Programming with Python
Part 6: Visualizing with Matplotlib

What is Data Visualization

- It refers to the graphical or visual representation of information and data using visual elements like charts, graphs, and maps etc.
- Helpful in decision making.
- It unveils pattern, trends, outliers, correlations etc. in the data, and thereby helps decision makers understand the meaning of data to drive business decisions.

Using Pyplot of Matplotlib

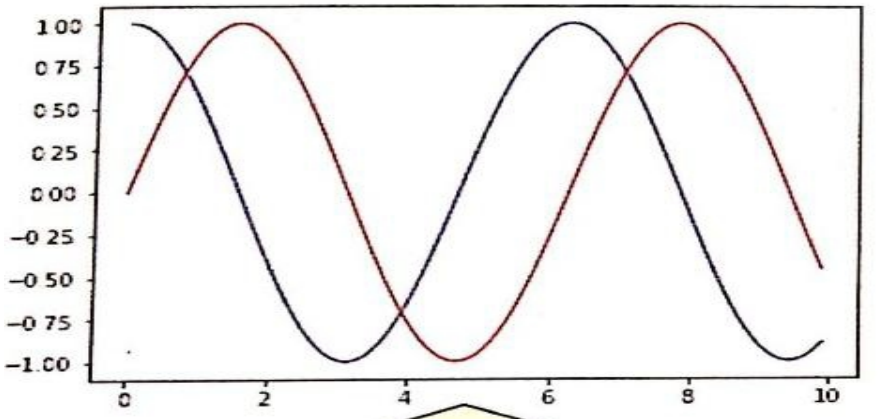
- The **matplotlib** is a Python library that provides many interfaces and functionality for 2D-graphics.
- matplotlib is a high quality plotting library of Python.
- - PyPlot is a collection of methods within matplotlib which allows user to construct 2D plots easily and interactively.

Importing PyPlot

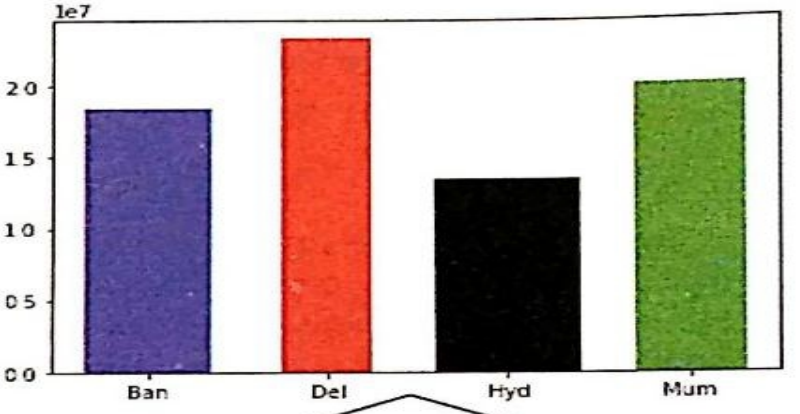
- In order to use pyplot methods on your computers, we need to import it by issuing one of the following command:

```
import matplotlib.pyplot as plt
```

Commonly used chart types



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 line segments.



A bar chart is a chart that presents categorical data with rectangular bars with heights proportional to the values that they represent.

A scatter plot is a two-dimensional data visualization that uses dots to represent the values obtained for two different variables - one plotted along the x-axis and the other plotted along the y-axis.

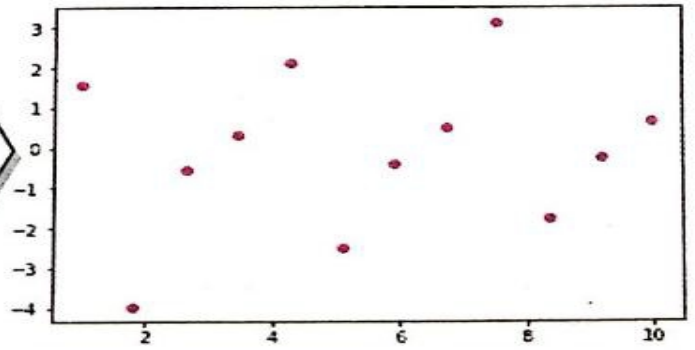


Figure 3.1 Some commonly used chart types.

Scanned with CamScanner

Line Chart

- 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 line segments.
- - The PyPlot interface offers plot( ) function for creating a line graph.
- - E.g.

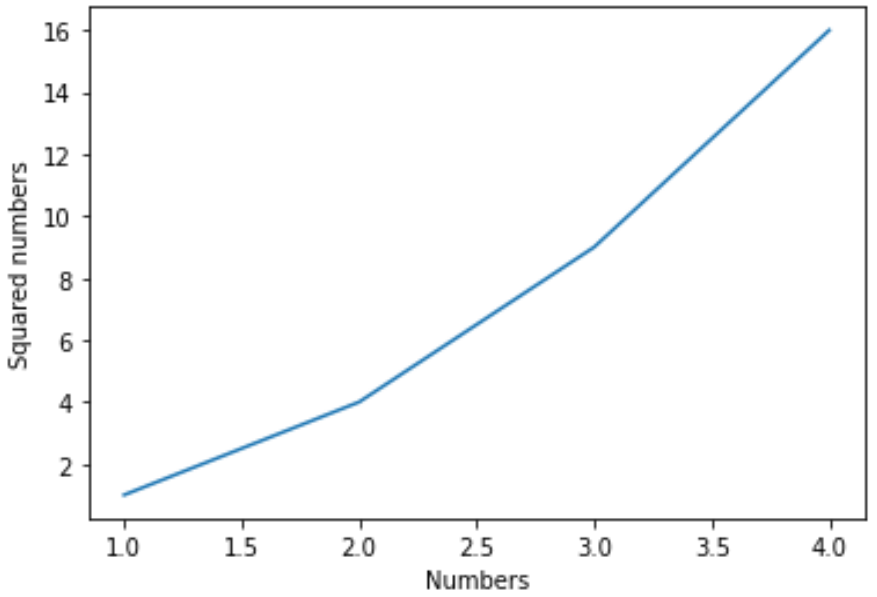
```
import matplotlib.pyplot as plt
import numpy as np

a=np.array([1,2,3,4])
b= a**2
plt.plot(a,b)
plt.xlabel('Numbers')
plt.ylabel('Squared numbers')
plt.show()
```

The import statement is to be given just once

List b containing values as squares of values in list a

show( ) method is used to display plot as per given specification

You can set x-axis' and y-axis' labels using functions xlabel( ) and ylabel( ) respectively,



Other Settings in Pyplot

The plot( ) function allows you to specify multiple settings for your chart/graph such as:

- color(line color/marker color)
- marker type
- marker size , etc.

Changing Line Color
plt.plot(<data1>,<data2> , **<color code>**)

Different color code

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

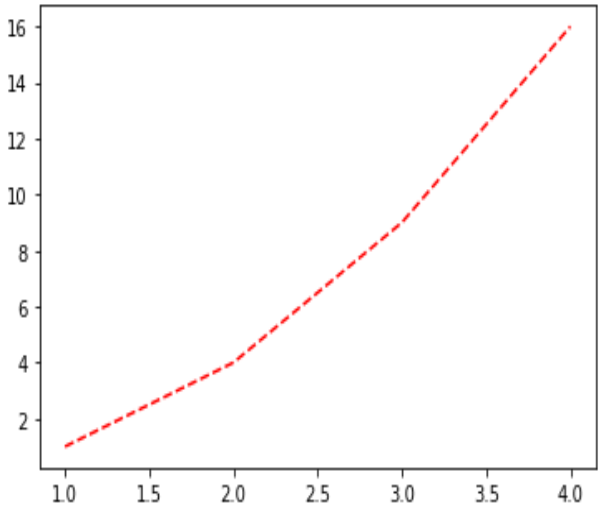
- Changing Line Style

  plt.plot(<data1>, <data2> , <linestyle>)

  linestyle or ls = [‘Solid’ | ‘dashed’ , ‘dashdot’ , ‘dotted’]

- **Example**

```
import matplotlib.pyplot as plt
import numpy as np
a=np.array([1,2,3,4])
b= a**2
plt.plot(a,b,'r',linestyle='dashed')
plt.show()
```

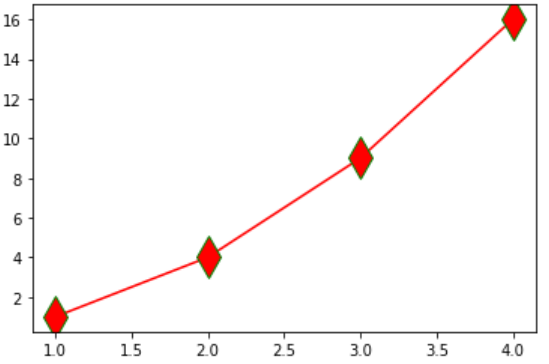


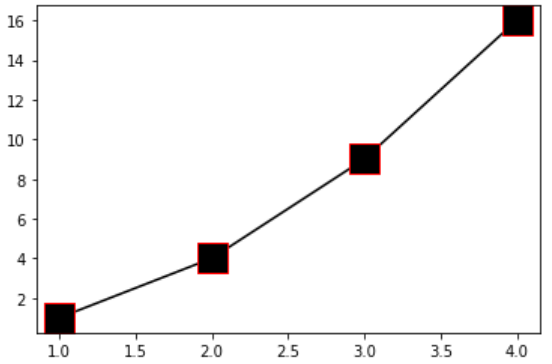
- Changing Marker Type, Size and Color
  - data points being plotted are called markers. To change market type, its size and color, the following arguments can be used in plot( ) function:

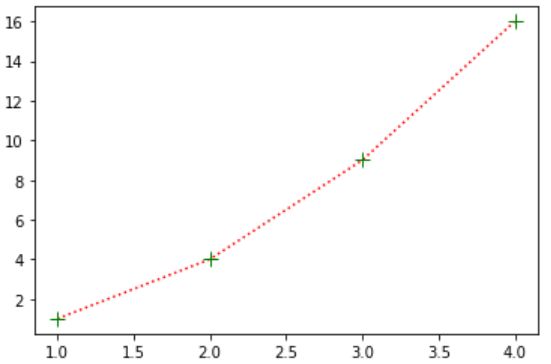
  - marker = <valid marker type> , markersize = <in points> , markeredgecolor = <valid color>

Example

```
import matplotlib.pyplot as plt

import numpy as np

a=np.array([1,2,3,4])

b= a**2

plt.plot(a,b,'r',marker='d',markersize=20,markeredgecolor
='green') #plot1

plt.show()

plt.plot(a,b,'k',linestyle='solid',marker='s',markersize=20,
markeredgecolor='red') #plot2

plt.show()

plt.plot(a,b,'r+',linestyle='dotted',markersize=10,markered
gecolor='green') #plot3

plt.show()
```







Creating a Scatter Chart

- It 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:
- 1. plot( ) function
- 2. scatter( ) function

Scatter charts using plot( ) function

  - If you specify the linecolor and markerstyle (e.g. “r+” or “bo” etc.) without the linestyle argument, then the plot created resembles a scatter chart as only the datapoints are plotted now.

Scatter Charts using scatter Function ( )

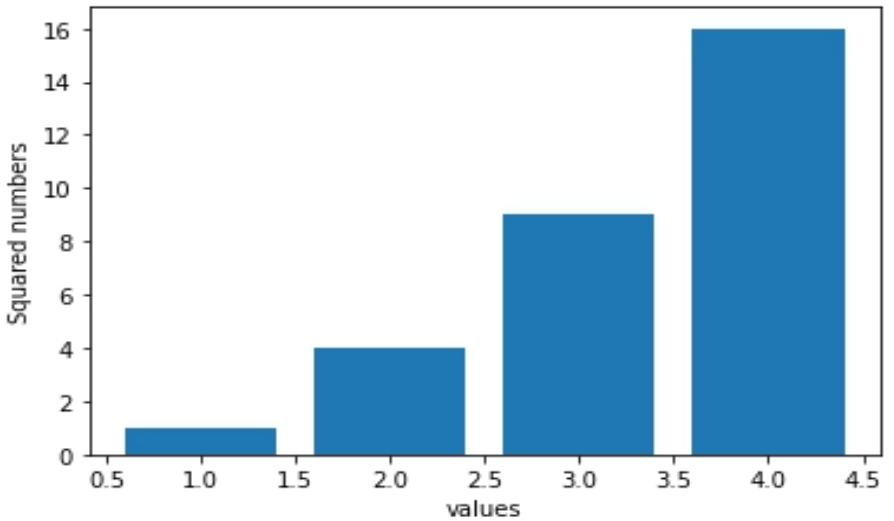
- This function can be used as:

  plt.scatter(<array1>, <array2>)

Creating Bar Charts

- A Bar Graph/ Chart is a graphical display of data using bars of different heights.
- Pyplot offers bar( ) function to create a bar chart where you can specify the sequences for x-axis and corresponding sequence to be plotted on y-axis .
- If you want to specify x-axis label and y-axis label, then you need to give commands:
- plt.xlabel(“your label “)
- plt.ylabel(“your label “)

```
import matplotlib.pyplot as plt

import numpy as np

a=np.array([1,2,3,4])

b= a**2

c= a+3

plt.bar(a,b)

plt.xlabel("values")

plt.ylabel("Doubles")

plt.show()
```



Mar, 2022

Adding Legends

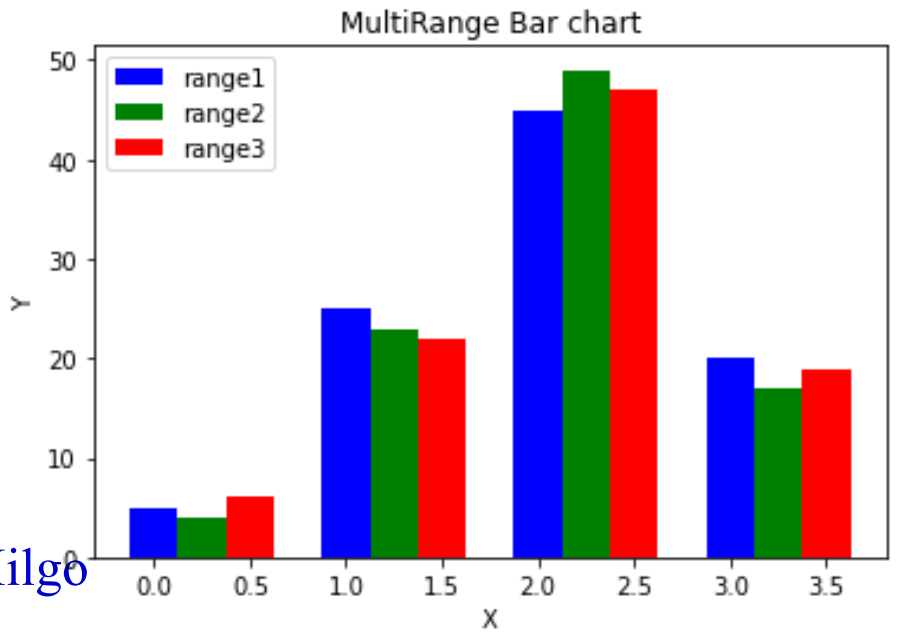
- When we plot multiple ranges on a single plot, it becomes necessary that legends are specified.
- **Two step process:**

1) In the plotting functions like plot( ) , bar( ) etc., give a specific label to data range using argument label.
2) Add legend to the plot using legend( ) as per format:

plt.legend(loc = <position number or string>)

The loc argument can either take values 1, 2, 3, 4 signifying the position strings ‘upper right’,’upper left’,’lower left’,’lower right’ respectively. Default position is ‘upper right’ or 1.

```
import matplotlib.pyplot as plt
import numpy as np
val=[[5.,25.,45.,20.],[4.,23.,49.,17.],[6.,22.,47.,19.]]
x=np.arange(4)
#step1: specify label for each range being plotted using label
plt.bar(x+0.00,val[0],color='b',width=0.25,label='range1')
plt.bar(x+0.25,val[1],color='g',width=0.25,label='range2')
plt.bar(x+0.50,val[2],color='r',width=0.25,label='range3')
#step2:add legend,i.e.
plt.legend(loc='upper left')
plt.title("MultiRange Bar chart")
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```



Saving a Figure

- If you want to save a plot created using pyplot functions for later use or for keeping records, you can use savefig( ) to save the plot.
- - You can use the pyplot's savefig( ) as per format:

  <matplotlib.pyplot>.savefig(<string with filename and path>)
- You can save figures in popular formats like .pdf , .png , etc.
- Example

  plt.savefig("multibar.pdf") # save the plot in current directory

  plt.savefig("c:\\COM418\\multibar.pdf") #save the plot at the given path