

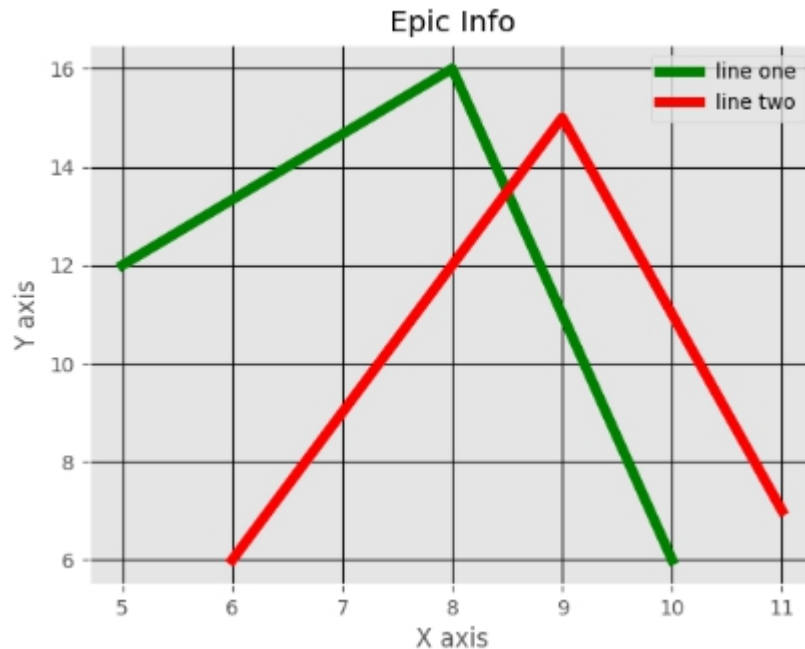
# Visualization

## Line Charts

A line chart can be created using the Matplotlib plot() function. While we can just plot a line, we are not limited to that. We can explicitly define the grid, the x and y-axis scale and labels, title and display options.

```
1  from matplotlib import pyplot as plt
2  from matplotlib import style
3
4  style.use('ggplot')
5  x = [5,8,10]
6  y = [12,16,6]
7  x2 = [6,9,11]
8  y2 = [6,15,7]
9  plt.plot(x,y,'g',label='line one', linewidth=5)
10 plt.plot(x2,y2,'c',label='line two',linewidth=5)
11 plt.title('Epic Info')
12 plt.ylabel('Y axis')
13 plt.xlabel('X axis')
14 plt.legend()
15 plt.grid(True,color='k')
16 plt.show()
```

Resultant Graph:

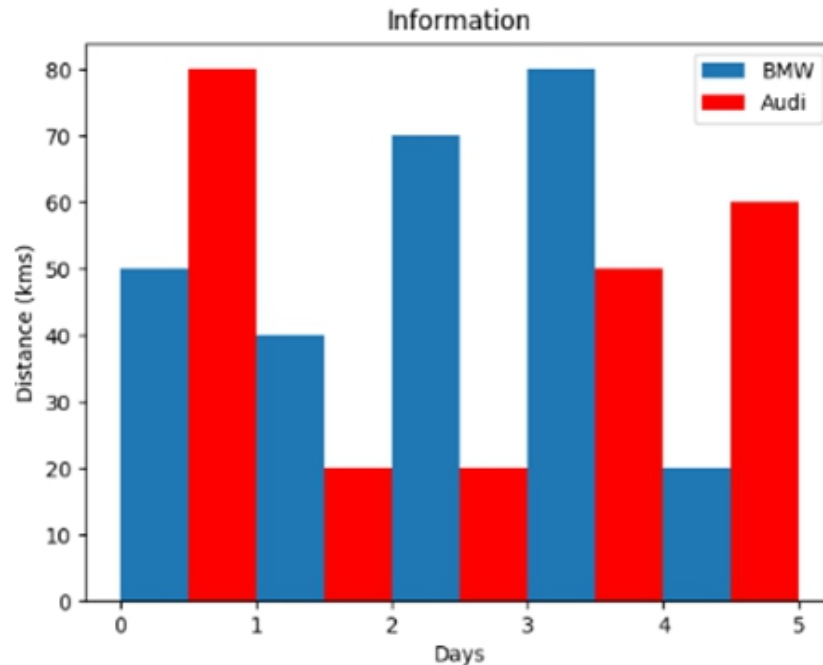


## Bar Charts

- A bar graph uses bars to compare data among different categories.
- It is well suited when you want to measure the changes over a period of time.
- It can be represented horizontally or vertically. Also, the important thing to keep in mind is that longer the bar, greater is the value.

```
1  from matplotlib import pyplot as plt
2
3  plt.bar([0.25,1.25,2.25,3.25,4.25],[50,40,70,80,20],
4  label="BMW",width=.5)
5  plt.bar([.75,1.75,2.75,3.75,4.75],[80,20,20,50,60],
6  label="Audi", color='r',width=.5)
7  plt.legend()
8  plt.xlabel('Days')
9  plt.ylabel('Distance (kms)')
10 plt.title('Information')
11 plt.show()
```

Resultant Graph:



## Histogram

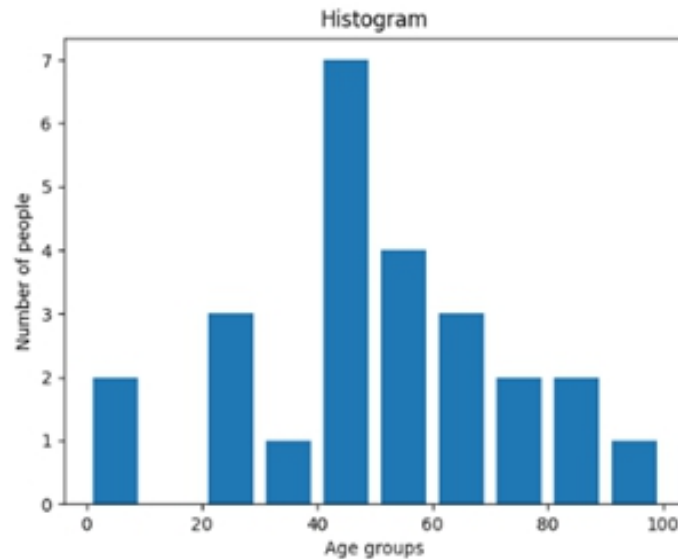
- Difference b/w Bar Charts and Histogram:
- Histograms are used to show a distribution whereas a bar chart is used to compare different entities.
- Histograms are useful when you have arrays or a very long list.
- Example: Let's consider an example where I have to plot the age of population with respect to the bin.
- Now, bin refers to the range of values that are divided into a series of intervals. Bins are usually created of the same size.

```

1 import matplotlib.pyplot as plt
2 population_age = [22,55,62,45,21,22,34,42,42,4,2,102,95,85,55,110,120,70,65,55,
3 bins = [0,10,20,30,40,50,60,70,80,90,100]
4 plt.hist(population_age, bins, histtype='bar', rwidth=0.8)
5 plt.xlabel('age groups')
6 plt.ylabel('Number of people')
7 plt.title('Histogram')
8 plt.show()

```

Resultant Graph:



## Scatter Plots

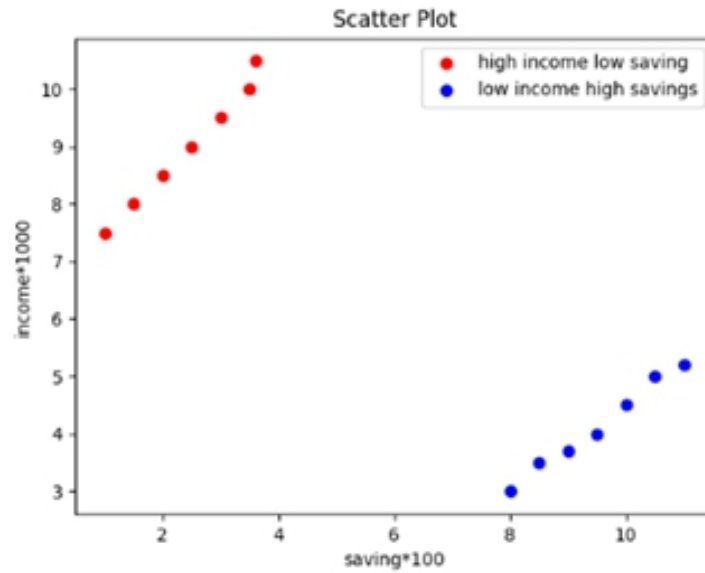
- Usually, we need scatter plots in order to compare variables, for example, how much one variable is affected by another variable to build a relation out of it.
- The data is displayed as a collection of points, each having the value of one variable which determines the position on the horizontal axis and the value of other variable determines the position on the vertical axis.
- Example: Let us plot a graph for segregating people with high income and low saving from people with low income and high savings.

```

1 import matplotlib.pyplot as plt
2 x = [1,1.5,2,2.5,3,3.5,3.6]
3 y = [7.5,8,8.5,9,9.5,10,10.5]
4
5 x1=[8,8.5,9,9.5,10,10.5,11]
6 y1=[3,3.5,3.7,4,4.5,5,5.2]
7
8 plt.scatter(x,y, label='high income low saving',color='r')
9 plt.scatter(x1,y1,label='low income high savings',color='b')
10 plt.xlabel('saving*100')
11 plt.ylabel('income*1000')
12 plt.title('Scatter Plot')
13 plt.legend()
14 plt.show()

```

Resultant Graph:



## Pie Charts

- A pie chart refers to a circular graph which is broken down into segments i.e. slices of pie.
- It is basically used to show the percentage or proportional data where each slice of pie represents a category.

```

1  import matplotlib.pyplot as plt
2
3  days = [1,2,3,4,5]
4
5  sleeping =[7,8,6,11,7]
6  eating = [2,3,4,3,2]
7  working =[7,8,7,2,2]
8  playing = [8,5,7,8,13]
9  slices = [7,2,2,13]
10 activities = ['sleeping','eating','working','playing']
11 cols = ['c','m','r','b']
12
13 plt.pie(slices,
14         labels=activities,
15         colors=cols,
16         startangle=90,
17         shadow=True,
18         explode=(0,0.1,0,0),
19         autopct='%1.1f%%')
20
21 plt.title('Pie Plot')
22 plt.show()

```

Resultant Graph:

