Experiment-2

Student Name: YANA SRIVASTAVA **UID: 20BCS2279**
Branch: CSE **Section/Group: 906/B**
Semester: 5th **Date of Performance: 18/08/2022**
Subject Name: ML Lab **Subject Code: 20CSP-317**

1.Aim/Overview of the practical: Implement Data Visualization.

2. Task to be done:

- Draw co-ordinates in pyplot.
- Draw line in pyplot.
- Draw curve with numpy array with the help of arange function in lower stride.
- Draw comparison graph of x^2 & x^3.
- Draw Bubble chart in pyplot.
- Draw pie chart in pyplot.
- Draw histogram in pyplot.
- Draw Bar graph in pyplot.

3. Algorithm/Flowchart (For programming based labs):

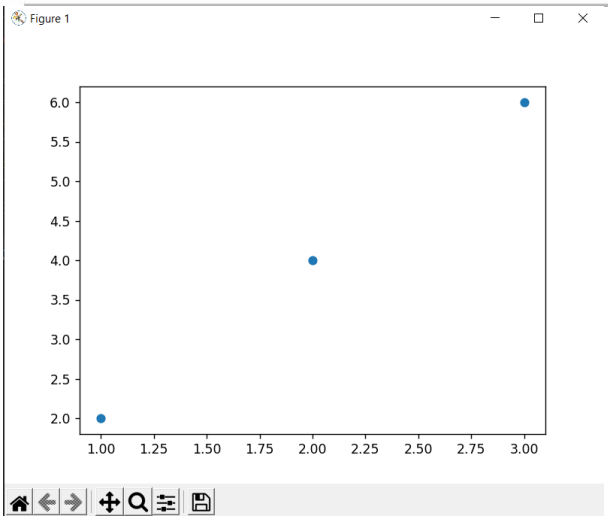
- **Analyse the data very well cause type of graph is decided by the kind of data ONLY.**
- **If we require to look trends between fewer data points: Go for line & bar graph.**
- **When you are sure about what graph is appropriate for your data, rite code for creating the basic graph.**
- **Then enhance its indication, label, details using: text, label, grid, legend, title, axis name, rotation (if required) etc.**

4. CODE:

```
#CHIRAG BITHER
#20BCS1838
# y = 2x graph :
import matplotlib.pyplot as plt

x = [1,2,3]
y = [2,4,6]
# Scatter points all over the graph :
# Takes 2 , 1D list of same dimension :
plt.scatter(x,y)
# To show graph :
plt.show()
```

5. OUTPUT:

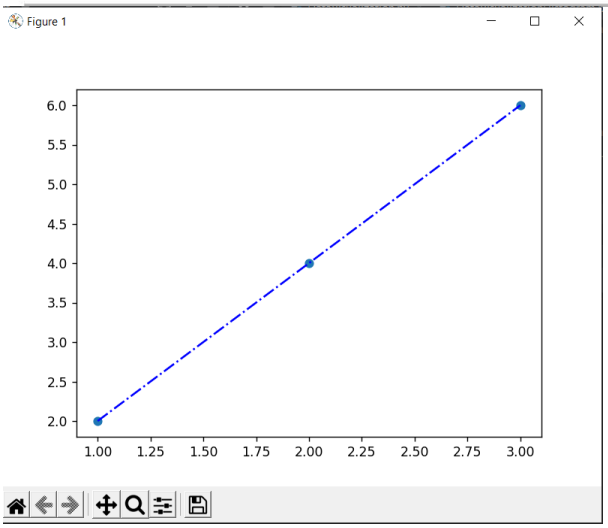


6. CODE:

```
# y = 2x : ( Superimposed co-ordinates & line )
import matplotlib.pyplot as plt

x = [1,2,3]
y = [2,4,6]
# Scatter points all over the graph :
# Takes 2 , 1D list of same dimension :
plt.scatter(x,y)
# To make a line we use plot :
plt.plot(x,y,"b-.")
# To show graph :
plt.show()
```

7. OUTPUT:



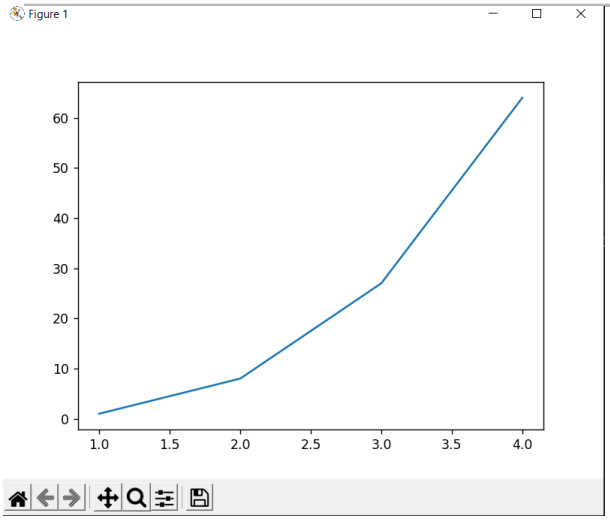
8. CODE:

```
# y = x^3 : ( Not smooth just combination of lines making curve )
import matplotlib.pyplot as plt
import numpy as np

x = np.array([1,2,3,4])

y = x**3
plt.plot(x,y)
plt.show()
```

9. OUTPUT:



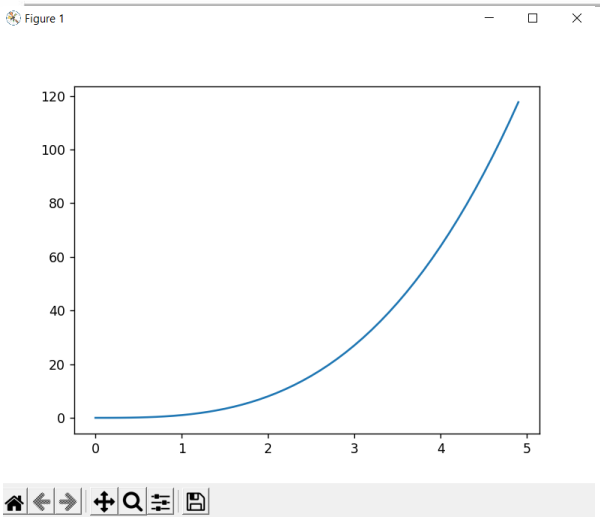
10. CODE:

```
# For smooth y = x^3 :

import matplotlib.pyplot as plt
import numpy as np

# arange arguments : [start,end) , stride/step/increment rate
x = np.arange(0,5,0.1)
y = x**3
plt.plot(x,y)
plt.show()
```

11. OUTPUT:



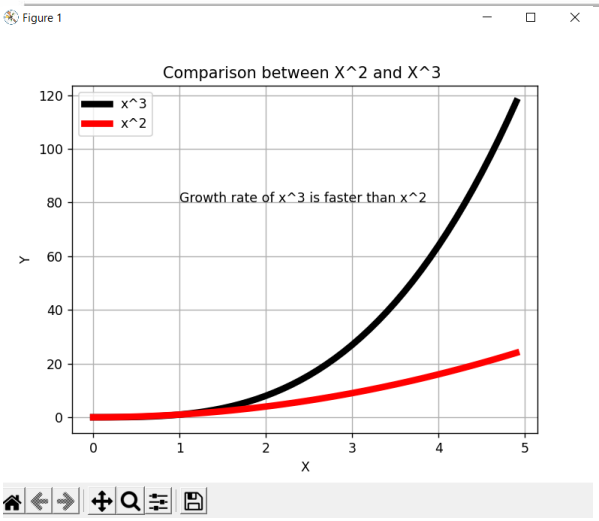
12. CODE:

```
import matplotlib.pyplot as plt
import numpy as np
x = np.arange(0,5,0.1)
y = x**3
y2 = x**2

plt.plot(x,y,color= 'black',linewidth = 5,label = "x^3")
plt.plot(x,y2,color= 'red',linewidth = 5 , label = "x^2")

plt.ylabel("Y")
plt.xlabel("X")
plt.legend()
plt.grid()
plt.text(1,80, "Growth rate of x^3 is faster than x^2")
plt.title("Comparison between X^2 and X^3 ")
plt.show()
```

13. OUTPUT:

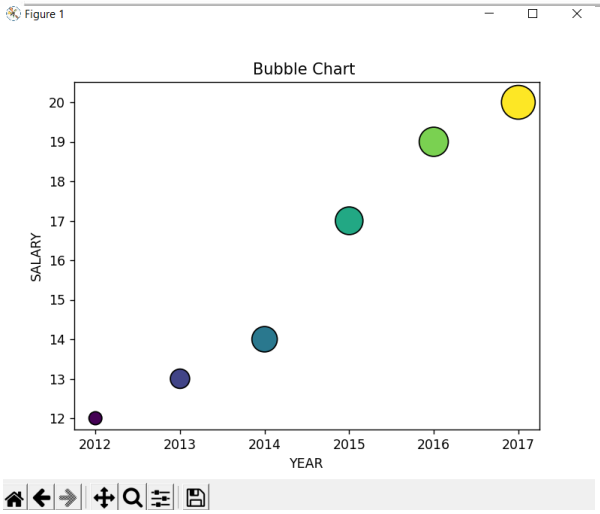


14. CODE:

```
import matplotlib.pyplot as plt
import numpy as np

year = [2012,2013,2014,2015,2016,2017]
salary = [12,13,14,17,19,20]
population = [100,220,380,450,500,670]
color = np.arange(len(year))
plt.xlabel("YEAR")
plt.ylabel("SALARY")
plt.scatter(year,salary,s = population , c = color ,alpha = 1 , edgecolors= "black",marker="o")
plt.title("Bubble Chart ")
plt.show()
```

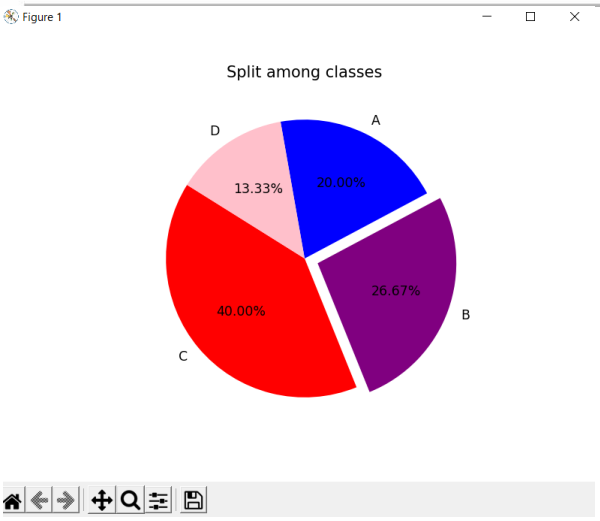
15. OUTPUT:



16. CODE:

```
# Pie Graphs :
import matplotlib.pyplot as plt
labels = ["A","B","C","D"]
sizes = [3,4,6,2]
colors = ["blue","purple","red","pink"]
explode = [0.,0.1,0,0]
plt.title("Split among classes")
plt.pie(sizes,colors = colors , explode = explode ,labels =
labels,autopct="%.2f%%",counterclock=False,startangle=100)
plt.show()
```

17. OUTPUT:

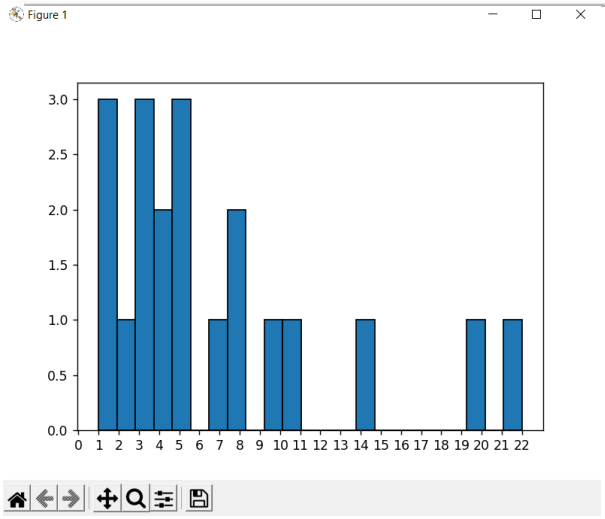


18. CODE:

```
# It is very useful when we want to see Frequency of data.
import matplotlib.pyplot as plt
import numpy as np

dataSet = [1,2,1,3,5,4,5,5,4,3,3,10,7,8,8,20,22,11,1,14]
xtickss = np.arange(23)
plt.hist(dataSet,bins = 23 , edgecolor = "black")
# bins  = (max. no. - min. no.)/total no. of bins
plt.xticks(xtickss)
plt.show()
```

19. OUTPUT:

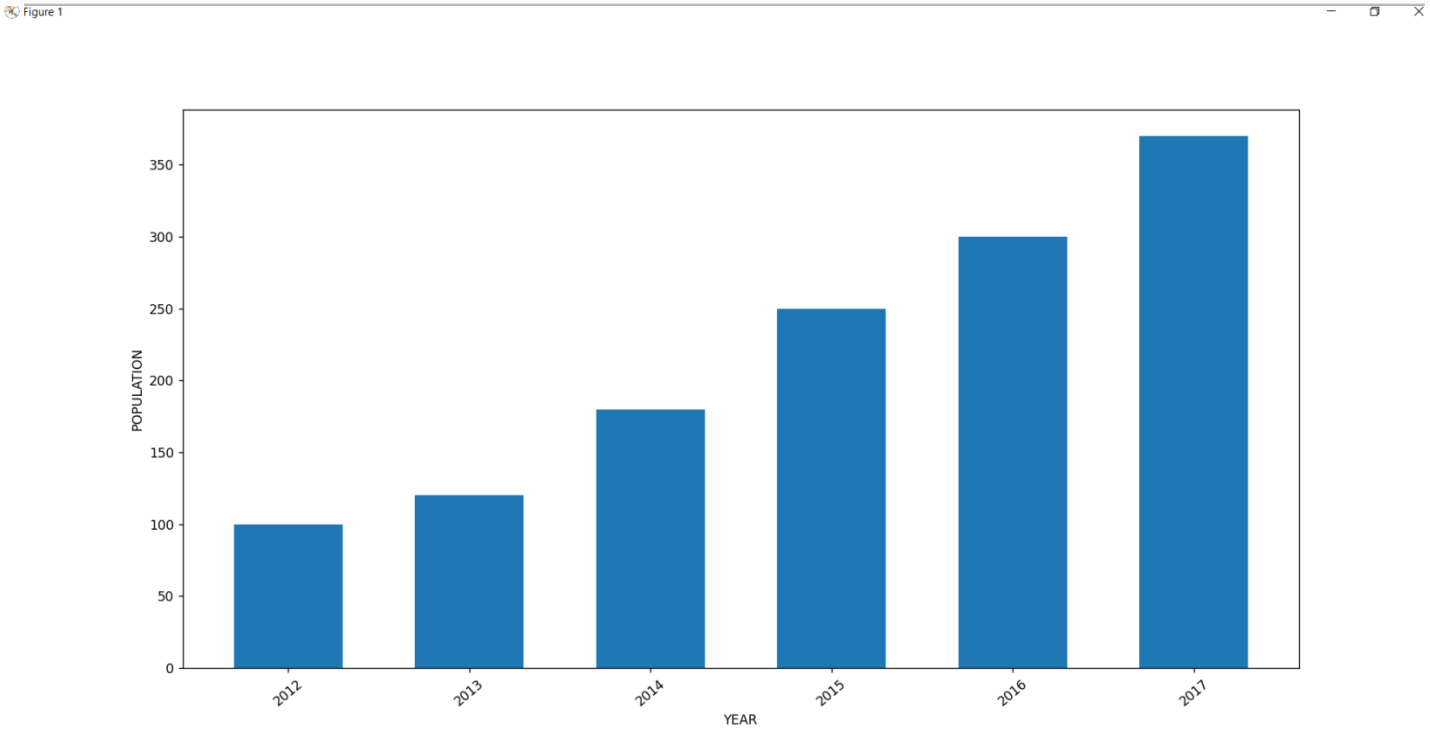


20. CODE:

```
# Bar Graph :
import matplotlib.pyplot as plt

year = [2012,2013,2014,2015,2016,2017]
salary = [12,13,14,17,19,20]
population = [100,120,180,250,300,370]
plt.xlabel("YEAR")
plt.ylabel("POPULATION")
plt.bar(year,population,width=0.6)
plt.xticks( rotation =40 )
plt.show()
```

21. OUTPUT:



Learning outcomes (What I have learnt):

1. learned different type of matplotlib function & in function specifically arguments need to be passed for better representation.

2. Application of numpy array to create smooth curves, bins for histogram & xticks in graph.

3. I learned in which type of data, which graph suits best like in frequency related data histogram, scatter datapoint to curve fitting representation with plot function, to represent three parameters in 2D graph with bubble graph & to show trend in data we use bar graph.

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			