EXPERIMENT NO : 10

AIM: To create various visualizations such as histograms, bar charts, pie charts, box plots, violin plots, and regression plots using Matplotlib and Seaborn for effective data representation.

Introduction:

Data visualization is an essential part of data analysis that helps in understanding trends, patterns, and relationships within data. Python provides powerful libraries for visualization, primarily **Matplotlib** and **Seaborn**.

- **Matplotlib**: A fundamental plotting library that provides a variety of basic and advanced visualization options.
- **Seaborn**: Built on top of Matplotlib, it offers more aesthetically pleasing and informative statistical graphics.

Types of Visualizations:

Histogram:

- **Used to visualize the distribution of numerical data.**
- **It divides the data into bins and shows the frequency of each bin.**
- **Useful for identifying the shape of the distribution (normal, skewed, etc.).**

Bar Charts:

- Used to compare categorical data.

Pie Charts:

- **Purpose:** Used to show proportions of a whole.

Box Plots (Box-and-Whisker Plots):

- Displays the distribution of numerical data and highlights outliers.
- Shows the median, quartiles, and potential outliers.
- Helps in identifying skewness and variability.

Violin PlotsPurpose:

- Combines a box plot with a KDE plot to show the full data distribution.

CODE:

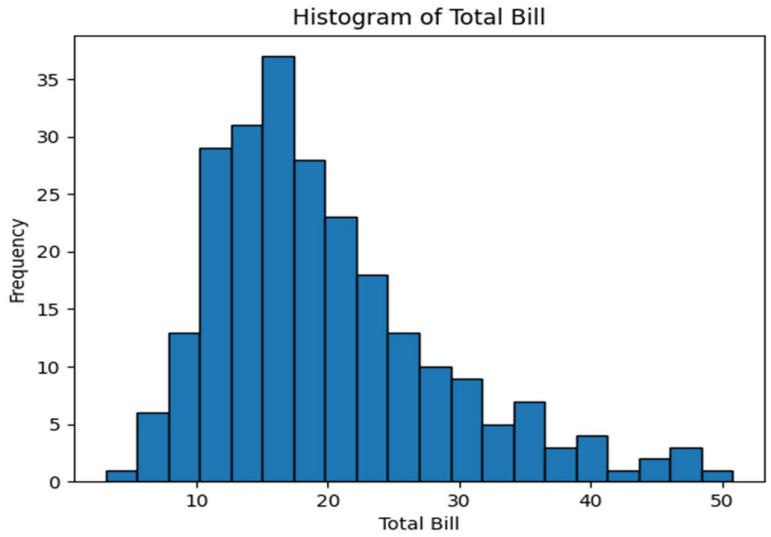
```
import matplotlib.pyplot as plt
```

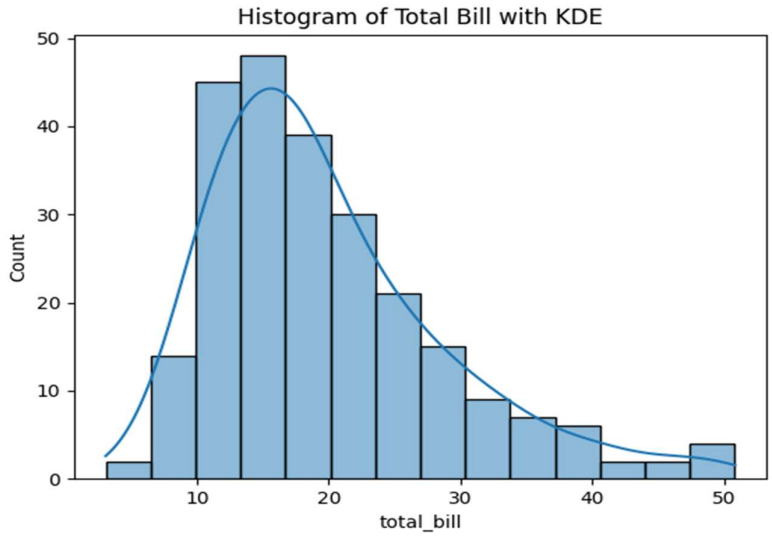
```
import seaborn as sns
import pandas as pd
import numpy as np
# Example dataset using seaborn's built-in dataset
df = sns.load_dataset('tips')
# Or create a random dataset for visualization purposes
np.random.seed(0)
data = np.random.randn(100)
# Matplotlib Histogram
plt.hist(df['total_bill'], bins=20, edgecolor='black')
plt.title('Histogram of Total Bill')
plt.xlabel('Total Bill')
plt.ylabel('Frequency')
plt.show()
# Seaborn Histogram
sns.histplot(df['total_bill'], kde=True)
plt.title('Histogram of Total Bill with KDE')
plt.show()
# Matplotlib Bar Chart
categories = df['day'].value_counts().index
counts = df['day'].value_counts()
plt.bar(categories, counts, color='skyblue')
plt.title('Bar Chart of Days')
plt.xlabel('Days')
plt.ylabel('Count')
```

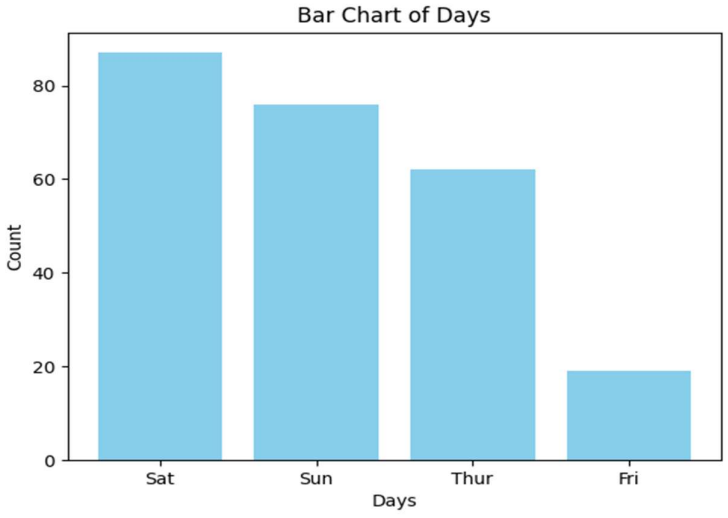
```
plt.show()

# Seaborn Bar Chart

sns.countplot(data=df, x='day', palette='coolwarm')

plt.title('Bar Chart of Days using Seaborn')

plt.show()

# Matplotlib Pie Chart

category_counts = df['day'].value_counts()

plt.pie(category_counts, labels=category_counts.index, autopct='%1.1f%%',
colors=sns.color_palette('pastel'))

plt.title('Pie Chart of Days')

plt.show()

# Matplotlib Box Plot

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

plt.title('Box Plot of Total Bill')

plt.xlabel('Total Bill')

plt.show()

# Seaborn Box Plot

sns.boxplot(data=df, x='total_bill', palette='Set2')

plt.title('Box Plot of Total Bill using Seaborn')

plt.show()

# Matplotlib Violin Plot (custom)

plt.violinplot(df['total_bill'])

plt.title('Violin Plot of Total Bill')

plt.xlabel('Total Bill')

plt.show()

# Seaborn Violin Plot
```

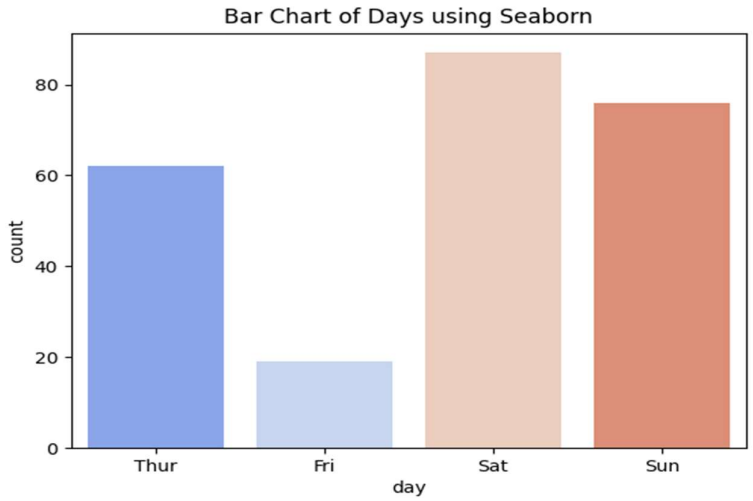
```
sns.violinplot(data=df, x='day', y='total_bill', palette='muted')
plt.title('Violin Plot of Total Bill by Day')
plt.show()
# Seaborn Regression Plot
sns.regplot(x='total_bill', y='tip', data=df, scatter_kws={'s': 50}, line_kws={'color': 'red'})
plt.title('Regression Plot of Total Bill vs Tip')
plt.show()
```

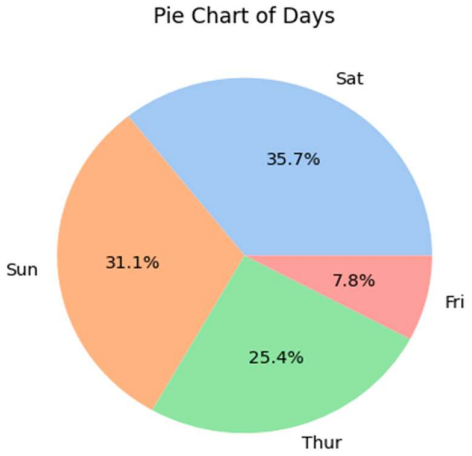
OUTPUT:

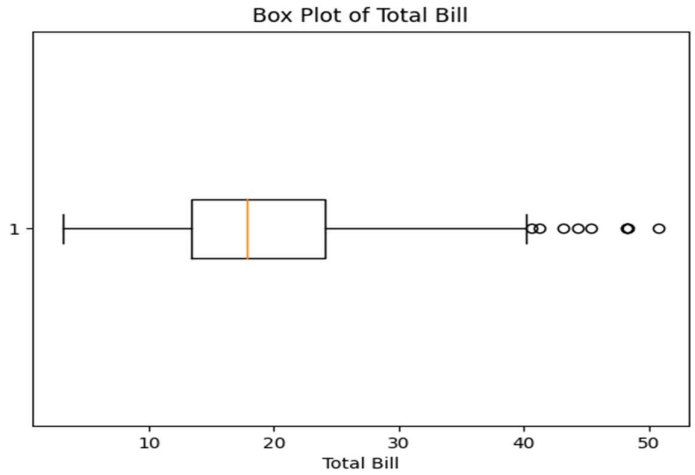


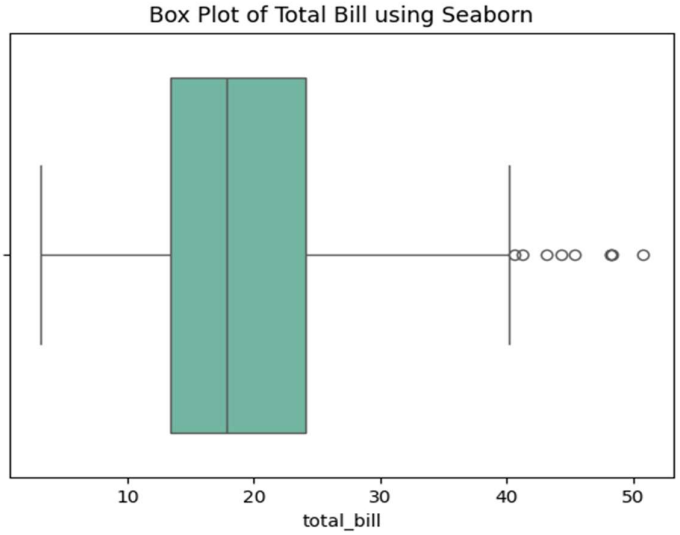



Bar Chart of Days
Count
0
20
40
60
80
Sat
Sun
Thur
Fri
Days

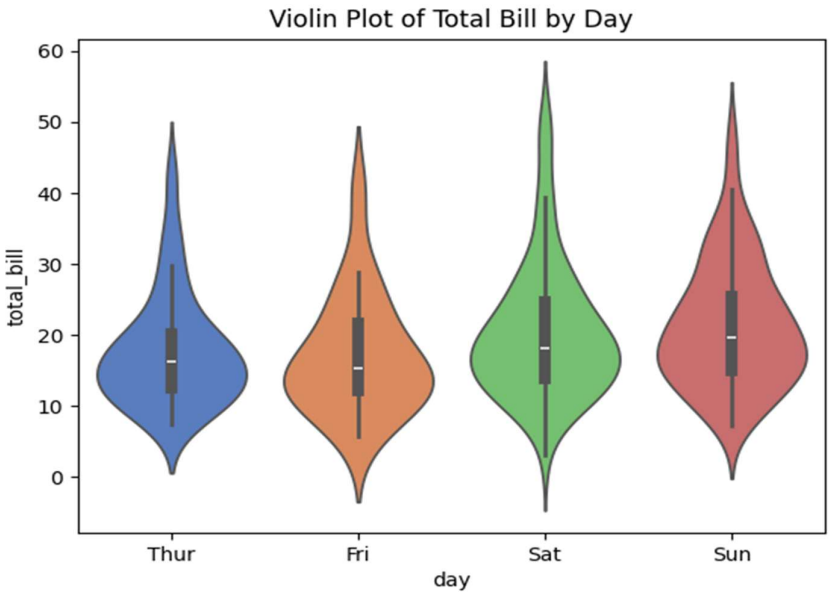

Bar Chart of Days using Seaborn
count
0
20
40
60
80
Thur
Fri
Sat
Sun
day

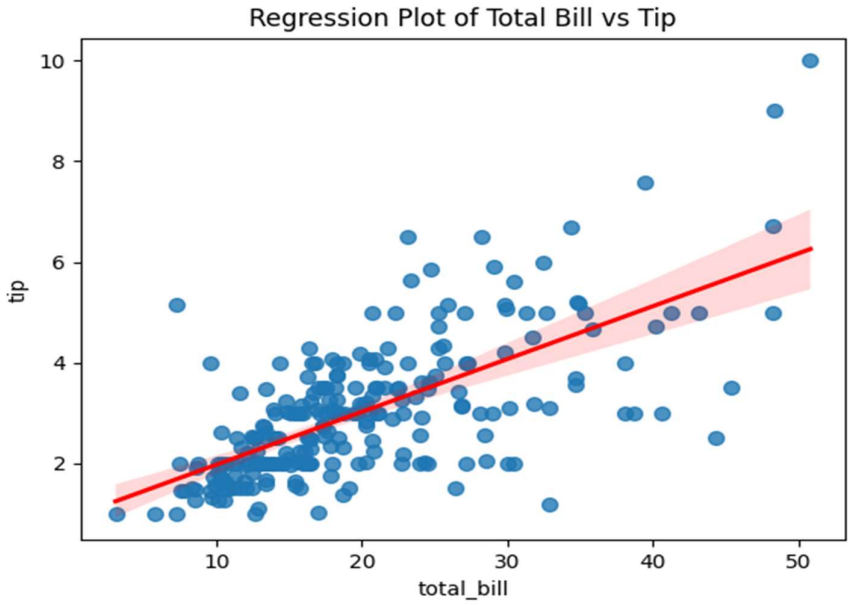

Pie Chart of Days
Sat
35.7%
Sun
31.1%
Fri
7.8%
25.4%
Thur


Box Plot of Total Bill
1
10
20
30
40
50
Total Bill


Box Plot of Total Bill using Seaborn
10
20
30
40
50
total_bill


Violin Plot of Total Bill
50
40
30
20
10
0.8
0.9
1.0
1.1
1.2
Total Bill





CONCLUSION: We successfully implemented various visualizations such as histograms, bar charts, pie charts, box plots, violin plots, and regression plots using Matplotlib and Seaborn for effective data representation.