TA Lecture Notes

December 3, 2014

Announcements:

- Final Project:
  - Last lab this week
  - Demo to two TAs that will be in the lab, no presentation or report needed
  - Upload your project and screenshot of the best part of your project to the website
  - Grading based on the demo in lab
- Final Exam:
  - 15th (for AL1) and 19th (AL2) from 8 AM to 11 AM
  - Room assignments will be posted soon
  - Review material will be posted by coming Monday

D3: Data Driven Documents (http://d3js.org/)

D3 is a JS library, made by other developers.

Download the code from Week 14 Part 2 on CS 105 website: It has an HTML page, JS code and an excel document

Midterm 2 Grades Visualization

1. Open HTML doc
   a. The JS file (d3.min.js) is imported into this file
   b. Don't worry about understanding the JS file, we just have to use it
2. The excel file contains all the midterm 2 grades in random order
   a. Task is to import this data into the script we will write inside the HTML file between the tags `<script type="text/javascript"> </script>`
3. Copy the column of grades and paste it into another excel document but transpose it into a row using paste special and save as grades.csv, a comma separated values file in the same folder as the excel file
   a. Open the csv file to check comma separated values format
   b. This is now in a format compatible with JS
4. In HTML file, between the script tags, write `var data = [//paste all the csv values from the grades.csv file];`
   a. i.e. copy all the data and make an array called 'data'
5. Let's make 'w' and 'h' as the variables for width and height
6. Add lines `var w = 1000;` and `var h = 500;`
7. Set up a drawing area in D3, it is called "svg", a simple vector graphic
8. `var svg = d3.select("body").append("svg").attr("height",h).attr("width",w);`
   a. Create a drawing area named svg (`var svg`..)

   b. set the height and width to 500 pixels and 1000 pixels respectively
   c. `.attr(param1,param2)` is the syntax for adding attributes with name param1 and value `param2`
9. Type this line next :
   `svg.selectAll("svg").data( data ).enter().append("rect");`
   a. i.e. select the area you want to work in (`"svg"` that we created in point 6 above) and feed the data (using `.data()`) in that area and processes that data (using `.enter()`)
   b. `.data(data);` first data is part of the command and the second is the variable we created in point 4 above
   c. `.append("rect")` is for making a rectangle for every grade
10. What you want to do with each data is dictated by what follows next
11. For every piece of data, we want to make a rectangle and we need width, height and position for that rectangle
12. After the `.append("rect")` part of the line from point 9, add these lines from points 13 through 16 below (add them one below the other to make your code more readable and manageable)
13. `.attr("width", w / data.length).attr("height", function(d,i) {return (d/150)*h;})`
   a. Width for every rectangle corresponding to individual grades is fixed `w / data.length`
   b. We are using a function here, to get heights of the rectangles for grades
   c. 'i' is the index and 'd' is the data in the for loop that is being implemented by the function when used in the format `function(d,i)`
   d. `(d/150)*h` is the height of the individual rectangle, here'd' is the grade value. The maximum grade on the test was 150.
   e. The function is a for loop that is going through all grade values
14. `.attr("x", function(d,i){return i*(w/data.length);})`
   a. this line is for the 'x' location, first rectangle starts at the edge (i=0) and then so on for locations of other rectangles
15. `.attr("y",0);`
   a. All rectangles start at the same y location
   b. x=0 and y=0 are at the top left corner
16. `.attr("fill","black")`
   a. To color the rectangles black
17. We have a chart at this point, the data is not sorted though
18. Let's sort the data, add `data.sort()` after the `var data …` line
   a. this is sorting as per the ASCII values, so the people with 100 or better are on the left in the data
   b. but we want sorting in terms of actual numerical values and not ASCII

   c. so we use a function inside the sort function
19. change the line above to `data.sort(function (a,b){return (a-b);})`
   a. read this reference to understand this usage http://www.w3schools.com/jsref/jsref_sort.asp
   b. go to the Parameter Value section of the above link
20. Alternatively, we could've sorted the data in excel itself
21. Now the chart should be for the sorted data, but is still kind of upside down
22. Let's align the bars at the bottom of the screen
23. Change the "y" location attribute for this
24. `.attr("y",function(d,i){return h-((d/150)*h);})`
   a. We have moved down by the amount of white space we had available, i.e. the bars are aligned at the bottom now since the y is that of the top left corner of individual rectangles at this point
25. Let's color the chart based on performance, by changing the "fill" attribute
26. `.attr("fill",function (d,i){if (d/150 > .9) return "green"; else if (d/150<0.6) return "red"; else return "black";});`
   a. To separate grades that were >90% (green) and <60% (red)
   b. You can use `rgb(146,208,80)` as the color, the numbers give the proportions of red green and blue