CSE 6242 / CX 4242: Data and Visual Analytics
Georgia Tech, Fall 2020

Homework 2: Tableau, D3 Graphs and Visualization

By 32+ awesome TAs of CSE6242A,Q,QSZ,OAN,O01,O3/CX4242A for our 1200+ students

Submission Instructions and Important Notes

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

1. Always check to make sure you are using the **most up-to-date assignment** (version number at bottom right of this document).
2. Submit a single zipped file, called "HW2-GTusername.zip" that unzips to a folder called "HW2-GTusername", containing all the deliverables including source code/scripts, data files, and readme. Example: "HW2-jdoe3.zip" if GT account username is "jdoe3". Your GT username is the one with letters and numbers. Only .zip is allowed; no other format will be accepted.
    a. At the end of this assignment, we have specified a folder structure you must use to organize your files. 5 points will be deducted for not following this strictly.
    b. Due to the large class size, we may need to use auto-grading scripts to grade some of your deliverables, to help speed up grading, so we can return graded work to you sooner. Thus, it is extremely important that you strictly follow the instructions.
    c. Do not include any intermediate files you may have generated while working on the task, unless your work is absolutely dependent on it to get the final result — there are rarely any situations that would justify such a need.
    d. Wherever you are asked to write down an explanation for the task you perform, stay within the word limit or you may lose points.
3. 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.
4. 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**.**
5. You may use JavaScript ES6 in your solutions for this homework.
6. You should submit your work by the official **Due** date on Canvas, the same date specified on the course schedule.
    a. Every homework assignment deliverable comes with a 48-hour "grace period". You do **not** need to ask before using this grace period.
    b. You may re-submit your work before the grace period expires **without penalty**, but Canvas will mark your submission as **"late"**.
    c. 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.**
    d. Any deliverable submitted after the grace period will get **0 credit**. We recommend that you submit your work before the grace period begins.
    e. We will **not** consider late submission of any missing parts of a deliverable. To make sure you have submitted everything, download your submitted files to double check. If you are submitting large files, you are responsible for making sure they get uploaded to the system in time. You have 48 hours to verify your submissions!

Download the HW2 Skeleton before you begin.

Grading

The maximum possible score for this homework is 100 points. Students can choose to complete any 90 points worth of work to receive the full 15% of the final course grade, and can receive more than 15% if additional work is submitted. For example, if a student scores 100 points, that student will receive (100 / 90) * 15 = 16.67% course grade..

Homework Overview

"Visualization gives you answers to questions you didn't know you have" - Ben Schneiderman

This homework focuses on exploring and creating data visualizations using two of the most popular tools in the field. Data visualization is an integral part of exploratory analysis and communicating key insights. All of the questions in this homework use data on the same topic in order to highlight some of the uses and strengths of different types of visualizations. The data for this homework comes from BoardGameGeek and includes information on games' ratings, popularity, and metadata.

Part 1 of the homework uses Tableau to connect to online data which feeds multiple visualizations including a table and bar charts. Part 2 of the homework uses D3 and includes graphs with different scales, network graphs, and a map.

Below are some terms you will often see in the questions:

- **Rating** – a value from 0 to 10 given to each game. BoardGameGeek calculates a game's overall rating in different ways including **Average** and **Bayes**, so make sure you are using the correct rating called for in a question. A higher rating is better than a lower rating.
- **Rank –** the overall rank of a boardgame from 1 to *n,* with ranks closer to 1 being better and *n* being the total number of games. The rank may be for all games or for a subgroup of games such as abstract games or family games.

In Q1, you will design a table, a grouped bar chart, and a stacked bar chart with filters. The data for this question is hosted online and will help you practice connecting Tableau to online data sources.

Questions 2-5 highlight different features of D3. The provided skeletons scaffold coding in D3 with the most complete template code being provided for Q2. Q4 and Q5 provide scaled back templates. Q3 does not provide template code, and is an excellent opportunity to separate html, css, and js files because a separate js file can be used for each of the visualizations.

Q2: a network graph shows relationships between games. You will add interactive features like pinning nodes to give the viewer some control over the visualization.

Q3: you will explore temporal patterns in the BoardGameGeek data, using line charts to compare how the number of ratings grew from month to month for 8 games. You will also integrate additional data about board game rankings onto these line charts and explore the effect of axis scale choice on what information is emphasized in the graph.

Q4: you will create line charts that use interactive elements to display additional data. This time, the line charts will show the number of games with each rating for multiple years. You will then implement a bar chart that appears when you mouse over a point on the line chart.

Q5: you will create a choropleth map to explore the average rating of each game in different countries.

Note the following important points

1. We highly recommend that you use the latest Firefox browser to complete this question. We will grade your work using **Firefox 80.0.**
2. You will work with **version 5** of D3 in this homework. You must **NOT** use any D3 libraries (d3*.js) other than the ones provided in the **lib** folder.
3. **You may need to setup a local HTTP server in the root (hw2-skeleton) folder to run your D3 visualizations**, depending on your web browser, 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. (for more details, see link).
4. **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="C:/Users/polo/hw2-skeleton/lib/d3.v5.min.js"></script>
5. 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 reads data from `earthquake.csv`, the path should simply be "earthquake.csv" and **NOT** an absolute path such as "C:/Users/polo/hw2-skeleton/Q/earthquake.csv".
6. 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 [25 points] Designing a good table. Visualizing data with Tableau.

Setting Up Tableau

Tableau has provided us with student licenses for *Tableau Desktop*, available for Mac and Windows. Go to Tableau and select "Products/Tableau Desktop". After 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 already have Tableau installed on your machine, for example from a previous trial, you may use this key to reactivate it.

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.

Connecting to the Data

Complete all parts of Q1 using **a single Tableau workbook**. (Technically, you could use multiple workbooks, but we do not recommend that here. The directions below assume you are using one workbook.)

1. You will need a data.world account (created using any email you want) to access the data for Q1.
2. Q1 will require connecting Tableau to multiple data sources. You can connect multiple data sources within one workbook by following the directions here.

3. Open Tableau and when prompted to connect to a data source choose To a Server – **Web Data Connector**. You may need to select “More...” to see Web Data Connector as an option.
4. Enter this URL (with SQL query embedded) to connect to part of the data.world data set on board games. This data will be used in Q1a and Q1b. You may be prompted to log in to data.world and authorize Tableau. Do not edit the provided SQL query.
5. We recommend renaming the data connection since you will have multiple connections to mjpetrey/boardgamegeek. Rename the connection to something that makes sense to you. (Clicking on the text lets you edit it.)
6. Click to create a new worksheet, and Tableau will then automatically create a data extract. You now have the data needed for Q1a and Q1b! (Live data connections are not an option when connecting to data.world. You can read a comparison of Tableau’s data connection options here.)
7. To add a new data source Click on Data – **New Data Source**. Then repeat steps 3-6 using this URL for Q1c.

If you are unable to connect to data.world for any reason, flat data files for Q1 have also been provided in the skeleton folder. The preferred data source is connecting online as that provides valuable experience (and something you may choose to use in your final projects). The provided csv files are identical to those hosted online and can be loaded directly into Tableau. That is, if data.world does not work for you, use the csv files.

a. **[5 points] Good table design.** You want to help a board game design company to analyze the current popular board game data from the website BoardGameGeek. Create a well-designed table to visualize the data contained in *popular_board_game.csv*. You can use any tool (e.g., Excel, HTML, Tableau) to create the table. If you choose to use a tool other than Tableau to make the table, you will still need to load the same data into Tableau for use in Q1b.

The company is interested in grouping popular games into “support solo” (minimum player = 1) and “not support solo” (minimum player > 1), because single-player games require a different design strategy.

Instructions:
Your table should clearly communicate information about these two groups (games that support solo & games that do not support solo) simultaneously. For each group, show:

1. Total game count in each category (fighting, economic, ...)
2. The most representative game (game with the most ratings) in each category. If more than one game have the same ratings, pick the game that you prefer.
3. Average rating of games in each category, rounded to the nearest 2 decimal places
4. Average playtime of games in each category, rounded to the nearest 2 decimal places
5. In the bottom left corner below your table include your GT username. In Tableau, this can be done by including a caption when exporting an image of a worksheet or by adding a text box to a dashboard. Refer to the tutorial here.
6. **Save the table as table.png**
7. In Tableau, to save a worksheet image, go to Worksheet - Export - Image. And to save a dashboard image, go to Dashboard - Export Image (Do not simply take a screenshot since your image should have a high resolution).

Note: If there is no game under a particular group and category, think about how to visually represent missing data in your table.

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 limited 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 52 and onwards of the lecture slides.

b. **[10 points] Grouped bar chart.** You want to help this board game design company better understand the relationship between game playtime and game category among popular board games. Visualize *popular_board_game.csv* as a grouped bar chart. Your chart should display game category (e.g., fighting, economic) along the horizontal axis and game count along the vertical axis. Also show game playtime (e.g., <=30, (30, 60]) for each game category.

The main goal here is for you to get familiarized with Tableau. 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 Figure 1a, based on the tutorial from Tableau, and you are not limited to the techniques presented there.

Instructions:

1. Your design should be a grouped bar chart. For each game category, show the game count for each game playtime.
2. Your design should have clear labeled axes and a clear chart title. Include a legend for your chart.
3. In the bottom left corner of your image include your GT username. In Tableau, this can be done by including a caption when exporting an image of a worksheet or by adding a text box to a dashboard. Refer to the tutorial here.
4. **Save the chart as grouped_barchart.png**
5. To save a worksheet image, go to Worksheet - Export - Image. And to save a dashboard image, go to Dashboard - Export Image (Do not simply take a screenshot since your image should have a high resolution).

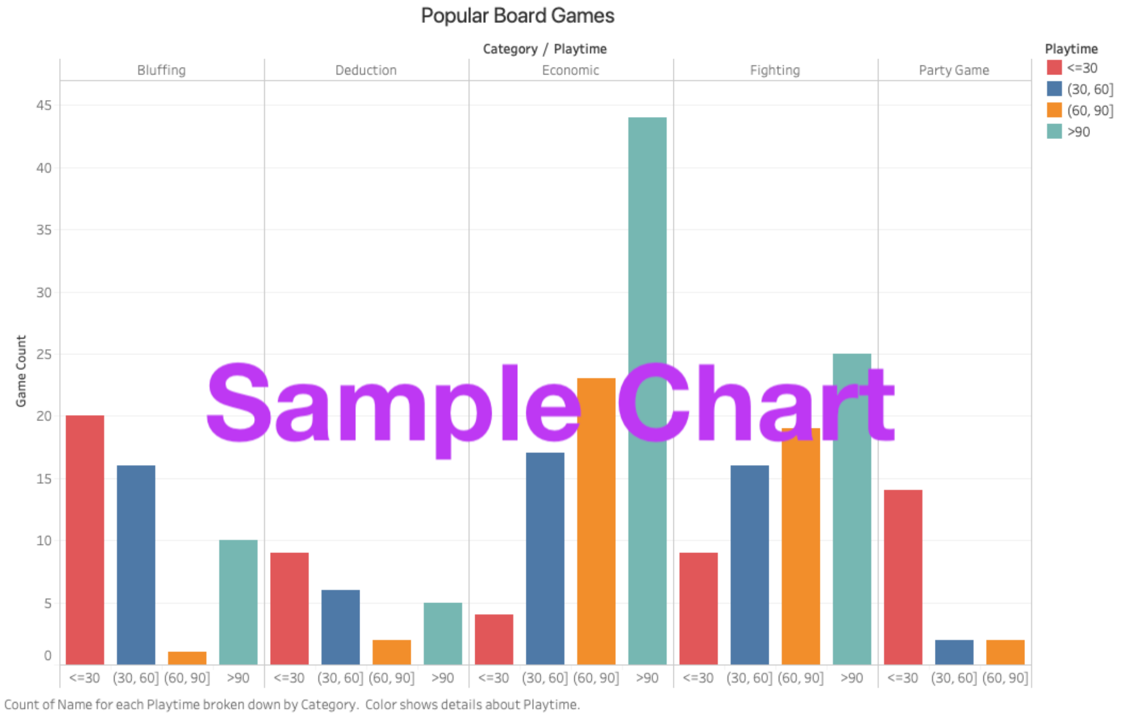


Figure 1a: Example of a grouped bar chart. Your chart may appear different, and can earn full credit if it meets all stated requirements. Your submitted image should include your GT username in the bottom left.

c. **[10 points] Stacked bar chart.** After understanding the relationship between game category and their playtime, the game company now wants to know the count of games in different category, and if there is any relationship between game category and how they are played (their playing mechanics). They also want to know how player size changes this information.

A stacked bar chart is one way that can help understand this kind of information, where each bar represents a game category. A bar's length represents the total game count in that category. For a bar, its "sub bars" show how games are divided into different game mechanics. (Optional reading: the effectiveness of stacked bar charts is often debated --- sometimes, they can be confusing, difficult to understand, and may make data series comparisons challenging.)

Instructions**:**

1. Create a 'Worksheet' with a stacked bar chart that shows game count for each game's playing mechanics (sub-bars) for each game category
2. Your chart should display game counts along the vertical axis and category along the horizontal axis
3. Your design should have clear axes labels and a clear chart title. Include a legend for your chart.
4. Create a **dashboard** using the sheet you created in the step 1
5. Add a filter for number of 'Max.Players' allowed in each game. Then update the chart using this filter to generate the following chart images (Refer to the tutorial on how to add filter in a dashboard here. Make sure to add 'Max.Players' in the filter shelf in the Worksheet first, like this.):
    a. Select "2 Players" only in the filter. Save the resulting chart as '**stacked_barchart_1.png**'
    b. Select "4 Players" only in the filter. Save the resulting chart as '**stacked_barchart_2.png**'
    c. Both images should include your GT username in the bottom left. This can be added using a text box. Refer to the tutorial here.
6. To save a dashboard image, go to Dashboard - Export Image. Do not submit screenshots.

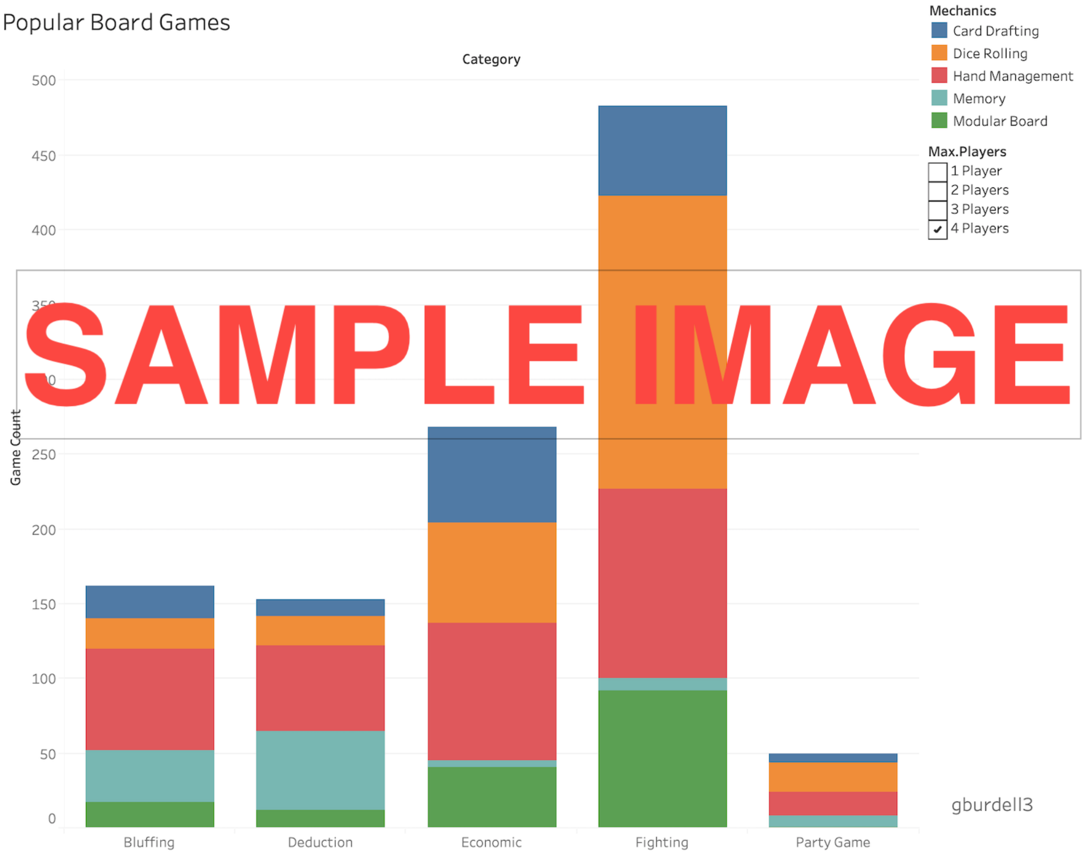


Figure 1b: Example of a stacked bar chart after selecting "4 Players" in Max.Players filter. Your chart may appear different, and can earn full credit if it meets all the stated requirements. Your submitted image should include your GT username in the bottom left.

Q1 Deliverables:
The directory structure should be as follows:

Q1/

table.png
grouped_barchart.png
stacked_barchart_1.png
stacked_barchart_2.png

- **table.png** - An image/screenshot of the table in Q1.a (png format **only**).
- **grouped_barchart.png** - An image of the chart in Q1.b
- **stacked_barchart_1.png** - An image of the chart in Q1.c after filtering data for Max.Players = 2
- **stacked_barchart_2.png** - An image of the chart in Q1.c after filtering data for Max.Players = 4

Note: Your Tableau workbooks will not be graded. Your images should be clear and of high resolution.

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) and an undirected graph dataset of boardgames, board_games.csv file (in the Q2 folder). The dataset for this question had inspiration from this post on reddit network visualization using boardgames in which the author calculates the similarity between board games based on categories and game mechanics where the edge value between each board game (node) is the total weighted similarity index. This dataset has been modified and simplified for this question and does not fully represent actual data found from this post.

Note: You are welcome to split graph.html into graph.html, graph.css, and graph.js. Make sure that all paths in your code are **relative paths**. Nonfunctioning code will result in a **five point deduction.**

a. **[2 points] Adding node labels**: Modify graph.html to show the node label (the node name, i.e., the source) at the **top right** of each node in **bold**. 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:

- If the value of the edge is equal to 0 (similar), the edge should be **gray, thick,** and **solid**.
- If the value of the edge is equal to 1 (not similar), the edge should be **green, thin,** and **dashed**.

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, which will likely lead to low-quality visualization (e.g., nodes that are mere points, barely visible, or of huge sizes).

   Note: D3 v5 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 is here.

2. **[1.5 points]** The degree of each node should be represented by varying colors. Pick a meaningful color scheme (hint: color gradients). There should be at least 3 color gradations and it must be visually evident that the nodes with a higher degree use darker/deeper colors and the nodes with lower degrees use lighter colors. You can find example color gradients at Color Brewer.

d. **[6 points] Pinning nodes**:

1. **[2 points]** Modify the code so that dragging a node will fix the node’s position such that it will not be modified by the graph layout algorithm (note: pinned nodes can be further dragged around by the

user). Node pinning is an effective interaction technique to help users spatially organize nodes during graph exploration. The d3 API for pinning nodes have evolved over time. We recommend reading this post when you work on this sub-question.

2. **[2 points]** Mark pinned nodes to visually distinguish them from unpinned nodes, e.g., show pinned nodes 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. When a node is no longer pinned, it should move freely again.

e. **[1 point] Add GT username:** Add your Georgia Tech username (usually includes a mix of letters and numbers, e.g., gburdell3) to the top right corner of the force-directed graph (see example image).

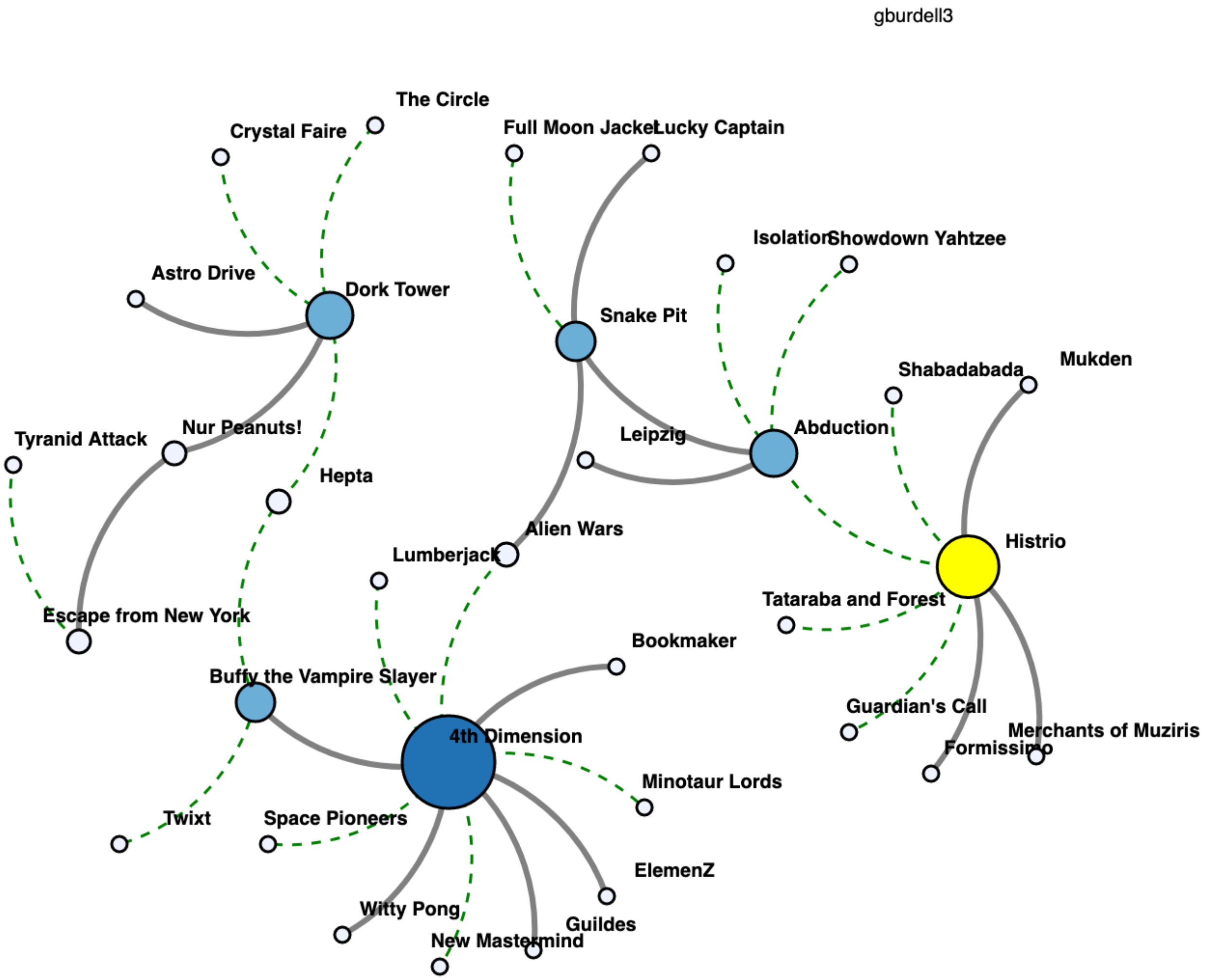


Figure 2a. Example of Visualization with pinned node (yellow). Your chart may appear different, and can earn full credit if it meets all the stated requirements.

Q2 Deliverables:

The directory structure should be as follows:

Q2/
graph.(html / js / css)
board_games.csv

- **graph.(html / js / css)** - the html file created, and the js / css files (if you decide to save js and css in their own separate files)

- **board_games.csv** - the dataset

Q3 [15 points] Line Charts

Use the dataset provided in the file *boardgame_ratings.csv* (in the Q3 folder) to create line charts.

Refer to the tutorial for line chart here.

Note: You will create four charts 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 (Figure 3a) that visualizes the number of board game ratings from November 2016 to August 2020 (inclusively), for the eight board games: ['Catan', 'Dominion', 'Codenames', 'Terraforming Mars', 'Gloomhaven', 'Magic: The Gathering', 'Dixit', 'Monopoly']. Use D3 Tableau10 color scheme to differentiate these board games. Add each board game's name next to its corresponding line. For the x-axis, show a tick label for every three months. Use D3 axis.tickFormat() and d3.timeFormat() to format the ticks to display abbreviated months and years. For example, Jan 17, Apr 17, Jul 17. (See Figure 3a and its x-axis ticks).

- Chart title: Number of Ratings 2016-2020
- Horizontal axis label: Month. Use D3.scaleTime().
- Vertical axis label: Num of Ratings. Use a linear scale (for this part a).

b. [5 points] Adding board game rankings. Create a line chart (Figure 3b) 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. Modify the code to visualize how the rankings of ['Catan', 'Codenames', 'Terraforming Mars', 'Gloomhaven'] change over time by adding a symbol with the ranking text on their corresponding lines. Show the symbol for every three months, similar to the x-axis ticks in part a. (See Figure 3b). Add a legend to explain what this symbol represents next to your chart (See the Figure 3b bottom right).

- Chart title: Number of Ratings 2016-2020 with Rankings

c. [5 points] Axis scales in D3. Create two line charts (Figure 3c-1,2) 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 chart uses the log scale for its vertical axis (only). Keep the symbols and the symbol legend you implemented in part b. At the bottom right of the last chart, add your **GT username** (e.g. gburdell3, see Figure 3c-2 for example). 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 (Figure 3c-1)
  - ○ Chart title: Number of Ratings 2016-2020 (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 (Figure 3c-2)
  - ○ Chart title: Number of Ratings 2016-2020 (Log Scale)
  - ○ This chart uses the **log scale** for its vertical axis (only)
  - ○ Other features should be the same as part b.

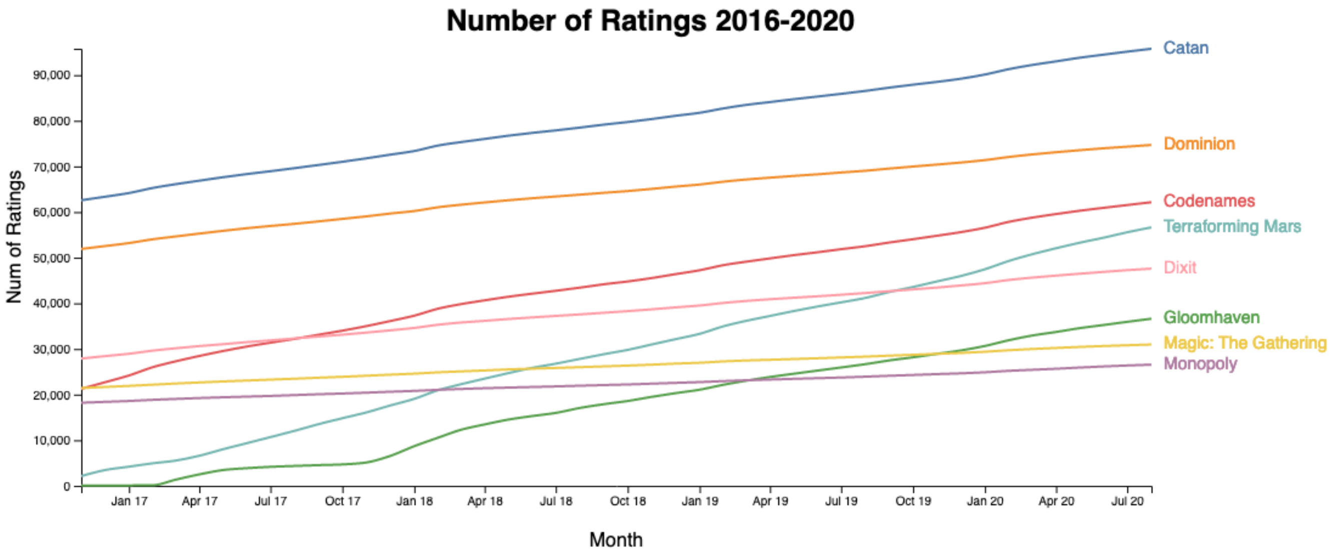


Figure 3a: Example line chart. Your chart may appear different, and can earn full credit if it meets all stated requirements.

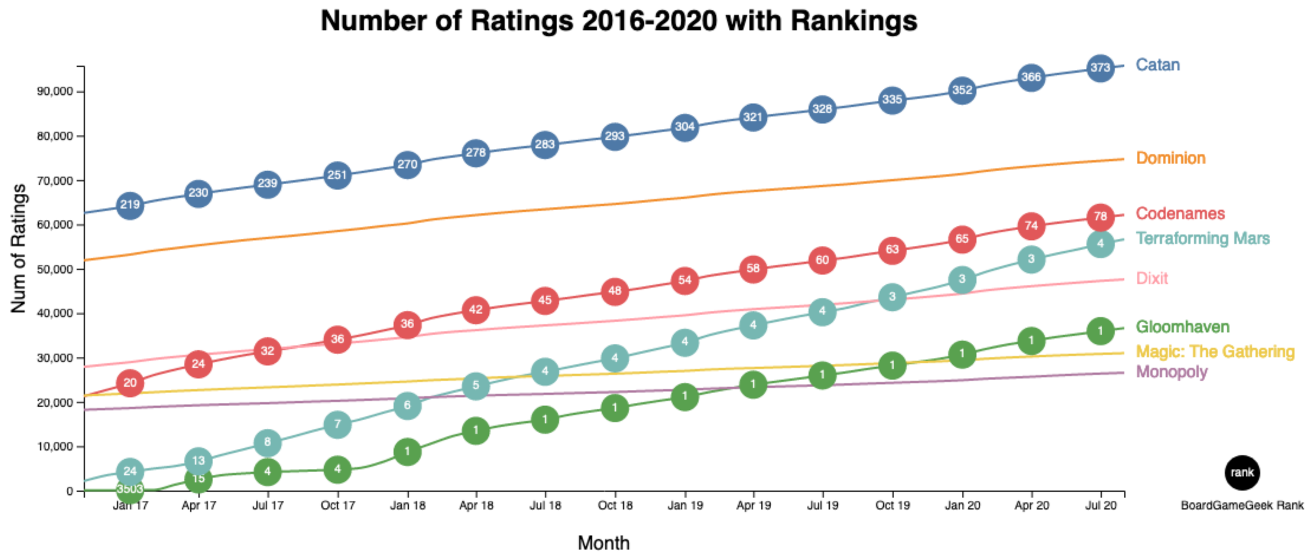


Figure 3b: Example of a line chart with rankings. Your chart may appear different, and can earn full credit if it meets all stated requirements.

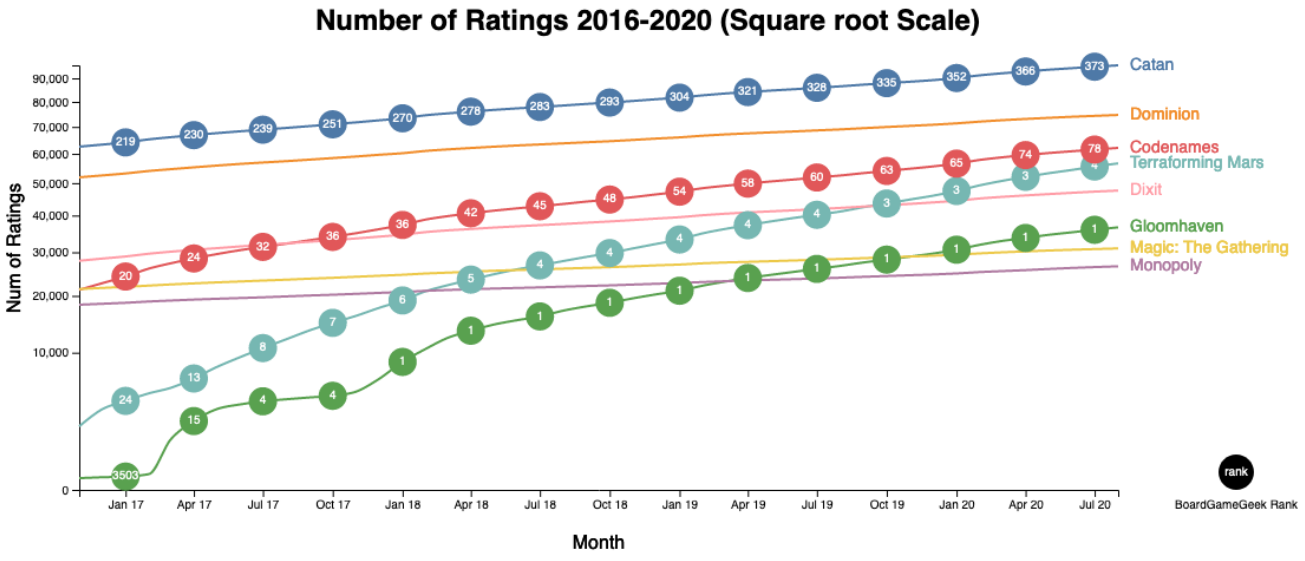


Figure 3c-1: Example of a line chart using square root scale. Your chart may appear different, and can earn full credit if it meets all stated requirements.

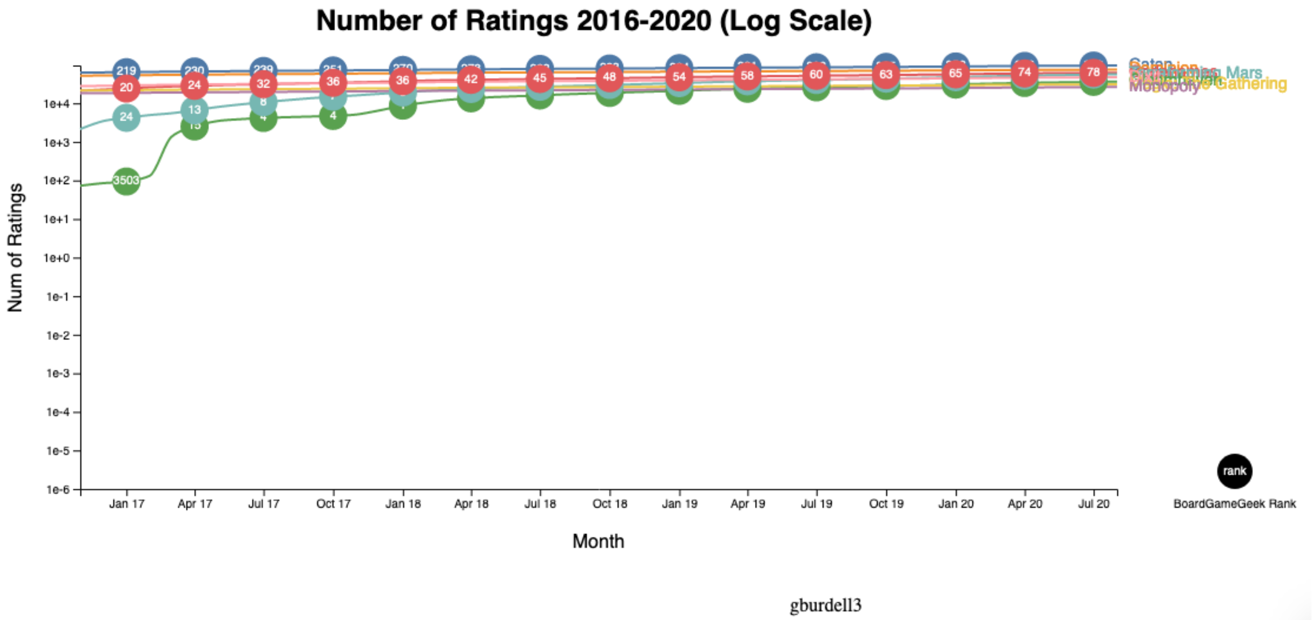


Figure 3c-2: Example of a line chart using log scale. Your chart may appear different, and can earn full credit if it meets all stated requirements.

Q3 Deliverables:
The directory structure should be organized as follows:

Q3/
boardgame_ratings.csv
linecharts.(html / js / css)
linecharts.pdf
explanation.txt

- **boardgame_ratings.csv** - the dataset.
- **linecharts.(html / js / css)** - the html file created, and the js / css files (if you decide to save js and css in their own separate files).
- **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 chart (**hint:** use CSS page break). Clearly title the charts as instructed (see examples in Figure 3). Make sure your Georgia Tech username that you added in the last chart (see example figure 3c-2) is shown in this PDF document.

- **explanation.txt** - the text file explaining your observations for Q3.c.

Q4 [20 points] Interactive Visualization

Use the dataset `average-rating.csv` provided in the Q4 folder to create an interactive frequency polygon line chart. This dataset contains a list of games, their ratings and supporting information like numbers of users who rated and year it was published. In the data sample below, each row under the header represents a game name, year, average rating, and number of users who rated the game.

```
name,year,average_rating,users_rated
Codenames,2015,7.71148,51209
King of Tokyo,2011,7.23048,48611
```

a. **[3 points] Create a line chart.** Summarize the data by displaying the count of board games by rating for each year. Round each rating down to the nearest integer, using Math.floor(). For example, a rating of 7.71148 becomes 7. For each year, sum the count of board games by rating. Display one line for each of the 5 years (2015 – 2019) in the dataset.

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

   - The vertical axis represents the count of board games for a given rating. Use a linear scale.
   - The horizontal axis represents the ratings. Use a linear scale.

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

   - For each line, use a different color of your choosing. Display a filled circle for each rating-count data point.
   - Display a legend on the right-hand portion of the chart to show how line colors map to years.
   - Display the title “Board games by Rating 2015-2019” at the top of the chart.
   - Add your GT username (usually includes a mix of lowercase letters and numbers, e.g., gburdell3) beneath the title (see example figure 4b).

Figure 4b shows an example line chart design. Yours may look different, and can earn full credit if it meets all stated requirements.

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

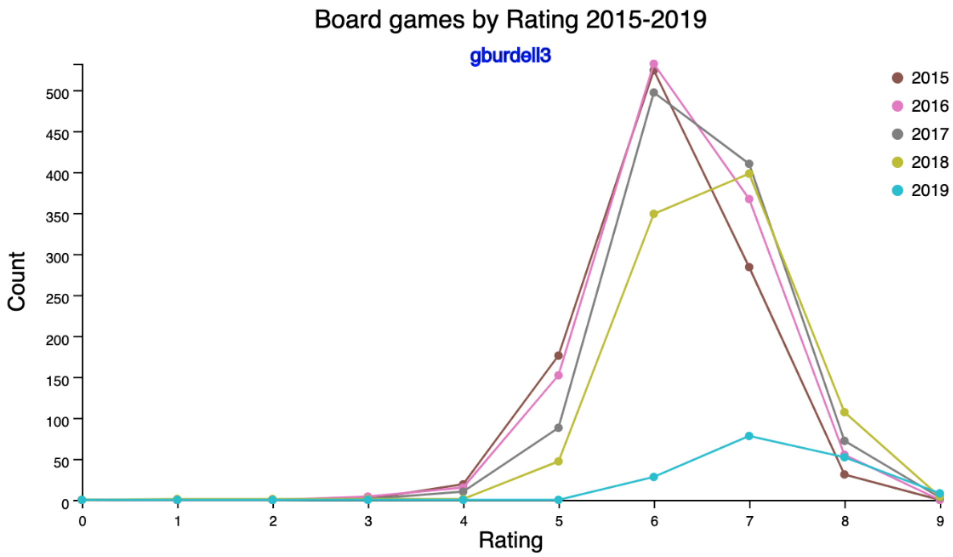


Figure 4b: Line chart representing count of board games by rating for each year. Your chart may appear different, and can earn full credit if it meets all stated requirements.

Interactivity and sub-chart. In the next few sub-questions, you will create event handlers to detect mouseover and mouseout events over each circle that you added in Q4.b.

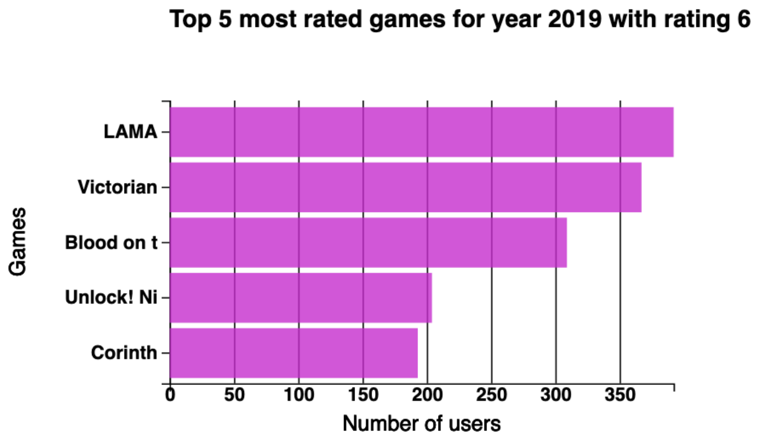


Figure 4c: Bar chart representing the number of users who rated the top 5 board games with the rating 6 in year 2019. Your chart may appear different, and can earn full credit if it meets all stated requirements.

c. **[5 points] Create a horizontal bar chart**, so that when hovering over a circle, that bar chart will be shown *below* the line chart. The bar chart displays the top 5 board games that received the highest numbers of user ratings (`users_rated`), for the hovered year and rating. For example, hovering over the rating-6 circle for 2019 will display the bar chart for the number of users who rated the top 5 board games. Figure 4c shows an example design. Show one bar per game. The bar length represents the number of users who rated the game.
Note: No bar chart should be displayed when the count of games is 0 for hovered year and rating.

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

- The vertical axis represents the board games. Sort the game names in ascending order, such that the game with the smallest `users_rated` is at the bottom, and the game with the highest `users_rated` is at the top. Some boardgame names are quite long. For each game name, display its first 10 characters (if a name has fewer than 10 characters, display them all). A space counts as a character. The horizontal axis represents the number of users who rated the game (for the hovered year and rating). Use a linear scale.

- Set horizontal axis label to ‘Number of users’ and vertical axis label to ‘Games’.

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

- Bars: All bars should have the same color and a uniform bar thickness.
- Title: Display a title with the format “Top 5 Most Rated Games of <Year> with Rating <Rating>” at the top of the chart where <Year> and <Rating> are what the user hovers over in the line chart. For example, hovering over rating 6 in 2015, the title would read: “Top 5 Most Rated Games of 2015 with Rating 6”

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

- The bar chart and its title should only be displayed during mouseover events for a circle in the line chart.
- The circle in the line chart should change to a larger size during mouseover to emphasize that it is the selected point.
- When count of games is 0 for hovered year and rating, no bar chart should be displayed. The hovered-over circle on the line graph should still change to a larger size to show it is selected.

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

- The bar chart and its title should be hidden from view on mouseout and the circle previously mouseover-ed should return to its original size in the line chart.

The graph should exhibit interactivity similar to Figure 4f where the mouse is over the larger circle.

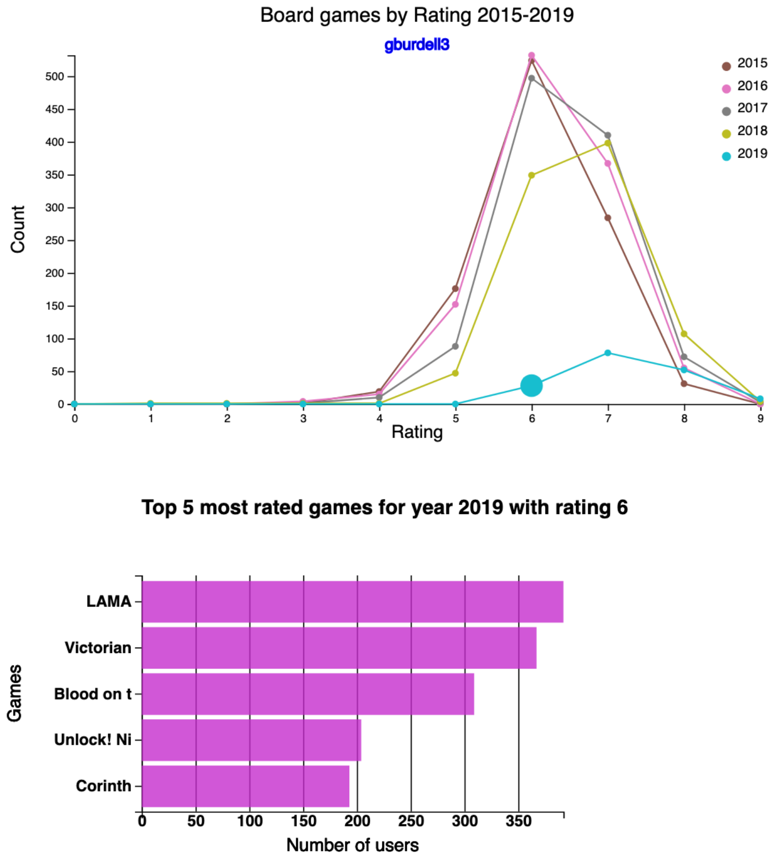


Figure 4f: Line chart and bar chart demonstrating interactivity. Your chart may appear different, and can earn full credit if it meets all stated requirements.

Q4 Deliverables:
The directory structure should be as follows:

Q4/
interactive.(html/js/css)
average-rating.csv

- **interactive.(html/js/css)** - The HTML, JavaScript, CSS to render the visualization in Q4.
- **average-rating.csv** - The dataset of game information.

Q5 [25 points] Choropleth Map of Board Game Ratings

Choropleth maps are a very common visualization in which different geographic areas are colored based on the value of a variable for each geographic area. You have most probably seen choropleth maps showing quantities like unemployment rates for each county in the US, or the number of confirmed COVID-19 cases per 10,000 people at the county level.

We will use choropleth maps to examine the popularity of different board games across the world. We have provided two files in the Q5 folder, ***ratings-by-country.csv*** and ***world_countries.json***.

- Each row in *ratings-by-country.csv* represents about a game's information for a country, in the form of `<Game,Country,Number of Users,Average Rating>`, where
  - `Game:` the name of a game, e.g., Catan.

  - `Country`: a country in the world e.g., United States of America.
  - `Number of Users`: the number of users who have rated `Game` who are from `Country`.
  - `Average Rating`: the mean rating given to `Game` by users who are from `Country`.

  This dataset has been preprocessed and filtered to include only those games that have been rated by more than 1000 users in the world.
- The *world_countries.json* file is a TopoJSON topology containing a single geometry collection: *countries*.

a. **[20 points]** Create a choropleth map using the provided data, and use Figure 5a and 5b as references.

1. **[5 points]** Dropdown lists are commonly used on dashboards to enable filtering of data. Create a dropdown list (see example in Figure 5a) to allow users to select which game's data are displayed.
   - The list options should be obtained from the `Game` column of the csv file.
   - Sort the list options in alphabetical order. Set the default display value to the first option.
   - Selecting a different game from the dropdown list should update both the choropleth map (see part 2) and the legend (see part 3) accordingly.

2. **[10 points]** Load the data from *ratings-by-country.csv* and create a choropleth map such that the color of each country in the map corresponds to the **average rating of the game selected in the dropdown** in each country.

   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, and use topojson (present in the lib folder) to draw the map.

   Many countries have no ratings for some games—these should be **colored gray**.

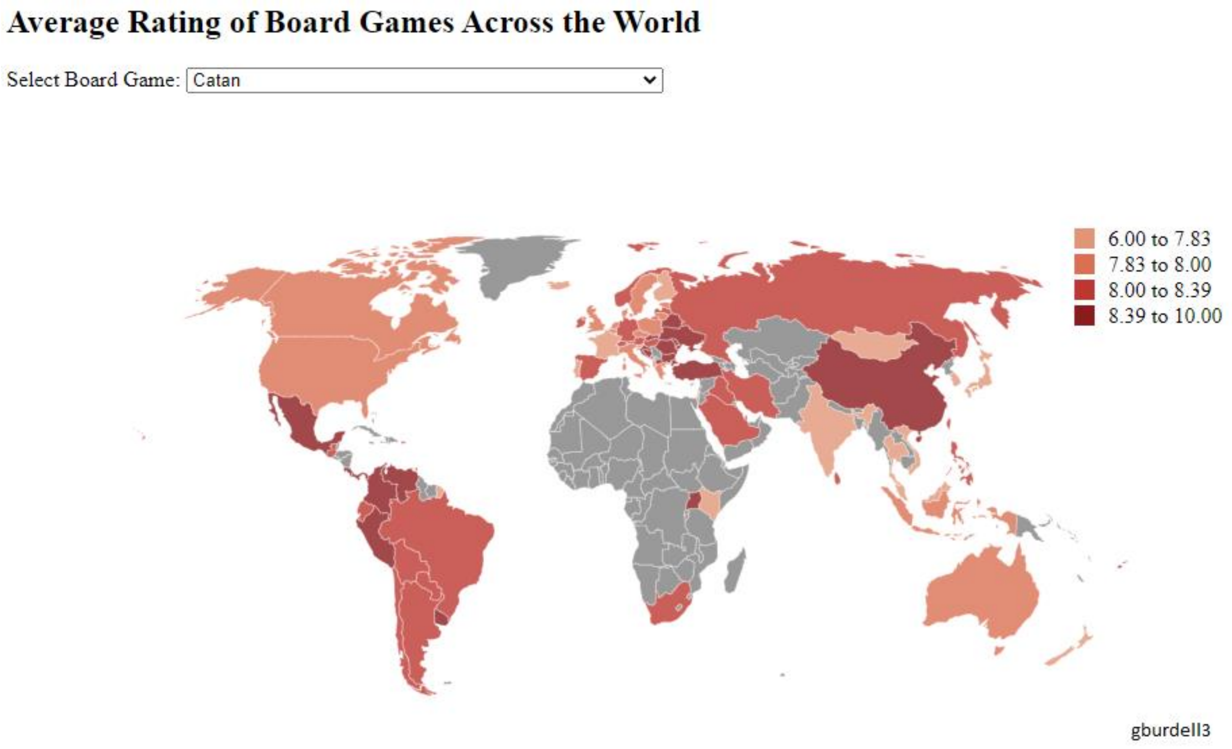
   For those countries that do have ratings for the selected game, use a quantile scale to generate the color scheme based on the average rating by country. Color them along a gradient of exactly 4 gradations from a single hue, with darker colors corresponding to higher rating values and lighter colors correspond to lower values (see gradient examples at Color Brewer).

   About Scaling Colormaps: In order to create effective visualizations that highlight patterns of interest, it is important to carefully think about the relationship between the range and distribution of values being displayed (the domain) and the color scale the values are mapped to (the range). Many types of mapping functions are possible, e.g., we could use a linear mapping where the lowest game rating is mapped to the first value in the color scheme, the highest game rating is mapped to the highest value in the color scheme, and intermediate ratings are mapped to hues in the middle. This article illustrates the value of choosing appropriate endpoints for linear color maps, or log-scaling the domain so that large but relatively infrequent values do not cause differences between smaller values to be washed out. In our case, most board games have similar average ratings across countries, e.g. Catan has an average rating close to 9.3 in almost all countries, making it challenging to perceive relative differences in popularity. To address this, we can compute quantiles of the domain data—game rating values that divide the ordered list of average ratings per country into roughly equally-sized groups. Here, we will get 4 groups, a special case of quantiles called "quartiles" since the data are divided into quarters.

3. **[5 points]** Add a **vertical** legend showing how colors map to the average rating for a particular game. You must use **exactly 4** color gradations in your submission. You could use d3-legend.min.js (in the **lib** folder) to create the legend for the scale you use. Also, **display your GT username** (e.g., gburdell3) beneath the map.

b. **[5 points]** Add a tooltip using the d3-tip.min library (in the **lib** folder). On hovering over a country, the tooltip should show the following information on separate lines: (1) country name, (2) game, (3) number of users form the country who rated the game, and (3) average rating for that game in that country (Figure 5b demonstrates this for Catan). The tooltip should appear when the **mouse** hovers over the country. On mouseout, the tooltip should disappear. Position the tooltip a small 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). If the tooltip becomes clipped you may lose points.

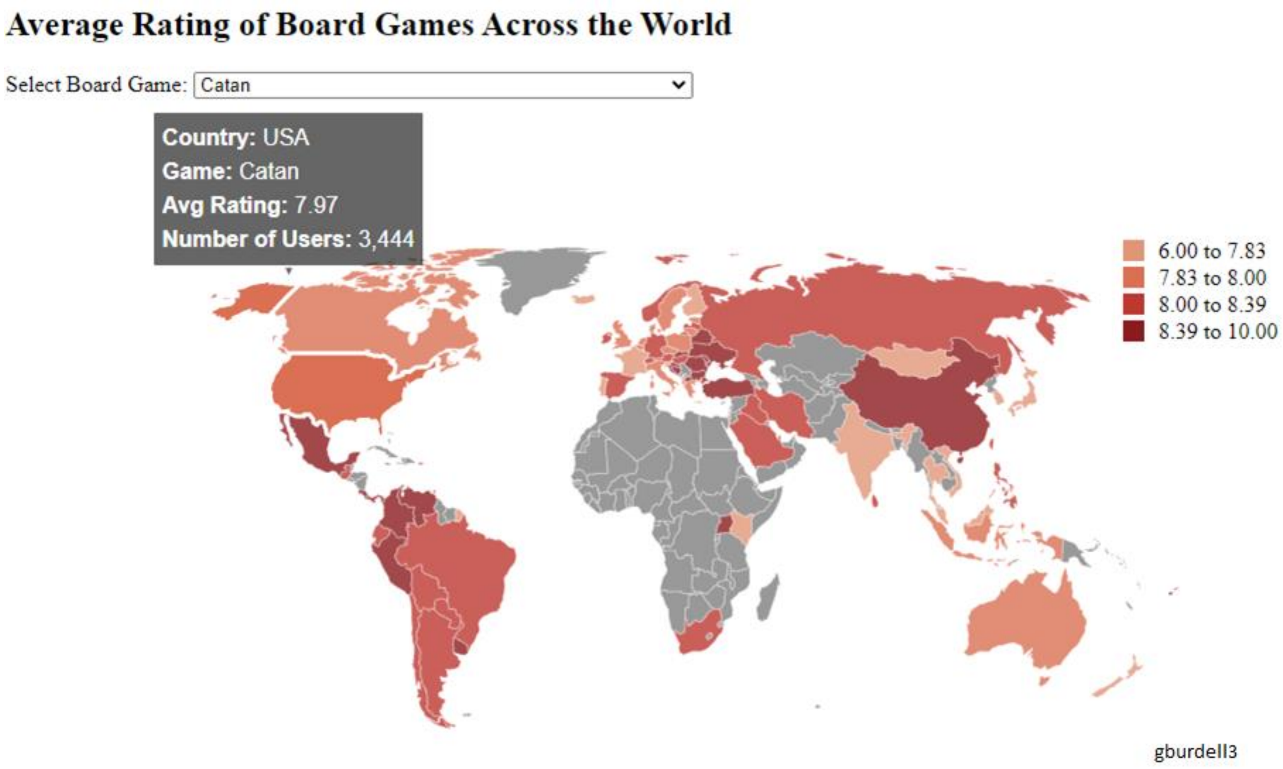
Note: You **must** create the tooltip by **only** using **d3-tip.min.js** in the **lib** folder.



Average Rating of Board Games Across the World

Select Board Game: Catan

Figure 5a: Reference example for Choropleth Map showing average rating of Catan. Your chart may appear different and earn full credit as long as it meets all stated requirements.

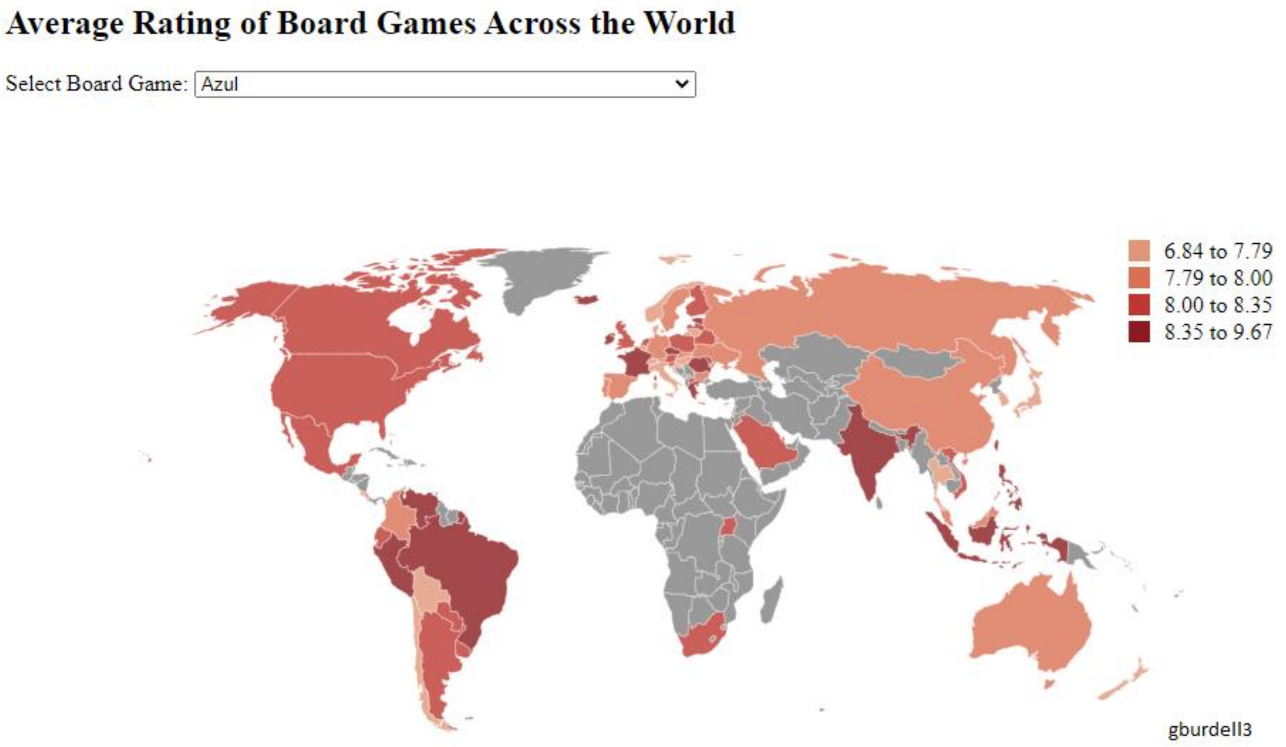


Average Rating of Board Games Across the World

Select Board Game: Catan

Country: USA
Game: Catan
Avg Rating: 7.97
Number of Users: 3,444

Figure 5b: Reference example for Choropleth Map showing tooltip. Your chart may appear different and earn full credit as long as it meets all stated requirements.



Average Rating of Board Games Across the World

Select Board Game: Azul

Figure 5c: Reference example showing updated Choropleth and legend for Azul. Your chart may appear different and earn full credit as long as it meets all stated requirements.

Q5 Deliverables:
The directory structure should be organized as follows:

Q5/
choropleth.(html/js/css)
ratings-by-country.csv
world_countries.json

- **choropleth.(html /js /css)**- The html/js/css file to render the visualization.
- **ratings-by-country.csv** - The datasets used to show the information of each state.
- **world_countries.json** - Dataset needed to draw the map.

Extremely Important: folder structure & content of submission zip file

We understand that some of you may work on this assignment until just prior to the deadline, rushing to submit your work before the submission window closes. **Please take the time** to validate that **all files** are present in your submission and that you have not forgotten to include any deliverables! If a deliverable is not submitted, you will receive **zero** credit for the affected portion of the assignment — this is a very sad way to lose points, since you have already done the work!

You are submitting a single **zip** file **HW2-GTusername.zip** (e.g., HW2-jdoe3.zip).

The files included in each question's folder have been clearly specified at the end of the question's problem description.

The zip file's folder structure must exactly be (when unzipped):

```
HW2-GTUsername/
    lib/
        d3.v5.min.js
        d3-tip.min.js
        d3-geo-projection.v2.min.js
        d3-dsv.min.js
        d3-legend.min.js
        topojson.v2.min.js
    Q1/
        table.png
        grouped_barchart.png
        stacked_barchart_1.png
        stacked_barchart_2.png
    Q2/
        graph.(html / js / css)
        board_games.csv
    Q3/
        linecharts.(html / js / css)
        linecharts.pdf
        boardgame_ratings.csv
        explanation.txt
    Q4/
        interactive.(html / js / css)
        average-rating.csv
    Q5/
        choropleth.(html / js / css)
        ratings-by-country.csv
        world_countries.json
```