

Data Visualisation with Python Programming

- Presentation By Uplatz
- Contact us: <https://training.uplatz.com>
- Email: info@uplatz.com
- Phone: +44 7836 212635

Data Visualisation tools

Learning outcomes:

- Bar chart
- Histogram
- Pie Chart

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.

A bar graph shows comparisons among discrete categories. One axis of the chart shows the specific categories being compared, and the other axis represents a measured value.

Bar chart

Bar charts are used to present categorical data with rectangular bars. The bars can be plotted vertically or horizontally, and their heights/lengths are proportional to the values that they represent.

Bar chart

Creating a bar plot:

The **matplotlib** API in Python provides the **bar()** function which can be used in MATLAB style use or as an object-oriented API. The syntax of the **bar()** function to be used with the axes is as follows:-

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

The function creates a bar plot bounded with a rectangle depending on the given parameters.

Bar chart

Creating a bar plot:

Example:

```
from matplotlib import pyplot as plt
Subject=["Mathematics","Physics",
"Chemistry","Biology"]
Marks = [90,70,75,85]
plt.bar(Subject, Marks, color='maroon',width =
0.4)
plt.show()
```

Bar chart

Creating a bar plot:

Example: **Creating Horizontal bar**

You can create the horizontal bar chart also with the help of

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

Let's see the example.

Bar chart

Creating a bar plot: comparing two data series:

We can also plot bar charts by comparing two data series side by side; this allows us to observe their differences more easily.

Let's see the example.

Bar chart

Creating a bar plot using a data from CSV file:

We can also plot bar charts by reading data from CSV file.

Let's see the example.

Histogram

A histogram is a great tool for quickly assessing a [probability distribution](#) that is intuitively understood by almost any audience. Python offers a handful of different options for building and plotting histograms. Most people know a histogram by its graphical representation, which is similar to a bar graph.

Histogram

A histogram is basically used to represent data provided in a form of some groups. It is accurate method for the graphical representation of numerical data distribution. It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency.

Histogram

Creating a Histogram:

To construct a histogram, follow these steps –

- **Bin** the range of values.
- Divide the entire range of values into a series of intervals.
- Count how many values fall into each interval.

The bins are usually specified as consecutive, non-overlapping intervals of a variable.

The **matplotlib.pyplot.hist()** function plots a histogram. It computes and draws the histogram of x .

Histogram

Creating a Histogram:

Following example plots a histogram of marks obtained by students in a class. Four bins, 0-25, 26-50, 51-75, and 76-100 are defined. The Histogram shows number of students falling in this range.

```
from matplotlib import pyplot as plt
import numpy as np
a=np.array([22,87,5,43,56,73,55,54,11,20,51,5,79,
31,27])
plt.hist(a, bins = [0,25,50,75,100])
plt.show()
```

Histogram

Creating a Histogram:

We can also set the **bins to any number** that shows the number of bins in that histogram.

Let's see the example.

Histogram

Creating a Histogram:

If needed, you can further [style your histogram](#).

One way to style your histogram is by adding this syntax towards the end of the code:

```
plt.style.use('ggplot')
```

Let's see the example.

Histogram

Creating a Histogram:

let's try some real data example.

It took me a while to figure out which dataset should I pick and finally I got settled with dataset of [Pakistan v India final in Champions Trophy 2017](#)

The thing I want to know how many runs were made per [delivery](#). The possible outcomes are: 0–6, provided we avoid wide and no-balls. So I want to find out the frequency of runs by both teams.

Since this is just for sake of education purpose I picked stats of first four overs of both teams. Data I prepared you can see in [**Cricket.csv**](#)

Histogram

Creating a Histogram:

Let's see the example.

Both X and Y axes are displaying data in decimal format and it does not look good because our data will always be in integer format. After setting range for X and Y ticks using `plt.xticks(range(0, 6))` `plt.yticks(range(1, 20))` the graph now looks like (See the example)

Pie Chart

A **Pie Chart** is a circular statistical plot that can display only one series of data. Pie charts show the size of items (called wedge) in one data series, proportional to the sum of the items. The data points in a pie chart are shown as a percentage of the whole pie. The area of the wedge is determined by the length of the arc of the wedge. The area of a wedge represents the relative percentage of that part with respect to whole data. Pie charts are commonly used in business presentations like sales, operations, survey results, resources, etc. as they provide a quick summary.

Pie Chart

Matplotlib API has `pie()` function in its `pyplot` module which create a pie chart representing the data in an array.

Syntax:

`matplotlib.pyplot.pie(data, explode=None, labels=None, colors=None, autopct=None, shadow=False)`

Pie Chart

Creating Pie Chart:

Let's create a simple pie chart using the `pie()` function.

Let's see the example.

Pie Chart

Creating Pie Chart:

The default **startangle** is 0, which would start the "AUDI" slice on the positive x-axis. This example sets **startangle = 90** such that everything is rotated counter-clockwise by 90 degrees, and the **AUDI** slice starts on the positive y-axis.

autopct parameter of the **arc()** function controls how the percentages are displayed in the wedges.

Let's see the example.

Pie Chart

Creating Pie Chart:

We can also add the colours to each wedge as per our choice.

Let's see the example.



Thank you