# Matplotlib

**Assignment**

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

Matplotlib is a popular Python library used for creating data visualizations. It provides a comprehensive collection of tools for producing a wide range of high-quality 2D and limited 3D plots. Matplotlib is widely used in various fields, including data analysis, scientific research, engineering, and more.

Here are five plots that can be created using the Pyplot module of Matplotlib:

1. Line Plot: A line plot is a basic plot that displays data points connected by straight lines. It is commonly used to visualize trends or patterns over time or across different variables.
2. Bar Plot: A bar plot represents data using rectangular bars, where the height of each bar corresponds to the magnitude of the data. It is effective for comparing categories or displaying discrete data.
3. Scatter Plot: A scatter plot displays individual data points as markers on a two-dimensional coordinate system. It is useful for showing the relationship between two variables and identifying any patterns or clusters.
4. Histogram: A histogram is a graphical representation of the distribution of numerical data. It divides the data into bins and displays the frequency or density of data points falling into each bin. Histograms are helpful in understanding the underlying distribution of a dataset.
5. Pie Chart: A pie chart presents data as a circular graph, divided into slices that represent different categories or proportions. It is commonly used to show the composition or percentage distribution of categorical data.

These are just a few examples of the plots that can be created with Matplotlib's Pyplot module. Matplotlib offers a wide range of customization options and additional plot types to cater to various 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.

A scatter plot is a type of graph that displays the relationship between two variables. It is used to examine the correlation or distribution of data points and identify any patterns or trends. Each data point is represented by a marker on the plot, with the position of the marker indicating the values of the variables being compared.

To generate the data and create a scatter plot using the provided code, you can use the matplotlib library in Python. Here's an example of how you can do it:

import numpy as np

import matplotlib.pyplot as plt

np.random.seed(3)

x = 3 + np.random.normal(0, 2, 50)

y = 3 + np.random.normal(0, 2, len(x))

# Create the scatter plot

plt.scatter(x, y)

# Add title, xlabel, and ylabel

plt.title("Scatter Plot")

plt.xlabel("X")

plt.ylabel("Y")

# Display the plot

plt.show()

# 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])

The subplot() function is used in plotting libraries like Matplotlib to create a grid of subplots within a single figure. It allows you to display multiple plots in a single figure, arranged in rows and columns. This is particularly useful when you want to compare or visualize different data or aspects of a dataset side by side.

Each subplot is identified by its position in the grid, which is specified using the row, column, and index values. The subplot() function takes three arguments: the number of rows, the number of columns, and the index of the current subplot.

Now, let's draw the four line plots using the subplot() function with the given data:

import numpy as np

import matplotlib.pyplot as plt

# Data for line 1

x1 = np.array([0, 1, 2, 3, 4, 5])

y1 = np.array([0, 100, 200, 300, 400, 500])

# Data for line 2

x2 = np.array([0, 1, 2, 3, 4, 5])

y2 = np.array([50, 20, 40, 20, 60, 70])

# Data for line 3

x3 = np.array([0, 1, 2, 3, 4, 5])

y3 = np.array([10, 20, 30, 40, 50, 60])

# Data for line 4

x4 = np.array([0, 1, 2, 3, 4, 5])

y4 = np.array([200, 350, 250, 550, 450, 150])

# Create a 2x2 grid of subplots

plt.subplot(2, 2, 1)

plt.plot(x1, y1)

plt.title('Line 1')

plt.subplot(2, 2, 2)

plt.plot(x2, y2)

plt.title('Line 2')

plt.subplot(2, 2, 3)

plt.plot(x3, y3)

plt.title('Line 3')

plt.subplot(2, 2, 4)

plt.plot(x4, y4)

plt.title('Line 4')

# Adjust the spacing between subplots

plt.tight\_layout()

# Display the figure

plt.show()

# 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])

A bar plot, also known as a bar chart or bar graph, is a graphical representation of categorical data using rectangular bars. It displays the values of different categories as bars, with the length or height of each bar representing the magnitude or quantity of the data.

Bar plots are commonly used to:

1. Compare and display the values of different categories or groups.
2. Show the distribution or frequency of categorical data.
3. Visualize trends or changes over time.
4. Highlight the relationship between different variables.

Now, let's create a bar plot and a horizontal bar plot using the provided data:

import matplotlib.pyplot as plt

company = np.array(["Apple", "Microsoft", "Google", "AMD"])

profit = np.array([3000, 8000, 1000, 10000])

# Bar plot

plt.bar(company, profit)

plt.xlabel("Company")

plt.ylabel("Profit")

plt.title("Profit of Companies")

plt.show()

# Horizontal bar plot

plt.barh(company, profit)

plt.xlabel("Profit")

plt.ylabel("Company")

plt.title("Profit of Companies")

plt.show()

The first plot is a bar plot where the x-axis represents the company names and the y-axis represents the profit values. Each company is represented by a rectangular bar showing its corresponding profit.

The second plot is a horizontal bar plot where the y-axis represents the company names, and the x-axis represents the profit values. The bars are now displayed horizontally.

Both plots provide a visual comparison of the profit values for different companies, allowing you to analyze and interpret the data more easily.

# 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)

A box plot, also known as a box-and-whisker plot, is a graphical representation of the distribution of a dataset. It provides a summary of the data's key statistical properties, including the minimum, maximum, median, and quartiles.

The box plot consists of several components:

1. Median: It is represented by a horizontal line inside a box and represents the middle value of the dataset when arranged in ascending order.
2. Quartiles: The dataset is divided into four equal parts called quartiles. The first quartile (Q1) represents the 25th percentile, the third quartile (Q3) represents the 75th percentile, and the second quartile represents the median.
3. Interquartile Range (IQR): It is the range between the first and third quartiles (Q1 and Q3) and provides information about the spread of the middle 50% of the data. It is shown as the length of the box.
4. Whiskers: They extend from the box and represent the minimum and maximum values within a certain range. Typically, they extend up to 1.5 times the IQR from the first and third quartiles. Data points lying outside this range are considered outliers and are represented as individual points or asterisks.
5. Outliers: Data points that fall outside the whiskers are considered outliers and are plotted as individual points or asterisks.

Box plots are used to compare distributions between different groups or variables. They provide a visual summary of the range, central tendency, and spread of the data, allowing for quick comparisons and identification of outliers.

Now, let's plot a box plot for the given data using Python:

import numpy as np

import matplotlib.pyplot as plt

box1 = np.random.normal(100, 10, 200)

box2 = np.random.normal(90, 20, 200)

data = [box1, box2]

fig, ax = plt.subplots()

ax.boxplot(data)

# Adding labels to the plot

ax.set\_xticklabels(['Box 1', 'Box 2'])

ax.set\_ylabel('Values')

plt.title('Box Plot')

plt.show()