Double-click (or enter) to edit

05 MARCH ASSIGNMENT

1.

To create a Bokeh plot using Python code, you need to follow these general steps:

1. Install the Bokeh library using pip or conda: pip install bokeh or conda install bokeh.
2. Import the necessary functions from Bokeh: from bokeh.plotting import figure, output_file, show.
3. Create a figure using the figure() function. This is where you specify the title, axis labels, and other plot properties.
4. Add glyphs (e.g., lines, circles, bars) to the figure using the corresponding functions (line(), circle(), vbar(), etc.). Specify the data source and any necessary parameters for each glyph.
5. Configure the output using the output_file() function (to save to an HTML file) or output_notebook() function (to display in a Jupyter notebook).
6. Display the plot using the show() function.

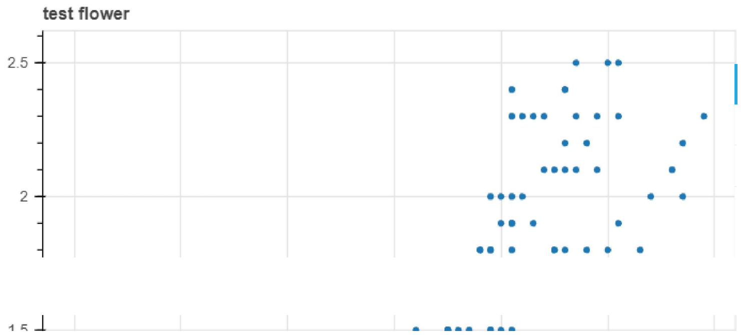
```
pip install bokeh
```

```
    Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-wheels/public/simple/
    Requirement already satisfied: bokeh in /usr/local/lib/python3.9/dist-packages (2.4.3)
    Requirement already satisfied: tornado>=5.1 in /usr/local/lib/python3.9/dist-packages (from bokeh) (6.2)
    Requirement already satisfied: numpy>=1.11.3 in /usr/local/lib/python3.9/dist-packages (from bokeh) (1.22.4)
    Requirement already satisfied: PyYAML>=3.10 in /usr/local/lib/python3.9/dist-packages (from bokeh) (6.0)
    Requirement already satisfied: Jinja2>=2.9 in /usr/local/lib/python3.9/dist-packages (from bokeh) (3.1.2)
    Requirement already satisfied: pillow>=7.1.0 in /usr/local/lib/python3.9/dist-packages (from bokeh) (8.4.0)
    Requirement already satisfied: typing-extensions>=3.10.0 in /usr/local/lib/python3.9/dist-packages (from bokeh) (4.5.0)
    Requirement already satisfied: packaging>=16.8 in /usr/local/lib/python3.9/dist-packages (from bokeh) (23.0)
    Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.9/dist-packages (from Jinja2>=2.9->bokeh) (2.1.2)
```

```
import bokeh.io
import bokeh.plotting
bokeh.io.output_notebook()
```

```
from bokeh.plotting import figure, output_file, show
from bokeh.sampledata.iris import flowers
```

```
output_file('test.html')
p = figure(title = "test flower")
p.xaxis.axis_label = "patel length"
p.yaxis.axis_label = "patel width"
p.circle(flowers['petal_length'], flowers['petal_width'])
show(p)
```

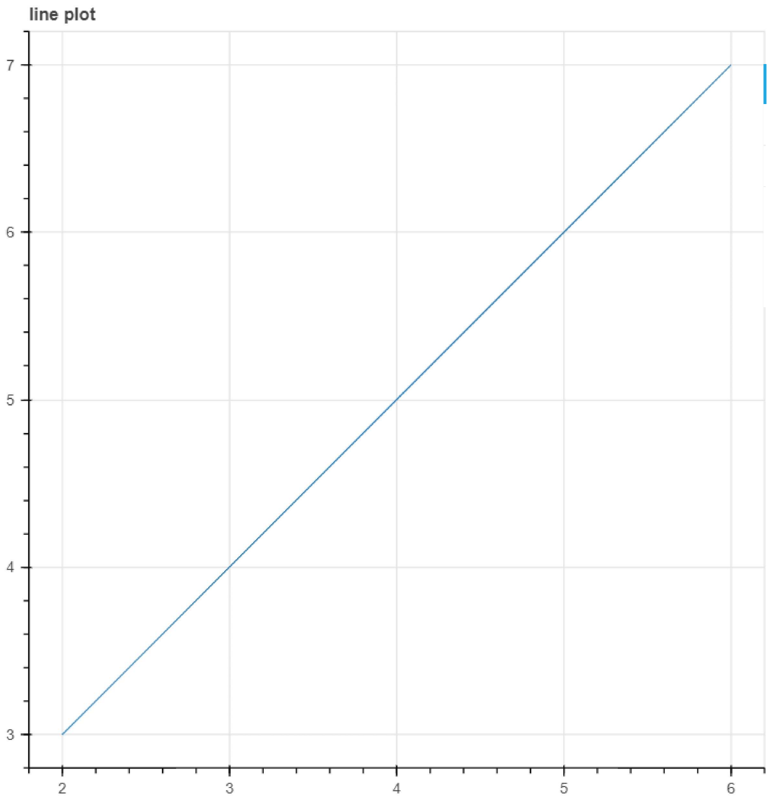


2.

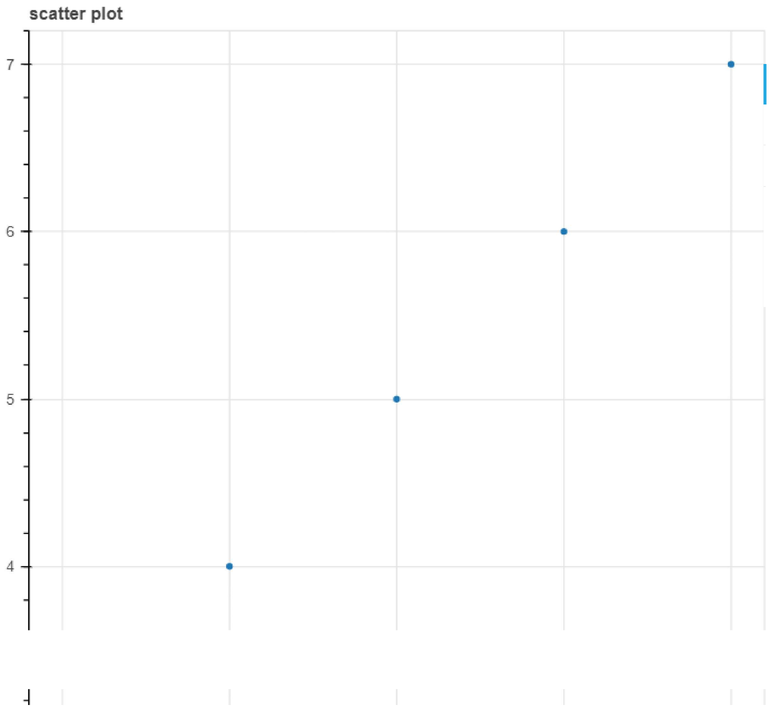
Glyphs in Bokeh refer to the graphical objects that can be added to a plot, such as lines, markers, patches, and more. Bokeh provides a variety of glyph types that can be used to visualize data in different ways.

To add glyphs to a Bokeh plot, you first create a data source that contains the data you want to visualize, and then create a glyph object with the appropriate properties, such as color, size, and shape. You then add the glyph to the plot by calling the add_glyph method of the plot object and passing in the data source and glyph object.

```
x = [2,3,4,5,6]
y = [3,4,5,6,7]
output_file("line.html")
p = figure(title = "line plot")
p.line(x,y)
show(p)
```



```
x = [2,3,4,5,6]
y = [3,4,5,6,7]
output_file("scatter.html")
p = figure(title = "scatter plot")
p.scatter(x,y)
show(p)
```



4.

A Bokeh server is a Python process that allows you to create and serve interactive Bokeh plots that can be updated in real time. The Bokeh server works by running a web application that serves Bokeh plots and receives events from the user's web browser. When the user interacts with a Bokeh plot (e.g., by zooming, panning, or clicking), the web browser sends an event to the Bokeh server, which can then update the plot with new data or modify its properties.

To use the Bokeh server, you need to create a Python script that defines the layout and behavior of the Bokeh plot. This script should contain a curdoc() function that returns a Document object, which represents the Bokeh plot and its associated data sources, glyphs, and tools. You can then run the Bokeh server by running the bokeh serve command with the name of the script as an argument.

5.

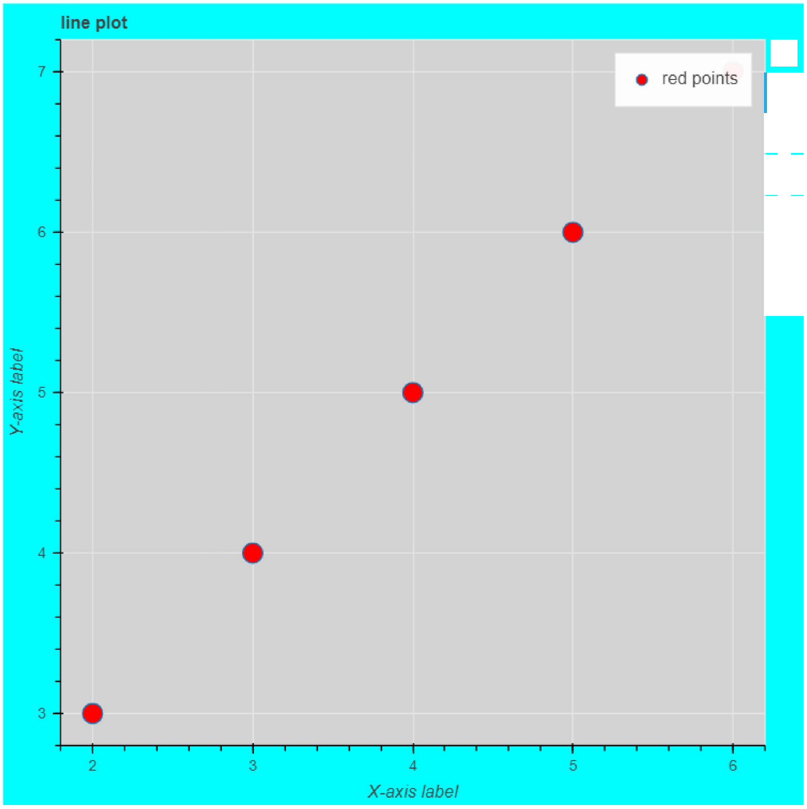
To embed a Bokeh plot into a web page or dashboard using Flask or Django, you can use the Bokeh components function to generate the HTML and JavaScript code necessary to display the plot in a web page, and then pass this code to your Flask or Django template.

3.

Bokeh provides a range of options for customizing the appearance of a plot, including the axes, title, legend, and more. Here are some of the ways you can customize a Bokeh plot:

1. Axis labels
2. Plot Title
3. Legend
4. Background and Border

```
x = [2,3,4,5,6]
y = [3,4,5,6,7]
output_file("line.html")
p = figure(title = "line plot", x_axis_label='X-axis label', y_axis_label='Y-axis label' )
p.scatter(x,y, fill_color = 'red', legend_label = 'red points', size = 15)
p.background_fill_color = 'lightgray'
p.border_fill_color = 'cyan'
show(p)
```



✓ 0s completed at 2:32 PM ● ×