CSE6242 / CX4242: Data and Visual Analytics | Georgia Tech | Spring 2016

Homework 2 : D3 Graphs and Visualization

Due: Friday, February 26, 2016, 11:55PM EST

Prepared by Gopi Krishnan Nambiar, Nilaksh Das, Pradeep Vairamani, Ajitesh Jain,

Vishakha Singh, Polo Chau

Submission Instructions:

It is important that you read the following instructions carefully and also those about the deliverables at the end of each question or you may lose points.

❏ Submit a single zipped file, called
  "HW2-{YOUR_LAST_NAME}-{YOUR_FIRST_NAME}.zip", containing all the
  deliverables including source code/scripts, data files, and readme. Example:
  'HW2-Doe-John.zip' if your name is John Doe. Only .zip is allowed (no .rar, etc.)

❏ You may collaborate with other students on this assignment, but you must write your
  own code and give the explanations in your own words, and also mention the
  collaborators' names on T-Square's submission page. All GT students must observe the
  honor code. Suspected plagiarism and academic misconduct will be reported and
  directly handled by the Office of Student Integrity (OSI). Here are some examples
  similar to Prof. Jacob Eisenstein's NLP course page (grading policy):

  ❏ OK: discuss concepts (e.g., how cross-validation works) and strategies (e.g., use
    hashmap instead of array)

  ❏ Not OK: several students work on one master copy together (e.g., by dividing it up),
    sharing solutions, or using solution from previous years or from the web.

❏ If you use any "slip days", you must write down the number of days used in the
  T-square submission page. For example, "Slip days used: 1". Each slip day equals 24
  hours. E.g., if a submission is late for 30 hours, that counts as 2 slip days.

❏ At the end of this assignment, we have specified a folder structure of how to organize
  your files in a single zipped file. 5 points will be deducted for not following this strictly.

❏ Wherever you are asked to write down an explanation for the task you perform, stay
  within the word limit or you may lose points.

❏ In your final zip file, do not include any intermediate files you may have generated to
  work on the task, unless your script is absolutely dependent on it to get the final result
  (which it ideally should not be).

❏ After all slip days are used up, 5% deduction for every 24 hours of delay. (e.g., 5 points
  for a 100-point homework)

❏ We will not consider late submission of any missing parts of an homework assignment or
  project deliverable. To make sure you have submitted everything, download your
  submitted files to double check.

Grading

The maximum possible score for this homework is 120 points, which includes 5 bonus points. Students in the undergraduate section (CX4242) can choose to complete any 100 points worth of work to score the full 15% of the final course grade. Students in the grad section (CSE6242) can choose to complete any 115 points. So, for example, if an undergraduate student scores 120 pts, they will receive (120 / 100) * 15 = 18 pts towards their final course grade.

Part 0: Prerequisites

Download this zip file, which contains all the datasets to be used in this assignment.

You may need to setup your own HTTP server to properly run your D3 visualizations (as we mentioned in the D3 lecture). The easiest way is using SimpleHTTPServer in Python. See here for details.

You can and are encouraged to decouple the style, functionality and markup in the code for each question, i.e., you can use separate files for css, javascript and html.

Include all the files related to d3*.js in the lib folder, and use a relative path to reference these files from html files in other folders, i.e., Q1, Q2, etc.

Part 1: Visualizing Data using D3 [35 pts]

1. **[10 pts]** Use the data[1] provided in the file *ForestFires.csv* (in the Q1 folder) to create a scatter plot visualization. Refer to the tutorial for scatter plot here.

   Features/ attributes of the data:
   1. Month of the year
   2. Wind speed in km/h
   3. Burnt area of the forest in hectares

   a. [6 pts] **Creating scatter plot**:
      Create a scatter plot with the following specifications:
      1. X-axis represents the wind speed, in *linear scale*
      2. Y-axis represents the burnt area, in *linear scale*
      3. Data points are shown as squares. All squares have same size.
      4. Use a *categorical scale* to assign different colors for squares corresponding to data points having different months
      5. Add a legend of how colors map to the months
      You can choose any appropriate range for linear scale in parts 1 and 2, and any appropriate size for squares in part 3. Take a screenshot of this scatter plot and include it in **scatterplots.pdf**.

   b. [3 pts] **Symbols and Log scale:**

---

[1] This is a subset of data taken from http://archive.ics.uci.edu/ml/datasets/Forest+Fires

Modify the above plot to:
1. Use circles for data points
2. Set the size (radius in pixels) of each circle in the plot to be proportional to square root of the the burnt area
3. Use *log scale* for Y-axis

Take a screenshot of this scatter plot and include it in **scatterplots.pdf**.

c. [1 pt] Explain in no more than 20 words why using log scale for Y-axis is better than using linear scale for this data. Write down your answer in **scatterplots.pdf**

Q1 Deliverables:
- **The directory structure should be as follows:**
  Q1/
      scatterplot.(html / js / css)
      scatterplots.pdf
      ForestFires.csv

- **scatterplot.(html / js / css)** - the html/js/css files after Q1.b. You do not have to submit code for Q1.a
- **scatterplots.pdf** - screenshot of the two scatter plots created in Q1.a and Q1.b and answer to Q1.c

2. **[10 pts]** Use the dataset[2] provided in the file *nepal.csv* (in the Q2 folder) to create a **stacked bar chart** visualization.This question requires to visualize the number of teachers in Nepal from year 2007-2011.

a. [3 pts] Manipulate the data file **nepal.csv** into **nepal_new.csv** such that the file looks like Figure 1. You must write a script in any language (Python, Java etc.) to do the same. Submit the script **as Q2.yyy**.

```
District,2007,2008,2009,2010,2011
Bajhang,2011,2284,3066,3104,1875
Dadeldhura,1419,1702,2021,2061,1261
Arghakhanchi,1859,2284,2365,2535,2063
Darchula,1483,1505,1838,1957,1382
Chitwan,4468,4634,6488,6488,1527
Panchthar,2063,2173,3177,3184,2096
Sindhupalchok,2699,2699,3201,3460,1802
Udayapur,2230,2230,2908,2988,1924
Mugu,486,486,589,682,468
Khotang,2164,2257,2294,2763,902
Kathmandu,18501,18501,18018,18018,10894
Banke,2900,3200,3535,3845,2114
Gorkha,3156,3319,3483,3485,2185
Baitadi,2276,2525,3073,3080,2016
Jajarkot,1477,1479,1647,1655,909
Taplejung,1733,1824,2255,2321,1669
Dolpa,523,523,533,635,480
Kavrepalanchok,6504,6504,6989,7136,4416
```

Figure 1

b. [1 pt] Select any 9 districts from the file and use appropriate scale to adjust them on the x-axis.
c. [1 pt] Select appropriate formatting technique to present the y-column as shown in Figure 2.
d. [2 pts] Add a legend to the graph such that each colour clearly represents the years from 2007 to 1011.
e. [3 pts] *description.txt:* Explain all the methods used to create the graph in not more than 50 words. The description must *explicitly* mention the methods.

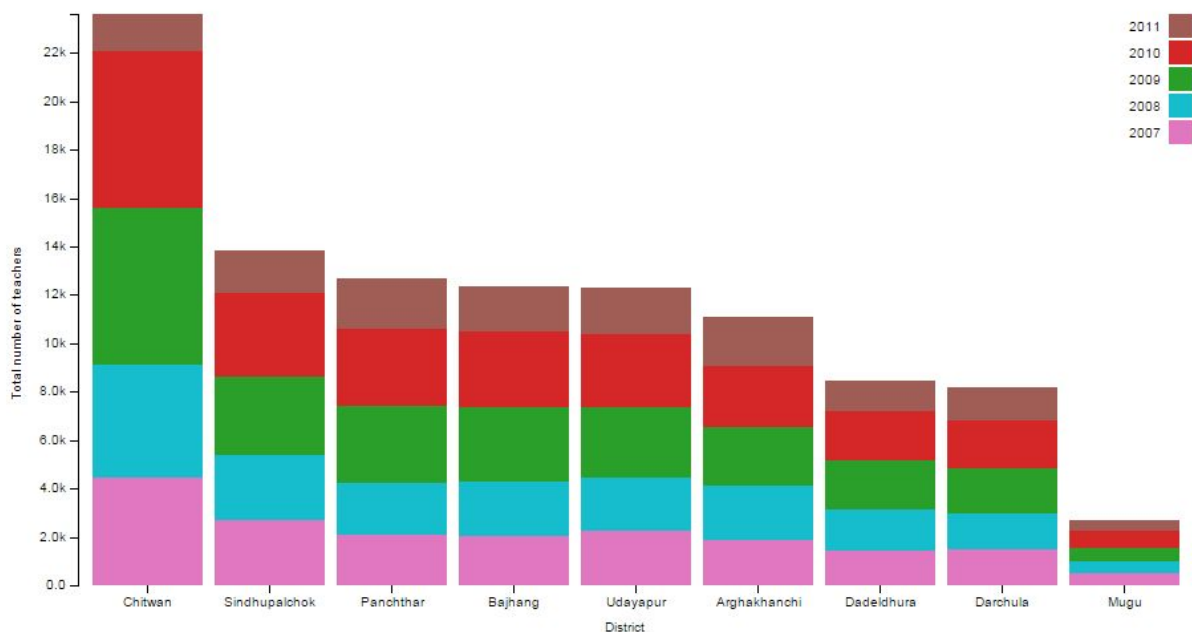Take a screenshot of the map and save it as **stacked.pdf**.

Figure 2

Q2 Deliverables:

- **The directory structure should look like (remember to include the d3 library):**
  - **Q2/**
    - Q2.yyy (Python, Java etc.)
    - nepal_new.csv
    - stacked.pdf
    - stacked_barchart.(html / css / js)
    - explanation.txt

- **Q2.yyy** - The script (in Python/Java etc.) used to manipulate the data file nepal.csv.
- **nepal_new.csv** - the resultant .csv file
- **stacked.pdf** - Screenshot of the resultant graph.
- **stacked_barchart.(html / css / js)** - html, javascript, css files to render the visualization made in Q2.
- **explanation.txt** - The text file explaining all the methods used to create the graph.

4

3. **[15 pts] Force-Directed Graph Layout in D3**

   You will experiment with many aspects of D3 for graph visualization. To get you started, we provide graph.html (in the folder Q3).

a. [4 pts] **Adding node labels**: Modify the graph.html to show labels to the right of each node in the graph. The label should be the name of company that the node represents. If a node is dragged, its label must also move with the node. (You are free to split graph.html into graph.html, graph.js and graph.css)

b. [2 pts] **Coloring links**: Color the links based on the "type" field in the links array. Assign the following colors:

Type	Color
licensing	blue
suit	red
resolved	green

c. [4 pts] **Pinning nodes** (fixing node positions). Modify the html so that when you double click a node it fixes the node's position (The pinned nodes can still be dragged directly but they will remain in their position when the other nodes are moved). 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.

d. [5 pts] **Adding Tooltips**: Using the d3-tip library, add a tooltip for each node. A "mouseover" event on the node should display a tooltip containing the list of neighbors of the node (both in-bound and out-bound neighbors).

The final graph could look like this:
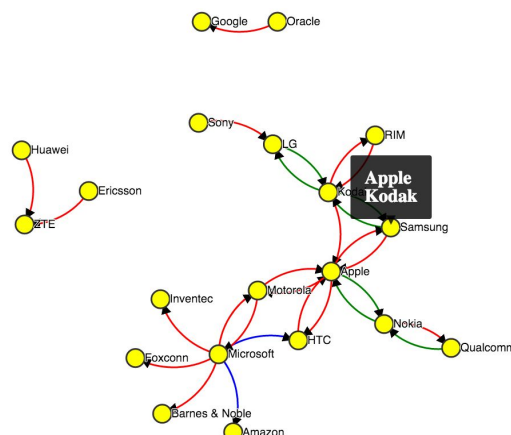
Figure 3

Q3 Deliverables:
- **The directory structure should be as follows:**
    Q3/
        graph.(html / js / css)

- **graph.(html /js /css)** - the html file based on the initial code that contains the changes made in (a-d) above and js/css files created (if any).

Part 2: Visualizing Statistics of Refugees in Europe [65 pts]

4. **[65 pts]** After the training in Part 1, assume that you are analyzing data for United Nations High Commissioner for Refugees (UNHCR) and need to perform the following tasks to aid in the UNHCR's understanding of the persons of concern, considering the ongoing European migrant crisis.

a. [5 pts] Convert the dataset[3] provided in the file *unhcr_persons_of_concern.csv* (in the folder Q4) to a json file (poc.json). This dataset represents a subset of the information available from UNHCR about Persons of Concern during the timeframe 2005-2015 residing in select European countries. The conversion to json file 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.

b. [5 pts] **Table.** Create a table to display details of the refugees 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 tabular visualization needs to convey the data clearly and effectively to the reader (displaying all rows that match the filter may not be an effective visualization). No user interaction is required. Describe your reason for choosing the techniques you use in **explanation.txt** in not more than 50 words**.**

c. [50 pts] **Visualizations using D3.** Visualize country of origin, asylum and number of refugees over the years.
   i.   [20 pts] Use the poc.json file generated in the previous part to create a *Sankey Chart.* The chart should visualize the flow of refugees from one country to another over the years, as shown in the example below. You may refer to this example to create the chart (sankey.js is provided in the lib folder). You can keep the vertical position of the blocks static.
   ii. [10 pts] The countries should be sorted in alphabetical order.
   iii. [15 pts] Create a slider to navigate over the years. The chart should update based on the position of the slider. You should either use the d3.slider library or the noUiSlider library for this. This subpart is optional for undergraduate students, who may only plot the chart for the year 2012.
   iv. [5 pts] Use the d3-tip library to add tooltips as shown.

---

[3] Source: http://popstats.unhcr.org/en/persons_of_concern

HINT*:* You may find the following functions useful - d3.nest(), array.filter(), array.map(), array.sort()

NOTE: ECMAScript 6 was finalized in June 2015 that introduces significant changes to the new javascript version. **5 pts bonus** will be awarded for a **correct solution** that uses at least one of the new features introduced in ECMAScript 6 (we suggest using arrow functions; don't just use trivial things like *const* or *let*).
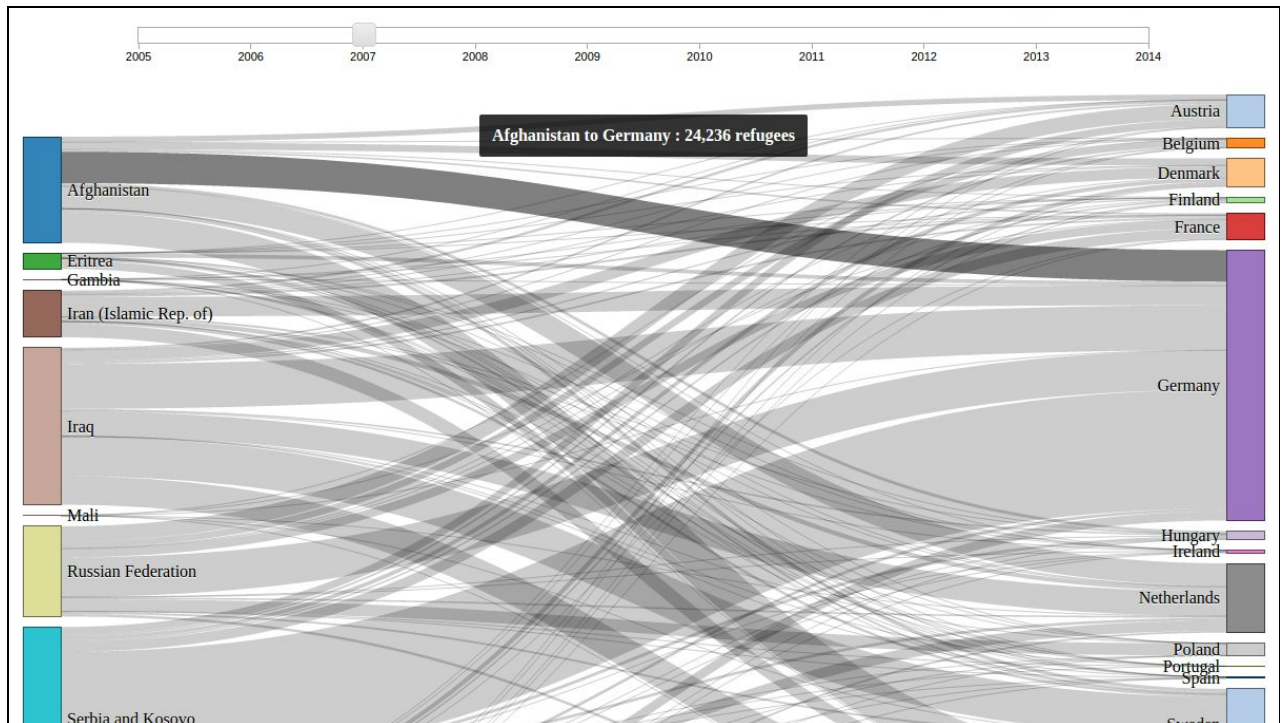
Figure 4

d. [5 pts] **Tableau:** Visualize the demographic attributes (age, sex, country of origin, asylum seeking country) in the file *unhcr_popstats_demographics.csv* (in the folder Q4) for any given year in one chart. Tableau is a popular InfoViz tool and the company has provided us with student licenses. Go to this link 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 available on T-Square. Do not share the key with anyone. Provide a rationale for your design choices in this step in the file **explanation.txt** in not more than 50 words.

Q4 Deliverables:
● **The directory structure should be as follows:**
> **Q4/**
>> poc.json
>> table.yyy
>> viz.(html / js/ css)
>> chart.xxx (from Tableau)

explanation.txt
- **table.yyy** - Any modern image format (e.g., jpg, png, pdf) showing the table created in Q4.b.
- **viz.html / js / css** - The html, javascript, css to render the visualization in Q4.c.
- **chart.xxx** - The figure for the chart generated from Tableau in Q4.d. You can use formats like png and pdf, but be sure to make it a high-quality and clear image).
- **explanation.txt** - Write the explanation for parts Q4.b and Q4.d in this file. Also explain how you used ECMAScript 6 in Q4.c. Keep it succinct. If you're an undergraduate student, mention it here, and say whether you've attempted Q.4.c.iii.

Part 3: Visualizing College Scorecard data [15 pts]

5. **[15 pts] Visualization.** Using D3, construct a visualization using the college scorecard dataset that may help students make college decisions. The dataset (**located in the Q5 folder**) contains many statistics about affordability and value.

   Create one large visualization or multiple small ones using the entire dataset or a subset of it. (The visualization does **not** need to support any interactions.)

   You may also refer to the information on this website to augment your data or for more ideas:
   http://catalog.data.gov/dataset/college-scorecard

   - The main goal of this task is for you to come up with a creative, practical and informative idea that could aim decision making, and implement it in D3 in the best way you can. Points will be awarded for functionality, and also for interesting ideas.
   - Summarize your main ideas behind the visualization in **explanation.txt** in no more than 50 words**.**

Q5 Deliverables:
- **The directory structure should be as follows:**
    Q5/
      q5.(html /jss /css)
      *.json
      explanation.txt

- **q5.(html /js /css**)- The html, javascript, css files to render the visualization made in Q5.
- ***.json** - Include all the json file(s) used as data sources.
- **explanation.txt** - As described in the question.

Important Instructions on Folder structure

We will be executing the following commands to validate the submission is all right. If some files are missing in our query of the zip folder, marks will be deducted. For example, if Q1/explanation.txt is missing, points could be deducted.

The directory structure should be as follows:

```
HW2-LastName-FirstName/
     |--- lib/
            |---- d3/
                     |----   d3.v3.min.js
                     |----   d3-tip.js
                     |----   sankey.js
                     |----   d3.slider.css
                     |----   d3.slider.js
                     |----   nouislider.css
                     |----   nouislider.js
                     |----   nouislider.pips.css
                     |----   nouislider.tooltips.css
     |--- Q1/
            |----   scatterplot.(html / js / css)
            |----   scatterplots.pdf
            |----   ForestFires.csv
     |--- Q2/
            |----   Q2.yyy
            |----   nepal_new.csv
            |----   stacked.pdf
            |----   stacked_barchart.(html / css / js)
            |----   explanation.txt
     |--- Q3/
            |----   graph.(html / js / css)
     |--- Q4/
            |----    poc.json
            |----    table.yyy
            |----    viz.(html / js / css)
            |----    chart.xxx (from Tableau)
            |----    explanation.txt
     |--- Q5/
            |----    *.json
            |----    q5.(html / js / css)
            |----    explanation.txt
```