

DATA VISUALIZATION USING MATPLOTLIB

Matplotlib is a low level graph plotting library in python that serves as visualization utility. It is an open source and we can use it freely. It is built in Numpy arrays and designed to work with the broader Scipy stack and consists of several plots like line, bar, scatter, histogram etc.

Pyplot is a matplotlib module that provides a MATLAB-like interface. It provides several functions to create plot, show, title, subplot, legend etc

In this article, we are discussing various types of plots and representing it graphically.

Types of Plot

Matplotlib contains a wide variety of plots. Plot helps to understand the latest trends, patterns and correlation. Some of the plots are given below:

1. Line graph
2. Bar plot
3. Histogram
4. Scatter plot
5. Pie chart

LINE GRAPH

Line graph is used to represent the relationship between 2 variables X and Y on different axis. It is also defined as the information changing over time. It is plotted using function plot() function.

Syntax

plot(x,y)

There are two types of line graph. They are

1. Simple Line graph: If there is only one line to connect all the datapoints.

Then the graph is called simple line graph.

2. Multiple Line graph: If there is more than one line to connect all the datapoints. Then the graph is called multiple line graph.

To draw a line plot, we use 3 steps:

1. Import libraries
2. Create data
3. Plot the lines over data using plot() function.
4. Label the axis and title.

The code for line graph is given by

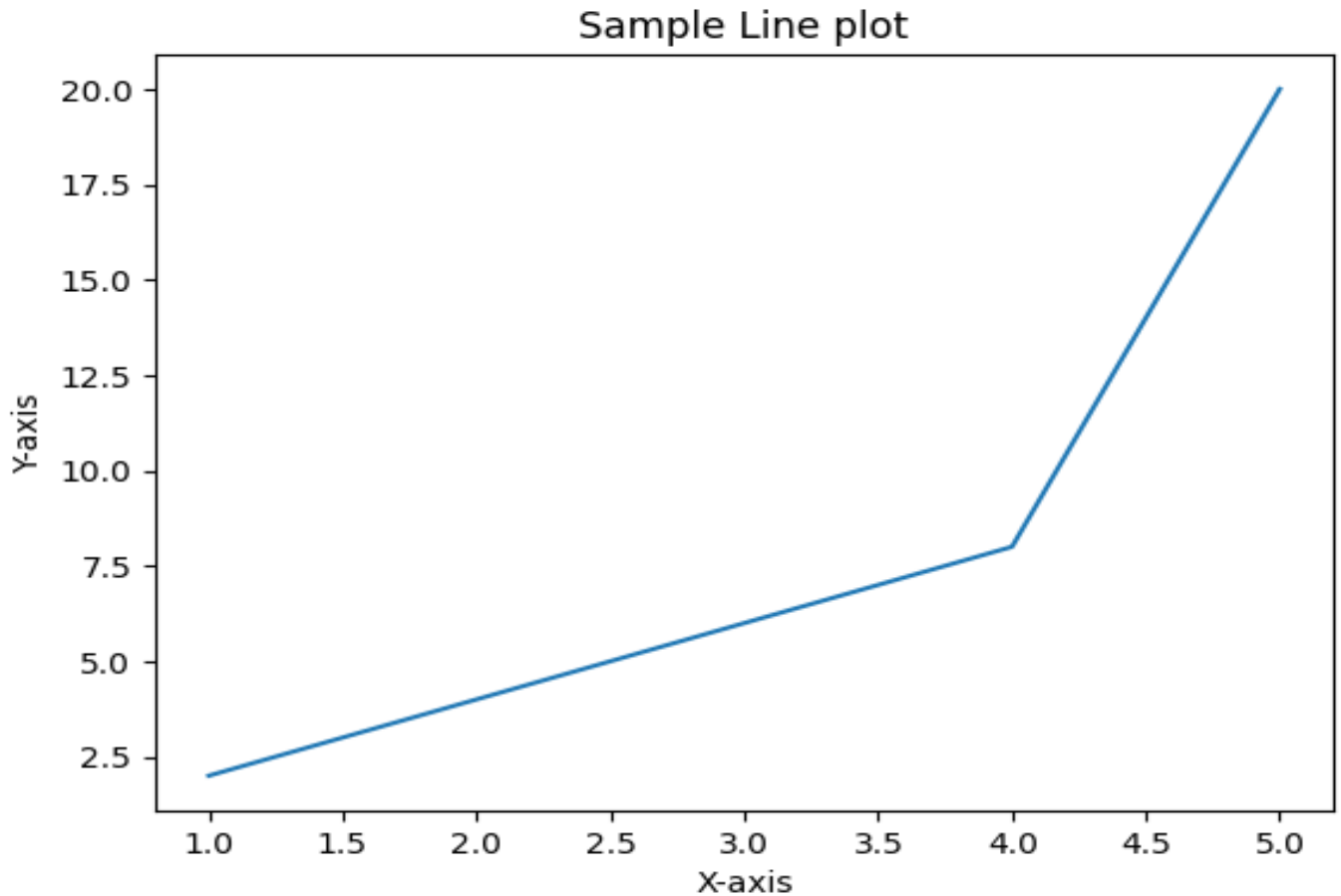
```
import matplotlib.pyplot as plt
x=[1,2,3,4,5]
y=[2,4,6,8,20]
plt.plot(x,y)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Sample Line plot')
plt.show()
```

Plt.xlabel shows the axis name corresponding to the X axis

Plt.ylabel shows the axis name corresponding to the Y axis

Plt.title shows the title corresponding to the plot

Plt.show gives the output.



BAR PLOT

It shows the relationship between a numeric and a categorical variables. Each entity of the categorical variable is represented as a bar. The size of the bar represents its numeric values. It can be plotted vertically or horizontally.

Syntax

`plt.bar(x,height,width,bottom,align)`

Creating vertical bar graph

- Import library
- Create list x for discrete values on x axis
- Create list y consisting only numeric dat
- `plt.bar(x,y)`
- Setting labels using `x_label` and `y_label`

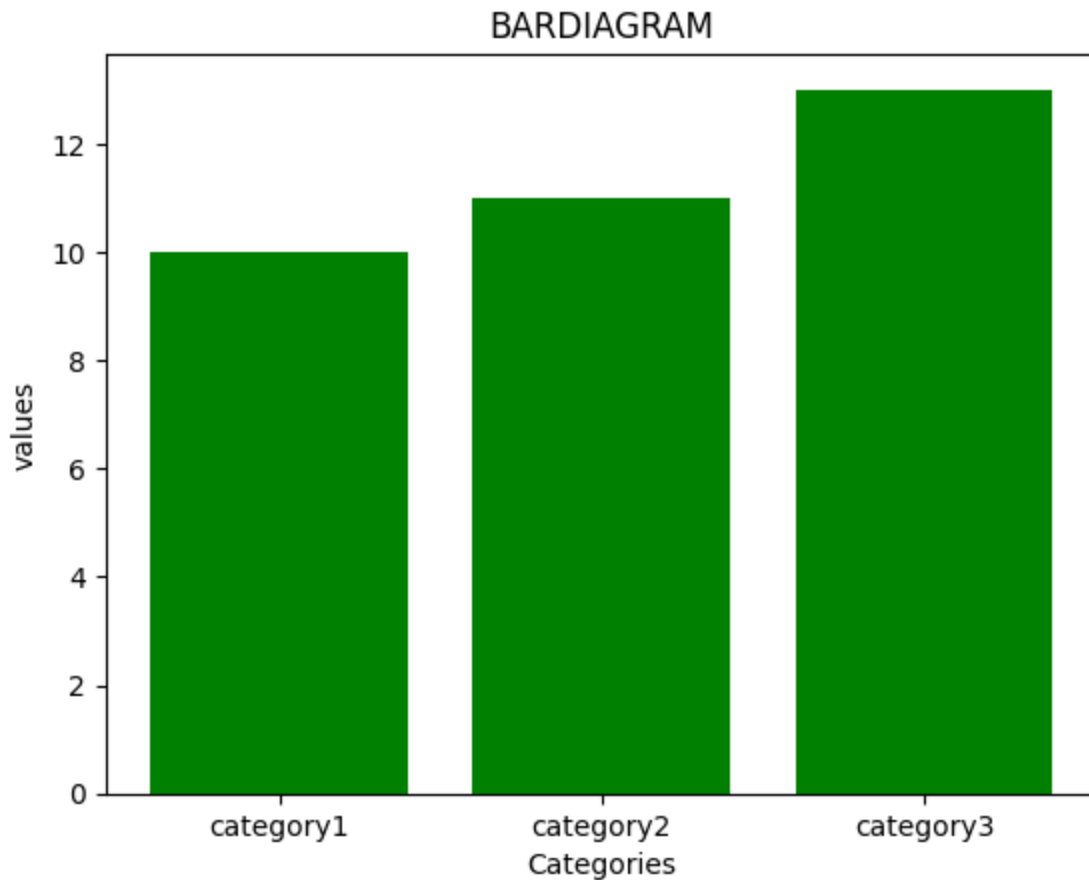
- Setting the title
- plt.show()

Creating horizontal bar graph

Follow other steps in vertical bar graph except plotting the bar. for plotting horizontal bar graph, we use the command plt.barh()

The code for plotting bar plot is

```
import matplotlib.pyplot as plt
Categories=['category1','category2','category3']
values=[10,11,13]
plt.barh(Categories,values,color='green')
plt.xlabel('Categories')
plt.ylabel('values')
plt.title('BARDIAGRAM')
plt.show()
```



Histogram

Histogram is a graphical representation of the distribution of data. It is represented by a set of rectangles adjacent to each other, where each bar represents data.

The histogram is used when the data are numerical. It is also used to check the shape of data distribution and to determine the output is different when it involves 2 or more processes. It is used to analyze whether the given process meets customer requirements.

How to plot histogram

- ☐ By marking class intervals on x axis and frequencies on y axis.
- ☐ Scales for both axis must be same
- ☐ Class intervals need be exclusive
- ☐ Height of each rectangle is proportional to the class frequency if the

class intervals are equal.

- ☐ Draw rectangles with base as class intervals and corresponding frequencies as heights.
- ☐ If the intervals are not equal then area of rectangle is proportional to the class frequency.

The difference between histogram and bar graph is the dimension difference i.e one is 2D and other is 1D. In histogram, rectangles touch each other and in bar graph, rectangles are separated.

The following table shows the parameter accepted by histogram.

Attribute	Parameter
x	array
Bins	range
Density	Optional parameter contain boolean values
histtype	Types of histogram
color	setcolor
label	Sequence string to match with multiple dataset

Creating basic histogram

The following are the steps to plot histogram;

- 1.Import libraries
- 2.Create dataset
- 3.Plot the data set using hist() function.
- 4.Label the axis and title of the set.

The code is given by

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

```
import numpy as np
```

```
data=np.random.random(100)
```

```
print(data)
```

```
[0.4739894 0.1502426 0.18929974 0.12426991 0.53019006 0.96792237  
0.22202452 0.25051486 0.10638543 0.19515614 0.2017147 0.63630412  
0.39352945 0.34618325 0.15426803 0.96364488 0.90318777 0.87245027  
0.54252227 0.62945485 0.06666067 0.67843117 0.03722104 0.02048776  
0.17284349 0.77583708 0.14107413 0.0699803 0.8992533 0.53136073  
0.99274416 0.67665445 0.6428408 0.63369514 0.10642902 0.94390483  
0.69995663 0.28367712 0.8098224 0.40277044 0.16854478 0.0480835  
0.58609918 0.6802093 0.07951991 0.12420307 0.99608717 0.45481698  
0.39700173 0.79801061 0.11791016 0.65397733 0.18935765 0.52960247  
0.92374098 0.03815731 0.17107829 0.02617943 0.45635893 0.35088794  
0.58547279 0.26135464 0.87449348 0.41520241 0.39355397 0.93152181  
0.7488007 0.24255372 0.58784643 0.13149456 0.41449409 0.81178789  
0.83798848 0.75838004 0.8527512 0.44310398 0.47557349 0.97207455  
0.36210654 0.02967896 0.71324153 0.3378346 0.0985566 0.14597959  
0.32106066 0.42219502 0.97138874 0.6832819 0.94890003 0.53145712  
0.72625509 0.51656608 0.77522098 0.31417567 0.79463661 0.43998459  
0.6878298 0.98146213 0.52534638 0.7940832 ]
```

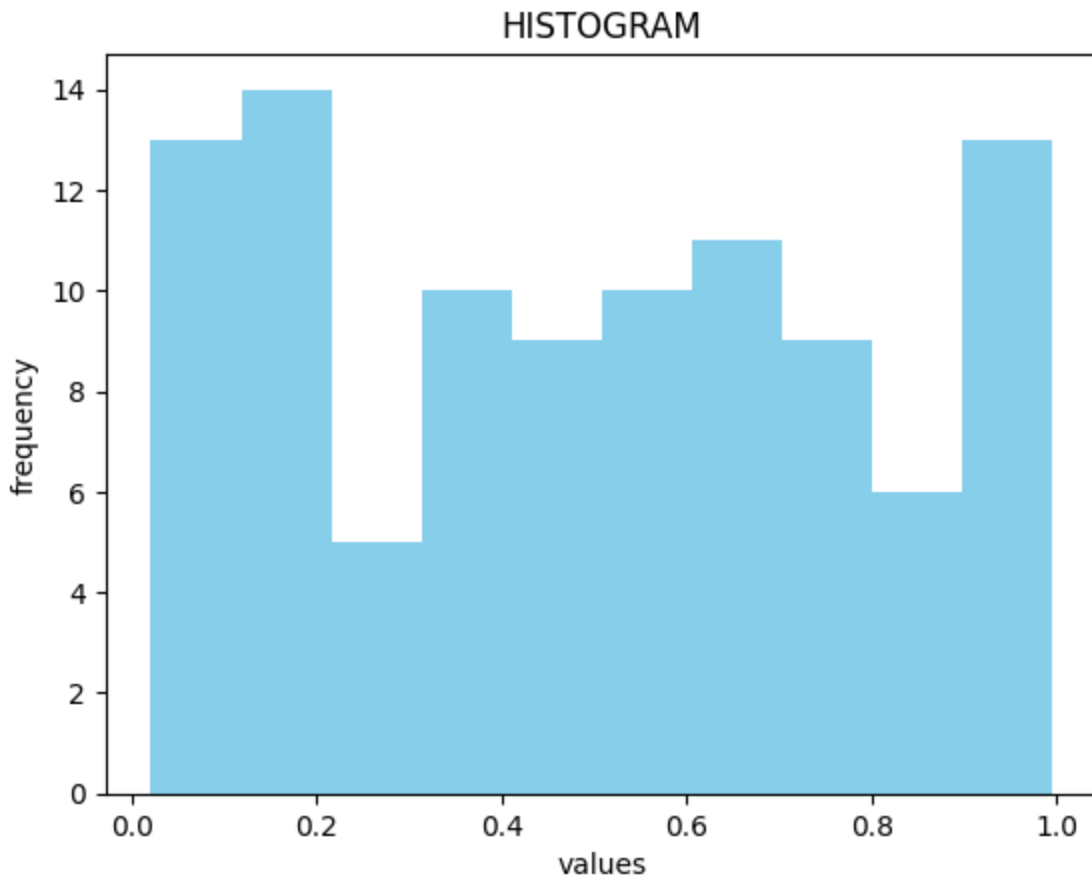
```
plt.hist(data,color='skyblue',bins=10)
```

```
plt.xlabel('values')
```

```
plt.ylabel('frequency')
```

```
plt.title('HISTOGRAM')
```

```
plt.show()
```



Scatter plot

Scatter plot are the graphs that present the relationship between variables in a dataset. It represent datapoints on a 2D plane. The independent variable is plotted on x axis and dependent variable is plotted on the y axis.

Scatter plots are used in following situations:

- When we have passed numeric data
- When there are more than one dependent variable for a unique value of an independent variable.
- Relationship between variables.

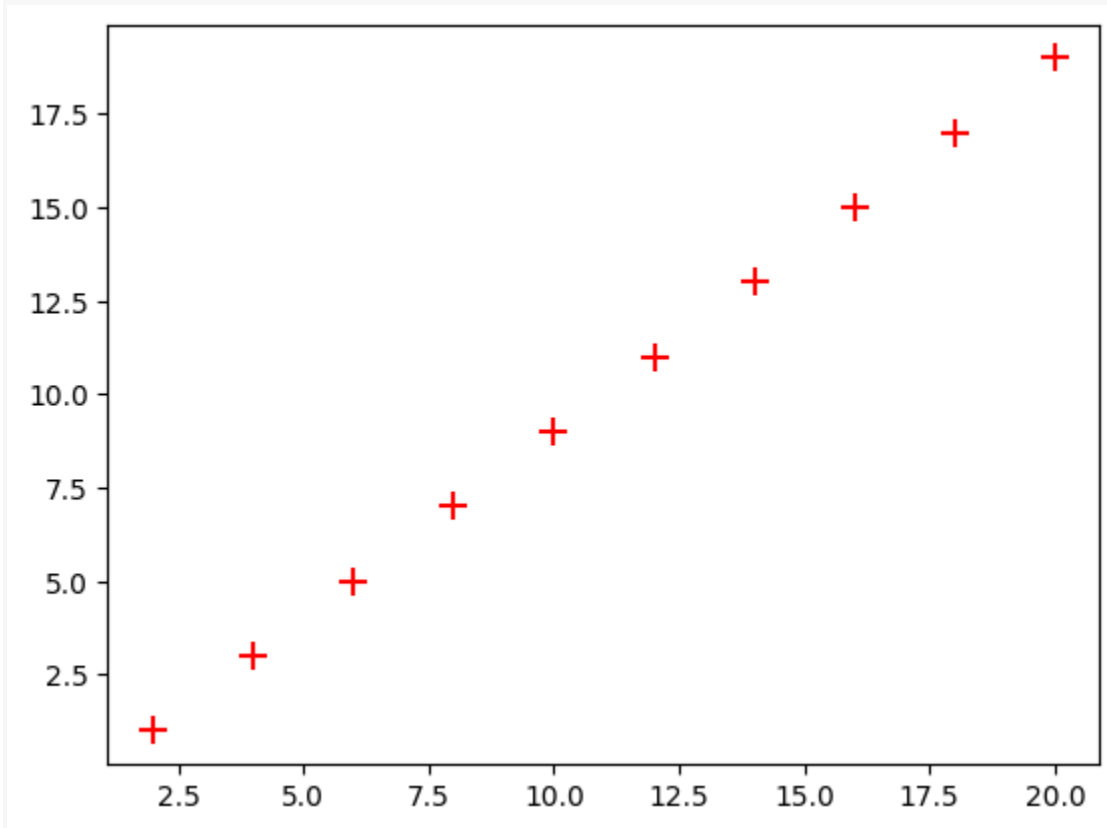
Scatter plot is mainly used in determining the correlation. If the points touch the lines then we can say that the variables are highly correlated.

The code for scatter plot is:

```
import matplotlib.pyplot as plt  
x=[2,4,6,8,10,12,14,16,18,20]
```



```
y=[1,3,5,7,9,11,13,15,17,19]  
plt.scatter(x,y,marker='+',color='red',s=100)
```



The marker is used to plot the points in '+' sign. We can also plot multiple plot and bubble sort.

Pie chart

Pie chart is also known as circle chart, dividing the circular statistical graphics into sectors. Each sector denotes a proportionate part of the whole.

The sum of all data = 360 degree
The total value of the pie = 100%

The steps for workout is:

1. Categorize the data
2. Calculate the total
3. Divide the categories
4. Convert into percentage.
5. Finally, calculate the degrees.

Formula

$$\text{Degree} = (\text{givendata} / \text{total value}) * 100$$

Syntax

The syntax for pie chart is plt.pie() function.

The code for pie chart is:

```
import matplotlib.pyplot as plt
sizes=[25,35,20,20]
labels=['cate A','cate B','cate C','cate D']
plt.pie(sizes,labels=labels,autopct='%1.1f%%')
plt.show()
```

