

Fundamentals of Python

MATPLOTLIB AND SEABORN



Shruti Bharat
[@shrutibharat0105](#)



Table of Contents

1. Matplotlib
2. 2D line plots
3. Scatter plot
4. Histogram
5. Pie chart
6. Bar chart
7. Seaborn
8. Types of function
9. Relational plot
10. Distribution plots
11. Categorical plots
12. Metrix plots
13. Regression plots
14. Multigrid plots



Matplotlib

- Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python.
- It is highly customizable and allows users to create a wide variety of plots, such as line plots, bar plots, scatter plots, histograms, and more.
- Matplotlib is designed to work seamlessly with NumPy and Pandas data structures, making it easy to visualize data from these libraries.
- It serves as the foundation for many other plotting libraries in Python.

Types of data

1. **Numerical Data:** Quantitative data that can be measured.
2. **Categorical Data:** Qualitative data that can be grouped into categories.



2D Line plots

- It is used for bivariate analysis.
- One column is categorical and the other is numerical.
- Use case: Time series data.

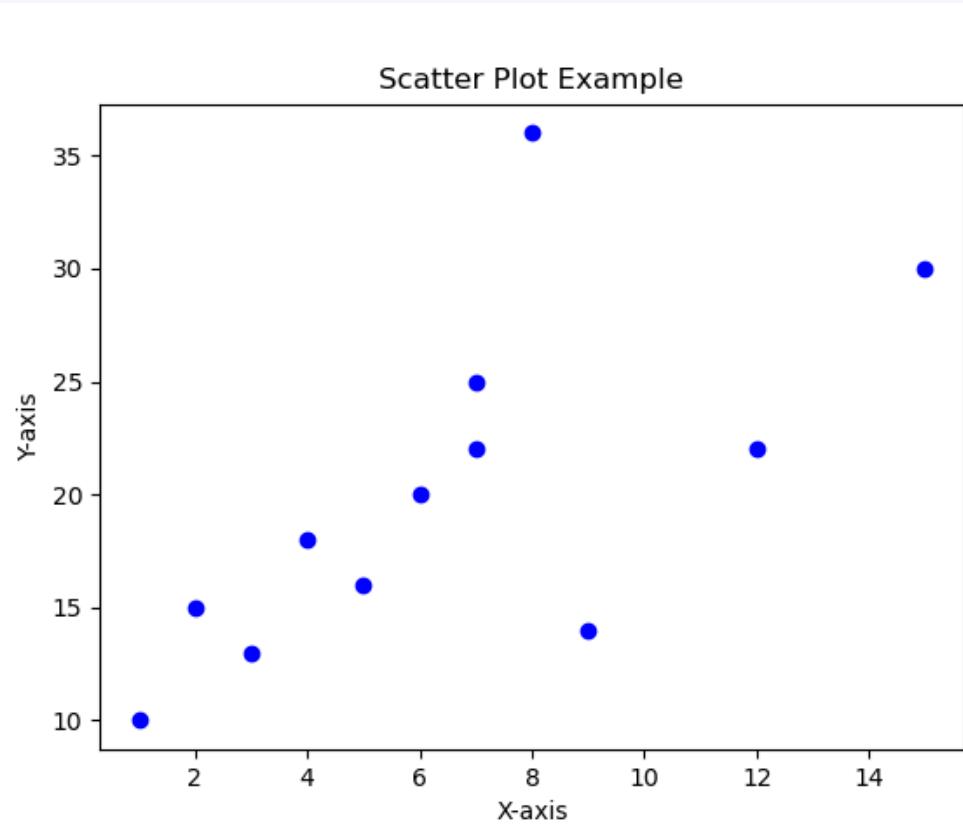
```
● ● ●  
import matplotlib.pyplot as plt  
  
# Data  
year = [2015, 2016, 2017, 2018, 2019, 2020]  
price = [48000, 54000, 57000, 49000, 47000, 45000]  
  
# Plot  
plt.plot(year, price, color='blue', linestyle='solid', linewidth=2,  
         marker='o', markersize=6)  
plt.title('Yearly Prices')  
plt.xlabel('Year')  
plt.ylabel('Price')  
plt.legend(['Price over Years'], loc='upper right')  
plt.show()
```



Scatter plot

- Used to display the relationship between two numerical variables.
- Each point represents an observation.

```
● ● ●  
import matplotlib.pyplot as plt  
  
# Data  
x = [1, 2, 3, 4, 5]  
y = [10, 15, 13, 18, 16]  
  
# Plot  
plt.scatter(x, y, color='red', marker='o')  
plt.title('Scatter Plot Example')  
plt.xlabel('X-axis')  
plt.ylabel('Y-axis')  
plt.show()
```



Histogram

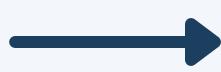
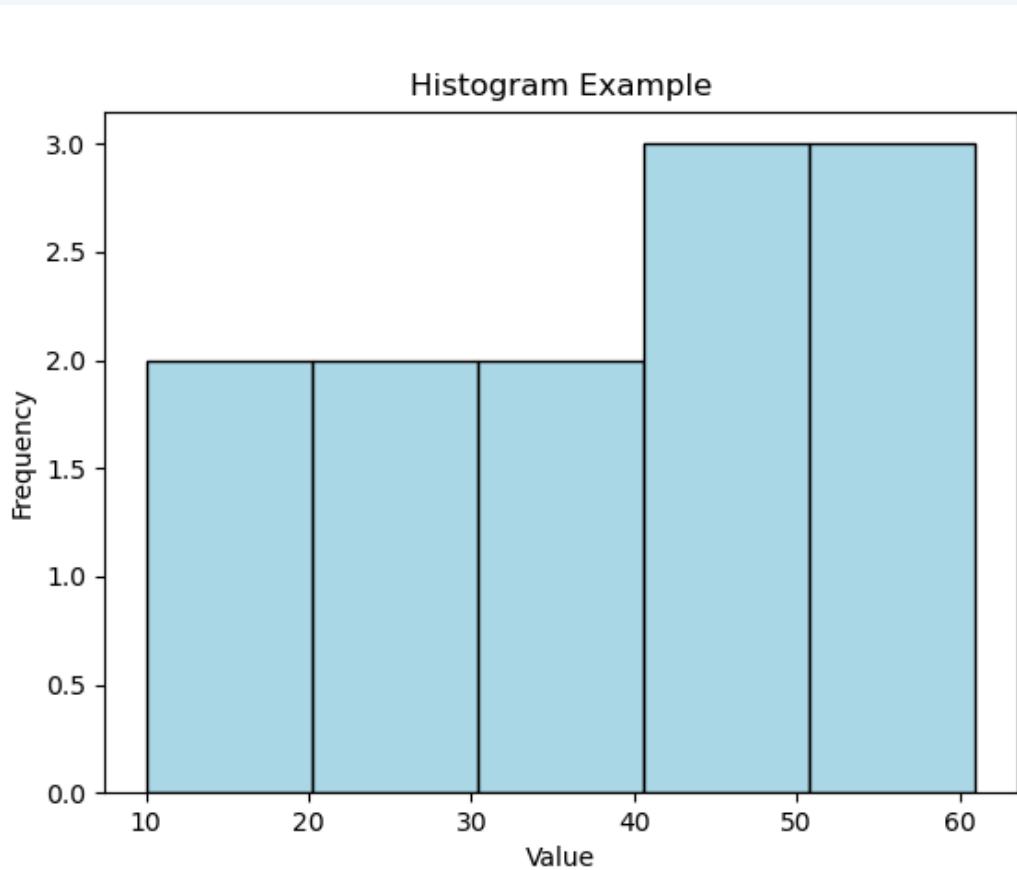
- Used to display the distribution of a numerical variable.
- Data is divided into bins.

```
● ● ●

import matplotlib.pyplot as plt

# Data
data = [32, 45, 56, 10, 15, 27, 61, 23, 42, 33, 55, 44]

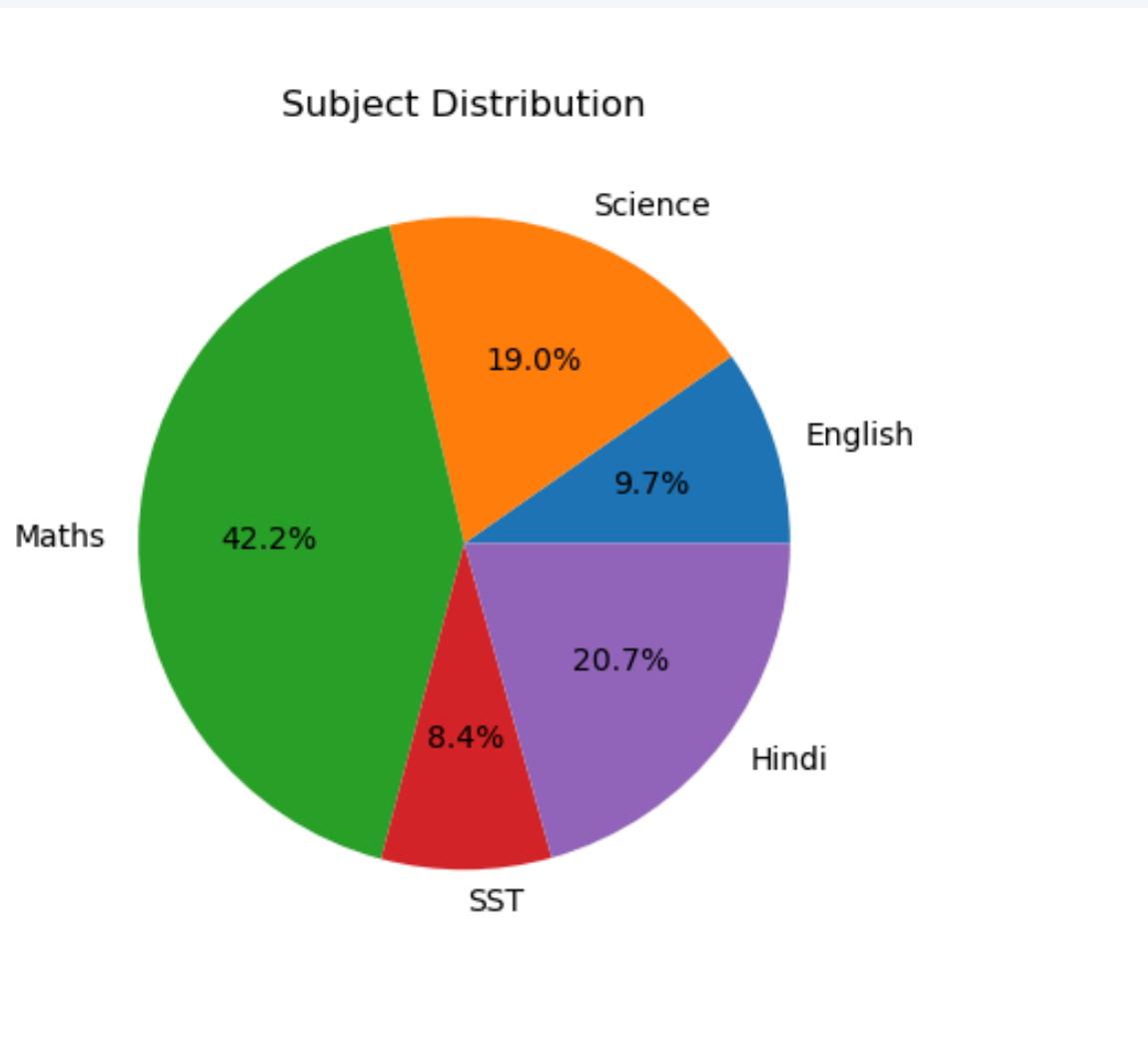
# Plot
plt.hist(data, bins=5, color='green', edgecolor='black')
plt.title('Histogram Example')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



Pie chart

- Represents data in a circular graph.
- Useful for showing proportions and contributions of parts to a whole.

```
● ● ●  
import matplotlib.pyplot as plt  
  
# Data  
data = [23, 45, 100, 20, 49]  
subjects = ['English', 'Science', 'Maths', 'SST', 'Hindi']  
  
# Plot  
plt.pie(data, labels=subjects, autopct='%0.1f%%')  
plt.title('Subject Distribution')  
plt.show()
```

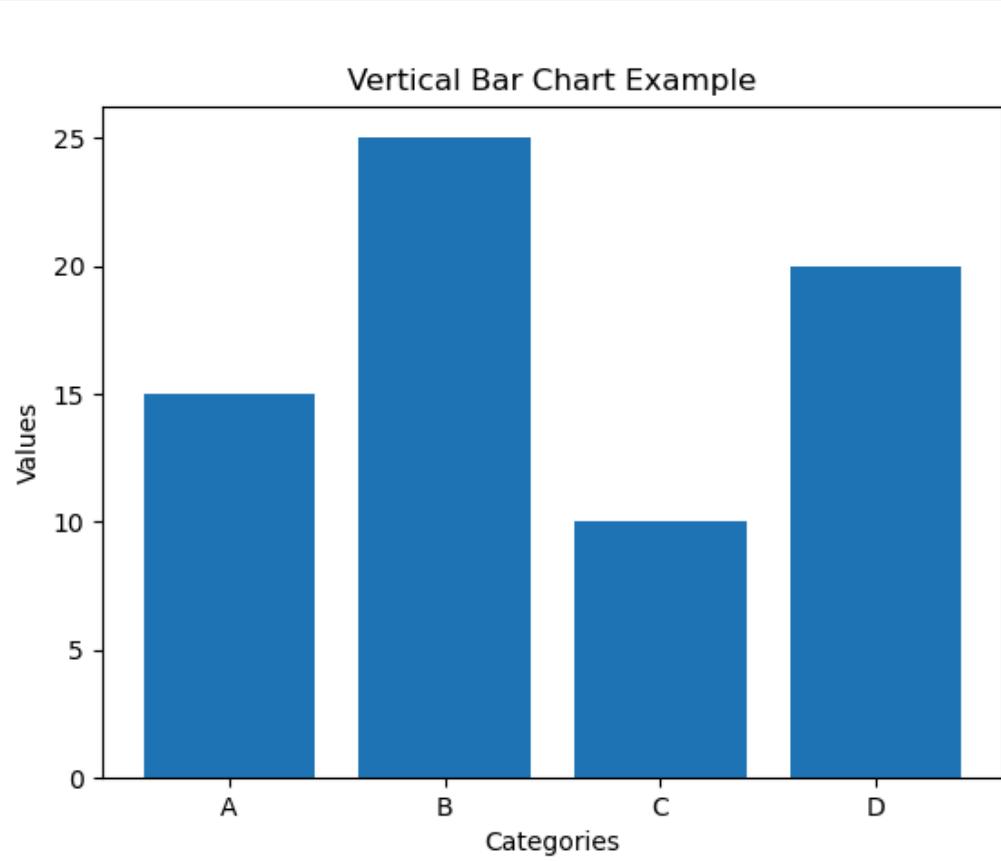


Bar Chart

3 Types of Bar chart:

1. Vertical bar chart (By default)
2. Horizontal bar chart
3. Stacked bar chart

```
● ● ●  
# Example data  
categories = ['A', 'B', 'C', 'D']  
values = [15, 25, 10, 20]  
  
# Plotting the vertical bar chart  
plt.bar(categories, values)  
plt.xlabel('Categories')  
plt.ylabel('Values')  
plt.title('Vertical Bar Chart Example')  
plt.savefig('vertical_barchart')
```

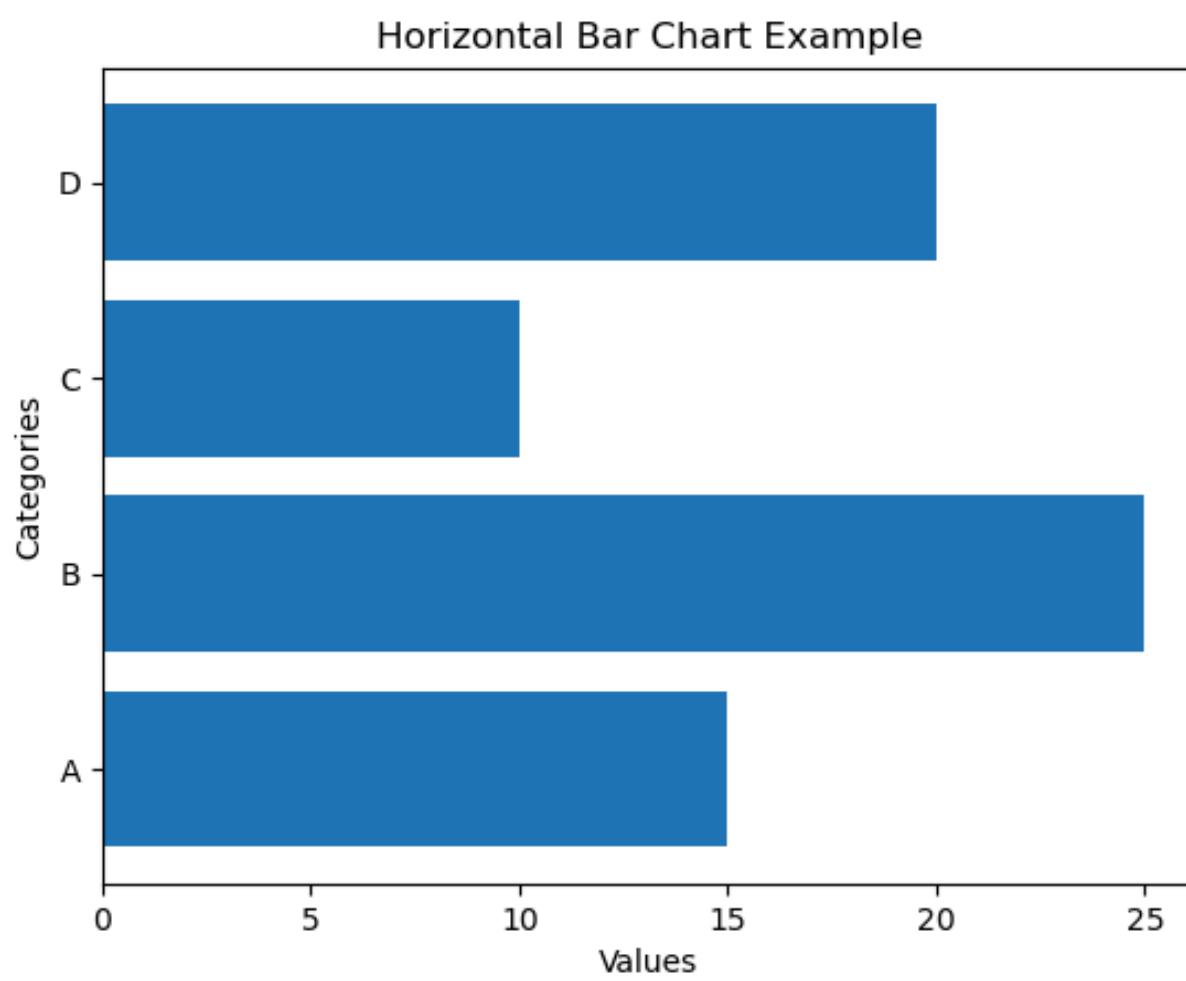


Bar Chart



```
categories = ['A', 'B', 'C', 'D']
values = [15, 25, 10, 20]

# Plotting the horizontal bar chart
plt.barh(categories, values)
plt.xlabel('Values')
plt.ylabel('Categories')
plt.title('Horizontal Bar Chart Example')
plt.savefig('horizontal_barchart')
```



Bar Chart

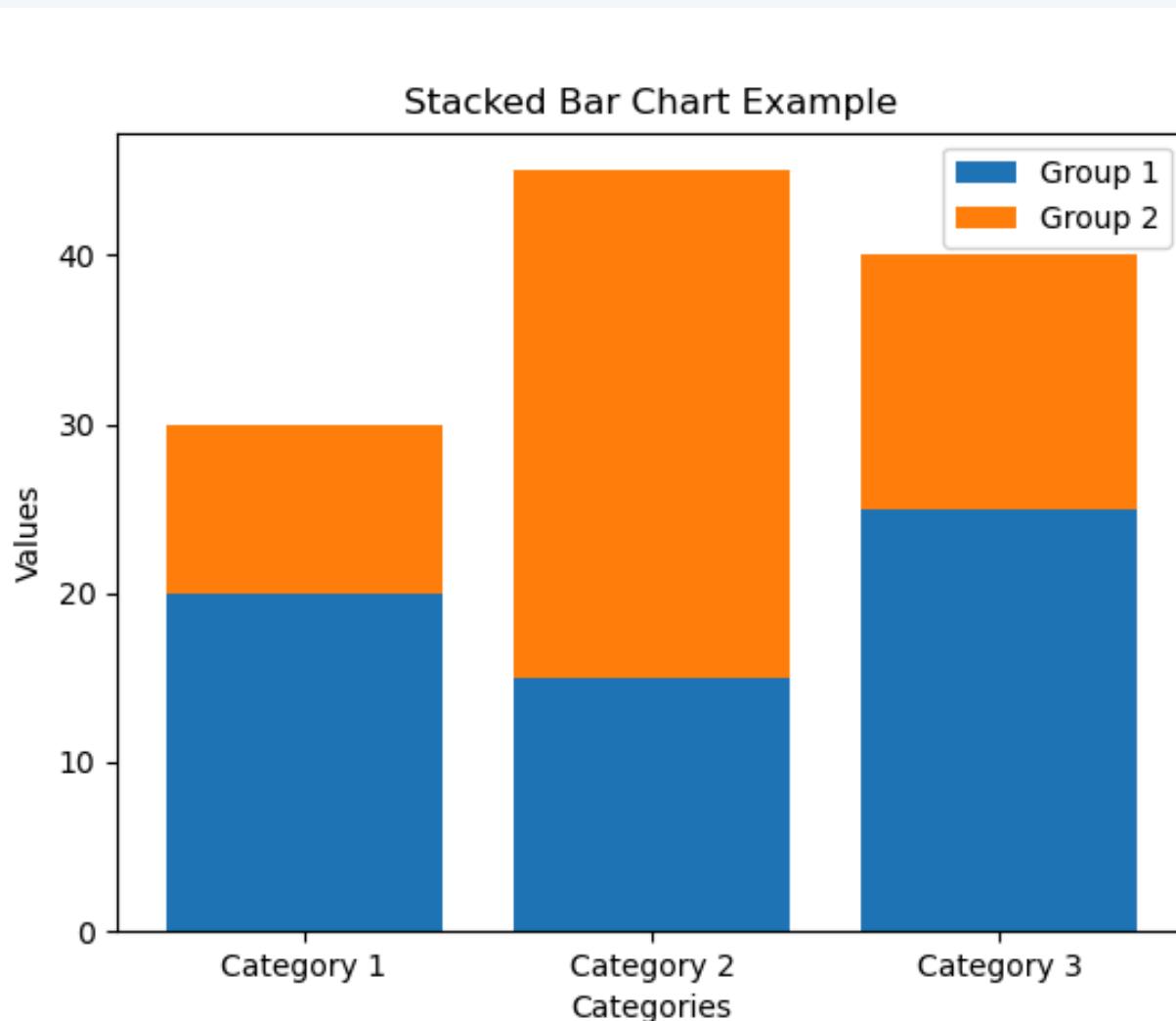


```
categories = ['Category 1', 'Category 2', 'Category 3']
values1 = [20, 15, 25] # Values for the first stack
values2 = [10, 30, 15] # Values for the second stack

# Plotting the stacked bar chart
plt.bar(categories, values1, label='Group 1')
plt.bar(categories, values2, bottom=values1, label='Group 2')

plt.xlabel('Categories')
plt.ylabel('Values')
plt.title('Stacked Bar Chart Example')
plt.legend()

plt.savefig('stacked_barchart')
```



Seaborn

Seaborn is a Python data visualization library built on top of Matplotlib. It provides a high-level interface for drawing attractive and informative statistical graphics.

Types of Functions in Seaborn

1. Figure-Level Functions:

- These functions manage the creation of entire figures, including setting up the axes, figure aesthetics, and the layout.
- Example: `sns.catplot()`, `sns.relplot()`

2. Axes-Level Functions:

- These functions work at the level of individual plots, allowing for more granular control of plot elements.
- Example: `sns.barplot()`, `sns.scatterplot()`, `sns.boxplot()`



Seaborn

Classification of Seaborn Plots

- **Relational Plots:**

- **relplot()**: Combines scatterplot() and lineplot() into a single **figure-level** function.
- **scatterplot()**: Creates scatter plots.
- **lineplot()**: Creates line plots.

- **Distribution Plots:**

- **distplot() (deprecated, use histplot() or kdeplot())**: Creates histograms and KDEs. (Figure level function)
- **histplot()**: Creates histograms.
- **kdeplot()**: Creates Kernel Density Estimate plots.



Seaborn

- **Regression Plots:**

- **Implot():** Combines regplot() into a **figure-level function.**
- **regplot():** Creates scatter plots with regression lines.
- **residplot():** Creates residual plots.

- **Categorical Plots:**

- **catplot():** Combines various categorical plots into a single figure-level function.
- **stripplot():** Creates strip plots.
- **swarmplot():** Creates swarm plots.
- **boxplot():** Creates box plots.
- **violinplot():** Creates violin plots.
- **barplot():** Creates bar plots.
- **pointplot():** Creates point plots.
- **countplot():** Show the counts of observations in each categorical bin using bars.



Seaborn

- **Matrix Plots:**

1. No figure level function.
2. **heatmap():** Creates heatmaps.
3. **clustermap():** Creates clustered heatmaps.

- **Multi-Grid Plots:**

1. **FacetGrid:** Facilitates the creation of multi-plot grids.
2. **PairGrid:** Facilitates the creation of pairwise plot grids.
3. **JointGrid:** Facilitates bivariate plot with marginal univariate plots.

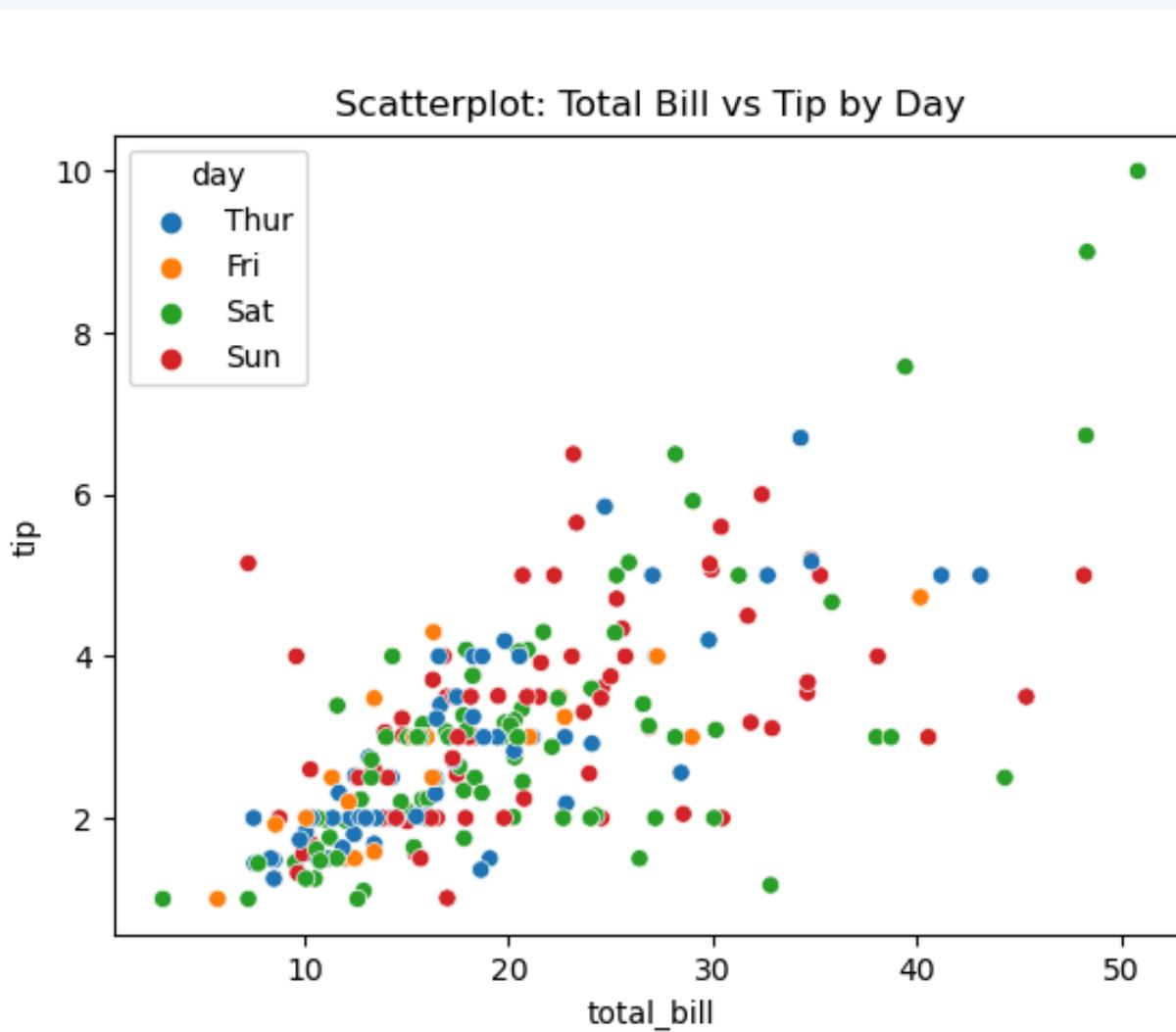


Seaborn

Relational Plots:

- **scatterplot():**

```
● ● ●  
import seaborn as sns  
import matplotlib.pyplot as plt  
  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Scatterplot  
sns.scatterplot(data=tips, x='total_bill', y='tip', hue='day')  
plt.title('Scatterplot: Total Bill vs Tip by Day')  
plt.savefig('Scatter_plot')
```

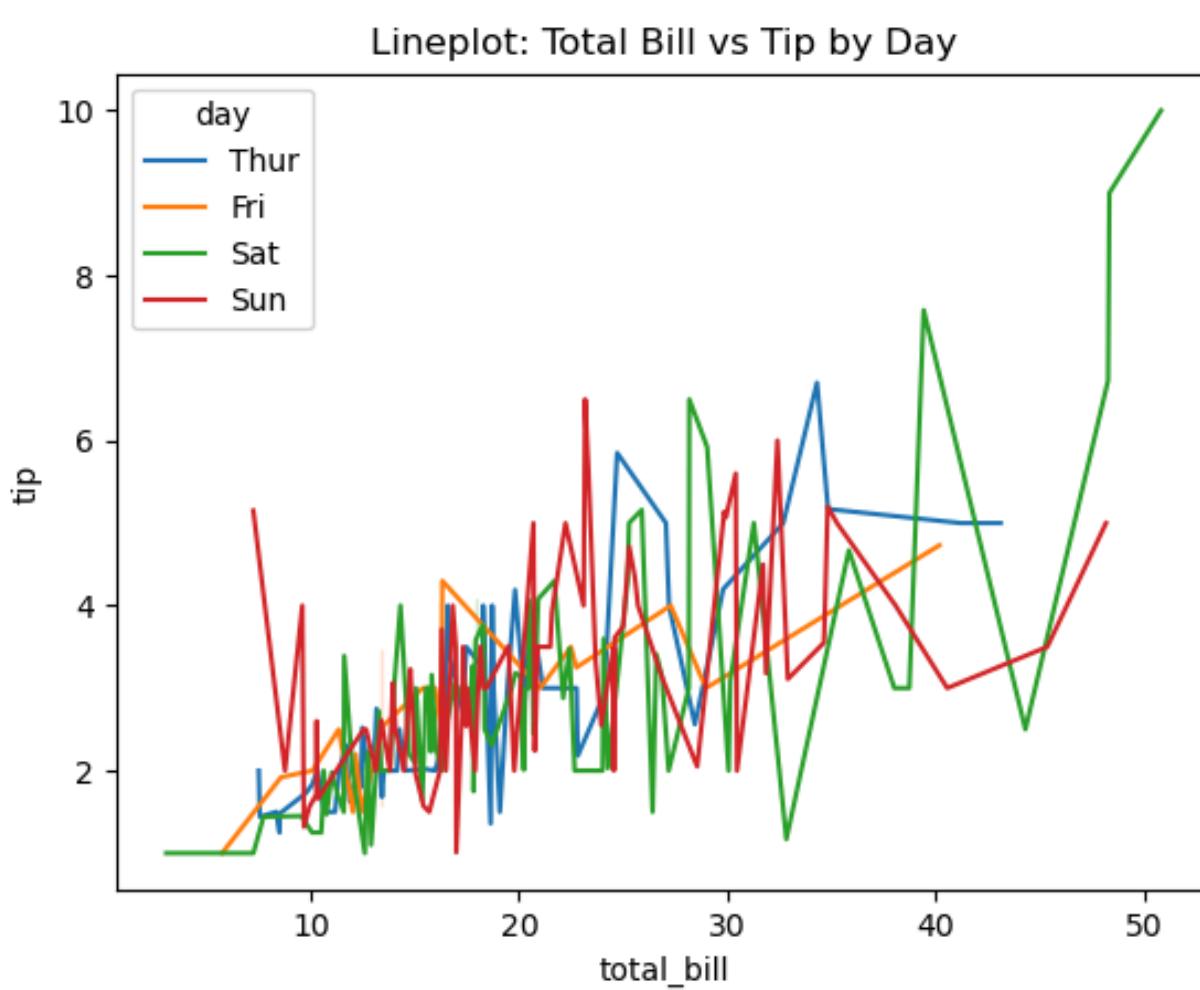


Seaborn

Relational Plots:

- **lineplot():**

```
● ● ●  
# Creating a Lineplot  
sns.lineplot(data=tips, x='total_bill', y='tip', hue='day')  
plt.title('Lineplot: Total Bill vs Tip by Day')  
plt.savefig('Line_plot')
```

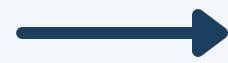
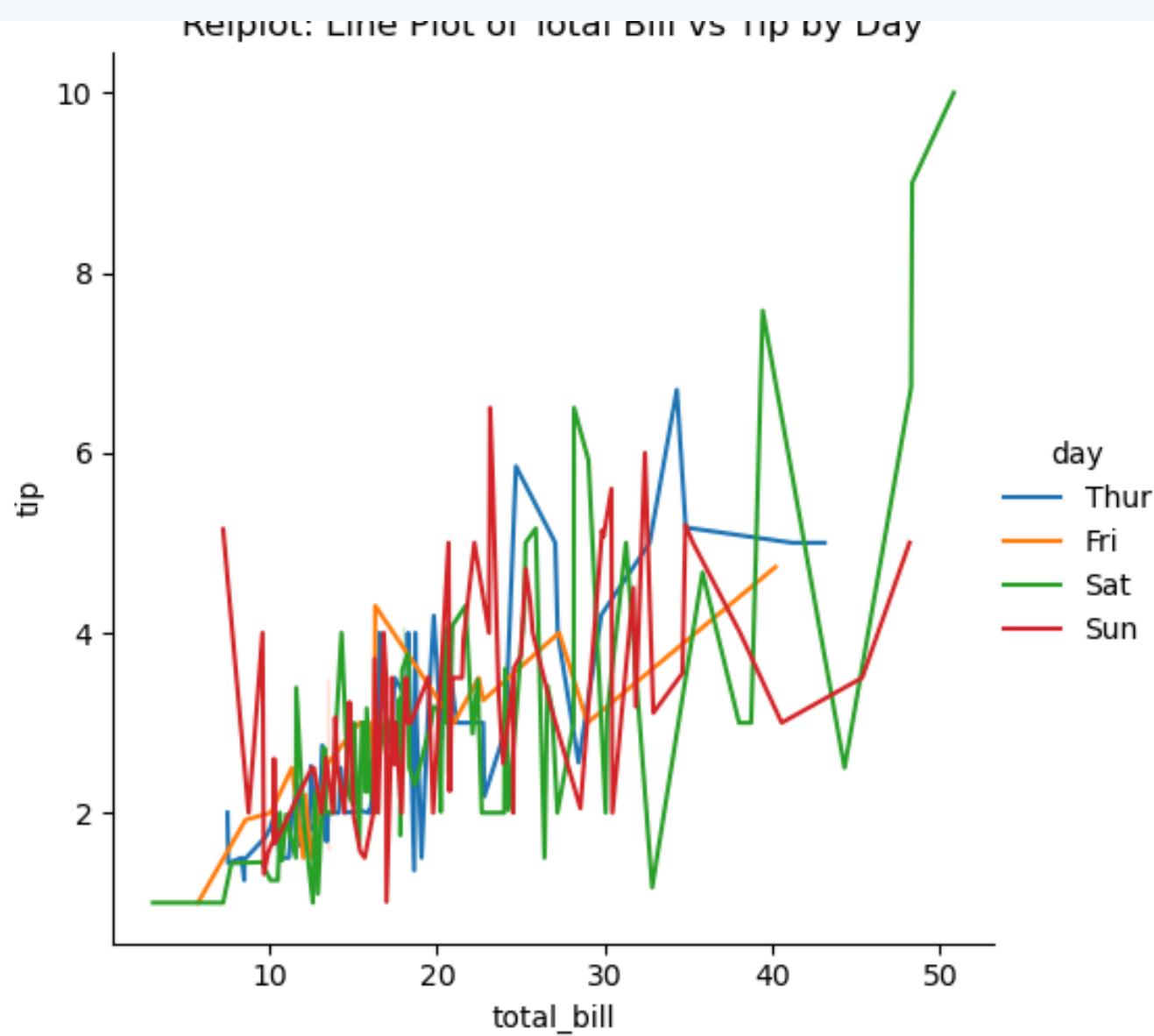


Seaborn

Relational Plots:

- **relplot(): (Figure level function)**

```
● ● ●  
# Creating a Relplot (line)  
sns.relplot(data=tips, x='total_bill', y='tip', hue='day',  
             kind='line')  
plt.title('Relplot: Line Plot of Total Bill vs Tip by Day')  
plt.savefig('relplot')
```



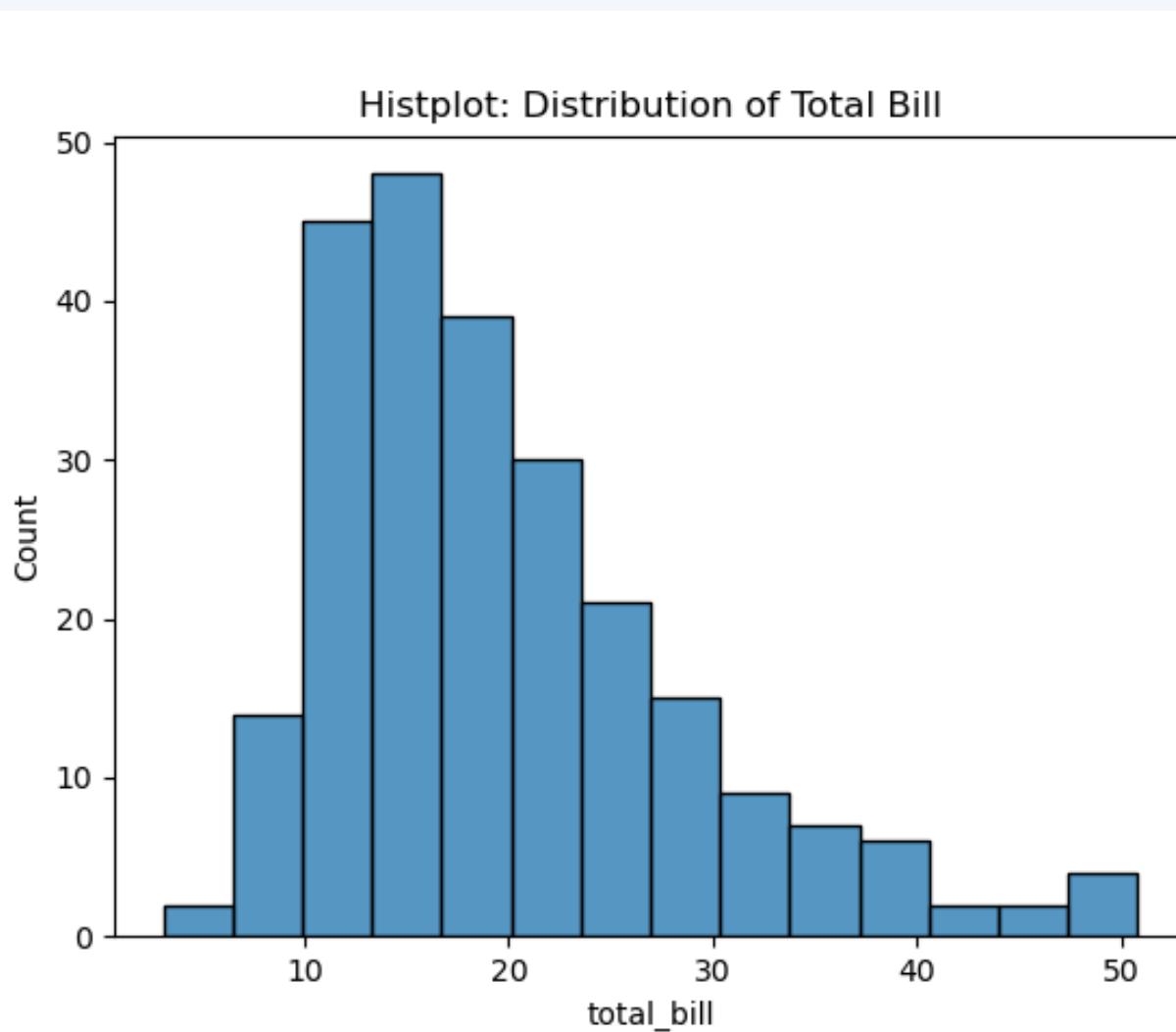
Seaborn

Distribution Plots:

- **histplot():**

```
# Load example dataset
tips = sns.load_dataset('tips')

# Creating a Histplot
sns.histplot(data=tips, x='total_bill', kde=True)
plt.title('Histplot: Distribution of Total Bill with KDE')
plt.show()
```



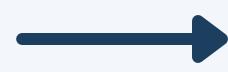
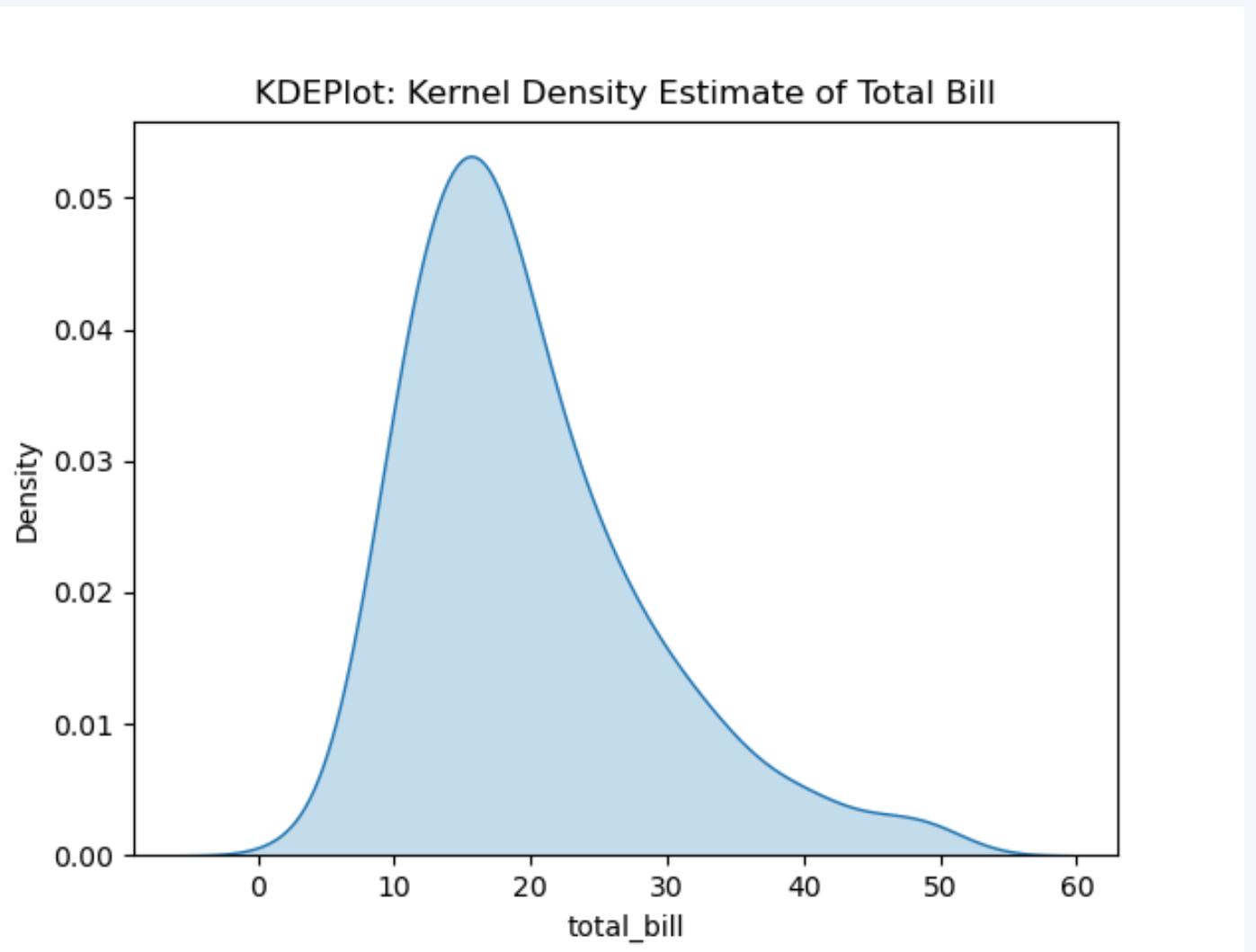
Seaborn

Distribution Plots:

- **kdeplot():**

```
# Load example dataset
tips = sns.load_dataset('tips')

# Creating a KDEPlot
sns.kdeplot(data=tips, x='total_bill', shade=True)
plt.title('KDEPlot: Kernel Density Estimate of Total Bill')
plt.show()
```



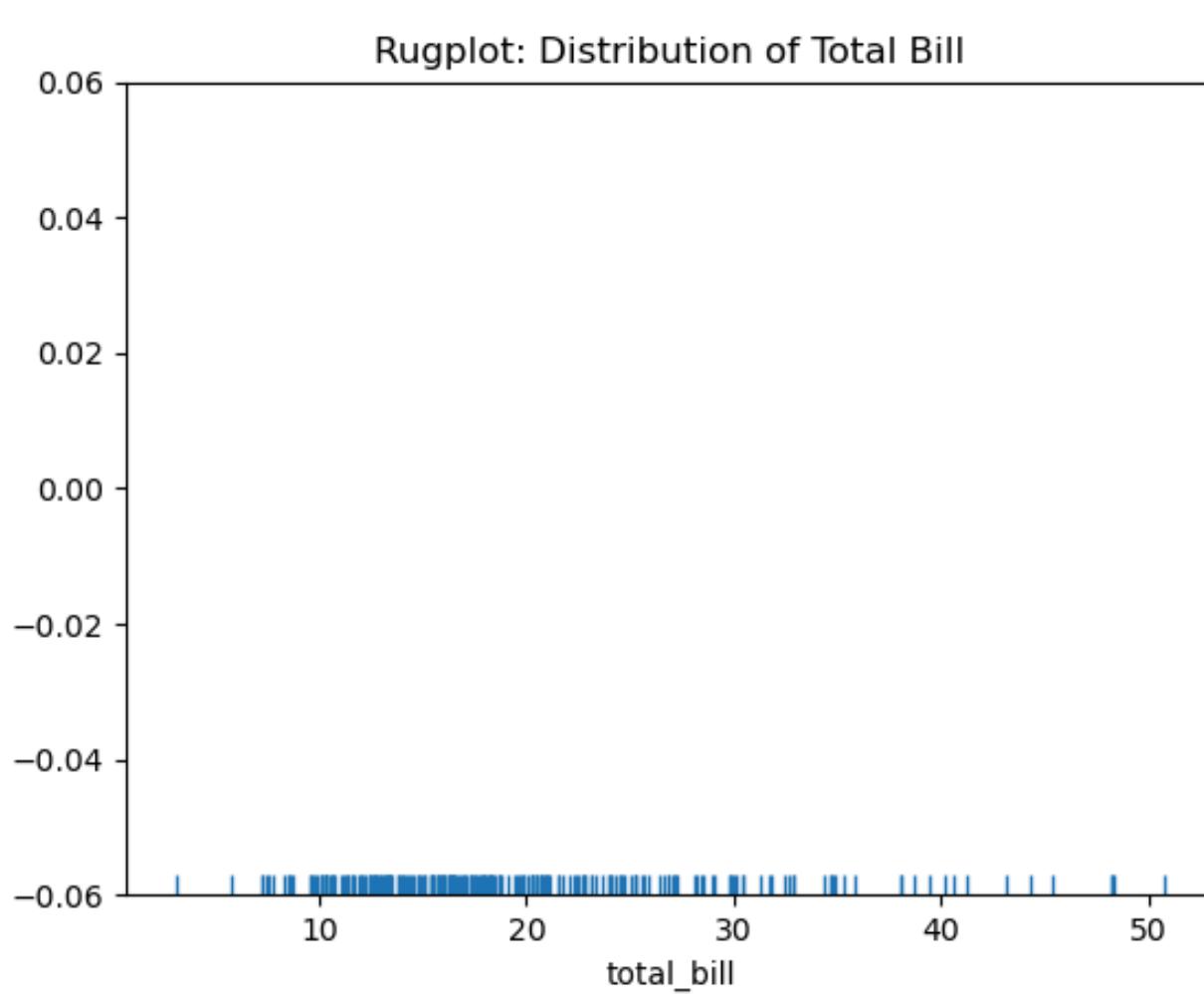
Seaborn

Distribution Plots:

- **rugplot():**

```
# Load example dataset
tips = sns.load_dataset('tips')

# Creating a Rugplot
sns.rugplot(data=tips, x='total_bill')
plt.title('Rugplot: Distribution of Total Bill')
plt.show()
```



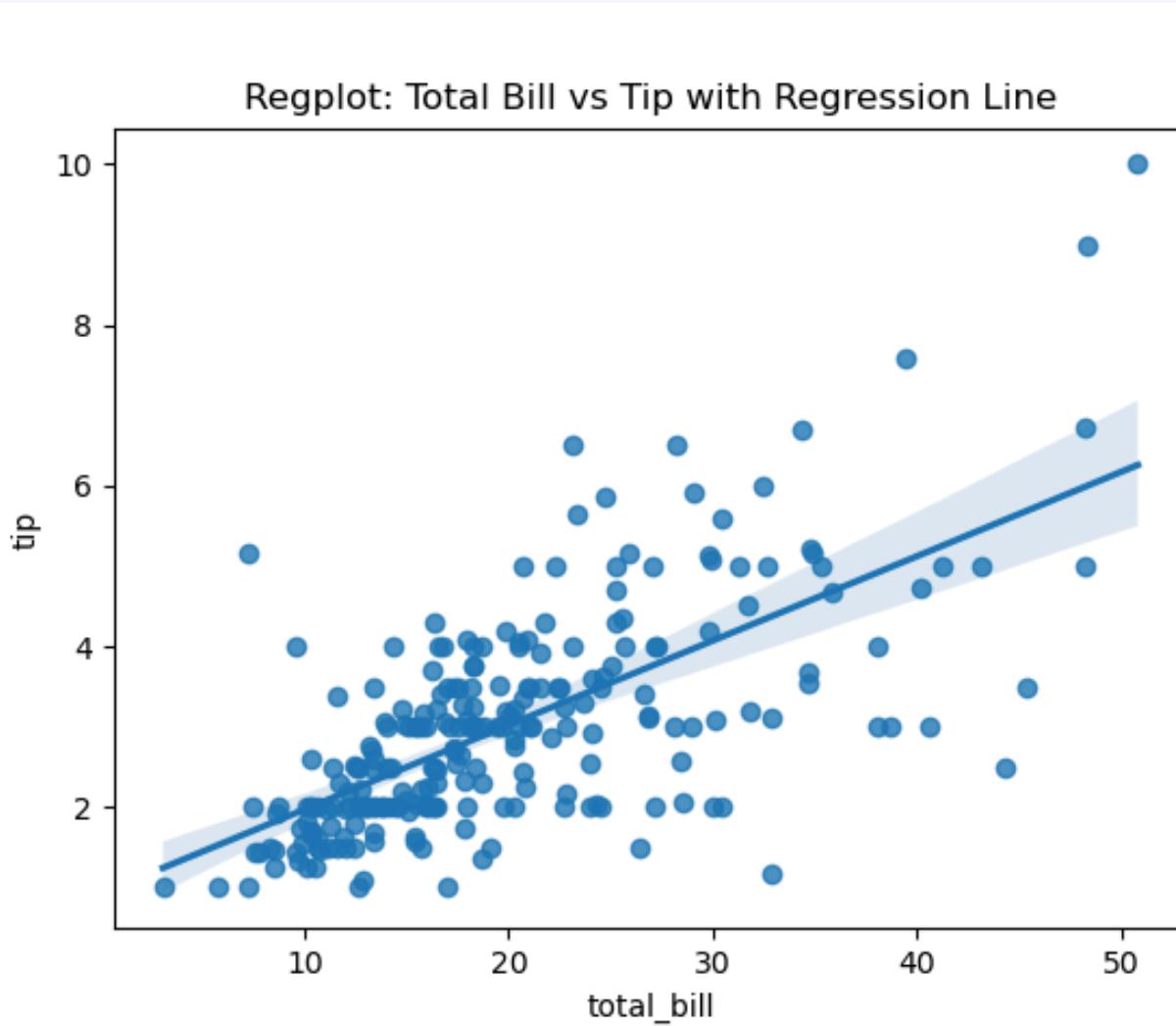
Seaborn

Regression Plots:

- **regplot():**

```
# Load example dataset
tips = sns.load_dataset('tips')

# Creating a Regplot
sns.regplot(data=tips, x='total_bill', y='tip')
plt.title('Regplot: Total Bill vs Tip with Regression Line')
plt.show()
```



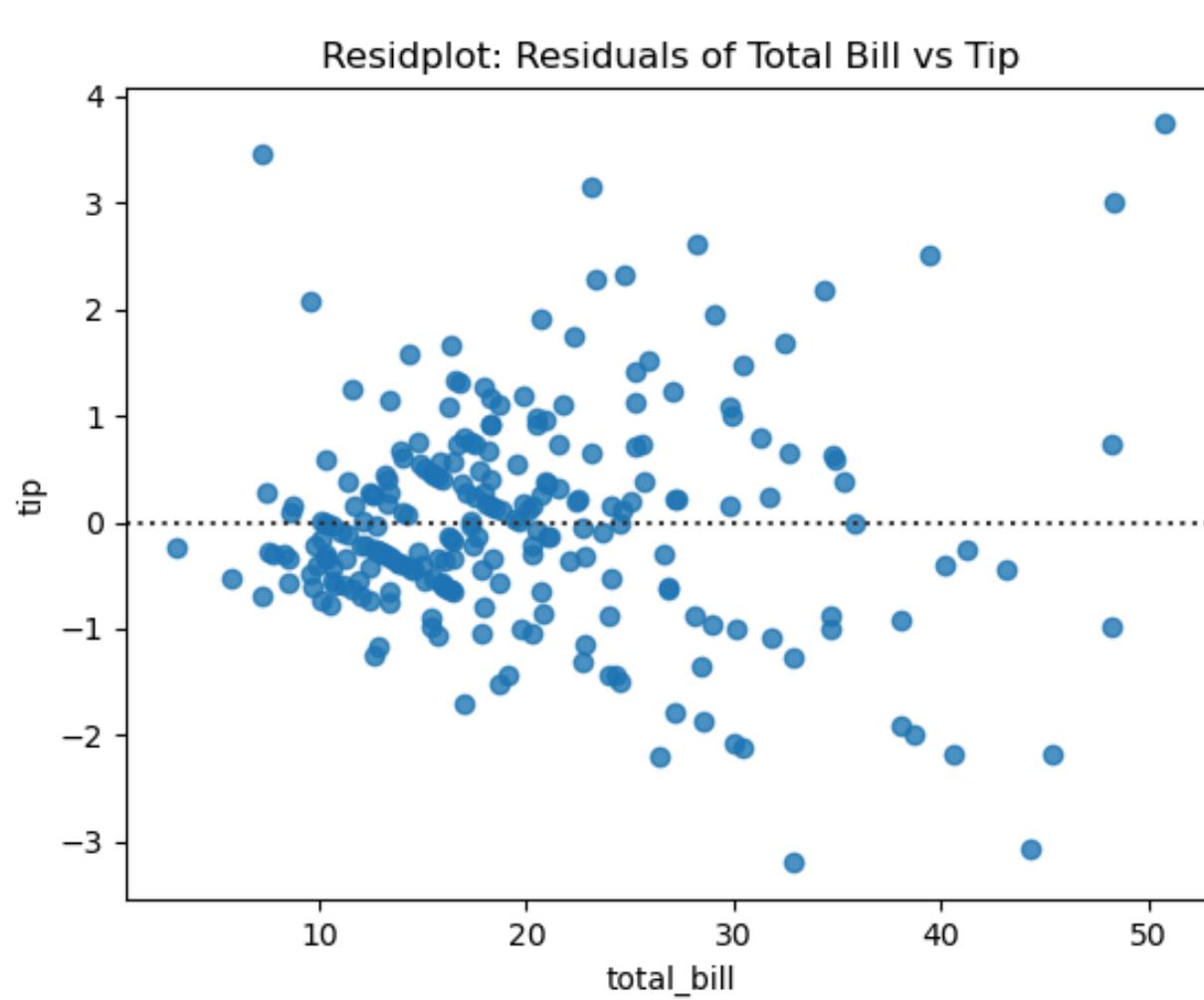
Seaborn

Regression Plots:

- **residplot():**

```
# Load example dataset
tips = sns.load_dataset('tips')

# Creating a Residplot
sns.residplot(data=tips, x='total_bill', y='tip')
plt.title('Residplot: Residuals of Total Bill vs Tip')
plt.show()
```

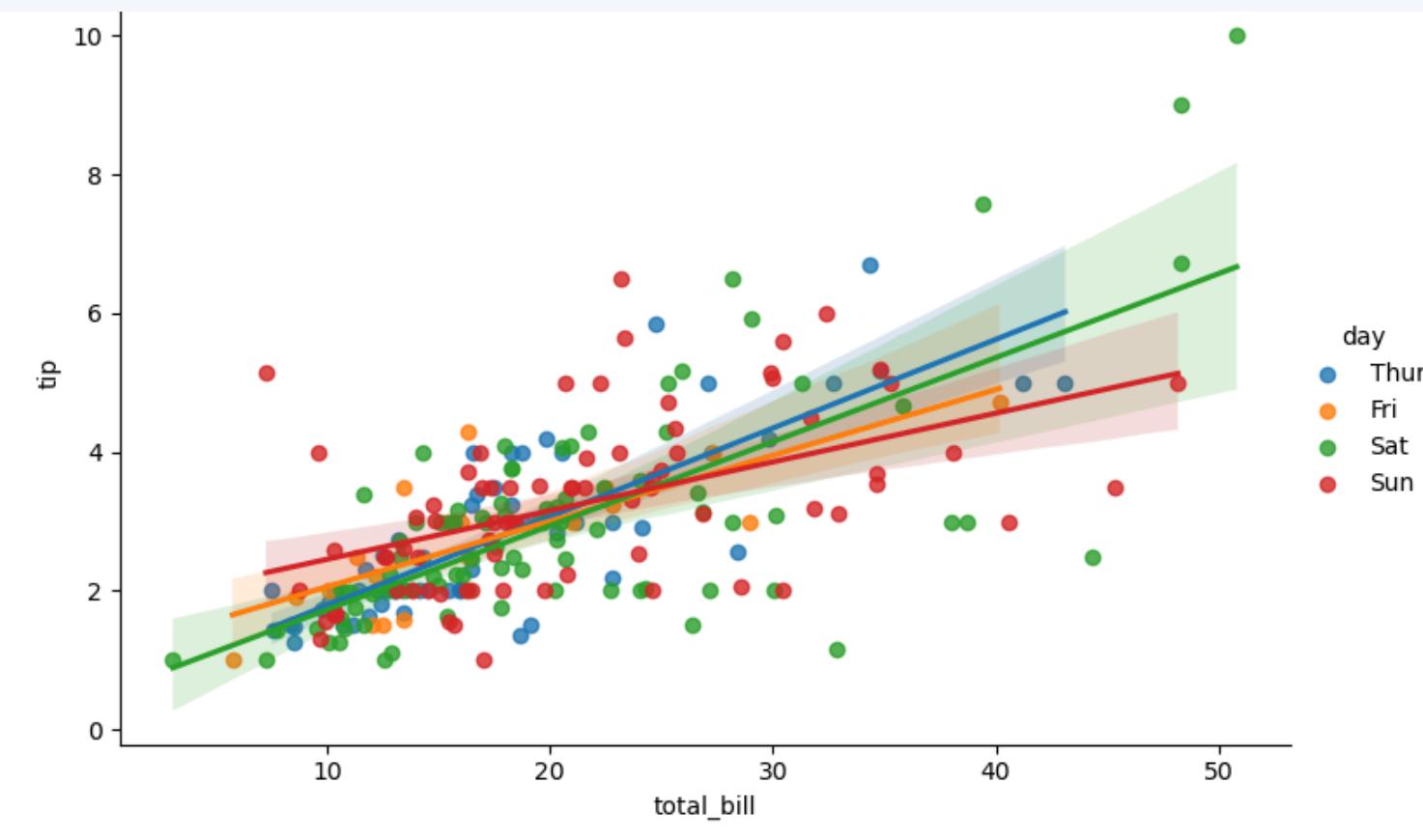


Seaborn

Regression Plots:

- **lmplot(): (Figure level function)**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating an Lmplot  
sns.lmplot(data=tips, x='total_bill', y='tip', hue='day', aspect=1.5)  
plt.title('Lmplot: Total Bill vs Tip with Regression Line by Day')  
plt.show()
```

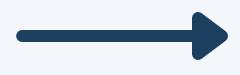
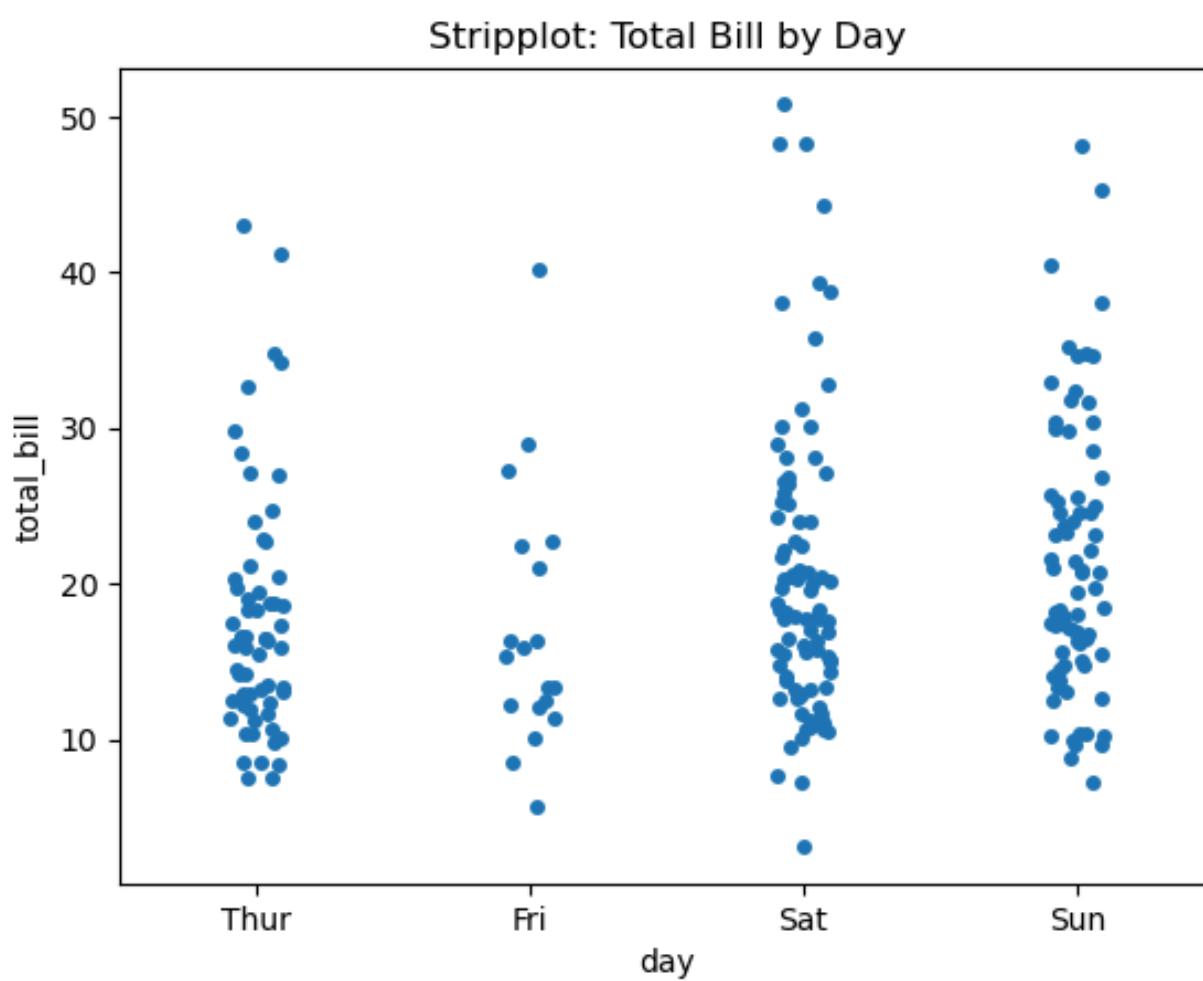


Seaborn

Categorical Scatter Plots:

- **Stripplot():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Stripplot  
sns.stripplot(data=tips, x='day', y='total_bill', jitter=True)  
plt.title('Stripplot: Total Bill by Day')  
plt.show()
```

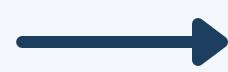
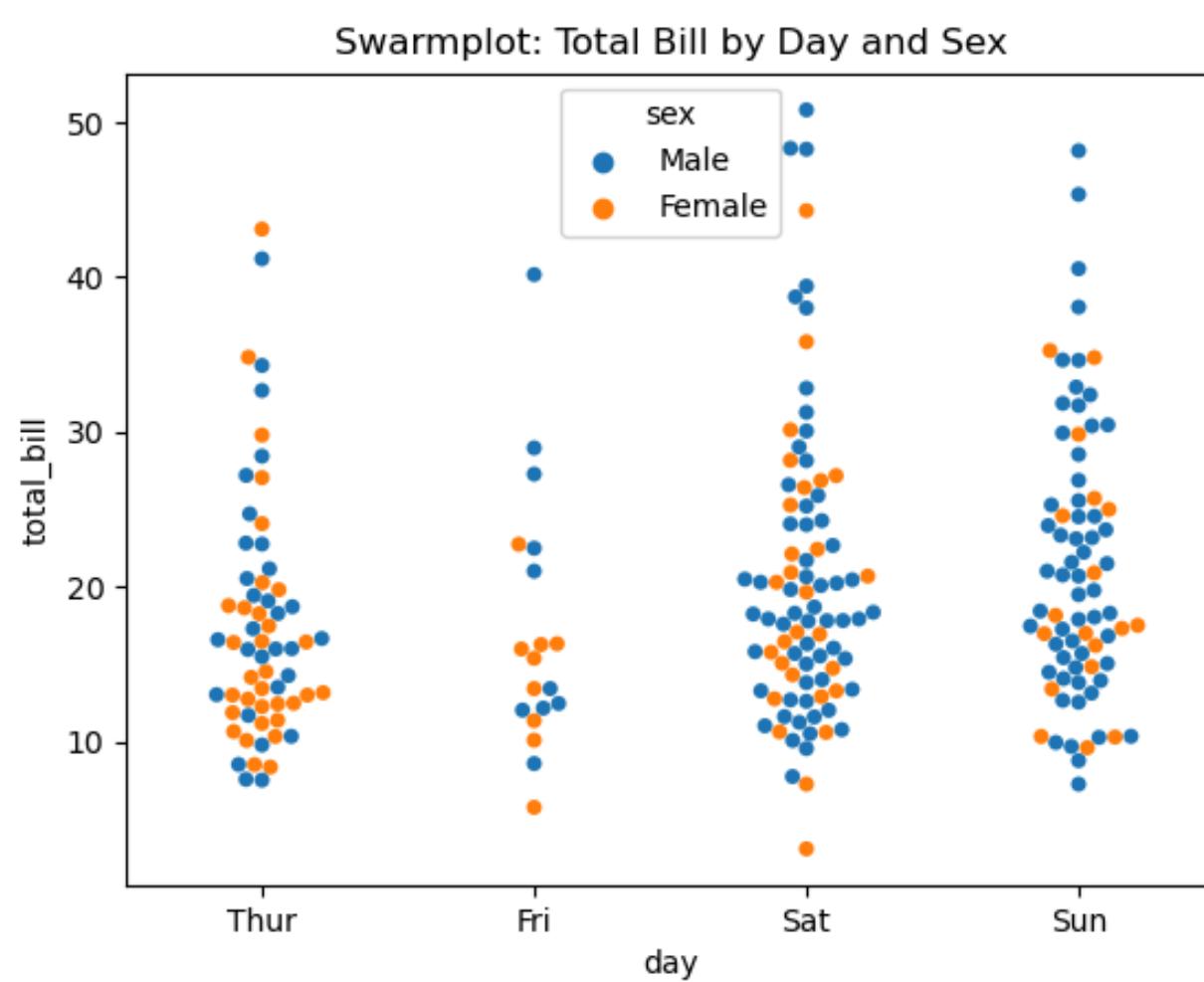


Seaborn

Categorical Scatter Plots:

- **Swarmplot():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Swarmplot  
sns.swarmplot(data=tips, x='day', y='total_bill')  
plt.title('Swarmplot: Total Bill by Day')  
plt.show()
```

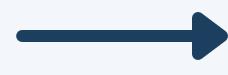
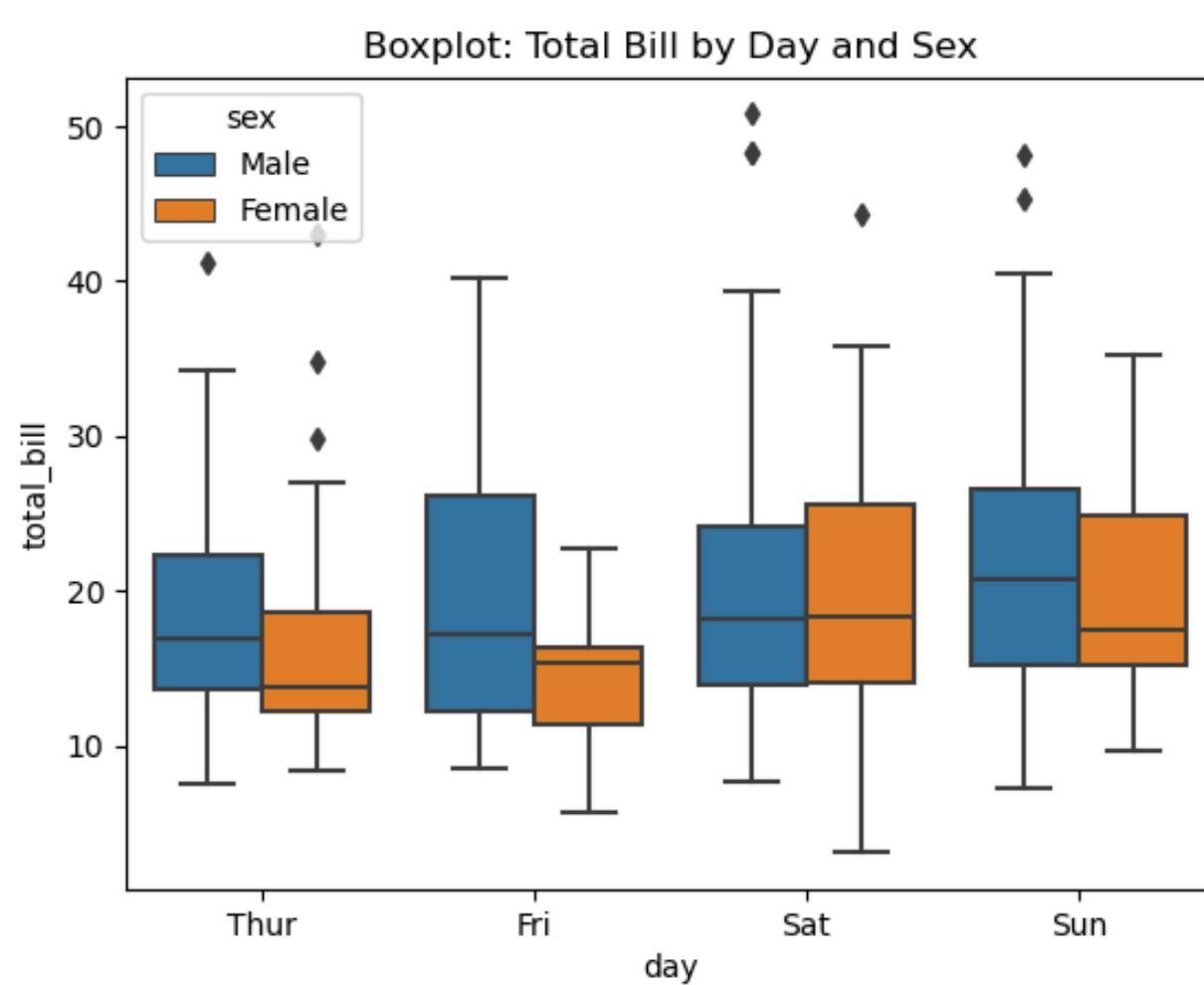


Seaborn

Categorical Distribution Plots:

- **Boxplot():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Boxplot  
sns.boxplot(data=tips, x='day', y='total_bill')  
plt.title('Boxplot: Total Bill by Day')  
plt.show()
```

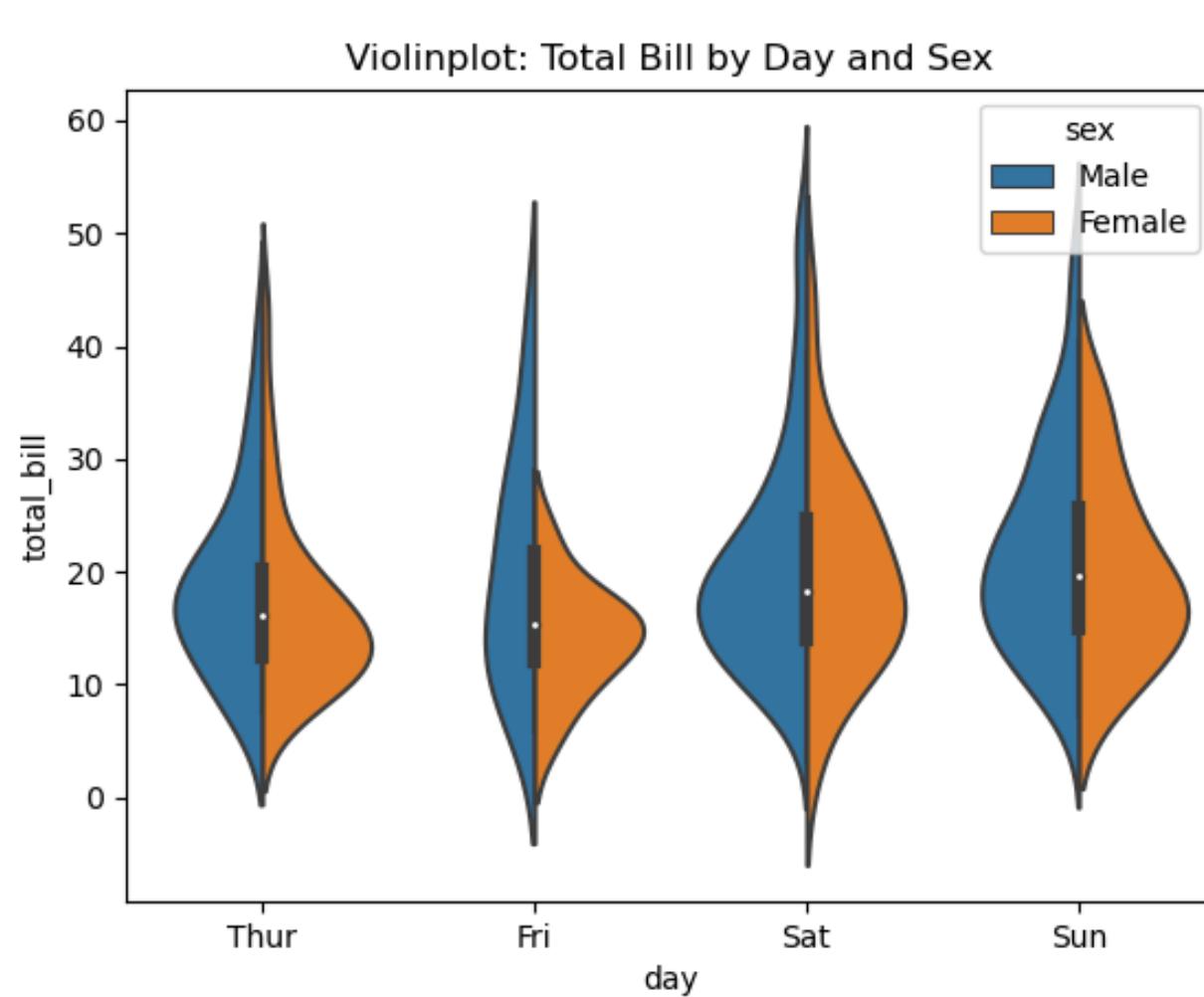


Seaborn

Categorical Distribution Plots:

- **Violinplot():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Violinplot  
sns.violinplot(data=tips, x='day', y='total_bill')  
plt.title('Violinplot: Total Bill by Day')  
plt.show()
```

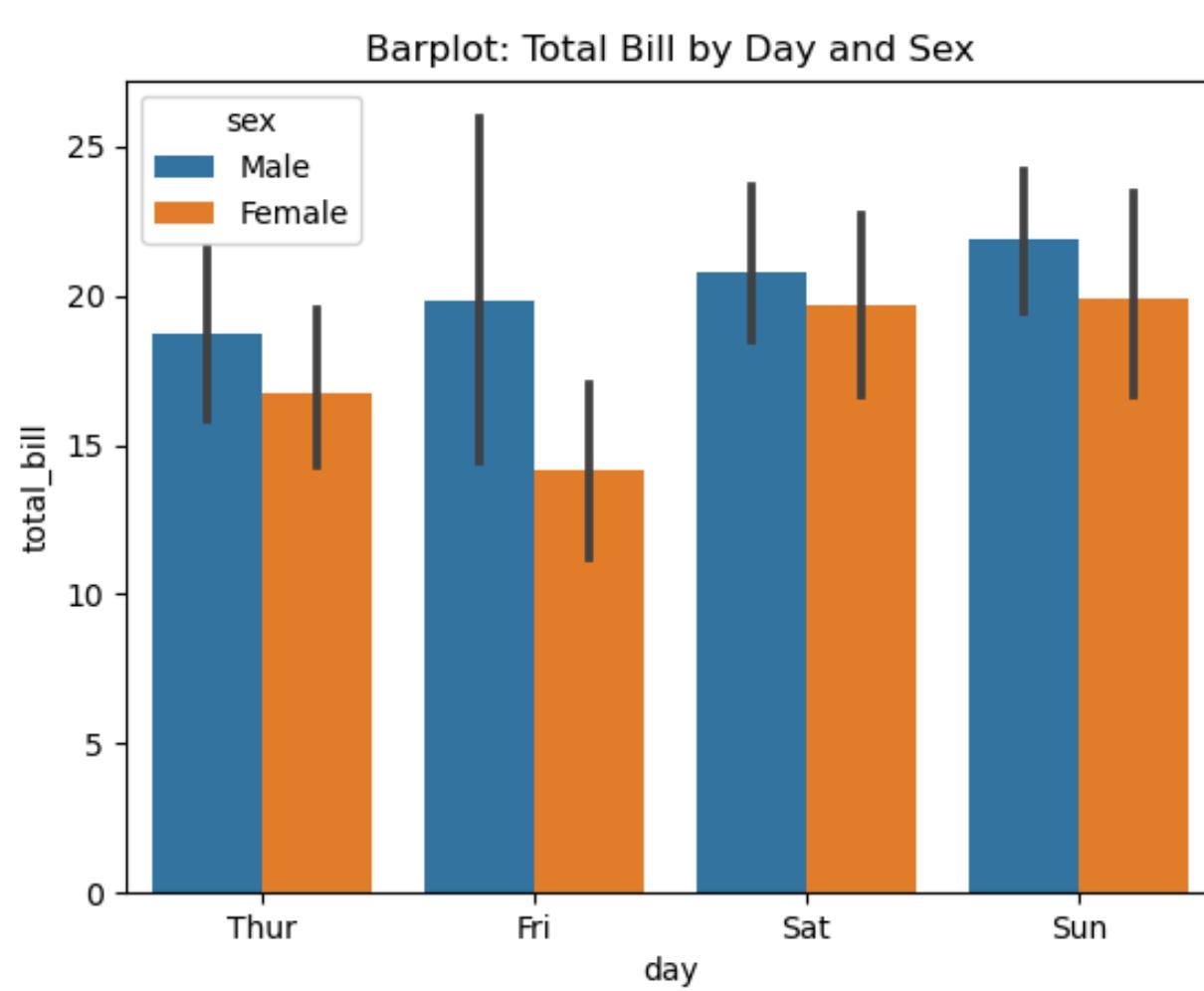


Seaborn

Categorical Estimate Plots:

- Barplot():

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Barplot  
sns.barplot(data=tips, x='day', y='total_bill')  
plt.title('Barplot: Total Bill by Day')  
plt.show()
```

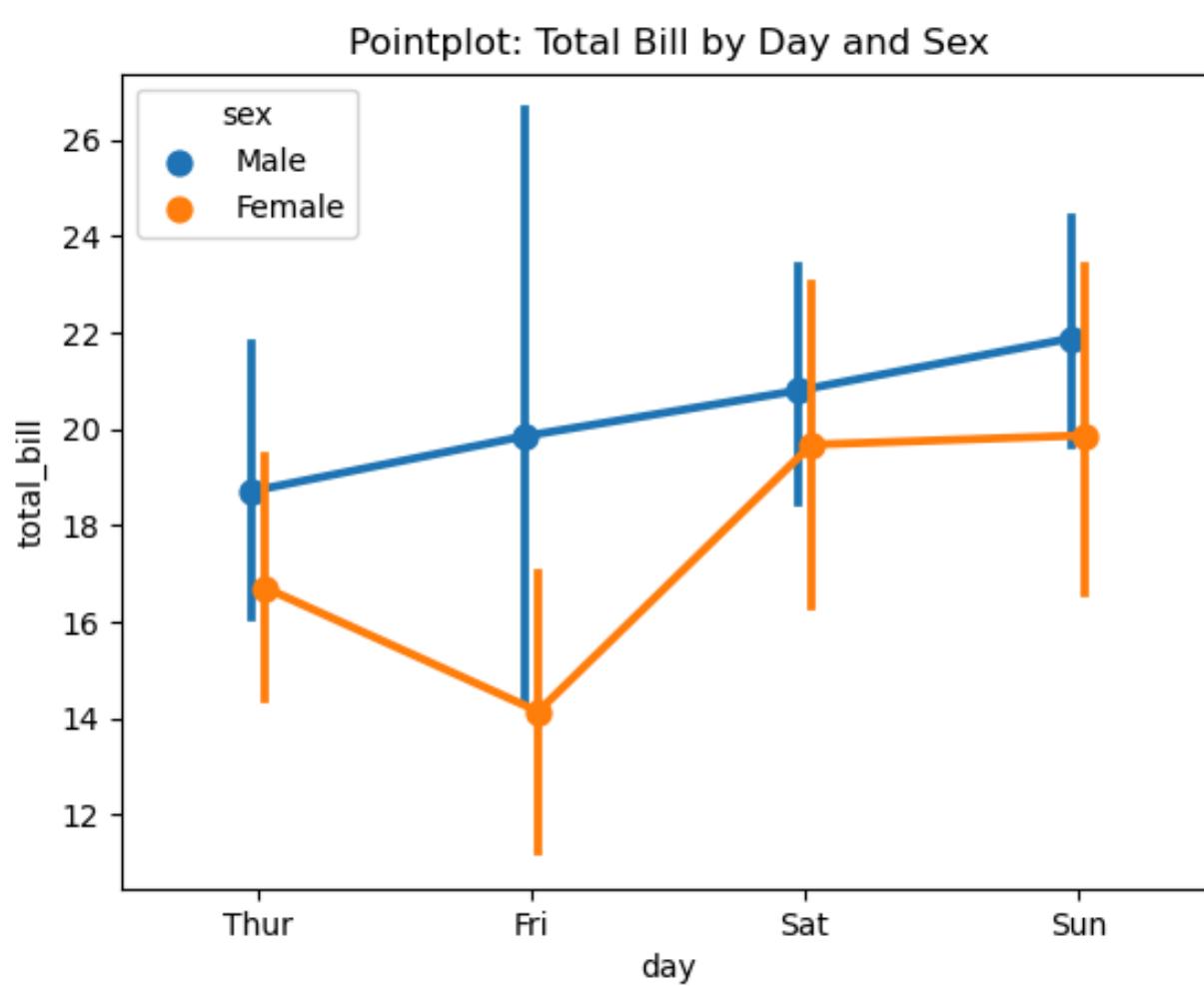


Seaborn

Categorical Estimate Plots:

- **Pointplot():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Pointplot  
sns.pointplot(data=tips, x='day', y='total_bill')  
plt.title('Pointplot: Total Bill by Day')  
plt.show()
```



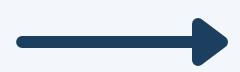
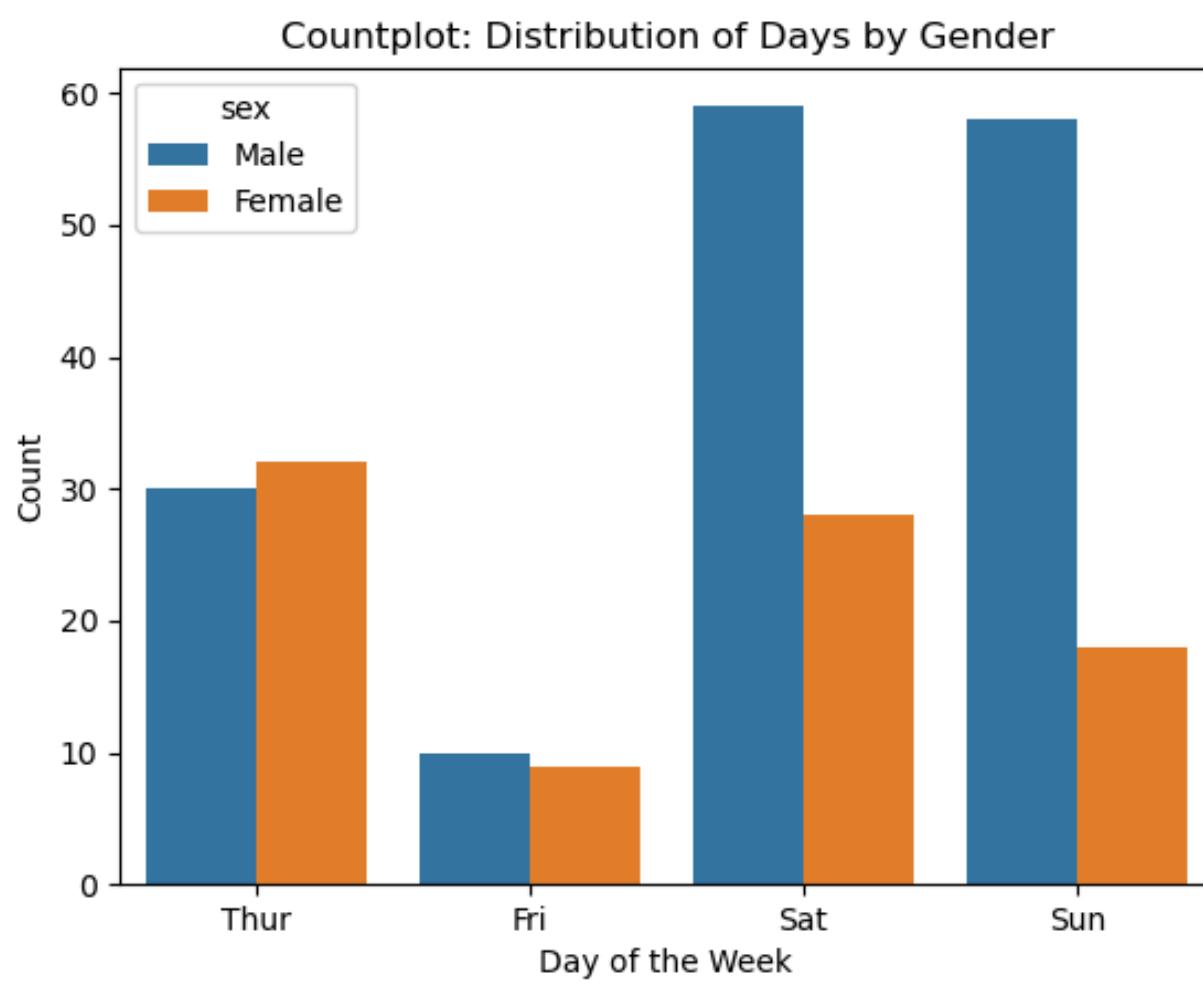
Seaborn

Categorical Estimate Plots:

- **Countplot():**



```
# Creating a Countplot
sns.countplot(data=tips, x='day', hue='sex')
plt.xlabel('Day of the Week')
plt.ylabel('Count')
plt.title('Countplot: Distribution of Days by Gender')
plt.show()
```

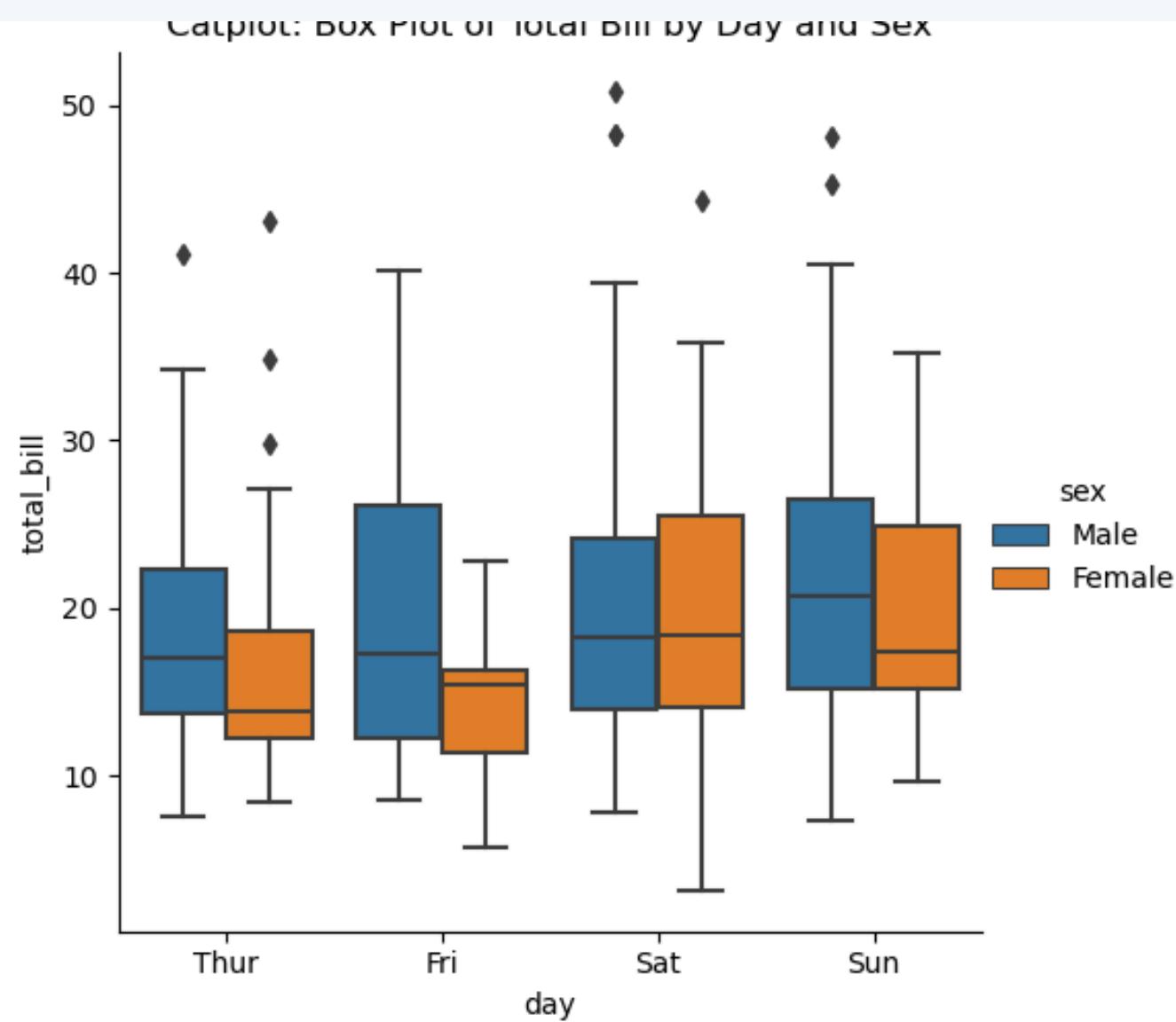


Seaborn

Categorical Estimate Plots:

- **Catplot(): (Figure level function)**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Catplot  
sns.catplot(data=tips, x='day', y='total_bill', kind='box', hue='sex')  
plt.title('Catplot: Box Plot of Total Bill by Day and Sex')  
plt.show()
```

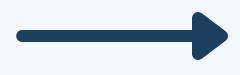
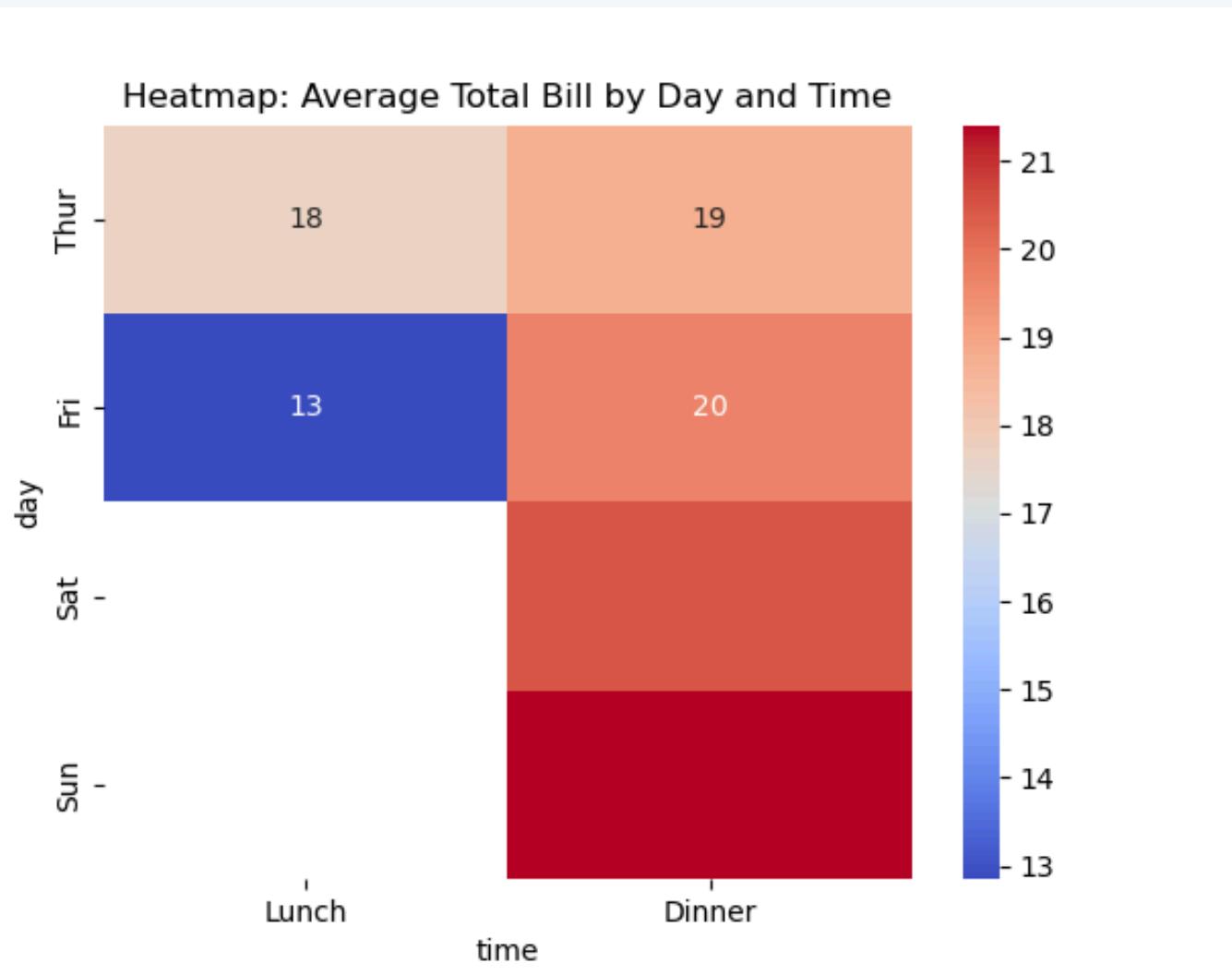


Seaborn

Matrix Plots:

- **Heatmap():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a Clustermap with numerical data  
clustermap = sns.clustermap(tips.select_dtypes(include='number'),  
                             cmap='coolwarm', annot=True)  
clustermap.ax_heatmap.set_title('Clustermap')  
plt.show()
```

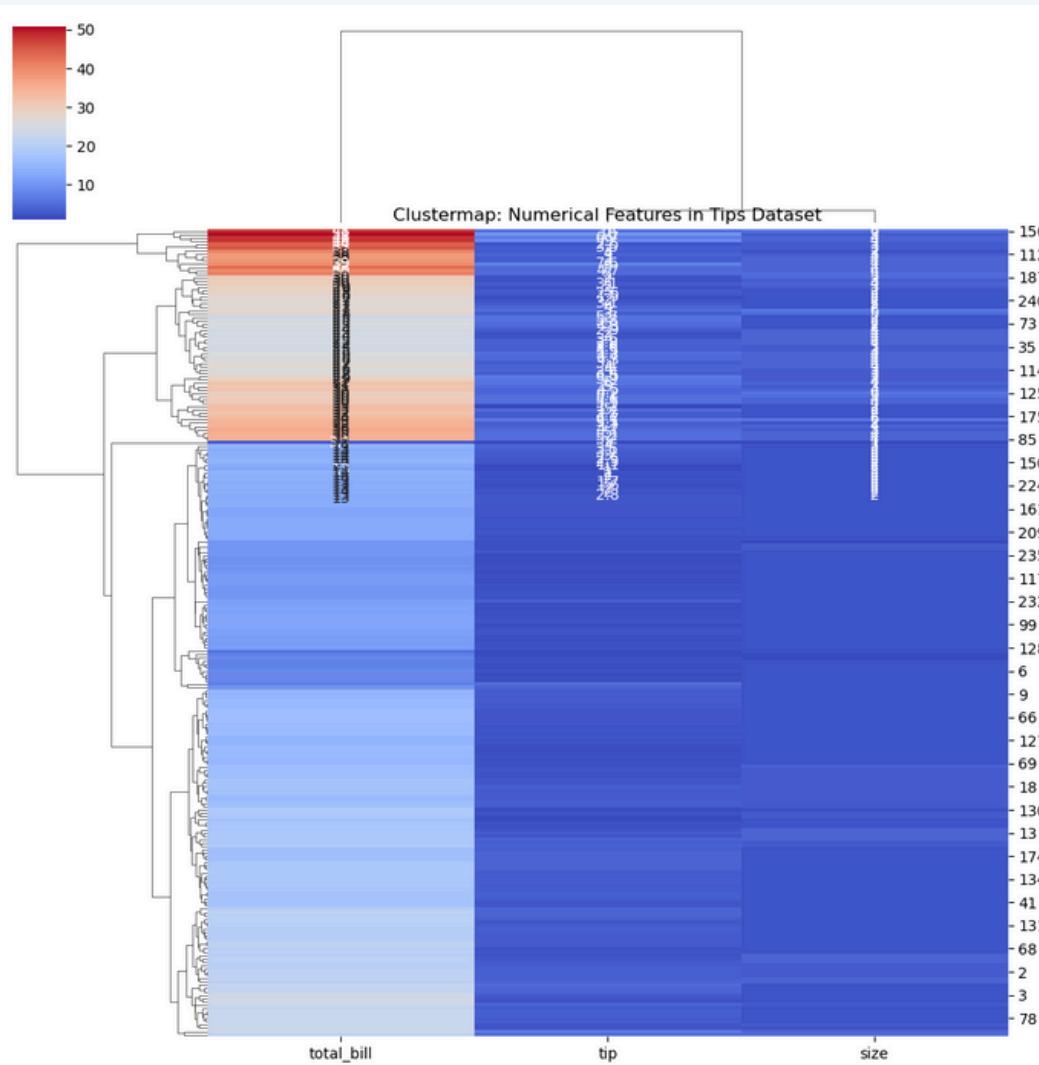


Seaborn

Matrix Plots:

- **Clustermap():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Pivot table to prepare data for heatmap  
pivot_table = tips.pivot_table(values='total_bill', index='day',  
                                columns='time', aggfunc='mean')  
  
# Creating a Heatmap  
sns.heatmap(pivot_table, annot=True, cmap='coolwarm')  
plt.title('Heatmap: Average Total Bill by Day and Time')  
plt.savefig('heatmap')
```

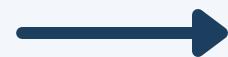
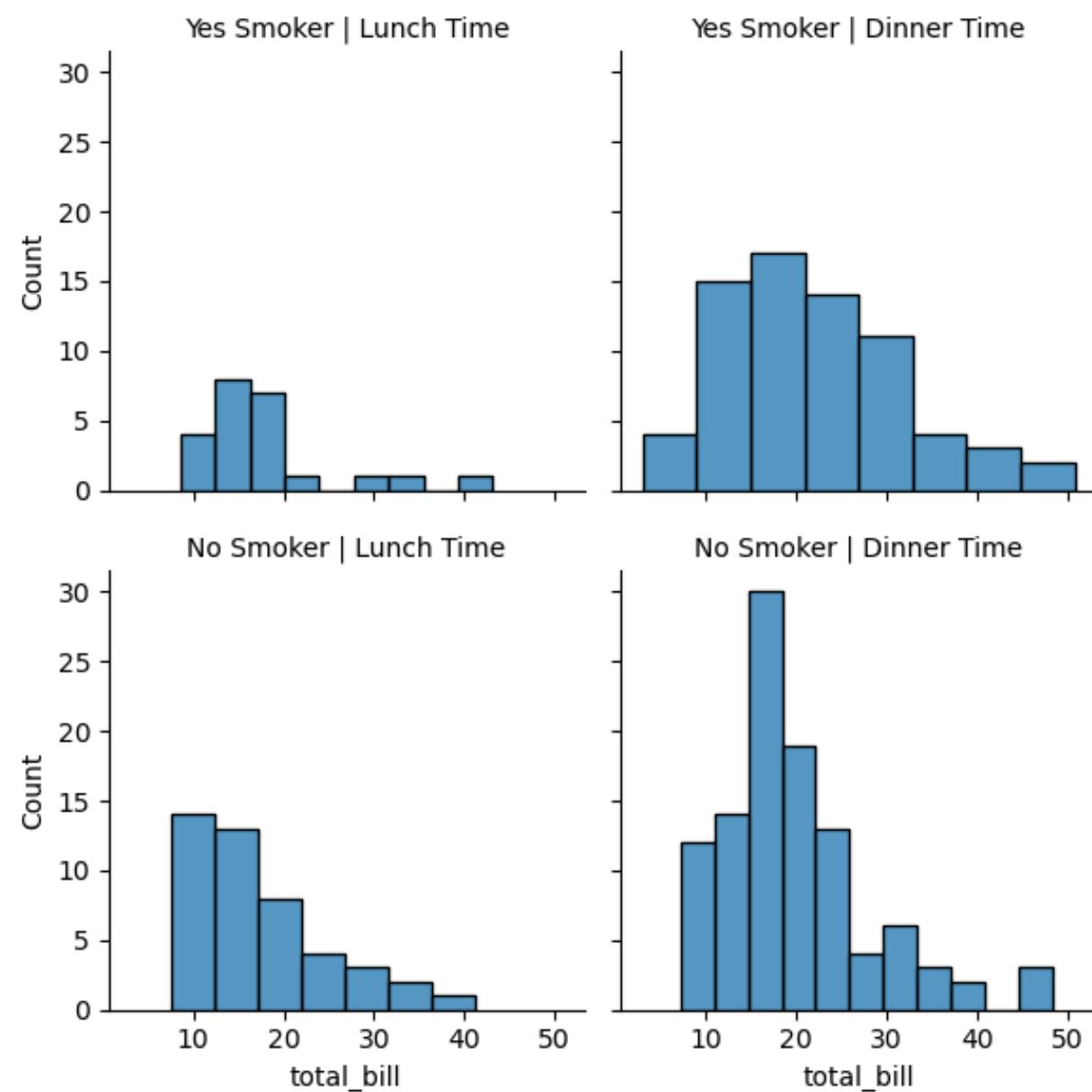


Seaborn

Multigrid Plots:

- **Facetgrid():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a FacetGrid  
g = sns.FacetGrid(tips, col='time', row='smoker')  
g.map(sns.histplot, 'total_bill')  
g.set_titles(col_template='{col_name} Time', row_template='{row_name} Smoker')  
plt.suptitle('FacetGrid: Distribution of Total Bill by Time and Smoker', y=1.02)  
plt.show()
```

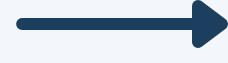
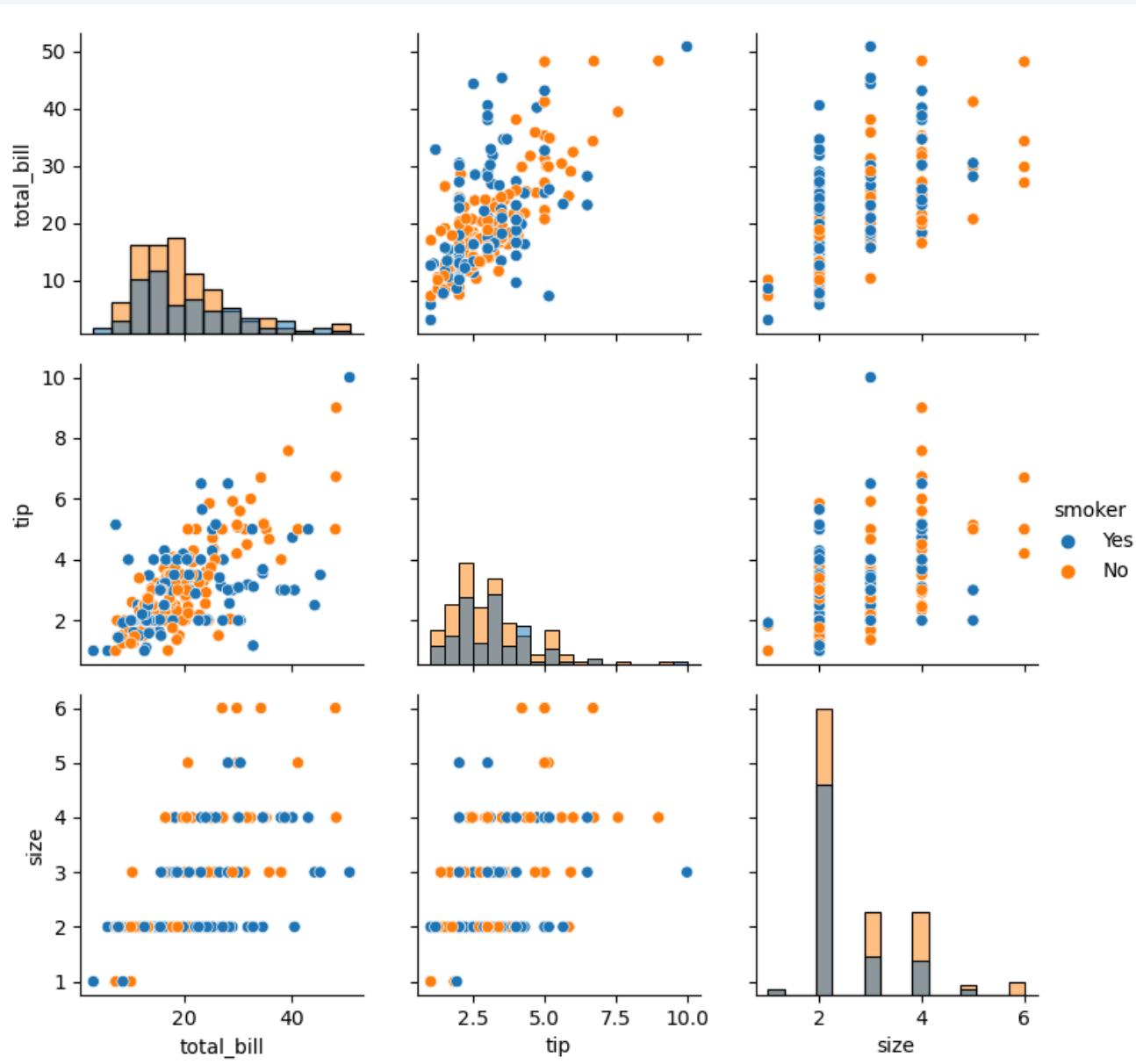


Seaborn

Multigrid Plots:

- **Pairgrid():**

```
● ● ●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a PairGrid  
g = sns.PairGrid(tips, hue='smoker')  
g.map_diag(sns.histplot)  
g.map_offdiag(sns.scatterplot)  
g.add_legend()  
plt.suptitle('PairGrid: Tips Data by Smoker Status', y=1.02)  
plt.show()
```

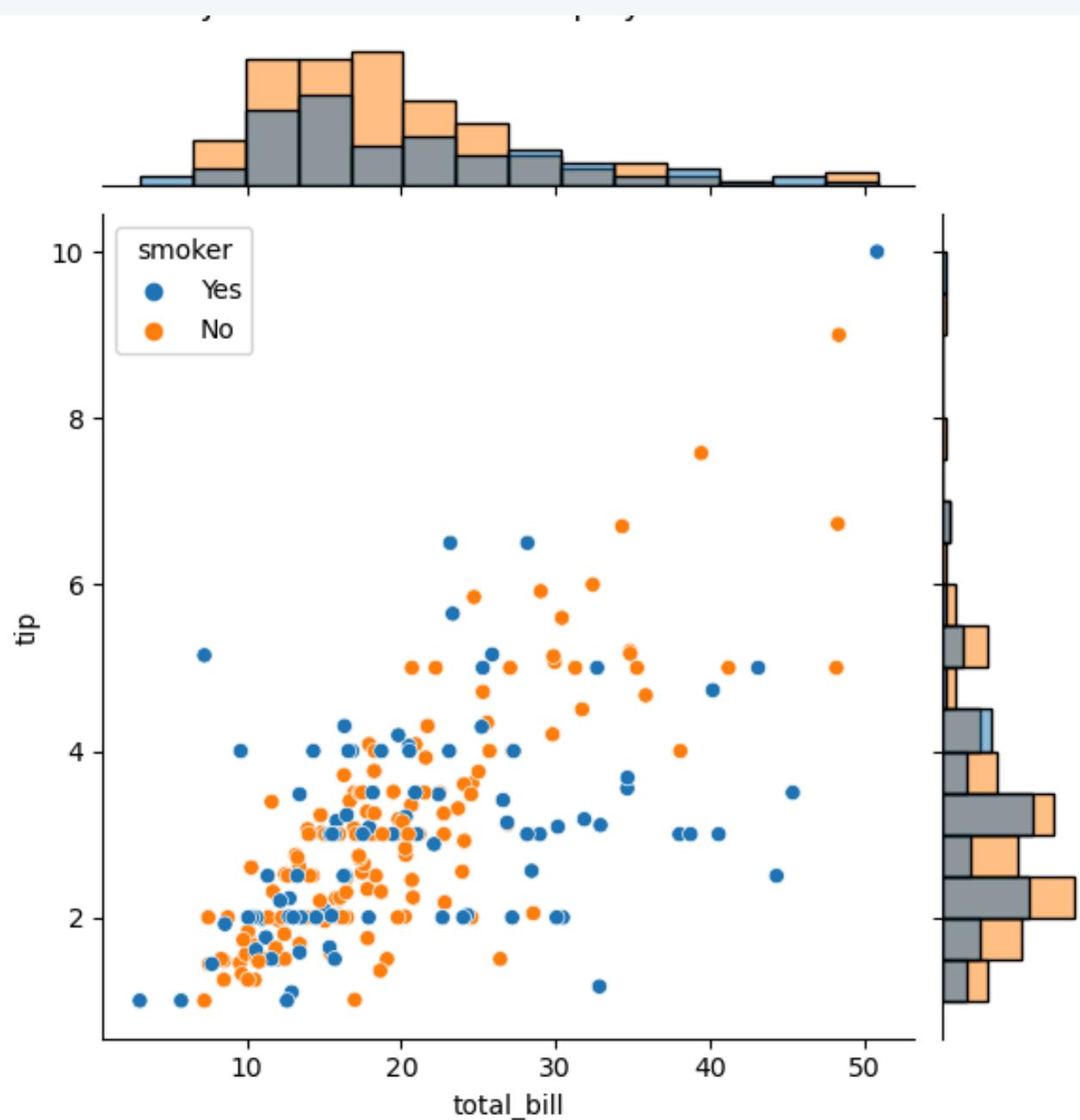


Seaborn

Multigrid Plots:

- **Jointgrid():**

```
●●●  
# Load example dataset  
tips = sns.load_dataset('tips')  
  
# Creating a JointGrid  
g = sns.JointGrid(data=tips, x='total_bill', y='tip', hue='smoker')  
g = g.plot(sns.scatterplot, sns.histplot)  
plt.suptitle('JointGrid: Total Bill vs Tip by Smoker Status', y=1.02)  
plt.show()
```





Ready to Level Up?

CONNECT FOR MORE INSIGHTS AND
FUTURE UPDATES!



Shruti Bharat
[@shrutibharat0105](#)