1 Glossary

In order to make the best use of this User Guide, it is important to have context for some high level concepts and terms. Here is a small glossary of some of the most important concepts in Bokeh.

BokehJS

The JavaScript client library that actually renders the visuals and handles the UI interactions for Bokeh plots and widgets in the browser. Typically, users will not have to think about this aspect of Bokeh much (We write the JavaScript, so you dont have to!) but it is good to have basic knowledge of this dichotomy. For full details, see the BokehJS chapter of the Developer Guide.

Charts

Schematic statistical plots such as bar charts, horizon plots, time series, etc. that may include faceting, grouping, or stacking based on the structure of the data. Bokeh provides a high level bokeh.charts interface to quickly construct these kinds of plots. See Using High-level Charts for examples and usage.

Embedding

Various methods of including Bokeh plots and widgets into web apps and pages, or the IPython notebook.

Glyphs

The basic visual building blocks of Bokeh plots, e.g. lines, rectangles, squares, wedges, patches, etc. The `bokeh.plotting` interface provides a convenient way to create plots centered around glyphs.

Models

The lowest-level objects that comprise Bokeh scenegraphs. These live in the `bokeh.models` interface. Most users will not use this level of interface to assemble plots directly.

However, ultimately all Bokeh plots consist of collections of models, so it is important to understand them enough to configure their attributes and properties.

Server

The bokeh-server is an optional component that can be used for sharing and publishing Bokeh plots and apps, for handling streaming of large data sets, or for enabling sophisticated user interactions based off of widgets and selections.

Widgets

User interface elements outside of a Bokeh plot such as sliders, drop down menus, buttons, etc. Events and updates from widgets can inform additional computations, or cause Bokeh plots to update.