SI649 Fall 2016 – Lab 2

Sep. 19/20, 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 this Sunday, Sep. 25).
- Please work with your lab partner in class. You can continue to work with them on the assignment outside of class. 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_lab2.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_lab2.html)

☐ The extra credit HTML file (basically your original file modified to also answer question 9: e.g., eadar_extracredit.html)

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

Step 0) A few basics

You need 1) an text editor 2) a browser

Few basic text editor:

- Emacs
- textmate, http://macromates.com/,
- atom, https://atom.io/
- sublime, http://www.sublimetext.com/).
- collaborative editor: http://stackoverflow.com/questions/10274181/online-real-time-shared-javascript-console.

Browser: We are going to use Chrome for this lab, but most other browsers also have developer environment.

☐ Download the starting file lab2.html

☐ Open up chrome the developer environment (F12 in chrome). If you run into problems this is useful for seeing the console.

Scenario: You are a HR for a top technology company. Before offering a salary package to a new grad, you want to make sure the package is reasonable by analyzing the salaries of various employees in your company. You are especially interested in whether specific group of people are under- or overpaid.

Although there are a lot of steps here, we've broken them down to make it easy. You will be implementing probably around 30 lines of code total.

Part 1 process data

Our data is stored in a json format. In the lab2.html file, you will find:

```
var salaryData =
      [{name:"A",salary:4,dept:"IT"},
      {name:"B",salary:2,dept:"HR"},
      {name:"C",salary:3,dept:"HR"},
      {name:"D",salary:1,dept:"MG"},
      {name:"E",salary:5,dept:"IT"},
      {name:"F",salary:5,dept:"SL"}];
```

Question 1: What's the salary of E? Which department does E belong to?

This is just something for you to play with. We have provided a number of ways of displaying this information (console, alert, etc.) which are useful for debugging. Try to uncomment each (one at a time) to see what happens. Why would you prefer one over the other? Make sure you understand why this pulls out the data for "E."

```
alert("Name = E, salary ="+salaryData[4].salary+" dept ="+ salaryData[4].dept)
console.log("Name = E, salary ="+salaryData[4].salary+" dept ="+
salaryData[4].dept)
confirm("Name = E, salary ="+salaryData[4].salary+" dept ="+
salaryData[4].dept)
```

Question 2: Practicing with a simple encoding loop

Write a for loop to output each person and their salary on a new line. You should get something that looks like this:

A 4
B 2
C 3
D 1
E 5
F 5

Let's create a simple **encoding**. We're going to pick a different color for every department. Modify your loop to support this. Hint:
document.write(""+...+"");
You should get something like this (your colors should vary):

A 4
B 2
C 3
D 1
E 5
F 5

Question 3: What's the mean salary of all employees?

Modify the meansalary function under step 2 to calculate the mean salary. Fill in what you expect to see as a comment:

```
// should return: XXXX
function meansalary(employees) {
      … calculate the salary mean …
   return mean;
}
```

Question 4: What's mean salary of each department?

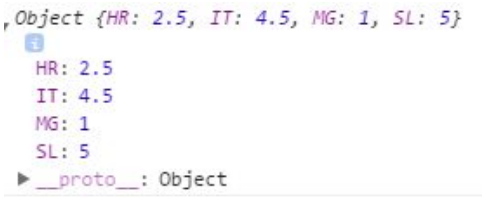
Complete the function `dept_mean_salary(data, dept)` to calculate the mean of a particular department:

■ We have created a variable called meansByDept:

```
var meansByDept={"HR":0, "IT":0, "MG":0, "SL":0}
```

This had the average salary per department. We want to update this dictionary using two functions: `dept_mean_salary(data, dept)` and `populatemeans(data)`.

☐ Complete the function `populatemeans(data)`. It calculates the means of EACH department and store these values in `meansByDept`. When you type `console.log(meansByDept)` you should have:



HINT: `populatemeans` can loop through each department and call `dept_mean_salary(...)`.

Question 5: find out if a specific employee is under- or over- paid, and by how much.

Ok, now we can easily calculate how much over or underpaid someone is (relative to the mean in their group). When we do this, let's add it to the employee database so we can access it later.
So for example, A is in IT and gets paid $4 but the average is $4.5. So they are "underpaid" by $0.5. E is overpaid by $0.5.

☐Implement the function `addOverUnderPayInfo(data)` so that for each employee, we add an "`overunder`" data to each employee's information.

If you did this manually, you'd see something like this:

```
[{name:"A",salary:4,dept:"IT",overunder:-0.5},
      {name:"B",sala ry:2,dept:"HR", overunder:-0.5},
      {name:"C",salary:3,dept:"HR", overunder:0.5},
      {name:"D",salary:1,dept:"MG", overunder:0},
…….
];
```

Question 6: What's the minimum and maximum salary?

In this step, we want to find out the minimum and maximum salary of all employees by completing the following functions:

☐ `minSalary(data) { // return minSalary }`
☐ `maxSalary(data) { // return maxSalary }`
☐ Store the min and max salary information in a dictionary `inputSalaryRange`: `inputSalaryRange={"min": minSalary(data), "max": maxSalary(data)}`

Part 2: Creating Visualizations

We are going to create a bar chart for this dataset. But before we draw the bar, we need to figure out some basic settings:

Question 7: What is the height of the bar?

In our dataset, the salary goes from 1 to 5. If we directly plot the salary onto the screen, it would be too small too see. Because the sizes of our screens are limited, we cannot enlarge the data using a constant. Let's say, the height of our bars are limited from 10 to 100 pixels. We need to write a map function so that the smallest salary is 10 pixels high and the largest is 100 pixels.

For our data above, this means (calculate the rest and fill in the table):

Name	Salary	Pixel height
A	4	
B	2	
C	3	55
D	1	10
E	5	
F	5	100

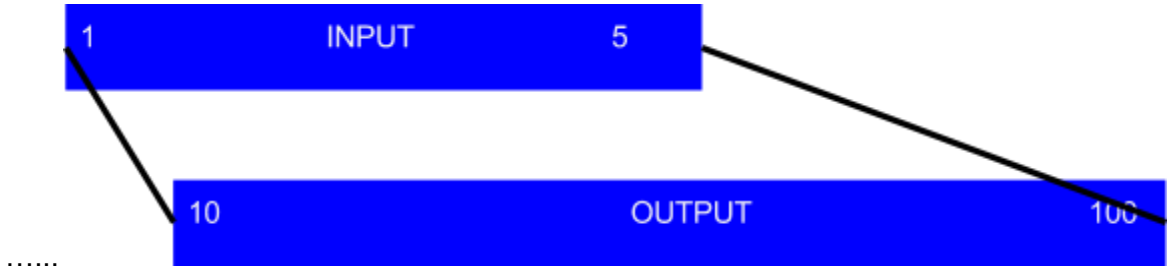


Figure 1

Do this on paper first (or excel if it helps).
Write the equation that takes salary as input and produces pixel height as output that matches the numbers above. HINT 1: If the minimum value of salary and height both start from 0, how would you map the height? HINT 2: start with the assumption that you know the maximum and minimum salaries (5 and 1 respectively). HINT 3: What you're looking to do is to map from the input range to the output range:

What your function should do is map any point in the top range (input) to the bottom range (output). Think through what it means if your input value is 3 (exactly halfway between 1 and 5). What should the output value be?

☐Once you've figured out how to do this on paper implement the function:

```
function height(salary) {
// return the height of the bar that represent the amount
of salary
}
```

So if you run height(1) you'll get 10, height(5) and you'll get 100, and height(2) you'll get 32.5.

Make sure your function isn't hard coded to the values 10-100. If you modify the `heightRange` it should still do the right thing (there's a "test" for you in the code).

Question 8: How to find employees who are underpaid and overpaid by adding color encoding?

We've created the basic code to render a new bar chart for you (see the bottom of the HTML file), but we still need to encode additional things. We want to know how much each person is under/overpaid (in this case we'll use bar colors).

Step 8.1:

In order to find employees who are underpaid or overpaid easily using our chart, we are going to add data stored in the field `overunder` using color encoding. For overpaid employees, the bar representing their salary should be darker in comparison to these of underpaid employees.

To make this work you will need to do a very similar thing for overunder and colors as you did above for salary(similar to Q5 and Q6). In this case we want to map the min and max overunder (-0.5 to 0.5) in this case to a range between 0 and 255 (recall that rgb values range from 0 to 255).

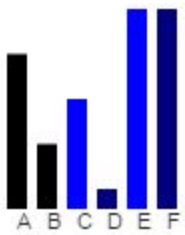
- ☐Complete `function minOverunder(data)`
- ☐Complete `function maxOverunder(data)`
- ☐Store the min and max overunder information in a dictionary `overunderRange`.
  E.g. `overunderRange={"min": minOverunder(data), "max": maxOverunder(data)}`

Step 8.2:

☐ Implement the f`unction calculate_color(overUnder){}` which returns the scaled color value for each overunder value.

calculate_color(-0.5) should give you back 0,
calculate_color(0) should give you back 128,
calculate_color(0.5) should give you 255 (for this particular setup).

HINT 1: Before you try programming this, write out the equation to solve this. For this example, you're basically mapping the input range to the output range (see Figure 1):
HIINT 2: rgb() expects integer values. You should round numbers like 127.5 to 128 (see Math.round) when you get to the last step
HINT 3: While you can "bullet proof" your code and make it work if the input range > 0, but the reality is that this isn't necessary given the current framing. The minimum overunder will always be <= 0.
If everything works correctly, you will see an image similar to this:



Question 9:

Argue that this solution is or isn't expressive given our initial question: How much does each employee make and are they being over or under paid? Can you think of a more effective way to answer this question? (Please write your answer in the html file.)

Extra credit:

Modify the visualization to make it more effective based on your answer to question 9. You can make a simple modification to the bar chart or you can go crazy and make a completely different visualization.

If you do the extra credit, please copy your file after the 9 questions above and create a new one that you will modify. Turn this in as: userid_extracredit.html (replacing userid with your umichid)