Chapter 3: Plotting with PyPlot

Data Visualization

- The role of data is to empower decision makers to make decisions based on facts ,trends and statistical numbers available in the form of data. But since data is so huge these days that the decision makers must be able to sift through the unnecessary, unwanted data and get the right information presented in compact and right way,so that best decisions taken.

- Data visualization basically refers to the graphical or visual representation of information and data using visual elements like charts, graphs and maps etc.

  - Data visualization is immensely useful in decision making

- Data visualization unveils patterns , trends, outliers, correlations etc.Helps decision makers to understands the meaning of data to drive business decisions.

Using Pyplot of Matplotlib Library

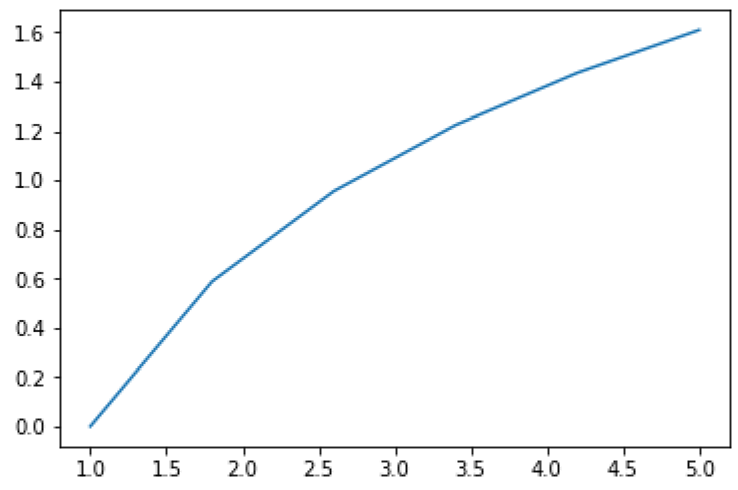
- **For** data visualization in Python , the Matplotlib library's Pyplot interface is used.
- **PyPlot is a collection of methods within matplotlib library(of Python)Which allows user to construct 2D plots easily and interactively.**

  - The matplotlib library is preinstalled with Anaconda distribution.

Working with PyPlot Methods

- PyPlot interface provides many methods for 2D plotting of data.
- The matplotlib's Pyplot interface lets one plot data in multiple ways such as line chart, bar chart, pie chart, scatter chart etc.
- Plotting a simple chart using an ndarray.

```
In [1]: import numpy as np

In [2]: import matplotlib.pyplot as pl
In [3]: x=np.linspace(1,5,6)

In [4]: y=np.log(x)

In [5]: pl.plot(x,y)

Out[5]: [<matplotlib.lines.Line2D at 0x2231e40a208>]
```



Basic of Simple Plotting

- As we know that Data visualization essentially means graphical representation of compiled data.
- We can create many different types of graphs and charts using PyPlot.
- Different type of chart types are

Line chart:- A line chart or line graph is atype of chart which displays information as a series of data points called ‘markers’ connected by straight line segments.

Bar Chart:- A bar chart or bar graph is a chart or graph that presents categorical data with rectangular bars with heights or lengths proportional to the values that they represent. The bars can be plotted vertically or horizontally.With PyPlot, a bar chart is created using bar() and barh() functions.

Scatter Plot:- The scatter plot is similar to a line chart , the major difference is that while line graph connects the data points with a line, scatter chart simply plots the data points to show the trend in the data. With PyPlot, a scatter chart is created using scatter() function.

Pie Chart:- A pie chart (or a circle chart) is a circular statistical graphic , which is divided into slices to illustrate numerical proportion. With PyPlot, a pie chart is created using pie() function.But pie chart can plot only one data sequence unlike other chart types.

Histogram Plot:- A histogram is a type of graph that provides a visual interpretation of numerical data by indicating the number of data points that lie within a range of values. With Pyplot, a histogram is created using hist() function.

BoxPlot Chart:- A box plot is the visual representation of the statistical five number summary of a given data set. With PyPlot, a boxplot is created using boxplot() function.

Creating Line charts and scatter charts".

- Before we plot any chart or graph type, make sure to import the matplotlib.pyplot library

  Import matplotlib.pyplot

  Import matplotlib.pyplot as pl

- Galong with pyplot, if we are using any NumPy or Pandas functionality, make sure to import them too using import commands.

Line chart using plot() Function

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

Eg

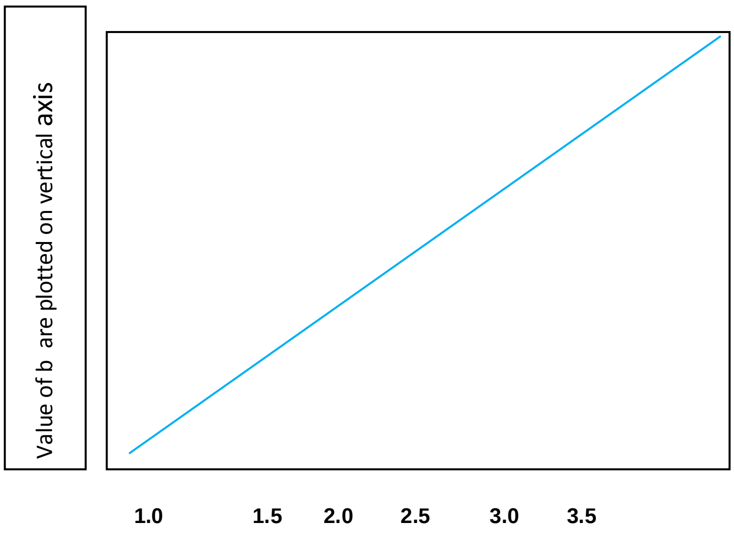
>>> a=[1,2,3,4]

>>>b=[2,4,6,8]

>>> import matplotlib.pyplot as pl

>>>pl.plot(a,b)

>>>pl.plot(a,b)



4 .0 (value of a)

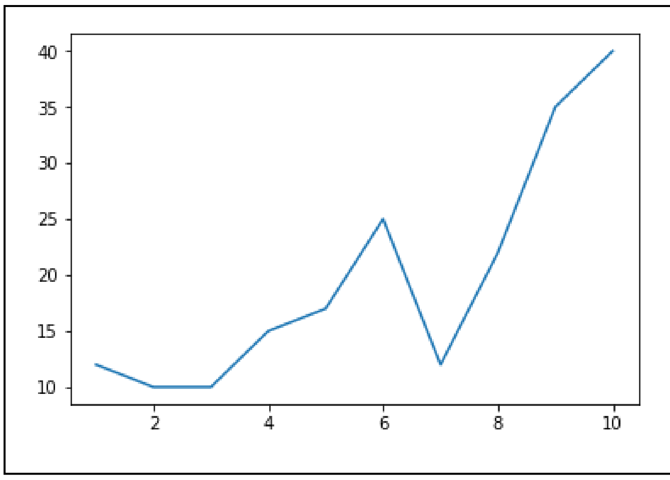
A program to plot a line chart to depict the changing weekly onion prices for four weeks. "

- Solution

```
Import matplotlib.pyplot as pl

Week=[1,2,3,4]

Prices=[40,80,100,50]

Pl.plot(week,Prices)

Pl.xlabel('week')

Pl.ylabel('Onion Prices(Rs.))

 Pl.show()
```

A program that plot the students performance of marks list which stores marks of a student in 10 unit tests.”

```
Import matplotlib. Pyplot as pl

week=[1,2,3,4,5,6,7,8,9,10]

Marks=[12,10,10,15,17,25,12,22,35,40]

Pl.plot(week,marks)

Pl.xlabel(‘week’)

Pl.ylabel(‘Unit test marks’)

Pl.show
```



APPLYING VARIOUS SETTINGS IN PLOT()

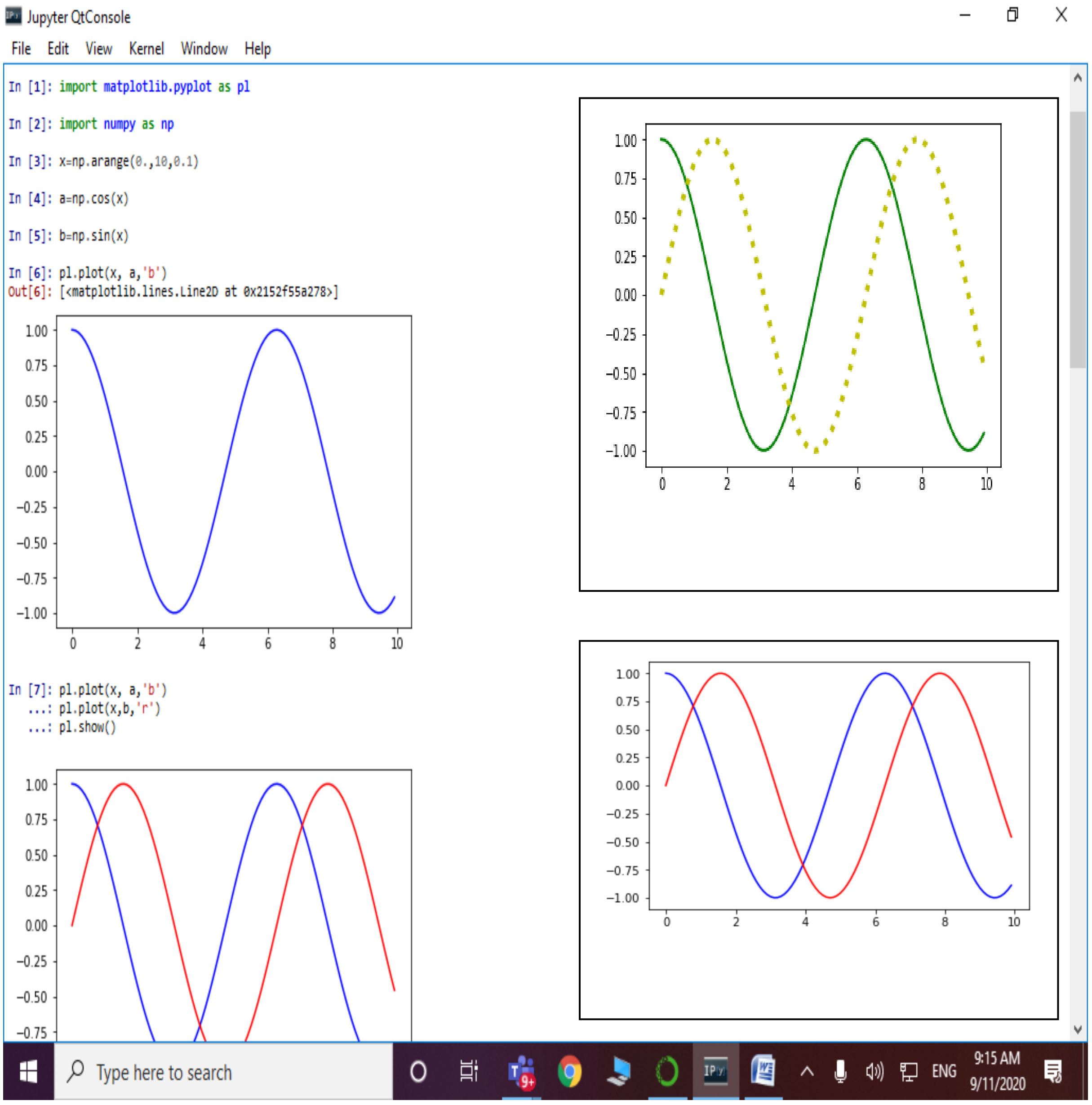
Color

Marker type

Marker size

CHANGING LINE COLOUR AND STYLE

1. <matplotlib.pyplot>.plot(<data1>,[,data2],<color code>)
2. We can use color codes
   as ‘r’ for red
   ‘y’ for yellow
   ‘g’ for green
   ‘b’ or blue etc
3. Type of line style
   - Solid
   - Dashed
   - Dashdot
   - Dotted



```
In [1]: import matplotlib.pyplot as pl

In [2]: import numpy as np

In [3]: x=np.arange(0.,10,0.1)

In [4]: a=np.cos(x)

In [5]: b=np.sin(x)

In [6]: pl.plot(x, a,'b')
Out[6]: [<matplotlib.lines.Line2D at 0x2152f55a278>]
```

```
In [7]: pl.plot(x, a,'b')
   ...: pl.plot(x,b,'r')
   ...: pl.show()
```

Tanushree is doing some research. She has a stored line of Pascal's

Triangle numbers as ar2 as shown below:

ar3=[1,7,21,35,35,21,7,1]

She wants to plot the sine (numpy.sin()),cosine(numpy.cos))and tangent

values (nump.tan()) for the same array(aar2).

- She wants cyan color for sine plot line, red color for cosine plot line and the black color for tangent plot line.
- Also, the tangent line should be dashed.

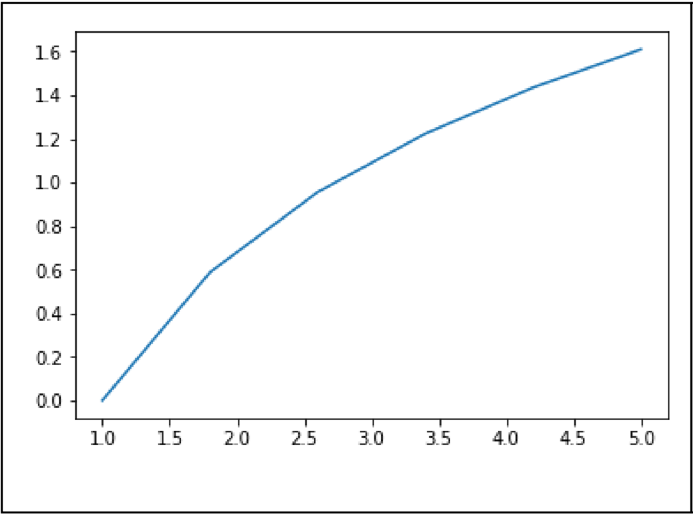
Write a program to accomplish all this.

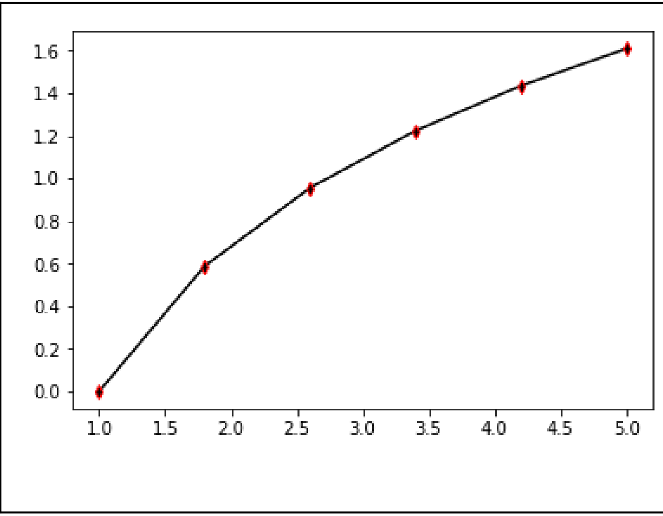
Solution

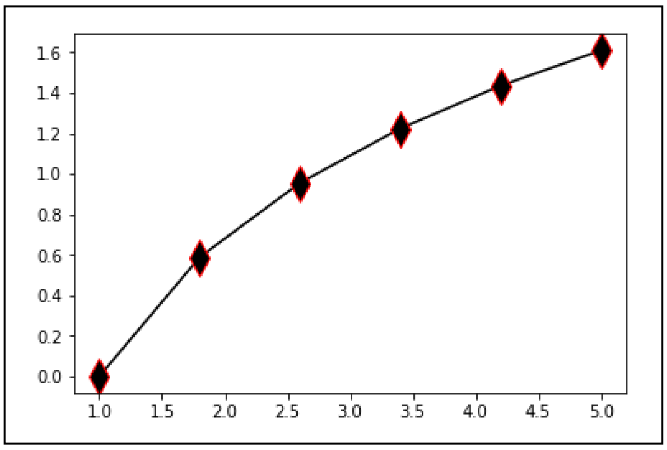
```
import matplotlib.pyplot as pl

import numpy as np

ar2=[1,7,21,35,35,21,7,1]

s2=np.sin(ar2)

c2=np.cos(ar2)

t2=np.tan(ar2)

pl.figure(figsize=(15,7))

pl.plot(ar2,s2,’c’)

pl.plot(ar2,c2,’r’)

pl.plot(ar2,t2,’k’,linestyle= “dashed”)

pl.xlabel(‘Array values’)

pl.ylabel(‘Sine, Cosine and Tangent Values’)

pl.show()
```

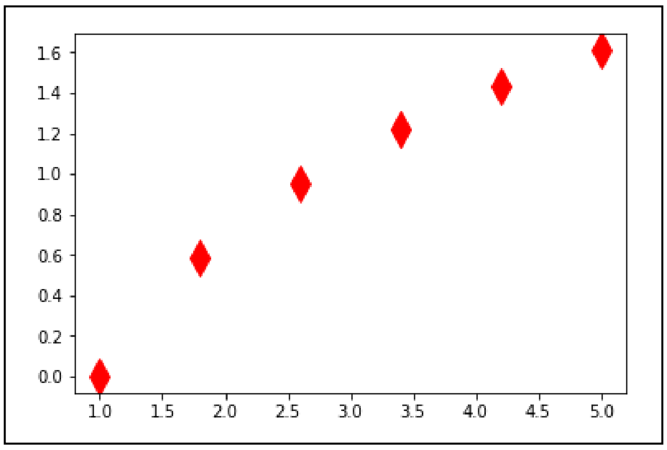
Changing Marker Type, Size and color

- The data points being plotted on a graph/chart are called markers.
- To change marker type, its size and color .
- marker=<valid marker type>
- markersize=<in points>
- markeredgecolor=<valid color>









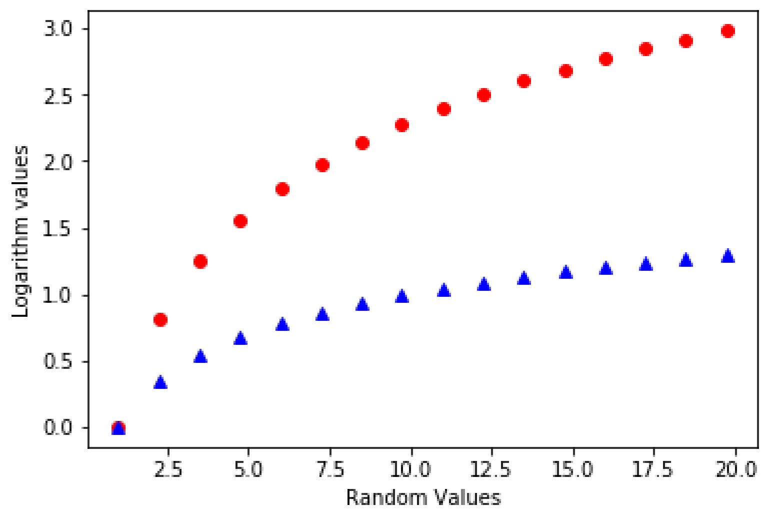
Creating Scatter Charts :

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

- **The scatter charts can be created through 02 functions of PyPlot library:**

(i) Plot() function
(ii) Scatter() function

```
In [1]: import numpy as np

In [2]: import matplotlib.pyplot as pl

In [3]: A=np.arange(1,20,1.25)

In [4]: B=np.log(A)

In [5]: C=np.log10(A)

In [6]: pl.plot(A,B,'ro')


...: pl.plot(A,C,'b^')
   ...: pl.xlabel('Random Values')
   ...: pl.ylabel('Logarithm values')
   ...: pl.show()
```



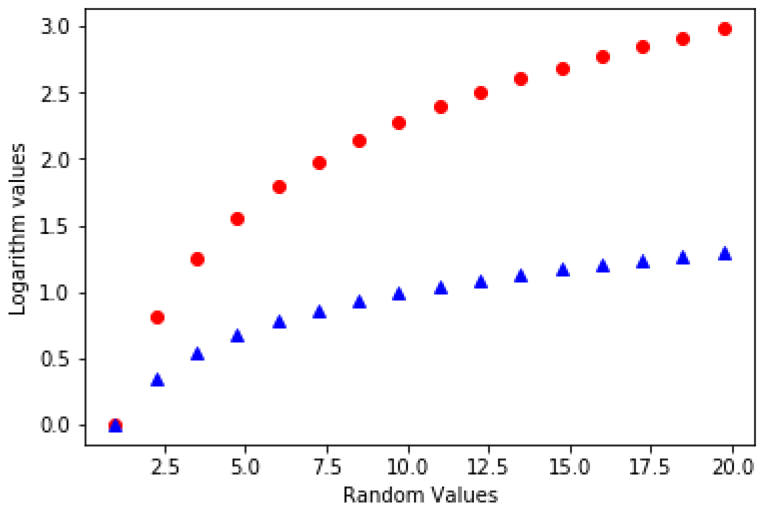
Creating Scatter Charts :

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

- **The scatter charts can be created through 02 functions of PyPlot library:**

(iii) Plot() function
(iv) Scatter() function

```
In [1]: import numpy as np

In [2]: import matplotlib.pyplot as pl

In [3]: A=np.arange(1,20,1.25)

In [4]: B=np.log(A)

In [5]: C=np.log10(A)

In [6]: pl.plot(A,B,'ro')


...: pl.plot(A,C,'b^')
   ...: pl.xlabel('Random Values')
   ...: pl.ylabel('Logarithm values')
   ...: pl.show()
```



Creating Bar Charts:

- A Bar Graph or a Bar Chart is a graphical display of data using bars of different heights.
- A bar chart can be drawn vertically or horizontally using rectangles or bars of diiferent heights/widths.
- PyPlot offers bar() function to create a bar chart where we can specify the sequences for x-axis and corresponding sequence to be plotted on y-axis.Each y value is plotted as bar on corresponding x-value on x-axis.
- Example

```
>>> Cities =['Delhi','Mumbai','Banglore','Hyderabad']
>>> Population =[23456123,20083104,18456123,13411093]
>>>plt.bar(Cities,Population)
>>>pl.xlabel('Cities')
>>>pl.ylabel('Population')
>>>pl.show()
```

Note:- The order of bars plotted may bedifferent from the order in actual data Sequence.

Changing Widths of the Bars in a Bar Chart:

(i)Changing Width

(ii)Colour of Bars chart

(i)Changing Width

(a) By default, bar chart draws bars with equal widths and having a

Default width f 0.8 units on a bar chart.

(b) We can specify a different width (other than the default width)

For all the bars of a bar chart.

(c) We can also specify different widths for different bars of a bar chart.

(ii) Changing Colors of the bars in a Bar Chart .

(a) By default , a bar chart draws bars with same default color.

(b) We can specify a different color for all the bars of a bar chart.

```
(c) Example :   Write a program to plot a barchart from the medals won by Australia.
    Make sure that the Gold, Silver,Broze and Total tally is represented through
    different widths
(d)
(e) >>> import matplotlib.pyplot as pl
(f) >>> Info=[“Gold”,”Silver”,”Bronze”,”Total”]
(g) >>> Australia =[80,59,59,198]
(h) >>> pl.bar(Info,Australia, width=[0.7,0.5,0.3,1])
(i) >>> pl.xlabel(“Medal type”)
(j) Pl.ylabel(“Austrailia Medial Count”)
(k) Pl.show()
```

Creating Multiple Bars Chart

Steps :-

(i) Decide number of x points

(ii) Decide thickness of each bar and accordingly adjust x points on X –axis.

(iii) Give different color to different data ranges using color Argument of bar() function.

(iv) The width argument remains the same for all ranges being Plotted

(v) Plot using bar() for each range separately

```
Example :   Write a program to plot  a bar chart from the medals won by top four
countries .

Make sure that bars are separately visible.


>>>import  matplotlib.pyplot as pl
>>>import numpy as np
>>>Pl.figure(figsize=(10,7))
>>>Info=[‘Gold’,’Silver’,’Bronze’,’Total’]
>>>Australia=[80,59,59,198]
>>>England=[45,45,46,136]
>>>India=[26,20,20,66]
>>>Canada=[15,40,27,82]
>>>X=np.arange(len(Info))
>>>Pl.bar(Info,Australia, width=.15)
>>>Pl.bar(X+0.15,England,width=.15)
```

```
>>>Pl.bar(x+0.30, India, width=.15)
>>>Pl.bar(x+0.45, canada, width=.15)
Pl.show()
```

Customizing the Plot Anatomy of a Chart

(i) Axes

a) Axis

- It is define the area on which actual plot (line or bar or graph etc.) will appear.
- Axes have properties like label, limits and tick marks on them.
- There are 02 axes in a plot:
  - X-axis, the horizontal axis
  - Y-axis, the vertical axis

- Axis Label:-
  - It defines the name for an axis.
  - It is individually defined for X-axis and Y-axis each.
- Limits :-

  These define the range of values and number of values marked on X-axis and Y-axis.
- Tick_marks are individual points marked on the X-axis or Y-axis.

(ii) Title :

(a) This is the text that appears on the top of the plot .
(b) It defines what the chart is about.
(c) Example to add a tile in graph

>>>import myplotlib. Pyplot as pl
>>>pl. title(“ This my personal chart”)

(iii) Legend:- This is the different colors that identify different
Sets of data plotted on the plot.
The legend are shown in corner of chart.

Setting Xlimit and YLimits
xlim() and Ylim() are used to set the limit of x and Y

Creating Histograms with PyPlot

- A histogram is a summarisation tool for discrete or continuous data.
- A histogram provides a visual interpretation of numerical data by showing the number of data points that fall within a specified range of **values (called bins).**
- It is similar to a vertical bar graph.
- Histogram **shows no gaps between the bars.**
- Histogram are a great way to show results of continuous data, such as: weight, height, how much time and so forth.
- Note :But when the data is in categories (such as Country or subject etc), one should use a bar chart.

- Example
  Prof Awasthi is doing some research in the field of Environment . For some plotting purposes, he has generated some data as :

```
  mu=100
Sigma=15
X=mu+sigma* numpy.random.randn(10000)
```

Write a program to plot this data on a horizontal histogram with this data.

```
>>>  import numpy as np

>>> import matpoltlib.pyplot as pl

>>>mu=100

Sigma=15

x=mu+sigma* np.random.randn(10000)

pl.hist(x, bins=30, orientation='horizontal')

pl.title('Research data Histogram')

pl.show()
```