EXP:2 30/01/2025	PERFORMING LINEAR REGRESSION ON THE TIME SERIES DATASET

AIM:

To implement a program for time series data cleaning,loading and handling time series data & preprocessing techniques.

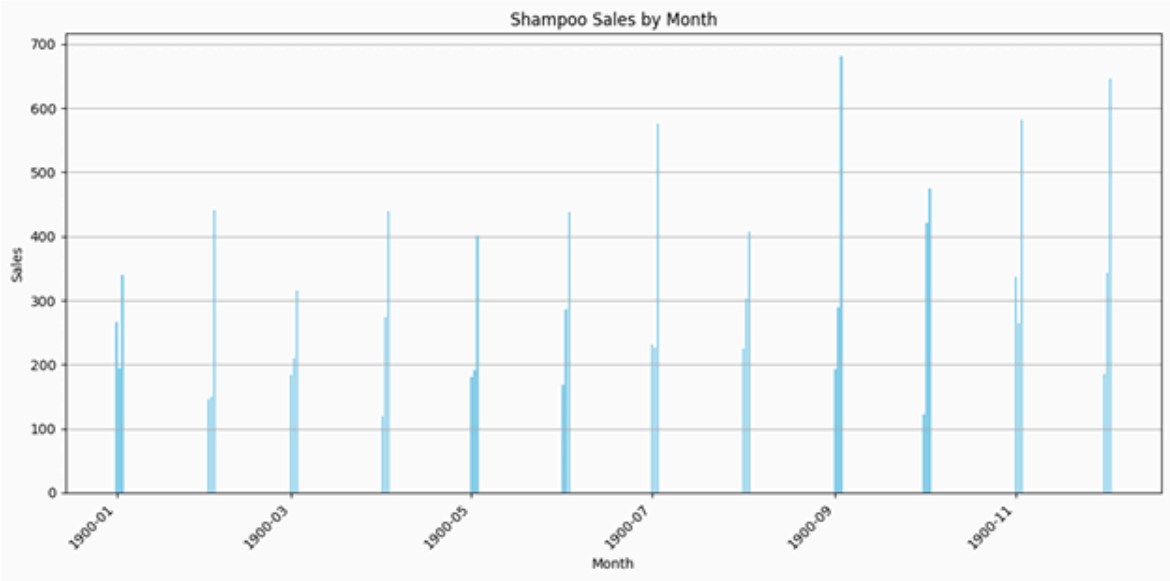
PROCEDURE:

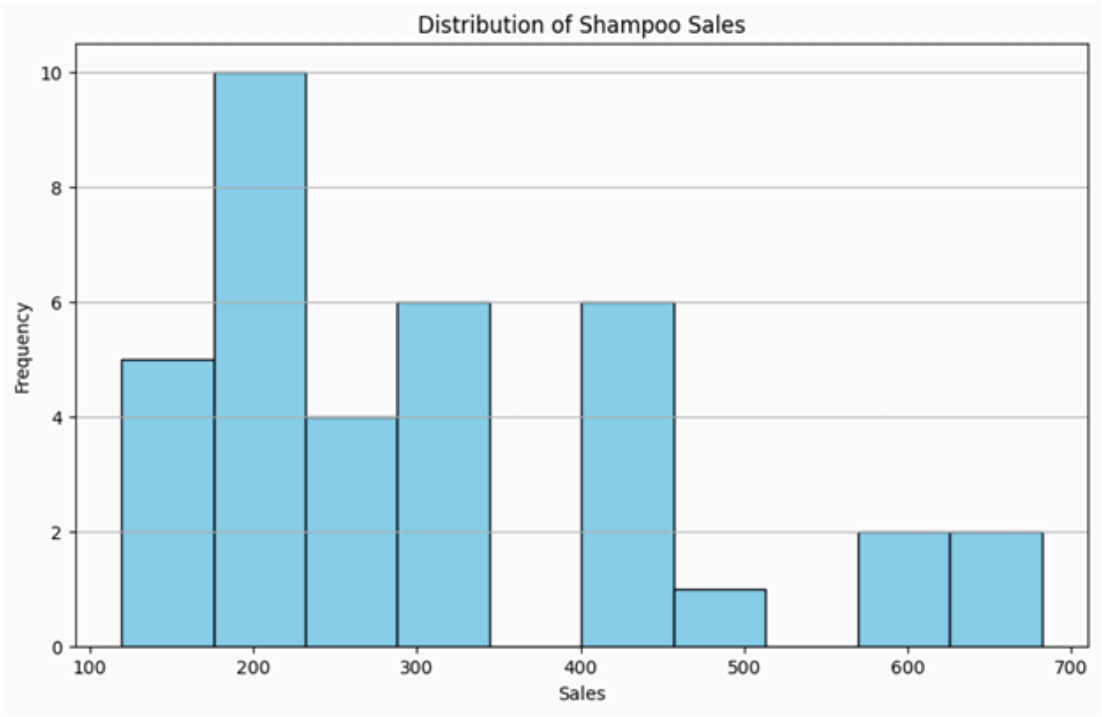
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import files
uploaded = files.upload("saleOfShampoo.csv")

file_name = list(uploaded.keys())[0]
shampoo_sales_data = pd.read_csv(file_name)

# Rename columns for easier handling
shampoo_sales_data.columns = ['Month', 'Sales']

# Convert 'Month' to a datetime object and handle missing or incorrect
data
shampoo_sales_data['Month'] = pd.to_datetime(shampoo_sales_data['Month'],
format='%d-%b', errors='coerce')
shampoo_sales_data['Sales'] = pd.to_numeric(shampoo_sales_data['Sales'],
errors='coerce')

# Drop rows with invalid data
shampoo_sales_data.dropna(inplace=True)

# Reset index
shampoo_sales_data = shampoo_sales_data.reset_index(drop=True)
```

```
import matplotlib.pyplot as plt
# Create a bar plot of sales for each month
plt.figure(figsize=(12, 6))
plt.bar(shampoo_sales_data['Month'], shampoo_sales_data['Sales'],
color='skyblue')
plt.title("Shampoo Sales by Month")
plt.xlabel("Month")
plt.ylabel("Sales")
plt.xticks(rotation=45, ha='right') # Rotate x-axis labels for readability
plt.grid(axis='y')
plt.tight_layout() # Adjust layout to prevent labels from overlapping
plt.show()


# Create a histogram of sales distribution
plt.figure(figsize=(10, 6))
plt.hist(shampoo_sales_data['Sales'], bins=10, color='skyblue',
edgecolor='black')
plt.title("Distribution of Shampoo Sales")
plt.xlabel("Sales")
plt.ylabel("Frequency")
plt.grid(axis='y')
plt.show()

#bar plot

plt.figure(figsize=(10, 5))
sns.boxplot(data=shampoo_sales_data, x='Sales', color='skyblue')
plt.title("Boxplot of Shampoo Sales")
plt.xlabel("Sales")
plt.grid(axis='y')
plt.show()



import matplotlib.pyplot as plt
```

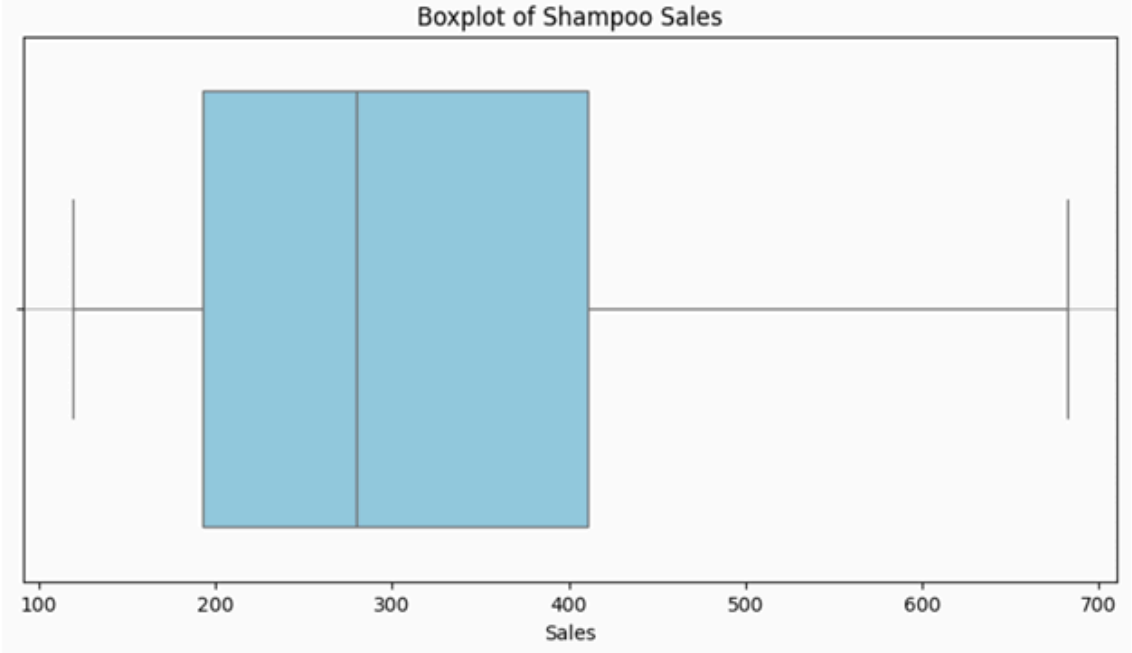
```
# Create a line plot
plt.figure(figsize=(10, 6))
plt.plot(shampoo_sales_data['Month'], shampoo_sales_data['Sales'],
linestyle='-', marker='o')
plt.title('Shampoo Sales Over Time (Line Plot)')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.grid(True)
plt.show()

# Create a scatter plot
plt.figure(figsize=(10, 6))
plt.scatter(shampoo_sales_data.index, shampoo_sales_data['Sales'],
color='green', marker='x') # Use index for x-axis
plt.title('Shampoo Sales (Scatter Plot)')
plt.xlabel('Data Point Index') # Changed x-axis label
plt.ylabel('Sales')
plt.grid(True)
plt.show()
```

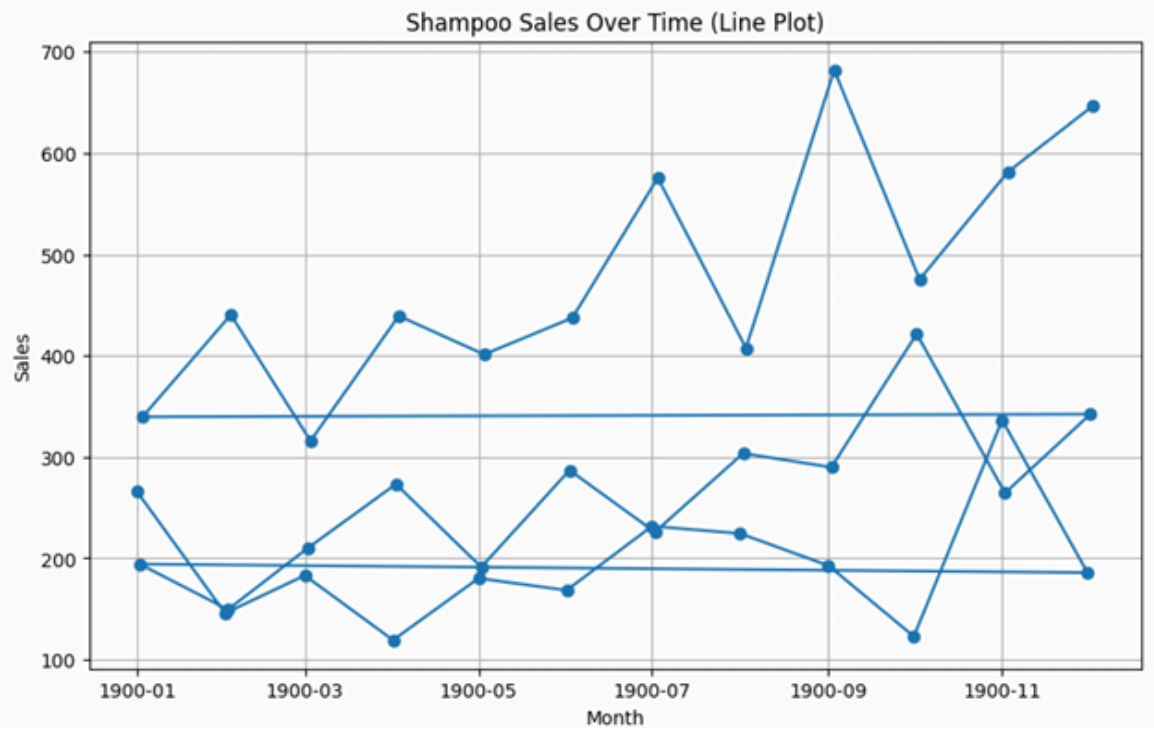
OUTPUT:

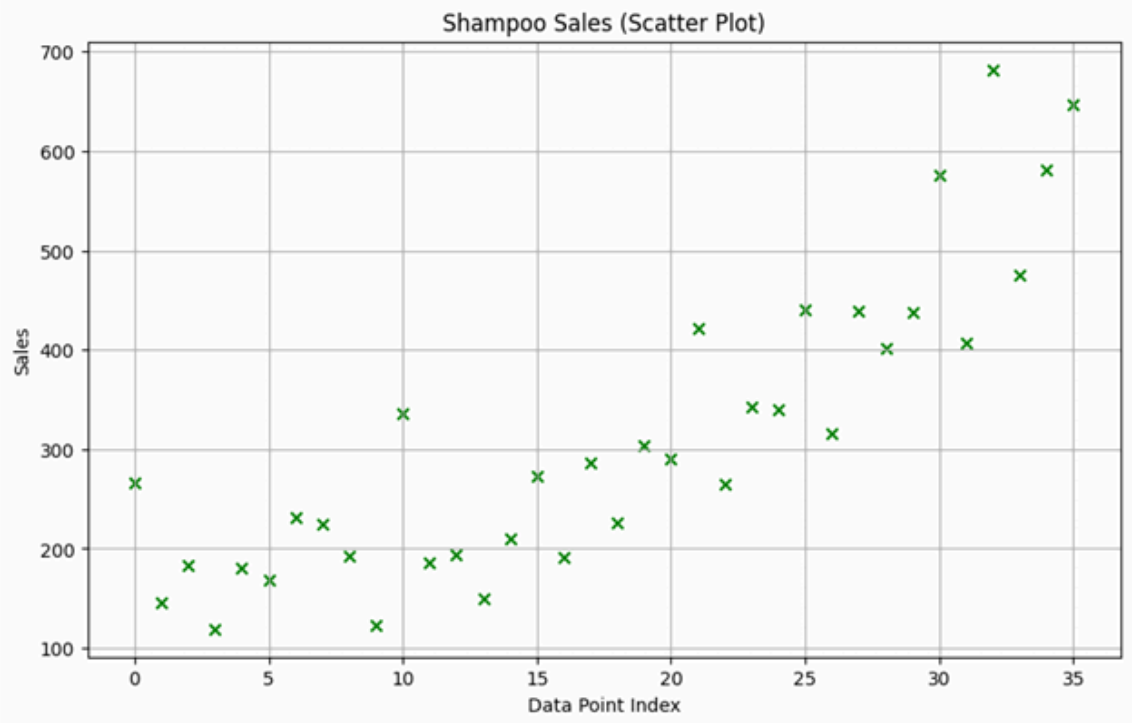



Distribution of Shampoo Sales
Frequency
Sales
0
2
4
6
8
10
100
200
300
400
500
600
700

Sales


Boxplot of Shampoo Sales
Sales
100
200
300
400
500
600
700


Shampoo Sales Over Time (Line Plot)
Sales
Month
700
600
500
400
300
200
100
1900-01
1900-03
1900-05
1900-07
1900-09
1900-11


Shampoo Sales (Scatter Plot)
Sales
Data Point Index
700
600
500
400
300
200
100
0
5
10
15
20
25
30
35

RESULT:

Thus the program has been executed successfully.