What is Data Visualization

Data visualization is the technique to present the data in a pictorial or graphical format. It enables stakeholders and decision makers to analyze data visually. The data in a graphical format allows them to identify new trends and patterns easily.

The main benefits of data visualization are as follows:

- ➢ It simplifies the complex quantitativeinformation
- ➢ It helps analyze and explore big dataeasily
- ➢ It identifies the areas that need attention orimprovement
- ➢ It identifies the relationship between data points and variables
- ➢ It explores new patterns and reveals hidden patterns in the data

Using Pyplot of MATPLOTLIB Library

- The matplotlib is a python library that provides many interfaces functionally for 2D graphics similar to MATLAB's in variousform.
- In short we can call mattplotlib as a high quality plotting library of Python.
- The matplotlib library offers many different named collections of methods, PyPlotis one suchinterface.
- PyPlot is a collection of methods within matplotlib which allows user toconstruct 2D plots easily and interactively.

Importing PyPlot

To import Pyplot following syntax is

```
import matplotlib.pyplot
```

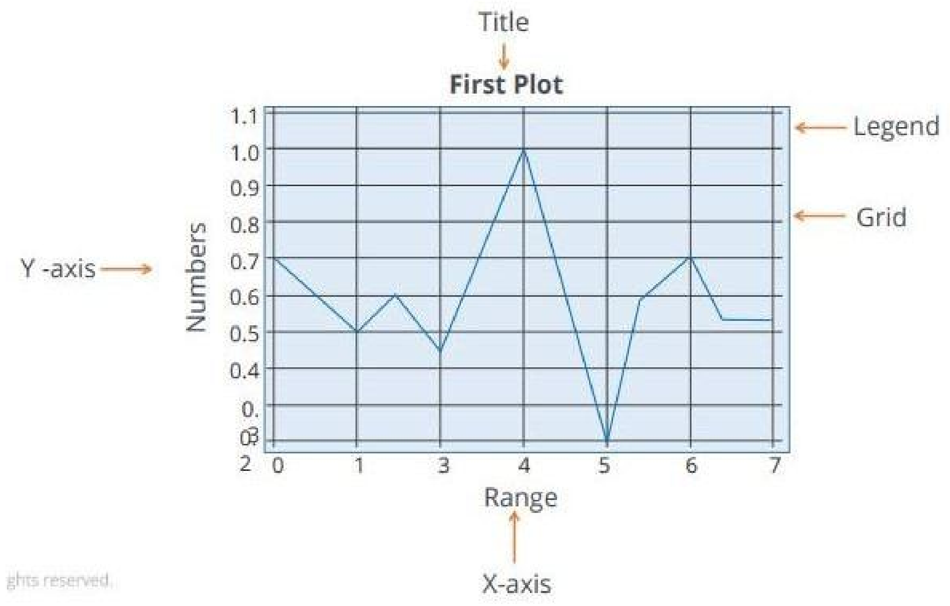
or

```
import matplotlib.pyplot as pl
```

After importing matplotlib in the form of pl we can use pl for accessing any function of matplotlib.

We can create a plot using four simple steps:-

1. Import the required libraries
2. Define or import the required dataset
3. Set the plotparameters
4. Display the createdplot

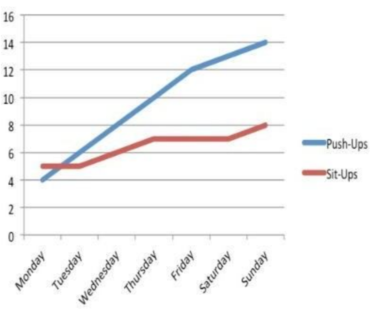


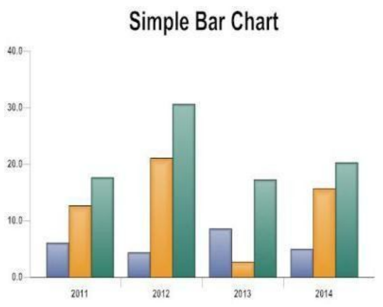
Following chart types

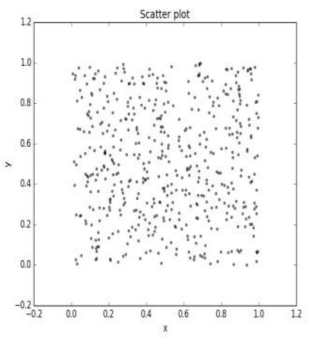
– Linechart

– BarChart

– ScatterPlot





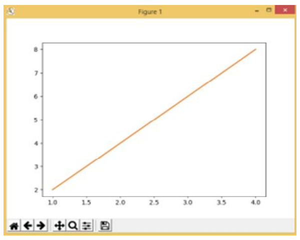


Line Chart using plot function ()

- A line chart or line graph is a type of chart which displays information as a series of data points called markers connected by straight linesegments.

The example codes given below to understand the working of plot().

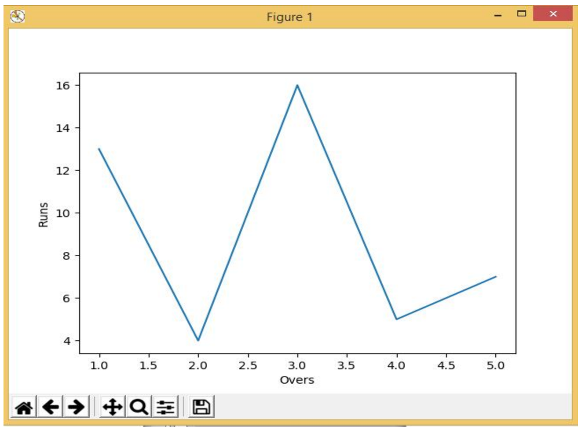
```
>>> import matplotlib.pyplot as pl
>>> a=[1,2,3,4]
>>> b=[2,4,6,8]
>>> pl.plot(a,b)
```



In the next example we take data from a cricket match where

runs of 5 overs are given. We place name of X axis as overs and name of Y axis as runs.

```
import matplotlib.pyplot as pl
over = [1,2,3,4,5]
run = [13,4,16,5,7]
pl.xlabel("Overs")
pl.ylabel("Runs")
pl.plot(over,run)
pl.show()
```

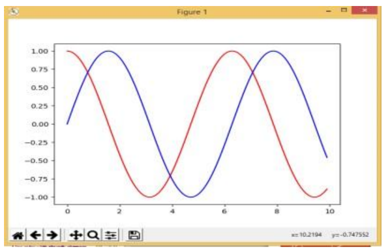


Change of Line color, width and style

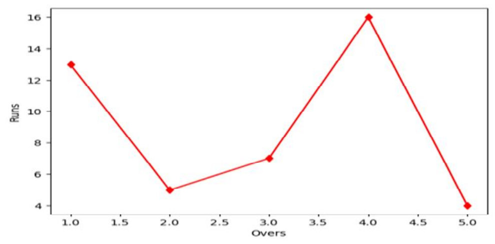
We can use following syntax -

matplotlib.pyplot.plot(<data1>,<data2>,<color code>)

```
import matplotlib.pyplot as pl
import numpy as np
x=np.arange(0,10,0.1)
a=np.cos(x)
b=np.sin(x)
pl.plot(x,a,'r')
pl.plot(x,b,'b')
pl.show()
```



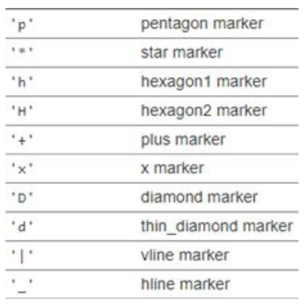
character	color
'b'	blue
'g'	green
'r'	red
'c'	cyan
'm'	magenta
'y'	yellow
'k'	black
'w'	white



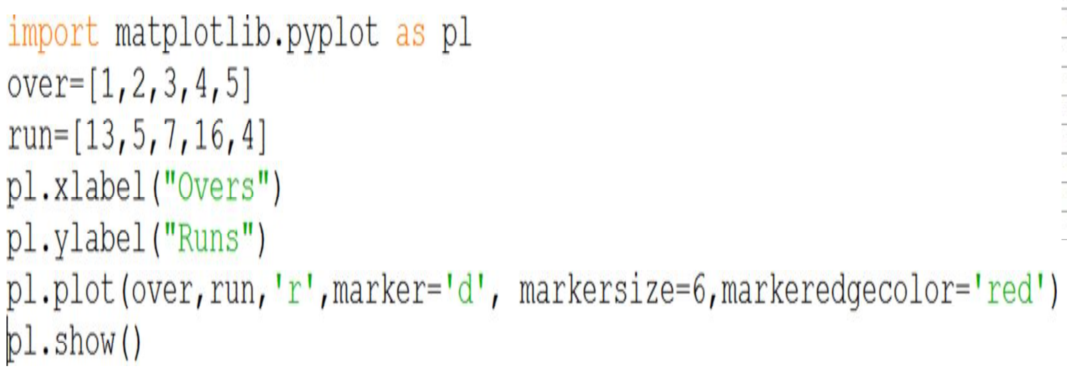
Change of Marker type, size and color

Use following -

matplotlib.pyplot.plot(<data1>,<data2>,linestyle=<val>…)



'p'	pentagon marker
'*'	star marker
'h'	hexagon1 marker
'H'	hexagon2 marker
'+'	plus marker
'x'	x marker
'D'	diamond marker
'd'	thin_diamond marker
'|'	vline marker
'_'	hline marker



```
import matplotlib.pyplot as pl
over=[1,2,3,4,5]
run=[13,5,7,16,4]
pl.xlabel("Overs")
pl.ylabel("Runs")
pl.plot(over,run,'r',marker='d', markersize=6,markeredgecolor='red')
pl.show()
```

The scatter chart is a graph of plotted points on two axes that show the relationship between two sets of data.

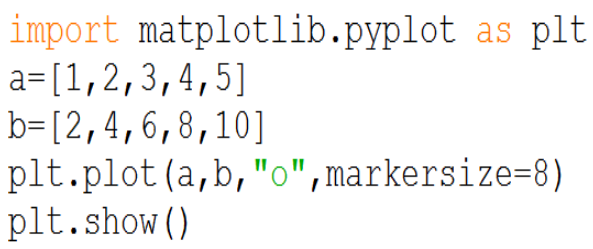
The scatter charts can be created through two functions of pyplot library:

– plot( )function

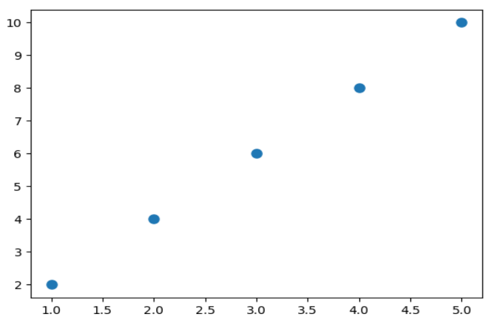
– scatter( )function

Syntax of plot() function is –

matplotlib.pyplot.plot(a,b,<point style >, markersize=<value>)

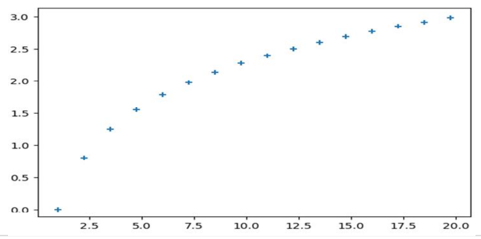


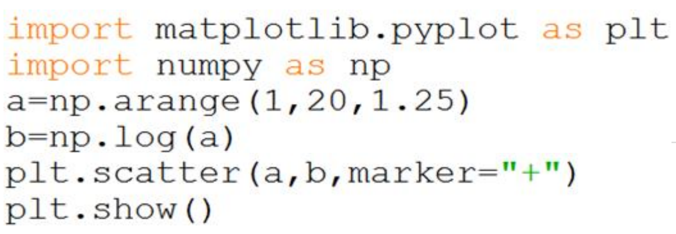
```
import matplotlib.pyplot as plt
a=[1,2,3,4,5]
b=[2,4,6,8,10]
plt.plot(a,b,"o",markersize=8)
plt.show()
```



Creating Scatter Chart

Syntax of scatter () function –
matplotlib.pyplot.scatter(a, b, marker=<type>)





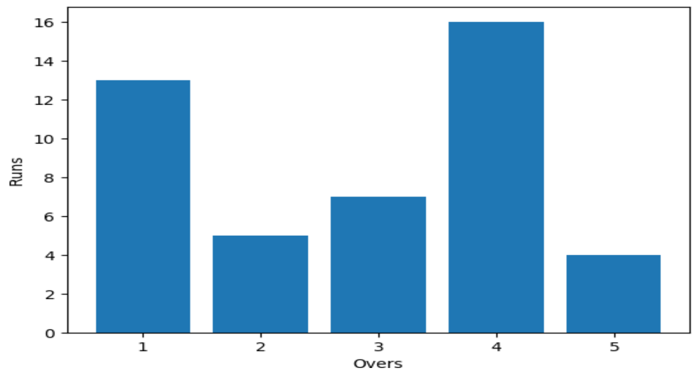
```
import matplotlib.pyplot as plt
import numpy as np
a=np.arange(1,20,1.25)
b=np.log(a)
plt.scatter(a,b,marker="+")
plt.show()
```

Bar Chart:-

A Bar Graph / Chart is a graphical display of data using bars of different heights.

syntax – matplotlib.pyplot.bar(a,b)

```python
import matplotlib.pyplot as pl
over=[1,2,3,4,5]
run=[13,5,7,16,4]
pl.xlabel("Overs")
pl.ylabel("Runs")
pl.bar(over,run)
pl.show()
```



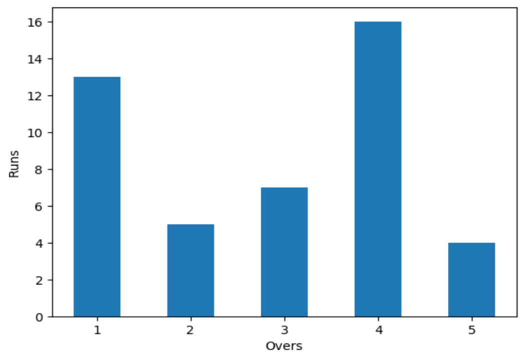
Changing widths of the Bars in a Bar Chart

To specify common width (other than the default width) for all bars we can specify width argument having a scalar float value in the bar() function.

syntax –

matplotlib.pyplot.bar(a, b, width=<Value>)

```python
import matplotlib.pyplot as pl
over=[1,2,3,4,5]
run=[13,5,7,16,4]
pl.xlabel("Overs")
pl.ylabel("Runs")
pl.bar(over,run,width=1/2)
pl.show()
```

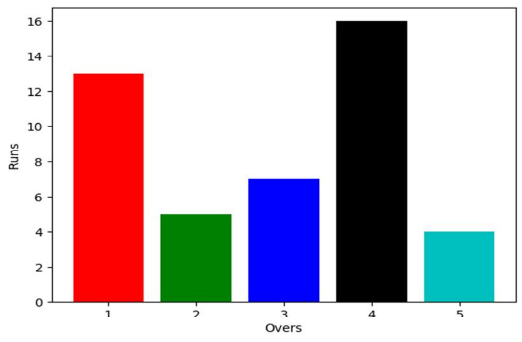


To specify common color (other than the default color) for all bars we can specify color argument having a valid color code/name in the bar() function.

Syntax –

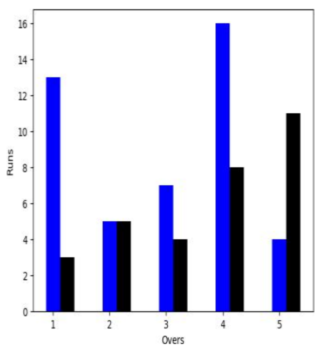
matplotlib.pyplot.bar(a, b, color=<code>)

```python
import matplotlib.pyplot as pl
over=[1,2,3,4,5]
run=[13,5,7,16,4]
pl.xlabel("Overs")
pl.ylabel("Runs")
pl.bar(over,run,color =['r','g','b','k','c'])
pl.show()
```

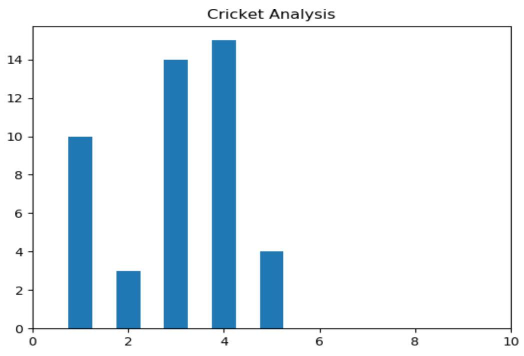


Creating Multiple Bar Chart

```
import matplotlib.pyplot as pl
import numpy as np
over=np.arange(1.0,6.0,1.0)
Ind=[13,5,7,16,4]
Nz=[3,5,4,8,11]
pl.xlabel("Overs")
pl.ylabel("Runs")
pl.bar(over,Ind,color ='b',width=0.25)
pl.bar(over+0.25,Nz,color='k',width=0.25)
pl.show()
```

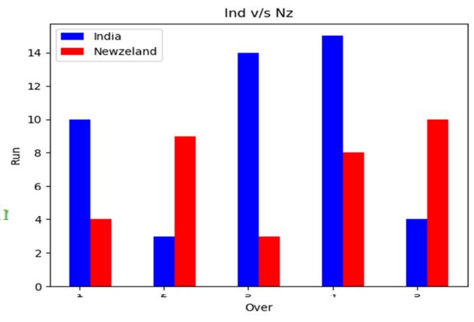


```
import matplotlib.pyplot as pl
import numpy as np
over=[1,2,3,4,5]
run=[10,3,14,15,4]
pl.xlim(0,10)
pl.title("Cricket Analysis")
pl.bar(over,run,width=1/2)
pl.show()
```



Adding Legends

```
import matplotlib.pyplot as pl
import numpy as np
over=np.arange(1.0,6.0,1.0)
ind=[10,3,14,15,4]
nz=[4,9,3,8,10]
pl.title("Ind v/s Nz")
pl.bar(over,ind,color='b',width=0.25,label='India')
pl.bar(over+0.25,nz,color='r',width=0.25,label='Newzela
pl.legend(loc='upper left')
pl.xlabel("Over")
pl.ylabel("Run")
pl.show()
```



Saving a Graph in a local disk:-

```
import matplotlib.pyplot as pl
import numpy as np
over=np.arange(1.0,6.0,1.0)
ind=[10,3,14,15,4]
nz=[4,9,3,8,10]
pl.title("Ind v/s Nz")
pl.bar(over,ind,color='b',width=0.25,label='India')
pl.bar(over+0.25,nz,color='r',width=0.25,label='Newzeland')
pl.legend(loc='upper left')
pl.xlabel("Over")
pl.ylabel("Run")
pl.savefig("C:\\MyData\\myfig.png")
pl.show()
```

Check Point:-

1. Which of the following is not a valid plotting function of pyplot?.

    a. Plot() b.bar()

    c. line() d.pie()

Ans:-c

2. Which of the following plotting functions does not plot multiple dataservices?.

    a.plot() b. bar()

    c.pie() d.barh()

Ans:-pie()

3. Name the function you will use to create a horizontal barchart.

Ans:-barh()

4. What is Marker? How can you change the marker type and colour in a plot ?.

Ans:- Data points being plotted are calledmarkers.

Change of Marker type, size and color use following -

matplotlib.pyplot.plot(<data1>,<data2>,linestyle=<val>…)

5. What is the use of **loc** argument in a legend()function.

Ans:- It specifies the keyword argument loc determines where the legend will be placed in a graph, which by default is 1 or “upper right”.

6. Compare bar() and barh()functions.

Ans:- A bar graph displays the values in a vector or matrix as horizontal or vertical bars. A barh graph displays the values in a vector or matrix as horizontal bars.

7. A Data Frame pdas:

	1990	2000	2010
1.	54	345	895
2.	64	485	562
3.	79	690	1100
4.	96	770	890

Write Code to Create:-

(a) A scatter chart from 1990 and 2010 of data framepd.

(b) A line chart from the 1990 and 2000 of data framepd.

(c) A bar chart to plotting the three columns of data framepd.

Ans:-(a)

```
        import matplotlib.pyplot aspl
        import pandas as pd
    data = {'1990':[54,64,79,96],'2000':[345,485,690,770],'2010':[895,562,1100,890]}
    d = pd.DataFrame(data,columns={'1990','2000','2010'})
    pl.scatter(d['1990'],d['2010'])
    pl.show()
```

(b)

```
        import matplotlib.pyplot aspl
        import pandas aspd
    data = {'1990':[54,64,79,96],'2000':[345,485,690,770],'2010':[895,562,1100,890]}
    d = pd.DataFrame(data,columns={'1990','2000','2010'})
    pl.plot(d['1990'],d['2000'])
    pl.show()
```

(c)

```
        import matplotlib.pyplot as pl
        import pandas aspd
    data = {'1990':[54,64,79,96],'2000':[345,485,690,770],'2010':[895,562,1100,890]}
    d = pd.DataFrame(data,columns={'1990','2000','2010'})
    pl.bar(d['1990'],d['2000'],d['2010'])
    pl.show()
```

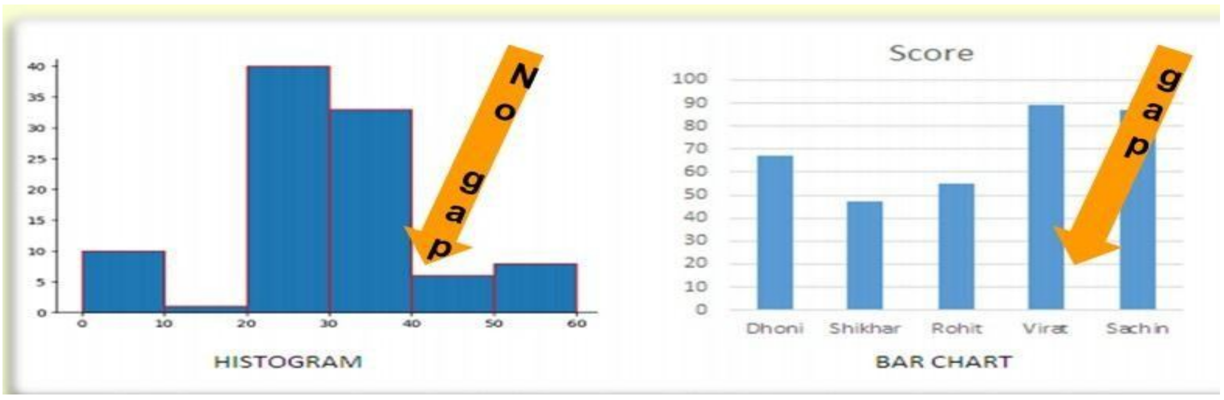
Histogram:-

A histogram is a powerful technique in data visualization. It is an accurate graphical representation of the distribution of numerical data. It was first introduced by Karl Pearson.

It is an estimate of the distribution of a continuous variable (quantitative variable). It is similar to a bar graph. To construct a histogram, the first step is to “bin” the range of values—means divide the entire range of values in to a series of intervals—and then count how many values fall into each interval. The bins are usually specified as consecutive, no over lapping intervals of a variable. The bins(intervals) must be adjacent, and are often(but are not required to be) of equalsize.

Difference between a histogram and a bar chart / graph –

A bar chart majorly represents categorical data (data that has some labels associated with it), they are usually represented using rectangular bars with lengths proportional to the values that they represent. While histograms on the other hand, is used to describe distributions. Given a set of data, what are their distributions?



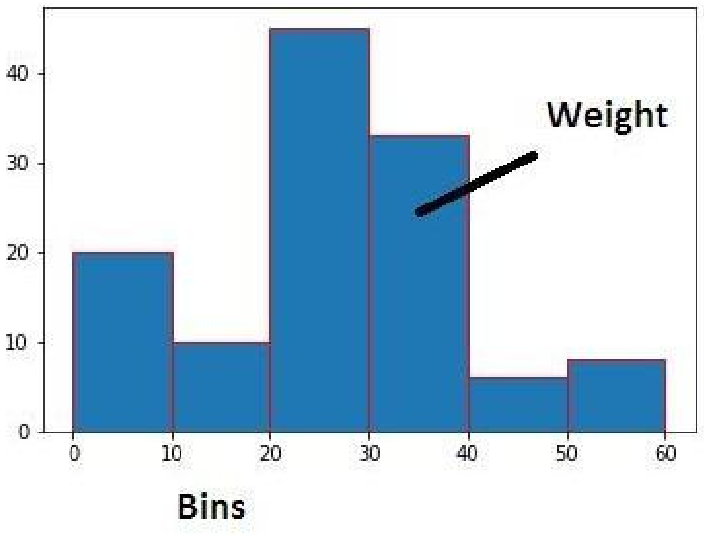
There are various ways to create histogram in python pandas. One of them is using matplotlib python library. Using this library we can easily create histogram. We have to write just few statements to create histogram.

Program in python. Develop a python program with the code below and execute it.

```
import numpy as np
import matplotlib.pyplot as plt
data = [1,11,21,31,41]
```

```
 plt.hist([5,15,25,35,45, 55], bins=[0,10,20,30,40,50, 60],
weights=[20,10,45,33,6,8], edgecolor="red")
plt.show()
```

Output:-

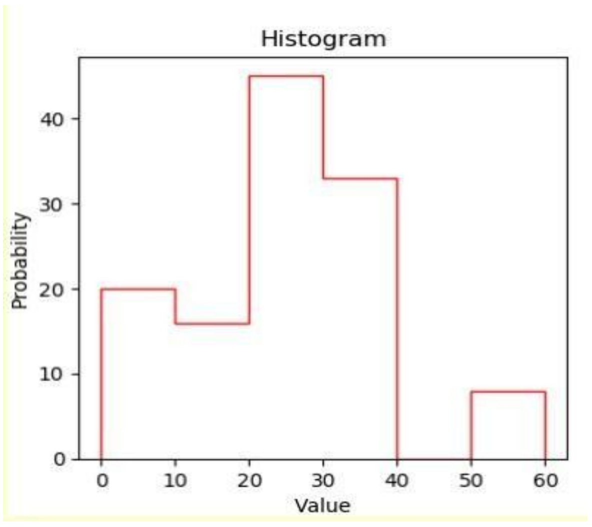


Frequency polygons:-

A frequency polygon is a type of frequency distribution graph. In a frequency polygon, the number of obervations is marked with a single point at the midpoint of an interval.

```
 import numpy as np
 import matplotlib.pyplot as plt
data = [1,11,21,31,41]
plt.hist([5,15,25,35,15, 55], bins=[0,10,20,30,40,50, 60], weights=[20,10,45,33,6,8],
edgecolor="red",histtype='step')
plt.xlabel('Value')
plt.ylabel('Probability')
plt.title('Histogram')
plt.show()
```

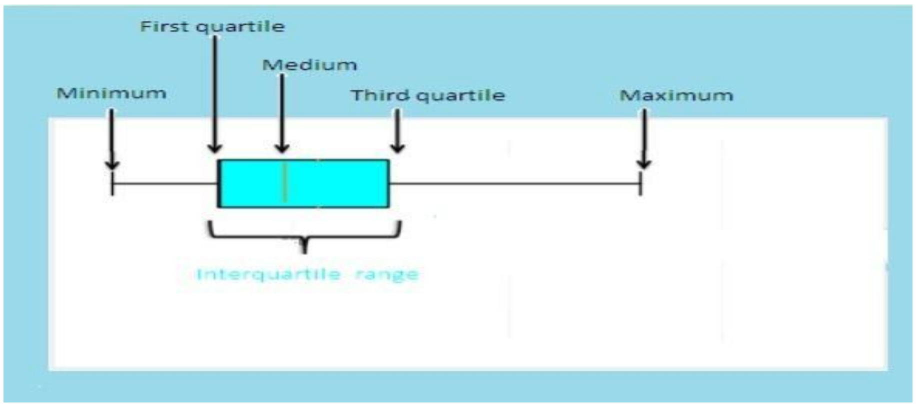
Output is:-



Box Plots:-

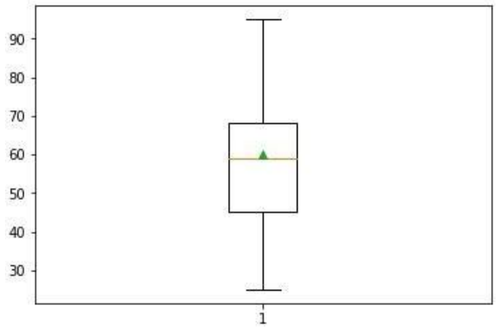
A Box Plot is the visual representation of the statistical five number summary of a given data set. A Five Number Summary includes:

- Minimum
- FirstQuartile
- Median (Second Quartile)
- ThirdQuartile
- Maximum



1. Draw the boxplotfrom the following data:- 25,42,63,45,59,56,87,68,95

Ans:-



```
import matplotlib.pyplot as plt
Data=[25,42,63,45,59,56,87,68,95]
plt.boxplot(Data,showmeans=True)
plt.show()
```

2. Which function of Pyplot can you plot histograms?

Ans:- hist()

3. What is the use of boxplot?

Ans:- A box plot or boxplot is a method for graphically depicting groups of numerical data through their quartiles