D3 Lab 2

What to turn in to the Discussion board: You may hand in different HTML/JS files depending on the step. Please name your file with your name and step (e.g., jhullman_step5.html and jhullman_step5.js). Make sure that you are pointing at the right javascript file (relative or absolute path). We will deduct points if we need to modify your assignment so that it runs.

Note: There are many examples on the Web for all these steps. You're more than welcome to look at how other people do it as a starting point.

Step 1)

D3 scales make certain mapping operations much easier. Remember that much of what we do in visualization is mapping from some input variable (the ***domain***) to some output (the ***range***). The domain can be nominal, quantitative, or ordinal. Similarly, the range can also be quantitative, nominal, or ordinal.

Some examples:

- we have weights ranging from 100 to 300 (the domain is 100-300), and we want to map it to an x-coordinate ranging from 10 to 1000 (the range is 10-1000)
- we have unemployment levels ranging from 0 to 10 and we want to map them to colors ranging from white to light blue
- We have temperatures ranging from -32 to 32 and we want negative temps to be mapped to blue, 0 temp to white, and positive temps to red
- we have nodes with a power-law degree distribution ranging from 0 to 1,000,000 and we want to rescale those to a range of 0-15 based on the square root of the degree

Load the template file step1.html into your editor. There are 4 rescaling questions. Implement the code for these like the example at the top of the file. **Do not change the variable names**. To test your code, load up the javascript console and run the sample commands to make sure you did the right thing. We will be testing using a similar technique.

https://github.com/d3/d3/blob/master/API.md#scales-d3-scale

Turn in your html/javascript as emailid_step1.html (and if you need emailid_step1.js)

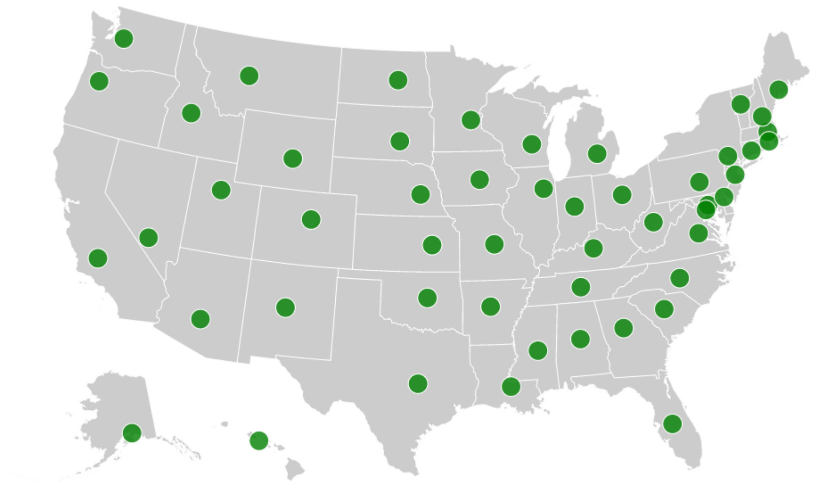
Step 2)

If you haven't installed a local webserver, this is where you'll want it. A webserver is required to load external data from a json or csv file.You can use python to run a local web server on either Mac OSX or Windows; google to find the instructions specific to your python version and operating system.

Grab the files:

step2.html, states.json and us-state-centroids.json

Once you have the server running, you should be able to see:



If you look at the code, you'll see that there is a call to queue() which loads the two json objects (one containing the maps, and one containing the coordinates with population values). When it is done loading, it will call the ready function which will store out the values as two variables (states and centroid) and draw the initial map.

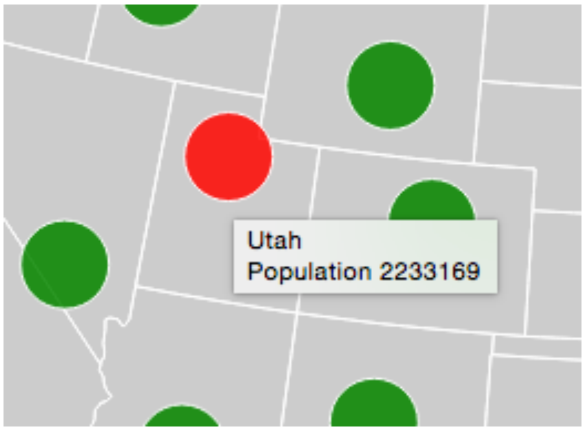
There's nothing to turn in here, keep going

Step 3)

Modify the code so that as you mouse over the circles you get the name of the state and a population count.

Also, the color of the circle changes to red when mouse hovering and when mouse moving away from the circle, its color changes back to green.

Something like this:



Utah
Population 2233169

Mouseover effect

Hint: SVG objects can have a title, google for append("title") and d3 if you're stuck. Also, check try to find examples of the on('mouseover) method.

You do not need to turn in a step 3 html, but include this as part of your step 4.

Step 4)

Notice that I created a function for you that recolors the symbols. So for example, recolor('purple') will make all the circles purple if you call it in the console.

Create buttons at the bottom of your screen that will control what is being displayed.

Something like:

```
<FORM>
<INPUT TYPE="Button" value="make it blue" onclick="recolor('blue')"/>
</FORM>
```

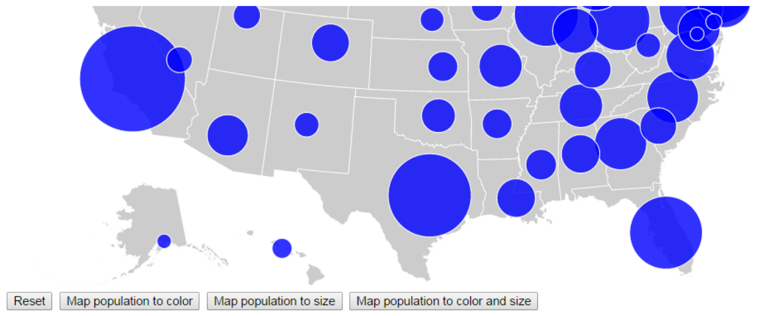
Is a place to start.

The buttons should be:

- Reset – revert back to the original
- Map population to color – map the circle so that the minimum population is green and the max is blue (you may want to use some kind of non-linear scaling so that you can see the smaller differences between populations in spite of the large disparity in values)
- Map population to size – map the circles so that the minimum population is 5 pixels and max is 20 (use non-linear scaling, maybe sqrt, again because of the disparity in values)
- Map population to color + size – do both of the above

Use the scale functions as you did in step 1.

So for example this is what I get if I click on mapping by size (I use blue for the symbol here, but you can do what you want)



Turn in your code as emailid_step4.html and emailed_step.js (if you used a separate js file). The 4 buttons should work as described above (and the title should pop up). Do not hard code *ranges* (you can hardcode the *domains* if you want—the min and max populations). Use the scale functions.

Step 5)

Make the transitions when you press the button animated.

Turn this file in as emailid_step5.html and emailed_step5.js