**Implement Program For Visualization OF Time Series Data**

**EX.No:2**

**DATE: 25/01/2**

**AIM:**

To analyze and visualize electricity production data using various statistical and graphical techniques, including histograms, box plots, scatter plots, bar charts, and heatmaps, for trend analysis and outlier detection.

**ALGORITHM:**

1. Load the electricity production data from a CSV file.
2. Convert the date column to datetime format and set it as the index.
3. Convert numerical columns to appropriate data types.
4. Handle missing values using forward fill.
5. Plot the histogram to visualize the distribution of electricity production.
6. Generate a box plot to detect outliers in the data.
7. Create a scatter plot to analyze electricity production trends over time.
8. Compute and visualize the monthly average electricity production using a bar chart.
9. Generate a heatmap to analyze electricity production trends across different months and years.
10. Analyze and interpret trends, patterns, and anomalies in the visualizations.

**CODE:**

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

def load\_and\_clean\_data(file\_path):

"""Loads and cleans the dataset."""

df = pd.read\_csv(file\_path)

# Convert 'month' to datetime format

df['month'] = pd.to\_datetime(df['month'], format='%d-%b', errors='coerce')

# Drop rows with missing 'month' or 'price'

df = df.dropna(subset=['month', 'price'])

df = df.sort\_values('month')

return df

def plot\_line\_chart(df):

"""Plots a time series line chart."""

plt.figure(figsize=(10, 5))

plt.plot(df['month'], df['price'], marker='o', linestyle='-', color='b', label='Price')

plt.xlabel("Month")

plt.ylabel("Price")

plt.title("Time Series Visualization of Price Over Months")

plt.xticks(rotation=45)

plt.legend()

plt.grid(True)

plt.show()

def plot\_bar\_chart(df):

"""Plots a bar chart."""

plt.figure(figsize=(10, 5))

plt.bar(df['month'], df['price'], color='skyblue')

plt.xlabel("Month")

plt.ylabel("Price")

plt.title("Monthly Shampoo Sales (Bar Chart)")

plt.xticks(rotation=45)

plt.show()

def plot\_scatter\_chart(df):

"""Plots a scatter plot."""

plt.figure(figsize=(10, 5))

plt.scatter(df['month'], df['price'], color='red', label='Price')

plt.xlabel("Month")

plt.ylabel("Price")

plt.title("Shampoo Sales (Scatter Plot)")

plt.xticks(rotation=45)

plt.legend()

plt.grid(True)

plt.show()

def plot\_box\_plot(df):

"""Plots a box plot to show distribution and outliers."""

plt.figure(figsize=(6, 5))

plt.boxplot(df['price'], vert=False)

plt.xlabel("Price")

plt.title("Box Plot of Shampoo Sales Prices")

plt.show()

def plot\_histogram(df):

"""Plots a histogram to show frequency distribution."""

plt.figure(figsize=(8, 5))

plt.hist(df['price'], bins=10, color='green', edgecolor='black')

plt.xlabel("Price")

plt.ylabel("Frequency")

plt.title("Histogram of Shampoo Sales Prices")

plt.show()

def plot\_heatmap(df):

"""Plots a heatmap to show price trends over time."""

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

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

pivot\_table = df.pivot\_table(values='price', index='Year', columns='Month', aggfunc='mean')

plt.figure(figsize=(10, 6))

sns.heatmap(pivot\_table, cmap='coolwarm', annot=True, fmt=".1f", linewidths=0.5)

plt.xlabel("Month")

plt.ylabel("Year")

plt.title("Heatmap of Shampoo Sales Prices")

plt.show()

if \_\_name\_\_ == "\_\_main\_\_":

file\_path = r"D:\TSA\Exp-1\Dataset.csv" # Update with your file path

df = load\_and\_clean\_data(file\_path)

print("Loaded Data:")

print(df.head())

plot\_line\_chart(df)

plot\_bar\_chart(df)

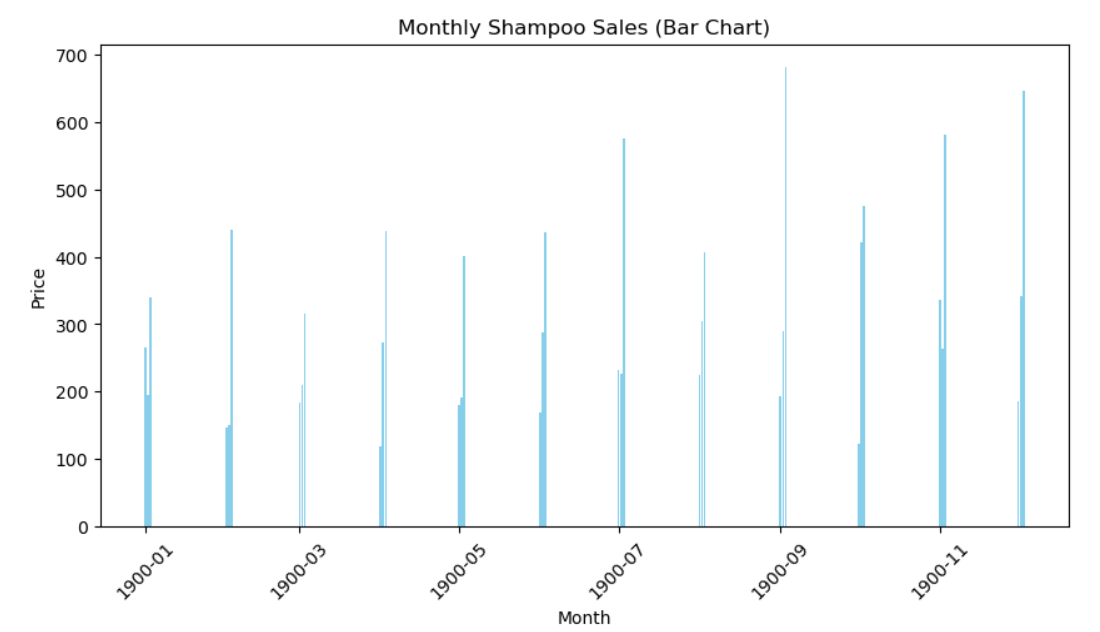
plot\_scatter\_chart(df)

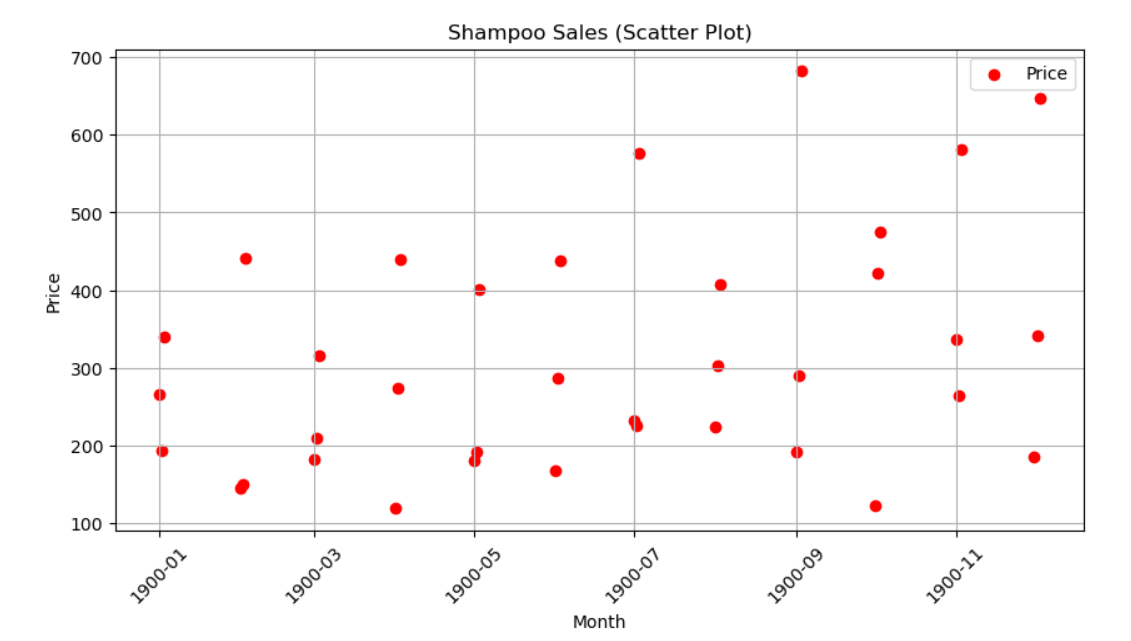
plot\_box\_plot(df)

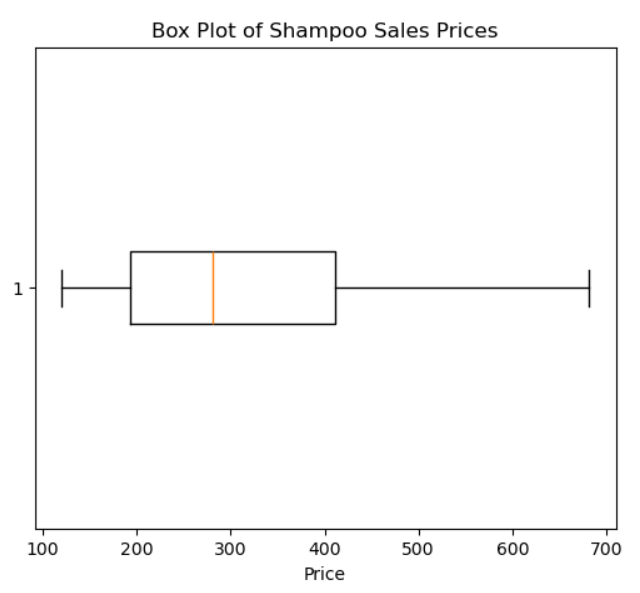
plot\_histogram(df)

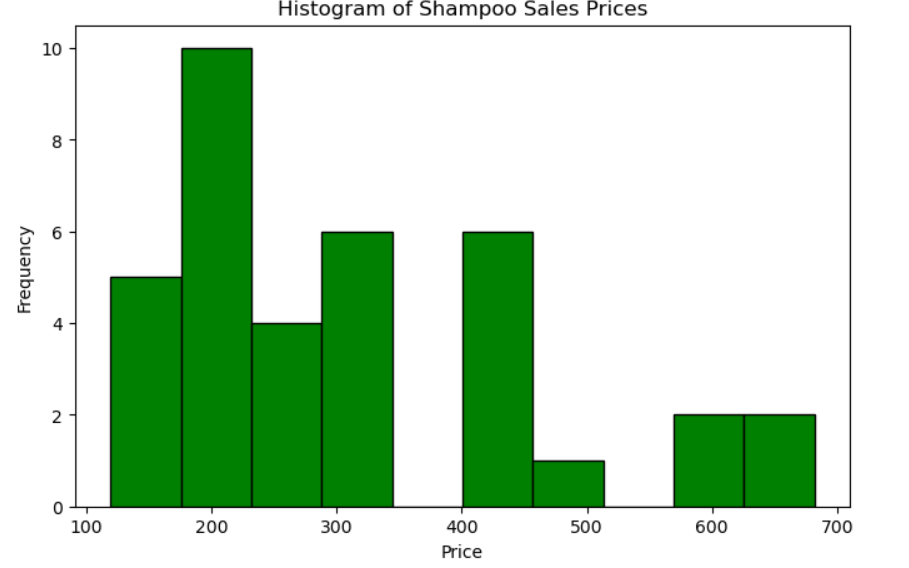
plot\_heatmap(df)

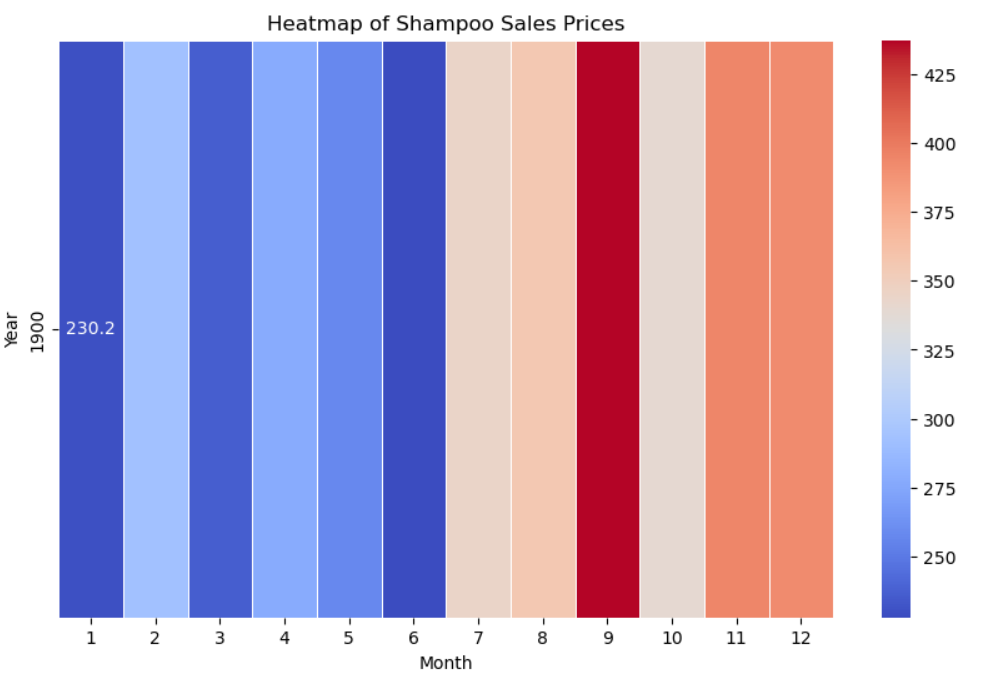
**OUTPUT:**

****

****

****

****

****

**RESULT:**

Thus the program has been completed and verified successfully.