CS 3300

Top 250 IMDB Movies
Data Visualization

By Kevin Lin, Chris Huber, and Marcus Boeck-Chenevier

Introduction

Our visualization allows the user to explore the Internet Movie Database's (IMDb) 250 top ranked movies. Our goal is for users to interact with the visualization and understand economic trends affecting films and filmmaking in the past century. As a focal point for the project, we were particularly interested in the "yield" of each movie, which we define as the ratio of its gross to its budget. Another key trend we wanted to highlight was how the movie industry has developed economically over time.

Data Collection

Scraping IMDb's webpages

Our visualization allows the user to explore the Internet Movie Database's (IMDb) 250 top ranked movies. The characteristics about each movie were scraped directly from webpages on imDb.com using a Python script Kevin wrote. In addition to base Python, Kevin used an API called imdbPY. The script loops through each movie ID in the IMDb list to find the corresponding urls to the movies. Next, using BeautifulSoup in python, it collects the html page for each movie and scrapes the desired elements to a pandas datatable.
From this, we obtained a csv file which has the following fields: ID, Title, Director, Rating, Year, Summary, Runtime, Image [of the poster], Genre, and Budget. Most of the fields are self-explanatory, but "Rating" refers to a rating, out of 10, for a movie generated by users with IMDb accounts. The "Summary" is a brief 1-2 sentence description of the movie's plot.

Filtering the scraped data

Each field was initially a string. In our mutateRow function, we converted the Rating, Rank, and Year fields to numbers. We also converted the Date field to a proper Date object. Any of the monetary amounts, such as gross and budget, were filtered with regular expressions and converted to raw numbers in new fields. IMDb's information was not perfect. For some movies, their gross and/or budget was unknown. So, we decided to filter out any of the movies which had an unknown value in either of these fields. This decision shrunk our original data set from 250 movies to 210 movies, a small enough change that we were comfortable making it.

Adjusting for Inflation

Our second data set is a table of average ticket prices each year from http://www.boxofficemojo.com/about/adjuster.htm. For this dataset, we had to clean the price column with regular expressions, and then converted the price and Year columns to Number types.. We used this dataset to contextualize our inflation adjustment calculation to allow users to better compare movie better across time. We also used this dataset to estimate the number of tickets each movie sold. We performed the inflation calculations in our mutateRow function, creating the new fields "Adjusted [Revenue]" and "Tickets". We estimated the number of tickets by dividing the unadjusted gross by the price of tickets in the year the movie was released. We performed the adjustment for inflation using an estimated rate of 3.2% per year applied to the domestic gross.

Mappings and Visuals

The visualization is designed to for today's high-resolution screens. To avoid scrolling, simply zoom out in your browser.

To allow users to explore the dataset from multiple perspectives, we included a series of buttons in the top right of our visualization to allow users to pick the independent and dependent variables. On the x-axis, the user can choose between sorting the movies by chronological release date, or by ranking according to IMDb. The release dates option employs a time scale mapping the time period 1900-2018 to the x-axis. The rank option is a simple linear scale plotting the ranks 0-250 in ascending order. For the y-axis (dependent

variable), the user can choose between unadjusted domestic gross, which we labelled "revenue", and adjusted domestic gross, labelled "adjusted" [for inflation, described in the "Data Collection" section]. Each of these variables used linear scales that mapped the monetary amounts to pixels in the y-space.

Each bar, then, corresponds to the current y-axis variable for each of the 210 movies we plotted. We chose vertical bars for our visualization because in terms of visual perception, they are the easiest visual element for the viewer to make quantitative comparisons with. As one of the focal points of our visualization is to identify how the financials of the film industry have evolved over time, bars were a clear choice for our most prominent visual element. We also wanted to focus on the "yield" of films, the gross:budget ratio. To represent this on our visualization, we used a two tone color gradient. A red color indicates a "negative" yield, meaning the the budget was larger than the gross. This color transitions to a light red, then a light green, where the gross becomes larger than the budget. Finally, a deep green color indicates a very impressive yield. We used red to indicate a loss and green to indicate a gain because users will most likely already psychologically associate "green=favorable" and "red=unfavorable".

An information box at the top dynamically updates based on which bar the user mouses over. It contains text which displays the stats and information about each individual film. The movie poster also updates on mouseover. A legend is presented in the the top left of the plot to orient the user on the meaning of the color scale. The buttons in the top left turn green when selected, providing the user with feedback on the visualization's current state.

Several of the films were released within a very short span of other films. Additionally, the data includes films from the '20s and 2000's, so it covers a very wide timespan. We wanted to create a large graphic so users could easily see the overarching, temporal/rank-based trends in the corpus of films. However, we also wanted users to be able to zoom in and look at each individual film in the context of its neighbors. To this end, we employed a d3 plugin called fisheye. It works by distorting the area around the mouse like a magnifying glass--expanding the space between the "in-focus" bars, will shrinking the space between the "out-of-focus" bars. However, unlike a magnifying glass, we employed a Cartesian fisheye *only in the horizontal direction*, as opposed a radial distortion. This allows a user to

easily zoom and pan through closely clustered movies., while still preserving overall trends. The lack of vertical distortion is key, affording users the ability to compare bars regardless if they are in focus or not. To further aid the viewer, a bar hovered by the mouse cursor will display a light blue stroke.

Discussion

Our visualization highlights several interesting films and trends. Films that grossed the most (unadjusted) included *The Dark Knight, Toy Story 3, Jurassic Park, and Harry Potter 7 Part 2*-- all modern films. This implicates inflation as a likely culprit. When viewed in terms of adjusted revenue, *Gone with the Wind* is far and away the top grosser, followed by *Star Wars: A New Hope, the Lion King, and Jurassic Park.* One reason *Gone with The Wind* is so head-and-shoulders above the rest of the films in revenue is that it has benefitted from multiple theatrical re-releases. There are also several movies which have very impressive yields while not grossing an astronomical amount. These “sleepers” include now-classics like *Psycho, Rocky, and Jaws.* In 1976 *Rocky* grossed $225,000,000 on a $960,000 budget. They multiplied their initial investment by 225, an incredible yield.

An interesting temporal trend we found was the movies we analyzed tended to have much higher yields before the 1980’s than after it. A viewer can see this by noting the chronologically earlier bars are more likely to be a darker shade of green. Why this shift? Are movies becoming less profitable? Our personal hypothesis is the contrary. The data we obtained has an important caveat-- it is *domestic* gross data. It does not take into account international ticket sales. So while movies made in the earlier part of the 20th century seem to have higher yields, they actually only have higher *domestic yields.* We predict that if international sales were taken into account, the post 1980 movies would fare much better in comparison. Still, the trend of declining domestic yield is a very intriguing result. It reveals to the user to continuing globalization of the film industry, and the increasing reliance on global markets such as China for tickets sales and profit margins.

For further directions, it would be interesting to explore which movies do the best globally. Specifically, we wonder, in the wake of the #MeToo and #OscarsSoWhite movements, does Hollywood have an additional financial impetus to make their casts and production crews

more diverse? Do movies that feature actors from traditional minorities perform or are rated higher overseas?

Sources

Data:
http://www.boxofficemojo.com/about/adjuster.htm
http://www.imdb.com/chart/top
Web Scraping Reference:
https://medium.freecodecamp.org/how-to-scrape-websites-with-python-and-beautifulsoup-5946935d93fe

Buttons Guide:

https://bl.ocks.org/eesur/bbcd0543cd284ba3c116

Legend Guide:

https://www.visualcinnamon.com/2016/05/smooth-color-legend-d3-svg-gradient.html

Appendix: Design Evolution

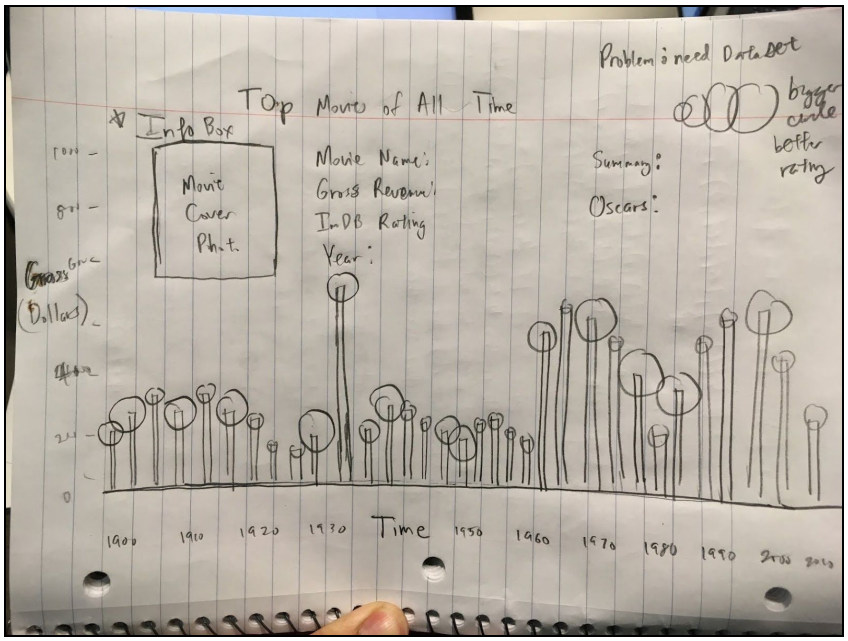

Figure 1: Initial Paper Prototype

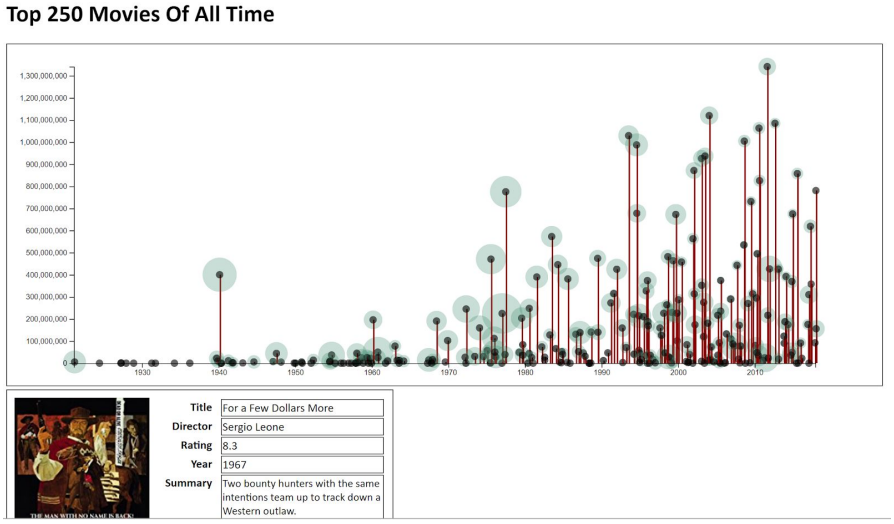

Figure 2: First iteration of informational box and graph. Minimal styling.

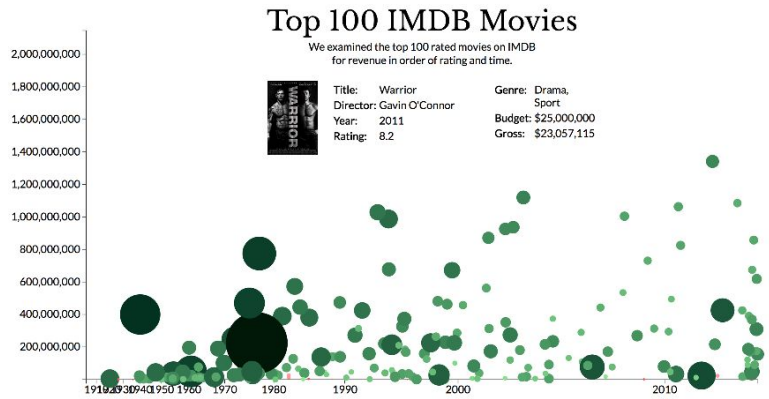

Figure 3: Updated styling, fonts, and color scale. Moved the information box position to be visible as the user interacts with the plot. Note here we experimented with circles as visual elements: cool looking, but less clear in our opinion.

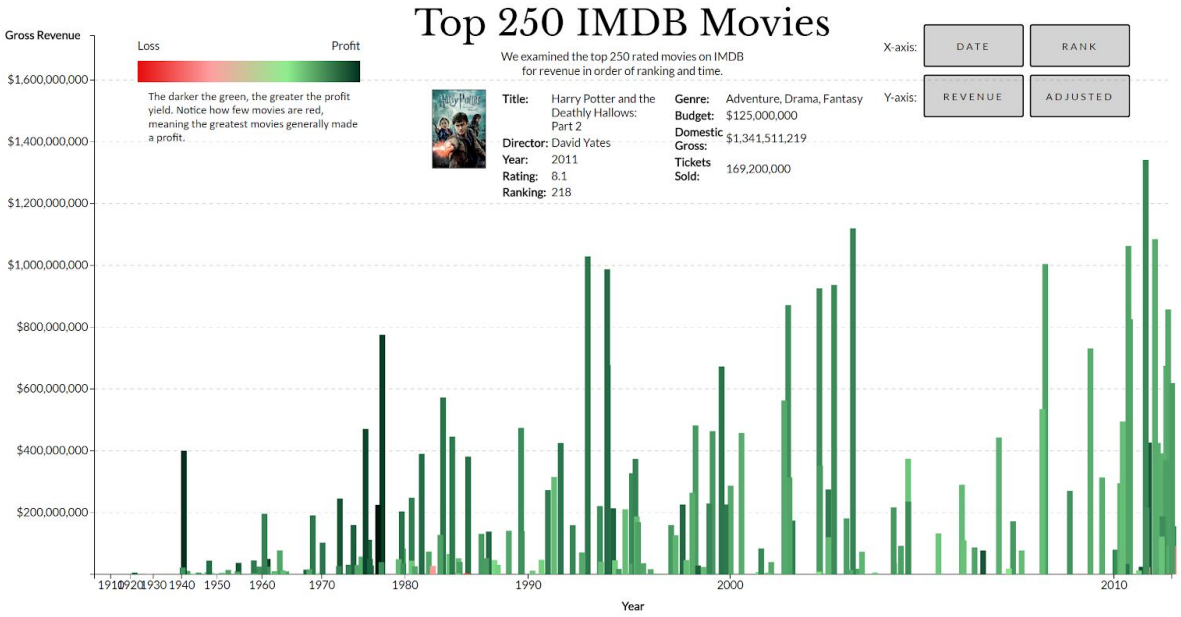

Figure 4: The final visualization. Note we decided on bars, made the visualization larger, and added the legend. We also implemented the buttons which allow the user to choose axes. The fisheye functionality is also complete in this iteration.