EX:No.1 DATE: 25/01/2	**Implement Programs For Time Series Data Cleaning, Loading, And Handling Time Series Data And Pre-Processing Techniques**

AIM:

To clean, preprocess, and visualize stock data, focusing on trend analysis and handling missing values.

ALGORITHM:

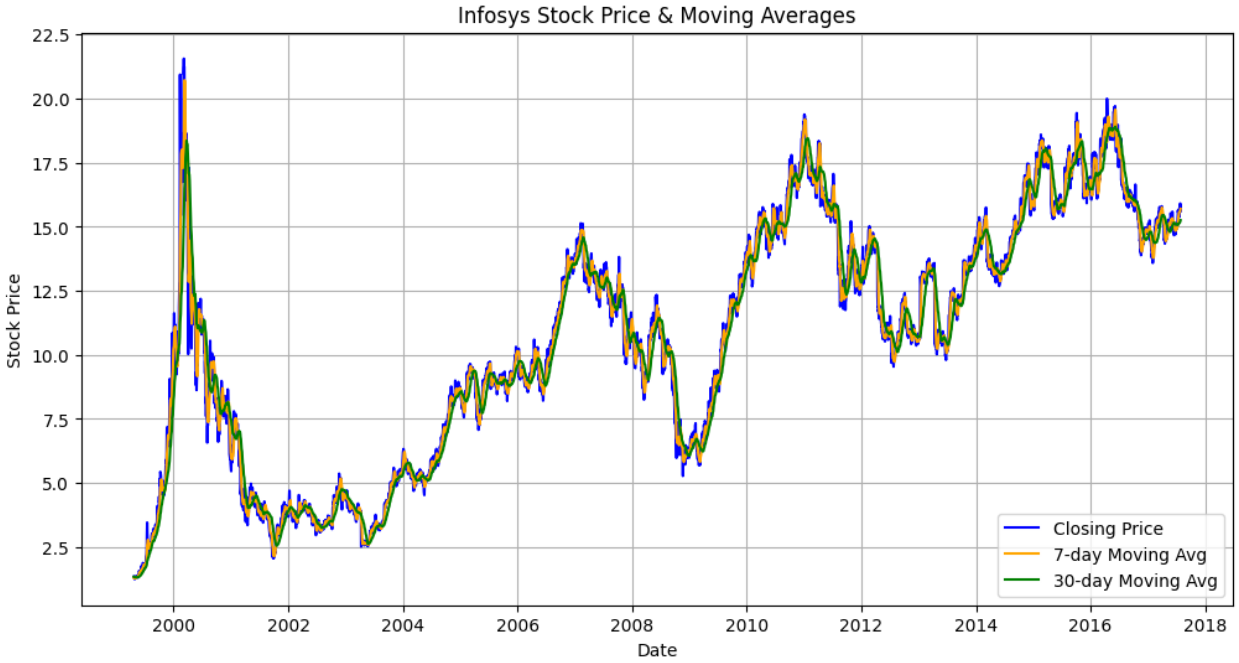
1. Load the stock data from the CSV file.
2. Parse the date column and set it as the index.
3. Handle missing values by filling them with forward fill.
4. Convert columns like Open, Close, Volume to numeric values.
5. Compute moving averages (7-day and 30-day) for trend analysis.
6. Drop any rows with NaN values created during moving average computation.
7. Visualize the closing price along with the moving averages using a line plot.

CODE:

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

# Load the dataset
df = pd.read_csv("infy_stock.csv", parse_dates=["Date"], index_col="Date")

# Display basic info
print(df.info())
print(df.head())

# Check for missing values
print("\nMissing Values:\n", df.isnull().sum())

# Fill missing values (if any) using forward fill method
df.fillna(method='ffill', inplace=True)
```

```
# Ensure data types are correct
df["Open"] = pd.to_numeric(df["Open"], errors="coerce")
df["High"] = pd.to_numeric(df["High"], errors="coerce")
df["Low"] = pd.to_numeric(df["Low"], errors="coerce")
df["Close"] = pd.to_numeric(df["Close"], errors="coerce")
df["Volume"] = pd.to_numeric(df["Volume"], errors="coerce")

# Create Moving Averages (7-day & 30-day)
df["7-day MA"] = df["Close"].rolling(window=7).mean()
df["30-day MA"] = df["Close"].rolling(window=30).mean()

# Drop any remaining NaN values caused by rolling averages
df.dropna(inplace=True)

# Visualization: Plot Closing Price with Moving Averages
plt.figure(figsize=(12,6))
sns.lineplot(x=df.index, y=df["Close"], label="Closing Price", color="blue")
sns.lineplot(x=df.index, y=df["7-day MA"], label="7-day Moving Avg", color="orange")
sns.lineplot(x=df.index, y=df["30-day MA"], label="30-day Moving Avg", color="green")

plt.title("Infosys Stock Price & Moving Averages")
plt.xlabel("Date")


plt.ylabel("Stock Price")
plt.legend()
plt.grid()
plt.show()
```

OUTPUT:



RESULT:

Thus the program has been completed and verified successfully.