**VISUALIZATION IN TIME SERIES ANALYSIS**

**Aim:**

To implement a program for visualization in time series analysis.

**Procedure & Code:**

**Step 1: Importing the required libraries**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

**Step 2: Loading and visualizing the dataset**

df=pd.read\_csv("D:/tsa\_107/data/Microsoft\_Stock.csv")

print(df.describe())

df.head()

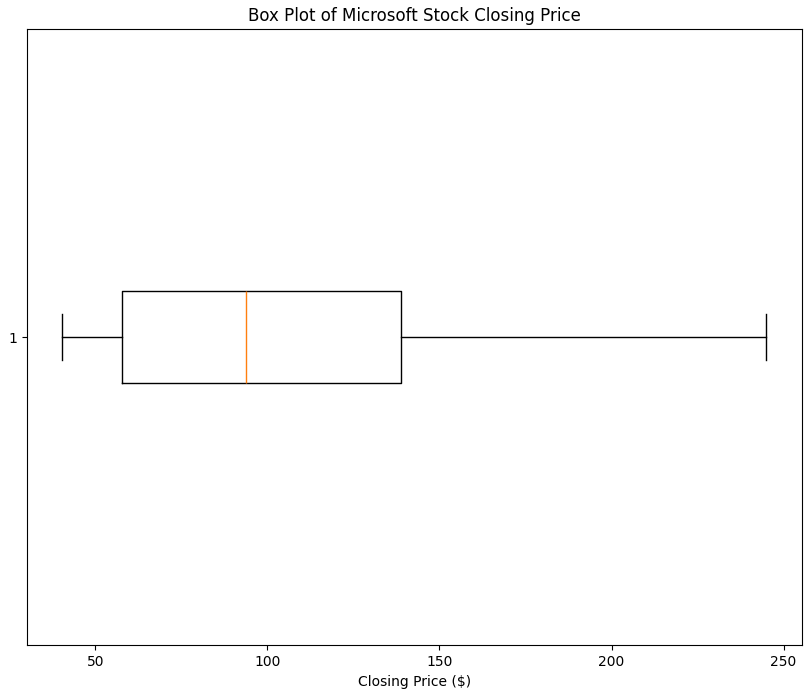
plt.figure(figsize=(10, 8))

plt.boxplot(df['Close'], vert=False)

plt.title('Box Plot of Microsoft Stock Closing Price')

plt.xlabel('Closing Price ($)')

plt.show()



plt.figure(figsize=(8, 6))

plt.hist(df['Close'], bins=20, color='purple', edgecolor='black')

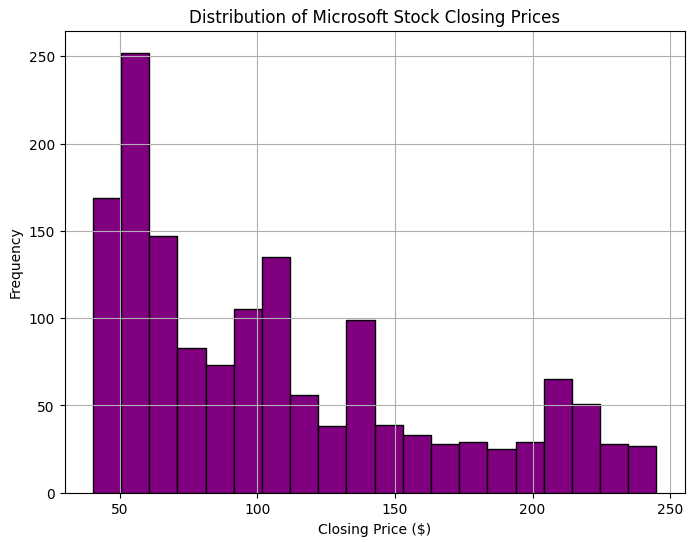
plt.title('Distribution of Microsoft Stock Closing Prices')

plt.xlabel('Closing Price ($)')

plt.ylabel('Frequency')

plt.grid(True)

plt.show()



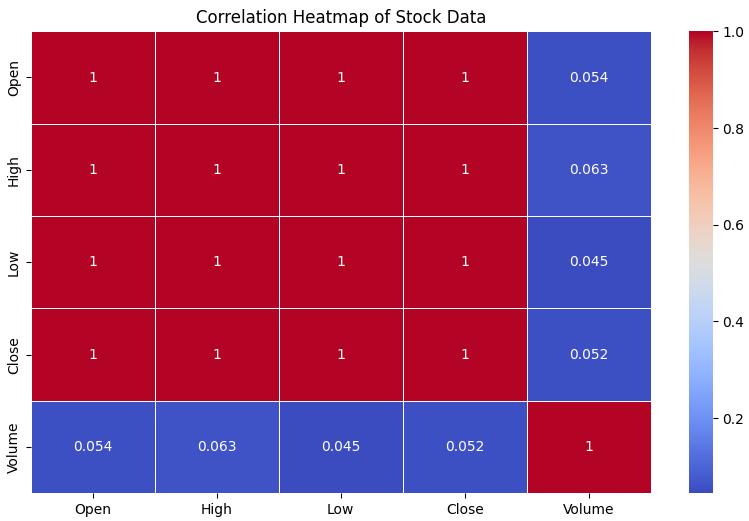
df\_numeric = df.select\_dtypes(include=[np.number])

plt.figure(figsize=(10, 6))

sns.heatmap(df\_numeric.corr(), annot=True, cmap='coolwarm', linewidths=0.5)

plt.title('Correlation Heatmap of Stock Data')

plt.show()



import matplotlib.dates as mdates

plt.figure(figsize=(12, 6))

plt.plot(df['Date'], df['Close'], color='blue')

plt.title('Stock Price Trend Over Time')

plt.xlabel('Date')

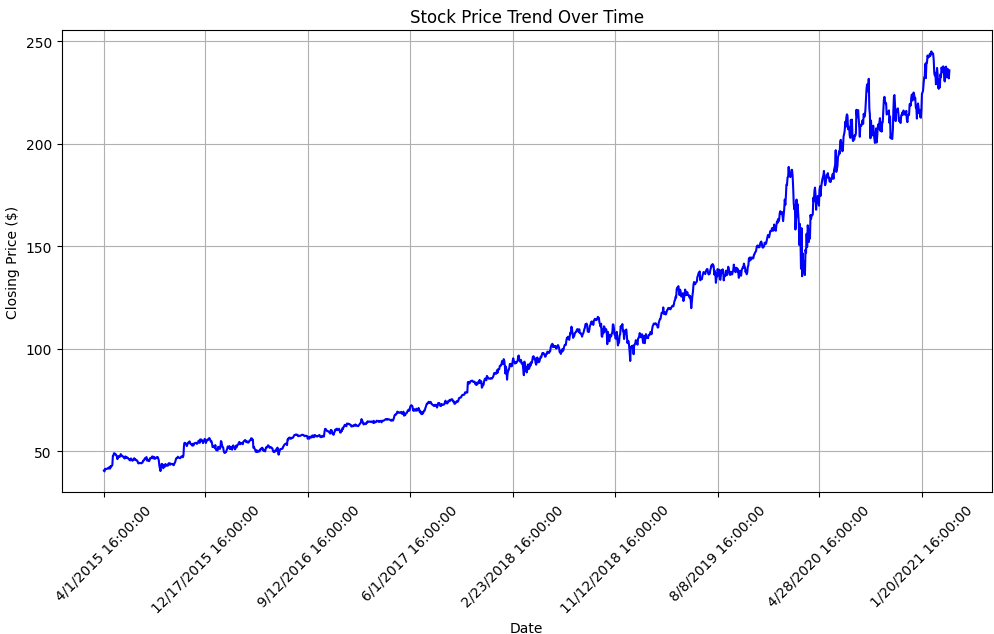
plt.ylabel('Closing Price ($)')

plt.xticks(rotation=45)

plt.gca().xaxis.set\_major\_locator(mdates.AutoDateLocator())

plt.grid(True)

plt.show()



import networkx as nx

df\_numeric = df.select\_dtypes(include=[np.number])

corr\_matrix = df\_numeric.corr()

G = nx.Graph()

threshold = 0.5

for i in corr\_matrix.columns:

for j in corr\_matrix.columns:

if i != j and abs(corr\_matrix.loc[i, j]) > threshold:

G.add\_edge(i, j, weight=corr\_matrix.loc[i, j])

plt.figure(figsize=(8, 6))

pos = nx.spring\_layout(G)

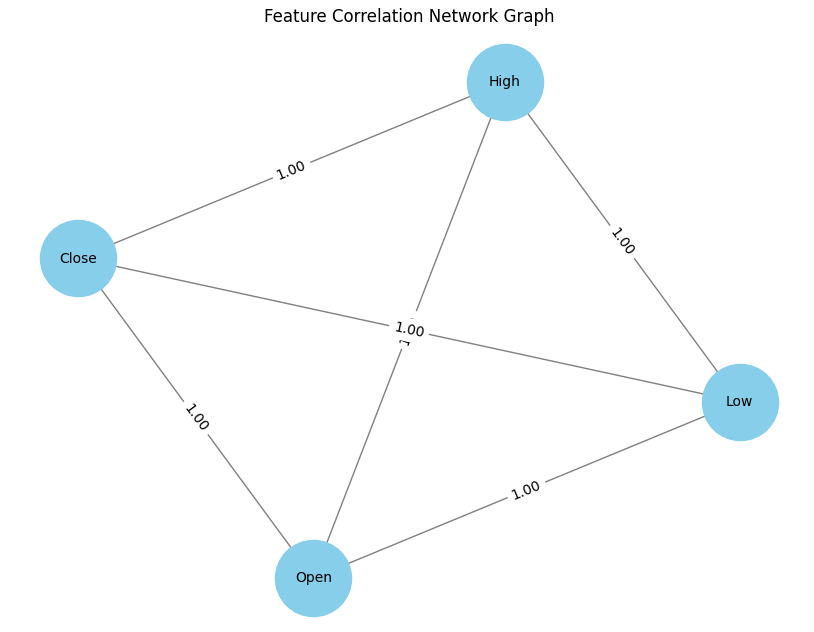
nx.draw(G, pos, with\_labels=True, node\_color='skyblue', edge\_color='gray', node\_size=3000, font\_size=10)

edge\_labels = {(i, j): f"{corr\_matrix.loc[i, j]:.2f}" for i, j in G.edges()}

nx.draw\_networkx\_edge\_labels(G, pos, edge\_labels=edge\_labels)

plt.title("Feature Correlation Network Graph")

plt.show()



**Result:**

Thus the implementation of a program for time series data cleaning, loading, handling and preprocessing techniques has been successfully written and executed successfully.