DEPARTMENT OF ARTIFICIAL INTELLIGENCE & DATASCIENCE

FACULTY OF ENGINEERING AND TECHNOLOGY (CO-ED)

SHARNBASVA UNIVERSITY KALABURAGI

Data Visualization Lab Manual

For 6th Semester

Course Code:**22ADL66**

By

Prof.Suhasini Patil

List of Experiments

Sl. NO	Title of the Experiment
1	a Write a program that plots a bar graph depicting number of students who are interested in learning Mathematics, Computers, English, Accountancy and Economics. b Write a program that plot marks obtained by two students in four subjects using bar graph.
2	a Draw a green-colored histogram plotting 100 random integers with number of bins being optimum. b Write a program that plots a stacked histogram of five data sets.
3	Write a program that creates boxplots of class wise performance. Make use of additional optional parameters.
4	Write a python program to draw a scatter plot comparing two subject marks of Mathematics and science. Use marks of 10 students.
5	Use the pima-Indians-diabetes-database and draw a density plot to depict the relation among variables.
6	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.
7	Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions.
8	a) Write a Python program to explain working with bokeh line graph using Annotations and Legends. (b) Write a Python program for plotting different types of plots using Bokeh.
9	Write a Python program to draw 3D Plots using Plotly Libraries.
10	a) Write a Python program to draw Time Series using Plotly Libraries. b) Write a Python program for creating Maps using Plotly Libraries.

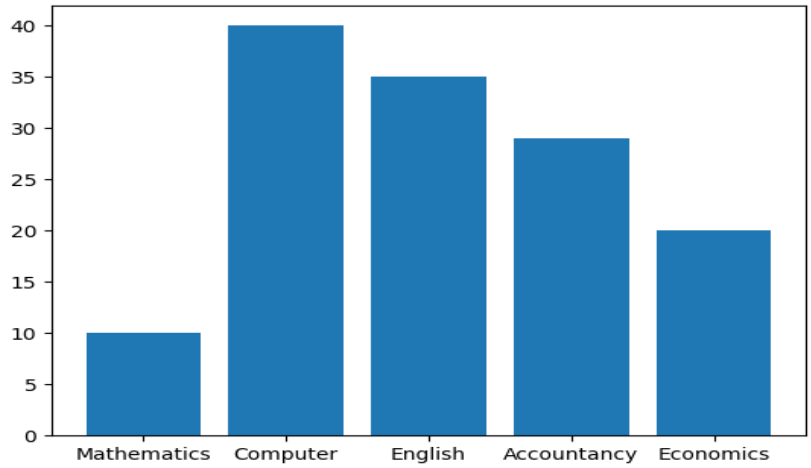
INSTRUCTIONSTO STUDENTS

- Before entering the lab the student should carry the following things (MANDATORY)
  1. Identity card issued by the college.
  2. Class notes
  3. Lab observation book
  4. Lab Manual
  5. Lab Record
- Copy all the programs to observation which are taught in class be for attending the laboratory session.
- All Students should wear uniforms while entering the laboratory
- Workspace must be kept clean and tidy after experiment is completed.
- All bags must be placed at the indicated place.
- Refer concerned staff if you need any help in using the laboratory.
- Shut down the computer properly after done with your experiment.
- Lab records need to be submitted on or be for the date of submission.
- Do not use floppy disks, pen drives on computers without permission of lab- in charge.
- Do not enter restricted areas with out permission.
- Do not talk aloud or crack jokes in laboratory.
- Do not touch any part of the computer with wet hands.
- Do not hit the keys on the computer too hard.
- Do not open any irrelevant internet sites on computer/laptop.
- Do not install or download any software or modify or delete any system files on any computers in the laboratory.
- Do not wander in the lab room and distract other students.
- You are not allowed to work in the laboratory alone or without presence of the faculty or instructor.
- Please report any unsafe behaviour or condition to the instructor or staff.

1a. Write a program that plots a bar graph depicting number of students who are interested in learning Mathematics, Computers, English, Accountancy and Economics.

```
import numpy as np
import pandas as pd
from pandas import Series, DataFrame
import matplotlib.pyplot as plt
fav_subjs=['Mathematics','Computer','English','Accountancy','Economics']
counts=[10,40,35,29,20]
plt.bar(fav_subjs,counts)
plt.show()
```

OUTPUT:

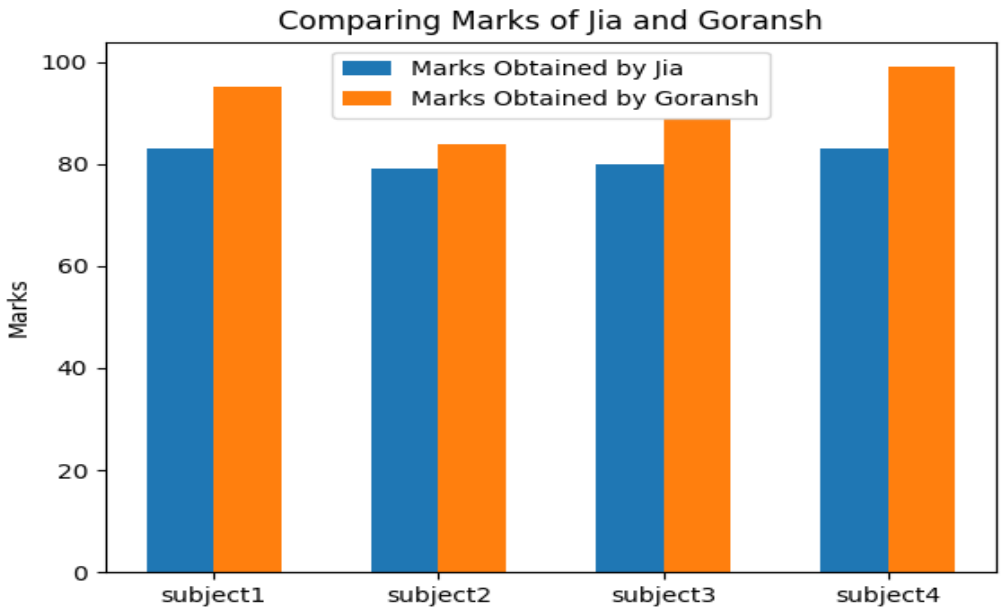


1b. Write a program that plot marks obtained by two students in four subjects using bar graph.

```
import numpy as np
import matplotlib.pyplot as plt
marks_Jia=(83,79,80,83)
marks_Goransh=(95,84,89,99)
index=np.array([1,2,3,4])
```

```
width=0.30
plt.bar(index,marks_Jia,width,label='Marks Obtained by Jia')
plt.bar(index+width,marks_Goransh,width,label='Marks Obtained by Goransh')
plt.ylabel('Marks')
plt.title('Comparing Marks of Jia and Goransh')
plt.xticks(index+width/2,('subject1','subject2','subject3','subject4'))
plt.legend(loc='best')
plt.show()
```

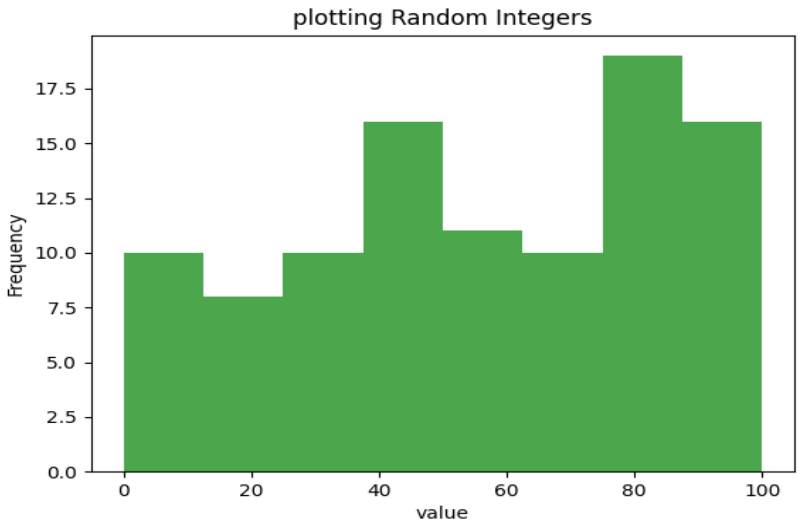
OUTPUT:



2a. Draw a green-colored histogram plotting 100 random integers with number of bins being optimum

```
import numpy as np
import matplotlib.pyplot as plt
from random import seed
from random import randint
data=[]
for i in range(100):
    value=randint(0, 100)
    data.append(value)
print(data)
plt.hist(data, bins='auto', color='green', alpha=0.7)
plt.xlabel('value')
plt.ylabel('Frequency')
plt.title('plotting Random Integers')
plt.show()
```

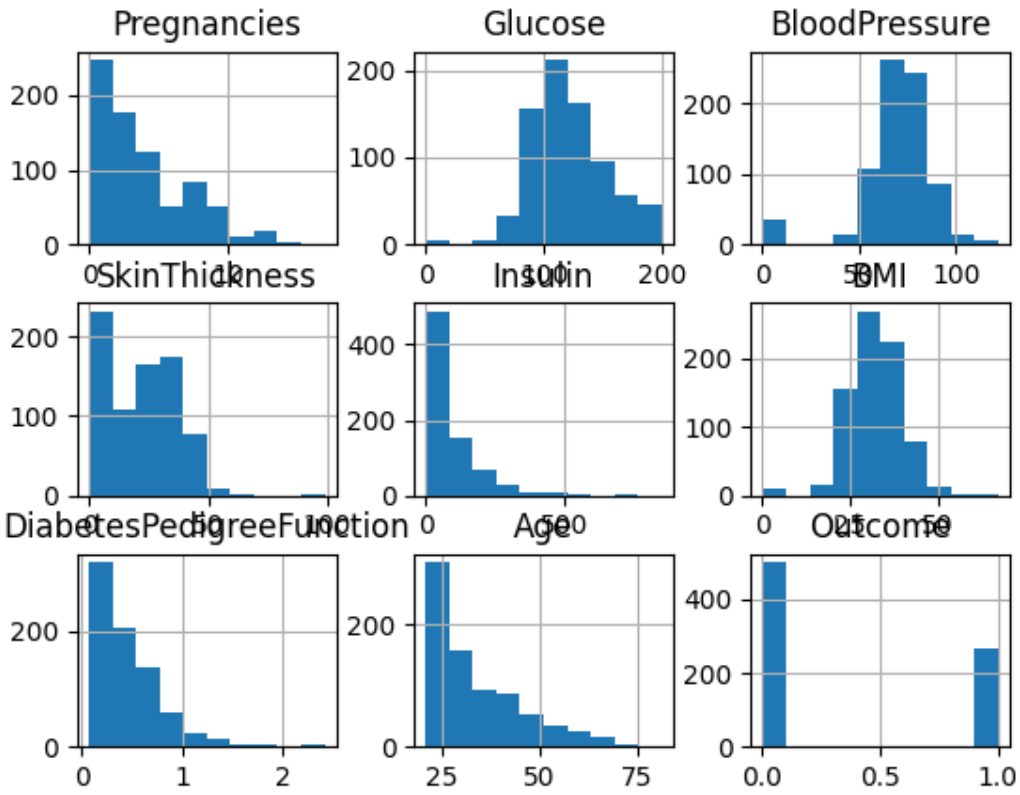
Output:



2b. Write a program that plots a stacked histogram of five data sets.

```
from matplotlib import pyplot
from pandas import read_csv
path = r"diabetes.csv"
data=read_csv(path)
data.hist()
pyplot.show()
```

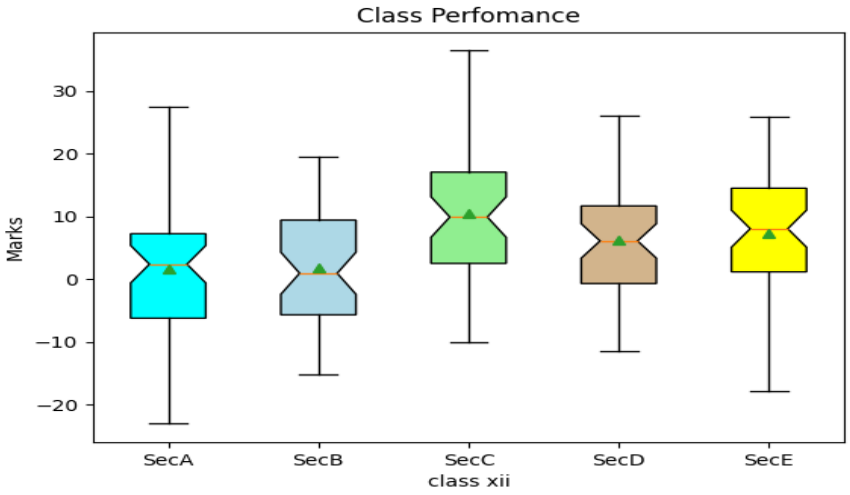
Output:



3. Write a program that creates boxplots of class wise performance. Make use of additional optional parameters.

```
import numpy as np
import matplotlib.pyplot as plt
from random import seed
from random import randint
import pandas as pd
x1 = np.random.normal(3,10,50)
x2 = np.random.normal(4,10,50)
x3 = np.random.normal(6,10,50)
x4 = np.random.normal(5,10,50)
x5 = np.random.normal(7,10,50)
data=[x1,x2,x3,x4,x5]
box=plt.boxplot(data,patch_artist=True,notch=True,vert=True,showmeans=True,labels=['SecA','SecB',
'SecC','SecD','SecE'])
plt.title('Class Perfomance')
plt.xlabel('class xii')
plt.ylabel('Marks')
colors=['cyan','lightblue','lightgreen','tan','yellow']
for patch, color in zip(box['boxes'],colors):
    patch.set_facecolor(color)
plt.show()
```

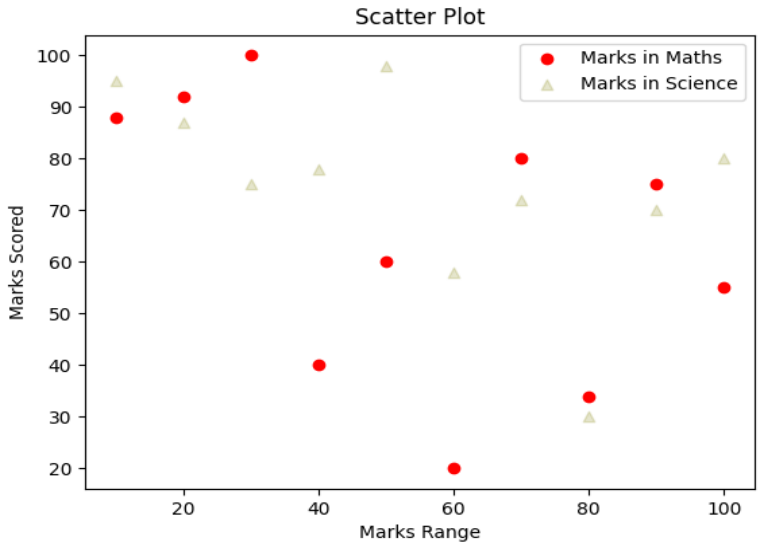
Output:


Class Perfomance
Marks
30
20
10
0
−10
−20
SecA
SecB
SecC
SecD
SecE
class xii

4 Write a python program to draw a scatter plot comparing two subject marks of Mathematics and science. Use marks of 10 students.

```
import matplotlib.pyplot as plt

math_marks = [88, 92, 100, 40, 60, 20, 80, 34, 75, 55]  # Now 10 elements
science_marks = [95, 87, 75, 78, 98, 58, 72, 30, 70, 80]  # Now 10 elements
marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]  # 10 elements

plt.scatter(marks_range, math_marks, label='Marks in Maths', color='red', marker='o')
plt.scatter(marks_range, science_marks, label='Marks in Science', color='olive', alpha=0.2, marker='^')

plt.title('Scatter Plot')
plt.xlabel('Marks Range')
plt.ylabel('Marks Scored')
plt.legend()
plt.show()
```

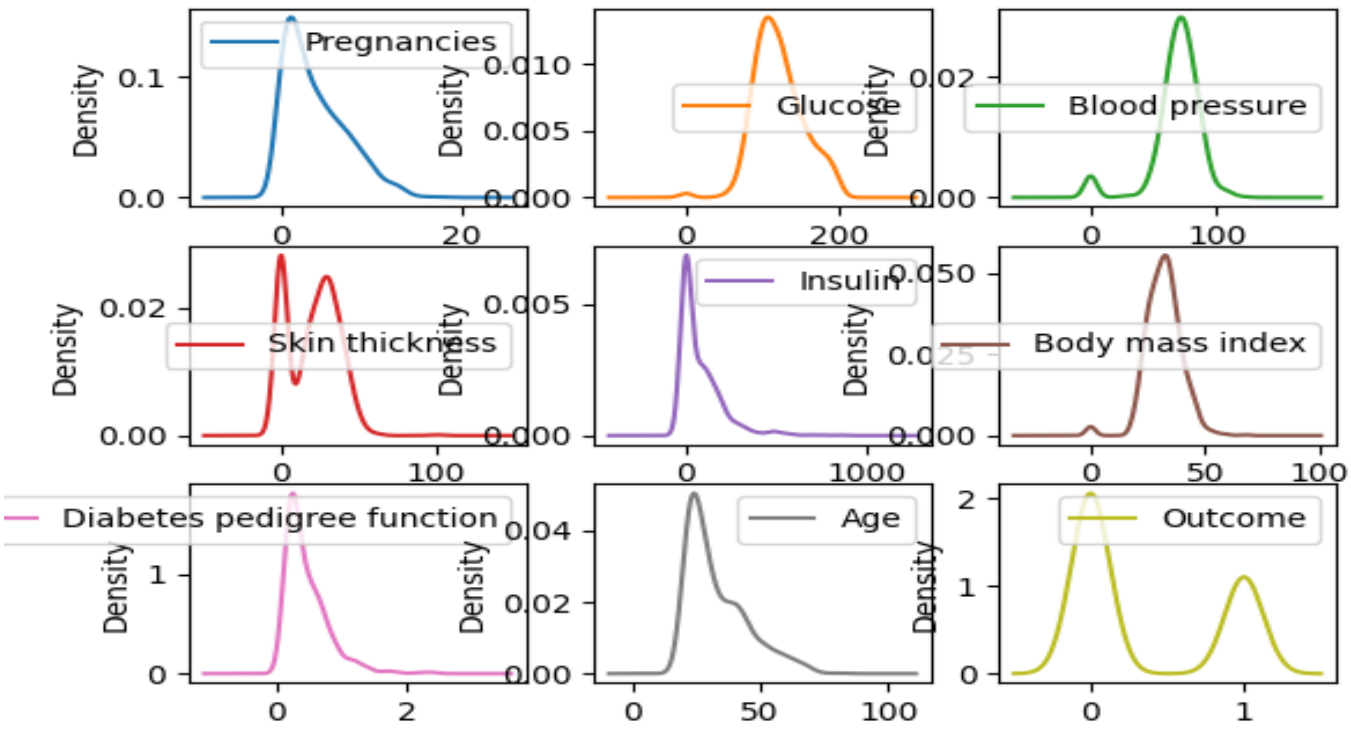
output:



5 Use the pima-Indians-diabetes-database and draw a density plot to depict the relation among variables.

```
from matplotlib import pyplot
from pandas import read_excel  # Using read_excel for an XLSX file

path = r"C:\Users\HP\Downloads\Dataset 1 _ Pima Indians diabetes dataset (PIDD).xlsx"
data = read_excel(path)  # Corrected

# Generate density plots for each column
data.plot(kind='density', subplots=True, layout=(3,3), sharex=False)

pyplot.show()
```

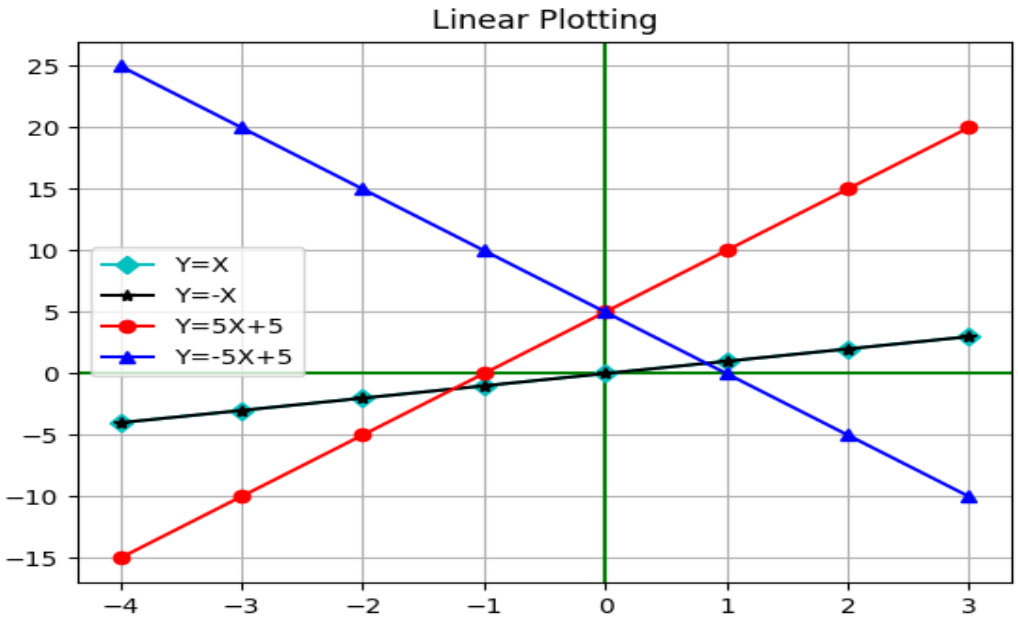
output



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

```
import matplotlib.pyplot as plt
import numpy as np
m=5
c=5
x=np.arange(-4,4)
y=np.arange(-4,4)
plt.axvline(color='g')
plt.axhline(color='g')
plt.title("Linear Plotting")
plt.plot(x,y,color='c',marker='D', label='Y=X')
plt.plot(x,y,color='k', marker='*', label='Y=-X')
plt.plot(x, m*x+c, color='r', marker='o', label='Y=5X+5')
plt.plot(x, -m*x+c, color='b',marker='^', label='Y=-5X+5')
plt.grid()
plt.legend()
plt.show()
```

OUTPUT

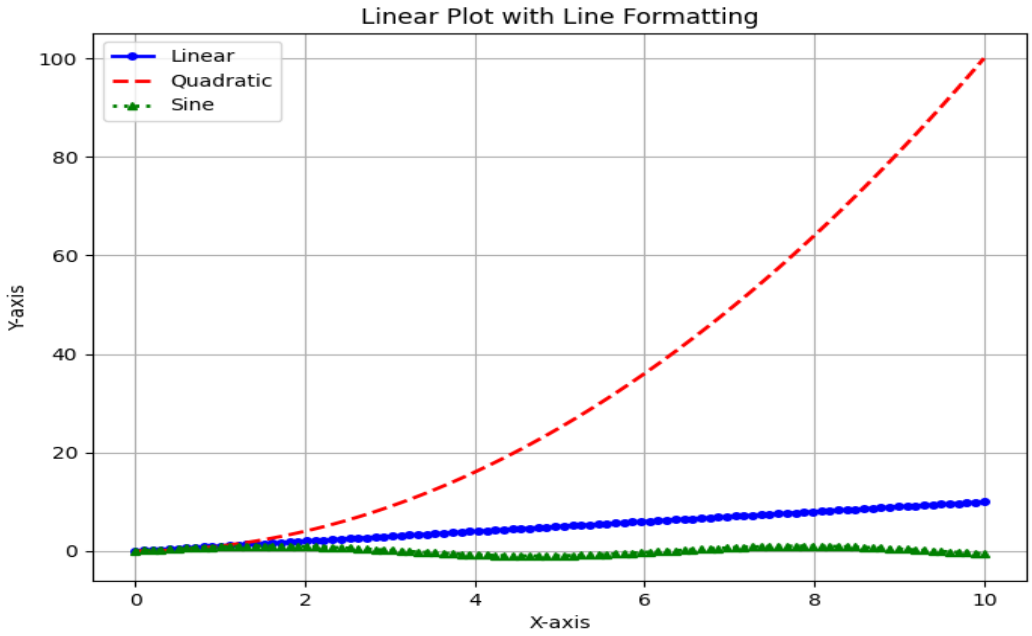


---

6 b. Write a python program to illustrate linear plotting with line formatting using Matplotlib.

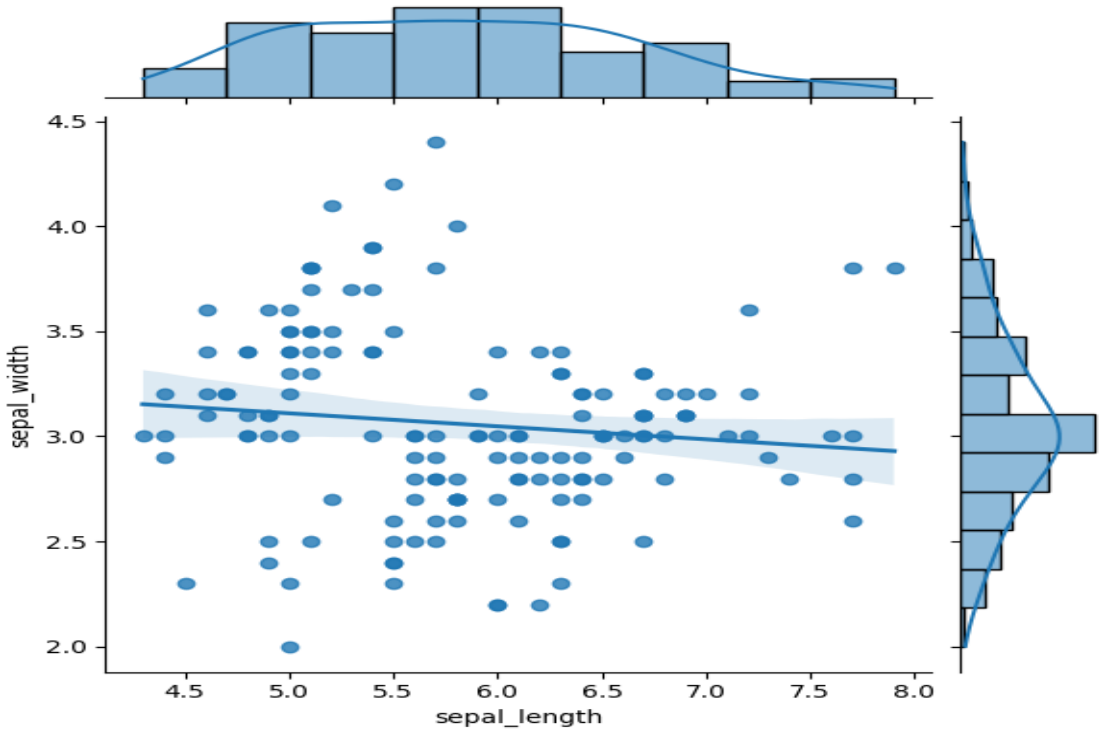
```
import matplotlib.pyplot as plt
import numpy as np
x= np.linspace (0, 10, 100)
y1 = x
y2 = x**2
y3 = np.sin(x)
plt.figure(figsize=(8, 6))
plt.plot(x, y1, label='Linear', color='blue', linestyle='-', marker='o', markersize=4, linewidth=2)
plt.plot(x, y2, label='Quadratic', color='red', linestyle='--', linewidth=2)
plt.plot(x, y3, label='Sine', color='green', linestyle=':', marker='^', markersize=4, linewidth=2)
plt.xlabel('X-axis')
plt.ylabel('Y-axis')
plt.title('Linear Plot with Line Formatting')
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT


Linear Plot with Line Formatting
Linear
Quadratic
Sine
100
80
60
40
20
0
Y-axis
0
2
4
6
8
10
X-axis

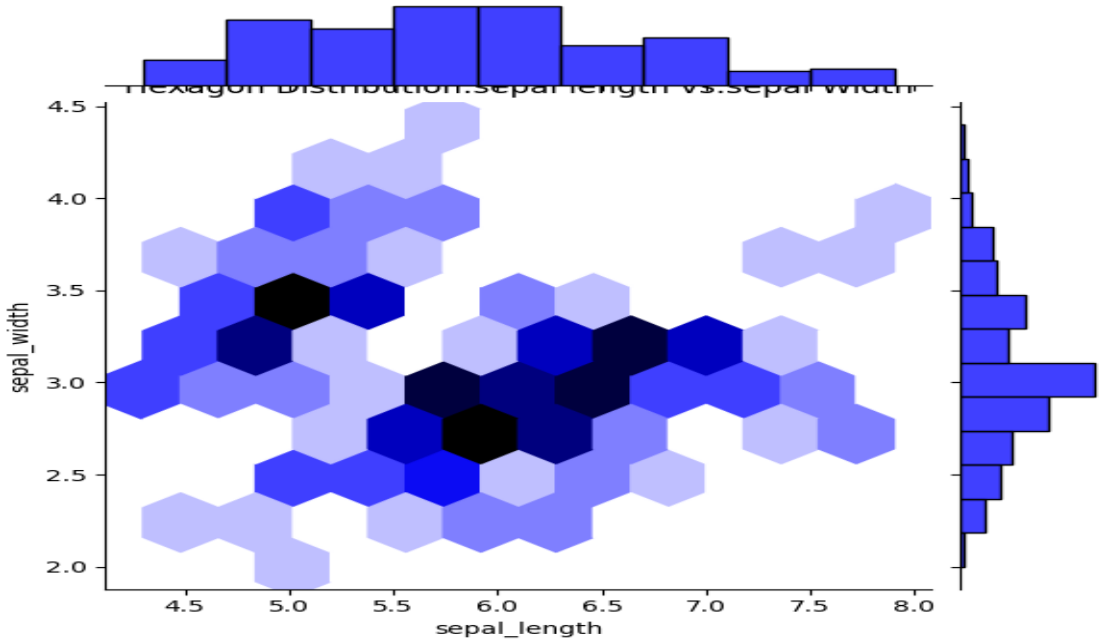
7a Write a Python program which explains uses of customizing seaborn plots with Aesthetic functions.

```
import seaborn as sns
import matplotlib.pyplot as plt
iris=sns.load_dataset("iris")
sns.jointplot(x="sepal_length",y="sepal_width",data=iris,kind="reg")
plt.show()
```



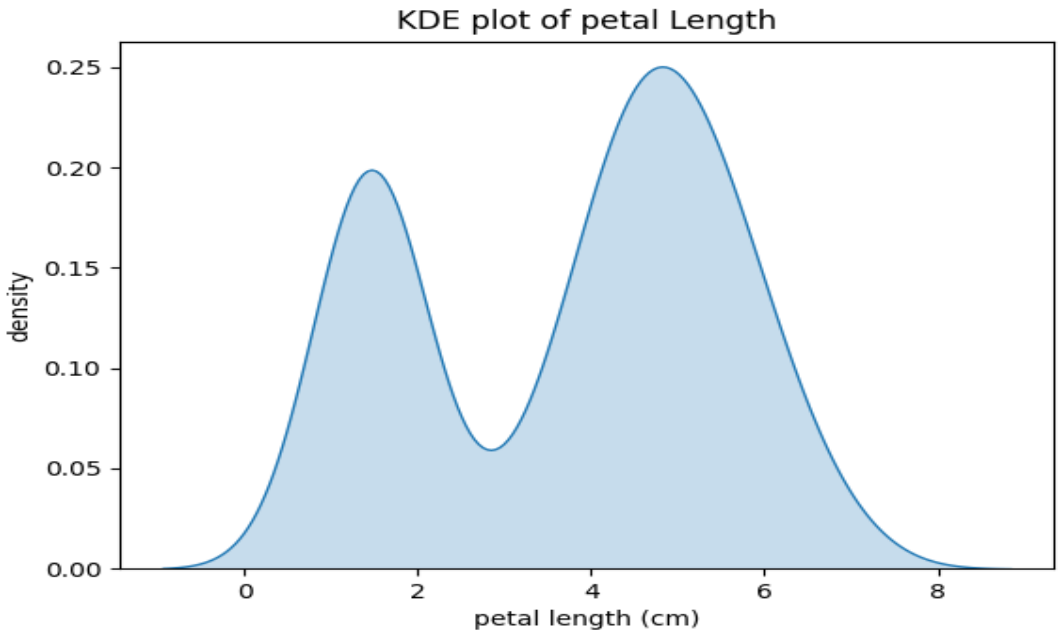
7B

```
import seaborn as sns
import matplotlib.pyplot as plt
iris=sns.load_dataset("iris")
sns.jointplot(x="sepal_length",y="sepal_width",data=iris,kind="hex",color="b")
plt.title("Hexagon Distribution:sepal length vs.sepal Width")
plt.show()
```



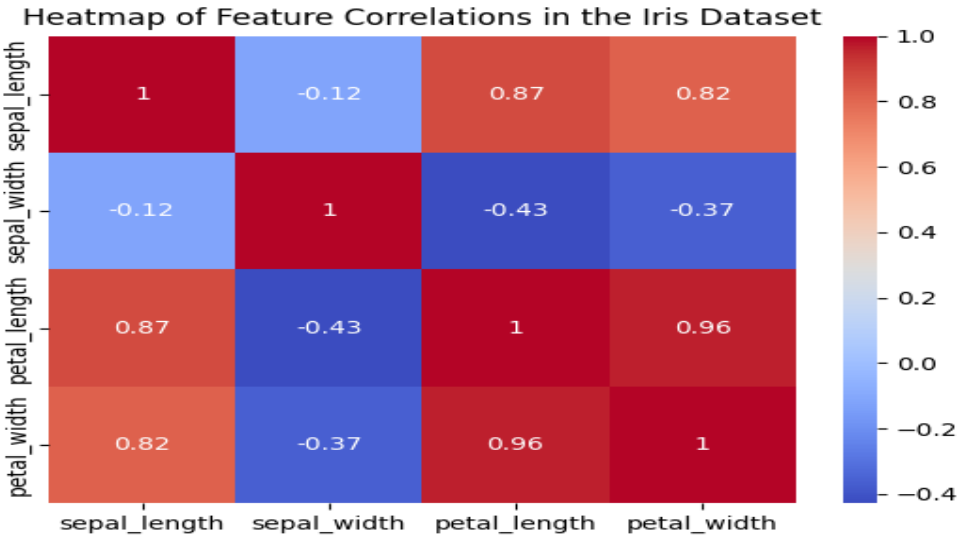
7C

```
import seaborn as sns
import matplotlib.pyplot as plt
iris=sns.load_dataset("iris")
sns.kdeplot(iris["petal_length"],shade=True)
plt.title("KDE plot of petal Length")
plt.xlabel("petal length (cm)")
plt.ylabel("density")
plt.show()
```



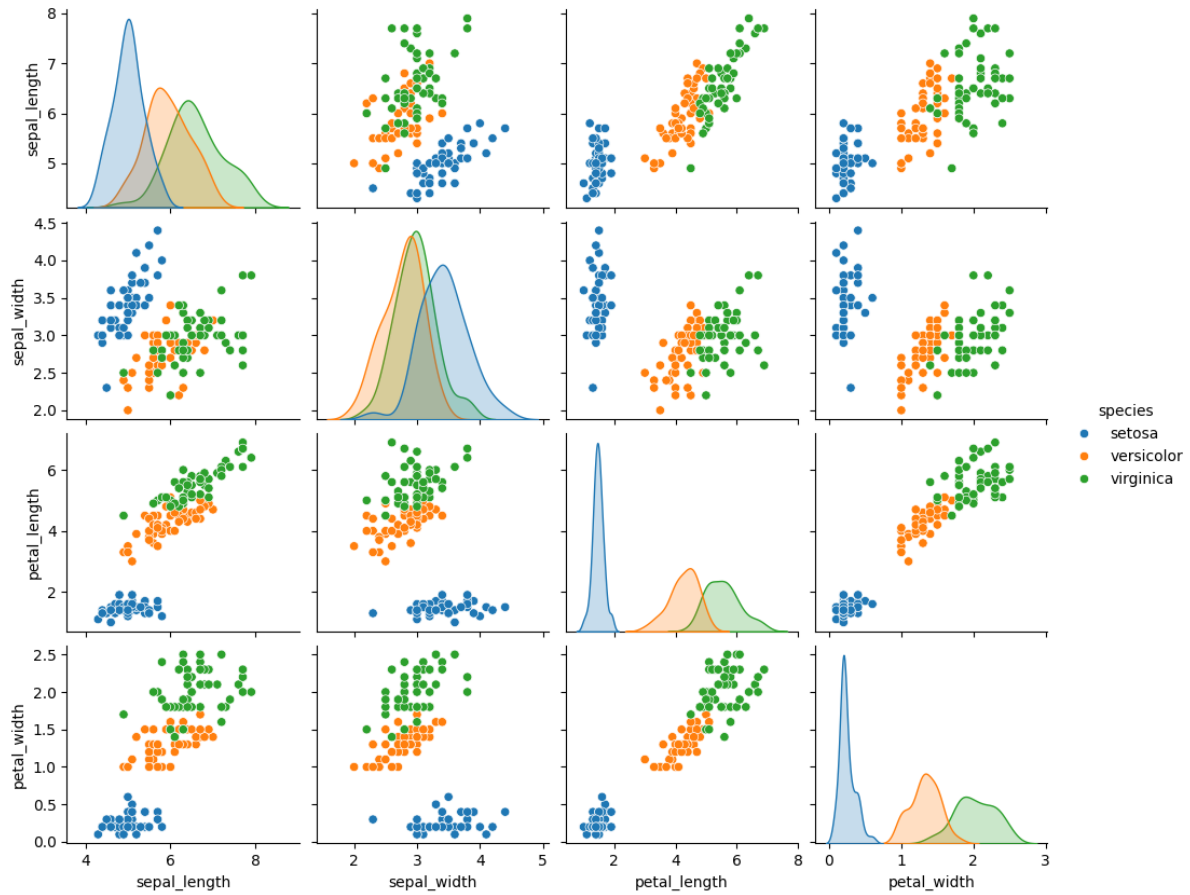
7D.

```
import seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset("iris")
correlation_matrix = iris.corr(numeric_only=True)
sns.heatmap(correlation_matrix, annot=True, cmap="coolwarm")
plt.title("Heatmap of Feature Correlations in the Iris Dataset")
plt.show()
```



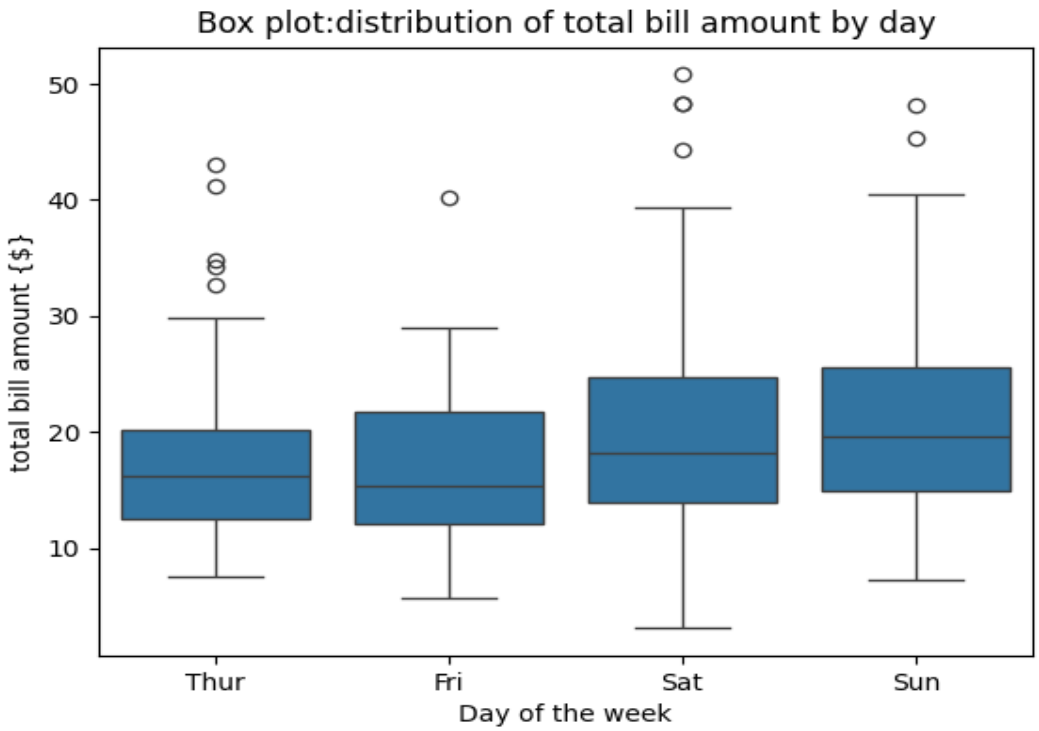
7E.

```
mport seaborn as sns
import matplotlib.pyplot as plt
iris = sns.load_dataset("iris")
sns.pairplot(iris, hue="species")
plt.show()
```



7F

```
import seaborn as sns

import matplotlib.pyplot as plt

tips=sns.load_dataset("tips")

sns.boxplot(x="day",y="total_bill",data=tips)

plt.title("Box plot:distribution of total bill amount by day")

plt.xlabel("Day of the week")

plt.ylabel("total bill amount {$}")

plt.show()
```

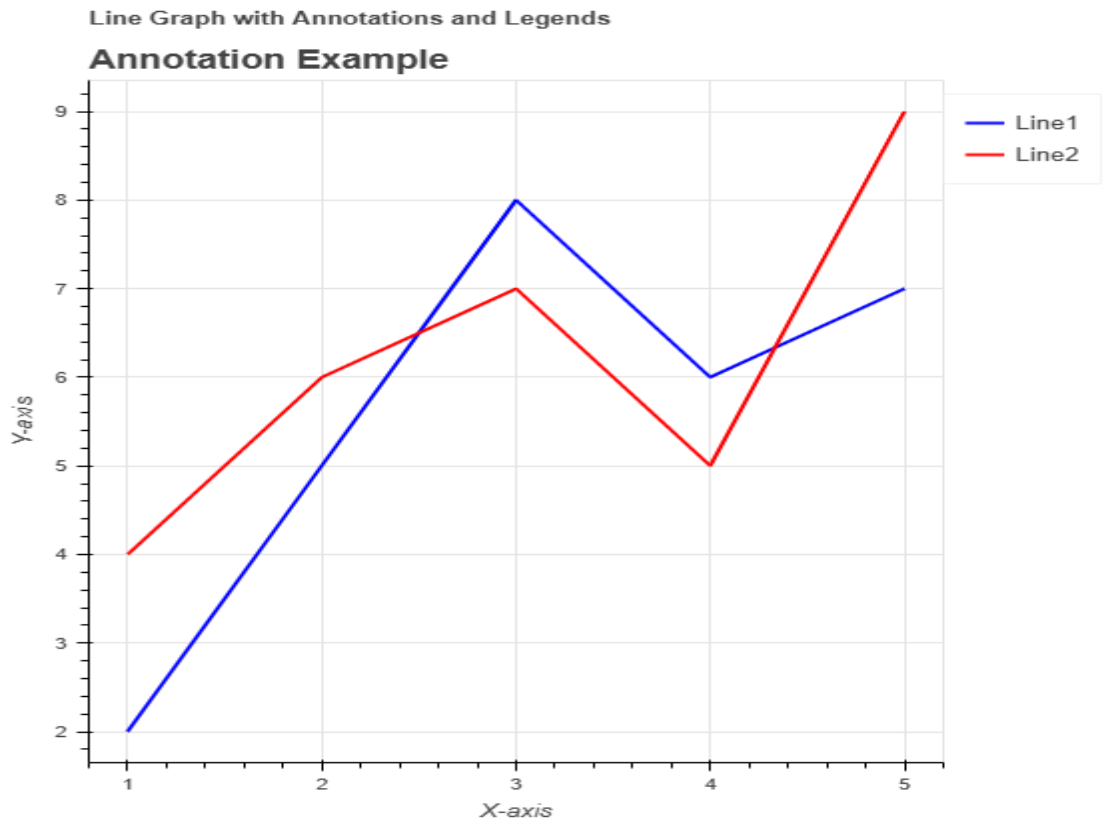

Box plot:distribution of total bill amount by day
total bill amount {$}
50
40
30
20
10
Thur
Fri
Sat
Sun
Day of the week

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

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

# This will create an HTML file and open it in your browser
output_file("line_graph_with_annotations.html")

# Sample data
x = [1, 2, 3, 4, 5]
y1 = [2, 5, 8, 6, 7]
y2 = [4, 6, 7, 5, 9]

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

# Create data sources
source1 = ColumnDataSource(data=dict(x=x, y=y1))
source2 = ColumnDataSource(data=dict(x=x, y=y2))

# Plot lines
line1 = p.line('x', 'y', source=source1, line_width=2, line_color="blue")
line2 = p.line('x', 'y', source=source2, line_width=2, line_color="red")

# Add annotation title above main title
```

```
annotation = Title(text="Annotation Example", text_font_size="14pt")
p.add_layout(annotation, 'above')

# Custom legend
legend = Legend(items=[
    LegendItem(label="Line1", renderers=[line1]),
    LegendItem(label="Line2", renderers=[line2])
])
p.add_layout(legend, 'right')

# Show the plot in browser
show(p)
```

Output:-

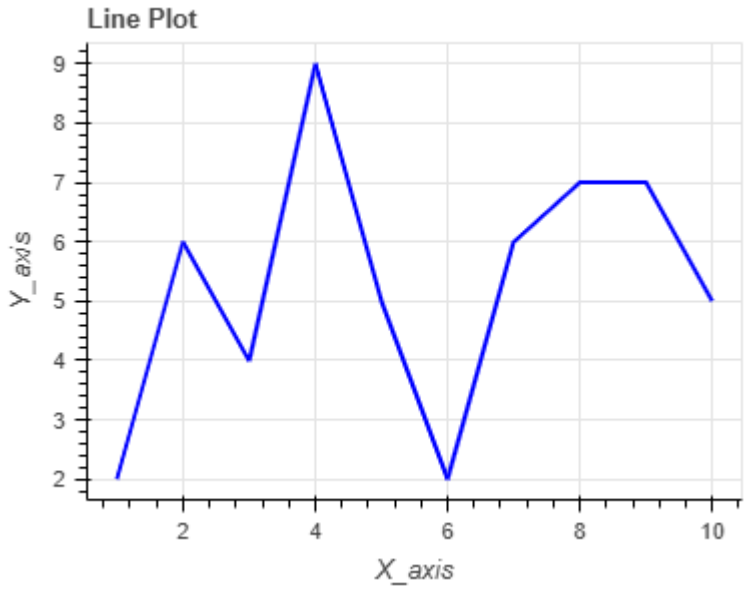


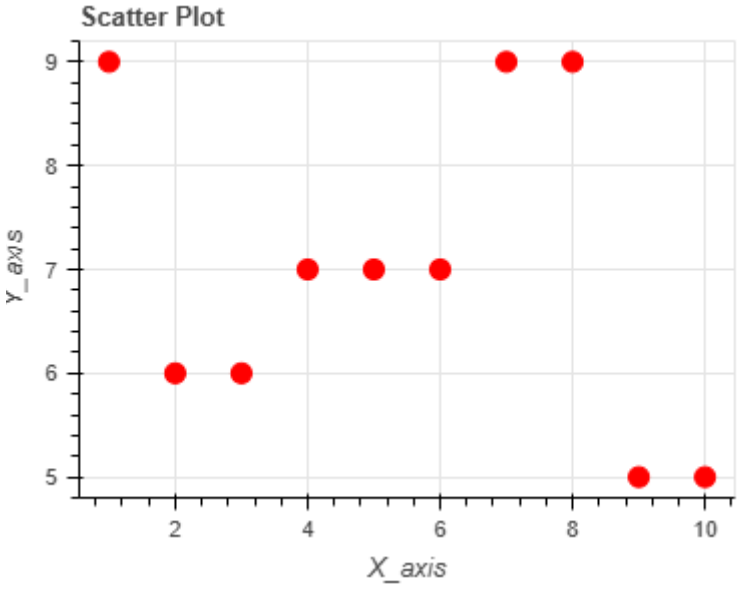
8.b Write a Python program for plotting different types of plots using Bokeh

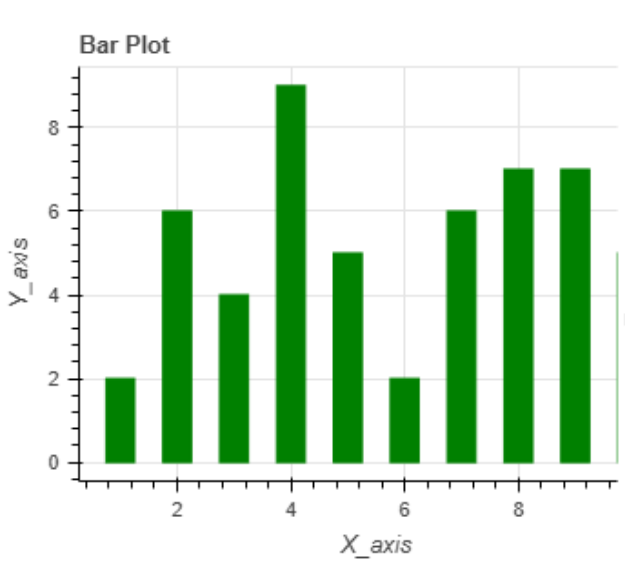
```
from bokeh.plotting import figure, output_file, show

from bokeh.layouts import layout

import random

import numpy as np


x = list(range(1, 11))

y1 = [random.randint(1, 10) for _ in x]   # fixed for loop syntax

y2 = [random.randint(1, 10) for _ in x]


p1 = figure(title="Line Plot", x_axis_label="X_axis", y_axis_label="Y_axis", width=400, height=300)

p1.line(x, y1, line_width=2, line_color="blue")


p2 = figure(title="Scatter Plot", x_axis_label="X_axis", y_axis_label="Y_axis", width=400, height=300)

p2.scatter(x, y2, size=10, color="red", marker="circle")


p3 = figure(title="Bar Plot", x_axis_label="X_axis", y_axis_label="Y_axis", width=400, height=300)

p3.vbar(x=x, top=y1, width=0.5, color="green")


# Create data for histogram plot


p4 = figure(title="Histogram", x_axis_label="Value", y_axis_label="Frequency", width=400, height=300)

hist, edges = np.histogram(y1, bins=5)

p4.quad(top=hist, bottom=0, left=edges[:-1], right=edges[1:], fill_color="purple", line_color="black")


# Use a different variable name to avoid shadowing

layout = layout([
```

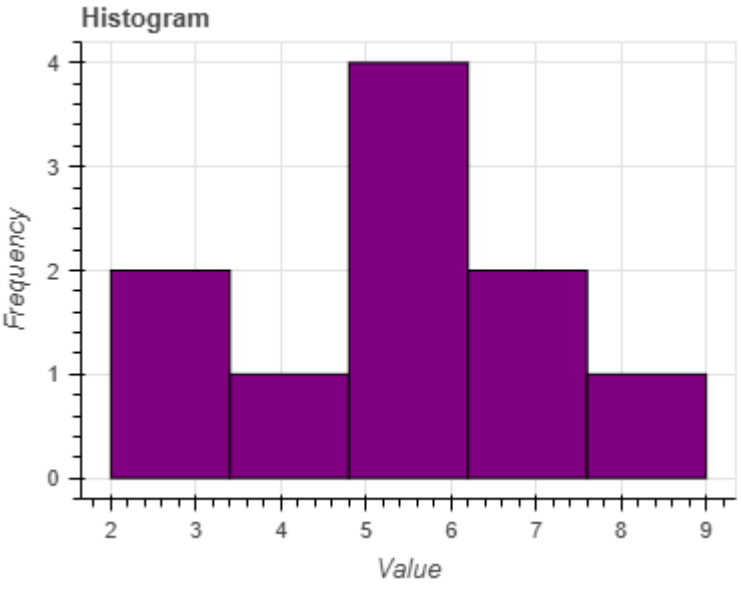
```
    [p1, p2],
    [p3, p4]
])

output_file("bokeh_plots.html")
```

Output:-









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

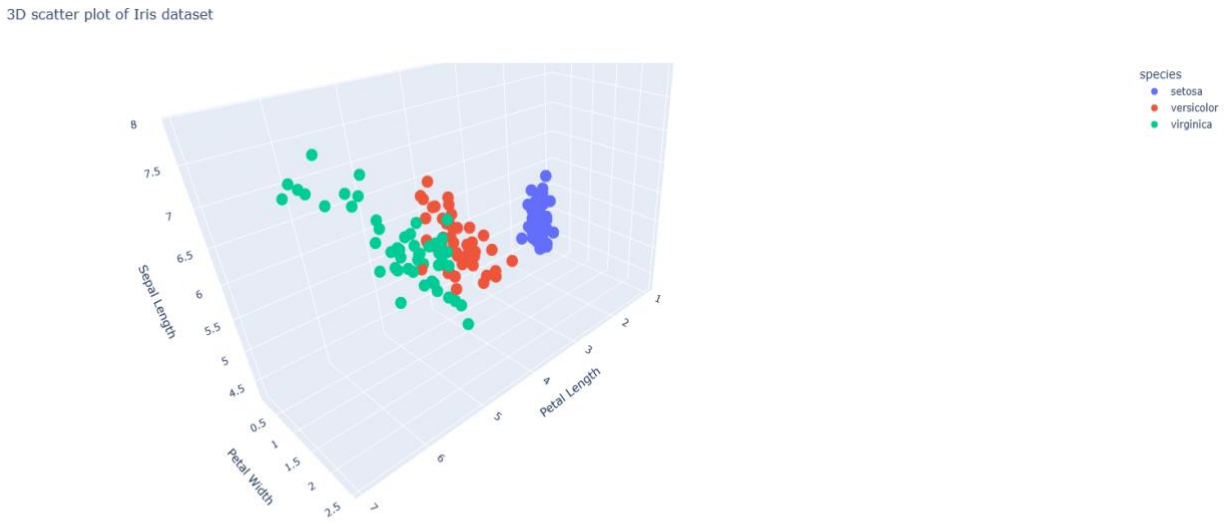
```
import plotly.express as px

df = px.data.iris()

fig = px.scatter_3d(
    df,
    x="petal_length",
    y="petal_width",
    z="sepal_length",
    color="species",
    title="3D scatter plot of Iris dataset"
)

fig.update_layout(
    scene=dict(
        xaxis_title="Petal Length",
        yaxis_title="Petal Width",
        zaxis_title="Sepal Length"
    )
```

output

9b.

```
import plotly.express as px

df = px.data.iris()

fig = px.scatter_3d(
    df,
    x="petal_length",
    y="petal_width",
    z="sepal_length",
    color="species",
    title="3D scatter plot of Iris dataset"
)

fig.update_layout(
    scene=dict(
        xaxis_title="Petal Length",
        yaxis_title="Petal Width",
        zaxis_title="Sepal Length"
    )
)

fig.show()
```

OUTPUT



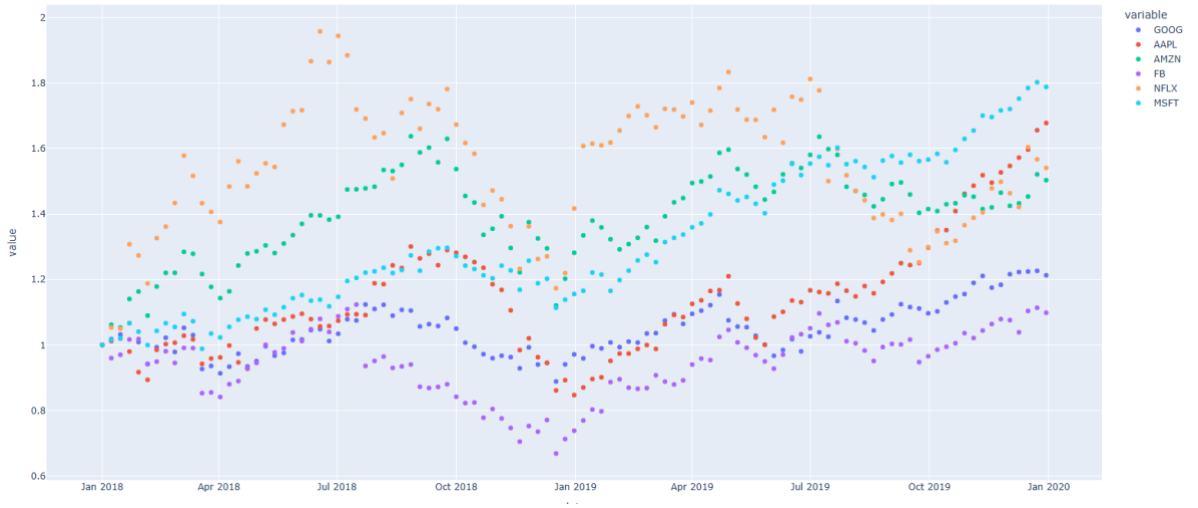
10a. Write a Python program to draw Time Series using Plotly Libraries.

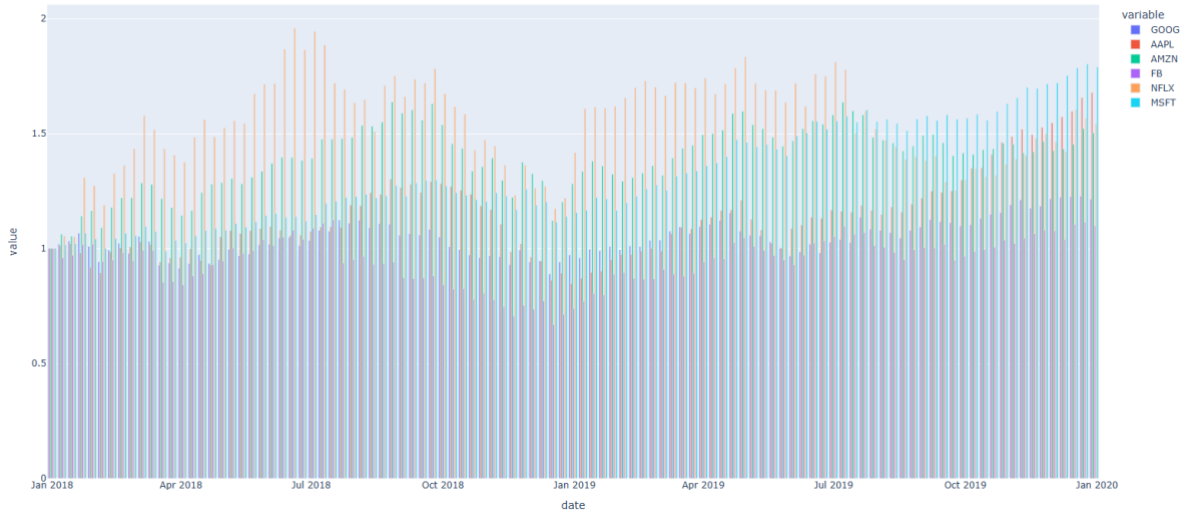
```
import plotly.express as px
import pandas as pd
import plotly.io as pio

# Force Plotly to open charts in the default web browser
pio.renderers.default = 'browser'

# Load sample stock data
df = px.data.stocks()
print(df.head(5))
```

```
# Line plot
fig = px.line(
    df,
    x="date",
    y=df.columns[1:],  # Exclude 'date' from y-values
    hover_data={"date": "|%B %d, %Y"},  # Format hover date
    title='Custom Tick Labels'
)
fig.update_xaxes(
    dtick="M1",  # Monthly ticks
    tickformat="%b\n%Y"  # Show month and year
)
fig.show()

# Scatter plot
fig = px.scatter(df, x="date", y=df.columns[1:])  # Exclude 'date'
fig.show()

# Bar plot
fig = px.bar(df, x="date", y=df.columns[1:], barmode='group')  # Exclude 'date'
```

fig.show()


variable
GOOG
AAPL
AMZN
FB
NFLX
MSFT
variable=NFLX
date=December 31, 2018
value=1.417067
value
date
Feb 2018
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec
Jan 2019
Feb
Mar
Apr
May
Jun
Jul
Aug
Sep
Oct
Nov
Dec


variable
GOOG
AAPL
AMZN
FB
NFLX
MSFT
value
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Jan 2019
Apr 2019
Jul 2019
Oct 2019
Jan 2020


variable
GOOG
AAPL
AMZN
FB
NFLX
MSFT
value
date
Jan 2018
Apr 2018
Jul 2018
Oct 2018
Jan 2019
Apr 2019
Jul 2019
Oct 2019
Jan 2020

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

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

# Import data from USGS
url = ("https://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_month.csv")
data = pd.read_csv(url)

# Drop rows with missing or invalid values in the 'mag' column
data = data.dropna(subset=['mag'])
data = data[data.mag >= 0]

# Create scatter map
fig = px.scatter_geo(
    data,
    lat='latitude',
    lon='longitude',
    color='mag',
    hover_name='place',
    title='Earthquakes Around the World (Past 30 Days)'
)

fig.show()
```

Earthquakes Around the World (Past 30 Days)