

bqplot:

Interactive Visualization in Jupyter

PyData Meetup (Ann Arbor)

Dhruv Madeka (dmadeka1@bloomberg.net)
Quantitative Researcher, Bloomberg L.P.

April 13, 2017

techatbloomberg.com

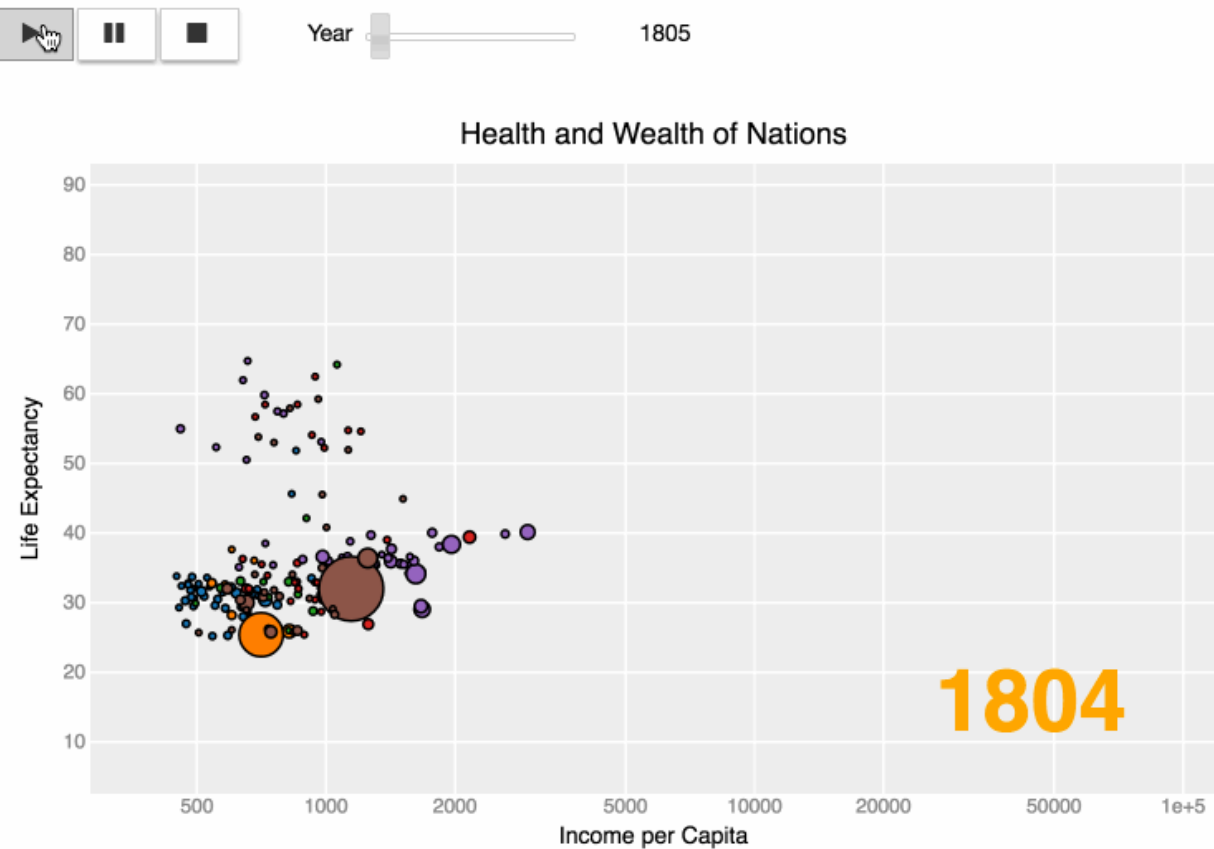
© 2017 Bloomberg Finance L.P. All rights reserved.

Bloomberg

bqplot

- bqplot is a plotting library for the Jupyter Notebook
- <https://github.com/bloomberg/bqplot>
- Another plotting library? Geez, it's hard enough to keep track already:
 - matplotlib
 - ggplot
 - Seaborn
 - Altair
 - Bokeh
- So why learn bqplot?

bqplot

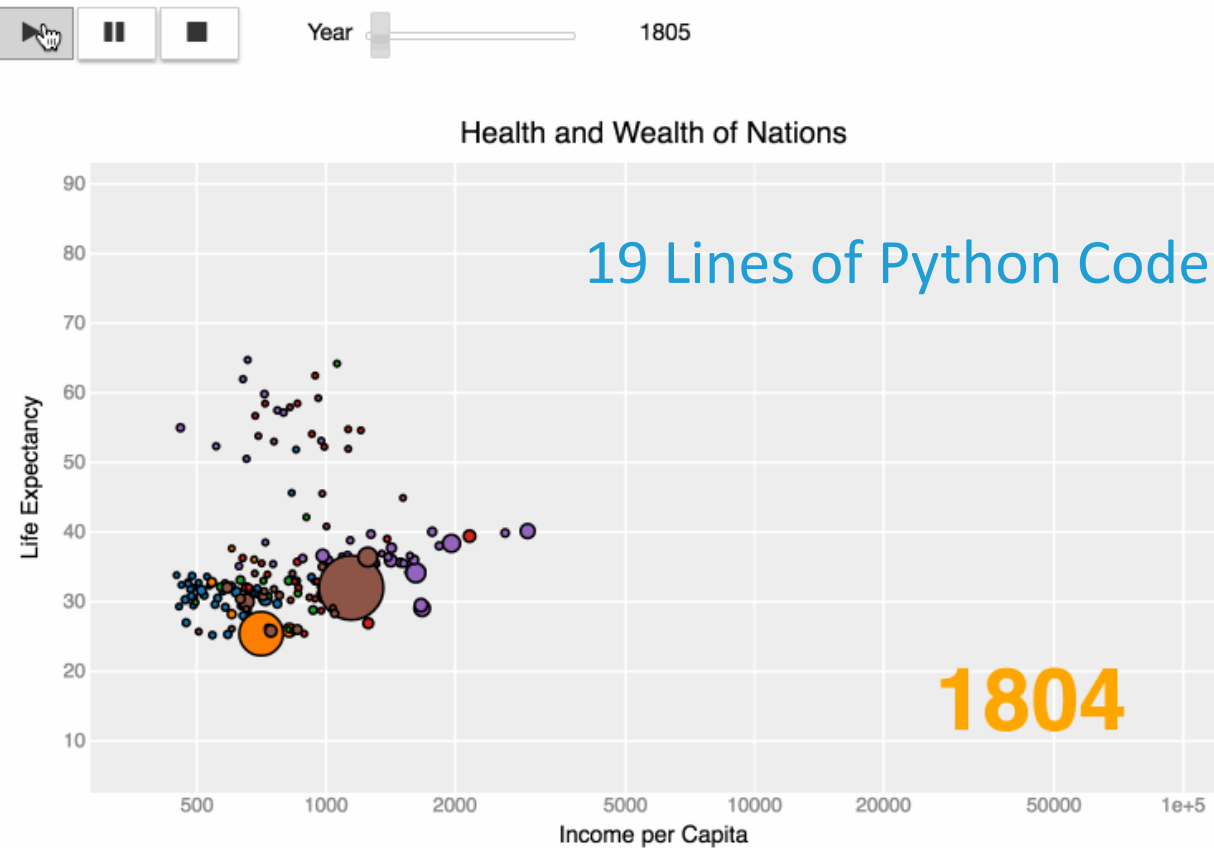


```
In [12]: from bqplot import pyplot as plt
         from bqplot import *
         import pandas as pd

In [13]: data = pd.read_csv('Data/2016-results.csv')

In [15]: map_fig = plt.figure(title='Visualizing the 2016 Election Results', min_width=1300, min_height=800)
         map_tt = Tooltip(fields=['name', 'color'], labels=['County Name', 'Democrat %'])
         map_res = plt.geo(map_data=topo_load('map_data/USCountiesMap.json'), stroke_color='black',
                           colors=['default_color': 'black'],
                           scales={'projection': AlbersUSA(),
                                   'color': ColorScale(colors=['Red', 'Gray', 'DeepSkyBlue'])},
                           color=dict(zip(data['fips'], data['dem.pct'])), tooltip=map_tt)
         map_fig
```

bqplot



```
In [12]: from bqplot import pyplot as plt
         from bqplot import *
         import pandas as pd

In [13]: data = pd.read_csv('Data/2016-results.csv')

In [15]: map_fig = plt.figure(title='Visualizing the 2016 Election Results', min_width=1300, min_height=800)
         map_tt = Tooltip(fields=['name', 'color'], labels=['County Name', 'Democrat %'])
         map_res = plt.geo(map_data=topo_load('map_data/USCountiesMap.json'), stroke_color='black',
                           colors={'default_color': 'black'},
                           scales={'projection': AlbersUSA(),
                                   'color': ColorScale(colors=['Red', 'Gray', 'DeepSkyBlue'])},
                           color=dict(zip(data['fips'], data['dem.pct'])), tooltip=map_tt)
         map_fig
```

8 Lines of Python Code

bqplot

- bqplot is a web visualization library for the Jupyter Notebook
- Exposes the full power of web applications through an imperative Python syntax
- Allows selections, interactions and CSS customizations directly from Python
- Every element of the chart is a Python widget, which can be bound to a Python function – **this makes every chart an interactive application**

bqplot – How does it work?

- The Python side:
 - Based on the ipywidgets infrastructure of the Jupyter Notebook
 - Bi-directional communication between JavaScript and Python
 - Can be used to create full-fledged web applications – **entirely handled by just a few lines of Python code**
- The JavaScript side:
 - Built on top of D3.js, which handles all the data binding, scales, axes, etc.
 - Uses an extension of the Grammar of Graphics to include interactions
- User never sees any JS; the data/code is on the Python side
- Can leverage the entire Python scientific stack, with easy D3-powered applications

Conclusion

bqplot:

- Exposes a D3.js based web visualization library directly in Python
- Enables scientists/researchers and other programmers unfamiliar with web applications to add them to their workflow
- Is looking for contributors!

Find out more on GitHub: <https://github.com/bloomberg/bqplot>

Email: bqplot@bloomberg.net