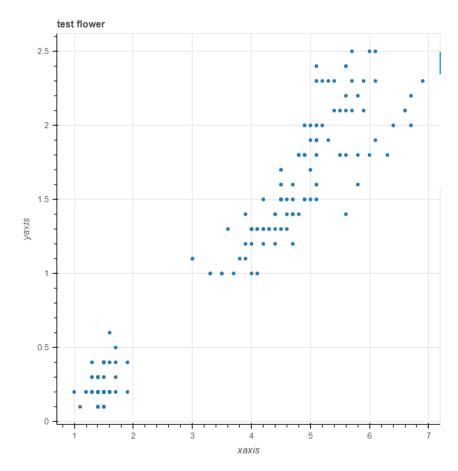
Q1. How can you create a Bokeh plot using Python code?

```
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='xaxis'
p.yaxis.axis_label='yaxis'
p.circle(flowers['petal_length'],flowers['petal_width'])
show(p)
```



Double-click (or enter) to edit

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

```
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='xaxis'
p.yaxis.axis_label='yaxis'

p.circle(x=[1, 2, 3], y=[4, 5, 6], size=10, color="red")
p.square(x=[4, 5, 6], y=[7, 8, 9], size=10, color="blue")

# Display the plot
show(p)
```

```
test flower
```

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

```
x=[1,2,3,4,5,6]
y=[7,8,9,10,11,12]
output_file('legend.html')
p=figure(title='legend_plot')
p.xaxis.axis_label='x-axis'
p.yaxis.axis_label='y_axis'
p.scatter(x,y,fill_color='red',legend='red_point')
show(p)
```

```
BokehDeprecationWarning: 'legend' keyword is deprecated, use explicit 'legend label', 'legend field', c
          legend plot
                 ■ Lea_hour
Q4. What is a Bokeh server, and how can you use it to create interactive plots that can be updated in real time?
from bokeh.plotting import figure
from bokeh.layouts import column
from bokeh.models import ColumnDataSource, Slider
from bokeh.server.server import Server
def modify doc(doc):
   # Create a data source with some sample data
   source = ColumnDataSource(data=dict(x=[1, 2, 3], y=[4, 5, 6]))
   # Create a scatter plot with circles
   p = figure(plot width=400, plot height=400)
   p.circle(x='x', y='y', source=source, size=10, color='blue')
   # Create sliders to control the size and color of the markers
   size slider = Slider(start=0, end=20, value=10, step=1, title='Marker Size')
   color slider = Slider(start=0, end=255, value=0, step=1, title='Marker Color')
   # Define a callback function that updates the plot when the sliders are changed
   def update(attr, old, new):
       source.data['size'] = [size slider.value] * len(source.data['x'])
       source.data['color'] = ['rgb(0, 0, %d)' % color slider.value] * len(source.data['x'])
   # Attach the callback function to the sliders
   size slider.on change('value', update)
   color slider.on change('value', update)
   # Combine the plot and sliders into a layout
   layout = column(p, size_slider, color_slider)
   # Add the layout to the document
   doc.add root(layout)
# Start the Bokeh server and open the plot in a web browser
server = Server({'/': modify_doc})
server.start()
server.io_loop.add_callback(server.show, "/")
server.io_loop.start()
```

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

Os completed at 5:09 PM

×