1. Matplotlib

Matplotlib is a comprehensive library in Python used for data visualization. It offers a wide array of plotting functions for producing static, animated, and interactive visualizations. With its extensive customization options, users can create line plots, scatter plots, histograms, and much more. Being open-source and community-driven, it integrates well with many other Python data science libraries.

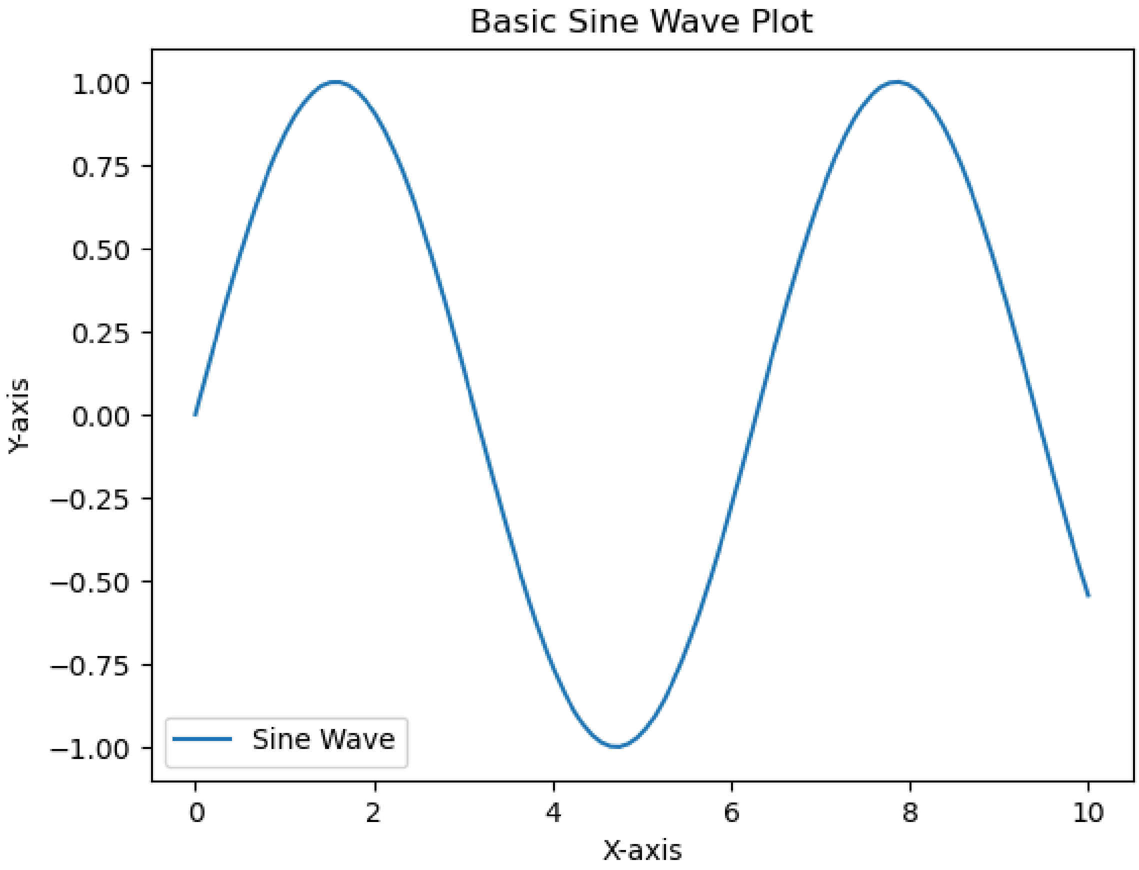
Importing Libraries

In [1]:
```
import numpy as np
import matplotlib.pyplot as plt
```

Basic Plotting

In [2]:

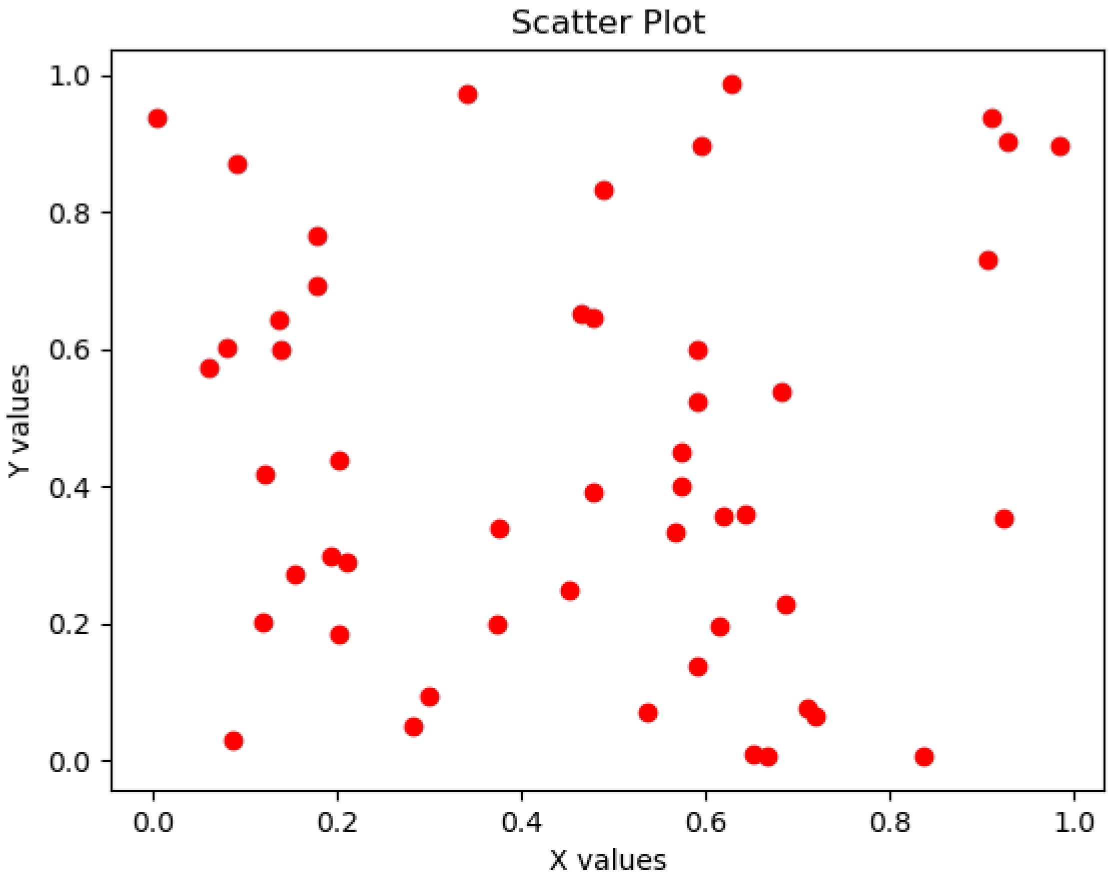
```
# Generate sample data
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Plot the data
plt.plot(x, y, label='Sine Wave')
plt.title('Basic Sine Wave Plot')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.legend()
plt.show()
```



Scatter Plot

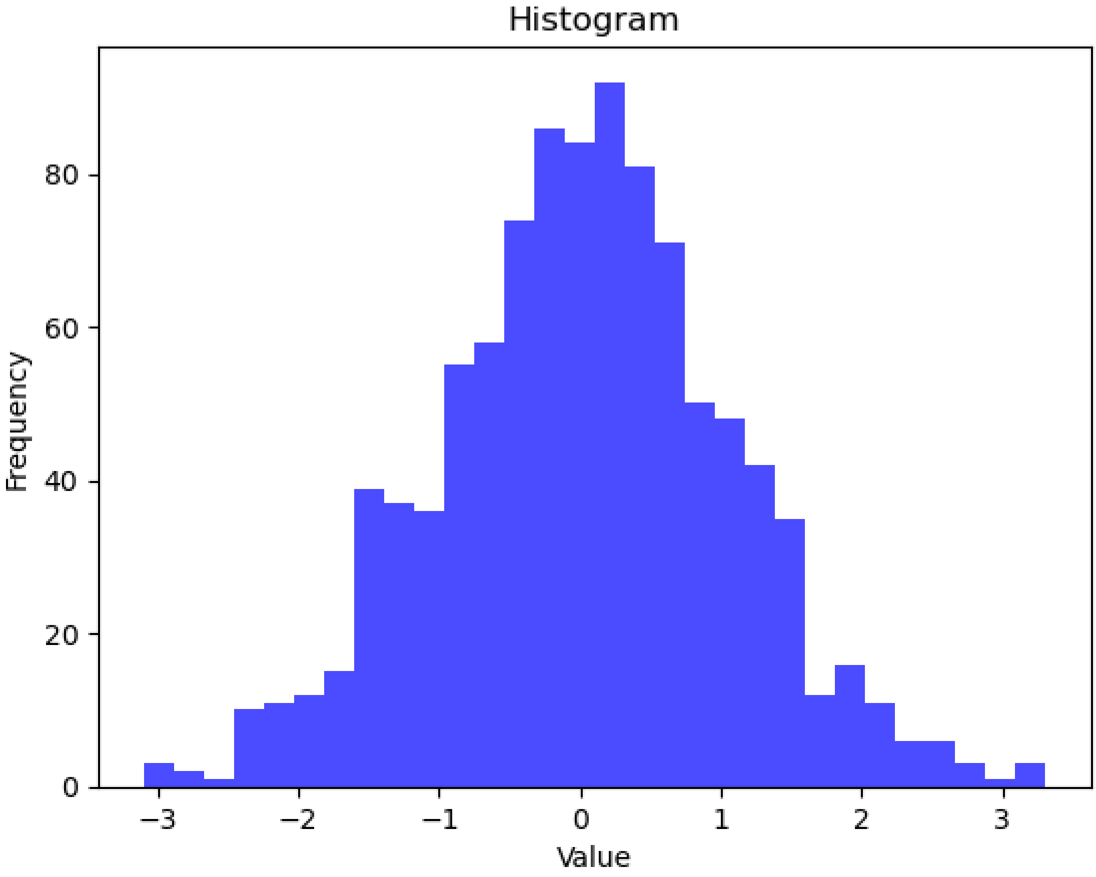
In [3]:

```
# Generate random data for scatter plot
x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y, color='red', marker='o')
plt.title('Scatter Plot')
plt.xlabel('X values')
plt.ylabel('Y values')
plt.show()
```



Histogram

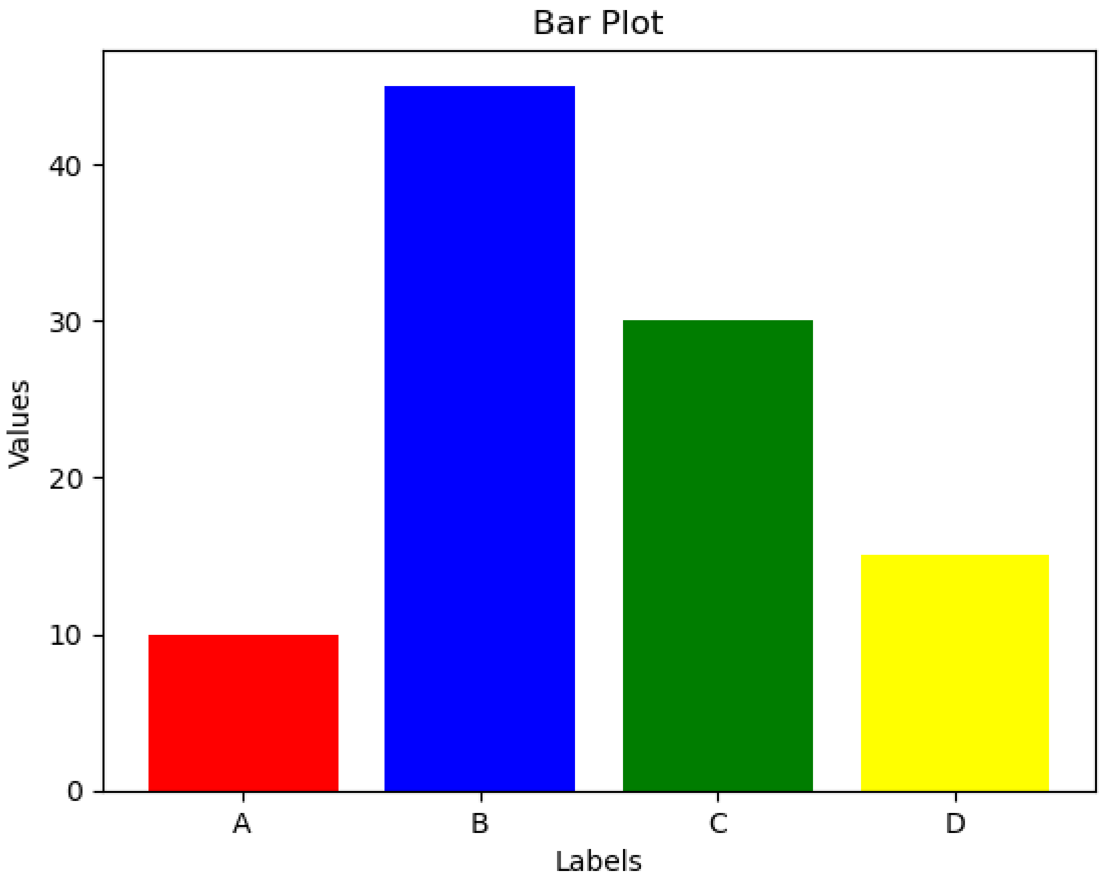
In [4]:
```
# Generating random data
data = np.random.randn(1000)

# Plotting the histogram
plt.hist(data, bins=30, color='blue', alpha=0.7)
plt.title('Histogram')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



Bar Plot

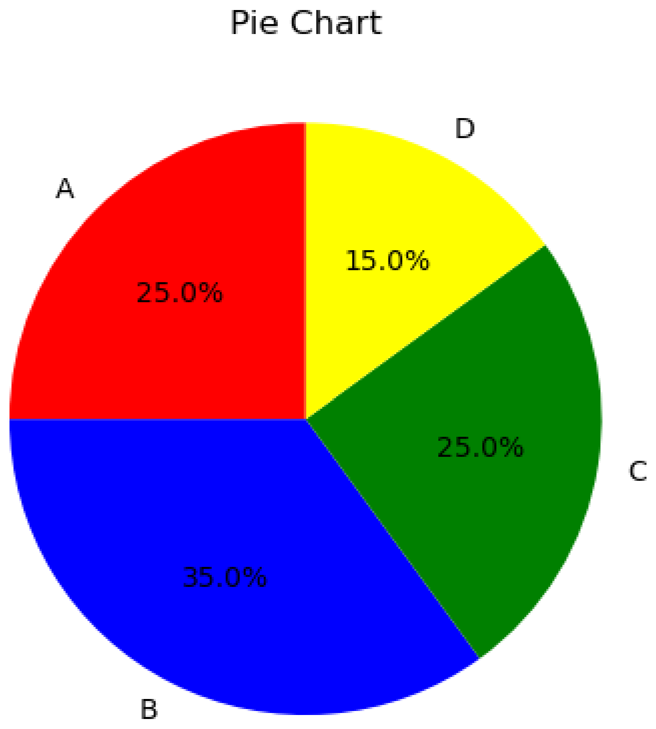
In [5]:

```
labels = ['A', 'B', 'C', 'D']
values = [10, 45, 30, 15]

plt.bar(labels, values, color=['red', 'blue', 'green', 'yellow'])
plt.title('Bar Plot')
plt.xlabel('Labels')
plt.ylabel('Values')
plt.show()
```



Pie Chart

```
In [6]: sizes = [25, 35, 25, 15]
        labels = ['A', 'B', 'C', 'D']
        colors = ['red', 'blue', 'green', 'yellow']

        plt.pie(sizes, labels=labels, colors=colors, autopct='%1.1f%%', starta
        plt.title('Pie Chart')
        plt.show()
```



2. Seaborn

Seaborn is a data visualization library in Python that provides a high-level interface for creating attractive graphs. Built on top of Matplotlib, it is integrated with Pandas data structures. Seaborn comes with several built-in themes and color palettes to enhance the visual appeal of plots. It excels in creating statistical graphics and simplifies the process of deriving insights from data. Seaborn's functionalities extend beyond Matplotlib, offering a variety of plot types useful for statistical data exploration.

Importing the library and loading the dataset

In [12]:

```
# Importing necessary libraries
import seaborn as sns
import matplotlib.pyplot as plt

# Load a built-in dataset
data = sns.load_dataset("tips")
data.head(5)
dtype_df = data.dtypes
dtype_df.describe
```

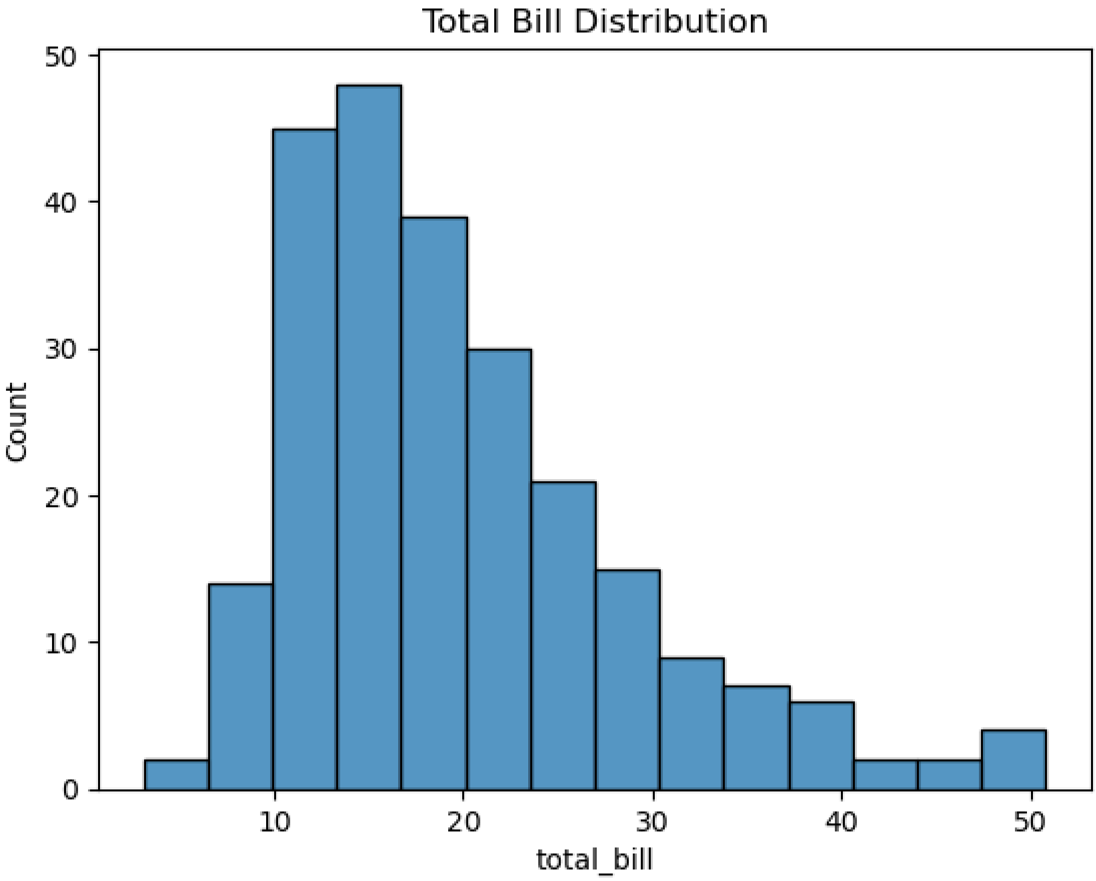
Out[12]:

```
<bound method NDFrame.describe of total_bill     float64
tip              float64
sex             category
smoker          category
day             category
time            category
size               int64
dtype: object>
```

Basic Plotting with Seaborn

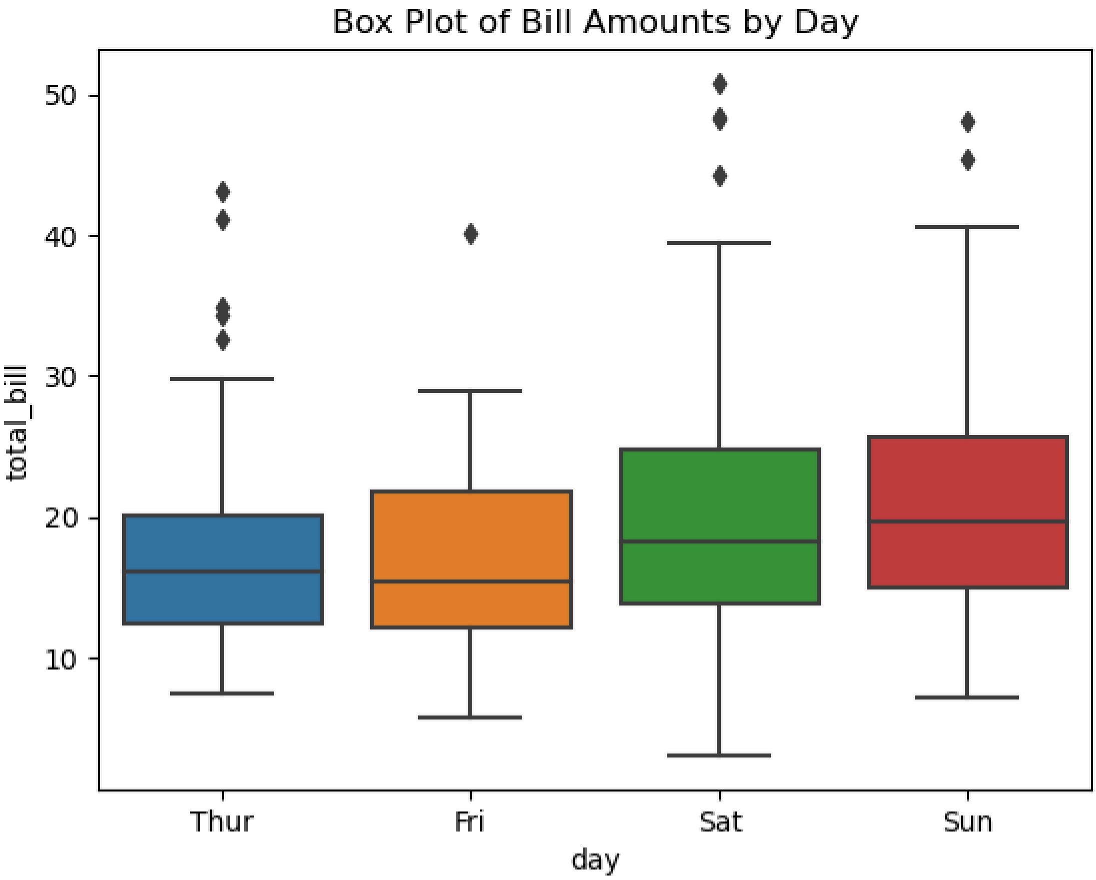
In [8]:

```
# Displaying a histogram
sns.histplot(data['total_bill'])
plt.title("Total Bill Distribution")
plt.show()
```



Box Plots

In [9]:
```
# Visualizing the distribution of bills by day
sns.boxplot(x='day', y='total_bill', data=data)
plt.title("Box Plot of Bill Amounts by Day")
plt.show()
```

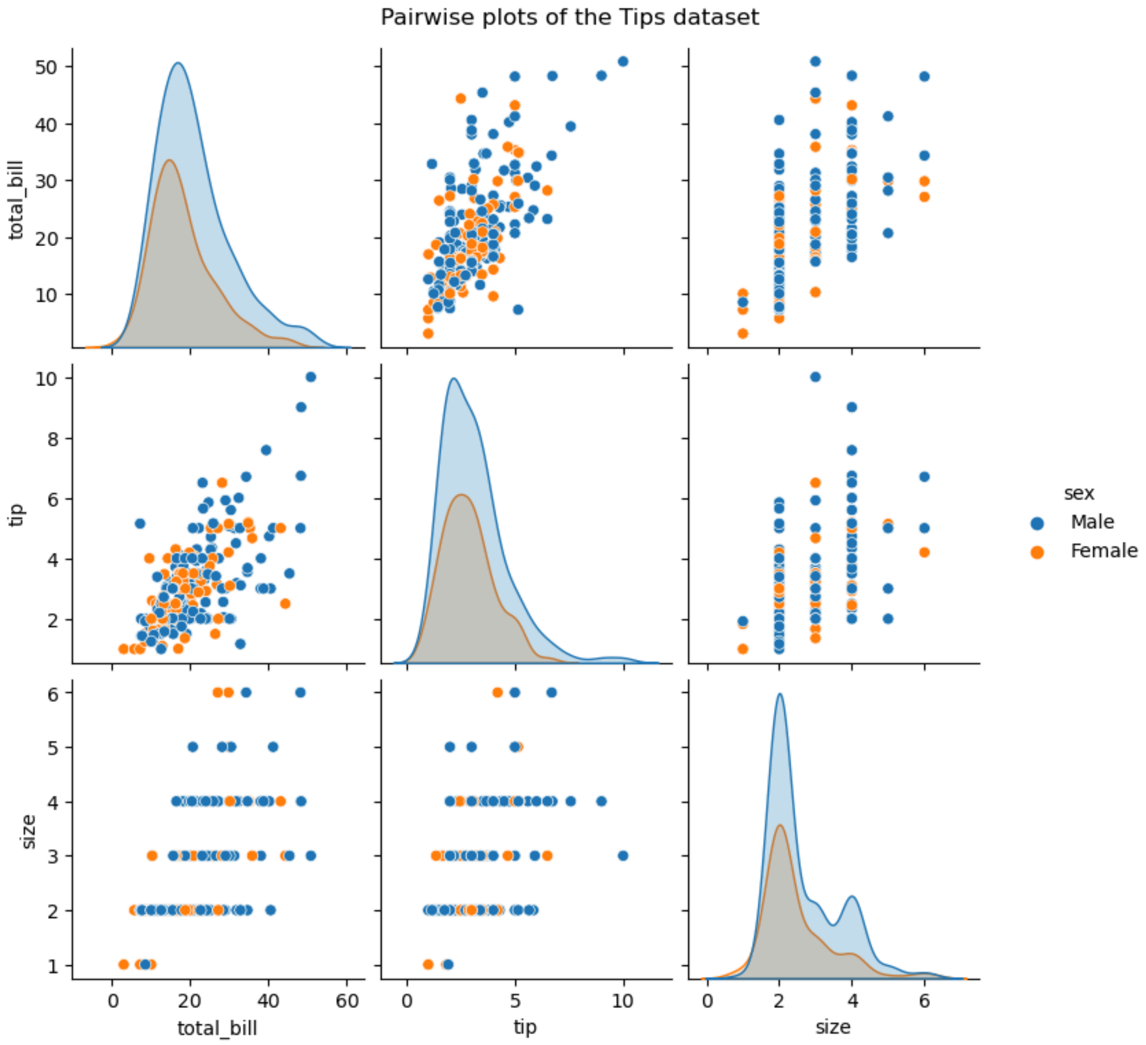


Pair Plots

In [17]:

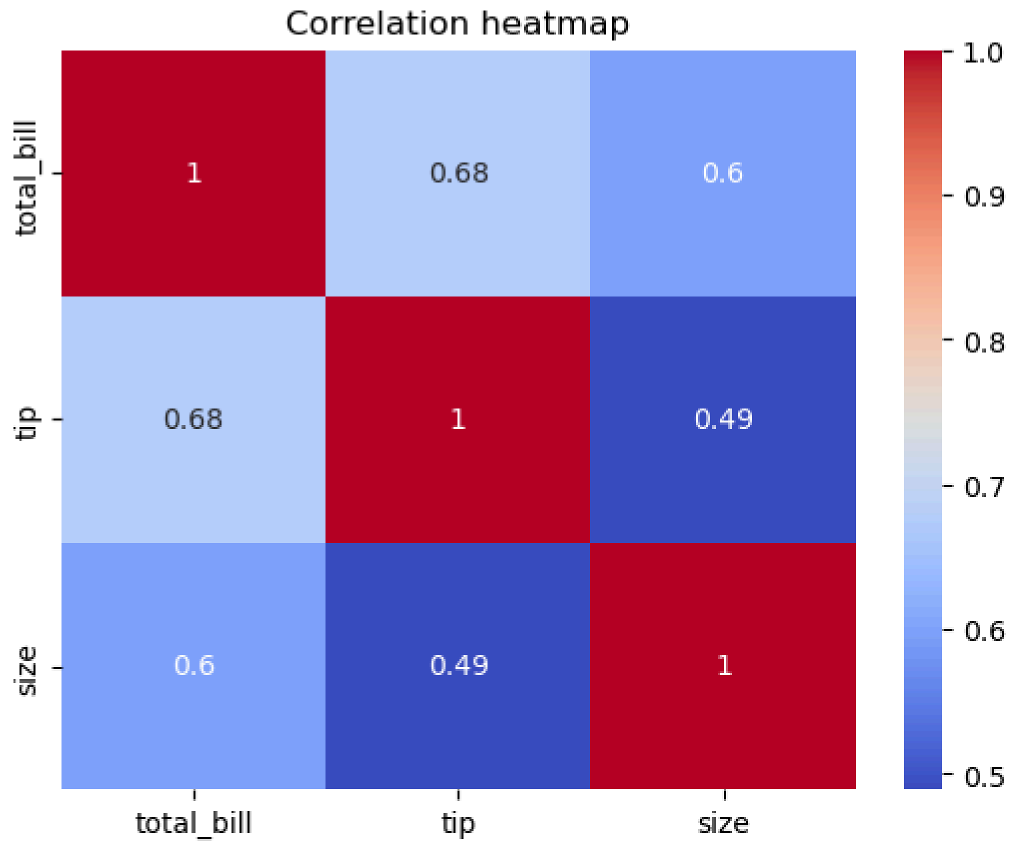
```
# Pair plot gives pairwise relationships in a dataset
sns.pairplot(data, hue="sex")
plt.suptitle("Pairwise plots of the Tips dataset", y=1.02)
plt.tight_layout
plt.show()
```

```
/Users/josephkambham/anaconda3/lib/python3.11/site-packages/seaborn/
axisgrid.py:118: UserWarning: The figure layout has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```



Heatmaps

```
In [13]: # Compute correlations among features and visualize using a heatmap
         correlation = data.corr(numeric_only=True)
         sns.heatmap(correlation, annot=True, cmap="coolwarm")
         plt.title("Correlation heatmap")
         plt.show()
```



Closing Thoughts

```
In [14]: # Always close your plots to free up resources
         plt.close()
```

```
In [ ]:
```

```
In [ ]:
```