Matplotlib & Seaborn: The Ultimate Syntax Guide

This guide is designed to help you understand the **structure** of Python plotting libraries and remember the **syntax patterns** you'll use every day.

1. The Mental Model

Understanding *how* these libraries think makes remembering the code much easier.

Matplotlib: The "Artist" Model

Think of Matplotlib as building a painting layer by layer.

- **Figure**: The blank canvas (the whole window).
- **Axes**: The specific plot area where you draw (x-y coordinates).
- **Artists**: Everything you see (lines, dots, text, legends) are "artists" drawn onto the Axes.

The Golden Rule: Always create a Figure and Axes first.

```
fig, ax = plt.subplots()  # Create Figure (fig) and Axes (ax)
ax.plot(x, y)             # Draw on the Axes
```

Seaborn: The "Data-Centric" Model

Seaborn is built on top of Matplotlib but thinks in terms of **DataFrames**.

- **Data**: You pass your entire DataFrame.
- **Mapping**: You tell it which columns map to which visual element (x, y, color, size).

The Golden Rule: Pass the DataFrame and map column names.

```
sns.plot_type(data=df, x='column_A', y='column_B', hue='category_C')
```

---

2. Universal Syntax Patterns

Memorize these patterns, and you can create almost any plot.

Matplotlib Pattern

`plt.plot_type(x_values, y_values, formatting_options)`

Component	Example	Description
x_values	`[1, 2, 3]`	List or array for X-axis

Component	Example	Description
y_values	`[4, 5, 6]`	List or array for Y-axis
Formatting	`color='red'`, `label='My Line'`	Optional styling

Seaborn Pattern

```
sns.plot_type(data=df, x='col_name', y='col_name', hue='group_col')
```

Component	Example	Description
data	`data=df`	The Pandas DataFrame
x, y	`x='Age'`, `y='Salary'`	Column names for axes
hue	`hue='Gender'`	Column to group by color (Automatic legend!)

3. Essential Methods Cheat Sheet

These are the methods you will use 90% of the time.

Setup & Layout

Method	Syntax	Description
Create Plot	`plt.figure(figsize=(10, 6))`	Start a new figure with specific size
Subplots	`plt.subplot(rows, cols, index)`	Create multiple plots in a grid
Show	`plt.show()`	Display the plot (required in scripts)

Labeling & Decorating

Method	Syntax	Description
Title	`plt.title("My Title")`	Add a title to the current axes
Labels	`plt.xlabel("Time")`, `plt.ylabel("Value")`	Label the X and Y axes
Legend	`plt.legend()`	Show the legend (requires `label=` in plot)
Grid	`plt.grid(True)`	Turn on background grid lines

4. Plot Types Reference

Relational Plots (Trends & Relationships)

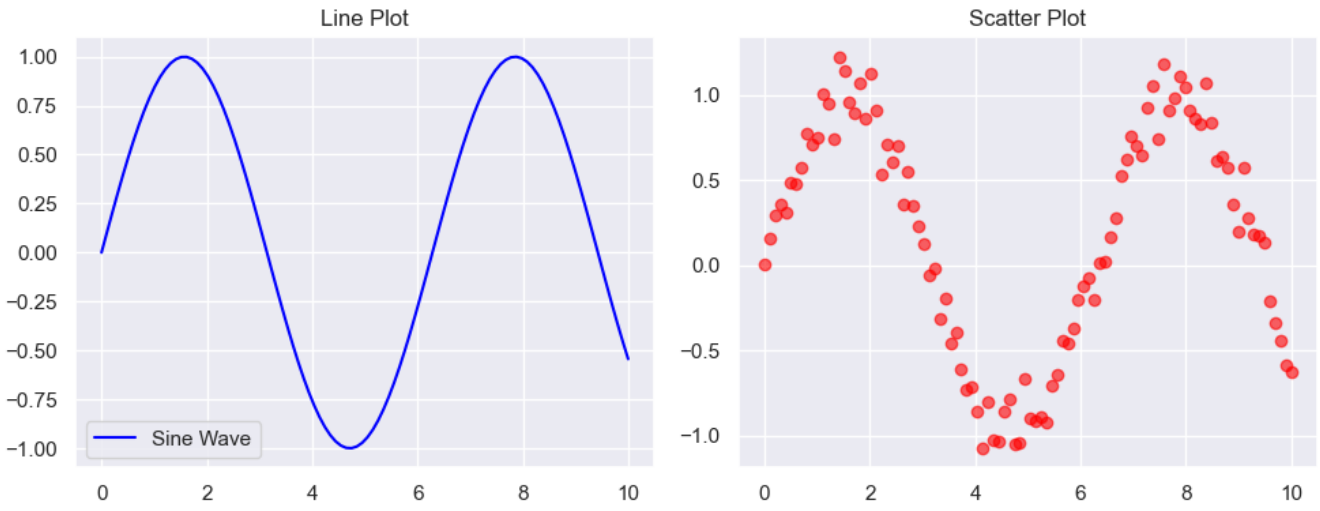
Line Plot

- **Use when**: Tracking changes over time or ordered data.
- **Matplotlib**: `plt.plot(x, y)`
- **Seaborn**: `sns.lineplot(data=df, x='time', y='value')`

Scatter Plot

- **Use when**: Seeing relationships/correlations between two variables.
- **Matplotlib**: `plt.scatter(x, y)`
- **Seaborn**: `sns.scatterplot(data=df, x='age', y='income', hue='gender')`

Visual Example:

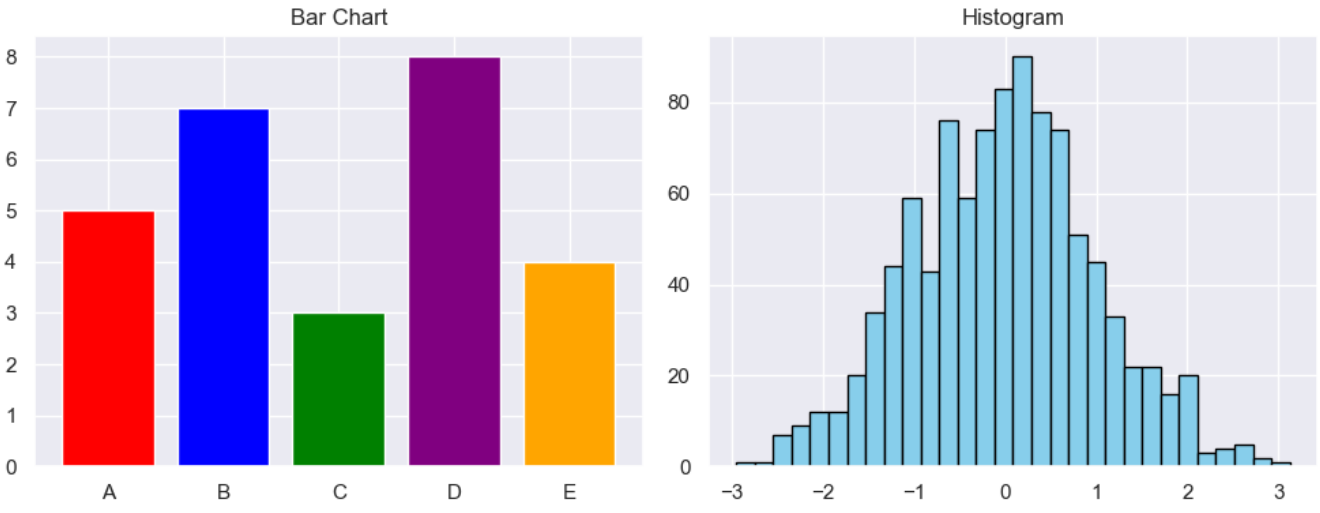


Distribution Plots (Spread & Frequency)

Histogram

- **Use when**: Checking the distribution of a single numerical variable.
- **Matplotlib**: `plt.hist(x, bins=20)`
- **Seaborn**: `sns.histplot(data=df, x='age', kde=True)` (KDE adds the smooth curve)

Visual Example:

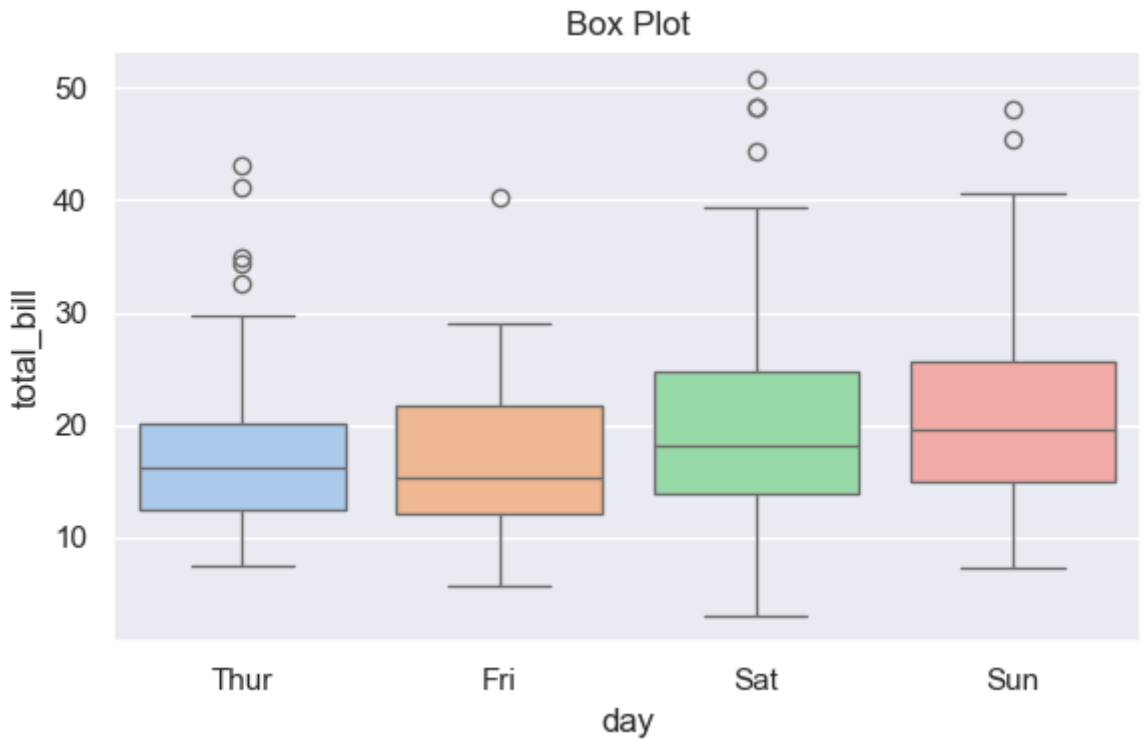


(Right side shows Histogram)

Box Plot

- **Use when**: Comparing distributions across categories (e.g., Salary by Department).
- **Seaborn Box**: `sns.boxplot(data=df, x='dept', y='salary')`

Visual Example:

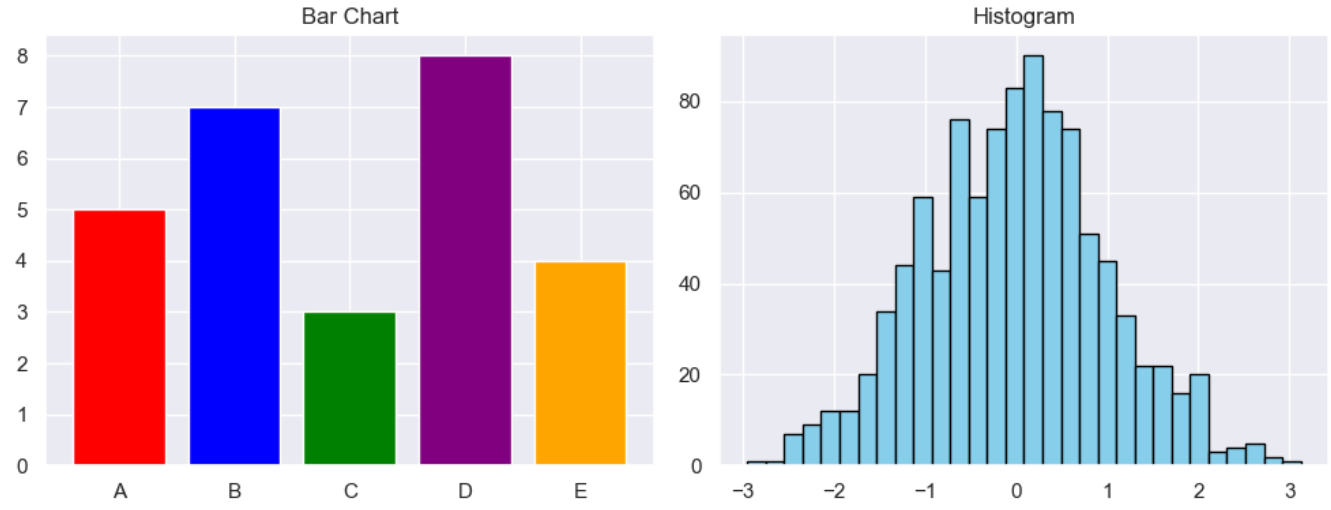


Categorical Plots (Comparison)

Bar Chart

- **Use when**: Comparing counts or values of categorical groups.
- **Matplotlib**: `plt.bar(categories, values)`
- **Seaborn**: `sns.barplot(data=df, x='category', y='value')`

Visual Example:



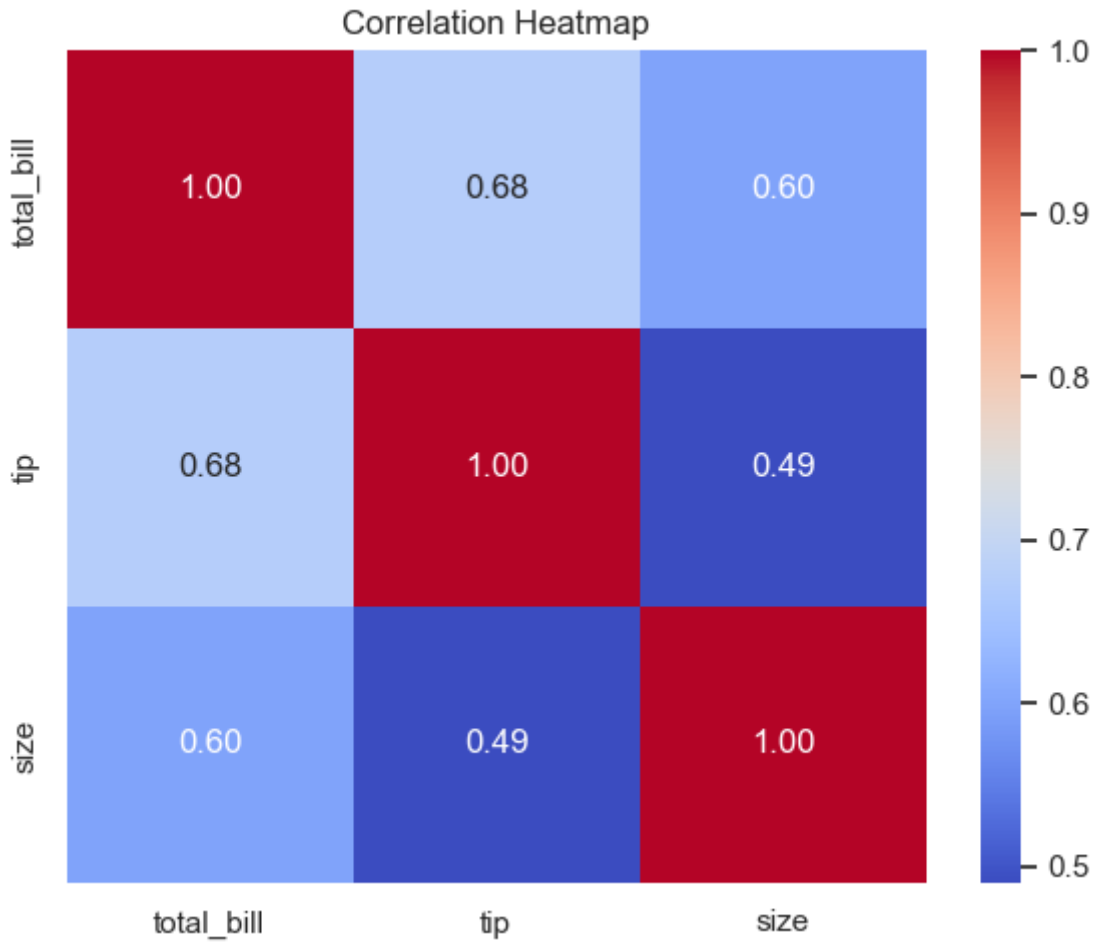
(Left side shows Bar Chart)

Matrix Plots (Correlations)

Heatmap

- **Use when**: Visualizing a matrix of numbers (like correlations) with color.
- **Seaborn**: `sns.heatmap(df.corr(), annot=True, cmap='coolwarm')`

Visual Example:



Exploration

Pair Plot

- **Use when**: Quickly exploring relationships between ALL variables in a dataset.
- **Seaborn**: `sns.pairplot(df, hue='species')`

Visual Example: