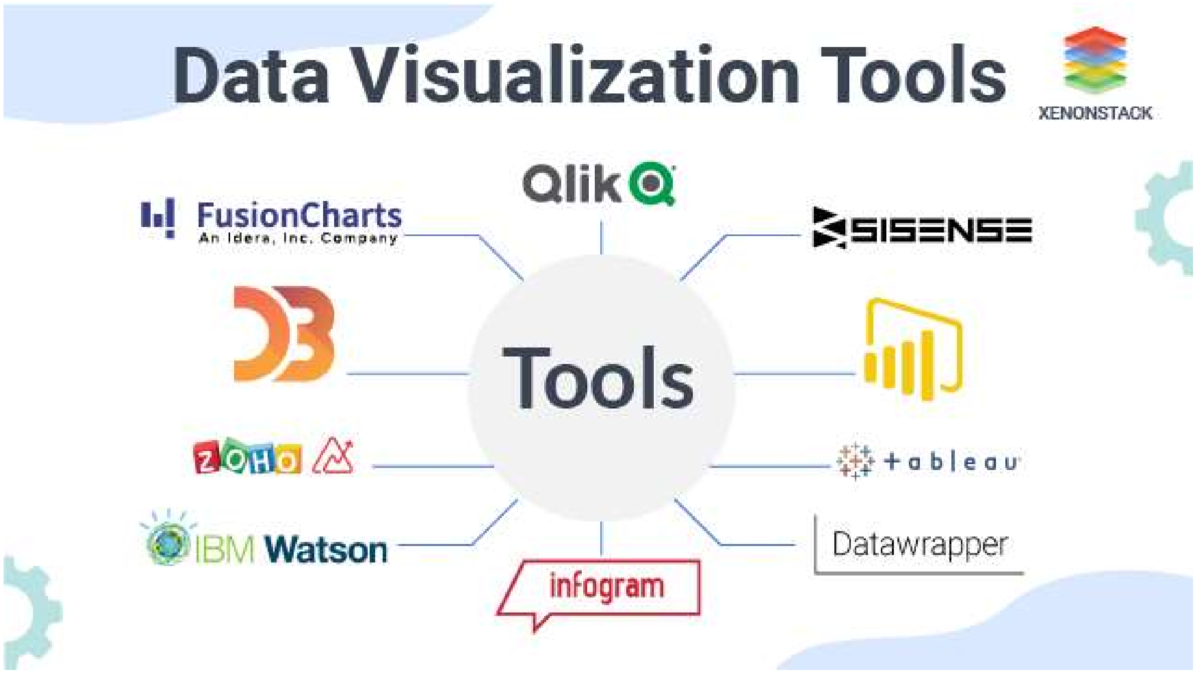
What is Data Visualization?

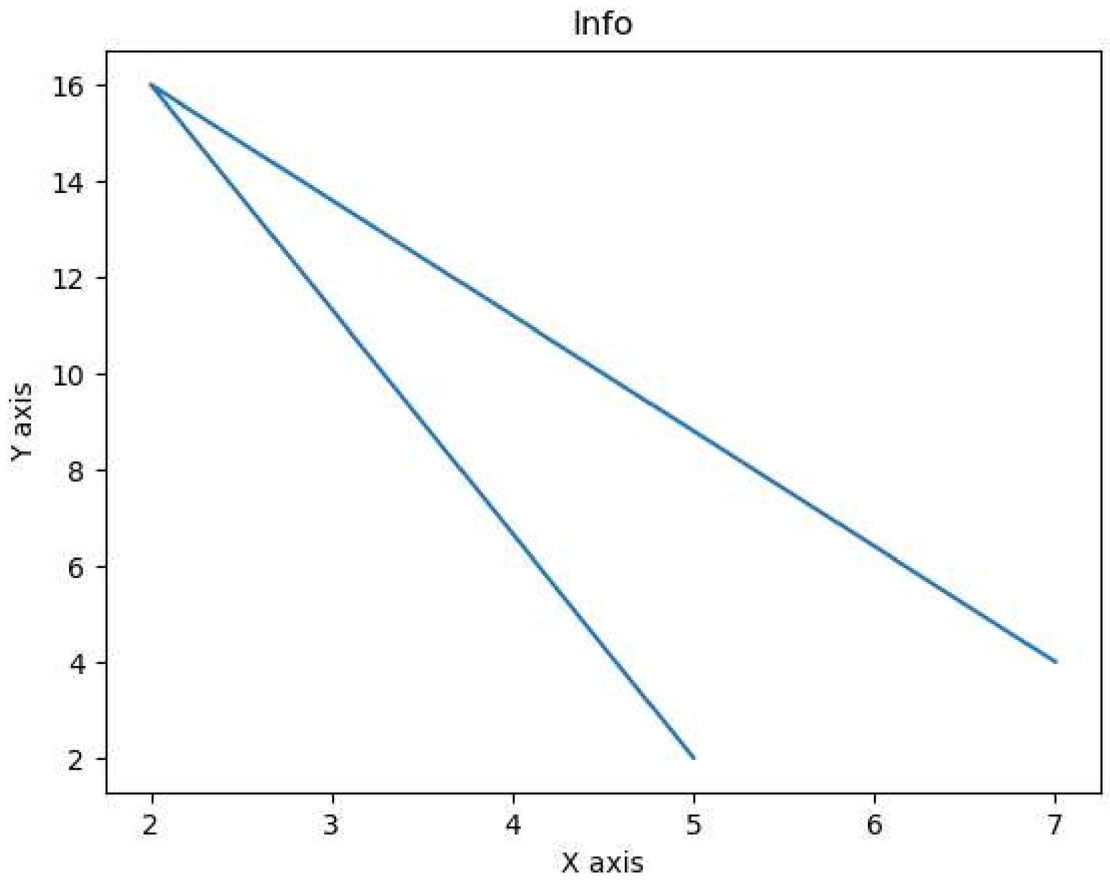
Data visualization is the graphical representation of data to help people understand context and significance. Interactive data visualization enables companies to drill down to explore details, identify patterns and outliers.



Matplotlib is easy to use and an amazing visualizing library in Python.

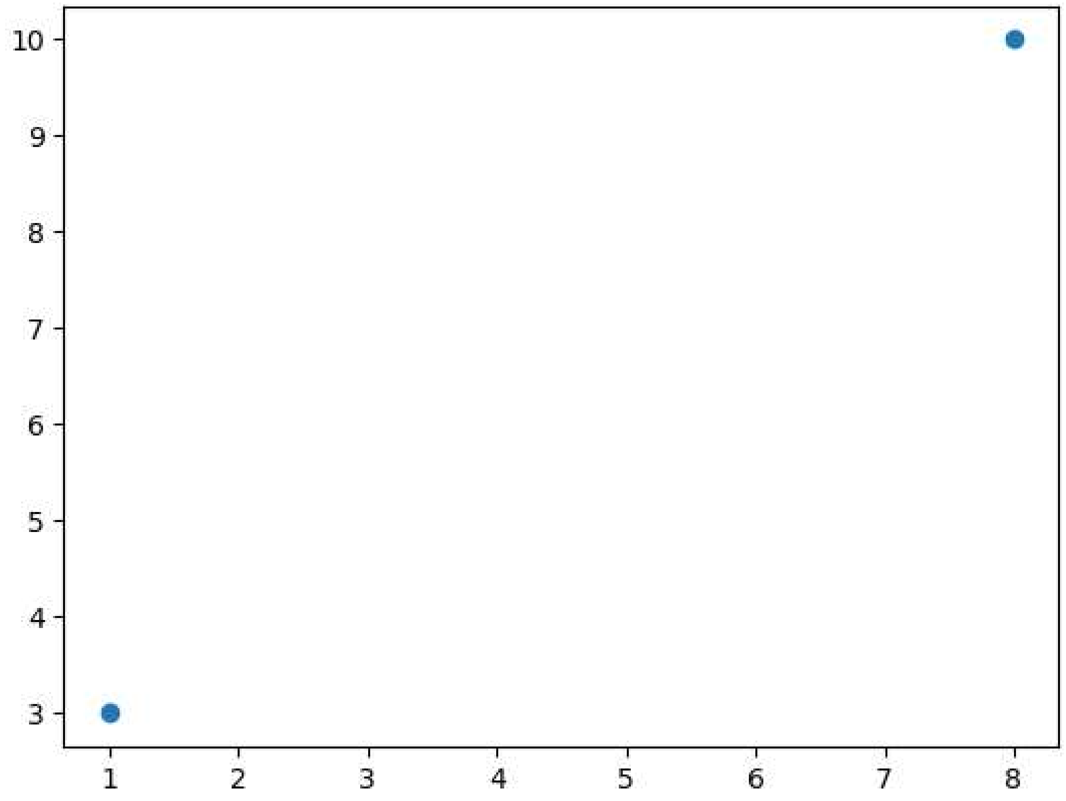
```
In [1]: #!pip install matplotlib
        from matplotlib import pyplot as plt
        import numpy as np
```

```
In [2]: x = [5,2,7] #(5,2) (2,16) (7,4)
        y = [2,16,4]
        plt.plot(x,y)
        plt.title('Info')
        plt.ylabel('Y axis')
        plt.xlabel('X axis')
        plt.show()
```

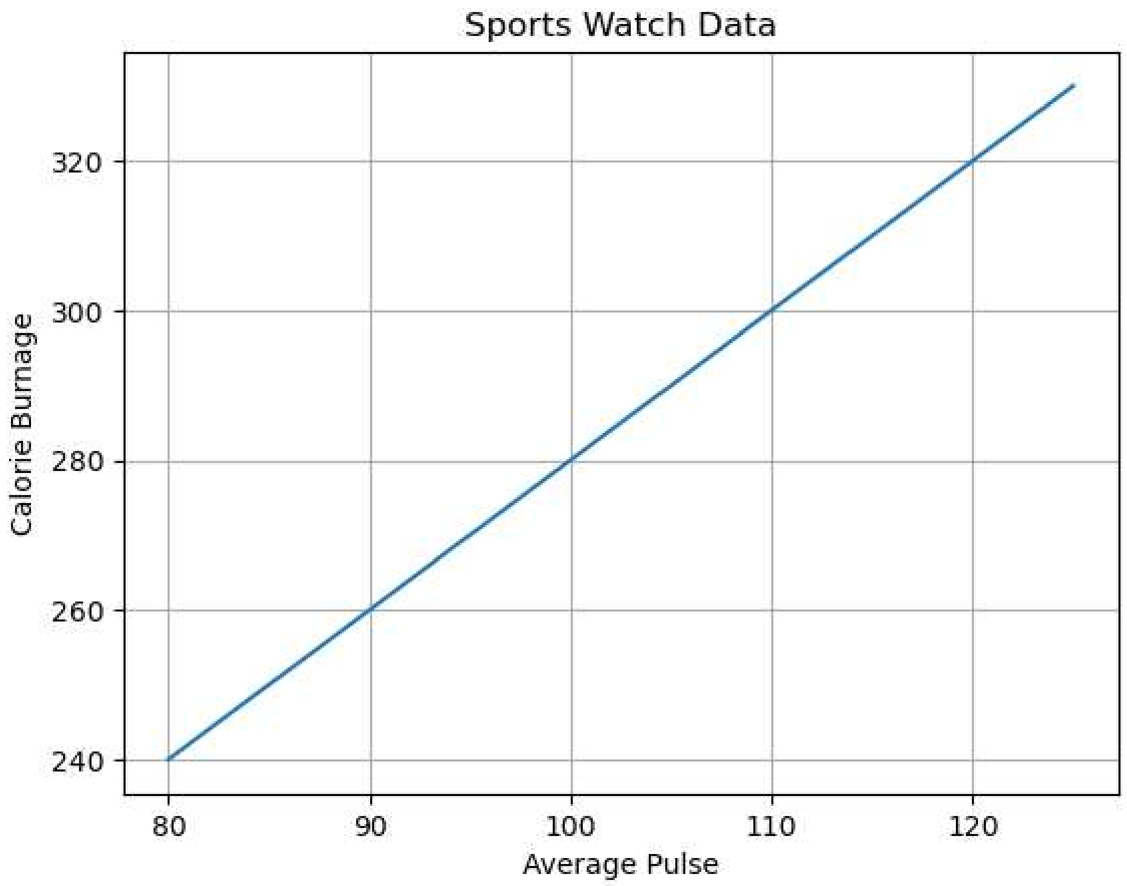


ploting without line

```
In [3]: xpoints = np.array([1, 8])
        ypoints = np.array([3, 10])

        plt.plot(xpoints, ypoints,'o')
        plt.show()
```

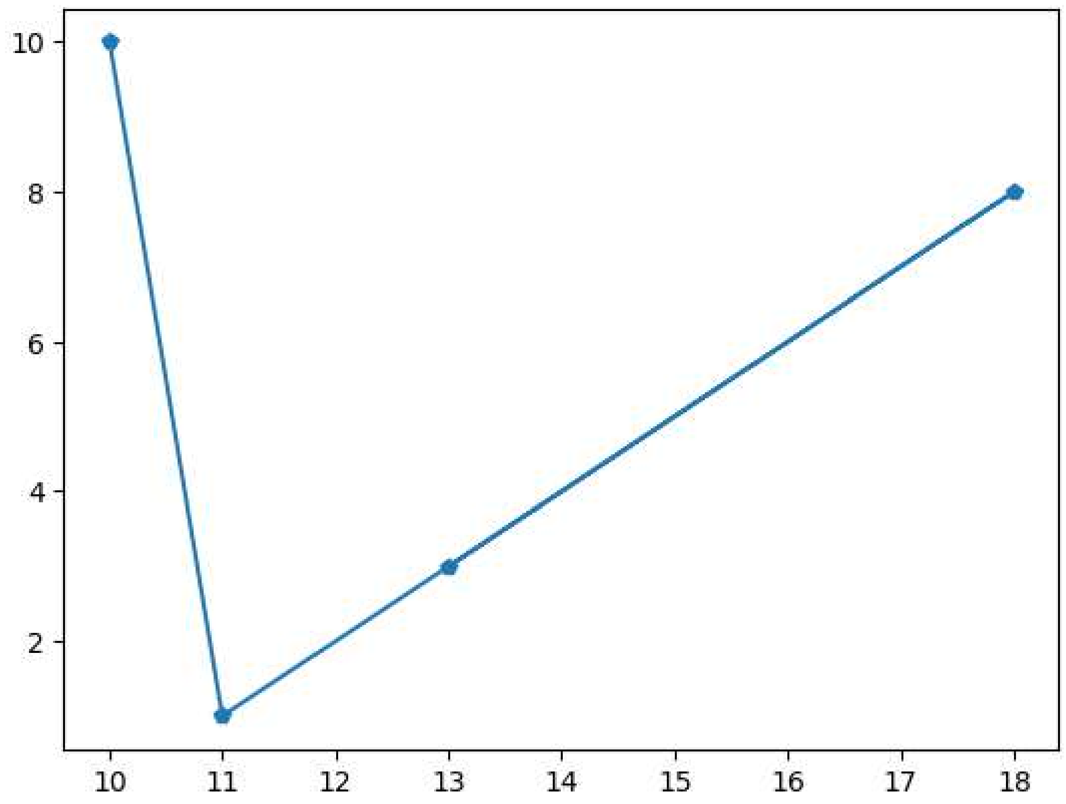


In [4]:
```
# With Pyplot, you can use the grid() function to add grid lines to the plot.

x = np.array([80, 85, 90, 95, 100, 105, 110, 115, 120, 125])
y = np.array([240, 250, 260, 270, 280, 290, 300, 310, 320, 330])

plt.title("Sports Watch Data")
plt.xlabel("Average Pulse")
plt.ylabel("Calorie Burnage")

plt.plot(x, y)

plt.grid()

plt.show()
```



Markers

In [5]:
```
ypoints = np.array([3, 8, 1, 10])
xpoints = np.array([13, 18, 11, 10])
#plt.plot(ypoints, marker = 'o')
plt.plot(xpoints,ypoints, marker = 'p')
plt.show()
```



'''Marker Description 'o' Circle

'*' Star

'.' Point

',' Pixel

'x' X

'X' X (filled)

'+' Plus

'P' Plus (filled)

's' Square

'D' Diamond 'd' Diamond (thin)

'p' Pentagon

'H' Hexagon 'h' Hexagon 'v' Triangle Down

'^' Triangle Up '<' Triangle Left

'>' Triangle Right

'1' Tri Down

'2' Tri Up

'3' Tri Left

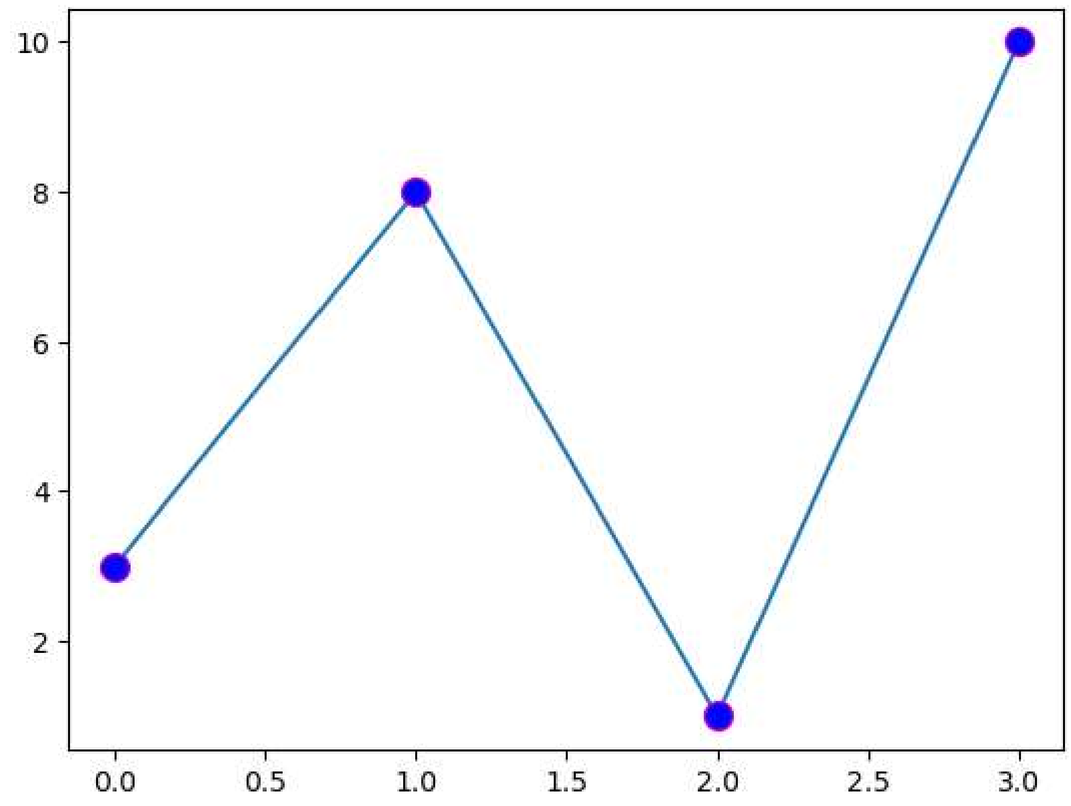
'4' Tri Right

'|' Vline

'_' Hline'''

In [6]:
```
plt.plot(ypoints, marker = 'o', ms = 10, mec = 'm', mfc = 'b') #edge color #face colo
```

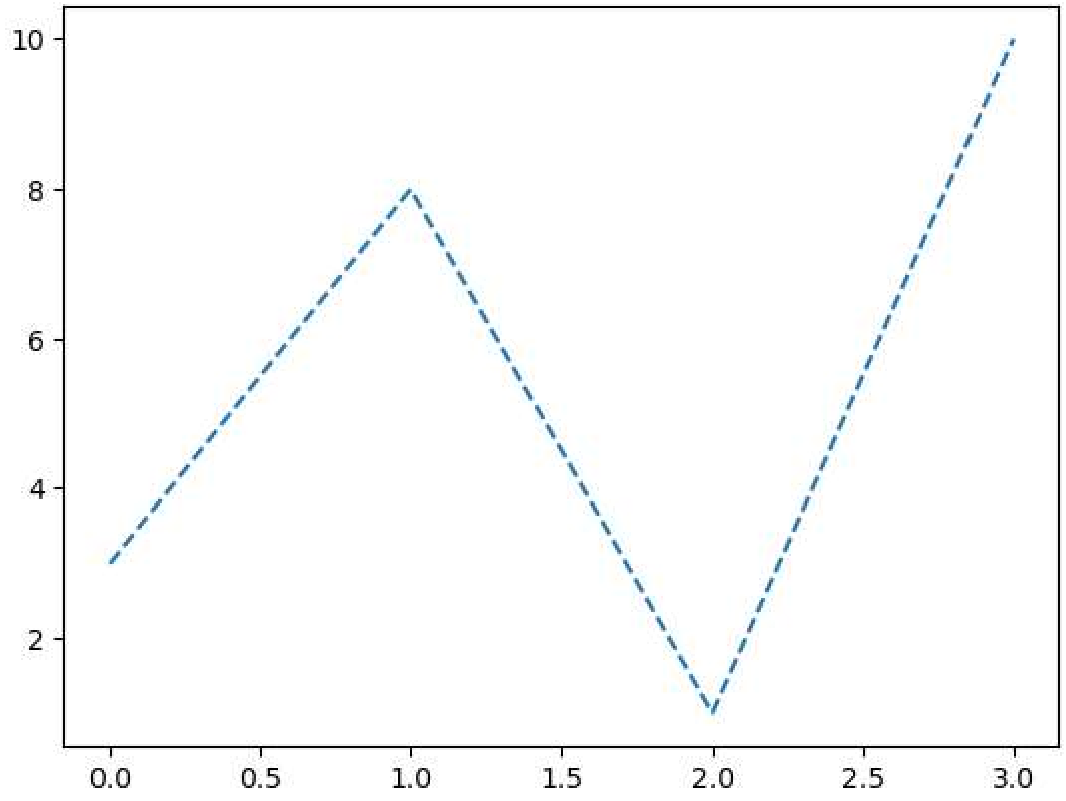
Out[6]:
```
[<matplotlib.lines.Line2D at 0x1a4132977f0>]
```



Color Syntax Description 'r' Red 'g' Green
'b' Blue
'c' Cyan
'm' Magenta 'y' Yellow
'k' Black
'w' White

line plot

In [7]:
```
ypoints = np.array([3, 8, 1, 10])

plt.plot(ypoints, linestyle = 'dashed')
plt.show()
```

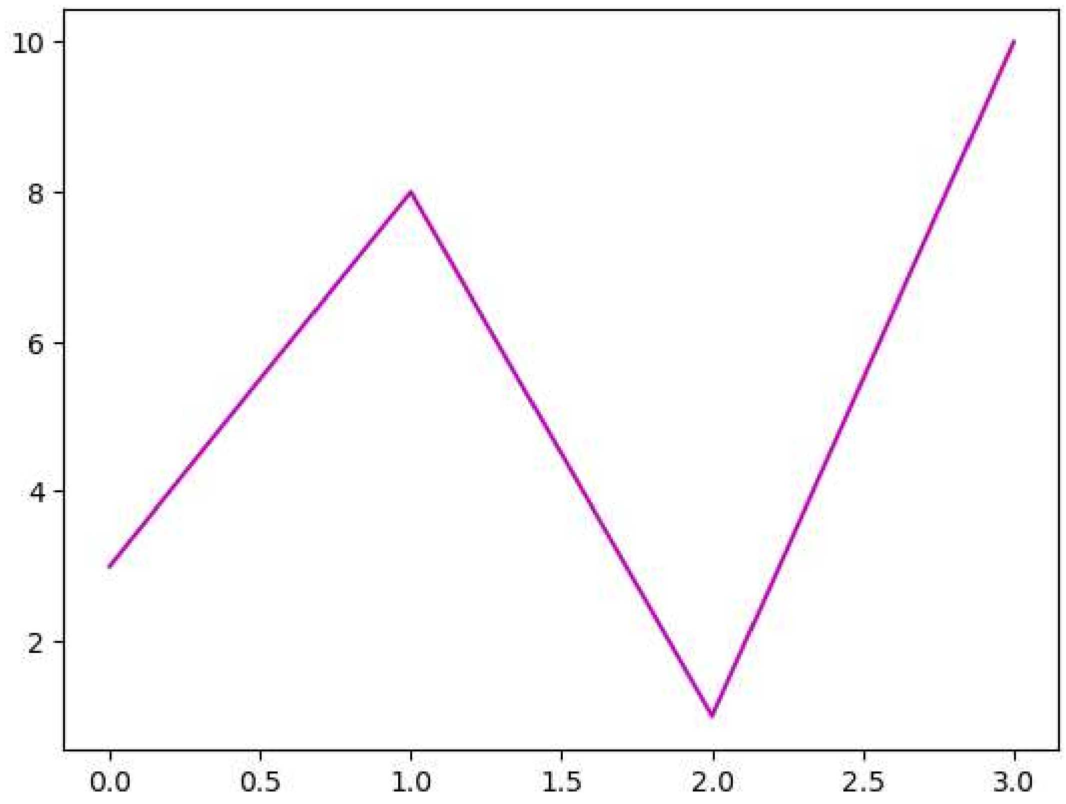


'solid' (default) '-' 'dotted' ':' 'dashed' '--'

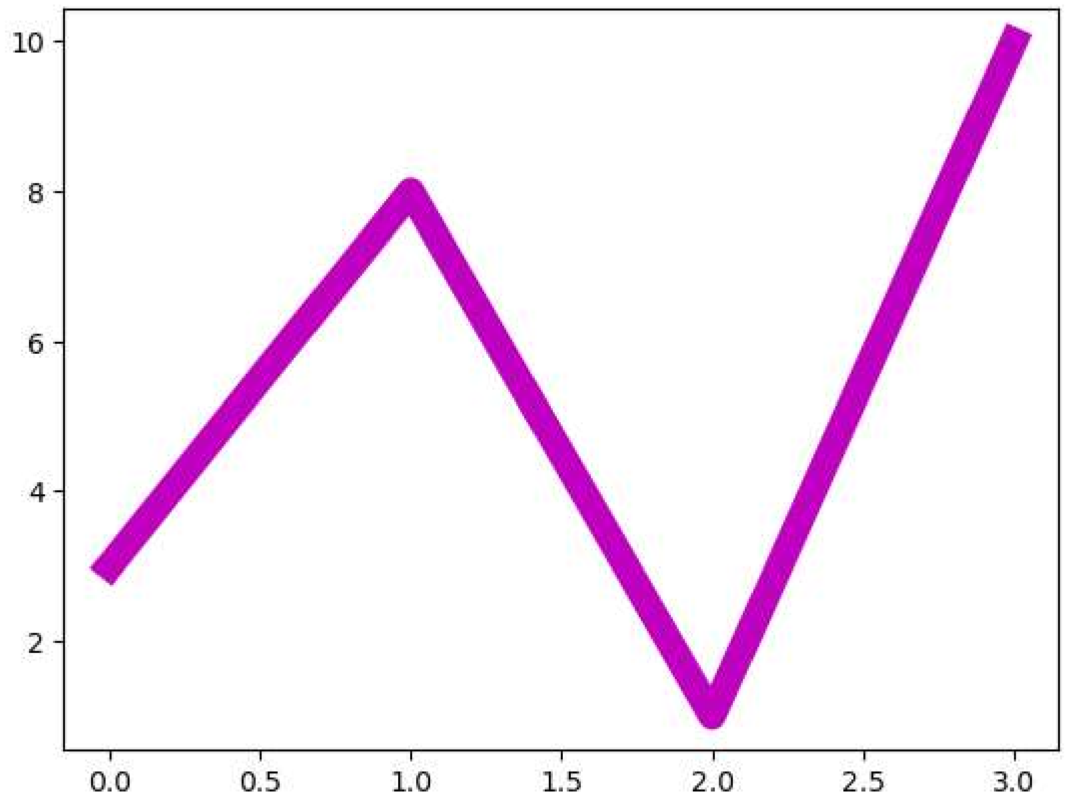
'dashdot' '-.'

'None' '' or ' '

In [8]:
```
plt.plot(ypoints, color = 'm')
plt.show()
```



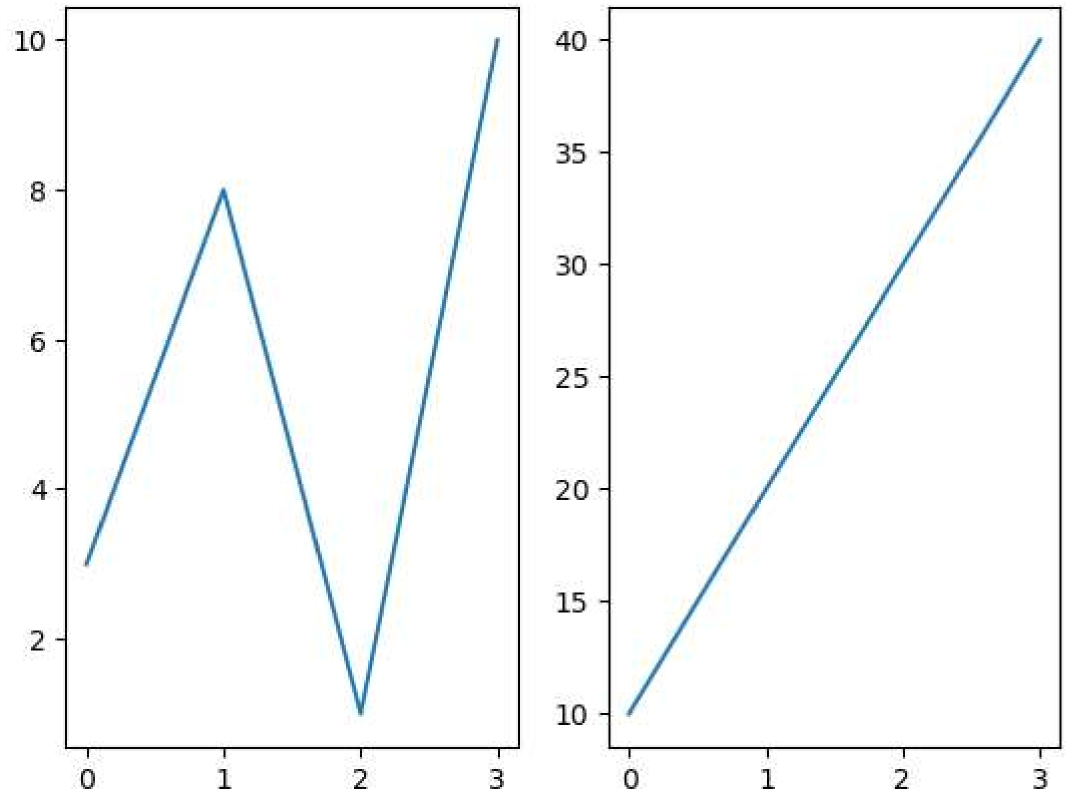
In [9]:
```
plt.plot(ypoints, linewidth = '10',color='m')#Line width
plt.show()
```



subplot

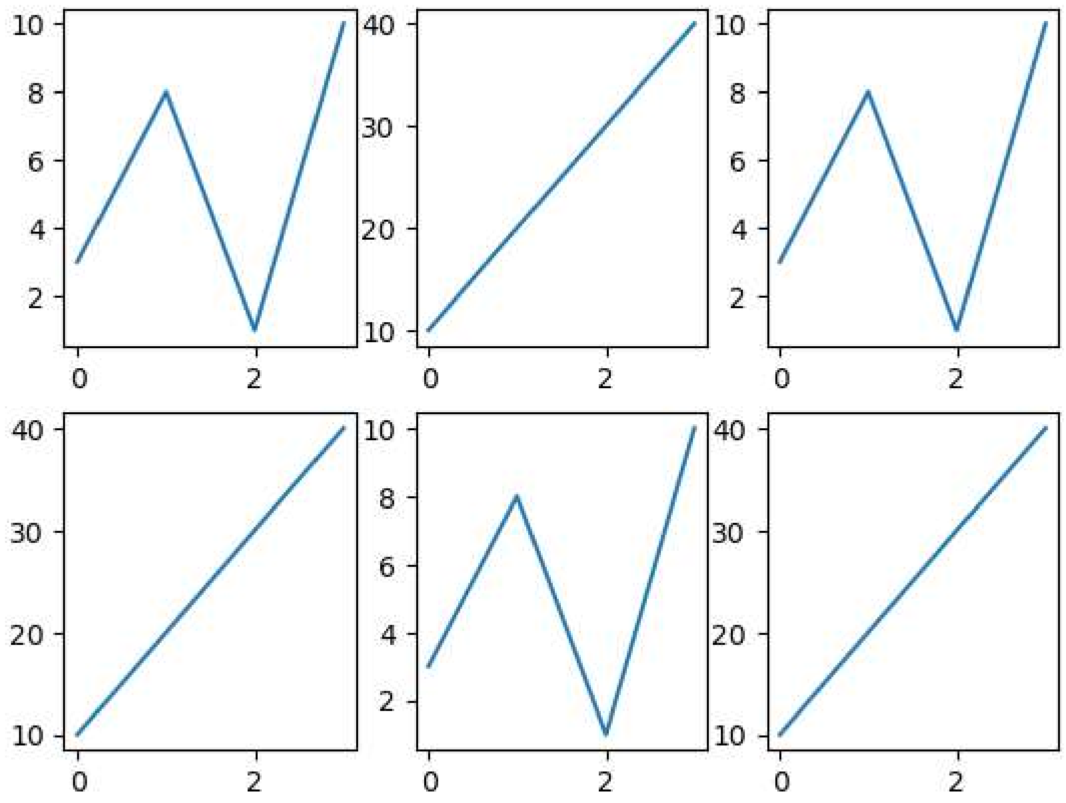
In [10]:
```
'''Display Multiple Plots
With the subplots() function you can draw multiple plots in one figure:
'''
#plot 1:
x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])
print(x)
print(y)

plt.subplot(1,2,1)
plt.plot(x,y)

#plot 2:
x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(1, 2, 2)
plt.plot(x,y)

plt.show()
```

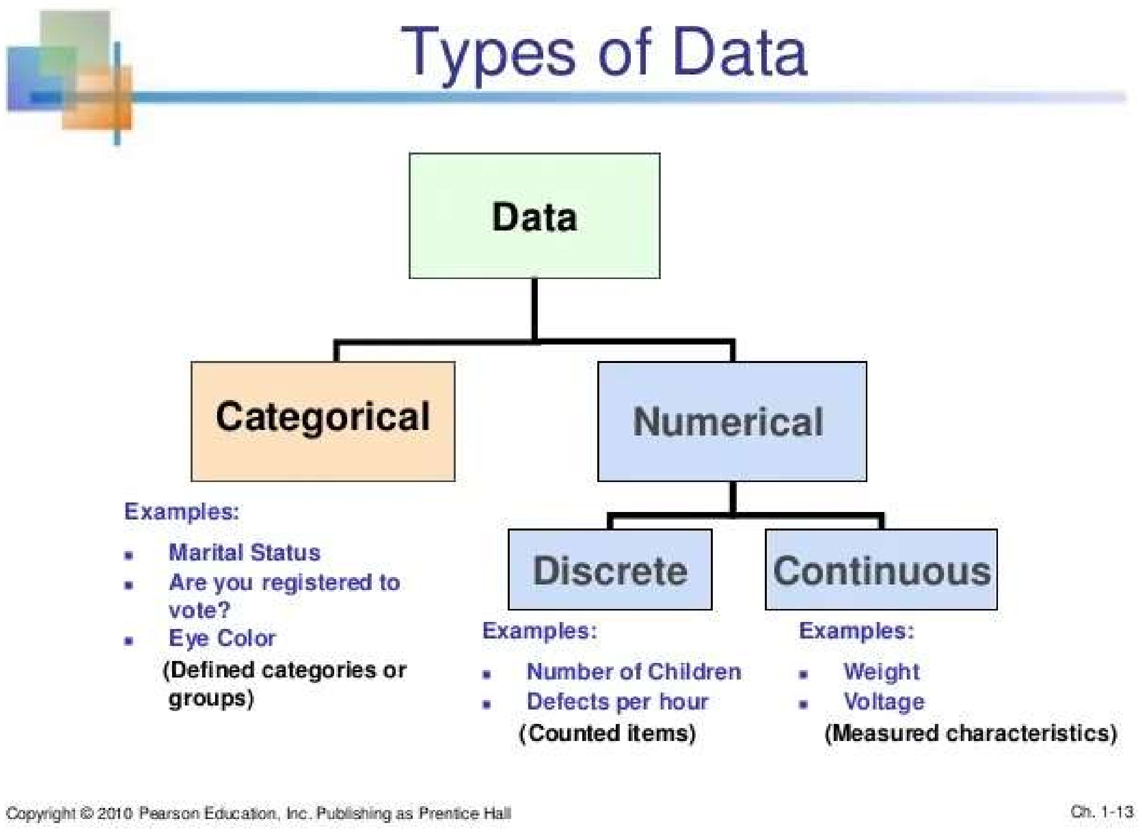
```
[0 1 2 3]
[ 3  8  1 10]
```



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

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 1)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
```

```
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 2)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 3)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 4)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([3, 8, 1, 10])

plt.subplot(2, 3, 5)
plt.plot(x,y)

x = np.array([0, 1, 2, 3])
y = np.array([10, 20, 30, 40])

plt.subplot(2, 3, 6)
plt.plot(x,y)

plt.show()
```



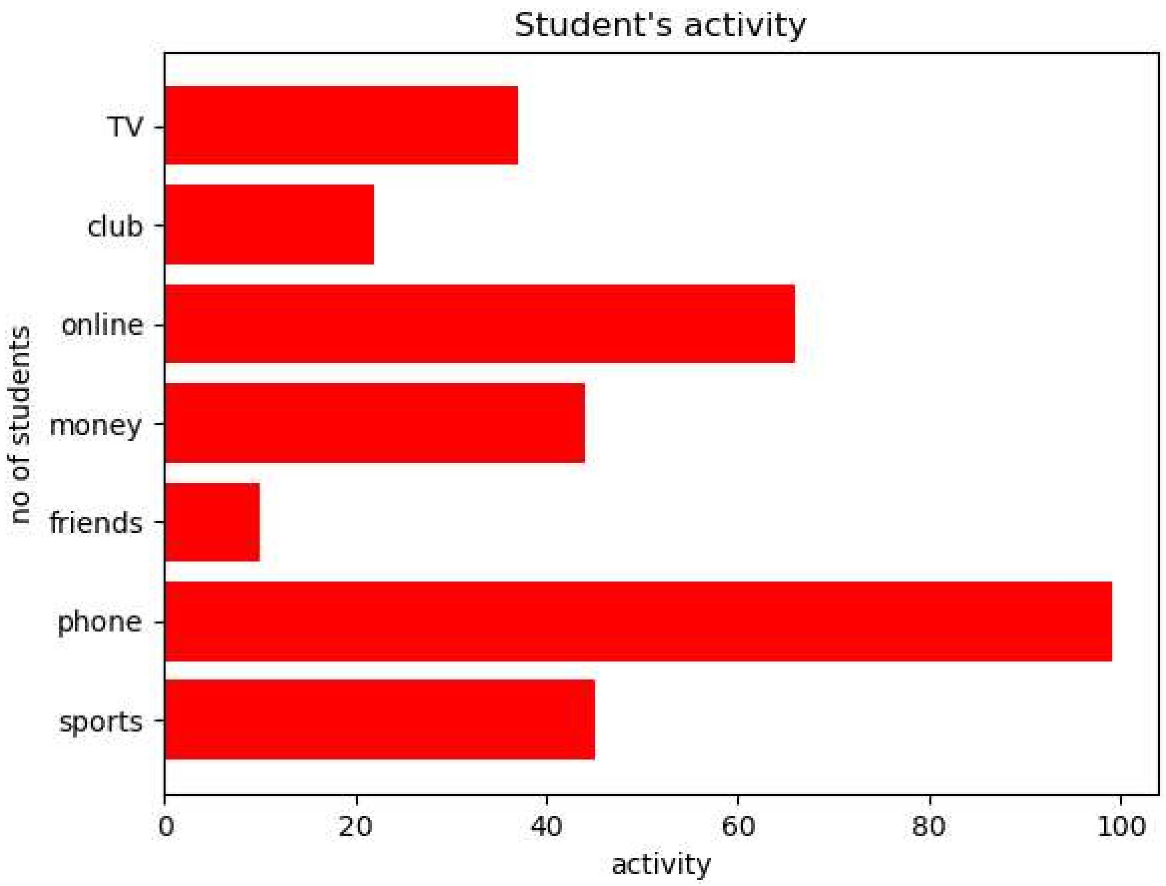


Bar plot

Bar graphs are one of the most common types of graphs and are used to show data associated with the categorical variables. Matplotlib provides a bar() to make bar graphs which accepts arguments such as: categorical variables, their value and color.

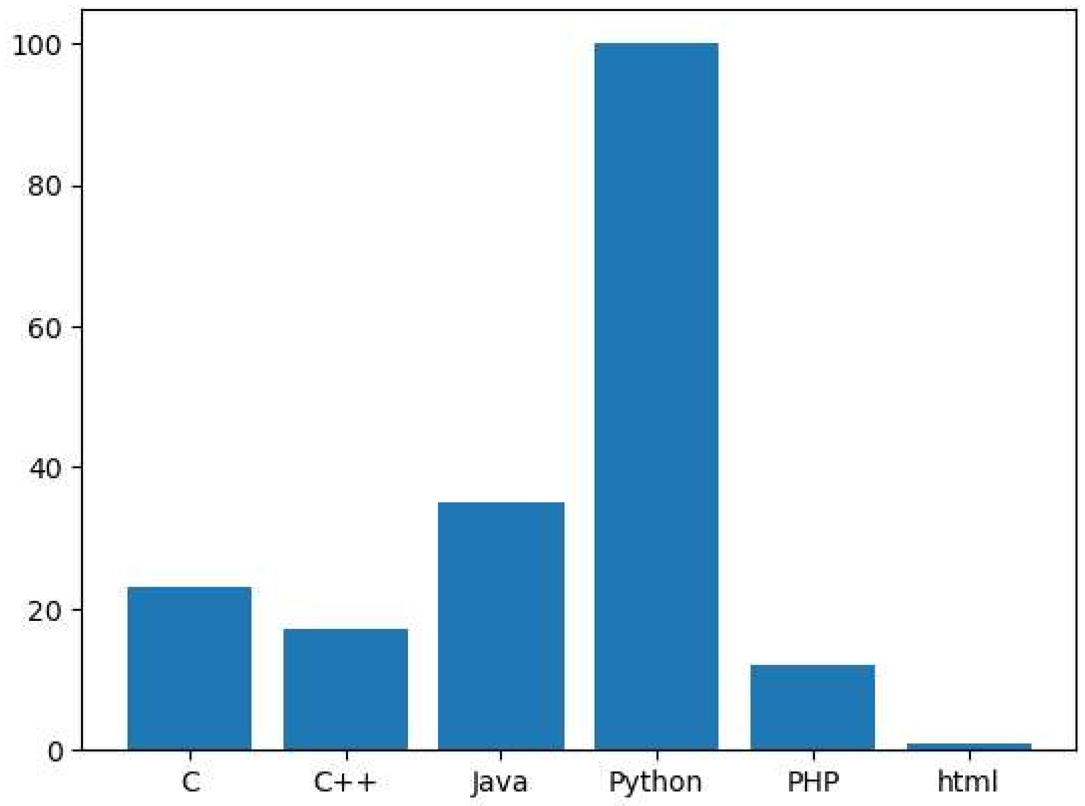
Horizontal Bar Chart

In [12]:
```
activities = ["sports","phone","friends","money","online","club","TV"]
frequency = [45,99,10,44,66,22,37]

plt.barh(activities, frequency, color="r") # default colour is blue, color is used to
plt.title("Student's activity")

plt.xlabel("activity")
plt.ylabel("no of students")

plt.show()
```

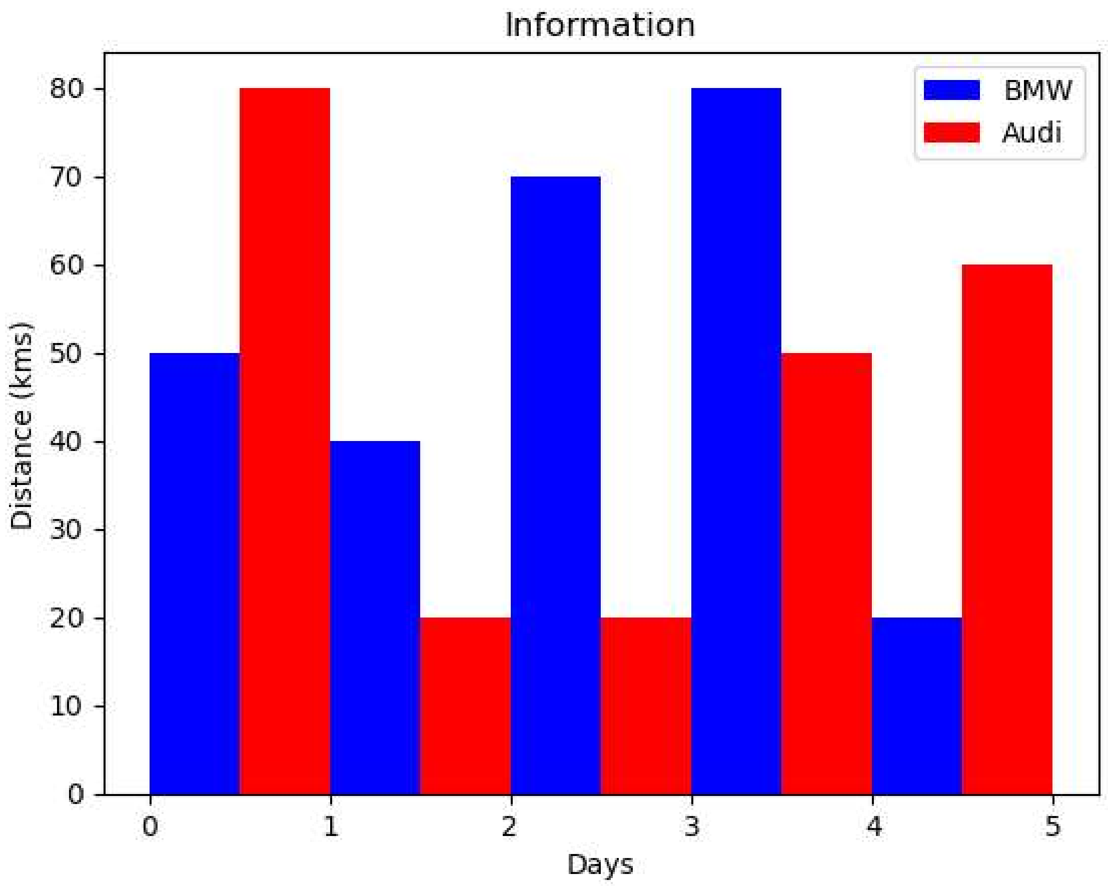


Vertical Bar Chart

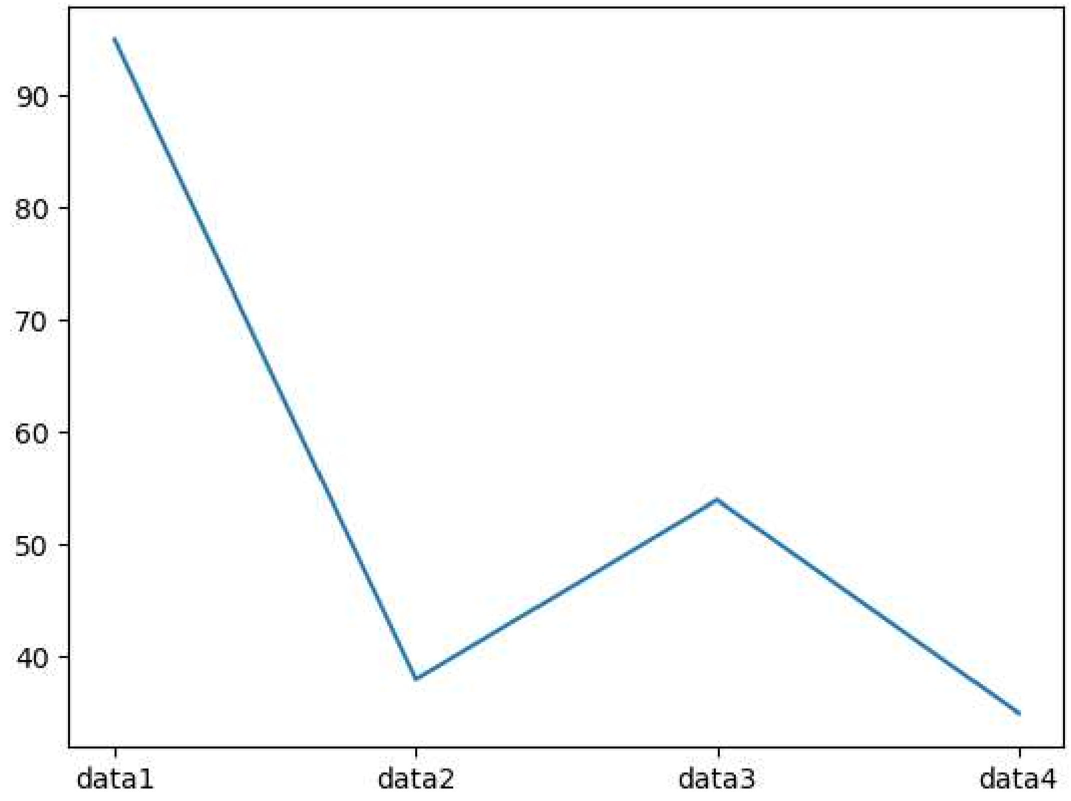
```
In [13]: langs = ['C', 'C++', 'Java', 'Python', 'PHP','html']
         students = [23,17,35,100,12,1]
         plt.bar(langs,students)
         plt.show()
```



In [14]:
```
plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],
label="BMW",color='b',width=.5)
plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],
label="Audi", color='r',width=.5)
plt.legend()
plt.xlabel('Days')
plt.ylabel('Distance (kms)')
plt.title('Information')
plt.show()
```



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

x = [1, 2, 3, 4]
y = [95, 38, 54, 35]
labels = ['data1', 'data2', 'data3', 'data4']

plt.plot(x, y)

plt.xticks(x, labels)

plt.show()
```

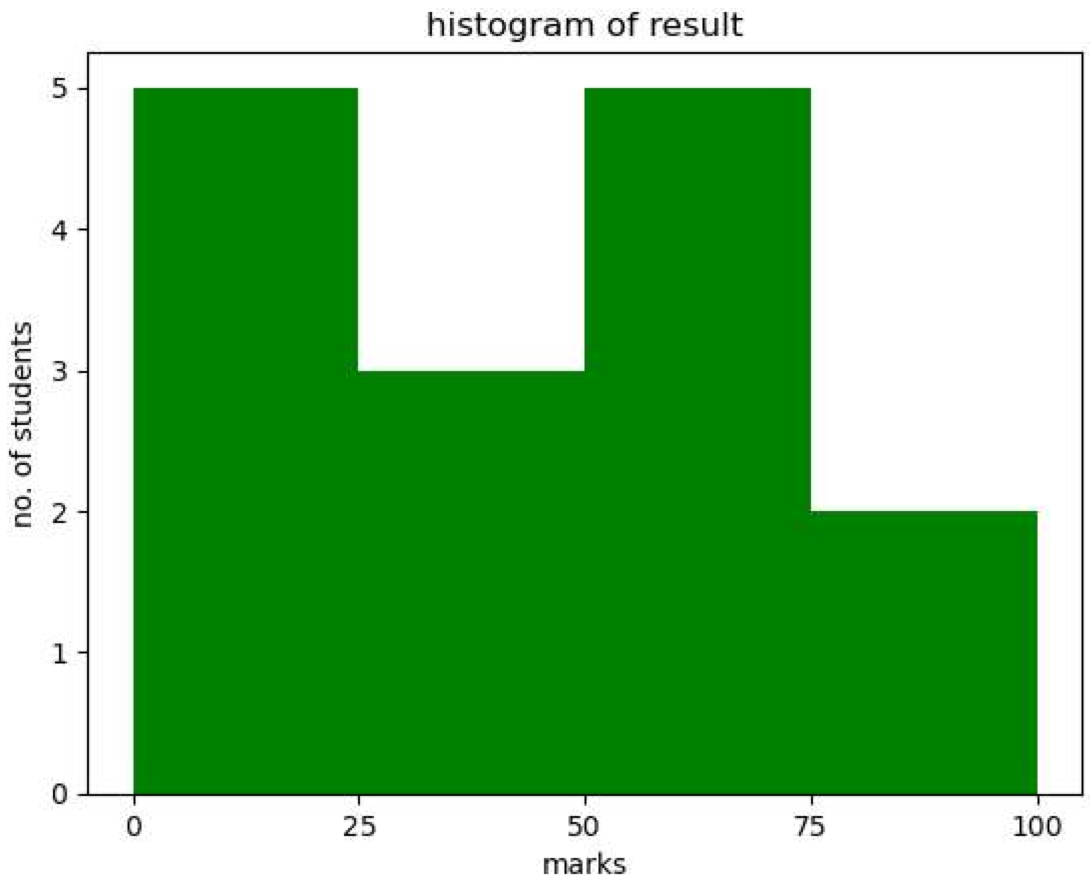


Histogram plot

First, we need to understand the difference between the bar graph and histogram. A histogram is used for the distribution, whereas a bar chart is used to compare different entities. A histogram is a type of bar plot that shows the frequency of a number of values compared to a set of values ranges.

For example we take the data of the different age group of the people and plot a histogram with respect to the bin. Now, bin represents the range of values that are divided into series of intervals. Bins are generally created of the same size.

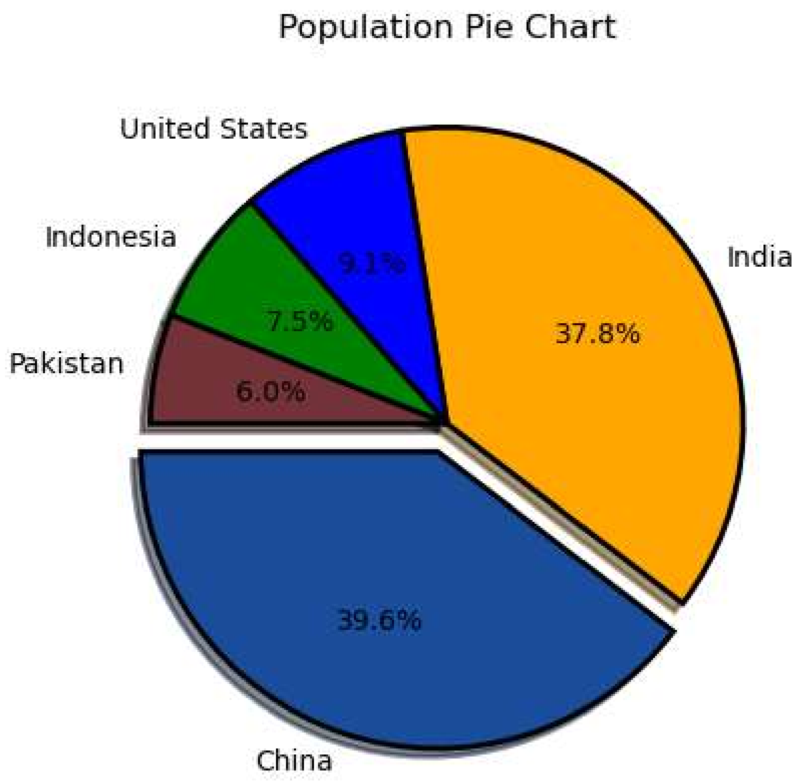
In [16]:
```
a = np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,31,27])
plt.hist(a, bins = [0,25,50,75,100],color="g")
plt.title("histogram of result")
plt.xticks([0,25,50,75,100])
plt.xlabel('marks')
plt.ylabel('no. of students')
plt.show()
```



pie plot

A pie chart is a circular graph that is broken down in the segment or slices of pie. It is generally used to represent the percentage or proportional data where each slice of pie represents a particular category

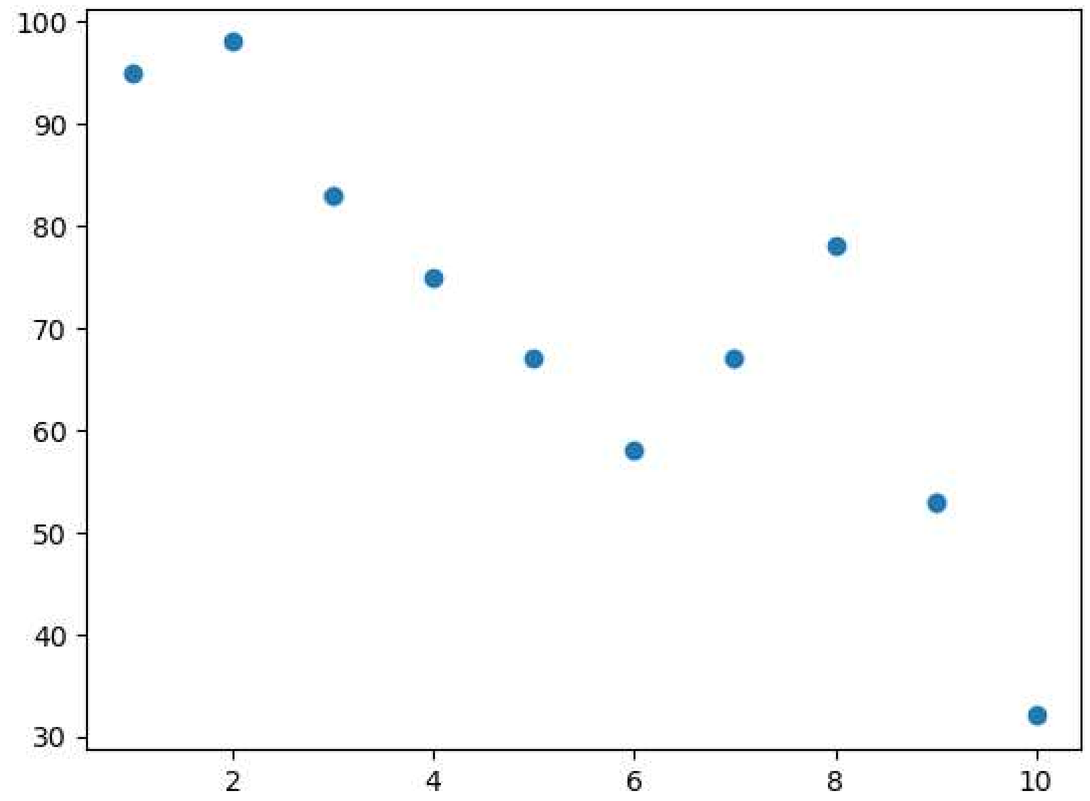
In [17]:
```
data = [1433783686, 1366417754, 329064917, 270625568, 216565318 ]

l = ['China', 'India', 'United States', 'Indonesia', 'Pakistan', ]

e = [0.1, 0.0, 0.0, 0.0, 0.0]

colors = ['#1a4c9c', 'orange', 'blue', 'green','#733238']

plt.pie(data, labels=l, explode=e, shadow=True, startangle=180,autopct="%1.1f%%",
        wedgeprops={'edgecolor':'black', 'linewidth': 2 },colors=colors)


plt.title("Population Pie Chart")
plt.show()
```



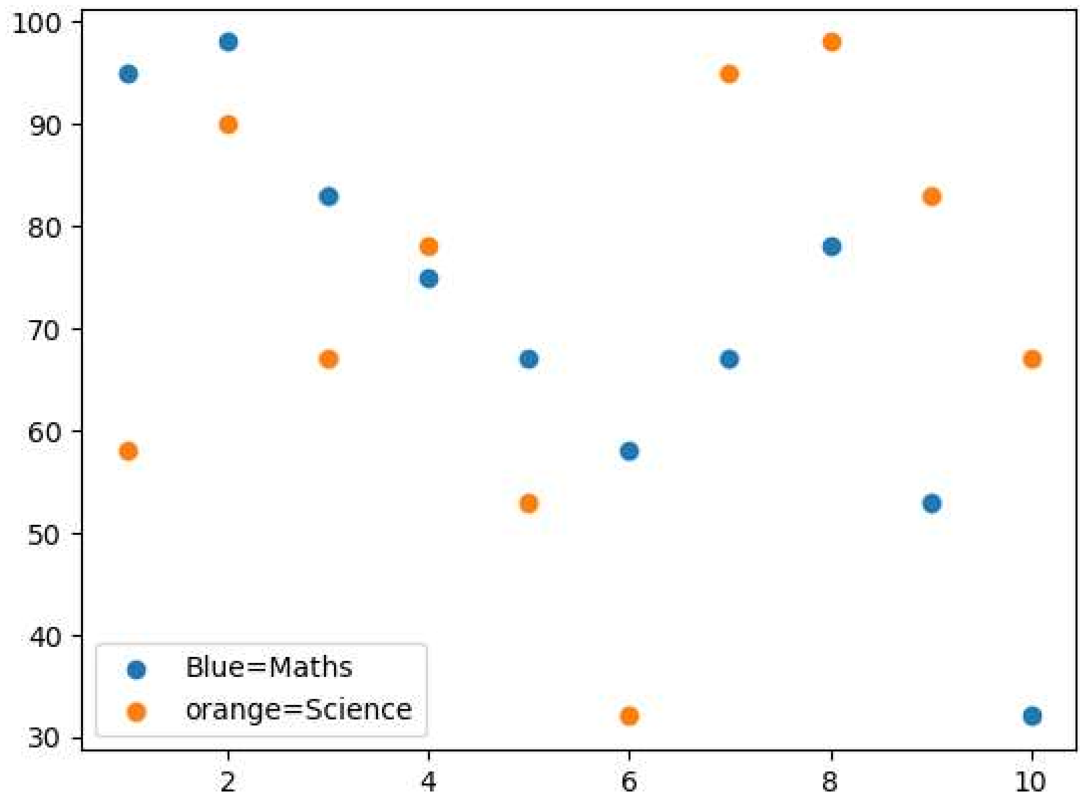
Scatter plot

The scatter plots are mostly used for comparing variables when we need to define how much one variable is affected by another variable. The data is displayed as a collection of points. Each point has the value of one variable, which defines the position on the horizontal axes, and the value of other variable represents the position on the vertical axis.

In [18]:
```
#importing library
import matplotlib.pyplot as plt
#datasets
students_id = [1,2,3,4,5,6,7,8,9,10]
students_marks = [95,98,83,75,67,58,67,78,53,32]
#scatter plot for the dataset
plt.scatter(students_id, students_marks)
plt.show()
```



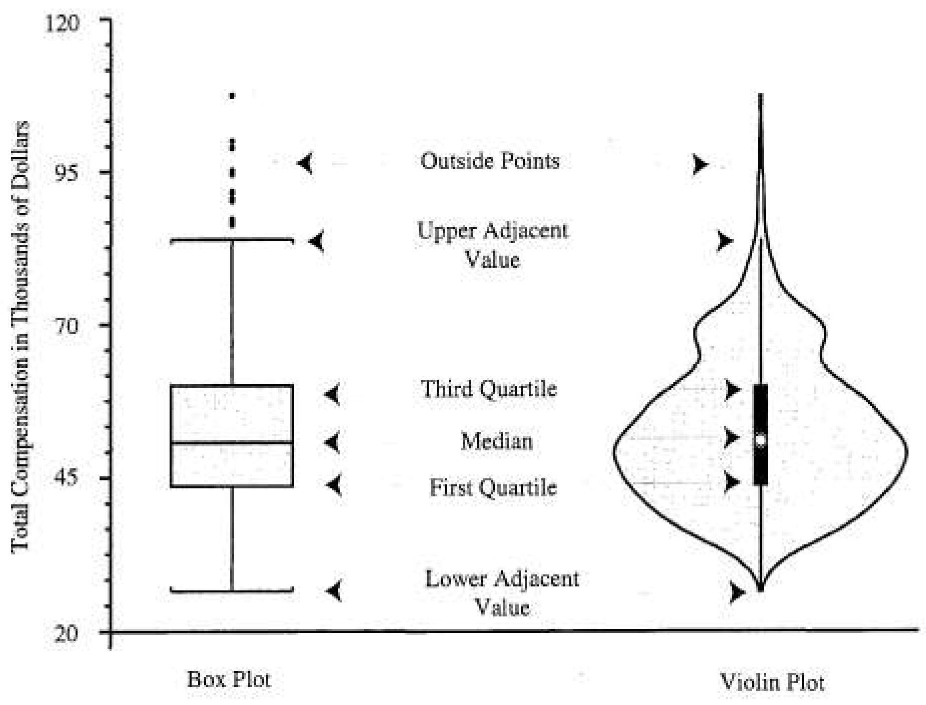
In [19]:
```
#Maths Marks
students_id = np.array([1,2,3,4,5,6,7,8,9,10])
students_marks = np.array([95,98,83,75,67,58,67,78,53,32])
plt.scatter(students_id, students_marks,label='Blue=Maths')

#science marks
students_id = np.array([1,2,3,4,5,6,7,8,9,10])
students_marks = np.array([58,90,67,78,53,32,95,98,83,67,])
plt.scatter(students_id, students_marks,label='orange=Science')
plt.legend()
plt.show()
```



These plots are mainly a combination of Box Plots and Histograms.

The box plot usually portrays the distribution, median, interquartile range of data.

box plot is used to find outliers in the dataset

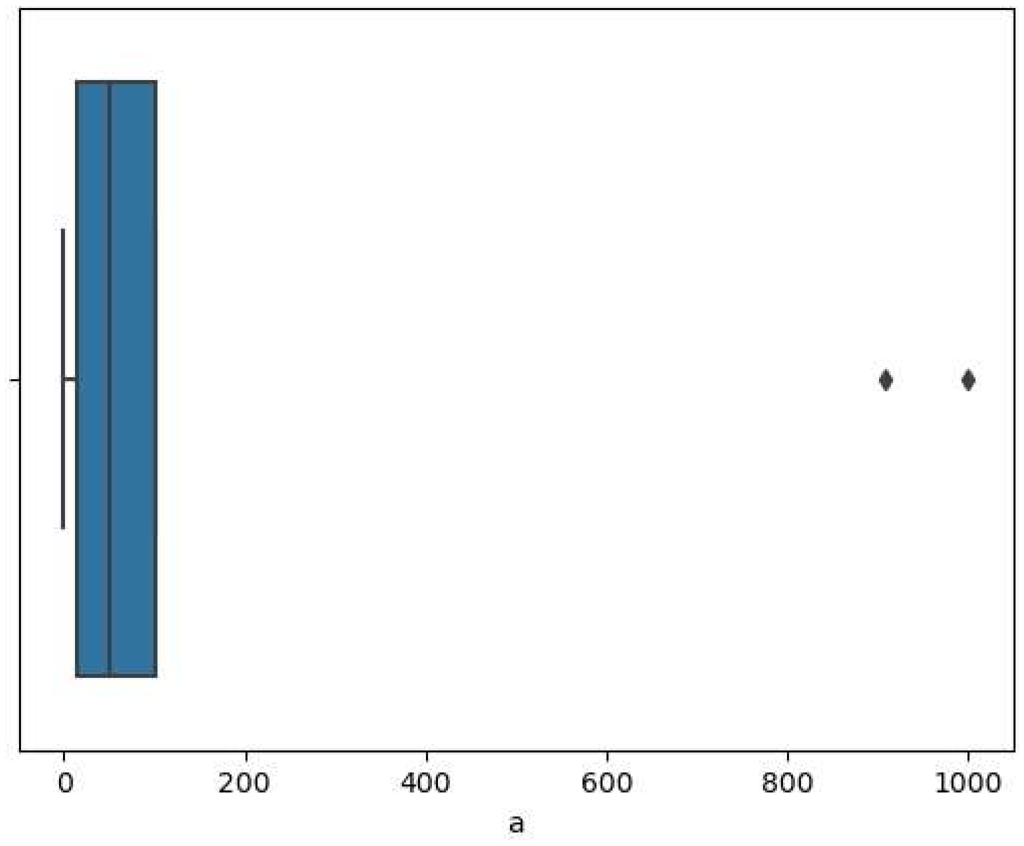


Box Plot

In [20]:
```
import seaborn as s
import pandas as pd
l=[0,1,50,60,50,14,909,1000,101]
df=pd.DataFrame(l,columns=['a'])
df
s.boxplot(df['a'])
```

```
C:\Users\Hi\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pas
s the following variable as a keyword arg: x. From version 0.12, the only valid posit
ional argument will be `data`, and passing other arguments without an explicit keywor
d will result in an error or misinterpretation.
  warnings.warn(
```

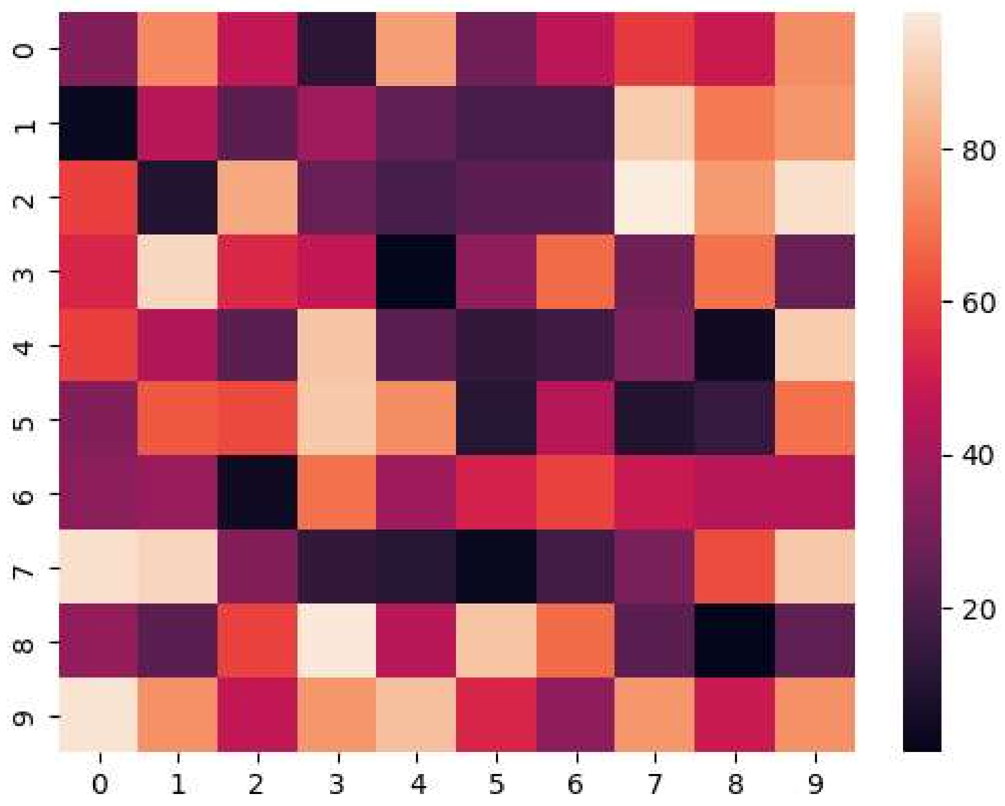
Out[20]:
```
<AxesSubplot:xlabel='a'>
```



In [22]:
```
# importing the modules
import numpy as np
import seaborn as sn
import matplotlib.pyplot as plt

# generating 2-D 10x10 matrix of random numbers
# from 1 to 100
data = np.random.randint(low = 1,high = 100,size = (10, 10))
print("The data to be plotted:\n")
print(data)

# plotting the heatmap
hm = sn.heatmap(data = data)
```

```
# displaying the plotted heatmap
plt.show()
```

```
The data to be plotted:

[[33 74 47 12 79 29 46 58 49 75]
 [ 3 45 24 40 25 19 19 90 71 77]
 [59 10 81 27 19 24 24 98 78 95]
 [53 93 54 47  1 36 68 29 69 27]
 [59 43 24 88 24 14 17 32  5 90]
 [33 64 61 89 75 11 44 10 15 69]
 [35 38  5 69 39 52 60 49 44 44]
 [95 92 33 14 11  3 18 31 62 89]
 [37 24 60 97 45 88 68 24  1 25]
 [96 76 47 77 87 53 36 77 49 76]]
```



```
In [ ]:
```

```
In [ ]:
```