Presentation Video Link: https://youtu.be/4WTNoTVWzNo

1 Introduction

For this final assignment I chose to visualise my year of movies, given that they had such an positive impact on me earlier in the year while Ireland was still in lockdown. The data was gathered using the mobile app Letterboxd, where a review of every movie watched was entered since the start of the year. The aim for this assignment was to investigate how my reviews compared against the majority of Letterboxd users, rank key attributes of movies and identify if there were any factors that might lead me to give a movie a lower or higher score. Furthermore I wanted to determine if any real-world factors had an impact on the scores that I gave to movies.

The primary idiom that I wanted to produce for this assignment was a calendar view of the year, showing the date that a movie was watched, the review that was given alongside the average review, and to superimpose this calendar with some real-world events that might have an impact. Then I also wanted to produce a combination of sub-plots based on the trends of my personal viewing history.

My plan was to develop this visualisation purely using Tableau, but it was quickly discovered that Tableau could not offer me the calendar view that I wanted to produce so it was developed in Processing.js and used Tableau for the rest of the plots and to culminate the plots in a single dashboard.

2 Data Gathering and Processing

The initial gathering of data for this visualisation was very straightforward and required logging in to the desktop version of the Letterboxd application and using the Import/Export feature found in settings. This downloaded a zipped folder containing six CSV files; ‘comments’, ‘profile’, ‘ratings’, ‘reviews’, ‘watched’, ‘watchlist’ and ‘diary’. Of these files, only ‘diary’ was relevant for this visualisation as it contained the name of each movie and the date that it was watched, the year that it was released, the URI to the review on Letterboxd, the review that I gave it out of 5, and a re-watch column that had ‘Yes’ for every movie that was previously watched.

From this CSV file, additional columns were appended that were deemed to be relevant metadata for each movie – runtime, average rating, director, cast member(s) and genre(s), which had to be extracted manually from each movie page on Letterboxd. This CSV file was then used to generate a vector and two scalar values; ‘Difference’, ‘Most positive difference’ and ‘Most negative difference’. ‘Difference’ was calculated by subtracting my score for a movie from the average score, and the two scalar values were generated by selecting the max/min of this vector, respectively. These scalars were used as the reference for the colour that is assigned to movies on the primary idiom of the visualisation. With these extra columns appended, enough data was available to construct the primary idiom of this visualisation – how my reviews compared against the average user on Letterboxd.

For the next set of idioms I wanted to look at the trends in my own viewing history which required a copy of the original CSV to be made and convert it to an Excel file so that it could be used in Tableau. Some success was had with the built in functions in Tableau for grouping columns but it was too time-consuming at times so for one of the visualisations it was decided to use the ‘Advanced filter’ Excel function to extract the unique values out of the three columns which detail the main cast members in a movie, to be stored in a new column called ‘Distinct Cast’. The ‘COUNTIF’ Excel function was then used to calculate the number of times that each cast member appeared in a movie, stored in the ‘NumTimesCastAppear’ column. A similar approach was attempted with the ‘AVERAGIF’ function to determine the mean rating of movies that each actor appears in, but there were errors and inaccuracies in the implementation so the generated figures were not included in the visualisation. This value, along with the formula, is stored in the ‘AvgCastRating’ column.

The final dataset used is shown below in Table 1:

Attribute Name	Data Type	Auto-generated
Watched Date	Discrete Quantitative	Yes
Name	Categorical	Yes
Year	Discrete Quantitative	Yes
Letterboxd URI	Categorical	Yes
Rating	Discrete Quantitative	Yes
Rewatch	Categorical	Yes
Runtime	Discrete Quantitative	No
Average Rating	Discrete Quantitative	No
Director	Categorical	No
Cast 1 / Cast 2 / Cast 3	Categorical	No
Genre 1 / Genre 2	Categorical	No
Difference	Discrete Quantitative	Yes
Most + diff / Most – diff	Discrete Quantitative	Yes
Distinct Cast	Categorical	Yes
NumTimesCastAppear	Discrete Quantitative	Yes
AvgCastRating	Continuous Quantitative	Yes

Table 1 - Dataset with additional extracted and Excel-generated columns

3 Implementation

3.1 Primary Idiom

The main idiom of this visualisation was developed purely in Processing.js. A circle is drawn and first divided into 4 quarters and then sub-divided into 12 slices, to represent the seasons and months of the year. 5 rings are drawn outside this circle to be used as references for the number of stars that any review has. A line is drawn out from the date that a movie is watched, with a length proportional to the number of stars that I gave it. The function to draw a star was found online [1].In the event that two movies were watched on the same date (2 occurrences), an additional 5 arcs were drawn outside the original 5 rings for that month, to view clearly the review for the second movie which occurred on the same day. The average review that any movie has is encoded using position by a small blue dot at that date and at the corresponding ring position for the number of stars. The number of stars at I gave that movie is encoded by the length of a line up to the ring position, and the colour of this line is either blue, or on some spectrum of red or green, depending on the polarity of the difference between my review and the average review. Therefore movies where I gave a much higher review than the average user appear as a deep shade of green. The sum of the differences between my reviews and the average user are then displayed above each month name.

It was determined that position was the only suitable encoding channel to use for dates, especially when some form of calendar view was desired. Using position along a single axis meant that it could be used again (perpendicularly) to encode the review scores that a movie gets, along with the size of the line originating at the circle perimeter. Colour and brightness were determined as the only two remaining suitable channels to encode the difference between my review and the average. The choice of colour depends on whether this difference was positive, negative or zero, and brightness is used to encode the magnitude of the difference.

Real world events were superimposed on top of this calendar, such as exam period, time that I was away on summer holiday, or easing of COVID restrictions, to explain the gap in movies during Summer and Autumn, but found that it made the view too cluttered so they were omitted from the final output.

This idiom allows the viewer to compare my reviews to the average review for all of the movies that were watched this year, and identify movies where the difference between the two figures is particularly large. It also allows the viewer to quickly identify movies that received a review at the high/low extremes, by examining the size of the lines. Lastly it allows the user to examine the spread of data across the year and identify peaks/troughs of activity.

The strength of this idiom is that it uses the chosen encoding channels effectively and allows the user to see a clear distribution of reviews across the year. The choice of colours is appropriate although it would be ideal to use a 3-colour scale for the colour of each line, ranging from red to blue to green. This is because most of the reviews don’t deviate much from the average review and therefore appear as a feint green/red when it would be more appropriate to be some shade of blue mixed with green/red. There is also some clustering of movie names at times, but this was unavoidable as any smaller text was illegible. A blog post was found online which allowed a high resolution PDF of the output to be downloaded [2], but when this is converted to a JPG or some other image type it loses some of its quality.

3.2 Secondary Idioms

Tableau was then used to generate a number of secondary idioms to try to summarise appropriate metrics of the data. These were mainly bar charts which allow the viewer to rank attributes that appeared across multiple movies throughout the year, for example actors or genres. The position and size were optimal for use in these bar charts, although effective use of colour was ideal so that plots could be easily distinguishable from each other. For the 3 leftmost bar charts, position is used to encode individual names and size is used to encode the number of occurrences, which was kept consistent for clarity. The last bar chart uses position to encode the bins of release dates, and size is used to encode the average review for movies released within that date range. This is intentionally similar to the stacked bar chart in the visualisation for consistency. The stacked bar chart also uses the colour channel to encode genres appearing in particular movies. To group together genres the pivot functionality in Tableau was used, but this meant that movies with only one genre are displayed with ‘Null’ as a second genre.

In the rightmost plot each of the movies are displayed along with their runtimes and reviews (where reviews are encoded using a two-tone colour scale), to determine if there was a correlation between runtime and review, but it was found that there was no way to sort/organise a Gantt chart in Tableau by some metric so it is difficult to see if there is a clear relationship. The font size also cannot be reduced further than 8, meaning that some boxes have no meaning until the detail tooltip is used.

These visualisations allow the viewer to identify common values (genres, runtime, release date etc.) across the year and to determine whether there is a relationship between these values and the score that a movie will get. Additional plots such as Average Rating of / Top Rated Actors/Directors/Genres, most common genres in each month of the year etc, or distribution of reviews, could have been added but the aim was to have the entire visualisation fit on one screen.

As mentioned above in Section 2, the average score of each individual actor was attempted to be gathered but trouble was encountered with the formula both in Excel and Tableau, which stopped plans of displaying the average score across each of the bar charts on the left hand side as a text label, resulting in some unused space along the bar chart.

Another disadvantage with these secondary idioms is that a lot of data points share similar attributes but are not included due to size constraints, for example in the stacked bar chart a lot of movies had a 4.5 star review, but the data was first ordered by score and then alphabetically, where the filter chose the top 10 of this list.

Although the data was not particularly complex, the primary visualisation developed in Processing.js is sufficiently novel and successful in its purpose of comparing review scores and visualising the spread of data. Combining this with the basic plots generated in Tableau gives a good summary of the year of movies, which was the aim of this visualisation. I also think that the visualisation as a whole is easily interpretable and visually appealing due to the placement of plots and colour scheme.

References

[1] Star. [online] Available at: <https://processing.org/examples/star.html>.

[2] Processing 2.0 Forum. 2021. Processing 2.x and 3.x Forum. [online] Available at: <https://forum.processing.org/two/discussion/24646/how-to-save-a-sketch-as-a-high-res-image.html>.