Interactive Visualization with Bokeh - 1

One should look for what is and not what he thinks should be. (Albert Einstein)

Interactive plots: Topic introduction

In this part of the course, we will cover the following concepts:

- Introduction to `bokeh`
- Organize and visualize data with `bokeh`

Module completion checklist

Objective	Complete
Explain how to create basic visualizations in bokeh	
Generate a figure using Bokeh and add customized glyphs to it	

Warm up

- Do you know Python has over 137000 libraries which can be helpful in creating applications in machine learning, data science, data manipulation, data visualization?
- For creating data visualizations, Python offers a large community and a vast number of built-in modules

- Take 5 minutes to read about ***Top 8 Python Libraries for Data Visualization*** and then share your thoughts on the following questions with the whole group:
  - Do you visualize data regularly at work? If yes, what do you use for creating visualizations?
  - Have you used any of the Python libraries for data visualization? Which one?

Visualizing data with Bokeh

- `bokeh` is an interactive visualization library that targets modern web browsers for presentation

- Bokeh offers two interfaces to users:
  - **`bokeh.models`**: low-level interface with the most flexibility (most users will not use this level of interface to assemble plots directly)
  - **`bokeh.plotting`**: higher-level interface centered around composing visual glyphs

- **Note:** The `bokeh.plotting` interface is handy when we need to customize the output by adding more data series, glyphs, etc.

Plotting with Bokeh

Here are the basic steps for creating plots with the `bokeh.plotting` interface:

- **Prepare data:**
  - Could be NumPy arrays or Pandas series

- Tell Bokeh **where to generate output:**
  - In this case, it's `output_notebook()` for use in Jupyter notebooks

- **Call `figure()`**
  - This creates a plot with default options and easy customization of title, tools, and axes labels

Plotting with Bokeh (cont'd)

- **Add renderers:**
  - Use functions specifying visual customization like colors, legends, and widths

- Ask Bokeh to **`show()` or `save()` the results:**
  - These functions save the plot to an HTML file and optionally display it in a browser

- **Note:** The last two steps can be repeated to create more than one plot

Output methods using Bokeh

Here are some common methods to view Bokeh plots, such as:

- `output_file()`
  - Generates HTML documents for Bokeh visualizations

- `output_notebook()`
  - Displays inline visualizations in Jupyter notebook

Module completion checklist

Objective	Complete
Explain how to create basic visualizations in bokeh	✔
Generate a figure using Bokeh and add customized glyphs to it	

Load the libraries

- Let’s import the libraries we will be using in this module

```
import pandas as pd
import numpy as np
import os
import matplotlib.pyplot as plt

from bokeh.io import output_notebook
from bokeh.plotting import figure, output_file, show, output_notebook, save
from bokeh.transform import factor_cmap, factor_mark
from bokeh.layouts import column, row, gridplot
from bokeh.models import HoverTool, ColumnDataSource, NumeralTickFormatter, GroupFilter, CDSView
import ipywidgets as widgets
from ipywidgets import interact, interact_manual
```

Bokeh: simple plot

- At first, we will create simple plots using data points assigned to variables `x_values` and `y_values`

```
# Input the sample data below.
x_values = [1, 2, 3, 4, 5, 6]
y_values = [6, 7, 2, 3, 6, 4]
```

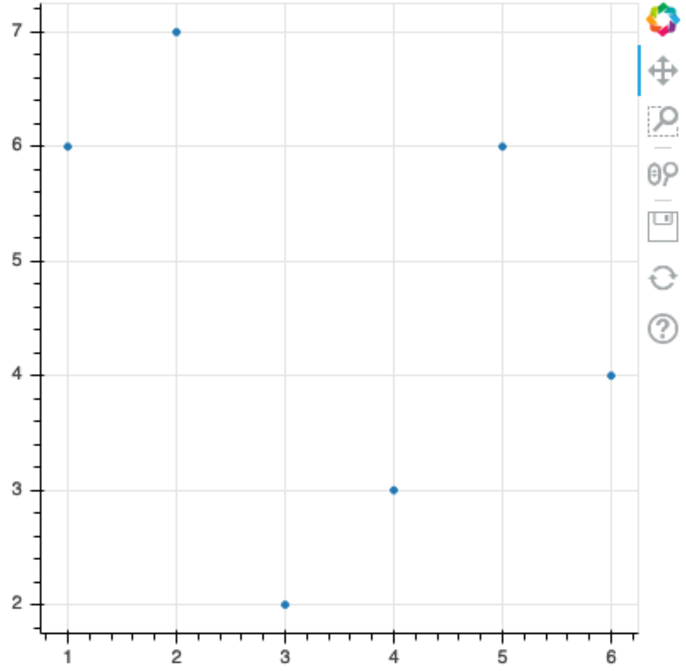
Bokeh: simple plot

Steps to create a simple plot:

1. Make a plot using the `figure()` method
2. Append the glyphs to the plot by calling an appropriate method and passing in data
3. Show the plot

```
# Set the output method
output_notebook()

p = figure()
p.scatter(x = x_values, y = y_values,
marker="circle")
show(p)
```

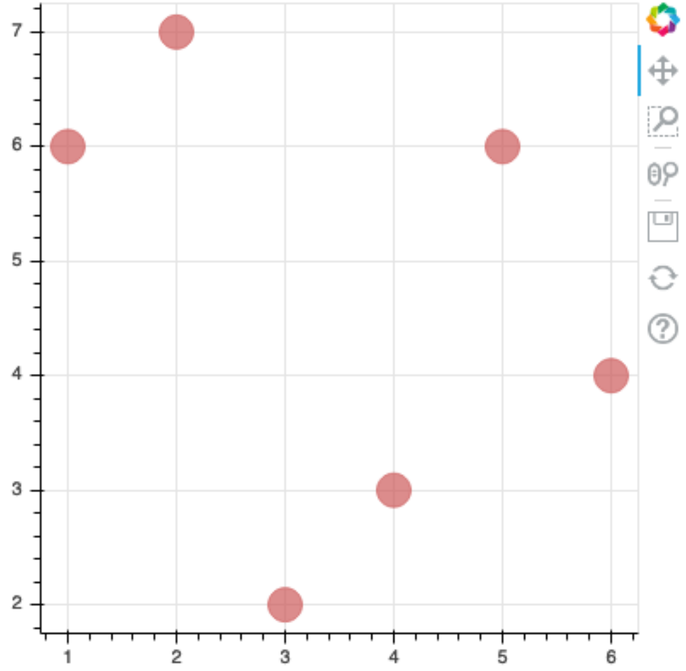


Bokeh: add size, color, and opacity

- We can also modify the same circle glyph by adding functions with a specific size, color, and alpha

```
# Create the blank plot.
p = figure(width = 400, height = 400)

# Add a circle glyph with a size, color, and
alpha.
p.scatter(x_values,
          y_values,
          size = 20,
          color = "red",
          alpha = 0.7,
          marker="circle")
show(p)
```



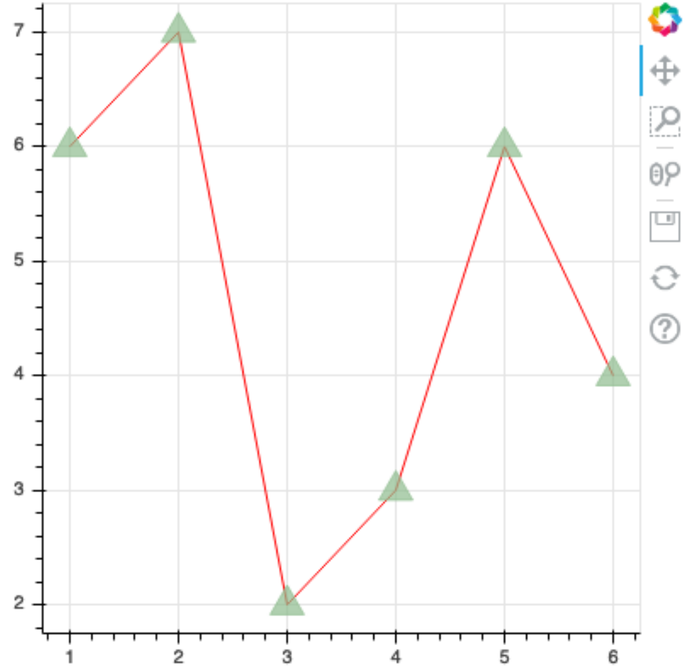
Bokeh: triangle glyph

- Further, we can add two different glyphs to the graph

```
p = figure(width = 400, height = 400)

p.line(x_values,
       y_values,
       color = 'red')

p.scatter(x_values,
          y_values,
          size = 20,
          color = "darkseagreen",
          alpha = 0.7,
          marker="triangle")
show(p)
```



Bokeh: marker types

- A glyph is a vectorized graphical shape or **marker** that is used to represent your data
- There are a lot of marker types you can try out
- ***Click here*** to see examples of plots with different markers

- asterisk()
- circle()
- circle_cross()
- circle_x()
- cross()
- dash()
- diamond()
- diamond_cross()
- inverted_triangle()
- square()
- square_cross()
- square_x()
- triangle()
- x()

Knowledge check

Module completion checklist

Objective	Complete
Explain how to create basic visualizations in bokeh	✔
Generate a figure using Bokeh and add customized glyphs to it	✔

Congratulations on completing this module!

You are now ready to try tasks 1-2 in the Exercise for this topic