

11. IMPLEMENT PROGRAMS FOR TIME SERIES DATA CLEANING, LOADING AND HANDLING TIMES SERIES DATA AND PRE-PROCESSING TECHNIQUES.

<u>EX.No: 01</u>	IMPLEMENT PROGRAMS FOR TIME SERIES DATA CLEANING, LOADING AND HANDLING TIMES SERIES DATA AND PRE-PROCESSING TECHNIQUES.
<u>DATE : 25/01/2025</u>	

AIM:

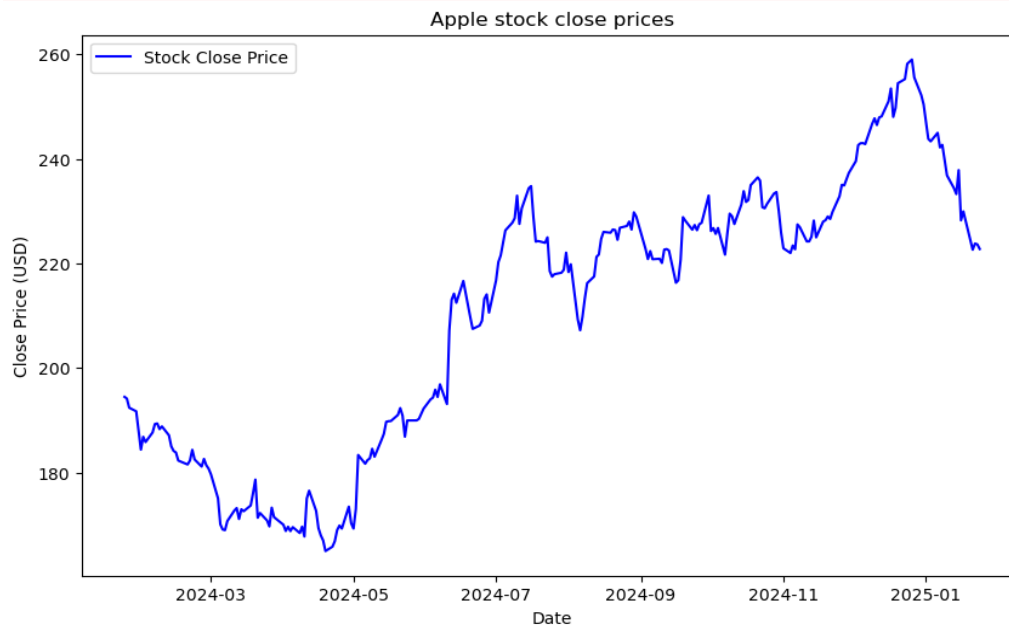
To Implement programs for time series data cleaning, loading and handling times series data and pre-processing techniques.

PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
def load_data(file_path):
    df = pd.read_csv(file_path, parse_dates=['Date'])
    df.set_index('Date', inplace=True)
    return df
def clean_data(df):
    df.replace(',', '', regex=True, inplace=True) # Remove commas from numeric values
    df = df.astype(float) # Convert all values to float
    df = df.dropna() # Drop missing values
    df = df[df > 0] # Remove negative or zero values if present
    return df
def handle_missing_data(df):
    df.fillna(method='ffill', inplace=True) # Forward fill
    df.fillna(method='bfill', inplace=True) # Backward fill
    return df
def preprocess_data(df):
    df['Return'] = df['Close'].pct_change() # Calculate daily returns
    df['Volatility'] = df['Return'].rolling(window=5).std() # Rolling volatility
    return df
def visualize_data(df):
```

```
plt.figure(figsize=(10, 6))
plt.plot(df['Close'], label='Stock Close Price', color='blue')
plt.title('Stock Close Price Over Time')
plt.xlabel('Date')
plt.ylabel('Close Price (USD)')
plt.legend()
plt.show()
file_path = "D:/221501511/Download Data - STOCK_US_XNAS_AAPL (1).csv"
df = load_data(file_path)
df = clean_data(df)
df = handle_missing_data(df)
df = preprocess_data(df)
visualize_data(df)
print(df.head())
```

OUTPUT:



RESULT: Thus, the program for Implement programs for time series data cleaning, loading and handling times series data and pre-processing techniques is executed successfully.

11. Implement programs for visualizing time series data

EX.N0 : 02	Implement programs for visualizing time series data
<u>DATE : 01/05/2024</u>	

AIM:

To Implement programs for visualizing time series data

PROGRAM:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm

# Step 1: Load dataset
def load_data(file_path):
    df = pd.read_csv(file_path, parse_dates=['Date'])
    df.set_index('Date', inplace=True)
    return df

# Step 2: Scatter Plot - Visualizing stock closing price over time
def scatter_plot(df):
    plt.figure(figsize=(10, 6))
    plt.scatter(df.index, df['Close'], label='Stock Close Price', color='blue', alpha=0.5)
    plt.title('Stock Close Price Over Time (Scatter Plot)')
    plt.xlabel('Date')
    plt.ylabel('Close Price (USD)')
    plt.legend()
    plt.show()

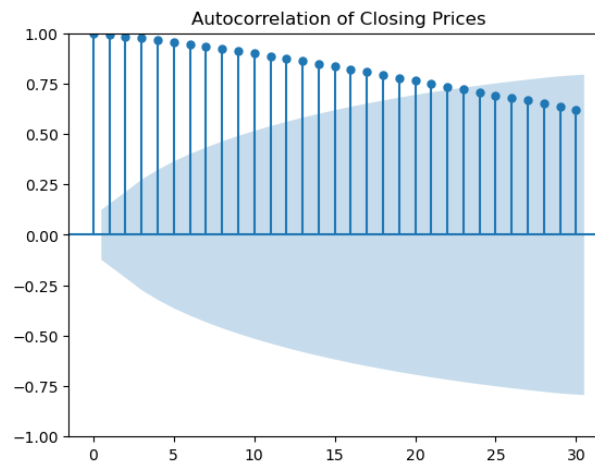
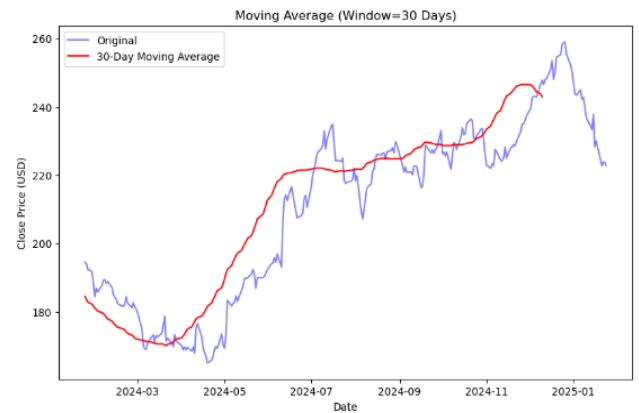
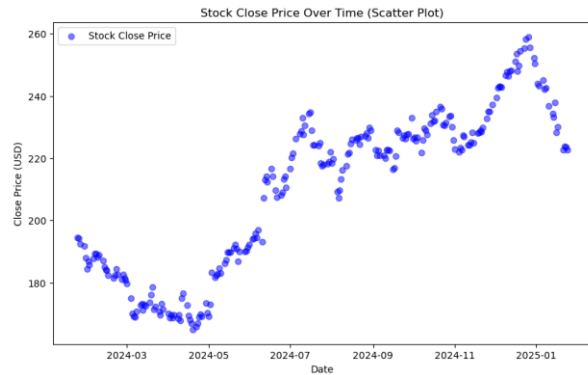
# Step 3: Histogram - Distribution of Closing Prices
def histogram(df):
    plt.figure(figsize=(10, 6))
    sns.histplot(df['Close'], bins=30, kde=True, color='blue')
    plt.title('Distribution of Closing Prices')
```

```

plt.xlabel('Close Price (USD)')
plt.ylabel('Frequency')
plt.show()
# Step 4: Box Plot - Outlier Detection in Closing Prices
def box_plot(df):
    plt.figure(figsize=(8, 6))
    sns.boxplot(y=df['Close'], color='blue')
    plt.title('Box Plot of Closing Prices')
    plt.ylabel('Close Price (USD)')
    plt.show()
# Step 5: Moving Average Plot
def moving_average_plot(df, window=30):
    df['Moving_Avg'] = df['Close'].rolling(window=window).mean()
    plt.figure(figsize=(10, 6))
    plt.plot(df['Close'], label='Original', color='blue', alpha=0.5)
    plt.plot(df['Moving_Avg'], label=f'{window}-Day Moving Average', color='red')
    plt.title(f'Moving Average (Window={window} Days)')
    plt.xlabel('Date')
    plt.ylabel('Close Price (USD)')
    plt.legend()
    plt.show()
# Step 6: Autocorrelation Plot
def autocorrelation_plot(df):
    sm.graphics.tsa.plot_acf(df['Close'].dropna(), lags=30)
    plt.title('Autocorrelation of Closing Prices')
    plt.show()
file_path = "D:/221501511/Download Data - STOCK_US_XNAS_AAPL (1).csv"
df = load_data(file_path)
scatter_plot(df)
histogram(df)
box_plot(df)
moving_average_plot(df)
autocorrelation_plot(df)

```

OUTPUT:



RESULT: Thus, the program for Implement programs for time series data cleaning, loading and handling times series data and pre-processing techniques is executed successfully.