**Ex.No:5**

**AIM:**

**To** perform Time Series Analysis and apply the various visualization techniques.

**Dataset:**

The C3.AI Stocks Dataset is a collection of stock market data sourced from Yahoo for the period from March 2022 to March 2023. The dataset provides comprehensive information on various stocks and their performance, including stock prices, trading volumes, market capitalization, and other key metrics. This data is useful for researchers, analysts, and investors looking to gain insights into the stock market and make informed investment decisions. With the C3 AI Platform, users can build enterprise-scale AI applications more efficiently and cost-effectively than alternative approaches, enabling them to extract valuable insights from this rich and diverse dataset.

**PROGRAM:**

from google.colab import drive  
drive.mount('/content/drive')

Mounted at /content/drive

import pandas as pd  
import numpy as np  
import matplotlib.pyplot as plt  
import seaborn as sns  
df=pd.read\_csv('/content/AI.csv')  
  
df.head(10)

Date Open High Low Close Adj Close Volume  
0 2022-03-10 20.799999 20.950001 19.709999 19.959999 19.959999 3528900  
1 2022-03-11 20.120001 20.290001 18.000000 18.000000 18.000000 5216800  
2 2022-03-14 17.790001 18.340000 16.709999 16.910000 16.910000 4588400  
3 2022-03-15 16.690001 17.490000 16.584999 17.469999 17.469999 3055800  
4 2022-03-16 17.889999 19.299999 17.809999 19.299999 19.299999 4208200  
5 2022-03-17 19.000000 20.770000 18.700001 20.750000 20.750000 3544100  
6 2022-03-18 20.410000 21.959999 20.110001 21.959999 21.959999 4758200  
7 2022-03-21 21.590000 22.379999 21.209999 21.809999 21.809999 2442100  
8 2022-03-22 21.680000 23.340000 21.530001 23.209999 23.209999 3023100  
9 2022-03-23 22.660000 23.900000 22.389999 23.150000 23.150000 1807700

df.isna()

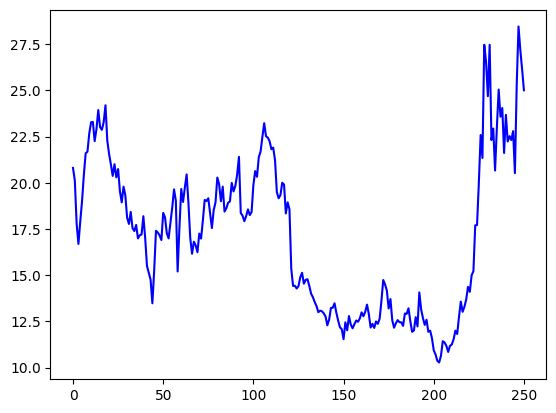
Date Open High Low Close Adj Close Volume  
0 False False False False False False False  
1 False False False False False False False  
2 False False False False False False False  
3 False False False False False False False  
4 False False False False False False False  
.. ... ... ... ... ... ... ...  
246 False False False False False False False  
247 False False False False False False False  
248 False False False False False False False  
249 False False False False False False False  
250 False False False False False False False  
  
[251 rows x 7 columns]

df.isna().sum()

Date 0  
Open 0  
High 0  
Low 0  
Close 0  
Adj Close 0  
Volume 0  
dtype: int64

df['Open'].plot(color='blue')

<Axes: >

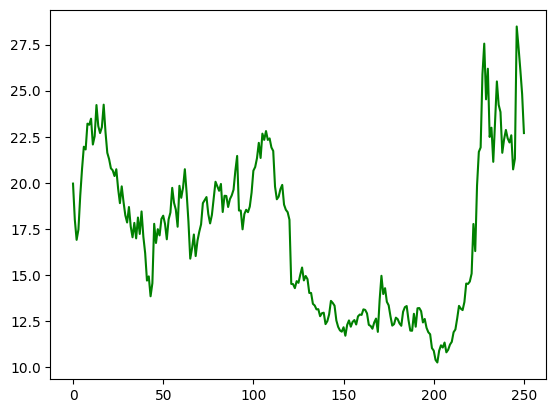


**OBSERVATION:**

**T**he above graph shows the open time of the stock from the given dataset. The plot is plotted with blue line graph. The stock goes upper in the cost about 27.5 rupees. The stock prices goes down at the minimum is about 10 to 11 rupees. The stock price is medium between 17.5 to 20 rupees.

df['Close'].plot(color='green',kind='line')

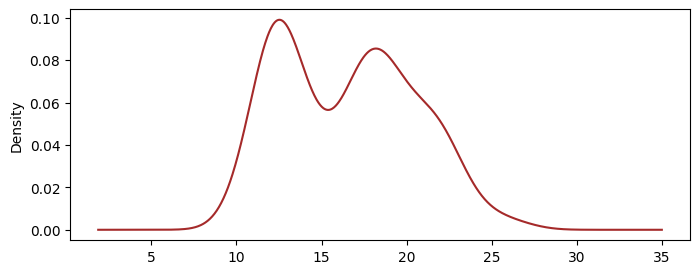
<Axes: >



**OBSERVATION:**

**T**he above graph shows the close time of the stock from the given dataset. The plot is plotted with green line graph. The stock is at upper in the cost about 27.5 rupees. The stock prices at down at the minimum is about 11 rupees. The stock price is medium between 17.5 to 20 rupees.

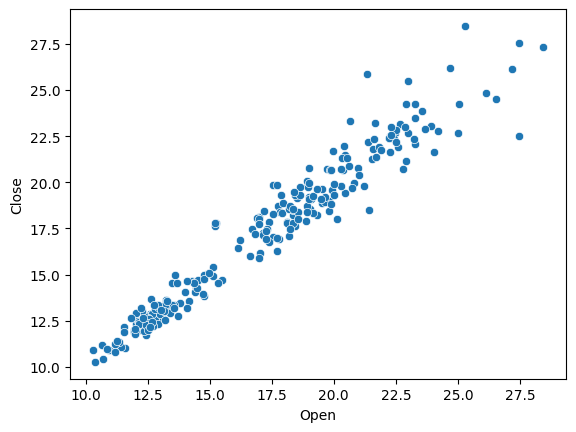
df['Low'].plot(kind='kde',figsize=(8,3),color='brown')  
plt.show()



**OBSERVATION:**

The above graph shows that the low stock prize from the given dataset.The plot is plotted with the brown line in the graph. The lowest stock from the given dataset is 8 rupees. The maximum number of lowest stock price is 12 that means the stock price is maximum in low stock prize.

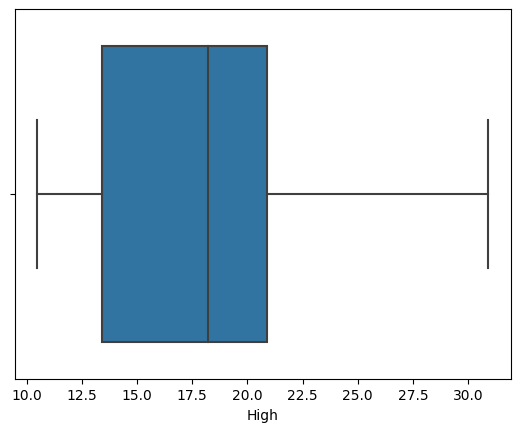
sns.scatterplot(x='Open',y='Close',data=df)  
plt.show()



**OBSERVATION:**

The above scatter plot graph shows that the open and the close time value of the stock price. The x axis represent the open stock price and the y axis represent the the close stock price of the open stock price. The stock is at upper in the cost about 27.5 rupees. The stock prices at down at the minimum is about 11 rupees. The stock price is medium between 17.5 to 20 rupees. The stock goes upper in the cost about 27.5 rupees. The stock prices goes down at the minimum is about 10 to 11 rupees. The stock price is medium between 17.5 to 20 rupees.

sns.boxplot(x='High',data=df)  
plt.show()

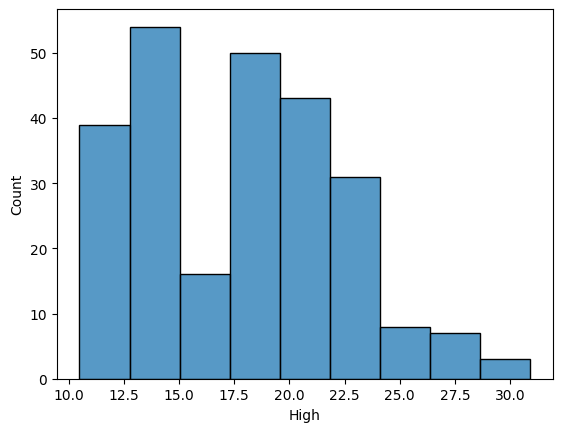


**OBSERVATION:**

The above box plot represent the high in stock price in the given the dataset. The graph was plotted between the 12.5 to 21.5 . Therefore the stock price is in most highest in the month is 21.5. The stock price which the lowest which had been compared with the other month in column is 13.0 rupees.

sns.histplot(x='High', data=df)

<Axes: xlabel='High', ylabel='Count'>

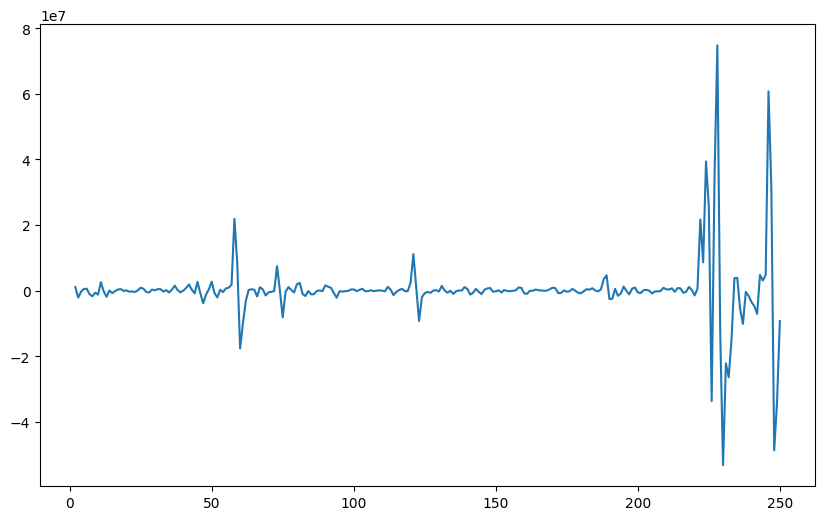


**OBSERVATION**:

The above histogram shows the high in the every month . Therefore the stock price is in most highest in the month is 30. The stock price which the lowest which had been compared with the other month in column is between 15.0 to 17.5 rupees. The medium is between the 22.5 to 25.0.

df.Volume.diff(2).plot(figsize=(10,6))

<Axes: >

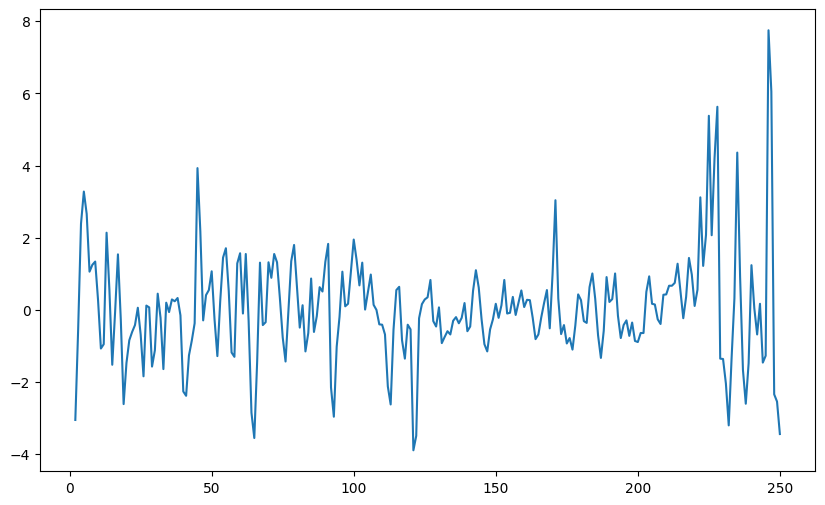


**OBSERVATION:**

The above graph shows the total number of shares traded in the stock on a given day from the dataset. The volume of the share is most highest between the 200 to 250 in the day in the given dataset and most lowest in the given dataset is between 0 to 50.

df.Close.diff(2).plot(figsize=(10,6))

<Axes: >

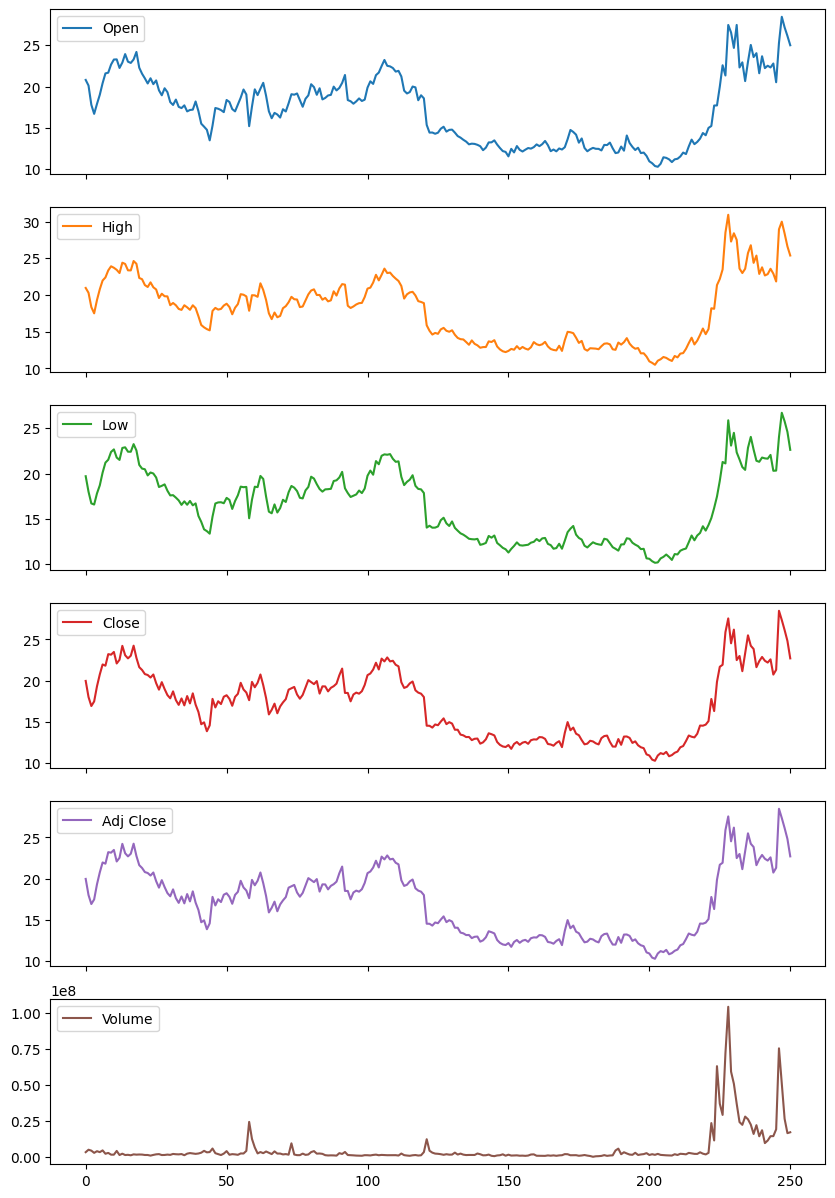


**OBSERVATION:**

**T**he above graph shows the close time of the stock from the given dataset. The plot is plotted with green line graph. The stock is at upper in the cost about 27.5 rupees. The stock prices at down at the minimum is about 11 rupees. The stock price is medium between 17.5 to 20 rupees.

df.plot(subplots=True,figsize=(10,15))

array([<Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >, <Axes: >],  
 dtype=object)

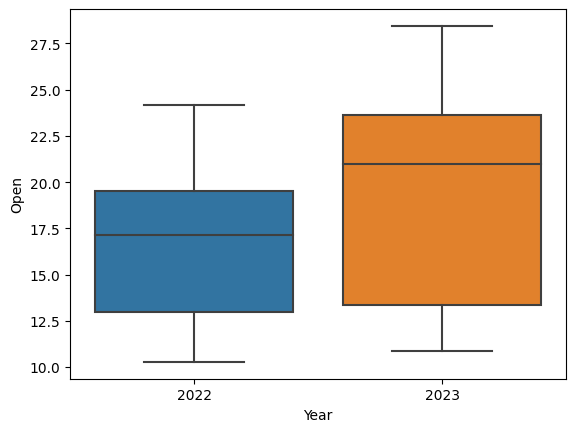


**OBSERVATION:**

The above graph shows that the time analysis of the volume, adj close, close, low, high, open from the dataset. The above graph shows the maximum and minimum in the graph

df['Date']= pd.to\_datetime(df['Date'])  
  
# extract year from date column  
df["Year"] = df["Date"].dt.year  
  
# box plot grouped by year  
sns.boxplot(data=df, x="Year", y="Open")

<Axes: xlabel='Year', ylabel='Open'>



**OBSERVATION:**

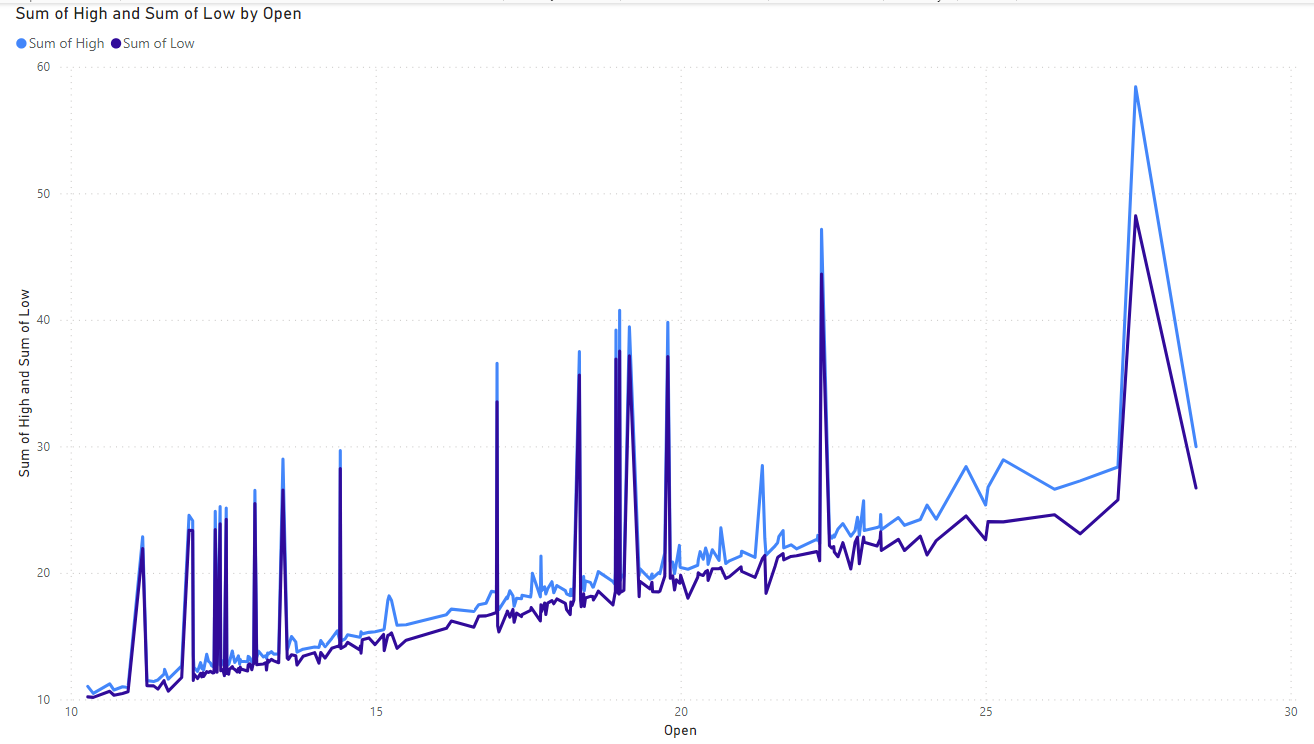
**T**he above graph shows the open time of the stock from the given dataset. The graph is plotted for the two year for all the month. In the year 2022 the stock goes which lower. In the year 2023 the stock price is higher than the before year. In year 2022 the stock price is in between the 13.0 to 19.0. In the year stock price is in between the 13.5 to the maximum.

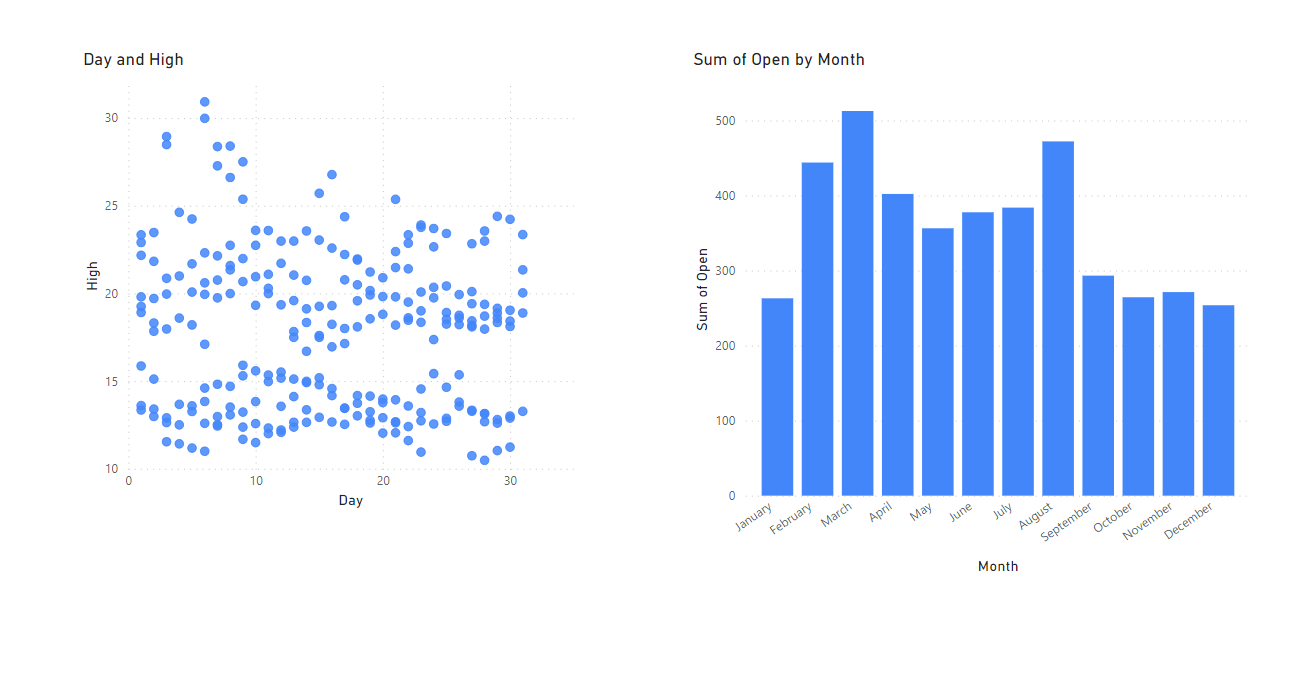
**RESULT:**

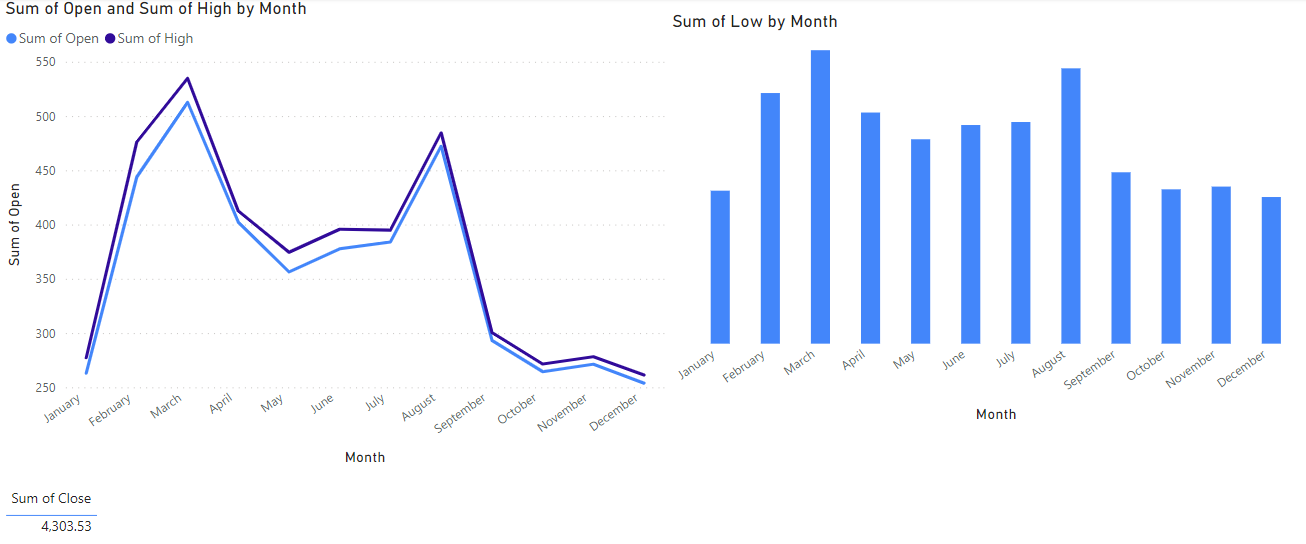
Theperforming Time Series Analysis and apply the various visualization techniques done successfully.

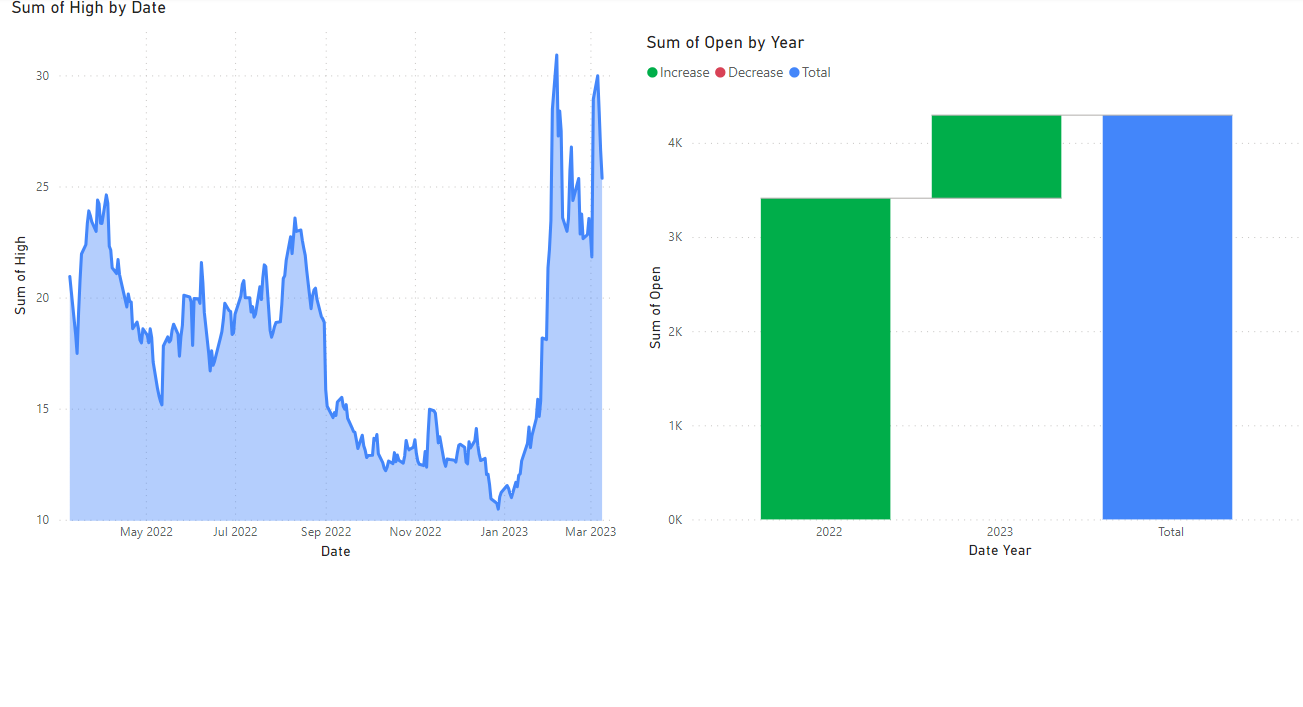
**POWER BI:**

****

****

****

****

****