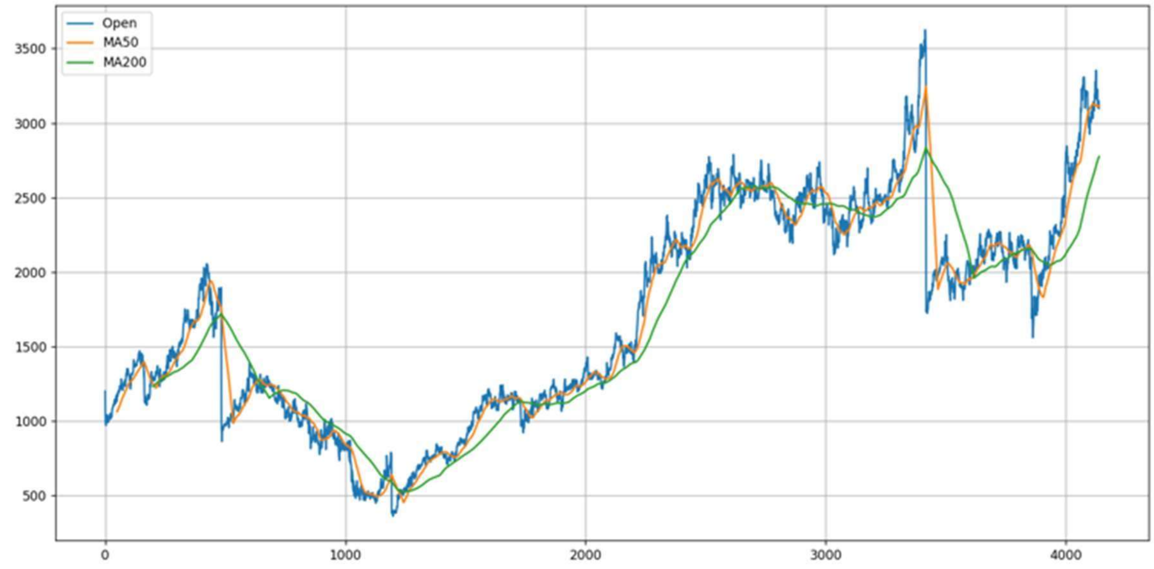


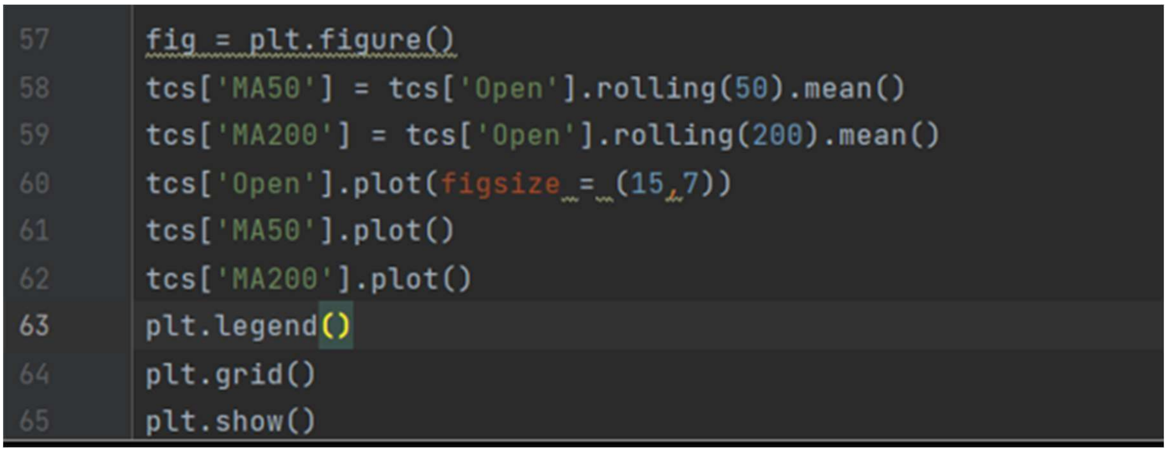


```
plt.figure(figsize=(20,7))
plt.plot(tata_motors['Date'],tata_motors['Volume'],color='blue',label='Tata Motors')
plt.plot(tata_steel['Date'],tata_steel['Volume'],color='grey',label='Tata Steel')
plt.plot(tcs['Date'],tcs['Volume'],color='orange',label='TCS')
plt.title("Relation between Tata Motors, Tata Steel and TCS Volume")
plt.xlabel("Year")
plt.ylabel("Volume")
plt.legend(title="")
plt.show()
```

The next plot shows the relationship between the volume of each company's stock over time. It plots the 'Volume' column against the 'Date' column for each company, sets the title and axes labels, and adds a legend for clarity.

Moving Averages

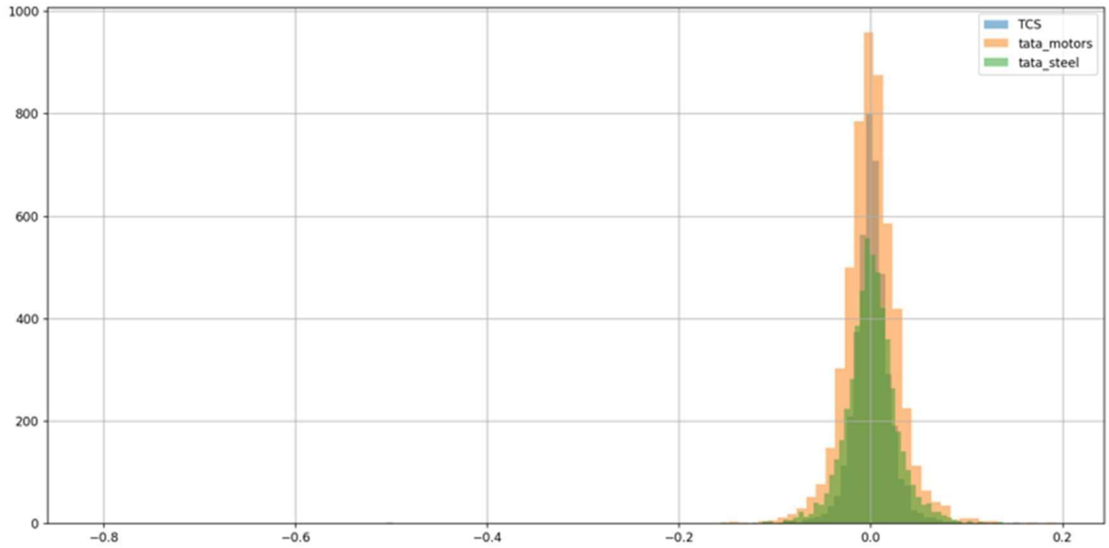


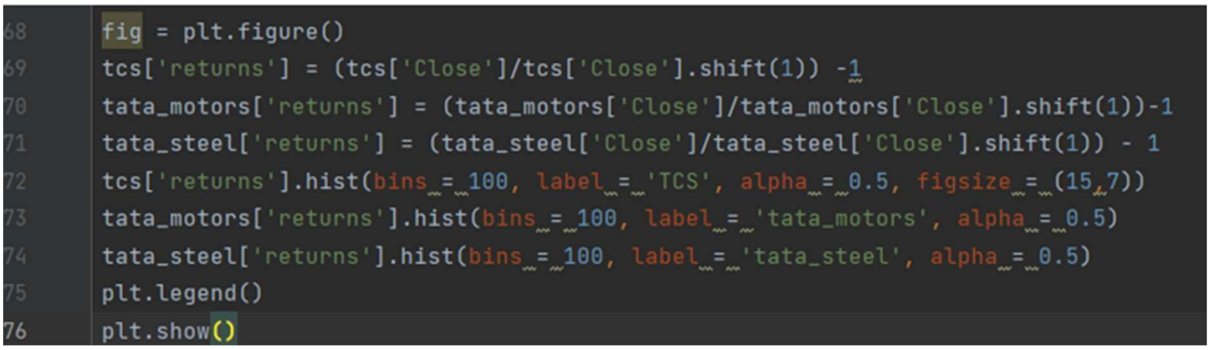


```
fig = plt.figure()
tcs['MA50'] = tcs['Open'].rolling(50).mean()
tcs['MA200'] = tcs['Open'].rolling(200).mean()
tcs['Open'].plot(figsize = (15,7))
tcs['MA50'].plot()
tcs['MA200'].plot()
plt.legend()
plt.grid()
plt.show()
```

The following segment of code creates a plot showing the rolling average of TCS's opening stock price over time. It calculates the rolling average using the 'rolling' function and plots the 'Open', 'MA50', and 'MA200' columns against the 'Date' column, sets the title, and adds a legend and gridlines for clarity.

Volatility Analysis





```
fig = plt.figure()
tcs['returns'] = (tcs['Close']/tcs['Close'].shift(1)) -1
tata_motors['returns'] = (tata_motors['Close']/tata_motors['Close'].shift(1))-1
tata_steel['returns'] = (tata_steel['Close']/tata_steel['Close'].shift(1)) - 1
tcs['returns'].hist(bins = 100, label = 'TCS', alpha = 0.5, figsize = (15,7))
tata_motors['returns'].hist(bins = 100, label = 'tata_motors', alpha = 0.5)
tata_steel['returns'].hist(bins = 100, label = 'tata_steel', alpha = 0.5)
plt.legend()
plt.show()
```

The final segment of code creates a histogram showing the distribution of returns for each company's stock. It calculates the daily returns for each company by dividing the 'Close' column by the previous day's 'Close' value and subtracting 1. It then creates a histogram with 100 bins for each company's returns, sets the title, adds a legend, and displays the plot.

Code

```
import pandas as pd

import matplotlib.pyplot as plt

tata_motors=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TA
TAMOTORS.csv")
tata_steel=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TAT ASTEEL.csv")
tcs=pd.read_csv("C:/Users/svukk/PycharmProjects/TheoryEL/TCS.csv")



tata_motors["Date"]=pd.to_datetime(tata_motors["Date"])
tata_steel["Date"]=pd.to_datetime(tata_steel["Date"])
tcs["Date"]=pd.to_datetime(tcs["Date"])

tata_motors=tata_motors.drop(['Trades','Deliverable Volume','%Deliverble'], axis=1)
tata_steel=tata_steel.drop(['Trades','Deliverable Volume','%Deliverble'], axis=1)
tcs=tcs.drop(['Trades','Deliverable Volume','%Deliverble'], axis=1)

tata_motors['Month']=tata_motors["Date"].dt.month
tata_motors['Year']=tata_motors["Date"].dt.year
tata_motors['Day']=tata_motors["Date"].dt.day

tata_steel['Month']=tata_steel["Date"].dt.month tata_steel['Year']=tata_steel["Date"].dt.year
```

```
tata_steel['Day']=tata_steel["Date"].dt.day

tcs['Day']=tcs['Date'].dt.day tcs['Year']=tcs['Date'].dt.year
tcs['Month']=tcs['Date'].dt.month

plt.figure(figsize=(20,7))
plt.plot(tata_motors['Date'],tata_motors['Open'],color='blue',label='Tata Motors')
plt.plot(tata_steel['Date'],tata_steel['Open'],color='grey',label='Tata Steel')
plt.plot(tcs['Date'],tcs['Open'],color='orange',label='TCS') plt.title("Relation
between Tata Motors, Tata Steel and TCS Price") plt.xlabel("Year")
plt.ylabel("Price") plt.legend(title="")
#plt.show()


plt.figure(figsize=(20,7))
plt.plot(tata_motors['Date'],tata_motors['Volume'],color='blue',label='Tata Motors')
plt.plot(tata_steel['Date'],tata_steel['Volume'],color='grey',label='Tata Steel')
plt.plot(tcs['Date'],tcs['Volume'],color='orange',label='TCS') plt.title("Relation between
Tata Motors, Tata Steel and TCS Volume") plt.xlabel("Year") plt.ylabel("Volume")
plt.legend(title="")
#plt.show()
```

```
fig = plt.figure() tcs['MA50'] =
tcs['Open'].rolling(50).mean() tcs['MA200'] =
tcs['Open'].rolling(200).mean()
tcs['Open'].plot(figsize = (15,7))
tcs['MA50'].plot()
tcs['MA200'].plot() plt.legend()
plt.grid()
plt.show()



fig = plt.figure()
tcs['returns'] = (tcs['Close']/tcs['Close'].shift(1)) -1 tata_motors['returns'] =
(tata_motors['Close']/tata_motors['Close'].shift(1))-1 tata_steel['returns'] =
(tata_steel['Close']/tata_steel['Close'].shift(1)) - 1 tcs['returns'].hist(bins = 100, label =
'TCS', alpha = 0.5, figsize = (15,7)) tata_motors['returns'].hist(bins = 100, label =
'tata_motors', alpha = 0.5) tata_steel['returns'].hist(bins = 100, label = 'tata_steel', alpha
= 0.5) plt.legend() plt.show()
```