

```
# Install (most are pre-installed in Colab)
!pip install matplotlib seaborn pandas
```

```
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.59.2)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: numpy>=1.23 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.0.2)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (25.0)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.2.3)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib)
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

sns.set(style="whitegrid") # nicer default style
```

```
np.random.seed(42)

months = np.arange(1, 13)
values = np.random.randint(20, 100, size=12)
category = np.random.choice(['A', 'B', 'C'], size=12)

df = pd.DataFrame({
    'Month': months,
    'Value': values,
    'Category': category
})

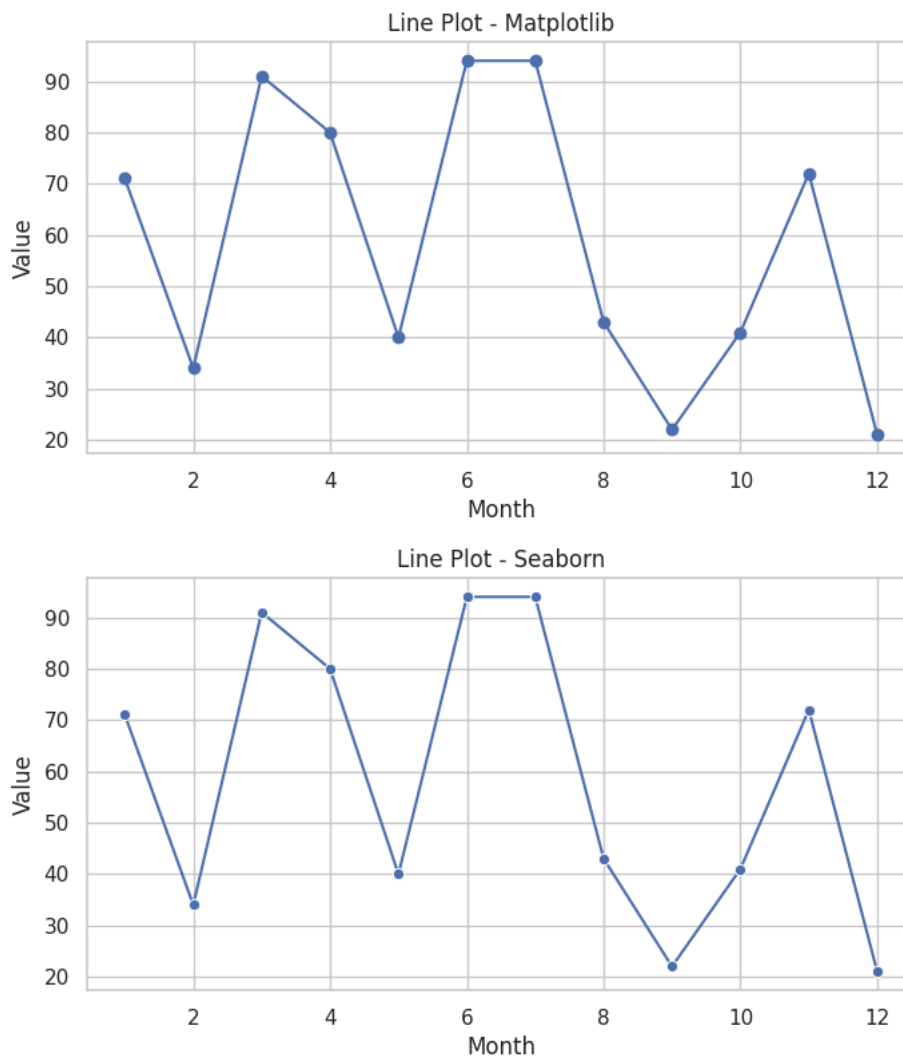
df.head()
```

	Month	Value	Category
0	1	71	B
1	2	34	B
2	3	91	B
3	4	80	A
4	5	40	A

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
# Matplotlib
plt.figure(figsize=(8,4))
plt.plot(df['Month'], df['Value'], marker='o')
plt.title("Line Plot - Matplotlib")
plt.xlabel("Month")
plt.ylabel("Value")
plt.show()

# Seaborn
plt.figure(figsize=(8,4))
sns.lineplot(data=df, x="Month", y="Value", marker="o")
plt.title("Line Plot - Seaborn")
plt.show()
```

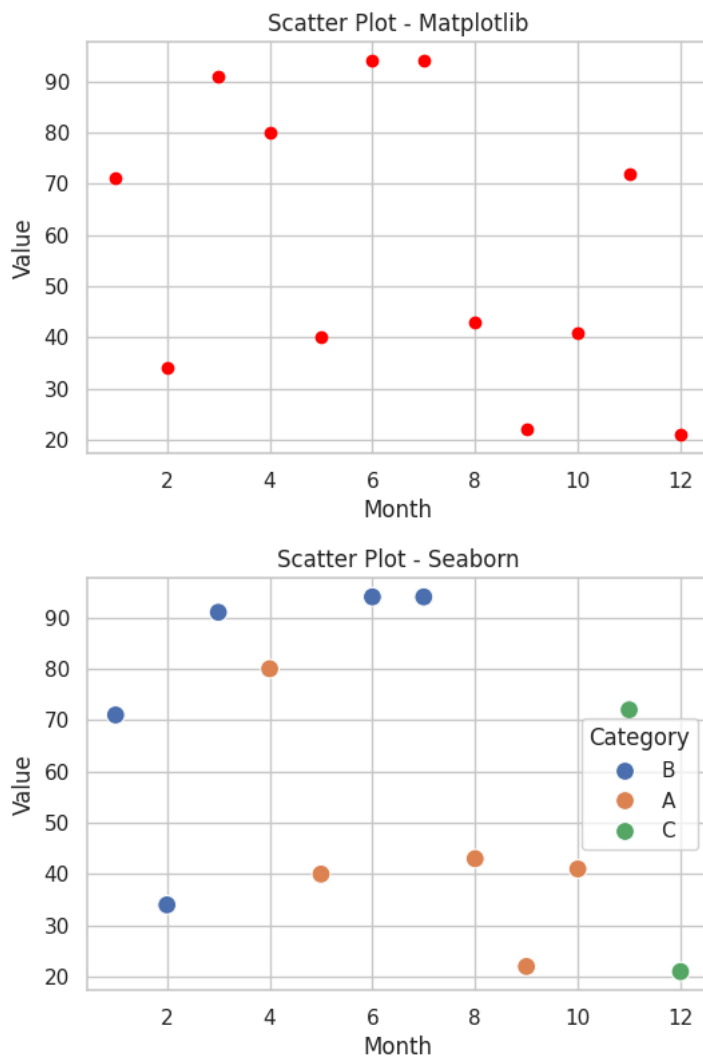


Line Plot

- **Description:** Shows how a variable changes over time.
- **Use case:** Stock prices, sales per month.
- **Example:** We used months vs. values.

```
# Matplotlib
plt.figure(figsize=(6,4))
plt.scatter(df['Month'], df['Value'], c='red')
plt.title("Scatter Plot - Matplotlib")
plt.xlabel("Month")
plt.ylabel("Value")
plt.show()

# Seaborn
plt.figure(figsize=(6,4))
sns.scatterplot(data=df, x="Month", y="Value", hue="Category", s=100)
plt.title("Scatter Plot - Seaborn")
plt.show()
```

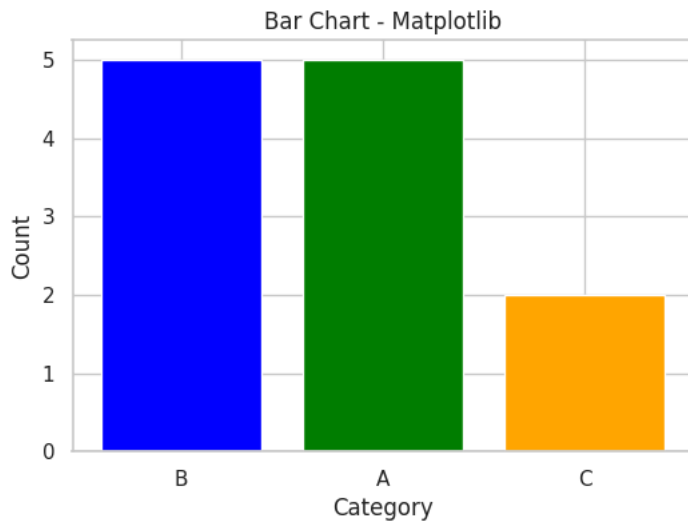


✓ Scatter Plot

- **Description:** Shows how a variable changes over time.
- **Use case:** Stock prices, sales per month.
- **Example:** We used months vs. values.

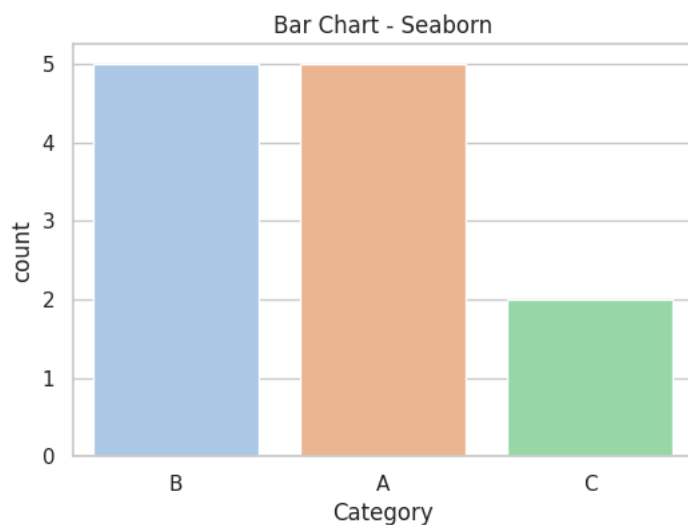
```
# Matplotlib
counts = df['Category'].value_counts()
plt.figure(figsize=(6,4))
plt.bar(counts.index, counts.values, color=['blue','green','orange'])
plt.title("Bar Chart - Matplotlib")
plt.xlabel("Category")
plt.ylabel("Count")
plt.show()

# Seaborn
plt.figure(figsize=(6,4))
sns.countplot(data=df, x="Category", palette="pastel")
plt.title("Bar Chart - Seaborn")
plt.show()
```



/tmp/ipython-input-293514649.py:12: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `x` variable to `hue` and
 sns.countplot(data=df, x="Category", palette="pastel")

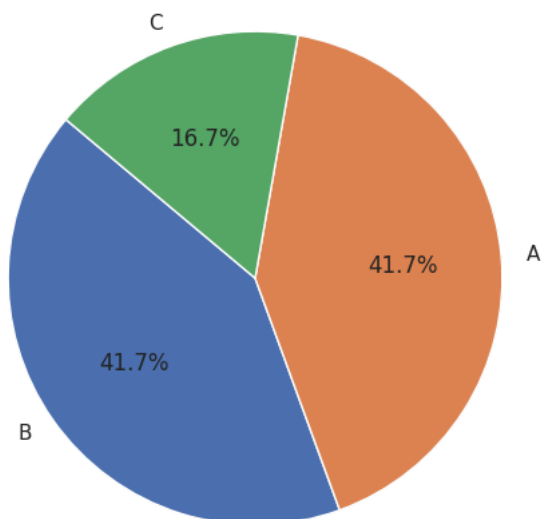


Bar Chart

- **Description:** Shows how a variable changes over time.
- **Use case:** Stock prices, sales per month.
- **Example:** We used months vs. values.

```
# Matplotlib only (Seaborn doesn't support pie)
sizes = counts.values
labels = counts.index
plt.figure(figsize=(6,6))
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title("Pie Chart - Matplotlib")
plt.show()
```

Pie Chart - Matplotlib



✓ Pie Chart

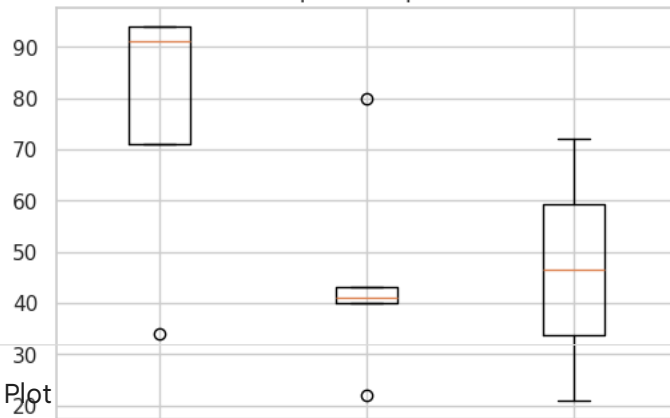
- **Description:** Shows how a variable changes over time.
- **Use case:** Stock prices, sales per month.
- **Example:** We used months vs. values.

```
# Matplotlib
plt.figure(figsize=(6,4))
plt.boxplot([df[df['Category']==c]['Value'] for c in df['Category'].unique()],
            labels=df['Category'].unique())
plt.title("Boxplot - Matplotlib")
plt.show()

# Seaborn
plt.figure(figsize=(6,4))
sns.boxplot(data=df, x="Category", y="Value", palette="Set2")
plt.title("Boxplot - Seaborn")
plt.show()
```

```
/tmp/ipython-input-3453003975.py:3: MatplotlibDeprecationWarning: The 'labels' parameter of boxplot() has been renamed 'tick_labels'; please update your code.
plt.boxplot([df[df['Category']==c]['Value'] for c in df['Category'].unique()],
```

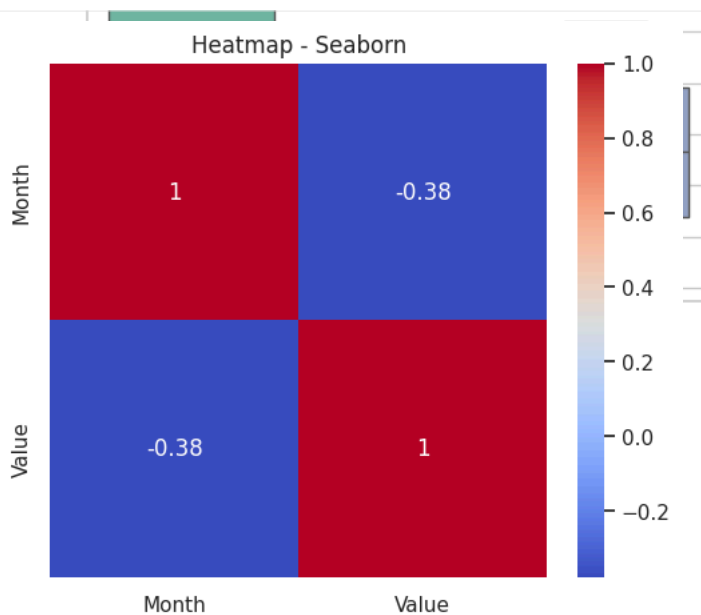
Boxplot - Matplotlib



Box Plot

- **Description:** Shows how a variable changes over time.
- **Use case:** Stock prices, sales per month.
- **Example:** We use month as values

```
plt.figure(figsize=(6,5))
corr = df.corr(numeric_only=True)
sns.heatmap(corr, annot=True, cmap="coolwarm")
plt.title("Heatmap - Seaborn")
plt.show()
```



Heatman