CSE6242 / CX4242: Data and Visual Analytics | Georgia Tech | Fall 2014

Homework 2:
D3 Graphs and Visualization
Due: Wednesday, October 1, 2014, 11:59PM EDT

Prepared by Alan Zhang, Drew Wei, Brian Minsuk Kahng, Seungyeon Kim, Polo Chau

Submission details: Submit a **single zipped file**, called “`HW2-YOUR_LAST_NAME-YOUR_FIRST_NAME.zip`”, containing all your deliverables including source code, scripts, data files, etc. Read all instructions carefully about the deliverables for each question.

If you collaborate with other students on this assignment, please specify their names in the text box on the T-square submission site. Each student must write his/her own answers. All GT students must observe the honor code. **Don’t wait until the night before! Start early.**

Q1: Force-Directed Graph Layout in D3 [55 pts]

In Q1, you will experiment with many aspects of D3 for graph visualization. To get you started, we provide **graph.html** for you here. Currently it is only an uncolored node graph, with each node representing a character from Victor Hugo’s *Les Misérables*. In the following steps, we will train you to build more powerful features and functionality. This code requires a json data file as the input, which describes the graph’s edges, nodes and their attributes. Download the json file here.

Note: You may need to setup your own HTTP server to properly run your D3 visualization. The easiest way is using SimpleHTTPServer in Python. See this for details.

1. [5 pts] **Adding node labels.** Modify the html to **show a label** to the right of each node in the graph. The label should be the name of the character that node represents. If a node is dragged, its label must also move with the node.

2. [5 pts] **Coloring nodes.** Color the nodes based on the “groups” field provided in the json file (i.e., all nodes from the same group have the same color). You **need to choose the colors**. The goal is to make the groups visually distinguishable from each other. We suggest using a color scheme (can be grayscale) that also works for people with colorblindness. (Hint and color brewer.)

3. [10 pts] **Scaling node sizes.** Adjust the radius of each node in the graph based on how “cool” each character is. In the provided json file there is a “coolness” rating for each character.
    a. [3 pts] Use this rating to scale the radii of the nodes **linearly**. This means cooler characters (higher score) will be represented as larger nodes. Take a screenshot of the whole graph with linearly scaled node size (Polo recommends Skitch for

taking screenshots).

   b. [3 pts] Now scale the radii by the **square root** of coolness scores. Take a screenshot of the whole graph with square root scaled node size.
   c. [4 pts] In fewer than 40 words, discuss which approach works better for this problem and why in **short_answers.txt.**

4. [10 pts] **Filtering node labels.** Only show the labels for "cool" characters, whose coolness factors are **above 25** (i.e., coolness > 25). Compare with the previous version and explain in **short_answers.txt.**

5. [10 pts] **Pinning nodes (fixing node positions).** Modify the html so that when you **double click** a node it fixes the node's position. Mark fixed nodes so that they are **visually distinguishable** from unfixed nodes, e.g., pinned nodes shown in a different color, or border thickness, or visually annotated with a "star" (*), etc. The rest of the nodes' positions should remain unfixed. Double clicking a fixed node should **unfreeze** its position and **unmark** it.
   Hint: the easiest way to do this is to add a doubleclick event-listener to the nodes.

6. [10 pts] **Tooltips.** Using d3-tip library, add a tooltip for each node. When **mouse over** the node, a tooltip should appear displaying its name, group and coolness.
   Note: You should also add tooltips for nodes whose labels have already been filtered out in Q1.4.

7. [5 pts] Improve the visualization by making the text labels more readable (Hint: balancing font size and the graph layout parameters to reduce clutter).
   a. [3 pts] The changes you make should be included in the final version of **graph.html**.
   b. [2 pts] In 40 words or fewer tell us what you did to make your graph less cluttered in **short_answers.txt**.

Q1 Deliverables:

- **The directory structure should look like (don't forget to add d3 library):**

```
Q1/
   graph.html
   miserables.json
   short_answers.txt
   linear_nodes.yyy
   squareroot_nodes.yyy
   d3/
      d3.v3.min.js
      d3-tip.js
```

- **graph.html** - the html file based off our initial code that contains the changes made in 1-6 above.

- **short_answers.txt** - the text file containing your short answers for Q1.3-c, Q1.4, and Q1.6-b. Keep it succinct (each answer should be less than 40 words).
- **linear_nodes.yyy** - a png, jpg, or pdf screenshot of the linearly scaled nodes from Q1.3-a.
- **squareroot_nodes.yyy** - a png, jpg, or pdf screenshot of the square root scaled nodes from Q1.3-b.

Q2: KAA Data Scientist: Visualizing Car Data using D3 and Tableau [45 pts]

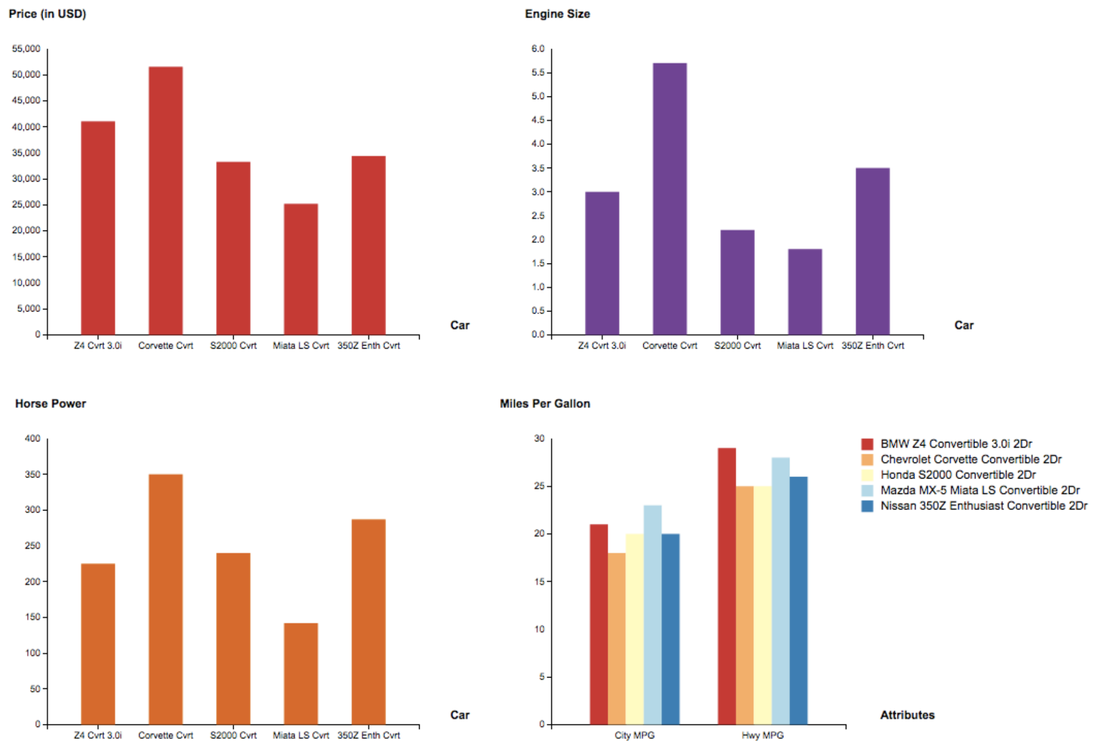
After the training in Q1, assume you become a data scientist at KAA, the largest website providing automobile information. Your job is to analyze and present data to customers.

1. [5 pts] Download this dataset of 428 new vehicles from year 2004[1]. Convert the dataset into a json file **(cars.json)**. This can be done by hand, a tool, or a script of your choice. Only the json file will be graded so you don't need to turn in the script.

2. [10 pts] **Table.** Make a table to display cars of which the retail prices are higher than $70,000. You can use **any tool** (e.g., Excel, HTML, Tableau, D3) you want to make the table. Keep suggestions from class in mind when designing your table (see lectures slides for what to and what not to do, but you are not limited to the techniques in the slides). Your visualization needs to convey the data clearly and effectively to the reader. No user interaction is required. Describe your reason for choosing the techniques you use in **explanation.txt** in fewer than 40 words**.**

3. [15 pts] **Bar charts.** A Sports Car enthusiasts has locked down 5 targets and wants to visually compare the properties of these cars. ***Using D3 & Tableau***, pick any **5** cars from the **Sports Car** category and use **bar charts** to visualize the differences in the 5 most important attributes -- **Retail Price, Engine Size, HP, City MPG and Hwy MPG**.
    a. [10 pts] **D3:** Visualize these attributes in **4** separate bar charts (one chart for Retail Price, one for Engine Size, one for HP, and one side-by-side bar chart for City MPG and Hwy MPG). We need separate charts since the scales are different for most of the attributes. All charts must be shown on the same webpage. Below are example **bar charts** we would like you to use. For all other properties of the chart, use what you have learned from class (see lectures slides) to make it effective in conveying information and visually pleasing (e.g., include axis labels, choose a good color scheme, etc).

[1] This data set is from http://www.idvbook.com/teaching-aid/data-sets/2004-cars-and-trucks-data/



b. [5 pts] **Tableau:** Visualize City MPG and Hwy MPG only, in one chart. Tableau is a popular infoviz tool and the company has provided us student licenses. Go to http://www.tableausoftware.com/tft/activation, and select Get Started. On the form, enter your Georgia Tech email address for "Business email" and "Georgia Institute of technology" for "Organization". The Desktop Key for activation is on T-Square. (Please don't share the key outside the class.)

The exact choice of colors (can be grayscale) and size is up to you. Explain your choices in 3a and 3b in **explanation.txt** in fewer than 40 words.

4. [15 pts] **Creative visualization. Using D3**, construct a creative visualization that compares 4 manufacturers of your choice (e.g., Audi, Ford, Honda, etc.) You should use 5 attributes (**Retail Price, Engine Size, HP, City MPG and Hwy MPG**) used in Q2.3. Note that we are not comparing car models but car manufacturers, so you need some data pre-processing. For example, you might want to aggregate values of cars (e.g. average) to get the data values for manufactures. You can have one large visualization or multiple small ones. (The visualization does **not** need to support any interactions.)

- **Do not turn in a bar graph or table.** The point of this task is for you to take a design idea you've created yourself and implement it in D3 as best you can.

Points will be awarded for functionality, and also for interesting ideas.

- Discuss the brilliant idea that you come up with in order to make this creative visualization in **explanation.txt** in fewer than 40 words**.**

Q2 Deliverables:

- **The directory structure should look like (don't forget to add d3 library):**

```
Q2/
   cars.json
   table.yyy
   bars.html
   bars.xxx (from Tableau)
   comparison.html
   comparison.json
   explanation.txt
   d3/
      d3.v3.min.js
```

- **cars.json** - The json file created in Q2.1
- **table.yyy** - Any modern image format (e.g., jpg, png, pdf) showing the table in Q2.2
- **bars.html** - The html and javascript to render the bar graphs requested in Q2.3
- **bars.xxx** - The figure for bar charts generated from Tableau requested in Q2.3. You can use formats like png and pdf, but be sure to make it a high-quality and clear image).
- **comparison.html** - The html and javascript to render the comparative visualization made in D3 for Q2.4.
- **comparison.json** - The json file used in Q2.4 (unless you use cars.json)
- **explanation.txt** - Include Q2.2, Q2.3 and Q2.4. Keep it succinct.
- **Any other json files -** If you use your own json files as the inputs for Q2.2 and/or Q2.3 other than cars.json.