CSE 6242/CX 4242: Data and Visual Analytics | Georgia Tech | Spring 2017

Homework 2 : D3 Graphs and Visualization

Due: Wednesday, March 1, 2017, 11:55 PM EST

Prepared by Meghna Natraj, Bhanu Verma, Fred Hohman, Kiran Sudhir,
Varun Bezzam, Chirag Tailor, Polo Chau

Submission Instructions and Important Notes:

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 other format will be accepted)
- 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 to 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 about how to organize your files in a single zipped file. 5 points will be deducted for not following this strictly.
- We will use auto-grading scripts to grade some of your deliverables (there are hundreds of students), so it is extremely important that you strictly follow our requirements. Marks may be deducted if our grading scripts cannot execute on your deliverables.
- 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, please 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 a 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. Students in the undergraduate section (CX4242) can choose to complete any 100 points worth of work to receive the full 15% of the final course grade. For example, if a CX4242 student scores 120 pts, that student will receive (120 / 100) * 15 = 18 pts towards the final course grade. To receive the full 15% score, students in the CSE6242 sections will need to complete all 120 points.

Important Prerequisites

Download the HW2 Skeleton that contains files you will use in this homework.

We highly recommend that you use the latest Firefox browser to complete this homework. We will grade your work using **Firefox 50.1.0 (or newer)**.

For this homework, you will work with version 3 of D3, provided to you in the **lib** folder. You must NOT use any other d3 libraries (d3*.js) other than the ones provided.

You may need to setup an HTTP server to run your D3 visualizations (depending on which web browser you are using, as discussed in the D3 lecture). The easiest way is to use SimpleHTTPServer in Python (for Python version 2.x). **You should run your local HTTP server in the root (hw2-skeleton) folder.**

All d3*.js files in the **lib** folder must be referenced using relative paths, e.g., **"../lib/<filename>**" in your html files (e.g., those in folders Q2, Q3, etc.). For example, suppose the file "Q2/graph.html" uses d3, its header should contain:

```
<script  type="text/javascript" src="../lib/d3.v3.min.js"></script>
```

It is incorrect to use an absolute path such as:

```
<script type="text/javascript" src="http://d3js.org/d3.v3.min.js"></script>
```

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

Q1 [10 pts] Designing a good table. Visualizing data with Tableau.

Imagine you are a data scientist working with United Nations High Commissioner for Refugees (UNHCR). Perform the following tasks to aid UNHCR's understanding of persons of concern.

a. **[5 pts] Good table design.** Create a table to display the details of the refugees (Total Population) in the year 2005 from the data provided in *unhcr_persons_of_concern.csv*. You can use any tool (e.g., Excel, HTML) to create the table. Keep suggestions from class in mind when designing your table (see lectures slides, specifically slide 43, for what to, but you are not limited to the techniques described). Describe your reason for choosing the techniques you use in **explanation.txt** in no more than 50 words**.**

b. **[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 Q1) for any given year in one chart. Tableau is a popular InfoViz tool and the company has provided us with student licenses. Go to tableau 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 available in T-Square Resources as "Tableau Desktop Key". This key is for your use in this course only. Do not share the key with anyone. Provide a rationale for your design choices in this step in the file **explanation.txt** in no more than 50 words.

Q1 Deliverables:
The directory structure should be as follows:

Q1/
table.(png / pdf)
chart.(png / pdf)
explanation.txt
unhcr_persons_of_concern.csv
unhcr_popstats_demographics.csv

- **table.(png / pdf)** - An image/screenshot of the table in Q1.a (png or pdf format **only**).
- **chart.(png / pdf)** - An image of the chart in Q1.b (png or pdf format **only**, Tableau workbooks will not be graded!). The image should be clear and of high-quality.
- **explanation.txt** - Your explanations for parts Q1.a and Q1.b in this file.
- **unhcr_persons_of_concern.csv** and **unhcr_popstats_demographics.csv** - the datasets

Q2 [15 pts] Force-directed graph layout

You will experiment with many aspects of D3 for graph visualization. To help you get started, we have provided the graph.html file (in the folder Q2). **Note:** You are welcome to split graph.html into graph.html, graph.css, and graph.js.

a. **[3 pts] Adding node labels**: Modify graph.html to show a node label (the node *name*, i.e., the *source*) to the right of each node. If a node is dragged, its label must also move with the node.

b. **[3 pts] Coloring links**: Color the links based on the "value" field in the links array. Assign the following colors:

If the value of the edge is < 1.0 : assign Blue color to the link.
If the value of the edge is >= 1.0 and <= 2.0 : assign Green color to the link.
If the value of the edge is > 2.0 : assign Red color to the link.

c. **[3 pts] Scaling node sizes:**

1. Scale the radius of each node in the graph based on the degree of the node.
2. In **explanation.txt**, using no more than 40 words, discuss your scaling method you have used and explain why you think it is a good choice. There are many possible ways to scale, e.g., scale the radii linearly, by the square root of the degree, etc.

d. **[6 pts] Pinning nodes** (fixing node positions):

1. Modify the html so that when you double click a node, it pins the node's position such that it will not be modified by the graph layout algorithm (note: pinned nodes can still be dragged around by the user but they will remain at their positions otherwise). Node pinning is an effective interaction technique to help users spatially organize nodes during graph exploration.
2. Mark pinned nodes to visually distinguish them from unpinned nodes, e.g., pinned nodes are shown in a different color, border thickness or visually annotated with an "asterisk" (*), etc.
3. Double clicking a pinned node should unpin (unfreeze) its position and unmark it.

Q2 Deliverables:
The directory structure should be as follows:

Q2/
graph.html
explanation.txt
graph.js, graph.css (if not included in graph.html)

Q3 [15 pts] Scatter plots

Tutorial: [Making a scatter plot]

Use the dataset[1] provided in the file *data.tsv* (in the folder Q3) to create two scatter plots.

a. **[8 pts]** Create a scatter plot with the distribution feature on the Y-axis and the body mass feature on the X-axis. Use different symbols and colors to indicate the different species:

- Red circles for Lagomorpha
- Blue squares for Didelphimorphia
- Green triangles for Dasyuromorphia

b. **[2 pts]** Add a legend to the scatter plot to show how species names map to the colored symbols.

c. **[3 pts]** Create another scatter plot using the same data, symbols, and legend as above, but use the log scale instead for both axes.
Note: The two scatter plots should be placed on a single html page, one after the other, as shown in Figure 1; your plots' visual design can be different from what is shown.

d. **[2 pts]** Explain in no more than 50 words, in **explanation.txt**, when we may want to use log scales in charts (e.g., in scatter plots).

[1] Derived from source: http://www.esapubs.org/archive/ecol/E084/094/#data

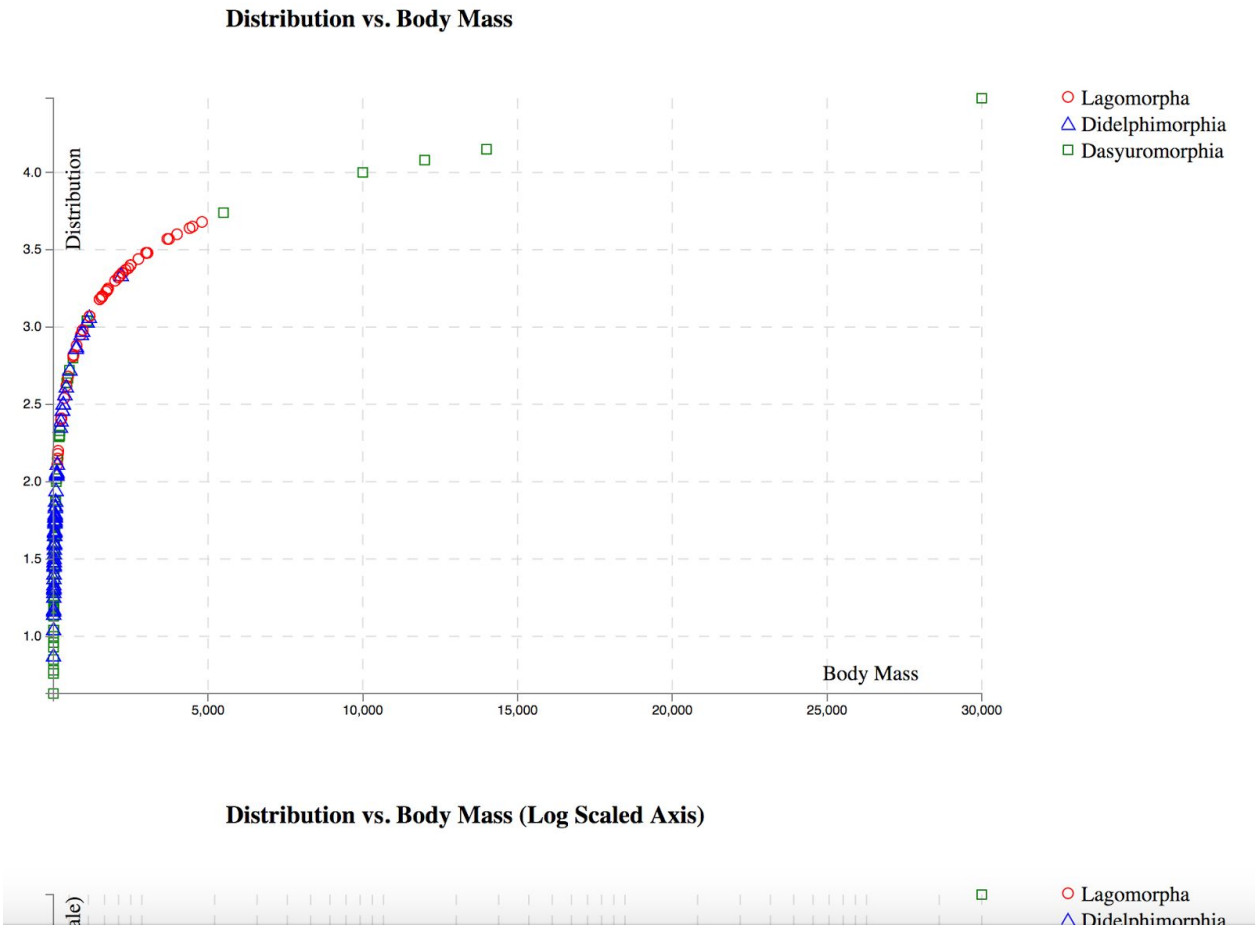


Figure 1. Example of how the two scatter plots should be arranged on a single HTML page. First show the plot from part a, then the one from part c.

Q3 Deliverables:

The directory structure should be organized as follows:

Q3/

scatterplot.(html / js / css)
explanation.txt
scatterplot.(pdf / png)
data.tsv

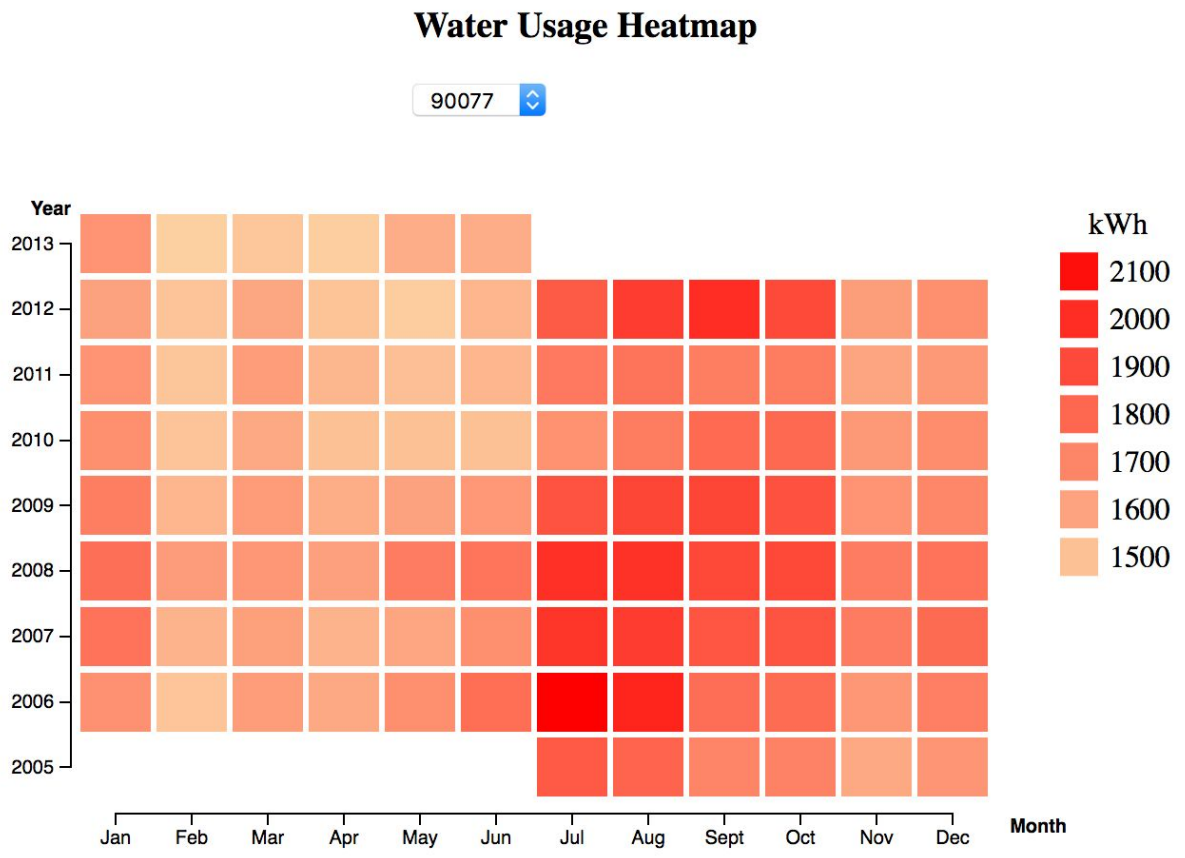
- **scatterplot.(html / js / css)** - the html/js/css files created.
- **explanation.txt** - the text file with your answer for 3d.
- **scatterplot.(pdf / png)** - a screenshot (png or pdf format) showing the two scatter plots created above.
- **data.tsv** - the dataset

Q4 [15 pts] Heatmap and Select Box

Example: 2D Histogram, Select Options

Use the dataset[2] provided in *heatmap.csv* (in the folder Q4) that describes power usage (kWh) across multiple zip codes in Los Angeles and visualize it using D3 heatmaps.

a. **[6 pts]** Create a heatmap of the power usage over time for zip code 90077. Place the month on the heatmap's horizontal axis and the year on its vertical axis. Power readings will be represented by colors in the heatmap.
b. **[3 pt]** Add axes and legends to both charts similar to the 2D Histogram example. Instead of placing the month number on the horizontal axis, place the name of the month ("Jan", "Feb", "Mar", etc.). Use d3.axis()'s member function .tickFormat() to provide a custom format to each tick value on the axis.
c. **[6 pt]** Now create a drop down select box with D3 that is populated with the unique zip codes in ascending order. When the user selects a different zip code in this select box, the heatmap for power usage should be updated with the values corresponding to the selected zip code. The default zip code when the page loads should be 90077.



[2] Source: https://catalog.data.gov/dataset/water-and-electric-usage-from-2005-2013-83298

Q4 Deliverables:

The directory structure should look like (remember to include the d3 library):

Q4/

heatmap.(html / js /css)

heatmap.(png / pdf)

heatmap.csv

- **heatmap.(html / js/ css)** - the html / js / css files created.
- **heatmap.(png / pdf)** - a screenshot (png or pdf format) of the plots created in Q4.b
- **heatmap.csv** - the dataset

Q5 [25 pts] Sankey Chart

Example: [Sankey diagram from formatted JSON]

Formula One racing is a championship sport in which race drivers represent teams to compete for points over several races (also called Grand Prix) in a season. The team with the most points at the end of a season wins the prestigious Formula One World Constructors' Championship award. You will visualize the flow of points for the races held in 2016[3]. The drivers win points according to their final standing in each race, which finally get added to their respective team's total.

Note: The implementation of certain parts in this question may be quite challenging.

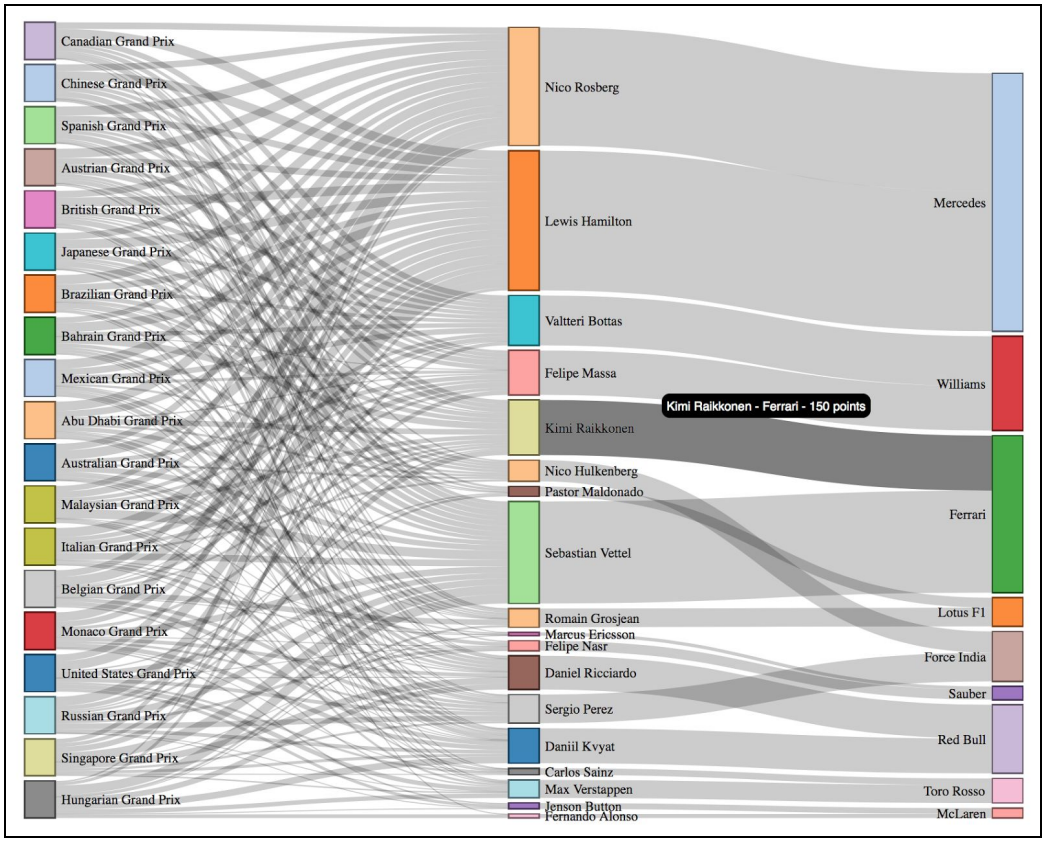


Figure 2. Example Sankey Chart visualizing the flow of points for the 2015 season

[3] Source: http://ergast.com/mrd/

a. **[15 pts]** Create a *Sankey Chart* using the provided datasets (*races.csv* and *teams.csv*) in the Q5 folder. The chart should visualize the flow of points in the order:

```
race → driver → team
```

You must use the sankey.js provided in the lib folder. You can keep the blocks' vertical positions static. Your chart should look similar to the example Sankey Chart for the 2015 season as shown in the above image.
Note: For this part, you will have to read in the csv files and combine the data into a format that can be passed to the sankey library. To accomplish this, you may find the following javascript functions useful: d3.nest(), array.filter(), array.map()

b. **[6 pts]** Use the d3-tip library to add tooltips as shown in the above image. You are welcome to make your own visual style choices using css properties.
Note: You must create the tooltip by only using **d3.tip.v0.6.3.js** present in the **lib** folder.

c. **[4 pts]** From the visualization you have created, determine the following:
   1. [1 pt] Which driver won the Grand Prix 2016?
   2. [1 pt] Which team won the Grand Prix 2016?
   3. [1 pt] Which driver won the Spanish Grand Prix?
   4. [1 pt] Which team has the maximum number of players?

Put your answers in **observations.txt**. Modify the template provided to you (in Q5 folder) by replacing team_name/driver_name with your answer

Sample **observations.txt**
1.driver_name 2.team_name 3.driver_name 4.team_name

Q5 Deliverables:
The directory structure should be as follows:

Q5/
races.csv
teams.csv
viz.(html/js/css)
observations.txt

- **races.csv** and **teams.csv** - the data sets (unmodified)
- **viz.(html/js/css)** - The html, javascript, css to render the visualization in Q5.a and b.
- **observations.txt** - Your answer for Q5.c.

Q6 [20 pts] Interactive visualization

Mr. Fluke runs a small company named FooBar. His company manufactures eight products around the year. He wants you to create an interactive visualization report using D3 so that he can see the total revenue generated per product type and the revenue breakdown across product types for the four quarters in 2015. Use the dataset provided in the Q6 folder. Integrate the dataset provided in dataset.txt directly in an array variable in the script.

Example: `<script> var data=[<paste data file conte nt here>];</script>`

a. **[5 pts]** Create a **horizontal bar chart** with its vertical axis denoting the product names and its horizontal axis denoting the total revenue. Each bar should have the total revenue amount in dollars labelled inside it. Refer to the example shown in Figure 6a.

b. **[10 pts]** On hovering over a bar, another smaller bar chart representing the revenue of each quarter for that product should be displayed in the top right corner. For example, product B generates revenue of $959, $1653, $1999 and $697 for the four quarters. On hovering over the bar representing product B, a bar chart depicting these 4 values is displayed. See Figure 6b for an example.

c**. [3 pts]** On mouse out, the bar chart of the quarters should no longer be visible.

d. **[2 pts]** On hovering over any horizontal bar representing a product, the color of the bar should change. You can use any color that is visually distinct from the regular bars. On mouseout, the color should be reset.

Note: The vertical axis of the chart should use product names as labels for the products, and quarter numbers for the quarters.

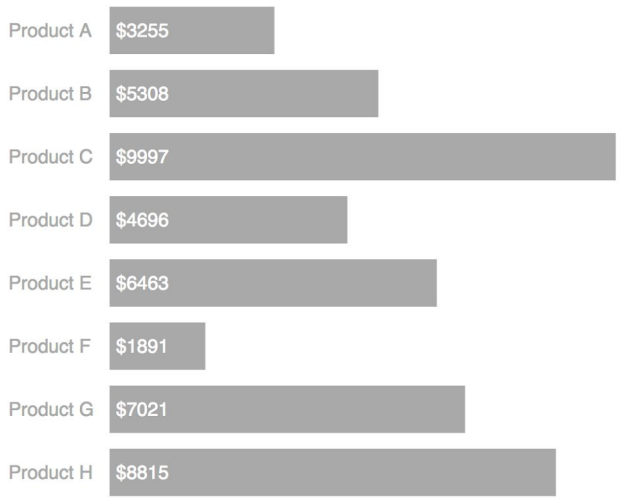


Figure 6a. Bars representing total revenue of each product.

Figure 6b. On hovering over the bar for product B, a smaller bar chart representing the revenue generated by product B in the 4 quarters is displayed at the top right corner.

Q6 Deliverables:
The directory structure should be as follows:

Q6/
interactive.(html/js/css)

interactive.(html/js/css) - The html, javascript, css to render the visualization in Q6 (dataset.txt is *not* required to be included in the final directory structure as the data provided in dataset.txt should have already been integrated into the "data" variable).

Q7 [20 pts] Choropleth Map of College Data

Example: Unemployment rates

Use the provided datasets in *sat_scores.csv*, *us.json* and *median_earnings.json* (in the folder Q7) and visualize them as a choropleth map.

- Each record in *sat_scores.csv* represents a college and is of the form `<id,name,sat_avg>`, where `id` corresponds to the state the college is in, `name` corresponds to the name of the college and `sat_avg` corresponds to the average SAT score of admitted students.
- The *median_earnings.json* file contains a list of JSON objects, each having two fields: an `id` field corresponding to a state in the United States, and a `median_earnings` field corresponding to the median earnings of students in that state after 10 years.
- The *us.json* file is a TopoJSON *topology* containing three geometry collections: *counties*, *states*, and *nation*.

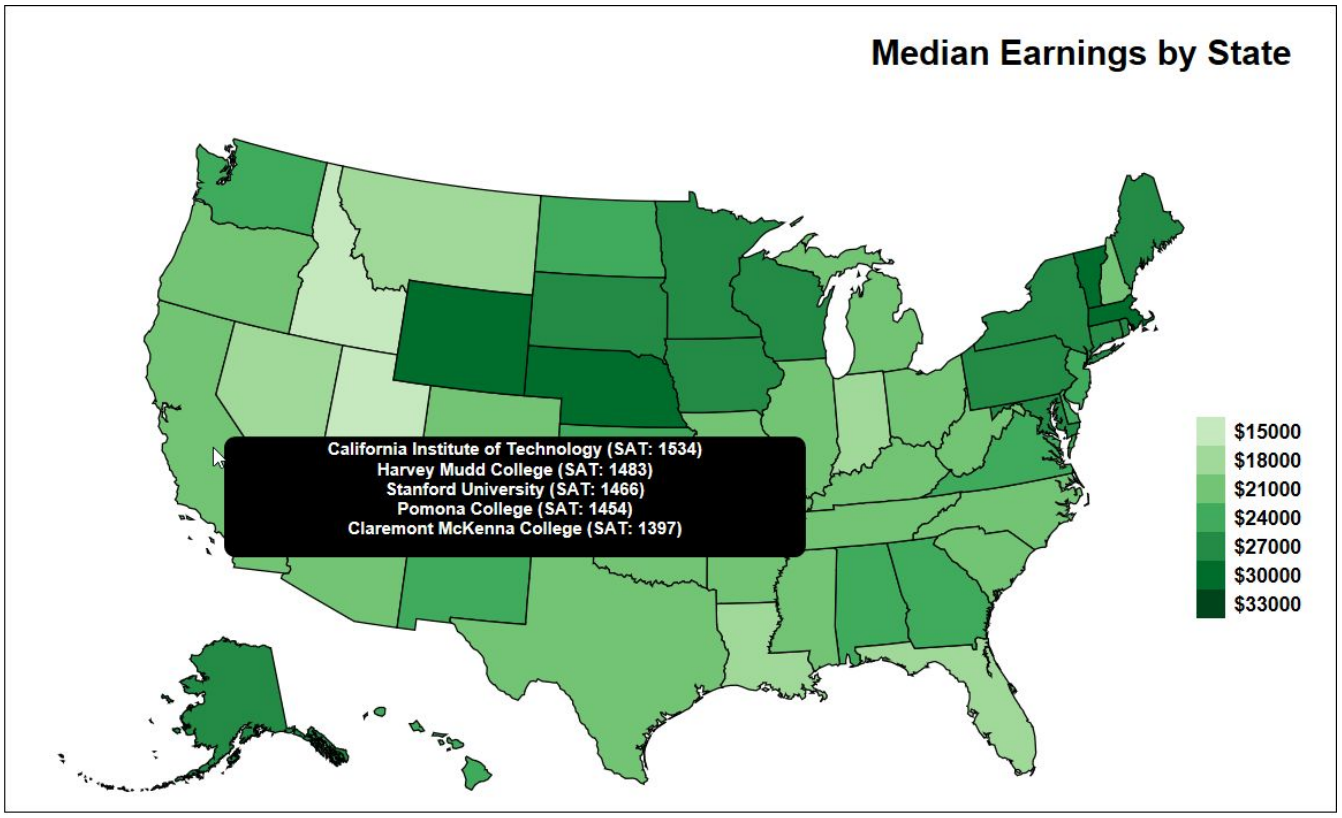


Figure 4. Reference example for Choropleth Maps

a. **[15 pts]** Create a choropleth map using the provided datasets. The color of each state should correspond to the median earnings in that state, i.e., darker colors correspond to higher median earnings in that state and lighter colors correspond to lower median earnings in that state. Add a legend showing how colors map to median earnings. Use d3-queue (in the lib folder) to easily load data from multiple files into a function[4]. Use topojson (present in lib) to draw the choropleth map.

b. **[5 pts]** Add a tooltip using the d3.tip library (in the **lib** folder) that, on hovering over a state, shows the top 5 colleges in that state with the highest SAT scores, along with those scores.
Note: You must create the tooltip by only using **d3.tip.v0.6.3.js** present in the **lib** folder.

Q7 Deliverables:
The directory structure should be organized as follows:

Q7/
q7.(html/js/css)
sat_scores.csv
median_earnings.json
us.json

- **q7.(html /js /css**)- The html/js/css file to render the visualization.
- **sat_scores.csv and median_earnings.json** - The datasets used.
- **us.json** - Dataset needed to draw the map.

[4] d3-queue evaluates a number of asynchronous tasks concurrently -- in this question, each task would be loading one data file. When all tasks have finished, d3-queue passes the results to a user-defined callback function.

Important Instructions on Folder structure

The directory structure must be as follows. The files that should be included in each question's folder (e.g., Q1 for question 1) have been clearly specified at the end of each question's problem description above.

```
HW2-LastName-FirstName/
        |--- lib/
              |----    d3.v3.min.js
              |----    d3.tip.v0.6.3.js
              |----    sankey.js
        |--- Q1/
              |----    ...
        |--- Q2/
              |----    ...
        |--- Q3/
              |----    ...
        |--- Q4/
              |----    ...
        |--- Q5/
              |----    ...
        |--- Q6/
              |----    ...
        |--- Q7/
              |----    ...
```