

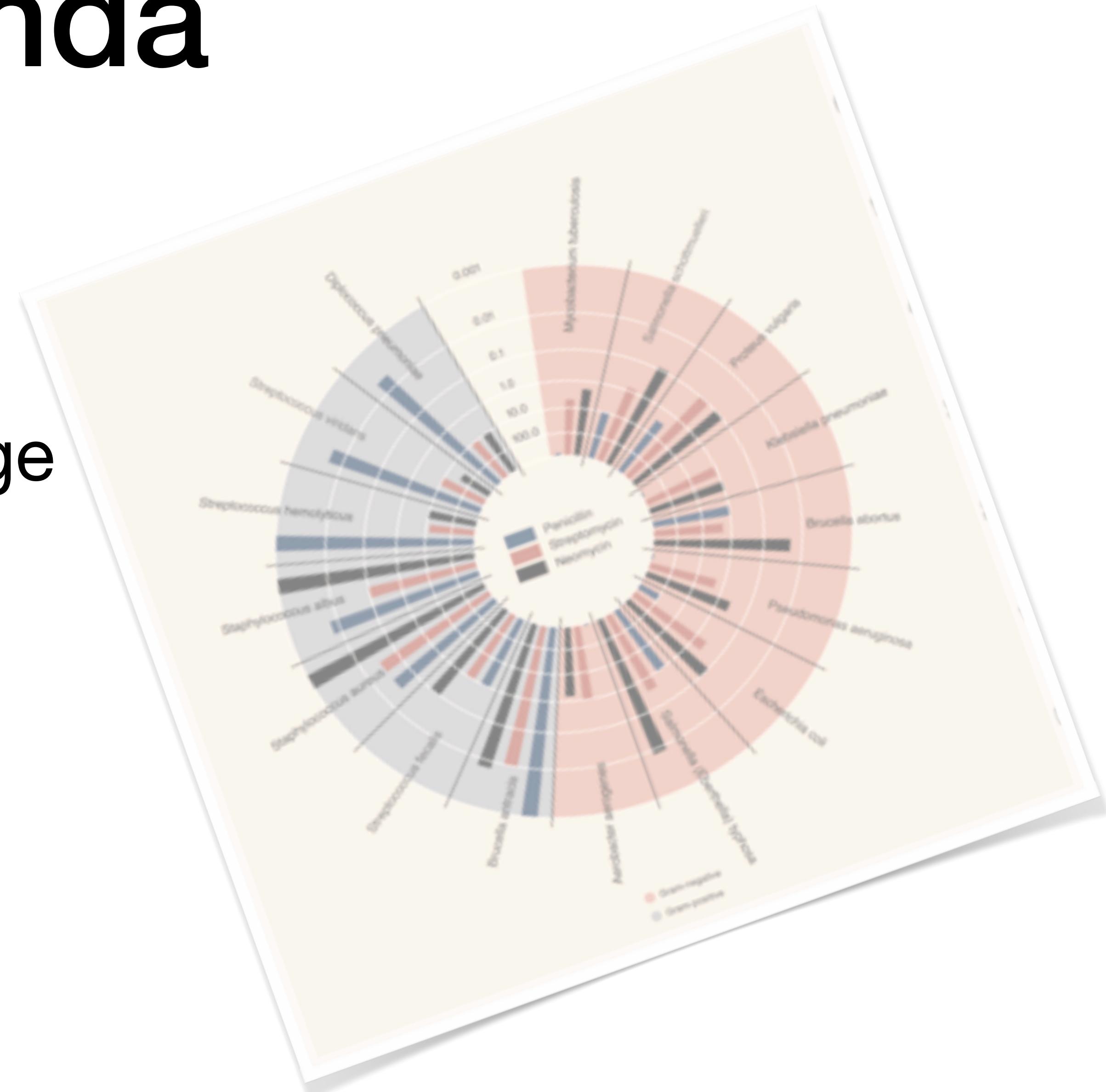
Bokeh



making interactive visualization in Python

Agenda

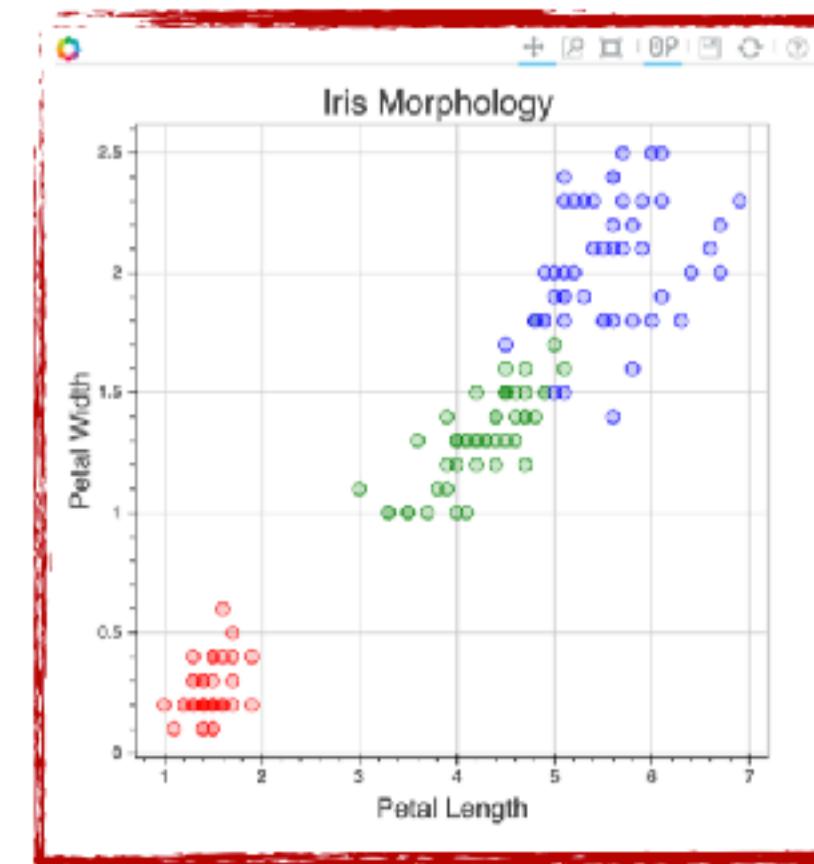
- Introduction
 - Strengths and Challenge
 - Example
 - Appendix



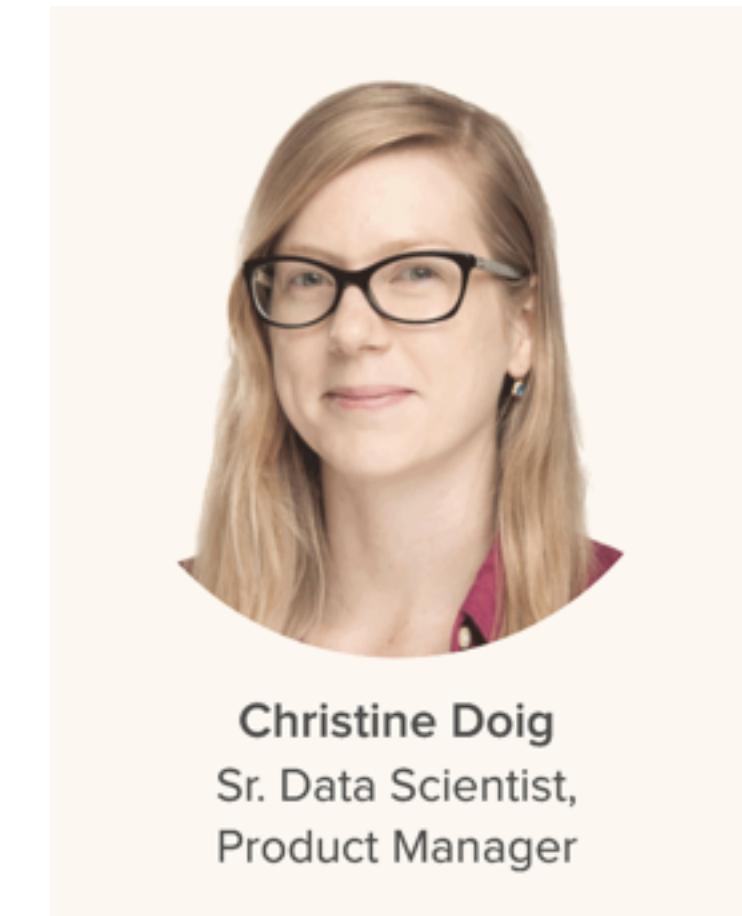
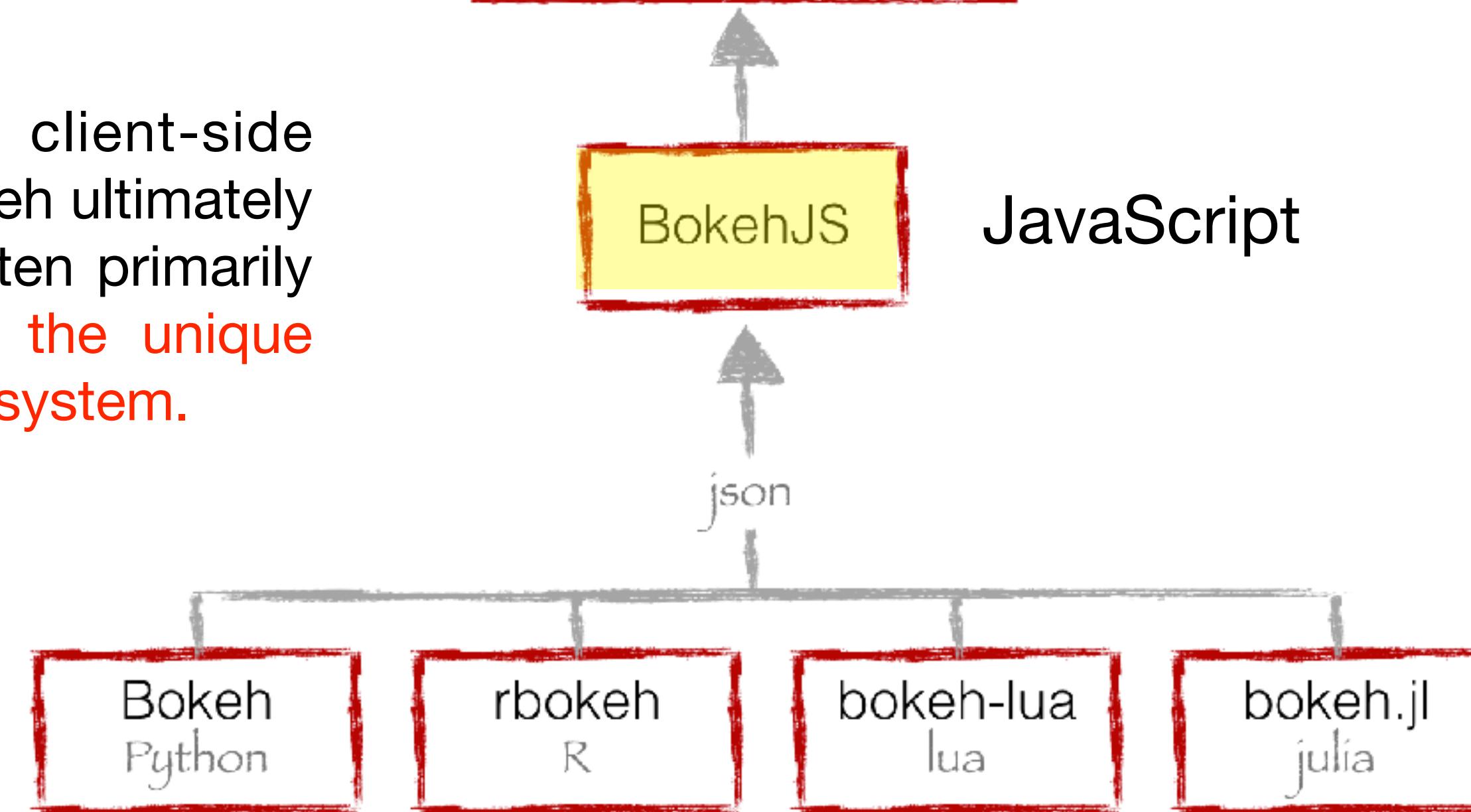


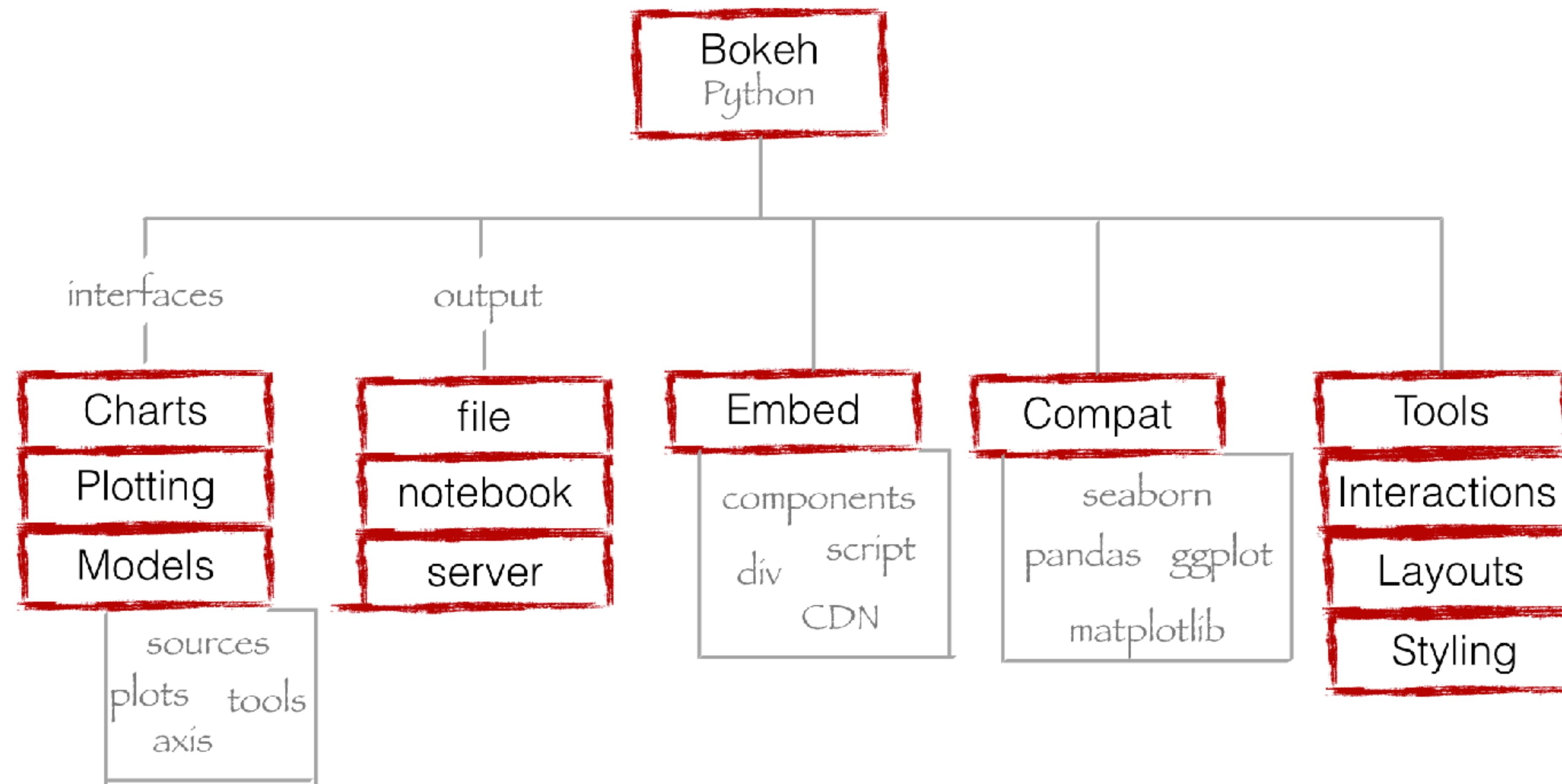
Bokeh 為 Python 套件，**提供 Python 與 D3.js 之間的橋梁**，不需要親自寫 JavaScript。

```
<html>
  <script>...</script>
  <div>
    <canvas>...</canvas>
  </div>
</html>
```



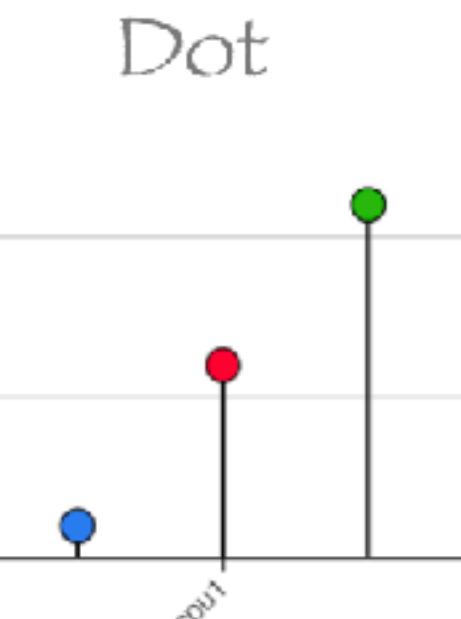
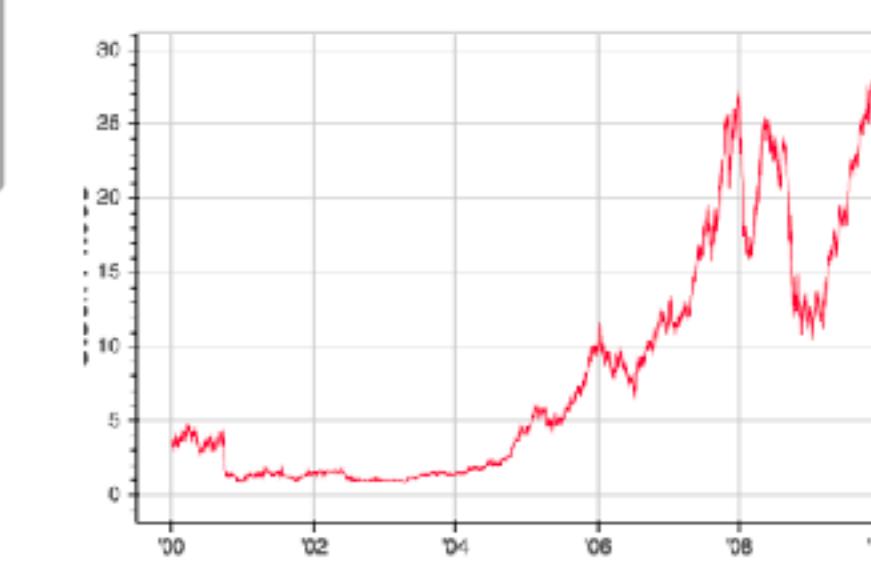
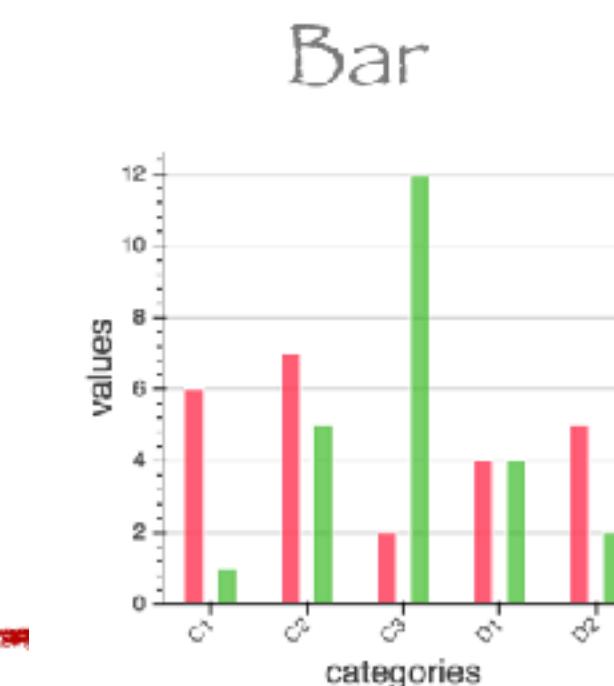
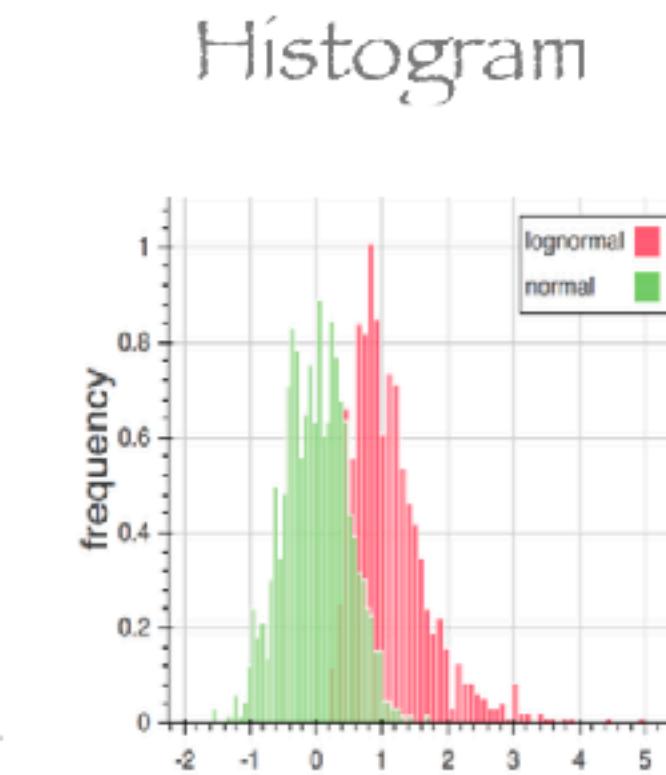
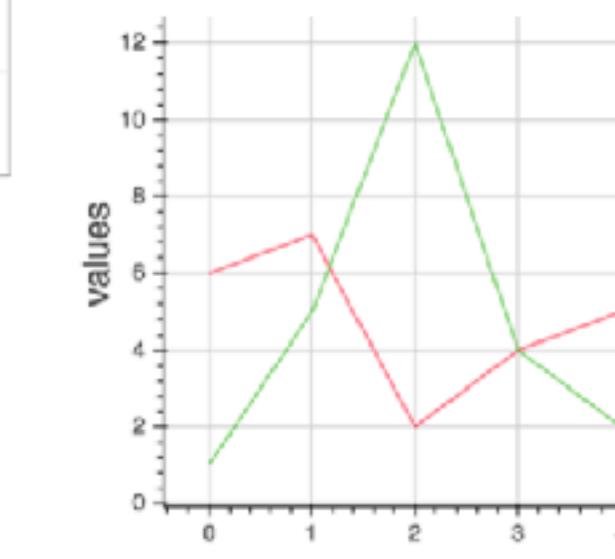
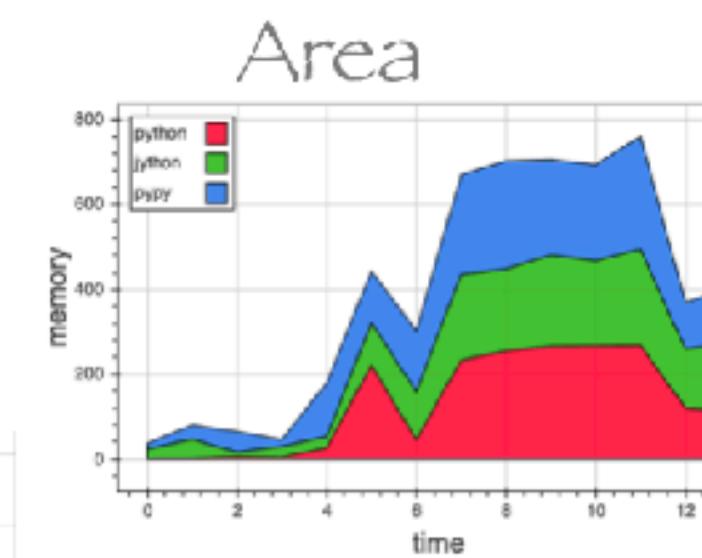
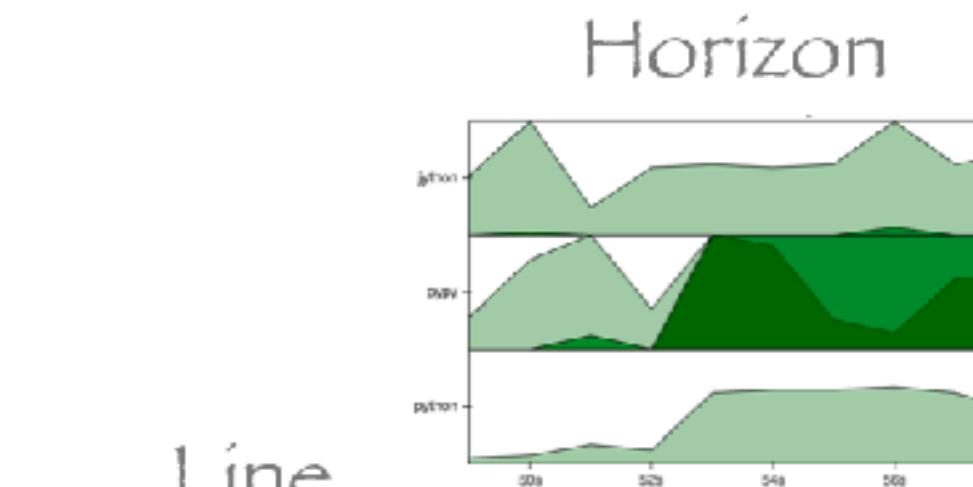
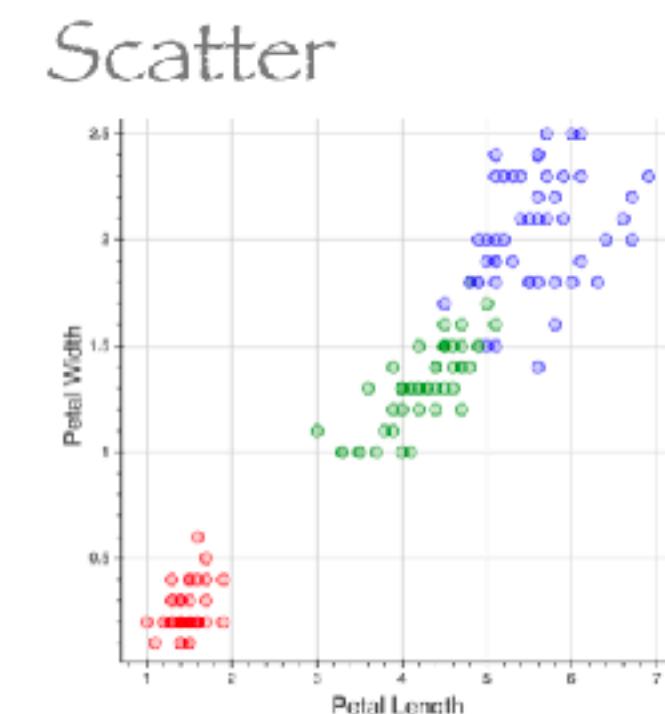
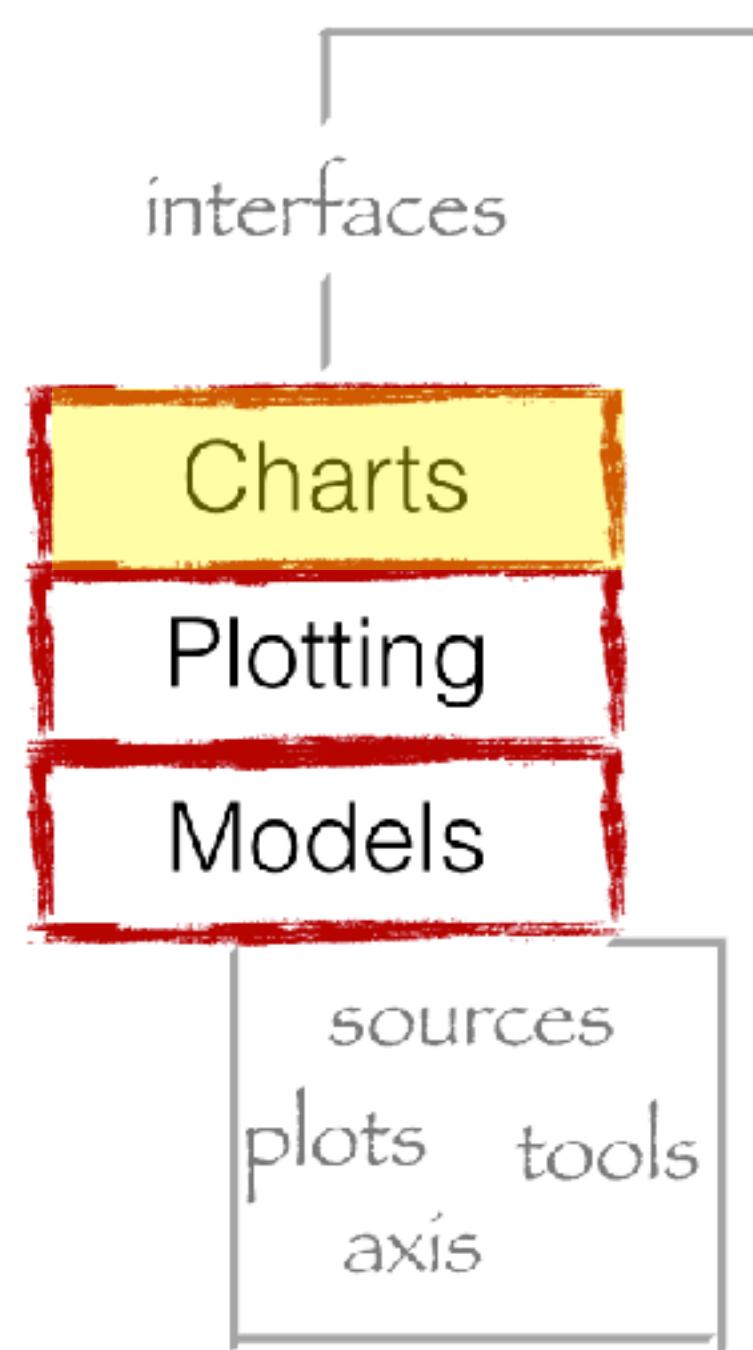
BokehJS is the in-browser client-side runtime library that users of Bokeh ultimately interact with. This library is written primarily in CoffeeScript and is one of the unique things about the Bokeh plotting system.





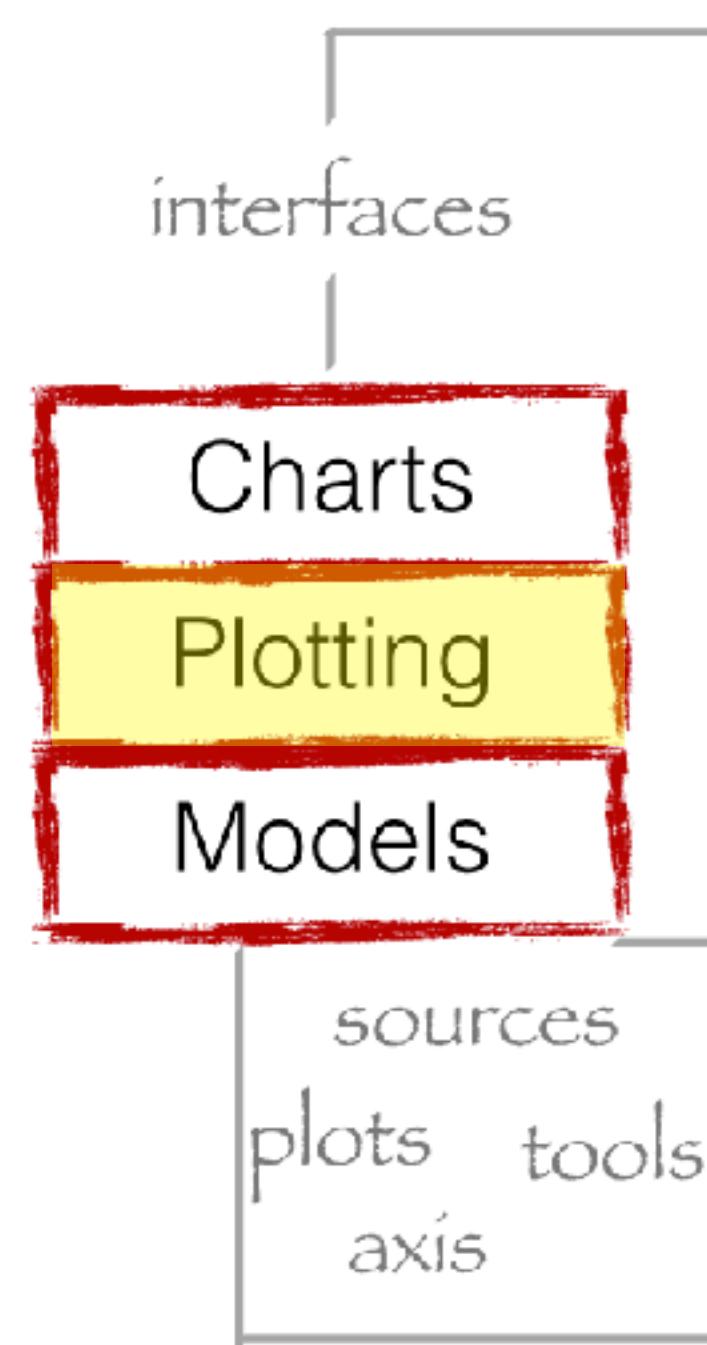
bokeh.charts

— 直接使用內建各種圖表。



bokeh.plotting

— 我們可以自由繪製圖表。

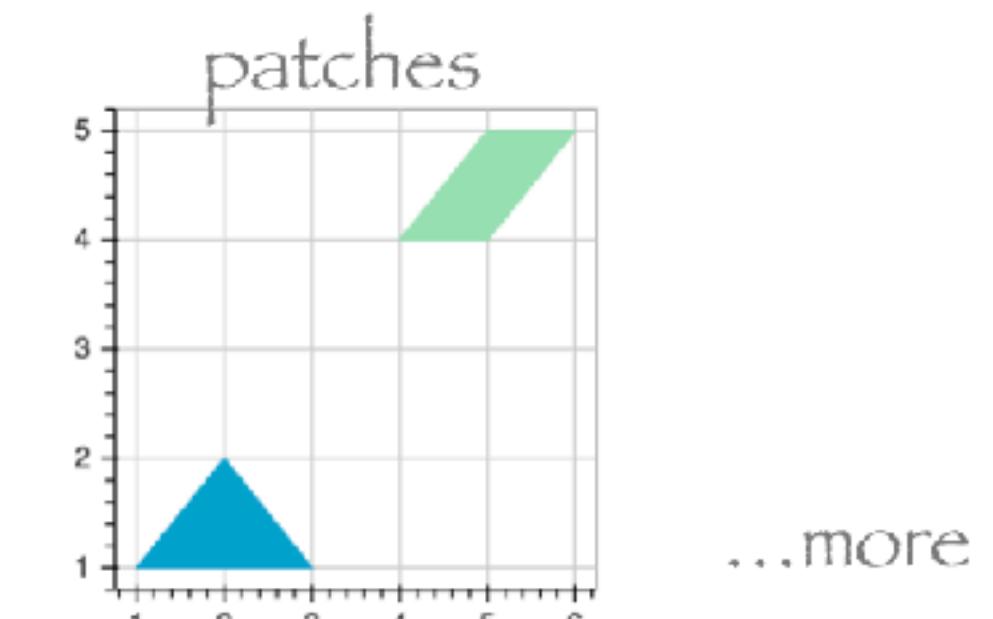
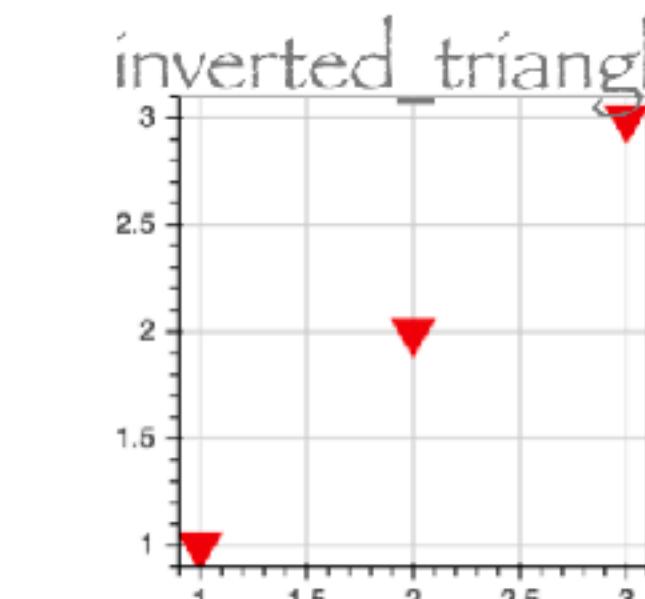
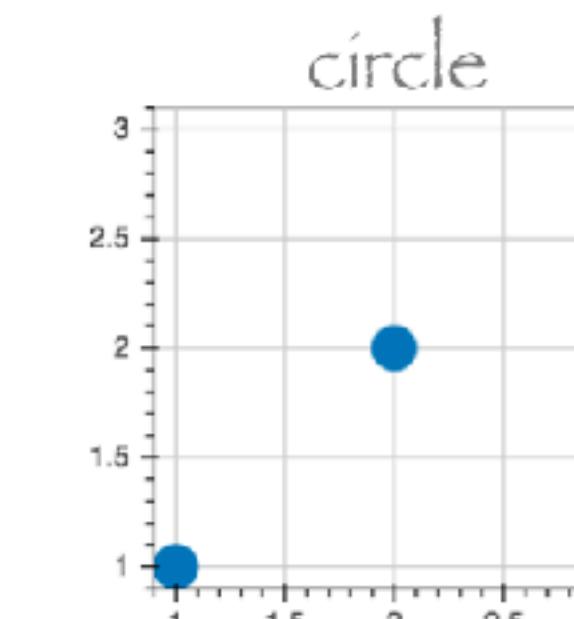
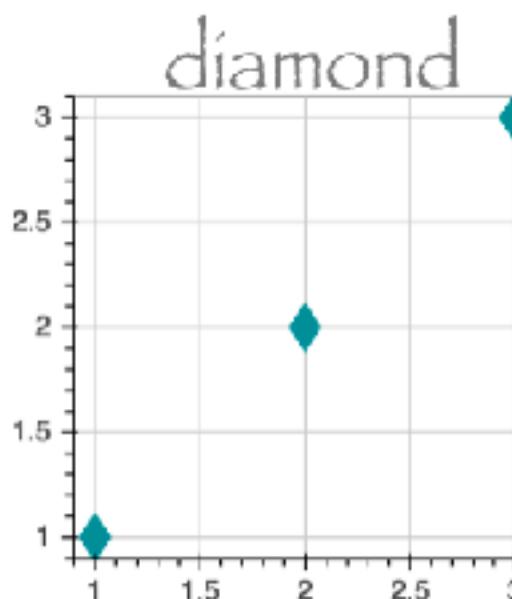
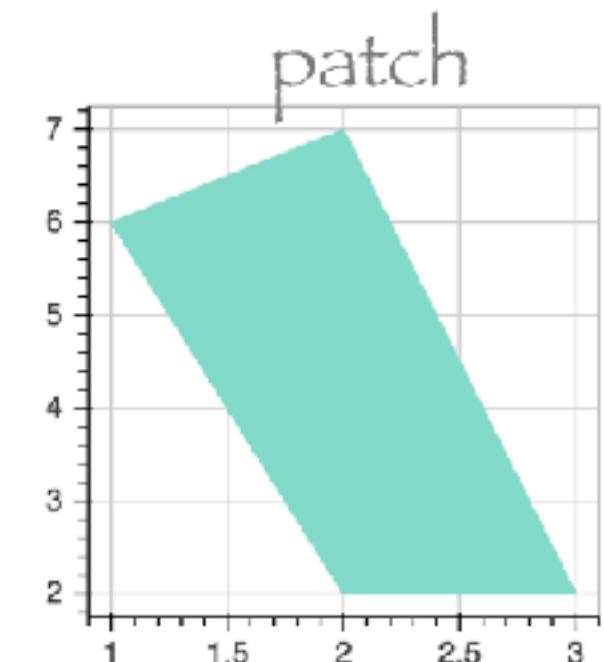
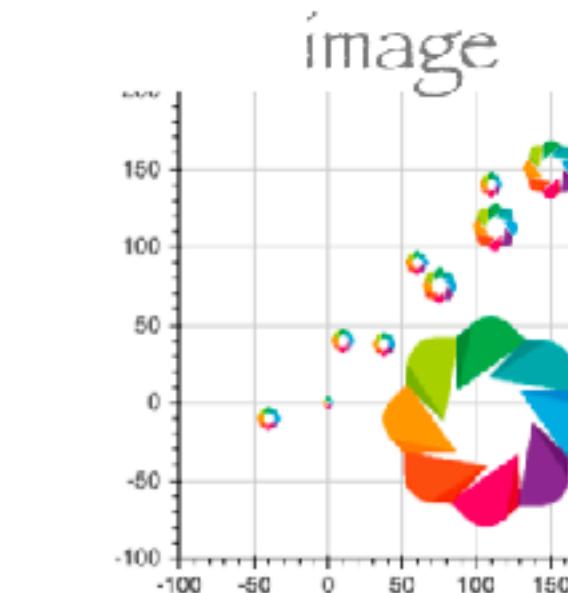
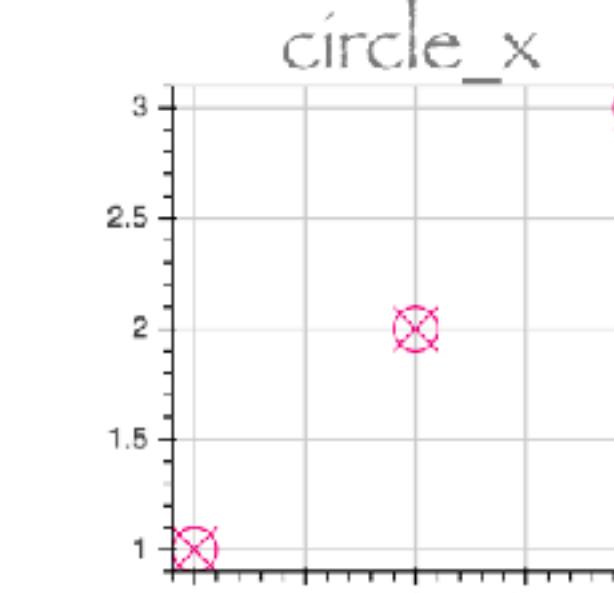
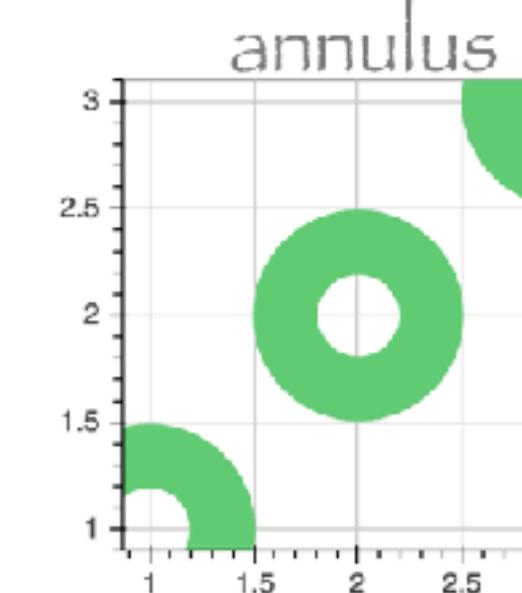


Plotting

figure Activate a new figure for plotting

```
plot = figure(width=300, height=300)
```

methods



```
plot.asterisk(x=[1,2,3], y=[1,2,3], size=20, color="#F0027F")
```

```
show(plot)
```

bokeh.models

Models

— 為開發人員提供靈活性。

interfaces

Charts

Plotting

Models

sources

plots tools

axis

sources ColumnDataSource Maps names of columns to sequences or arrays.

```
x = np.arange(-2*np.pi, 2*np.pi, 0.1)
y = np.sin(x)
source = ColumnDataSource(data=dict(x=x, y=y))
```

ranges DataRange1d An auto-fitting range in a continuous scalar dimension.

```
xdr = DataRange1d()
ydr = DataRange1d()
```

plots Plot Model representing a plot, containing glyphs, guides, annotations.

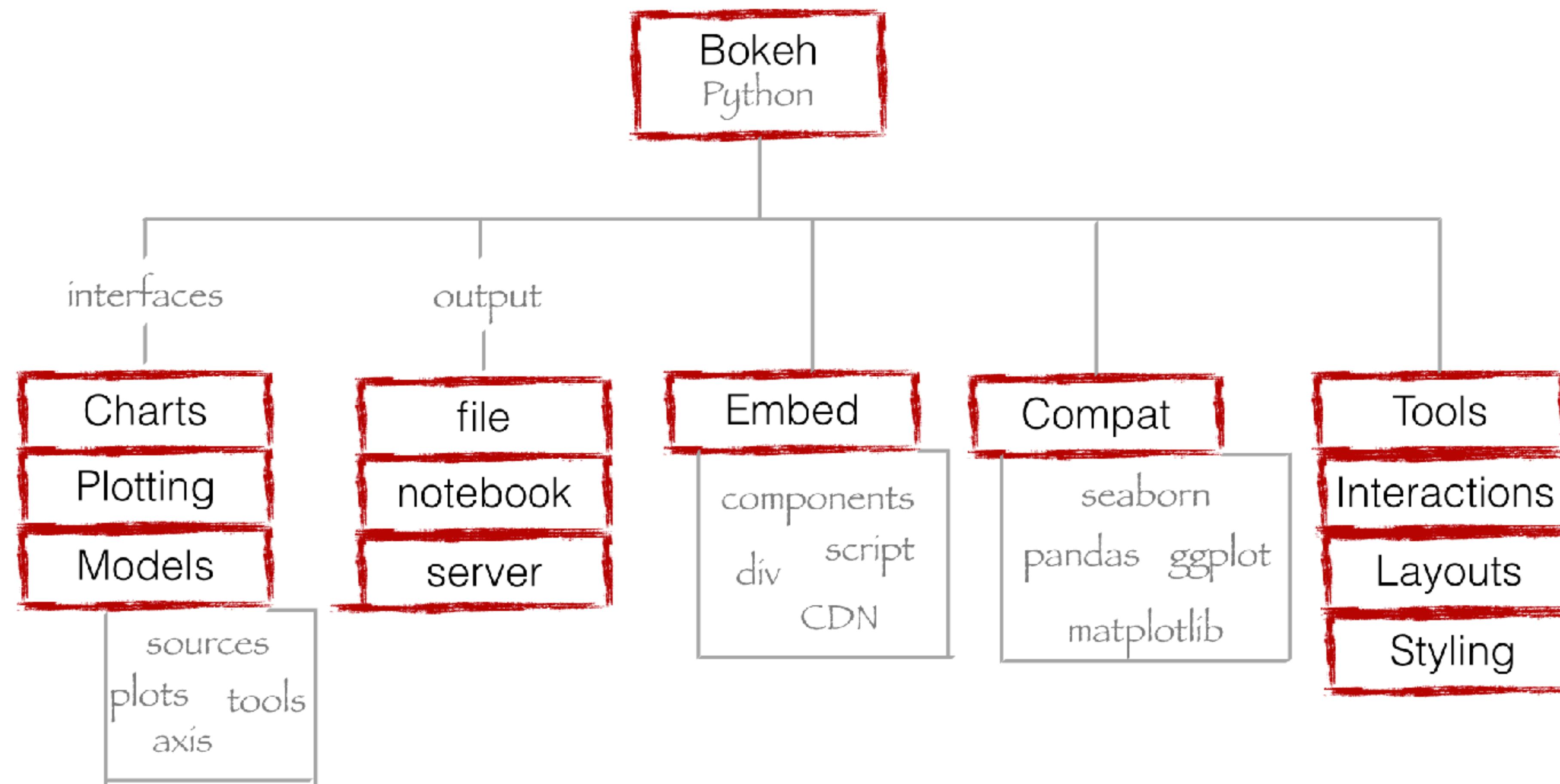
```
plot = Plot(x_range=xdr, y_range=ydr)
```

methods

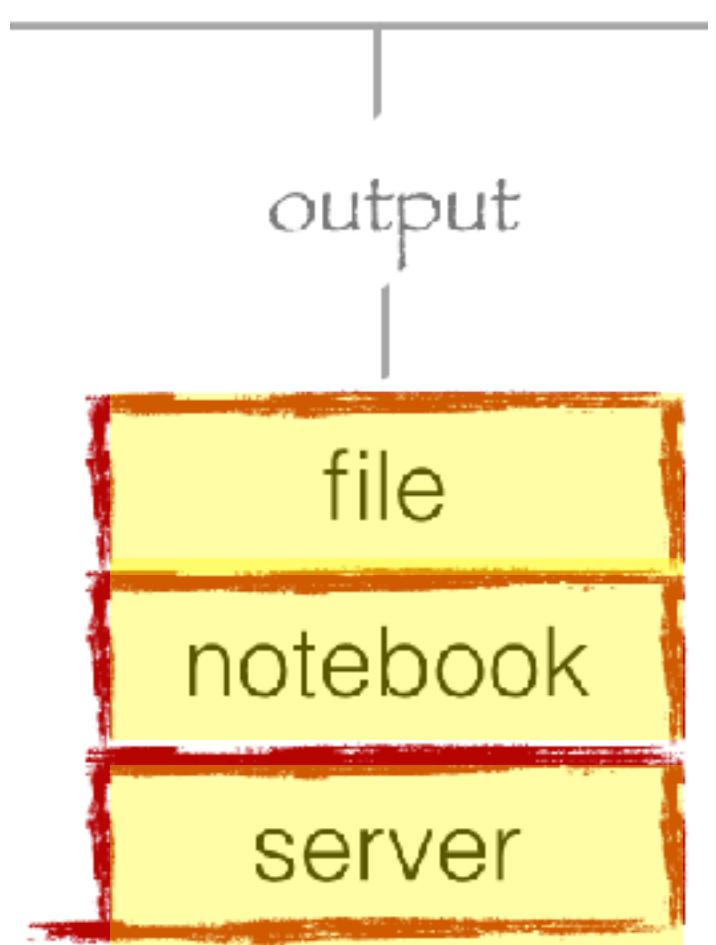
add_glyph plot.add_glyph(source, circle)

add_layout plot.add_layout(LinearAxis(), 'below')

add_tools plot.add_tools(PanTool(), WheelZoomTool())



I/O



Plots can be displayed inline in an IPython Notebook:

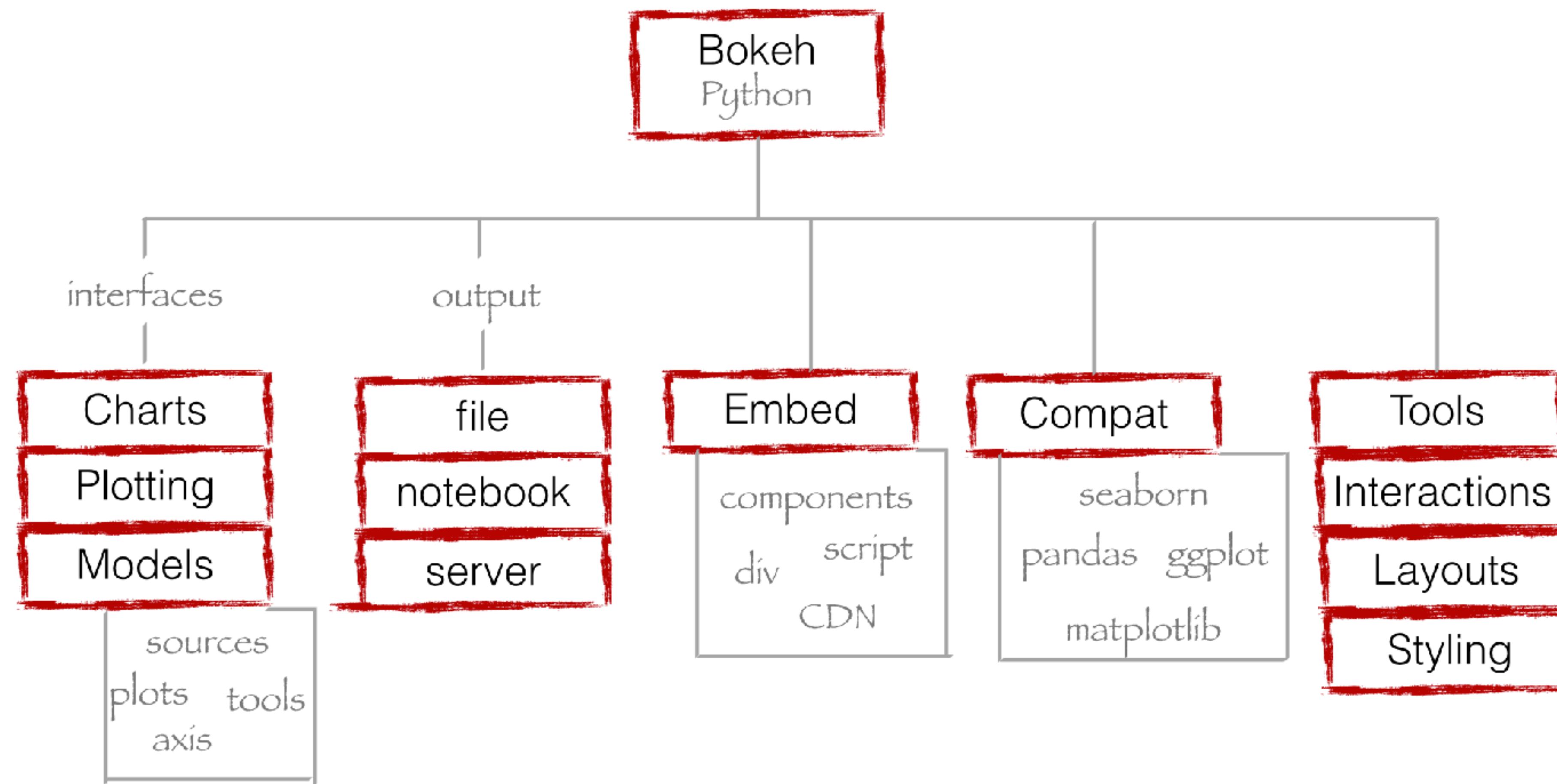
```
bk.plotting.output_notebook()
```

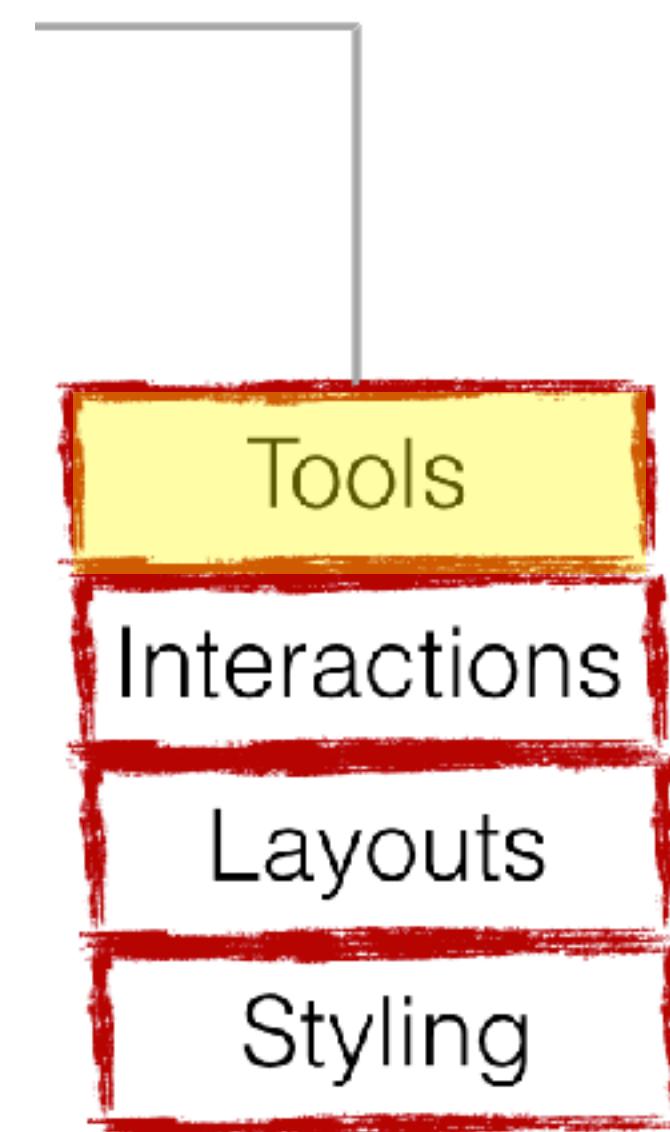
They can also be saved to file:

```
bk.plotting.output_file("output_filename.html")
```

Or to the bokeh-server:

```
bk.plotting.output_server("output_name")
```





Pan/Drag Tools

- BoxSelectTool
- BoxZoomTool
- LassoSelectTool
- PanTool
- Resize Tool

Click/Tap Tools

- PolySelectTool
- TapSelectTool

Scroll/Pinch Tools

- WheelZoomTool

Actions

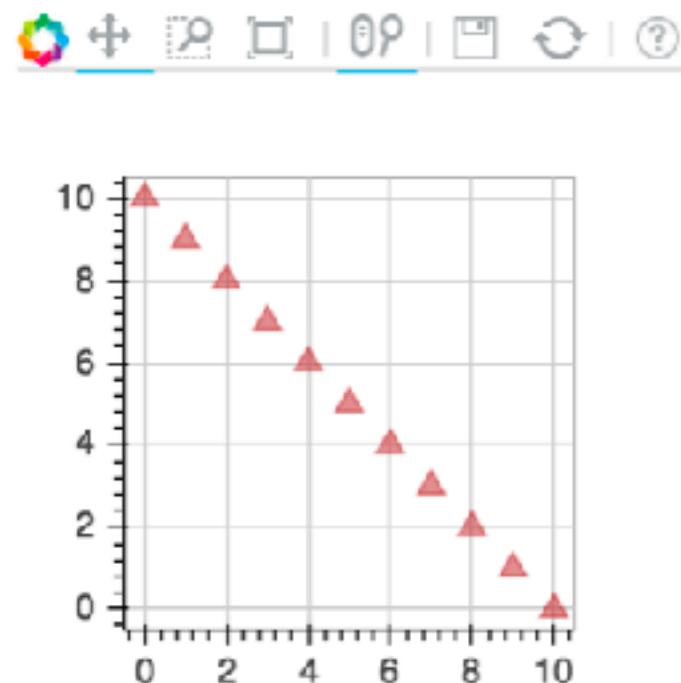
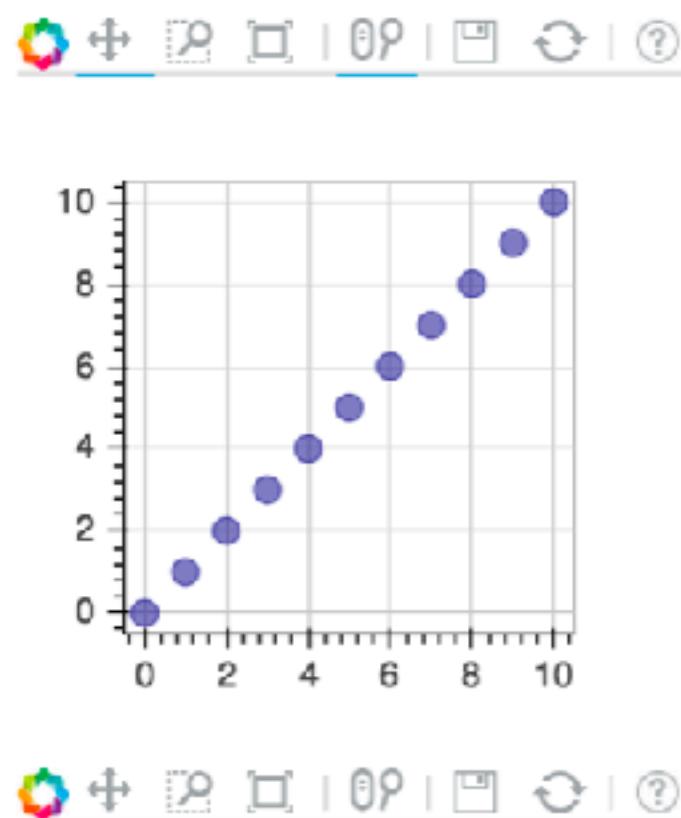
- ResetTool
- SaveTool

Inspectors

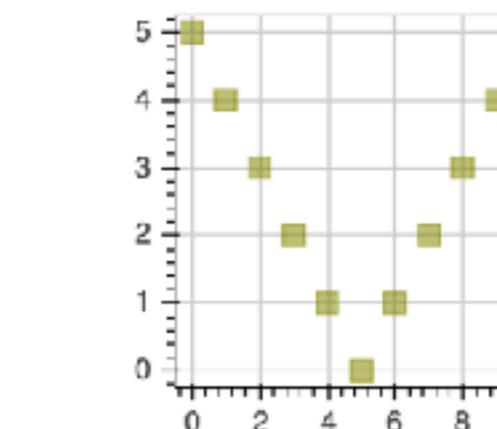
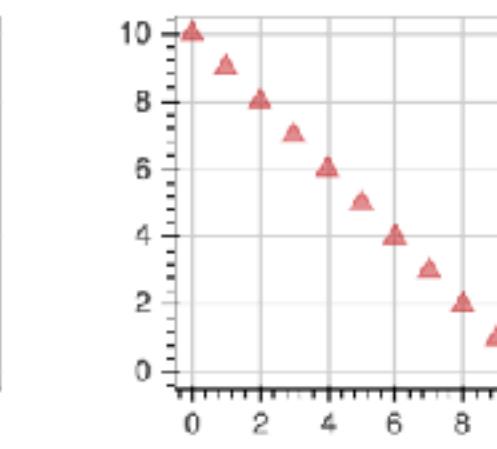
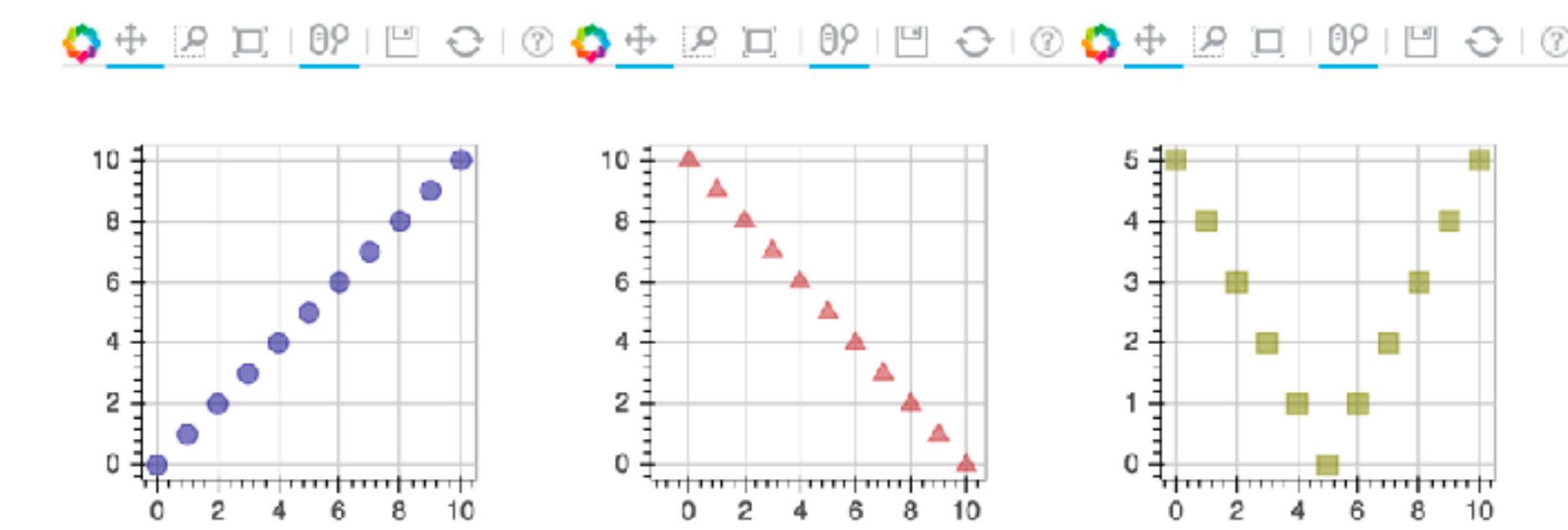
- CrosshairTool
- HoverTool

Layouts

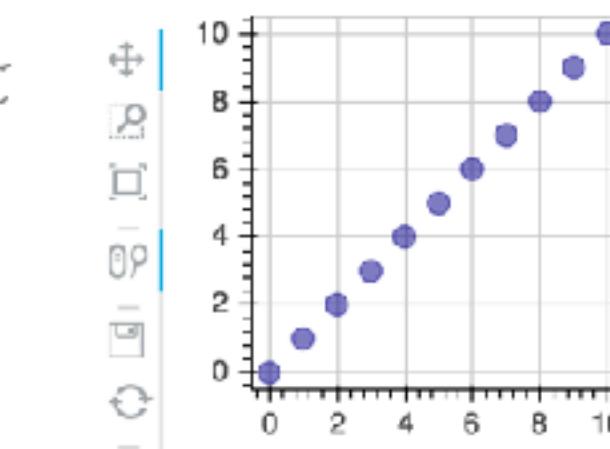
vplot



hplot



gridplot





Strengths and Challenge

Strengths and Challenge

允許通過簡單指令快速創建複雜統計圖

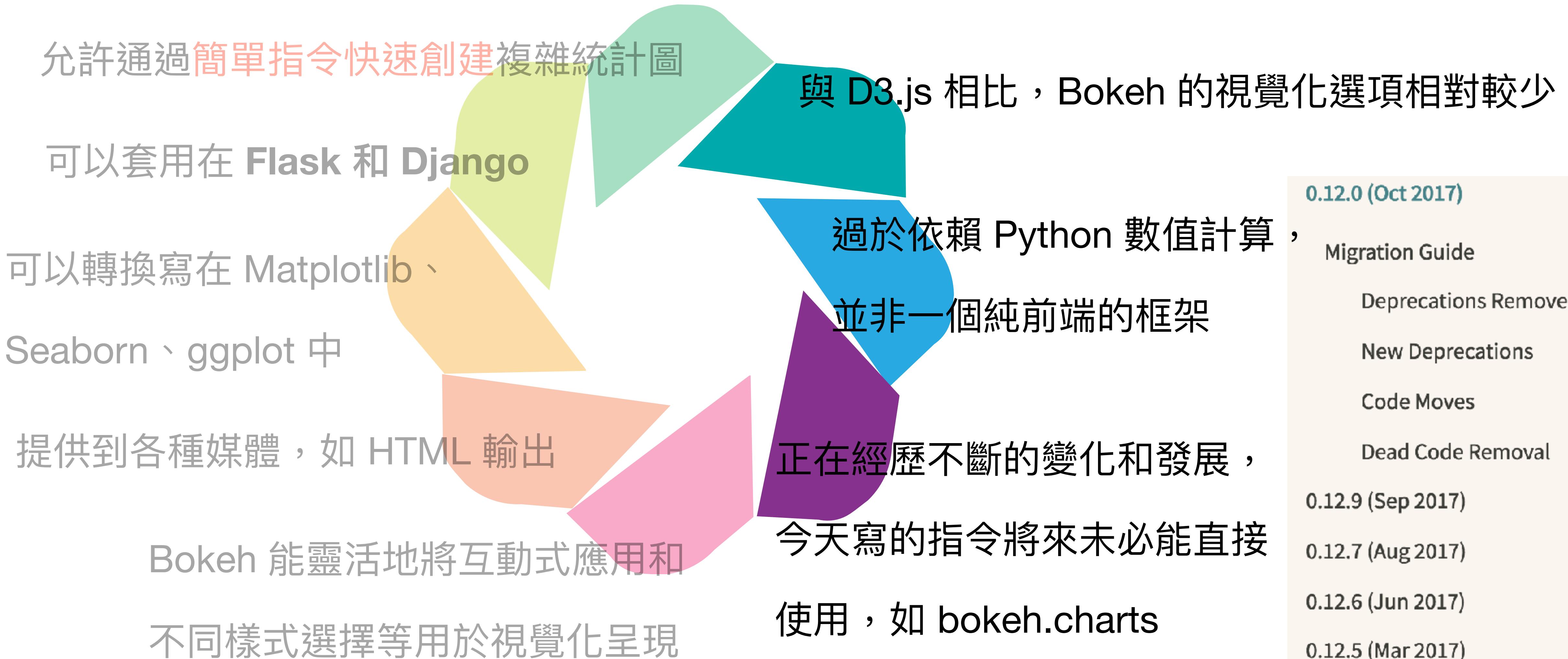
可以套用在 Flask 和 Django

可以轉換寫在 Matplotlib、
Seaborn、ggplot 中

提供到各種媒體，如 HTML 輸出

Bokeh 能靈活地將互動式應用和
不同樣式選擇等用於視覺化呈現

Strengths and Challenge



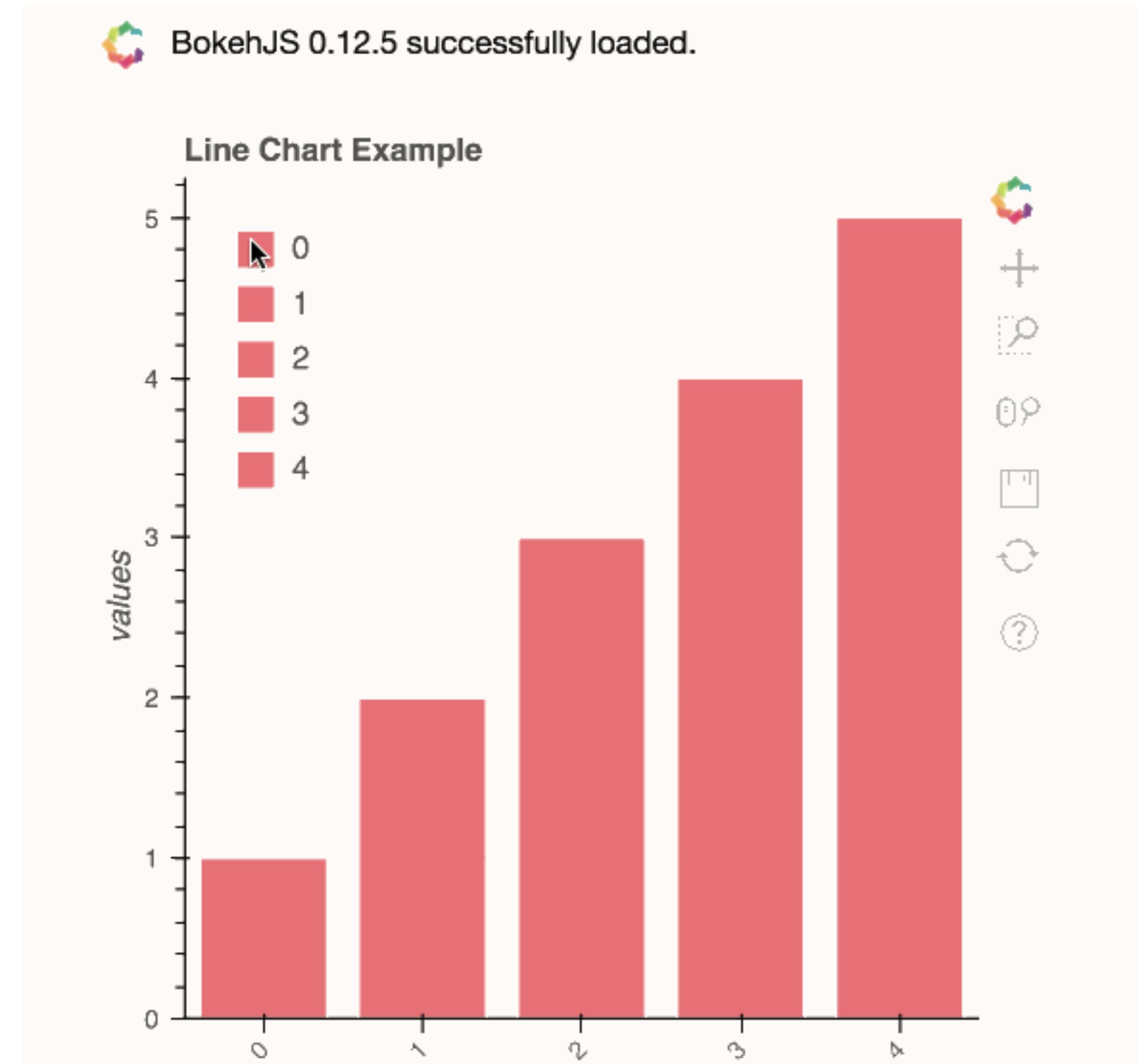


The graphic consists of several overlapping, rounded, irregular shapes in various colors: orange, yellow-green, green, cyan, blue, purple, pink, and red-orange. These shapes overlap in a non-uniform, organic manner, creating a sense of depth and complexity.

Example



bokeh.charts example



bokeh.plotting example

jupyter bokeh_plotting Last Checkpoint: 4 minutes ago (autosaved) [Logout](#)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 2

`p3.circle("mpg", "displ", size="cyl", line_color="red", fill_color=None, source=source)`

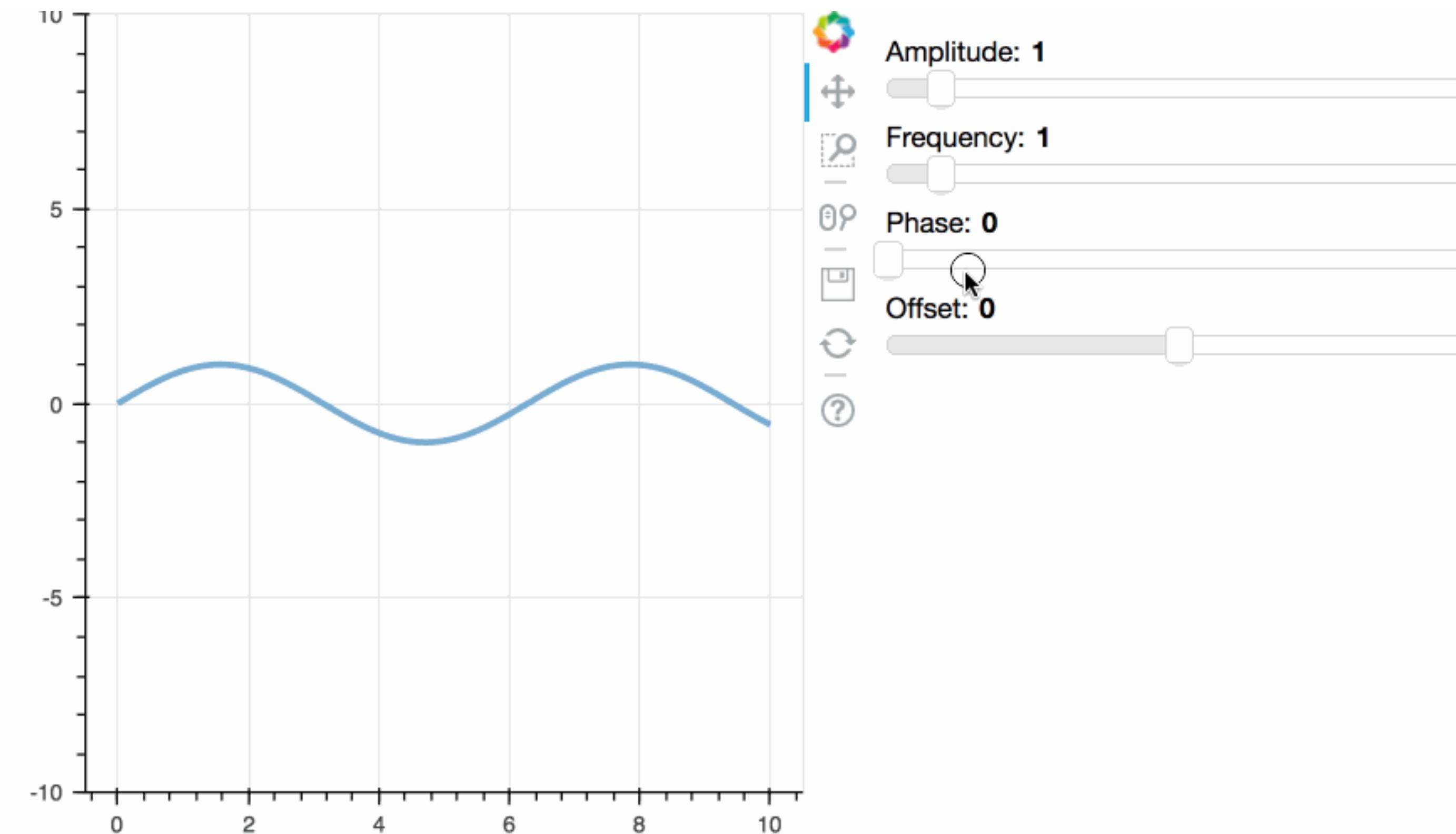
`p = gridplot([[p1, p2, p3]], toolbar_location="right")`

`show(p)`

The image shows three Bokeh plots generated from a Jupyter notebook cell. The first plot, titled 'MPG by Year', is a scatter plot of MPG versus Year, showing a general downward trend over time. The second plot, titled 'HP vs. Displacement', is a scatter plot of Horsepower (HP) versus Displacement, showing a positive correlation where larger engines tend to have higher horsepower. The third plot, titled 'MPG vs. Displacement', is a scatter plot of MPG versus Displacement, showing a negative correlation where smaller engines tend to have better fuel economy. All plots use a color palette consisting of various shades of red, green, blue, and black.

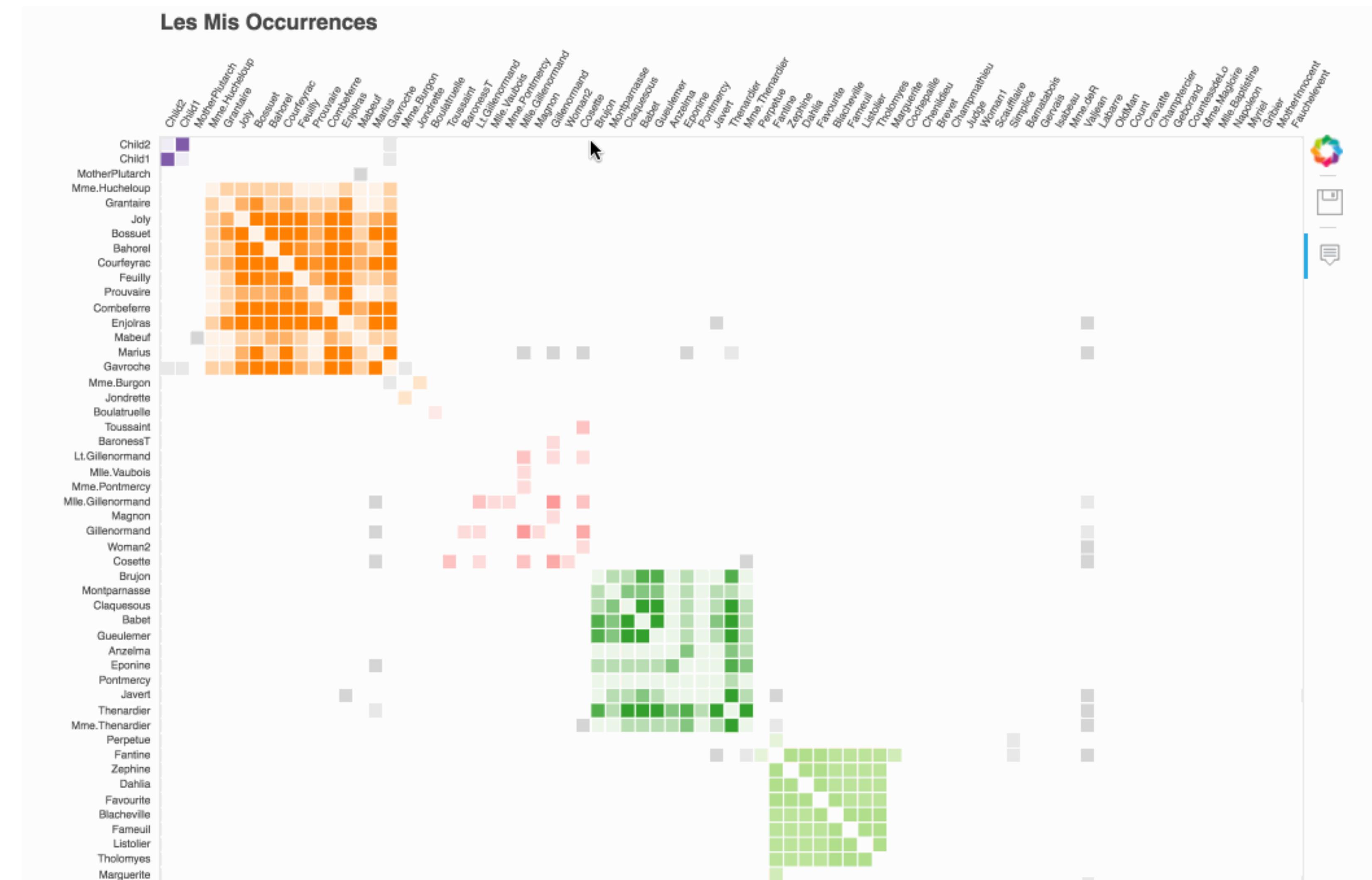


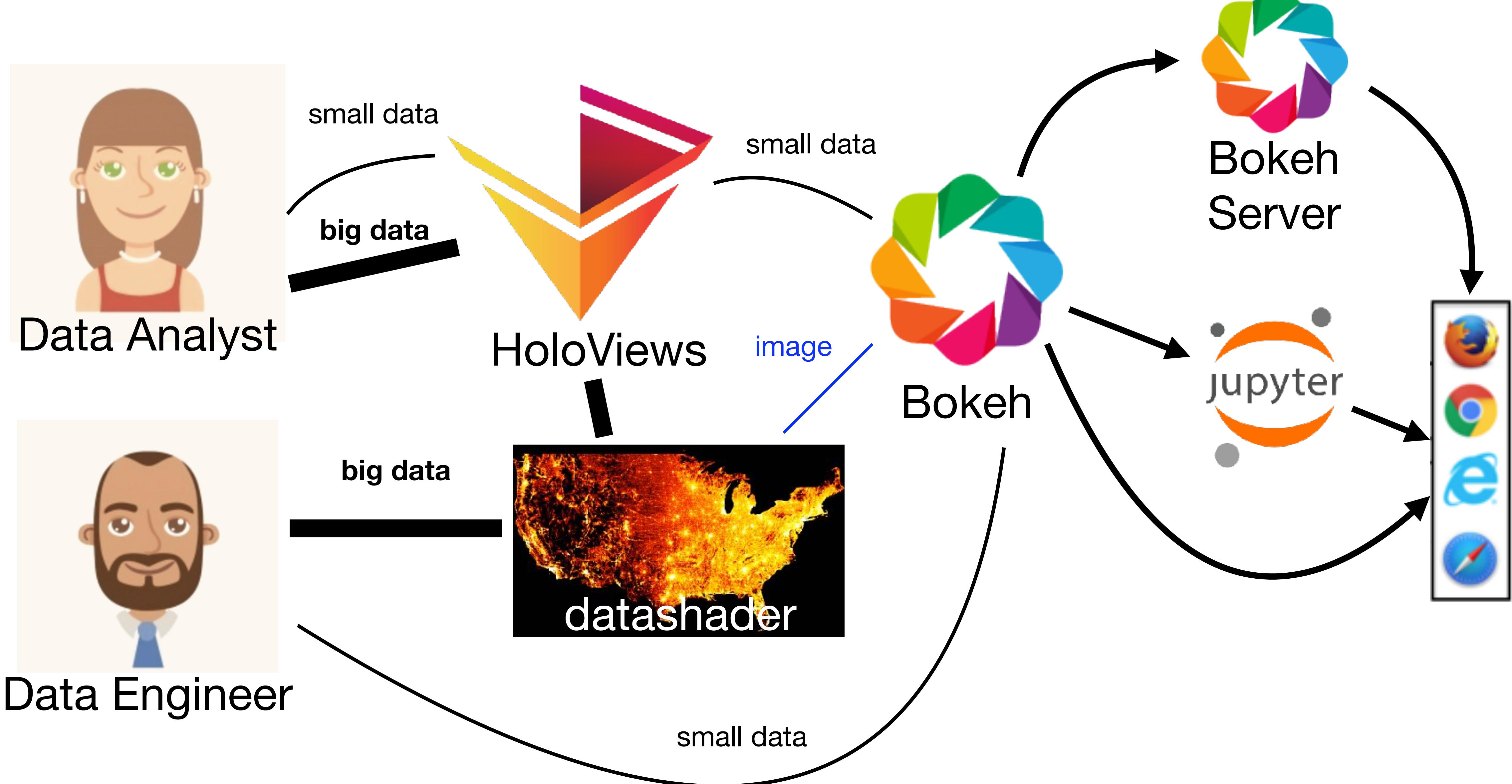
bokeh.plotting example





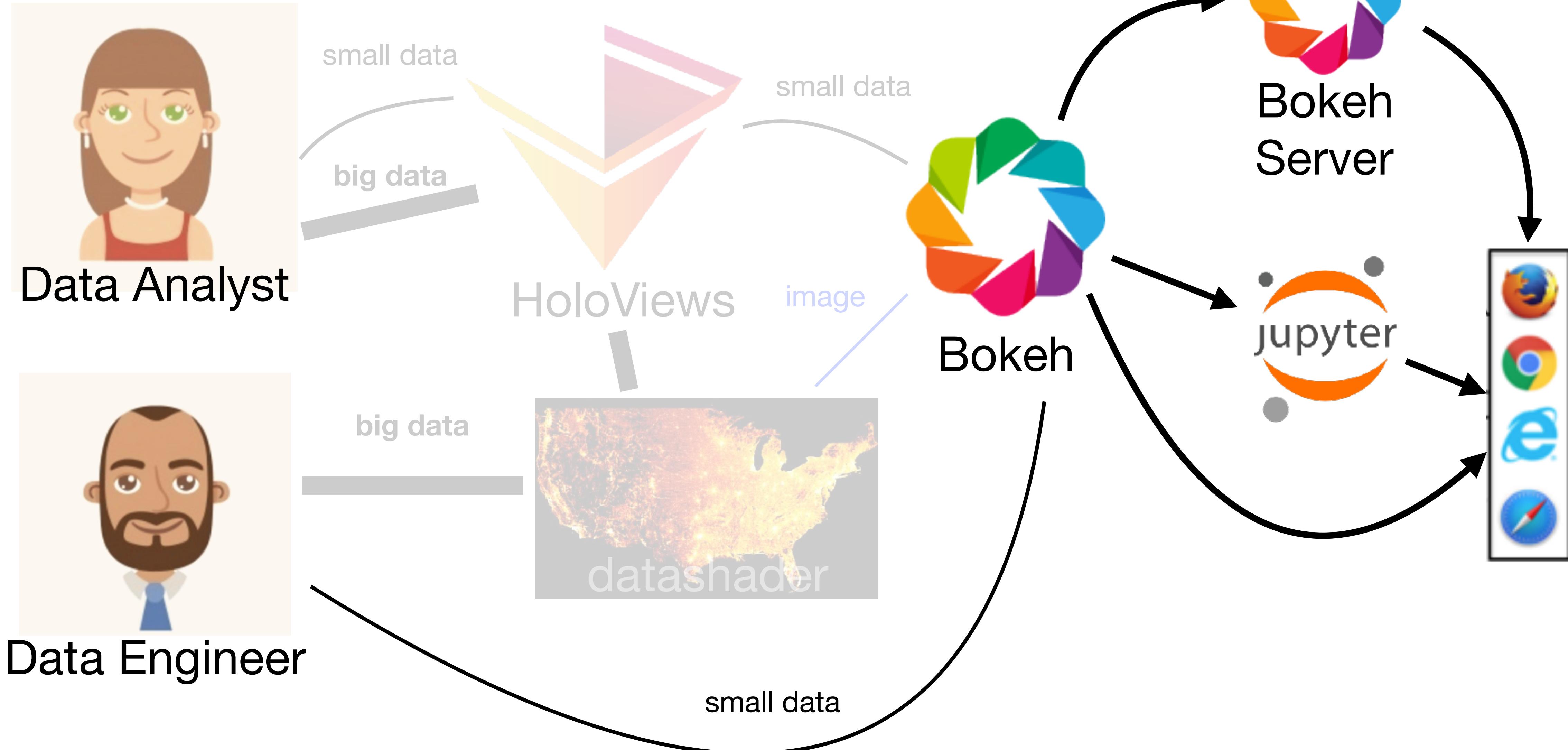
bokeh.models example



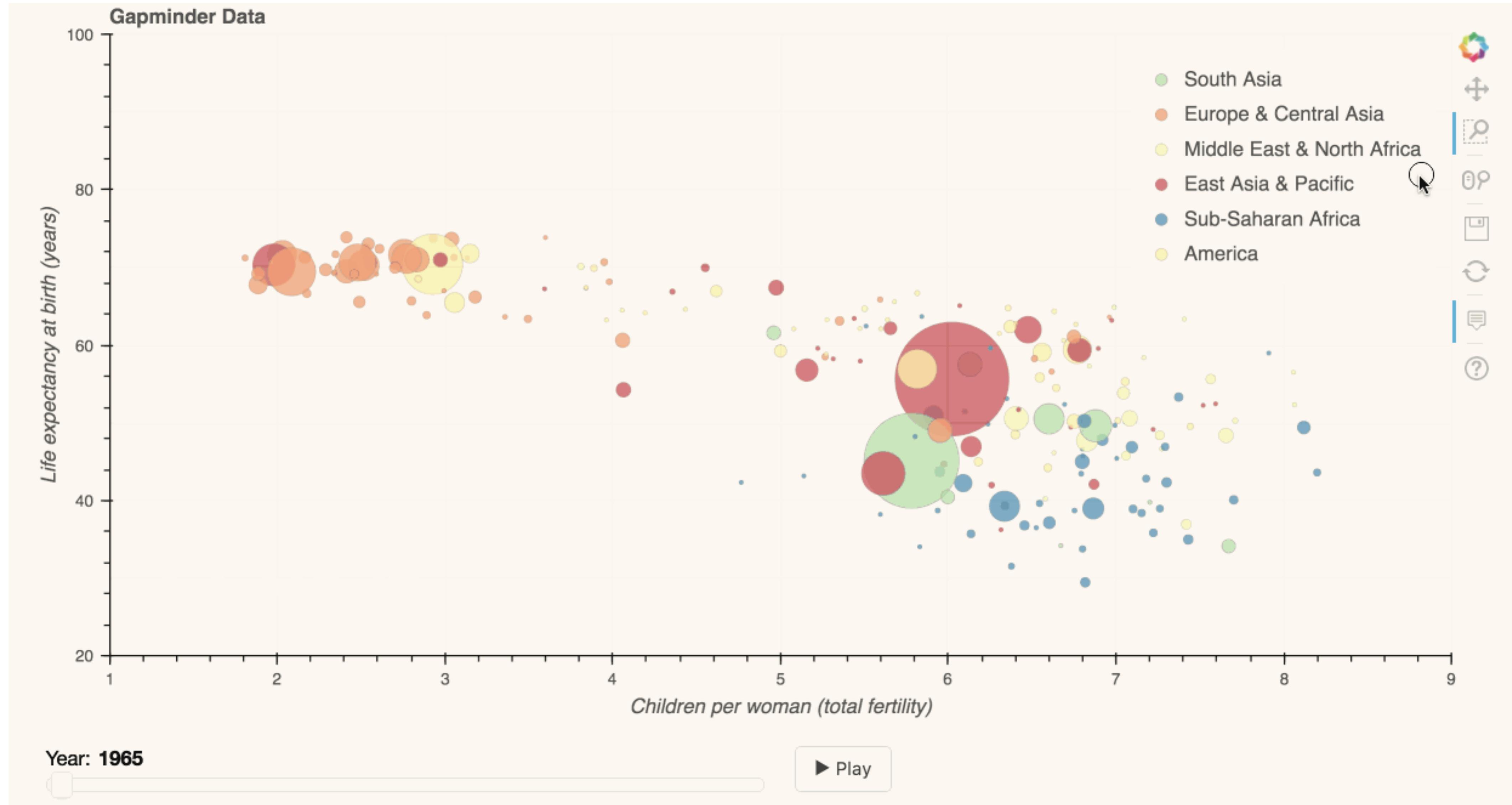


https://github.com/bokeh/datashader/blob/master/examples/holoviews_datashader.ipynb

Bokeh example



Bokeh example





Bokeh example

Step 1: 打開 Terminal

Step 2: 找一個合適的位置

```
conda create -n bokeh_streaming_app  
source activate bokeh_streaming_app  
mkdir bokeh_streaming_app  
cd bokeh_streaming_app/  
touch main.py  
vim main.py
```

按 i 開始編輯

將 main.py 的內容複製貼上

輸入 :wq 儲存並關閉

touch data.py

vim data.py

按 i 開始編輯

將 data.py 的內容複製貼上

輸入 :wq 儲存並關閉

Step 3:
bokeh serve .

Step 4: 在 Chrome 貼上連結位址

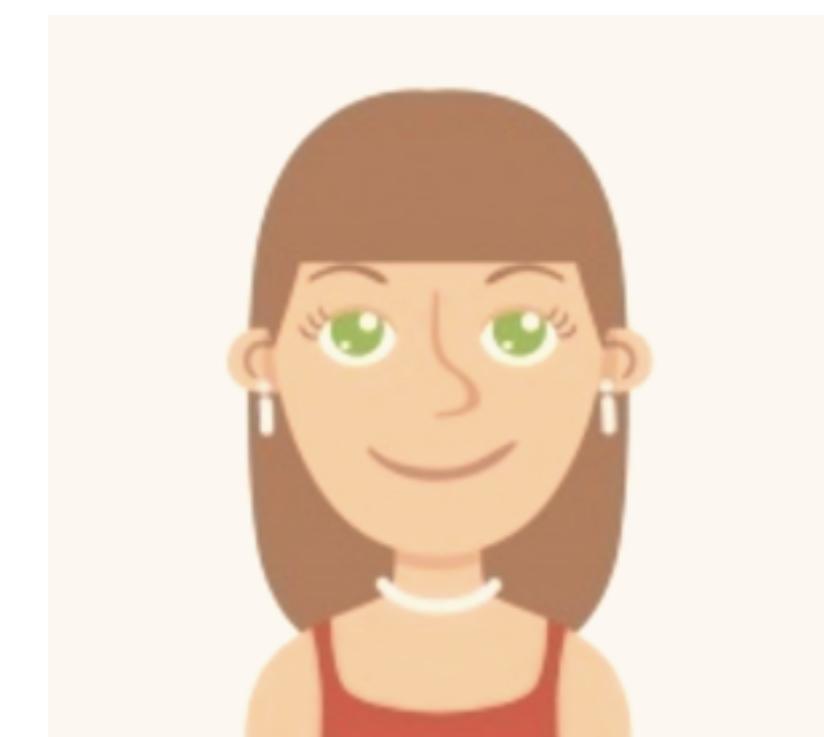
localhost:5006/bokeh_streaming_app

or

Step 3:

bokeh serve --show main.py

Bokeh + HoloViews example



Data Analyst

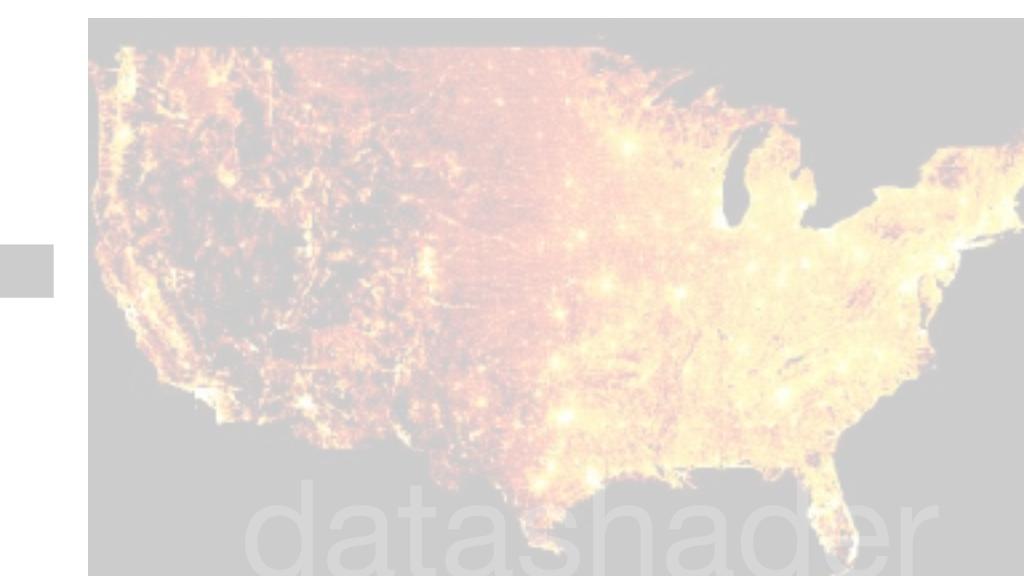


Data Engineer

small data

big data

HoloViews



small data

small data

image



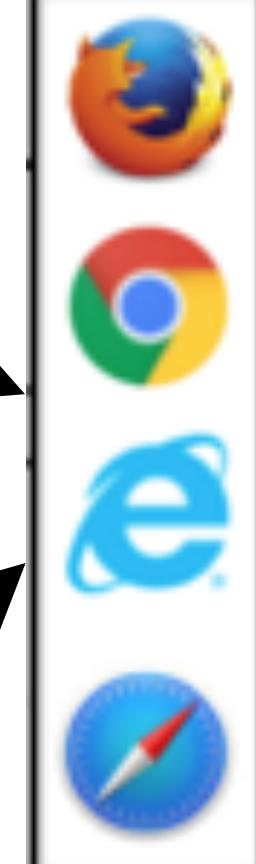
Bokeh



Bokeh
Server



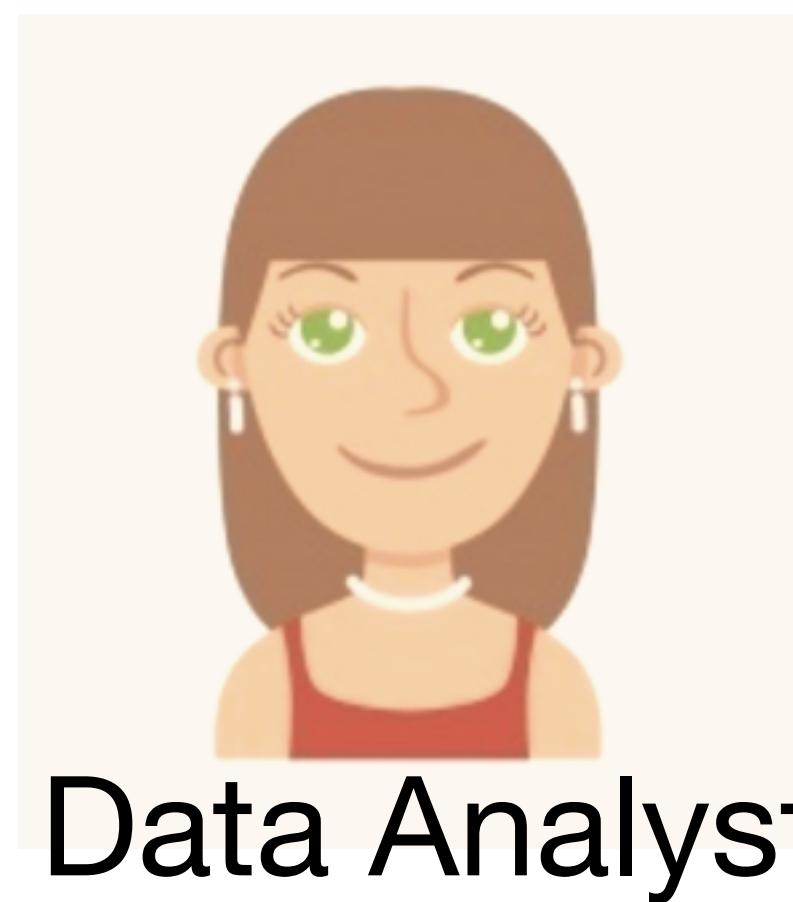
jupyter



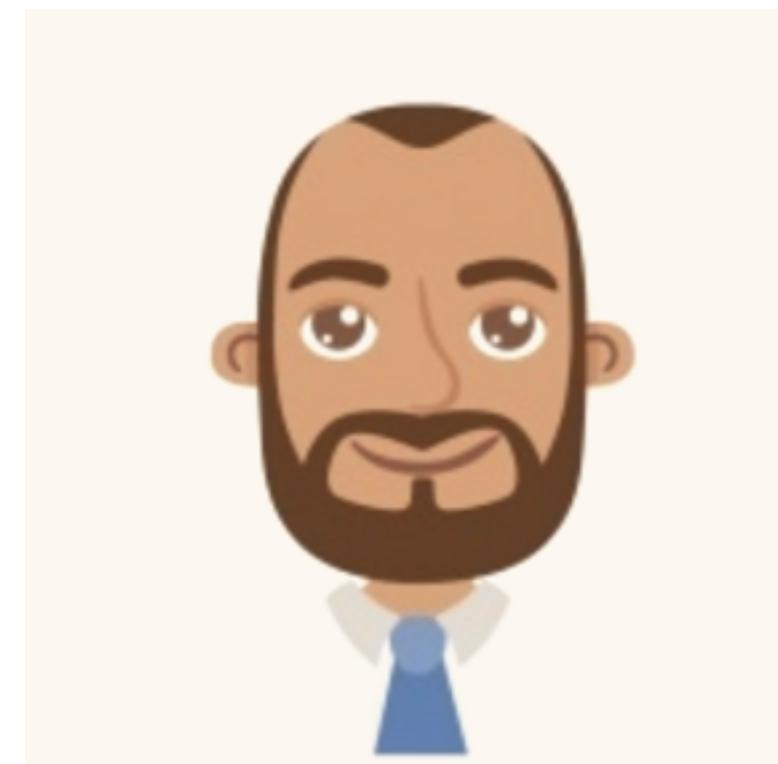
See you next sharing!



Bokeh + HoloViews + datashader example



Data Analyst



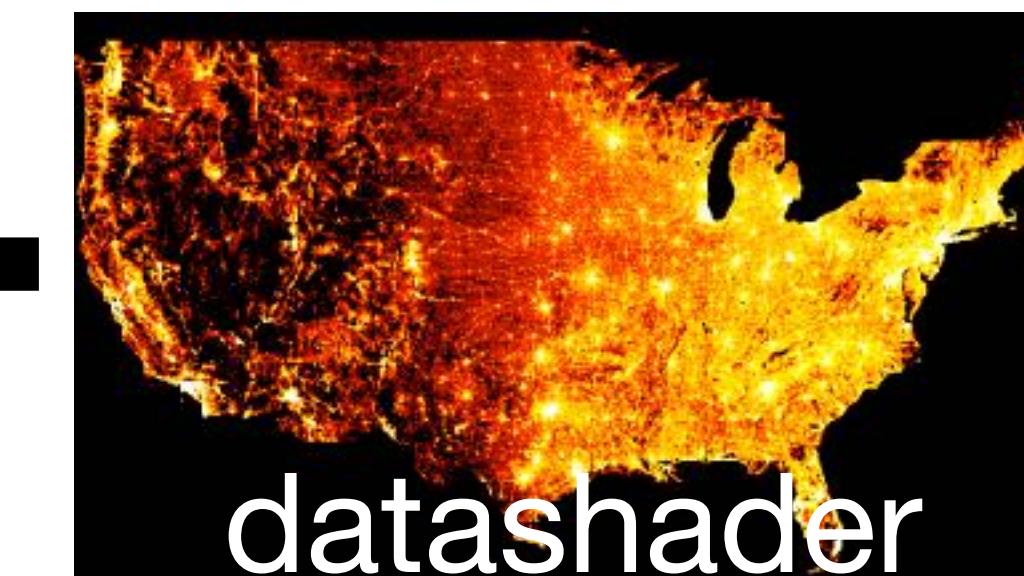
Data Engineer

small data

big data

HoloViews

big data



datashader

small data

image

Bokeh

small data



Bokeh
Server



jupyter



See you next sharing!

Appendix



HoloViews

**Stop plotting your data
- annotate your data and let it visualize itself.**

HoloViews 實際上不是一個繪圖庫，主要提供的互動功能是滾動式捲軸，將視覺化的過程變得簡單，目的是讓數據分析師能夠專注於探索數據和嘗試表達數據，而不是摸索繪圖過程。

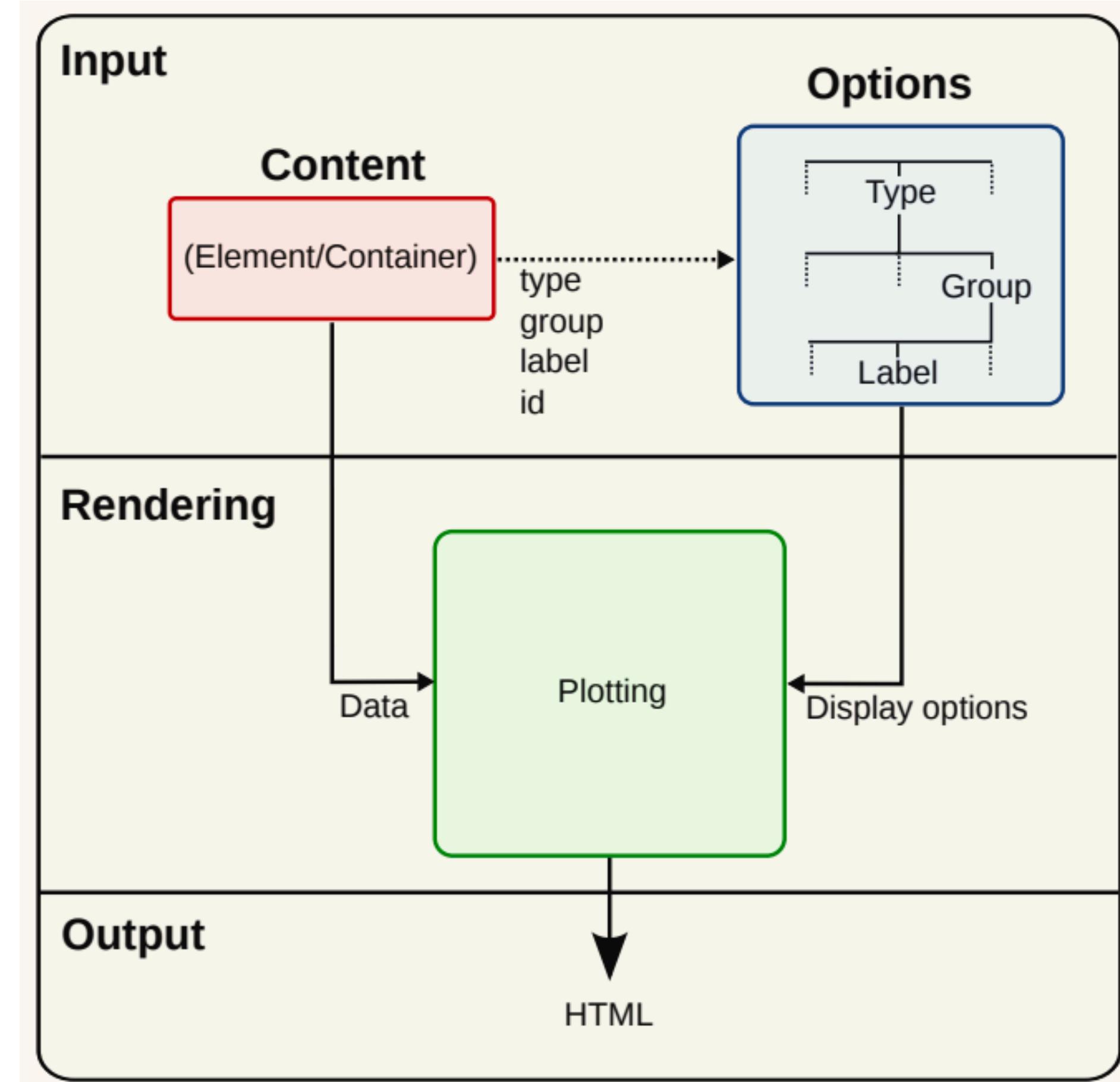


Fig. 3: This view of the HoloViews display and customization systems illustrates the complete separation between the content (data) to be displayed, the display options, and the rendering/plotting system. The display options are stored entirely separately from the content as a tree structure, with the appropriate options being selected with user-controllable levels of specificity: general options for all objects of a given type, more specific options controlled by user-definable group and label strings, or arbitrarily specific options based on the integer `id` assigned to each content object. Plotting and rendering happens automatically through the use of IPython display formatters. These combine the content with the specified display options, call an external plotting library, which returns an HTML representation that can then be rendered in the notebook.