📊 Visualization Library Guide – Matplotlib vs Seaborn

Welcome! This notebook is a beginner-friendly guide comparing two powerful Python visualization libraries: **Matplotlib** and **Seaborn**.

We'll cover:

- What each library does
- Common graph types and how to create them
- A side-by-side comparison

Let's dive in!

🔧 Library Overview

Matplotlib

Matplotlib is a flexible plotting library that lets you build all sorts of visualizations from scratch. It’s great for detailed control and customization.

Seaborn

Seaborn is built on top of Matplotlib. It makes it easier to create beautiful statistical plots with less code.

In [1]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import pandas as pd
import numpy as np

# For inline display
%matplotlib inline
```

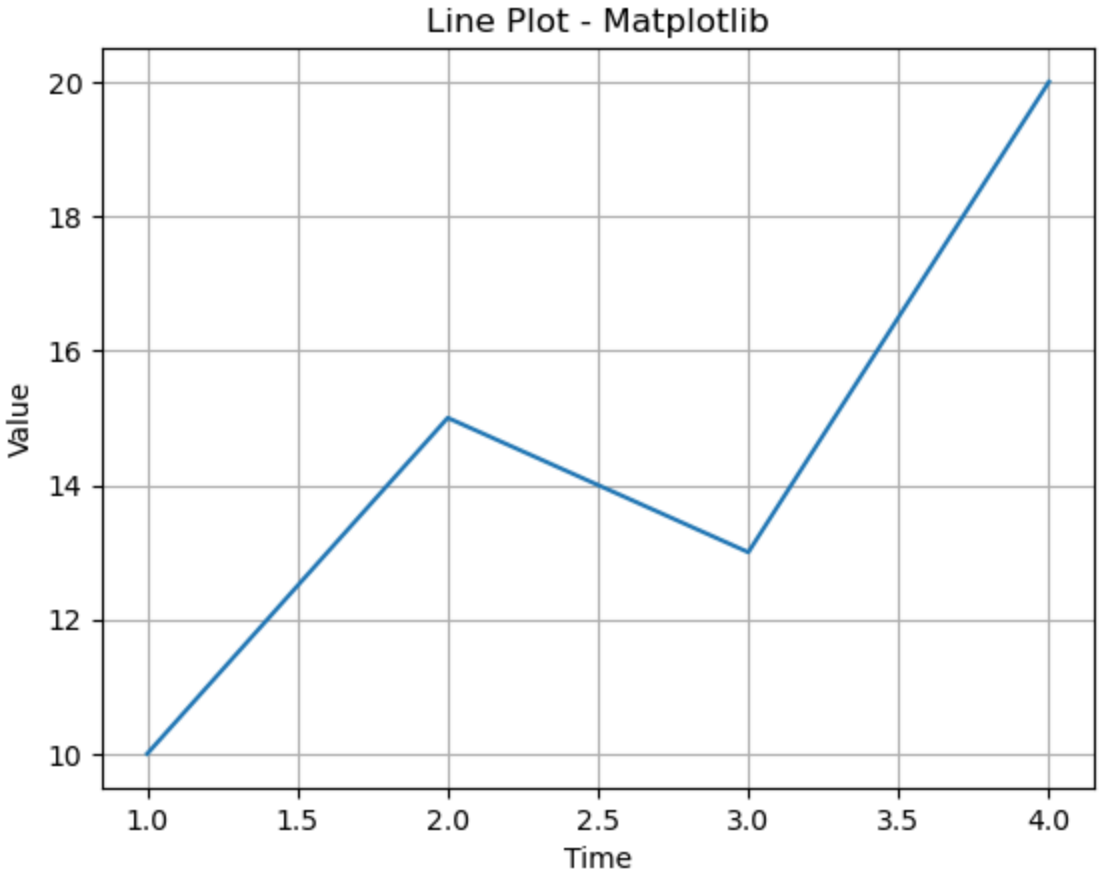
📈 Line Plot

A line plot is useful for tracking changes over time.

In [4]:
```
x = [1, 2, 3, 4]
y = [10, 15, 13, 20]

# Matplotlib
plt.plot(x, y)
plt.title("Line Plot - Matplotlib")
```

```
plt.xlabel("Time")
plt.ylabel("Value")
plt.grid(True)
plt.show()
```



In [18]:
```
# Seaborn
df_line = pd.DataFrame({"Time": x, "Value": y})
sns.lineplot(data=df_line, x="Time", y="Value").set(title="Line Plot - Seabo
```

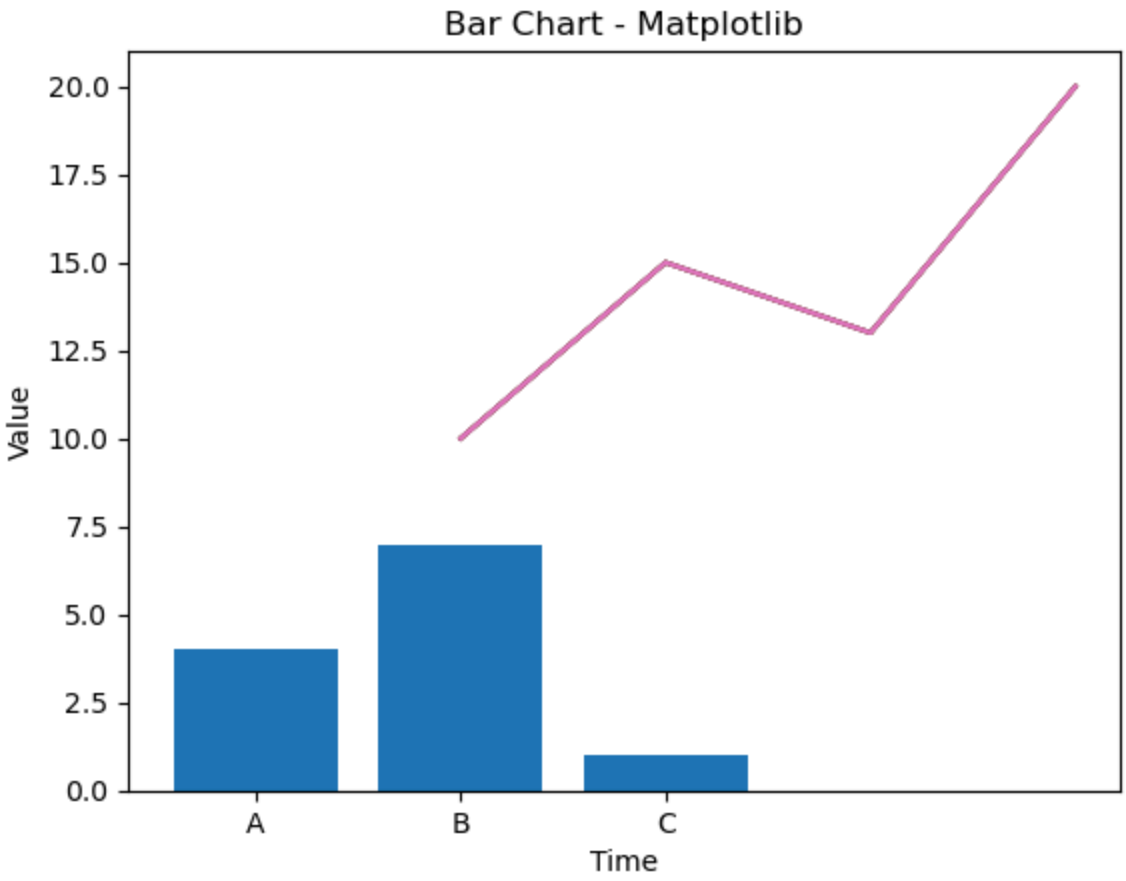
Out[18]:
```
[Text(0.5, 1.0, 'Line Plot - Seaborn')]
```

📊 Bar Chart

Bar charts are great for comparing categories.

In [20]:
```
categories = ["A", "B", "C"]
values = [4, 7, 1]

# Matplotlib
plt.bar(categories, values)
plt.title("Bar Chart - Matplotlib")
plt.show()
```



In [22]:
```
# Seaborn
df_bar = pd.DataFrame({"Category": categories, "Score": values})
sns.barplot(x="Category", y="Score", data=df_bar).set(title="Bar Chart - Sea
```

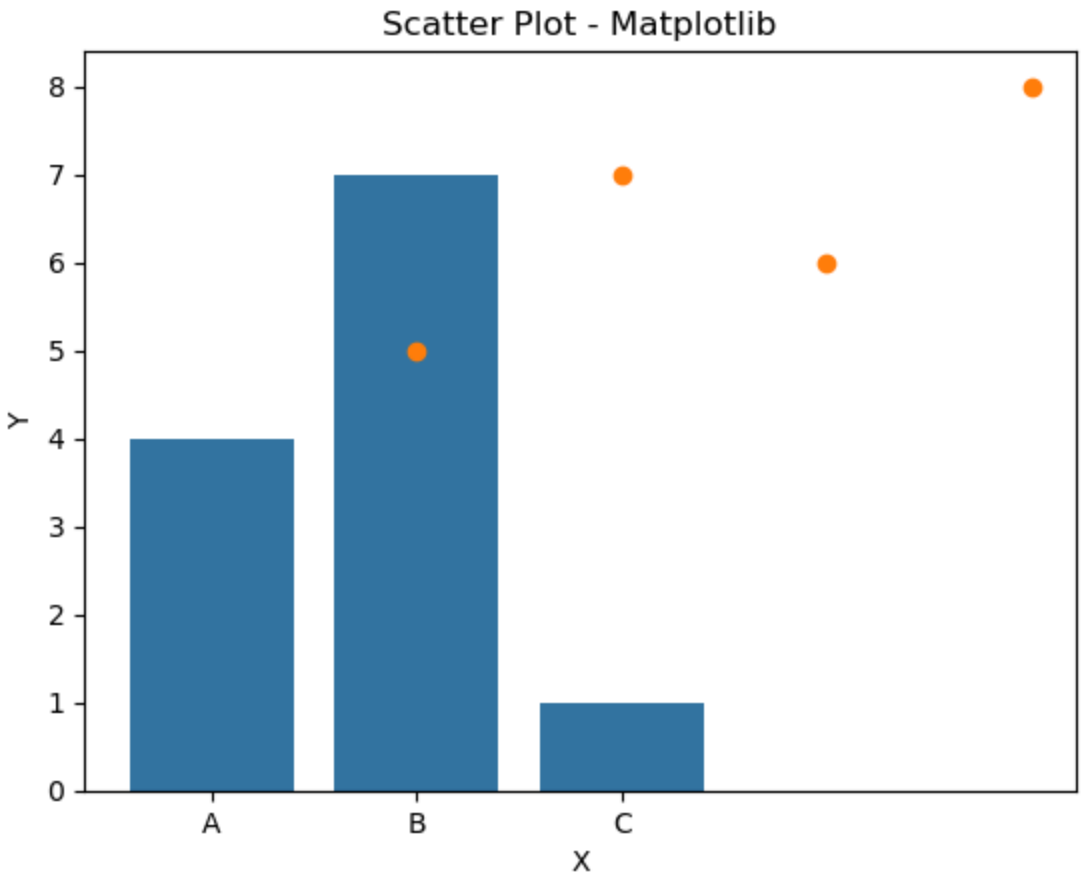
Out[22]:
```
[Text(0.5, 1.0, 'Bar Chart - Seaborn')]
```

📍 Scatter Plot

Used to display the relationship between two numeric variables.

In [24]:
```
x = [1, 2, 3, 4]
y = [5, 7, 6, 8]

# Matplotlib
plt.scatter(x, y)
plt.title("Scatter Plot - Matplotlib")
plt.xlabel("X")
plt.ylabel("Y")
plt.show()
```



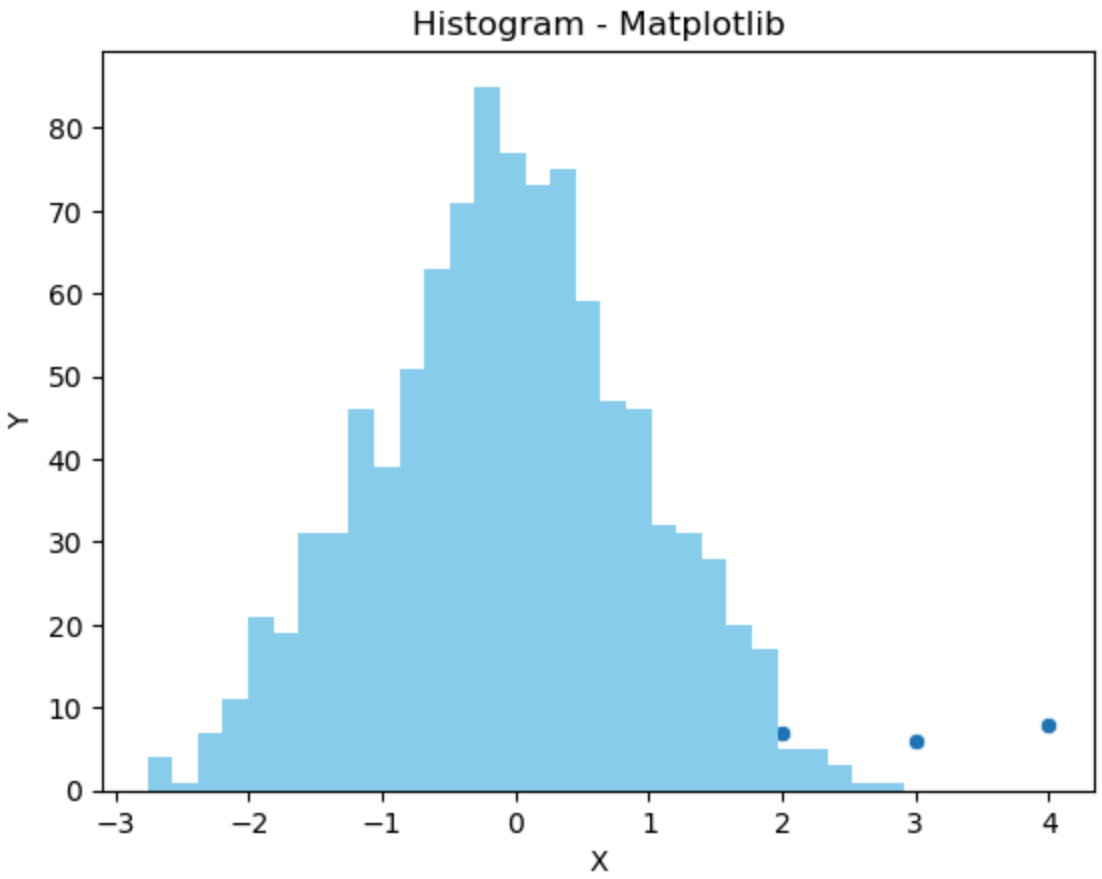
In [26]:
```
# Seaborn
df_scatter = pd.DataFrame({"X": x, "Y": y})
sns.scatterplot(data=df_scatter, x="X", y="Y").set(title="Scatter Plot - Sea
```

Out[26]:
```
[Text(0.5, 1.0, 'Scatter Plot - Seaborn')]
```

📊 Histogram

Used to understand the distribution of a variable.

In [28]:
```
data = np.random.randn(1000)
plt.hist(data, bins=30, color='skyblue')
plt.title("Histogram - Matplotlib")
plt.show()
```



In [34]:
```
sns.histplot(data, bins=30, kde=True, color='skyblue').set(title="Histogram
```

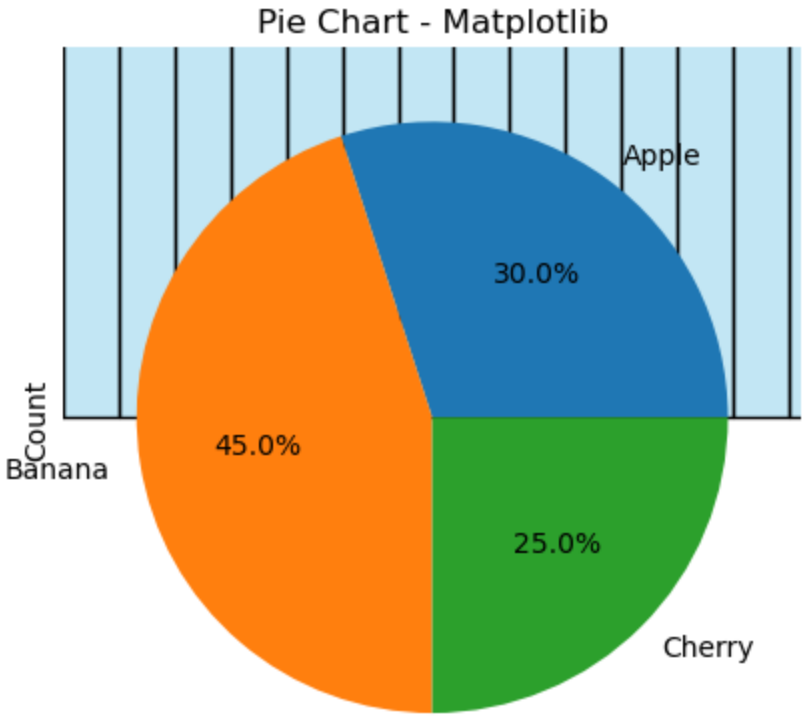
Out[34]:
```
[Text(0.5, 1.0, 'Histogram - Seaborn')]
```

🥧 Pie Chart (Matplotlib only)

Shows proportional data.

In [32]:
```
labels = ['Apple', 'Banana', 'Cherry']
sizes = [30, 45, 25]

plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.title("Pie Chart - Matplotlib")
plt.show()
```



In [ ]:
```
📝 Final Thoughts

    Use Matplotlib when you want full control or need unique charts.

    Use Seaborn when you want quick, stylish plots for data analysis.

Both are useful tools – and once you learn one, the other becomes much easie
```