Seaborn

Seaborn is a Python data visualization library built on top of Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.

Key Features of Seaborn for Plotting

- Simplifies complex statistical plots with fewer lines of code.
- Integrates well with Pandas DataFrames for direct plotting from data.
- Provides beautiful default styles and color palettes.
- Supports visualizing distributions, relationships, categorical data, and matrix data.

Common Plot Types in Seaborn

- Scatter Plot (sns.scatterplot): Visualizes relationships between two variables.
- Line Plot (sns.lineplot): Displays trends over an interval or ordered categories.
- Bar Plot (sns.barplot): Shows aggregate values across categories.
- Histogram & KDE Plot (sns.histplot, sns.kdeplot): Displays data distributions.
- Box Plot (sns.boxplot): Summarizes data distribution by quartiles and outliers.
- Violin Plot (sns.violinplot): Combines box plot and KDE to show distribution shape.
- Heatmap (sns.heatmap): Shows matrix-style data with color coding.
- Pair Plot (sns.pairplot): Plots pairwise relationships in a dataset.

Example: Simple Bar Plot

python

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

# Load example dataset
tips = sns.load_dataset("tips")

# Create bar plot
sns.barplot(x="day", y="total_bill", data=tips)

plt.title("Total Bill by Day")
plt.show()
```

Example: Scatter Plot with Regression Line

python

```python
sns.lmplot(x="total_bill", y="tip", data=tips)
plt.title("Tip vs Total Bill with Regression Line")
plt.show()
```

Customization

- Change color palettes with palette parameter.
- Use hue to add a third categorical variable.
- Control plot elements like confidence intervals, markers, line styles.

Visualizing distributions, correlations, and relationships

Visualizing distributions, correlations, and relationships is essential in exploratory data analysis to understand data patterns and dependencies. Python libraries like Matplotlib and Seaborn provide versatile tools for these tasks.

Visualizing Distributions

Histogram: Shows frequency distribution of numerical data.

python

```
import matplotlib.pyplot as plt
plt.hist(data, bins=30)
plt.show()
```

Seaborn histplot & kdeplot: Combines histogram and Kernel Density Estimate (KDE) for smooth distribution.

python

```
import seaborn as sns
sns.histplot(data, kde=True)
plt.show()
```

Box Plot: Summarizes data distribution highlighting median, quartiles, and outliers.

python

```
sns.boxplot(x=data)
plt.show()
```

Violin Plot: Combines box plot and KDE to show distribution shape.

python

```
sns.violinplot(x=data)
plt.show()
```

Visualizing Correlations

Heatmap of correlation matrix: Shows the strength and direction of correlations between variables.

python

```
corr = df.corr()
sns.heatmap(corr, annot=True, cmap='coolwarm')
plt.show()
```

Visualizing Relationships

Scatter Plot: Shows relationship between two numerical variables.

python

```
sns.scatterplot(x='var1', y='var2', data=df)
plt.show()
```

Pair Plot: Visualizes pairwise relationships and distributions for all variables in a DataFrame.

python

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

Joint Plot: Combines scatter plot and distribution plots for two variables.

python

```
sns.jointplot(x='var1', y='var2', data=df, kind='reg')
plt.show()
```

Summary

- Use histograms, box, and violin plots to explore data distributions.
- Use heatmaps to understand correlations between variables.
- Use scatter, pair, and joint plots to analyze relationships and dependencies between variables.

These visualizations help reveal underlying trends, outliers, and relationships that guide further analysis or modeling.