Generating Bokeh Plots with Haskell: BokeHS Proof of Concept

Mark Hay
Github: ahaym

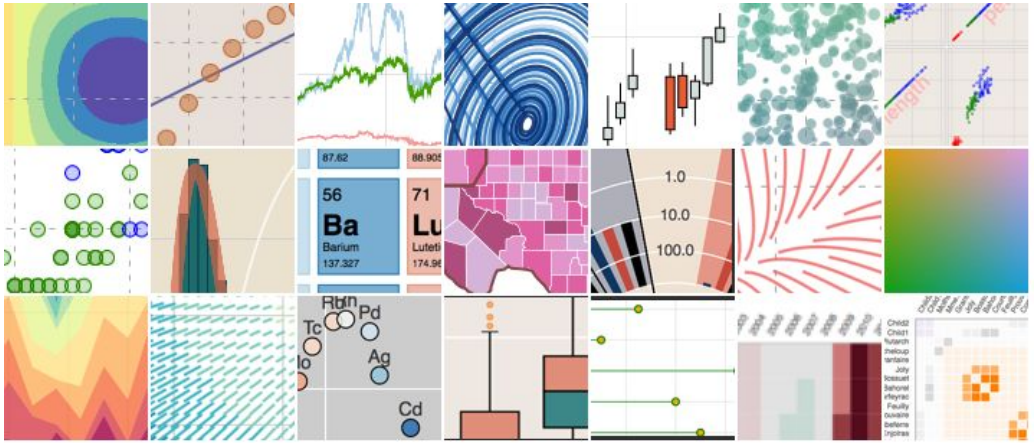
What is Bokeh?

Bokeh is a powerful data visualisation library

Plots are described as a Typed DSL that is serialized to JSON

Bokeh officially supports only Python and Javascript, but can technically accommodate any language as long as well-formed plot JSON can be emitted

Your JSON + Bokeh Renderer File



Why Haskell?

```
xdr = Range1d(start=-0.5, end= 20.5)
ydr = Range1d(start=-0.5, end= 20.5)

plot = Plot(
        title=Title(text="Sample Python Plot"),
        x_range=xdr,
        y_range=ydr,
        plot_width=400,
        plot_height=400,
        background_fill_color= "#F2F2F7"
        )
plot.add_layout(LinearAxis(), "below")
plot.add_layout(LinearAxis(), "left")

line = Line(x="x", y="y", line_color="blue")
plot.add_glyph(source, line)
```

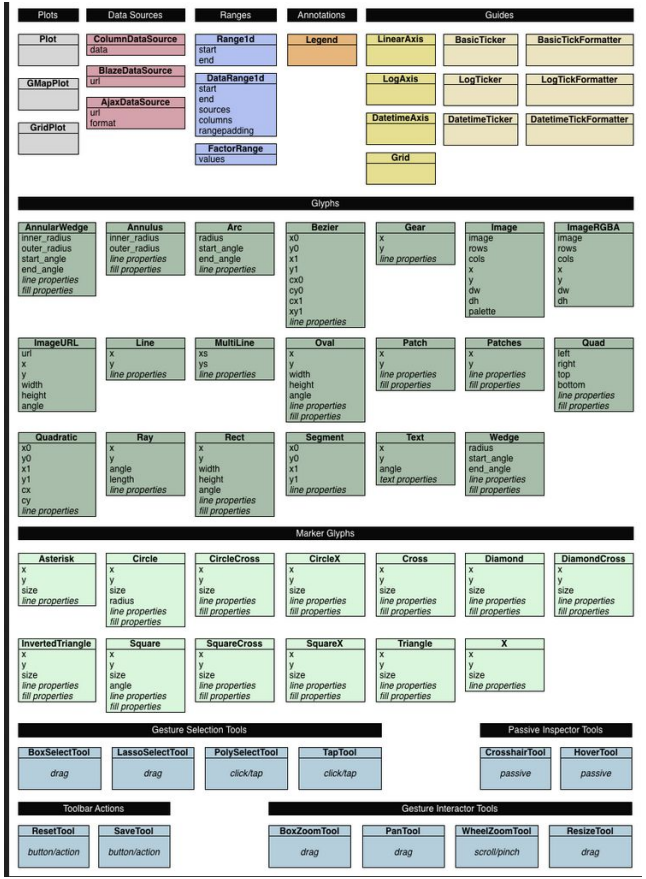
Goal

Bokeh's Plot DSL is huge

I had almost nothing before ZuriHac

Write enough of a Haskell DSL to represent a simple line plot

Write a function to compile this DSL to a Bokeh-readable format



This library proves that Bokeh plots can be conveniently specified and serialized in Haskell. Next Steps:

Extend to take advantage of all of Bokeh’s features

Create more convenient functions to construct plots instead of writing out the ADTs by hand

Current Data Store Representation: [(Text, Array Scientific)]

Would like a way to verify that fields referred to are actually in the Data