**CS4460 P5**
**Team Members: Mehnaz Ruksana & Nouzhan Vakili Dastjerd**
**Data Set: Movies.csv**

---

**Overview of the design:**

As avid fans of cinema, we were immediately drawn to the movies dataset. Upon browsing it, we saw the clear possibility of a vis centered around movie popularity. Yet with further inspection, we saw that there were plenty of holes in the data. Almost all the movie titles had bizarre characters, and around ⅓ of the data cases had blanks instead of values for many of their attributes. To remedy these problems, we parsed the data with Excel and both removed the weird characters and filled in 0s for all the blank data attributes. This allowed for a much easier time handling the data in d3, though we still needed to perform calculations to derive values that we needed to populate some of the charts and graphs in our vis -- we will discuss how those were handled throughout the document.

In our visualization, we chose to focus on the relationships between a movie's popularity and its received success. This is demonstrated through the multiple graphs we have implemented that are all accompanied by our (novice attempt) at a scrolly-telling experience. Some questions we had included: Are movies with more popular cast members rated higher than those with lesser popular actors? Do people tend to like movies that are longer or shorter in duration? What is the average IMDb score among all the data points. Which genre of movie tends to receive the most amount of public approval? Has this number changed for the genres over time (as in, have certain genres gotten more popular over time?) We created these four visualizations in order to answer these questions.

The first visualization you see is a line chart that shows the change in popularity over time of different movie genres like comedy, action, and drama. Using the selector, the user is able to choose a specific genre they want to see the line for while dimming the others. Below that, there is a set of two linked scatter plots that show the relationship between IMDb scores with both film duration and cast popularity. These two linked graphs have a brush implemented that allows one to highlight certain points on one graph and see the corresponding points on the other. Additionally, there is a bar chart that shows the frequency of IMDb scores for all the movies. The IMDb values were rounded using JavaScript's built in Math.round() function.

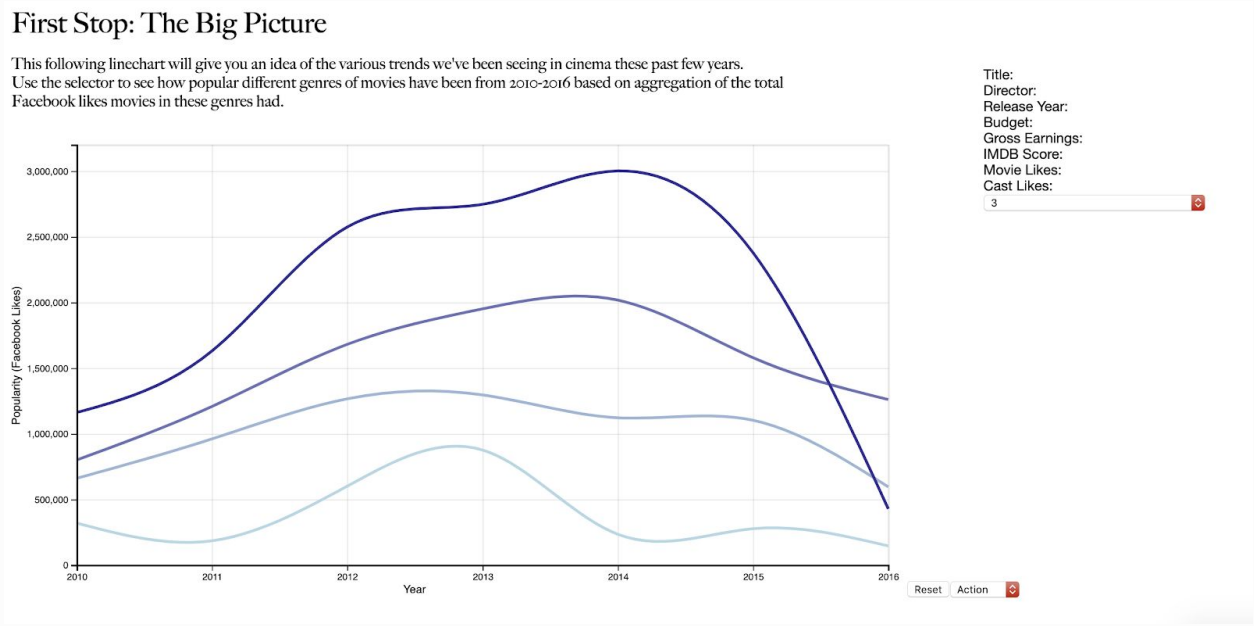
In terms of visual design, we opted for a monochrome color scheme with purple at its center, since it is easier to detect changes in the same color, rather than changes amongst different colors. The color denotes the IMDb score — the darker the color, the higher the score.

We also chose to utilize size of a circle to indicate the amount of Facebook likes a movie has; the larger a point's diameter, the more Facebook likes the associated movie has. This allows users to see three data attributes in a visualization that is usually used for only two, increasing the amount of information one draws from a single vis.

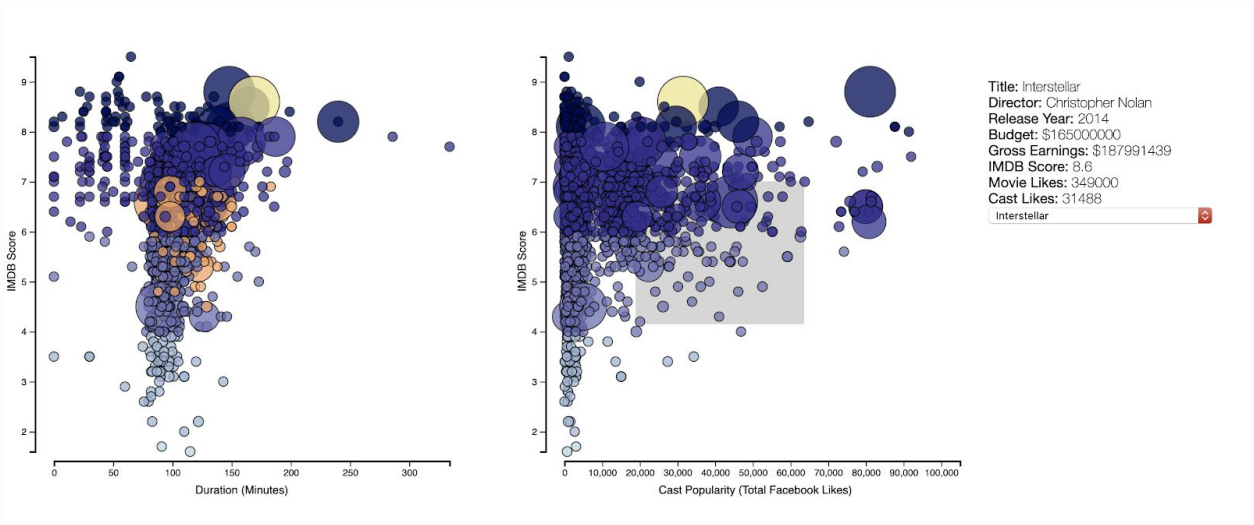
**Users tasks/taxonomies our visualization supports:**

- **Find extrema** — Using the linked graphs, one is able to highlight the specific move we find to be interesting; the extremum values are easily identifiable.
- **Correlate** — The linked graphs show correlation between multiple attributes such as IMDb rating, movie FB likes, cast FB likes, and movie duration.
- **Filter** — Using the line chart, one is able to filter the genre displayed to see that genre's change in popularity over time. Another filter is applied when brushing over certain points in the linked graphs.
- **Details on Demand** — When clicking on a point in the linked graphs, one is able to see all the specific info about that data case. One is also able to choose a specific movie from the drop down menu and see its details that way.
- **Compute Derived Value** — The bar chart acts as a computation of the total number of movies that have a rounded IMDb score of 1-10. The line chart also does a sum of all the likes of a certain genre for a specific year.
- **Overview first** — our first graph shows a nice overview of general trends seen in each genre, and the linked graphs show the relationships between three attributes. Users are then allowed to click on a point or choose a movie from a drop down list to get its details.
- **Search vs Browse** — One can browse by simply looking at our graphs and exploring the trends they see through the interactive line graph and brushes, but they can also search for a specific film by selecting it from the drop down menu.

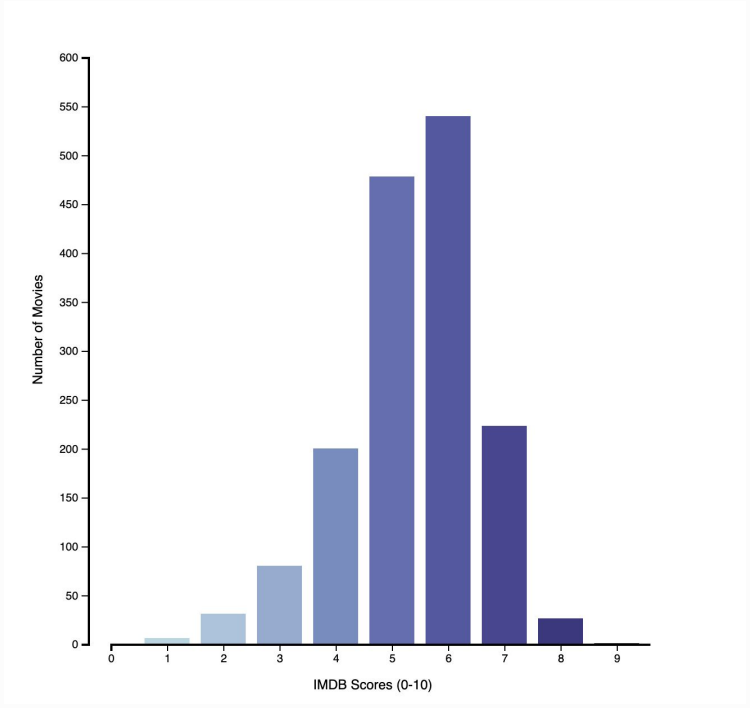
## Screenshots & Further Information:




Above you can see the overview graph and the detail window to the right. Note, the detail window is fixed and scrolls with the user, allowing for direct access to it at all times. One can choose the genre they are interested in seeing highlighted from the selector on the bottom right, and they can hover over the desired line to get more information about it. You may have noticed that these data points were not provided in the movie dataset csv. We computed these values using python (the file is provided in our submission) and exported it to a csv, which we then used in JavaScript with d3 to create this chart. This all goes to show the immense amount of work that can go into a vis.



Above are the linked graphs. The graphs are linked with a brush that allows one to select a certain group of elements and see where they correspond on the other graph. While the brushing feature is somewhat redundant, it is still useful considering how densely populated the two charts are. Also shown is the “details on demand” feature, that allows one to click a certain point and get specific information on it. The yellow dot signifies a point has been clicked, and its corresponding dot is highlighted on the opposite chart. Note that both these graphs are very densely populated. While some may say this hinders interaction, we counter that claim by asserting that the point of these graphs is to observe the *overall trends* of the data, not to specifically monitor exact points. While we have allowed the ability to observe specific points, this capability is not the entire point.




Lastly, we have this simple bar chart that shows a final summary of the distribution of rounded IMDb scores. It fits nicely with the “story” of our scrolly-telling vis as a closer. Again, these values were not provided in the dataset but were instead computed using Python. However, this time we computed them and saved their values in a Python dictionary object, and translated its values and keys to a JavaScript object.

**Closing:**

We hope our vis was interesting! It was rather difficult for us to make, but the creativity it took not only to design our overall visualization, but also the code that made it functional was surprising and made it a worthwhile endeavor! Thanks for a great semester.