LINE PLOT

Purpose: Tracks trends or change over continuous intervals—great for showing time series, sensor data, or mathematical functions.

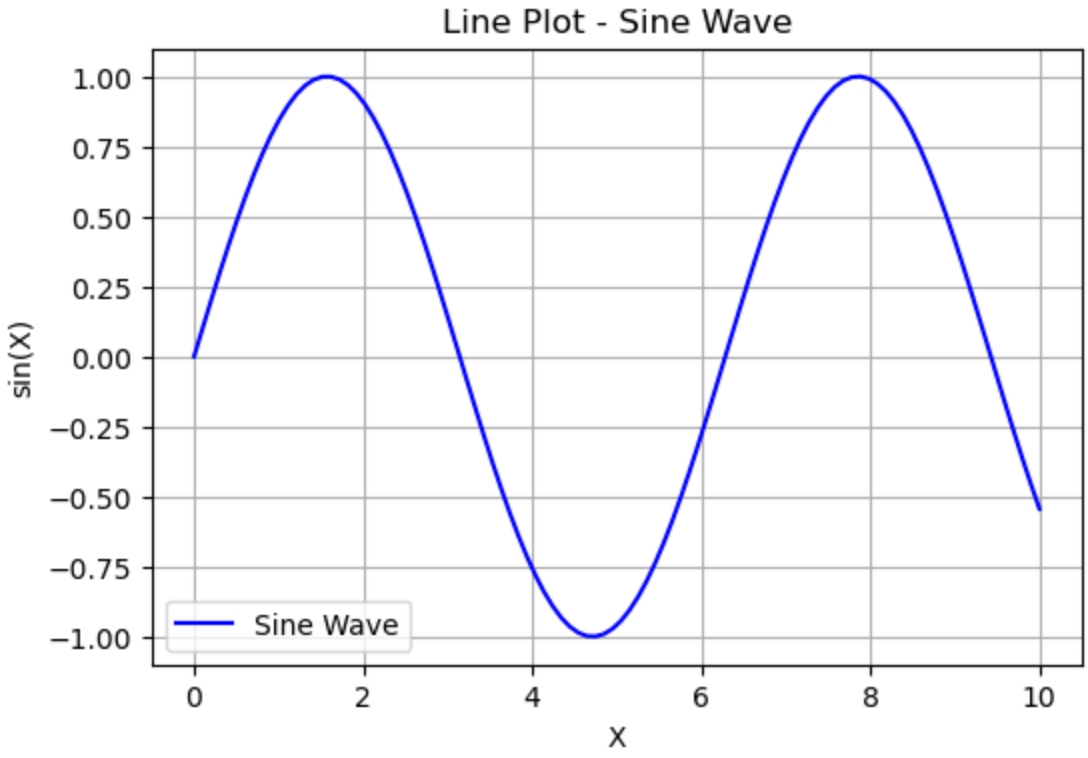
Example: Plotting a sine wave across 100 data points.

Use: It's simple and very customizable—ideal for clear trend visualization.

In [11]:
```
import matplotlib.pyplot as plt
import numpy as np

# Line Plot
x = np.linspace(0, 10, 100)
y = np.sin(x)

plt.figure(figsize=(6,4))
plt.plot(x, y, label='Sine Wave', color='blue')
plt.title('Line Plot - Sine Wave')
plt.xlabel('X')
plt.ylabel('sin(X)')
plt.legend()
plt.grid(True)
plt.show()
```



In [ ]:
```
BAR CHART
```

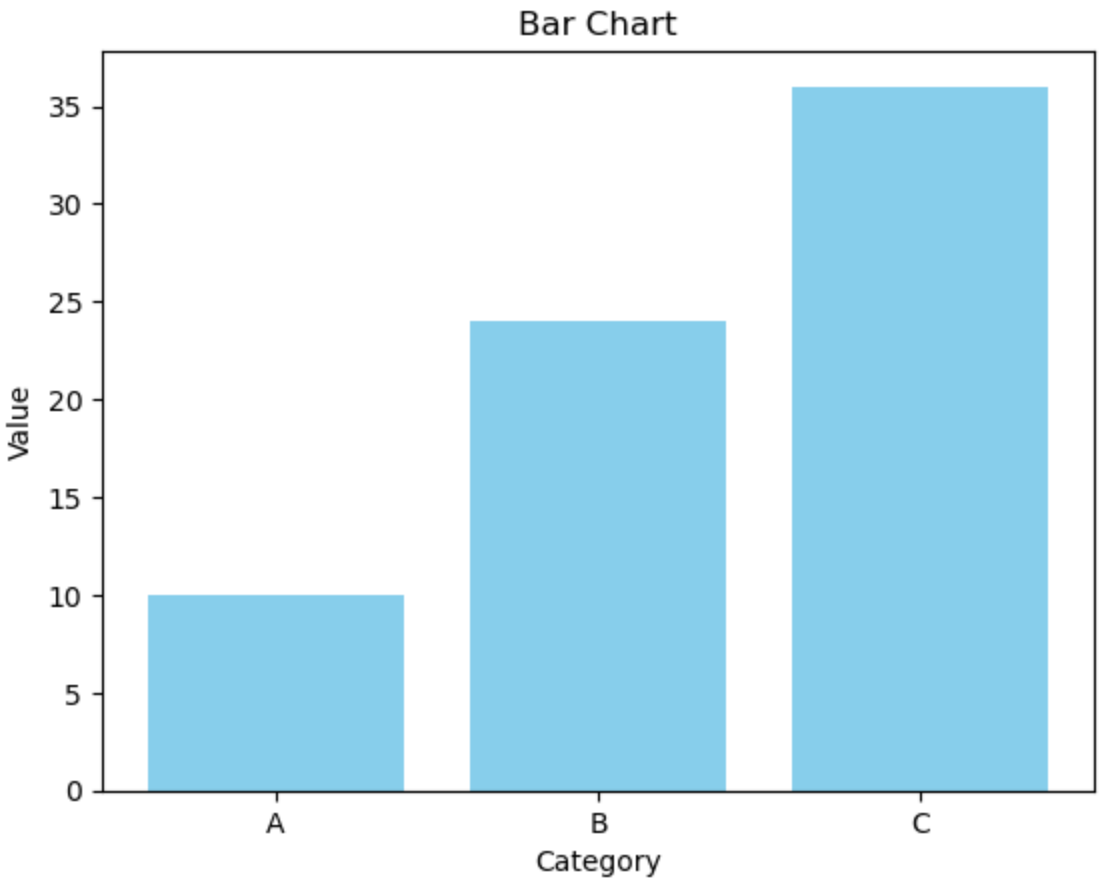
Purpose: Compares quantities across discrete categories.

Example: Number of runs scored by three players.

Why Use: It’s intuitive for presenting comparisons like sales, scores, or counts.

In [2]:
```
# Bar Chart
categories = ['A', 'B', 'C']
values = [10, 24, 36]

plt.bar(categories, values, color='skyblue')
plt.title('Bar Chart')
plt.xlabel('Category')
plt.ylabel('Value')
plt.show()
```



In [ ]:
```
PIE CHART
```

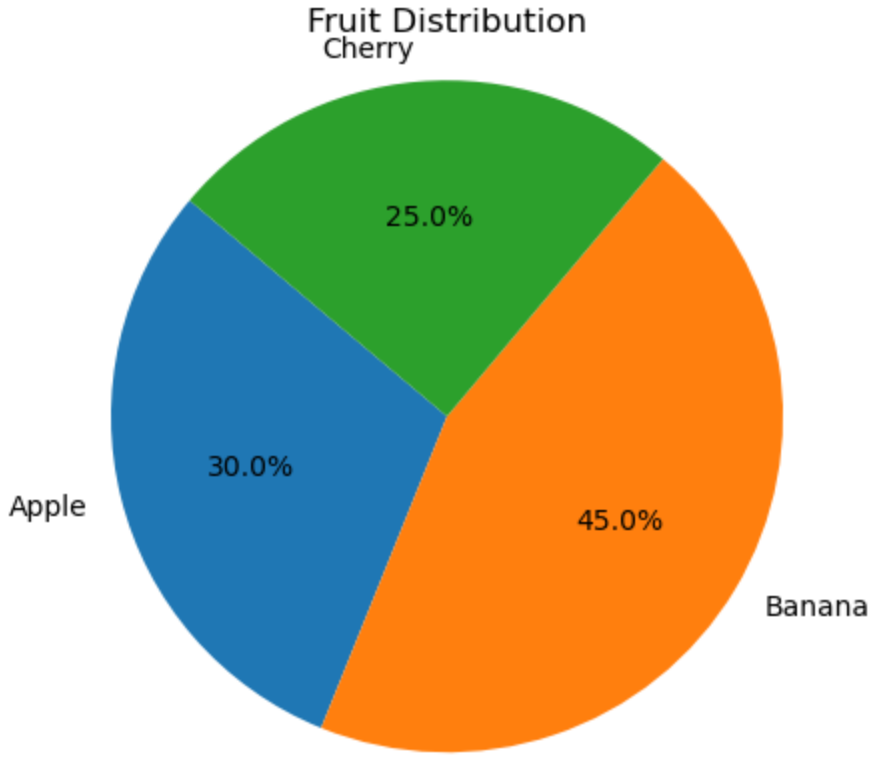
Purpose: Shows part-to-whole relationships.

Example: Market share distribution or fruit sales ratio.

Why Use: Visually emphasizes proportions but best for simple comparisons with few categories.

In [3]:
```
# Pie Chart
labels = ['Apple', 'Banana', 'Cherry']
sizes = [30, 45, 25]
```

```
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title('Fruit Distribution')
plt.axis('equal')
plt.show()
```



In [ ]:
```
HISTOGRAM
```

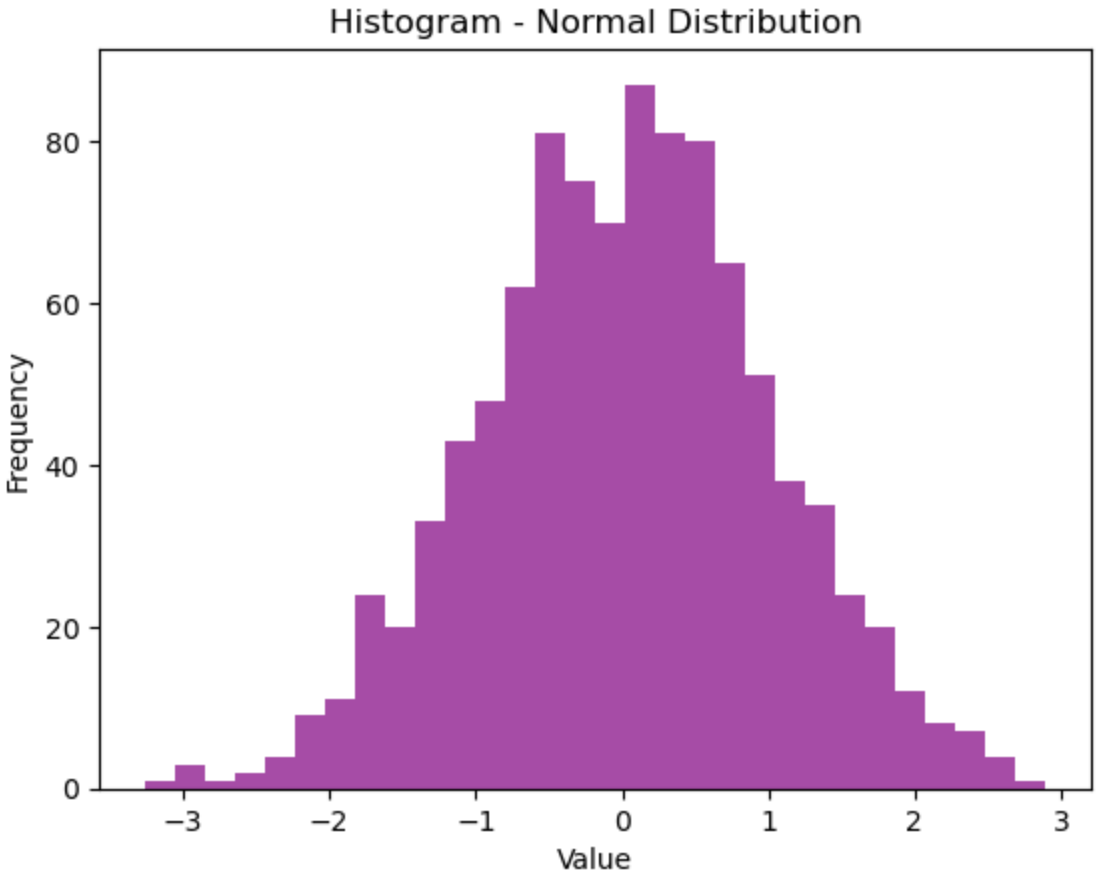
Purpose: Shows the frequency distribution of numerical data.

Example: Distribution of player scores or delivery speeds in a match.

Why Use: Helps assess skewness, spread, or presence of outliers in continuous data.

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

plt.hist(data, bins=30, color='purple', alpha=0.7)
plt.title('Histogram - Normal Distribution')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



In [ ]:
```
SCATTER PLOT
```

Purpose: Reveals relationships between two numeric variables.

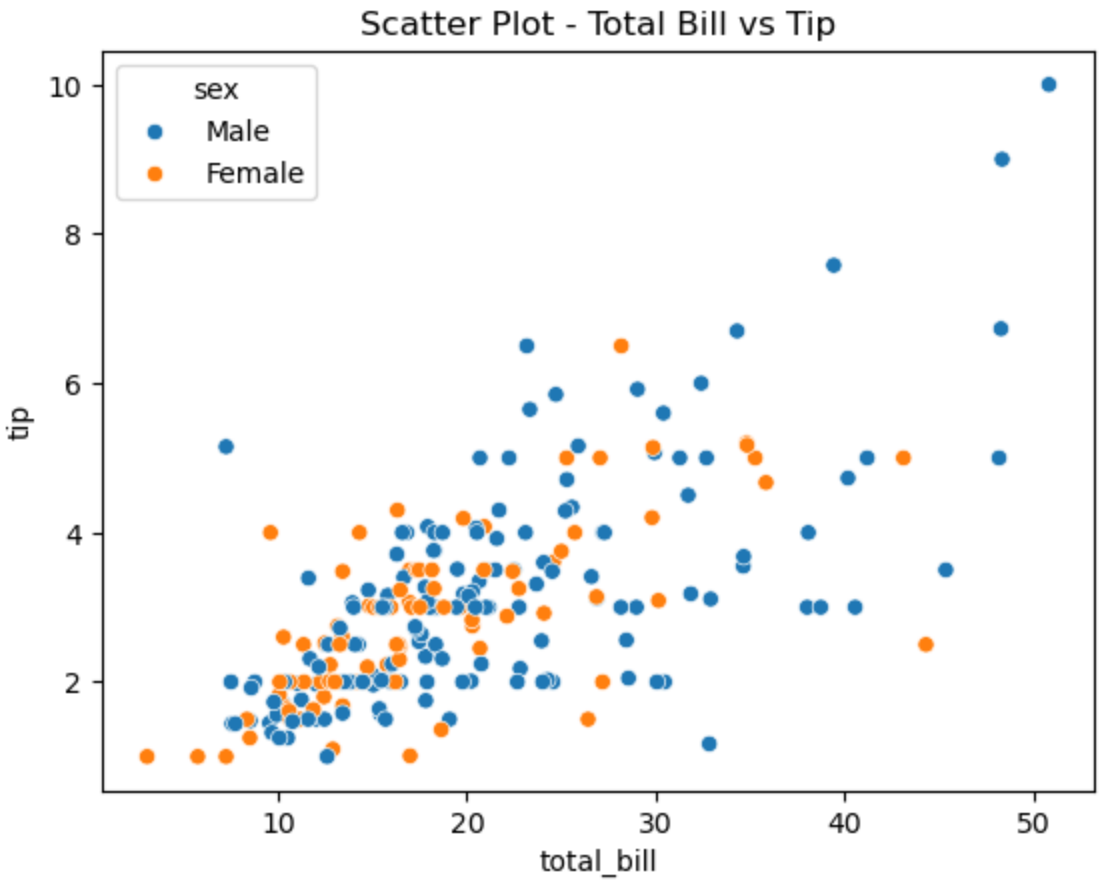
Example: Visualizing if higher total bills tend to yield higher tips.

Why Use: Perfect for correlation analysis, and the hue shows subgroup patterns (like gender).

In [5]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Sample Data
tips = sns.load_dataset("tips")

# Scatter Plot
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="sex")
plt.title("Scatter Plot - Total Bill vs Tip")
plt.show()
```



In [ ]:
```
BOX PLOT
```

Purpose: Displays distribution based on 5-number summary (min, Q1, median, Q3, max).

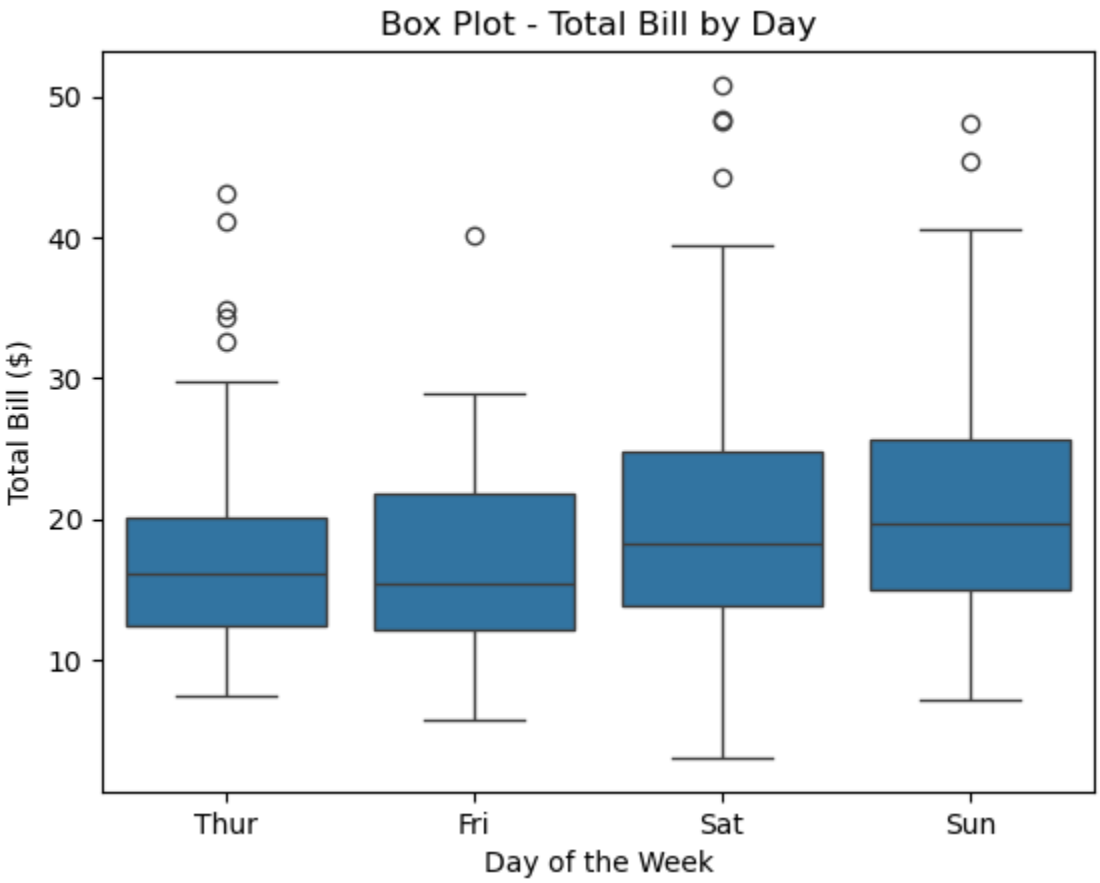
Example: Variation in bills per day of the week.

Why Use: Highlights median, spread, and outliers—great for comparative analytics.

In [13]:
```
import seaborn as sns
import matplotlib.pyplot as plt

# Load sample data
tips = sns.load_dataset("tips")

# Create box plot without palette (avoids deprecation warning)
sns.boxplot(data=tips, x="day", y="total_bill")
plt.title("Box Plot - Total Bill by Day")
plt.xlabel("Day of the Week")
plt.ylabel("Total Bill ($)")
plt.show()
```



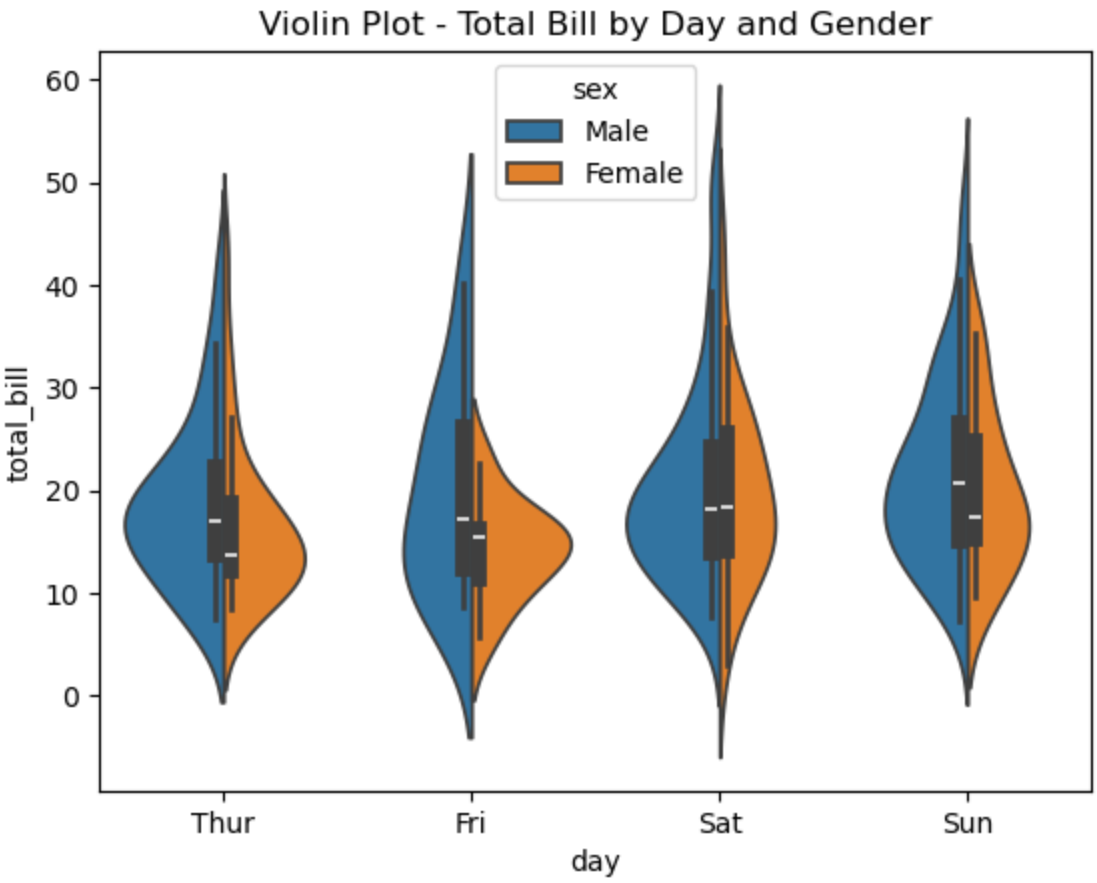
In [ ]:
```
VIOLIN PLOT
```

Purpose: Combines box plot with a kernel density estimate (KDE).

Example: Comparing bill amounts across gender and day.

Why Use: Gives richer distribution details than a box plot.

In [7]:
```
# Violin Plot
sns.violinplot(data=tips, x="day", y="total_bill", hue="sex", split=True)
plt.title("Violin Plot - Total Bill by Day and Gender")
plt.show()
```



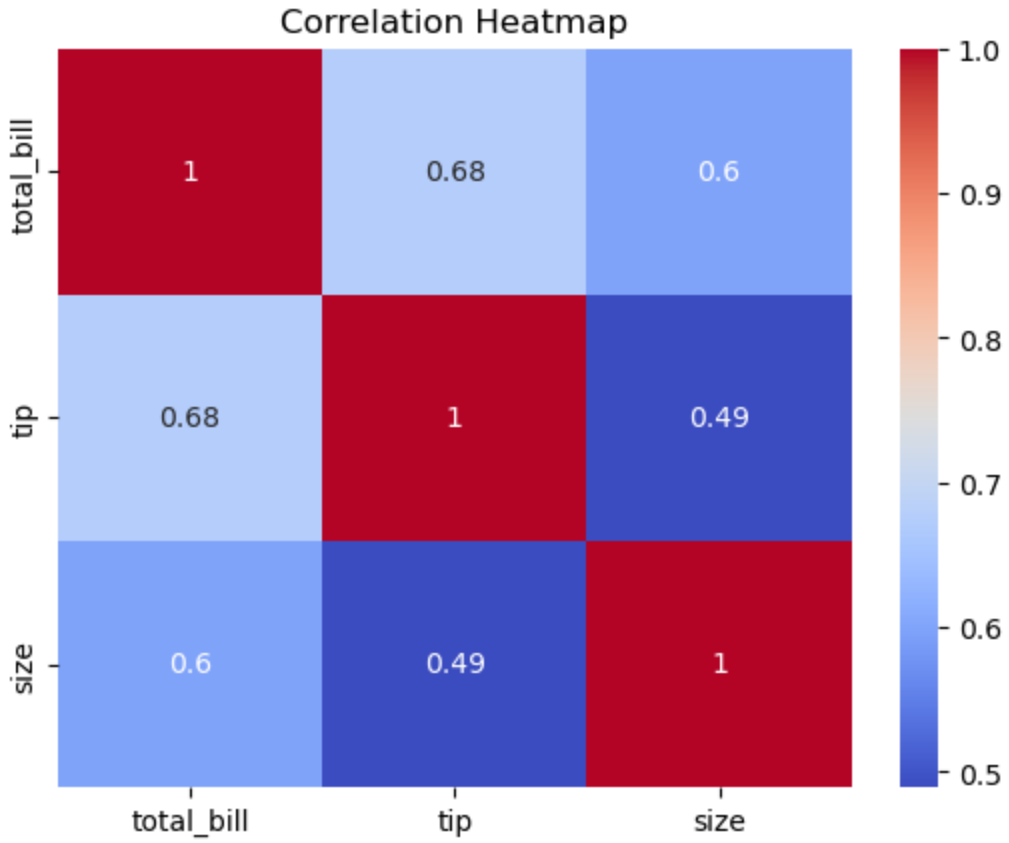
In [ ]:
```
HEATMAP
```

Purpose: Displays a matrix of correlations or values with color encoding.

Example: Correlation between variables like total bill, tip, size.

Why Use: Excellent for identifying linear relationships or dependencies.

In [8]:
```
# Heatmap
corr = tips.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```



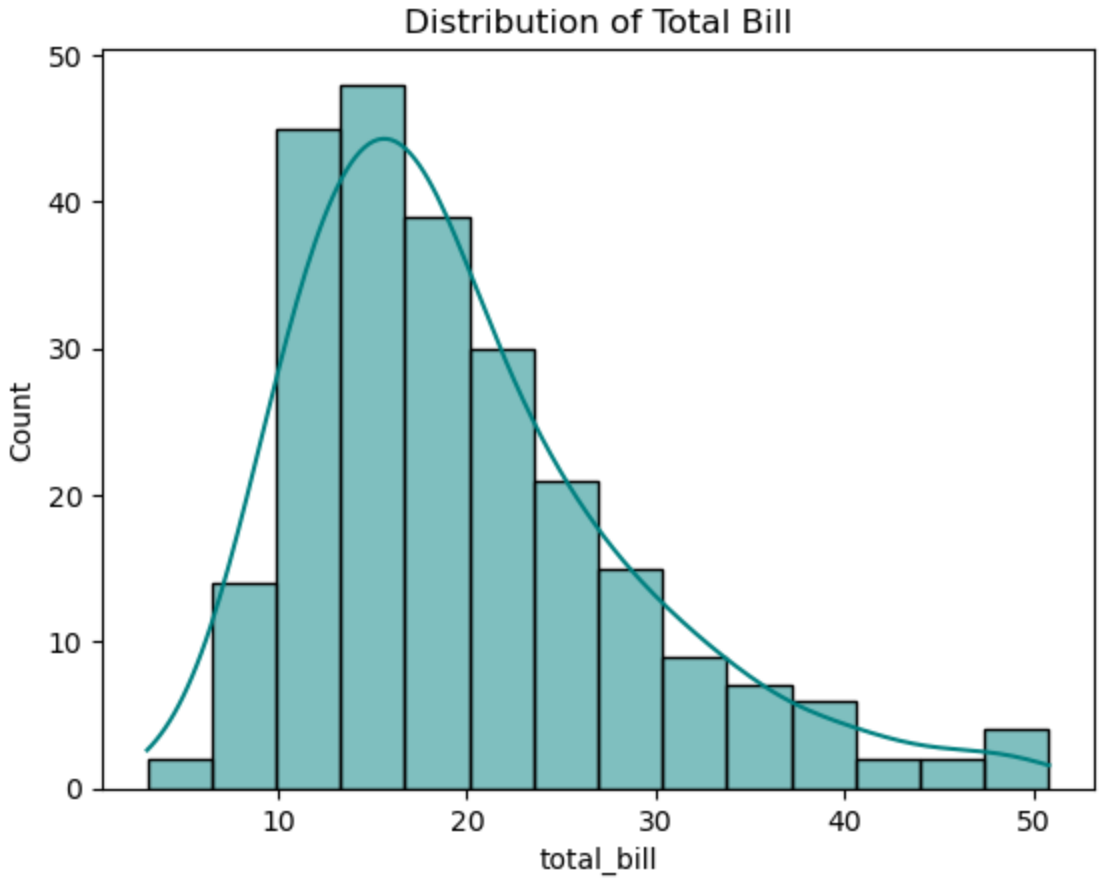
In [ ]:
```
DISTRIBUTION PLOT
```

Purpose: Shows distribution of a numeric variable with optional density overlay.

Example: Frequency of bill sizes in the dataset.

Why Use: Helps see skewness, central tendency, and approximate shape of data distribution.

In [9]:
```
# Distribution Plot
sns.histplot(data=tips, x="total_bill", kde=True, color='teal')
plt.title("Distribution of Total Bill")
plt.show()
```


Distribution of Total Bill
Count
total_bill
0
10
20
30
40
50
10
20
30
40
50

In [ ]: