

---

## 1. Categorical Plots

Categorical plots are used when **one variable is categorical** (like gender, day, class) and the other is numerical.

---

### ◆ Stripplot

- **Use:** Shows individual data points by category.
- **Best for:** Viewing spread and overlaps.

```
sns.stripplot(x="day", y="total_bill", data=tips, jitter=True)
```

#### Plot Output:

Dots placed vertically for each day, possibly overlapping.

---

### ◆ Swarmplot

- **Use:** Similar to stripplot, but adjusts points to avoid overlap.
- **Best for:** Clean view of all points.

```
sns.swarmplot(x="day", y="total_bill", data=tips)
```

#### Plot Output:

Bee-swarm-like alignment of dots with no overlap.

---

### ◆ catplot()

- **Use:** A high-level wrapper for categorical plots like box, violin, strip, etc.
- **Faceting supported.**

```
sns.catplot(x="day", y="total_bill", kind="swarm", data=tips)
```

#### Plot Output:

Swarmplot within a figure-level plot object, with subplot options.

---

## 2. Categorical Distribution Plots

Used to compare **distribution** of a numeric variable across categories.

---

### ◆ Boxplot

- **Use:** Visualize quartiles, median, and outliers.

```
sns.boxplot(x="day", y="total_bill", data=tips)
```

**Plot Output:**

Box with whiskers, center line (median), and dots (outliers).

---

◆ **Violinplot**

- **Use:** Boxplot + KDE (shows shape of distribution).

```
sns.violinplot(x="day", y="total_bill", data=tips)
```

**Plot Output:**

Symmetric violin-shaped curves around a boxplot.

---



### 3. Categorical Estimate Plots (Central Tendency)

Focus on showing **mean** and **confidence intervals**.

---

◆ **Barplot**

- **Use:** Mean of y-value per category, with error bars.

```
sns.barplot(x="day", y="total_bill", data=tips)
```

**Plot Output:**

Bars showing average total bill for each day.

---

◆ **Pointplot**

- **Use:** Shows trends across categories with points + lines.

```
sns.pointplot(x="day", y="total_bill", data=tips)
```

**Plot Output:**

Dots connected by lines with error bars.

---

◆ **Countplot**

- **Use:** Shows count (frequency) of items in each category.

```
sns.countplot(x="day", data=tips)
```

### Plot Output:

Bars representing number of observations per day.

---

#### ◆ Faceting

- **Use:** Splits plots by category using col or row.

```
sns.catplot(x="sex", y="total_bill", kind="box", col="day", data=tips)
```

### Plot Output:

4 boxplots in 4 columns, one per day.

---

## 4. Regression Plots

Used to understand relationships between two **numeric variables**.

---

#### ◆ regplot()

- **Use:** Basic scatter + linear regression line.

```
sns.regplot(x="total_bill", y="tip", data=tips)
```

### Plot Output:

Scatter with a straight regression line.

---

#### ◆ lmpplot()

- **Use:** Like regplot, but figure-level with hue, col, row.

```
sns.lmpplot(x="total_bill", y="tip", hue="sex", data=tips)
```

### Plot Output:

Two scatter + regression lines, one per gender.

---

#### ◆ Residual Plot

- **Use:** Shows residuals (errors) from regression.

```
sns.residplot(x="total_bill", y="tip", data=tips)
```

### Plot Output:

Scatter of residuals; good for diagnosing linear fit.

---

## ◆ 5. Grid Plots

For **multi-dimensional relationships** and customizable plotting.

---

### ◆ FacetGrid

- **Use:** Custom subplot grid using row/col/hue.

```
g = sns.FacetGrid(tips, col="sex")  
g.map(sns.scatterplot, "total_bill", "tip")
```

#### **Plot Output:**

One scatter plot per gender.

---

### ◆ Pairplot

- **Use:** Automatically plots all pairwise combinations.

```
sns.pairplot(tips, hue="sex")
```

#### **Plot Output:**

Grid of scatter and hist plots for all variable pairs.

---

### ◆ PairGrid

- **Use:** Customizable version of pairplot.

```
g = sns.PairGrid(tips)  
g.map_diag(sns.histplot)  
g.map_offdiag(sns.scatterplot)
```

#### **Plot Output:**

Same as pairplot but with manual control.

---

### ◆ Jointplot

- **Use:** Combines scatter + hist + KDE for 2 variables.

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

#### **Plot Output:**

Scatter + regression line, with histograms on sides.

---

### ◆ JointGrid

- **Use:** Like jointplot, but more flexible.

```
g = sns.JointGrid(data=tips, x="total_bill", y="tip")
```

```
g.plot(sns.scatterplot, sns.histplot)
```

#### Plot Output:

Same layout, but fully customizable.

---

### 6. Utility Function: load\_dataset()

Loads Seaborn's **built-in datasets** like tips, penguins, etc.

```
df = sns.load_dataset("tips")
```

Use it for learning and demo purposes.

---