D3 Lab 1

Today we begin working with d3 in class. **You may not finish this entire lab in class.** Work through as much as you can and submit your work to the Discussion for today. If you don't finish, I strongly recommend that you finish the tasks after class as practice for A3.

What to post to Canvas: Zip your HTML/JS files (corresponding to different steps in the lab) and name the zipped folder with your name. Name each individual HTML/JS file with your name and which step it corresponds to (e.g., jhullman_step5.html and jhullman_step5.js).

Step 1)

To create visualizations we map data to retinal variables. The most effective mapping is to coordinates in space. Unfortunately, the raw values don't often directly fit inside the space we have available to us paper or on screen so we need to rescale.

Step 1a) As a simple example, let's make a scatterplot. Start with the template HTML code that is provided on Canvas.

Step 1b) Now copy and paste the following dataset into your file:

```
var dataset = [[5, 20], [480, 90], [250, 50], [100, 33], [330, 95], [410, 12], [475, 44], [25, 67], [85, 21], [220, 88]];
```

Notice that if you were to directly draw the points, they'd exceed the canvas size. This is where the scale function becomes necessary. If you wanted to map the domain 100 – 500 (i.e., the data range) to the range of 10 - 350 (i.e., the pixel range) we could do:

Example:

```
var scale = d3.scaleLinear()
   .domain([100, 500])
   .range([10, 350]);
```

Given this function, calling scale(300) would give you 180 (halfway between 10 and and 350), because 300 is halfway between the domain 100 and 500.

Step 1c) Create two variables xScale and yScale that map the dataset to the range 0-400 (on the x-axis) and 0-400 (on the y-axis). Use d3.max(...), and d3.min(...) to make this easier.

Step 1d) Paste this in:

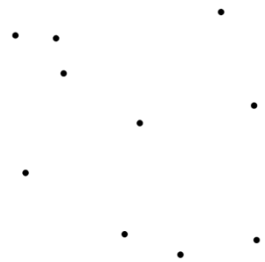
```
svg.selectAll("circle")
  .data(dataset)
  .enter().append("circle")
    .attr("cx", function(d) { return xScale(d[0]); })
    .attr("cy", function(d) { return yScale(d[1]); })
    .attr("r", 5);
```

If you've done things correctly, you'll probably see:

Note: this is a rescaled (smaller) picture, so the radius here looks smaller than 5. Your circles should be larger than these. The same is true below.

Step 1e) Change the code so that your origin, 0,0, which is currently in the upper left corner, is instead in the bottom left.

Step 1f) It's a little annoying that the dots are off the screen. Modify your xScale and yScale so they entirely inside the drawing surface (assume you have a fixed radius of 5). If you're doing more work than changing 4 numbers, you're probably doing too much work:



Turn in your html file for this step (youremailid_step1.html). Make sure your variables/functions are called dataset, xScale and yScale as specified above.

Step 2)

Time to add some axes. Copy your last file and make a new one (call it youremailid_step2.html). Let's make a horizontal axis:

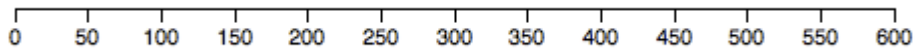
```
var xAxis = d3.axisBottom()
  .scale(xScale);

svg.append("g")
.attr("transform", "translate(0," + height  + ")")
.call(xAxis);
```

Step 2a) Now add a vertical y-axis.

Step 2b) Chances are the axes don't look as nice as they could – too thick and too crowded. Modify your

code (e.g., see the 'ticks()' command) and play with the css to make the axes look a little nicer, for instance, styled more like this:



Turn in your html file for this step (youremailid_step2.html).

Step 3)

Copy your step 2 code to a new step 3 file and copy this code to the bottom of your script

```
// reset everything to color=black, radius=3
circles = svg.selectAll("circle")
  .data(dataset)
  .attr("fill","black")
  .attr("r",3);

// transition to magenta with a radius of 6 over 1 seconds
transition1 = circles.transition()
  .duration(1000)
  .attr("fill","magenta")
  .attr("r", 6);


// transition to black with a radius of 3 after a delay of 1 seconds // and make it take 1 seconds again
transition2 = transition1.transition()
  .delay(1000)
  .duration(1000)
  .attr("fill", "black")
  .attr("r", 3);
```

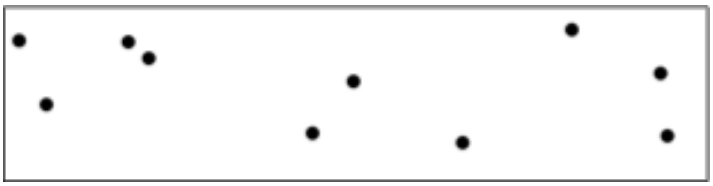
What this will do is: resets everything to black with a radius of 3, smoothly transforms the color to magenta with a radius of 6 and THEN to a black circle with a radius of 3. There are other ways to chain transitions, but this happens to be a simple one (that requires adding delays so the transitions run on after the other).

Step 3a) *Modify the code so that in addition to doing the above, the points move to a random position*. The new coordinate should be somewhere between 0 and the max X in the dataset, and 0 and the max Y in the dataset. You can make the coordinate change either happen in the first or second step (when it is going to the old color or when it's going to black). Make the last step happen over 3 seconds. Hint: Look up Math.random()

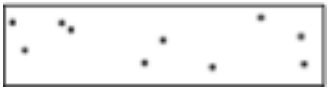
Turn in your html file for this step (youremailid_step3.html).

Step 4)

Copy your file from step 3 and make a new one (call it youremailid_step4.html). Let's say for some reason we want to maintain the aspect ratio of the original dataset which is very long and not so tall. (none of these are drawn completely to scale)



What we want to do is use a scale to ensure that the maximum dimension (either width or height) fits inside 400x400. So if our picture looks like the one above, we'd want something like:



(where the width of this is 400). If the dataset were pivoted (taller rather than wider), we'd want to see:

var dataset = [[20, 5], [90, 480], [50, 250], [33, 100], [95,330], [12, 410], [44, 475], [67, 25], [21, 85], [88, 220]];

Modify your xScale and yScale to support this. You can assume a minimum value on both the x and y dimensions of the dataset to be 0, but you will need to (1) figure out which is the bigger dimension, (2) rescale that dimension to fit inside 400, (3) rescale the other dimension proportionally (i.e., if you had to shrink the x-direction 50% to make it fit, you should shrink the y dimension by that much).

Turn in your html file for this step (youremailid_step4.html). Make sure your variables/functions are called dataset, xScale and yScale as specified above and *please document your code.* Make sure you test for datasets that are bigger than the canvas space of 400x400 in each of the dimensions, both of the dimensions, or neither of the dimensions.