Assignment

Q1: What is Matplotlib? Why is it used? Name five plots that can be plotted using the Pyplot module of
Matplotlib.

Ans:Matplotlib is a comprehensive 2D plotting library for Python. It is widely used for creating static, animated, and interactive visualizations in Python. Matplotlib is designed to work seamlessly with NumPy, and it integrates well with other libraries and frameworks.

Why Matplotlib is used:

- Data Visualization: Matplotlib is primarily used for creating static, animated, and interactive visualizations of data. It provides a wide range of options for customizing plots and charts.
- Publication-Quality Graphics: Matplotlib is capable of producing high-quality plots suitable for publication. It allows users to control every aspect of a plot, ensuring that the final result meets professional standards.
- Compatibility: Matplotlib can be used in various contexts, including web applications, GUI toolkits, and Jupyter notebooks. It is widely used in scientific computing, data analysis, machine learning, and more.

Five plots that can be plotted using the Pyplot module of Matplotlib:

Line Plot:
- `plt.plot()`: Used for creating line plots, where data points are connected by straight lines.

Scatter Plot:
- `plt.scatter()`: Useful for visualizing the relationship between two numerical variables. Each data point is represented as a marker on the plot.

Histogram:
- `plt.hist()`: Displays the distribution of a single variable by dividing the data into bins and counting the number of occurrences in each bin.

Bar Plot:
- `plt.bar()`: Used to represent categorical data with rectangular bars. The length of each bar corresponds to the value of a specific category.

Pie Chart:
- `plt.pie()`: Represents data in a circular statistical graphic. Each wedge of the pie represents a proportion of the whole.

These are just a few examples, and Matplotlib provides a wide range of functions for creating various types of plots to suit different visualization needs.

Q2: What is a scatter plot? Use the following code to generate data for x and y. Using this generated data
plot a scatter plot.
import numpy as np
np.random.seed(3)
x = 3 + np.random.normal(0, 2, 50)
y = 3 + np.random.normal(0, 2, len(x))
Note: Also add title, xlabel, and ylabel to the plot.
Ans:A scatter plot is a type of plot that displays individual data points on a two-dimensional graph. Each point represents the values of two variables, and the position of the point on the graph corresponds to its values on the two axes. Scatter plots are useful for visualizing the relationship between two variables and identifying patterns or trends.

Now, let's use the provided code to generate data for `x` and `y` and create a scatter plot using

Matplotlib:

python

Copy code

```
import       as
import                  as


              3


    3                   0  2  50
    3                   0  2  len


                       'Data Points'


          'Scatter Plot of x and y'
           'X-axis Label'
           'Y-axis Label'
```

In this code:

- `plt.scatter(x, y, label='Data Points')` creates a scatter plot with `x` values on the x-axis and `y` values on the y-axis.
- `plt.title('Scatter Plot of x and y')` adds a title to the plot.
- `plt.xlabel('X-axis Label')` and `plt.ylabel('Y-axis Label')` add labels to the x-axis and y-axis, respectively.
- `plt.legend()` shows the legend with the label 'Data Points'.
- `plt.show()` displays the plot.

You can run this code in a Python environment that supports Matplotlib to visualize the scatter plot.

Q3: Why is the subplot() function used? Draw four line plots using the subplot() function.
Use the following data:
import numpy as np
For line 1: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([0, 100, 200, 300, 400, 500])
For line 2: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([50, 20, 40, 20, 60, 70])
For line 3: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([10, 20, 30, 40, 50, 60])
For line 4: x = np.array([0, 1, 2, 3, 4, 5]) and y = np.array([200, 350, 250, 550, 450, 150])
Ans:The `subplot()` function in Matplotlib is used to create multiple plots within the same figure. It is particularly useful when you want to compare different plots side by side. The `subplot()` function takes three arguments: the number of rows, the number of columns, and the index of the current subplot.

Now, let's use the `subplot()` function to draw four line plots with the provided data:

python

Copy code

```
import       as
import                  as


             0  1  2  3  4  5

              0  100  200  300  400  500
              50  20  40  20  60  70
              10  20  30  40  50  60
              200  350  250  550  450  150


            2  2  1
```

```
'Line Plot 1'

  2  2  2

'Line Plot 2'

  2  2  3

'Line Plot 3'

  2  2  4

'Line Plot 4'
```

In this code:

- `plt.subplot(2, 2, 1)` creates a 2x2 grid of subplots and selects the first subplot.
- `plt.plot(x, y1)` plots the first line on the first subplot.
- Similar steps are repeated for the other three lines and subplots.

The `plt.tight_layout()` function is used to automatically adjust subplot parameters for better spacing. Finally, `plt.show()` is used to display the plots.

Q4: What is a bar plot? Why is it used? Using the following data plot a bar plot and a horizontal bar plot.
import numpy as np
company = np.array(["Apple", "Microsoft", "Google", "AMD"])
profit = np.array([3000, 8000, 1000, 10000])
Ans:A bar plot is a chart that presents categorical data with rectangular bars. The lengths of the bars are proportional to the values they represent. Bar plots are useful for comparing the values of different categories and identifying patterns or trends in the data.

Here's how you can create a bar plot and a horizontal bar plot using the provided data:

python

Copy code

```
import       as
import                   as


                  "Apple"  "Microsoft"  "Google"  "AMD"
                 3000  8000  1000  10000


                  8  5
                            'skyblue'
          'Bar Plot of Company Profits'
           'Company'
           'Profit (in billions)'



                  8  5
                             'salmon'
          'Horizontal Bar Plot of Company Profits'
           'Profit (in billions)'
           'Company'
```

In the code above:

- `plt.bar(company, profit, color='skyblue')` creates a vertical bar plot with company names on the x-axis and profit values on the y-axis.
- `plt.barh(company, profit, color='salmon')` creates a horizontal bar plot with company names on the y-axis and profit values on the x-axis.

The `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` functions are used to add a title and labels to the plot for better interpretation. The `plt.show()` function is used to display the plots.

Q5: What is a box plot? Why is it used? Using the following data plot a box plot.
box1 = np.random.normal(100, 10, 200)
box2 = np.random.normal(90, 20, 200)

Ans:A box plot, also known as a box-and-whisker plot, is a statistical representation of the distribution of a dataset. It displays the minimum, first quartile, median, third

quartile, and maximum values. Box plots are used to visualize the spread and skewness of the data, identify outliers, and compare the distributions of different datasets.

Here's how you can create a box plot using the provided data:

python

Copy code

```
import       as
import                    as


                      100  10  200
                      90  20  200




                       'Box 1'  'Box 2'
          'Box Plot of Box1 and Box2'
           'Box'
           'Values'
```

In the code above:

- `plt.boxplot(data, labels=['Box 1', 'Box 2'])` creates a box plot with two boxes (Box 1 and Box 2) representing the two datasets.
- `plt.title()`, `plt.xlabel()`, and `plt.ylabel()` are used to add a title and labels for better interpretation.
- `plt.show()` displays the box plot.

The box in the box plot represents the interquartile range (IQR) of the data, and the whiskers extend to show the range of the data. Outliers, if present, may be represented as individual points outside the whiskers. Box plots are useful for summarizing the distribution of the data and identifying patterns and variations.