EXPERIMENT 6

Aim:

To analyze and visualize data using the Seaborn library in Python through various types of plots such as distribution, joint, pair, box, count, and heatmap plots.

Algorithm:

1. Import the required libraries – Seaborn, Pandas, NumPy, and Matplotlib.
2. Load the built-in tips dataset using Seaborn.
3. Display the first few rows of the dataset using head().
4. Visualize data distributions using displot() with and without KDE.
5. Explore relationships between numerical variables using jointplot().
6. Compare multiple variables simultaneously using pairplot().
7. Visualize correlation between numerical variables using a heatmap().
8. Identify outliers using boxplot().
9. Display categorical distributions using countplot() and bar/pie charts.
10. Display the visualizations for analysis.

Code:

```
import seaborn as sns
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline

tips = sns.load_dataset('tips')
tips.head()
```

Output:

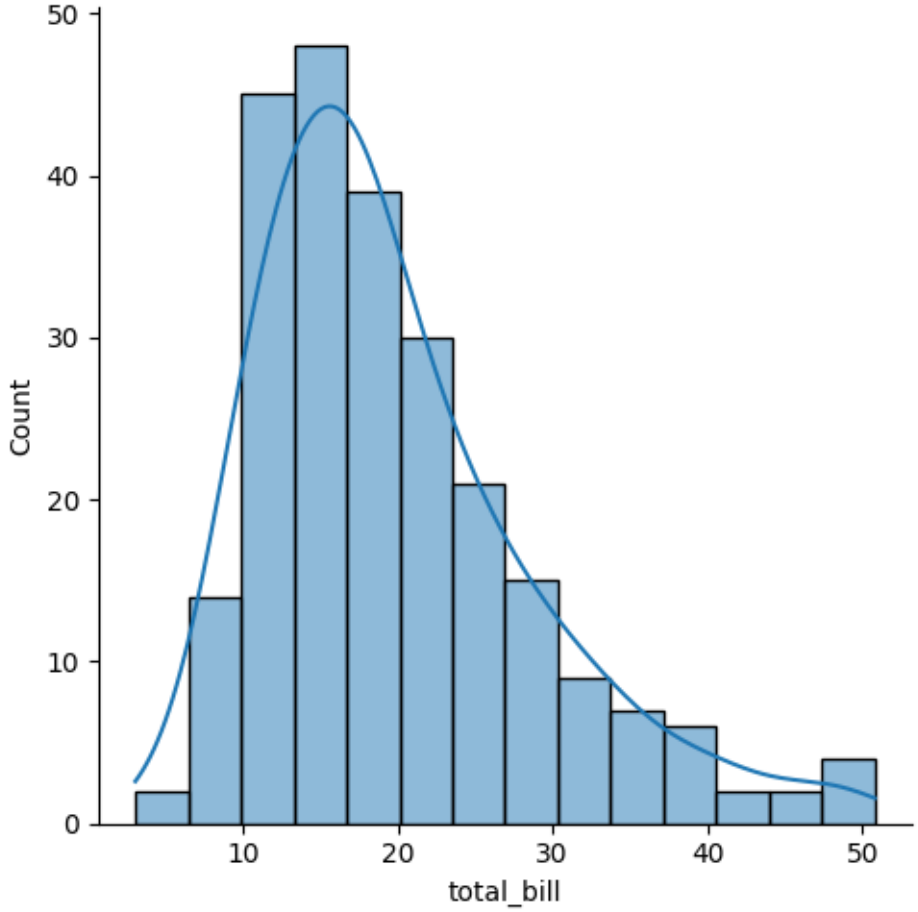
	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3

	total_bill	tip	sex	smoker	day	time	size
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

---

sns.displot(tips.total_bill, kde=True)

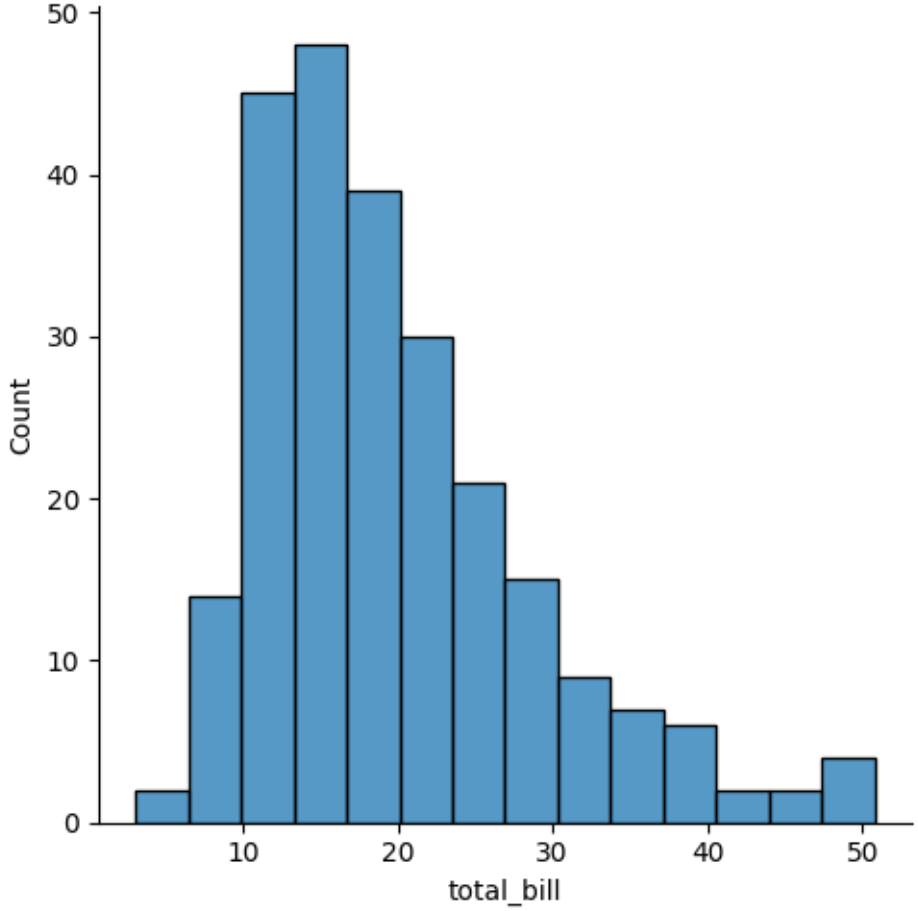
Output:



---

sns.displot(tips.total_bill, kde=False)

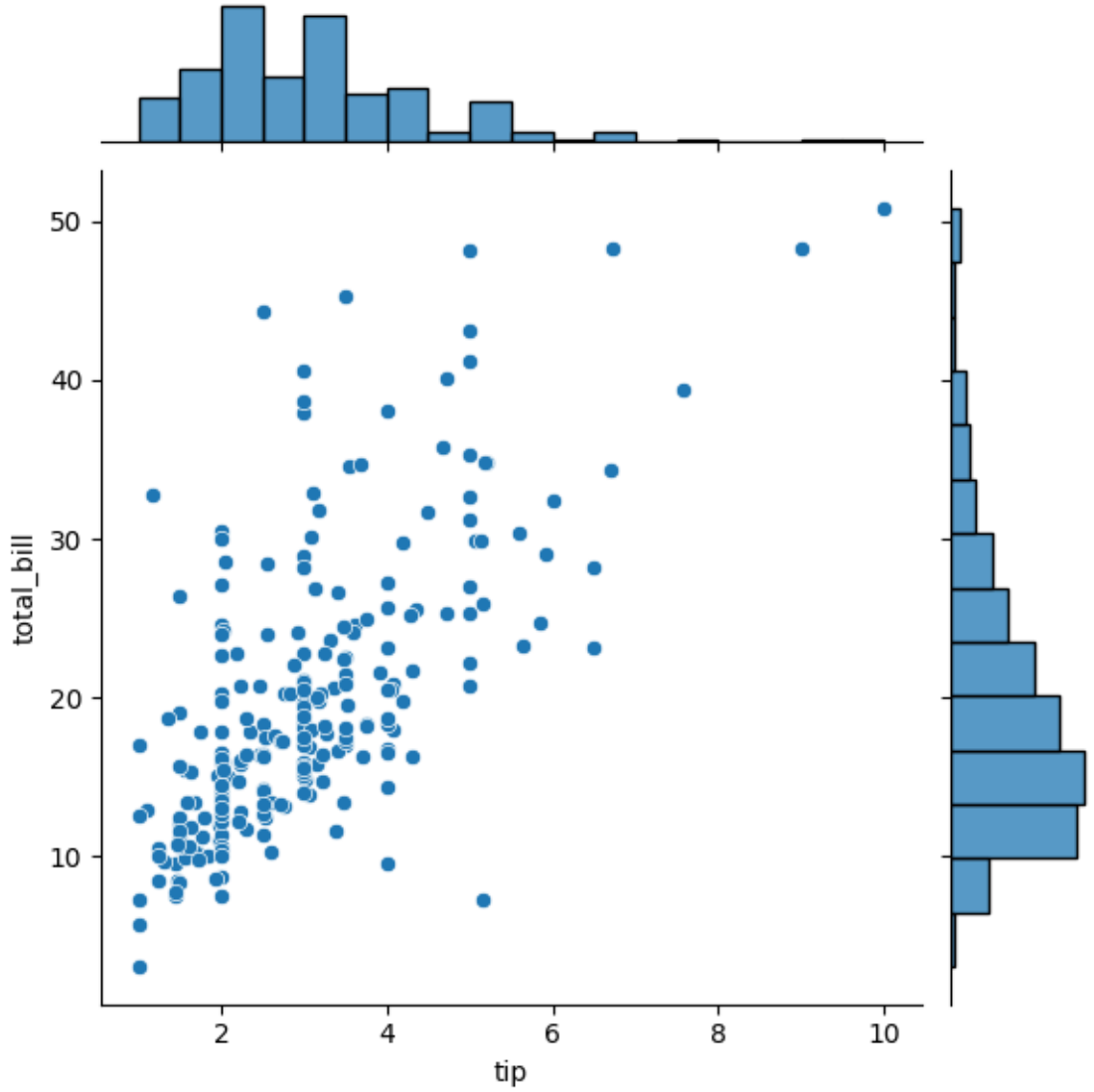
Output:



---

sns.jointplot(x=tips.tip, y=tips.total_bill)

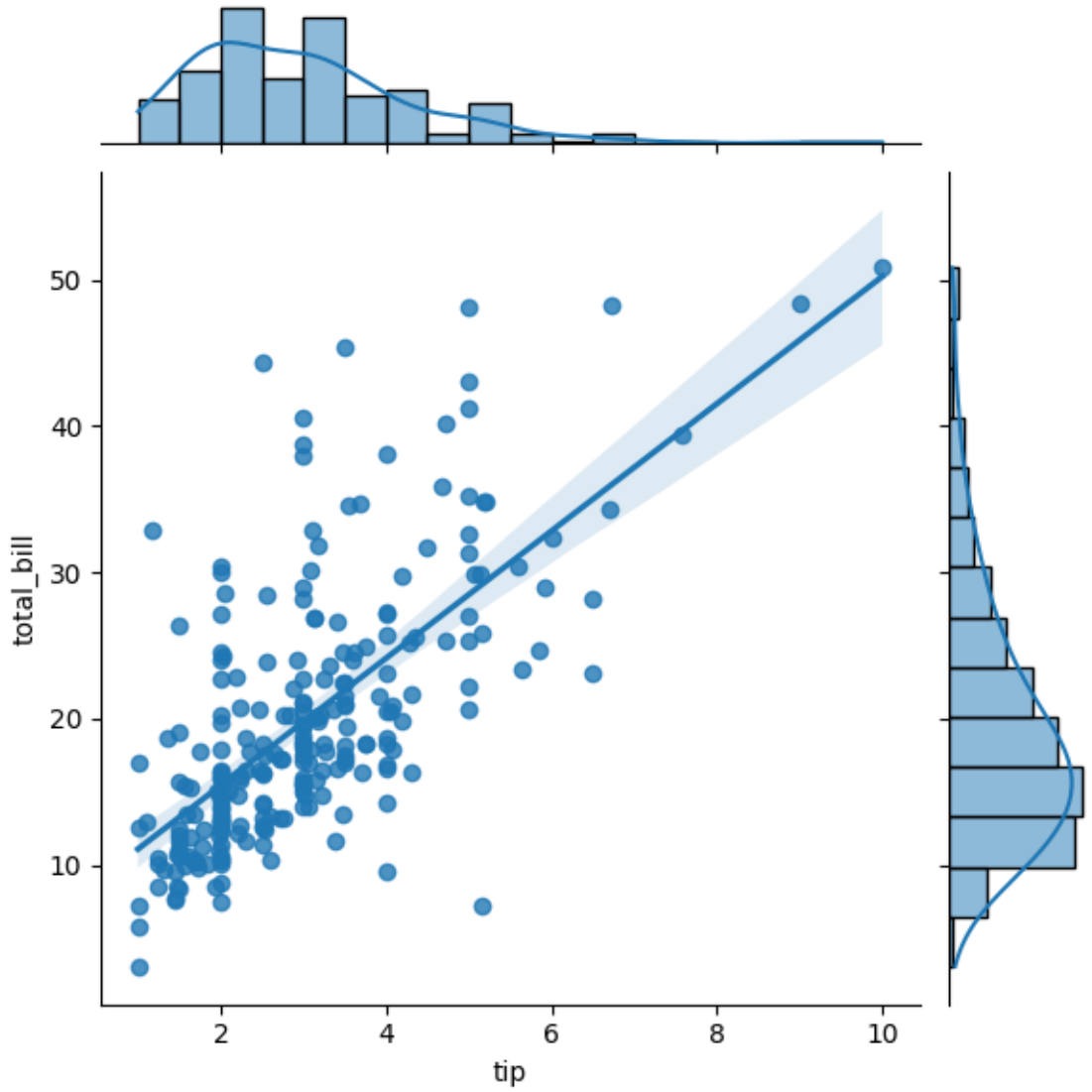
Output:



---

sns.jointplot(x=tips.tip, y=tips.total_bill, kind="reg")

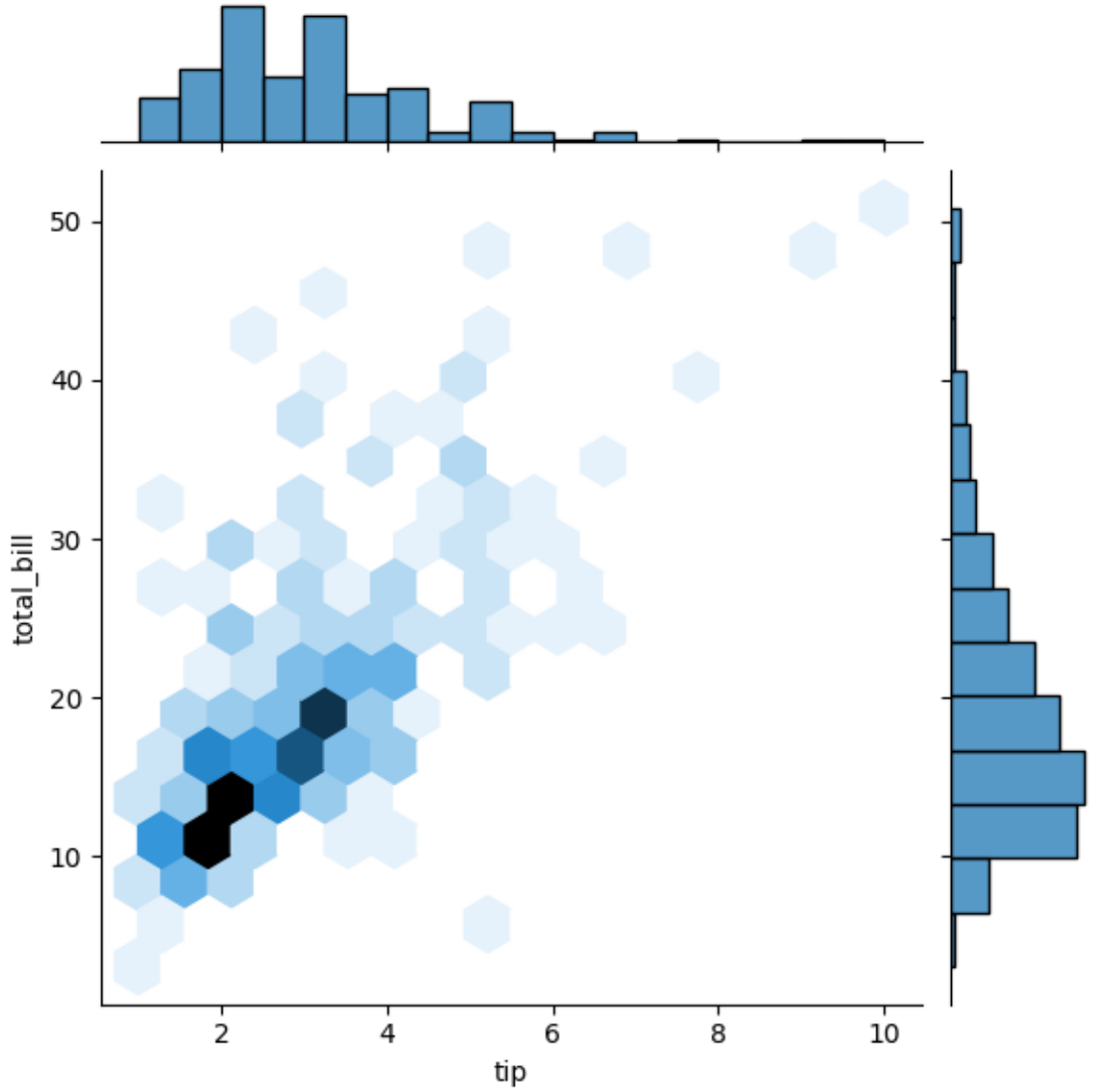
Output:



---

sns.jointplot(x=tips.tip, y=tips.total_bill, kind="hex")

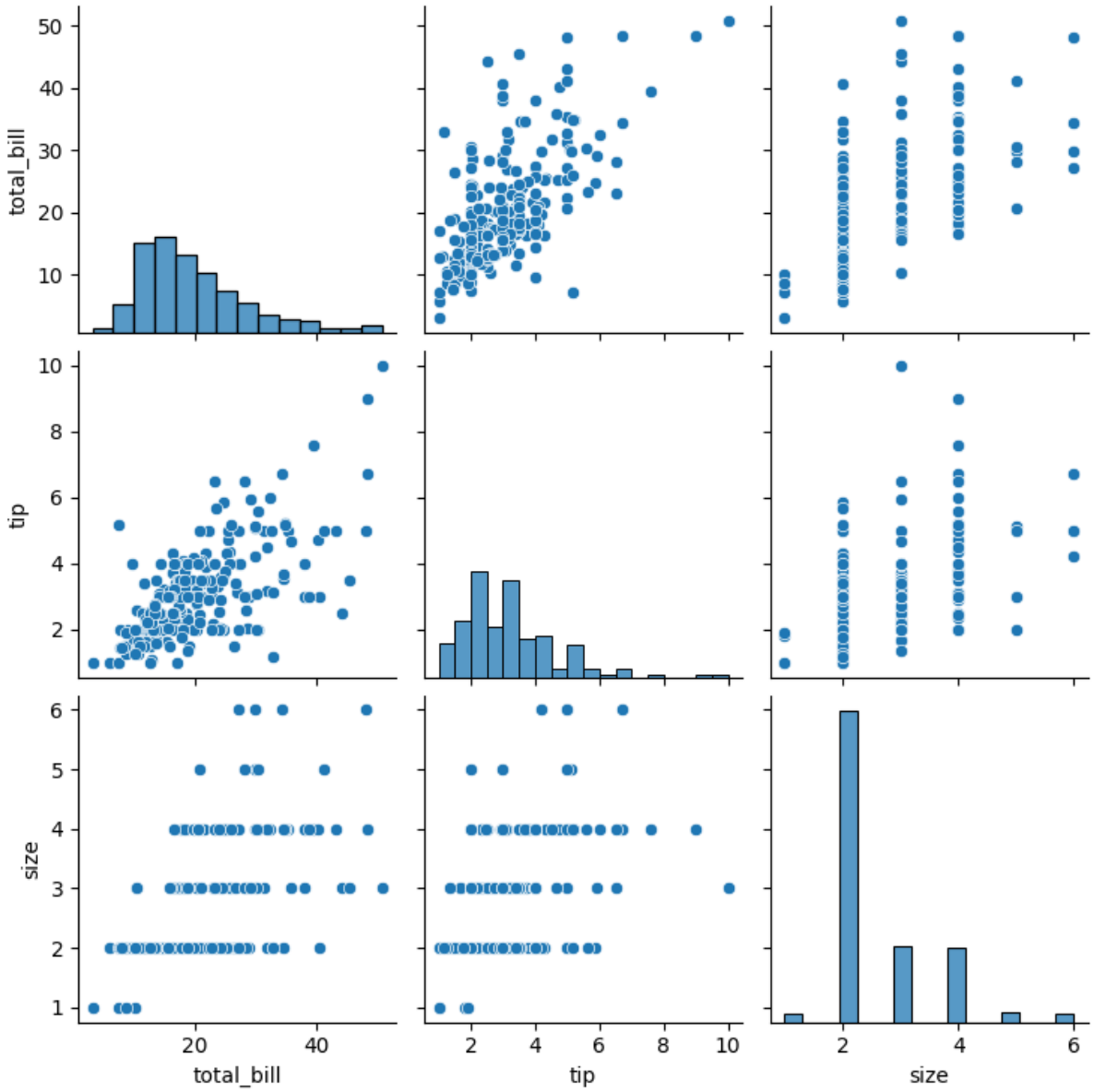
Output:



---

sns.pairplot(tips)

Output:



---

tips.time.value_counts()

Output:

time

Dinner 176

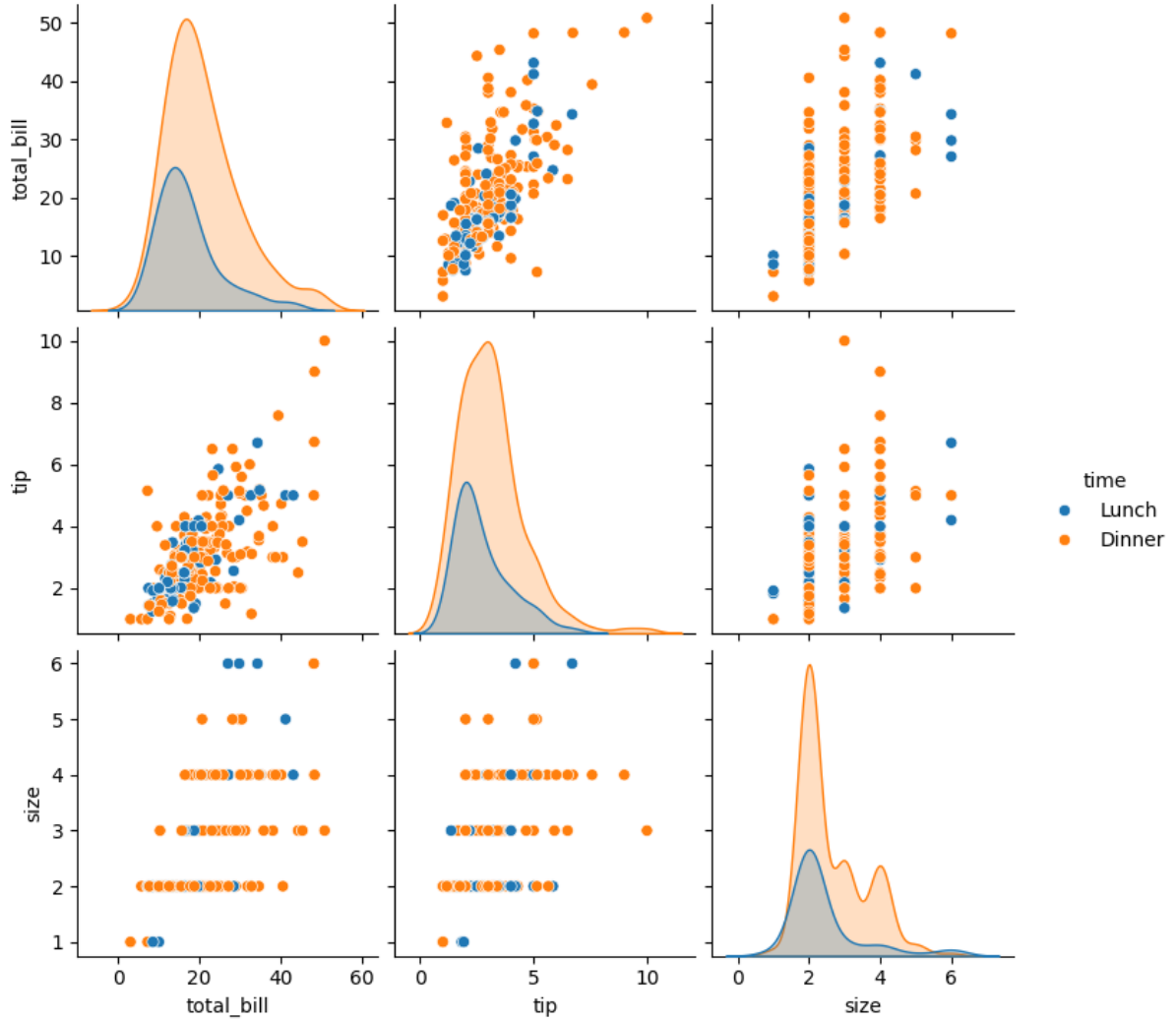
Lunch 68

Name: count, dtype: int64

---

sns.pairplot(tips, hue='time')

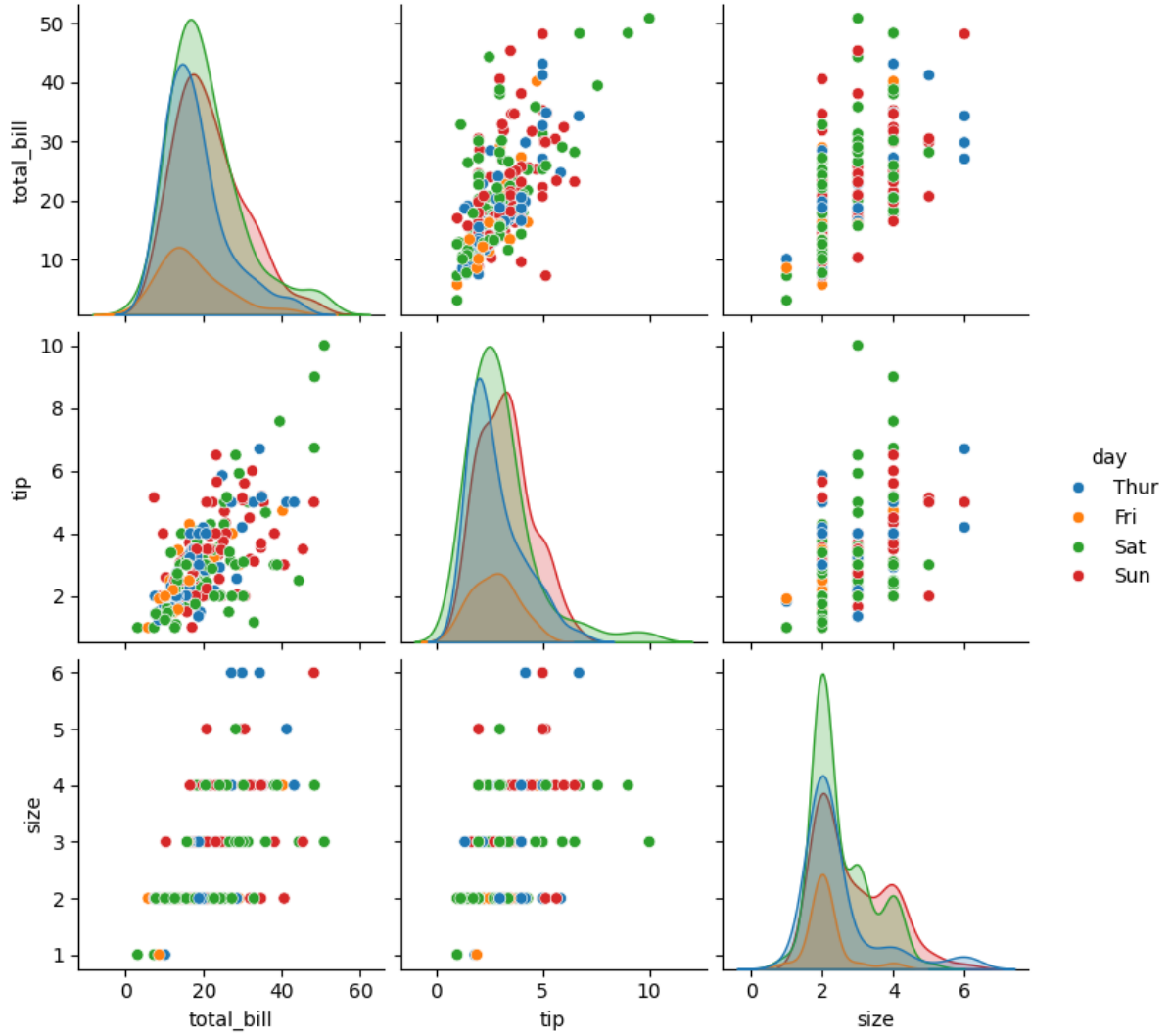
Output:



---

sns.pairplot(tips, hue='day')

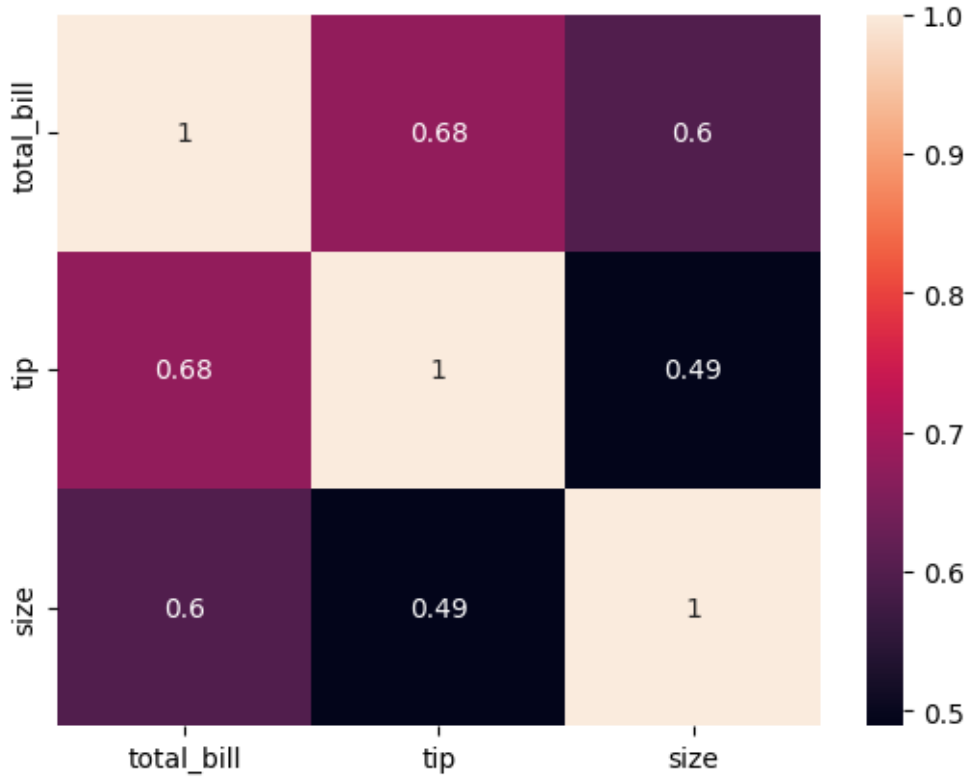
Output:



---

sns.heatmap(tips.corr(numeric_only=True), annot=True)

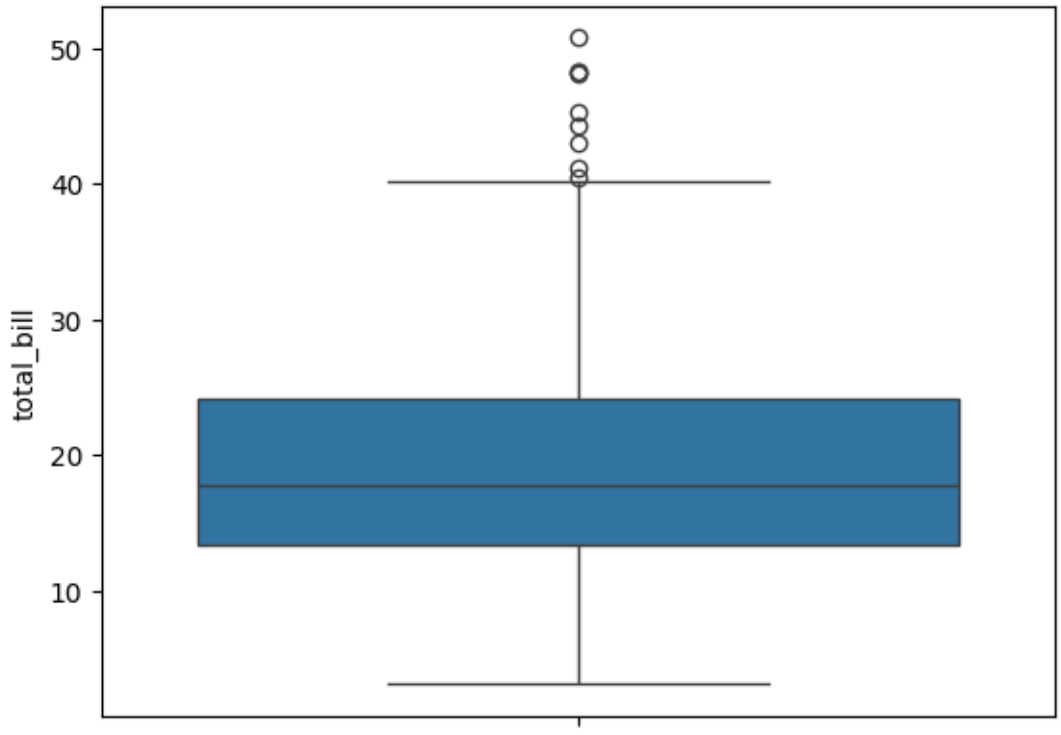
Output:



---

sns.boxplot(tips.total_bill)

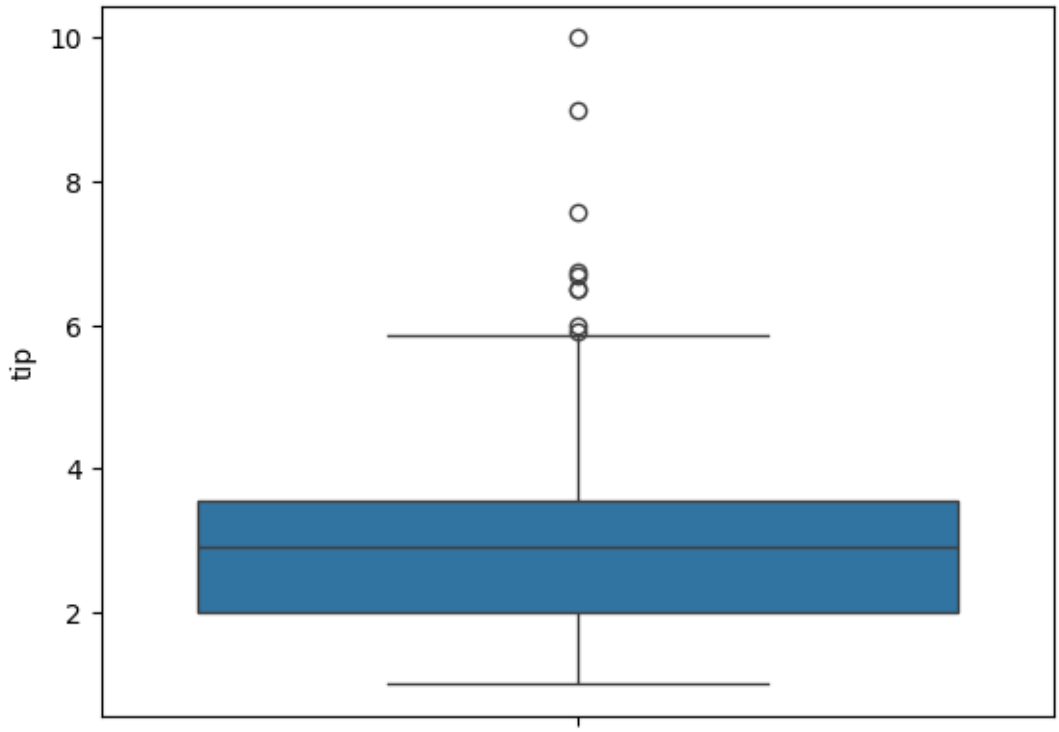
Output:



---

sns.boxplot(tips.tip)

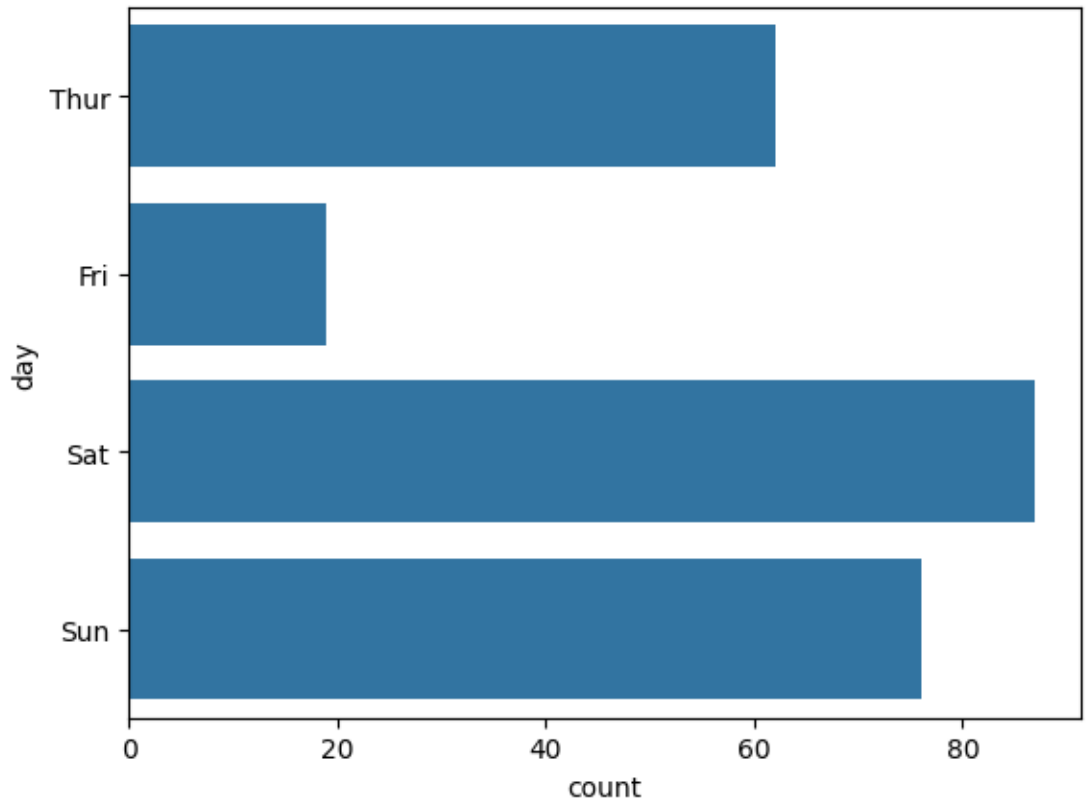
Output:



---

sns.countplot(tips.day)

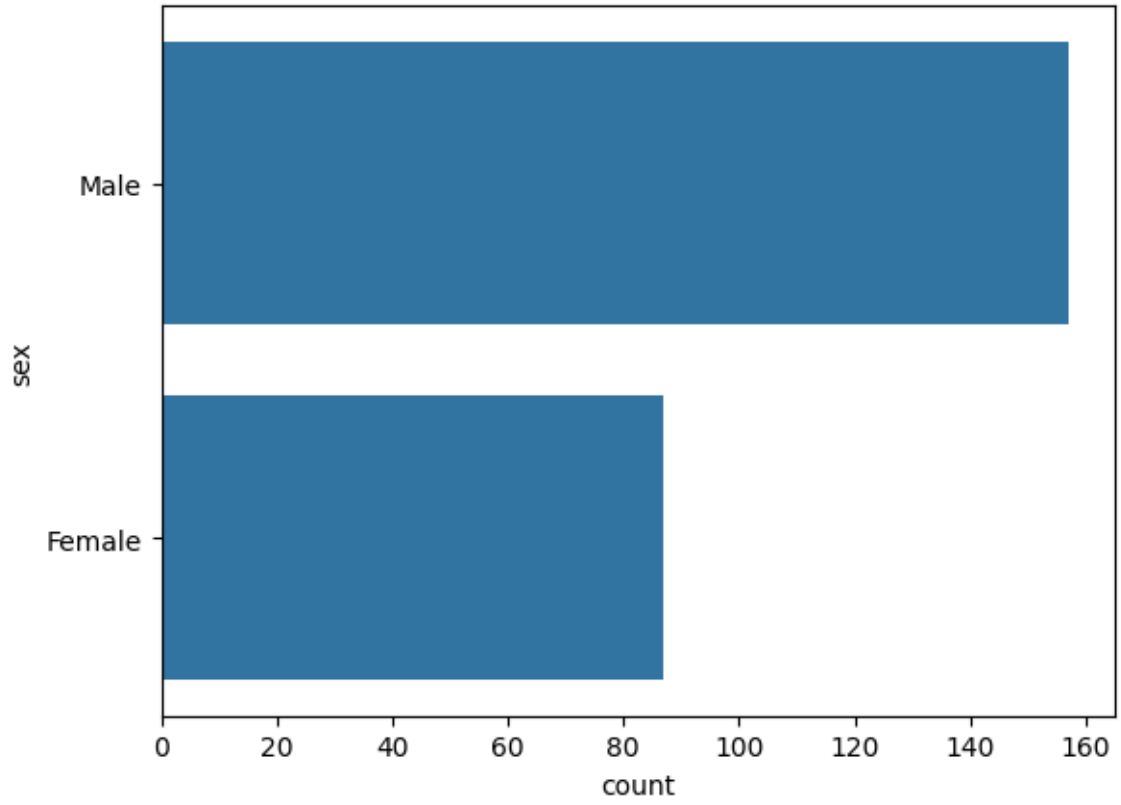
Output:



---

sns.countplot(tips.sex)

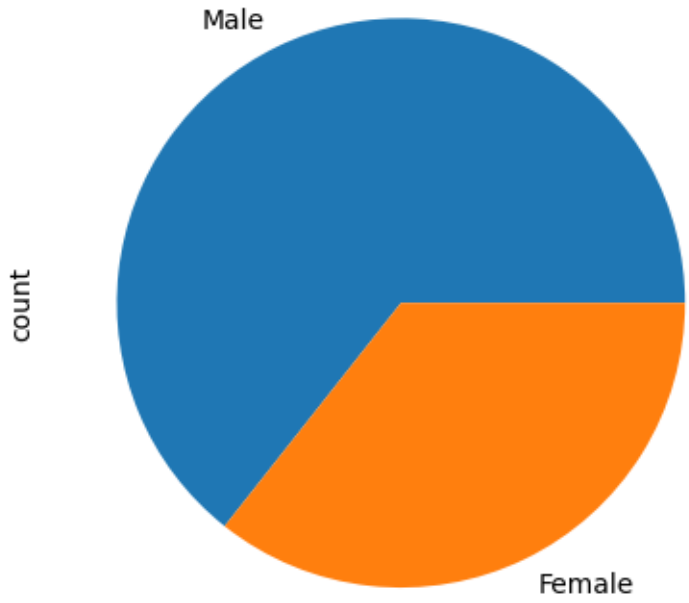
Output:



---

tips.sex.value_counts().plot(kind='pie')

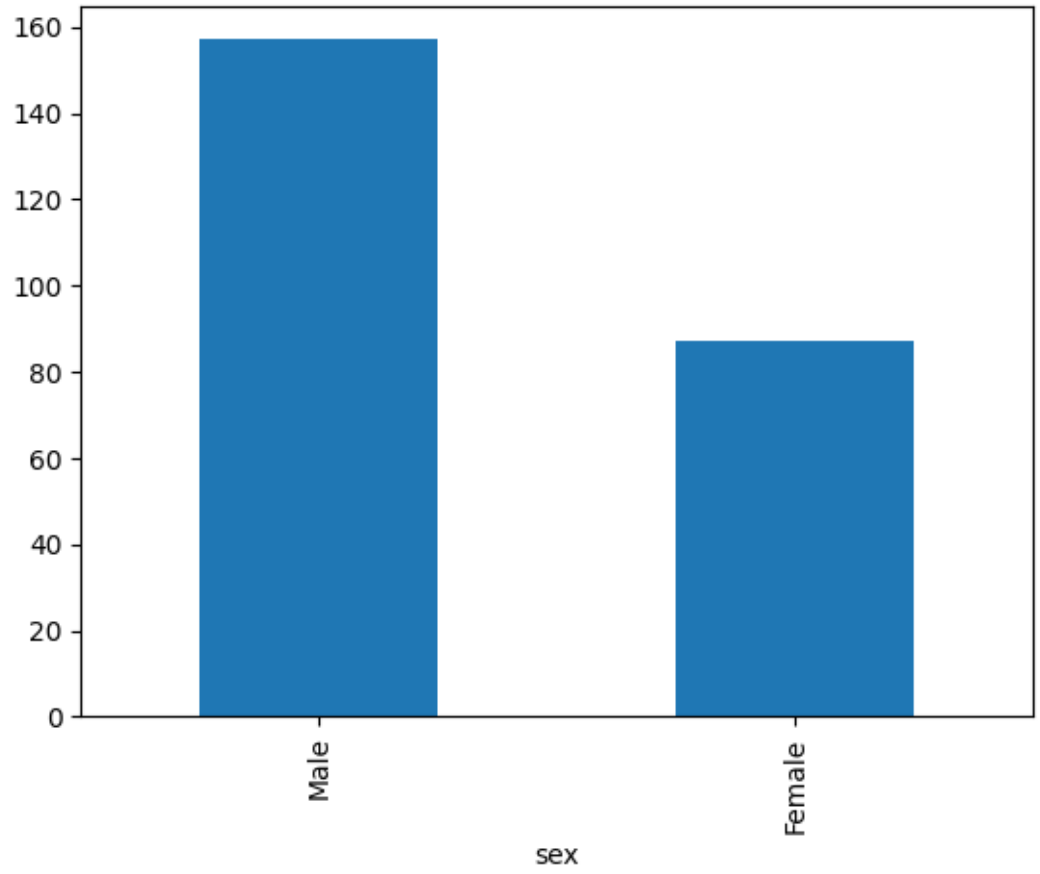
Output:



---

tips.sex.value_counts().plot(kind='bar')

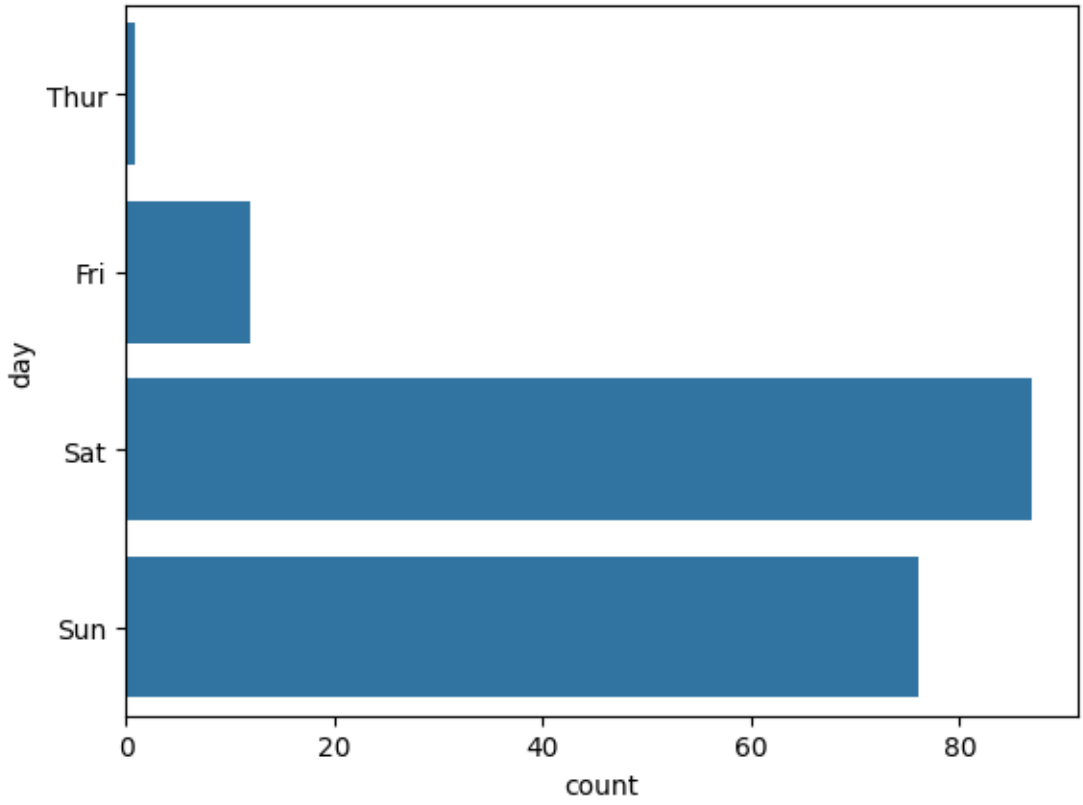
Output:



---

sns.countplot(tips[tips.time == 'Dinner']['day'])

Output:



Result:

Thus, the Python program to analyze and visualize data using the Seaborn library was executed successfully, and various plots were generated to understand data distribution, correlation, and relationships effectively.