# Data Visualization – Seaborn

## Seaborn

**Seaborn** is a high-level Python data visualization library **built on top of Matplotlib**. It provides a simpler and more attractive way to create statistical graphics, with default themes and color palettes that look modern and clean.

### Key Features

1. **Integration with Pandas** – Works seamlessly with DataFrames and handles column names directly.
2. **Complex plots made easy** – Automatically handles things like regression lines, aggregation, and confidence intervals.
3. **Statistical plotting** – Supports visualization of distributions, relationships, and categorical data (e.g., `boxplot`, `violinplot`, `heatmap`).
4. Beautiful default styles & in-built datasets for learning.

### Usage

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

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

# Scatter plot of total bill vs tip
sns.scatterplot(data=tips, x="total_bill", y="tip", hue="day", style="sex", size="size")
plt.title("Tips Scatter Plot")
plt.show()
```

- `hue` - color by category
- `style` - marker style by category
- `size` - size of points by a variable

### Why Use Seaborn

- Saves time with prettier defaults than Matplotlib.
- Great for exploratory data analysis (EDA).
- Works well with complex datasets and statistical visualizations.

# Common Plots with Seaborn

## Relational Plots

Relational plots are plots that show the relationship between two (or more) variables, typically numerical x and y variables.

### Line Plots

```
# Example dataset
data = sns.load_dataset("tips")

# Line plot: average tip by total bill
sns.lineplot(data=data, x="total_bill", y="tip")
plt.title ("Line Plot: Total Bill vs Tip")
plt.show()
```

We can also use `ci` i.e. Confidence Interval.

### Scatter Plots

```
# Scatter plot
sns.scatterplot(data=data, x="total_bill", y="tip")
plt.title("Scatter Plot: Total Bill vs Tip")
plt.show()
```

## Categorical Plots

Categorical plots are used to visualize data where one variable is categorical (discrete) and the other is usually numerical.

### Bar Plots

```
tips = sns.load_dataset("tips")
sns.barplot(data=tips, x="day", y="total_bill", hue="sex")
plt.title("Average Total Bill per Day by Sex")
plt.show()
```

### Box Plots

```
sns.boxplot(data=tips, x="day", y="total_bill", hue="sex")
plt.title("Box Plot of Total Bill by Day and Sex")
plt.show()
```

## Distribution Plots

Distribution plots are used to visualize the distribution of a single numerical variable.

### Histograms

```
tips = sns.load_dataset("tips")
sns.histplot(data=tips, x="total_bill", bins=10, kde=False)
plt.title("Histogram of Total Bill")
plt.show()
```

## Matrix Plots

Matrix plots are used to visualize 2D data arranged in a matrix or table, such as correlation matrices, confusion matrices, or pivot tables.

### Heatmap

```
# Example: correlation matrix of tips dataset
tips = sns.load_dataset("tips")
corr = tips.corr() # correlation between numerical columns

sns.heatmap (corr, annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

## Best Practices

1. When working with data visualization we use Seaborn for **High-Level Plots** and Matplotlib for **Low-Level Customizations**.
2. We work with `fig` and `ax` objects for modular and reusable code.

### Creating Simple Plot

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

fig, ax = plt.subplots(figsize=(7,5))

sns.scatterplot(data=sns.load_dataset("tips"), x="total_ bill", y="tip", hue="day", ax=ax)
sns.set_style("whitegrid") # options: white, dark, whitegrid, ticks

ax.set_title("Scatter Plot with Seaborn and Matplotlib")
plt.show()
```

### Using Subplots

```
fig, ax = plt.subplots()

sns.boxplot(data=sns.load_dataset("tips"), x="day", y="total_bill", hue="sex", ax=ax)

ax.set_title("Boxplot: Total Bill by Day and Sex")
ax.set_xlabel("Day of the Week")
ax.set_ylabel("Total Bill ($)")
plt.legend(title="Sex")

plt.show()
```

*Keep Learning & Keep Exploring!*