

# Matplotlib & Seaborn: A Simple Guide to Data Visualization

## 1. Introduction

Data visualization is an essential skill for anyone working with data. Whether you're analyzing trends, comparing categories, or exploring relationships, graphs make your findings much clearer.

This guide introduces two popular Python libraries for visualization: **Matplotlib** and **Seaborn**. We'll cover their features, common graph types, and simple examples to get you started.

### 1.1 Matplotlib: The Classic Visualization Tool

Matplotlib is one of the oldest and most widely used Python libraries for data visualization. It provides complete control over plots, making it useful for both simple and complex charts.

#### Why use Matplotlib?

- Supports many types of plots (line charts, bar charts, scatter plots, etc.).
- Highly customizable.
- Can be used for both basic and advanced visualizations.

### 1.2 Seaborn: Simplified and Stylish Charts

Seaborn is built on top of Matplotlib and makes creating attractive statistical visualizations much easier. It's especially useful for handling datasets with built-in themes and color palettes.

#### Why use Seaborn?

- Simplifies complex plots.
- Automatically styles charts for better readability.
- Ideal for statistical and categorical data.

---

## 2. Common Graph Types & Examples

Here are some of the most commonly used plots in Matplotlib and Seaborn, along with basic code examples.

### 2.1 Line Plot (Used to show trends over time)

**Matplotlib Example:**

```
import matplotlib.pyplot as plt
```

```
x = [1, 2, 3, 4, 5]
```

```
y = [10, 15, 20, 25, 30]
```

```
plt.plot(x, y)
```

```
plt.show()
```

**Seaborn Example:**

```
import seaborn as sns
```

```
sns.lineplot(x=[1, 2, 3, 4, 5], y=[10, 15, 20, 25, 30])
```

---

## 2.2 Scatter Plot (Used to show relationships between two variables)

**Matplotlib Example:**

```
plt.scatter([1, 2, 3, 4], [5, 15, 10, 20])
```

```
plt.show()
```

**Seaborn Example:**

```
sns.scatterplot(x=[1, 2, 3, 4], y=[5, 15, 10, 20])
```

---

## 2.3 Bar Chart (Used to compare different categories)

**Matplotlib Example:**

```
plt.bar(['A', 'B', 'C'], [10, 20, 15])
```

```
plt.show()
```

**Seaborn Example:**

```
sns.barplot(x=['A', 'B', 'C'], y=[10, 20, 15])
```

---

## 2.4 Histogram (Used to show the distribution of data)

**Matplotlib Example:**

```
import numpy as np
```

```
data = np.random.randn(100)
```

```
plt.hist(data)
```

```
plt.show()
```

**Seaborn Example:**

```
sns.histplot(data)
```

---

## 2.5 Heatmap (Only available in Seaborn, useful for visualizing matrix-like data)

### Seaborn Example:

```
import numpy as np
import seaborn as sns
```

```
matrix = np.random.rand(3, 3)
sns.heatmap(matrix)
```

---

## 3. Matplotlib vs. Seaborn: A Quick Comparison

Feature	Matplotlib	Seaborn
Ease of Use	Requires more code	Simpler to use
Customization	Fully customizable	Limited customization
Interactivity	Supports interactivity	Mostly static plots
Best For	General plotting needs	Statistical and categorical data

---

## 4. Further Learning Resources

If you'd like to explore more, check out these official documentation pages:

- [Matplotlib Quick Start Guide](#)
  - [Seaborn Introduction](#)
- 

## Conclusion

Both Matplotlib and Seaborn are excellent tools for data visualization in Python. If you want full control over your plots, go with Matplotlib. If you prefer quick and beautiful statistical charts, Seaborn is a great choice.