📌 What is Seaborn?

Seaborn is a **data visualization library** built on top of Matplotlib. It makes graphs:

✅ More **beautiful**
✅ More **informative**
✅ **Easier to create**

🔹 Install Seaborn

If you haven't installed it yet, run:

```
pip install seaborn
```

🔹 Import Seaborn

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

🔥 1. Basic Seaborn Plot (Line Plot)

Seaborn works with Pandas **DataFrames**. Let's use a built-in dataset!

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


# Load built-in dataset
data = sns.load_dataset("fmri")


# Line plot
sns.lineplot(x="timepoint", y="signal", hue="region", data=data)


plt.show()
```

✅ **hue="region"** → Different colors for each region
✅ **Automatic styling** (no need for plt.plot)

🔥 2. Scatter Plot (Relationship Between Two Variables)

```
# Load dataset
tips = sns.load_dataset("tips")


# Scatter plot
sns.scatterplot(x="total_bill", y="tip", hue="sex", style="time", size="size",
data=tips)


plt.show()
```

✅ **hue="sex"** → Color by gender
✅ **style="time"** → Different markers for lunch/dinner
✅ **size="size"** → Point size based on party size

🔥 **3. Histogram & KDE Plot (Data Distribution)**

✅ **Histogram**

```
sns.histplot(tips["total_bill"], bins=20, kde=True)
plt.show()
```

✅ Shows **frequency** of total bill amounts
✅ kde=True adds a smooth **density curve**

🔥 **4. Boxplot (Outliers & Data Distribution)**

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

✅ Boxplots show **median, quartiles, and outliers**

🔥 **5. Heatmap (Correlation Between Variables)**

```
import numpy as np

# Compute correlation matrix
corr = tips.corr()

# Draw heatmap
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.show()
```

✅ Shows **correlation** between variables
✅ annot=True → Displays correlation values

🔥 **6. Pairplot (Multi-Variable Relationships)**

A **pairplot** shows scatter plots between all numerical columns in a dataset.

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

# Load dataset
iris = sns.load_dataset("iris")

# Create pairplot
```

```
sns.pairplot(iris, hue="species", diag_kind="kde")


plt.show()
```

✅ **hue="species"** → Different colors for each species
✅ **diag_kind="kde"** → Adds a smooth density curve instead of histograms

---

🔥 **7. Jointplot (Two Variable Relationship)**

A **jointplot** combines a scatter plot with histograms.

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

✅ **kind="reg"** → Adds a regression line

Other options for kind:

- "scatter" (default) → Just scatter
- "hex" → Hexagonal binning (for large datasets)
- "kde" → Kernel density estimation

🔥 **8. Categorical Plots**

✅ **Barplot (Mean of a Category)**

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

✅ Shows the **sum** of total_bill per day

✅ **Countplot (Category Frequency)**

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

✅ Shows **how many times each day appears**

✅ **Violin Plot (Distribution of Data)**

```
sns.violinplot(x="day", y="total_bill", data=tips, hue="sex", split=True)
plt.show()
```

✅ Combines **boxplot & KDE plot** for a better data spread visualization

---

🔥 **9. Regression Plot (Best-Fit Line)**

A regression plot helps visualize relationships between two variables.

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

✅ Automatically **fits a regression line**

🔥 10. FacetGrid (Multiple Plots in One Figure)

A **FacetGrid** allows you to create multiple graphs for different categories.

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

plt.show()
```

✅ Creates separate **scatter plots for each category**

🔥 11. Customizing Seaborn Styles

Seaborn provides built-in themes to make your plots look professional.

✅ Change Theme

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

# Set style
sns.set_style("darkgrid")  # Options: "white", "dark", "whitegrid", "ticks", "darkgrid"

# Load dataset
tips = sns.load_dataset("tips")

# Create plot
sns.scatterplot(x="total_bill", y="tip", data=tips)

plt.show()
```

✅ "darkgrid" adds a grid for better readability

✅ Change Color Palette

```
sns.set_palette("coolwarm")  # Try "pastel", "deep", "muted", "bright", etc.
sns.boxplot(x="day", y="total_bill", data=tips)
plt.show()
```

✅ "coolwarm" gives a blue-red gradient

🔥 12. Saving Plots

You can save Seaborn plots as images for reports.

```
plt.figure(figsize=(8,6))  # Set figure size
sns.histplot(tips["total_bill"], bins=20, kde=True)

plt.savefig("my_plot.png", dpi=300)  # Save as PNG with high resolution
plt.show()
```

✅ dpi=300 ensures **high-quality images**

🔥 13. Combining Seaborn with Matplotlib

You can use Seaborn for styling but still modify plots using **Matplotlib**.

```
sns.set_style("whitegrid")
plt.figure(figsize=(8,6))

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

# Add Matplotlib customizations
plt.title("Total Bill by Day", fontsize=14, fontweight="bold")
plt.xlabel("Day of the Week", fontsize=12)
plt.ylabel("Total Bill ($)", fontsize=12)

plt.show()
```

✅ Seaborn for styling, Matplotlib for fine-tuning

🔥 14. Real-Time Data Visualization

For **live data updates**, use plt.pause().

```
import numpy as np
import time

for i in range(10):
    data = np.random.rand(10)
    sns.lineplot(x=range(10), y=data)
    plt.pause(0.5)  # Pause for half a second
    plt.clf()  # Clear the plot for the next update
```

✅ This creates a **real-time updating graph**