**Exp no: 2 Implement programs for visualizing time series data**

**Date:** 20/03/25

**Objectives:**

The primary objective of this experiment is to visualize and analyze time series stock market data. By applying various visualization techniques, we aim to identify trends, patterns, and seasonality in stock prices over time. This analysis will help in understanding stock price fluctuations and assist in financial forecasting.

**Background Scope:**

Time series data consists of observations recorded sequentially over time. The dataset used in this experiment contains historical stock prices, including attributes like closing price, adjusted closing price, opening price, highest and lowest price, and trading volume. By applying resampling, lag plots, and autocorrelation analysis, we gain insights into the structure and trends within the data.

Steps for Time Series Data Preprocessing and Visualization:

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 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: Comparing 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')

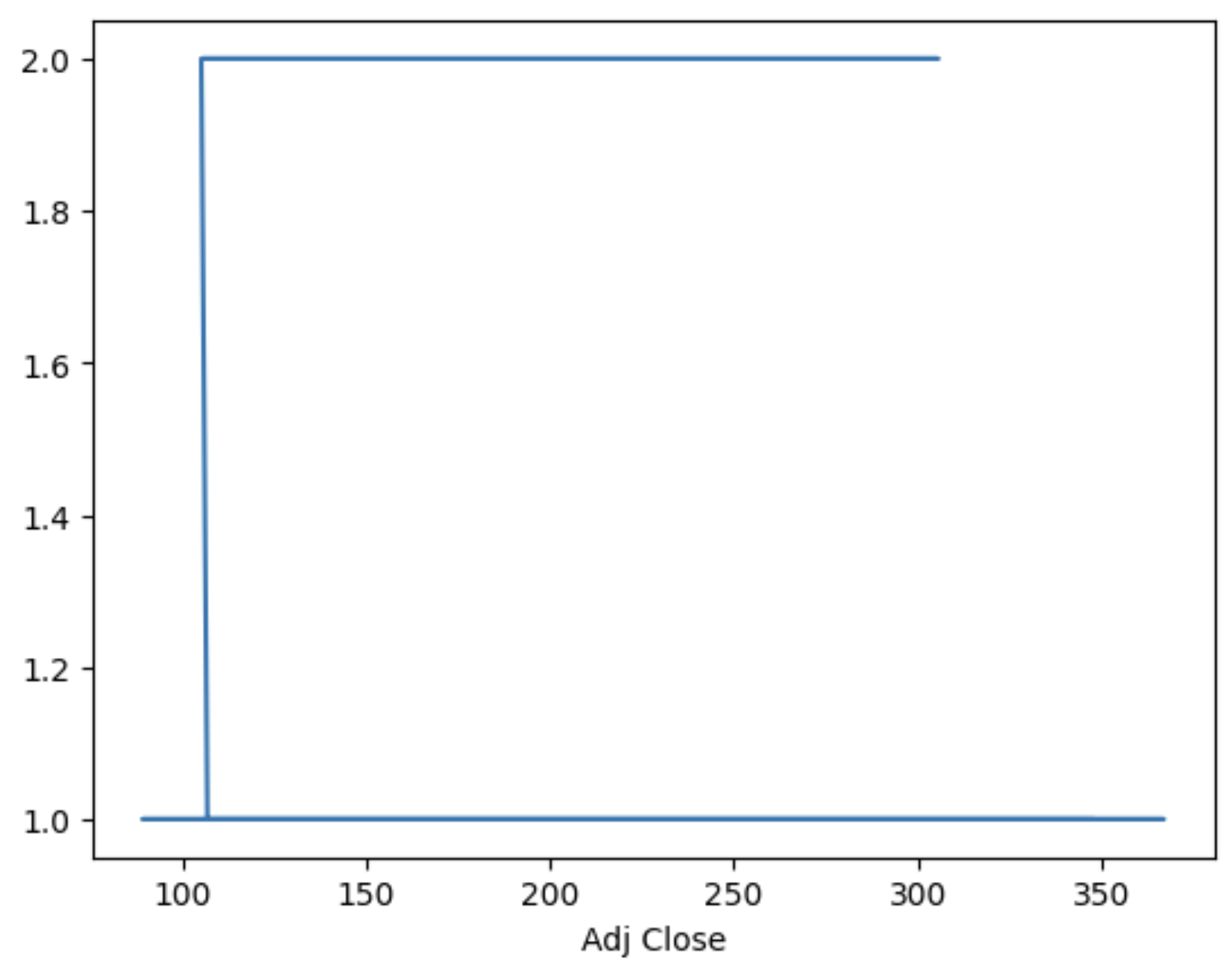
axs[row, col].legend(loc='right')

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

plt.tight\_layout()

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

plt.show()

Step 7: Box Plot for Seasonal Analysis

Create a box plot to analyze stock price variations across different months.

import seaborn as sns

# Extract year and month

df['Year'] = df['Date'].dt.year

df['Month'] = df['Date'].dt.month

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

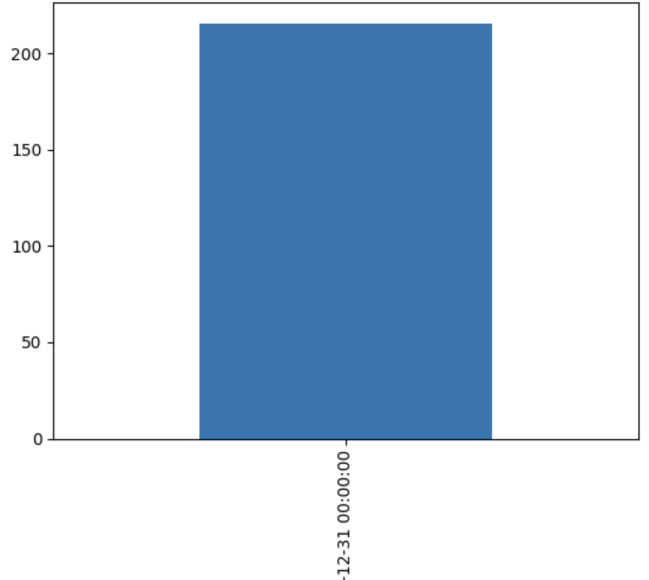
sns.boxplot(x='Month', y='Close', data=df)

plt.title('Monthly Distribution of Closing Prices')

plt.xlabel('Month')

plt.ylabel('Closing Price')

plt.show()

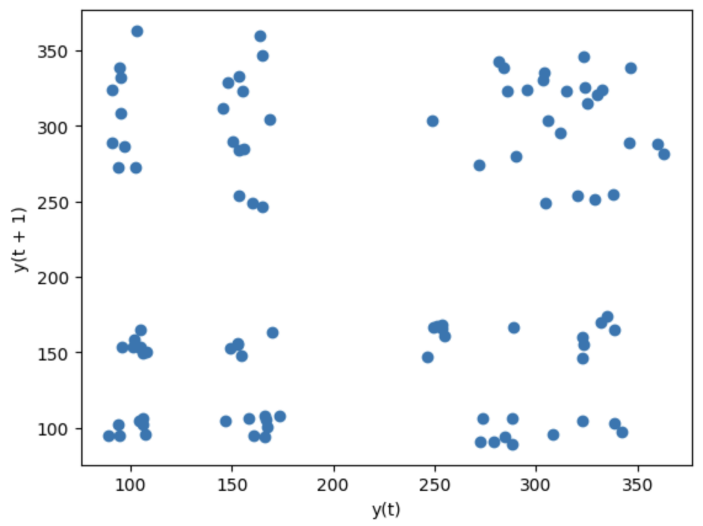


Step 8: Lag Plot for Trend Analysis

Lag plots help in identifying patterns and relationships in time series data.

from pandas.plotting import lag\_plot

lag\_plot(df['Close'].sample(100))

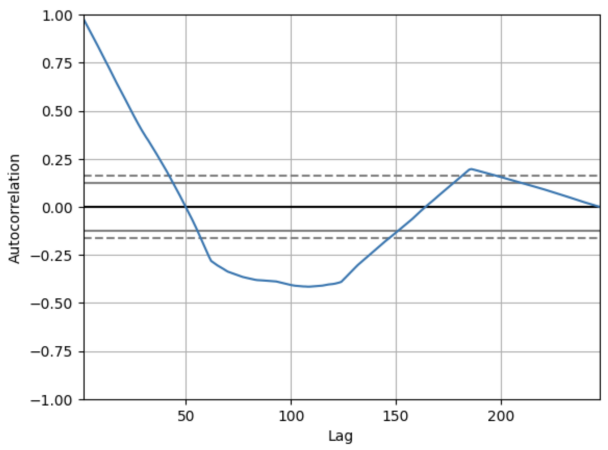


Step 9: Autocorrelation Analysis

Autocorrelation plots help determine how past values of a time series influence future values.

from pandas.plotting import autocorrelation\_plot

autocorrelation\_plot(df['Close'])



Step 10: Resampling for Trend Analysis

Aggregate data yearly to observe long-term trends in stock prices.

df['Close'].resample('Y').mean().plot.bar()

**Result:**

Thus, the time series dataset is successfully visualized, highlighting trends, seasonality, and patterns in stock prices over time.