**Exp no: 1 Implement programs for time series data cleaning, loading and handling times series data and pre-processing techniques.**

Date: 20/03/25

**Objectives:**

The primary objective of this experiment is to analyze and visualize stock market data using Python. By processing the dataset, cleaning missing values, and applying data visualization techniques, we aim to identify trends, variations, and patterns in stock prices. The analysis will provide insights into how different stock tickers have performed over time.

**Background Scope:**

The dataset contains historical stock prices, including information such as closing price, adjusted closing price, opening price, highest and lowest price, and trading volume. The dataset is useful for understanding stock market trends, detecting outliers, and performing exploratory data analysis (EDA) before building predictive models.

Steps for Stock Market Data Preprocessing and Analysis:

Step 1: Load the Dataset

Load the dataset from a local CSV file and display the first few rows to understand its structure.

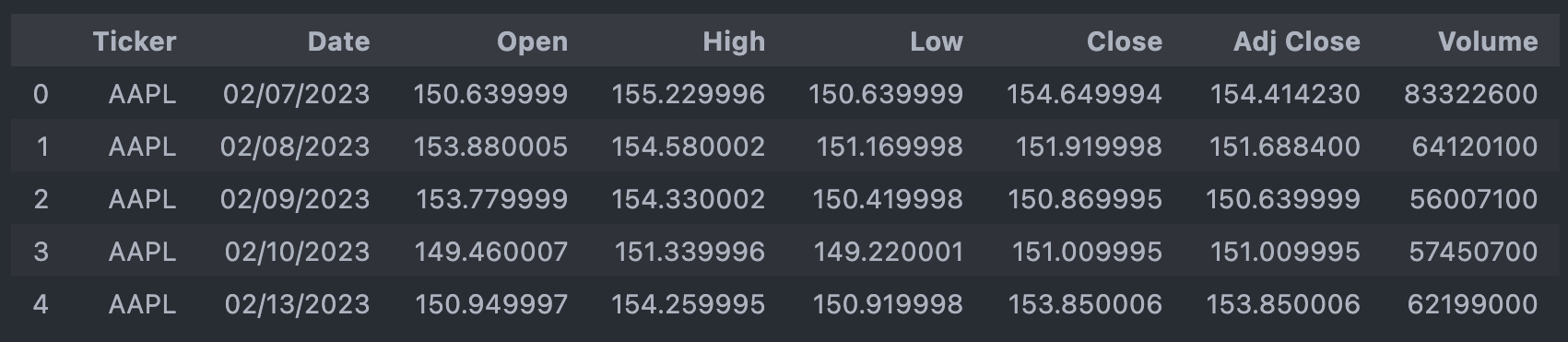
import pandas as pd

# Load the dataset

df = pd.read\_csv("market.csv")

# Display the first few rows

print(df.head())



Step 2: Data Cleaning and Preprocessing

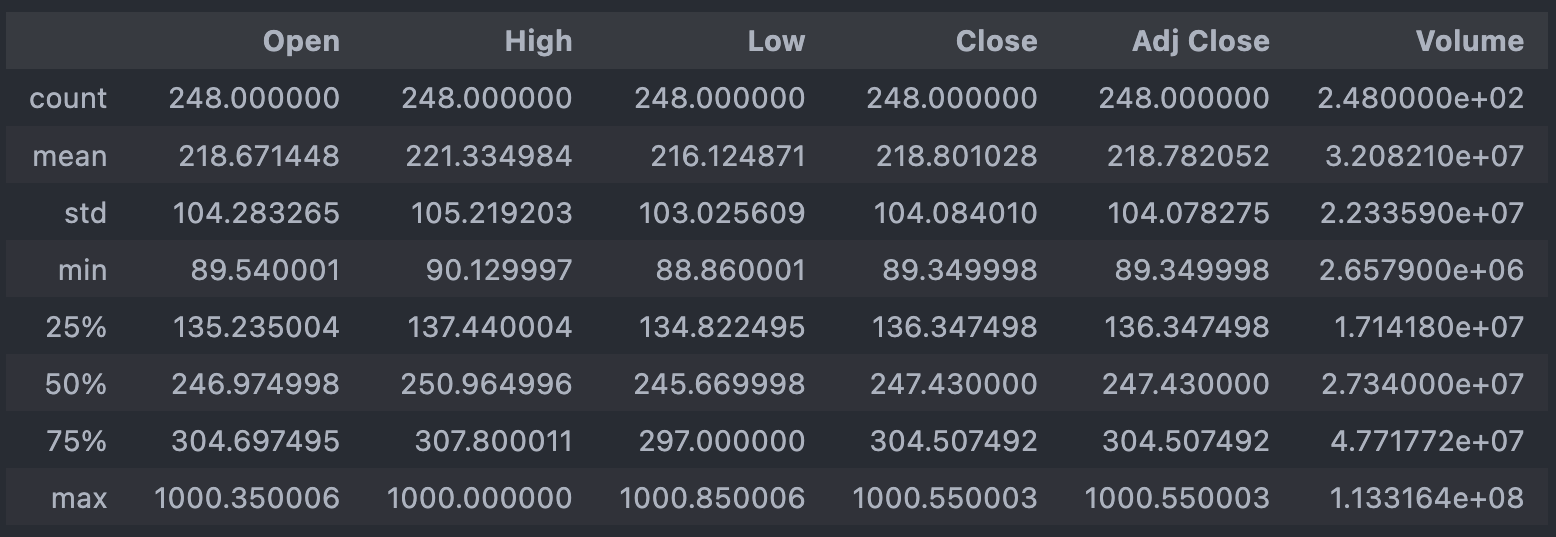
Convert the date column to a datetime object and remove missing values.

# Convert 'Date' to datetime format

df['Date'] = pd.to\_datetime(df['Date'])

# Remove missing values

df.dropna(subset=['Date', 'Close', 'Adj Close', 'Open', 'High', 'Low', 'Volume'], inplace=True)



Step 3: Handle Duplicates

Remove duplicate records based on date and ticker to ensure data integrity.

df.drop\_duplicates(subset=['Date', 'Ticker'], inplace=True)



Step 4: Outlier Removal

Use the interquartile range (IQR) method to detect and remove outliers from the closing price.

Q1 = df['Close'].quantile(0.25)

Q3 = df['Close'].quantile(0.75)

IQR = Q3 - Q1

df = df[(df['Close'] >= (Q1 - 1.5 \* IQR)) & (df['Close'] <= (Q3 + 1.5 \* IQR))]

Step 5: Visualizing Stock Prices Over Time

Plot stock prices to observe trends for different tickers over time.

import matplotlib.pyplot as plt

# Plot the time series data

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

plt.plot(df['Date'], df['Close'], color='blue', marker='o', linestyle='-')

plt.title('Stock Prices Over Time')

plt.xlabel('Date')

plt.ylabel('Closing Price')

plt.grid(True)

plt.show()

Step 6: Compare Different Stock Tickers

Generate subplots to compare different stock tickers.

fig, axs = plt.subplots(3, 2, figsize=(15, 16))

fig.suptitle('Stock Data by Ticker Type')

cols = ['Close', 'Adj Close', 'Open', 'High', 'Low', 'Volume']

for i, col in enumerate(cols):

row = i // 2

col = i % 2

for ticker, data in df.groupby('Ticker'):

axs[row, col].plot(data['Date'], data[cols[i]], label=ticker)

axs[row, col].set\_title(cols[i])

axs[row, col].set\_xlabel('Date')

axs[row, col].set\_ylabel('Price')

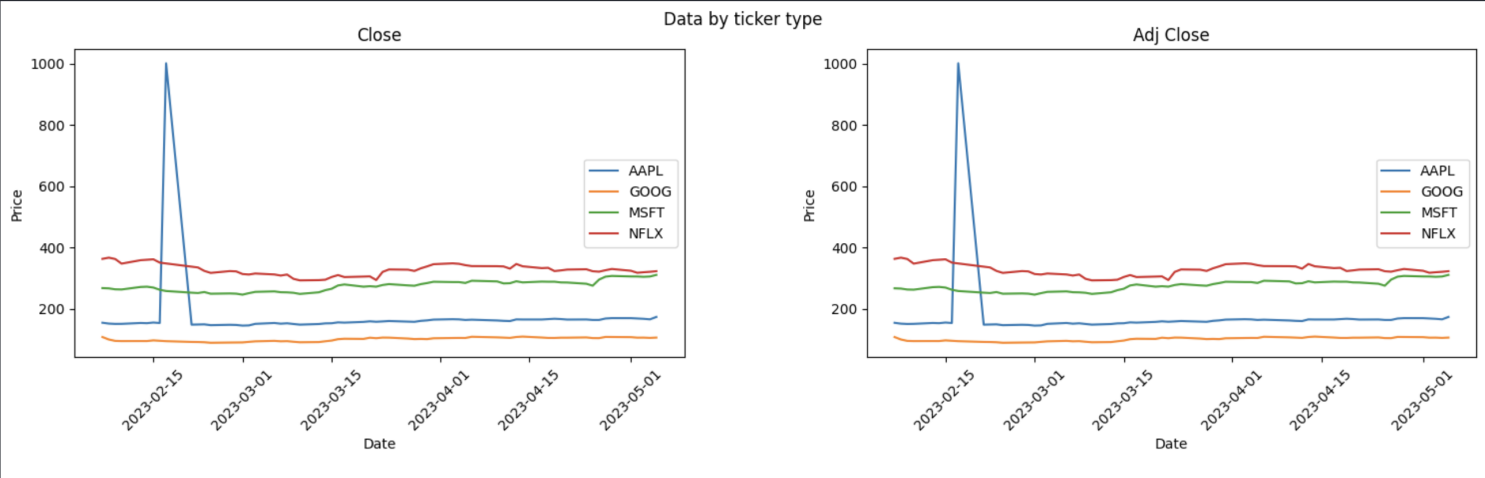
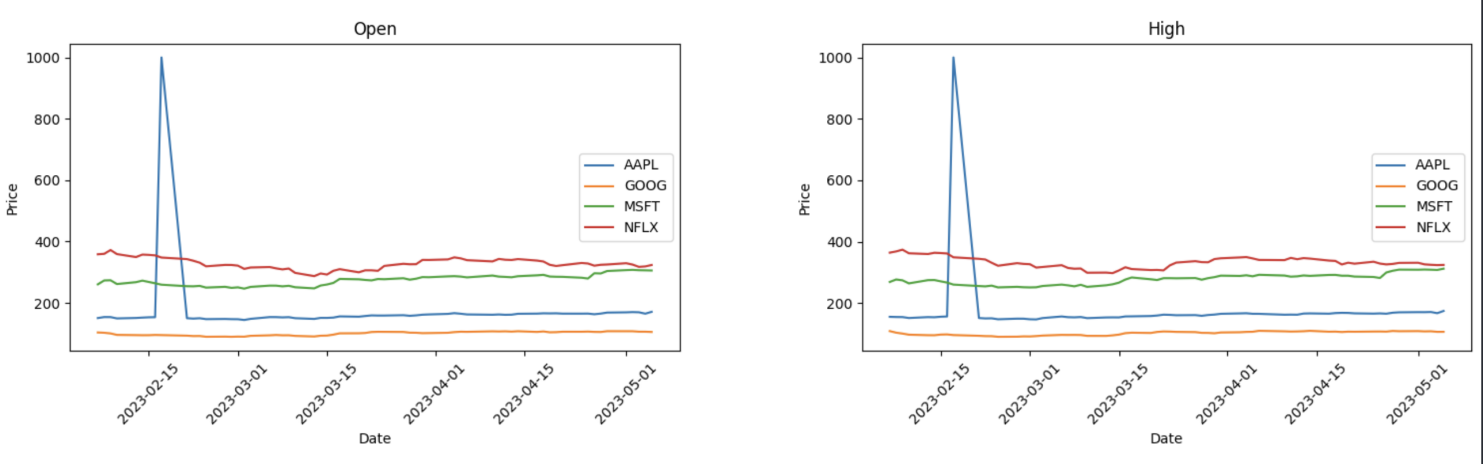
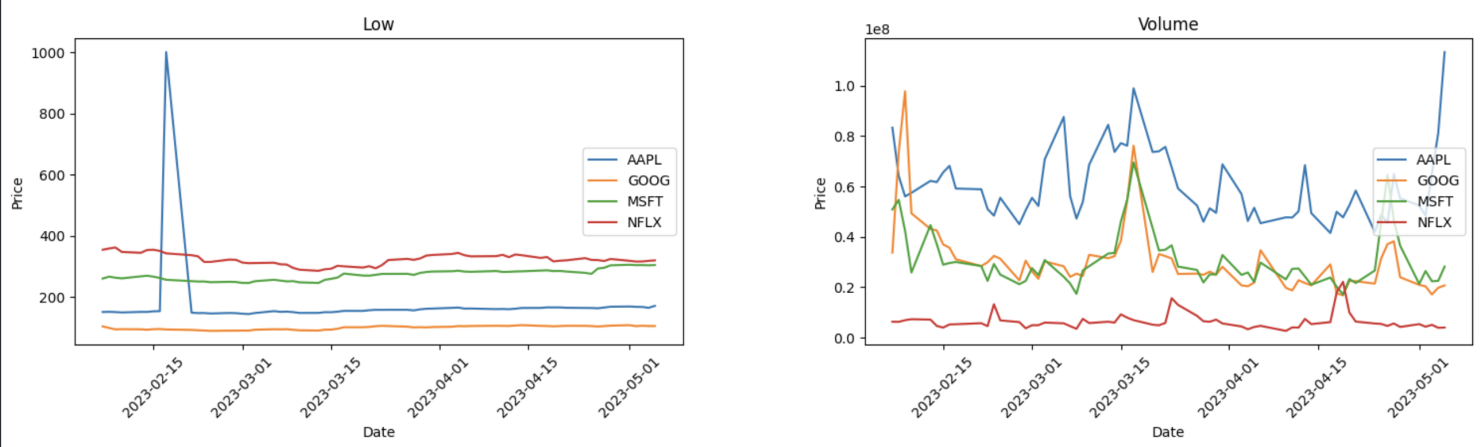
axs[row, col].legend(loc='right')

axs[row, col].tick\_params(axis='x', rotation=45)

plt.tight\_layout()

plt.subplots\_adjust(wspace=0.3, hspace=0.8)

plt.show()



**Result:**

Thus, the stock market dataset is successfully analyzed and visualized. The processed data can be further used for predictive modeling or deeper financial analysis.