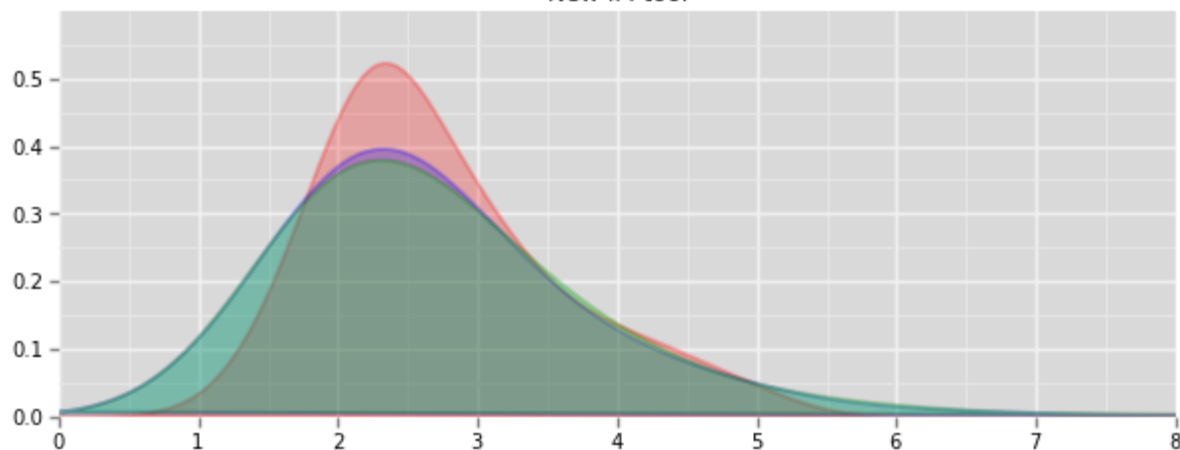
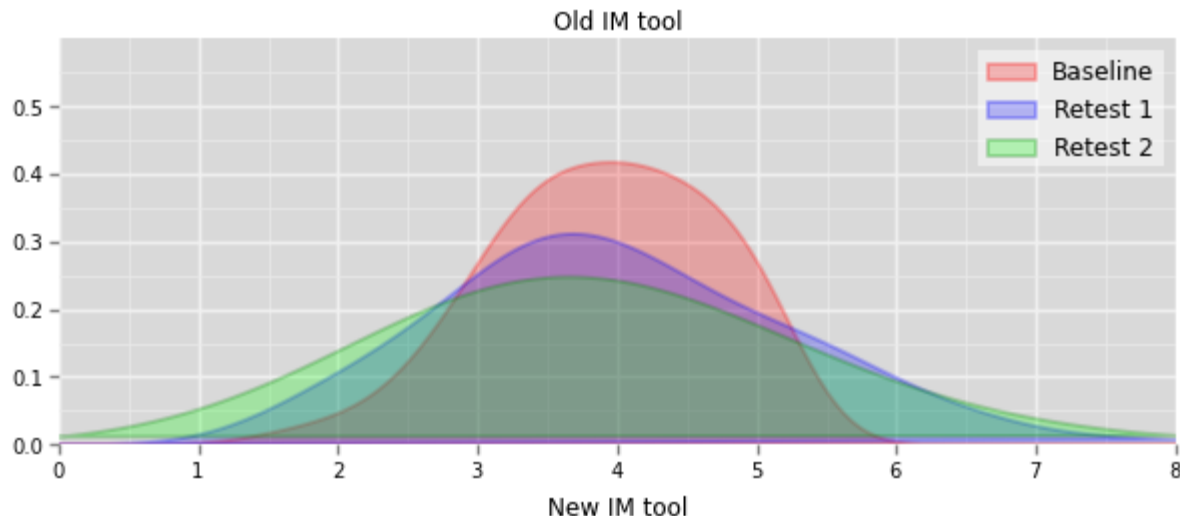


Scientific Plotting in Python

Jack Parmer
CEO, Plotly
BS Engineering Physics
Stanford University

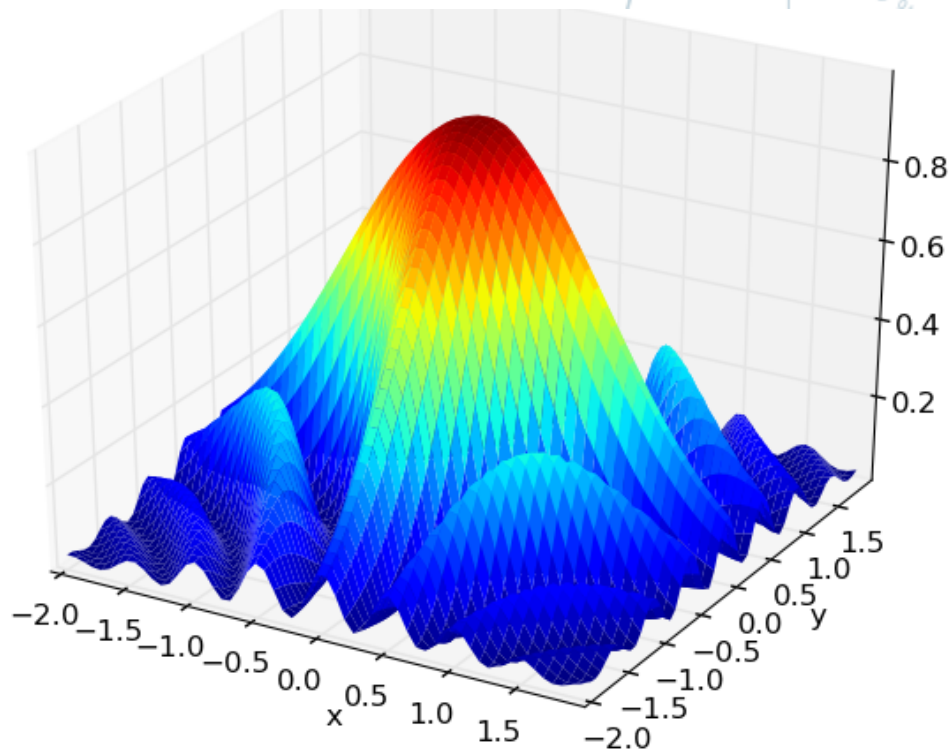




Developed over
10+ years.

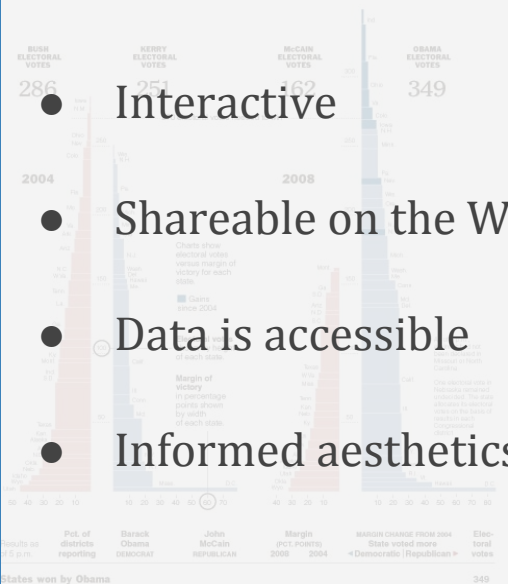
Cornerstone of scientific
graphing in Python.

Open-source.



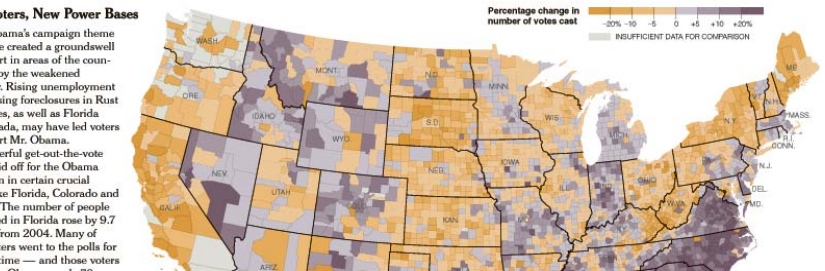
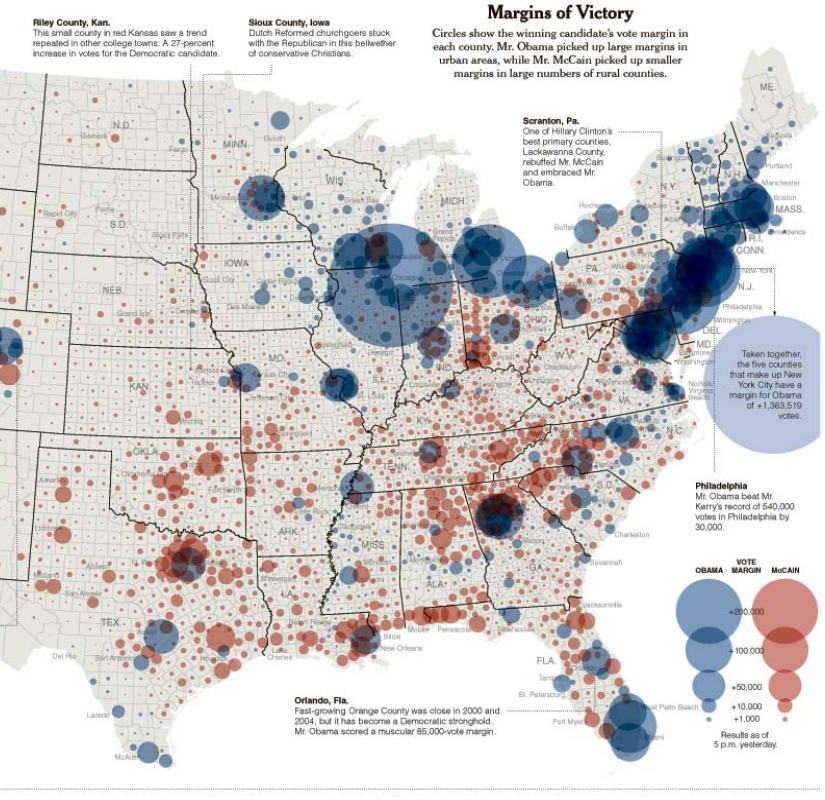
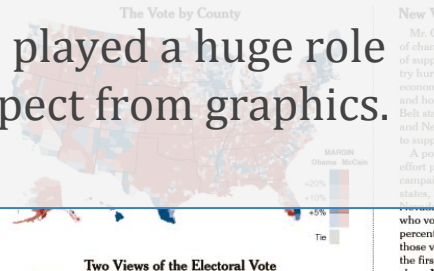
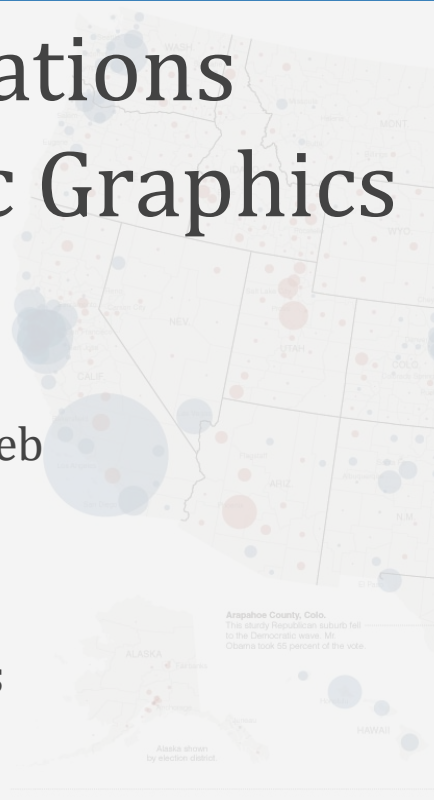
In a Decisive Victory, Obama New Expectations for Scientific Graphics

- Interactive
- Shareable on the Web
- Data is accessible
- Informed aesthetics



The Web and D3.js have played a huge role in what scientists can expect from graphics.

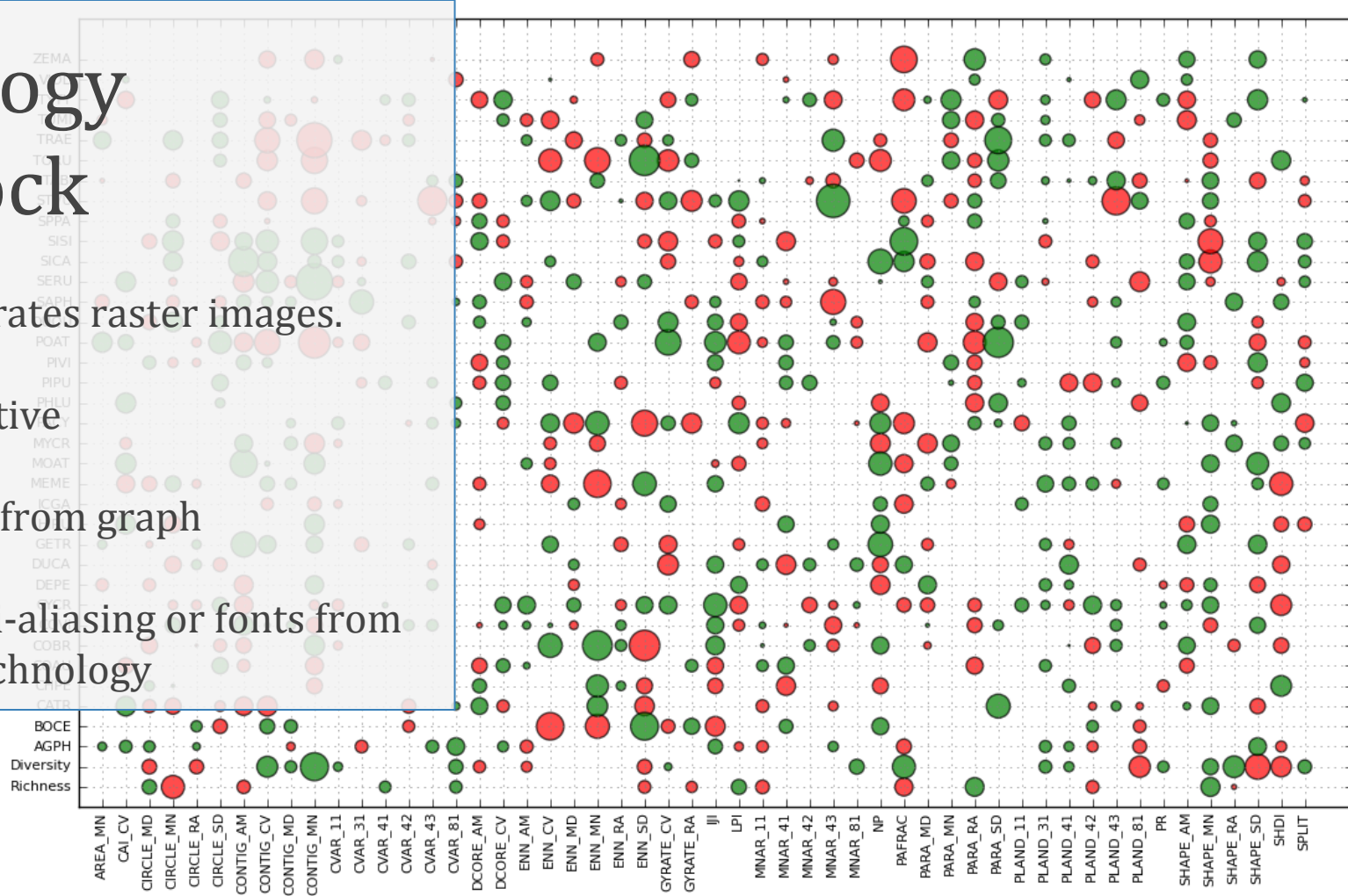
State	Obama	McCain	Margin
Alaska	63.0	36.9	+26.1
Arizona	39.1	60.8	-21.7
Arkansas	35.2	64.7	-29.5
California	68.4	31.5	+36.9
Colorado	53.3	46.6	+6.7
Connecticut	67.2	32.7	+34.5
Delaware	71.8	28.1	+43.7
Florida	51.3	48.6	+2.7
Georgia	38.6	61.3	-22.7
Idaho	32.9	67.0	-34.1
Illinois	62.0	37.9	+24.1
Indiana	35.3	64.6	-29.3
Iowa	67.0	32.9	+34.1
Kansas	37.6	62.3	-24.7
Kentucky	35.1	64.8	-29.7
Louisiana	34.4	65.5	-31.1
Maine	68.8	31.1	+37.7
Maryland	70.1	29.8	+40.3
Massachusetts	68.0	31.9	+36.1
Michigan	52.1	47.8	+4.3
Minnesota	60.9	38.9	+22.0
Mississippi	31.9	67.9	-36.0
Missouri	41.7	58.2	-16.5
Montana	52.0	47.9	+4.1
Nebraska	37.1	62.8	-25.7
Nevada	31.6	68.3	-36.7
New Hampshire	70.8	29.1	+41.7
New Jersey	68.4	31.5	+36.9
New Mexico	31.2	68.7	-37.5
New York	68.0	31.9	+36.1
North Carolina	35.8	64.1	-28.3
North Dakota	35.1	64.8	-29.7
Ohio	37.6	62.3	-24.7
Oklahoma	31.2	68.7	-37.5
Oregon	68.0	31.9	+36.1
Pennsylvania	60.3	39.6	+20.7
Rhode Island	68.0	31.9	+36.1
South Carolina	35.8	64.1	-28.3
South Dakota	35.1	64.8	-29.7
Tennessee	34.4	65.5	-31.1
Texas	35.1	64.8	-29.7
Utah	31.2	68.7	-37.5
Vermont	68.8	31.1	+37.7
Virginia	53.3	46.6	+6.7
Washington	68.0	31.9	+36.1
West Virginia	31.2	68.7	-37.5
Wisconsin	60.9	38.9	+22.0
Wyoming	35.1	64.8	-29.7



Technology Roadblock

Matplotlib generates raster images.

- Not interactive
- Data is lost from graph
- No free anti-aliasing or fonts from browser technology



New Project - mplexporter

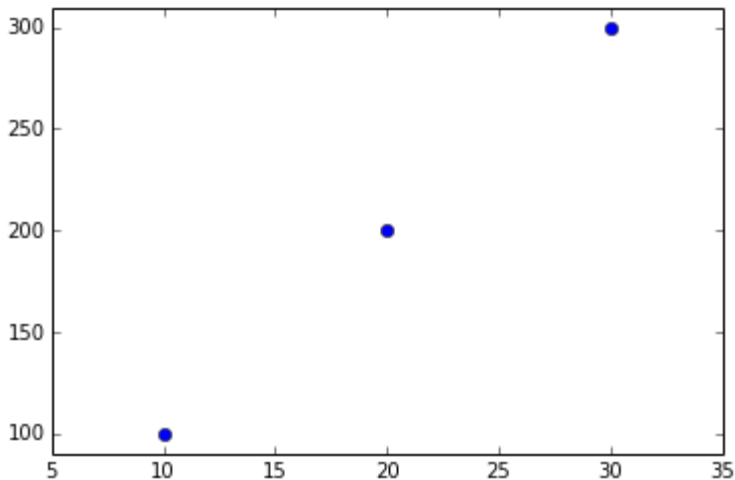
Extract matplotlib “graph DNA” to a universal JSON format, that anyone can read.

IPython Notebook:

<http://nbviewer.ipython.org/github/plotly/IPython-plotly/blob/master/See%20more/Plotly%20and%20mpld3.ipynb>

GitHub Repo:

<https://github.com/mpld3/matplotliblylib>

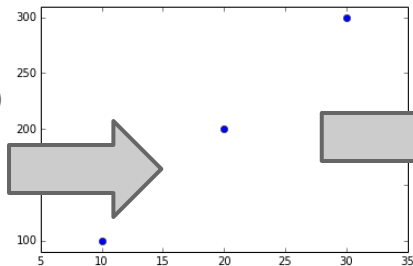


mplexporter

```
[{'marker': {'color': '#0000FF',  
  'line': {'color': '#000000', 'width': 0.5},  
  'opacity': 1,  
  'symbol': 'dot'},  
  'mode': 'markers',  
  'x': [10.0, 20.0, 30.0],  
  'y': [100.0, 200.0, 300.0]}]
```

Improving Data Liquidity

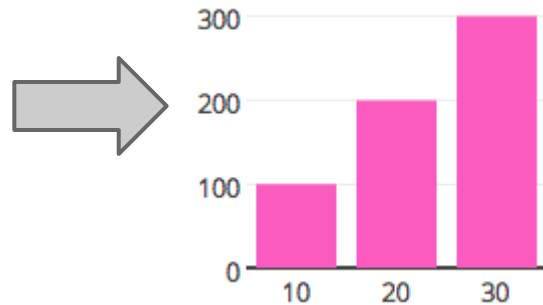
```
fig, ax = plt.subplots()  
ax.plot([10, 20, 30],  
        [100, 200, 300], 'o')
```



Resulting
matplotlib-specific
figure

```
[ {  
  'x': [10, 20, 30],  
  'y': [100, 200, 300]  
} ]
```

Neutral JSON figure
representation



New figure representations
based on neutral JSON
representations

*Goal is to make data transfer bidirectional - Data can be put in *and* taken out of figures.*

Plans to expand Matplotlib so that all figure objects include and expose their data.

SVG and Web Graphics are Incredible

Several projects are leveraging rich web graphics technology for plotting with Python

- **mpld3** (Using SVG / d3.js)
- <http://mpld3.github.io/>
- **Plotly** (Using SVG / d3.js)
- <http://plot.ly/python/>
- **Bokeh** (Using HTML5 Canvas)
- <http://bokeh.pydata.org/>

ROUND 1

ROUND 2

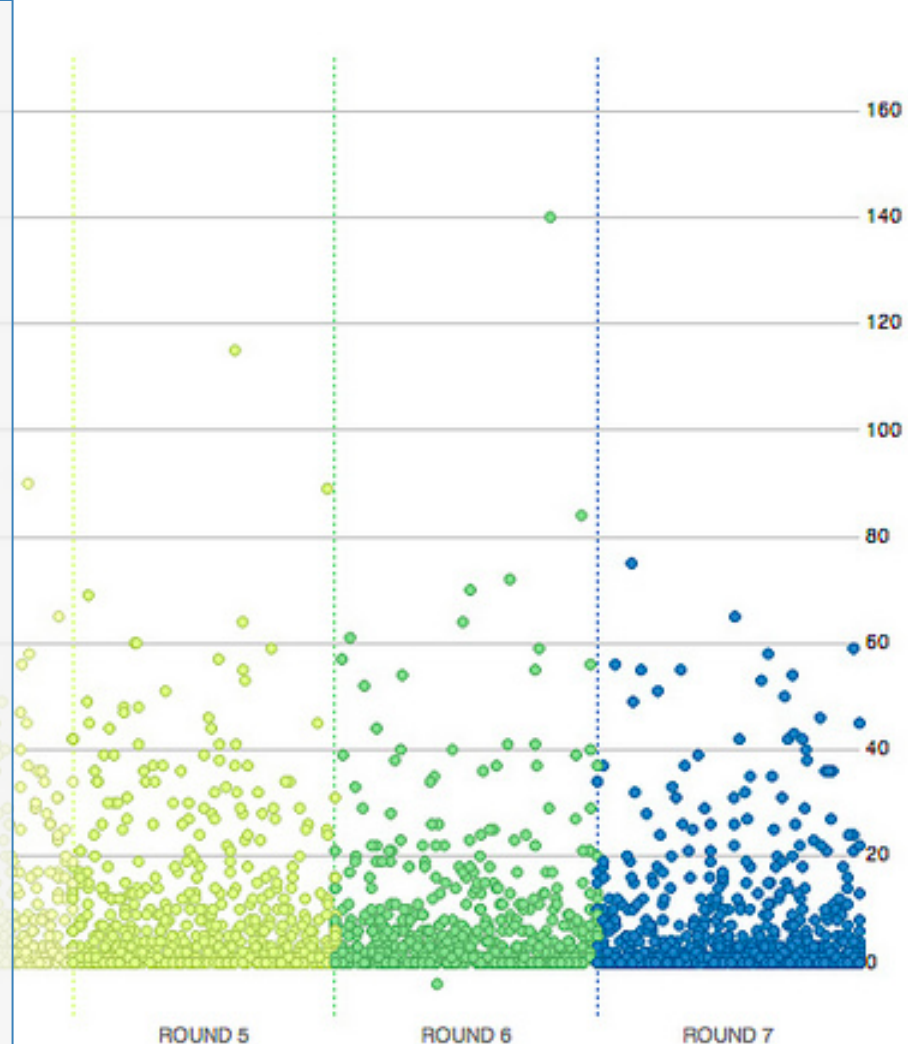
ROUND 3

ROUND 4

ROUND 5

ROUND 6

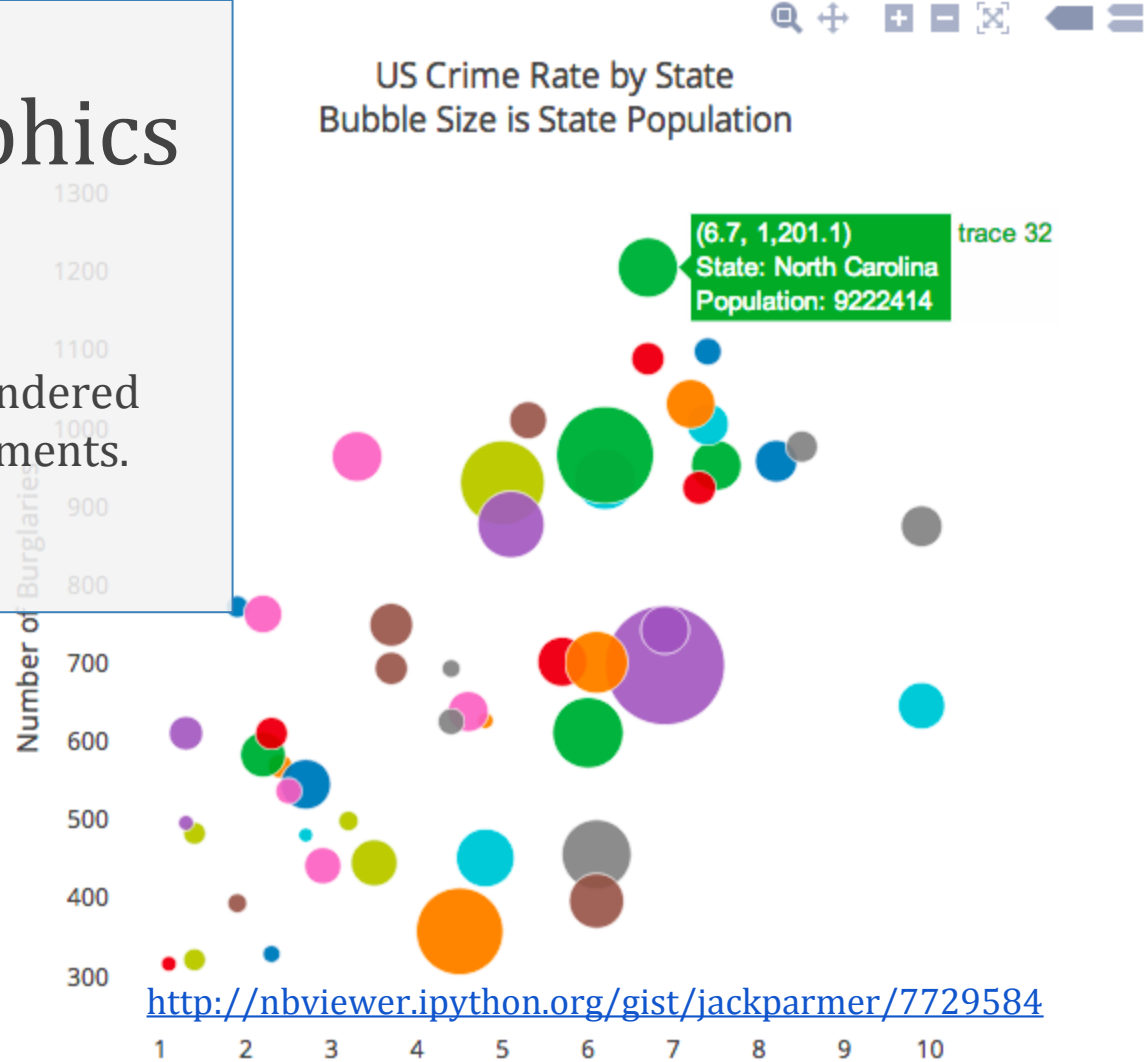
ROUND 7



SVG and Web Graphics are Interactive

Using JavaScript, scientific figures rendered with Python can have interactive elements.

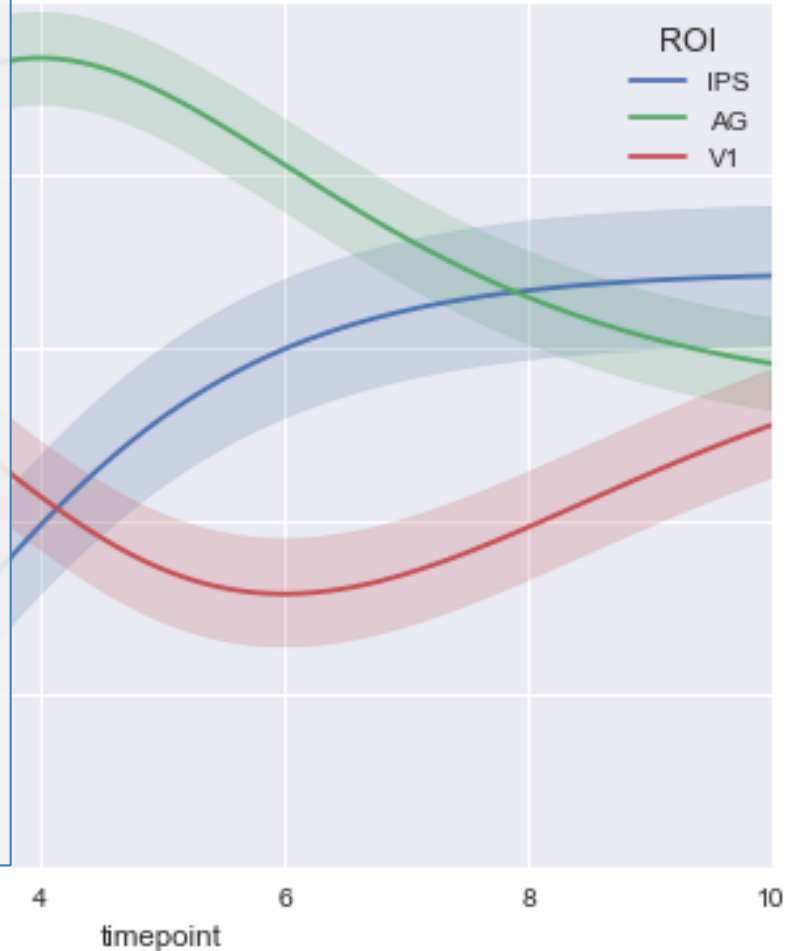
Zoom, pan, hover...



Aesthetic Layering for matplotlib

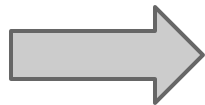
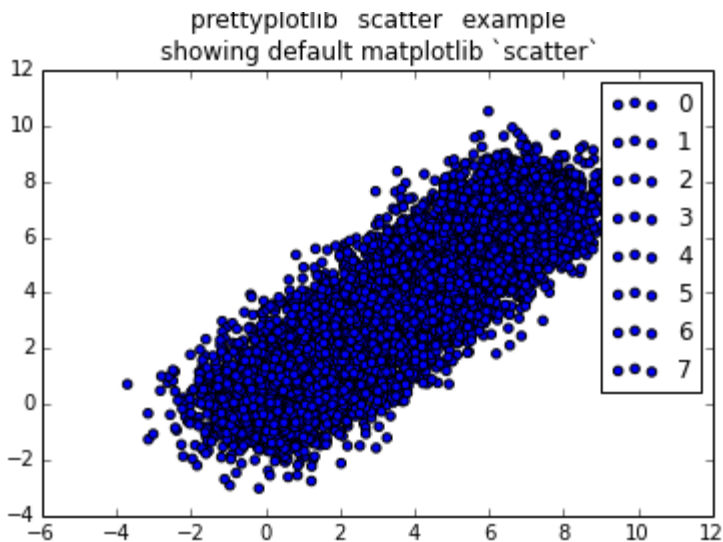
Several projects work with Matplotlib to improve its default styling, without using web graphics technology (SVG or Canvas).

- prettyplotlib
- <http://olgabot.github.io/prettyplotlib/>
- Seaborn
- <http://www.stanford.edu/~mwaskom/software/seaborn/>

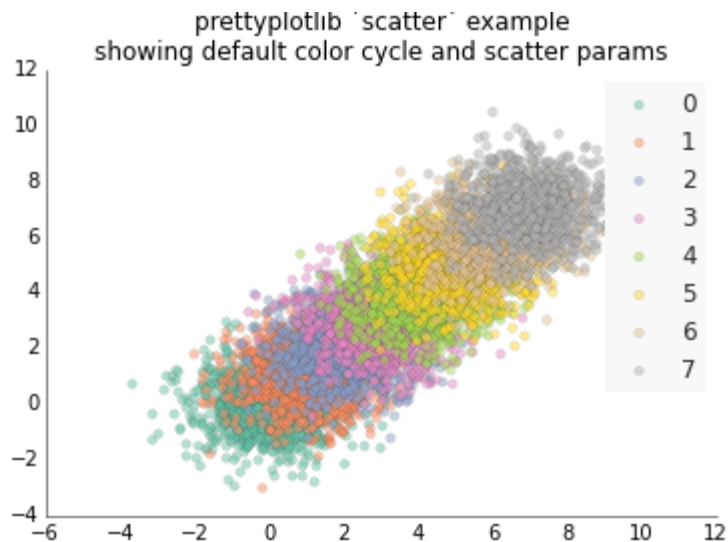


prettyplotlib

<http://olgabot.github.io/prettyplotlib/>



prettyplotlib

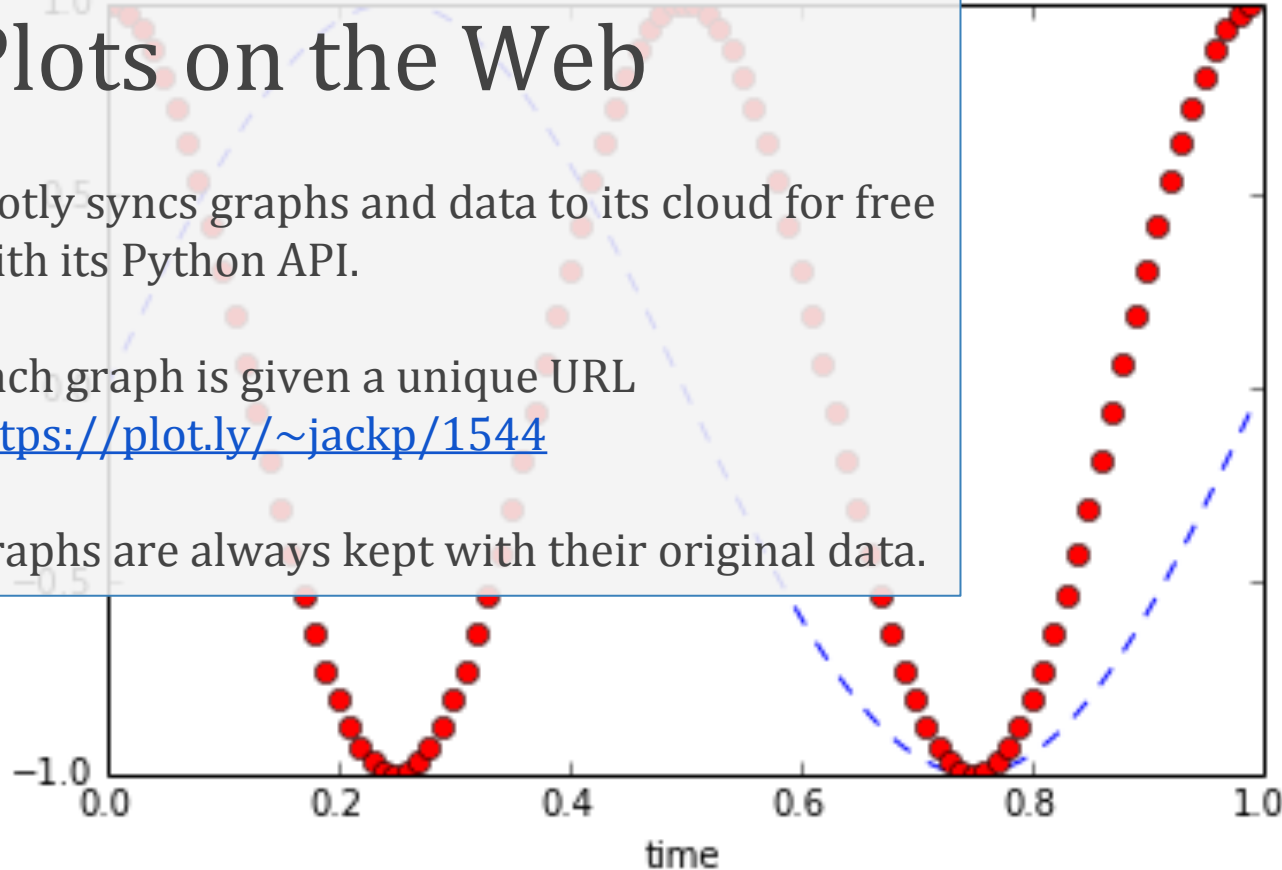


Sharing Scientific Plots on the Web

Plotly syncs graphs and data to its cloud for free with its Python API.

Each graph is given a unique URL
<https://plot.ly/~jackp/1544>

Graphs are always kept with their original data.



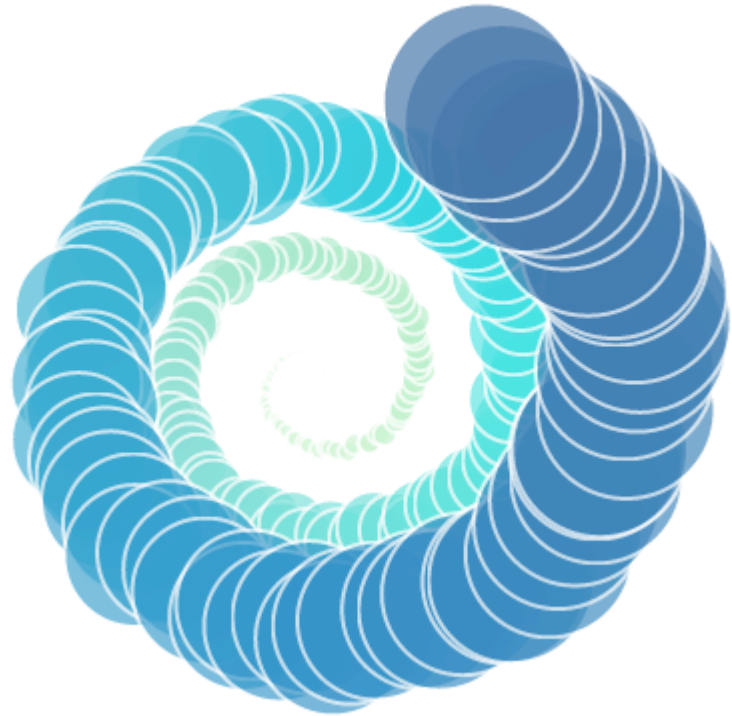
```
fig1 = plt.figure()
x = np.arange(0,1.0,0.01)
y1 = np.sin(2*np.pi*x)
y2 = np.cos(4*np.pi*x)
ax = fig1.add_subplot(111)
ax.plot(x, y1, 'b--',
        label='sin')
ax.plot(x, y2, 'ro',
        label='cos')
ax.set_title("It's a Sign")
ax.set_xlabel('time')
ax.set_ylabel('amplitude')
```

```
fig_to_plotly(fig1, \
              plotly_username, \
              plotly_api_key, \
              notebook=True)\
```



<https://plot.ly/~jackp/1544>

Questions?
jack@plot.ly



<http://nbviewer.ipython.org/gist/jackparmer/7729584>