

① What is Seaborn?

Seaborn is a Python data visualization library used to create statistical graphs easily.

It is built on top of Matplotlib, so internally it uses Matplotlib but provides simpler syntax and better design.

Use

- To analyze data visually
 - To find relationships between variables
 - To create professional-looking charts
 - Widely used in Data Science & Machine Learning
-

② Importing Seaborn (Important)

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Explanation

- seaborn → main visualization library
 - sns → short name (convention)
 - matplotlib.pyplot → used to display graphs (plt.show())
-

③ Loading Dataset

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

Explanation of variables & method

- load_dataset() → Seaborn method to load built-in datasets
- "tips" → dataset name
- df → Pandas DataFrame where data is stored

☞ Why DataFrame?

Because Seaborn works best with Pandas DataFrames

4 Line Plot

Use

- To show trend between two numerical variables

Code

```
sns.lineplot(x="size", y="total_bill", data=df)
plt.show()
```

Explanation (Line by Line)

- `sns.lineplot()` → creates a line graph
- `x="size"` → column name for X-axis
- `y="total_bill"` → column name for Y-axis
- `data=df` → tells Seaborn where the data is
- `plt.show()` → displays the graph window

🔗 Seaborn automatically calculates average values

5 Bar Plot

Use

- To compare average values of categories

Code

```
sns.barplot(x="day", y="total_bill", data=df)
plt.show()
```

Explanation

- `barplot()` → creates bar chart
 - `x="day"` → categorical variable (Mon, Tue, etc.)
 - `y="total_bill"` → numerical variable
 - Bars represent mean (average) value
 - Error line shows confidence interval
-

6 Scatter Plot

Use

- To check relationship between two variables

Code

```
sns.scatterplot(x="total_bill", y="tip", data=df)
plt.show()
```

Explanation

- scatterplot() → dot-based graph
 - x="total_bill" → independent variable
 - y="tip" → dependent variable
 - Each dot → one row of dataset
-

7 Histogram

Use

- To see distribution of data

Code

```
sns.histplot(df["total_bill"])
plt.show()
```

Explanation

- histplot() → histogram graph
 - df["total_bill"] → selects one column
 - Bars show frequency of values
-

8 Count Plot

Use

- To count categorical values

Code

```
sns.countplot(x="day", data=df)
plt.show()
```

Explanation

- `countplot()` → counts entries
- `x="day"` → category column
- Height of bar = number of records

☞ No `y` needed because counting is automatic

9 Box Plot

Use

- To detect outliers
- To understand data spread

Code

```
sns.boxplot(x="day", y="total_bill", data=df)
plt.show()
```

Explanation

- `boxplot()` → statistical graph
 - Middle line → median
 - Box → Q1 to Q3 range
 - Dots → outliers
-

10 Violin Plot

Use

- To show distribution + density

Code

```
sns.violinplot(x="day", y="total_bill", data=df)
plt.show()
```

Explanation

- `violinplot()` → advanced distribution plot
 - Width shows data density
 - Combines box plot + histogram concept
-

11 Heatmap

Use

- To show correlation between variables

Code

```
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True)
plt.show()
```

Explanation

- `df.corr()` → calculates correlation
 - `numeric_only=True` → only numeric columns
 - `corr` → correlation matrix
 - `heatmap()` → color-based graph
 - `annot=True` → shows numeric values on cells
-

12 Pair Plot

Use

- To analyze entire dataset at once
- Used in EDA (Exploratory Data Analysis)

Code

```
sns.pairplot(df)
plt.show()
```

Explanation

- `pairplot()` → creates multiple plots
 - Shows relationship between all numerical columns
 - Diagonal → distribution plots
-

13 Styling Graphs

Code

```
sns.set_style("darkgrid")
sns.barplot(x="day", y="total_bill", data=df)
plt.show()
```

Explanation

- `set_style()` → changes background style
- "darkgrid" → grid-based dark style

Other styles:

- whitegrid
 - dark
 - white
-

✦ Summary Table (Methods & Uses)

Method	Use
<code>load_dataset()</code>	Load sample dataset
<code>lineplot()</code>	Trend analysis
<code>barplot()</code>	Category comparison
<code>scatterplot()</code>	Relationship check
<code>histplot()</code>	Distribution
<code>countplot()</code>	Count categories
<code>boxplot()</code>	Outliers & spread
<code>violinplot()</code>	Density + distribution
<code>heatmap()</code>	Correlation
<code>pairplot()</code>	Full data analysis