Creating interactive browser visualizations with Bokeh

Feb 21, 2014

About Me

- Employee at Continuum, Analytics
- Open-source contributor (Bokeh, Chaco, NumPy)
- Scientific, financial, engineering domains using Python, C, C++, etc.
- Interactive Visualization of "Big Data"
- Background in Physics, Mathematics

About Continuum

- Founded in 2012 by Travis Oliphant and Peter Wang
- Headquartered in Austin, TX
- Products, consulting, training
  - "big data" analytics
  - scientific & high-performance computing
  - interactive visualization, dashboards, web apps
  - collaborative analysis

Visualization

Bokeh: Interactive, browser-based visualization for big data, driven from Python (and others!)

http://bokeh.pydata.org

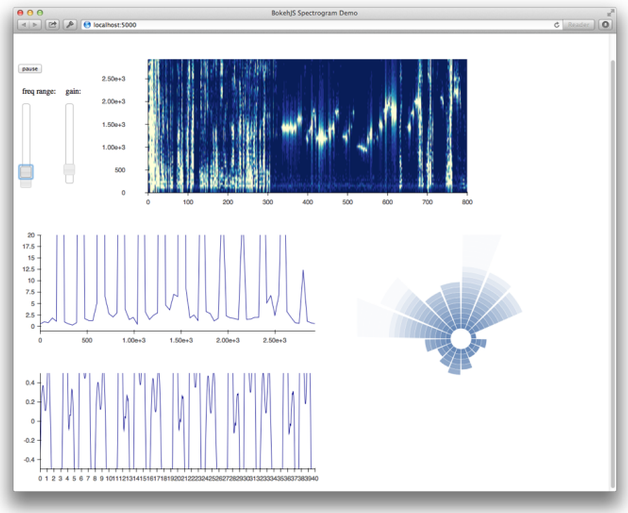
Bokeh

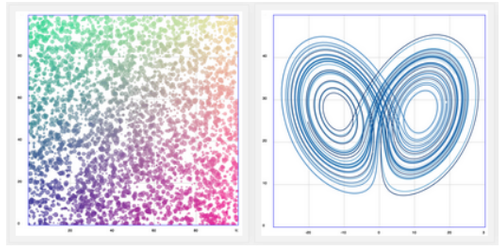
Interactive web viz without Javascript

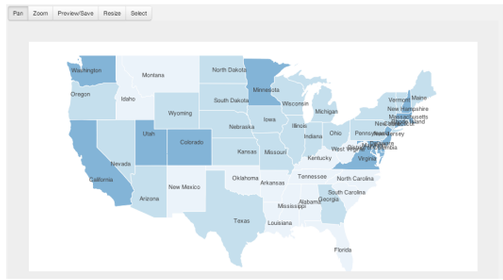
Object-oriented JS runtime library for dynamic, novel, interactive web graphics

Python interfaces to output static plots or drive live ones

Interop with IPython Notebook







CONTINUUM ANALYTICS

Bokeh

- Language-based (instead of GUI) visualization system
  - High-level expressions of data binding, statistical transforms, interactivity and linked data
  - Easy to learn, but expressive depth for power users

- Interactive
  - Data space configuration as well as data selection
  - Specified from high-level language constructs

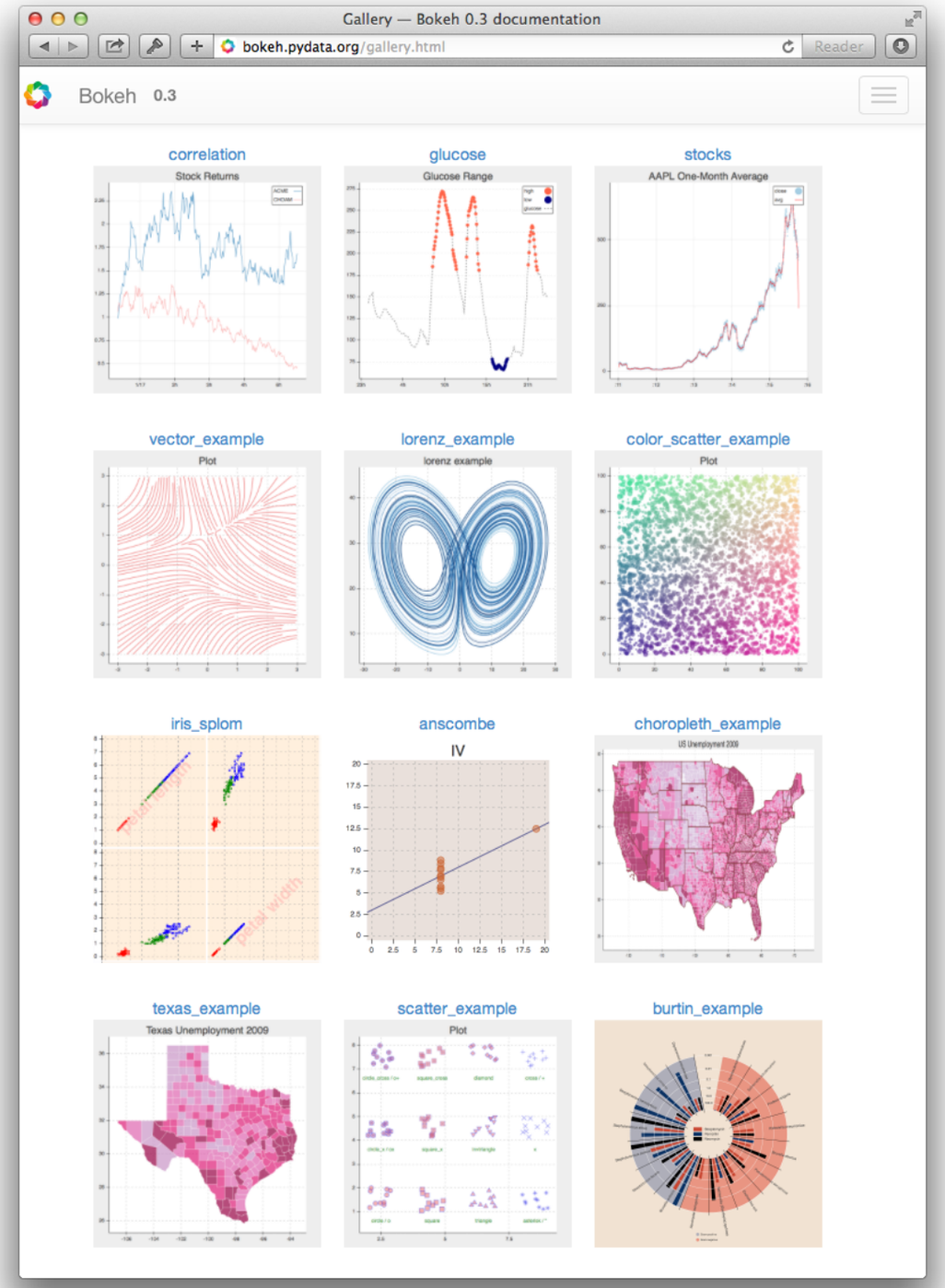
- Web as first class interface target

- Support for large datasets via intelligent downsampling ("abstract rendering")

Bokeh

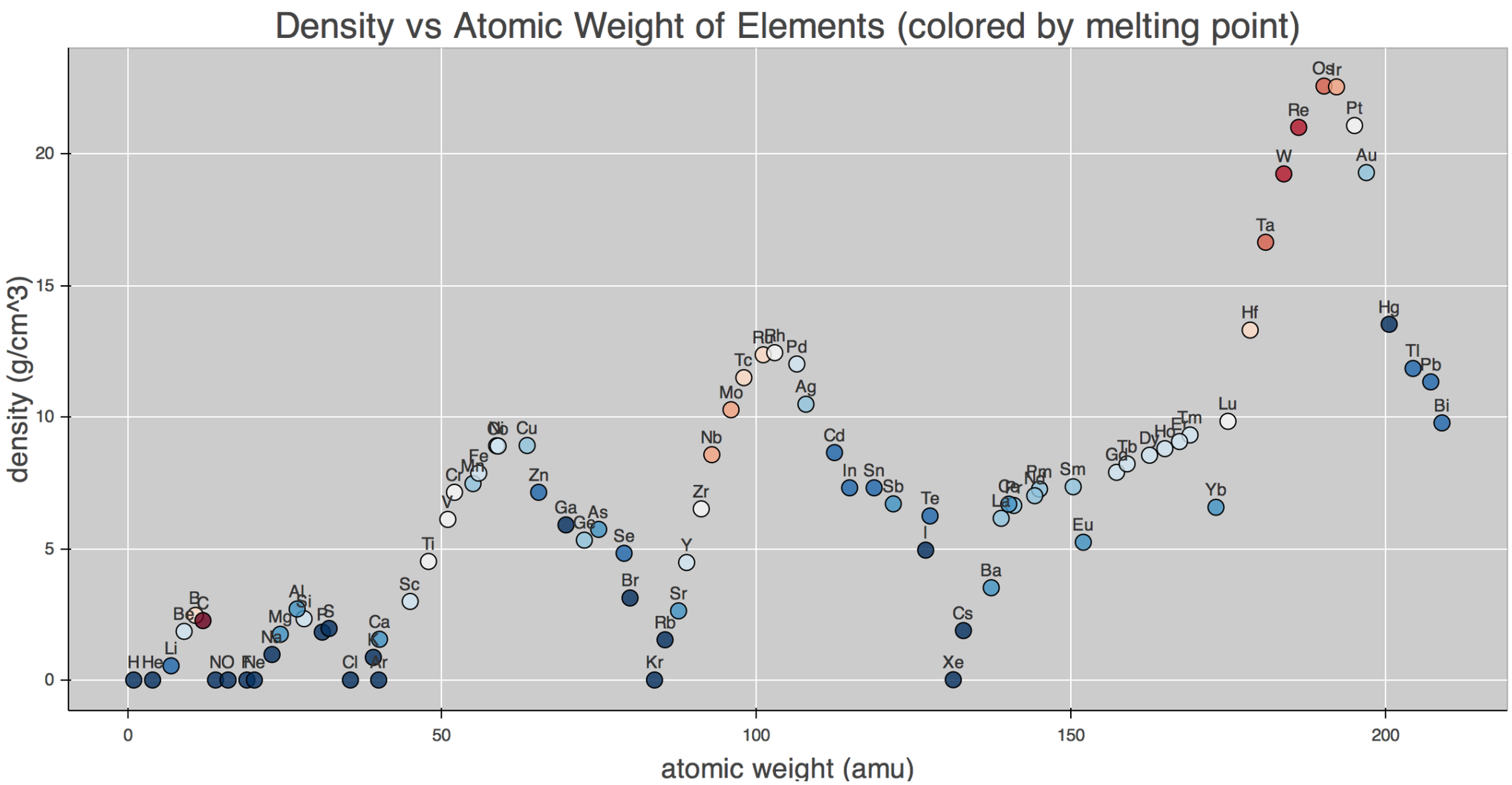
- Rich interactivity over large datasets
- HTML5 Canvas (faster than SVG)
- Handles realtime streaming and updating data
- Novel & custom visualizations
- Integration with Google Maps
- No need to learn Javascript - easy interfaces from Python & other langs

http://bokeh.pydata.org

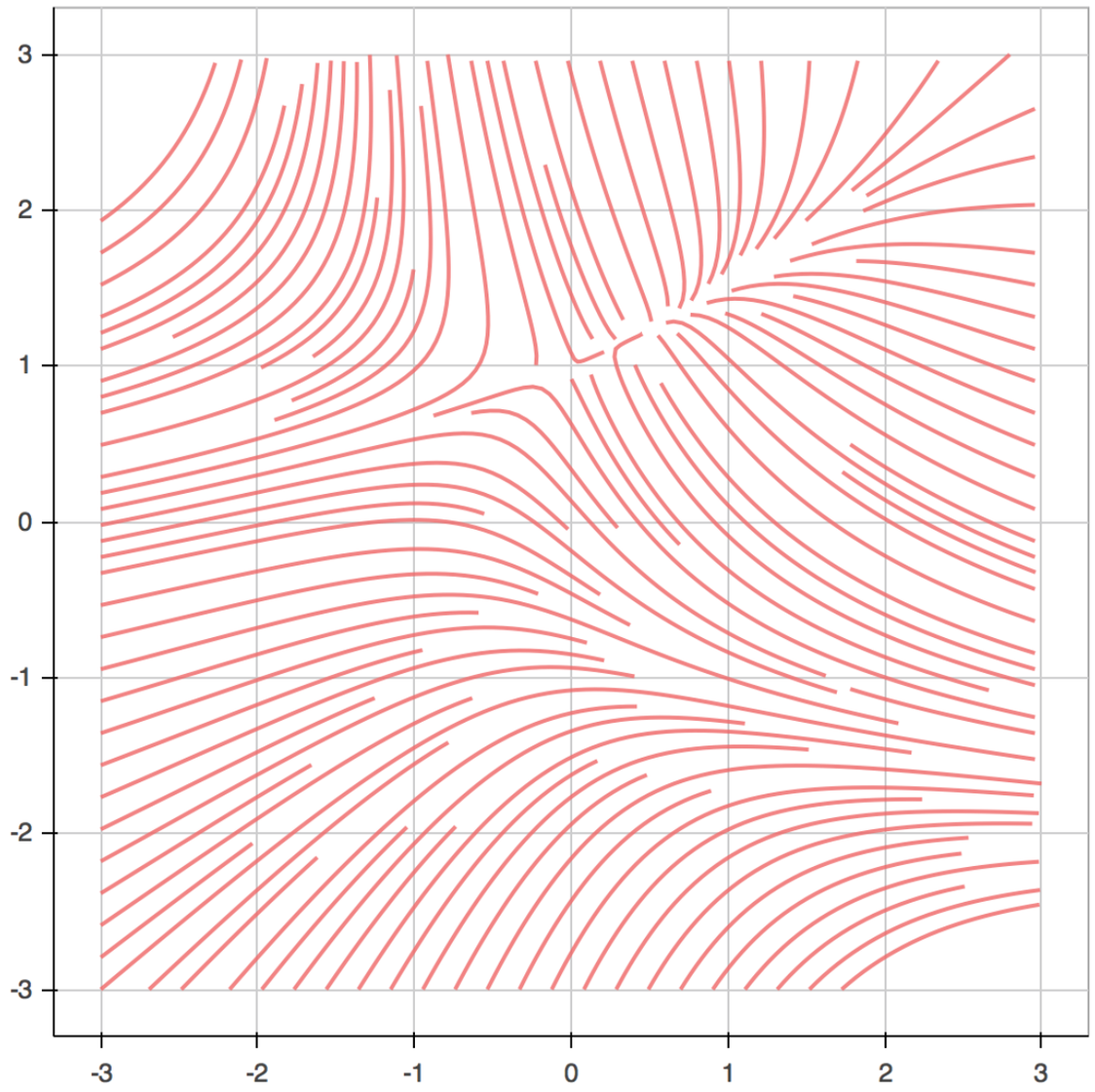


Bokeh Interface Concepts

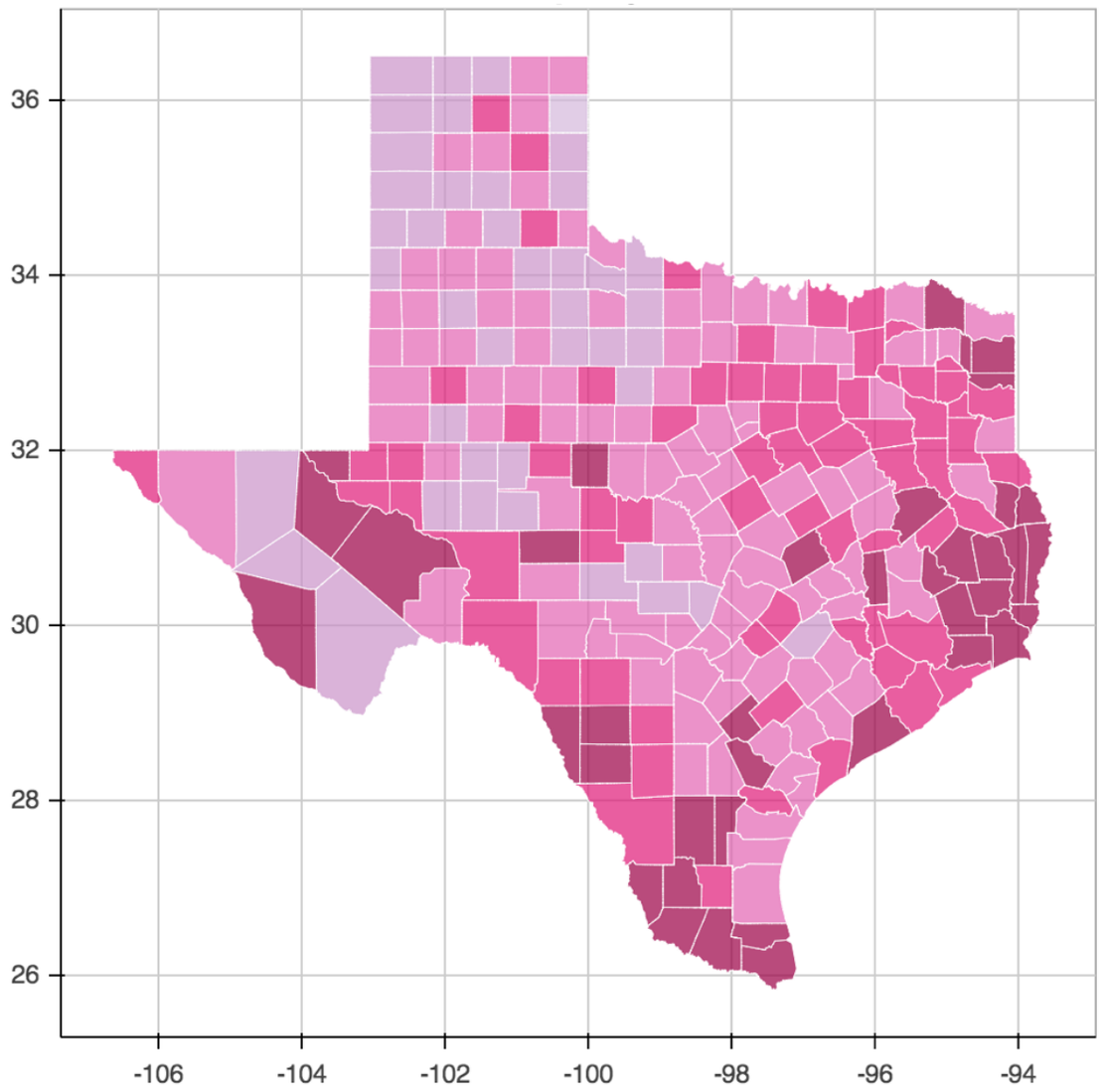
- Plots are based on glyphs
- All or almost all visual elements of a glyph can be attached to a vector of data.



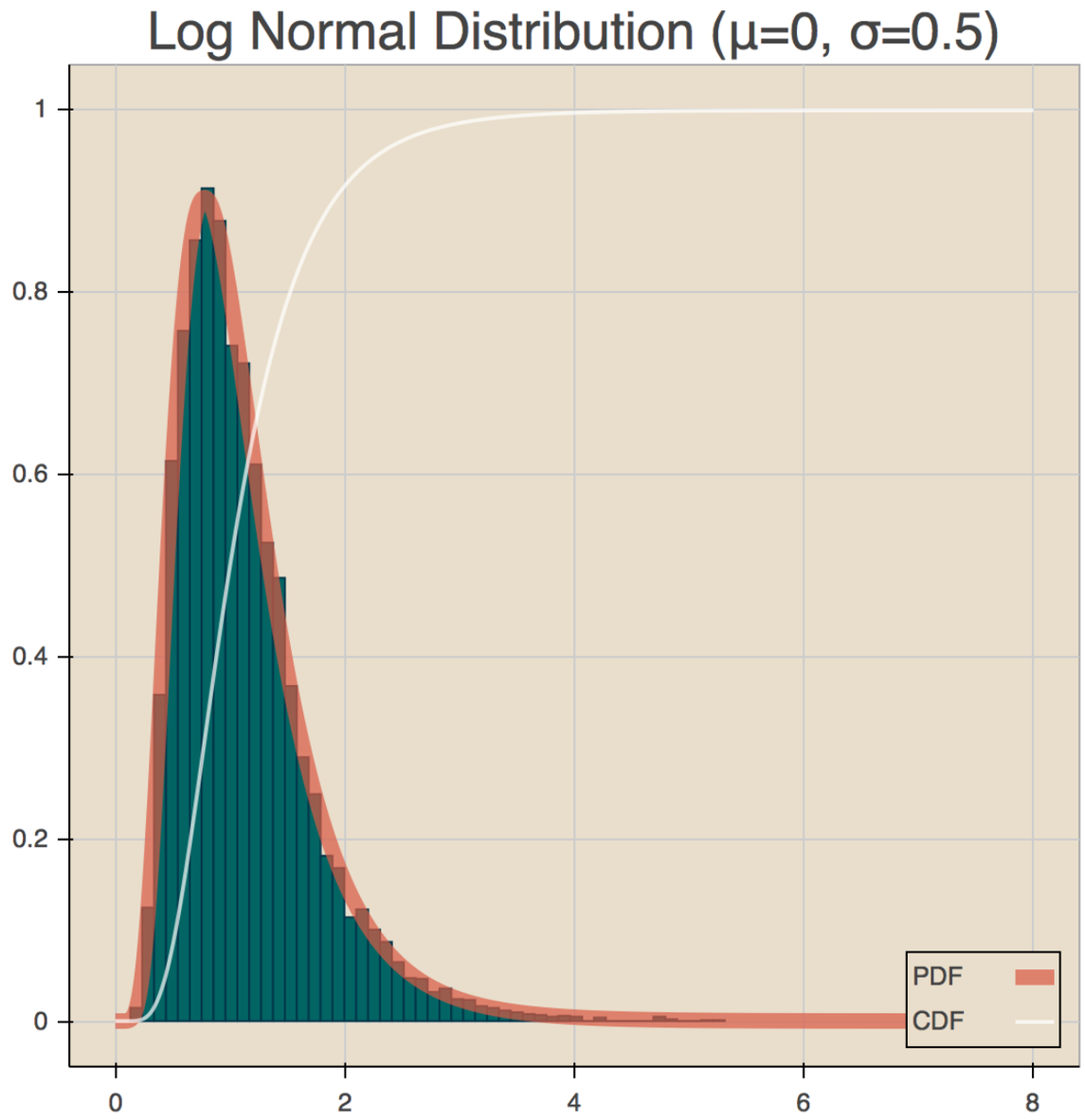

```
circle(elements['Atomic Mass'], elements['Density'] ,
       color=meltingpointcolors, plot_width=1200, line_color='black',fill_alpha=0.8,
       size=12, title='Density vs Atomic Weight of Elements (colored by melting point)',
       background_fill= '#cccccc', tools='pan, wheel_zoom, box_zoom, reset')

text(elements['Atomic Mass'], elements['Density'] +0.3,
    text=elements['Symbol'],angle=0, text_color='#333333',
    text_align="center", text_font_size="10pt")

xaxis().axis_label='atomic weight (amu)'
yaxis().axis_label='density (g/cm^3)'
grid().grid_line_color='white'

show()
```



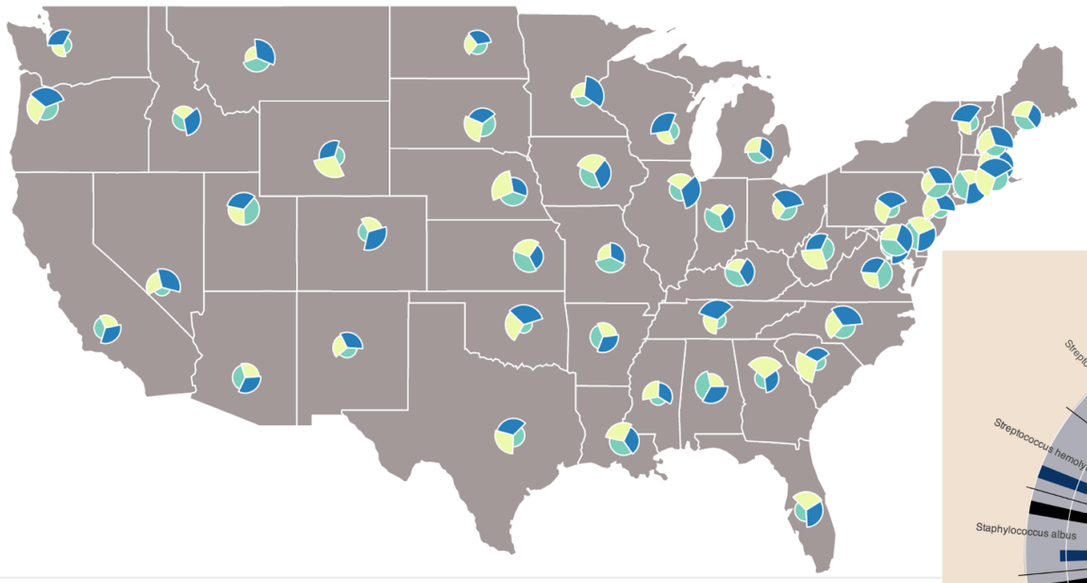

```
multi_line(xs, ys,
    line_color="#ee6666", line_width=2, line_alpha=0.8,
    name="vector example", tools="pan,wheel_zoom,box_zoom,reset,previewsave"
)

show()  # open a browser
```

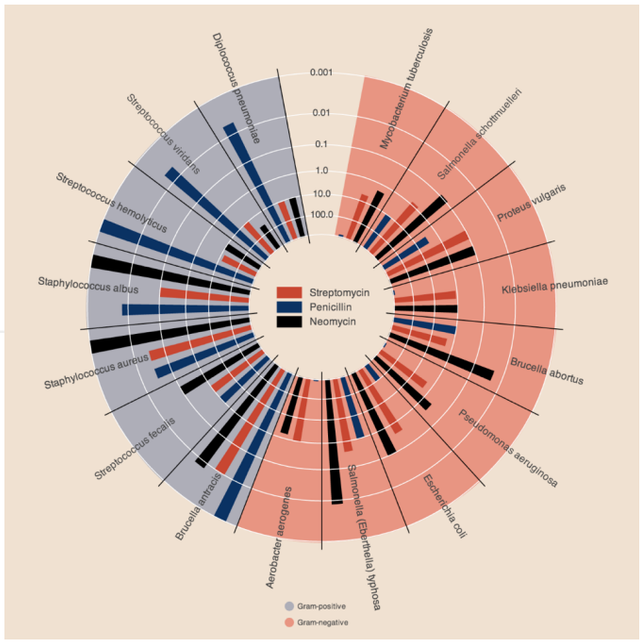


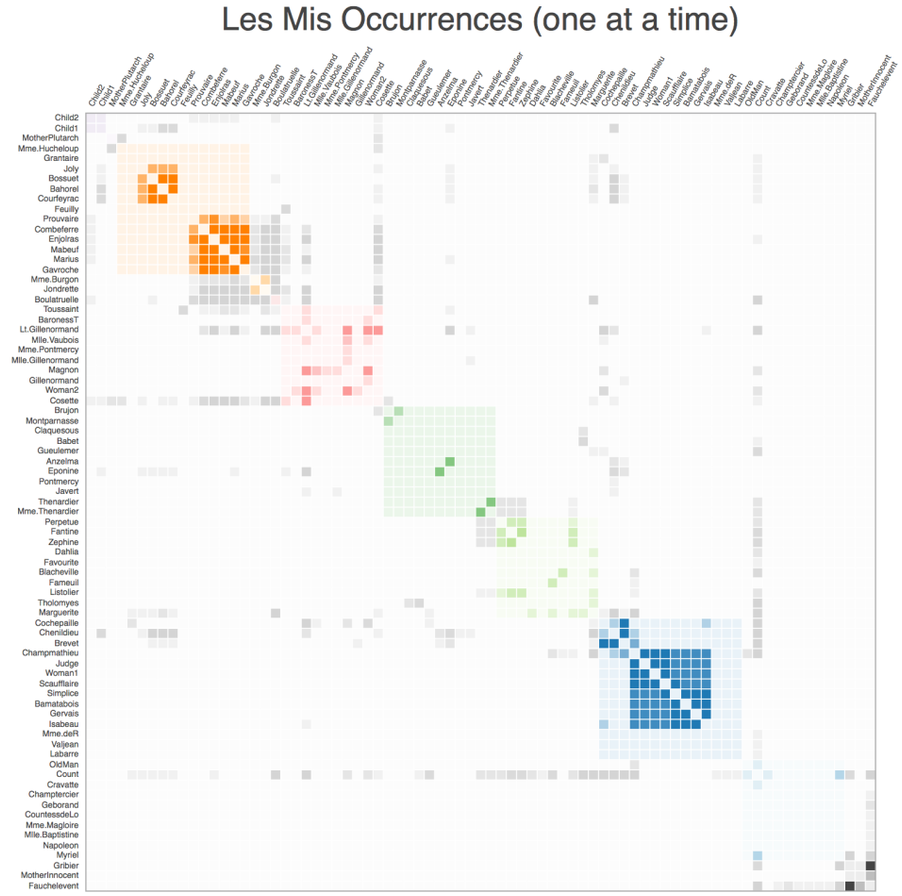

```
output_file("texas.html", title="texas.py example")

patches(county_xs, county_ys, fill_color=county_colors, fill_alpha=0.7,
        line_color="white", line_width=0.5, title="Texas Unemployment 2009")

show()
```

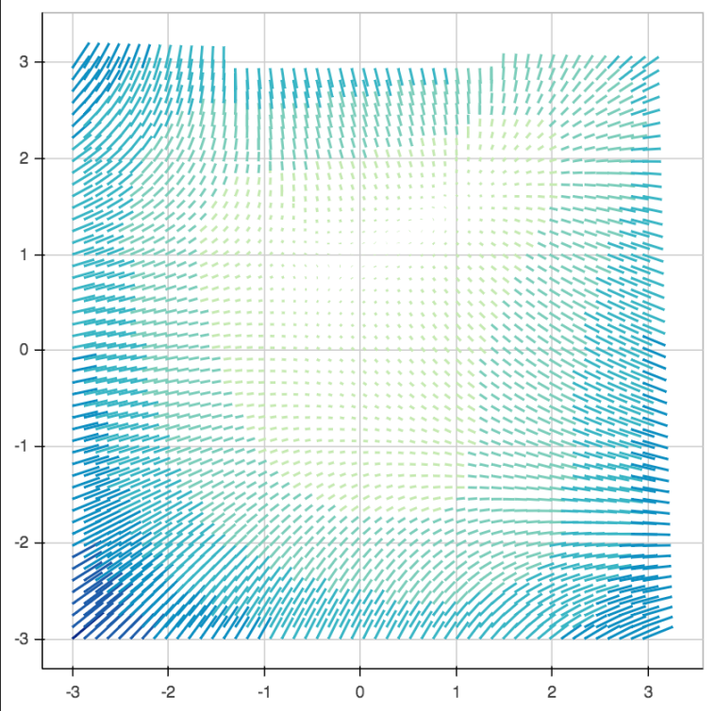


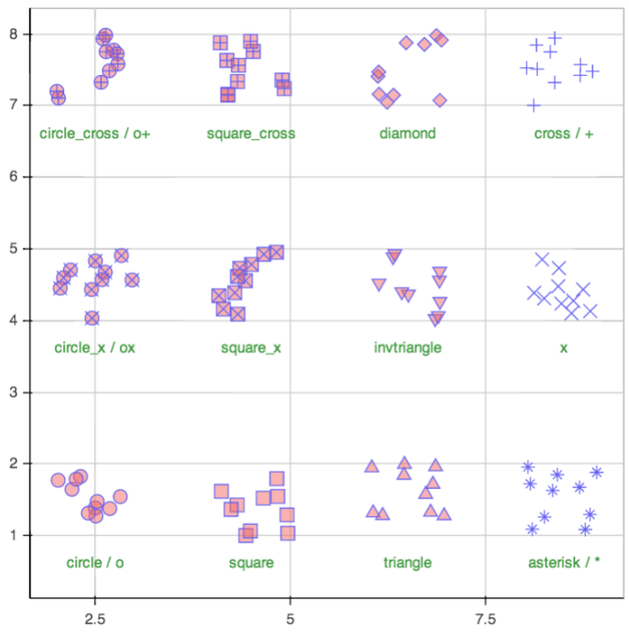

```
quad(top=hist, bottom=np.zeros(len(hist)), left=edges[:-1], right=edges[1:],
     fill_color="#036564", line_color="#033649", background_fill="#E8DDCB",
     title="Log Normal Distribution (μ=0, σ=0.5)", tools="")
line(x, pdf, line_color="#D95B43", line_width=8, alpha=0.7, legend="PDF")
line(x, cdf, line_color="white", line_width=2, alpha=0.7, legend="CDF")
legend().orientation = "bottom_right"
```

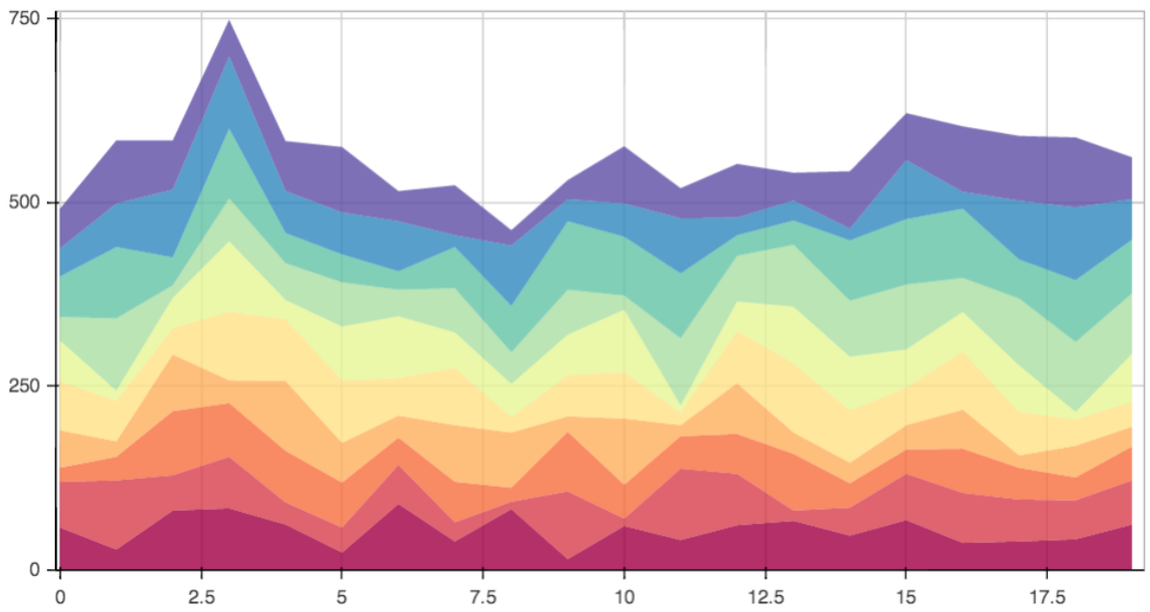




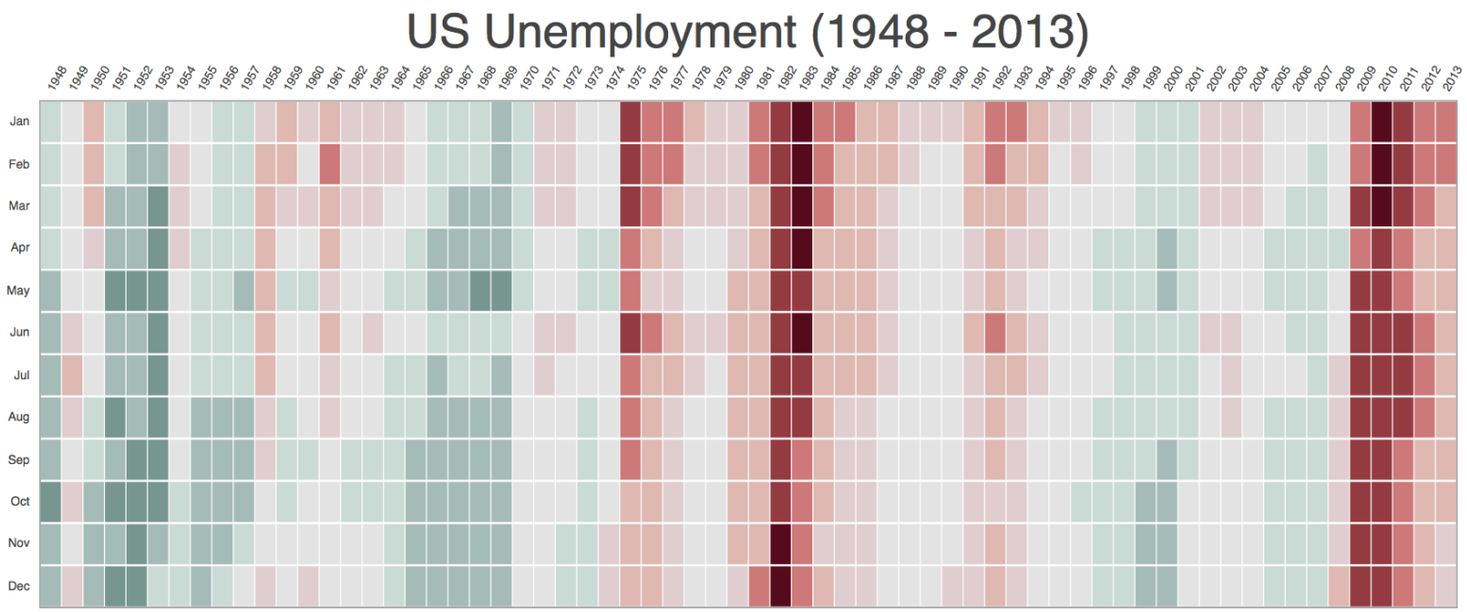


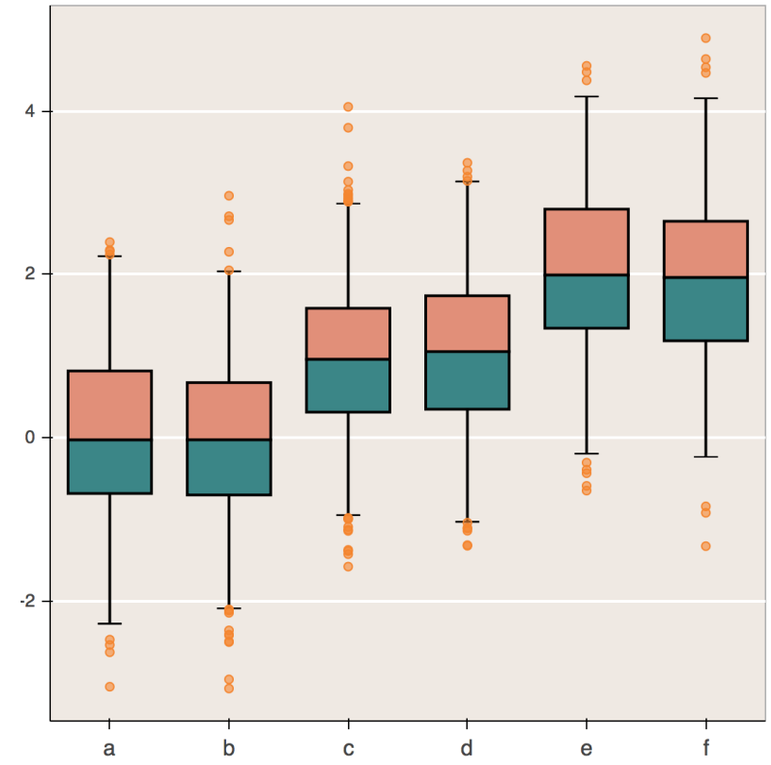


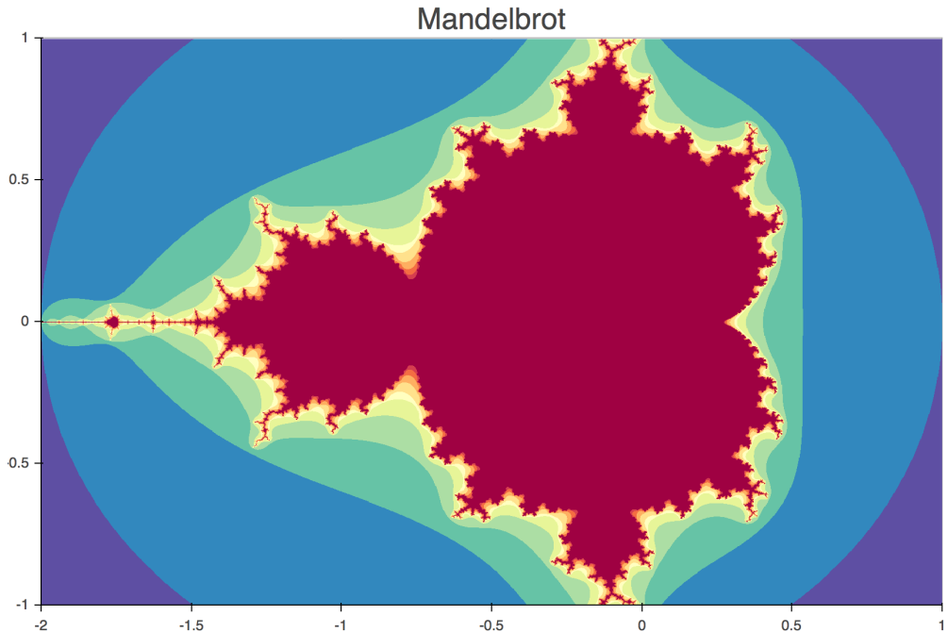


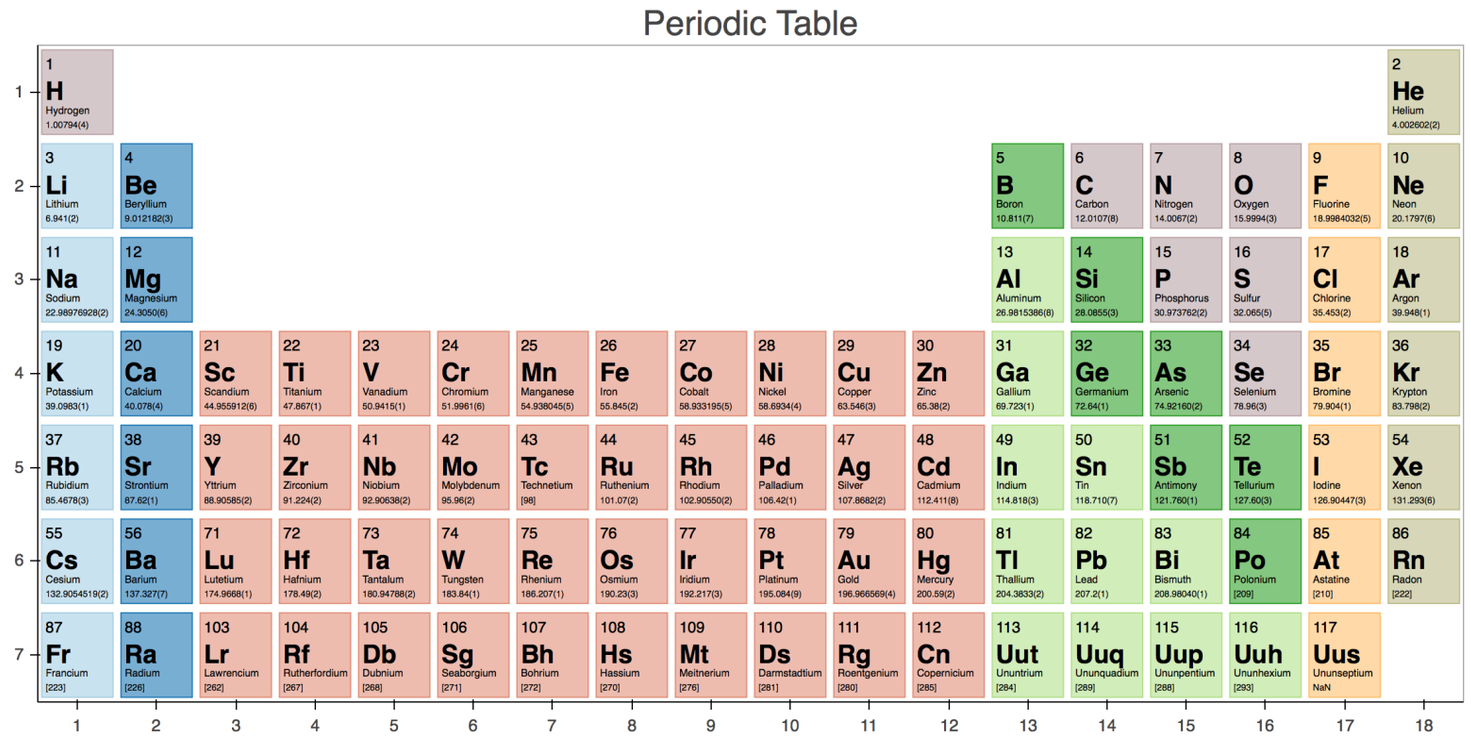


CONTINUUM ANALYTICS









CONTINUUM ANALYTICS

Coming soon

- Abstract Rendering — dynamic downsampling and data shading for millions of points
- Contraints based layout system
- Interactive tool improvements and additional tools
- Matplotlib compatibility — use Bokeh from pandas, ggplot.py, Seaborn
- Language bindings — Scala underway, more later
- Widget interactors and plugins

But don't forget

- Usability improvements
  - Discoverable parameters
  - Informative error messaging
  - Expanded live gallery
  - "Do the right thing" when it is possible
  - expose capability when it's not

Need feedback from users (you!)

More information and Contributing

Public Github repos

- https://github.com/ContinuumIO/bokeh
- https://github.com/JosephCottam/AbstractRendering

Videos

- Python & the Future of Data Analysis
- Bokeh Workshop

Blogs

- http://continuum.io/blog/index
- http://continuum.io/blog/painless_streaming_plots_w_bokeh
- http://continuum.io/blog/realtime-analytics-twitter