**Experiment 3c:**

**Matplotlib:**

* **Installation of matplotlib:**

pip install matplotlib #installing matplotlib package

**Output:**

Requirement already satisfied: matplotlib in /usr/local/lib/python3.10/dist-packages (3.7.1)

Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.1.0)

Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (0.11.0)

Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (4.42.1)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.4.4)

Requirement already satisfied: numpy>=1.20 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (1.23.5)

Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (23.1)

Requirement already satisfied: pillow>=6.2.0 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (9.4.0)

Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (3.1.1)

Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.10/dist-packages (from matplotlib) (2.8.2)

Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)

1. Line Chart:

#initialing the data

import matplotlib.pyplot as plt

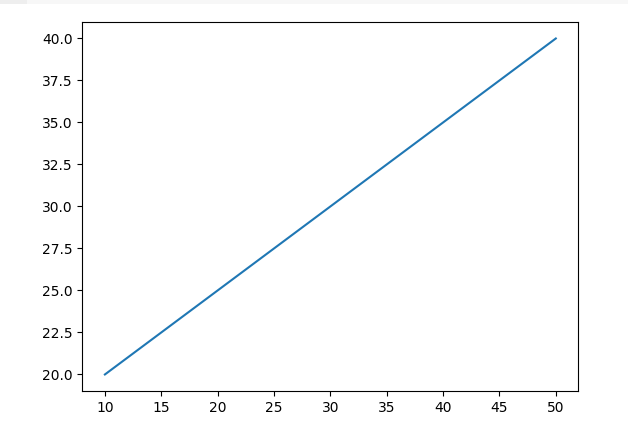
x=[10,20,30,40,50]

y=[20,25,30,35,40]

plt.plot(x, y)

plt.show()

**Output:**



1. Title to the graph:

#initialing the data

import matplotlib.pyplot as plt

x=[10,20,30,40,50]

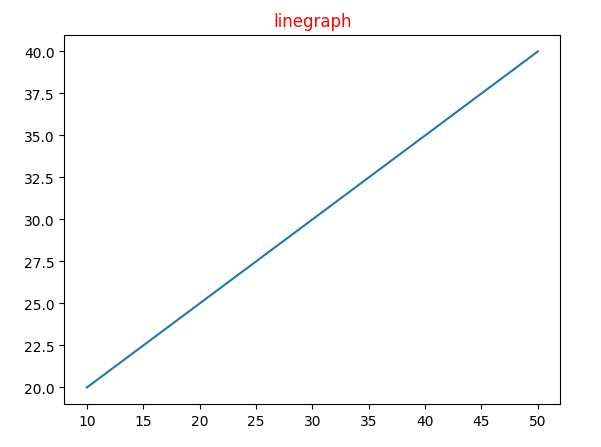
y=[20,25,30,35,40]

plt.plot(x, y)

plt.title('linegraph',fontsize=12,color='red')

plt.show()

**Output:**



3.Result count by using Line plot:

import matplotlib.pyplot as plt

categories = ['Category A', 'Category B', 'Category C', 'Category D']

counts = [15, 28, 10, 35]

plt.plot(categories, counts, marker='o')

plt.xlabel('Categories')

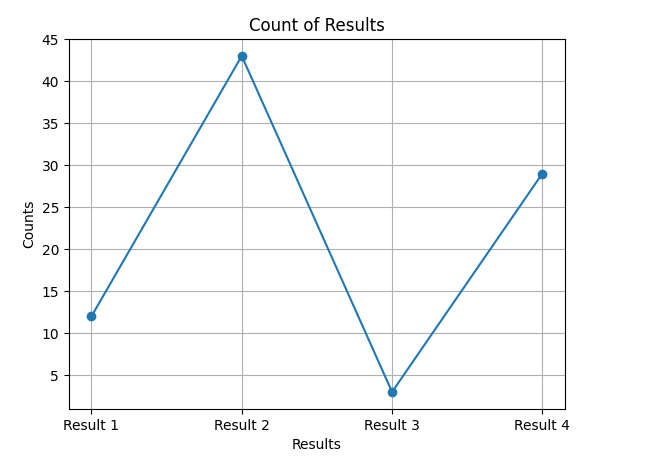
plt.ylabel('Counts')

plt.title('Count of Data by Category')

plt.grid(True)

plt.show()

Output:



1. Scatter Plot:

import matplotlib.pyplot as plt

results = ['Result 1', 'Result 2', 'Result 3', 'Result 4']

counts = [12, 43, 3, 29]

plt.scatter(results, counts, color='blue', marker='o')

plt.xlabel('Results')

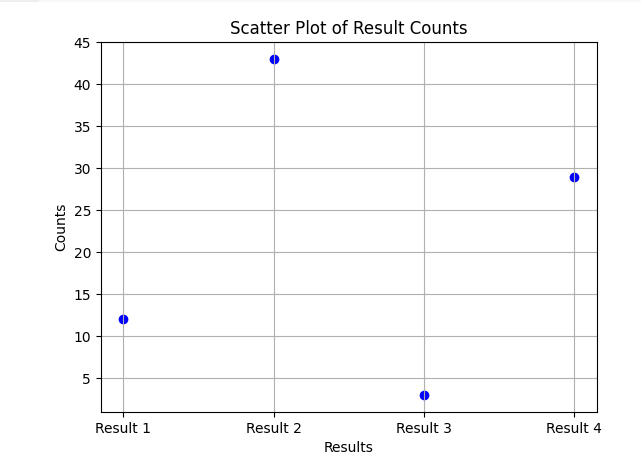
plt.ylabel('Counts')

plt.title('Scatter Plot of Result Counts')

plt.grid(True)

plt.show()

**Output:**



1. Box plot:

import matplotlib.pyplot as plt

import pandas as pd

# Sample data (replace this with your actual data)

data = {

    'company': ['Company A', 'Company A', 'Company B', 'Company C', 'Company C'],

    'role': ['Role 1', 'Role 2', 'Role 1', 'Role 2', 'Role 1'],

    'result\_count': [15, 28, 10, 22, 14]

}

# Create a DataFrame

df = pd.DataFrame(data)

# Create a box plot

plt.figure(figsize=(8, 6))  # Set the figure size

plt.boxplot(df['result\_count'], vert=True, patch\_artist=True)

plt.xlabel('Role')

plt.ylabel('Result Count')

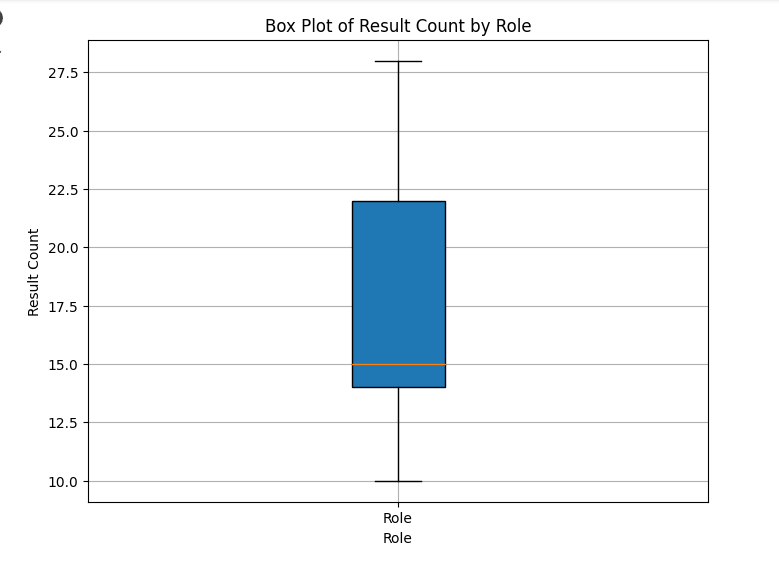
plt.title('Box Plot of Result Count by Role')

plt.xticks([1], ['Role'])  # Set x-axis label

plt.grid(True)

plt.show()

**Output:**



* Adding Legend to the line chart:

#initialization of data

x=[10,20,30,40]

y=[20,30,40,50]

fig=plt.figure(figsize=(5,4))

ax=fig.add\_axes([1,1,1,1])

#ploting first dataset to gif

ax1=ax.plot(x,y)

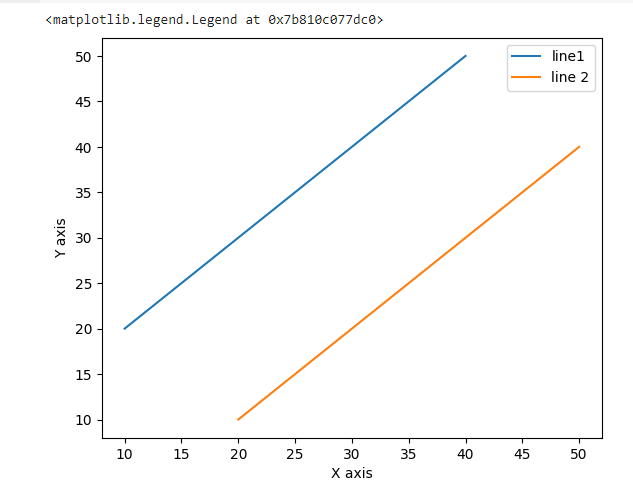
ax2=ax.plot(y,x)

ax.set\_xlabel("X axis")

ax.set\_ylabel("Y axis")

ax.legend(labels=('line1','line 2'))

**Output:**



* Dotted line chart:

import matplotlib.pyplot as plt

from matplotlib import markers

x=[10,20,30,40]

y=[20,30,40,50]

plt.plot(x, y,color='green',linewidth=3,marker='o',markersize=15,linestyle='--')

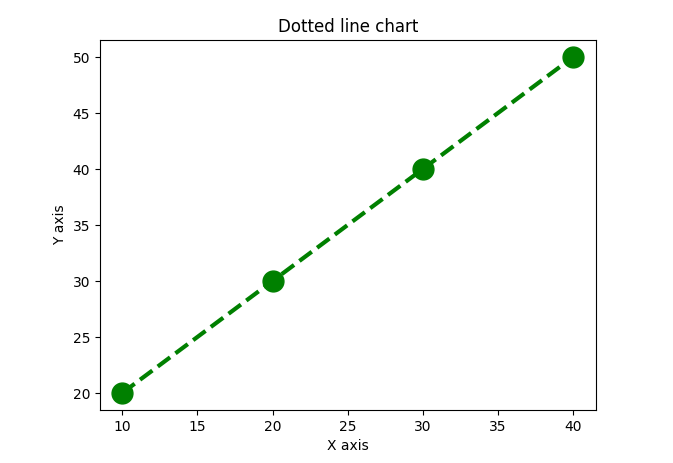
plt.title('Dotted line chart')

plt.ylabel('Y axis')

plt.xlabel('X axis')

plt.show()

**Output:**



1. Bar chart:

import matplotlib.pyplot as plt

import pandas as pd

# Sample data (replace this with your actual data)

data = {

'seekers': [1, 2, 3, 4],

'result\_count': [10, 25, 15, 30]

}

# Convert data to DataFrame

df = pd.DataFrame(data)

# Create a bar plot

plt.bar(df['seekers'], df['result\_count'], color='green', edgecolor='black', linewidth=2)

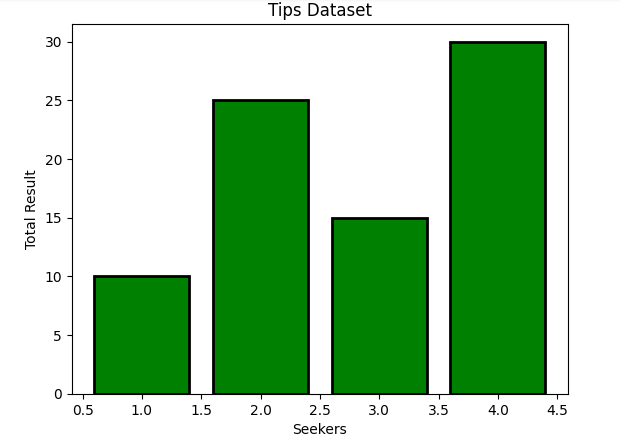
plt.title('Tips Dataset')

plt.ylabel('Total Result')

plt.xlabel('Seekers')

plt.show()

Output:



1. **Histogram:**

**import matplotlib.pyplot as plt**

**import pandas as pd**

**# Sample data (replace this with your actual data)**

**data = {**

**'result\_count': [7,10,13,16,19,22,25,28,31]**

**}**

**# Convert data to DataFrame**

**df = pd.DataFrame(data)**

**# Create a histogram**

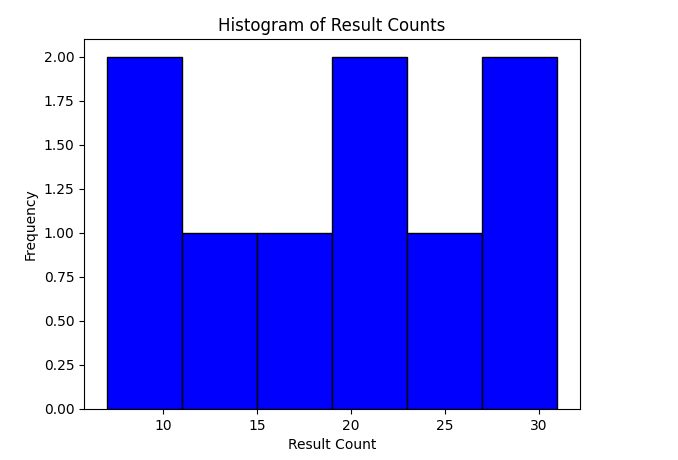
**plt.hist(df['result\_count'], bins=6, color='blue', edgecolor='black')**

**plt.title('Histogram of Result Counts')**

**plt.xlabel('Result Count')**

**plt.ylabel('Frequency')**

**plt.show()**



**7.Pie chart1:**

import matplotlib.pyplot as plt

seekers=['Company','Role','Function','Designation','Location']

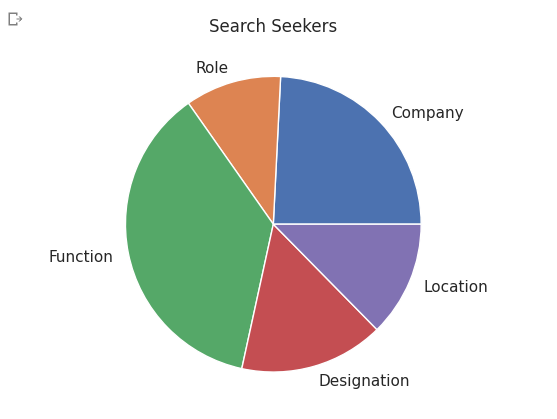
data=[23,10,35,15,12]

plt.pie(data,labels=seekers)

plt.title('Search Seekers')

plt.show()

Output:



8.Pie Chart 2:

import matplotlib.pyplot as plt

seekers=['Company','Role','function','Designation','Location']

data=[23,10,15,15,12]

explore=[0.1,0.5,0,0,0]

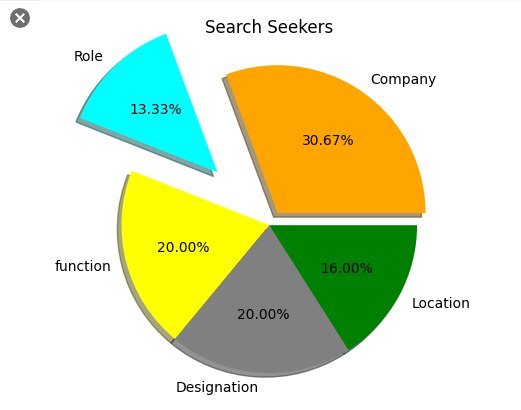
color=['Orange','Cyan','yellow','grey','green']

plt.pie(data,labels=seekers,explode=explore,autopct='%1.2f%%',colors=colors,shadow=True)

plt.title('Search Seekers')

plt.show()

Output:



9.Saving a Plot:

import matplotlib.pyplot as plt

year=['2001','2002','2004','2006','2008']

production=[25,15,35,30,10]

plt.bar(year,production)

plt.savefig("outpute.jpg")

plt.savefig("output1",facecolor='y',bbox\_inches="tight",pad\_inches=0.3,transparent=True)

Output:

