
📊 Seaborn Visualization Complete Guide

🔹 Why Seaborn?

A high-level API built on Matplotlib — easier syntax, beautiful default styles, and support for complex plots with minimal code.

🔹 Seaborn Roadmap

Understanding the two main types of functions:

- **Figure-Level Functions**: Create the entire figure, support multiple subplots (e.g., relplot, displot)
- **Axis-Level Functions**: Draw on a specific axis (e.g., scatterplot, lineplot)

🧩 Main Classification of Seaborn Plots

1. Relational Plots

- scatterplot() – Axis-level scatter plot for relationship between two numeric variables
- relplot() – Figure-level scatter/line plot for faceted plots
- Parameters: hue, size, style to represent more dimensions
- lineplot() – For trends over time or index
- Facet plots with col, row, and col_wrap for subplots

2. Distribution Plots

- histplot() – Histogram of variable distribution
- displot() – Figure-level distribution plot (combines hist, kde, rug)
- Parameters: bins, element, hue for customization
- kdeplot() – Smooth probability density estimate
- rugplot() – Small ticks on axis for individual observations
- Bivariate versions: joint distributions using kdeplot & histplot

3. Categorical Plots

- barplot() – Aggregate categorical data with error bars
- violinplot() – Combines boxplot & KDE for categorical data

4. Regression Plot

- regplot() – Fits a linear regression line to the scatter data

5. Matrix Plots

- heatmap() – Shows data values as color intensity in a matrix
  - Customization: annot, linewidths, cmap, etc.
- clustermap() – Hierarchical clustering + heatmap

6. MultiPlots

- jointplot() – Combines scatter + hist/kde plots for bivariate analysis

---

💡 Extras

- Plot sizing, customizing with parameters, subplot layout using FacetGrid, and doubts discussed in depth

---

✅ Libraries Used

- seaborn, matplotlib.pyplot, pandas, and CSV data for demonstration

---

🔹 Importing Libraries & Loading Data

Before any plotting:

```
import seaborn as sns

import matplotlib.pyplot as plt

import pandas as pd
```

Load dataset (example):

```
df = sns.load_dataset("tips")  # built-in dataset
```

🔶 1. Relational Plots

Relational plots show the relationship between two or more numerical (or categorical) variables.

📌 scatterplot() – Axis-level

Used to show relationships between two variables.

```
sns.scatterplot(x='total_bill', y='tip', data=df)
```

✔️ Simple and fast
✔️ Allows hue, style, size to encode more data

📌 relplot() – Figure-level

More flexible than scatterplot, supports multiple subplots.

```
sns.relplot(x='total_bill', y='tip', data=df)
```

- Returns a **FacetGrid** object
- Allows for easy faceting (col, row)

❓ scatterplot vs relplot

Feature	scatterplot()	relplot()
Type	Axis-level	Figure-level
Returns	AxesSubplot	FacetGrid
Subplots	❌	✅ with col, row
Use when	Simple single plot	Multiple subplots / advanced layout

📌 Parameters in Both

```
sns.scatterplot(x='total_bill', y='tip', data=df,
                hue='sex', size='size', style='smoker')
```

- hue: color based on category
- style: different markers
- size: different point sizes

📌 lineplot() – Axis-level

For trends over continuous or time-based variables

```
sns.lineplot(x='size', y='total_bill', data=df)
```

📌 relplot(kind='line') – Figure-level Line Plot

```
sns.relplot(x='size', y='total_bill', kind='line', data=df)
```

✅ Supports multiple facets just like figure-level scatter

📌 Facet Plots

```
sns.relplot(x='total_bill', y='tip', hue='sex',
            col='day', data=df)
```

- col, row: Creates grid of subplots
- col_wrap: Wrap columns for better layout

```
sns.relplot(x='total_bill', y='tip', col='day', col_wrap=2, data=df)
```

🔶 2. Distribution Plots

Used to visualize the distribution of a variable (univariate or bivariate)

📌 histplot() – Histogram

```
sns.histplot(df['total_bill'], bins=10)
```

- bins: number of intervals
- Can be used on numeric or categorical columns

📌 displot() – Figure-level Distribution Plot

```
sns.displot(df['total_bill'], bins=10)
```

- Supports faceting and plotting multiple distributions
- Returns FacetGrid

📌 Parameters in Histogram

```
sns.histplot(data=df, x='total_bill', hue='sex', element='step')
```

- hue: separate colors for categories

- element: bars, step, or poly (style of bars)

📌 Histogram on Categorical Column

```
sns.histplot(x='day', data=df)
```

- Works even for object/categorical columns
- Useful to count frequency

📌 Facet Plot with displot

```
sns.displot(df, x='total_bill', col='sex')
```

📌 kdeplot() – Kernel Density Estimation

```
sns.kdeplot(data=df['total_bill'], fill=True)
```

- Smooth curve over histogram
- Use fill=True for shaded area

📌 rugplot()

```
sns.rugplot(x=df['total_bill'])
```

- Small ticks on x-axis for each observation
- Best combined with kde or hist

📌 Bivariate Histogram

```
sns.histplot(data=df, x='total_bill', y='tip')
```

- 2D histogram for joint distribution
- bins=(x, y) for control

📌 Bivariate KDE Plot

```
sns.kdeplot(data=df, x='total_bill', y='tip', fill=True)
```

- Smooth 2D distribution

- fill=True shows contours as shades

🔶 3. Matrix Plots

Best used for visualizing grid-like or tabular data such as correlation matrices.

📌 heatmap()

```
sns.heatmap(df.corr())
```

- Shows correlation matrix as color-intensity
- By default, darker = higher correlation

🔧 Customizations:

```
sns.heatmap(df.corr(), annot=True, linewidths=0.5, cmap='coolwarm')
```

- annot=True: Show numeric values
- linewidths: lines between cells
- cmap: color palette

📌 clustermap()

```
sns.clustermap(df.corr())
```

- Performs hierarchical clustering
- Automatically reorders rows/columns
- Good for grouping similar variables

🔹 Additional Notes

📐 Changing Size in Figure-Level Plots

Use height and aspect:

```
sns.relplot(x='total_bill', y='tip', data=df, height=5, aspect=1.5)
```

- height: height of each facet
- aspect: width = height × aspect
