5th Mar Assignment

March 14, 2023

1 Assignment 29

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

Ans.

[4]:
```
from bokeh.plotting import figure, output_file, show

# Create a blank figure with labels
p = figure(title="My Bokeh Plot", x_axis_label='X Axis', y_axis_label='Y Axis')

# Add a line plot with some data
x = [1, 2, 3, 4, 5]
y = [6, 7, 2, 4, 5]
p.line(x, y, legend_label="Line 1", line_width=2)

# Add a scatter plot with some data
x2 = [3, 8, 5, 1, 10]
y2 = [9, 4, 6, 2, 7]
p.circle(x2, y2, legend_label="Circles", fill_color="white", size=8)

# Set output file and show the plot
output_file("myplot.html")
show(p)
```

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

Ans. In Bokeh, glyphs are visual markers that can be used to represent data points on a plot. Glyphs can be used to create scatter plots, line plots, bar charts, and many other types of visualizations.

[5]:
```
from bokeh.plotting import figure, output_file, show

# Create a blank figure with labels
p = figure(title="My Bokeh Glyphs", x_axis_label='X Axis', y_axis_label='Y␣
 ↪Axis')

# Add a circle glyph with some data
```

```
x = [1, 2, 3, 4, 5]
y = [6, 7, 2, 4, 5]
p.circle(x, y, legend_label="Circles", fill_color="white", size=12)

# Add a line glyph with some data
x2 = [1, 2, 3, 4, 5]
y2 = [2, 5, 4, 8, 6]
p.line(x2, y2, legend_label="Line 1", line_width=2)

# Add a triangle glyph with some data
x3 = [3, 4, 2, 5, 1]
y3 = [9, 7, 8, 6, 10]
p.triangle(x3, y3, legend_label="Triangles", fill_color="blue", size=10)

# Set output file and show the plot
output_file("myglyphs.html")
show(p)
```

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

Ans.Bokeh provides many options for customizing the appearance of a plot, including the axes, title, and legend. Here are some examples of how to customize these elements:

[6]:
```
# Axes
p.xaxis.axis_label = "X Axis Label"
p.xaxis.axis_label_text_color = "blue"
p.xaxis.axis_label_standoff = 20

p.yaxis.axis_label = "Y Axis Label"
p.yaxis.axis_label_text_color = "red"
p.yaxis.axis_label_standoff = 20

p.xaxis.major_label_text_color = "green"
p.yaxis.major_label_text_color = "purple"
p.xaxis.axis_line_color = "gray"
p.yaxis.axis_line_color = "gray"
p.xaxis.major_tick_line_color = "blue"
p.yaxis.major_tick_line_color = "red"
p.xaxis.minor_tick_line_color = "green"
p.yaxis.minor_tick_line_color = "purple"
p.xaxis.major_tick_out = 10
p.yaxis.major_tick_out = 10
```

[7]:
```
#Legend
p.legend.title = "Legend Title"
```

```
p.legend.title_text_color = "green"
p.legend.title_text_font_size = "12pt"
p.legend.label_text_font_size = "10pt"
p.legend.location = "top_left"
p.legend.background_fill_alpha = 0.5
p.legend.border_line_color = "black"
p.legend.border_line_width = 2
```

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

Ans.A Bokeh server is a Python application that allows you to create and serve interactive Bokeh plots that can be updated in real-time based on user input or external data sources. The Bokeh server works by maintaining a WebSocket connection between the client (i.e., web browser) and the server, which allows for bi-directional communication and dynamic updates to the plot.

[8]:
```
from bokeh.plotting import figure
from bokeh.layouts import column
from bokeh.models import ColumnDataSource, Slider
from bokeh.server.server import Server
from bokeh.application import Application
from bokeh.application.handlers.function import FunctionHandler
from numpy.random import random

# Define the layout and behavior of the plot
def make_plot(doc):
    source = ColumnDataSource(data=dict(x=random(100), y=random(100)))
    plot = figure()
    plot.scatter('x', 'y', source=source)

    # Define a callback function to update the plot based on slider input
    def callback(attr, old, new):
        source.data = dict(x=random(new), y=random(new))

    # Create a slider and add the callback function
    slider = Slider(start=10, end=100, value=50, step=10, title="Number of␣
 ↪points")
    slider.on_change('value', callback)

    # Add the plot and slider to a layout
    layout = column(plot, slider)
    doc.add_root(layout)

# Define the Bokeh server application
handler = FunctionHandler(make_plot)
app = Application(handler)
```

```
# Start the Bokeh server and display the plot
server = Server(app)
server.start()
server.io_loop.add_callback(server.show, "/")
server.io_loop.start()
```

```
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
Cell In[8], line 35
     33 server.start()
     34 server.io_loop.add_callback(server.show, "/")
---> 35 server.io_loop.start()

File /opt/conda/lib/python3.10/site-packages/tornado/platform/asyncio.py:199, i
 ↪BaseAsyncIOLoop.start(self)
    197     self._setup_logging()
    198     asyncio.set_event_loop(self.asyncio_loop)
--> 199     self.asyncio_loop.run_forever()
    200 finally:
    201     asyncio.set_event_loop(old_loop)

File /opt/conda/lib/python3.10/asyncio/base_events.py:592, in BaseEventLoop.
 ↪run_forever(self)
    590 """Run until stop() is called."""
    591 self._check_closed()
--> 592 self._check_running()
    593 self._set_coroutine_origin_tracking(self._debug)
    595 old_agen_hooks = sys.get_asyncgen_hooks()

File /opt/conda/lib/python3.10/asyncio/base_events.py:584, in BaseEventLoop.
 ↪_check_running(self)
    582 def _check_running(self):
    583     if self.is_running():
--> 584         raise RuntimeError('This event loop is already running')
    585     if events._get_running_loop() is not None:
    586         raise RuntimeError(
    587             'Cannot run the event loop while another loop is running')

RuntimeError: This event loop is already running
```

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

Ans.

[9]:
```python
from flask import Flask, render_template
from bokeh.plotting import figure
from bokeh.embed import components
from numpy.random import random

app = Flask(__name__)

# Define the route for the web page that will display the plot
@app.route('/')
def index():
    # Create the plot
    plot = figure()
    plot.scatter(random(100), random(100))

    # Generate the HTML and JavaScript code for the plot
    script, div = components(plot)

    # Pass the code to the Flask template
    return render_template('index.html', script=script, div=div)

if __name__ == '__main__':
    app.run(debug=True)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[9], line 1
----> 1 from flask import Flask, render_template
      2 from bokeh.plotting import figure
      3 from bokeh.embed import components

ModuleNotFoundError: No module named 'flask'
```