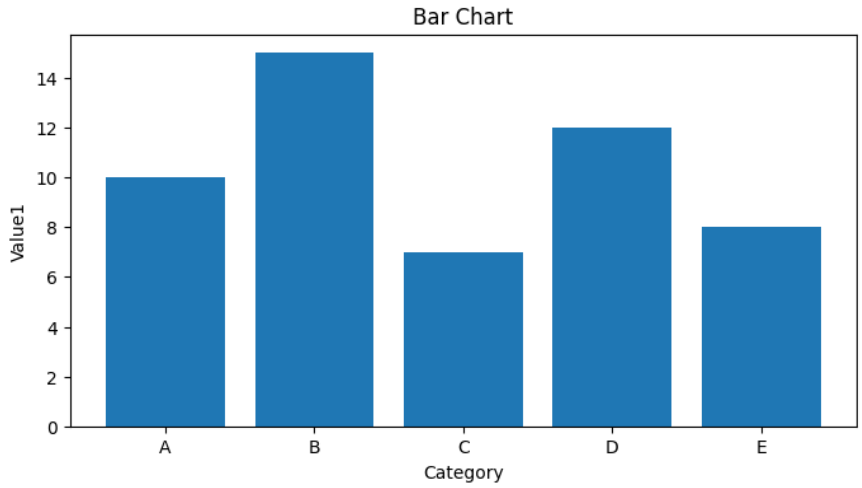
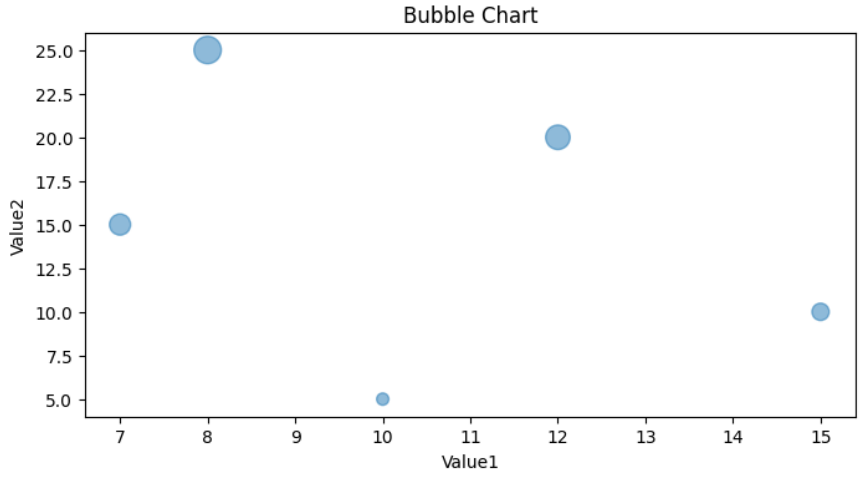
BAR CHART

```
import pandas as pd
import matplotlib.pyplot as plt

# Sample DataFrame
data = {
    'Category': ['A', 'B', 'C', 'D', 'E'],
    'Value1': [10, 15, 7, 12, 8],
    'Value2': [5, 10, 15, 20, 25]
}

df = pd.DataFrame(data)

# Create a Bar Chart
plt.figure(figsize=(8, 4))
plt.bar(df['Category'], df['Value1'])
plt.xlabel('Category')
plt.ylabel('Value1')
plt.title('Bar Chart')
plt.show()
```



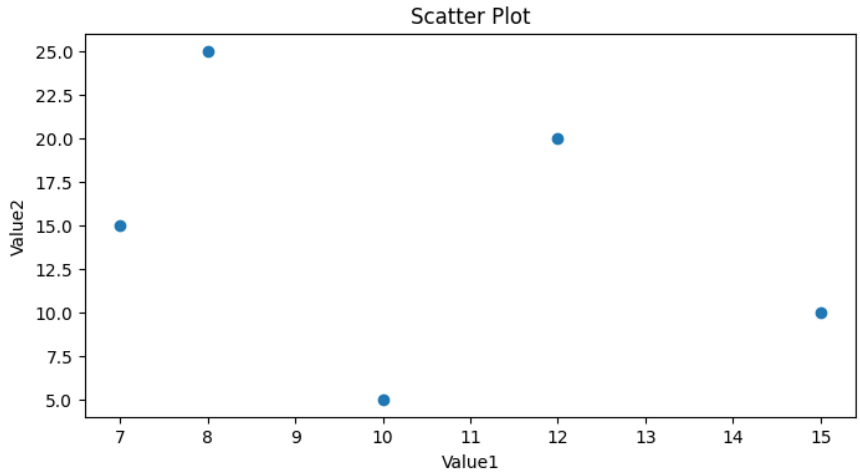

BUBBLE CHART

```
# Create a Bubble Chart
plt.figure(figsize=(8, 4))
plt.scatter(df['Value1'], df['Value2'], s=df['Value2'] * 10, alpha=0.5)
plt.xlabel('Value1')
plt.ylabel('Value2')
plt.title('Bubble Chart')
plt.show()
```



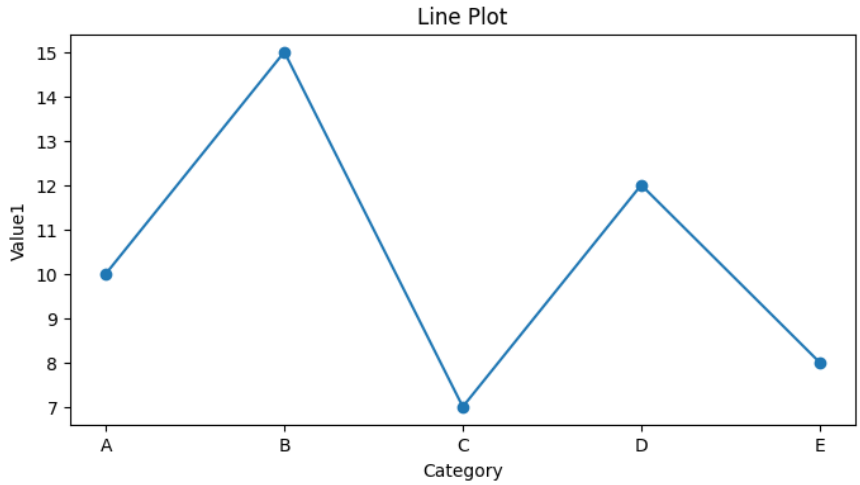

SCATTER PLOT

```
# Create a Scatter Plot
plt.figure(figsize=(8, 4))
plt.scatter(df['Value1'], df['Value2'])
plt.xlabel('Value1')
plt.ylabel('Value2')
plt.title('Scatter Plot')
plt.show()
```



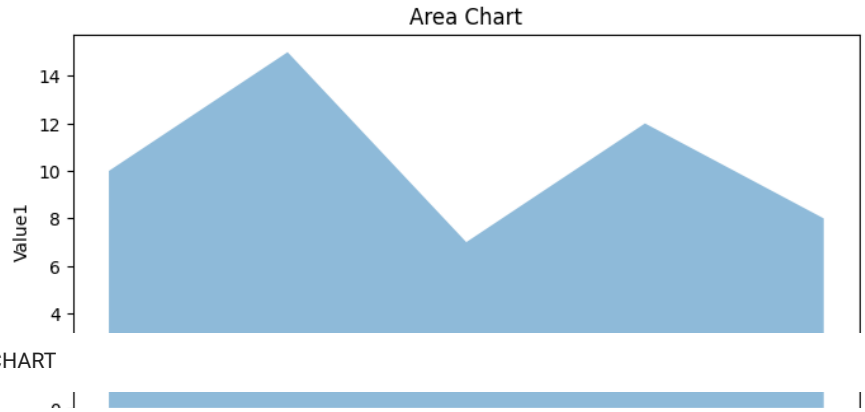

LINE PLOT

```
# Create a Line Plot
plt.figure(figsize=(8, 4))
plt.plot(df['Category'], df['Value1'], marker='o')
plt.xlabel('Category')
plt.ylabel('Value1')
plt.title('Line Plot')
plt.show()
```



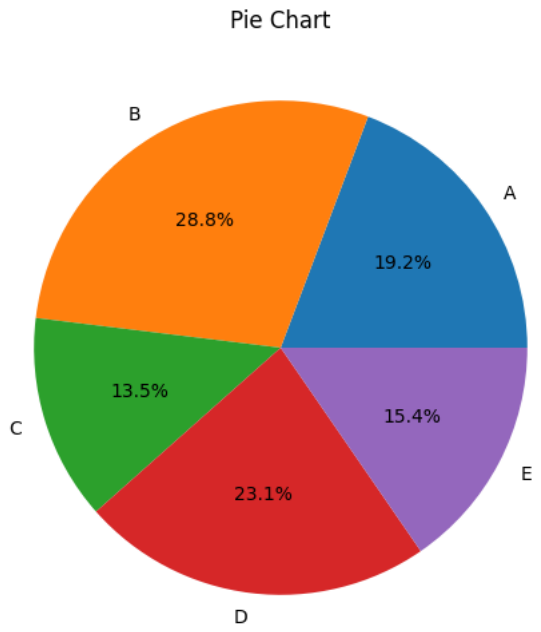

AREA CHART

```
# Create an Area Chart
plt.figure(figsize=(8, 4))
plt.fill_between(df['Category'], df['Value1'], alpha=0.5)
plt.xlabel('Category')
plt.ylabel('Value1')
plt.title('Area Chart')
plt.show()
```




PIE CHART

```
# Create a Pie Chart
plt.figure(figsize=(6, 6))
plt.pie(df['Value1'], labels=df['Category'], autopct='%1.1f%%')
plt.title('Pie Chart')
plt.show()
```




HISTOGRAM

```
# Create a Histogram
plt.figure(figsize=(8, 4))
plt.hist(df['Value1'], bins=5, edgecolor='black')
plt.xlabel('Value1')
plt.ylabel('Frequency')
plt.title('Histogram')
plt.show()
```


HEATMAP

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Sample DataFrame
data = {
    'Category': ['A', 'B', 'C', 'D', 'E'],
    'Value1': [10, 15, 7, 12, 8],
    'Value2': [5, 10, 15, 20, 25]
}

df = pd.DataFrame(data)

# Create a heatmap
values = df.pivot(index='Category', columns='Value1', values='Value2')

plt.figure(figsize=(8, 6))
plt.imshow(values, cmap='YlOrRd', aspect='auto', origin='lower')
plt.colorbar(label='Value2')
plt.xticks(np.arange(len(df['Value1'])), df['Value1'])
plt.yticks(np.arange(len(df['Category'])), df['Category'])
plt.xlabel('Value1')
plt.ylabel('Category')
plt.title('Heatmap')
plt.show()
```

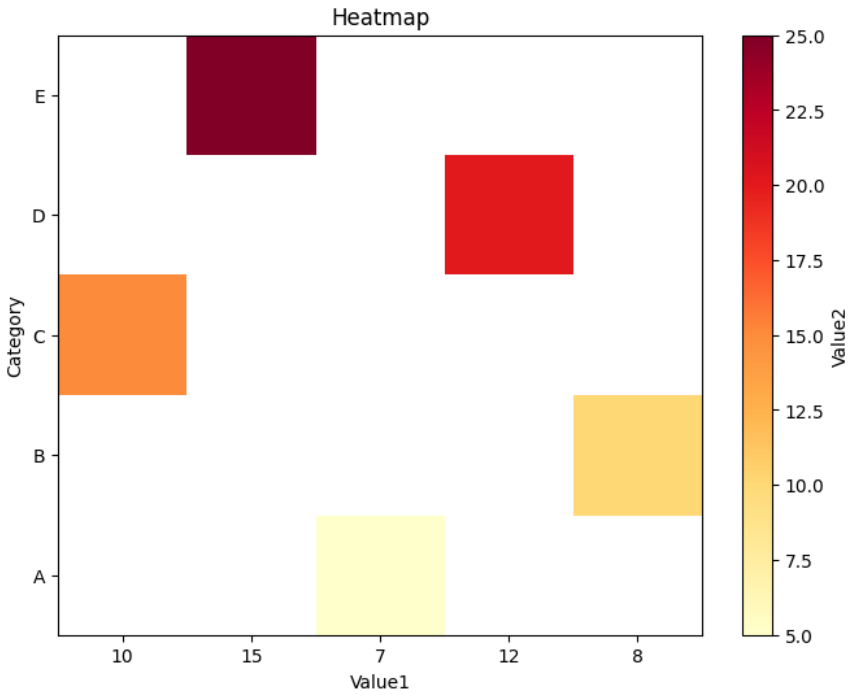



TABLE CHART

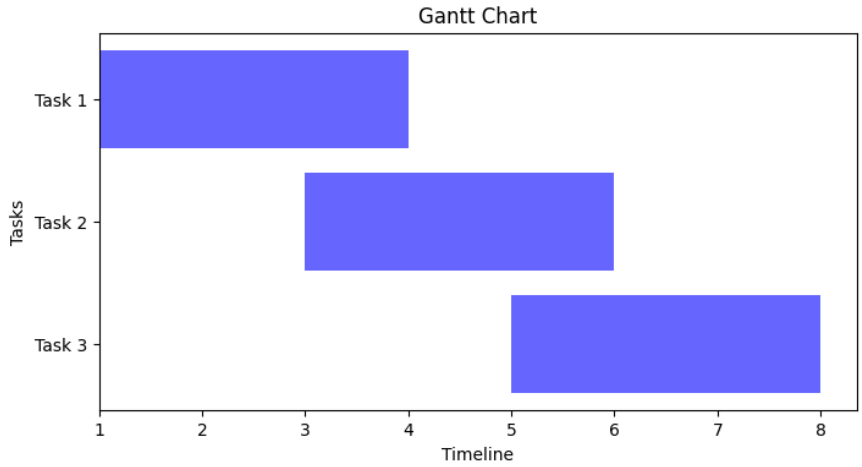
```
# Create a Table Chart
fig, ax = plt.subplots(figsize=(8, 4))
ax.axis('tight')
ax.axis('off')
ax.table(cellText=df.values, colLabels=df.columns, cellLoc='center', loc='center')
plt.title('Table Chart')
plt.show()
```

Table Chart

| Category | Value1 | Value2 |
|---|---|---|
| A | 10 | 5 |
| B | 15 | 10 |
| C | 7 | 15 |

GANTT CHART

```
import matplotlib.pyplot as plt

# Sample Gantt chart data
tasks = ['Task 1', 'Task 2', 'Task 3']
start_dates = [1, 3, 5]
end_dates = [4, 6, 8]

# Create a Gantt chart
fig, ax = plt.subplots(figsize=(8, 4))

for i, task in enumerate(tasks):
    ax.barh(task, end_dates[i] - start_dates[i], left=start_dates[i], color='b', alpha=0.6)

ax.set_xlabel('Timeline')
ax.set_ylabel('Tasks')
ax.set_title('Gantt Chart')
ax.invert_yaxis()
plt.show()
```




TREEMAP

```
import matplotlib.pyplot as plt

# Sample Treemap data
categories = ['Root', 'A', 'B', 'C']
values = [100, 40, 30, 30]

# Create a Treemap
fig, ax = plt.subplots(figsize=(6, 6))
ax.axis('off')

# Calculate rectangle sizes
total = sum(values)
sizes = [(value / total) for value in values]

# Define the position and size of rectangles
rectangles = [(0, 0, 1, 1)]
for i in range(1, len(categories)):
    x, y, w, h = rectangles[i - 1]
    if w > h:
        w1 = w * sizes[i]
        h1 = h
        x1 = x + w - w1
        y1 = y
```

```
            w2 = w - w1
            h2 = h
            x2 = x
            y2 = y
        else:
            w1 = w
            h1 = h * sizes[i]
            x1 = x
            y1 = y + h - h1
            w2 = w
            h2 = h - h1
            x2 = x
            y2 = y
        rectangles.extend([(x1, y1, w1, h1), (x2, y2, w2, h2)])

    colors = ['skyblue', 'lightgreen', 'lightcoral', 'lightsalmon']

    for i, category in enumerate(categories):
        x, y, w, h = rectangles[i]
        ax.add_patch(plt.Rectangle((x, y), w, h, fill=True, color=colors[i]))
        ax.text(x + w / 2, y + h / 2, category, ha='center', va='center')

    plt.title('Treemap')
    plt.show()
```