MUKESH PATEL SCHOOL OF TECHNOLOGY MANAGEMENT AND ENGINEERING

(Affiliated to NMIMS Deemed to be University, Mumbai)

Data Extraction and Processing

Project Report

on

"Stock Price Comparison of Tata Motors and Tata Steel"

Submitted by:

Group 3		
Name	**Roll No.**	**SAP ID**
Armaan Shah	C037	70322000107
Siddhi Pandya	C052	70322000057
Khushl Vora	C058	70322000108
Lakshay Dahiya	C063	70322000114
Semester/Year: VII/IV		
Division: B		
Batch: B2		

B. Tech Integrated Program

Department of Computer Science Engineering

MPSTME, Mumbai

About the Dataset

Dataset Title: Stock price comparison of Tata Motors and Tata Steel

URL for Dataset Download:

-*https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data?select=TATASTEEL.csv*
- *https://www.kaggle.com/datasets/rohanrao/nifty50-stock-market-data?select=TATAMOTORS.csv*

We have taken 2 datasets pertaining to the stock of tatasteel and tatamotors from the nifty-50 stock market data (2000-2021) on Kaggle. we have performed analysis on the vast data to understand how different attributes affect the stock prices and how tatasteel stock could affect tatamotors stock. This dataset contains the open, high, low, close and last traded prices for a day. It also contains the number of trades made, the volume of the trades and the turnover on the trades.
The various columns present are:
Date – contains the data.
Symbol – name of the company
Series – type of security
Prev close – previous day's close price
Open – open price of the day
High - Highest price in day
Low – Lowest price in a day
Last – Lat traded price in a day
Close – Close price of the day
VWAP – Volume weighted average price
Volume – Number of shares bought and sold
Turnover – Profit
Trades – Number of trades
Deliverable Volume - Amount of Deliverable volume
%Deliverable – percentage of deliverable volume

DATA EXPLORATION

Imported the NumPy library, a popular package for numerical computing in Python.
Imported the Pandas library, which is used for data manipulation and analysis.
Importing the matplotlib library, a widely used plotting library in Python
Importing the plotly.express module, a part of the Plotly library used for creating interactive visualizations.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import plotly.express as px
```

Importing the tata motors data set

```
df = pd.read_csv('/content/TATAMOTORS.csv')
```

Importing the tata steel data set

```
df2 = pd.read_csv('/content/TATASTEEL.csv')
```

Displaying the Frame

```
df
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	2000-01-03	TELCO	EQ	201.60	207.4	217.25	207.40	217.00	216.75	214.28	676126	1.448775e+13	NaN	NaN	NaN
1	2000-01-04	TELCO	EQ	216.75	217.0	219.00	206.00	211.90	208.20	209.50	679215	1.422962e+13	NaN	NaN	NaN
2	2000-01-05	TELCO	EQ	208.20	194.0	217.80	194.00	213.10	213.25	210.33	1120951	2.357684e+13	NaN	NaN	NaN
3	2000-01-06	TELCO	EQ	213.25	215.0	229.90	215.00	222.00	222.10	225.29	1968998	4.435932e+13	NaN	NaN	NaN
4	2000-01-07	TELCO	EQ	222.10	224.0	239.90	223.10	239.90	239.90	236.32	2199431	5.197636e+13	NaN	NaN	NaN
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
5301	2021-04-26	TATAMOTORS	EQ	294.00	297.0	299.00	294.60	295.85	295.40	296.48	30583139	9.067386e+14	173282.0	3291167.0	0.1076
5302	2021-04-27	TATAMOTORS	EQ	295.40	295.7	302.50	295.10	302.10	301.50	299.05	35040532	1.047879e+15	191627.0	5215100.0	0.1488
5303	2021-04-28	TATAMOTORS	EQ	301.50	303.5	309.50	303.00	305.15	305.90	307.22	44668126	1.372278e+15	264159.0	5116568.0	0.1145
5304	2021-04-29	TATAMOTORS	EQ	305.90	308.9	310.00	301.25	302.20	301.90	304.77	36647292	1.116900e+15	186996.0	4193346.0	0.1144
5305	2021-04-30	TATAMOTORS	EQ	301.90	298.2	301.30	292.55	293.10	293.85	297.01	36121668	1.072849e+15	227806.0	7898988.0	0.2187

5306 rows × 15 columns

Displaying the first 5

```
[ ] df.head()
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover	Trades	Deliverable Volume	%Deliverble
0	2000-01-03	TELCO	EQ	201.60	207.4	217.25	207.4	217.0	216.75	214.28	676126	1.448775e+13	NaN	NaN	NaN
1	2000-01-04	TELCO	EQ	216.75	217.0	219.00	206.0	211.9	208.20	209.50	679215	1.422962e+13	NaN	NaN	NaN
2	2000-01-05	TELCO	EQ	208.20	194.0	217.80	194.0	213.1	213.25	210.33	1120951	2.357684e+13	NaN	NaN	NaN
3	2000-01-06	TELCO	EQ	213.25	215.0	229.90	215.0	222.0	222.10	225.29	1968998	4.435932e+13	NaN	NaN	NaN
4	2000-01-07	TELCO	EQ	222.10	224.0	239.90	223.1	239.9	239.90	236.32	2199431	5.197636e+13	NaN	NaN	NaN

Displaying the information of the dataset

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5306 entries, 0 to 5305
Data columns (total 15 columns):
 #   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
dtypes: float64(11), int64(1), object(3)
memory usage: 621.9+ KB
```

Max and Min date

```
print(df.Date.min())
print(df.Date.max())
```

```
2000-01-03
2021-04-30
```

Datatypes

```
df.dtypes
```

```
Date              object
Symbol            object
Series            object
Prev Close       float64
Open             float64
High             float64
Low              float64
Last             float64
Close            float64
VWAP             float64
Volume             int64
Turnover         float64
dtype: object
```

DATA PRE-PROCESSING

Finding null values

```
[ ] df.isna().sum()
```

```
Date                    0
Symbol                  0
Series                  0
Prev Close              0
Open                    0
High                    0
Low                     0
Last                    0
Close                   0
VWAP                    0
Volume                  0
Turnover                0
Trades               2850
Deliverable Volume    514
%Deliverble           514
dtype: int64
```

Removing Trades column

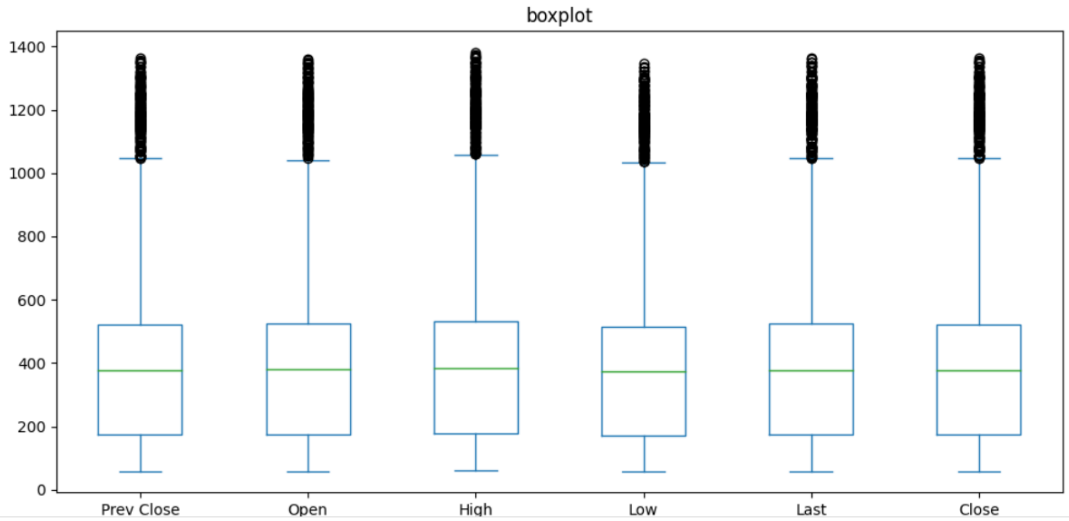
```
[ ] df.drop(['Trades','Deliverable Volume','%Deliverble'],axis=1,inplace=True)
```

```
df
```

	Date	Symbol	Series	Prev Close	Open	High	Low	Last	Close	VWAP	Volume	Turnover
0	2000-01-03	TELCO	EQ	201.60	207.4	217.25	207.40	217.00	216.75	214.28	676126	1.448775e+13
1	2000-01-04	TELCO	EQ	216.75	217.0	219.00	206.00	211.90	208.20	209.50	679215	1.422962e+13
2	2000-01-05	TELCO	EQ	208.20	194.0	217.80	194.00	213.10	213.25	210.33	1120951	2.357684e+13
3	2000-01-06	TELCO	EQ	213.25	215.0	229.90	215.00	222.00	222.10	225.29	1968998	4.435932e+13
4	2000-01-07	TELCO	EQ	222.10	224.0	239.90	223.10	239.90	239.90	236.32	2199431	5.197636e+13
...	...	...	...	...	...	...	...	...	...	...	...	...
5301	2021-04-26	TATAMOTORS	EQ	294.00	297.0	299.00	294.60	295.85	295.40	296.48	30583139	9.067386e+14
5302	2021-04-27	TATAMOTORS	EQ	295.40	295.7	302.50	295.10	302.10	301.50	299.05	35040532	1.047879e+15
5303	2021-04-28	TATAMOTORS	EQ	301.50	303.5	309.50	303.00	305.15	305.90	307.22	44668126	1.372278e+15
5304	2021-04-29	TATAMOTORS	EQ	305.90	308.9	310.00	301.25	302.20	301.90	304.77	36647292	1.116900e+15
5305	2021-04-30	TATAMOTORS	EQ	301.90	298.2	301.30	292.55	293.10	293.85	297.01	36121668	1.072849e+15

5306 rows × 12 columns

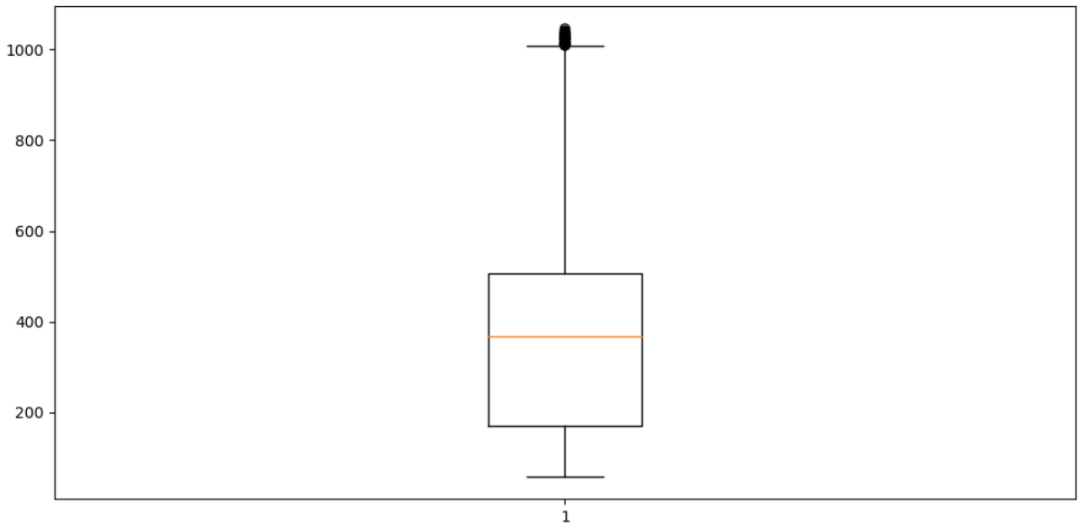
Finding Outliers



```
Q1 = df['Prev Close'].quantile(0.25)
Q3 = df['Prev Close'].quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5*IQR
upper = Q3 + 1.5*IQR


df.drop(df[(df['Prev Close'] < lower) | (df['Prev Close'] > upper)].index, inplace = True)

plt.boxplot(df['Prev Close'])
```

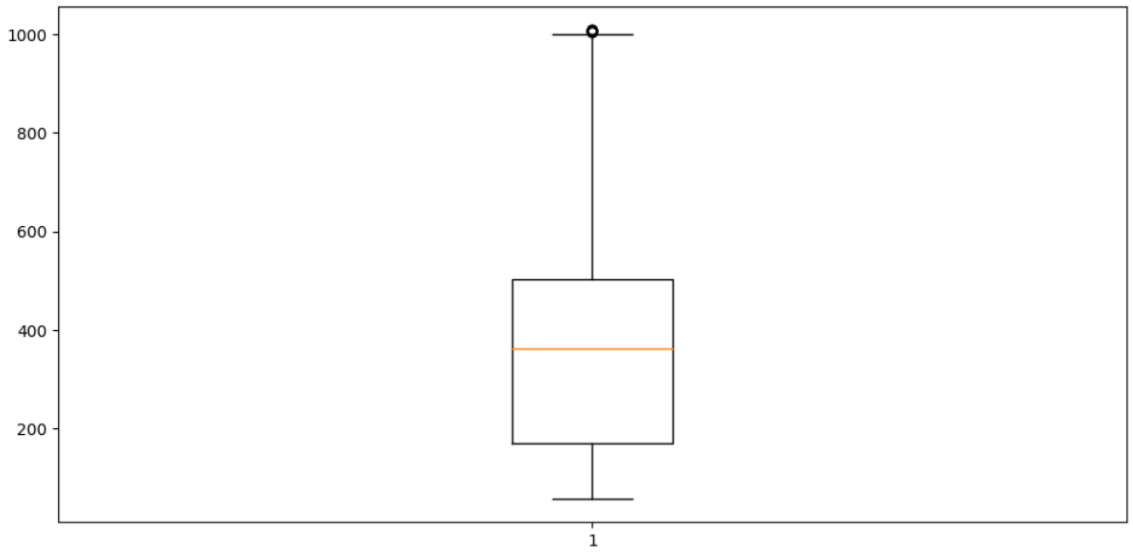
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d6e8d41b1c0>,
  <matplotlib.lines.Line2D at 0x7d6e8d41b460>],
 'caps': [<matplotlib.lines.Line2D at 0x7d6e8d41b700>,
  <matplotlib.lines.Line2D at 0x7d6e8d41b9a0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d6e8d41af20>],
 'medians': [<matplotlib.lines.Line2D at 0x7d6e8d41bc40>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d6e8d41bee0>],
 'means': []}
```

]



```
Q1 = df['Open'].quantile(0.25)
Q3 = df['Open'].quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5*IQR
upper = Q3 + 1.5*IQR


df.drop(df[(df['Open'] < lower) | (df['Open'] > upper)].index, inplace = True)

plt.boxplot(df['Open'])
```

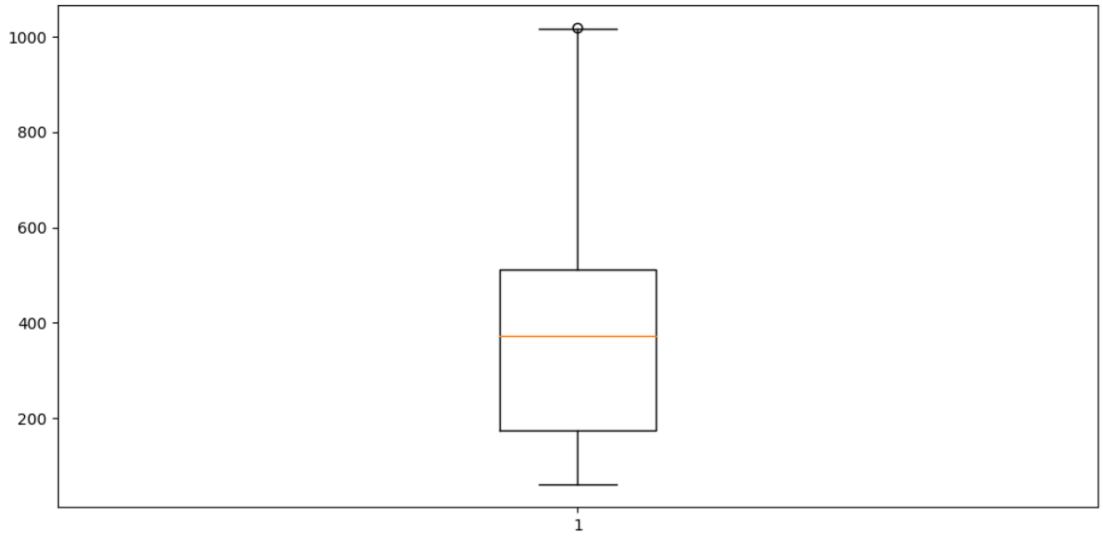
```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d6e8d4a13c0>,
  <matplotlib.lines.Line2D at 0x7d6e8d4a1660>],
 'caps': [<matplotlib.lines.Line2D at 0x7d6e8d4a1900>,
  <matplotlib.lines.Line2D at 0x7d6e8d4a1ba0>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d6e8d4a1240>],
 'medians': [<matplotlib.lines.Line2D at 0x7d6e8d4a1e40>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d6e8d4a20e0>],
 'means': []}
```



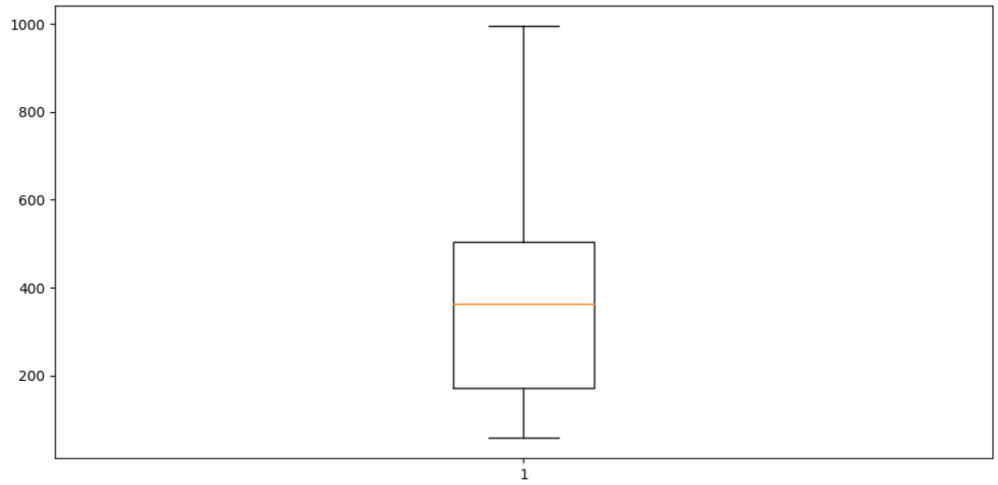

```
Q1 = df['High'].quantile(0.25)
Q3 = df['High'].quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5*IQR
upper = Q3 + 1.5*IQR


df.drop(df[(df['High'] < lower) | (df['High'] > upper)].index, inplace = True)

plt.boxplot(df['High'])
```

```
{'whiskers': [<matplotlib.lines.Line2D at 0x7d6e8d2f3910>,
  <matplotlib.lines.Line2D at 0x7d6e8d2f3bb0>],
 'caps': [<matplotlib.lines.Line2D at 0x7d6e8d2f3d60>,
  <matplotlib.lines.Line2D at 0x7d6e8d324040>],
 'boxes': [<matplotlib.lines.Line2D at 0x7d6e8d2f3670>],
 'medians': [<matplotlib.lines.Line2D at 0x7d6e8d3242e0>],
 'fliers': [<matplotlib.lines.Line2D at 0x7d6e8d324580>],
 'means': []}
```

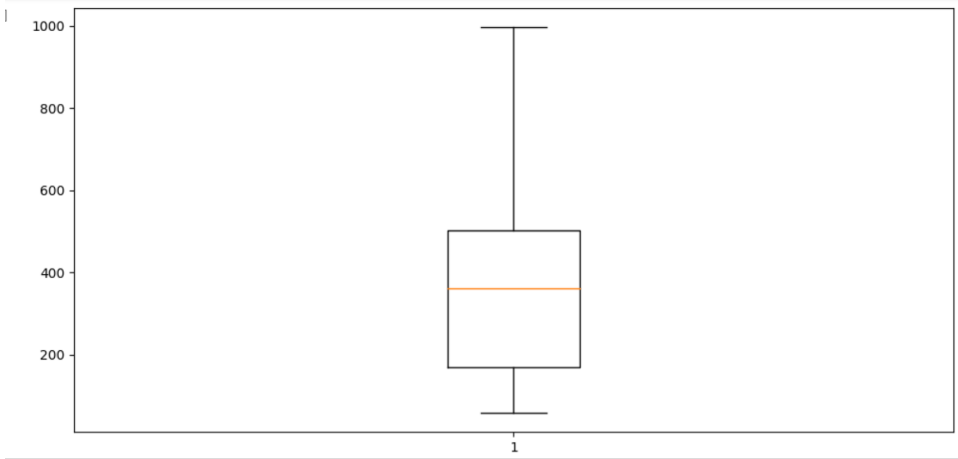
```
'means': []}
```



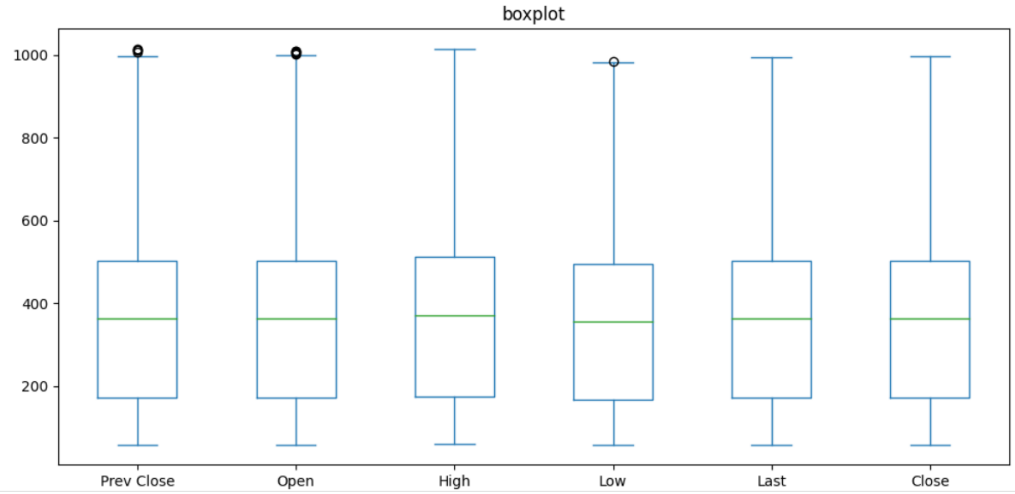

```
Q1 = df['Low'].quantile(0.25)
Q3 = df['Low'].quantile(0.75)
IQR = Q3 - Q1
lower = Q1 - 1.5*IQR
upper = Q3 + 1.5*IQR


df.drop(df[(df['Low'] < lower) | (df['Low'] > upper)].index, inplace = True)

plt.boxplot(df['Low'])
```




```
[ ]  Q1 = df['Close'].quantile(0.25)
     Q3 = df['Close'].quantile(0.75)
     IQR = Q3 - Q1
     lower = Q1 - 1.5*IQR
     upper = Q3 + 1.5*IQR


     df.drop(df[(df['Close'] < lower) | (df['Close'] > upper)].index, inplace = True)

     plt.boxplot(df['Close'])

     from google.colab import files
     df.to_csv('correct_one.csv')
     files.download('correct_one.csv')
```




```
[ ]  plt.rcParams["figure.figsize"] = [10, 5]
     plt.rcParams["figure.autolayout"] = True
     ax = df[['Prev Close', 'Open', 'High', 'Low', 'Last', 'Close']].plot(kind='box', title='boxplot')
     plt.show()
```

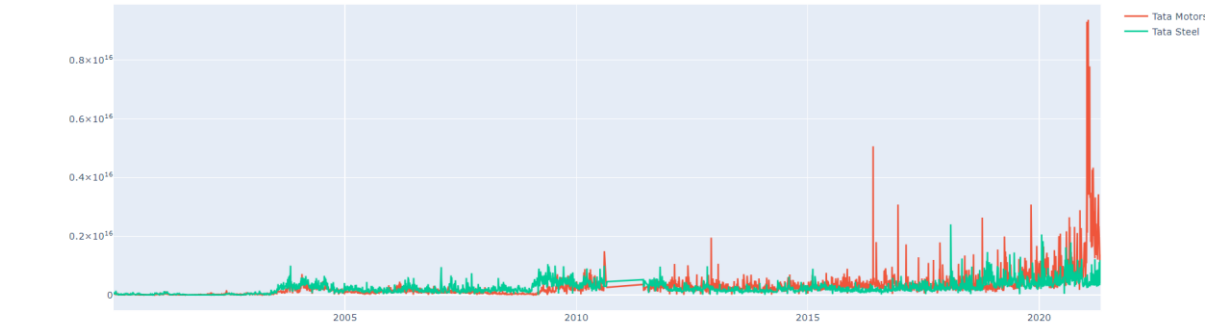


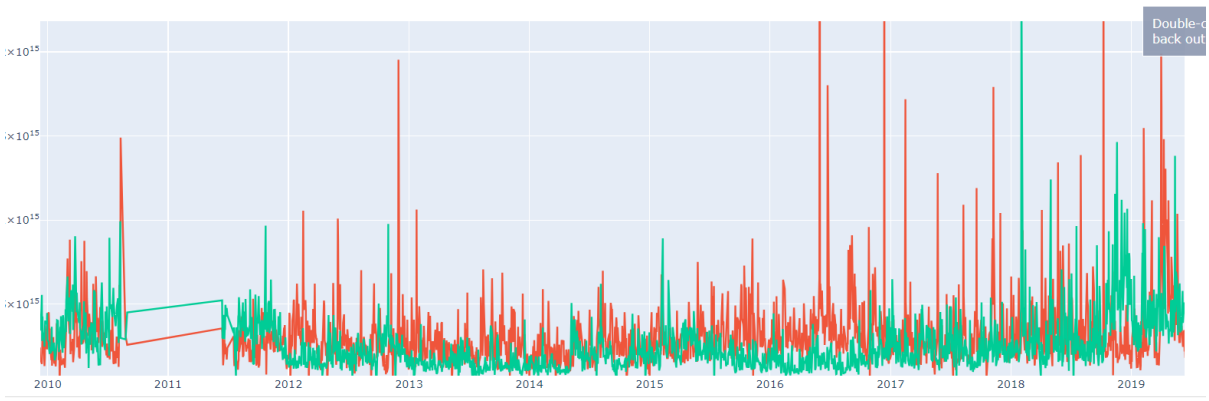
DATA VISUALIZATION

Bar Plot of Tata Motors

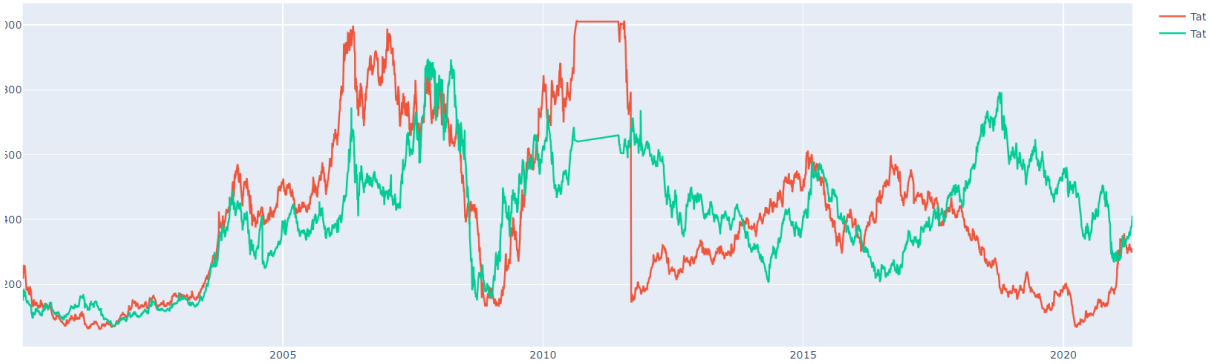
Comparison of tata motors and tata steel based on turnover

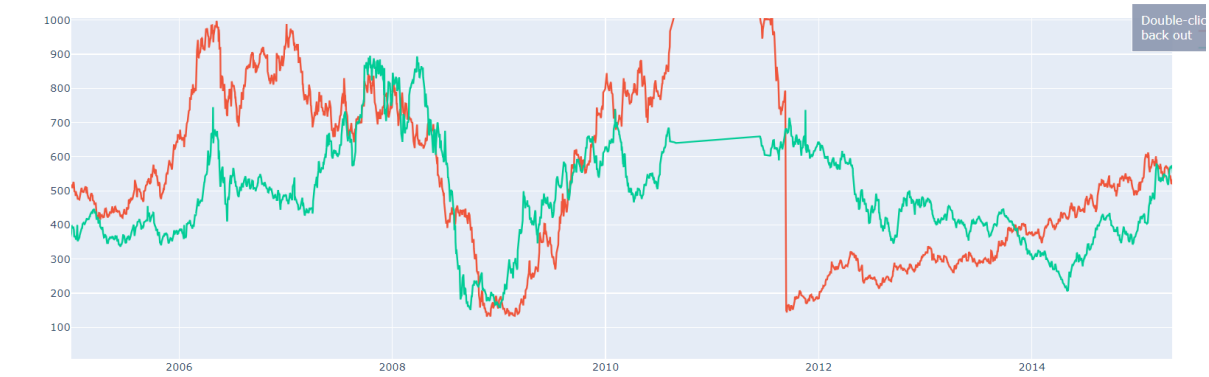
```
fig.add_scatter(x=df['Date'], y=df['Turnover'], name = "Tata Motors")
fig.add_scatter(x=df['Date'], y=df2['Turnover'], name = "Tata Steel")

fig.show()
```



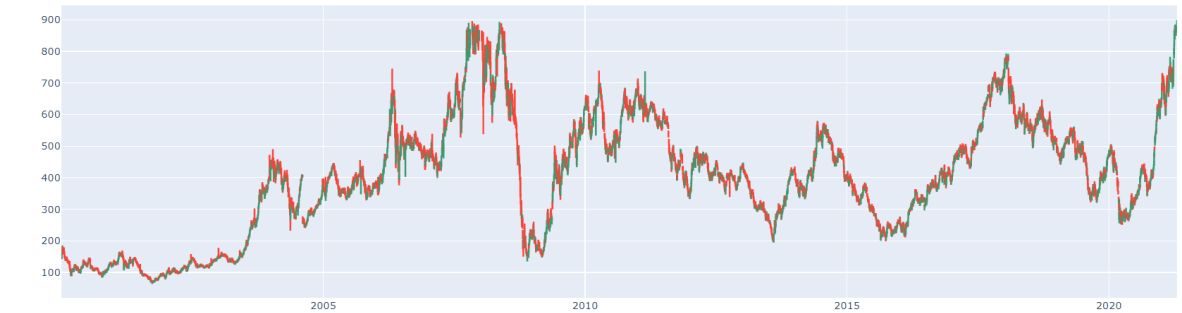


Comparison of tata motor and tata steel based on all time high



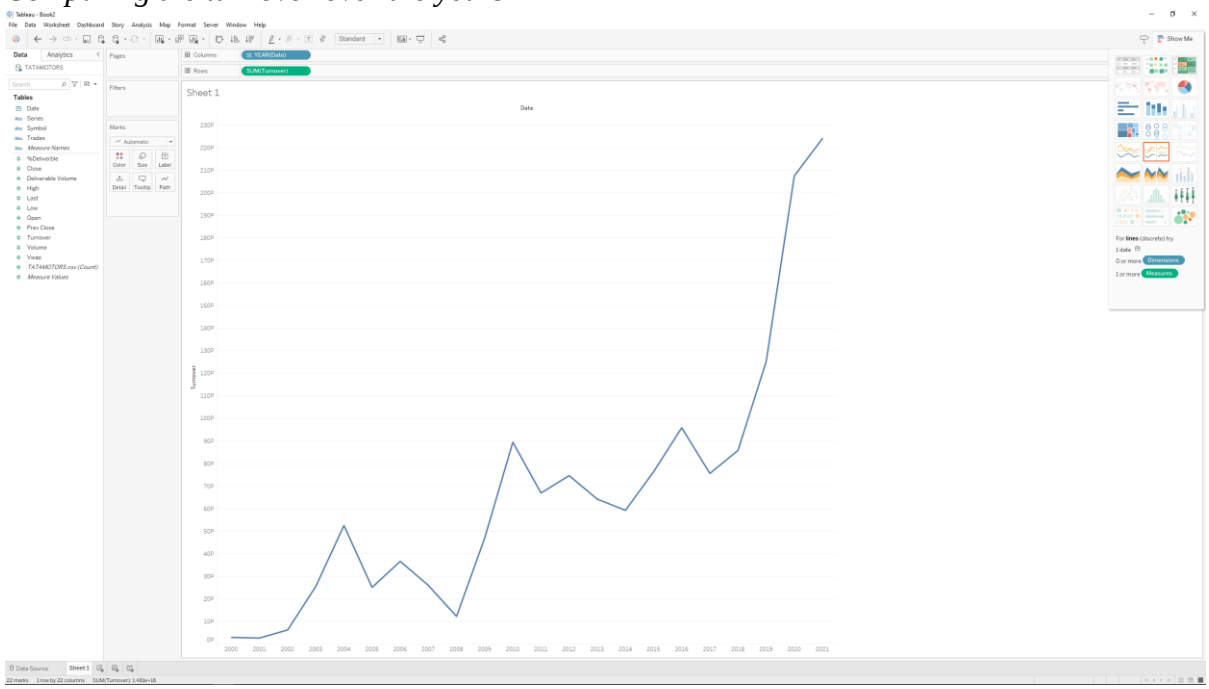


Tata Steel Candle Stick graph
Includes open, high, low, close



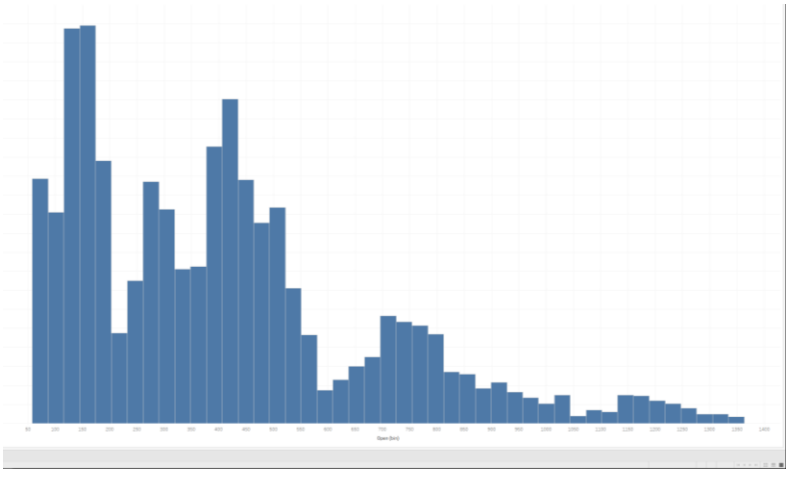
VISUALIZATION USING TABLEAU

Comparing the turnover over the years

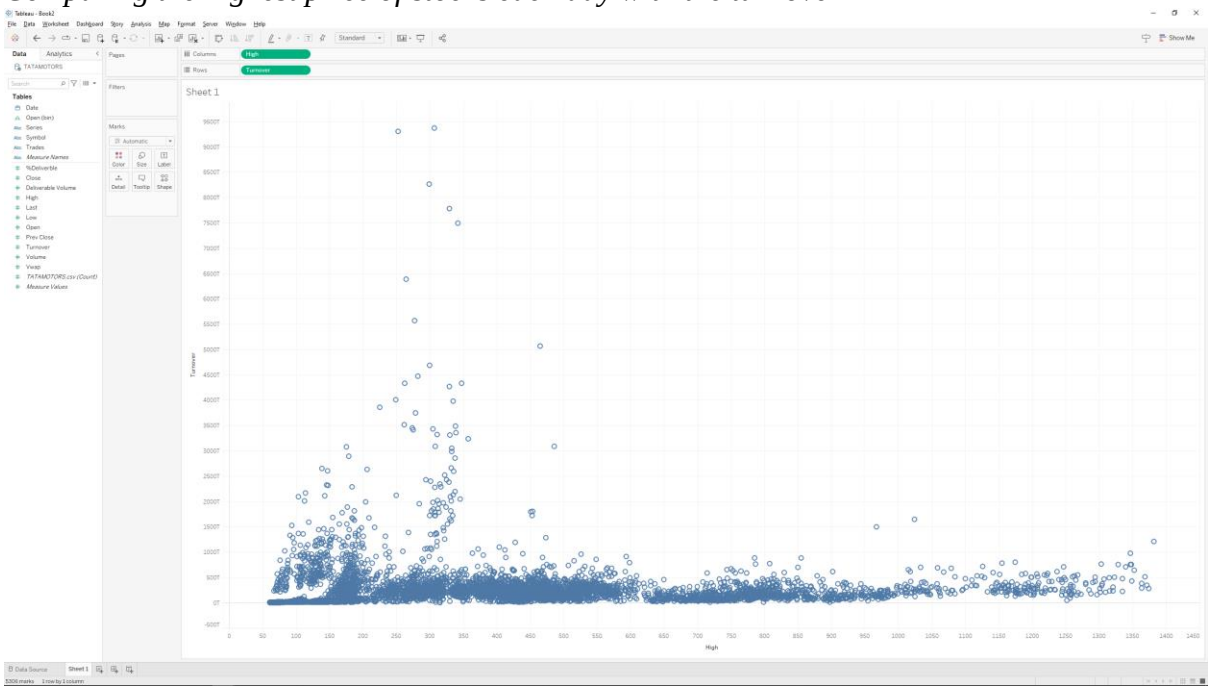


We can see that there is a overall growth in the turnover generated by Tata Motors over the years with a significant increase in the last couple of years

Histogram of the prices at which the stocks opened over the years



Comparing the highest price of stocks each day with the turnover

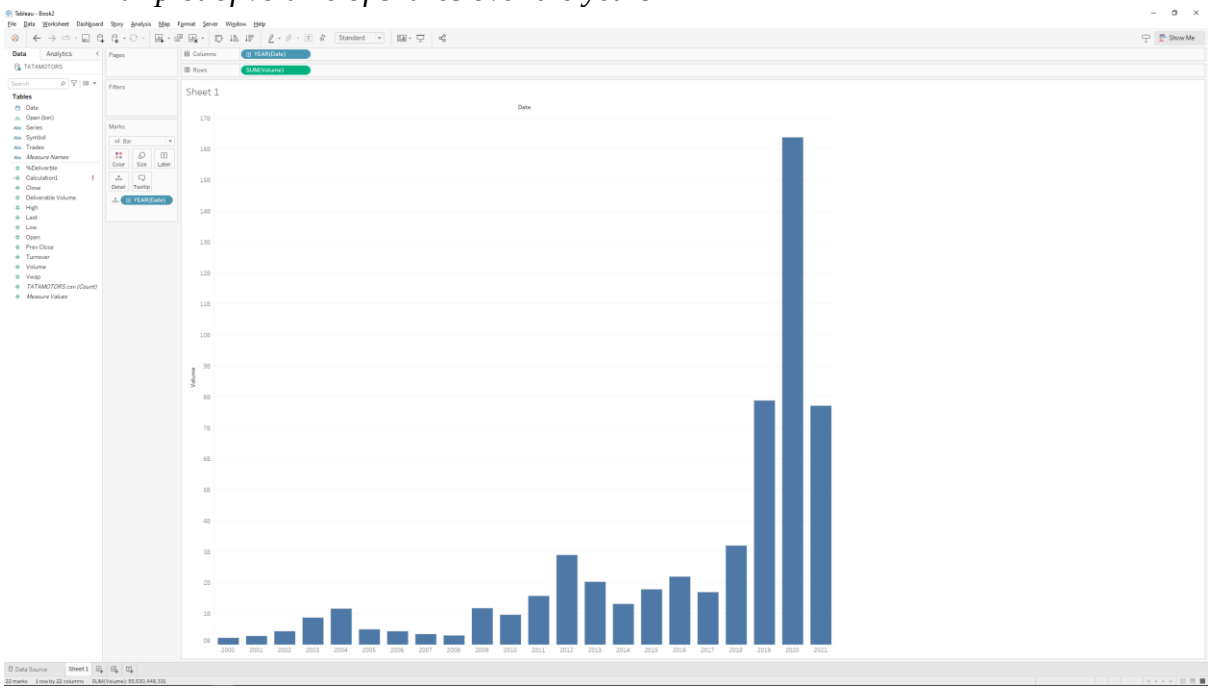


We can see that the highest price that the stock reaches everyday doesn't really affect the turnover.

Comparing the deliverable volume with the turnover

We can see that the turnover is positively correlated with the deliverable volume except a few outliers.

Bar plot of volume of shares over the years



CONCLUSION

Based on the data exploration, cleaning, and visualization of the share price data for Tata Motors and Tata Steel over the last 20 years, the following conclusions can be drawn:

- Trends Over 20 Years: A visual analysis of the share price data indicated that both Tata Motors and Tata Steel experienced fluctuations in their share prices over the 20-year period. There were notable periods of growth, decline, and stability in the stock prices for both companies.

- Correlation Analysis: Further analysis revealed a potential correlation between the share prices of Tata Motors and Tata Steel, suggesting a possible interdependence or similar market influences affecting both companies within the same industry.

- Outliers and Anomalies: During the data cleaning process, a few outliers or anomalies were identified, potentially caused by irregular market events, global economic shifts, or company-specific occurrences. These outliers were either treated or investigated further to assess their impact on the overall analysis.

- Visualization Insights: The visualizations, including box plots, scatter plots convey that the historical performance of both Tata Motors and Tata Steel stocks. Patterns, trends, and anomalies were clearly highlighted through the graphical representations, aiding in a better understanding of the market behaviour and stock price movements.

- Market Performance Comparison: By looking at the graphs the performance of Tata Motors and Tata Steel, it was evident that the two companies might have responded differently to market dynamics, macroeconomic trends, and industry-specific factors over the years.

- Investment Implications: These analyses could be beneficial for investors and stakeholders looking to make informed decisions regarding investments in Tata Motors and Tata Steel. Understanding the historical trends and patterns could provide insights into potential future market movements and assist in making more informed investment choices.

In conclusion, the data exploration, cleaning, and visualization of the 20-year share price data for Tata Motors and Tata Steel provided valuable insights into the historical performance and market behavior of the two companies. The analyses conducted can serve as a foundation for further in-depth research, modeling, and predictive analysis related to the stock market performance of these companies.

Collab code:
https://colab.research.google.com/drive/1_vwYsBDX3kYH4N4jnkh7dd7AVvCf14qe?usp=sharing