PRODIGY_DS_02

Task 2. To perform data cleaning and exploratory data analysis (EDA) , Explore the relationship between variables and trends in data .

INTRODUCTION :

We will perform various steps involved in Exploratory Data Analysis such as Data processing and cleaning ,Data reduction,Feature engineering,visualysing and analysis.

DATA DESCRIPTION:

Recently Tata groups has listed their shared so I have taken a data set set ot TCS stocks data from 2004-2021 .

STEP1. Loading Data set

In [141…
```
import pandas as pd
Tata_stocks_data=pd.read_csv("C:\\Users\\Ujjaval Raj\\Downloads\\TCS_stock_history.
```

STEP2. Understanding and Describing data set.

In [142…
```
Tata_stocks_data.head()
```

Out[142]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
0	2002-08-12	28.794172	29.742206	28.794172	29.519140	212976	0.0	0.0
1	2002-08-13	29.556316	30.030333	28.905705	29.119476	153576	0.0	0.0
2	2002-08-14	29.184536	29.184536	26.563503	27.111877	822776	0.0	0.0
3	2002-08-15	27.111877	27.111877	27.111877	27.111877	0	0.0	0.0
4	2002-08-16	26.972458	28.255089	26.582090	27.046812	811856	0.0	0.0

In [143…
```
Tata_stocks_data.tail()
```

Out[143]:

	Date	Open	High	Low	Close	Volume	Dividends	Stock Splits
4458	2021-09-24	3890.000000	3944.399902	3855.000000	3871.300049	2320754	0.0	0.0
4459	2021-09-27	3900.000000	3904.000000	3802.899902	3836.949951	1673362	0.0	0.0
4460	2021-09-28	3850.000000	3850.000000	3751.250000	3779.149902	2253075	0.0	0.0
4461	2021-09-29	3759.800049	3806.000000	3722.149902	3791.899902	2489161	0.0	0.0
4462	2021-09-30	3805.000000	3805.000000	3765.000000	3773.199951	640479	0.0	0.0

In [144…

```
Tata_stocks_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4463 entries, 0 to 4462
Data columns (total 8 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   Date          4463 non-null   object 
 1   Open          4463 non-null   float64
 2   High          4463 non-null   float64
 3   Low           4463 non-null   float64
 4   Close         4463 non-null   float64
 5   Volume        4463 non-null   int64  
 6   Dividends     4463 non-null   float64
 7   Stock Splits  4463 non-null   float64
dtypes: float64(6), int64(1), object(1)
memory usage: 279.1+ KB
```

In [145…

```
Tata_stocks_data.describe()
```

Out[145]:

	Open	High	Low	Close	Volume	Dividends	Stock Splits
count	4463.000000	4463.000000	4463.000000	4463.000000	4.463000e+03	4463.000000	4463.000000
mean	866.936239	876.675013	856.653850	866.537398	3.537876e+06	0.071533	0.001344
std	829.905368	838.267104	821.233477	829.611313	3.273531e+06	0.965401	0.051842
min	24.146938	27.102587	24.146938	26.377609	0.000000e+00	0.000000	0.000000
25%	188.951782	191.571816	185.979417	188.594620	1.860959e+06	0.000000	0.000000
50%	530.907530	534.751639	525.616849	529.713257	2.757742e+06	0.000000	0.000000
75%	1156.462421	1165.815854	1143.622800	1154.784851	4.278625e+06	0.000000	0.000000
max	3930.000000	3981.750000	3892.100098	3954.550049	8.806715e+07	40.000000	2.000000

In [146…

```
Tata_stocks_data.nunique()
```

Out[146]:
```
Date            4463
Open            4460
High            4461
Low             4462
Close           4397
Volume          4435
Dividends         29
Stock Splits       2
dtype: int64
```

In [147…
```
Tata_stocks_data.isnull().sum()
```

Out[147]:
```
Date            0
Open            0
High            0
Low             0
Close           0
Volume          0
Dividends       0
Stock Splits    0
dtype: int64
```

In [148…
```
Tata_stocks_data['Date'] = pd.to_datetime(Tata_stocks_data['Date'])
```

In [149…
```
Tata_stocks_data = Tata_stocks_data .drop(['Stock Splits'], axis = 1)
Tata_stocks_data .info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4463 entries, 0 to 4462
Data columns (total 7 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   Date       4463 non-null   datetime64[ns]
 1   Open       4463 non-null   float64
 2   High       4463 non-null   float64
 3   Low        4463 non-null   float64
 4   Close      4463 non-null   float64
 5   Volume     4463 non-null   int64
 6   Dividends  4463 non-null   float64
dtypes: datetime64[ns](1), float64(5), int64(1)
memory usage: 244.2 KB
```

Conclusion:

1. Data set has 8 columns and having variables Date , Open , High , Low , Close , Volume , Dividends , Stock Splits.

2. There are 4463 entries.

3. Date has data type object and volume has integer except these two all other have float data type.

4. No duplicates in date as all entries are unique.

5. No null values in the data set.

6. We get various basic statistic about data set from function .describe()

STEP3. Reduction of data set.

Conclusion:

1. I removed Stock Splits column as most of the entries were zero.

STEP4. Feature Engineering.

In [150…

```
Tata_stocks_data ['DayOfWeek'] = Tata_stocks_data ['Date'].dt.dayofweek
Tata_stocks_data ['Month'] = Tata_stocks_data ['Date'].dt.month
Tata_stocks_data ['Quarter'] = Tata_stocks_data ['Date'].dt.quarter
Tata_stocks_data ['Year'] = Tata_stocks_data ['Date'].dt.year
Tata_stocks_data ['DailyPriceChange'] = Tata_stocks_data ['Close'] - Tata_stocks_da
Tata_stocks_data ['Close_Lag1'] = Tata_stocks_data ['Close'].shift(1)
Tata_stocks_data['MA_7'] = Tata_stocks_data['Close'].rolling(window=7).mean()

Tata_stocks_data.head()
```

Out[150]:

	Date	Open	High	Low	Close	Volume	Dividends	DayOfWeek	Montl
0	2002-08-12	28.794172	29.742206	28.794172	29.519140	212976	0.0	0	
1	2002-08-13	29.556316	30.030333	28.905705	29.119476	153576	0.0	1	
2	2002-08-14	29.184536	29.184536	26.563503	27.111877	822776	0.0	2	
3	2002-08-15	27.111877	27.111877	27.111877	27.111877	0	0.0	3	
4	2002-08-16	26.972458	28.255089	26.582090	27.046812	811856	0.0	4	

Conclusion:

1.I have formed new variables based on Date like Dayofweek , Month, Quater ,Year ,Dailypricechance so that we can use these to visualise the data more efficiently .
2.Close_Lag1 is formed from last date close dependency (lag). 3.Moving averages for 7 dates as MA_7

STEP5. Data visualisation and interpretation .

OHLC plot

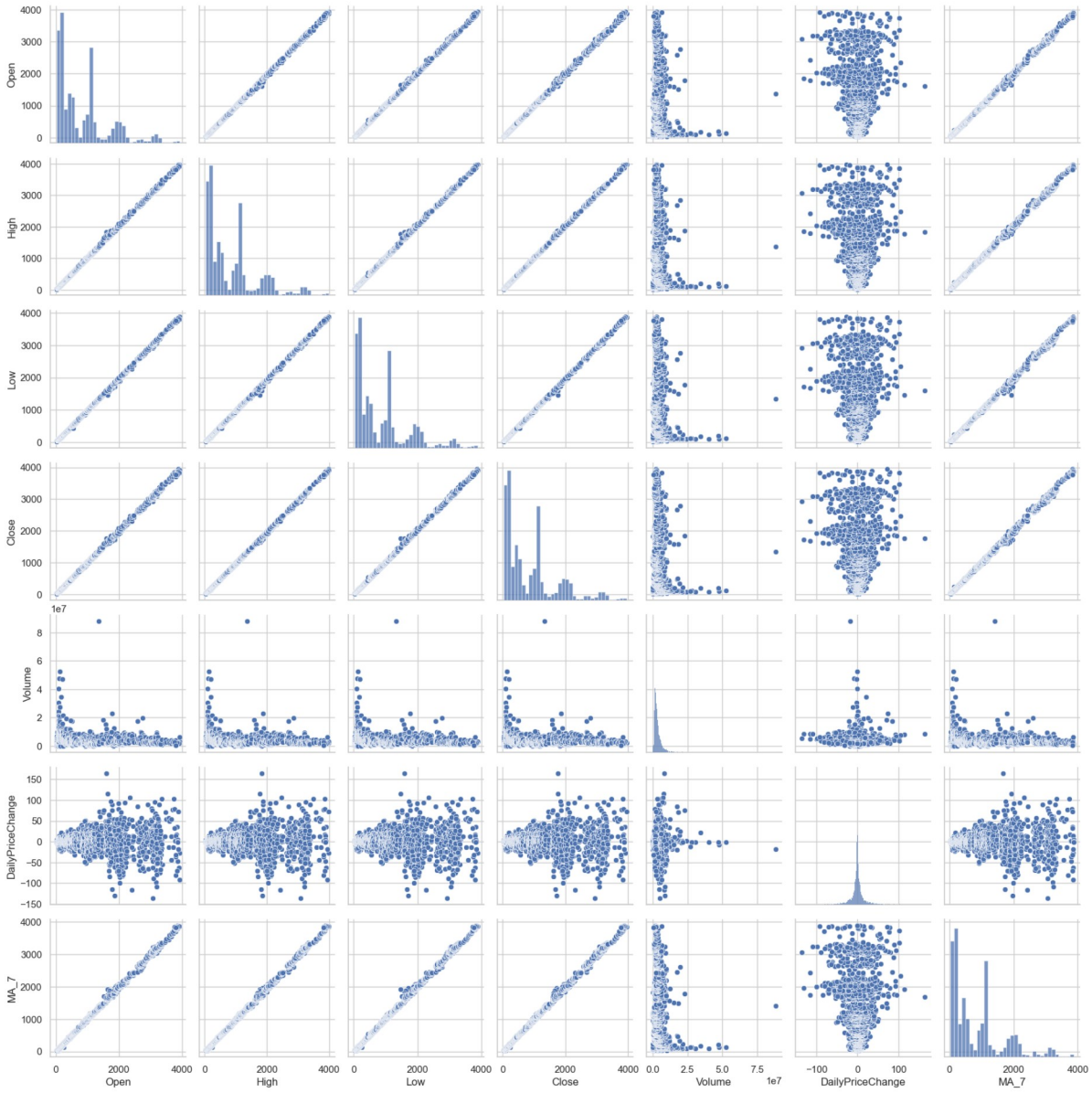
In [151…

```
import plotly.graph_objects as go

fig = go.Figure(data=go.Ohlc(x=Tata_stocks_data ['Date'],
        open=Tata_stocks_data ['Open'],
        high=Tata_stocks_data ['High'],
        low=Tata_stocks_data ['Low'],
        close=Tata_stocks_data ['Close']))
fig.show()
```

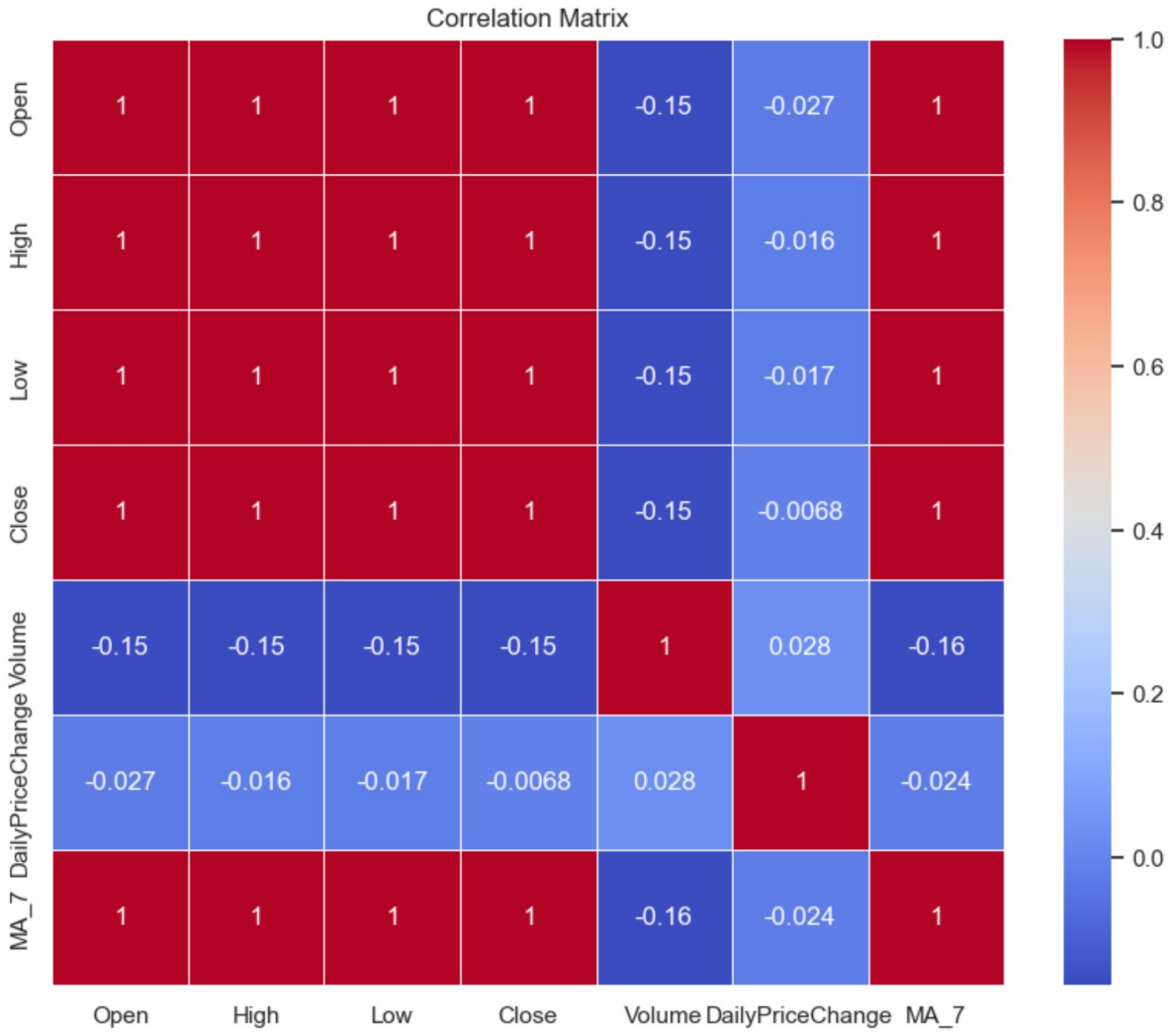
Pairs Plot

In [154…
```
import warnings
warnings.filterwarnings('ignore')
sns.pairplot(Tata_stocks_data[['Open', 'High', 'Low', 'Close', 'Volume', 'DailyPric
plt.show()
```



Heat maps for Correlation

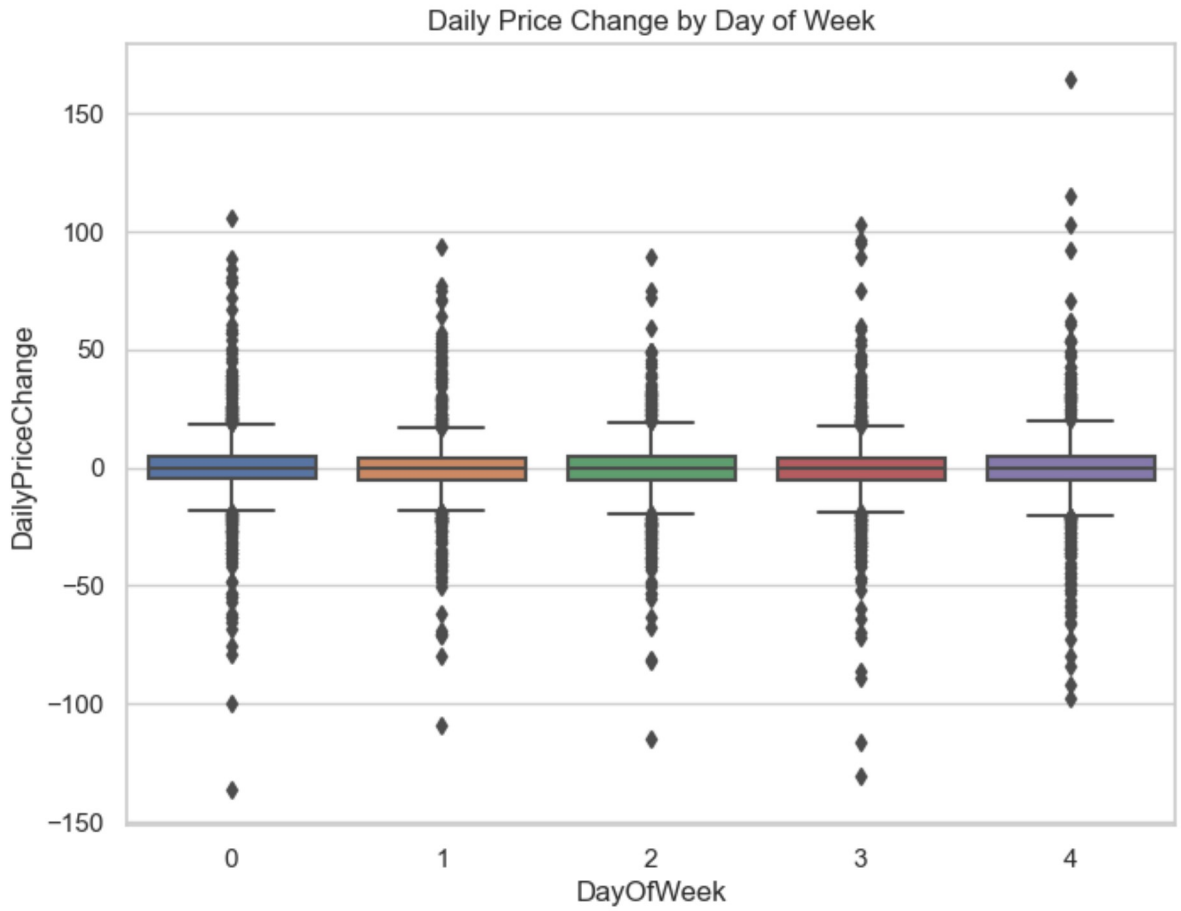
In [155…
```
plt.figure(figsize=(10, 8))
sns.heatmap(Tata_stocks_data[['Open', 'High', 'Low', 'Close', 'Volume', 'DailyPrice
plt.title('Correlation Matrix')
plt.show()
```



Box plot for 'DailyPriceChange' to identify outliers

In [156…

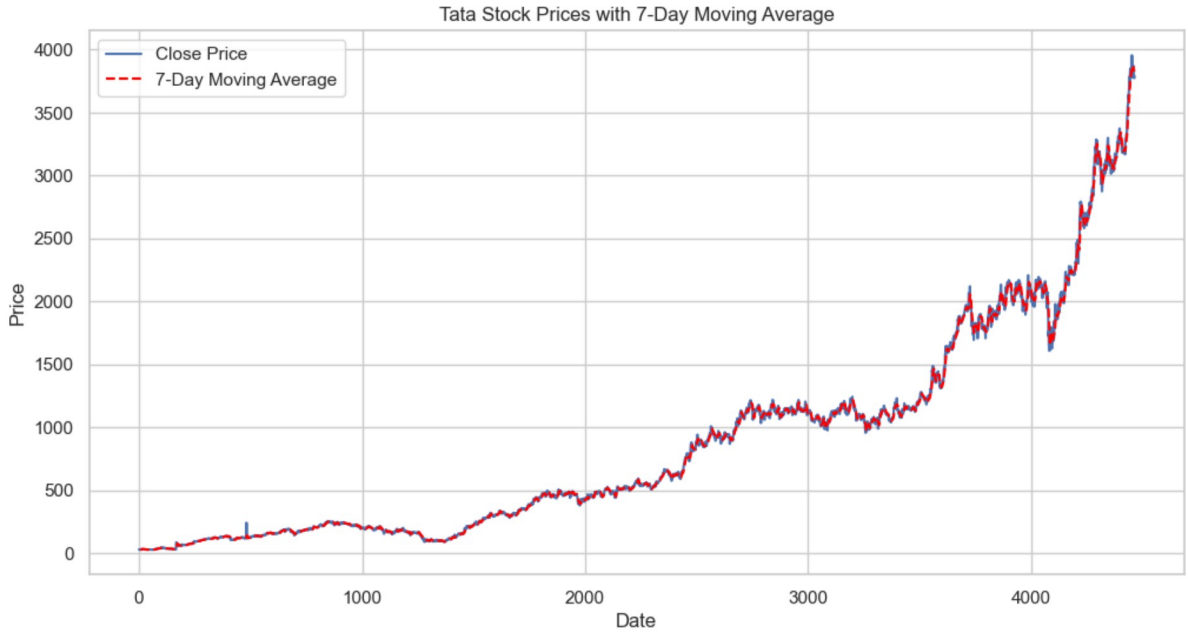
```
plt.figure(figsize=(8, 6))
sns.boxplot(x='DayOfWeek', y='DailyPriceChange', data=Tata_stocks_data)
plt.title('Daily Price Change by Day of Week')
plt.show()
```



Line plot for 'Close' prices with a 7-day moving average

In [157…

```
plt.figure(figsize=(12, 6))
plt.plot(Tata_stocks_data.index, Tata_stocks_data['Close'], label='Close Price')
plt.plot(Tata_stocks_data.index, Tata_stocks_data['MA_7'], label='7-Day Moving Aver
plt.title('Tata Stock Prices with 7-Day Moving Average')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```

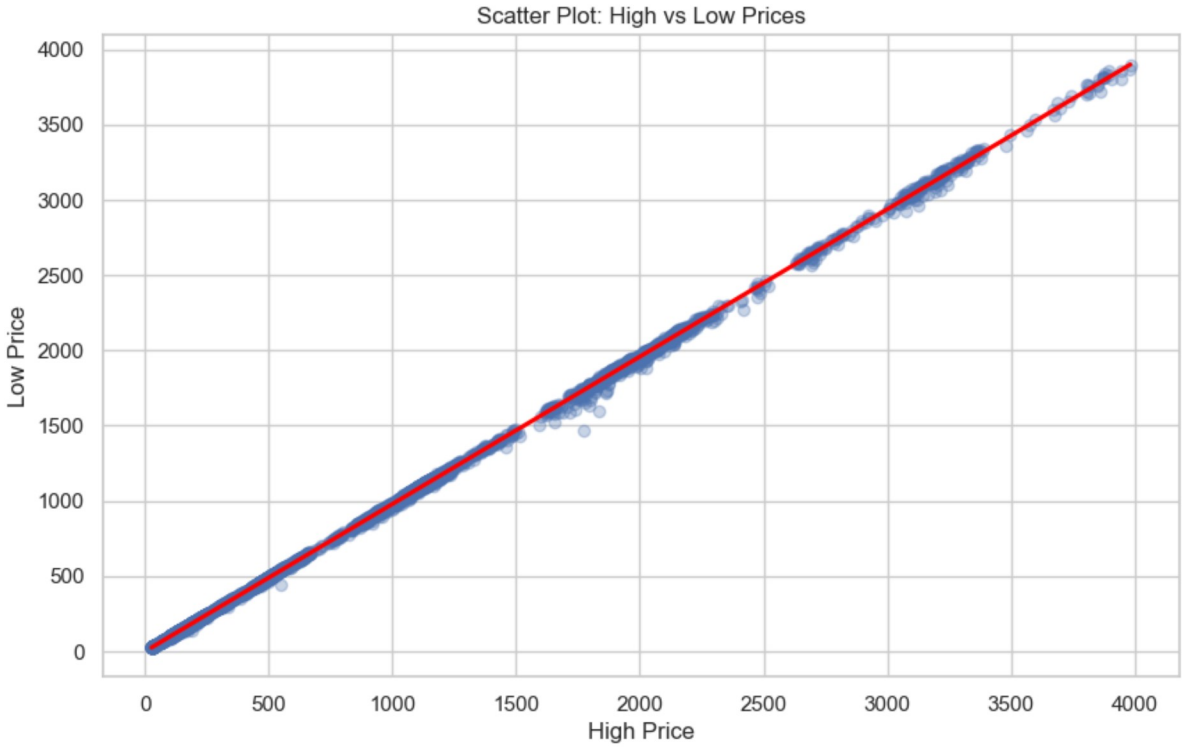


Volume Chart

In [165…
```
plt.figure(figsize=(12, 3))
plt.bar(Tata_stocks_data.index, Tata_stocks_data['Volume'], color='red', alpha=0.7)
plt.title('Trading Volume Over Time')
plt.xlabel('Date')
plt.ylabel('Volume')
plt.show()
```

Scatter Plot with Regression Line

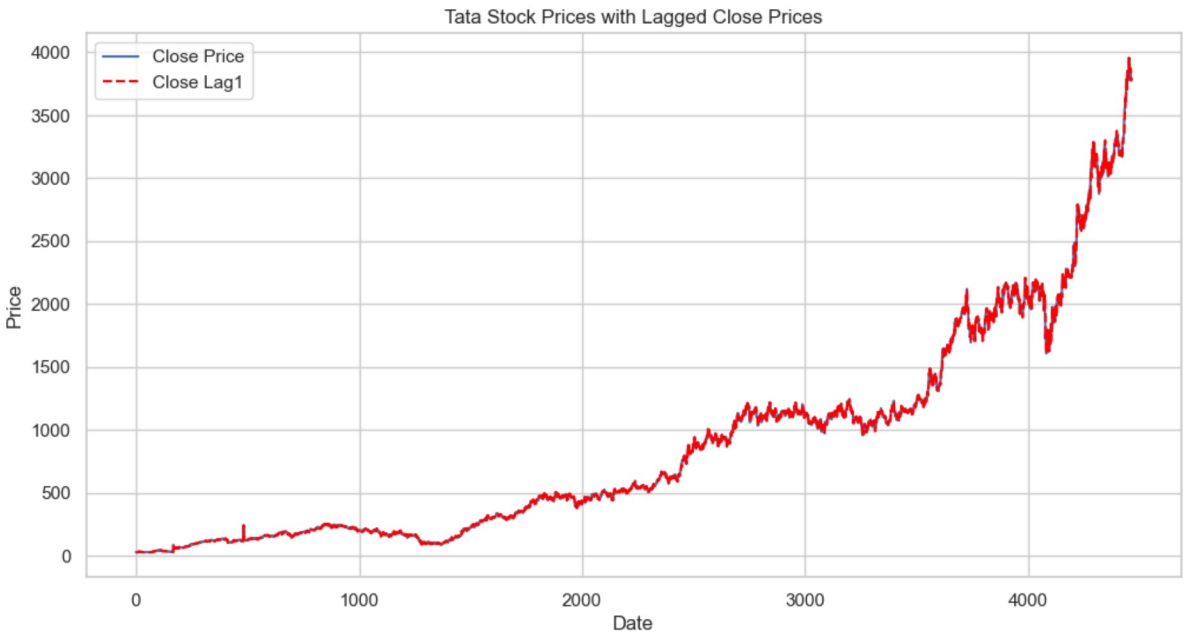
In [166…
```
plt.figure(figsize=(10, 6))
sns.regplot(x='High', y='Low', data=Tata_stocks_data, scatter_kws={'alpha':0.3}, li
plt.title('Scatter Plot: High vs Low Prices')
plt.xlabel('High Price')
plt.ylabel('Low Price')
plt.show()
```



Line Plot for Close Price and Close_lag1

In [168…

```
plt.figure(figsize=(12, 6))
plt.plot(Tata_stocks_data.index, Tata_stocks_data['Close'], label='Close Price')
plt.plot(Tata_stocks_data.index, Tata_stocks_data['Close_Lag1'], label='Close Lag1'
plt.title('Tata Stock Prices with Lagged Close Prices')
plt.xlabel('Date')
plt.ylabel('Price')
plt.legend()
plt.show()
```



Simple Moving Averages Chart

In [173…

```
Tata_stocks_data['SMA5'] = Tata_stocks_data.Close.rolling(5).mean()
Tata_stocks_data['SMA25'] = Tata_stocks_data.Close.rolling(25).mean()
Tata_stocks_data['SMA50'] = Tata_stocks_data.Close.rolling(50).mean()
Tata_stocks_data['SMA250'] = Tata_stocks_data.Close.rolling(250).mean()
Tata_stocks_data['SMA500'] = Tata_stocks_data.Close.rolling(500).mean()

fig = go.Figure(data=[go.Ohlc(x=Tata_stocks_data['Date'],
                              open=Tata_stocks_data['Open'],
                              high=Tata_stocks_data['High'],
                              low=Tata_stocks_data['Low'],
                              close=Tata_stocks_data['Close'], name = "OHLC"),
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA5,
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA25,
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA50,
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA250
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA500
fig.show()
```

Exponential Moving Averages Chart

In [174…

```
Tata_stocks_data['EMA5'] = Tata_stocks_data.Close.ewm(span=5, adjust=False).mean()
Tata_stocks_data['EMA25'] = Tata_stocks_data.Close.ewm(span=25, adjust=False).mean(
Tata_stocks_data['EMA50'] = Tata_stocks_data.Close.ewm(span=50, adjust=False).mean(
Tata_stocks_data['EMA250'] = Tata_stocks_data.Close.ewm(span=250, adjust=False).mea
Tata_stocks_data['EMA500'] = Tata_stocks_data.Close.ewm(span=500, adjust=False).mea

fig = go.Figure(data=[go.Ohlc(x=Tata_stocks_data['Date'],
                              open=Tata_stocks_data['Open'],
                              high=Tata_stocks_data['High'],
                              low=Tata_stocks_data['Low'],
                              close=Tata_stocks_data['Close'], name = "OHLC"),
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA5,
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA25,
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA50,
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA250
                      go.Scatter(x=Tata_stocks_data.Date, y=Tata_stocks_data.SMA500
fig.show()
```

INTERPRETATION

1. Tata_stocks_shares went up by margin 1260 within past 17 years .

2. Tata stocks shares keep on increasing year wise.

3. High, Open ,low ,close has a good correlation between them.

4. There exist some outliers in box plot that suggest that some times shares wont follow general trend .

5. Except for few date volumes are less than 2 .

6. There exist strong correlation of lag for close price.

7. With the help of SMA and EMA we can find that SMA25 AND EMA25 are closely related with data trend.

Thank You For Reading!