Python supports a **wide variety of plots** for data visualization. Below is a categorized list of **all major plot types**, with their **usage and syntax** using **Matplotlib, Seaborn, and Plotly**.

---

1. Basic Plots

1.1 Line Plot (Trend Analysis)

- **Used for:** Time-series data, trends
- **Library:** `matplotlib`

```
import matplotlib.pyplot as plt

x = [1, 2, 3, 4, 5]
y = [10, 12, 15, 18, 22]

plt.plot(x, y, marker='o', linestyle='-', color='b')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Line Plot')
plt.show()
```

1.2 Scatter Plot (Relationship Between Two Variables)

- **Used for:** Finding correlations and patterns
- **Library:** `matplotlib`

```
import numpy as np

x = np.random.rand(50)
y = np.random.rand(50)

plt.scatter(x, y, color='r')
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')
plt.show()
```

1.3 Bar Plot (Comparing Categorical Data)

- **Used for:** Comparing discrete categories
- **Library:** `matplotlib`

```
categories = ['A', 'B', 'C', 'D']
values = [10, 20, 15, 25]

plt.bar(categories, values, color='g')
plt.xlabel('Category')
plt.ylabel('Value')
plt.title('Bar Plot')
plt.show()
```

1.4 Histogram (Distribution of Data)

- **Used for:** Understanding the frequency distribution of data
- **Library:** `matplotlib`

```
data = np.random.randn(1000)

plt.hist(data, bins=30, color='purple', edgecolor='black')
plt.xlabel('Bins')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```

2. Advanced Statistical Plots

2.1 Box Plot (Outlier Detection)

- **Used for:** Detecting outliers and understanding data spread
- **Library:** `seaborn`

```
import seaborn as sns

data = [10, 12, 15, 18, 22, 25, 30, 100]  # 100 is an outlier

sns.boxplot(data=data)
plt.title('Box Plot')
plt.show()
```

2.2 Violin Plot (Data Distribution + Density)

- **Used for:** Distribution of data with KDE (Kernel Density Estimation)
- **Library:** `seaborn`

```
tips = sns.load_dataset('tips')
sns.violinplot(x='day', y='total_bill', data=tips)
plt.title('Violin Plot')
plt.show()
```

2.3 Pair Plot (Relationship Between All Features)

- **Used for:** Visualizing pairwise relationships in a dataset
- **Library:** `seaborn`

```
sns.pairplot(tips)
plt.show()
```

---

3. Distribution Plots

3.1 KDE Plot (Kernel Density Estimation)

- **Used for:** Estimating data distribution
- **Library:** `seaborn`

```
sns.kdeplot(data=tips['total_bill'], shade=True)
```

```
plt.title('KDE Plot')
plt.show()
```

3.2 Joint Plot (Scatter + KDE)

- **Used for:** Analyzing the relationship and distribution between two variables
- **Library:** `seaborn`

```
sns.jointplot(x='total_bill', y='tip', data=tips, kind='scatter')
plt.show()
```

---

4. Categorical Plots

4.1 Count Plot (Category Frequency)

- **Used for:** Counting occurrences of each category
- **Library:** `seaborn`

```
sns.countplot(x='day', data=tips)
plt.title('Count Plot')
plt.show()
```

4.2 Heatmap (Correlation Matrix)

- **Used for:** Displaying relationships between multiple variables
- **Library:** `seaborn`

```
import pandas as pd

correlation_matrix = tips.corr()
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm')
plt.title('Heatmap')
plt.show()
```

5. Time Series & Area Plots

5.1 Time Series Plot

- **Used for:** Visualizing trends over time
- **Library:** `matplotlib`

```
dates = pd.date_range('20230101', periods=10)
values = np.random.randn(10)

plt.plot(dates, values, marker='o')
plt.xticks(rotation=45)
plt.title('Time Series Plot')
plt.show()
```

5.2 Area Plot (Stacked Line Plot)

- **Used for:** Cumulative trends
- **Library:** `matplotlib`

```
x = range(1, 6)
y = [1, 3, 5, 7, 9]

plt.fill_between(x, y, color="skyblue", alpha=0.4)
plt.plot(x, y, color="Slateblue", alpha=0.6)
plt.title('Area Plot')
plt.show()
```

6. Pie & Donut Charts

6.1 Pie Chart (Percentage Distribution)

- **Used for:** Visualizing proportions

- **Library:** `matplotlib`

```
labels = ['A', 'B', 'C', 'D']
sizes = [30, 20, 25, 25]

plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140)
plt.title('Pie Chart')
plt.show()
```

6.2 Donut Chart (Pie Chart with a Hole)

```
plt.pie(sizes, labels=labels, autopct='%1.1f%%', startangle=140, wedgeprops={'edgecolor':
'black'})
plt.gca().add_artist(plt.Circle((0,0), 0.6, color='white'))
plt.title('Donut Chart')
plt.show()
```

---

7. 3D Plots

7.1 3D Scatter Plot

- **Used for:** Visualizing 3D relationships
- **Library:** `matplotlib`

```
from mpl_toolkits.mplot3d import Axes3D

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')

x = np.random.rand(50)
y = np.random.rand(50)
z = np.random.rand(50)

ax.scatter(x, y, z)
plt.title('3D Scatter Plot')
plt.show()
```

7.2 3D Surface Plot

```
X = np.linspace(-5, 5, 100)
Y = np.linspace(-5, 5, 100)
X, Y = np.meshgrid(X, Y)
Z = np.sin(np.sqrt(X**2 + Y**2))

fig = plt.figure()
ax = fig.add_subplot(111, projection='3d')
ax.plot_surface(X, Y, Z, cmap='viridis')
plt.title('3D Surface Plot')
plt.show()
```

Conclusion

📌 **Python offers powerful visualization libraries** like:

- **Matplotlib** (Basic plots)
- **Seaborn** (Statistical and advanced plots)
- **Plotly** (Interactive plots)

🔹 **Most Used Plots:**

Category	Plot
Basic	Line, Scatter, Bar, Histogram
Statistical	Box, Violin, Pair, KDE
Categorical	Count, Heatmap
Advanced	3D Scatter, Surface

Let me know if you need **interactive Plotly examples!** 🚀📊