Q1. How can you create a Bokeh plot using Python code?

1.Installing Bokeh in Python using pip

2.Importing Bokeh in Python

3.Import sample data for bokeh

4.Check Bokeh version in Python

Q2. What are glyphs in Bokeh, and how can you add them to a Bokeh plot? Explain with an example.

A glyph is a vectorized graphical shape or marker that is used to represent your data. A Few Categories of Glyphs: Marker: Shapes like circles, diamonds, squares and triangles. Effective for scatter and bubble charts. Line: Single, step and multi-line shapes.You can display images on Bokeh plots using the image() , image_rgba() , and image_url() glyph methods. You can use hovering tooltips with image glyphs to let the user see the values of each pixel.

In [ ]:

```
from bokeh.plotting import figure, output_file, show
output_file("line.html")
p = figure(width=400, height=400)
p.circle([1, 2, 3, 4, 5], [6, 7, 2, 4, 5], size=20, color="navy", alpha=0.5)
show(p)
```

Q3. How can you customize the appearance of a Bokeh plot, including the axes, title, and legend?

1.setting labels for your axes 2.styling the numbers displayed with your axes 3.defining colors and other layout properties for the axes themselves

we can customize the plot by adding title to the plot and labelling the xaxis and yaxis.

Q4. What is a Bokeh server, and how can you use it to create interactive plots that can be updated in real time?

Bokeh server makes it easy to create interactive web applications that connect front-end UI events to running Python code. Bokeh creates high-level Python models, such as plots, ranges, axes, and glyphs, and then converts these objects to JSON to pass them to its client library, BokehJS.

Q5. How can you embed a Bokeh plot into a web page or dashboard using Flask or Django?

You can save the returned HTML text to a file using standard Python file operations. You can also provide your own template for the HTML output and pass in custom, or additional, template variables. For more details, see the file_html() documentation.

Bokeh can also supply JSON data that BokehJS can use to render a standalone Bokeh document in a specified . The json_item() function accepts a Bokeh model (for example, a plot) and an optional ID of the target.

You can also have Bokeh return individual components of a standalone document to embed them one by one with the components() function. This function returns a that contains the data for your plot and provides a target to display the plot view. You can use these elements in HTML documents however you like.

You can also embed standalone documents with the autoload_static() function. This function provides a tag that replaces itself with a Bokeh plot. This script also checks for BokehJS and loads it if necessary. This function lets you embed a plot with nothing but this tag.