Study Material for XII Computer Science on Data Visualization using Pyplot

Prepared by Kushal Roy

What is Data Visualization?

- ❖ Data visualization refers to the graphical or visual representation of information and data using visual elements like charts, graphs, maps etc.
- ❖ It is immensely useful in decision making to understand the meaning of data to drive business decision.
- ❖ For data visualization in Python, the **Matplotlib** library's pyplot is used.
- Pyplot is a collection of methods within matplotlib library(of python) to construct 2D plot easily and interactively.
- ❖ Matplotlib library is preinstalled with Anaconda distribution.
- ❖ In order to use pyplot for data visualization, user needs to import the pyplot module as follows:

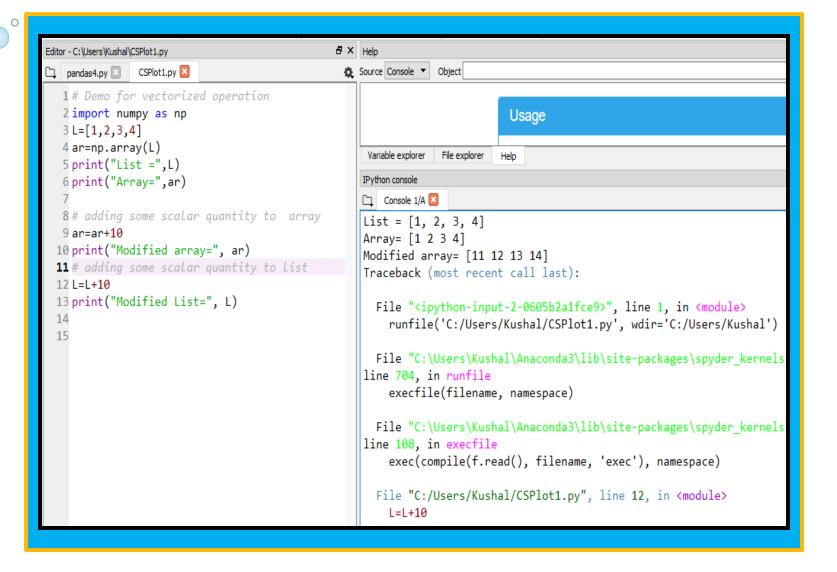
import matplotlib.pyplot as plt

Working with pyplot methods

- ❖ The pyplot interface provides many methods for 2D plotting of data in multiple ways such as line chart, bar chart, pie chart, scatter chart etc.
- ❖ But before proceeds, it is recommended to have an idea of using **numpy** library and some of it s functions, as:
 - ➤ **Numpy** offers some useful function to create arrays of data, which provides useful while plotting data.
 - ➤ Numpy arrays which is like a list supports **vectorized operation** that is not supported by list object in python.
- ❖ To use numpy arrays one needs to import the respective module as:
 import numpy as np
- ❖ We can also create a numpy array from a list object as follows:

```
import numpy as np
L= [1,2,3,4]
ar= np.array(L)
print(ar)
```

Vectorized operation



array vs. List: Comparison

* Both the data structures allow indexing, slicing, iterating and mutability.

But:

- List is built-in data structure in python where as to use an array in Python, we need to import this data structure from the *NumPy* package.
- ❖ List does not support vectorized operation while array supports the same.
- ❖ Arrays are great for numerical operations with compared to list
- ❖ Lists can be re-sized where array can not be
- ❖ List is heterogeneous where array is homogeneous structure
- ❖ Nested list is possible but nested array is not possible

Creating numpy array

```
<array name>=numpy.arange([start,] stop [,step] [,dtype])
         arr=np.arange(1,7,2,np.int32)
                                           In [4]: runfile('C:/Users/Kushal/CSPlot1.py',
         print(arr)-
         arr=np.arange(7)
                                          In [7]: runfile('C:/Users/Kushal/CSPlot1.py
[0 1 2 3 4 5 6]
         print(arr)
  <array name>=numpy.linspace(<start>, <stop>, <number of
                                                 values to be generated>)
         import numpy as np
         arr=np.linspace(2,3,6)
                                          In [11]: runfile('C:/Users/Kushal/CSPlot1.py'
         print(arr)
                                           [2. 2.2 2.4 2.6 2.8 3. ]
```

Plotting the values

```
1 import numpy as np
 2 import matplotlib.pyplot as plt
 3 arr=np.linspace(2,3,6)
 4 print("values are :", end=" ")
 5 print(arr)
 6 print("plotting the values in graph...")
 7 plt.plot(arr)
In [15]: runfile('C:/Users/Kushal/CSPlot1.py',
values are : [2. 2.2 2.4 2.6 2.8 3. ]
plotting the values in graph...
 3.0
 2.8
 2.6
 2.4
 2.2
 2.0
```

Some useful numpy functions:

Function Name	Description
sin(x)	Trigonometric sine, element wise
$\cos(x)$	Trigonometric cosine, element wise
arcsin(x)	Inverse sine element wise
floor(x)	Returns floor of the input, element wise
ceil(x)	Returns ceiling of the input, element wise
trunc(x)	Returns truncated value of the input, element wise
exp(x)	Calculate exponential of all values in the array
exp2(x)	Calculate 2**p for all p in input array
log(x)	Natural logarithm element wise
log10(x)	Returns base 10 log of the input array
log2(x)	Base 2 log of x

Example

```
Console 1/A 🔀
In [24]: runfile('C:/Users/Kushal/CSPlot1.py';
1.6
1.4
1.2
1.0
0.8
0.6
0.4
0.2
0.0
                              3.5
    10
         15
              2.0
                   2.5
                         3.0
                                   4.0
                                        4.5
                                             5.0
import numpy as np
import matplotlib.pyplot as plt
x=np.linspace(1,5,6)
y=np.log(x)
plt.plot(x,y)
```



***** Line chart:

- It displays information as a series of data points called 'Markers' by straight line segment using the **plot()**.
- A line chart is often used to visualize a trend in data over intervals of time

❖ Bar chart:

■ It is the graphical display of categorical data using bars of different height using **bar()**. It can be either vertical or horizontal

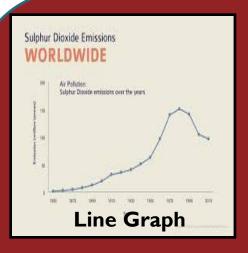
Pie chart:

• It is a circular statistical graph which is divided into slices to illustrate the numerical proportion.

Scatter chart:

 Scatter plots are used to plot data points on horizontal and vertical axis in the attempt to show how much one variable is affected by another(correlation).

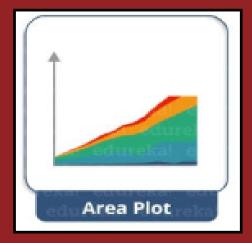
Commonly used chart type:

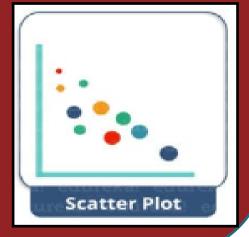












Customization of Plot()

import matplotlib.pyplot as plt

x = [1,2,3,4,5,6]

y = [2,4,1,5,2,6]

plt.plot(x, y, color='green', linestyle='dashed', linewidth = 3, marker='o', markerfacecolor='blue', markersize=12)

plt.ylim(1,8)

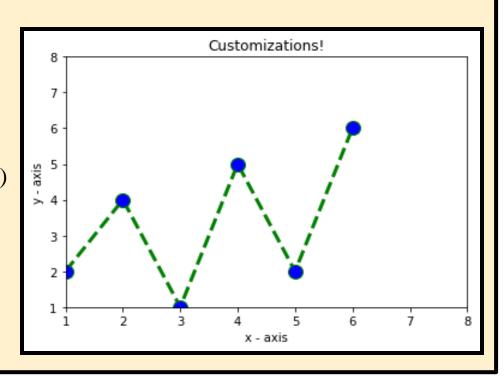
plt.xlim(1,8)

plt.xlabel('x - axis')

plt.ylabel('y - axis')

plt.title('customizations!')

plt.show()

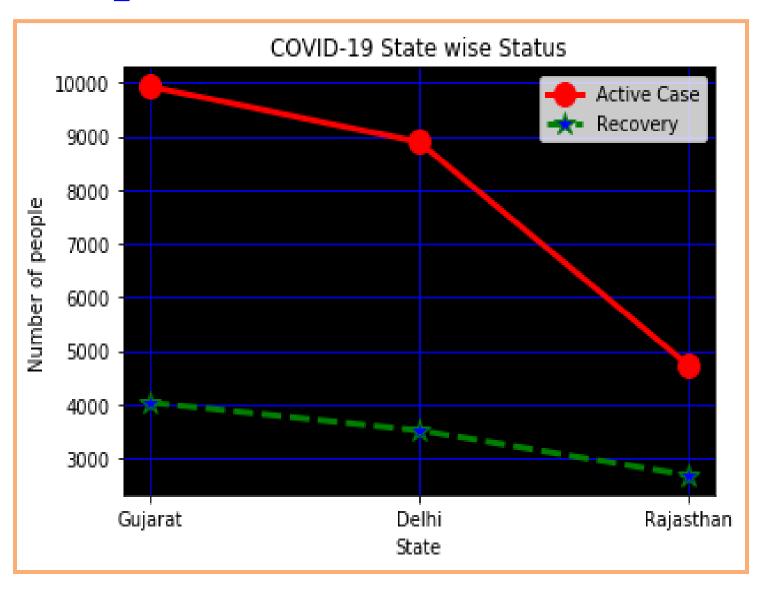


What is legend?

- The **legend** of a **graph** reflects the data displayed in the **graph's** Y-axis, also called the **graph** series.
- With a **chart**, a **legend** is an area of a **chart** describing each of the parts of the **chart**.
- A Legend is a representation of legend keys or entries on the plotted area of chart or graph which are linked to the data table of the chart or graph.

```
import matplotlib.pyplot as plt
       = ['Gujarat', 'Delhi', 'Rajasthan']
state
active = [9931,8895,4727]
recovery = [4035,3518,2677]
plt.plot(state, active, color='red', linestyle='solid',
     linewidth = 3, marker='o', markerfacecolor='red', markersize=12)
plt.plot(state, recovery, color='green', linestyle='dashed',
     linewidth = 3, marker='*', markerfacecolor='blue', markersize=12)
plt.xlabel('State')
plt.ylabel('Number of people')
plt.title('COVID-19 State wise Status')
plt.legend(['Active Case', 'Recovery'])
plt.grid(True, color='blue')
ax=plt.axes()
ax.set_facecolor("k")
plt.show()
```

Output



import matplotlib.pyplot as plt

langs = ['C', 'C++', 'Java', 'Python', 'PHP']

students = [23,17,35,29,12]

plt.bar(langs, students, color = ['red', 'green', 'blue', 'magenta', 'orange'])

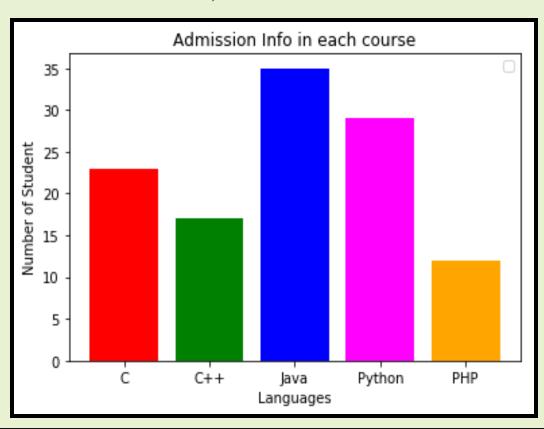
plt.xlabel('Languages')

plt.ylabel('Number of Student')

plt.title('Admission Info in each course')

plt.show()

Bar Chart



import matplotlib.pyplot as plt

frequencies

ages= [2,5,70,40,30,45,50,45,43,40,44,60,7,13,57,18,90,77,32,21,20,]

setting the ranges and no. of intervals

range = (0, 100)

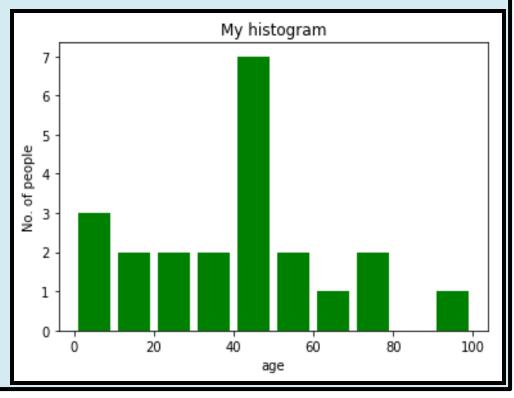
bins = 10

plotting a histogram

plt.hist(ages, bins, range, color = 'green', histtype = 'bar', rwidth=0.8)

plt.xlabel('age')
plt.ylabel('No. of people')
plt.title('My histogram')
plt.show()

Histogram



Difference between Bar Graph & Histogram

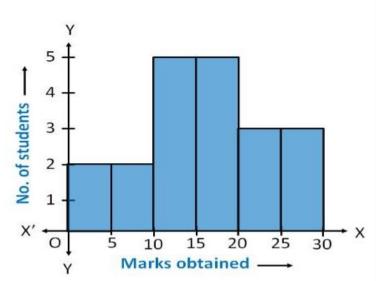




In Bar Graph

- Bars have equal space
- On the y-axis, we have numbers
 & on the x-axis, we have data
 which can be anything.

Histogram



In Histogram

- Bars are fixed
- On the y-axis, we have numbers & on the x-axis, we have data which is continuous & will always be number.

import matplotlib.pyplot as plt

x-axis values

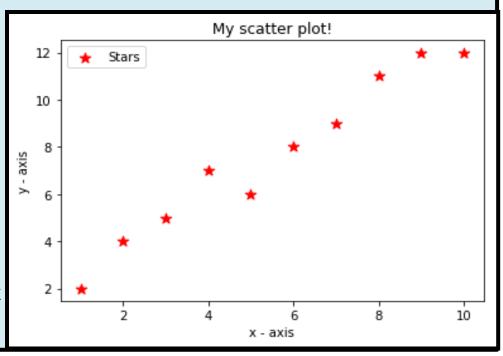
x = [1,2,3,4,5,6,7,8,9,10]

y-axis values

y = [2,4,5,7,6,8,9,11,12,12]

plotting points as a scatter plot plt.scatter(x, y, label= "Stars", color= "red", marker= "*", s=80)

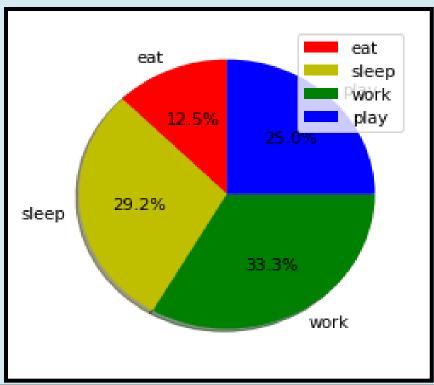
x-axis label
plt.xlabel('x - axis')
frequency label
plt.ylabel('y - axis')
plot title
plt.title('My scatter plot!')
showing legend
plt.legend()
function to show the plot
plt.show()



Scatter Diagram

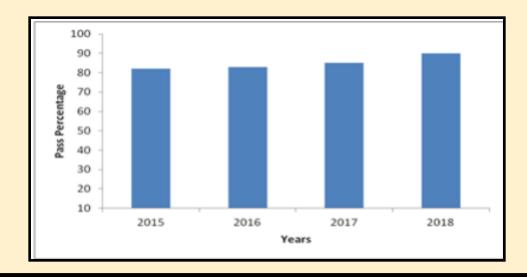
Pie Chart

plt.show()



Assignment 1:

- 1. A bar chart is drawn(using pyplot) to represent sales data of various models of cars, for a month. Write appropriate statements in Python to provide labels **Month June** and **Sale done** to x and y axis respectively.
- 2. Write a code to plot a bar chart to depict the pass percentage of students in CBSE exams for the years 2015 to 2018 as shown below



Assignment 2:

Give the output from the given python code: import matplotlib.pyplot as plt import numpy as np objects = ('Python', 'C++', 'Java', 'Perl', 'Scala', 'Lisp') = np.arange(len(objects)) y_pos performance = [10,8,6,4,2,1]plt.bar(y_pos, performance, align='center', alpha=0.5) plt.xticks(y_pos, objects) plt.ylabel('Usage') plt.title('Programming language usage') plt.show()