

# CSE 6242 / CX 4242: Data and Visual Analytics

## Georgia Tech, Fall 2019

### Homework 2 : Tableau, D3 Graphs and Visualization

Prepared by our 26+ wonderful TAs of [CSE6242A,Q,OAN,O01,O3/CX4242A](#) for our 1200+ students

#### 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**.

- ❑ **Always check to make sure you are using the most up-to-date assignment** (version number at bottom right of this document).

- ❑ Submit a single zipped file, called "HW2-GT username.zip", containing all the deliverables including source code/scripints, data files, and readme. Example: "HW2-jdoe3.zip" if GT account username is "jdoe3". **Only .zip is allowed** (no other format will be accepted). **Your GT username is the one with letters and numbers.**

- ❑ You may discuss high-level ideas with other students at the "whiteboard" level (e.g., how cross validation works, use hashmap instead of array) and review any relevant materials online. **However, each student must write up and submit his or her own answers.**

- ❑ All incidents of suspected dishonesty, plagiarism, or violations of the [Georgia Tech Honor Code](#) will be subject to the institute's Academic Integrity procedures (e.g., reported to and directly handled by the [Office of Student Integrity \(OSI\)](#)). **Consequences can be severe, e.g., academic probation or dismissal, grade penalties, a 0 grade for assignments concerned, and prohibition from withdrawing from the class.**

- ❑ At the end of this assignment, we have specified a folder structure **you must use** to organize your files in a single zipped file. **5 points will be deducted for not following this strictly.**

- ❑ 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).

- ❑ We may use auto-grading scripts to grade some of your deliverables, so it is extremely important that you strictly follow our requirements.

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

- ❑ Every homework assignment deliverable and every project deliverable comes with a 48-hour "grace period". **Any deliverable submitted after the grace period will get zero credit. We recommend that you plan to finish by the beginning of the grace period in order to leave yourself time for any unexpected issues which might arise.**

❑ **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. You may re-submit your work before the grace period expires. [Canvas automatically appends a “version number” to files that you re-submit](#). You do not need to worry about these version numbers, and there is no need to delete old submissions. **We will only grade the most recent submission.**

## Grading

The maximum possible score for this homework is 100 points.

Students in the CX4242 undergraduate section can choose to complete any 85 points worth of work to receive the full 15% of the final course grade. For example, if a CX4242 student scores 100 pts, that student will receive  $(100 / 85) * 15 = 17.65$  points towards the final course grade. To receive the full 15% score, students in the CSE6242 sections will need to complete all 100 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 68.0 (or newer)**.

For this homework, you will work with version 5 of D3, provided to you in the **lib** folder. You must NOT use any 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 (OMS students: the video “Week 5 - Data Visualization for the Web (D3) - Prerequisites: Javascript and SVG”. Campus students: see [lecture PDF](#)). The easiest way is to use [http.server](#) for Python 3.x, or [SimpleHTTPServer](#) for Python 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. For example, suppose the file “Q2/graph.html” uses d3, its header should contain:

```
<script type="text/javascript" src="../lib/d3.v5.min.js"></script>
```

**It is incorrect to use an absolute path such as:**

```
<script type="text/javascript" src="http://d3js.org/d3.v5.min.js"></script>
```

For questions that require reading from a dataset, you may be required to submit the dataset in the deliverables too. For such questions, in your html/js code, use a **relative path** to read in the dataset file. For example, suppose a question Q reads data from `heatmap.csv`, the path should simply be “heatmap.csv” and **NOT** an absolute path such as “C:/Users/polo/hw2-skeleton/Q/heatmap.csv”.

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 points] Designing a good table. Visualizing data with Tableau.

Imagine you are a data scientist working with data that documents population distribution according to ethnic group, age and gender across years.

a. **[5 points] Good table design.** You want to help your organization analyze the data in the years 2017 and 2018. Create a well-designed a table to visualize data contained in *age-distribution.csv*. You can use any tool (e.g., Excel, HTML) to create the table.

For each year, and for each ethnic group (treat “Other Ethnic Groups” as an ethnic group), your table should clearly communicate,:

- The total number of males (across all ages)
- The total number of females (across all ages)
- The total population (across all ages)
- The percentage of people that are 65 years and over, rounded to 2 decimal places (you will need to calculate this percentage)
- **Save the table as table.png.**

You may decide on the most meaningful column names to use, the number of columns, and the column order. Keep suggestions from lecture in mind when designing your table. You are not required to use only the techniques described in lecture. For OMS students, the online lecture video pertaining to this topic is *Week 4 - Fixing Common Visualization Issues - Fixing Bar Charts, Line Charts*). For campus student, please review slide 43 and onwards of the [lecture slides](#).

b. **[5 points] Tableau.** You want to help your organization better understand the yearly trend in population growth (in a city) and contribution of each ethnicity towards that growth. Visualize the data *population.csv*<sup>[1]</sup> as a *stacked bar chart*. Your chart should display years (1960 to 1970, inclusively) on the vertical axis and the total population on the horizontal axis. (Optional reading: the effectiveness of stacked bar charts is often debated --- sometimes, [they can be confusing, difficult to understand, and may make data series comparison challenging](#).)

Our main goal here is for you to try out Tableau, a popular information visualization tool. Thus, we keep this part more open-ended, so you can practice making design decisions. **We will accept most designs from you all.** We show one possible design in the figure below, based on the tutorial from [Tableau](#), and you are not limited to the techniques presented there.

Please follow the instructions below:+

- Your design should visualize the values of the categories *Total Malays*, *Total Indians*, *Total Chinese*, *Other Ethnic Groups (Total)* for each year.
- Your design should utilize a stacked bar chart to show the count for each of the aforementioned columns
- Your design should have clear label axes and a clear chart title. Include a legend for your chart.
- **Save the chart as barchart.png.**

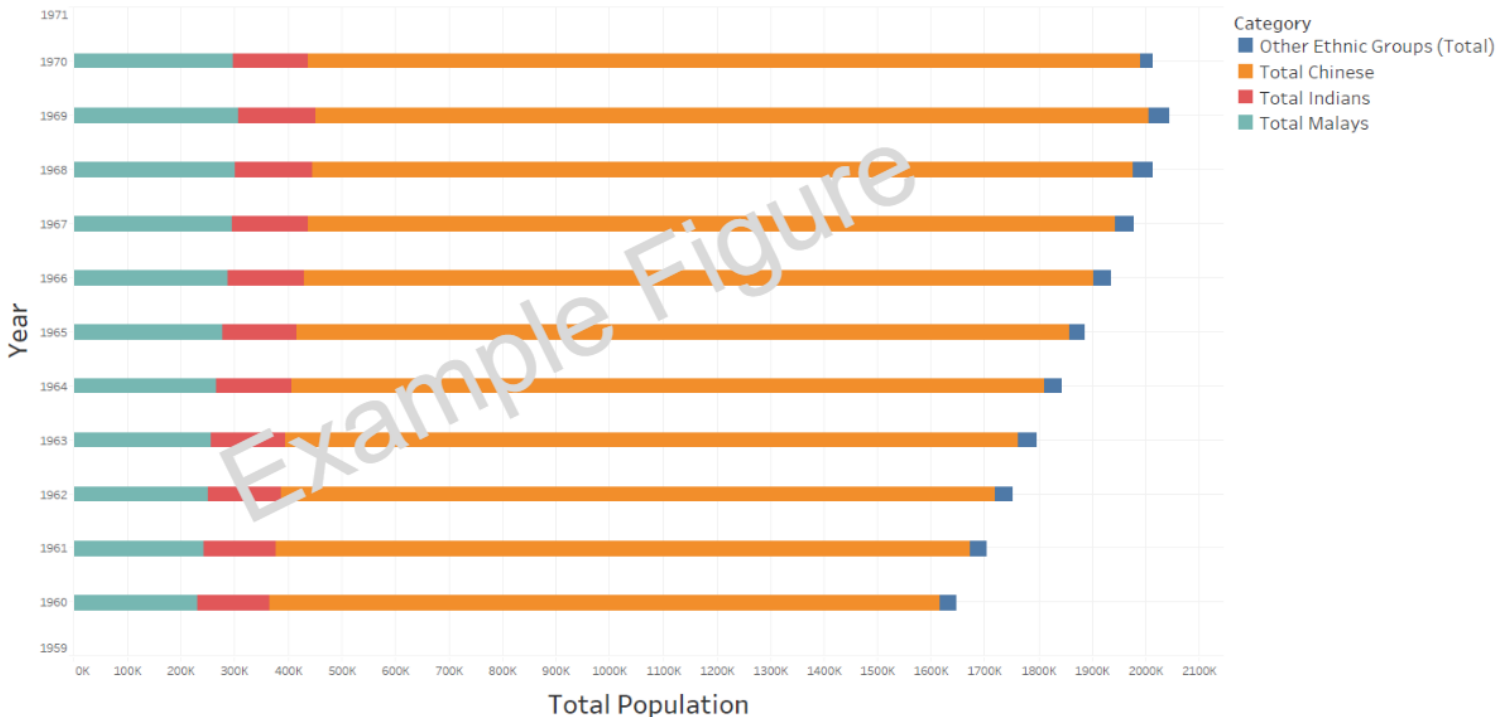
Tableau has provided us with student licenses for *Tableau Desktop*, available for Mac and Windows. Go to [tableau activation](#) and select “Tableau Desktop”. After the installation, you will be asked to provide an activation key, which you can find on the Canvas page for this assignment. This key is for your use in this course only. **Do not share the key with anyone.**

If you do not have access to a Mac or Windows machine, please use the 14-day trial version of *Tableau Online*:

1. Visit <https://www.tableau.com/trial/tableau-online>
2. Enter your information (name, email, GT details, etc)
3. You will then receive an email to access your Tableau Online site
4. Go to your Site and create a workbook

One final option, if neither of the above methods work, is to take advantage of [Tableau for Students](#). Follow the link and select “Get Tableau For Free”. You should be able to receive an activation key which offers you a one-year use of Tableau Desktop at no cost by providing a valid Georgia Tech email. Note that it is unclear whether Tableau intends for these licenses to be renewable, so you may only be eligible to receive one in the event that you have never used a *Tableau for Students* license before.

Yearly trend in population growth by ethnicity



**Figure 1:** Example of a stacked bar chart

**Q1 Deliverables:**

The directory structure should be as follows:

Q1/

table.png

barchart.png

age-distribution.csv

population.csv

- **table.png** - An image/screenshot of the table in Q1.a (png format **only**).
- **barchart.png** - An image of the chart in Q1.b (png format **only**), Tableau workbooks will not be graded!). The image should be clear and of high-quality.
- **age-distribution.csv** and **population.csv** - the datasets.

## Q2 [15 points] 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 Q2 folder).

**Note:** You are welcome to split graph.html into graph.html, graph.css, and graph.js. **Please also make certain that any paths in your code are relative paths. Nonfunctioning code will result in a five point deduction.**

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

b. **[3 points] Styling edges:** Style the edges based on the “value” field in the links array. Assign the following styles:

If the value of the edge is equal to 0, the edge should be **black**, **thin**, and **dashed**.

If the value of the edge is equal to 1, the edge should be **green**, **thick**, and **solid**.

c. **[3 points] Scaling nodes:**

1. **[1.5 points]** Scale the radius of each node in the graph based on the degree of the node (you may try linear or squared scale, but you are not limited to these choices).

**Note:** Regardless of which scale you decide to use, you should avoid extreme node sizes (e.g., nodes that are mere points, barely visible, or of huge sizes. Failure to do so will result in a poor quality visualization.

**Note:** D3 v4 (and above) does not support d.weight (which was the typical approach to obtain node degree in D3 v3). You may need to calculate node degrees yourself. Example relevant approach: <https://stackoverflow.com/questions/43906686/d3-node-radius-depends-on-number-of-links-weight-property>

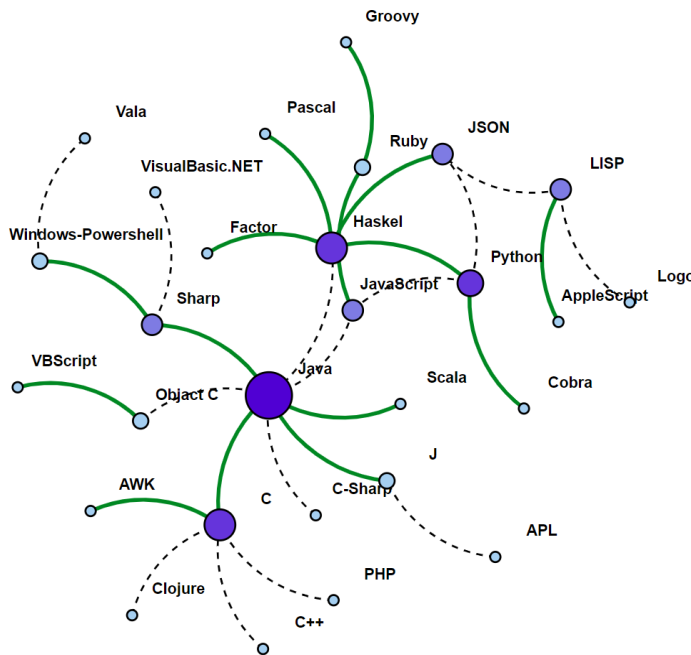
2. **[1.5 points]** The degree of each node should be represented by varying colors. Pick a meaningful color scheme (hint: color gradients). The number of color gradations is up to you, but it must be visually evident that the nodes with higher degree are colored a darker/deeper color and the nodes with lesser degree are colored lighter.

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

1. **[2 points]** Modify the code 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. **[2 points]** 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. **[2 points]** Double clicking a pinned node should unpin (unfreeze) its position and unmark it.



**Figure 2a.** Example Visualization

## Q2 Deliverables:

The directory structure should be as follows:

Q2/

graph.(html / js / css)

- **graph.(html / js / css)** - the html file created, and the js / css files if not included in graph.html

## Q3 [15 points] Line Charts

Use the dataset<sup>[2]</sup> provided in the file *earthquakes.csv* (in the Q3 folder) to create line charts.

Refer to the tutorial for line chart [here](#).

**Note:** You will create four plots in this question, which should be placed one after the other **on a single HTML page**, similar to the example image below (Figure 3). Note that your design need NOT be identical to the example.

**a. [5 points] Creating line chart.** Create a line chart that visualizes the number of earthquakes worldwide from 2000 to 2015 (inclusively), for the four magnitude ranges: [ '5\_5.9', '6\_6.9', '7\_7.9', '8.0+' ]. Use the color scheme provided below for the magnitude ranges. Add a legend at the top right corner of the chart showing the magnitude-color mapping.

- Chart title: Worldwide Earthquake stats 2000-2015
- Horizontal axis label: Year
  - Use `scaleTime` like you did in HW1Q3
- Vertical axis label: Num of Earthquakes
  - Use linear scale for this part a
- Colors scheme: { '5\_5.9': '#FFC300', '6\_6.9': '#FF5733', '7\_7.9': '#C70039', '8.0+': '#900C3F' }

**b. [4 points] Adding symbols and scaling symbol sizes.** Create a line chart for this part (append to the HTML page) whose design is a variant of what you have created in part a. Start with your chart from part a. Then modify the code to visualize each data point in the chart as a solid circle, whose size is proportional to “Estimated Deaths”. Use a good scaling coefficient (your choice) to make the chart legible, visually attractive and meaningful. Keep the legend.

- Chart title: Worldwide Earthquake stats 2000-2015 with symbols

**c. [6 points] Axis scales in D3.** Create two line charts for this part (append to the HTML page) to try out two axis scales in D3. Start with your chart from part b. Then modify the vertical axis scale for each chart: the first chart uses the square root scale for its vertical axis (only), and the second plot uses the log scale for its vertical axis (only). Keep the legend and symbols. In **explanation.txt**, explain when we may want to use such nonlinear scales as square root scale and log scale in charts, in no more than 50 words.

**Note:** the horizontal axes should be kept in linear scale, and only the vertical axes are affected.

**Hint:** You may need to carefully set the scale domain to handle the 0s in data.

### ■ First chart

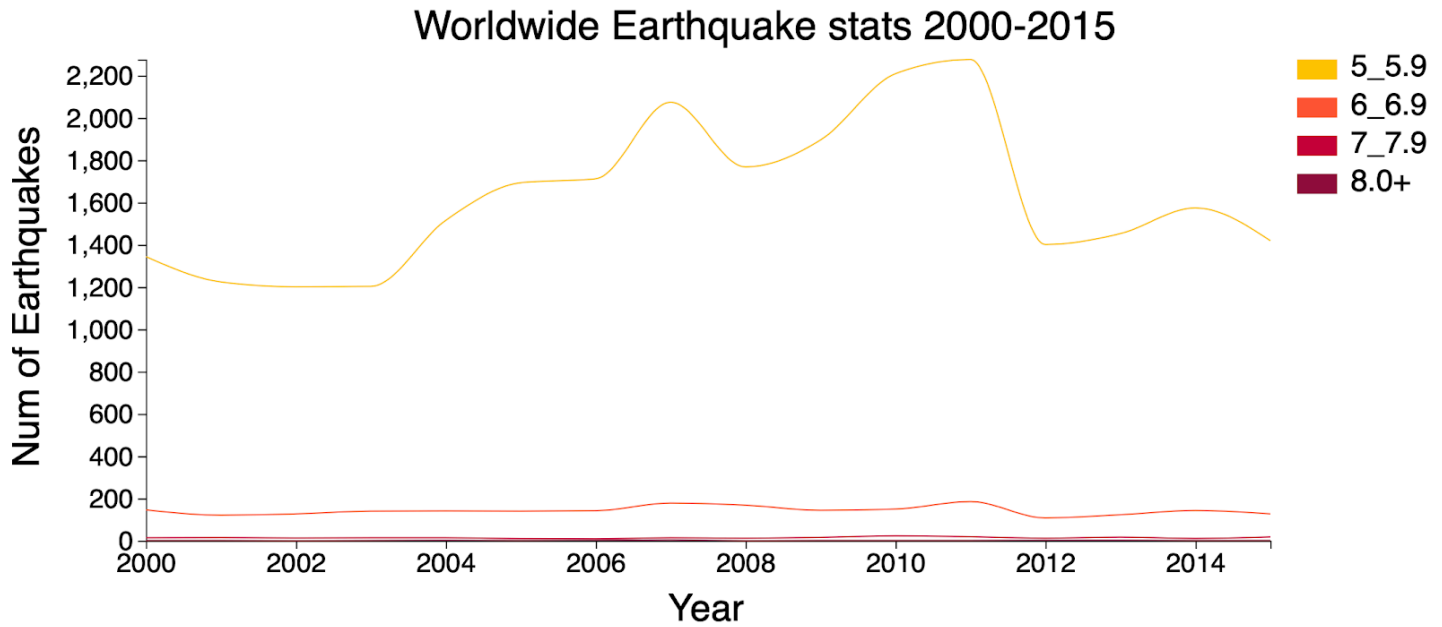
- Chart title: Worldwide Earthquake stats 2000-2015 square root scale
- This chart uses the square root scale for its vertical axis (only)
- Other features should be the same as part b.

### ■ Second chart

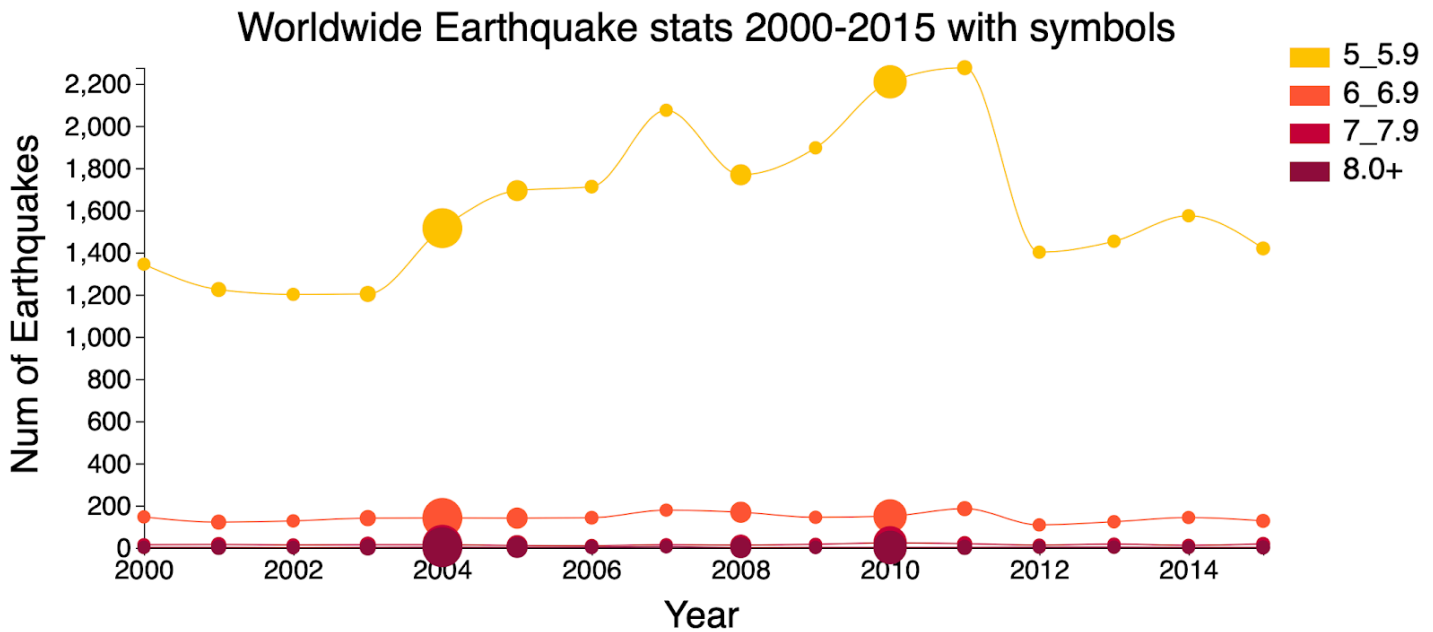
- Chart title: Worldwide Earthquake stats 2000-2015 log scale
- This chart uses the log scale for its vertical axis (only)



- Other features should be the same as part b.



**Figure 3a:** Example line chart



**Figure 3b:** Example line chart with symbols



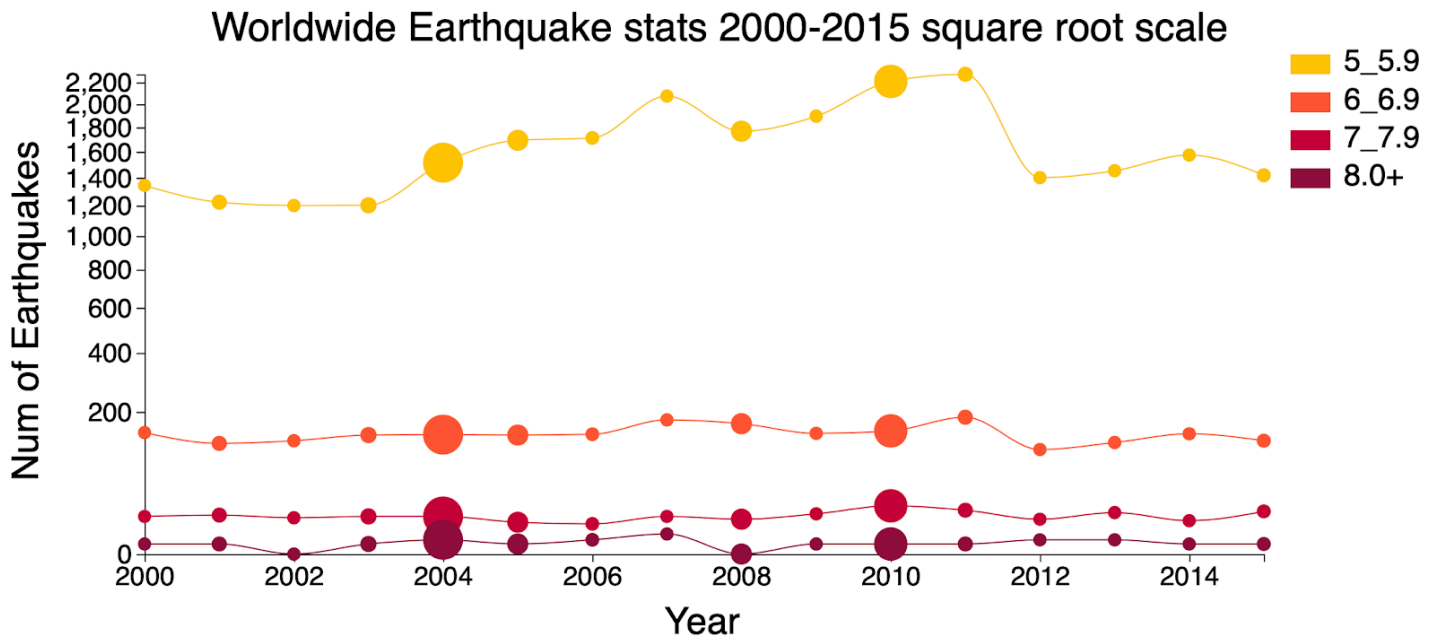


Figure 3c-1: Example line chart using square root scale

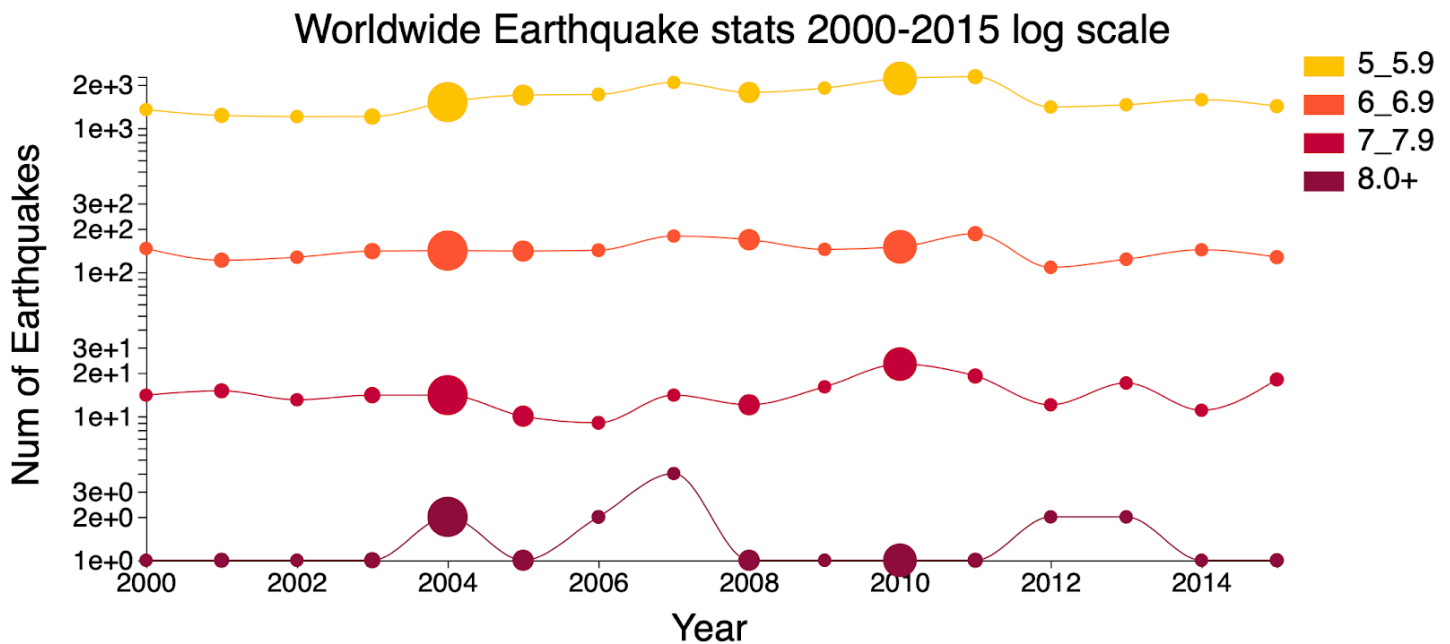


Figure 3c-2: Example line chart using log scale

**Q3 Deliverables:**

The directory structure should be organized as follows:

Q3/

earthquakes.csv  
 linecharts.(html / js / css)  
 linecharts.pdf  
 explanation.txt

- **earthquakes.csv** - the dataset.
  - **linecharts.(html / js / css)** - the html file created, and the js / css files if not included in linecharts.html
  - **linecharts.pdf** - a PDF document showing the screenshots of the four line charts created above (one for Q3.a, one for Q3.b and two for Q3.c). You should print the HTML page as a PDF file, and each PDF page shows one plot (**hint: use CSS page break**). Clearly title the plots as instructed (see examples in Figure 3).
- **explanation.txt** - the text file explaining your observations for Q3.c.

## Q4 [15 points] Heatmap and Select Box

### Example: [2D Histogram](#), [Select Options](#)

Use the dataset provided in *earthquakes.csv* (in the Q4 folder) that describes the earthquake counts for different states from 2010 to 2015 in the US. Visualize the data using D3 heatmaps.

- [3 points]** Create a file named *heatmap.html*. Within this file, create a heatmap of the earthquakes for different states from year 2010 to 2015 (inclusively). Place the state name on the heatmap's horizontal axis and the year on its vertical axis.
- [1 point]** A heatmap's color scheme is a very important design element that has a direct impact on the heatmap's effectiveness. [Colorize](#) the earthquake counts for each state, using a meaningful 9-gradation color gradient of your choice.
- [3 pt]** Add axis labels and a legend to the chart. Place the year (2010, 2011, 2012, etc.) on the vertical axis (i.e. top → bottom: 2010 → 2015). Place the state name ("Alabama", "Arizona", "Arkansas", etc.) on the horizontal axis also in alphabetical order (i.e. left → right: A → Z).
- [6 pt]** Create a drop down [select box](#) with D3 based on the total counts (from 2010 to 2015) of earthquakes of a state. The selections are "0 to 9", "10 to 99", "100 to 499", and "500 or above". When the user selects a different range in this select box, the heatmap and the legend should both be updated with values corresponding to the selected range. Note the differences in the horizontal axes and legends for "0 to 9" and "500 or above" in Figure 4a and Figure 4b below. **While the 9 color gradations in the legend remain the same, the threshold values are different.** The default category when the page loads should be "0 to 9".
- [2 pt]** Implement a mouseover effect. When the mouse cursor is on a heatmap cell, the value of that cell will be displayed between the chart title and the heatmap.

### Note:

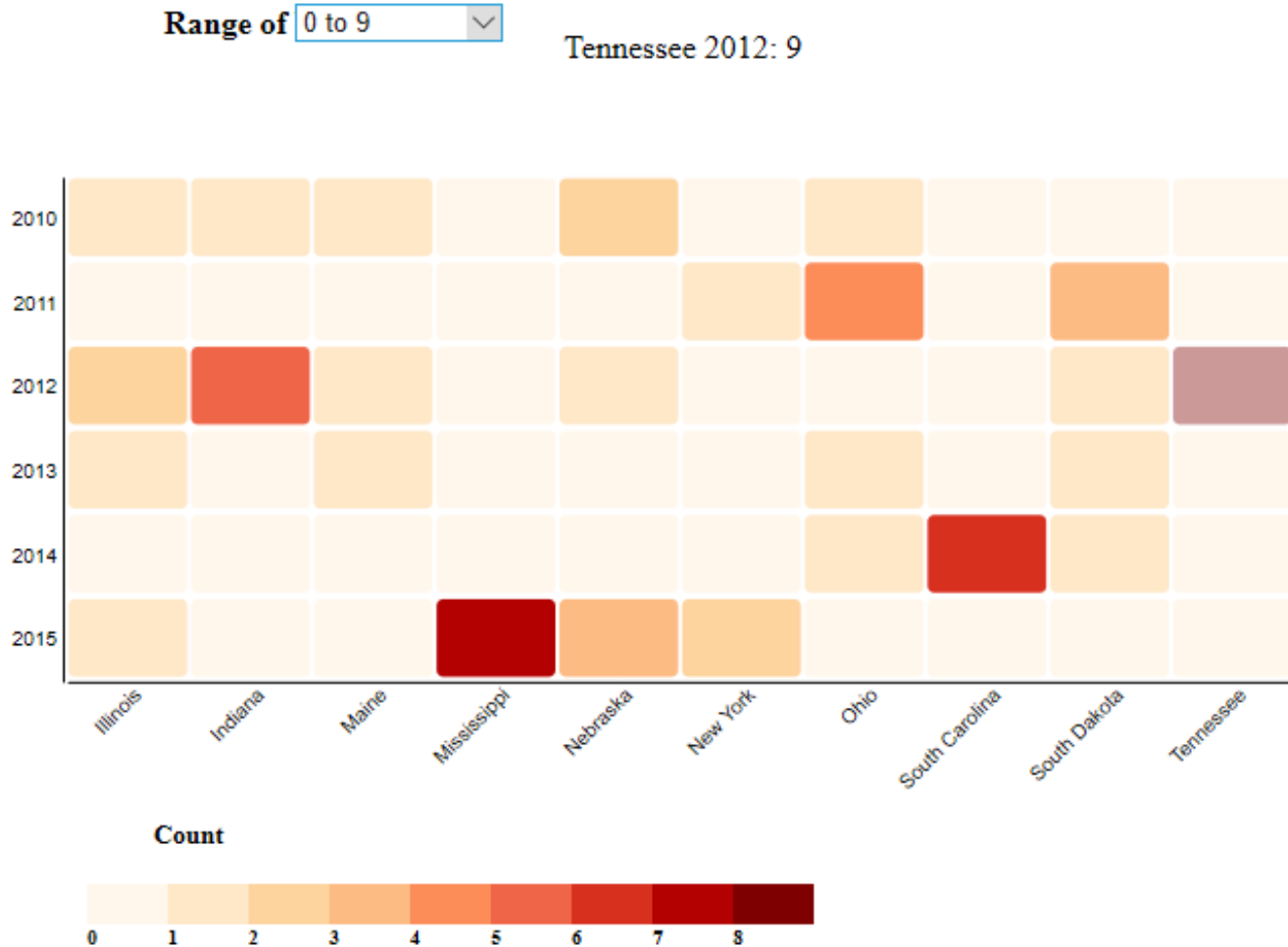
1. The Earthquake Statistics is from [USGS](#) with some modifications.
2. The data provided in earthquake.csv would need to be "reshaped" in such a way that it can produce the expected output. **All data reshaping must only be performed in javascript; you must not modify earthquake.csv.** That is, your code should first read the data from

earthquake.csv file as is, then you may reshape that loaded data using javascript, and then use it to create the heatmap.

3. **The threshold values should not be hardcoded.** They do not necessarily have to match the ones provided in the screenshots below.

The screenshots provided below serve as an example only. You are not expected to produce an exact copy of the screenshots. Please feel free to experiment with fonts, placement, color, etc. as long as the output looks reasonable for a heatmap and meets the functional requirements mentioned above.

## Visualizing Earthquake Counts by State 2010-2015 (M3+)

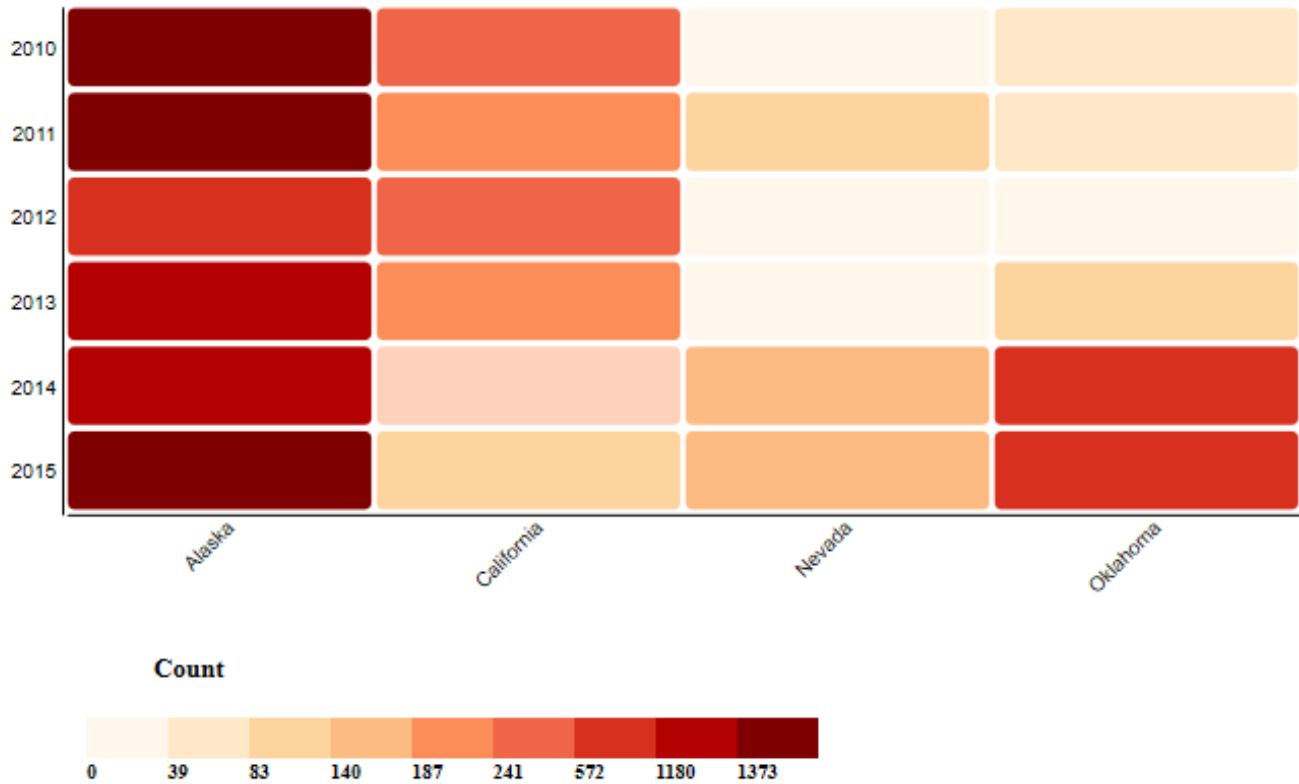


**Figure 4a:** Counts of earthquakes in the states that have 0-9 earthquakes in total from 2010 to 2015. When the mouse is placed on the grid (Tennessee, 2012), the value of 9 will show up.

## Visualizing Earthquake Counts by State 2010-2015 (M3+)

Range of

California 2014: 191



**Figure 4b:** Counts of earthquakes in the states that have 500 or above earthquakes in total from 2010 to 2015. When the mouse is placed on the grid (California, 2014), the value of 191 will show up.

### Q4 Deliverables:

The directory structure should look like:

Q4/

heatmap.(html / js / css)

earthquakes.csv

- **heatmap.(html / js / css)** - the html file created, and the js / css files if not included in heatmap.html
- **earthquakes.csv** - the dataset

## Q5 [20 points] Interactive Visualization

Use the dataset `state-year-earthquakes.csv` provided in the Q5 folder to create an interactive line chart and sub-chart.

This dataset<sup>[3]</sup> contains the earthquake counts by U.S. state and region, in the years 2010 to 2015 (inclusively). In the data sample below, each row under the header represents a state, its region, year, and count of earthquakes.

```
state, region, year, count
Hawaii, West, 2010, 17
Hawaii, West, 2011, 34
```

a. [3 points] **Create a line chart.**

Summarize the data by displaying the count of earthquakes by region for each year. You will need to sum the count of earthquakes by year for all states in their respective regions. Then, display one line for each of the 4 regions in the dataset.

Axes: All axes should automatically adjust based on the data. Do not hard-code any values.

- The vertical axis will represent the total count of earthquakes for a region. Display these values using a linear scale.
- The horizontal axis will represent the years. Display these values using a time scale.

b. [3 points] **Line styling, legend, and title.**

Lines: Each line should use a different color of your choosing to differentiate between regions. Display a dot shape over each data point in the line chart(i.e., a line should have one dot displayed for each year).

Legend: Display a legend on the right-hand portion of the chart that maps the line color to the name of the region.

Title: Display the title “US Earthquakes by Region 2010-2015” at the top of the plot.

The line chart should be similar in appearance to the chart provided in figure 5.b

**Note:** The data provided in `state-year-earthquakes.csv` requires some processing for aggregation. **All aggregation must only be performed in javascript; you must not modify `state-year-earthquakes.csv`.** That is, your code should first read the data from `.csv` file as is, then you may process the loaded data using javascript.

## US Earthquakes by Region 2010-2015

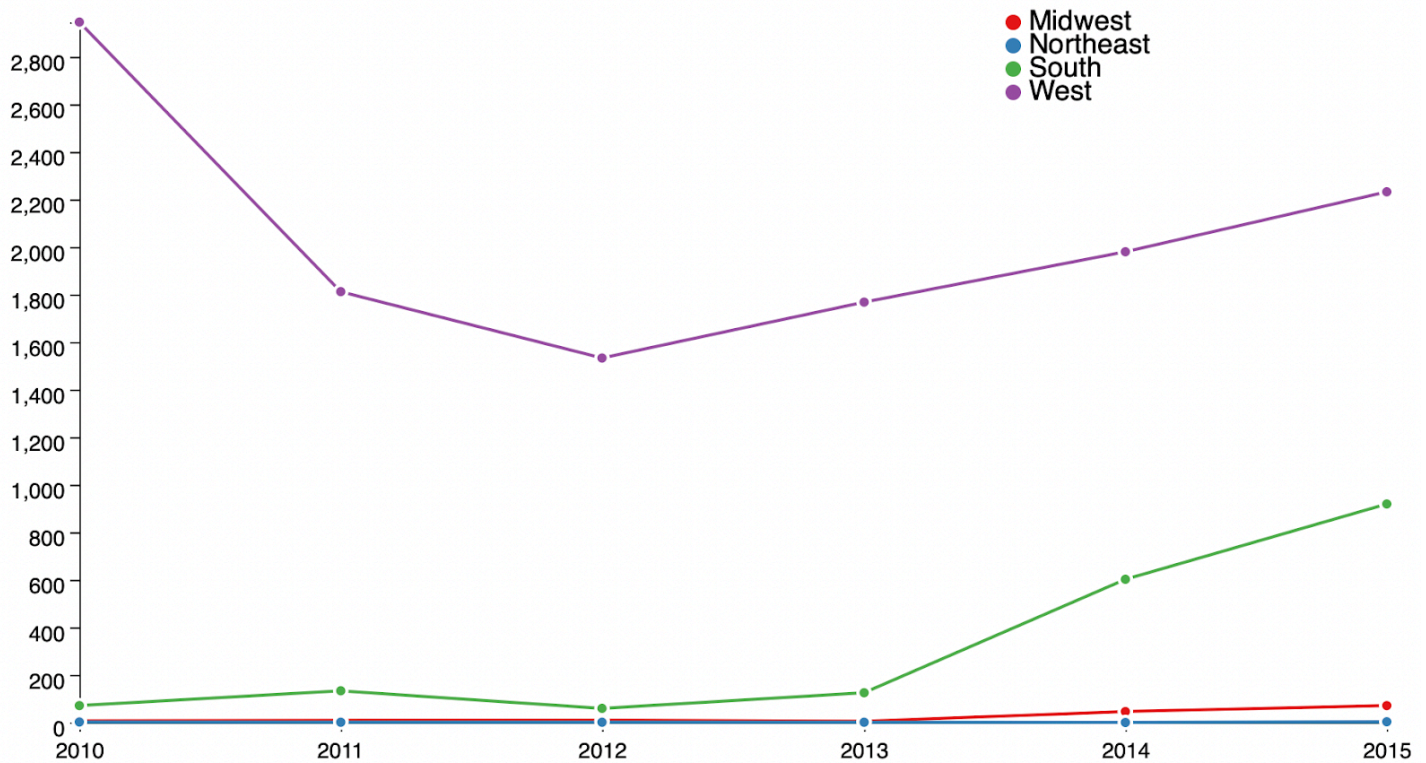


Figure 5b. Line Chart representing count of earthquakes by year for each region

**Interactivity and sub-chart.** In the next few parts of this question, you will create event handlers to detect mouseover and mouseout events over each dot shape that you added in Q5.b, so that when hovering over a dot, a horizontal bar chart representing the earthquake count for each state in a region will be shown below the line chart (for the year of that dot). For example, hovering over the dot for the West region in 2011 will display the bar chart for all states in the Western region and their individual earthquake counts in 2011. See Figure 5c for an example.



## Western Region Earthquakes 2011

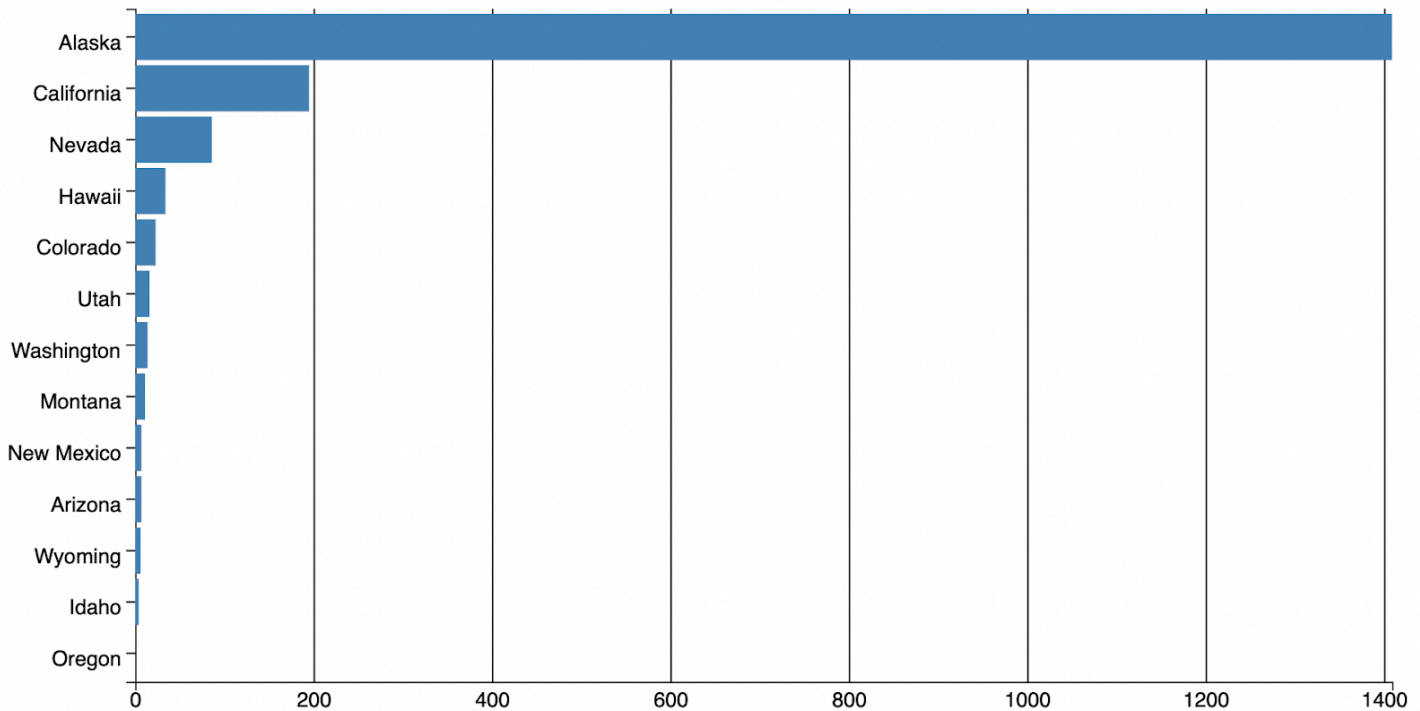


Figure 5c. Bar chart representing count of earthquakes for the Western region in 2011

c. [5 points] **Create a Bar chart**

Use a horizontal design for the bar chart, with one bar per state in the selected region. Each bar represents the count of earthquakes for one state in the selected year.

Axes: All axes should automatically adjust based on the data. Do not hard-code any values.

- The vertical axis represents states in a region. The state names should be sorted in ascending order on the vertical axis where the state with the lowest amount of earthquakes is at the bottom and the state with the highest order of earthquakes is at the top.

**Note:** If a region has multiple states with an equivalent count of earthquakes, then order those state names in ascending alphabetical order. e.g., Alabama, Delaware, and Florida have 0 earthquakes in 2013. They will be ordered as:

...
Florida
Delaware
Alabama

- The horizontal axis represents the count of earthquakes for the selected year. Display these values using a linear scale.

d. [3 points] **Bar styling and title**



- Bars: All bars should have the same color and a fixed bar width.
- Title: Display a title with format “<Region>ern Region Earthquakes <Year>” at the top of the plot where <Region>, and <Year> are the variables set by hovering over a dot in the line chart. e.g., If displaying earthquakes for the South in 2012, the title would read: “Southern Earthquakes 2012”

e. [3 points] **Mouseover Event Handling**

- The barchart and its title should only be displayed during mouseover events for a dot in the line chart.
- The dot in the line chart should change to a larger size during mouseover to emphasize that it is the selected point.

f. [3 points] **Mouseout Event Handling**

- The barchart and its title should be hidden from view on mouseout and the dot previously mouseovered should return to its original size.

The graph should exhibit interactivity similar to the .gif in Figure 5f.

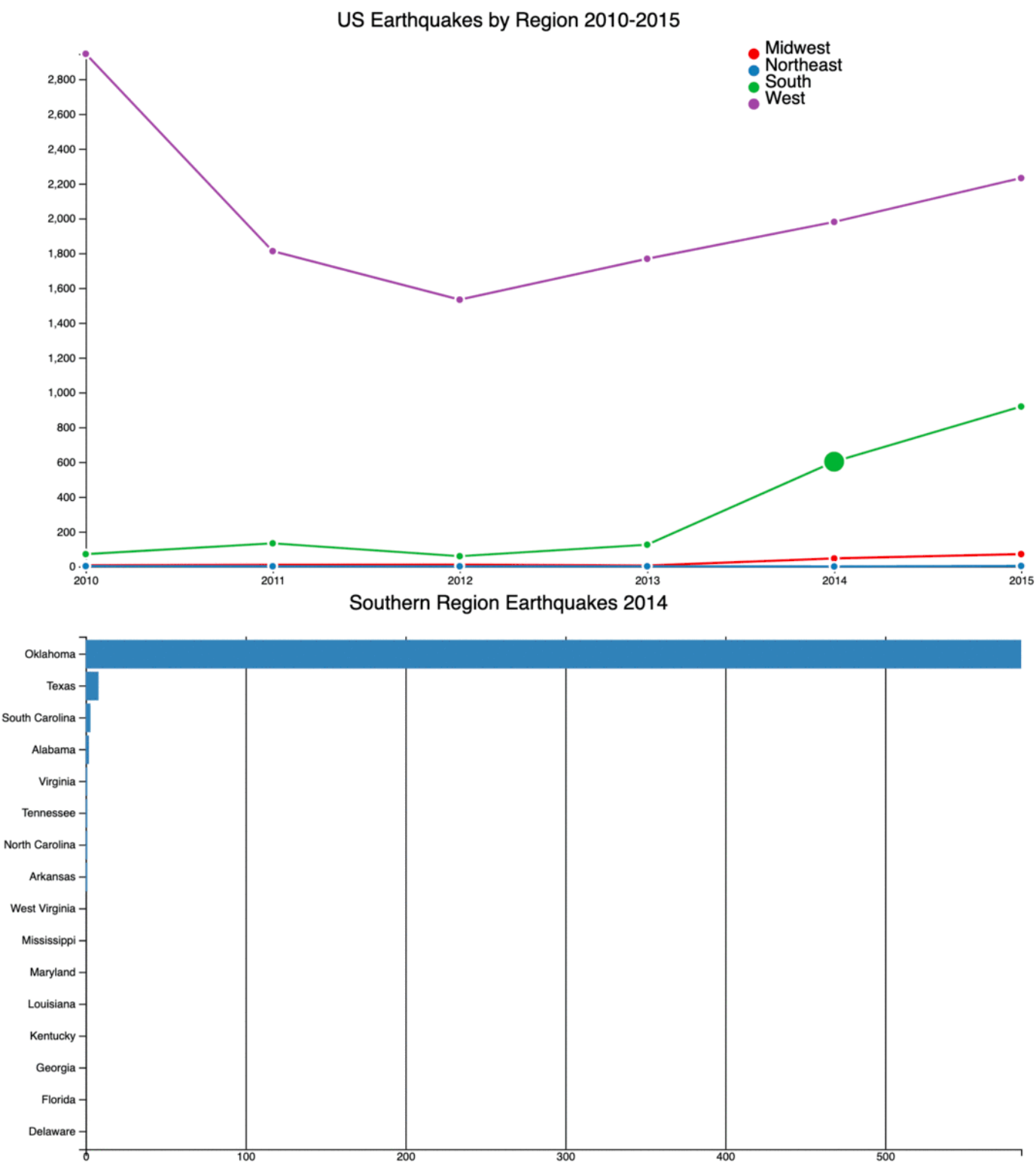


Figure 5f.Line Chart+BarChart demonstrating interactivity

**Q5 Deliverables:**

- The size of the dot in the line chart should be reset.

**The directory structure should be as follows:**

**Q5/**

interactive.(html/js/css)  
state-year-earthquakes.csv

- **interactive.(html/js/css)** - The html, javascript, css to render the visualization in Q5.
- **state-year-earthquakes.csv** - The datasets used to show the information of each state.

## Q6 [20 points] Choropleth Map of State Data

### Example of choropleth map: [Unemployment rates](#)

Use the dataset<sup>[4]</sup> provided in the file **state-earthquakes.csv** and **states-10m.json** (in the Q6 folder) and visualize them as a choropleth map.

- Each record in *state-earthquakes.csv* represents a state and is of the form `<State,Region,2010,2011,2012,2013,2014,2015>Total Earthquakes>`, where
  - **State**: the name of the state. e.g., Alabama.
  - **Region**: the region which the state belongs to. e.g., South.
  - **2010,...,2015**: the number of earthquakes in that state in 2010, ..., 2015, respectively.
  - **Total Earthquakes**: the total number of earthquakes in that state during 2010-2015 (the number of earthquakes in the *state-earthquakes.csv* file have been slightly modified from the original values and do not represent the official figures).
- The *states-10m.json* file is a [TopoJSON topology](#) containing two geometry collections: *states*, and *nation*.

- [15 points]** Create a choropleth map using the provided datasets, use Figure 6 below as reference.
  - [10 points]** The color of each state should correspond to the **log** of total earthquakes in that state (`Total Earthquakes` field in *state-earthquakes.csv*). i.e., darker colors correspond to higher total earthquakes in that state and lighter colors correspond to lower total earthquakes in that state **in log scale**. Use gradients of only **one** particular hue. Use [promises](#) (part of the *d3.v5.min.js* file present in the *lib* directory; there is no need to download or install anything) to easily load data from multiple files into a function. Use [topojson](#) (present in the *lib* folder) to draw the choropleth map.
  - [5 points]** Add a **vertical** legend showing how colors map to the total number of earthquakes. (In the example shown in Figure 6, there are 7 color gradations, but you must use **exactly 9** in your submission.)
- [5 points]** Add a tooltip using the [d3-tip.min](#) library (in the *lib* folder). On hovering over a state, the tooltip should show the following information on each line: (1) state name, (2) region, and (3) total earthquakes. The tooltip should appear when the **mouse** hovers over the state. On mouseout, the tooltip should disappear. Use Figure 6 below as reference. We recommend that you position the tooltip some distance away from the **mouse cursor**, which will prevent the tooltip from “flickering” as you move the mouse around quickly (the tooltip disappears when your mouse leaves a state and enters the tooltip’s bounding box). Please ensure that the tooltip is fully visible (i.e., not clipped, especially near the page edges).

**Note:** You **must** create the tooltip by **only** using **d3-tip.min.js** in the **lib** folder.

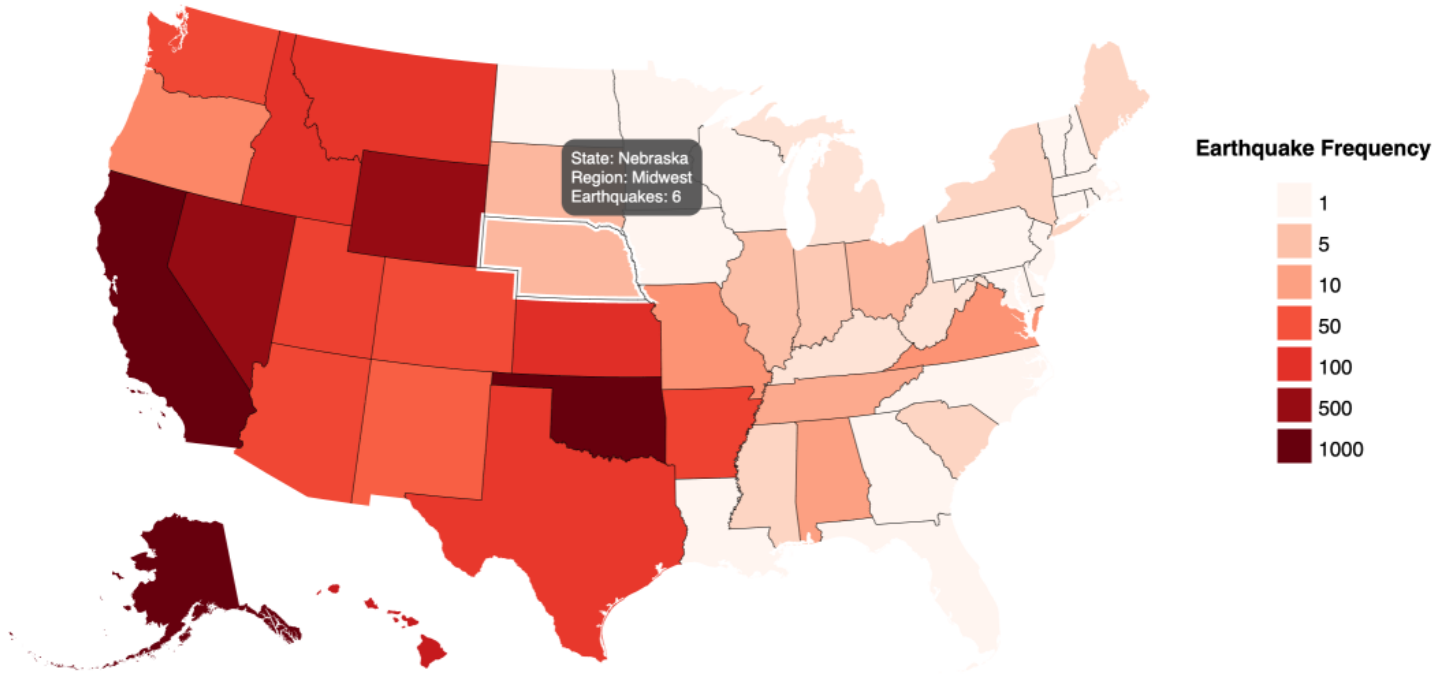


Figure 6. Reference example for Choropleth Maps

#### Q6 Deliverables:

The directory structure should be organized as follows:

Q6/

```
choropleth.(html/js/css)
state-earthquakes.csv
states-10m.json
```

- **choropleth.(html /js /css)**- The html/js/css file to render the visualization.
- **state-earthquakes.csv** - The datasets used to show the information of each state.
- **states-10m.json** - Dataset needed to draw the map.

## Q7 [5 points] Pros and Cons of Visualization Tools

This question has two parts. The first part is optional and WILL NOT be graded and the second part is required and WILL be graded.

a. **[OPTIONAL - NO points] Line chart using R.** Use R to create a line chart that looks the same as the 4th line chart in Q3, i.e., the line chart in Q3c with log scale y-axis.

b. **[5 points] Comparison write-up.** If you did part a, you may use your experience with R from part a to

complete this comparison write-up by comparing it with Tableau and D3 in the following aspects. If you did not do part a, pick a visualization system/tool/library/framework that you are familiar with (R, R Shiny, Python, Plotly, Excel, JMP, Matlab, Mathematica, Julia, etc.), and compare it with Tableau and D3 in the following aspect. Your write-up for each comparison aspect should be within the word limitations specified.

1. Ease to develop for developers [40 words]
2. Ease to maintain the visualization for developers (e.g., difficulty of the maintenance of the product as the requirements change, the data changes, the hosting platform changes, etc.) [40 words]
3. Usability of visualization developed for end users [40 words]
4. Scalability of visualization to “large” datasets [40 words]
5. System requirements to run the visualization (e.g., browsers, OS, software licensing) for end users [40 words]

Your answer will depend on what you have learned from working through the questions in this assignment, and your personal experience.

**Note:** Your claims should be well justified, supported with compelling reasons. Simply stating that a tool is better (or worse) than D3 without justifications will receive a low (or no) score.

We recommend formatting your answers as bullet lists for better readability. For example:

1. Ease to develop
  - R: ...
  - Tableau: ...
  - D3: ...
2. Ease to maintain the visualization
  - R: ...
  - Tableau: ...
  - D3: ...
- ...

Text (e.g., “Ease to develop”, “D3:” above) mainly for organizing you answers do not count towards the word limit.

### Q7 Deliverables:

The directory structure should be as follows:

Q7/

linechart.jpg (optional)

analysis.txt

- **chart.jpg** - the line chart you created using R. (note: this is optional and will not be graded)
- **analysis.txt** - comparison of R and D3.

**Important:** folder structure of the zip file that you submit

You are submitting a single zip file HW2-GTUsername.zip (e.g., HW2-jdoe3.zip, where “jdoe3” is your GT username), which must unzip to the following directory structure (i.e., a folder “HW2-jdoe3”, containing folders “Q1”, “Q2”, etc.). The files to be included in each question’s folder have been clearly specified at the end of each question’s problem description above.

```

HW2-GTUsername/
  lib/
    d3.v5.min.js
    d3-tip.min.js
    d3-scale-chromatic.v1.min.js
    topojson.v2.min.js
    d3-dsv.min.js
    d3-fetch.min.js
  Q1/
    table.png
    barchart.png
    age-distribution.csv
    population.csv
  Q2/
    graph.(html / js / css)
  Q3/
    linecharts.(html / js / css)
    linecharts.pdf
    earthquakes.csv
    explanation.txt
  Q4/
    heatmap.(html / js /css)
    earthquakes.csv
  Q5/
    interactive.(html / js / css)
    state-year-earthquakes.csv
  Q6/
    choropleth.(html / js / css)
    state-earthquakes.csv
    states-10m.json
  Q7/
    linechart.jpg (optional)
    analysis.txt

```

Version 2

- 
- [1] Source: [here](#)
  - [2] Source: USGS <https://earthquake.usgs.gov/earthquakes/browse/stats.php>
  - [3] Source: USGS <https://earthquake.usgs.gov/earthquakes/browse/stats.php>
  - [4] Source: USGS <https://earthquake.usgs.gov/earthquakes/browse/stats.php>
- 

Published by [Google Drive](#) – [Report Abuse](#) – Updated automatically every 5 minutes

---