SI649 Fall 2016 – Lab 3

Sep. 26/27, 2016

Lab Logistics:

- **Though some of you may get through this, we do not expect you to finish this entire lab in class.** Work through as much as you can today and then finish the rest at home to turn in for next week (upload through Canvas by midnight, Sunday, Oct. 2nd).
- Please work with your lab partner *in class*. You can continue to work with them on the assignment outside of class or find another partner (please try to keep groups to 2-3 people and report everyone you worked with on your assignment). You may not share code with other groups/individuals (though you can talk through solutions). You may either edit the code together with your lab partner in one file, on one computer (just don't forget to share it before you leave class today) or you can work on your own laptops. Whatever is more effective for you…
- Rename the html file with your name (e.g., eadar_lab3.html). Don't forget to modify your name and your partner's name in the HTML file.

What to turn in:

You will upload:

□ The modified HTML file for question 1-9 (e.g., eadar_lab3.html)

Note that to test your code we will put in a new salary table with different numbers. If you hardcode anything it will break!

Step 0) A few basics

□ Download the starting file lab3.html

We are going to work with an upgraded version of the salaryData from lab 2. On the top of the file, you will find the dataset. Our employees now have three components to their salaries based on the projects they work with (A,B,C)

```
var salaryData =
       [{name:"A",salary:4,dept:"IT",salaryA:1, salaryB:3, salaryC:0},
       {name:"B",salary:2,dept:"HR",salaryA:1, salaryB:0.5, salaryC:0.5},
       {name:"C",salary:3,dept:"HR",salaryA:1, salaryB:1, salaryC:1},
       ……
```

Recall that in order to create a bar chart that visualizes the salary and over-/under- pay information, we have implemented several functions in the previous lab, To summarize, we did the following steps (we've coded some parts of this for you… ~lines 50-100)

1. Calculate mean salary of a department (with `dept_mean_salary(data, dept)`)
2. Calculate mean salaries of each department, which are stored in the variable `meansByDept` using the function `populatemeans(data)`
3. Add over-/under- pay information into the dataset(`addOverUnderPayInfo(data)`)
4. Find the min and max value for salary (stored in `salaryRange`) and overunder. (stored in `overunderRa nge`)
5. Calculate height of the bar (using the function `height(salary)`)
6. Calculate color of the bar (using `calculate_color(overunder)`)
7. Plot the bar

After you execute the code we've given you, each employee will now have a "salary" field (sum of their salaries) and the "overunder" field. Make sure you understand our code.

We are going to replicate the bar chart with SVG and D3. But before that, let's take a look at some helpful functions from the d3 library.

Part 1 Process data with d3 *helper* functions

In the starter code, we have done the first 3 steps for you and included that in the dataset. Now let's go find the min value for salary and overunder.

Step 1.1: Find min and max range

In the previous lab, we found the min and max value for salary (`salaryRange`) and overunder. (`overunderRange`) by implementing four helper functions:

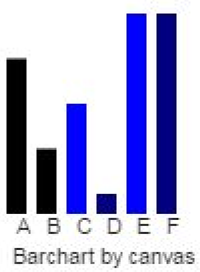
```
function maxSalary(data){ }
function minSalary(data){ }
function maxOverunder(data){ }
function minOverunder(data){ }
```

☐ **Because these four functions share similar purposes, let's create a more generic helper function, you should implement using the D3 helper functions (d3.min,d3.max, etc.): `findMinMaxRange(data,fieldName)`**

It should return an object of min and max value of the field with input fieldName. E.g.,
`overunderRange=findMinMaxRange(salaryData,"overunder")`
should return {"min":-0.5,"max": 0.5}.
`salaryRange=findMinMaxRange(salaryData,"salary")`
should return {"min":1,"max": 5}

You should be using the D3 helper functions (e.g. d3.min(), d3.max() and d3.mean()). Check https://github.com/d3/d3-array for more information.

After completing this step,the function `drawBars(data)` will work properly and you will see the following visualization identical to what you have created last time.



Part 2 : Creating the name-salary bar chart with svg.

To create the bar chart using the svg, we will create some bars (i.e., <rect>). Therefore, we need to know the x, y, height and width for each rectangle. We know that width is a constant (use width=10).

Step 2.1 find parameters x, height and color.

□Fill in all x values. Assuming x=0 for the leftmost bar, width=10 and spaceBetweenBars=5. You can look at `drawBars(data)` to see how the x values are calculated.

Name	x	y	height	color
A	0		77.5	(0,0,0)
B				
C				
D				
E	60		100	(0,0,255)
F				

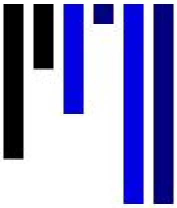
In the previous lab, we mapped the salary information to the height of the bar using (`height(salary)`). We also mapped the under/over pay information to the color of the bar using `calculate_colo r(overUnder)` . **Fill out the values to this table at the bottom of the HTML files (there's a table we started for you)**

Step 2.2 find y values.

We want to align our barchart by with the barchart by canvas. Therefore, we want to set the y-axis of our barchart to be 250. (The location to do this is at the top of the HTML file, look for 2.2)

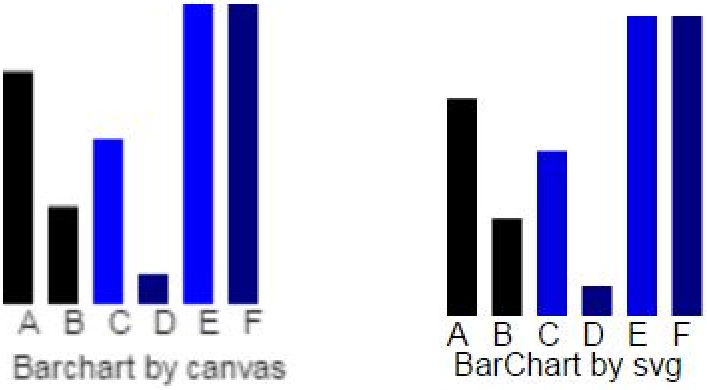
□ **Draw the bar chart by filling in x, y, height, color in the <rect></rect> tag.**

If you see an upside-down bar chart like the following image, what happened?



HINT: Which point of the rectangle does (x,y) control? Read the documentation of <rect></rect> in here: http://www.w3schools.com/graphics/svg_rect.asp.

If you draw set x, y, height and color correctly, you will see the following page:



Part 3: Creating the name-salary bar chart with D3

Step 3.1: Calculating Scale

Because the scaling processes for both salary->height and overunder->color are quite similar to each other (they are both linear), we can create a generic scaling function. In the starter code, we have provided this generic function

```
calculateScale(input, inputRange, outputRange).
```

E.g., calculateScale(5,{‘min’:0, ‘max’:5}, {‘min’:10, ‘max’:100}) will return 100.

Since we are going to perform the linear scaling quite often, and we don’t want to write `calculatingScale(input, inputRange, outputRange)` everytime, d3 has provided you several functions to help you with scaling. See https://github.com/d3/d3-scale for more details.

In the starter code, we have implemented the heightScale using d3.

```
var heightScale = d3.scale.linear()
  .domain([salaryRange.min, salaryRange.max])
  .range([heightRange.min, heightRange.max]);
```

□ **Implement colorScale such that colorScale will map the overunder value to the color range from 0 to 255.**

Test your values to make sure they are in line with what you were expecting (see the table you created)

Step3.2: Drawing rectangles in D3

In the starter code, we have

1. Created an <svg> element which has the class of ‘d3Chart’
2. Created a group (‘g’) within the <svg> element, which has the class ‘barChart’.
3. Within the barchart, we have created ‘bars’, a variable that we are going to use to draw all bars.

□ **Write the function that set the ‘x’ attribute of rect. HINT: you can look at drawBars(data) to see how the x values are calculated.**

□ **Write the function that set the ‘y’ attribute of rect. if you only use the heightScale(salary), the barchart is upside-down, why?**

Step 3.3: Enable colors

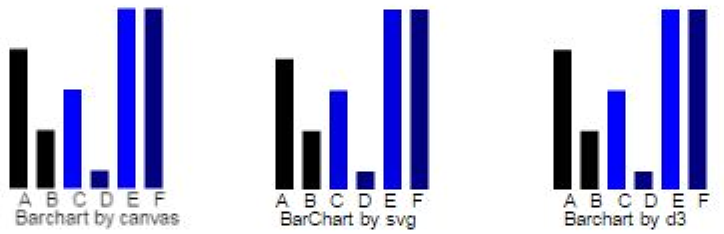
□Change the line `bars.style(‘fill’, ‘red’)` so that these bars are colored by overunder information. You want to use the colorScale that you have created in Question 2. Don’t forget to round the value.

Step 3.4: Adding text

□ Complete the nameLabel chain so that the name information is placed under the bar chart. To do this, you want to specify the following properties/attributes.

1) text
2) x
3) y .
4) font-family (e.g. "Arial")
5) font-size(e.g. "8pt”)

If everything works correctly, you will have three bar charts as follows:



Step 3.5 Bonus Question: Canvas v.s. SVG v.s. D3

Enlarge the webpage (ctrl + in chrome). Compare two visualizations, what do you find? What do you think happened here? If you inspect the html elements, does the bar chart created in the html file (question 2) and the one created using d3 different from each other? How about the canvas and svg? Please write your answer at the bottom of the HTML file, there’s a space for it.

Part 4: Creating the salary-composition bar chart for each employee.

Because the salary of these employees are project-based, there are three additional parameters salaryA, salaryB and salaryC which give the salary composition of each employee. For example, from `{name:"B",salar y:2,dept:"HR",salaryA:1, salaryB:0.5, salaryC:0.5}` we know that B's monthly salary is 2k. he/she earned 1k from project A, 0.5 k from project B and 0.5k from project C.
We want to plot a salary-composition bar chart to show each employee.

Step 4.1 Complete the salary-composition bar chart for employee A.

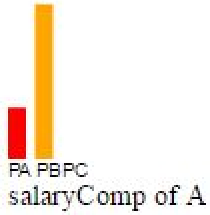
In the starter code, we have provided basic settings for creating the salary-composition bar chart. We have created a new <svg> with the name `salaryCompSVG` under the `salaryCompDIV` <div>. We will create the salary-comp bar chart in the following steps:

□ **Fill in the correct domain and range for `salaryCompHeightScale`, the new scale function for our salary composition charts. `salaryCompHeightScale(3.5)` should return 100 and `salaryCompHeightScale(0)` should return 0.**

□ **Bind the correct data to the `salaryCompA`, the group (<g>) where you are going to create your chart in**.

□ **Draw the salary-composition bar chart for employee A. You can use the same bar chart settings as we defined for previous questions (i.e., width=10, spaceBetweenBar=5, yAxis=120)**

After finishing this step (including bonus), you will have an image similar to the following one.

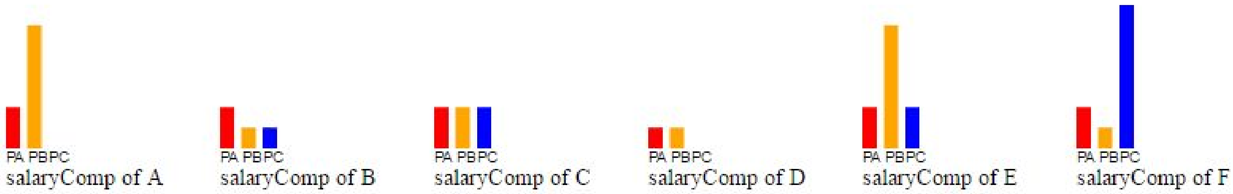


Step 4.2 Draw the salary-composition bar chart for all employees.

□ **Plot the salary-composition chart for employee B-F**

HINT 1: You can start by modifying your code for 4.1
HINT 2: You should not draw everything for employee A first, employee B next, etc. The "D3 way" is to first draw all the bars for project A (for all employees) and then project B (for all employees) and then project C (for all employees)

You will see an image similar to this if you plot correctly (the size of each chart is about 120 x 150 px).


PA PBPC
salaryComp of A
PA PBPC
salaryComp of B
PA PBPC
salaryComp of C
PA PBPC
salaryComp of D
PA PBPC
salaryComp of E
PA PBPC
salaryComp of F