

CS/INFO 3300

Homework 3

Due before class Friday 2/21

1. Write your own Javascript function that has the same behavior as the d3 function returned by this expression: (20 pts)

In Q1, provide a function that reproduces the exact behavior of the d3 scale returned by the specified d3 code (ie using the numbers specified).

```
d3.scale.linear().domain([-85, 85]).range([200, 0])
```

2. Revisiting HW1, Q5: Create three <div> elements with class "findable". Using a single d3 function, select all the nodes with this class. Set the content of these nodes to "found", and set the style to display in red text. Consult the d3js.org API documentation as necessary. (10 pts)

When you click a shape, only that element should change color. Include axes (tick marks optional) and text labels as in HW2, but use d3 commands to add them.

3. In problem 2 on HW2 you recreated Fig. 2 from the Wickham reading using SVG elements. Now create the same plot again, but this time using d3 functions. First create x and y scale functions that map from data coordinates to SVG coordinates (10 pts). Add circles and rectangles, with positions given by the x and y scales. You don't need to use a data array: it's fine if you do a separate command for each shape (10 pts). Add lines for the axes, again using the x and y scale functions (10 pts). Now add an event listener that changes the color of the circles and rectangles to blue when they are clicked, using d3 selections (10 pts).

4. In this problem we're going to plot some data about English word frequencies from Google Books. The file <http://mimno.infosci.cornell.edu/info3300/zipf.json> contains a Javascript snippet that defines an array of objects. Each object represents a word, sorted by its frequency. The most frequent word, "of", occurs 15 billion times. The 512th most frequent word, "middle", occurs 45 million times. Copy this snippet into a script element in your homework document.

A. Create a 200x200 pixel SVG element using d3 functions. Create two linear scale functions: an x scale for the "rank" and a y scale for the "count". Choose the "range" attributes to be appropriate for "rank" and "count". Use d3 to add text elements to the plot for each word in the data set. Use a loop or a "forEach" statement; you may not use a separate command for each word. Is this visualization useful? Why or why not? (10 pts)

B. In this section we'll transform the data as you create the text elements. Create a

second 200x200 SVG element. Add the same points, but this time set "x" to the log of word's rank (use `Math.log()`) and "y" to the log of the count. You will need to create new x and y scale functions using appropriate values for the "range". How does this version differ from the previous version? (10 pts)

C. Now rather than transforming the data, lets change the scale functions. Create a third 200x200 SVG element, and create two **log** scale functions using the same "range" values as in part A. See the d3js.org API documentation as necessary. Use d3 to add text elements to this new plot, again using the original values for "rank" and "count". (10 pts)