20INMCA501 - Data Science & Machine Learning

Assignment: 01

Data Visualization Techniques

Aleena Ginu

Roll no:05

INT MCA – S9

Data Visualization Techniques

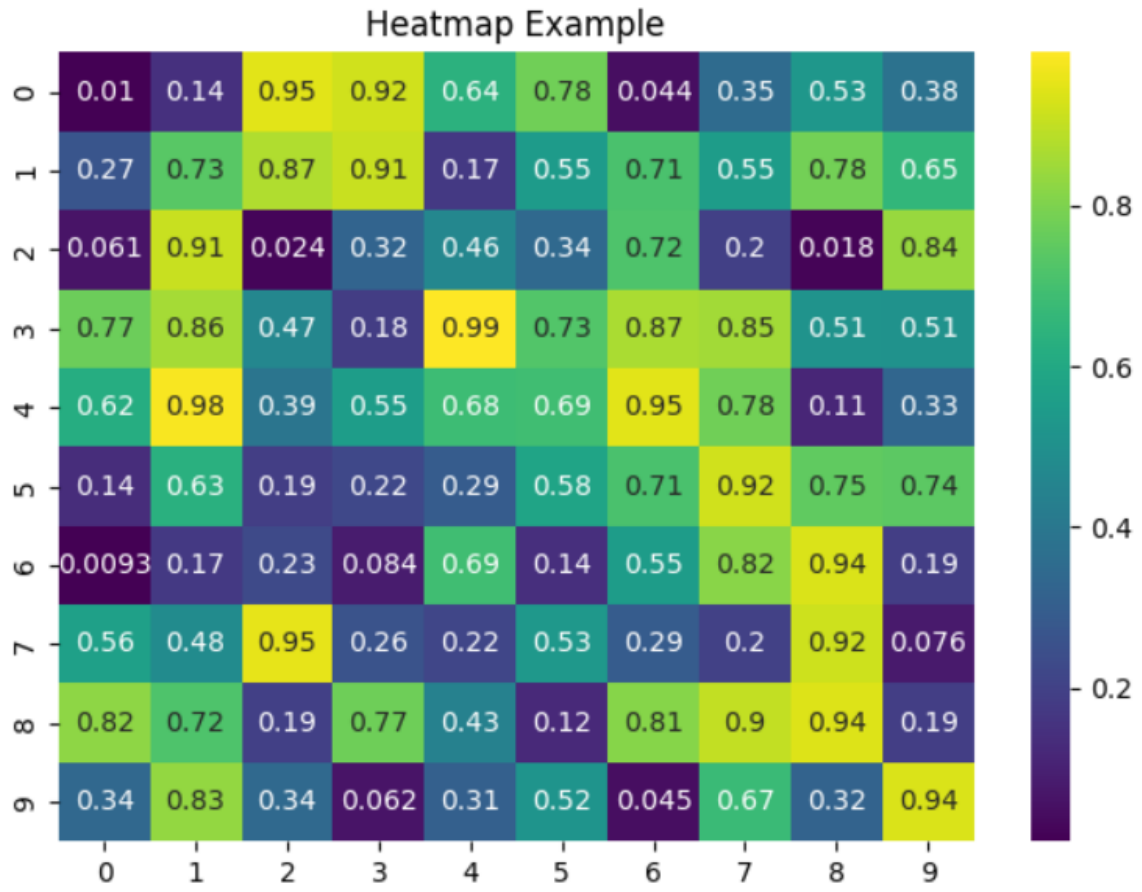
1.Heatmap

A heatmap uses color to represent data values in a matrix format. It is effective for visualizing data density and patterns over a grid. Colors vary based on intensity, which makes it easy to identify trends and correlations.

Python Code:

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

data = np.random.rand(10, 10)
sns.heatmap(data, annot=True, cmap='viridis')
plt.title('Heatmap Example')
plt.show()
```



Special Parameters:

- annot: If True, it writes the data value in each cell.
- cmap: Color map for the heatmap.
- linewidths: Width of the lines separating cells.

2. Bubble Chart

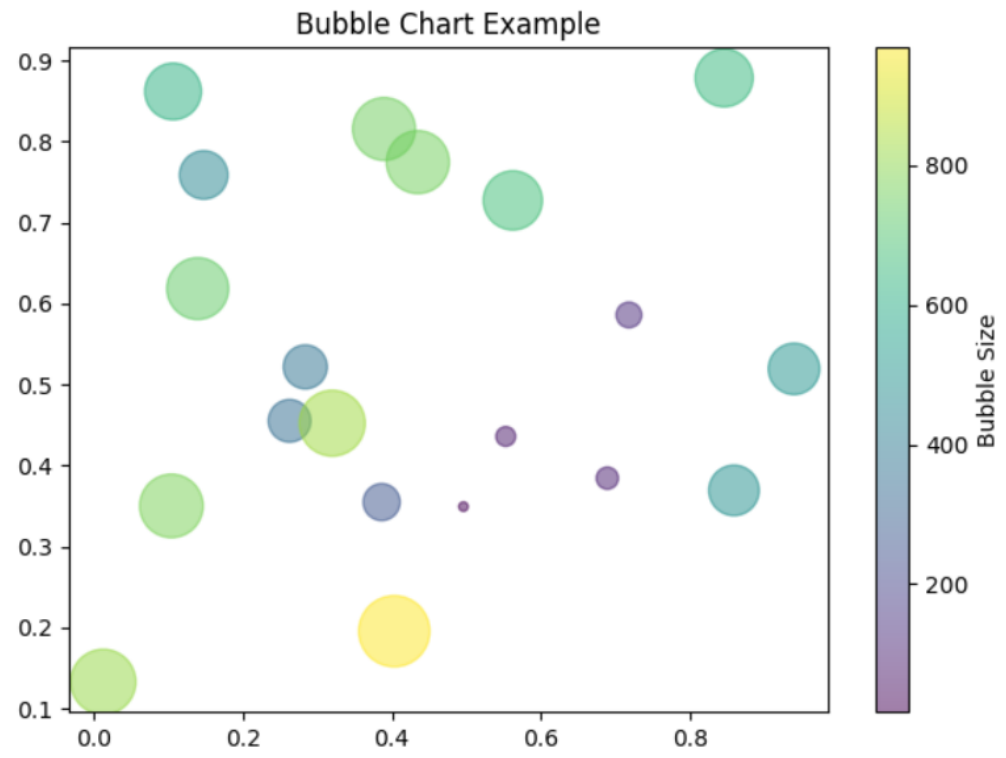
A bubble chart plots data points as bubbles, where the size and color of each bubble represent additional dimensions of data. It's useful for displaying the relationship between three variables and visualizing the magnitude of differences.

Python Code:

```python
import matplotlib.pyplot as plt

x = np.random.rand(20)
y = np.random.rand(20)
sizes = 1000 * np.random.rand(20)

plt.scatter(x, y, s=sizes, alpha=0.5, c=sizes, cmap='viridis')
plt.title('Bubble Chart Example')
plt.colorbar(label='Bubble Size')
plt.show()
```



Special Parameters:

- s: Bubble size.
- alpha: Transparency of the bubbles.
- c: Color of the bubbles.
- cmap: Color map for bubble color.

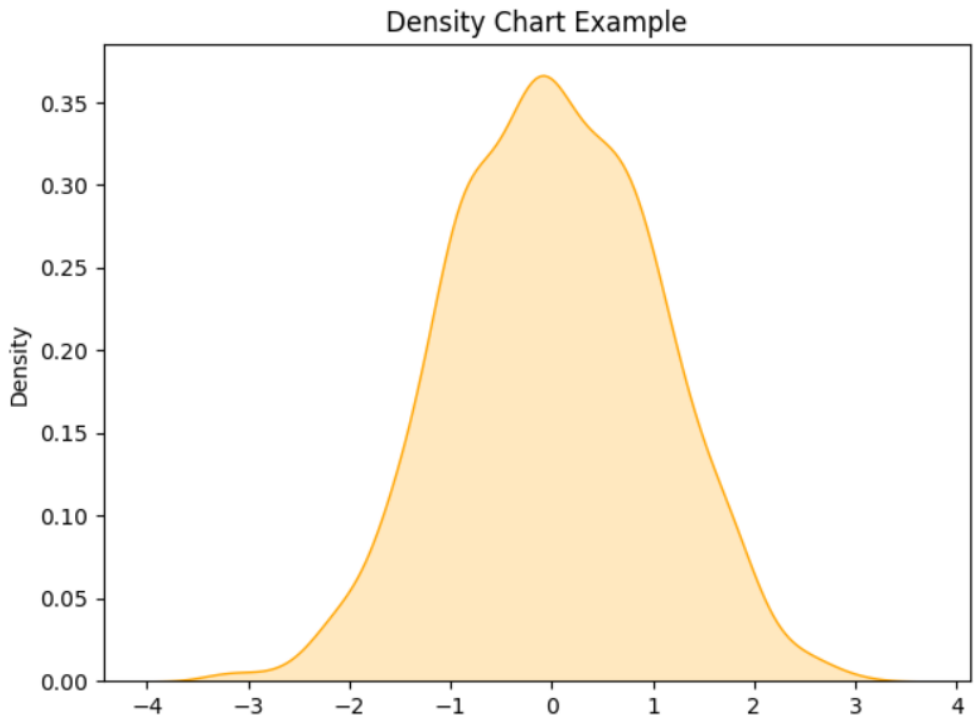
3. Density Chart

A density chart (or Kernel Density Estimate plot) shows the distribution of data over a continuous interval. It provides a smoothed estimate of the probability density function, helping to visualize the data's underlying distribution.

Python Code:

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

data = np.random.randn(1000)
sns.kdeplot(data, shade=True,color='orange')
plt.title('Density Chart Example')
plt.show()
```



Special Parameters:

- shade: If True, fills the area under the KDE curve.
- bw_adjust: Adjusts the bandwidth of the kernel.

4. Scatterplot

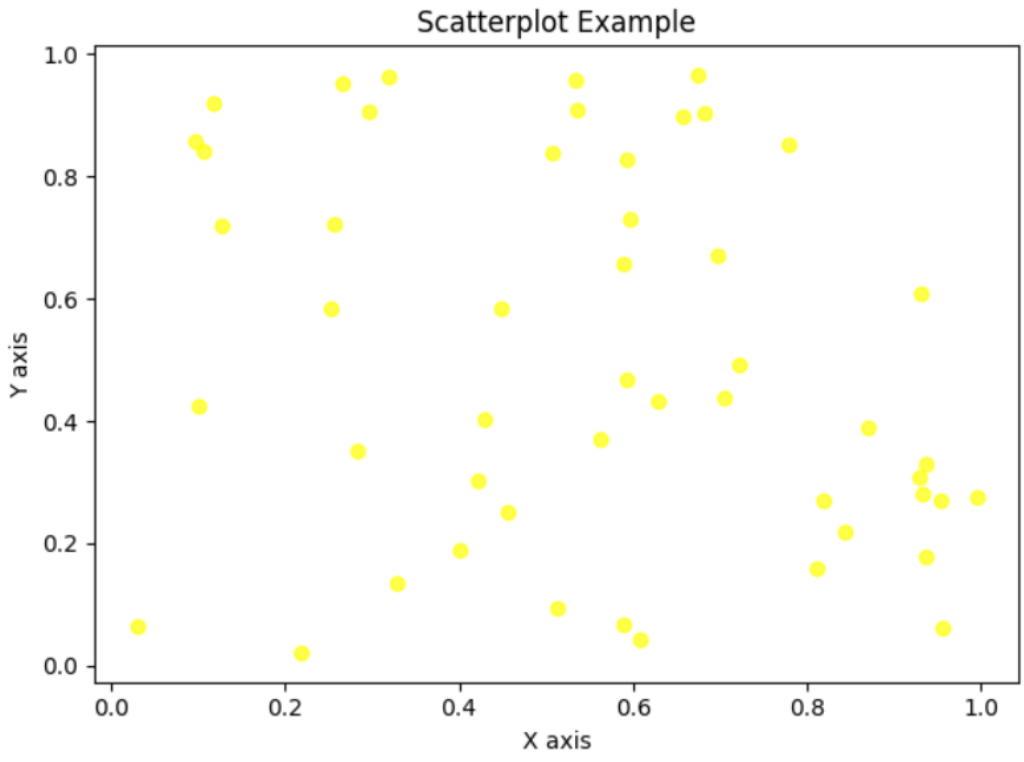
A scatterplot displays data points on a two-dimensional axis to show the relationship between two variables. It is useful for identifying correlations, clusters, and outliers in the dataset.

Python Code:

```
import matplotlib.pyplot as plt

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

plt.scatter(x, y, alpha=0.7,color='yellow')
plt.title('Scatterplot Example')
plt.xlabel('X axis')
plt.ylabel('Y axis')
plt.show()
```



Special Parameters:

- alpha: Transparency of points.
- c: Color of the points.
- s: Size of the points.

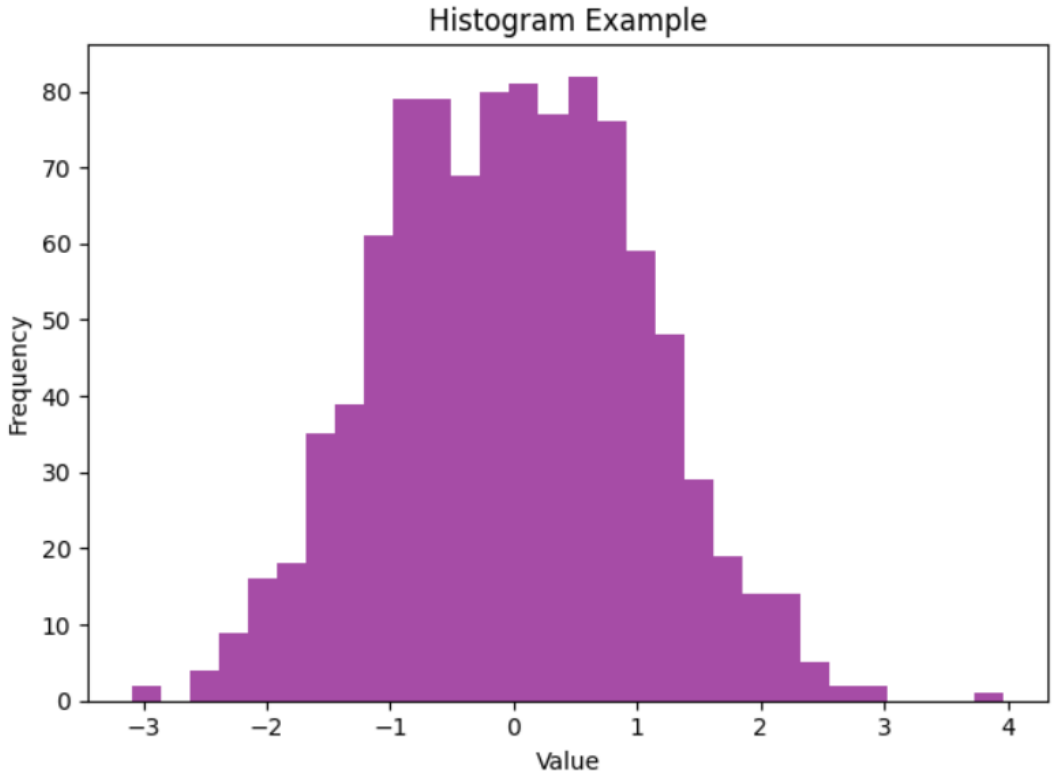
5. Histogram

A histogram represents the frequency distribution of a dataset using bars. Each bar corresponds to an interval or bin, making it easy to see how data values are distributed across different ranges.

Python Code:

```
import matplotlib.pyplot as plt

data = np.random.randn(1000)
plt.hist(data, bins=30, alpha=0.7, color='purple')
plt.title('Histogram Example')
plt.xlabel('Value')
plt.ylabel('Frequency')
plt.show()
```



Special Parameters:

- bins: Number of bins or intervals.
- alpha: Transparency of the bars.
- color: Color of the bars.

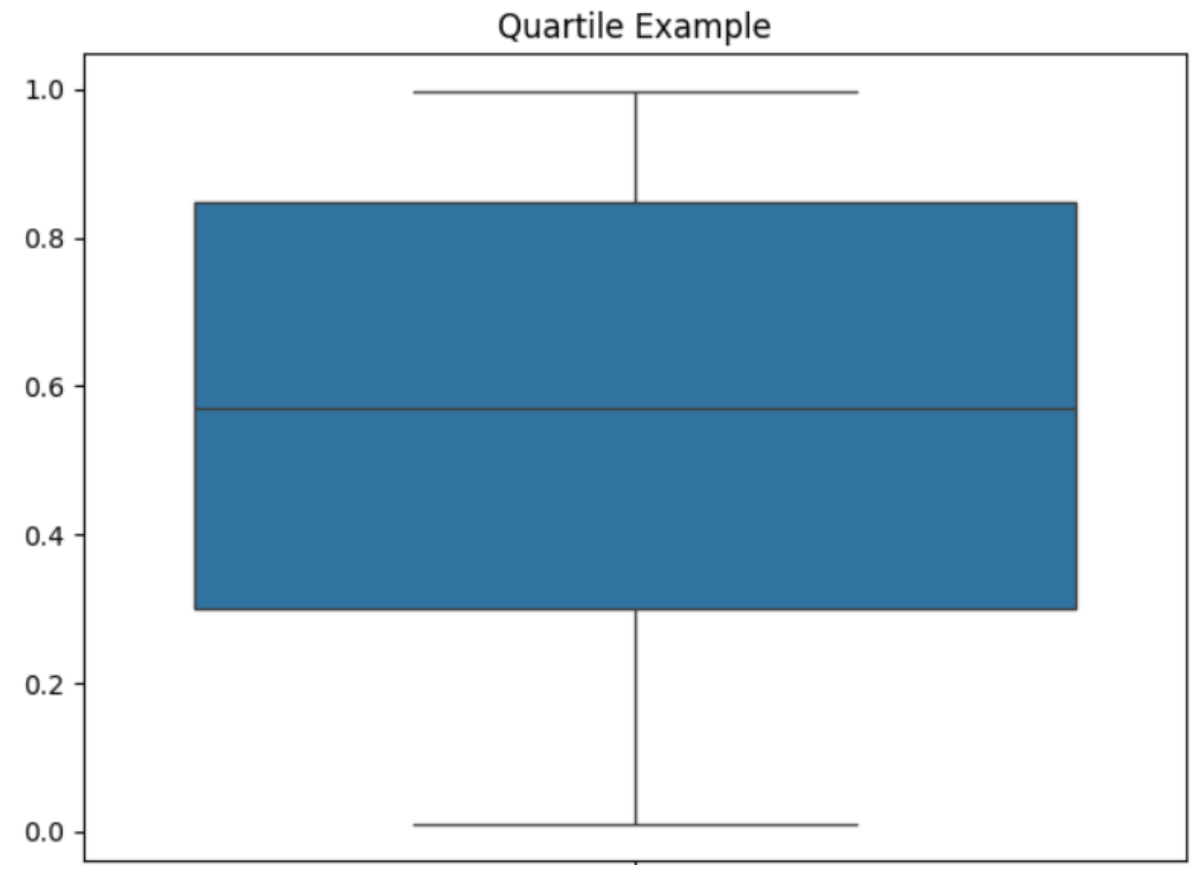
6. Quartile Plot (Boxplot)

A boxplot visualizes data distribution through quartiles, highlighting the median, upper and lower quartiles, and potential outliers. It provides a summary of the data's spread and central tendency.

Python Code:

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

data = np.random.rand(100)
sns.boxplot(data)
plt.title('Quartile Example')
plt.show()
```



Special Parameters:

- whis: Determines the reach of the whiskers (e.g., 1.5 for 1.5 times the IQR).
- notch: If True, makes a notched boxplot.
- vert: If False, makes a horizontal boxplot.

7. Pie Chart

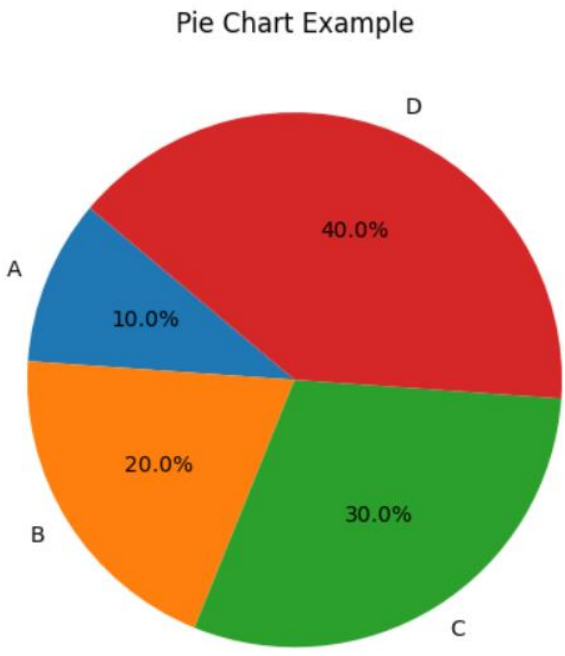
A pie chart shows proportions of a whole as slices of a circle. Each slice represents a category's relative contribution to the total, making it easy to see the composition of data at a glance.

Python Code:

```python
import matplotlib.pyplot as plt

sizes = [10, 20, 30, 40]
labels = ['A', 'B', 'C', 'D']

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



Special Parameters:

- autopct: Format string for showing percentages.
- startangle: Rotation angle of the pie chart.

- explode: A tuple to highlight specific slices.

8. Bar Chart

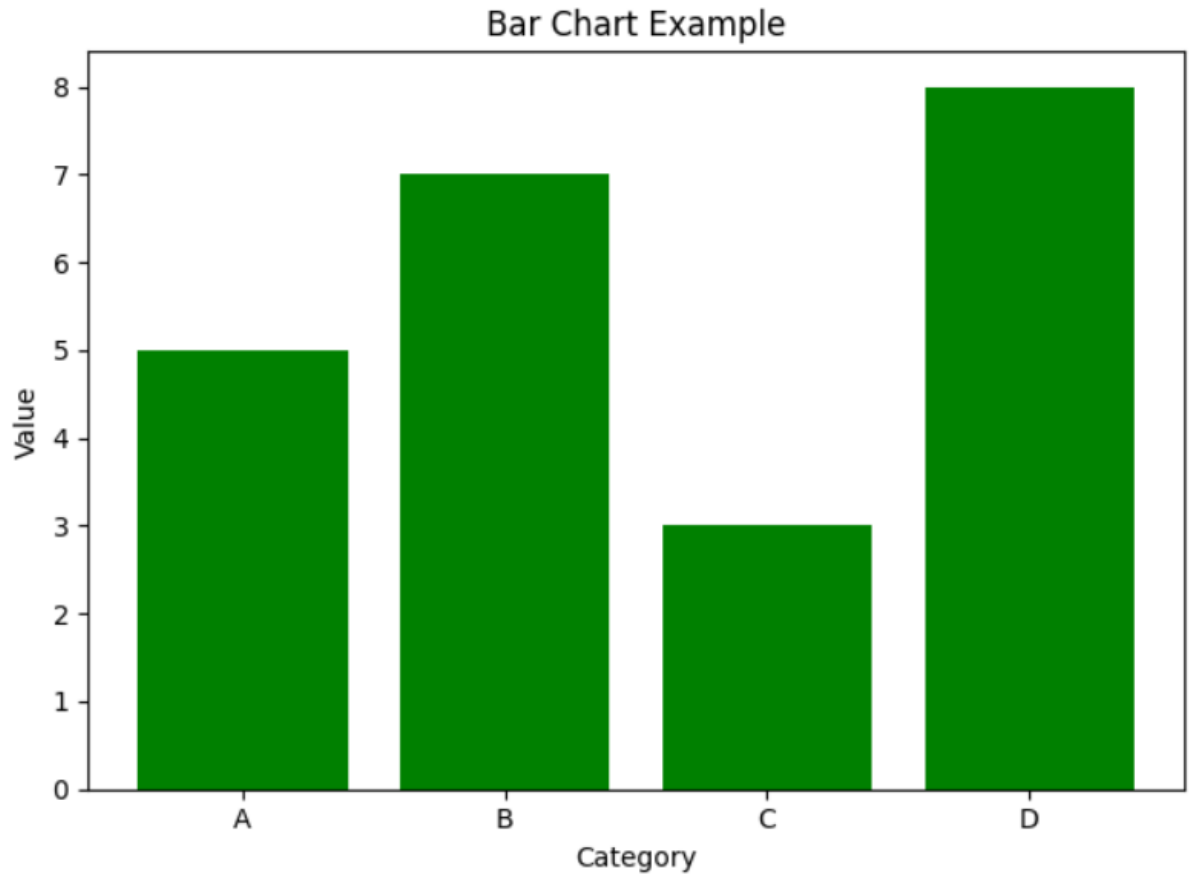
A bar chart uses rectangular bars to represent categorical data. The length of each bar correlates with the value it represents, making it effective for comparing quantities across different categories.

Python Code:

```
import matplotlib.pyplot as plt

categories = ['A', 'B', 'C', 'D']
values = [5, 7, 3, 8]

plt.bar(categories, values, color='Green')
plt.title('Bar Chart Example')
plt.xlabel('Category')
plt.ylabel('Value')
plt.show()
```



Special Parameters:

- color: Color of the bars.
- align: Alignment of bars ('center' or 'edge').
- width: Width of the bars.

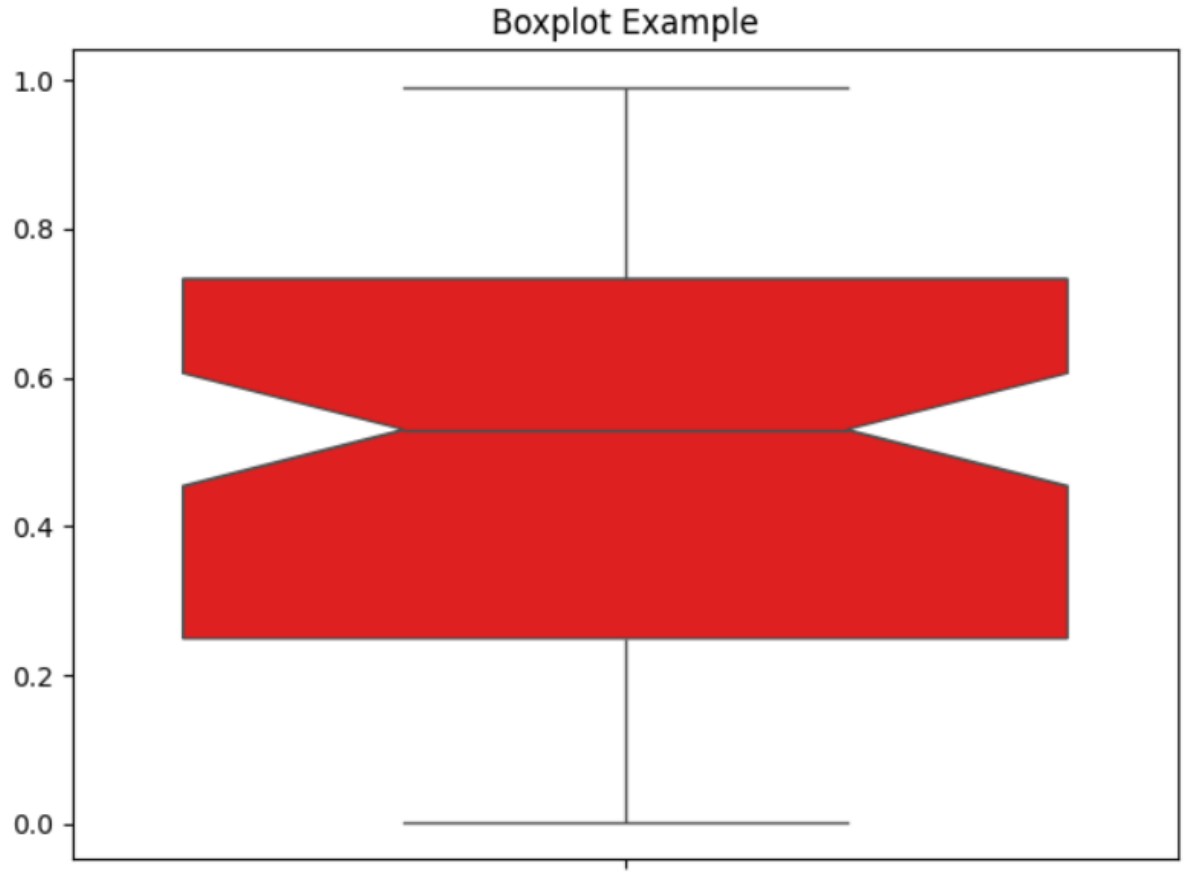
9. Boxplot

A boxplot displays data distribution through quartiles and visualizes outliers. It highlights the median, interquartile range, and variability, providing insights into the data's spread and central tendency.

Python Code:

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

data = np.random.rand(100)
sns.boxplot(data,notch=True,color='red')
plt.title('Boxplot Example')
plt.show()
```



Special Parameters:

- whis: Determines the reach of the whiskers (e.g., 1.5 for 1.5 times the IQR).
- notch: If True, makes a notched boxplot.
- vert: If False, makes a horizontal boxplot.

10. Area Chart

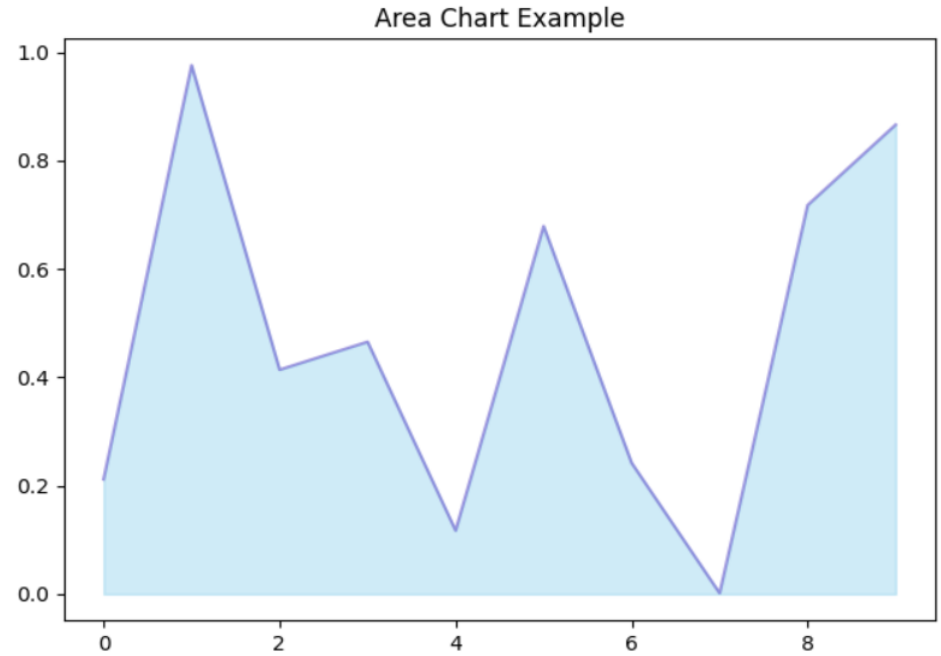
An area chart shows data trends over time with the area below the line filled in. It emphasizes the magnitude of change and the cumulative total of values, making it useful for visualizing time series data.

Python Code:

```
import matplotlib.pyplot as plt

x = np.arange(10)
y = np.random.rand(10)

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



Special Parameters:

- color: Fill color of the area.

- alpha: Transparency of the fill.