EX:No.2 DATE:1/02/25	**Implement programs for visualizing time series data.**

AIM:

To Implement programs for visualizing time series data.

OBJECTIVE:

To analyze and visualize air pollution trends from 2012 to 2021 using multiple time-series plots.

BACKGROUND:

- Load, clean, and analyze supermarket sales data (2012-2021).
- Handle missing values and outliers for better accuracy.
- Identify sales trends over time.
- Visualize sales performance using graphs and time-series plots.

SCOPE OF THE PROGRAM:

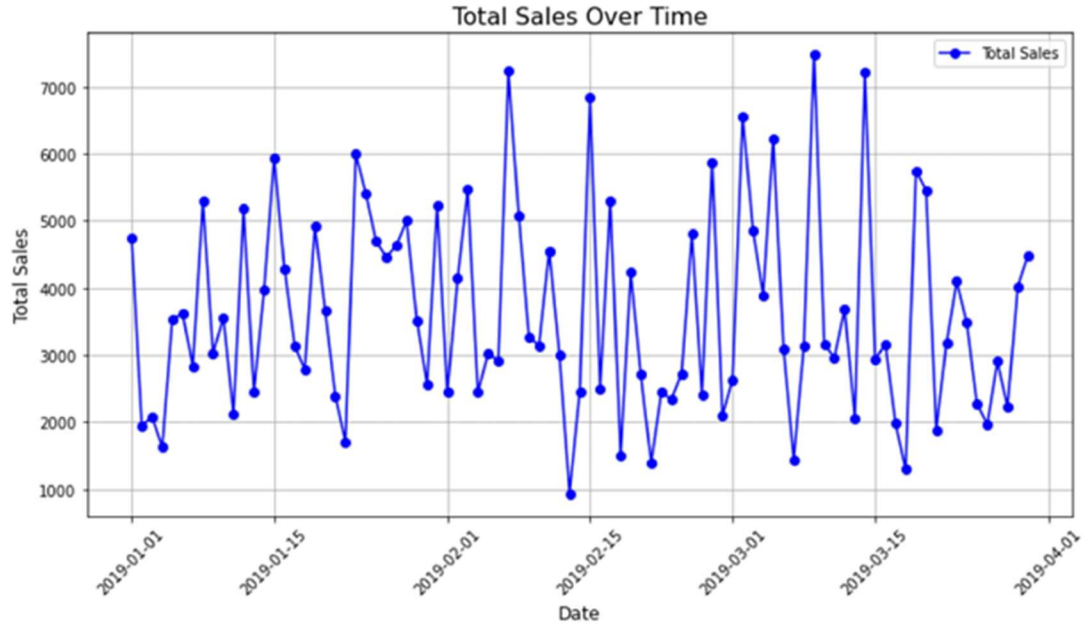
- Sales data analysis helps businesses understand performance and customer trends
- Key factors include revenue, customer purchases, seasonal trends, and promotions.
- Analyzing historical sales data aids in demand forecasting and business growth strategies.

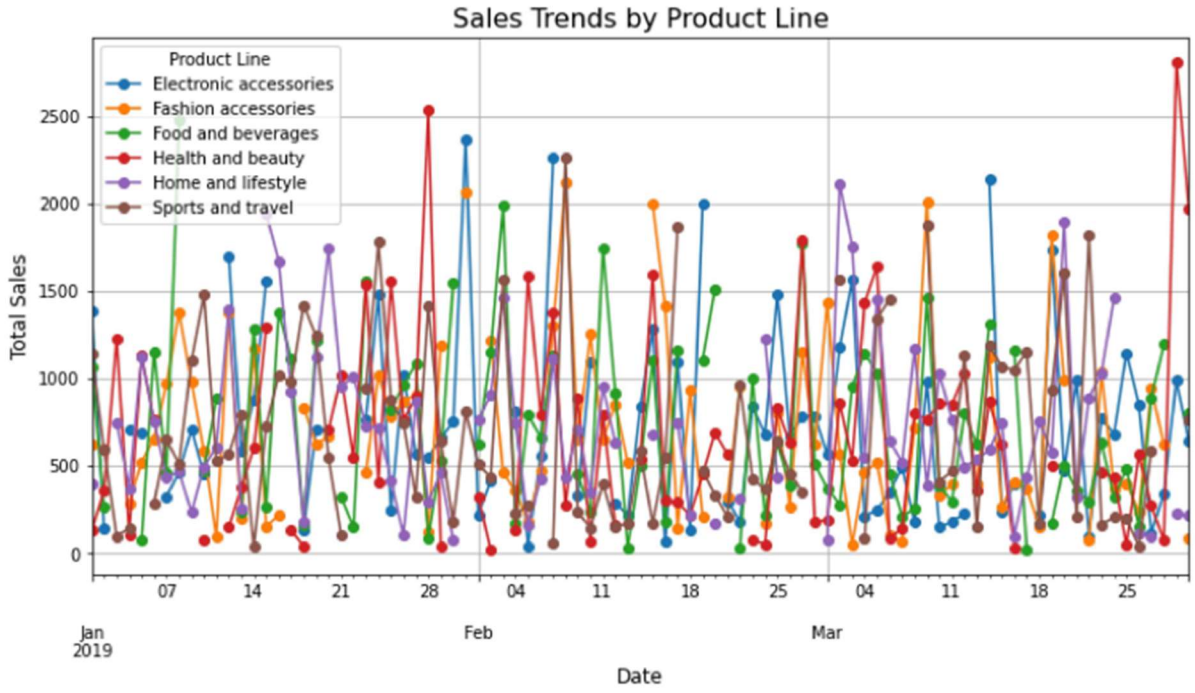
CODE:

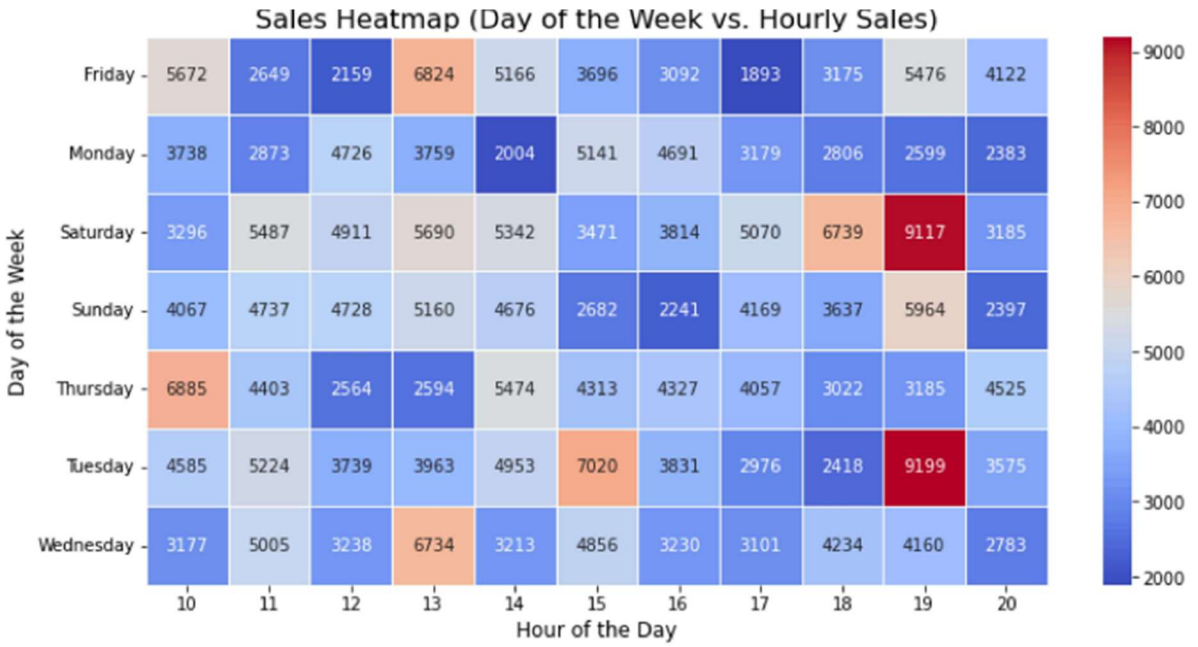
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

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

# Convert 'Date' column to datetime format
df['Date'] = pd.to_datetime(df['Date'], errors='coerce')

# Select the pollution column (update the name if different)
pollution_col = "PM2.5 (µg/m³)"  # Update based on actual column name

# Filter data for 2012-2021
df = df[(df['Date'].dt.year >= 2012) & (df['Date'].dt.year <= 2021)]

 # Handle missing values
df.fillna(method="ffill",
inplace=True)
df.fillna(method="bfill",
inplace=True)
```

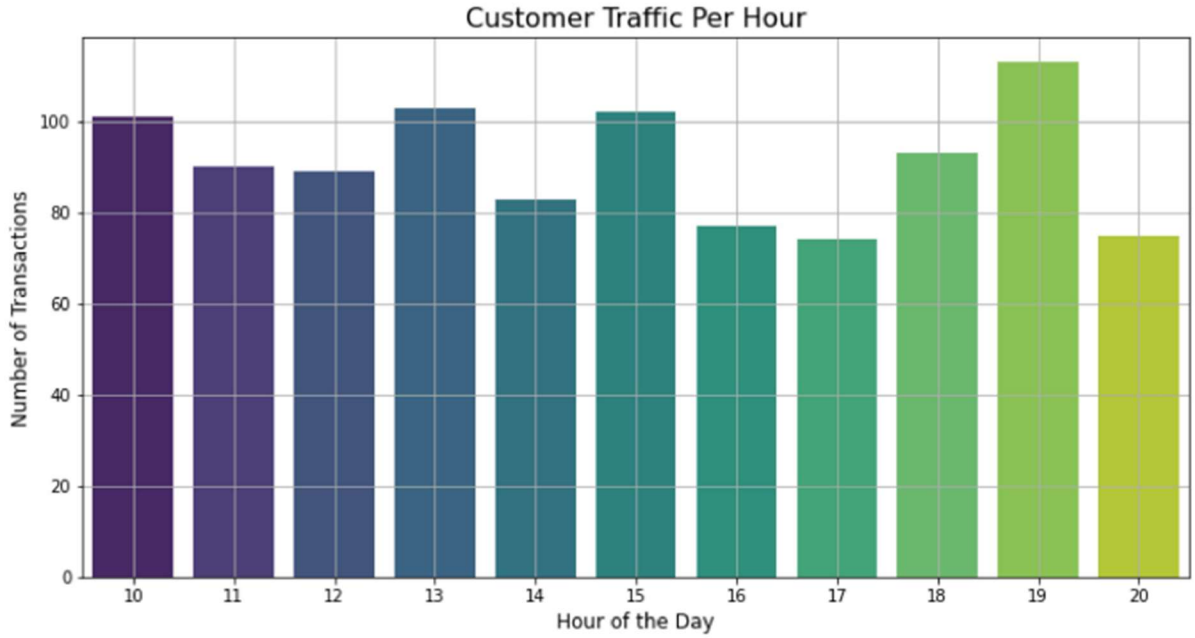
```
IQR = Q3 - Q1
df = df[(df[pollution_col] >= (Q1 - 1.5 * IQR)) & (df[pollution_col] <= (Q3 + 1.5 * IQR))]

# Select relevant column
df = df[['Total']]

# Remove outliers using IQR
method
Q1 = df.quantile(0.25)
Q3 = df.quantile(0.75)
IQR = Q3 - Q1
df = df[~((df < (Q1 - 1.5 * IQR)) |
(df > (Q3 + 1.5 *
IQR))).any(axis=1)]

# Plot 1: Line Plot
plt.figure(figsize=(10, 5))
plt.plot(df.index") plt.xlabel("Date")
plt.ylabel("Pollution Level")
plt.title("Line Plot - Air Pollution Over Time")
plt.legend()
plt.show()

# Plot 3: Area Chart
plt.figure(figsize=(10, 5))
plt.fill_between("Daily Sales", color="blue", alpha=0.6)
plt.xlabel("Date")
df = df[~((df < (Q1 - 1.5 * IQR)) | (df > (Q3 + 1.5 * IQR))).any(axis=1)]
plt.title("Area Chart - Air Pollution Over Time")
plt.show()

# Plot 4: Bar Chart (Monthly Average)
df['Month'] = df.index.month
monthly_avg = df.groupby("Month")[pollution_col].mean()
monthly_avg.plot(kind="bar", color='purple', figsize=(10, 5))
plt.xlabel("Month")
plt.ylabel("Average Pollution Level")
plt.title(""Supermarket Sales Performance Over Time (Cleaned)")
plt.show()

# Plot 5: Box Plot (Pollution Distribution)
plt.figure(figsize=(8, 5))
sns.boxplot(x=df.index.year, y=df[pollution_col], palette="coolwarm")
plt.xlabel("Year")
plt.ylabel("Pollution Level")
plt.title("Box Plot - Pollution Distribution by Year")
plt.xticks(rotation=45)
plt.show()
```

OUTPUT:


Total Sales Over Time
Total Sales
Total Sales
7000
6000
5000
4000
3000
2000
1000
2019-01-01
2019-01-15
2019-02-01
2019-02-15
2019-03-01
2019-03-15
2019-04-01
Date


Sales Trends by Product Line
Product Line
Electronic accessories
Fashion accessories
Food and beverages
Health and beauty
Home and lifestyle
Sports and travel
Total Sales
2500
2000
1500
1000
500
0
07
14
21
28
04
11
18
25
04
11
18
25
Jan
2019
Feb
Mar
Date





RESULT:

Thus, the program using the time series data implementation has been done successfully.