

Homework 2

D3 Graphs and Visualization

Due: February 20, 2015, 11:55PM EST

Prepared by Meera Kamath, Yichen Wang, Amir Afsharinejad, Chris Berlind, Polo Chau

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

You may collaborate with other students on this assignment, but **each student must write up their own answers in their own words**, and must write down the **collaborators’ names** on T-Square’s submission page. If you are using “slip days”, mention that while submitting.

Q1: Force-Directed Graph Layout in D3 [50 pts]

*We anticipate the time to complete this question to be around 5 hours.

In Q1, you will experiment with many aspects of D3 for graph visualization. To get you started, we provide **graph.html** for you [here](#). Each node of the graph represents 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 from [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 *get 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 [Sketch](#) 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. Place this response in **explanation.txt**.
4. [5 pts] **Filtering node labels.** Only show the labels for “cool” characters, whose coolness factor is above 25 (i.e., coolness > 25). Compare this with the earlier approach in part 3? Place this response(less than 40 words) in **explanation.txt**.
 5. [5 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 can be 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. [5 pts] Contain the graph within a fixed boundary i.e. you should not be able to drag the nodes outside the boundary created.
 7. [10 pts] **Tooltips.** Using [d3-tip](#) library, add a tooltip for each node. A “**mouseover**” event on the node, should display a tooltip containing the name, group and coolness associated with the node.
Note: You should also add tooltips for nodes whose labels have already been filtered out in Q1.4.
 8. [5 pts] Improve the visualization by making it easier to read the text labels. There are several possible routes to making the text 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** that you turn in.
 - b. [2 pts] In 40 words or fewer tell us what you did to make your graph less cluttered in **explanation.txt**.

Q1 Deliverables:

- The directory structure should look like (remember to include the 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 on the initial code that contains the changes made in 1-8 above.
- **explanation.txt** - the text file explaining your approach for Q1.3-c, Q1.4, Q1.8b. 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: Visualize the EuroVision Song Contest entries [40 pts]

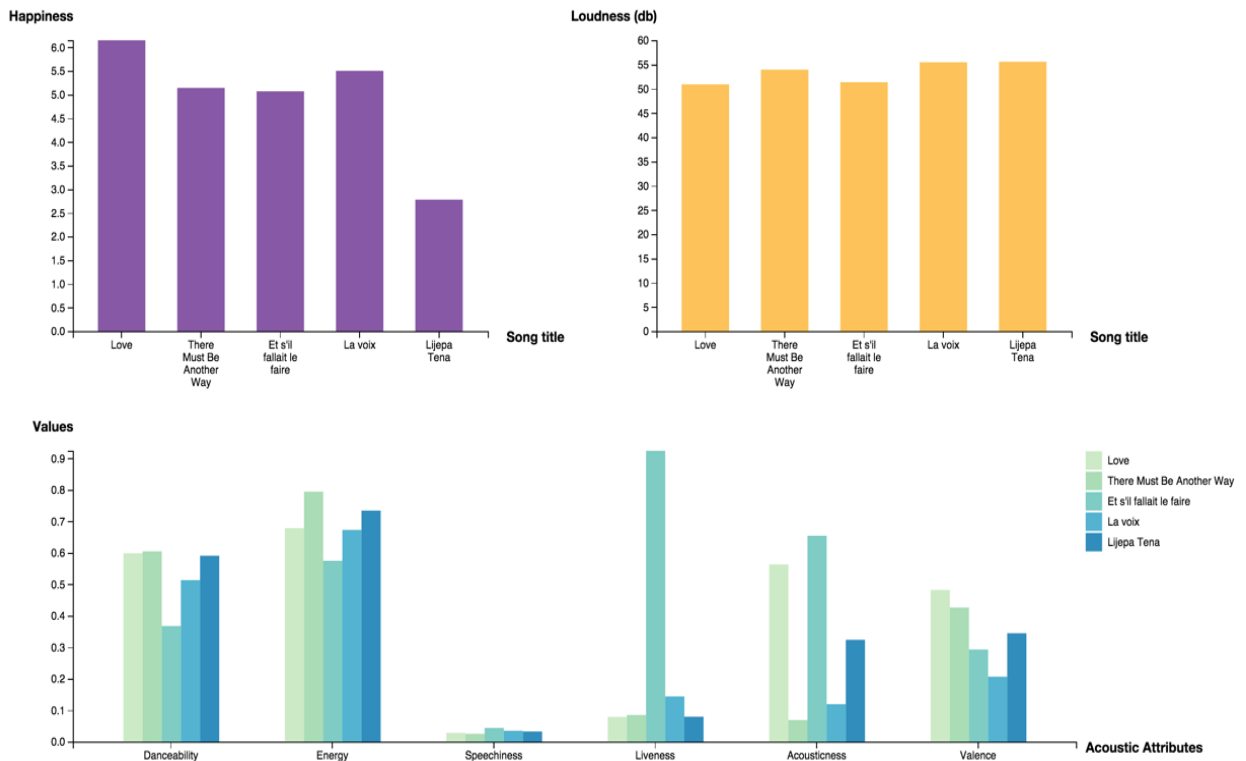
*We anticipate the time to complete this question to be around 7 hours.

1. [5 pts] Download [this](#) dataset of 648 songs that were part of the Eurovision Song Contest over the years 1998-2012.¹ A description of the columns can be found [here](#). Convert the dataset into a json file (**songs.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. [5 pts] **Table**. Make a table to display details of the finalists in the competition in the year 2005. 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. Using D3 & Tableau**, pick any **5** songs from the dataset and use **bar charts** to visualize the differences in the attributes: "Happiness", "Loudness" and various "Acoustic" properties of the songs.
 - a. [10 pts] **D3**: Visualize these attributes in **3** separate bar charts (one chart for Happiness, one for Loudness, and one side-by-side bar chart for the six Acoustic attributes - danceability, energy, speechiness, liveness, acousticness, and valence). 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

¹ This data set is from <http://www.student.dtu.dk/~s093020/dataAnalysisWebsite/#dataHeader>

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).

Note: The loudness attribute has values in the range [-60, 0], with -60 representing silence and 0 being extremely loud. While plotting the chart, make sure you transform the values to the range [0, 60] so that 0 represents silence, and 60 the highest loudness possible.



- a. [5 pts] **Tableau:** Visualize the six acoustic attributes used above (danceability, energy, speechiness, liveness, acousticness, and valence) 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](#). Do not share the key with anyone.

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 songs in 5 countries, of your choice. 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 (remember to include the d3 library):

Q2/

songs.json

table.yyy

bars.html

bars.xxx (from Tableau)

comparison.html

comparison.json

explanation.txt

d3/

d3.v3.min.js

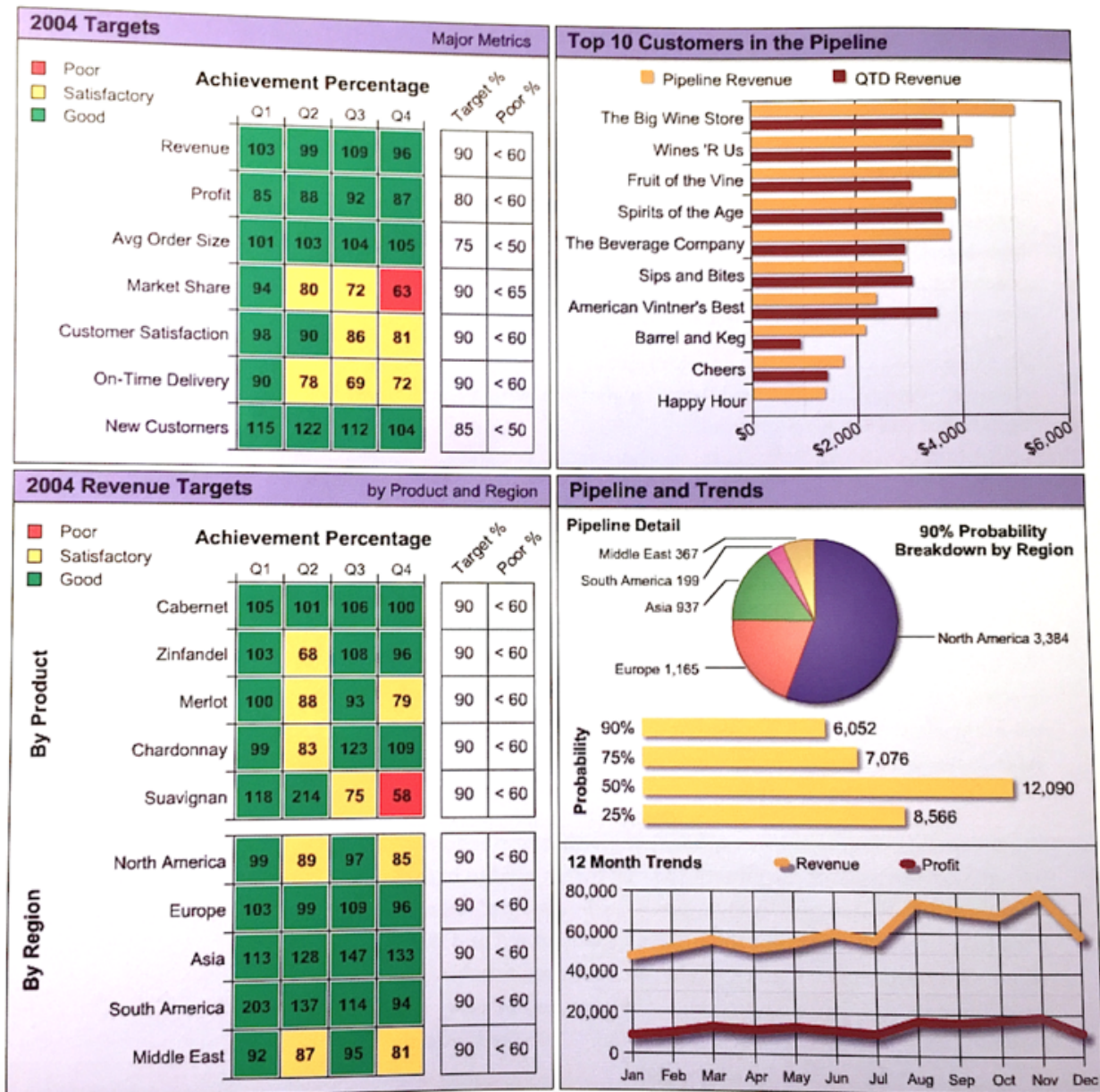
- **songs.json** - The json file created in Q2.1
- **table.yyy** - Any modern image format (e.g., jpg, png, pdf) showing the table in Q2.2
- **bars.html** - The html and javascript to render the bar graphs requested in Q2.3
- **bars.xxx** - The figure for bar charts generated from Tableau requested in 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 (i.e. if you decide to use a subset of songs.json)
- **explanation.txt** - Write the explanation for parts Q2.2, Q2.3 and Q2.4 in this file. 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 songs.json.

Q3: Dashboard Design [10 pts]

*We anticipate the time to complete this question to be around 1 hour.

Identify five items that you think could have been designed better in the sales dashboard shown below and suggest improvements for the same. Write down your answers in a file

called explanation.txt, in fewer than 100 words. To help us more easily grade your work, divide your writeup into 5 paragraphs, one for each pair of “issue” and “improvement”.



Q3 Deliverables:

Q3/

explanation.txt