Lab Assignment 5: Data Visualization with D3.js

Due: 11:59 PM Feb 18, Wednesday

In this lab, you will learn the basics of D3.js, another popular tool for interactive visualization. It's better if you have a background knowledge of HTML, CSS, JavaScript, but you should be able to do the lab even if you do not. An excellent 10 minutes overview of these fundamentals is on website, alignedleft.com. To be more proficient with D3.js, you are encouraged to go through tutorial,visualization with D3.js (time estimate: 10-15 hours).

Submission Please submit all your writeups, figures through blackboard.

1 A simple example of plotting circles

This tutorial illustrates how to plot circles, bind data with the plots, plot word cloud.

1. Open Chrome. (You should be also able to do the same on other browsers such as firefox and IE, but the instructions are specific to Chrome).

2. Open the web page http://nymph332088.github.io/CIS4340/index.html.

3. Stay on the page, open `Chrome -> More tools -> Developer Tool`. Or click `F12` button on the keyboard to open `Developer Tool` (depend on the version of Chrome). You will see tabs such as `Elements, Sources, Console, ....` Click on `Elements` tab. If you mouse hovers over any tag, the corresponding content will be highlighted on the web page with a blue background. Internal CSS styles are included within `<style></style>` tag under the `<head>`. External CSS sheets are linked in the head too.

4. Copy all the text on D3.js page to the Console. This will load D3.js functionality to the Console.

5. Type `var header = d3.select('#header');`. This will create an array variable which contains a DOM element referring to the header in the HTML. DOM stands for **D**ocument **O**bject **M**odel.

6. Third, we can use D3 chainable functions to manipulate the DOM element referenced by the header variable. For example, change the header of the page by typing `header.select('h1').text( 'Hello World!');` or change the header background color by typing `header.select('h1').style( 'background-color', 'green');` in the Console.

7. Let us draw a circle with D3. D3 draws **S**calable **V**ector **G**raphics, which is a text-based image format. Let us define a variable called `var svg = d3.select('#footer').append('svg');`. Then, set the width and height attributes of the SVG canvas by typing `svg.attr( 'width',60).attr( 'height',50)`. Then, draw a circle on the canvas by typing `var circle = svg.append('circle');`.

Then, you can also set the attributes of the circle to specify its appearance, type `circle.attr( ‘cx’,’25).attr( ‘cy’,’25’).attr( ‘r’,22)` to see the circle in the top right corner. Continue to set `circle.attr(‘fill’,‘blue’).attr(‘stroke’,‘gray’).attr(‘stroke-width’,’2’);`. (Notice the difference between `attr()` and `style()`). For more appearance of circles, here is the reference. Where is your circle placed and why? See SVG primer.

8. D3 data binding. Suppose you need to draw 5 circles, with radius of each specified by a different value. Would you append 5 circles by typing the above codes 5 times? Is there any easier way? Yes. Let’s do it. First, reload the page to clear what you did before. Then copy all the text again on D3.js page to include D3.js. Now, draw a bigger canvas by typing `var svg = d3.select(‘#footer’).append(‘svg’); svg.attr( ‘width’,500).attr( ‘height’,80);`. Let us define the radius array, `var radius= [10, 15,20, 25, 30];`. Then, declare 5 place holders on the canvas for the 5 circles in the future, typing `var circles = svg.selectAll(‘#varycircle` Then, type `circles = circles.append(‘circle’);`, you will append an circle node for each place holder. Then, type `circles.attr( ‘r’, function(d) {return d;})`. Then, specify the x position so that the 5 circles are not overlapping, `circles.attr(‘cx’,function(d,i){return (i*50) + 25; });`. To specify y position, `circles.attr(‘cy’,30)` or `circles.attr(‘cy’,function(d` `30;});`. (Data types in JavaScript are kind of similar to Python.)

1.1 Assignments

Plot 4 squares horizontally with 4 different colors and 4 different sizes.

2 More complicated word cloud example [1]

Figure 1 shows the word cloud of the sample tweets data. The size of each word is proportional to the word count of all the tweets. You can see the basketball has the largest size, that is because the example twitter data is crawled by keyword ‘basketball’. The program is set to show up to 500 words. We also see kobe, LSU, and some video links be the top 500 most frequent words.

Now, we illustrate how to plot the word cloud based on your own data.

1. Reload the page to clear what you did before. Then copy all the text again on D3.js page to include D3.js.

2. To plot word cloud, you need to write hundreds of code, but the good news is someone already did it. Let’s copy and paste all the text into the console on the page auxiliary_functions.js. Now, let’s call those functions to create a simple word cloud.

3. Open the link wordcloudgeneration.js. You can copy and paste all the code directly to the console. When you press enter, you will see a word cloud for a sample string `var textdata`. The last two lines specify a string variable `var textdata` and call the `load()` function. Try to modify the variable `textdata` to another string and call `load()` function again.

[1] Based on Jason David’s code

2.1 Assignments

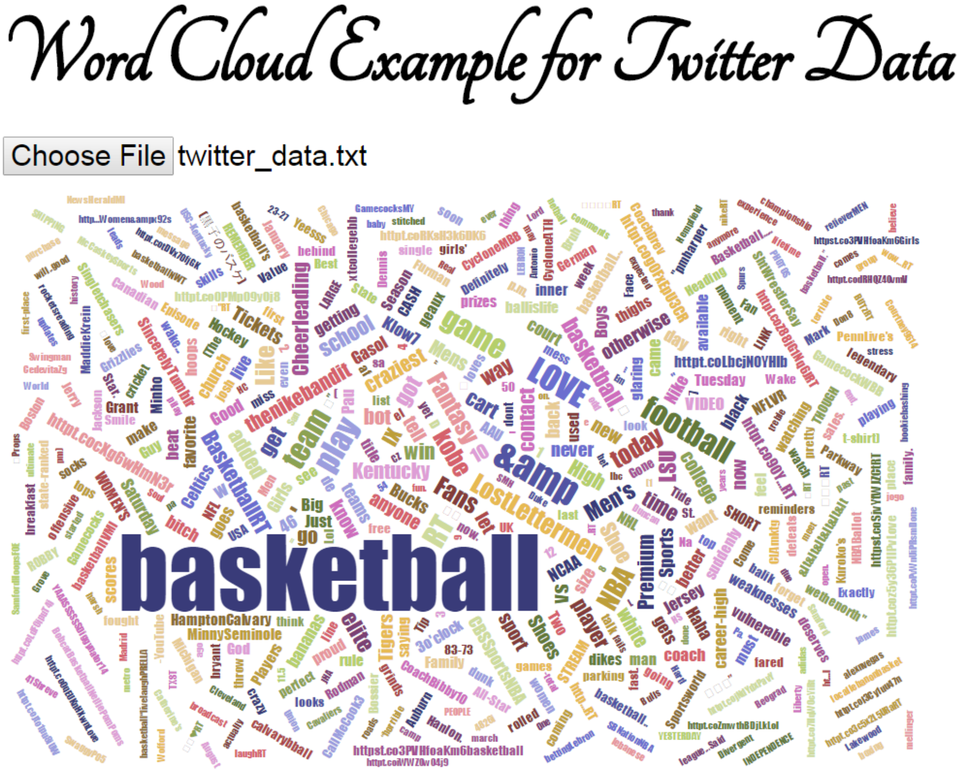
In Lab 2, you downloaded tweets data from different locations, e.g., Philadelphia and Boston.

1. Show the word clouds of tweets respectively that you crawled from two places.
2. Show the word cloud of the most recent two days of the two locations.
3. Show any word cloud that might interest you.
4. Write up your discoveries.

3 Extra Credits (30 points)

After you got the basic ideas of D3.js, it's time to explore yourself. The extra credit is to use other D3 visualization with any data we have used before, such as the auto car dataset, twitter data set or the household power consumption data. Some links might be insightful:

1. http://christopheviau.com/d3list/gallery.html (Basic D3 gallery)
2. http://www.gapminder.org/world/ (Wealth of nations)
3. http://www.jasondavies.com/wordcloud (word cloud)
4. http://mbostock.github.io/d3/talk/20111116/iris-splom.html (scatterplot)
5. ordered heatmaps
6. http://neuralengr.com/asifr/journals/ (showing 3 attributes)
7. http://square.github.io/cubism/ (time series)
8. http://bl.ocks.org/wizicer/f662a0b04425fc0f7489 (CS skills in time)
9. http://windhistory.com/map.html#4.00/36.00/-95.00 (spatial data)
10. http://www.trulia.com/vis/tru247/ http://www.trulia.com/vis/metro-movers/ (spatial temporal data)
11. http://remittances.herokuapp.com/?en (spatial temporal)
12. http://bost.ocks.org/mike/miserables/ (graph data)
13. http://www.brightpointinc.com/interactive/political_influence/index.html?source=d3js (circular plot)
14. http://air.nullschool.net/ (amazing demo)
15. http://bl.ocks.org/fhernand/9a9f93f2a6b0e83a9294 (passes in a game)



Word Cloud Example for Twitter Data

twitter_data.txt

Figure 1: Word cloud for tweets data