TRIBHUVAN UNIVERSITY

INSTITUTE OF SCIENCE AND TECHNOLOGY

HIMALAYA COLLEGE OF ENGINEERING

CHYASAL,LALITPUR

Lab Report No:- 6

Title:- Introduction to pyplot using matplotlib.

Submitted by:-

Name:-Surajan shrestha

Roll No:- 38

Submitted To:- Rubas mali

Department Of CSIT

Checked by:-

Date of submission:- 2081/06/02

TITLE: Introduction to pyplot using matplotlib.

OBJECTIVE:

The Objective of this lab is to learn how to visualize data using matplotlib's pyplot library python. The goal is to understand how to create different types of plots such as line plots, bar graphs, scatter plots, and histograms, and customize them effectively.

THEORY:

→ Matplotlib is one of the most popular libraries in python for data visualization. pyplot is a module within matplotlib that provides a MATLAB-like interface for creating plots and charts.

- Line plot: A type of plot that displays information as a series of data points connected by straight line segments.
- Bar plot: A chart with rectangle bars where the length of each bar is proportional to the value it represents.
- Scatter plot: A plot that uses dots to represent values for two different variables.
- Histogram: A representation of the distribution of numerical data.

Basic pyplot function:

1, plt.plot(): used for creating line plots.

2, plt.bar(): used for creating bar charts.

3, plt.scatter(): used for creating scatter plots.

4, plt.hist() : used for creating histograms.

5, plt.xlabel(), plt.ylabel(), plt.title(): used for labelling axes and setting a title.

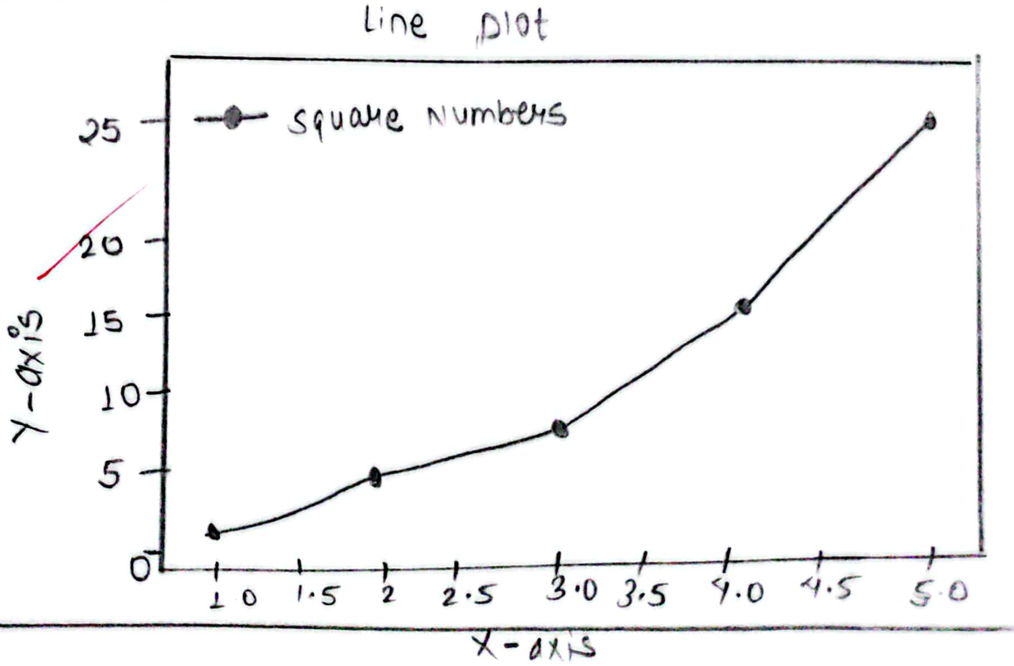
6, plt.legend() : used for adding legends.

7, plt.show() : used for displaying the plot.

OBSERVATION:

```
//Line plot
 import matplotlib.pyplot as plt
   x = [ 1, 2 , 3, 4,5]
   Y = [1,4, 9,16,25]
  plt.plot (x, Y, label ='square Numbers', color
   ='blue', marker ='0')
  plt.xlabel ('X-axis label')
  plt.ylabel ('Y-axis label')
  plt.title ('Line plot Example')
  plt.legend ( )
  plt.show( )
```

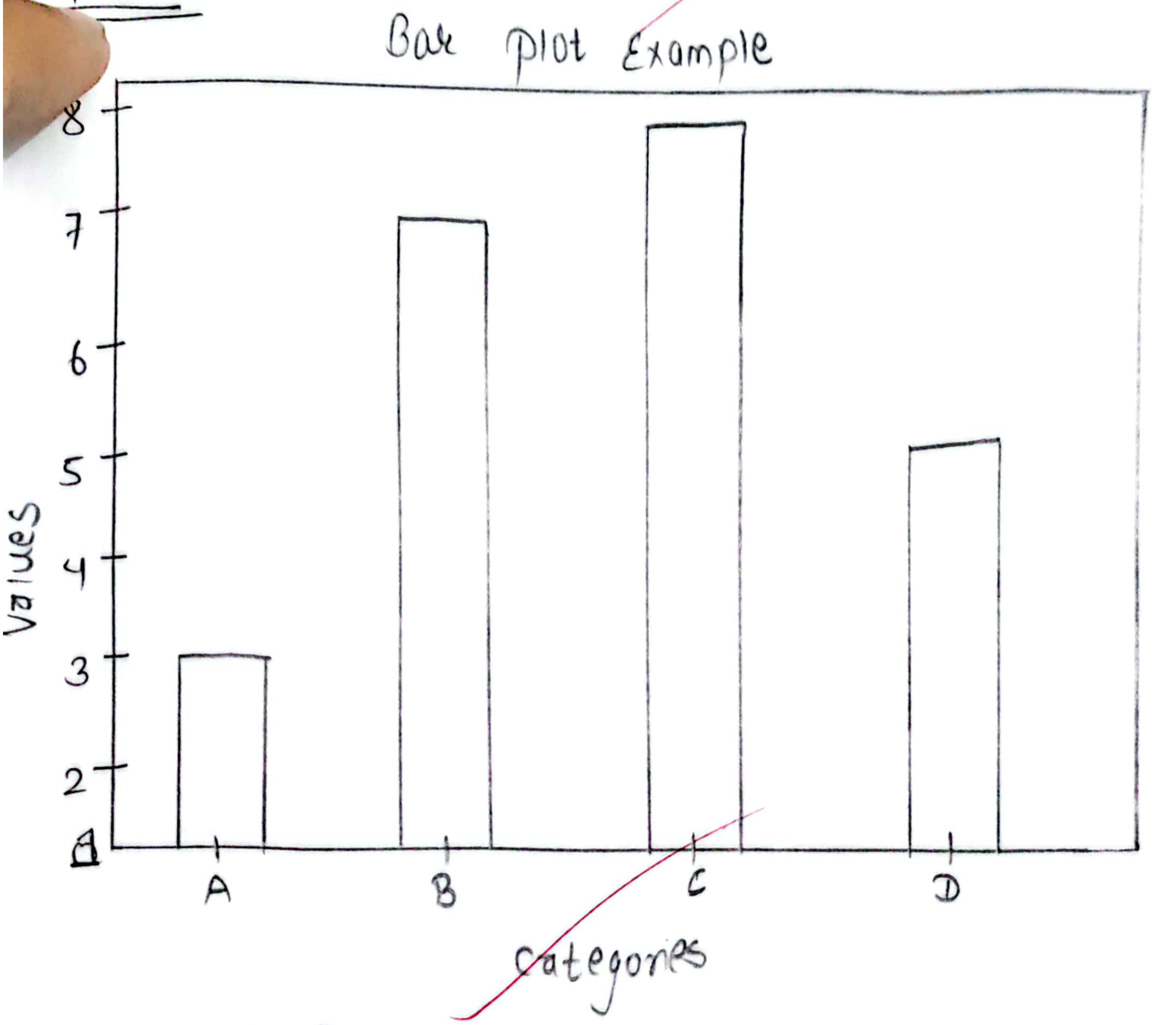
OUTPUT:



// Bar plot

```
import matplotlib.pyplot as plt
categories = ['A', 'B', 'C', 'D']
values = [3,7,8,5]
plt.bar(categories, values, color='green')
plt.xlabel('Categories')
plt.ylabel('values')
plt.title('Bar plot Example')
plt.show()
```

OUTPUT:

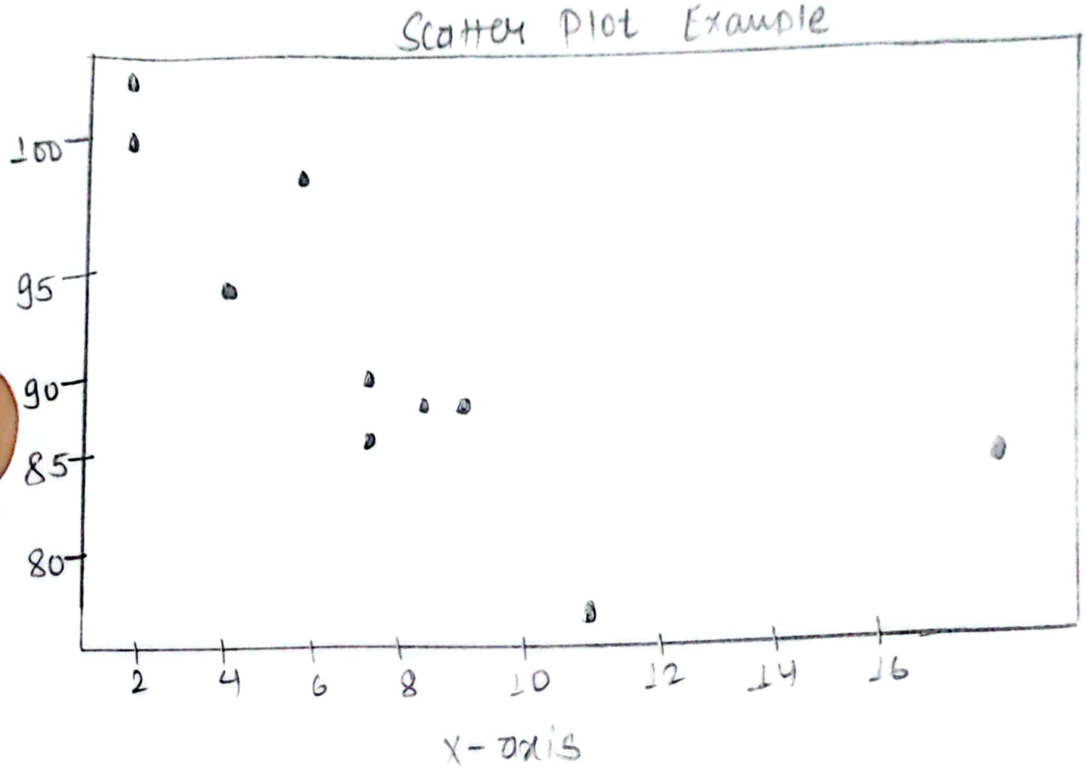


// scatter plot:

```
import matplotlib.pyplot as plt
x= [5,7,8,7,2,17,2,9,4,11]
y= [99,86,87,88,100,86,103,87,94,78]
```

```
plt.scatter(x,y,color='red')
plt.xlabel('x-axis')
plt.ylabel('y-axis')
plt.title('scatter plot Example')
plt.show()
```

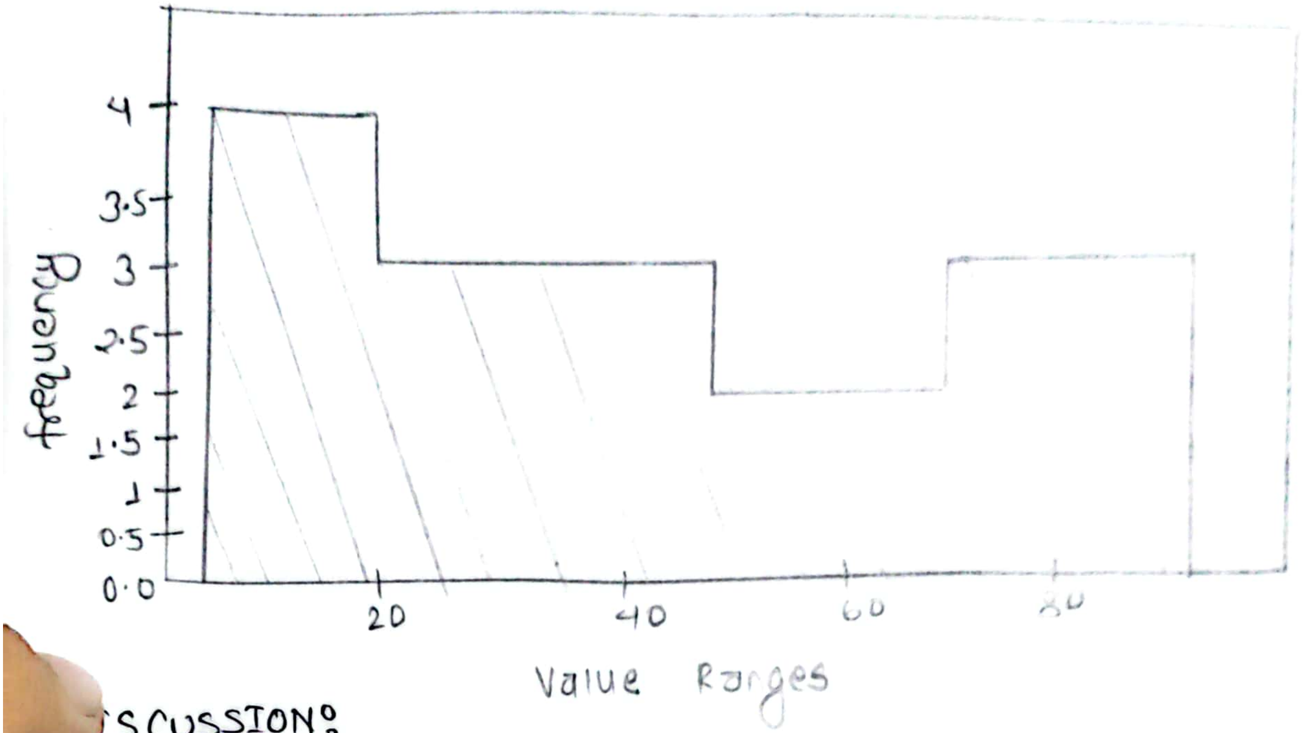
OUTPUT:



// Histogram

```
  import matplotlib.pyplot as plt
data = [22,87,5,43,56,73,55,54,11,20,51,5,79,31,
        27]
plt.hist(data, bins=5, color='purple')
plt.xlabel('value Ranges')
plt.ylabel('frequency')
plt.title('Histogram Example')
plt.show()
```

OUTPUT:



SCUSSION:

→ pylot offers users to customize plots by setting labels, titles and adjusting colors and markers for clarity. Each plot type serves a unique purposes: line plots show trends over time, bar charts compare quantities accross categories, scatter plots reveal the relationships between variables and histograms displays data distribution. while scatter plots and line plots can look similar, scatter plots the emphasize discrete points, while line plots indicat continuity.

Checked

CONCLUSION:

→ Pyplot is a versatile & powerful tool for creating various types of plots in python. It allows users to visualize data effectively and provides extensive customization options for enhancing the presentation of plots. Through this lab, we learned how to generate line plots, bar, charts, scatter plots and histograms.