

Interactive Programming
Louise Nielsen and Apurva Raman
SoftDes Spring 2016

Project Overview:

We created an interactive map that displays resources for queer youth in Massachusetts. We collected the name, address, location (in latitude and longitude) for each resource and categorized them. We then displayed this data using Bokeh on a Google Maps plot and had data points be different colors depending on their category.

Results:

Default state:

We displayed the data on a Google Maps plot and indicated the category by changing the color of the glyphs (the circles indicating the presence of a resource). Each glyph maps to the latitude and longitude locations displayed on the Google Maps plot. The default state shows most of Eastern Massachusetts and is centered at Boston.

Interactions:

When the user hovers over a glyph, they can see the name of the resource, its address, and the category it belongs to. The user can also pan, zoom, and box-select to see the portion of the map they are interested in. They can move the center of the map to their desired location, and change the zoom as they see fit.

Other options:

The map type can be changed from 'roadmap' to 'satellite' within the body of the init method for the MakePlot() class.

Implementation:

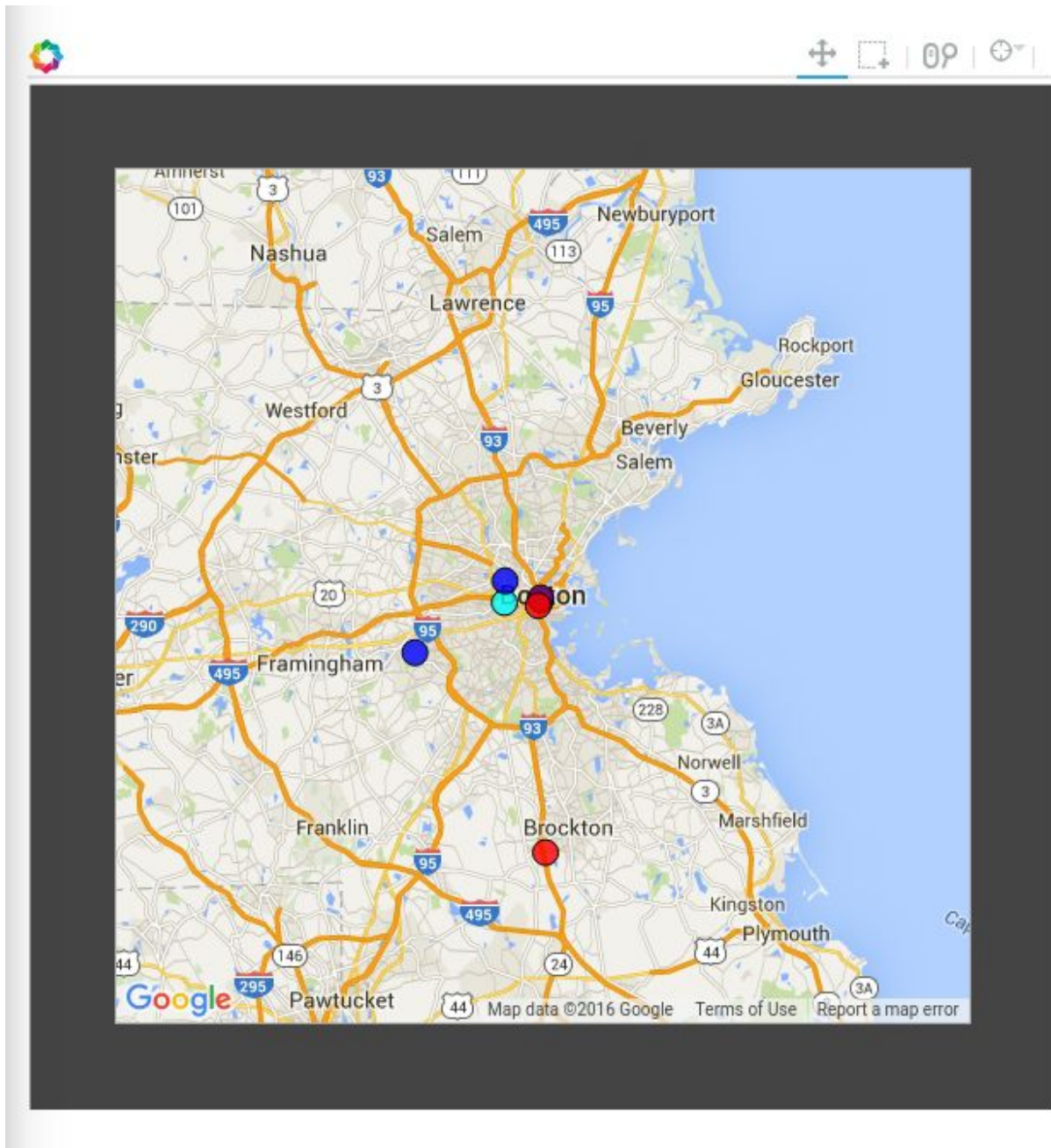


Fig. 1. The default state of the map.

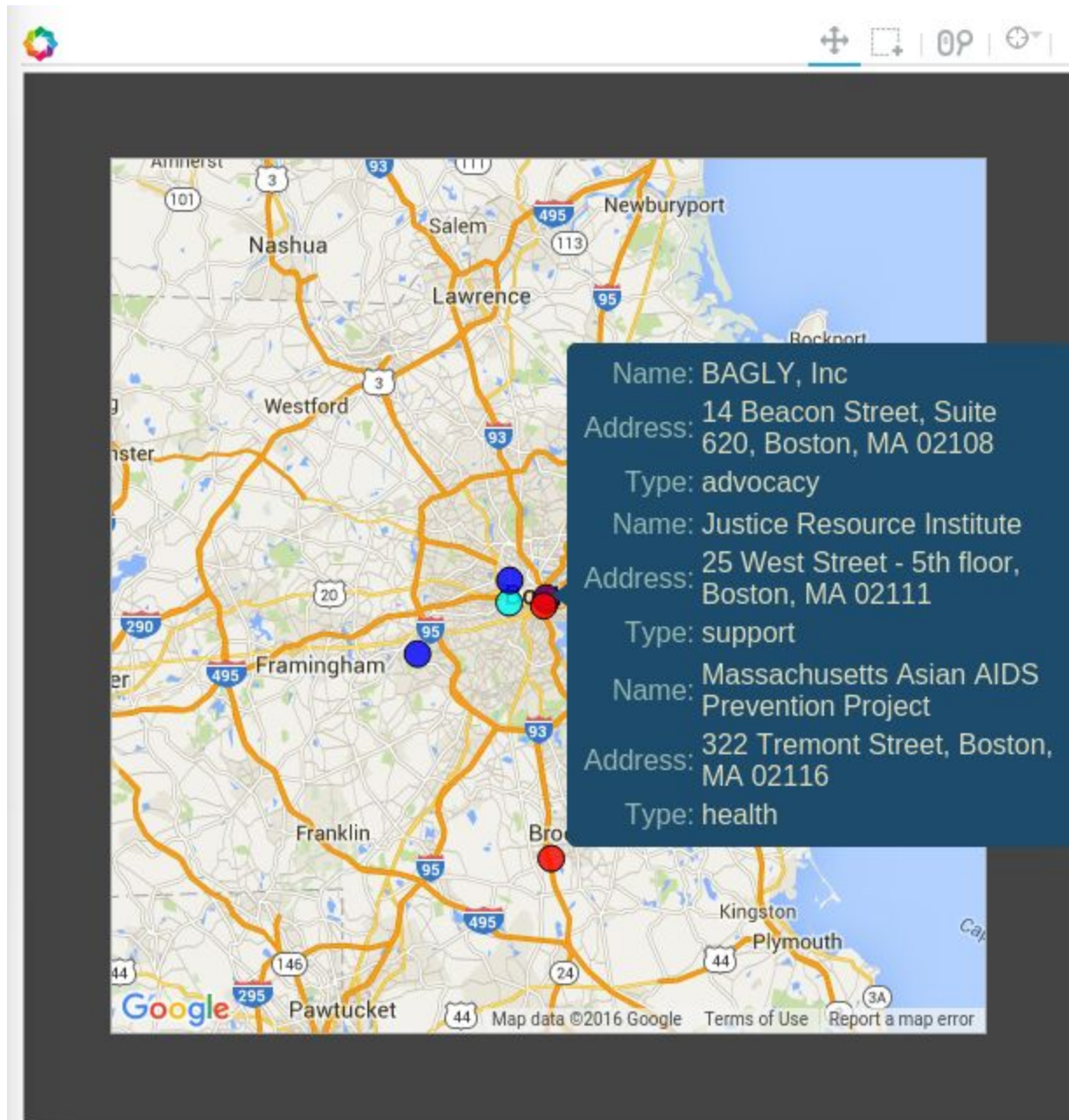


Fig. 2. Hovering over a cluster of resources.

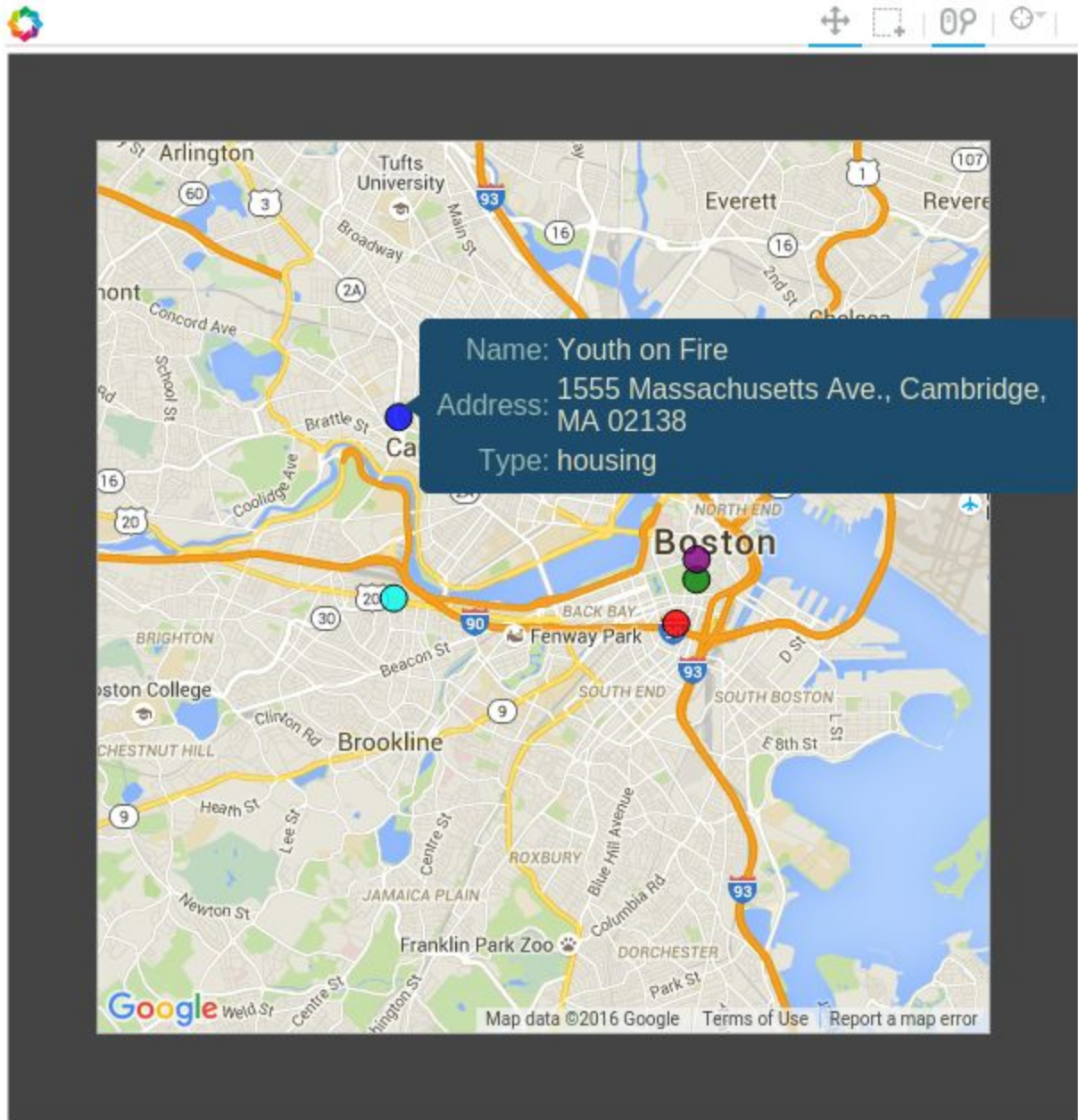


Fig 3. Map after zooming and panning

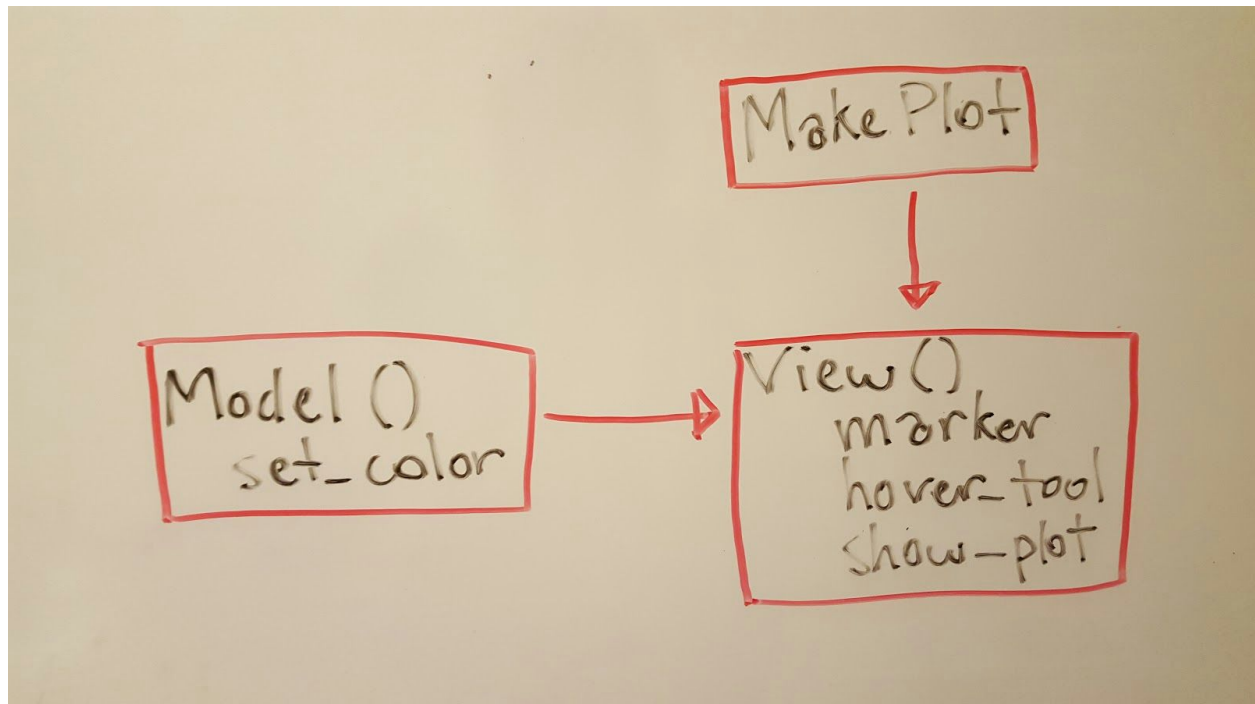


Fig 4. UML diagram.

We had three classes in our code.

The first class was Model(), which read a csv file and converted it to a pandas dataframe and set an additional color attribute based on category using Bokeh's colormap. This design decision to use the pandas dataframe came late in our process; we were using dictionaries before we had implemented the glyphs' color changes corresponding to category. When we attempted to change the glyphs' color, we realized that many of the Bokeh methods were written for interfacing with pandas, and although using a dictionary was possible, we would have to split our dictionary (which was just read from a csv) into a separate dictionary for each category, which would require iterating through the entire dictionary to find the keys associated with certain values in the category field. At this point we realized that using pandas (as was recommended in Bokeh documentation) would be the best way to change the color of the glyphs by category.

The second class, MakePlot(), created the base plot using Bokeh. It draws a map using Bokeh's Google Maps plotting functionality. It also defines the tools the user can use to interact with the plot.

The third class, View() calls Model() and MakePlot(). It plots the data from the Model() in marker; the marker. In hover_tool, the hover tool is added with keys to show the name, address, and type of each resource when the user hovers over the glyph. In show_plot, it shows the modified MakePlot() object in a webpage.

Reflection:

We pair programmed well and figured out how to unit test appropriately; we were never stuck on a single problem for a long time. We could definitely have better planned the architecture and used the documentation more effectively. Our project was not appropriately scoped. We assumed that Bokeh would be more intuitive and implementable than it turned out to be, especially for the Google Maps implementation. Going forward, we'll be more cautious about using libraries we've never seen before.

Our intent was to fully pair program the entire time, and this happened, except for one night in which Louise decided to implement pandas instead of sleeping. Scheduling issues arose, but those were solved by working during class and for hours-long meetings in the last two days. In the future, we'd do more sooner.