

CSH4W3 Data Visualization

Introduction Plotting with Bokeh Python

Irma Palupi

Telkom University

irmapalupi@telkomuniversity.ac.id

KK: Modeling and Computational Experiment (MCE)

Overview

Bokeh Python Intro

Plots with Glyphs

Bokeh-Pandas

Miscellaneous

Bokeh Python

- Bokeh is a data visualization library for Python.
- Unlike Matplotlib and Seaborn, Bokeh renders its plots using HTML and JavaScript. Hence, it proves to be extremely useful for developing web based dashboards.
- Bokeh can easily connect with NumPy, Pandas and more, to produce interactive plots, dashboards and data applications.
- The important feature : Bokeh primarily converts the data source into a JSON file which is used as input for BokehJS, a JavaScript library, which in turn is written in TypeScript and renders the visualizations in modern browsers.

Beneficial Features of Bokeh

- **Flexibility:** useful for common plotting requirements as well as custom and complex usecases.
- **Productivity:** can easily interact with other popular Pydata tools such as Pandas and Jupyter notebook.
- **Interactivity:** Bokeh creates interactive plots that change when the user interacts with them. You can give your audience a wide range of options and tools for inferring and looking at data from various angles so that user can perform “what if” analysis. This is the important advantage over Seaborn and Matplotlib.
- **Powerful:** By adding custom JavaScript, it is possible to generate visualizations for specialised usecases.
- **Sharable:** Plots can be embedded in output of Flask or Django enabled web applications. They can also be rendered in Jupyter notebooks.
- **OpenSource:** Its source code is available on <https://github.com/bokeh/bokeh>

Environment Setup

"pip install bokeh" to automatically install all the following Bokeh requiring packages.

- `jinjia2 >=2.7`
- `numpy >=1.7.1`
- `packaging >=16.8`
- `pillow >=4.0`
- `python-dateutil >=2.1`
- `pyyaml >=3.10`
- `six >=1.5.2`
- `tornado >=4.3`

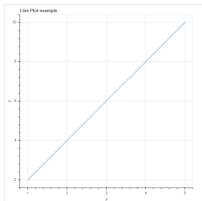
Bokeh Basic Concept

Offering two interfaces using which various plotting operations can be performed.

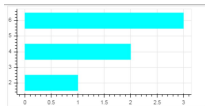
- **bokeh.models:** The low level interface. The low-level objects that comprise a Bokeh scene graph are called Models.
- **bokeh.plotting:** This is a higher level interface that has functionality for composing visual glyphs. Glyphs are the building blocks of Bokeh plot such as lines, circles, rectangles, and other shapes.
- **bokeh.application:** Bokeh package Application class which is a lightweight factory for creating Bokeh Documents. A Document is a container for Bokeh Models to be reflected to the client side BokehJS library.
- **bokeh.server:** It provides customizable Bokeh Server Tornado application. Server is used to share and publish interactive plots and apps to an audience of your choice.

Plots with Glyphs

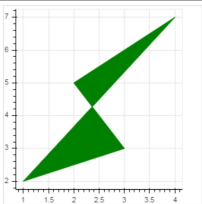
Glyphs is geometrical shapes, such as line, circle, and rectangle that have visual information about the corresponding set of data. The types of plots created using glyphs are given as:



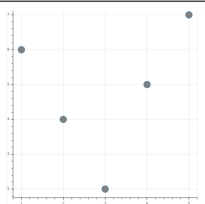
line plot



bar plot



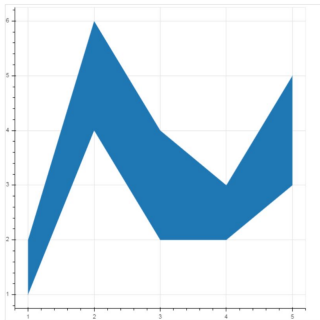
patch plot



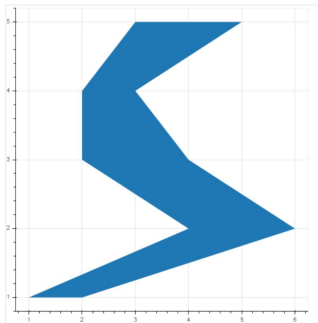
scatter plot

Area Plots

Area plots are filled regions between two series that share a common index. Bokeh's Figure class has two methods: **varea()** and **harea()**.



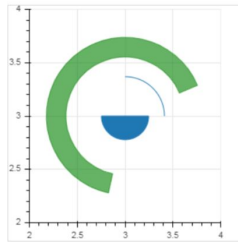
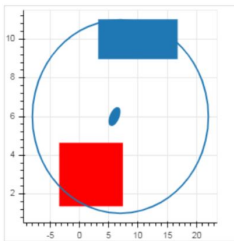
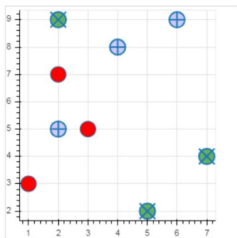
vertical area



horizontal area

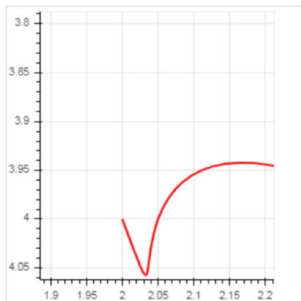
Circle Glyphs

The figure object has many methods using which vectorised glyphs of different shapes such as circle, rectangle, polygon, etc. can be drawn. Methods for drawing circle glyphs: **circle()**, **circle_cross()**, and **circle_x()**. It is possible to render rectangle, ellipse and polygons in a Bokeh figure.



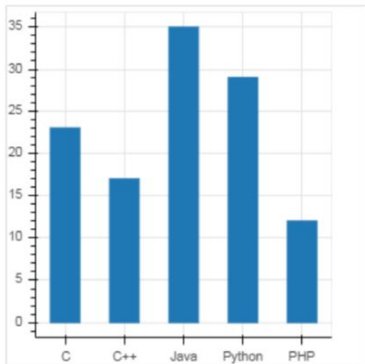
Bezier Curves

A Bézier curve is a parametric curve used in computer graphics. Other uses include the design of computer fonts and animation, user interface design and for smoothing cursor trajectory. In vector graphics, Bézier curves are used to model smooth curves that can be scaled indefinitely. A "Path" is combination of linked Bézier curves. We also have **quadratic()** to add a parabola glyph to bokeh figure. The function has same parameters as **beizer()**.



Categorical Axes

In the examples so far, the Bokeh plots show numerical data along both x and y axes. In order to use categorical data along either of axes, we need to specify a `FactorRange` to specify categorical dimensions for one of them.

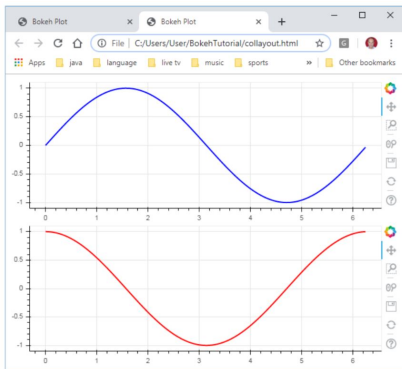


Bokeh-Pandas

In all the examples above, the data to be plotted has been provided in the form of Python lists or numpy arrays. It is also possible to provide the data source in the form of pandas DataFrame object. DataFrame is a two-dimensional data structure. Columns in the dataframe can be of different data types. The Pandas library has functions to create dataframe from various sources such as CSV file, Excel worksheet, SQL table, etc.

Bokeh- Layouts

Bokeh visualizations can be suitably arranged in different layout options. These layouts as well as sizing modes result in plots and widgets resizing automatically as per the size of browser window. For consistent appearance, all items in a layout must have same sizing mode. The widgets (buttons, menus, etc.) are kept in a separate widget box and not in plot figure.



Plots Tools

The tools can be classified under following categories: Pan/Drag Tools, Click/Tap Tools, and Scroll/Pinch.

Tool	Description	Icon
BoxSelectTool Name : 'box_select'	allows the user to define a rectangular selection region by left-dragging a mouse	
BoxZoomTool name: 'box_zoom'	allows the user to define a rectangular region to zoom the plot bounds too, by left-dragging a mouse	
LassoSelectTool name: 'lasso_select'	allows the user to define an arbitrary region for selection by left-dragging a mouse	
PanTool name: 'pan', 'xpan', 'ypan',	allows the user to pan the plot by left-dragging a mouse	
TapTool name: 'tap'	allows the user to select at single points by clicking a left mouse button	

WheelZoomTool name: 'wheel_zoom', 'xwheel_zoom', 'ywheel_zoom'	zoom the plot in and out, centered on the current mouse location.	
WheelPanTool name: 'xwheel_pan', 'ywheel_pan'	translate the plot window along the specified dimension without changing the window's aspect ratio.	
ResetTool name: 'reset'	restores the plot ranges to their original values.	
SaveTool name: 'save'	allows the user to save a PNG image of the plot.	
ZoomInTool name: 'zoom_in', 'xzoom_in', 'yzoom_in'	The zoom-in tool will increase the zoom of the plot in x, y or both coordinates	
ZoomOutTool name: 'zoom_out', 'xzoom_out', 'yzoom_out'	The zoom-out tool will decrease the zoom of the plot in x, y or both coordinates	
CrosshairTool name: 'crosshair'	draws a crosshair annotation over the plot, centered on the current mouse position.	

Adding Widget

The `bokeh.models.widgets` module contains definitions of GUI objects similar to HTML form elements, such as button, slider, checkbox, radio button, etc. These controls provide interactive interface to a plot.

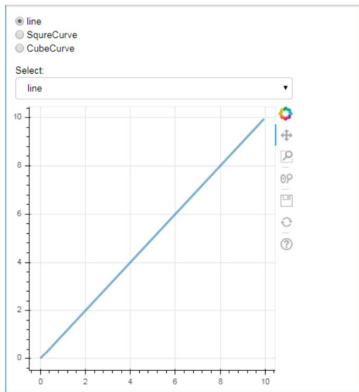
Bokeh allows call back functionality to be defined with two methods:

- Use the CustomJS callback so that the interactivity will work in standalone HTML documents.
- Use Bokeh server and set up event handlers.

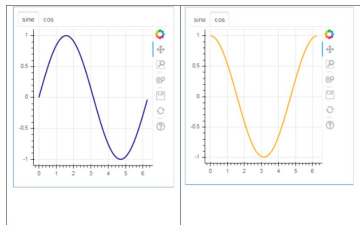
In this section, we shall see how to add Bokeh widgets and assign JavaScript callbacks.

- BUtton
- Slider
- RadioGroup
- Tab Widget

Bokeh output with widget



Button and select widget



Tab Widget

Miscellaneous

- Server
- Using Bokeh Subcommands
- Exporting Plots
- Embedding Plots and Apps

For more read:

Jolly, Kevin. Hands-On Data Visualization with Bokeh:
Interactive Web Plotting for Python Using Bokeh. Packt
Publishing Ltd, 2018.

Hands-on

Please download or copy to your gdrive the
following file Python
click this link

and open with your Google Colabs
<https://colab.research.google.com>

The End