Day 15 – Assignment Questions

1. What is Matplotlib? Explain about it.

Matplotlib is a widely used Python library for data visualization. It helps in creating static, animated, and interactive plots.
It is mainly used in data science, machine learning, and statistical analysis to visually represent data.
With Matplotlib, we can create line charts, bar charts, histograms, pie charts, scatter plots and many more.
It provides high customization options such as changing colors, line styles, markers, labels, and grid settings.
Because of its flexibility and simplicity, it is one of the most important visualization libraries in Python.

2. Difference between plot() and scatter()

plot() function is generally used to create line graphs. It connects the data points with straight lines.
It is mostly used when we want to show trends or changes over time.

scatter() function is used to create scatter plots. It displays individual data points without connecting them with lines.
It is mainly useful to understand the relationship or correlation between two variables.

3. What is pyplot? Explain about it.

Pyplot is a module in Matplotlib that provides a simple interface for plotting.
It contains commonly used functions such as plot(), scatter(), title(), xlabel(), ylabel(), legend(), and show().
Pyplot works in a state-based manner, meaning it automatically keeps track of the current figure and axes.
It is very beginner-friendly and similar to MATLAB plotting style.

4. How to add labels and title?

We can add labels and title to make the plot more understandable.
The functions used are:

```
plt.xlabel("X-axis label")
plt.ylabel("Y-axis label")
plt.title("Title of the plot")
```

Labels help identify what each axis represents, and the title describes the overall graph.

5. Difference between Figure and Axes.

Figure is the complete canvas or outer container of the plot.
It can contain one or more axes.

Axes is the region inside the figure where the actual data is plotted.
Each subplot has its own axes.
So, figure is the overall structure, and axes is where data visualization happens.

6. How to create multiple subplots?

Multiple subplots can be created using:

```
plt.subplot(rows, columns, position)
```

or

```
plt.subplots()
```

This allows us to display more than one graph in a single figure, which is useful for comparison and analysis.

7. What is the use of marker? What are the different markers?

Markers are used to highlight data points in a graph.
They make it easier to identify individual values.

Some common markers are:
'o' → circle
's' → square
'^' → triangle
'*' → star
'x' → cross

'+' → plus
'.' → point

Markers improve the readability of the plot, especially when data points are limited.

8. How to customize colors?

Colors can be customized using the color parameter:

plt.plot(x, y, color='red')

We can use:
- Color names like 'red', 'blue'
- Short forms like 'r', 'g', 'b'
- Hex codes like '#FF5733'

Color customization improves visual clarity and presentation quality.

9. How to handle large amount of data while visualization?

When dealing with large datasets:
- Use sampling to reduce data points.
- Reduce marker size.
- Use alpha (transparency).
- Avoid unnecessary grid or labels.
- Use efficient plotting methods.

These techniques help improve performance and readability.

10. Which plot is used for showing trends in the data?

Line plot is best suited for showing trends over time or sequential data.
It clearly shows upward or downward movement.

11. Which plot is used for data distributions?

Histogram is commonly used to show data distribution.
It divides data into bins and shows frequency of values in each bin.

12. How to display grid?

Grid lines can be displayed using:

plt.grid(True)

Grid improves readability and helps compare values easily.

13. How to change line color?

Line color can be changed by using:

plt.plot(x, y, color='blue')

This helps in distinguishing multiple lines in the same graph.

14. What are the different styles available in line?

Different line styles include:

'-' → solid line
'--' → dashed line
':' → dotted line
'-.' → dash-dot line

These styles can be applied using the linestyle parameter.
Line styles help differentiate between multiple data series in the same plot.