RNS INSTITUTE OF TECHNOLOGY

(AICTE Approved, VTU Affiliated and NAAC 'A' Accredited)

UG Programs - CSE, ECE, ISE, EIE and EEE have been Accredited by NBA for three Academic years

DR. VISHNUVARDHAN ROAD, CHANNASANDRA, RR NAGAR POST, BENGALURU – 560098

ESTD : 2001

An Institute with a Difference

DATA VISUALIZATION WITH PYTHON LABORATORY MANUAL

For Third Semester B.E [VTU/CBCS, 2022 syllabus]

Subject Code – BCS358D

NAME:_____

BRANCH:_____

SECTION:_____

USN:

VISION AND MISSION OF INSTITUTION

Vision

Building RNSIT into a World Class Institution

Mission

To impart high quality education in Engineering, Technology and Management with a Difference, Enabling Students to Excel in their Career by

1. Attracting quality Students and preparing them with a strong foundation in fundamentals so as to achieve distinctions in various walks of life leading to outstanding contributions
2. Imparting value based, need based, choice based and skill based professional education to the aspiring youth and carving them into disciplined, World class Professionals with social responsibility
3. Promoting excellence in Teaching, Research and Consultancy that galvanizes academic consciousness among Faculty and Students
4. Exposing Students to emerging frontiers of knowledge in various domains and make them suitable for Industry, Entrepreneurship, Higher studies, and Research & Development
5. Providing freedom of action and choice for all the Stake holders with better visibility

DATA VISUALIZATION WITH PYTHON LABORATORY
BCS358D
INTERNAL EVALUATION SHEET

EVALUATION (MAX MARKS 40)				
TEST A	**REGULAR EVALUATION (OBSERVATION+RECORD) B**	**VIVA B**	**CERTIFICATION C**	**TOTAL MARKS A+B+C+D**
20	**10**	**10**	**10**	**50**

R2: REGULAR LAB EVALUATION (OBSERVATION+RECORD) RUBRIC (MAX MARKS 10)				
Sl. No.	**Parameters**	**Good**	**Average**	**Needs improvement**
a.	**Understanding of problem (3 marks)**	Clear understanding of problem statement while designing and implementing the program (3)	Problem statement is understood clearly but few mistakes while designing and implementing program (2)	Problem statement is not clearly understood while designing the program (1)
b.	**Writing program (4 marks)**	Program handles all possible conditions (4)	Average condition is defined and verified. (3)	Program does not handle possible conditions (1)
c.	**Result and documentation (3 marks)**	Meticulous documentation and all conditions are taken care (3)	Acceptable documentation shown (2)	Documentation does not take care all conditions (1)

R3: REGULAR LAB EVALUATION VIVA RUBRIC (MAX MARKS 10)					
Sl. No.	**Parameter**	**Excellent**	**Good**	**Average**	**Needs Improvement**
a.	**Conceptual understanding (10 marks)**	Answers 80% of the viva questions asked(10)	Answers 60% of the viva questions asked (7)	Answers 30% of the viva questions asked (4)	Unable to relate the concepts (1)

R4: CERTIFICATION RUBRIC (MAX MARKS 10)				
Sl. No.	**Parameters**	**Excellent**	**Good**	**Needs Improvement**
a.	**Design, implementation and demonstration (10 marks)**	Student has passed the assessment with 65% or more percentage and are certified (10)	Student has completed the course with less than 65% in assessment and has attended all sessions (5)	Student has not taken up assessment and has attended all sessions (3)

TEST /LAB INTERNALS MARKS(MAX MARKS 20)

TEST #	Write up 6	Execution 28	Viva 6	Sign	Total 40	Avg. 40	Final 10
TEST-1							
TEST-2						40	20

REGULAR LAB EVALUATION (MAX MARKS 10)

Lab program	Date of Execution	Additional programs	Observation (5)	Record (5)	Viva (10)	Total 20	Teacher Signature
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
Total Marks		—		20			—

Final Marks obtained from test (20) + regular evaluation (20) + Certification (10)	**50**	**Lab in charge :** **HOD :**

PREFACE

We have developed this comprehensive laboratory manual on **Data Visualization with Python Programming Lab** with two primary objectives: Use Python programming constructs to develop programs for solving real-world problems and Demonstrate working with Seaborn, Bokeh for visualization.

This material is divided into two parts: Part-A and Part-B, which provides the students an exposure to problem solving approaches and solution to number of problems using Python programming language. The problems discussed in this manual comprises of a problem and its programming solution. Viva questions, frequently appeared examination questions and practicing programming problems constitute an indispensable part of this material.

Our profound and sincere efforts will be fruitful only when students acquire the extensive knowledge by reading this Data Visualization with Python manual and apply the concepts learnt apart from the requirements specified in Data Visualization with Python Programming Laboratory as prescribed by VTU, Belagavi.

Department of CSE

TABLE OF CONTENTS

ACKNOWLEDGMENT

A material of this scope would not have been possible without the contribution of many people. We express our sincere gratitude to Dr. R N Shetty, Former Chairman, RNS Group of Companies and Karan Shetty, Chairman RNS Group of Companies for their magnanimous support in all our endeavors.

We are grateful to Dr.Ramesh Babu H S, Principal, Dr M K Venkatesha, Director RNSIT, Dr. P Kiran, HOD, CSE for extending their constant encouragement and support.

Our heartfelt thanks to Mrs. Soumya C S, Mrs. Meenakshi S J and Ms Likitha R for their unparalleled contribution throughout the preparation of this comprehensive manual. We also acknowledge our colleagues for their timely suggestions and unconditional support.

Department of CSE

LABORATORY PROGRAMS

Lab Programs:

1.a)Write a python program to find the best of two test average marks out of three test's marks accepted from the use

```
m1 = int(input("Enter marks for test1 : "))
m2 = int(input("Enter marks for test2 : "))
m3 = int(input("Enter marks for test3 : "))
if m1 <= m2 and m1 <= m3:
avgMarks = (m2+m3)/2
elif m2 <= m1 and m2 <= m3:
avgMarks = (m1+m3)/2
elif m3 <= m1 and m2 <= m2:
avgMarks = (m1+m2)/2
print("Average of best two test marks  out of three test's marks is", avgMarks);
```

OR

```
m1=int(input("Enter the test1 Marks"))
m2=int(input("Enter the test2 Marks"))
m3=int(input("Enter the test3 Marks"))
L1=[]
L1.append(m1)
L1.append(m2)
L1.append(m3)
L1.sort()
print(L1)
Avg=(L1[1]+L1[2])/2
print("Average of best two test marks  out of three test's marks is ",Avg)
```

output

```
Enter the test1 Marks 25
Enter the test2 Marks 25
Enter the test3 Marks 12
[12, 25, 25]
Average of Best of two Test is 25.0
```

OR

```
m1=int(input("Enter the test1 Marks"))
m2=int(input("Enter the test2 Marks"))
m3=int(input("Enter the test3 Marks"))
Minimum=min(m1,m2,m3)
sum=m1+m2+m3-Minimum
Avg=sum/2
print("Average of best two test marks  out of three test's marks is ",Avg)
```

1.b)Develop a Python program to check whether a given number is palindrome or not and #also count the number of occurrences of each digit in the input number.

```
val = int(input("Enter a value : "))
str_val = str(val)
if str_val == str_val[::-1]:
   print("Palindrome")
else:
print("Not Palindrome")

for i in range(10):
   if str_val.count(str(i)) > 0:
      print(str(i),"appears", str_val.count(str(i)), "times");
```

OR

```
num=input("Enter the Number")
rev=num
if (rev==rev[::-1]):
        print("The Number is Palindrome", num)
else:
        print("The Number  is not Palindrome", num)
for i in set(num):
        print(i, "appears", num.count(str(i)),"times")
```

output

```
Enter the Number12321
The Number is Palindrome 12321
1 appears 2 times
3 appears 1 times
2 appears 2 times
```

2. a) Defined as a function F as Fn = Fn-1 + Fn-2. Write a Python program which accepts a value for N (where N >0) as input and pass this value to the function. Display suitable error message if the condition for input value is not followed.

```
def Fibonacci(n):
   if n==1:
      return 0
elif n==2:
      return 1
   else:
      return (Fibonacci(n-1)+Fibonacci(n-2))


num=int(input("Enter the number\n"))
if num>0:
   res=Fibonacci(num)
```

```
print("Fibonacci of ", num ,"is",res)
else:
print("Error in the  input")
```

Output

```
Enter the number  5
Fibonacci of  5 is 3

Enter the number -9
Error in input
```

2.b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions.

```
def Bin2dec(bin):
    l=len(bin)
dec=0
    for i in range(l):
dec+=int(bin[i])*(2**(l-i-1))
    return dec

def oct2hex(oct):
    l=len(oct)
dec=0
    for i in range(l):
dec+=int(oct[i])*(8**(l-i-1))
    hexa=['0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F']
octhex=' '
    while dec>0:
        rem=dec%16
octhex=hexa[rem]+octhex
dec=dec//16
    return octhex

bin=input("Enter the Binary Number")
print("Binary to Decimal is ",Bin2dec(bin))
oct=input("Enter the octal Number")
print("Octal to Decimal is",oct2hex(oct))
```

OUTPUT:

```
Enter the Binary Number1011
Binary to Decimal is  11
Enter the octal Number12
Octal to Decimal is A
```

3. a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.

```
    sentence = input("Enter a sentence : ")
    digCnt = upCnt = loCnt =wordcnt=0
    wordcont=sentence.split()
```

```
for ch in sentence:
   if ch>='0' and ch<='9':
digCnt += 1
   if ch>='A'andch<='Z':
upCnt += 1
   if ch>='a' and ch<='z':
loCnt += 1

print("This sentence has\n")
print("words: ", len(wordcont),"\n" "digits",digCnt, )
print("upper case letters",upCnt, "\n" "lower case letters ",loCnt)
```

Output

```
Enter a sentence : Hello RNSIT, 2001 &
This sentence has

words:  4
digits 4
upper case letters 6
lower case letters  4
```

3.b) Write a Python program to find the string similarity between two given strings

```
def compare(s,p):
   count=0
     n=min(len(s),len(p))
   for i in range(n):
      if s[i]==p[i]:
         count+=1
   return count
s1 = input("Enter String 1 \n")
s2 = input("Enter String 2 \n")
mx=max(len(s1),len(s2))
count=compare(s1,s2)
similarity=count/mx*100
      print ("Total number letter matched is",count)
      print("simirality between two string is",similarity)
```

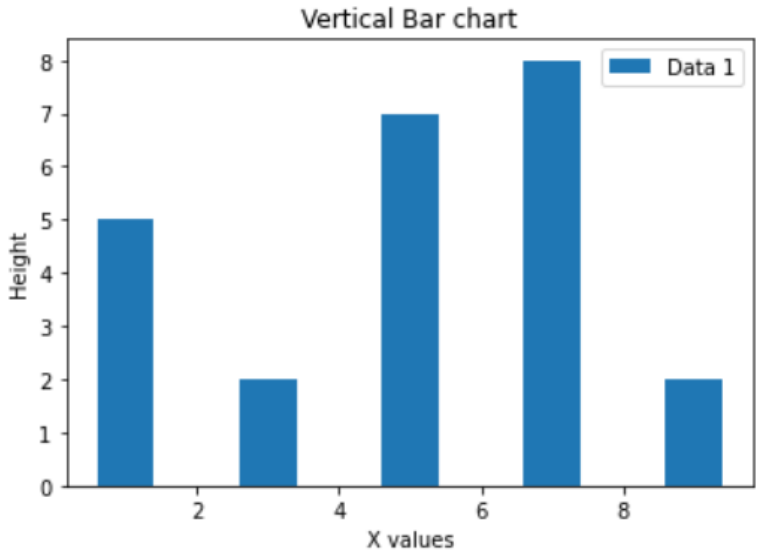
Output

```
Enter String 1
rnsit
Enter String 2
rnsit
Total number letter matched is 5
simirality between two string is 100.0

Enter String 1
welcome
Enter String 2
rnsit
Total number letter matched is 0
simirality between two string is 0.0
```

4.a)Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.

```
import matplotlib.pyplot as plt
plt.bar([1,3,5,7,9],[5,2,7,8,2], label="Data 1")
plt.legend()
# The following commands add labels to our figure.
plt.xlabel('X values')
plt.ylabel('Height')
plt.title('Vertical Bar chart')
plt.show()
```

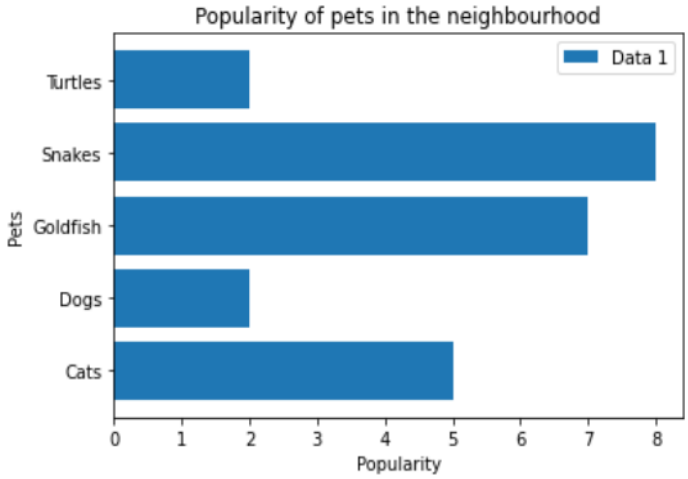
OUTPUT



Or

```
import matplotlib.pyplot as plt
plt.barh(["Cats","Dogs","Goldfish","Snakes","Turtles"],
[5,2,7,8,2], align='center', label="Data 1")
plt.legend()

plt.ylabel('Pets')
plt.xlabel('Popularity')
plt.title('Popularity of pets in the neighbourhood')
plt.show()
```

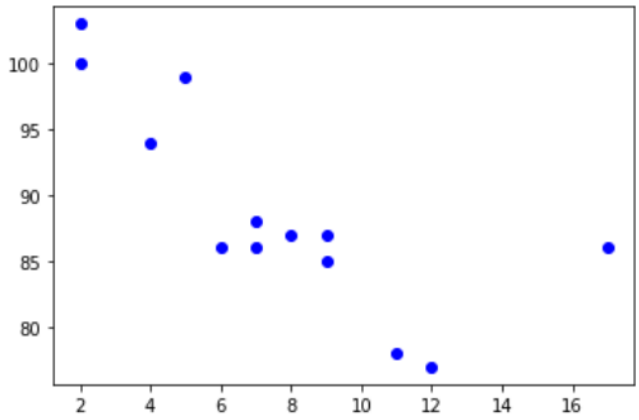
OUTPUT



4 b) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib.

```
import matplotlib.pyplot as plt

x =[5, 7, 8, 7, 2, 17, 2, 9, 4, 11, 12, 9, 6]

y =[99, 86, 87, 88, 100, 86,    103, 87, 94, 78, 77, 85, 86]

plt.scatter(x, y, c ="blue")

plt.show()
```

OUTPUT

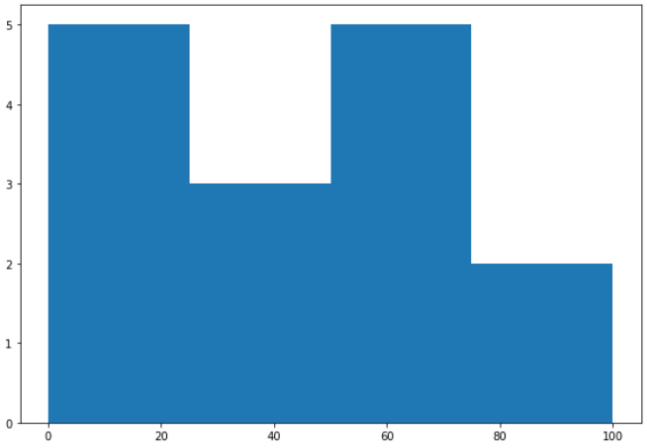


5 a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib.

```
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])
fig, ax = plt.subplots(figsize =(10, 7))
ax.hist(a, bins = [0, 25, 50, 75, 100])
plt.show()
```

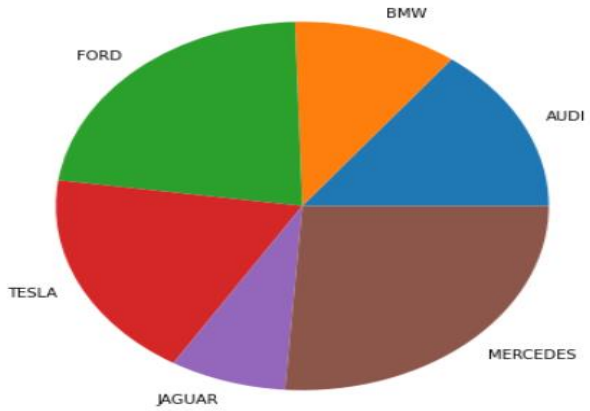
OUTPUT:



5 b) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib.

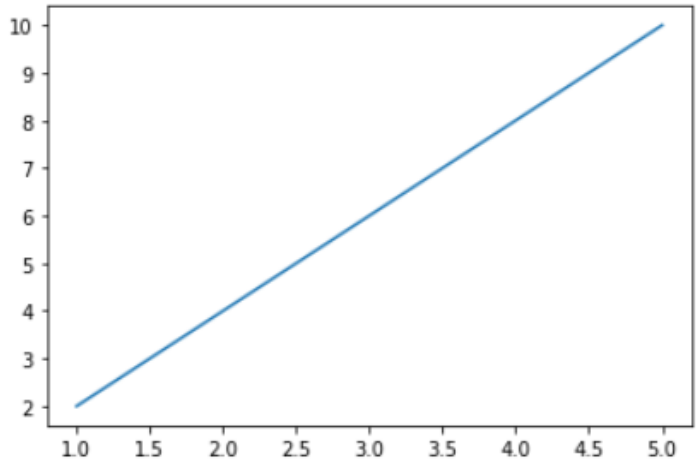
```
from matplotlib import pyplot as plt
import numpy as np
cars = ['AUDI', 'BMW', 'FORD','TESLA', 'JAGUAR', 'MERCEDES']
data = [23, 17, 35, 29, 12, 41]
fig = plt.figure(figsize =(10, 7))
plt.pie(data, labels = cars)
plt.show()
```

OUTPUT:



6 a) Write a Python program to illustrate Linear Plotting using Matplotlib.

```
import matplotlib.pyplot as plt

X = [1,2,3,4,5] # X-axis points

Y = [2,4,6,8,10] # Y-axis points

plt.plot(X,Y) # Plotting the line plot

plt.show() #Displaying the plot
```

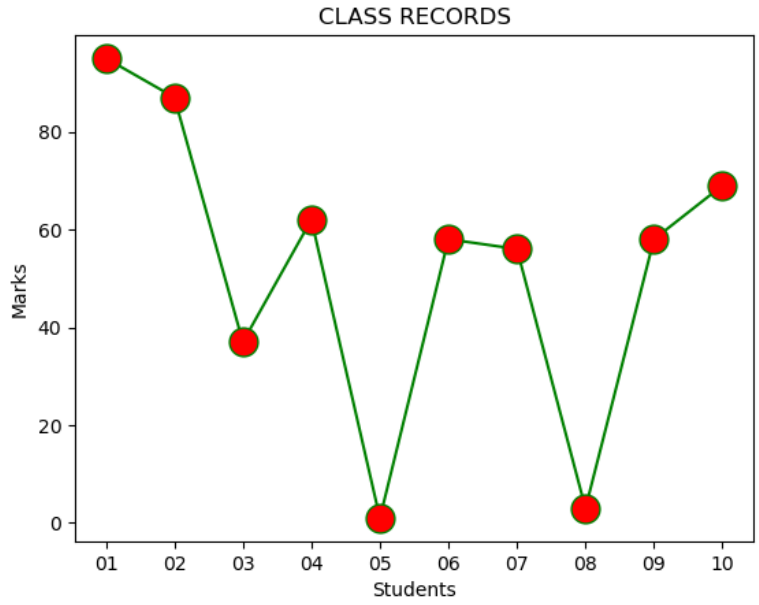


6 b) Write a Python program to illustrate liner plotting with line formatting using Matplotlib.

```
import matplotlib.pyplot as plt
import random as random
studentsusn = ["01","02","03","04","05","06","07","08","09","10"]
marks=[]
for i in range(0,len(studentsusn)):
marks.append(random.randint(0, 101))
```

```
plt.xlabel("Students")
plt.ylabel("Marks")
plt.title("CLASS RECORDS")
plt.plot(studentsusn,marks, color = 'green', linestyle = 'solid', marker = 'o', markerfacecolor =
'red', markersize = 15)
```

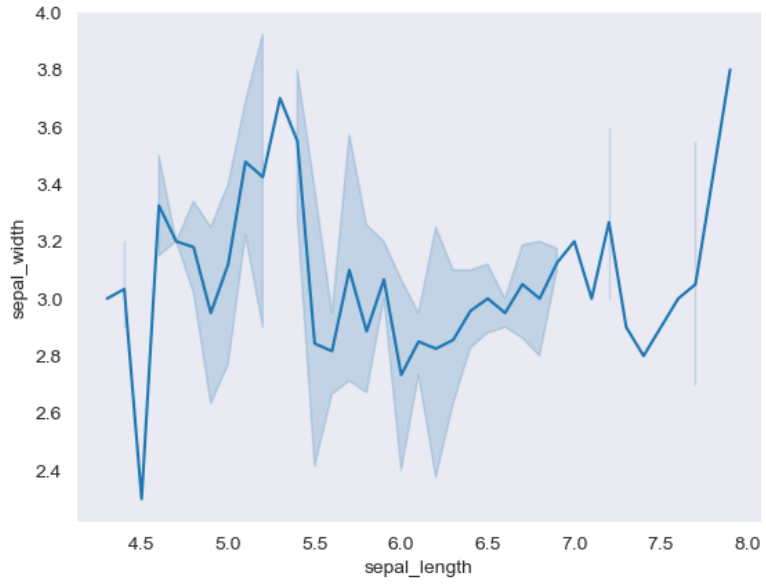
OUTPUT



7 a) Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions.

```
import seaborn as sns

import matplotlib.pyplot as plt

data = sns.load_dataset("iris")

sns.lineplot(x="sepal_length", y="sepal_width", data=data)

sns.set_style("darkgrid")

plt.show()
```

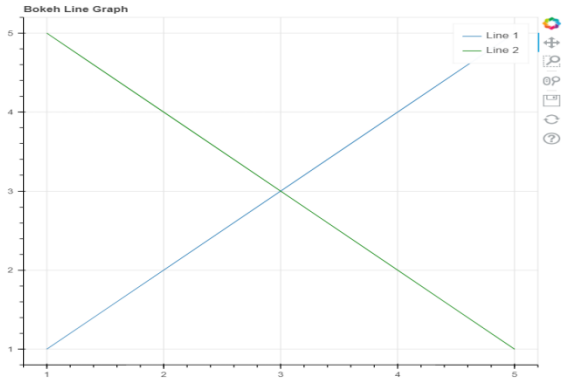
OUTPUT



8 a) Write a Python program to explain working with bokeh line graph using Annotations and Legends.

```
from bokeh.plotting import figure, output_file, show

graph = figure(title="Bokeh Line Graph")

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

y = [5, 4, 3, 2, 1]

graph.line(x, x, legend_label="Line 1")

graph.line(y, x, legend_label="Line 2", line_color="green")

show(graph)
```

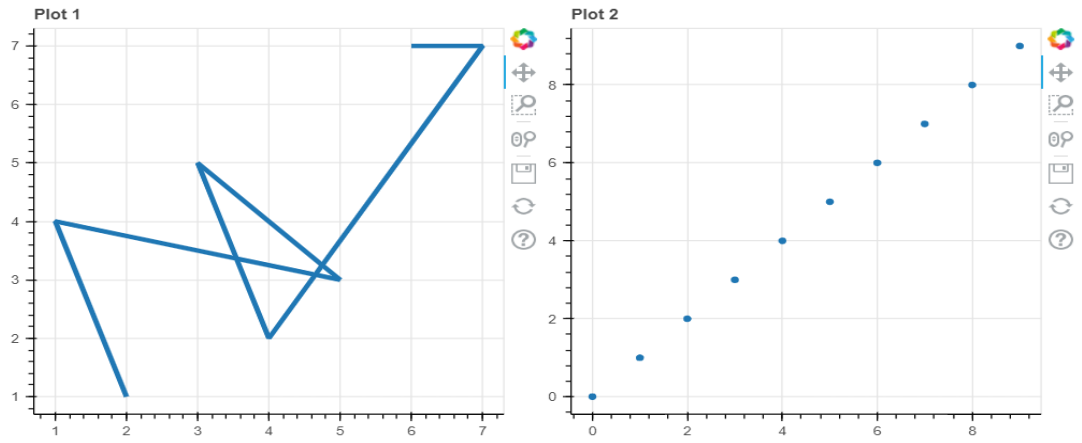
OUTPUT:



8. b) Write a Python program for plotting different types of plots using Bokeh.

```
from bokeh.io import output_file as OF

from bokeh.io import show

from bokeh.layouts import row

from bokeh.plotting import figure as figs

fig1 = figs(plot_width = 400,plot_height = 400, title = "Plot 1")

fig1.line([2, 1, 5, 3, 4, 7, 6],[1, 4, 3, 5, 2, 7, 7],line_width = 4)

x = y = list(range(10))

fig2 = figs(plot_width = 400,plot_height = 400, title = "Plot 2")

fig2.circle(x, y, size = 5)

show(row(fig1, fig2))
```

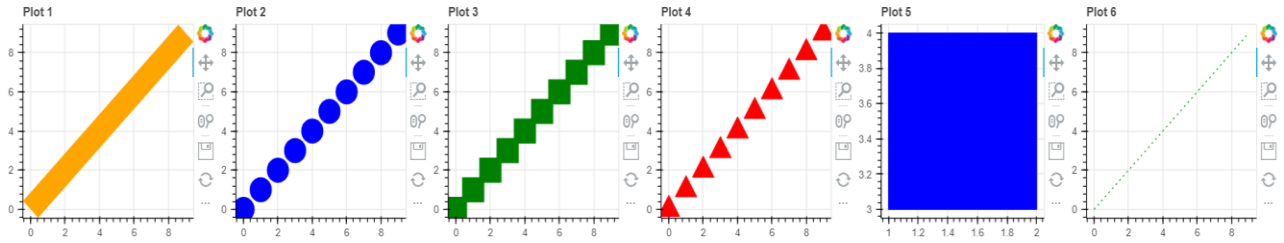
OUTPUT:



OR

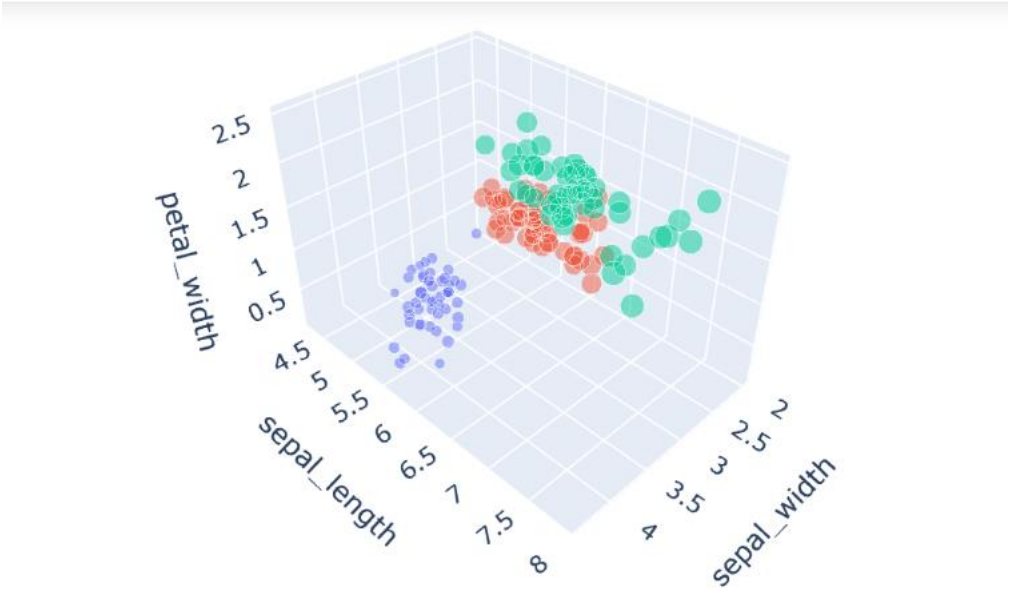
```
from bokeh.io import output_file as OF

from bokeh.io import show

from bokeh.layouts import row

from bokeh.plotting import figure as figs

x = y = list(range(10))

xs = [[[[1, 1, 2, 2]]]]

ys = [[[[4, 3, 3, 4]]]]

fig1 = figs(title = "Plot 1", plot_width = 250, plot_height = 250)

fig1.line(x, y, line_width = 25, color = "Orange")

fig2 = figs(title = "Plot 2", plot_width = 250, plot_height = 250)

fig2.circle(x, y, size = 25, color = "Blue")

fig3 = figs(title = "Plot 3", plot_width = 250,  plot_height = 250)

fig3.square(x, y, size = 25, color = "Green")

fig4 = figs(title = "Plot 4", plot_width = 250,plot_height = 250)

fig4.triangle(x, y, size = 25, color = "RED")

fig5 = figs(title = "Plot 5", plot_width = 250, plot_height = 250)

fig5.multi_polygons(xs, ys, color = "Blue")

fig6 = figs(title = "Plot 6", plot_width = 250,  plot_height = 250)

fig6.line(x, y, line_dash = "dotted", color = "Green")

show(row(fig1, fig2,  fig3, fig4, fig5, fig6))
```

output:



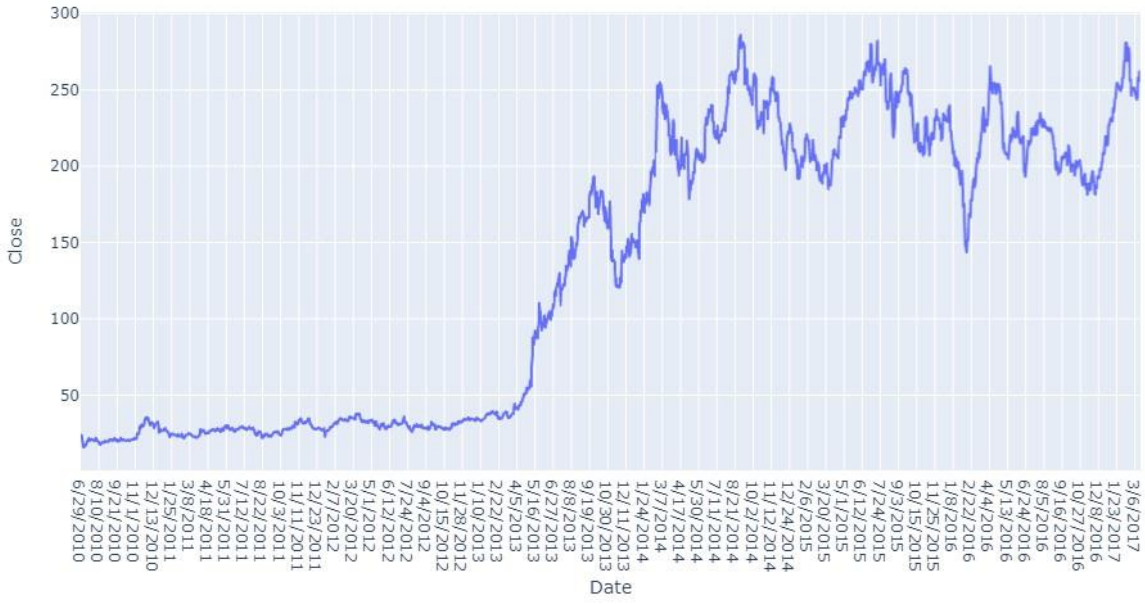
9. a) Write a Python program to draw 3D Plots using Plotly Libraries.

```
import plotly.express as px

df = px.data.iris()

 fig = px.scatter_3d(df, x = 'sepal_width',

                          y = 'sepal_length',

                           z = 'petal_width',

                           color = 'species',

                            size='petal_length',

size_max = 20,

                           opacity = 0.5)

fig.show()
```



10. a) Write a Python program to draw Time Series using Plotly Libraries.

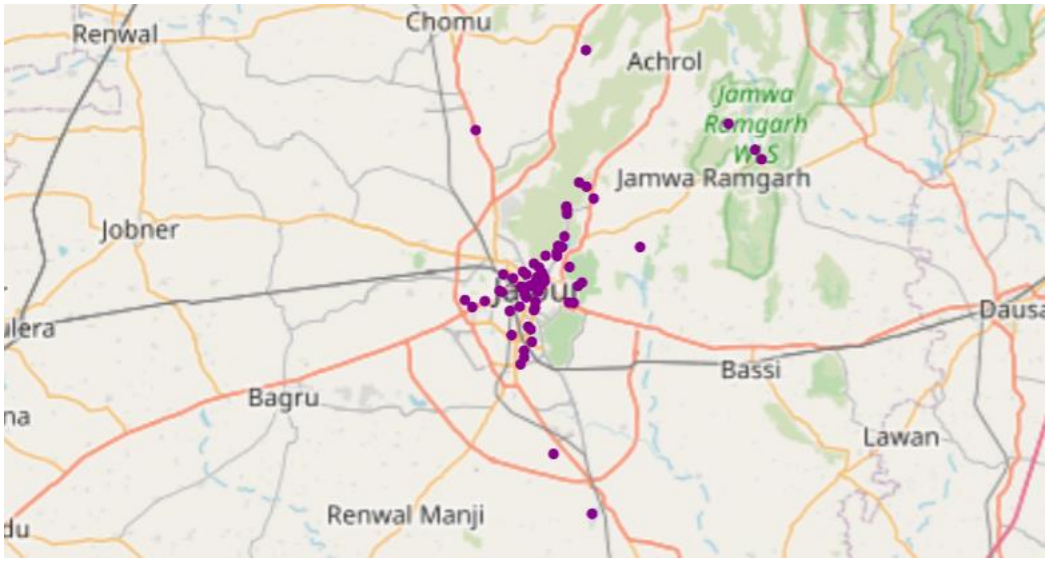
```
import pandas as pd

import plotly.express as px

tsla = pd.read_csv('C:/Users/Admin/OneDrive/Desktop/Tesla.csv')

tsla.head()

fig = px.line(tsla, x='Date', y="Close")

fig.show()

fig = px.line(tsla, x='Date', y="Close")

fig.show()
```



10 b) Write a Python program for creating Maps using Plotly Libraries.

```
import plotly.graph_objects as go

import plotly.express as px

url = 'C:/Users/Admin/Downloads/Places.csv'

data = pd.read_csv(url, header='infer')

data.head()
```

```
data = data.rename(columns = {'POIs':'Place'})

data.head()

fig = px.scatter_mapbox(data, lat="LATITUDE", lon="LONGITUDE", hover_name="Place",
hover_data=["Place"], color_discrete_sequence=["darkmagenta"], zoom=5.5, height=300)

fig.update_layout(mapbox_style="open-street-map")

fig.update_layout(margin={"r":0,"t":0,"l":0,"b":0})

fig.show()
```



Additional Programs for data visualization:

1. WAP to create a scatter plot in Matplotlib.
2. WAP to create a line plot in Matplotlib.
3. You have been provided with monthly sales data for a retail store. Visualize this data using a bar chart. Include labels for the x-axis, y-axis, and title.
4. A teacher wants to understand the distribution of exam scores for a class of students. Create a histogram to visualize the distribution. Adjust the number of bins to make the histogram informative.
5. A financial consultant wants to visualize a client's monthly expenses. Create a pie chart to represent the expense breakdown. Include percentage labels and a title for the chart.

6. You have recorded temperature data over a week. Visualize temperature trends using a line plot. Customize the line style, color, and markers.
7. Analyze the relationship between study hours and exam scores for a group of students. Create a scatter plot to visualize the data. Include labels for the x-axis, y-axis, and title.
8. A family wants to visualize the allocation of their monthly budget among different categories. Create a stacked bar chart to represent the budget allocation. Customize the colors and labels.
9. A web analyst wants to visualize the distribution of website traffic sources. Create a pie chart and adjust the start angle. Include a title for the chart.
10. Compare the sales trends of two products over several months. Create a line plot with multiple lines. Customize markers and include a legend.
11. Analyze the visualizations to draw conclusions and insights about the students' performance.

    i)Which students scored the highest and lowest on the test?

    ii) Is there a correlation between study hours and test scores?

    iii) What is the typical distribution of study hours?

    iv) How do attendance percentages compare among students?

    v) How have test scores changed over student IDs?

Sample Programs

1. **Python program to print "Hello Python"**

```
print ('Hello Python')
```

2. **Python program to do arithmetical operations**

```
# Store input numbers:
num1 = input('Enter first number: ')
num2 = input('Enter second number: ')

# Add two numbers
sum = float(num1) + float(num2)
# Subtract two numbers
min = float(num1) - float(num2)
# Multiply two numbers
mul = float(num1) * float(num2)
#Divide two numbers
div = float(num1) / float(num2)
# Display the sum
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))

# Display the subtraction
print('The subtraction of {0} and {1} is {2}'.format(num1, num2, min))
```

```
# Display the multiplication
print('The multiplication of {0} and {1} is {2}'.format(num1, num2, mul))
# Display the division
print('The division of {0} and {1} is {2}'.format(num1, num2, div))
```

3. **Python program to find the area of a triangle**

```
a = 5
b = 6
c = 7
# Uncomment below to take inputs from the user
# a = float(input('Enter first side: '))
# b = float(input('Enter second side: '))
# c = float(input('Enter third side: '))
# calculate the semi-perimeter
s = (a + b + c) / 2
# calculate the area
area = (s*(s-a)*(s-b)*(s-c)) ** 0.5
print('The area of the triangle is %0.2f' %area)
```

4. Python program to swap two variables

```
#swapping of 2 numbers
p=int(input("enter the value of a"))
q=int(input("enter the value of b"))
print('Before Swapping the value of p=',p,'and q=',q)
temp=p
p=q
q=temp
print('After swapping the value of p=',p,'and q=',q)
```

5. **Python program to find the sum and average of natural numbers up to n where n is provided by user.**

```
n=int(input("Enter upto which number you want sum and average"))
sum=0
for i in range(0,n+1):
    sum=sum+i
avg=sum/n
print("Result of sum is",sum)
print("Result of Average",avg)
```

6. WAP to find Factorial of a number using for loop

```
fact=1
n=int(input("enter the value of n to find factorial of a given number"))
for i in range(1,n+1) :
    fact=fact*i
print(fact)
```

7. **WAP to find Factorial of a number using while loop**

```
fact=1
i=1
n=int(input("enter the value of n to find factorial of a given number"))
while i<=n :
    fact=fact*i
i=i+1
print(fact)
```

8. WAP to find fibonacci series using Iterative:

```
n = int(input("How many terms? "))
n1, n2 = 0, 1
count = 0
if n <= 0:
print("Please enter a positive integer")
elif n == 1:
print("Fibonacci sequence upto",n,":")
   print(n1)
else:
print("Fibonacci sequence:")
while count < n:
   print(n1)
   next = n1 + n2
   n1 = n2
   n2 = next
   count += 1
```

9. **WAP to find fibonacci series using recursion:**

```
def fib(n):
   if n <= 1:
      return n
   else:
      return(fib(n-1) + fib(n-2))

nterms = int(input("How many terms? "))
if nterms<= 0:
print("Plese enter a positive integer")
else:
print("Fibonacci sequence:")
for i in range(nterms):
   print(fib(i))
```

10 **. WAP to find largest among three numbers, input by user**

```
n1=int(input("enter first number"))
n2=int(input("enter sec number"))
n3=int(input("enter third number"))

if n1>n2 and n1>n3 :
print("n1 is larger")
elif n2>n3 and n2>n1 :
print("n2 is larger")
else :
print("n3 is larger")
```

11. WAP to print first ten programs using for loop .

```
for i in range(1,11) :
   print(i)
```

12. WAP to print first ten programs using while loop.

```
I=1
while i<=10:
   print(i)
i=i+1
```

13. WAP to check whether a person is eligible for voting.

```
age=input()
type(age)
x=int(age)
if x>18 :
   print("eligible")
else :
print("not ")
```

14. **WAP to print grades obtained by the students and print the appropriate message.**

```
marks=input()
type(marks)
x=int(marks)
if x>=90 and x<100:
   print('distinction')
elif x>=80 and x<=90:
   print("first")
else :
   print("fail")
```

15.WAP to find factorial of given number using for loop.

```
 fact=1
for i in range(1,6) :
   fact*=i
print(fact)
```

16.**WAP to find factorial of given number using while loop.**

```
 fact=1
i=1
n=int(input())
while i<=n:
   fact=fact*i
i=i+1
print(fact)
```

17. **WAP to find factorial of given number using functions.**

```
def fact(n) :
   fact=1
i=1
   while i<=n :
      fact*=i
i=i+1
   return fact
   #print(fact)
n=int(input("enter the number to find factoral of a given number"))
print(fact(n))
```

18.**WAP to find gcd of 2 numbers.**

```
defgcd(a,b):
   if b==0:
      return a
else :
      return gcd(b,a%b)
n1=int(input("enter the first number"))
n2=int(input("enter the second number"))
```

```
print(gcd(n1,n2))
```

19. Python program to generate a random number
20. Python program to convert kilometers to miles.
21. Python program to convert Celsius to Fahrenheit
22. Python program to display calendar
23. Python Program to Check if a Number is Positive, Negative or Zero
24. Python Program to Check if a Number is Odd or Even
25. Python Program to Check Leap Year
26. Python Program to Check Prime Number
27. Python Program to Print all Prime Numbers in an Interval
28. Python Program to Find the Factorial of a Number
29. Python Program to Display the multiplication Table
30. Python Program to Print the Fibonacci sequence
31. Python Program to Check Armstrong Number
32. Python Program to Find Armstrong Number in an Interval

Python Additional Programs

1. Python program to print "Hello Python"
2. Python program to do arithmetical operations
3. Python program to find the area of a triangle
4. Python program to solve quadratic equation
5. Python program to swap two variables
6. Python program to generate a random number
7. Python program to convert kilometers to miles
8. Python program to convert Celsius to Fahrenheit
9. Python program to display calendar
10. Python Program to Check if a Number is Positive, Negative or Zero
11. Python Program to Check if a Number is Odd or Even
12. Python Program to Check Leap Year
13. Python Program to Check Prime Number
14. Python Program to Print all Prime Numbers in an Interval
15. Python Program to Find the Factorial of a Number
16. Python Program to Display the multiplication Table
17. Python Program to Print the Fibonacci sequence
18. Python Program to Check Armstrong Number
19. Python Program to Find Armstrong Number in an Interval
20. Python Program to Find the Sum of Natural Numbers
21. Python Function Programs
22. Python Program to Find LCM and HCF.
23. Python Program To Find ASCII value of a character
24. Python Program to Make a Simple Calculator
25. Python Program to Display Calendar
26. Python Program to Display Fibonacci Sequence Using Recursion
27. Python Program to Find Factorial of Number Using Recursion

28. Python program to check if the given number is Happy Number
29. Python program to print all happy numbers between 1 and 100
30. Python program to copy all elements of one array into another array
31. Python program to find the frequency of each element in the array
32. Python program to left rotate the elements of an array
33. Python program to print the duplicate elements of an array
34. Python program to print the elements of an array
35. Python program to print the elements of an array in reverse order
36. Python program to print the elements of an array present on even position
37. Python program to print the elements of an array present on odd position
38. Python program to print the largest element in an array
39. Python program to print the smallest element in an array
40. Python program to print the number of elements present in an array
41. Python program to print the sum of all elements in an array
42. Python program to right rotate the elements of an array
43. Python program to sort the elements of an array in ascending order
44. Python program to sort the elements of an array in descending order

Viva Questions:

1. What it the syntax of print function?
2. What is the usage of input function?
3. Define a variable.
4. What is type conversion?
5. Mention the data types in Python
6. What are the attributes of the complex datatype?
7. Mention a few escape sequences.
8. Define an expression
9. What is the usage of ** operator in Python?
10. Give the syntax of if else statement.
11. What is data visualization, and why is it important?
12. .What is Matplotlib?
13. What is Seaborn?
14. How do you install Matplotlib and Seaborn?
15. .How do you create a scatter plot in Matplotlib?
16. How do you create a line plot in Matplotlib?
17. What are some common tools used for data visualization?
18. What is the difference between a histogram and a bar chart?
19. What kinds of information are often shown by box plots? Why?
20. What is a Boxplot?
21. Why is data visualization necessary?
22. How do you decide whether to use a bar chart or a line chart?
23. What are some best practices for designing effective visualizations?
24. What are the 5 steps in data visualization?
25. What are the two main uses for data visualization?

26. .Is python a data visualization tool?
27. Give the syntax of for statement.
28. How is range function used in for?
29. Give the syntax of while statement.
30. What are multi way if statements?
31. How is random numbers generated?
32. Define a function.
33. Give the syntax of function.
34. What are the types of arguments in function.?
35. What is a recursive function?
36. What are anonymous functions?
37. What are default arguments?
38. What are variable length arguments?
39. What are keyword arguments?
40. Mention the use of map().
41. Mention the use of filter().
42. Mention the use of reduce().
43. Define a string.
44. How is string slicing done?
45. What is the usage of repetition operator?
46. How is string concatenation done using + operator>
47. Mention some string methods
48. How is length of a string found?
49. How is a string converted to its upper case?
50. `Differentiate isalpha() and isdigit().
51. What is the use of split()?
52. Define a file.
53. Give the syntax for opening a file.
54. Give the syntax for closing a file.
55. How is reading of file done?
56. How is writing of file done?
57. What is a list?
58. Lists are mutable-Justify.
59. How is a list created?
60. How can a list be sorted?
61. How are elements appended to the list?
62. How is insert() used in list?
63. What is the usage of pop() in list?
64. Define a tuple.
65. Are tuples mutable or immutable?
66. Mention the use of return statement.
67. What is a Boolean function?
68. How is main function defined?
69. What is a dictionary?

70. How are tuples created?
71. How is a dictionary created?
72. How to print the keys of a dictionary?
73. How to print the values of a dictionary?
74. How is del statement used?
75. Can tuple elements be deleted?
76. What is Python interpreter?
77. Why is Python called an interpreted language?
78. Mention some features of Python
79. What is Python IDLE?
80. Mention some rules for naming an identifier in Python.
81. Give points about Python Numbers.
82. What is bool datatype?
83. Give examples of mathematical functions.
84. What is string formatting operator?
85. Mention about membership operators in Python.
86. How is expression evaluated in Python?
87. What are the loop control statements in Python?
88. What is the use of break statement?
89. What is the use of continue statement?
90. What is the use of pass statement?
91. What is assert statement?
92. How is exception handled in python?
    77. What are required arguments ?
93. Differentiate pass by value and pass by reference.
94. Mention few advantages of function.
95. What is swapcase() method?
96. What is a local variable?
97. What are global variables?
98. What are Python decorators?
99. Are strings mutable or immutable?
100. What is join()?
101. What is replace() method?
102. What is an exception?How is exception handled in python?