

Bapuji Institute Of Engineering & Technology

Davanagere, Karnataka-570003

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



“Data Visualization with Python”

(BCS358D)

As per VTU Syllabus



PREPARED BY:

Dr. Naresh Patel K M

Associate Professor,

Department of Computer Science & Engineering

Bapuji Institute of Engineering & Technology,

Davanagere-04

Bapuji Educational Association (Regd.)
Bapuji Institution of Engineering and Technology, Davangere
Department of Computer Science and Engineering

Vision & Mission of the Program

Vision of the Department:

To be a centre-of-excellence by imbibing state-of-the-art technology in the field of Computer Science and Engineering, thereby enabling students to excel professionally and be ethical.

Mission of the Department:

M1	Adapting best teaching and learning techniques that cultivates Questioning and Reasoning culture among the students.
M2	Creating collaborative learning environment that ignites the critical thinking in students and leading to the innovation.
M3	Establishing Industry Institute relationship to bridge the skill gap and make them industry ready and relevant.
M4	Mentoring students to be socially responsible by inculcating ethical and moral values.

Program Specific Outcomes

A graduate of the Computer Science and Engineering Program will demonstrate:

PSOs	Program Specific Outcomes
PSO1	Analyze and develop solutions for problems that are complex in nature by applying the knowledge acquired from the core subjects of this program.
PSO2	Ability to develop Secure, Scalable, Resilient and distributed applications for industry and societal requirements.
PSO3	Ability to learn and apply the concepts and construct of emerging technologies like Artificial Intelligence, Machine learning, Deep learning, Big Data Analytics, IoT, Cloud Computing etc for any real time problems.

Program Educational Objectives

PEO1	To apply skills acquired in the discipline of Computer Science and Engineering for solving societal and industrial problems with apt technology intervention.
PEO2	To continue their career in industry/academia or to pursue higher studies and research.
PEO3	To become successful entrepreneurs, innovators to design and develop software products and services that meets the societal, technical and business challenges.
PEO4	To work in the diversified environment by acquiring leadership qualities with effective communication skills accompanied by professional and ethical values.

Program Outcomes

Computer Science & Engineering Graduates will be able to:

POs	Program Outcomes
PO1	Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and computer science and engineering to the solution of complex engineering problems.
PO2	Problem analysis: Identify, formulate, review research literature, and analyze complex computer engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
PO3	Design/development of solutions: Design solutions for complex computer engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO4	Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
PO5	Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern computer engineering and IT tools including prediction and modeling to complex computer engineering activities with an understanding of the limitations.
PO6	The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7	Environment and sustainability: Understand the impact of the professional computer engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO8	Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the computer engineering practice.
PO9	Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO10	Communication: Communicate effectively on complex computer engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO11	Project management and finance: Demonstrate knowledge and understanding of the computer engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO12	Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

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:			
<ul style="list-style-type: none">• 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=HZARImviDxg 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> <div><div>Sample Output: Original string: Python Exercises Python Exercises Similarity between two said strings:</div><div>Sample Output: Original string: Python Exercises Python Exercise Similarity between two said strings:1.0 0.967741935483871</div></div> <p>Strings: https://www.youtube.com/watch?v=ISItwlnF0eU String functions: https://www.youtube.com/watch?v=9a3CxJyTq00</p>		

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=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&index=3</p> <p>https://www.youtube.com/watch?v=7ABCuhWO9II&list=PLjVLYmrlmjGcC0B_FP3bkJ-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=PLjVLYmrlmjGcC0B_FP3bkJ-JIPkV5GuZR&index=6</p> <p>https://www.youtube.com/watch?v=PSji21jUNO0&list=PLjVLYmrlmjGcC0B_FP3bkJ-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-osiE80TeTvipOqomVEeZ1HRrcEvtZB</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-gkv9H3HtPbBVA8M94&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.

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 Test1 marks: "))
test2 = float(input("Enter Test2 marks: "))
test3 = float(input("Enter Test3 marks: "))

if test1 <= test2 and test1 <= test3:
    min_score = test1
elif test2 <= test1 and test2 <= test3:
    min_score = test2
else:
    min_score = test3

average = (test1 + test2 + test3 - min_score) / 2
print("The Average of Best Two Tests is: ", average)
```

Output:

```
Enter Test1 marks: 45
Enter Test2 marks: 20
Enter Test3 marks: 40
The Average of Best Two Tests is: 43
```

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 number: "))
str_val = str(val)
if str_val == str_val[::-1]:
    print("Number is Palindrome")
else:
    print("Number is not Palindrome")

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

Output1:

```
Enter a number: 1001
Number is Palindrome
1 appears 2 times
0 appears 2 times
```

Output2:

```
Enter a number: 10011
Number is not Palindrome
1 appears 3 times
0 appears 2 times
```

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:
        print("Incorrect input")
    # First Fibonacci number is 0
    elif n == 1:
        return 0
    # Second Fibonacci number is 1
    elif n == 2:
        return 1
    else:
        return Fibonacci(n-1)+Fibonacci(n-2)
```

```
n = int(input("Enter a number:"))
print("Fibonacci number is:")
print(Fibonacci(n))
```

Output1:

Enter a number: 5
Fibonacci number is: 3

Output2:

Enter a number: -5
Incorrect input

Output3:

Enter a number: 1
Fibonacci number is: 0

Output4:

Enter a number: 2
Fibonacci number is: 1

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

```
def BinToDec(b):
    return int(b, 2)

print("Enter the Binary Number: ")
bnum = input()
dnum = BinToDec(bnum)
print("\nEquivalent Decimal Value = ", dnum)

def OctToHex(o):
    return hex(int(o, 8))

print("Enter Octal Number: ")
onum = input()
hnum = OctToHex(onum)
print("\nEquivalent Hexadecimal Value = ", hnum[2:].upper())
```

Output1:

Enter the Binary Number: 1000
Equivalent Decimal Value = 8

Output2:

Enter Octal Number: 10

Equivalent Hexadecimal Value = 8

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

```
s = input("Enter a sentence: ")
w, d, u, l = 0, 0, 0, 0
l_w = s.split()
w = len(l_w)
for c in s:
    if c.isdigit():
        d = d + 1
    elif c.isupper():
        u = u + 1
    elif c.islower():
        l = l + 1

print ("No. of Words: ", w)
print ("No. of Digits: ", d)
print ("No. of Uppercase letters: ", u)
print ("No. of Lowercase letters: ", l)
```

Output:

Enter a sentence: Welcome to the World of Python Programming

No. of Words: 7

No. of Digits: 42

No. of Uppercase letters: 4

No. of Lowercase letters: 38

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

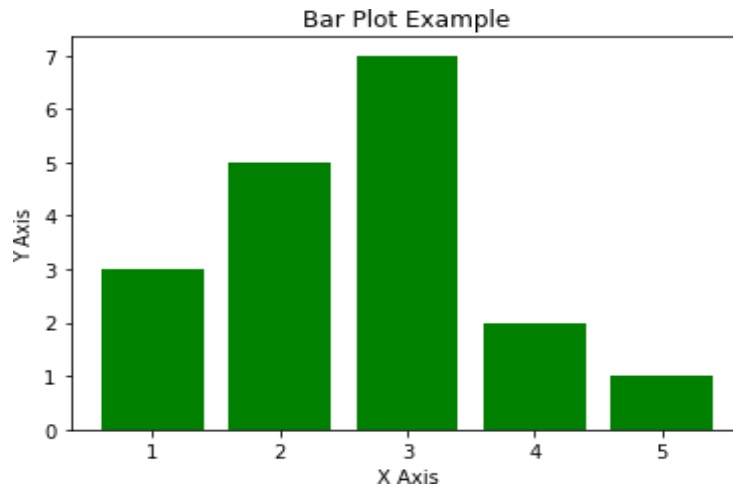
```
import difflib
str1 = "Welcome to Computer Science"
str2 = "Welcome to Computer Network"
sequence = difflib.SequenceMatcher(a=str1.lower(), b=str2.lower())
difference = sequence.ratio()*100
difference = round(difference,1)
print("Similarity between two said strings:"+str(difference))
```

Output:

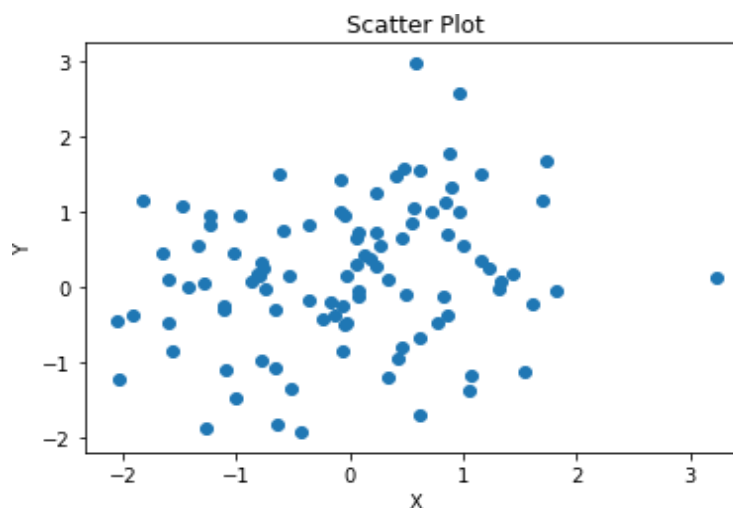
Similarity between two said strings: 77.8

4 a). Write a Python program to Demonstrate how to Draw a Bar Plot using Matplotlib. #before executing this program you have to install matplotlib package...

```
import matplotlib.pyplot as plt
x = [1, 2, 3, 4, 5]
y = [3, 5, 7, 2, 1]
plt.bar(x, y, color='green')
plt.title('Bar Plot Example')
plt.xlabel('X Axis')
plt.ylabel('Y Axis')
plt.show()
```

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

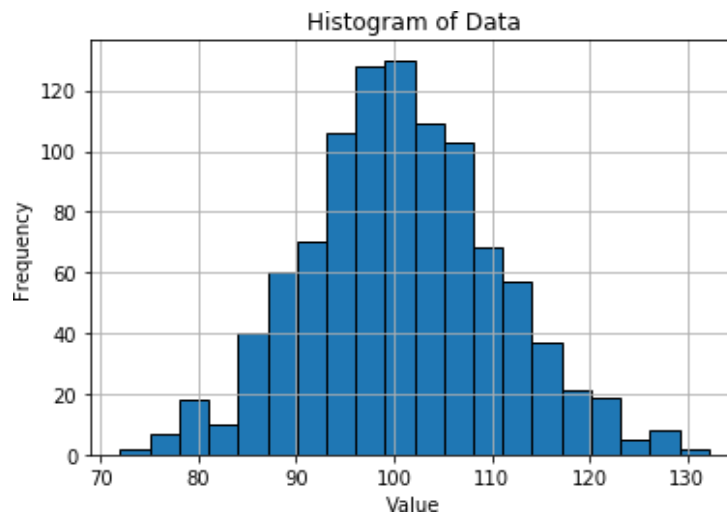
```
import matplotlib.pyplot as plt
import numpy as np
x = np.random.randn(100)
y = np.random.randn(100)
plt.scatter(x, y)
plt.title('Scatter Plot')
plt.xlabel('X')
plt.ylabel('Y')
plt.show()
```

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

```
import matplotlib.pyplot as plt
import numpy as np
data = np.random.normal(100, 10, 1000)
plt.hist(data, bins=20, edgecolor='black')
plt.xlabel('Value')
plt.ylabel('Frequency')
```

```
plt.title('Histogram of Data')
plt.grid(True)
plt.show()
```

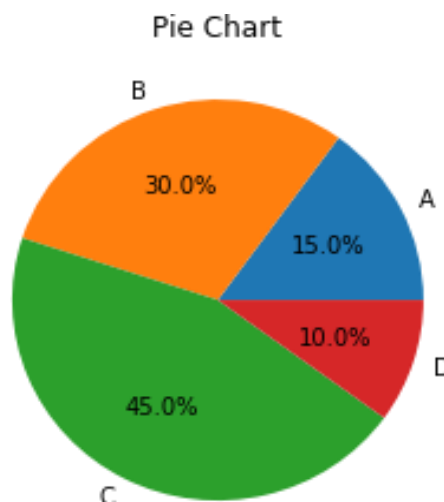
Output:



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

```
import matplotlib.pyplot as plt
labels = ['A', 'B', 'C', 'D']
sizes = [15, 30, 45, 10]
plt.pie(sizes, labels=labels, autopct="% 1.1f%% ")
plt.title("Pie Chart")
plt.show()
```

Output:



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

```
import matplotlib.pyplot as plt

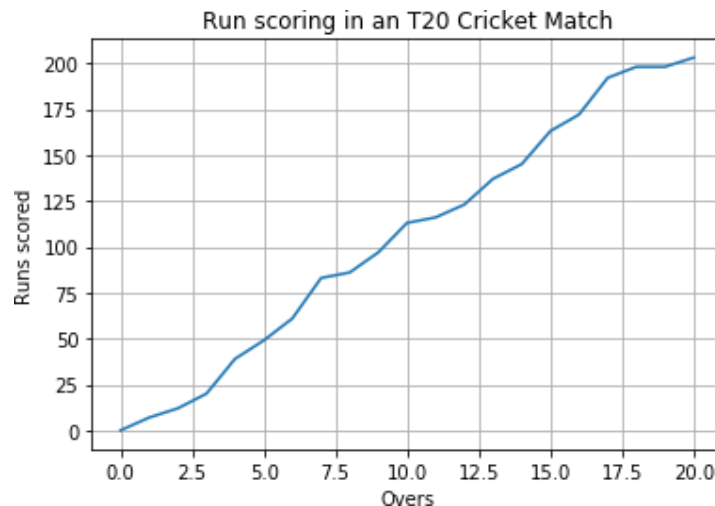
overs = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
runs_scored = [0,7,12,20,39,49,61,83,86,97,113,116,123,137,145,163,172,192,198,203]

plt.plot(overs, runs_scored)
```

```
plt.xlabel('Overs')
plt.ylabel('Runs scored')
plt.title('Run scoring in an T20 Cricket Match')

plt.grid(True)
plt.show()
```

Output:



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

```
import matplotlib.pyplot as plt
```

```
overs = [0,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20]
```

```
runs_scored = [0,7,12,20,39,49,61,83,86,97,113,116,123,137,145,163,172,192,198,198,203]
```

```
plt.plot(overs, runs_scored, marker='X', linestyle='dashed',color='red', linewidth=2,
markerfacecolor='blue', markersize=8)
```

```
plt.xlabel('Overs', color = 'green')
```

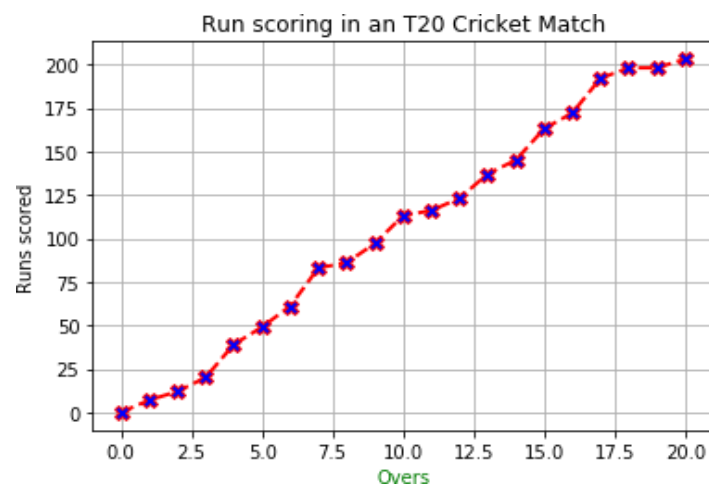
```
plt.ylabel('Runs scored')
```

```
plt.title('Run scoring in an T20 Cricket Match')
```

```
plt.grid(True)
```

```
plt.show()
```

Output:



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

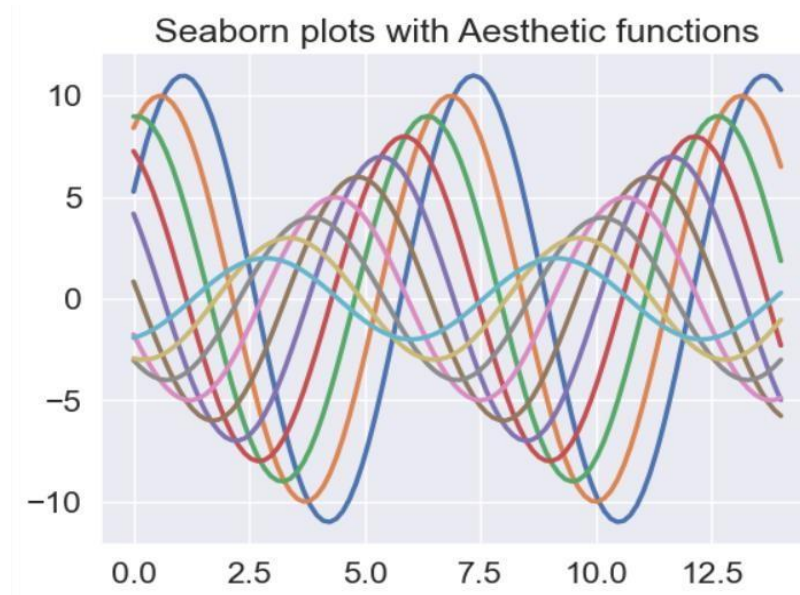
```
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

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))

sns.set_theme()
sns.set_context("notebook", font_scale=1.5, rc={"lines.linewidth": 2.5})

sinplot()
plt.title('Seaborn plots with Aesthetic functions')
plt.show()
```

Output:



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

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

```
import numpy as np
from bokeh.layouts import gridplot
from bokeh.plotting import figure, show
x = np.linspace(0, 4*np.pi, 100)
y = np.sin(x)

TOOLS = "pan,wheel_zoom,box_zoom,reset,save,box_select"

p1 = figure(title="Example 1", tools=TOOLS)

p1.circle(x, y, legend_label="sin(x)")
```

```

p1.circle(x, 2*y, legend_label="2*sin(x)", color="orange")
p1.circle(x, 3*y, legend_label="3*sin(x)", color="green")

p1.legend.title = 'Markers'

p2 = figure(title="Example 2", tools=TOOLS)

p2.circle(x, y, legend_label="sin(x)")
p2.line(x, y, legend_label="sin(x)")

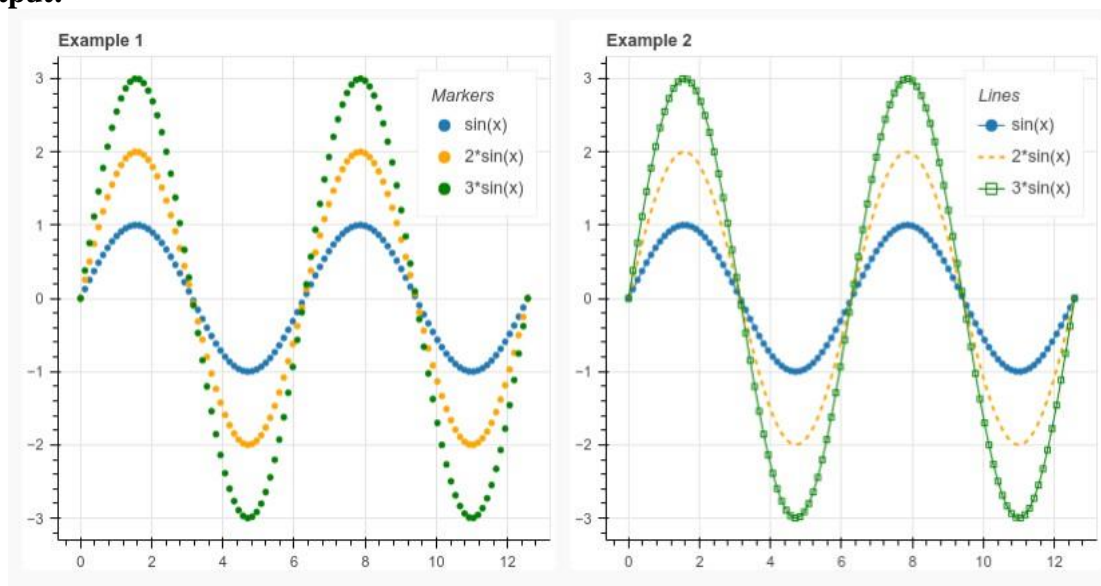
p2.line(x, 2*y, legend_label="2*sin(x)",
        line_dash=(4, 4), line_color="orange", line_width=2)

p2.square(x, 3*y, legend_label="3*sin(x)", fill_color=None, line_color="green")
p2.line(x, 3*y, legend_label="3*sin(x)", line_color="green")

p2.legend.title = 'Lines'

show(gridplot([p1, p2], ncols=2, width=400, height=400))

```

Output:**9. Write a Python program to draw 3D Plots using Plotly Libraries.**

```

import plotly.graph_objects as go
import numpy as np

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))

fig = go.Figure(data=[go.Surface(z=z, x=x, y=y)])

fig.update_layout(scene=dict(
    xaxis_title='X Axis',
    yaxis_title='Y Axis',

```

```

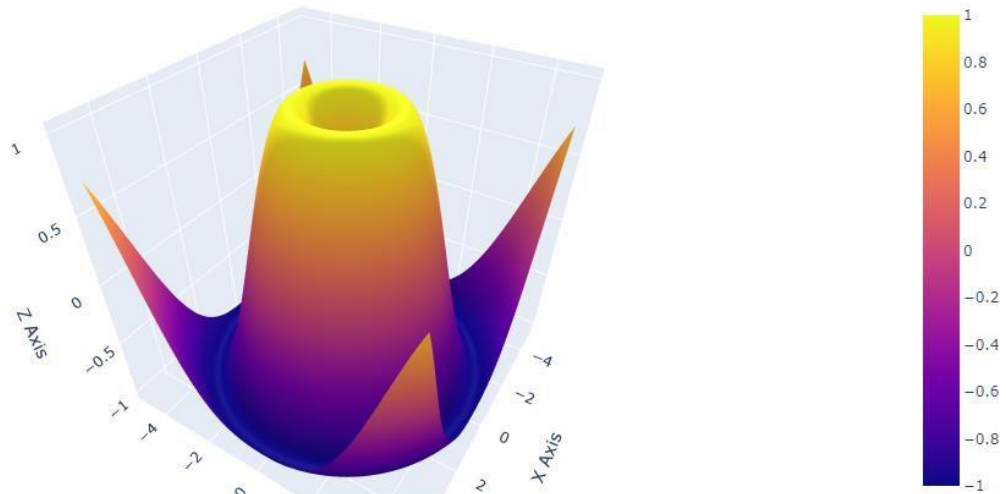
    zaxis_title='Z Axis'),
    margin=dict(l=0, r=0, b=0, t=40),
    title='3D Surface Plot of sin(sqrt(x^2 + y^2))')

```

```
fig.show()
```

Output:

3D Surface Plot of sin(sqrt(x^2+y^2))



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('CUR_DLR_INR.csv')
```

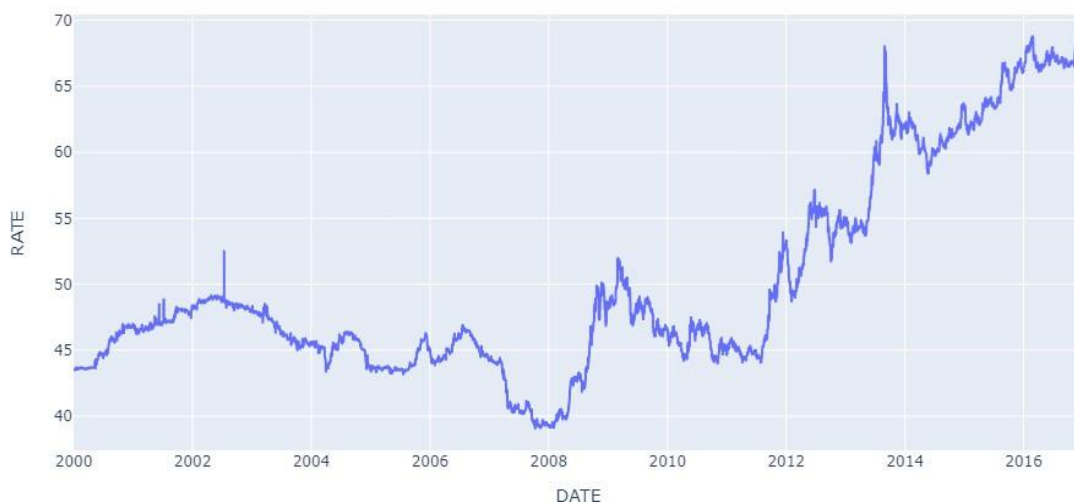
```

fig = px.line(dollar_conv, x='DATE', y='RATE', title='Dollar vs Rupee')
fig.show()

```

Output:

Dollar vs Rupee



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

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

```
data = pd.read_csv('gapminder_with_codes.csv')
```

```
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

