

K.S. SCHOOL OF ENGINEERIN AND MANAGEMENT, BENGALURU

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

DATA VISUALIZATION WITH PYTHON

INTEGRATED PROFESSIONAL CORE COURSE

BCS358D



Prepared by :-

Mrs. Jayashubha J
Associate Professor
Dept of CSE, KSSEM

Mrs. Bindu K. P
Assistant Professor
Dept of CSE, KSSEM

LAB MANUAL

ACADEMIC YEAR 2023 - 2024

Data Visualization with Python		Semester	III
Course Code	BCS358D	CIE Marks	50
Teaching Hours/Week (L:T:P: S)	0: 0: 2: 0	SEE Marks	50
Credits	01	Exam Hours	100
Examination type (SEE)	Practical		

Course objectives:

- CLO 1. Demonstrate the use of IDLE or PyCharm IDE to create Python Applications
- CLO 2. Using Python programming language to develop programs for solving real-world problems
- CLO 3. Implementation of Matplotlib for drawing different Plots
- CLO 4. Demonstrate working with Seaborn, Bokeh.
- CLO 5. Working with Plotly for 3D, Time Series and Maps.

	Experiments		
Sl. No.	PART A – List of problems for which student should develop program and execute in the Laboratory		
1	<p>a) Write a python program to find the best of two test average marks out of three test's marks accepted from the user.</p> <p>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.</p> <p>Datatypes: https://www.youtube.com/watch?v=gCCVsvgR2KU Operators: https://www.youtube.com/watch?v=v5MR5JnKcZI Flow Control: https://www.youtube.com/watch?v=PqFKRqpHrjw For loop: https://www.youtube.com/watch?v=0ZvaDa8eT5s While loop: https://www.youtube.com/watch?v=HZARIImviDxg Exceptions: https://www.youtube.com/watch?v=6SPDvPK38tw</p>		
2	<p>a) Defined as a function F as $F_n = F_{n-1} + F_{n-2}$. 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.</p> <p>b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions.</p> <p>Functions: https://www.youtube.com/watch?v=BVfCWuca9nw Arguments: https://www.youtube.com/watch?v=ijXMGpoMkhQ Return value: https://www.youtube.com/watch?v=nuNXiEDnM44</p>		
3	<p>a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters.</p> <p>b) Write a Python program to find the string similarity between two given strings</p> <table style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%; vertical-align: top;"> Sample Output: Original string: Python Exercises Python Exercises Similarity between two said strings: </td><td style="width: 50%; vertical-align: top;"> Sample Output: Original string: Python Exercises Python Exercise Similarity between two said strings: 1.0 0.967741935483871 </td></tr> </table> <p>Strings: https://www.youtube.com/watch?v=lSItwlnF0eU String functions: https://www.youtube.com/watch?v=9a3CxJyTq00</p>	Sample Output: Original string: Python Exercises Python Exercises Similarity between two said strings:	Sample Output: Original string: Python Exercises Python Exercise Similarity between two said strings: 1.0 0.967741935483871
Sample Output: Original string: Python Exercises Python Exercises Similarity between two said strings:	Sample Output: Original string: Python Exercises Python Exercise Similarity between two said strings: 1.0 0.967741935483871		

4	<p>a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.</p> <p>b) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib.</p> <p>https://www.youtube.com/watch?v=RRHQ6Fs1b8w&list=PLjVLYmrImjGcC0B_FlP3bkJ-JIPkV5GuZR&index=3</p> <p>https://www.youtube.com/watch?v=7ABCuhWO9II&list=PLjVLYmrImjGcC0B_FlP3bkJ-JIPkV5GuZR&index=4</p>
5	<p>a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib.</p> <p>b) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib.</p> <p>https://www.youtube.com/watch?v=Qk7caotaQUQ&list=PLjVLYmrImjGcC0B_FlP3bkJ-JIPkV5GuZR&index=6</p> <p>https://www.youtube.com/watch?v=PSji21jUNO0&list=PLjVLYmrImjGcC0B_FlP3bkJ-JIPkV5GuZR&index=7</p>
6	<p>a) Write a Python program to illustrate Linear Plotting using Matplotlib.</p> <p>b) Write a Python program to illustrate liner plotting with line formatting using Matplotlib.</p> <p>https://www.youtube.com/watch?v=UO98IJQ3QGI&list=PL-osiE80TeTvipQomVEeZ1HRrcEvtZB</p>
7	<p>Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions.</p> <p>https://www.youtube.com/watch?v=6GUZXDef2U0</p>
8	<p>Write a Python program to explain working with bokeh line graph using Annotations and Legends.</p> <p>a) Write a Python program for plotting different types of plots using Bokeh.</p> <p>https://www.youtube.com/watch?v=HDvxYoRadcA</p>
9	<p>Write a Python program to draw 3D Plots using Plotly Libraries.</p> <p>https://www.youtube.com/watch?v=cCck7hCanpw&list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&index=4</p>

10	<p>a) Write a Python program to draw Time Series using Plotly Libraries.</p> <p>b) Write a Python program for creating Maps using Plotly Libraries.</p> <p>https://www.youtube.com/watch?v=xnJ2TNrGYik&list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&index=5</p> <p>https://www.youtube.com/watch?v=D35m2CdMhVs&list=PLE50-dh6JzC4onX-qkv9H3HtPbBVA8M94&index=6</p>
Python (Full Course): https://www.youtube.com/watch?v=_uQrJ0TkZlc	
Pedagogy	For the above experiments the following pedagogy can be considered. Problem based learning, Active learning, MOOC, Chalk &Talk
<p>Course outcomes (Course Skill Set):</p> <p>At the end of the course the student will be able to:</p> <p>CO 1. Demonstrate the use of IDLE or PyCharm IDE to create Python Applications</p> <p>CO 2. Use Python programming constructs to develop programs for solving real-world problems</p> <p>CO 3. Use Matplotlib for drawing different Plots</p> <p>CO 4. Demonstrate working with Seaborn, Bokeh for visualization.</p> <p>CO 5. Use Plotly for drawing Time Series and Maps.</p>	

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50) and for the SEE minimum passing mark is 35% of the maximum marks (18 out of 50 marks). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/course if the student secures a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together

Continuous Internal Evaluation (CIE):

CIE marks for the practical course are **50 Marks**.

The split-up of CIE marks for record/ journal and test are in the ratio **60:40**.

- Each experiment is to be evaluated for conduction with an observation sheet and record write-up. Rubrics for the evaluation of the journal/write-up for hardware/software experiments are designed by the faculty who is handling the laboratory session and are made known to students at the beginning of the practical session.
- Record should contain all the specified experiments in the syllabus and each experiment write-up will be evaluated for 10 marks.
- Total marks scored by the students are scaled down to **30 marks** (60% of maximum marks).
- Weightage to be given for neatness and submission of record/write-up on time.
- Department shall conduct a test of 100 marks after the completion of all the experiments listed in the syllabus.
- In a test, test write-up, conduction of experiment, acceptable result, and procedural knowledge will carry a weightage of 60% and the rest 40% for viva-voce.
- The suitable rubrics can be designed to evaluate each student's performance and learning ability.
- The marks scored shall be scaled down to **20 marks** (40% of the maximum marks).

The Sum of scaled-down marks scored in the report write-up/journal and marks of a test is the total CIE marks scored by the student.

Semester End Evaluation (SEE):

- SEE marks for the practical course are 50 Marks.
- SEE shall be conducted jointly by the two examiners of the same institute, examiners are appointed by the Head of the Institute.
- The examination schedule and names of examiners are informed to the university before the conduction of the examination. These practical examinations are to be conducted between the schedule mentioned in the academic calendar of the University.
- All laboratory experiments are to be included for practical examination.
- (Rubrics) Breakup of marks and the instructions printed on the cover page of the answer script to be strictly adhered to by the examiners. **OR** based on the course requirement evaluation rubrics shall be decided jointly by examiners.
- Students can pick one question (experiment) from the questions lot prepared by the examiners jointly.
- Evaluation of test write-up/ conduction procedure and result/viva will be conducted jointly by examiners.
- General rubrics suggested for SEE are mentioned here, writeup-20%, Conduction procedure and result in -60%, Viva-voce 20% of maximum marks. SEE for practical shall be evaluated for 100 marks and scored marks shall be scaled down to 50 marks (however, based on course type, rubrics shall be decided by the examiners)
- Change of experiment is allowed only once and 15% of Marks allotted to the procedure part are to be made zero.

The minimum duration of SEE is 02 hours

- Weightage of marks for PART A is 80% and for PART B is 20%. General rubrics suggested to be followed for part A and part B.
- Change of experiment is allowed only once and Marks allotted to the procedure part to be made zero (Not allowed for Part B).
- The duration of SEE is 03 hours

Rubrics suggested in Annexure-II of Regulation book

Textbooks:

1. Al Sweigart, “Automate the Boring Stuff with Python”, 1st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at <https://automatetheboringstuff.com/>)
2. Reema Thareja “Python Programming Using Problem Solving Approach” Oxford University Press.
3. Allen B. Downey, “Think Python: How to Think Like a Computer Scientist”, 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at <http://greenteapress.com/thinkpython2/thinkpython2.pdf>)
4. Jake VanderPlas “Python Data Science Handbook” 1st Edition, O’REILLY.



K.S. SCHOOL OF ENGINEERING AND MANAGEMENT
BANGALORE - 560109
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CO-PO Mapping

Course: Data Visualization with Python		Course Code: BCS358D	
No of Hours			
Lab Hours	Practical/Field Work/Allied Activities	Total hours/Week	Total Number of Lab Contact Hours
2	0	2	28
Marks			
CIE	SEE	Total	Credits
50	50	100	01

Aim/Objectives of the Course

This laboratory course enables students to get practical experience in design, develop, implement, analyze and evaluation/testing of

1. Demonstrate the use Python programming constructs to develop programs for solving real-world problems.
2. Identify the different functions of Matplotlib for drawing different Plots.
3. Demonstrate working with Seaborn, Bokeh for visualization.
4. Discover the need of Plotly for drawing Time Series and Maps.

Course Learning Outcomes

After completing the course, the students will be able to

CO1	Demonstrate the use of IDLE or PyCharm IDE to create Python Applications.	Applying (K3)
CO2	Use Python programming constructs to develop programs for solving real-world problems	Applying (K3)
CO3	Implementation of Matplotlib for drawing different Plots.	Applying (K3)
CO4	Demonstrate working with Seaborn, Bokeh for data visualization	Applying (K3)
CO5	Apply Plotly for drawing Time Series and Maps	Applying (K3)

Syllabus Content	
Program – 1 <ul style="list-style-type: none"> a) Write a python program to find the best of two test average marks out of three test's marks accepted from the user. 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. 	CO1, CO2 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-2 PSO2-2
Program – 2 <ul style="list-style-type: none"> a) Defined as a function F as $F_n = F_{n-1} + F_{n-2}$. 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. b) Develop a python program to convert binary to decimal, octal to hexadecimal using functions. 	CO1, CO2 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-2 PSO2-2
Program – 3 <ul style="list-style-type: none"> a) Write a Python program that accepts a sentence and find the number of words, digits, uppercase letters and lowercase letters. b) Write a Python program to find the string similarity between two given strings. 	CO1,CO2 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-2 PSO2-2

Program - 4 <ul style="list-style-type: none"> a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib. b) Write a Python program to Demonstrate how to Draw a Scatter Plot using Matplotlib. 	CO3 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3
Program - 5 <ul style="list-style-type: none"> a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib. b) Write a Python program to Demonstrate how to Draw a Pie Chart using Matplotlib. 	CO3 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3
Program - 6 <ul style="list-style-type: none"> a) Write a Python program to illustrate Linear Plotting using Matplotlib. b) Write a Python program to illustrate liner plotting with line formatting using Matplotlib. 	CO3 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3
Program - 7 <ul style="list-style-type: none"> a) Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions. 	CO4 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3

<p>Program - 8</p> <p>Write a Python program to explain working with bokeh line graph using Annotations and Legends.</p> <p>a) Write a Python program for plotting different types of plots using Bokeh.</p>	CO4 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3
<p>Program - 9</p> <p>Write a Python program to draw 3D Plots using Plotly Libraries.</p>	CO5 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3
<p>Program - 10</p> <p>a) Write a Python program to draw Time Series using Plotly Libraries.</p> <p>b) Write a Python program for creating Maps using Plotly Libraries.</p>	CO5 2 hrs PO1-3 PO2-3 PO3-2 PO4-1 PO5-3 PO8-2 PO9-1 PO12 -1 PSO1-3 PSO2-3

CO to PO Mapping

<p>PO1: Science and engineering Knowledge</p> <p>PO2: Problem Analysis</p> <p>PO3: Design & Development</p> <p>PO4: Investigations of Complex Problems</p> <p>PO5: Modern Tool Usage</p> <p>PO6: Engineer & Society</p>	<p>PO7: Environment and Sustainability</p> <p>PO8: Ethics</p> <p>PO9: Individual & Team Work</p> <p>PO10: Communication</p> <p>PO11: Project Management & Finance</p> <p>PO12: Lifelong Learning</p>
---	--

PSO1: Understand fundamental and advanced concepts in the core areas of Computer Science and Engineering to analyze, design and implement the solutions for the real-world problems.

PSO2: Utilize modern technological innovations efficiently in various applications to work towards the betterment of society and solve engineering problems.

CO	PO	PO ₁	PO2	PO3	PO4	PO5	PO6	PO7	PO ₈	PO ₉	PO ₁₀	PO11	PO12	PSO ₁	PSO ₂
BCS 304	K-level														
CO1	K3	3	3	2	1	3	-	-	2	1	-	-	1	2	2
CO2	K3	3	3	2	1	3	-	-	2	1	-	-	1	2	2
CO3	K3	3	3	2	1	3	-	-	2	1	-	-	1	3	3
CO4	K3	3	3	2	1	3	-	-	2	1	-	-	1	3	3
CO5	K3	3	3	2	1	3	-	-	2	1	-	-	1	3	3

INSTITUTION VISION & MISSION

VISION:

“To impart quality education in engineering and management to meet technological, business and societal needs through holistic education and research”

MISSION:

K.S. School of Engineering and Management shall,

- ❖ Establish state-of-art infrastructure to facilitate effective dissemination of technical and Managerial knowledge.
- ❖ Provide comprehensive educational experience through a combination of curricular and experiential learning, strengthened by industry-institute-interaction.
- ❖ Pursue socially relevant research and disseminate knowledge.
- ❖ Inculcate leadership skills and foster entrepreneurial spirit among students.

DEPARTMENT VISION & MISSION

VISION:

“To produce quality Computer Science professional, possessing excellent technical knowledge, skills, personality through education and research.”

MISSION:

Department of Computer Science and Engineering shall,

- ❖ Provide good infrastructure and facilitate learning to become competent engineers who meet global challenges.
- ❖ Encourages industry institute interaction to give an edge to the students.
- ❖ Facilitates experimental learning through interdisciplinary projects.
- ❖ Strengthen soft skill to address global challenges.

LAB PROGRAMS

EXPERIMENTS 1: -

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

```
test1= float (input ("Enter the first test marks"))
test2= float (input ("Enter the second test marks"))
test3= float (input ("Enter the third test marks"))
marks= []
marks. append(test1)
marks. append(test2)
marks. append(test3)
print(marks)
marks. sort (reverse=1)
print ("Marks after sorting ")
print(marks)
avg = ((marks [0] +marks [1]))/2
print ("The average marks of the student", avg)
```

OUTPUT: -

```
Enter the first test marks34
Enter the second test marks69
Enter the third test marks5
[34.0, 69.0, 5.0]
Marks after sorting
[69.0, 34.0, 5.0]
The average marks of the student 51.5
```

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.

```
value = input("enter the value\n")
if (value==value [ :: -1]):
    print ("the given number is a palindrome")
else:
    print ("the given number is not a palindrome")

for i in range (10):
```

```
if value. count(str(i))>0:  
    print(f'{str(i)} appears {value. count(str(i))} times')
```

OUTPUT 1: -

enter the value

1551

the given number is a palindrome

1 appears 2 times

5 appears 2 times

OUTPUT 2: -

enter the value

2356

the given number is not a palindrome

2 appears 1 times

3 appears 1 times

5 appears 1 times

6 appears 1 times

EXPERIMENTS 2: -

2.a) Defined as a function F as $F_n = F_{n-1} + F_{n-2}$. 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==0 or n==1:
        return 1
    else:
        return Fibonacci(n-1) + Fibonacci(n-2)

num = int (input ("Enter the value of N:"))
if num>0:
    fib = Fibonacci(num)
    print ("Fibonacci value at nth position is:", fib)
else:
    print ("Incorrect value of N")
```

OUTPUT 1: -

Enter the value of N:8
Function value at nth position is: 34

OUTPUT 2: -

Enter the value of N:0
Incorrect value of N

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

```
def binTodeci (binary):
    return int(binary,2)

def octTohex (octal):
    print("Decimal equivalent ",int(octal,8))
    return hex(int(octal,8))

# Convert binary to decimal

binaryNo = input ("Enter a binary number ")
decimalNo = binTodeci(binaryNo)
print ("Decimal Equivalent ", decimalNo)

# Convert octal to hexadecimal

octalNo = input ("Enter an octal number ")
hexadecimalNo = octTohex(octalNo)
print ("Hexadecimal Equivalent ", hexadecimalNo)
```

OUTPUT: -

Enter a binary number 1100

Decimal Equivalent 12

Enter an octal number 57

Decimal equivalent 47

Hexadecimal Equivalent 0x2f

EXPERIMENTS 3: -

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

```
str=input ("Enter the Sentence ")
w, d, l, u=0,0,0,0
words=str.split ()
w=len (words)
for i in str:
    if i.isdigit():
        d=d+1
    elif i.isupper():
        u=u+1
    elif i.islower():
        l=l+1
print ("Number of words", w)
print ("Number of Digits", d)
print ("Number of Uppercase Letters", u)
print ("Number of Lowercase Letters", l)
```

OUTPUT: -

Enter the Sentence k S School of Engg 123
 Number of words 6
 Number of Digits 3
 Number of Uppercase Letters 3
 Number of Lowercase Letters 11

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

```
import difflib

def string_similarity (str1, str2):
    result=difflib.SequenceMatcher(a=str1.lower(), b=str2.lower())
    return result.ratio()

str1 = input ("Enter the first string :")
str2 = input ("Enter the second string :")
print ("Original string:")
print(str1)
print(str2)
print ("Similarity between two said strings:")
print (string_similarity(str1, str2))
```

OUTPUT 1: -

Enter the frist string :Python Excercises
 Enter the frist string :Python Excercises

Original string:
Python Excercises
Python Excercises
Similarity between two said strings:
1.0

OUTPUT 2: -

Enter the frist string :Python Lab
Enter the frist string :Python
Original string:
Python Lab
Python
Similarity between two said strings:
0.75

OUTPUT 3: -

Enter the frist string :Python
Enter the frist string :Excercises
Original string:
Python
Excercises
Similarity between two said strings:
0.0

EXPERIMENTS 4: -**4.a) Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib.**

```

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

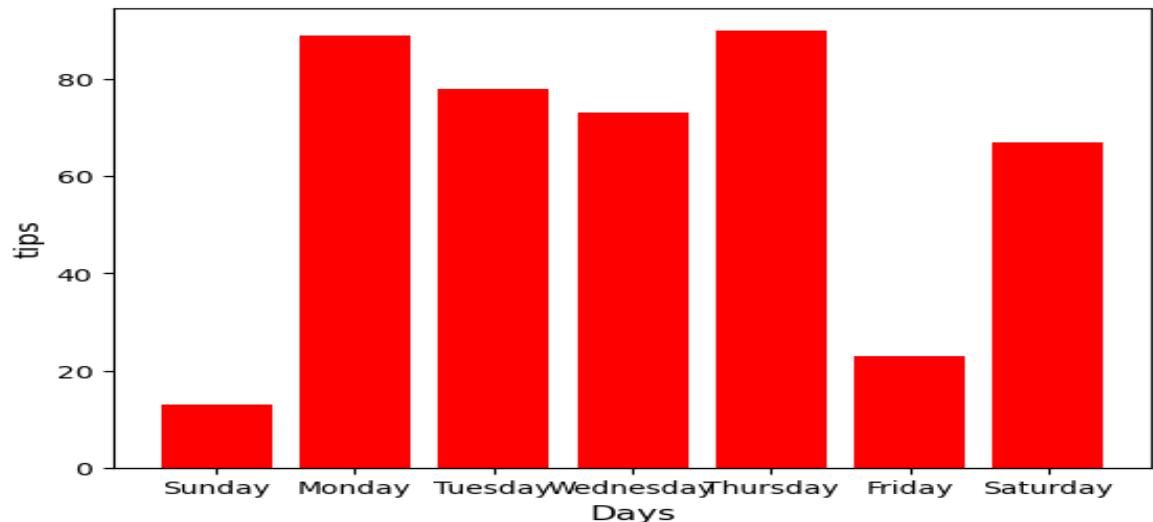
days = ['Sunday','Monday','Tuesday','Wednesday','Thursday','Friday','Saturday']
values = [13,89,78,73,90,23,67]

#Plot Bar Graph
plt.bar(days,values,color='red')

#Add labels and title
plt.xlabel("Days",fontsize=12)
plt.ylabel("tips",fontsize=12)

#Display the plot
plt.show()

```

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

```

import matplotlib.pyplot as plt
import pandas as pd
import numpy as np

country= ['Brazil','Russia','India','China','South Africa']
per_capita_income= [9600,11600,2300,11000,6500]

# Plot the scatter plot

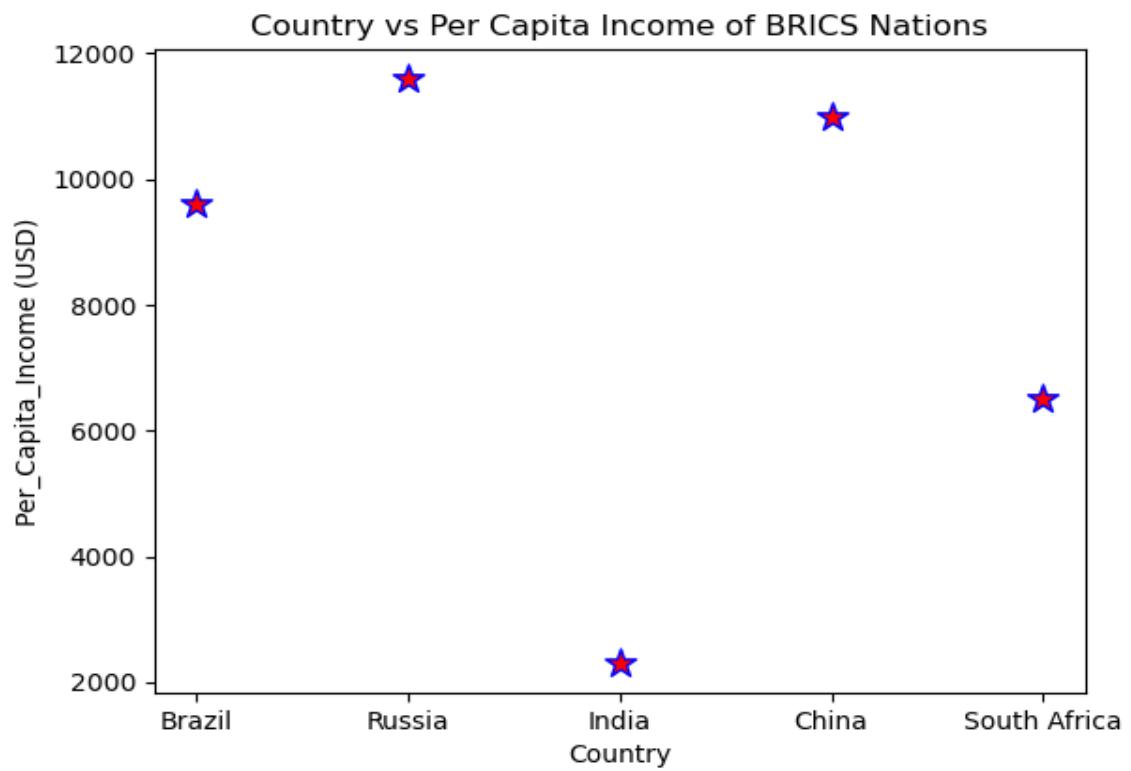
```

```
plt.scatter(country,per_capita_income,s=150,c='r',linewidth=1,marker='*',edgecolor='b')

#Add labels and title
plt.xlabel('Country')
plt.ylabel('Per_Capita_Income (USD)')
plt.title('Country vs Per Capita Income of BRICS Nations')

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

OUTPUT:-



EXPERIMENTS 5: -**5.a) Write a Python program to Demonstrate how to Draw a Histogram Plot using Matplotlib.**

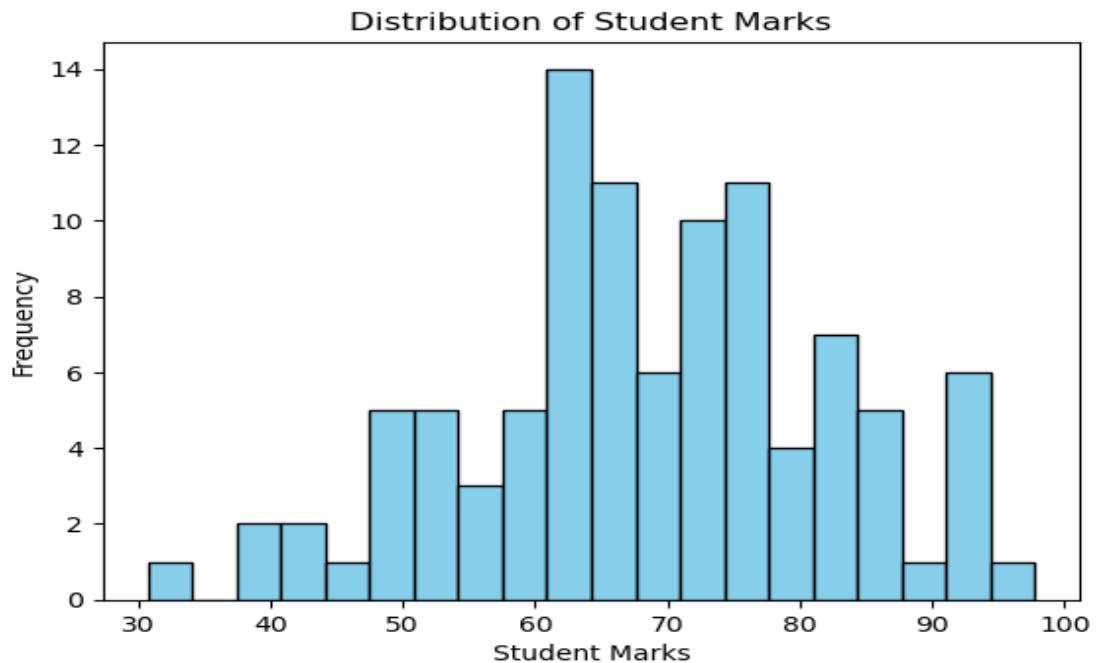
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import random

#Generate random students marks
np.random.seed(42)
student_marks=np.random.normal(loc=70,scale=15,size=100)

#Create Histogram
plt.hist(student_marks,bins=20,color='skyblue',edgecolor='black')

#Add labels and title
plt.xlabel('Student Marks')
plt.ylabel('Frequency')
plt.title('Distribution of Student Marks')

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

OUTPUT: -

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

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

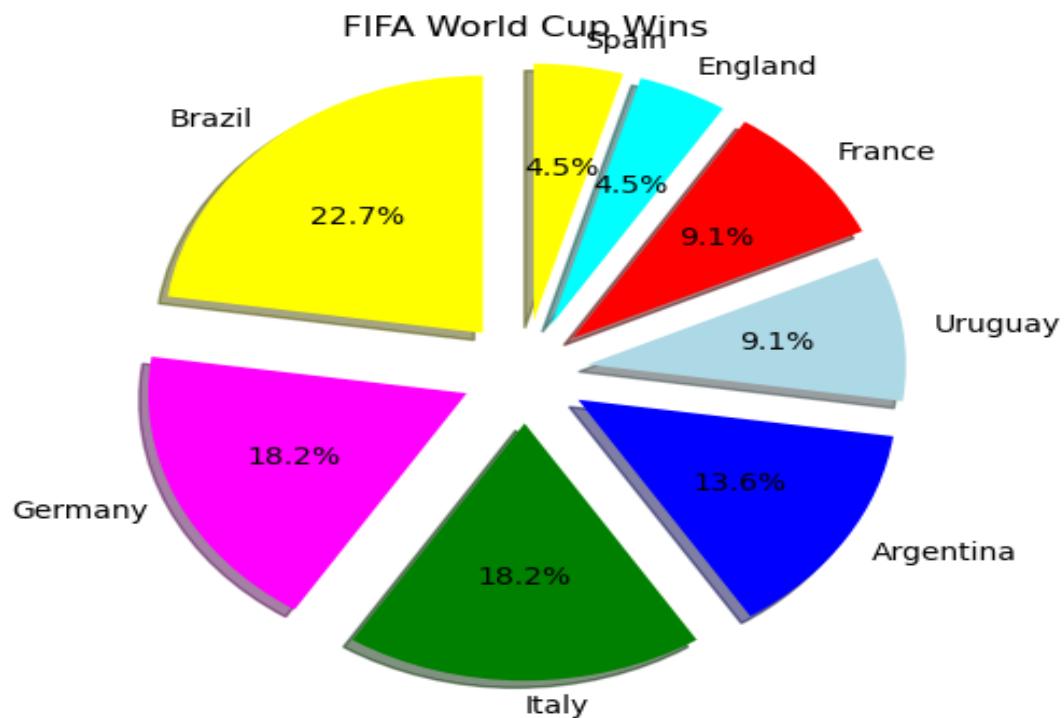
countries=['Brazil','Germany','Italy','Argentina','Uruguay','France','England','Spain']
wins=[5,4,4,3,2,2,1,1]

colors=['yellow','magenta','green','blue','lightblue','red','cyan']

plt.pie(wins,labels=countries,autopct='%1.1f%%',colors=colors,startangle=90,explode=[0.2,0.2,0.2,0.2,0.2,0.2,0.2,0.2],shadow=True)

#Add title
plt.title('FIFA World Cup Wins')

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

OUTPUT: -

EXPERIMENTS 6: -

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

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

X=np.linspace(1,20,10)
print(X)
Y1=X
Y2=np.square(X)
print(Y2)
Y3=np.sqrt(X)

# Line plot
plt.plot(X,Y1,'r',X,Y2,'b',X,Y3,'g')
plt.legend(['X','X-square','Square-root'])

# Add title to the plot
plt.title('line plot')
plt.grid()

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

OUTPUT: -

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

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

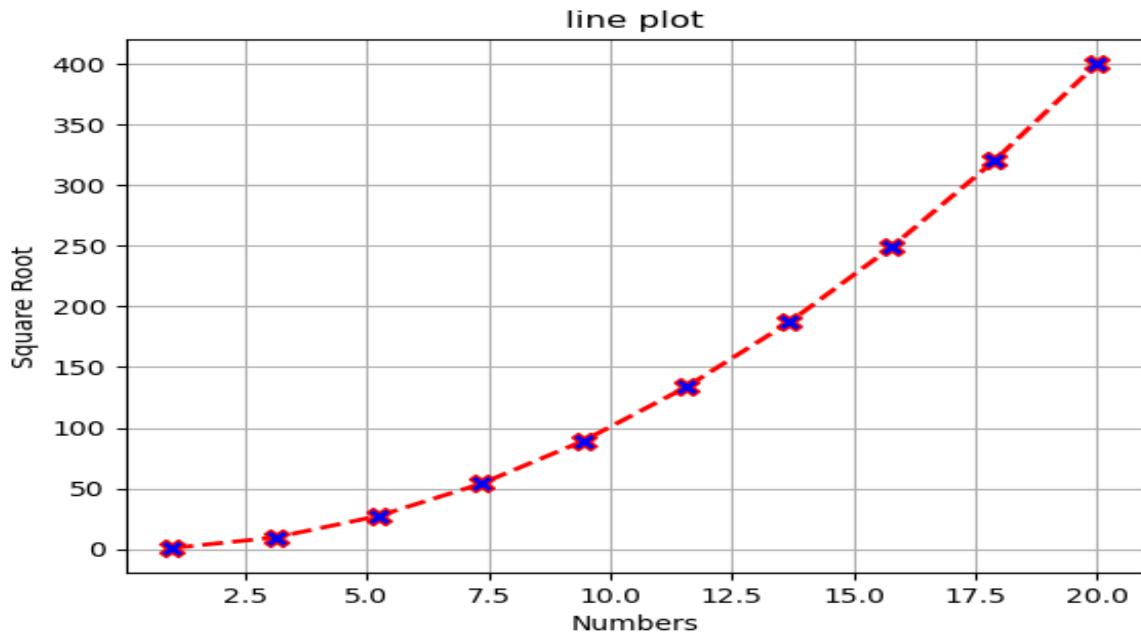
X=np.linspace(1,20,10)
print(X)

Y2=np.square(X)
print(Y2)

# Line plot
plt.plot(X,Y2,marker='X',linestyle='dashed',color='red',linewidth=2,markerfacecolor='blue',markersize=8)

# Add title to the plot
plt.title('line plot')
plt.xlabel('Numbers')
plt.ylabel('Square Root')
plt.grid()

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

OUTPUT:-

EXPERIMENTS 7: -

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

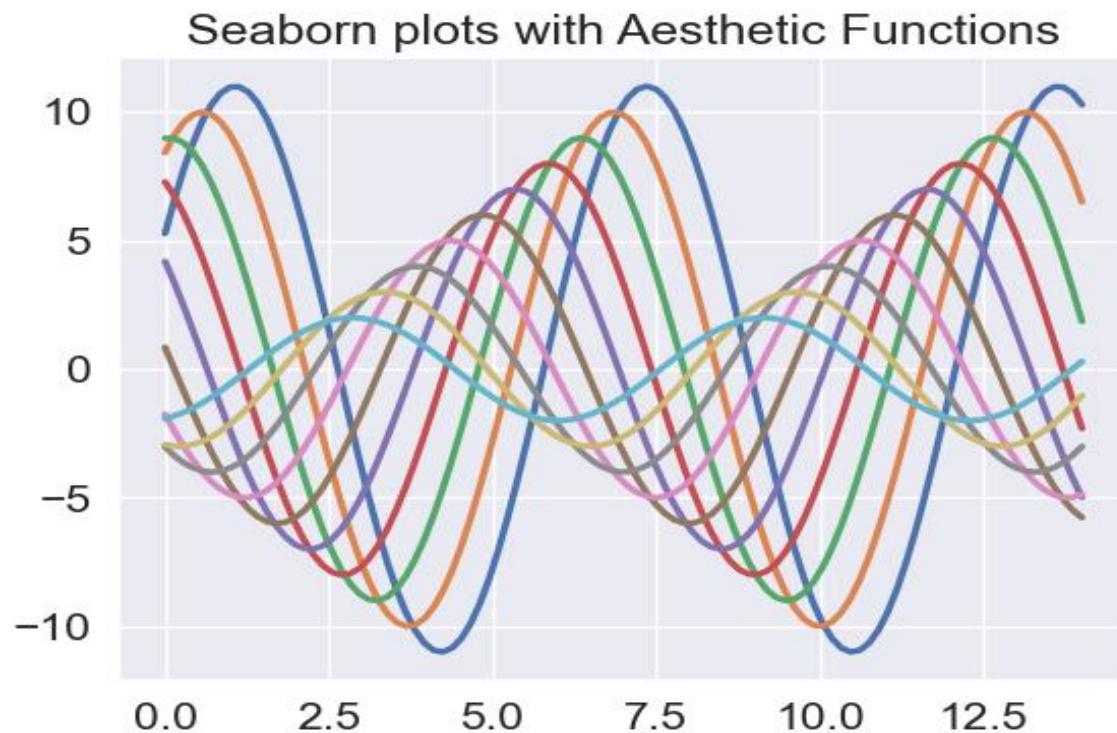
```
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

def sinplot(n=10):
    x=np.linspace(0,14,100)
    for i in range(1,n+1):
        plt.plot(x,np.sin(x+i*.5)*(n+2-i))

# To set the theme
sns.set()
sns.set_context("notebook",font_scale=1.5,rc={"lines.linewidth":2.5})

sinplot()
plt.title('Seaborn plots with Aesthetic Functions')

plt.show()
```

OUTPUT: -

EXPERIMENTS 8: -

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

```
from bokeh.plotting import figure, output_file, show
from bokeh.models import Legend, LegendItem, Title, Span

# Output to an HTML file
output_file("line_graph_with_annotations.html")

# Sample data
x = [1, 2, 3, 4, 5]
y1 = [2, 5, 7, 2, 8]
y2 = [1, 4, 5, 3, 6]

# Create a Bokeh figure
p = figure(title="Line Graph with Annotations and Legends", x_axis_label='X-axis', y_axis_label='Y-axis')

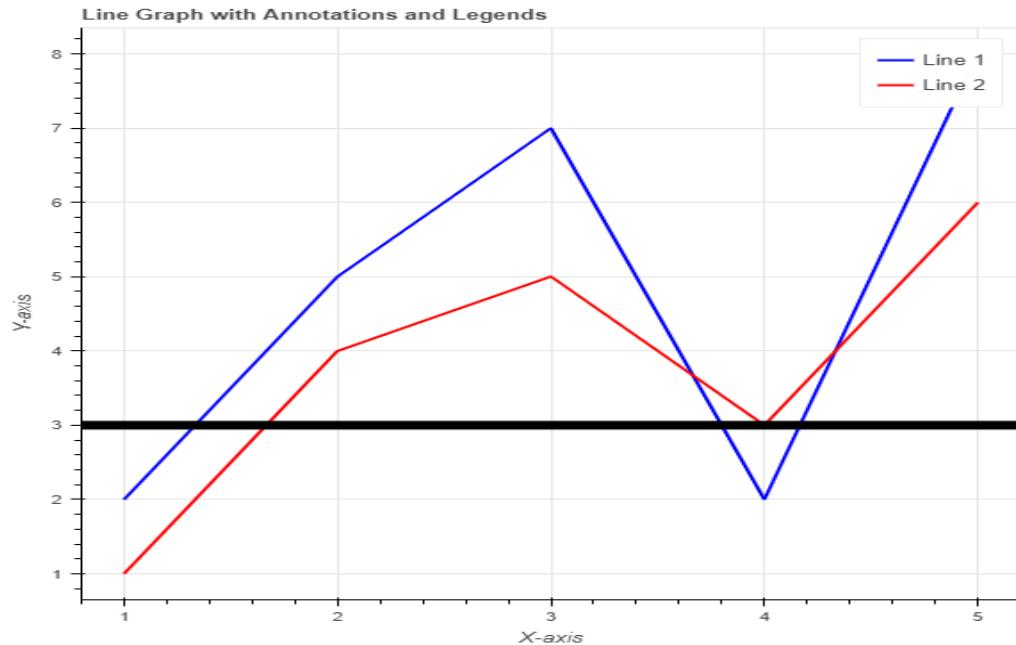
# Add lines to the figure
line1 = p.line(x, y1, line_width=2, color="blue", legend_label="Line 1")
line2 = p.line(x, y2, line_width=2, color="red", legend_label="Line 2")

# Add annotations (vertical line and a text label)
annotation = Span(location=3, dimension='width', line_color='black', line_width=8)

# Add the annotation to the figure
p.add_layout(annotation)

# Create a legend
legend = Legend(items=[
    LegendItem(label="Line 1", renderers=[line1]),
    LegendItem(label="Line 2", renderers=[line2]),
])
p.add_layout(legend)

# Show the plot
show(p)
```

OUTPUT: -**8.a) Write a Python program for plotting different types of plots using Bokeh.**

```
import numpy as np
from bokeh.io import output_file
from bokeh.io import show
from bokeh.layouts import row, column
from bokeh.plotting import figure

# Create a new plot object.
fig = figure(width=500, height=500, title='Different Types of Plots')

# Add data to the plot.
x = np.linspace(0, 10, 100)
y = np.sin(x)

# Add a line glyph to the plot.
fig.line(x, y, line_width=2, legend_label='Line Plot')

# Add a bar chart to the plot.
fig.vbar(x=x, top=y, legend_label='Bar Chart', width=0.5, bottom=0)

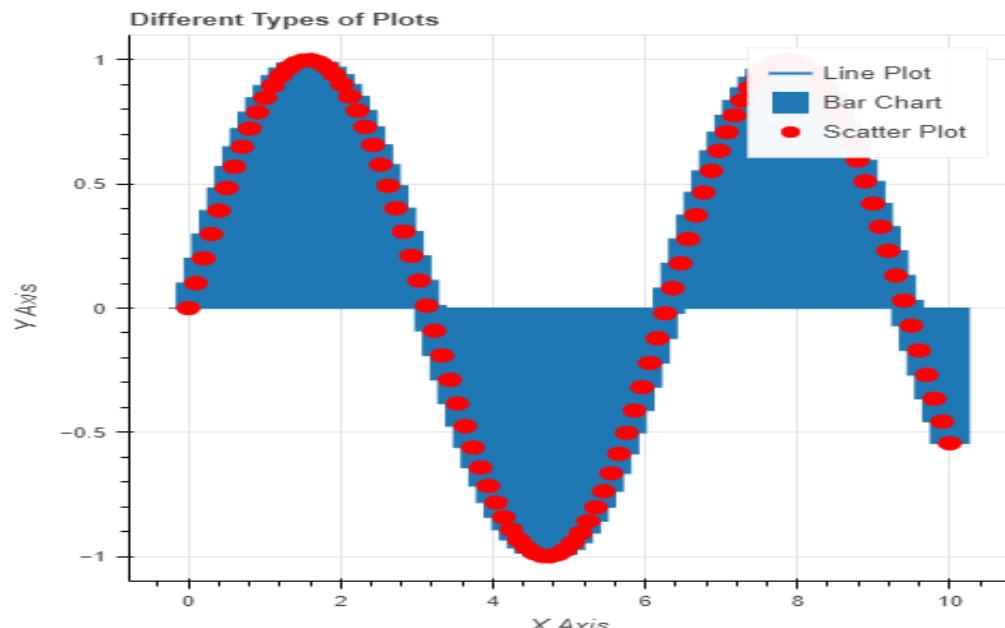
# Add a scatter plot to the plot.
fig.circle(x, y, size=10, color='red', legend_label='Scatter Plot')

# Customize the plot appearance.
fig.xaxis.axis_label = 'X Axis'
fig.yaxis.axis_label = 'Y Axis'
```

```
fig.legend.location = 'top_right'
```

```
# Show the plot.  
show(fig)
```

OUTPUT: -



EXPERIMENTS 9: -

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

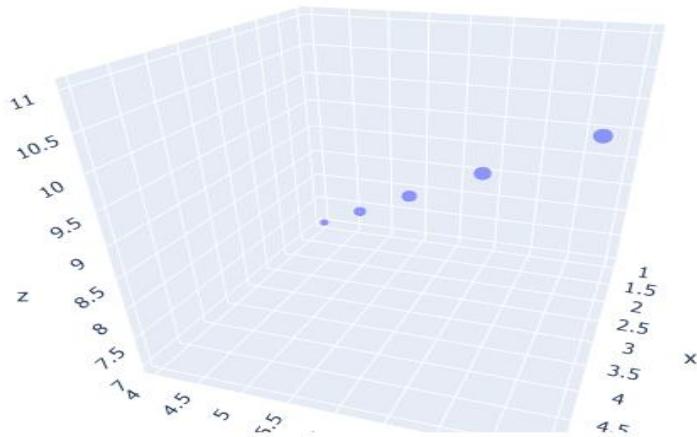
Scatter Plot

```
import plotly.express as px
import pandas as pd
import plotly.express as px

size = [10, 20, 30, 40, 50]
x = [1, 2, 3, 4, 5]
y = [4, 5, 6, 7, 8]
z = [7, 8, 9, 10, 11]

fig = px.scatter_3d(x=x, y=y, z=z, size=size)

fig.show()
```

OUTPUT: -

Line Plot

```
# Create a data frame
df = px.data.tips()

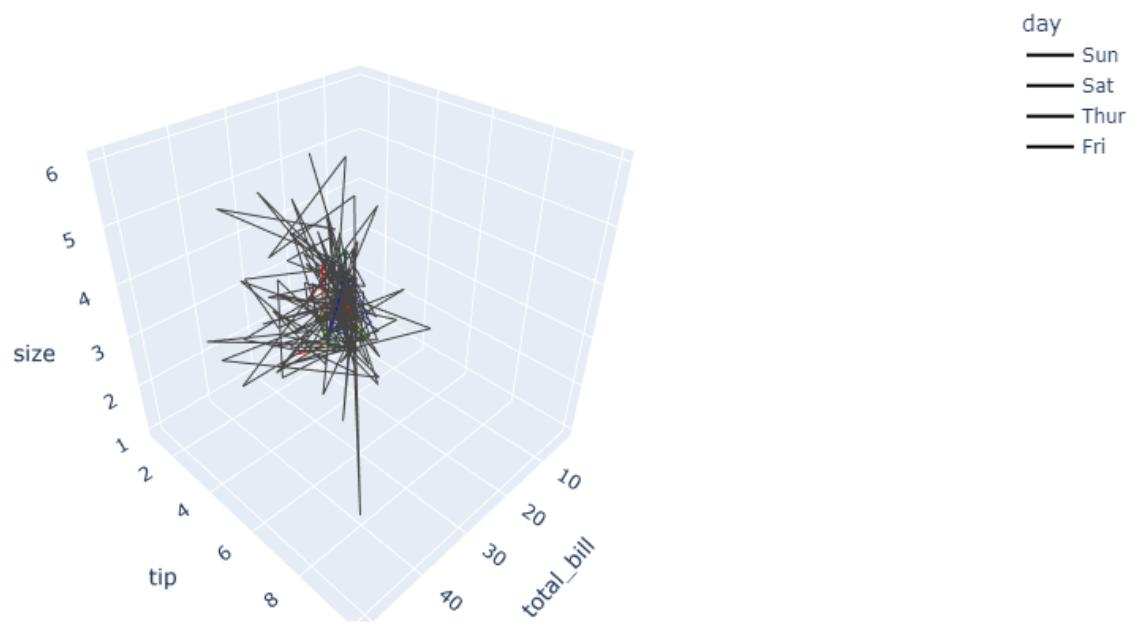
# View the contents of the data frame
print(df.head())

# Create a 3D line plot
fig = px.line_3d(
    df,
    x="total_bill",
    y="tip",
    z="size",
    color="day"
)

# Set the line color
fig.update_traces(line_color=["red", "green", "blue"])

# Display the plot
fig.show()
```

OUTPUT:-



3D Scatter plot

```
import plotly.express as px

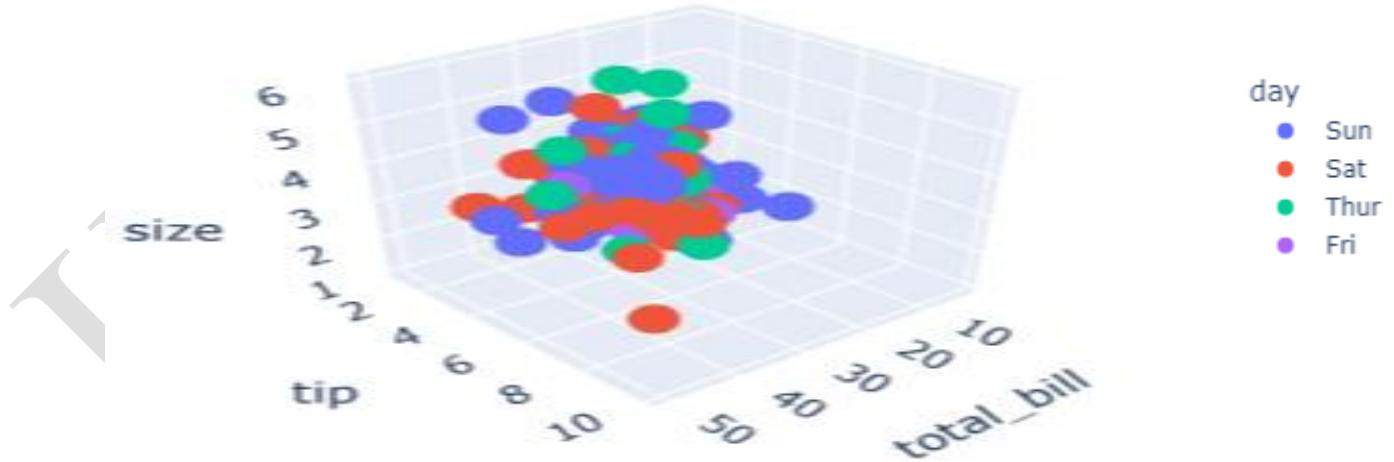
# Create a data frame
df = px.data.tips()

print(df)
# View the contents of the data frame
print(df.head())

# Create a 3D scatter plot
fig = px.scatter_3d(
    df,
    x="total_bill",
    y="tip",
    z="size",
    color="day"
)

# Display the plot
fig.show()
```

OUTPUT: -



Random Plot

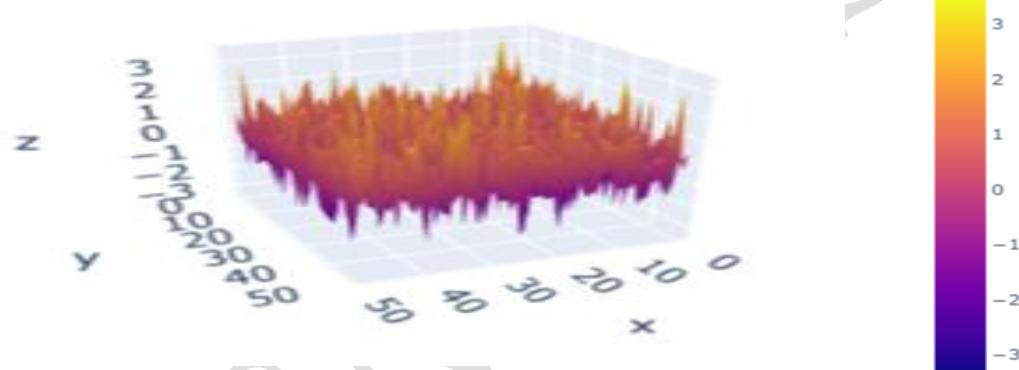
```
import plotly.graph_objects as go
import numpy as np

# Create a 50x50 array of random numbers
z = np.random.randn(50, 50)

# Create a Surface trace
surface = go.Surface(x=np.arange(50), y=np.arange(50), z=z)

# Create a Figure object
fig = go.Figure(data=[surface])

# Show the plot
fig.show()
```

OUTPUT: -

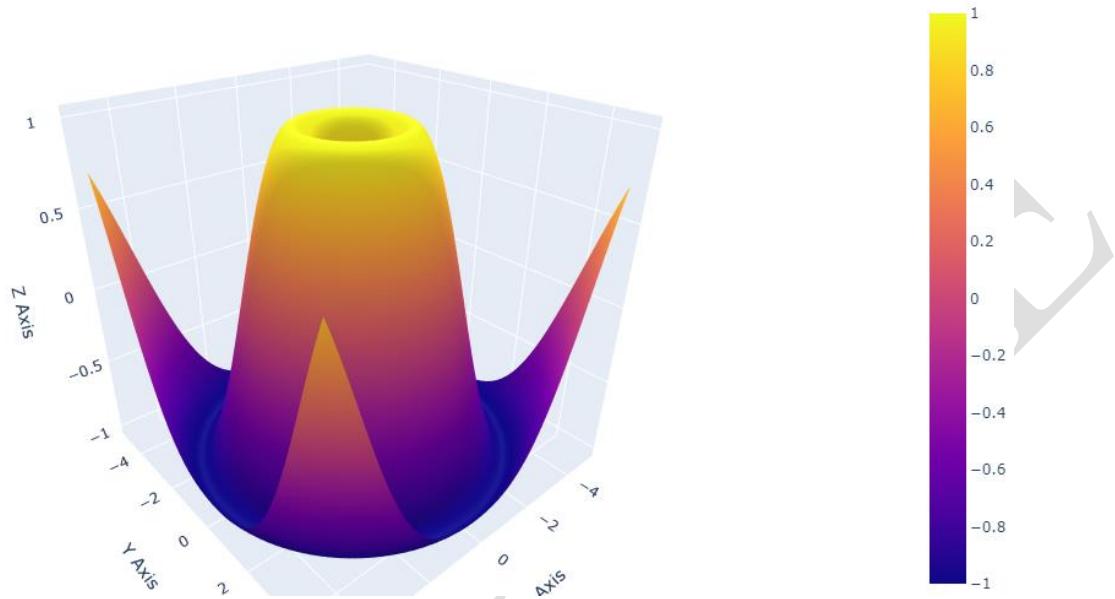
```
import plotly.graph_objects as go
import numpy as np

# Generate sample 3D data
x = np.linspace(-5, 5, 100)
y = np.linspace(-5, 5, 100)
x, y = np.meshgrid(x, y)
z = np.sin(np.sqrt(x**2 + y**2))

# Create a 3D surface plot
fig = go.Figure(data=[go.Surface(z=z, x=x, y=y)])

# Customize layout
fig.update_layout(scene=dict(
    xaxis_title='X Axis',
    yaxis_title='Y Axis',
    zaxis_title='Z Axis'),
    margin=dict(l=0, r=0, b=0, t=40),
    title='3D Surface Plot of sin(sqrt(x^2 + y^2)))')

# Display the 3D surface plot
fig.show()
```

OUTPUT: -3D Surface Plot of $\sin(\sqrt{x^2 + y^2})$ 

KSSEM

EXPERIMENTS 10: -**10.a) Write a Python program to draw Time Series using Plotly Libraries.**

```
import pandas as pd
import plotly.express as px
dollar_conv = pd.read_csv("C:\\Users\\KSSEM033\\Downloads\\CUR_DLR_INR.csv")
fig = px.line(dollar_conv, x='DATE', y='RATE', title='Dollar vs Rupee')
fig.show()
```

OUTPUT:-**10.b) Write a Python program for creating Maps using Plotly Libraries.**

```
import plotly.express as px
import pandas as pd

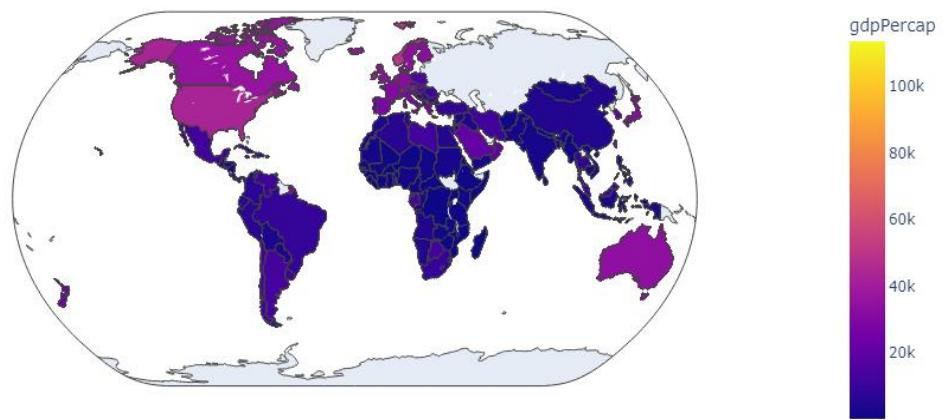
# Import data from GitHub
data=pd.read_csv('https://raw.githubusercontent.com/plotly/datasets/master/gapminder_with_codes.csv')

# Create basic choropleth map
fig = px.choropleth(data, locations='iso_alpha', color='gdpPercap',
```

```
hover_name='country',
projection='natural earth', title='GDP per Capita by Country')
fig.show()
```

OUTPUT: -

GDP per Capita by Country



VIVA QUESTIONS AND ANSWERS

1. What is Python?

Python is an interpreted scripting language that is known for its power, interactivity, and object-oriented nature. It utilizes English keywords extensively and has a simpler syntax compared to many other programming languages.

2. Python is an interpreted language. Explain.

An interpreted programming language refers to any language that executes its instructions sequentially, one line at a time. In the case of Python, programs are executed directly from the source code without the need for any intermediate compilation process.

3. What distinguishes lists from tuples?

Lists	Tuples
Lists are mutable, i.e., they can be edited	Tuples possess immutability, denoting their incapability of being modified like lists.
Lists are usually slower than tuples.	Tuples are faster than lists
Lists consume a lot of memory	Tuples consume less memory when compared to lists

4. What are the Key features of Python?

The key features of Python are as follows: Interpreted, dynamically typed, Object-oriented programming, Cross platform, General-purpose language.

5. How is Memory managed in Python?

Python makes use of automatic memory management through garbage collection. The garbage collector keeps track of objects and frees memory when they are no longer in use.

6. What are Python Modules?

Files containing Python codes are referred to as Python modules. This code can be of different types like classes, functions, or variables.

7. What is scope resolution?

In Python, a scope defines the region of code where an object remains valid and accessible. Every object in Python operates within its designated scope.

8. What is a dictionary in Python?

Python supports various data types, including dictionaries. A dictionary in Python is a collection of elements that are stored as key-value pairs.

9. What are the applications of Python?

The applications of Python are as follows:

- GUI-based desktop applications
- Image processing applications
- Business and Enterprise applications
- Prototyping
- Web and web framework applications

10. What are the two major loop statements?

The two major loop statements in python are for and while.

11. What are functions in Python?

A function is a segment of code that runs only when it is called. The “def” keyword is utilized to define a specific function.

12. What are the common built-in data types in Python?

Python supports the below-mentioned built-in data types:

Immutable data types: Number, String, Tuple

Mutable data types: List, Dictionary, Set

13. What are local variables and global variables in Python?

A local variable is a variable that is defined within a specific function and is only accessible within that function. It cannot be accessed by other functions within the program. A global variable is a variable that is declared outside of any function, allowing it to be accessed by all functions in the program

14. What is type conversion in Python?

Python offers a valuable feature that allows for the conversion of data types as needed. This process is referred to as type conversion in Python. Type conversion can be divided into two types:

Implicit Type Conversion: This type of conversion is automatically performed by the Python interpreter without requiring any user intervention.

Explicit Type Conversion: This type of conversion involves the user explicitly changing the data type to the desired type.

15. What is the difference between Python Arrays and lists?

Arrays are data structures that hold fixed-size elements of the same type. Lists are versatile data structures that can hold elements of different types and sizes.

16. Is python case sensitive?

Yes, Python is a case sensitive language. In Python, it is important to note that “Function” and “function” are distinct entities.

17. Is indentation required in Python?

Indentation is an essential aspect of Python syntax, ensuring proper code structure. It is a method used by programming languages to determine the scope and extent of code blocks.

18. How does break, continue, and pass work?

The following statements assist in altering the course of execution from the regular flow, which categorizes them as loop control statements.

- **Python break:** This statement aids in discontinuing the loop or the statement and transferring control to the subsequent statement.
- **Python continue:** This statement enforces the execution of the subsequent iteration when a particular condition is met, instead of terminating it.
- **Python pass:** This statement allows the syntactical writing of code while intending to bypass its execution. It is also recognized as a null operation, as no action is taken when the pass statement is executed.

19. How to comment with multiple lines in Python?

To include a multiline comment in Python, each line should begin with the # symbol. This practice ensures that the code is clear and easily understood.

20. What type of language is python? Programming or scripting?

Python is a versatile programming language that excels not only as a general-purpose language but also as a scripting tool.

21. What are indexes and why are they used?

To retrieve an item from a sequential collection, we can simply utilize its index, which represents the position of that specific item. Conventionally, the index commences at 0, implying that the initial element has an index of 0, the second element has an index of 1, and so forth.

22. What split(), sub(), subn() functions perform in Python?

- `split()`: This method is used to split a given string into a list.
- `sub()`: This method is used to find a substring where a regex pattern matches, and then it replaces the matched substring with a different string.
- `subn()`: This method is similar to the `sub()` method, but it returns the new string, along with the number of replacements.

23. What do you mean by Python literals?

In programming, literals are used to represent specific values assigned to variables or constants. Python offers various types of literals, including string literals, numeric literals, and boolean literals.

24. Do we need to declare variables with respective data types in Python?

No. Python is a dynamically typed language, i.e., the Python Interpreter automatically identifies the data type of a variable based on the type of value assigned.

25. What does len() do?

`len()` is an inbuilt function used to calculate the length of sequences like list, python string, and array.

26. How will you remove duplicate elements from a list?

To remove duplicate elements from the list we use the `set()` function.

27. What is the purpose of “is”, “not” and “in” operators?

Operators are referred to as special functions that take one or more values (operands) and produce a corresponding result.

- **is:** returns the true value when both the operands are true (Example: “x”is ‘x’)
- **not:** returns the inverse of the boolean value based upon the operands(example:”1” returns “0” and vice-versa.)
- **In:** helps to check if the element is present in a given Sequence or not.

28. How can you generate random numbers in Python?

This is achieved by importing the random module. It is the module that is used to generate random numbers.

29. How will you convert a string to all lowercase?

To convert a string to all lowercase in Python, you can use the built-in lower () method. The lower () method is available for strings and returns a new string with all characters converted to lowercase.

30. What is the difference between append () and extend() methods?

Both **append ()** and **extend ()** methods are methods used to add elements at the end of a list.

The primary differentiation between the append () and extend () methods in Python is that append () is used to add a single element to the end of a list. In contrast, extend () is used to append multiple aspects, such as another list or an iterable, to the end of a list.

31. What is slicing in Python?

Slicing is a technique employed to extract a specific range of elements from sequential data types, such as lists, strings, and tuples. Slicing is beneficial and easy to extract out elements. It requires a : (colon) which separates the start index and end index of the field.

32. How do you create a Python function?

Functions are defined using the **def** statement.

33. What is data visualization in Python?

Data visualization in Python refers to the process of creating graphical representations of data using Python programming language libraries such as Matplotlib, Seaborn, Plotly, etc. It allows for the exploration and communication of patterns and relationships within the data.

34. What is Matplotlib?

Matplotlib is a Python plotting library that provides a wide range of 2D and 3D plots for visualizing data. It is widely used for creating static, interactive, and animated visualizations in Python.

35. What is Seaborn?

Seaborn is a Python visualization library built on top of Matplotlib that provides a higher-level interface for creating statistical graphics. It is used for creating visually appealing and informative statistical graphics such as heat maps, pair plots, etc.

36. How do you install Matplotlib and Seaborn?

Matplotlib and Seaborn can be installed using pip, a Python package installer. To install Matplotlib, run "pip install matplotlib" in the command line. To install Seaborn, run "pip install seaborn" in the command line.

37. How do you create a scatter plot in Matplotlib?

A scatter plot can be created in Matplotlib using the "scatter" function.

38. How do you create a line plot in Matplotlib?

A line plot can be created in Matplotlib using the "plot" function.

39. How do you customize the appearance of a plot in Matplotlib?

The appearance of a plot in Matplotlib can be customized using various functions such as "title", "xlabel", "ylabel", "xlim", "ylim", "legend", "grid", etc.

40. What is data visualization, and why is it important?

Data visualization is the representation of data in a graphical or pictorial format that makes it easy to understand and interpret. It is important because it allows analysts to see patterns, trends, and insights that may be difficult to discern from raw data.

41. What are some common tools used for data visualization?

There are many tools available for data visualization, but some common ones include Python libraries like Matplotlib, Seaborn, and Plotly, as well as tools like Tableau, PowerBI, and Excel.

42. How do you choose the best visualization for your data?

The best visualization for your data depends on the type of data you have, the question you are trying to answer, and the audience you are presenting to. Generally, simple visualizations like bar charts and scatter plots work well for

basic analysis, while more complex visualizations like heat maps and network graphs are better for advanced analysis.

43. What is the difference between a histogram and a bar chart?

A histogram is used to represent the distribution of continuous numerical data, while a bar chart is used to compare categorical or discrete data.

44. How do you create a scatter plot in Python?

A scatter plot can be created in Python using libraries like Matplotlib or Seaborn. It is typically created using two variables, one for the x-axis and one for the y-axis. The "scatter" function in Matplotlib or Seaborn can be used to create a scatter plot.

45. How do you decide whether to use a bar chart or a line chart?

A bar chart is used to compare categorical or discrete data, while a line chart is used to show trends or changes over time. When deciding which to use, consider the type of data you have and the question you are trying to answer.

46. Can you differentiate between Matplotlib and Seaborn?

Matplotlib and Seaborn are both Python libraries used for data visualization, but they differ in their functionalities and use-cases. Matplotlib is a low-level library providing extensive control over every aspect of a plot, making it suitable for creating custom plots.

Seaborn is built on top of Matplotlib, offering a high-level interface for drawing attractive statistical graphics. It simplifies tasks such as generating complex visualizations of multi-dimensional data and provides built-in themes for styling matplotlib graphics.

47. What are Pandas?

Pandas is an open-source python library that has a very rich set of data structures for data-based operations. Pandas with their cool features fit in every role of data operation, whether it be academics or solving complex business problems. Pandas can deal with a large variety of files and are one of the most important tools to have a grip on.

48. Explain Bokeh?

Bokeh is a data visualization library in Python that provides high-performance interactive charts and plots. Bokeh output can be obtained in various mediums like notebook, html and server.

49. What is Plotly?

Plotly is a graphing library for creating interactive, publication-quality graphs online.

50. What are some common use cases for Plotly?

Plotly is a graphing library that is commonly used for creating interactive visualizations. Some common use cases for Plotly include creating line graphs, bar charts, scatter plots, and histograms.

51. What are some of the main features offered by Plotly?

Plotly is a graphing library that offers a number of features, including:

- The ability to create interactive, web-based graphs.
- A wide variety of graph types available.
- The ability to share graphs online.
- The ability to export graphs to a number of different formats.

52. What is the difference between Plotly and Matplotlib?

Plotly is a newer library that is built on top of Matplotlib. It offers some additional features, such as interactivity and online plotting, that Matplotlib does not.

53. What does Plotly offer that other data visualization libraries don't?

Plotly is unique in that it offers online graphing and data analysis, as well as the ability to easily share your graphs and data online. Other data visualization libraries may offer some of these features, but Plotly makes it easy and user-friendly.

THE END