Data Science with Python Programming

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

Uplatz

Data Visualisation with Matplotlib

Uplatz

Learning outcomes:

Data Visualisation with Matplotlib

- What is Data Visualisation?
- Introduction to Matplotlib
- Installation of Matplotlib

Types of data visualization charts/plots

- Line chart, Scatter plot
- Bar chart, Histogram
- Area Plot, Pie chart
- Boxplot, Contour plot

What is Data Visualisation?

Data visualization plays an essential role in the representation of both small and large-scale data. One of the key skills of a data scientist is the ability to tell a compelling story, visualizing data and findings in an approachable and stimulating way. Learning how to leverage a software tool to visualize data will also enable you to extract information, better understand the data, and make more effective decisions.

What is Data Visualisation?

Data visualisation is an efficient technique of gaining insights about data through a visual medium.

The main goal of this Data Visualisation with **matplotlib** in Python session is to teach you how to take data that at first glance has little meaning and present that data in a form that makes sense to people.

Now one might ask why would I need to learn how to visualize data. Well data visualization is a way to show a complex data in a form that is graphical and easy to understand.

What is Data Visualisation?

Data visualization is a technique used for the graphical representation of data. By using elements like scatter plots, charts, graphs, histograms, area plots etc., we make our data more understandable. Data visualization makes it easy to recognize patterns, trends, and exceptions in our data. It enables us to convey information and results in a quick and visual way.

What is Data Visualisation?

The main benefits of data visualization are as follows:

- It simplifies the complex quantitative information.
- It helps analyse and explore big data easily
- It identifies the areas that need attention or improvement.
- It identifies the relationship between data points and variables.
- It explores new patterns and reveals hidden patterns in the data.

Introduction to Matplotlib

Matplotlib is a plotting library for the Python programming language and its numerical mathematics extension NumPy.
It was conceived by John Hunter in 2002, originally as a patch to IPython for enabling interactive MATLAB-style plotting via gnu-plot from the IPython command line.

Introduction to Matplotlib

Matplotlib is the most popular data visualization library of Python and is a 2D plotting library. It is the most widely-used library for plotting in the Python community and is more than a decade old. It comes with an interactive environment across platforms. Matplotlib can be used in Python scripts, the Python and IPython shells, the Jupyter notebook, web application servers and for graphical user interface toolkits.

Introduction to Matplotlib

It is a very versatile visualization library. With this library, with just a few lines of code, one can generate plots, bar charts, histograms, power spectra, stemplots, scatterplots, error charts, pie charts and many other types. The pyplot module provides a MATLAB-like interface, particularly when combined with Ipython.

Installation of Matplotlib

If you are using Anaconda distribution, then you need not install matplotlib, separately as it is already installed with it.

Standard Python distribution doesn't come bundled with matplotlib module. A lightweight alternative is to install matplotlib using popular Python package installer, **pip.**

Installation of Matplotlib

Open the command prompt and change the directory to your current working directory and then type any one of the following command.

pip3 install matplotlib OR
python -m pip install matplotlib OR
pip install --user matplotlib

Types of data visualization charts/plots

We will learn how to create a simple plot with Matplotlib.

matplotlib.pyplot is a collection of command style functions that make Matplotlib work like MATLAB. Each **pyplot** function makes some change to a figure. For example, a function creates a figure, a plotting area in a figure, plots some lines in a plotting area, decorates the plot with labels, etc.

Types of data visualization charts/plots

Line Chart:

A line chart or line graph displays the evolution of one or several numeric variables. Data points are connected by straight line segments. A line chart is often used to visualize a trend in data over intervals of time – a time series – thus the line is often drawn chronologically. You can use the plot(x,y) method to create a line chart.

Types of data visualization charts/plots

Line Chart:

Example:

```
import matplotlib.pyplot as plt OR
from matplotlib import pyplot as plt
x=[1,2,3,4]
y=[10,20,30,40]
plt.plot(x,y)
plt.show()
```

You can also add the label on x-axis and y-axis, and title to your chart.

Uplatz

Types of data visualization charts/plots

Scatter plot:

A Scatterplot displays the value of **2 sets of data** on 2 dimensions. Each dot represents an observation. The position on the X (horizontal) and Y (vertical) axis represents the values of the 2 variables. It is really useful to study the **relationship** between both variables. A scatter plot is a diagram where each value in the data set is represented by a dot. Use the scatter() method to draw a scatter plot diagram

Types of data visualization charts/plots

Scatter plot:

Example:

```
import matplotlib.pyplot as plt OR
from matplotlib import pyplot as plt
x=[1,2,3,4]
y=[10,20,30,40]
plt.scatter(x,y)
plt.show()
```

You can also add the label on x-axis and y-axis, and title to your chart.

Uplatz

Types of data visualization charts/plots

Bar Chart (Bar Plot):

A barplot (or barchart) is one of the most common type of plot. It shows the relationship between a **numerical variable** and a **categorical variable**. For example, you can display the height of several individuals using 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. Use the bar() method to draw a bar plot diagram.

Types of data visualization charts/plots

Bar Chart (Bar Plot):

Example:

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

You can also add the label on x-axis and y-axis, and title to your chart.

Types of data visualization charts/plots

Histogram:

An histogram is an accurate graphical representation of the distribution of numerical data. It takes as input one numerical variable only. The variable is cut into several **bins**, and the number of observation per bin is represented by the **height** of the bar. It is a type of bar plot where X-axis represents the bin ranges while Y-axis gives information about frequency. Use the hist() method to draw a histogram diagram.

Types of data visualization charts/plots

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()
```

Types of data visualization charts/plots

AREA Plot/Chart:

An area chart is really similar to a line chart, except that the area between the x axis and the line is filled in with color or shading. It represents the evolution of a **numerical variable following another** numerical variable. Use the fill_between () method to draw a Area chart.

Types of data visualization charts/plots

AREA Plot/Chart:

Example:

```
import matplotlib.pyplot as plt OR
from matplotlib import pyplot as plt
x=[1,2,3,4]
y=[10,20,30,40]
plt.fill_between(x,y)
plt.show()
```

You can also add the label on x-axis and y-axis, and title to your chart.

Types of data visualization charts/plots

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. Use the pie () method to draw a Pie chart.

Types of data visualization charts/plots

Pie Chart :

Example:

```
from matplotlib import pyplot as plt
cars = ['AUDI','BMW','FORD','TESLA', 'JAGUAR',
'MERCEDES']
data = [23, 17, 35, 29, 12, 41]
plt.pie(data, labels = cars)
plt.show()
```

You can also add the label on x-axis and y-axis, and title to your chart.

Types of data visualization charts/plots

Boxplot:

Boxplot is probably one of the most common type of graphic. It gives a nice **summary** of one or several **numeric variables**. The line that divides the box into 2 parts represents the **median** of the data. The end of the box shows the upper and lower **quartiles**. The extreme lines shows the highest and lowest value excluding **outliers**.
Use the boxplot() method to draw a box plot.

Types of data visualization charts/plots

Boxplot:

Example:

```
from matplotlib import pyplot as plt
data1= [100,120,140,160,180]
plt.boxplot(data1)
plt.show()
```

You can also add the label on x-axis and y-axis, and title to your chart.

Types of data visualization charts/plots

Contour plots :

Contour plots (sometimes called **Level Plots**) are a way to show a **three-dimensional** surface on a **two-dimensional plane**. It graphs two predictor variables X Y on the y-axis and a response variable Z as contours. These contours are sometimes called the z-slices. A contour plot is appropriate if you want to see how value Z changes as a function of two inputs X and Y, such that Z = f(X,Y). A contour line of a function of two variables is a curve along which the function has a constant value.

Types of data visualization charts/plots

Contour plots :

The independent variables x and y are usually restricted to a regular grid called **meshgrid**. The **numpy.meshgrid** creates a rectangular grid out of an **array of x values** and an **array of y values**. Matplotlib API contains **contour()** and **contourf()** functions that draw contour lines and filled contours, respectively. Both functions need three parameters **x,y and z.**

Types of data visualization charts/plots

Contour plots :

The **matplotlib.pyplot.contour()** are usually useful when **Z = f(X, Y)** i.e Z changes as a function of input X and Y. A **contourf()** is also available which allows us to draw filled contours.

Types of data visualization charts/plots

Contour plots :

Example:

```
import numpy as np
import matplotlib.pyplot as plt
#Generating 100 numbers from -3.0 to 3.0
xlist = np.linspace(-3.0, 3.0, 100)
ylist = np.linspace(-3.0, 3.0, 100)
X, Y = np.meshgrid(xlist, ylist)
Z = np.sqrt(X**2 + Y**2)
cp = plt.contourf(X, Y, Z)
plt.colorbar(cp) # Add a colorbar to a plot
plt.show()
```