May 21, 2018

In Lab Final

CMSC 23900: Data Visualization

In this final you will complete a trio of small tasks, as enumerated below. To facilitate and monitor your completion of these tasks we have provided a small test suite (tests/final-tests.js). We have configured a visual test harness so that you can see if your tests succeed or fail in real time. To access this test suite, begin by getting everything installed and turning on the web server:

```
npm install
npm run start

# or if yarning

yarn
yarn start
```

Then point your browser to http://localhost:3001/. In this test you will **do all of your work in src/utils.js**. There are three functions written out there, you only need to fill them out. You **do not need to install any additional packages.** Your grade will consist of 80% for correctness (fraction of the tests that pass) and 20 % for style (all or nothing lint check).

1. To begin, you will build a function called sumByGroup, which takes three arguments an array of objects called data, a string called grouping key, and a string called summing key. Your function should execute a single pass across the data and build an object with keys equal to the each rows groupby key, and generate a sum for that group using the summing key.

2. Next, you will build a small implementation of a function that we have made extensive use of throughout the quarter: scaleLinear. As you know scaleLinear take a domain and range and provides a mapping between the two. The function you will build, myScaleLinear takes two arguments, domain and range, which are both arrays of pairs of numbers, and returns a function that implements the following function:

$$f(x) = \frac{x - domain_{min}}{domain_{max} - domain_{min}} * (range_{max} - range_{min}) + range_{min} \quad (1)$$

3. Finally, you will create a small visualization of some fake data in the form of donut chart. A donut chart is like a pie chart, but with the center cut out. In the test suite you should see an example donut chart, your goal is to make your chart look as close to that image as possible. You should feel free to check whatever documentation you feel is necessary to build this correctly. The reference solution places the labels for each of the donuts halfway in between the start and end angle for each of the wedges.

   Hint/Tip: d3's pie function starts it's angle at 12 o'clock (as opposed to the mathematically traditional 3 o'clock).