Visualization with Seaborn

Seaborn is a powerful Python library built on top of Matplotlib for creating attractive and informative statistical graphics. It simplifies complex visualizations and provides a higher-level interface for drawing graphs.

Key Features of Seaborn

1. Built-in themes for styling (darkgrid, whitegrid, etc.).
2. High-level interfaces for statistical plots.
3. Integration with Pandas DataFrames for seamless data handling.
4. Support for categorical, distribution, and relational data visualizations.

Importing Seaborn

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Basic Syntax for Seaborn

1. **Categorical Plots**:
   - **sns.catplot(data=..., x=..., y=..., kind='...')**: Combines categorical plots like strip, swarm, box, etc.
2. **Distribution Plots**:
   - **sns.histplot(data=..., x=...)**: Draws a histogram.
   - **sns.kdeplot(data=..., x=...)**: Kernel Density Estimation.
3. **Relational Plots**:
   - **sns.relplot(data=..., x=..., y=..., kind='scatter/line')**.
4. **Heatmaps**:
   - **sns.heatmap(data=..., annot=True)**.
5. **Pairplots**:
   - **sns.pairplot(data=...)**: Visualizes pairwise relationships.

Example 1: Distribution Plot

```
import seaborn as sns
import matplotlib.pyplot as plt

# Sample Data
data = [1, 2, 2, 3, 3, 3, 4, 4, 5, 5, 5, 5, 6, 6, 7, 8, 9]

# Plotting Histogram with KDE
```

```
sns.histplot(data, kde=True, color="blue", bins=10)
plt.title("Distribution Plot")
plt.xlabel("Values")
plt.ylabel("Frequency")
plt.show()
```

1. **sns.histplot(data, kde=True)**:
   - Creates a histogram of the data list.
   - Adds a Kernel Density Estimate (KDE) curve to show the data distribution.
2. **Customization**:
   - **color="blue"**: Sets the histogram color.
   - **bins=10**: Specifies the number of bins in the histogram.
3. **Title and Labels**:
   - Titles and axis labels are added using Matplotlib functions.

Example 2: Scatter Plot with Regression Line

```
import seaborn as sns
import matplotlib.pyplot as plt

# Load Sample Dataset
tips = sns.load_dataset("tips")

# Scatter Plot with Regression Line
sns.lmplot(data=tips, x="total_bill", y="tip", hue="sex", markers=["o", "s"], height=6)
plt.title("Scatter Plot with Regression Line")
plt.xlabel("Total Bill")
plt.ylabel("Tip Amount")
plt.show()
```

1. **Dataset**:
   - The tips dataset contains data about restaurant bills and tips.
2. **sns.lmplot()**:
   - Creates a scatter plot with a regression line.
   - **hue="sex"**: Colors the points based on gender (male or female).
   - **markers=["o", "s"]**: Specifies different markers for categories in hue.
   - **height=6**: Sets the size of the plot.
3. **Title and Labels**:
   - Customizes the plot with Matplotlib labels and titles.

Example 3: Heatmap

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

# Sample Data
data = np.random.rand(5, 5)

# Heatmap
sns.heatmap(data, annot=True, cmap="coolwarm", linewidths=0.5)
plt.title("Heatmap Example")
plt.show()
```

1. **np.random.rand(5, 5)**:
   - Generates a 5x5 matrix of random values.
2. **sns.heatmap()**:
   - Creates a heatmap of the matrix.
   - **annot=True**: Annotates each cell with its value.
   - **cmap="coolwarm"**: Sets the colormap to coolwarm.
   - **linewidths=0.5**: Adds gridlines between cells.
3. **Title**:
   - Adds a title to the heatmap.

Key Seaborn Functions at a Glance

Function	Description
sns.catplot()	Categorical plots like boxplot, barplot, etc.
sns.histplot()	Histograms and KDEs.
sns.scatterplot()	Scatter plots for relational data.
sns.lineplot()	Line plots for trends over intervals.
sns.heatmap()	Heatmaps for matrix-like data.
sns.pairplot()	Pairwise relationships in data.
sns.boxplot()	Boxplots for visualizing distributions.

Advantages of Seaborn

1. Easy integration with Pandas DataFrames.
2. Simplifies the creation of statistical graphics.
3. Built-in support for aesthetics and themes.