Visualizing Movies and Their Statistics

By: Angela Zhang (az337), Aaheli Chattopadhyay(ac923), Saachi Gopal(sg932)

A. A description of the data. Report where you got the data. Describe the variables. If you had to reformat the data or filter it in any way, provide enough details that someone could repeat your results. If you combined multiple datasets, specify how you integrated them. Mention any additional data that you used, such as shape files for maps. Editing is important! You are not required to use every part of the dataset. Selectively choosing a subset can improve usability. Describe any criteria you used for data selection.

The first dataset was obtained from IMDb and consisted of their ranking of the top 250 films of all time. The dataset was condensed from the top 250 to the top 100 movies. The second dataset was obtained from Rotten Tomatoes which was parsed to include the top 100 movies as predetermined from the IMDb dataset. The first dataset was used for our sunburst visualization which randomly generates a movie based on the selection criteria of genre, year, and rating. The second visualization was used for the plot comparing a movie's box office number to its academy award wins. Both the datasets contained several details about a movie including its title, year of release, rating, runtime, genre, director, oscars won, country of release, box office score, production company, plot, and cast list. The second dataset obtained from rotten tomatoes also contained separate scores given by the audience and critics for the movies. Both the datasets contained a comprehensive overview of the movies that were in them which allowed us to utilize as much information as we could to display a movie and compare its success to other movies.

B. A description of the mapping from data to visual elements. Describe the scales you used, such as position, color, or shape. Mention any transformations you performed, such as log scales.

The sunburst used the movies.json file to retrieve its data. The sunburst was created from a json that contained a hierarchy of parent and children nodes, where each child represented a filtering category (genre, year, and rating). The first level of children included the genre, the second level included the year range, and the third level included the movie rating. Each genre had 3 child nodes containing the year ranges, and each year range node had 4 child nodes containing the movie ratings. The genres consisted of animation/comedy, crime, horror/mystery/sci-fi, action/adventure, and drama/biography. The year ranges were 1920-1960, 1961-1990, and 1991-2016. The movie ratings were NR, PG/G,PG-13, and NR. The size of the child nodes were determined by filtering the movie data based on genre, then year range, and then ratings. The result of the filtration returned an array of movies and the function filterMovie returned the length of this array after performing the filtration on the movie data set for the three categories.

In order to display the sunburst, the json consisting of the hierarchical data was converted into a root node using d3.hierarchy(). This root node contained all the parent nodes and its children, where any node's ancestors and descendants could be accessed by using node.ancestors()

and node.descendents() respectively. After creating the root node, an svg path element containing the root node dataset was appended. Each node's path was created by the arc function with defined the start angle, end angle, and width of the arc chord that would be displayed on the sunburst. The color of each arc was generated by the function color which traverses to a node and gives it a color that is darker than its parent. The root node is always white and each of the five genres are represented by five different colors.

The sunburst arcs had event listeners for mouseover, mouseout, and click. The mouseover updated the sunburst by fading all the chords of the sunburst except for the immediate ancestors of the current node. It also updated the text at the center of the sunburst by displaying what categories were selected so far and what was to be selected next. The mouseover displayed the percentage of movies that matched the categories that were selected so far at the bottom of the center circle exactly underneath the last topic selected. Additionally, the function highlighted the immediate children of the current node to aid the user in selecting the next topic. This was done by accumulating the path of the child nodes and unfading its color. Using the same process, the ancestors of the current nodes were also unfaded. The mouseout function faded the entire sunburst except for the genre arcs if a mouse moved outside of it. The function also updated the text in the center of the sunburst to indicate what the user should select first on the chart. The click function first filtered the movie data set based on the arc nodes selected in the graph. After filtering out the movies, the function randomly chose a movie from the filtered array of movies. The randomly selected movie was then displayed by a modal. Before the modal was displayed, a loader was added to the middle of the sunburst to act as a transitioning step from the sunburst to the modal. The loader was set to transition for 1.2 seconds before the modal appeared.

Once a movie was selected, a modal was created to display more detailed information about the selected movie. An overlay was created in the background to fade out the sunburst and draw the user’s attention to the pop-up. The pop-up is centered to draw the user’s attention. The movie poster is displayed on the left and the text information is all grouped together to help create a smoother user flow. The title is at the top with the year displayed beside it in a smaller, opaque font and both are relatively large and bolded to highlight the text information. The rating for the movie is displayed in a circle graph (credits to Anders Ingemann for this code), which has a fill animation to show the progress of the rating. Underneath the rating graph are two text labels indicating what kind of score is being displayed and a prompt to instruct the user to click the rating to switch the rating type. To the right and underneath, additional text information about the movie is displayed, including the genre, rated, runtime, plot summary, cast list, director, production company, and box office, and each section is labeled with bolded, larger-sized.

The plot mapping the movies’ box office scores against the number of Academy Awards won used the box office and oscars information from the plot.json file. The buttons below the plot allowed for filtering of the data points based on the genre categories labelled upon them. This allows for a closer analysis of the plots based on the filtered distribution. The onclick

functionality of the circles displayed the audience and critics' score for the movie through two pie charts. The pie charts were animated by doing an arc sweep from a start angle 0 and the end angle dictated by the movie score. The end angle was determined by linearly scaling the audience and critics' score from a domain of (0,100) to a range of (0,360). Therefore the larger the score the greater the arc sweep of the pie chart. The percentages of the movie scores for the audience and critics were also displayed to the right of the respective pie charts, and the movie title was displayed above both of the charts.

C. The story. What does your visualization tell us? What was surprising about it?

On average, a user of platforms such as Netflix, Amazon Prime, and Hulu wastes up to 1 hour searching for a movie or television show to watch. This project aimed to engineer a solution to that problem by incorporating elements such as audience and critic scores from sites such as IMDb and Rotten Tomatoes as well as industry metrics like Academy Awards in a variety of ways. The first element is a sunburst movie generator that allows users to narrow their choices sequentially by inputting up to three categories (genre, year range, and rating). After selecting a certain number of inputs, a randomized movie fitting those criteria is chosen and a pop-up displaying detailed information about the movie such as plot summary, actors, and more.

When people decide what film to watch, critic reviews as well as audience reviews play a very important role in that decision. As such, we wanted to demonstrate how box office revenue and critical acclaim from film industries such as the Academy affected both the film's audience score and critic score. To do this, we created a graph that plots the top 100 films from IMDb based on each film's box office revenue and how many Academy Awards it won. Buttons beneath the graph allow the user to filter the movies based on genres and see trends regarding a film's success (from both an audience's and critics' perspective) based on its genre. In addition, clicking on the circles in the plot displays pie charts visualizing the film's audience score (from the IMDb user rating) and critic score (from Rotten Tomatoes).

It was surprising to see how films of certain genres fared against others when it came to the popularity with audience and with critics. Drama and Biography films tended to be the most successful in terms of winning Academy Awards while Horror/Mystery/Sci-Fi movies were the least. Furthermore, higher grossing films (more commercially appealing thus lower awards won) were more likely to have audience scores exceeding the critics score. On the other hand, prestige films that may be lower grossing often had critics scores exceeding the audience score. This speaks to a noticeable divide between the perception of the public and those of industry insiders yet both of these metrics play an important role in how people choose their entertainment.