# WEEK 3

**DATA VISUALIZATION USING MATPLOTLIB**

Data visualization refers to representing information in the form of visuals

Data Visualization can make your data speak! There is no doubt that when information is represented in the form of a picture like a graph or a chart, it can provide a much better understanding of the Data. Meaningful, effective, and aesthetically pleasing.

The key skill of a Data Scientist is to tell a compelling story after finding useful patterns and information from data. The plots and graphs can provide a clear description of the data. The Visuals can help support any claims you make based on the Data at hand.

They can be understood by any non-technical personnel which is the major advantage offered by them. While doing so, they let us convey most information while being very compact.

Data Visualization offers:

* Efficiency
* Clarity
* Accuracy

**list of Python data visualization tools**

* Matplotlib
* Seaborn
* Folium
* Plotly

## How to use the right visualization?

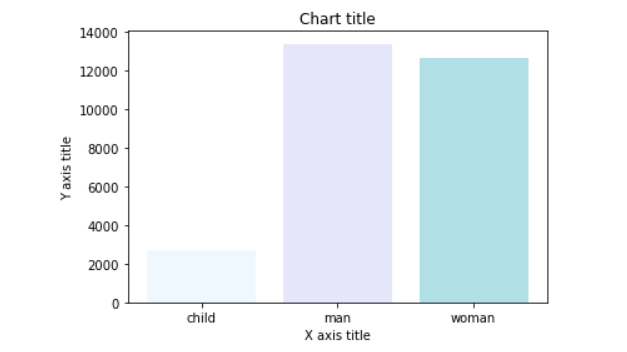
To extract the required information from the different visuals we create, it is quintessential that we use the correct representation based on the type of data and the questions that we are trying to understand. We will go through a set of most widely used representations below and how we can use them in the most effective manner.

### Bar chart

A bar chart is used when we want to compare metric values across different subgroups of the data. If we have a greater number of groups, a bar chart is preferred over a column chart.

Column charts are mostly used when we need to compare a single category of data between individual sub-items, for example, when comparing revenue between regions

#### Column chart using Matplotlib

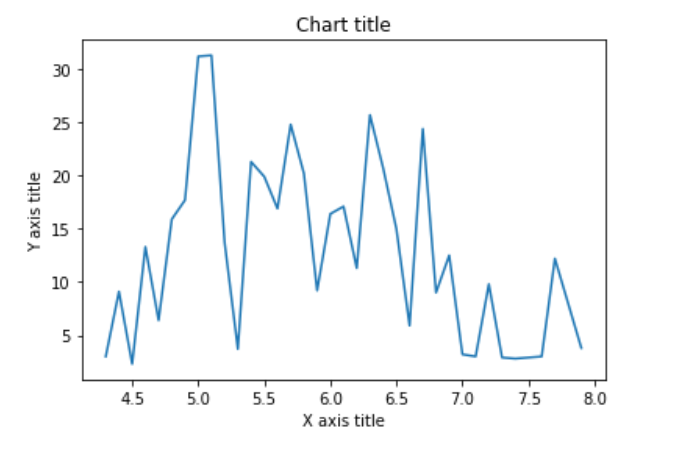


### Line chart

A line chart is used for the representation of continuous data points. This visual can be effectively utilized when we want to understand the trend across time.

### Pie chart

Pie charts can be used to identify proportions of the different components in a given whole.



### column histogram

### Scatter plot

Column histograms are used to observe the distribution for a single variable with few data points

**Scatter plots**

Scatter plots can be leveraged to identify relationships between two variables. It can be effectively used in circumstances where the dependent variable can have multiple values for the independent variable.

### Box plot

A box plot is used to show the shape of the distribution, its central value, and its variability.

import matplotlib.pyplot as plt

import pandas as pd

# initializing the data

x = [10, 20, 30, 40]

y = [20, 30, 40, 50]

# plotting the data

plt.plot(x, y)

# Adding the title

plt.title("Simple Plot")

# Adding the labels

plt.ylabel("y-axis")

plt.xlabel("x-axis")

plt.show()

#intializing data For BAR&PIE Chart

x = [ 25, 50, 36, 19]

y = ['Neural Network', 'Data analytics', 'Quantum Computing', 'Machine Learning']

#Bar graph

plt.bar([0.25,2.25,3.25,5.25,7.25],[300,400,200,600,700],

label="Y axis",color='b',width=0.5)

plt.bar([0.75,1.75,2.75,3.75,4.75],[50,30,20,50,60],

label="X axis", color='g',width=.5)

plt.legend()

plt.xlabel('Course')

plt.ylabel('Strudents ')

plt.title('Bar graph')

plt.show()

#pie chart

plt.pie(x, labels=y)

plt.title("Pie Chart")

cols = ['b','c','g', 'orange']

plt.pie(x,

labels =y,

colors = cols,

startangle = 90,

shadow = True,

explode =(0.1,0,0,0),

autopct ='%1.1f%%')

plt.show()

#Histogram

import matplotlib.pyplot as plt

# data to display on plots

x = [ 2, 3, 4, 5, 6, 7, 4]

y = [8, 8.5, 9, 9.5, 10, 10.5, 11]

# This will plot a simple histogram

plt.hist(x,)

# Title to the plot

plt.title("Histogram")

# Adding the legends

plt.legend(["bar"])

plt.show()

#Scatter plot

plt.scatter(x,y,label='Values of x&y',color='r')

plt.title('Smart Band Data Report')

plt.xlabel('x')

plt.ylabel('y')

plt.legend()

# Print the chart

plt.show()

# create a dataframe

data = pd.DataFrame({'Name': ['Akhil', 'Nikhil', 'Satyam', 'Sravan', 'Pavan'],

'Marks': [77, 95, 89, 78, 64],

'Credits': [8, 10, 9, 8, 7]})

# box plot

data['Marks'].plot(kind='box', title='Marks of students')

plt.show()

