



bokeh

■ ■ ■ Project Overview

Bokeh is an interactive visualization library that targets modern web browsers for presentation.

It provides an elegant and concise way to construct versatile graphics, and to extend this capability with high-performance interactivity over very large or streaming datasets.

Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications.

■ ■ ■ Roadmap in Priority Order

- Integrate with VegaLite / Altair
- Visual Design Improvements
- Develop BokehJS as a First-Class JavaScript Library
- Use / Improve WebGL Throughout
- Better Theming Support
- Simple RPC Mechanism
- Patches with Holes
- Scripted / Smooth Animations and Transitions
- Improve the Documentation Organization and Accessibility
- Built-in LaTeX / MathText Support
- Improvements to Layout

LEARN MORE
sales@quansight.com





bokeh

Integrate with VegaLite / Altair

Bokeh would like to be able to ingest the declarative VegaLite specification and produce Bokeh plots as a result. This would allow users to take advantage of the very high level Altair library for exploratory data analysis and obtain real Bokeh plots. These could be further customized beyond what is available in Altair directly.

Visual Design Improvements

Bokeh has never had the benefit of any dedicated web or visual designers, and this is one area where it suffers compared to some other projects. While advances have been made over time, there is still room for visual improvement. Additionally, Bokeh would benefit from **Better Theming Support**. This would be accomplished by having visual designers help craft a set of attractive themes that would be immediately available to users.

Develop BokehJS as a First-Class JavaScript Library

By and large none of the core contributors have been JavaScript or front-end experts, and for some time the client side BokehJS library was an implementation detail. Now, there is more interest to use BokehJS directly. With attention and contribution from experienced front-end JavaScript developers, BokehJS could integrate better with existing common web frameworks and tools. This includes making more accessible documentation for BokehJS itself. These improvements would also help increase the potential pool of core developers interested in maintaining BokehJS over the long run.

Use / Improve WebGL Throughout

Bokeh currently can use WebGL to accelerate plotting for larger datasets in some limited situations. In addition to not fully supporting all use-cases, there is a large internal cost in maintaining two separate code paths. It would be desirable to convert all of Bokeh's rendering to use WebGL (possibly via a higher level library such as ReGL). Such a change would ensure that better performance for larger data sets is always available. Moving to WebGL would also streamline maintenance for the future.

Better Theming Support

Bokeh currently has minimal theming capabilities. Bokeh would like to extend this to have support for more targets and to have more sophisticated ways to theme individual objects or collections. Additionally, work is needed to improve and create visual assets that work well with different themes, and to make DOM elements easily themeable. For example, it is hard to make a "dark" theme now, because the toolbar is always light by default and it is difficult to change.

Simple RPC Mechanism

Bokeh currently allows for users to execute JavaScript code or Python code (in Bokeh server applications) in response to data changes or various UI events. Some users have requested a more direct (simple) "remote procedure" capability that would enable them to execute a JavaScript function directly from Python in a Bokeh server application or vice versa.

Patches with Holes

Currently Bokeh can draw polygonal patch (or multi-patch) glyphs, but these may not have "holes" in them. Adding support for Patches with Holes is needed to support various kinds of scientific plots (such as contour plots) as well as to support certain types of Geo/GIS usage.

Scripted / Smooth Animations and Transitions

Many users have expressed interest in the ability to make different views of a plot transform smoothly from one state to another. This would afford the ability to create visually attractive "data stories" by scripting smooth transitions from one view of data to another.

Improve the Documentation Organization and Accessibility

The Bokeh documentation is extensive. However, many users report having difficulty finding the appropriate documentation or examples. The project would benefit greatly from experienced help to organize the structure of the docs, point out missing areas, and make general improvements.

Built-in LaTeX / MathText Support

Bokeh can display mathematical symbols and formulas by creating and using custom extensions. Users have expressed a desire for this capability to be more directly integrated into plot elements such as axis and tick labels.

Improvements to Layout

Bokeh has a built in layout system that is suitable for simple use cases. Unfortunately, there are certain use-cases that do not function well or properly. We would like to make Bokeh's layout system more robust and applicable as widely as possible. We would also like to demonstrate alternative options using external layout libraries for more sophisticated use cases.