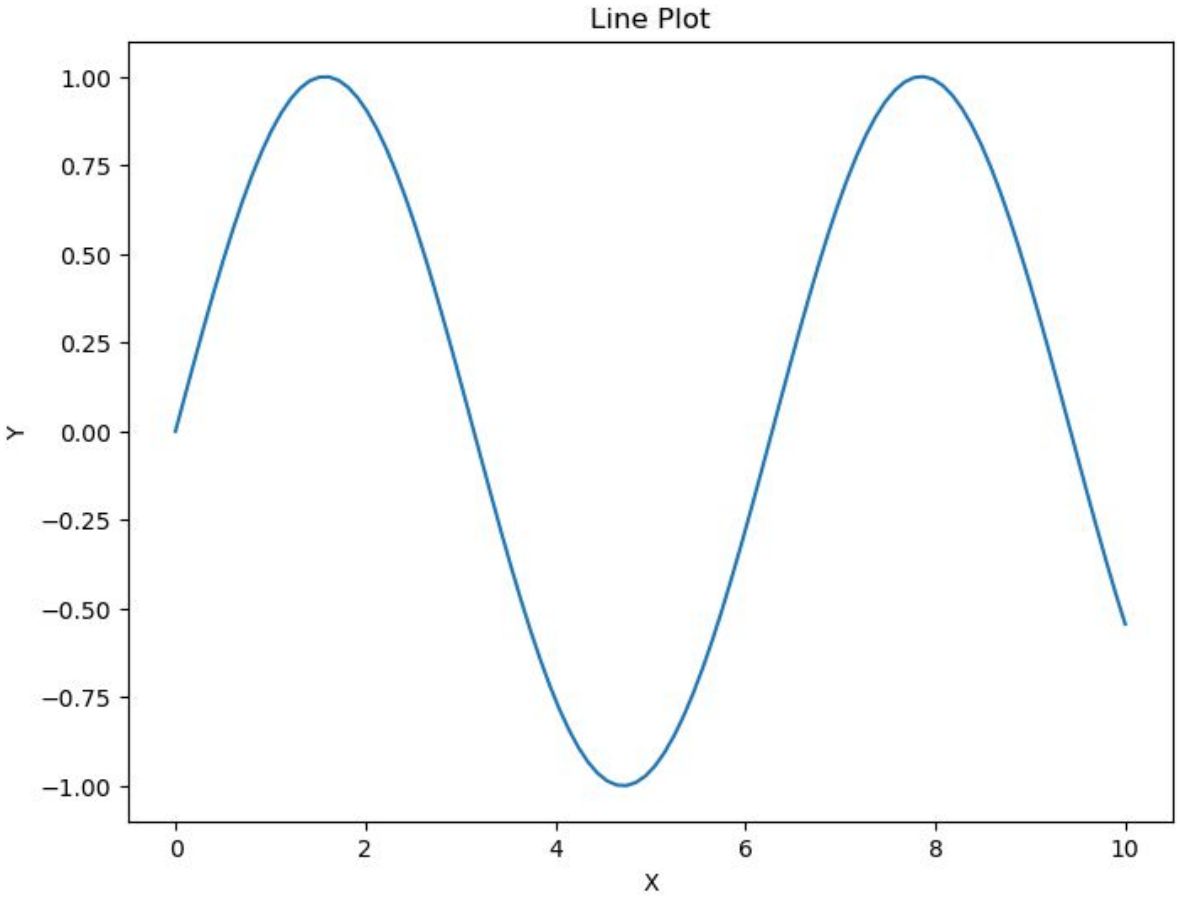
1. Line Plot:

Create a simple line plot using Matplotlib.

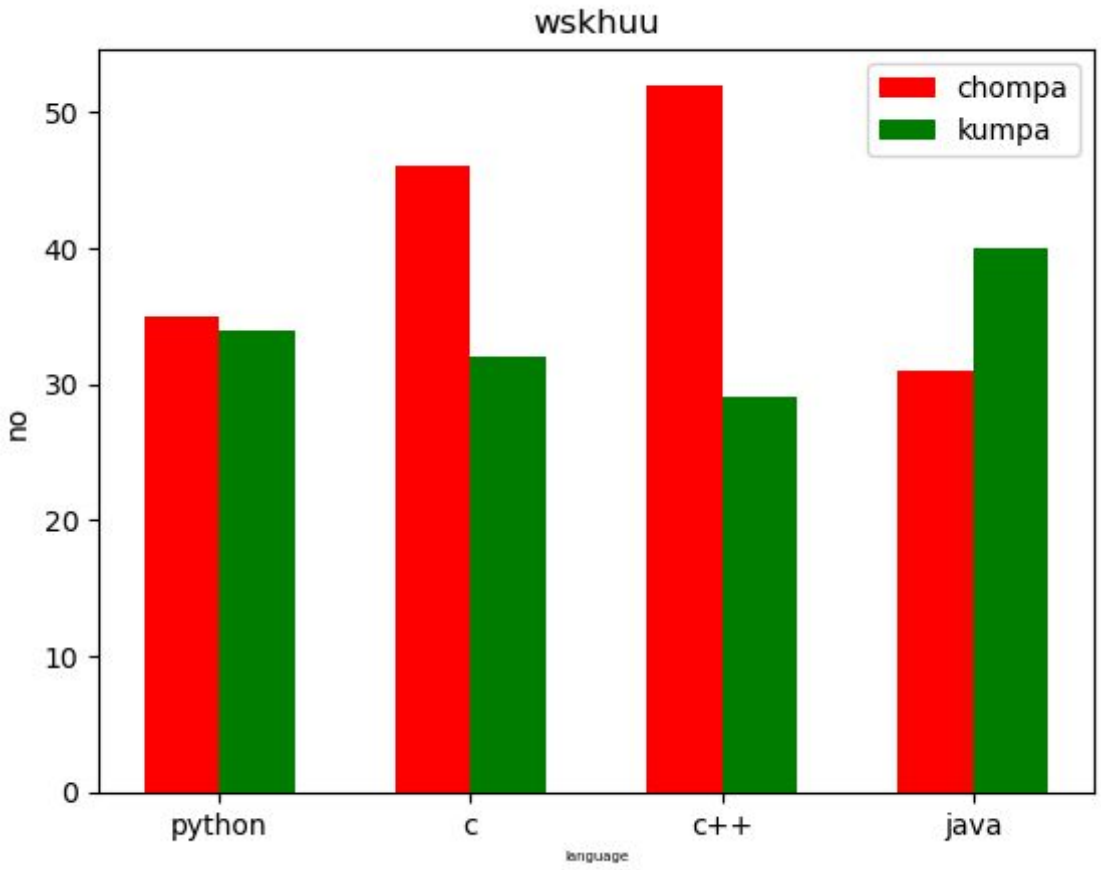
In [2]:
```
import matplotlib.pyplot as plt
import numpy as np

# Generate some sample data
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Create the line plot
plt.figure(figsize=(8, 6))
plt.plot(x, y)
plt.title('Line Plot')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



2. BAR PLOT

Create a bar plot to compare categories.

In [3]:
```
import matplotlib.pyplot as plt
import numpy as np

x=["python","c","c++","java"]
y=[35,46,52,31]
z=[34,32,29,40]
width=0.3
p=np.arange(len(x))
p1=[j+width for j in p]

plt.xlabel("language",fontsize=5)
plt.ylabel("no")
plt.title("wskhuu")

plt.bar(p,y,width,color="r",label="chompa")
plt.bar(p1,z,width,color="g",label="kumpa")

plt.xticks(p+width/2,x)
plt.legend()
plt.show()
```

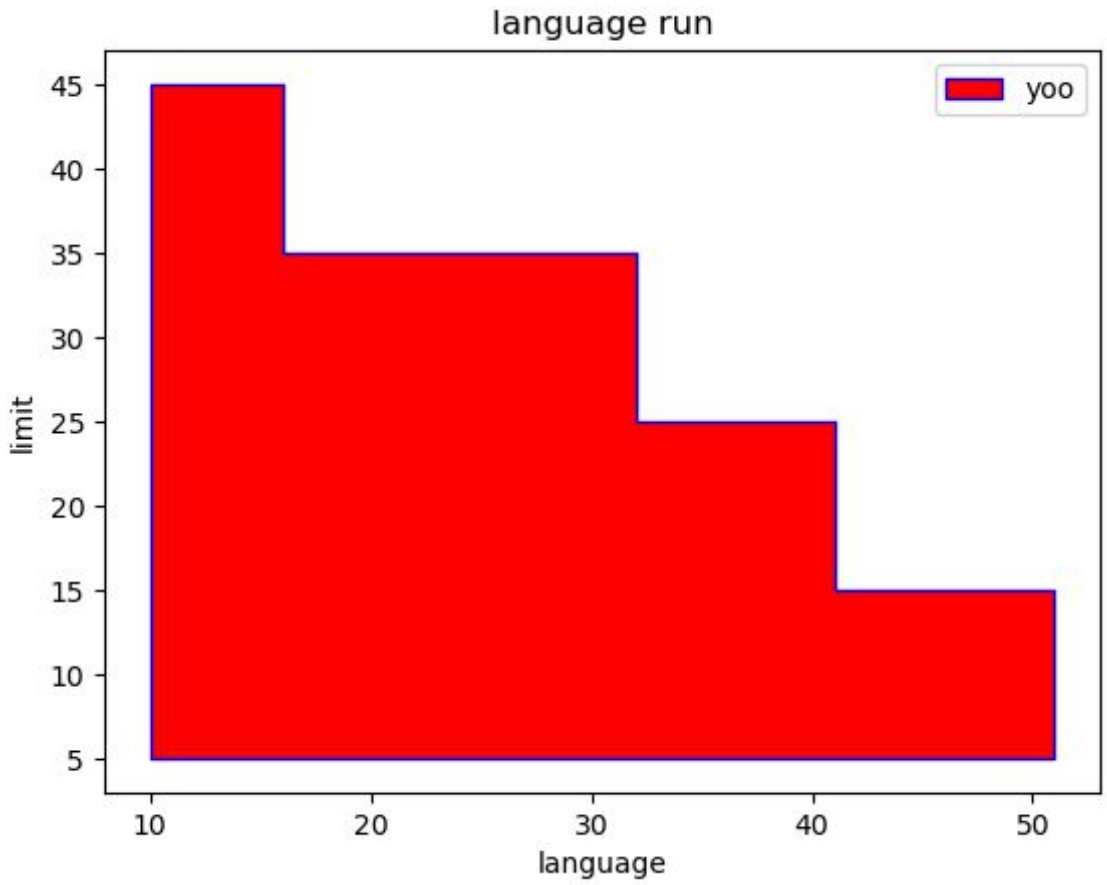


3. histogram

Create a histogram to visualize the distribution of a dataset.

In [4]:
```
no=[36, 25, 11, 19, 32, 52, 51, 21, 58, 31, 27, 36, 14, 21, 32, 30, 33, 32, 33, 15,
 15, 32, 30, 13, 25, 19, 46, 59, 43,55, 19, 28, 32, 20, 34, 51, 52, 37, 46, 17, 47,

l=[10,20,30,40,50]
```

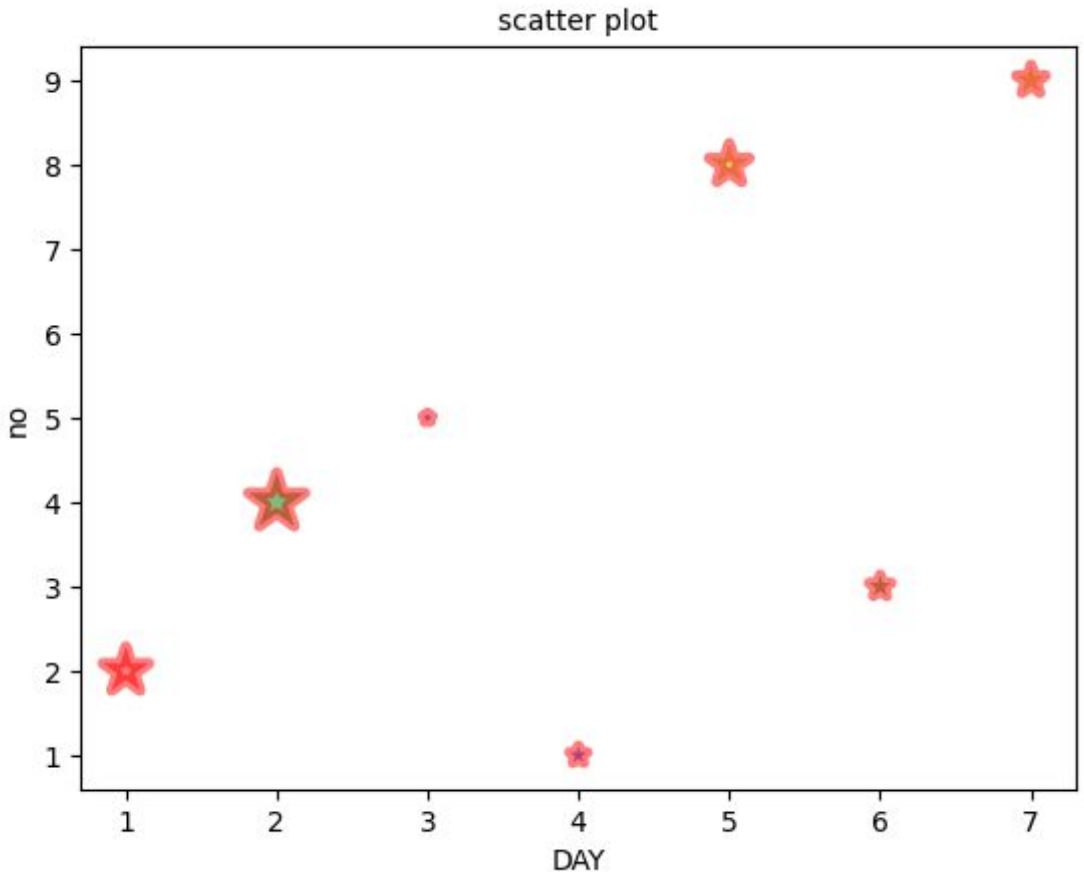
```
plt.hist(no,color='red',edgecolor='blue',bins=1,cumulative=-1,bottom=10,align="left
plt.xlabel('language')
plt.ylabel('limit')
plt.title('language run')
plt.legend()
plt.show()
```



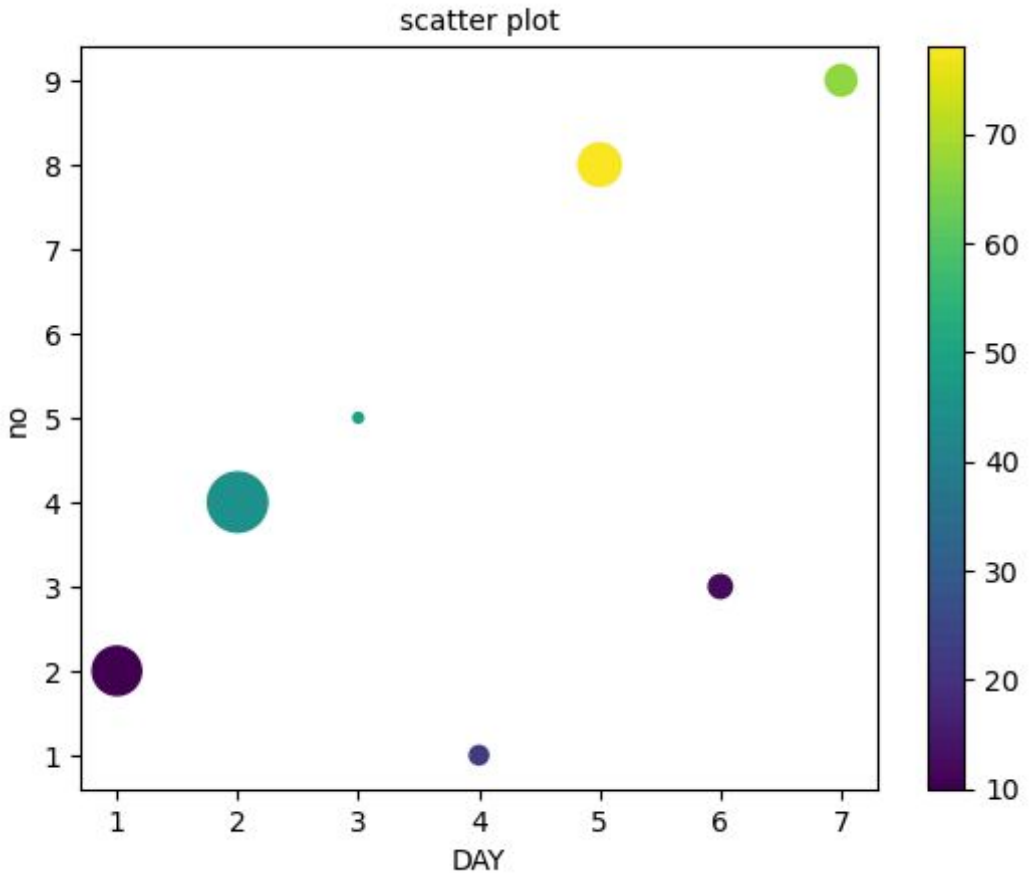
4. SCATTER PLOT

Create a scatter plot to show the relationship between two variables.

In [13]:
```
DAY=[1,2,3,4,5,6,7]
no=[2,4,5,1,8,3,9]
colors=["r","g","b","b","y","g","y"]
sizes=[300,450,12,43,230,68,120]
plt.scatter(DAY,no,c=colors,s=sizes,alpha=0.5,marker='*',edgecolor='r',linewidth=4)
plt.title("scatter plot",fontsize=10)
plt.xlabel("DAY",fontsize=10)
plt.ylabel("no",fontsize=10)
plt.show()
```



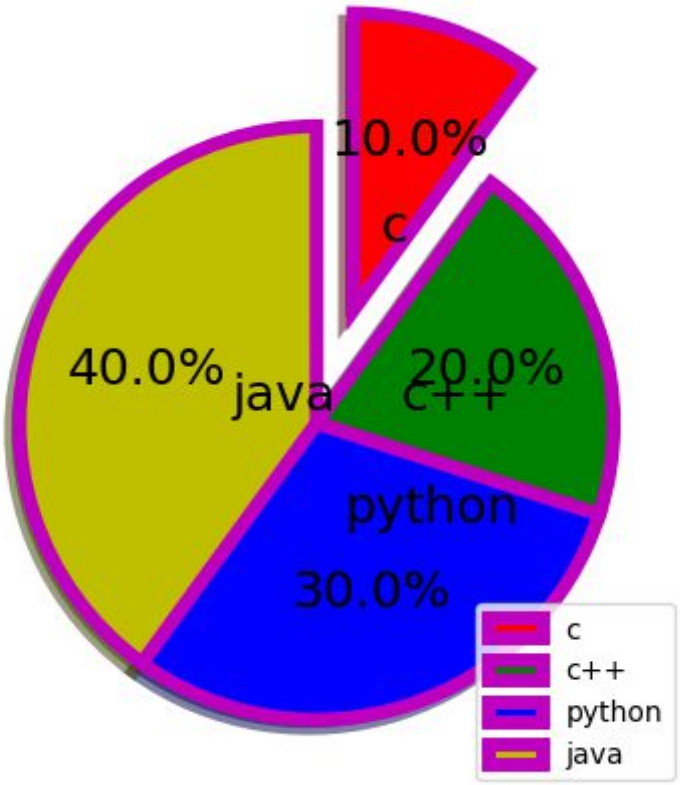
In [14]:
```
DAY=[1,2,3,4,5,6,7]
no=[2,4,5,1,8,3,9]
colors=[10,45,49,23,78,12,67]
sizes=[300,450,12,43,230,68,120]
plt.scatter(DAY,no,c=colors,s=sizes,cmap='viridis')
plt.title("scatter plot",fontsize=10)
plt.xlabel("DAY",fontsize=10)
plt.ylabel("no",fontsize=10)
plt.colorbar()
plt.show()
```



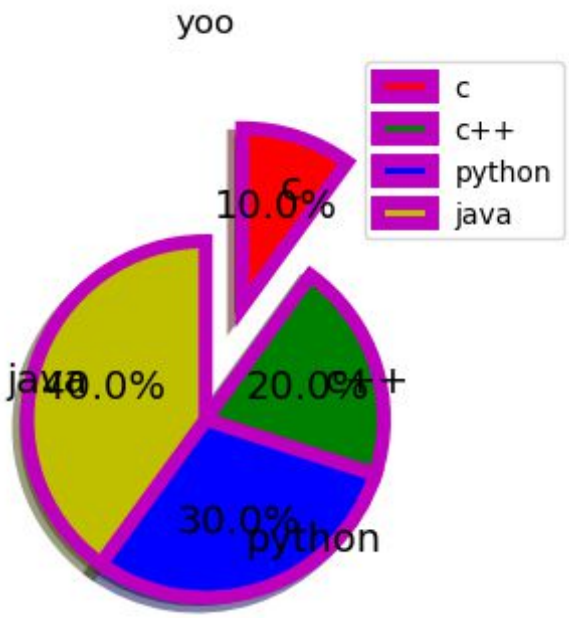
5. pie chart

Create a pie chart to show the proportions of different categories.

In [9]:
```
x=[10,20,30,40]
y=["c","c++","python","java"]
ex=[0.4,0.0,0.0,0.0]
c=["r","g","b","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=1.0,lab
        textprops={"fontsize":18},counterclock=False,wedgeprops={"linewidth":5,"edg
plt.legend(loc=4)
plt.show()
```



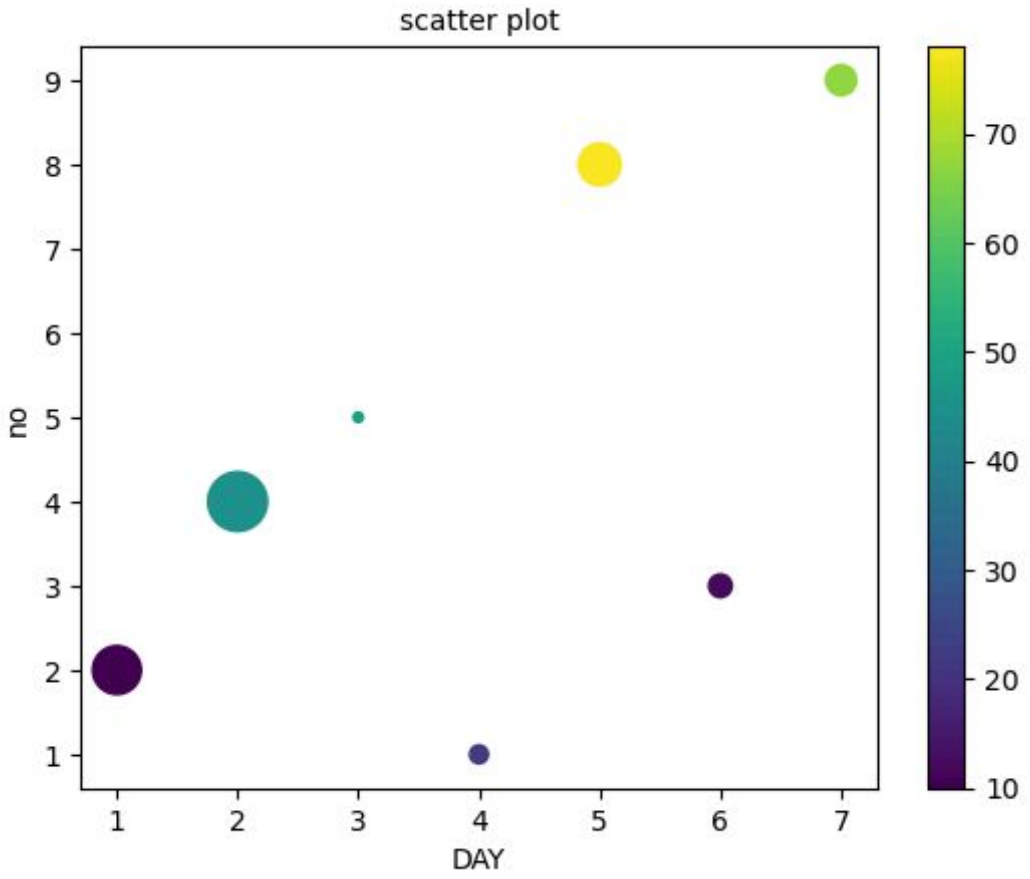
In [10]:
```
x=[10,20,30,40]
y=["c","c++","python","java"]
ex=[0.4,0.0,0.0,0.0]
c=["r","g","b","y"]

plt.pie(x,labels=y,explode=ex,colors=c,autopct="%0.1f%%",shadow=True,radius=0.6,lab
        textprops={"fontsize":14},counterclock=False,wedgeprops={"linewidth":5,"edg
plt.legend()
plt.title("yoo")
plt.show()
```



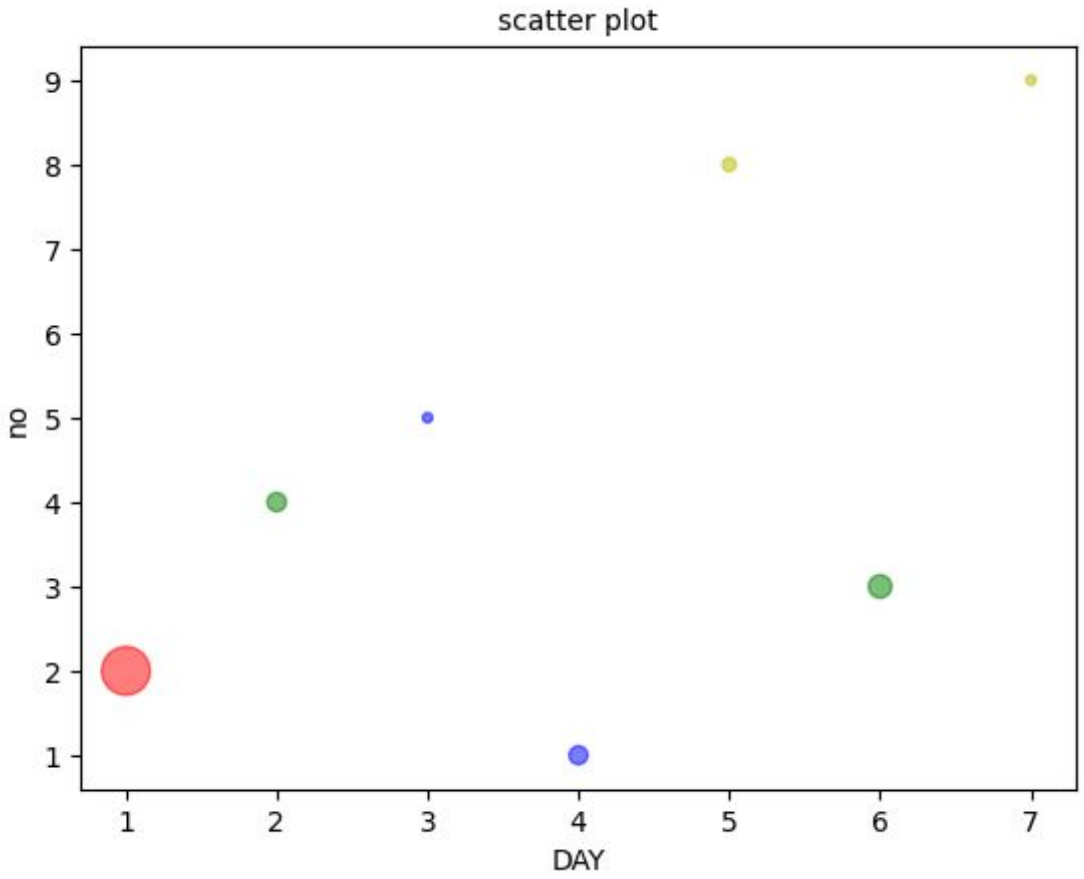
box plot

6. Create a box plot to show the distribution of a dataset.

In [7]:
```
DAY=[1,2,3,4,5,6,7]
no=[2,4,5,1,8,3,9]
colors=[10,45,49,23,78,12,67]
sizes=[300,450,12,43,230,68,120]
plt.scatter(DAY,no,c=colors,s=sizes,cmap='viridis')
plt.title("scatter plot",fontsize=10)
plt.xlabel("DAY",fontsize=10)
plt.ylabel("no",fontsize=10)
plt.colorbar()
plt.show()
```



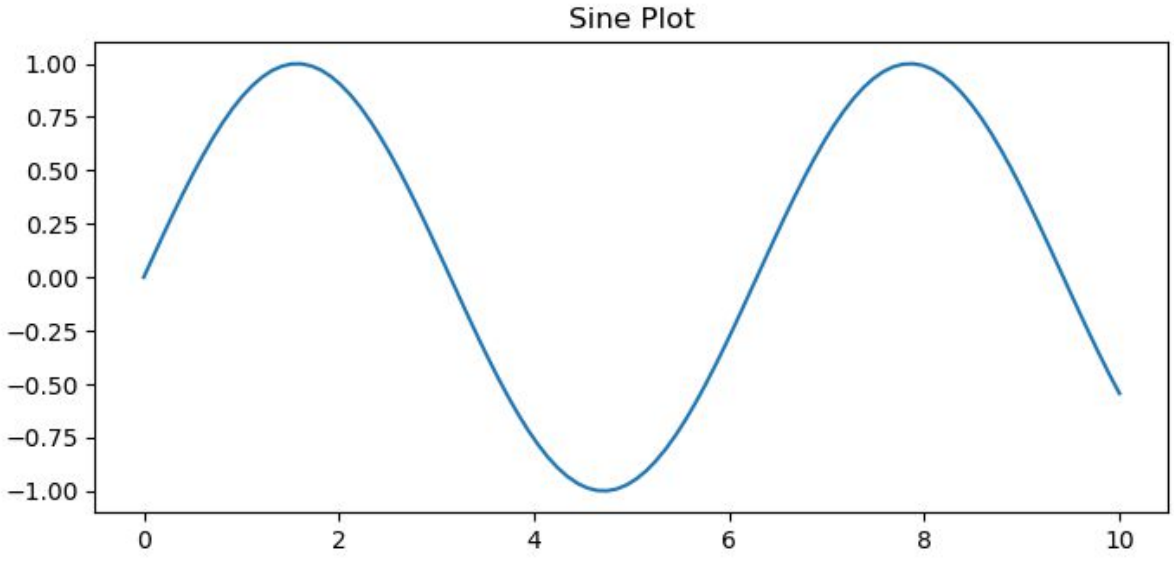
In [8]:
```
import matplotlib.pyplot as plt
DAY=[1,2,3,4,5,6,7]
no=[2,4,5,1,8,3,9]
colors=["r","g","b","b","y","g","y"]
sizes=[300,45,12,43,23,68,12]
plt.scatter(DAY,no,c=colors,s=sizes,alpha=0.5)
plt.title("scatter plot",fontsize=10)
plt.xlabel("DAY",fontsize=10)
plt.ylabel("no",fontsize=10)
plt.show()
```

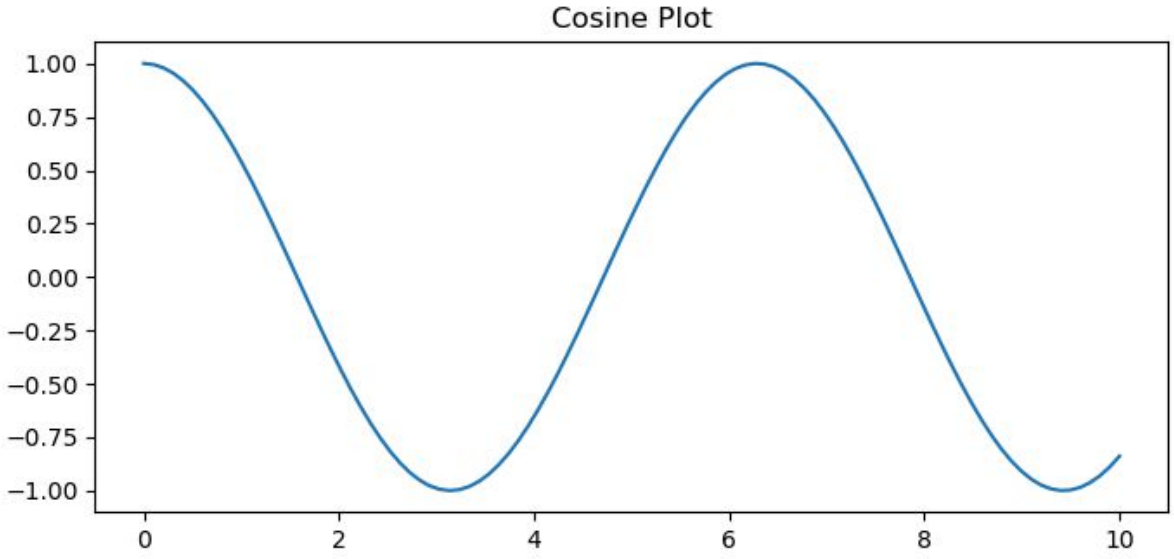


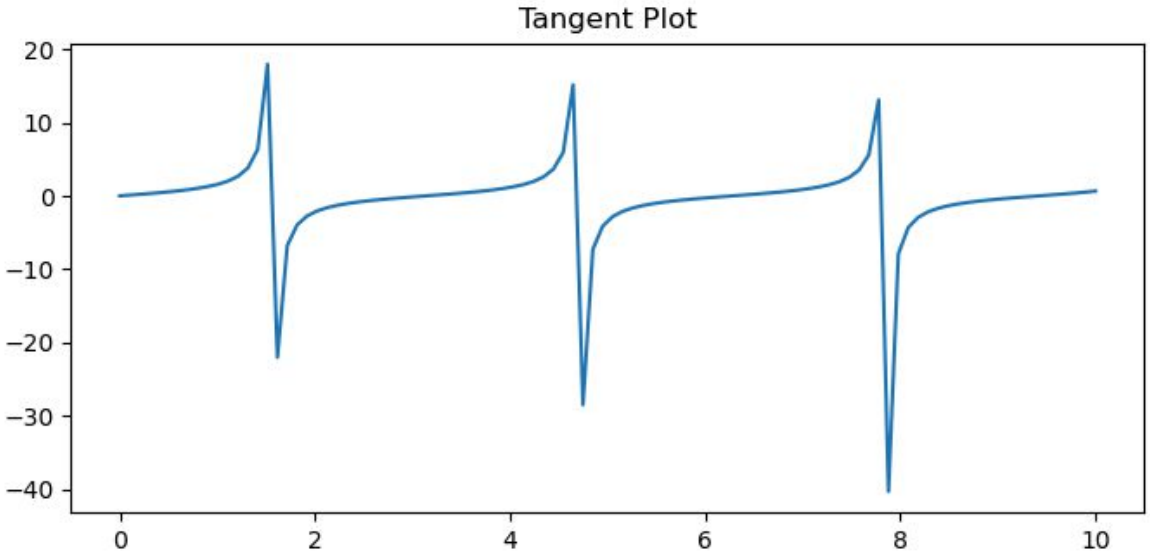
7. subplot

Create multiple subplots in a single figure.

In [11]:
```
import matplotlib.pyplot as plt
import numpy as np


x = np.linspace(0, 10, 100)
y1 = np.sin(x)
y2 = np.cos(x)
y3 = np.tan(x)

# Create the subplots
fig, (ax1, ax2, ax3) = plt.subplots(3, 1, figsize=(8, 12))

ax1.plot(x, y1)
ax1.set_title('Sine Plot')
ax2.plot(x, y2)
ax2.set_title('Cosine Plot')
ax3.plot(x, y3)
ax3.set_title('Tangent Plot')

plt.show()
```



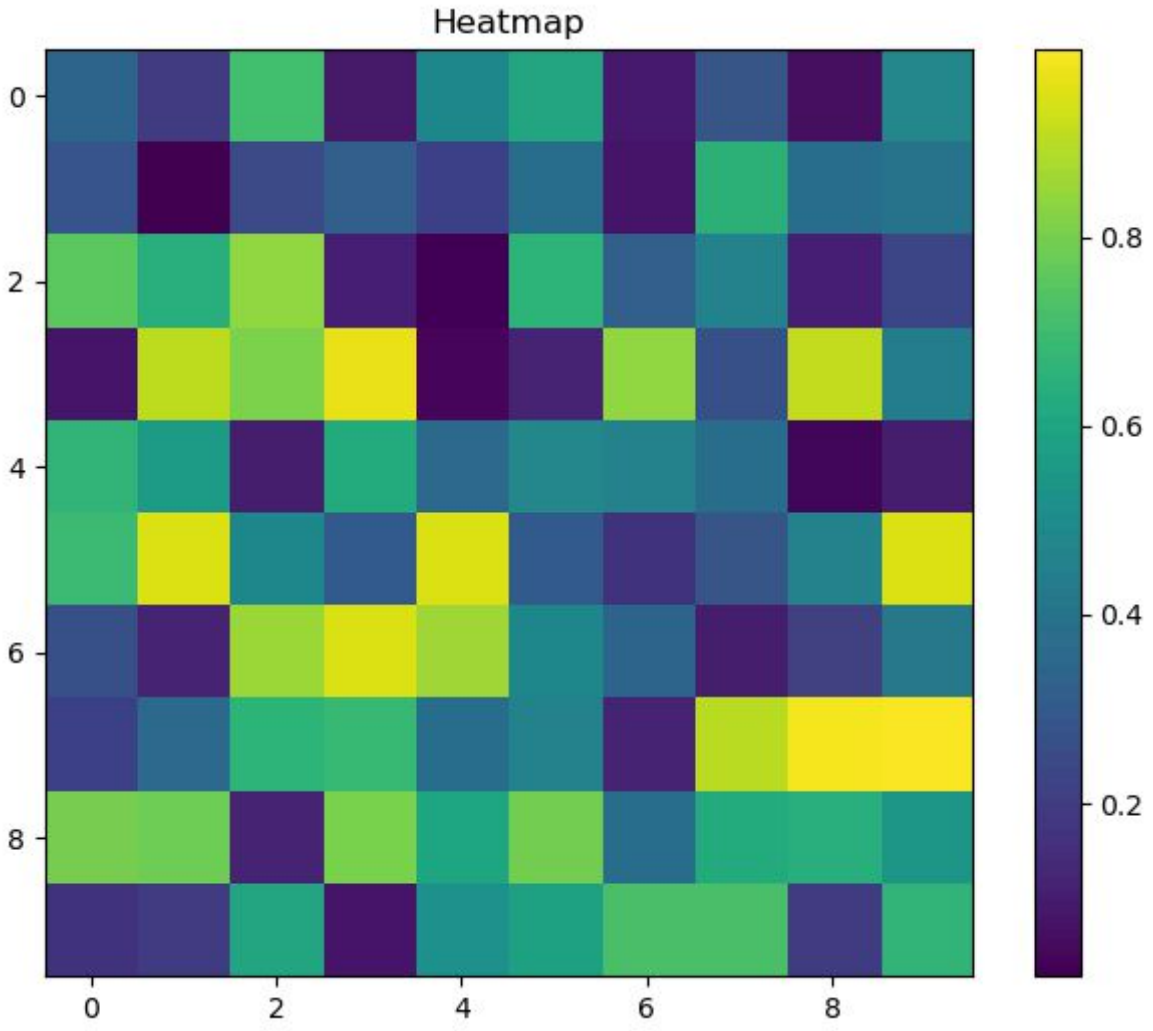




8. HEATMAP

Create a heatmap to show correlations between variables.

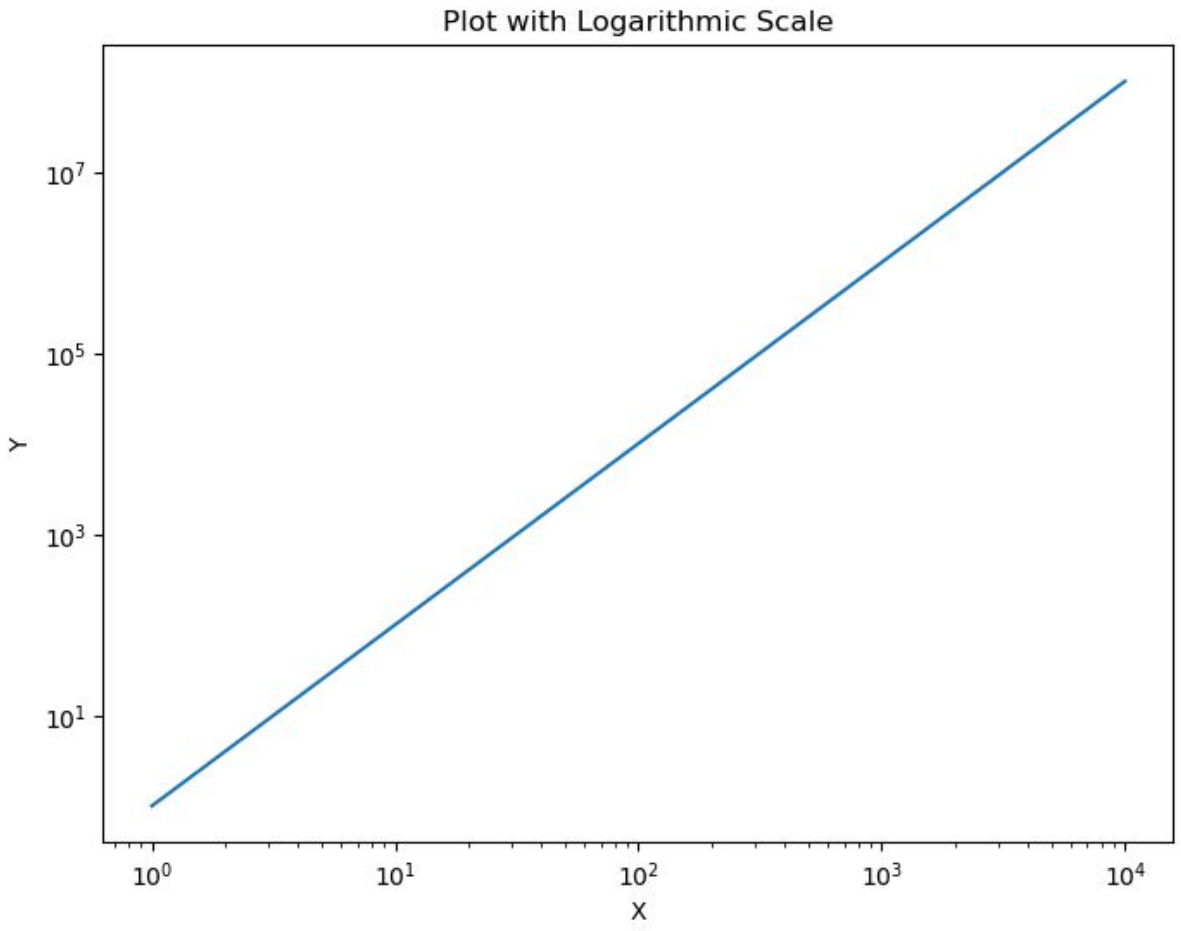
In [12]:
```
import matplotlib.pyplot as plt
import numpy as np


data = np.random.rand(10, 10)


plt.figure(figsize=(8, 6))
plt.imshow(data, cmap='viridis')
plt.colorbar()
plt.title('Heatmap')
plt.show()
```



Create a plot with a logarithmic scale.

In [15]:
```
import matplotlib.pyplot as plt
import numpy as np

x = np.logspace(0, 4, 100)
y = x ** 2


plt.figure(figsize=(8, 6))
plt.plot(x, y)
plt.xscale('log')
plt.yscale('log')
plt.title('Plot with Logarithmic Scale')
plt.xlabel('X')
```

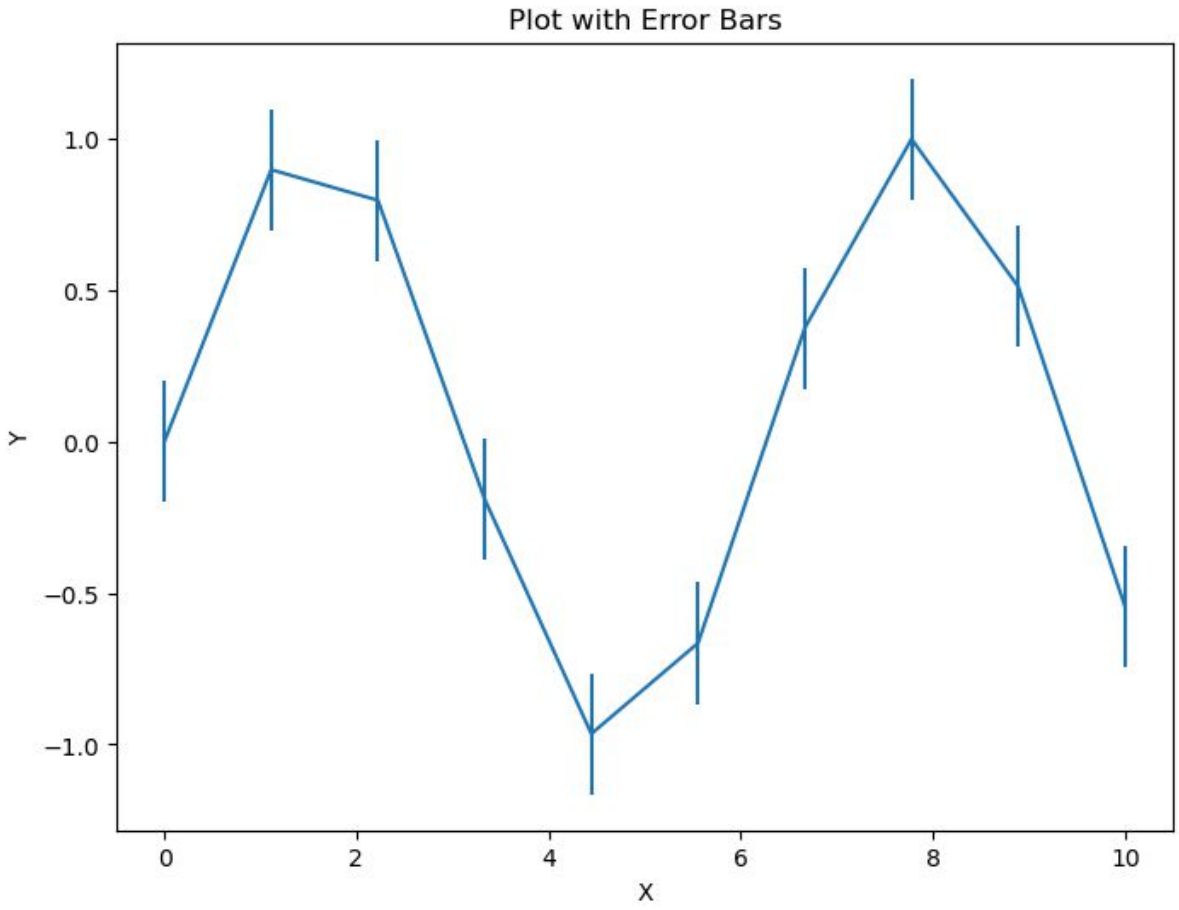
```
plt.ylabel('Y')
plt.show()
```



Error Bars:

Add error bars to a plot.

In [16]:
```
import matplotlib.pyplot as plt
import numpy as np


x = np.linspace(0, 10, 10)
y = np.sin(x)
yerr = 0.2 * np.ones_like(y)

plt.figure(figsize=(8, 6))
plt.errorbar(x, y, yerr=yerr)
plt.title('Plot with Error Bars')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

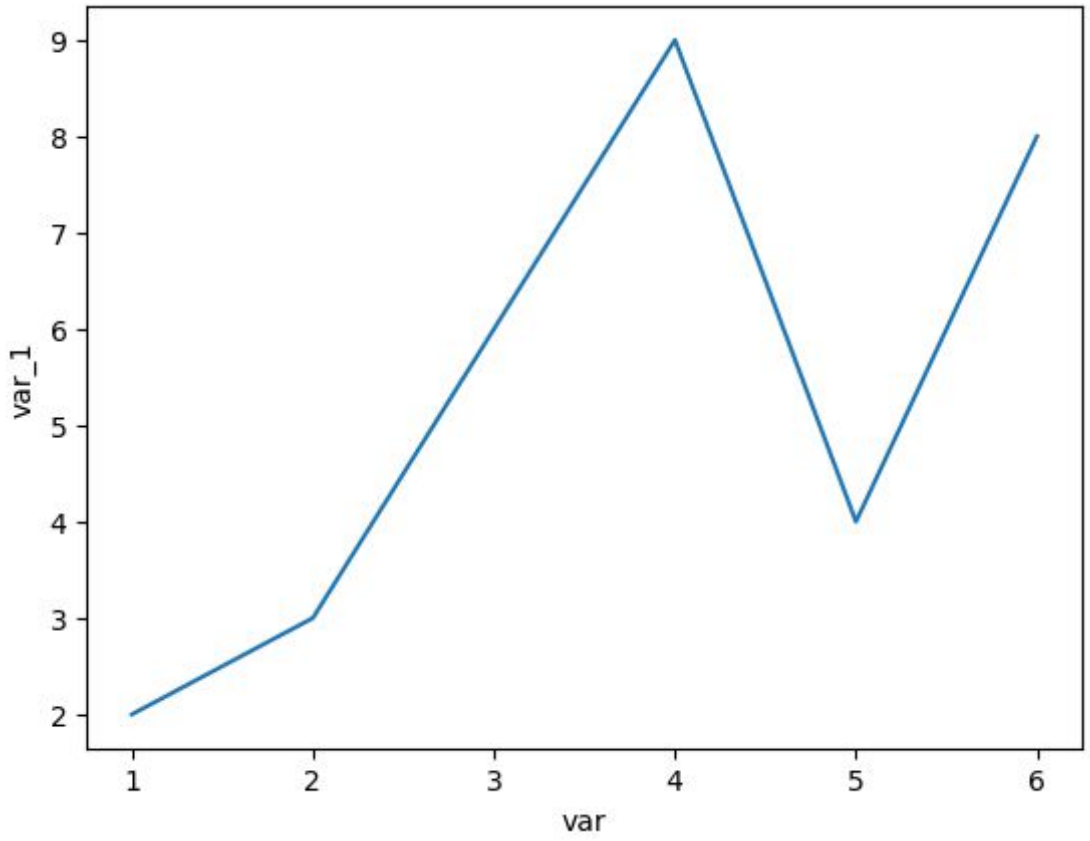


SEABORN

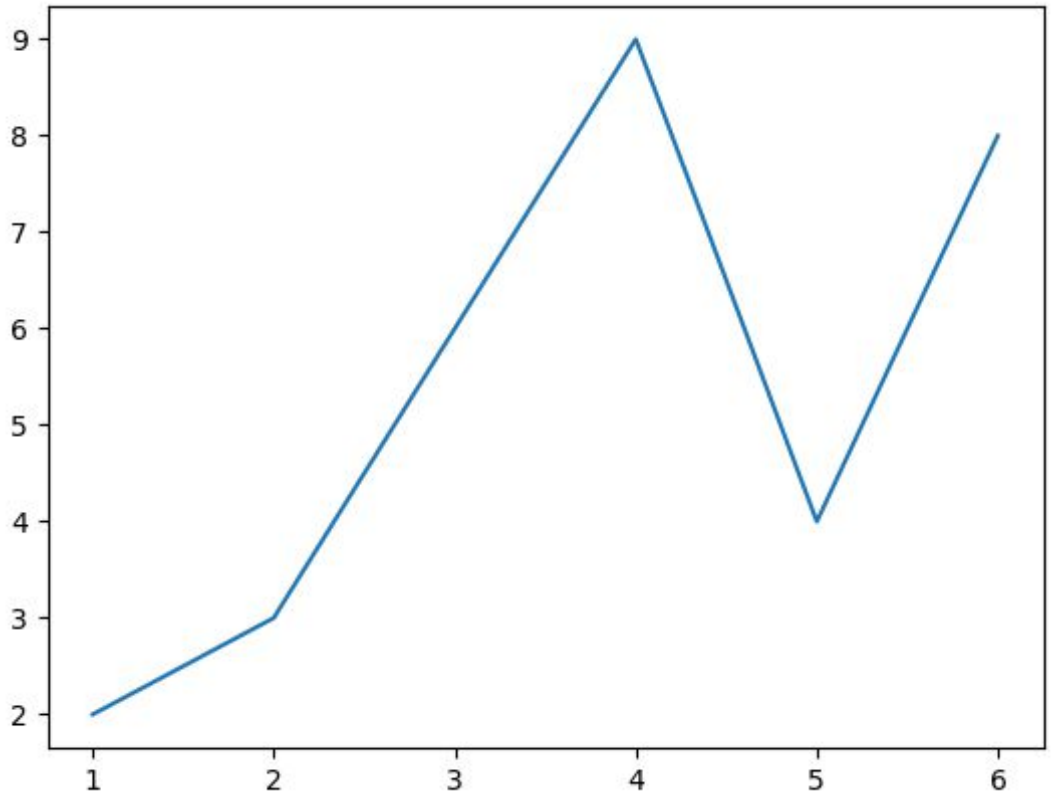
Create a line plot using Seaborn.

In [2]:
```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [3]:
```
var=[1,2,3,4,5,6]
var_1=[2,3,6,9,4,8]

x_1=pd.DataFrame({"var": var, "var_1":var_1})

sns.lineplot(x="var",y="var_1",data= x_1)
plt.show()
```

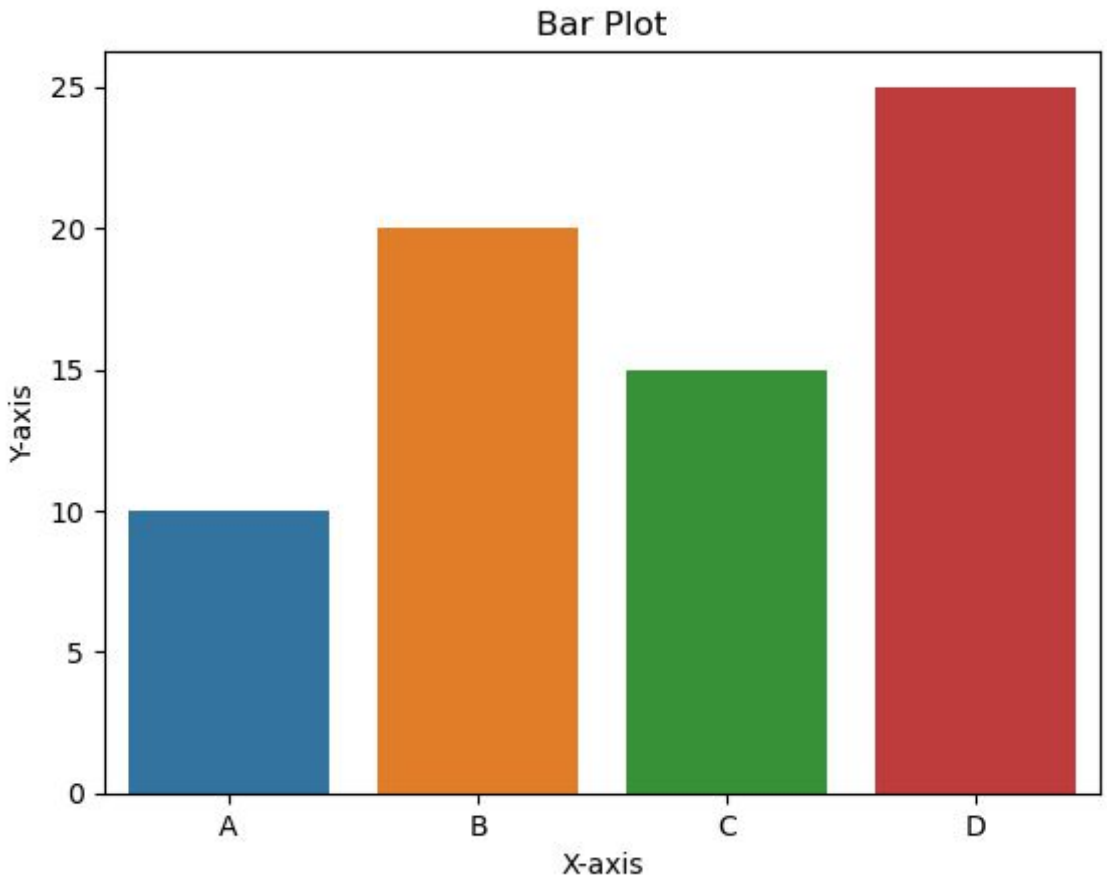


In [4]:
```
var=[1,2,3,4,5,6]
var_1=[2,3,6,9,4,8]


sns.lineplot(x=var,y=var_1)
plt.show()
```



Create a bar plot using Seaborn.

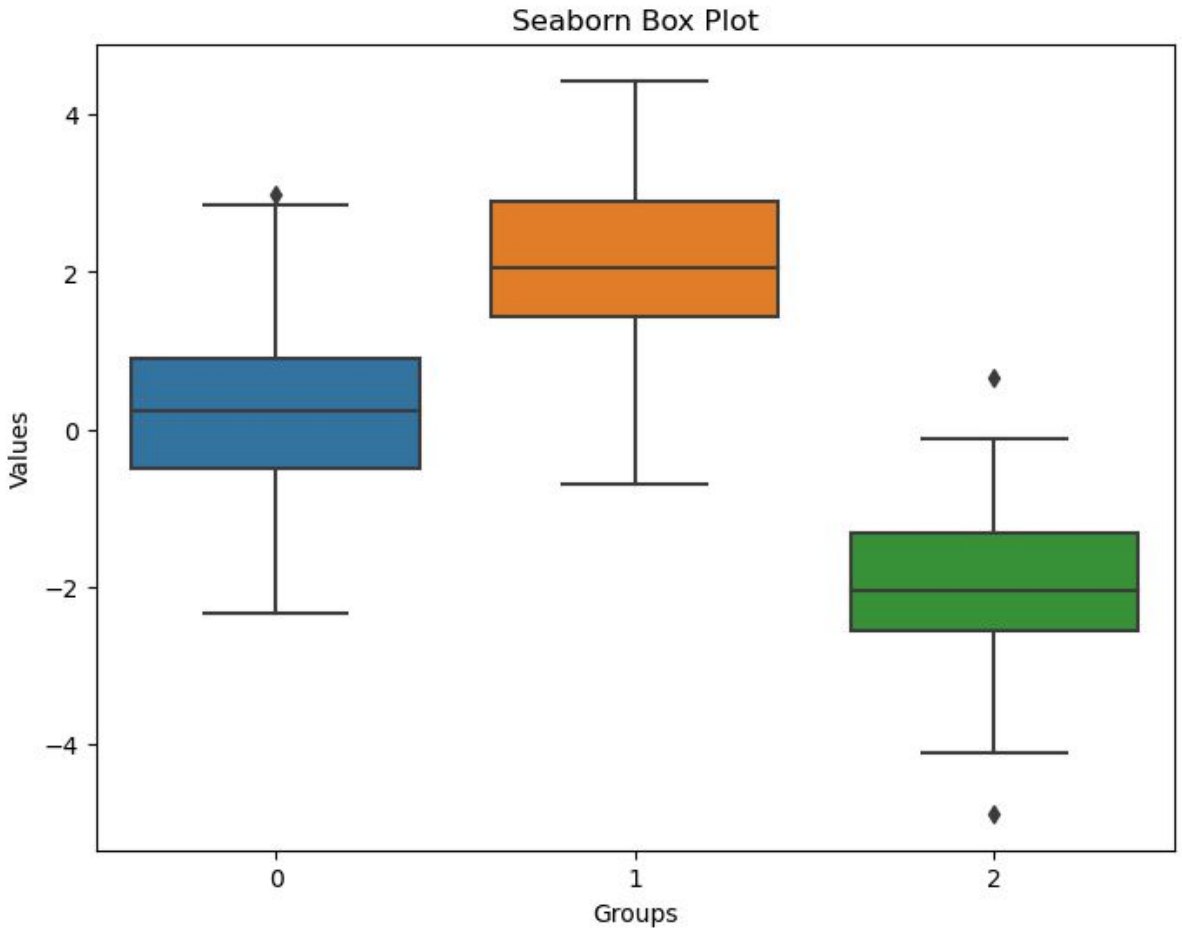
In [5]:
```
import seaborn as sns
import matplotlib.pyplot as plt


x = ['A', 'B', 'C', 'D']
y = [10, 20, 15, 25]


sns.barplot(x=x, y=y)


plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Bar Plot')


plt.show()
```



Create a box plot

In [6]:
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np


data1 = np.random.normal(0, 1, 100)
data2 = np.random.normal(2, 1, 100)
data3 = np.random.normal(-2, 1, 100)
```

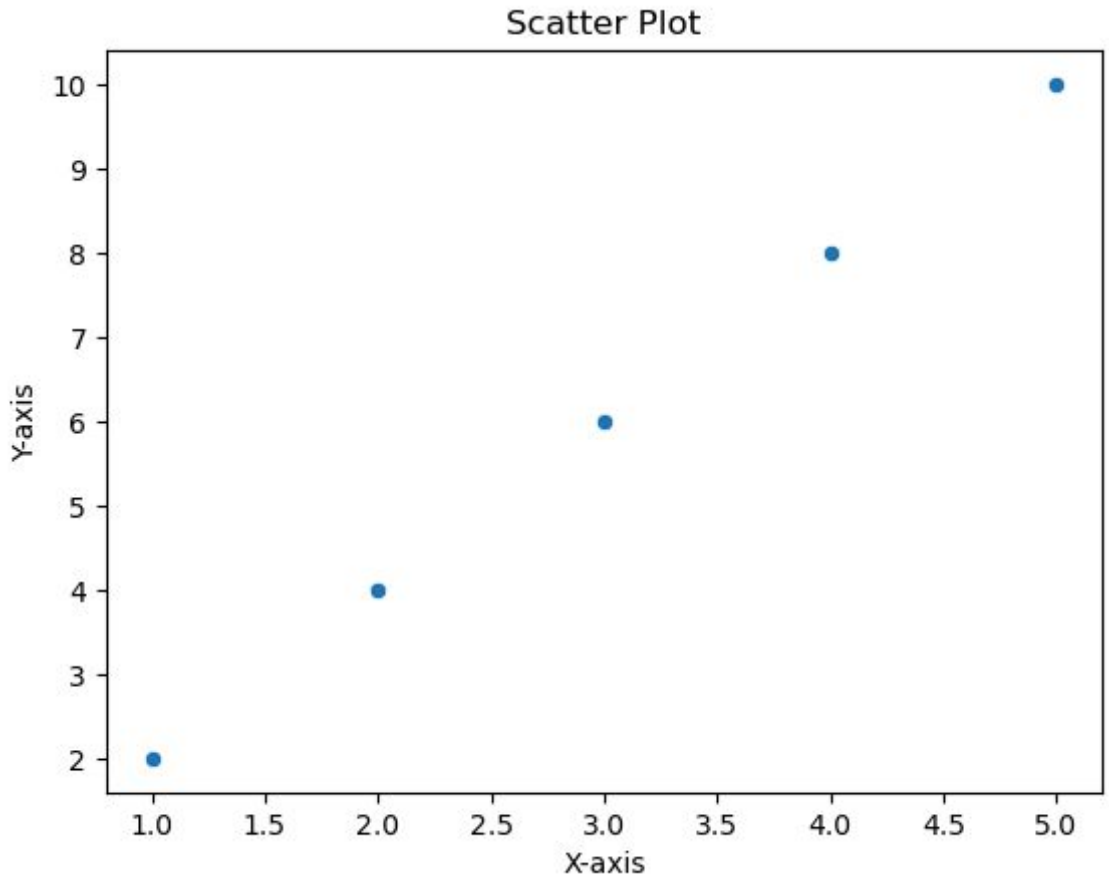
```
plt.figure(figsize=(8, 6))
sns.boxplot(data=[data1, data2, data3])
plt.title('Seaborn Box Plot')
plt.xlabel('Groups')
plt.ylabel('Values')
plt.show()
```



Create a scatter plot using Seaborn.

In [8]:

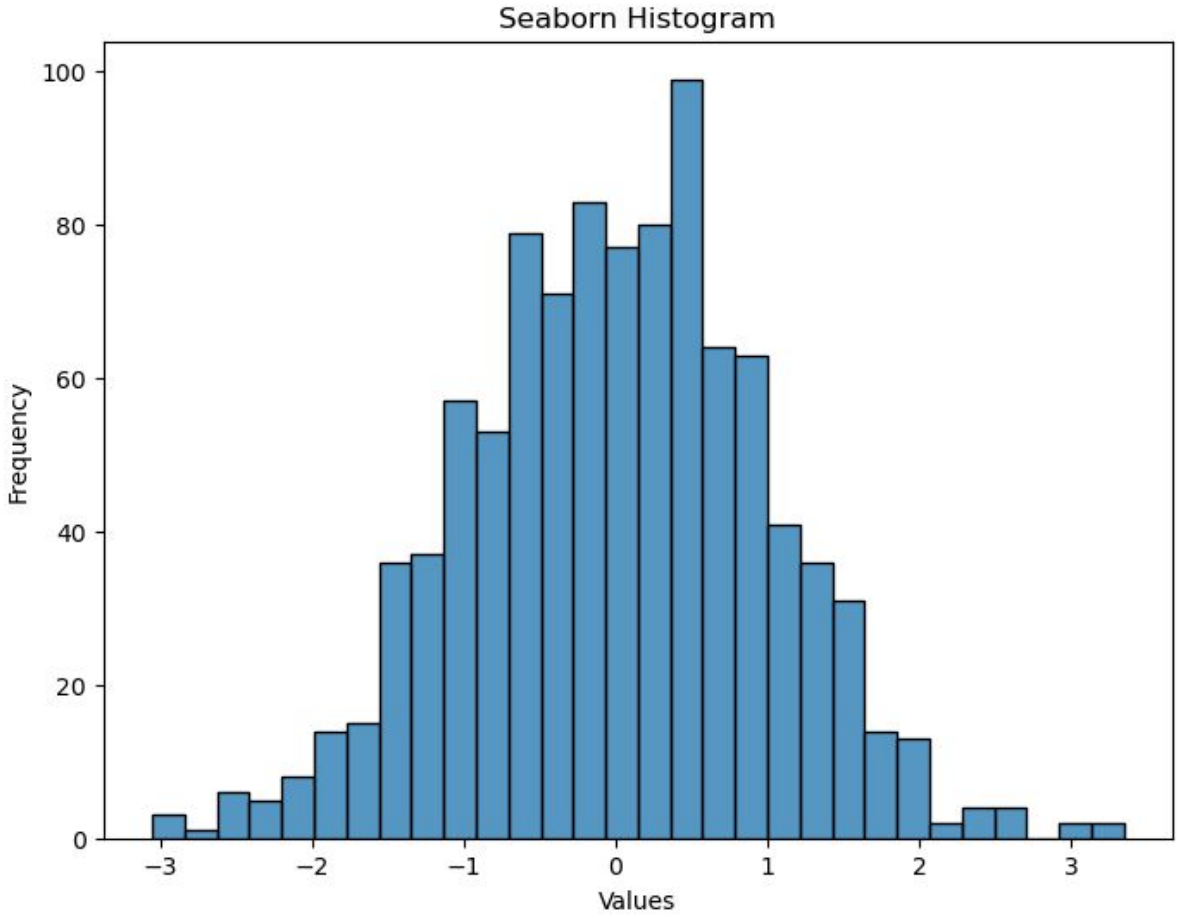
```
import seaborn as sns
import matplotlib.pyplot as plt


x = [1, 2, 3, 4, 5]
y = [2, 4, 6, 8, 10]

# Create the scatter plot
sns.scatterplot(x=x, y=y)


plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Scatter Plot')


plt.show()
```



Create a histogram plot using Seaborn.

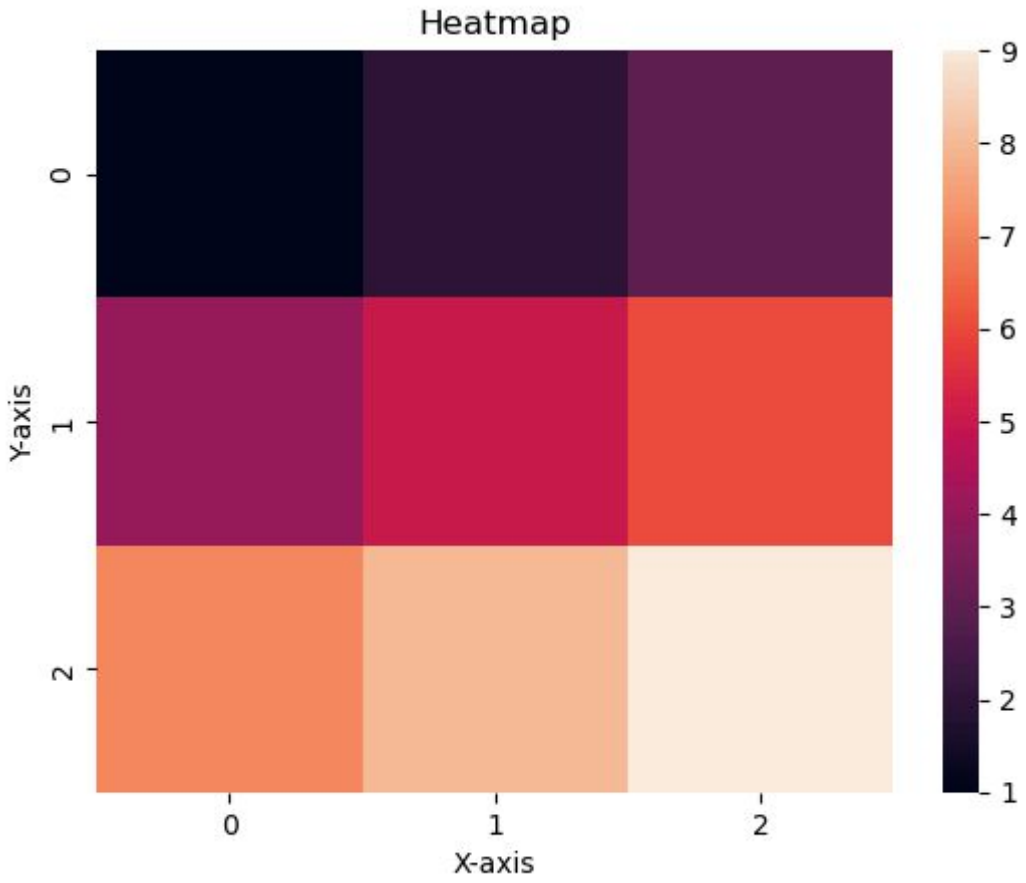
In [9]:

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

data = np.random.normal(0, 1, 1000)

# Create the histogram using Seaborn
plt.figure(figsize=(8, 6))
sns.histplot(data, bins=30)
plt.title('Seaborn Histogram')
plt.xlabel('Values')
plt.ylabel('Frequency')
plt.show()
```



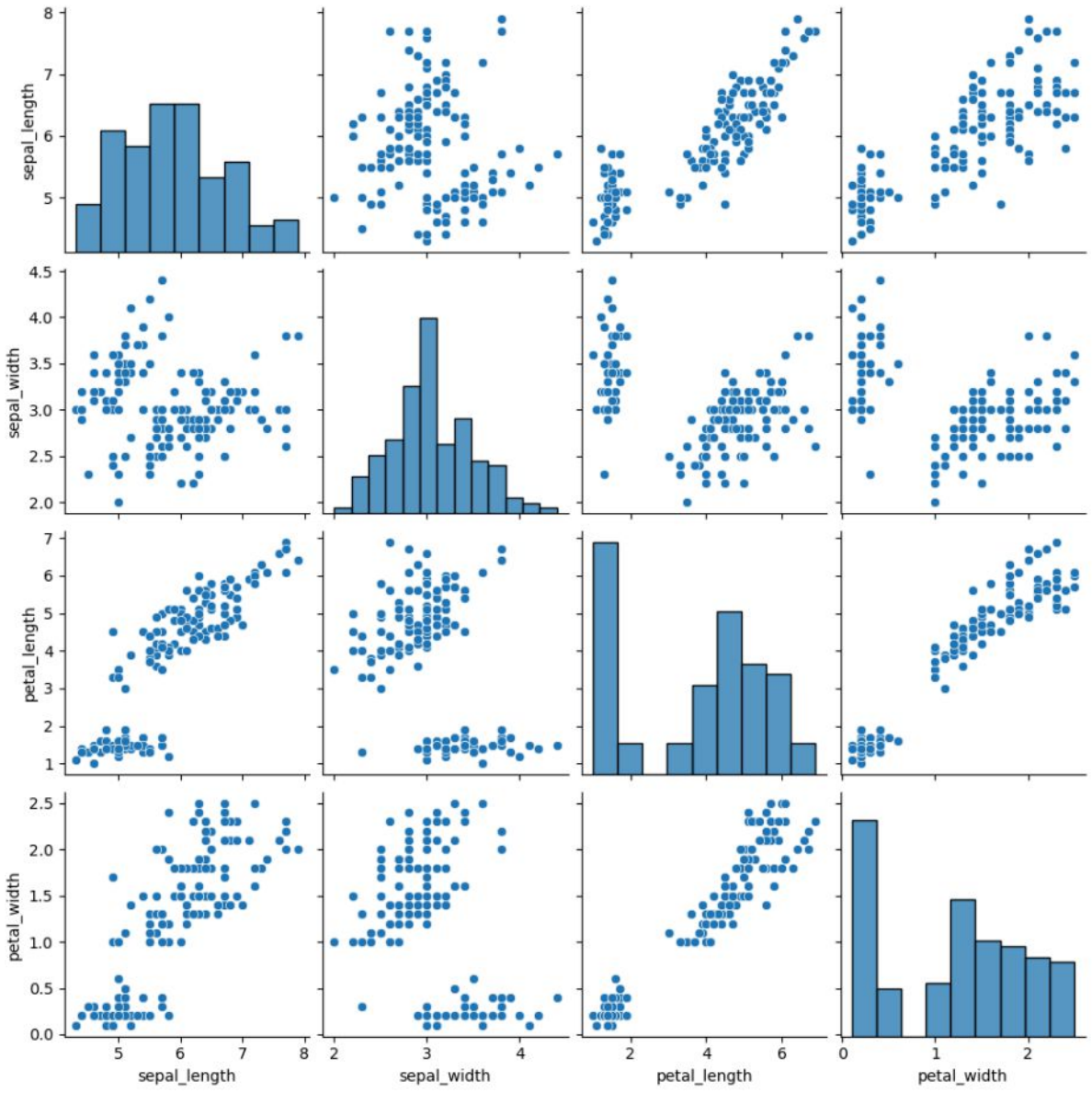
Create a heatmap using Seaborn.

In [10]:
```
import seaborn as sns
import matplotlib.pyplot as plt


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


sns.heatmap(data)

plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Heatmap')

plt.show()
```



Create a pairplot to visualize relationships between variables.

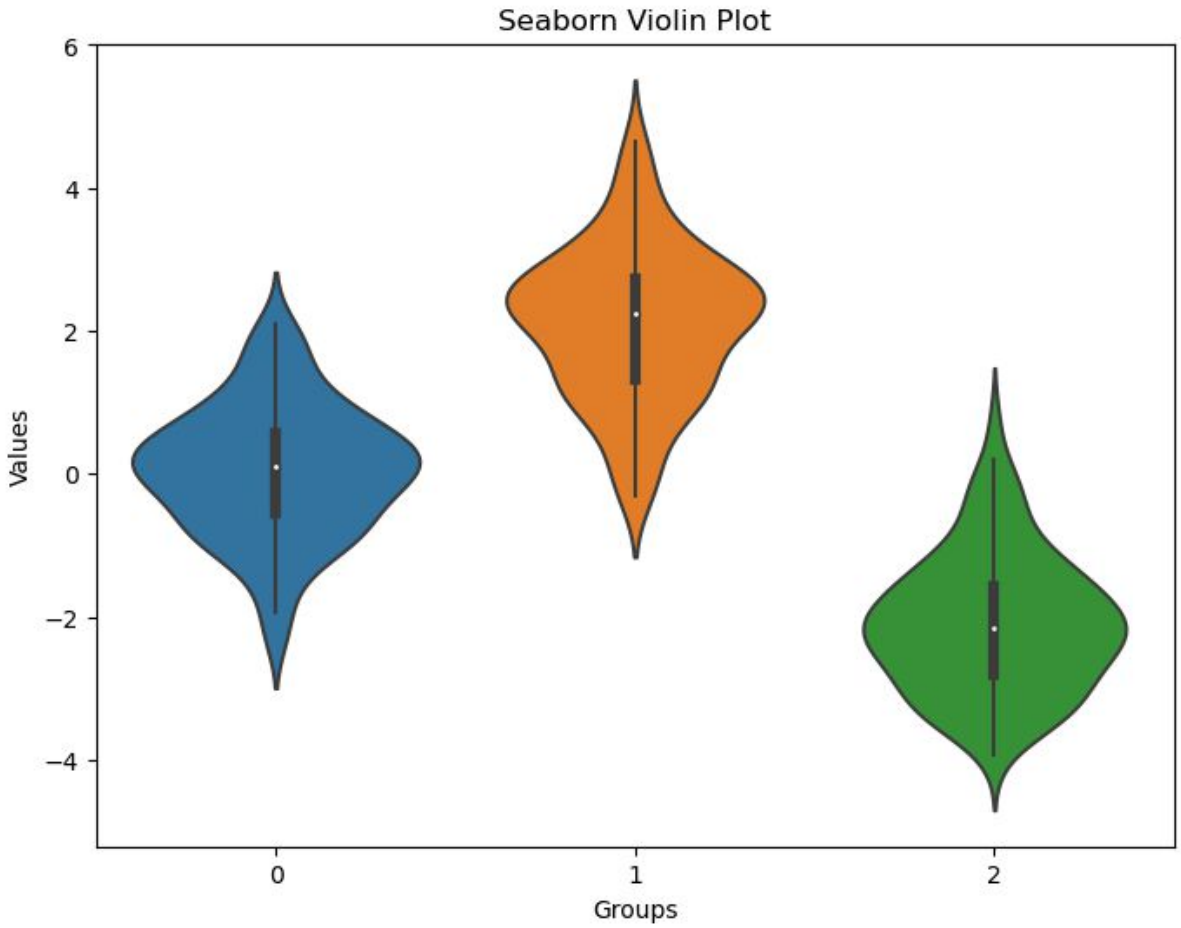
In [16]:
```
import seaborn as sns


iris = sns.load_dataset('iris')

# Create the pairplot
sns.pairplot(iris)


plt.show()
```



Create a violin plot to show the distribution of a dataset.

In [12]:

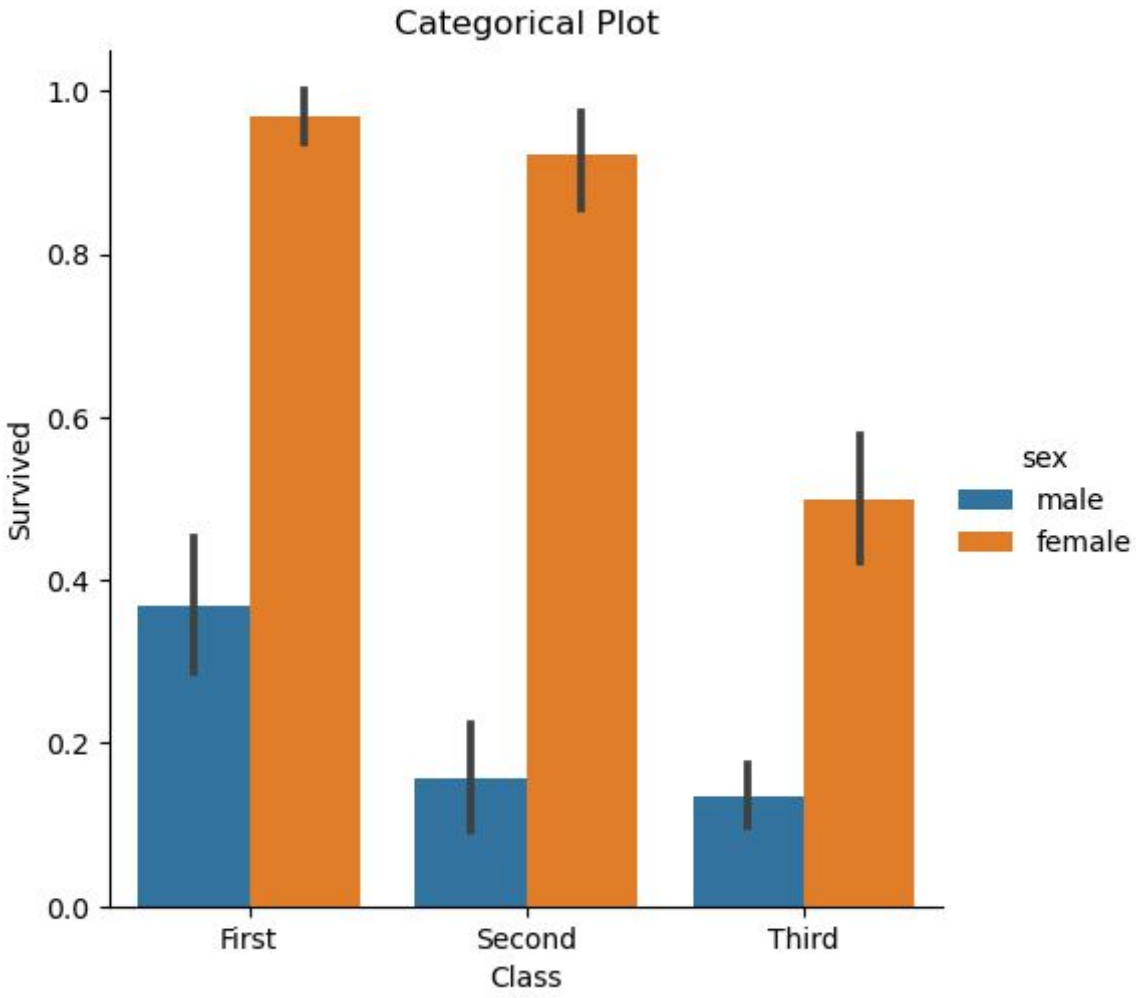
```
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np


data1 = np.random.normal(0, 1, 100)
data2 = np.random.normal(2, 1, 100)
data3 = np.random.normal(-2, 1, 100)


plt.figure(figsize=(8, 6))
sns.violinplot(data=[data1, data2, data3])
plt.title('Seaborn Violin Plot')
plt.xlabel('Groups')
plt.ylabel('Values')
plt.show()
```



Create a categorical plot.

CAT PLOT

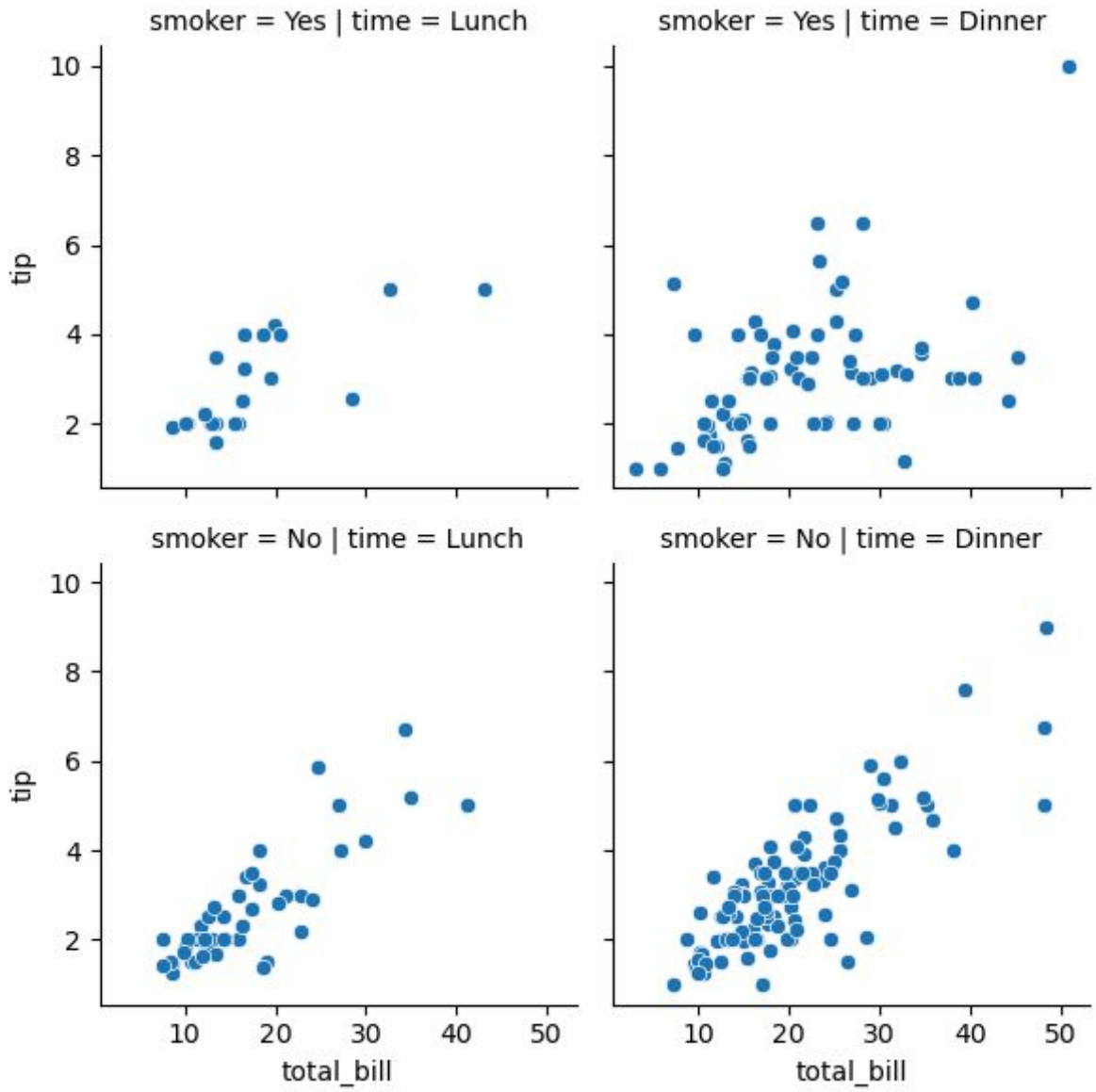
In [15]:
```
import seaborn as sns
import matplotlib.pyplot as plt

titanic = sns.load_dataset('titanic')


sns.catplot(x='class', y='survived', hue='sex', data=titanic, kind='bar')

e
plt.xlabel('Class')
plt.ylabel('Survived')
plt.title('Categorical Plot')


plt.show()
```



Seaborn FacetGrid:

Create a FacetGrid to visualize multiple subsets of data.

In [14]:
```
import seaborn as sns
import matplotlib.pyplot as plt


tips = sns.load_dataset('tips')


g = sns.FacetGrid(tips, col='time', row='smoker')


g.map(sns.scatterplot, 'total_bill', 'tip')

plt.show()
```


smoker = Yes | time = Lunch
smoker = Yes | time = Dinner
smoker = No | time = Lunch
smoker = No | time = Dinner
tip
total_bill
10
8
6
4
2
10
20
30
40
50

In [ ]: