Dylan Fontana
Kenedi Heather
Aura Velarde

Visualization Proposal

Process Version I

Initially, we were set on doing our data visualization on data about deep sea coral, with data taken from NOAA. This data was time-series data about cora observations taken from around the globe.

How we chose our dataset:

1. Explored data sources to see what might be available (get ideas generating)
    a. Sources
        i. Kaggle
        ii. Google
2. List of themes and interests of the group, what could we do? Filter down data sources to those fitting our themes.
    a. Nature, conservation, preservation
    b. Nautical, seafaring, navigation
3. Consider dimensionality - what potential does each data set have? What can't we do?
    a. Time
    b. Relationships
    c. Filters
    d. Groupings
4. What questions does the data or topic explicitly pose?

Questions Posed by Data:

1. Evolution of coral population by locality (& by species)
2. How deep is deep sea coral?
    a. Depth by locality
    b. Depth by species
3. How well is deep sea coral being tracked? (observations/time)
4. The deep sea cora
    a. Family
    b. Species
    c. Family (tree)
5. How does temperature affect coral
    a. Temp effect on coral population
    b. Do different species of coral like different temps

c. Temperature change by decade with change in coral by decade
d. Temperature by depth

How can these questions be visualized?

1. Map / Time Map
2. Map / Depth
3. Bar chart
4. Genome tree

We started the five design sheet, and during our 12th draft, we found that we were over stretching the data because we wanted the nautical theme. We didn't have what we thought we had. Although we lost all the work we had done since then, we scratched everything and started looking for new data again. This time focusing on the question we wanted to ask, rather than the visual and types of graphs we wanted to do. We used the same process to find our data.

Process Version II

Data Set.

The dataset we found is "The Movies" dataset provided by Kaggle. This dataset is an aggregate of Movies throughout the past century, with their cast, crews, plots, keywords, and more. Attached are also specific user ratings - which can be used in machine learning to predict movies users might like - but for our purposes we won't be needing those specifics. Overall, this means we have 45,000+ movies with about 26 dimensions after joining Keywords and Cast/Crew to the main metadata.

Project Objectives. Provide the primary questions you are trying to answer with your visualization. What would you like to learn and accomplish? List the benefits.

Our primary questions aim to explore the themes of each binned decade, which are determined through the popularity of keywords for a given bin. These keywords also serve as a mean to establish relationship between movies and who promotes these themes most. These we aim to answer either explicitly or through implicit exploration of the data:

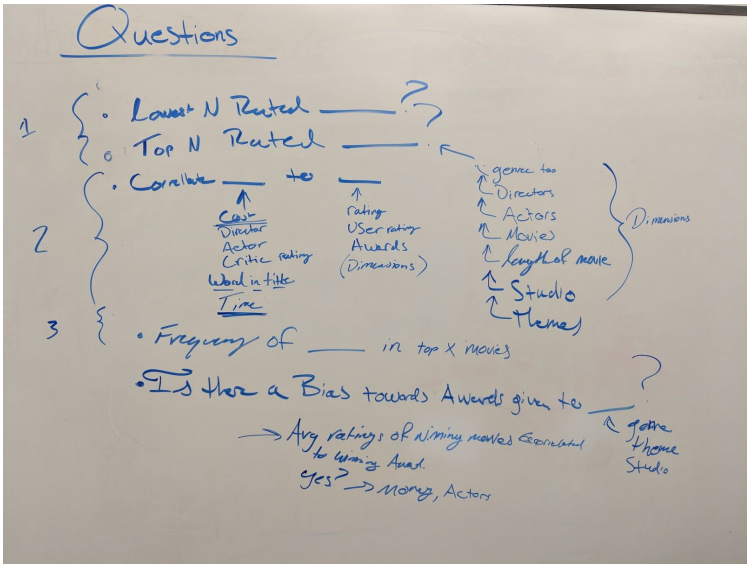
- How do keywords relate movies and actors? What actors/studios use the most frequent keywords? Are the most popular movies the ones who set these terms or do multiple?
- What are the most popular themes (defined by top 3 keywords) by decade over the past 100 years of cinema? (Note: This list will be what we call our "Top Themes").

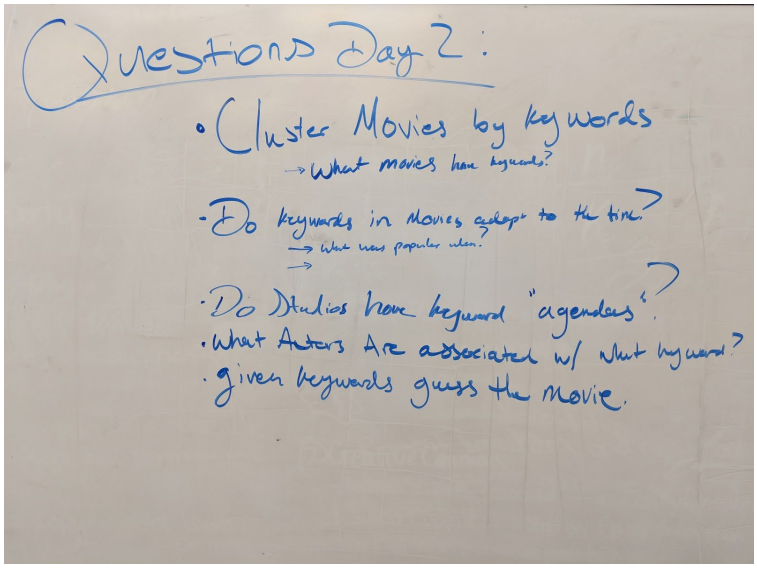
The next questions then look to evolve that fundamental understanding into higher level thought. These questions are what the end user should start to walk away with, with our visualization serving as a means to promote thought in them:

- What actors, movies, and studios align themselves with what keywords?
- What studios/actors are strongest in which top themes? How successful have those themes made them?
- How have a decade's themes evolved? How did they start and where are they now?

Through answering these questions we hope to learn of what themes we (as humans) are drawn most to. We want to understand do those themes change, and who has pushed those themes the most through time. Through doing this we can gain insight towards the sociological impacts these themes have upon ourselves, as well as understanding how influential specific entities are on our culture. For example, if we discovered the keyword "superhero" was a big theme in 1970, then we could explore when that theme started (by what movie?) and watch its rise each decade. We can then learn what actors and studio have had an influence on that keyword the most as time progressed, leading to its popularity. Thus, as a tangible outcome, we've now gained some sociological insight as to how "things are today" in the realm of cinema. Of course, alongside the sociological benefits, there's the "DataVis for fun" benefit where the common person can gain insight on a topic they can relate to while being exposed to the potential DataVis carries.

Background and Motivation





There's few mediums existent in society that reflects our cultures and ideals such as Cinema. Decades of "movie magic" and cinematography have brought to life some of humanity's deepest thoughts and funniest aspirations. What's more, movies are evolving from a

Big Screen treat to an on demand fix for boredom through streaming services like Netflix and Hulu. With the increasing availability of movies, and their rich cultural history, we beg to ask - are movies truly a reflection of us, of our desires? And what impact has time had on telling that story?

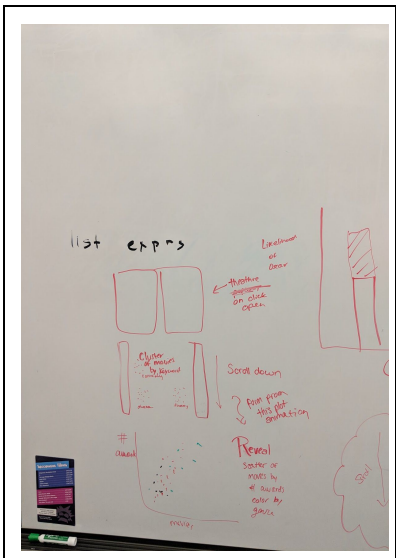
For our final project, we've set our sights on exploring the relationship between the keywords that describe a movie, the actors who brought them alive, and the studios that produced them. Using this data, we intend to learn more about what themes successful movies follow, what actors bring those themes most to life, and what studios are just the best at working those themes to their advantage. We'll aim at seeing how this changes with time - if perhaps movies reflect current day events or if there are simply common themes that work better than others. To do this, we'll need to have some working definitions; for example - "success" should be a measure of revenue and user rating. "Best" should follow a similar function, albeit perhaps also inspecting frequency of the keyword. The resulting outcome, then, should be able to answer: does purely having a keyword or theme that is descriptive for that time-period guarantee that it will bring attention? Who are the major influences of those keywords?

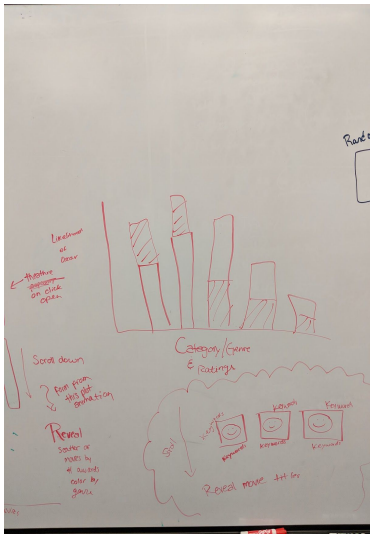
Motivation for this theme came two fold. After our first dataset failed to follow through (see our process book), we learned we needed more than just a question to answer but a means to answer it. This changed the way we sought our topic. Our fault came from looking for a dataset that reflected a general theme we wanted to pursue, rather than asking ourselves if there was enough sustenance to really approach the topic. When we realized we couldn't provide the missing sustenance, we needed to start over. That's when we shifted gears and took another perspective, browsing datasets and thinking of what topics might be interesting to us - but this time when filtering we carefully asked does this data carry meaning? Can we fill in what its missing? What potential does it hold? That is where the Movies topic shined. The topic is relatable, engaging, and integrated deeply with our culture - and data is boundless on it. We saw a lot of potential in what could be said, and when we dug deeper we realized there was something worth exploring to the high-dimensionality of this dataset. Instead of trying to find questions to ask about the data, the data was presenting questions to answer.

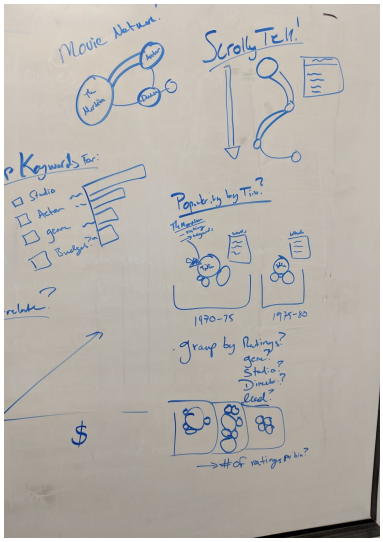
The Five Sheet Visualization Design

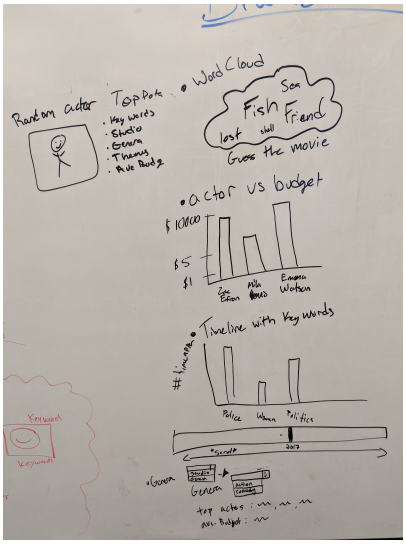
Five Design Sheets: Brainstorm

1. Generate Ideas.

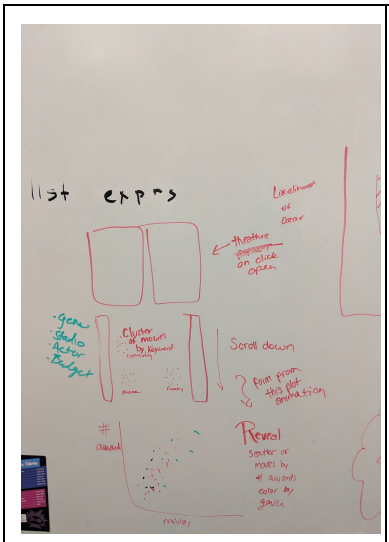


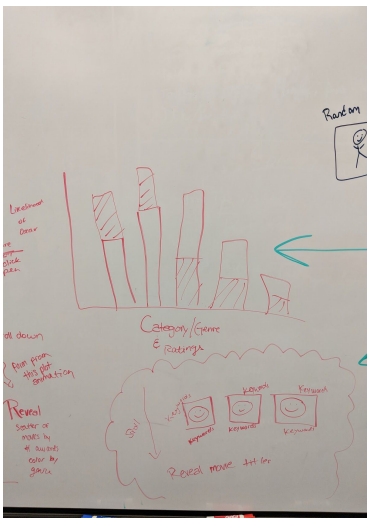


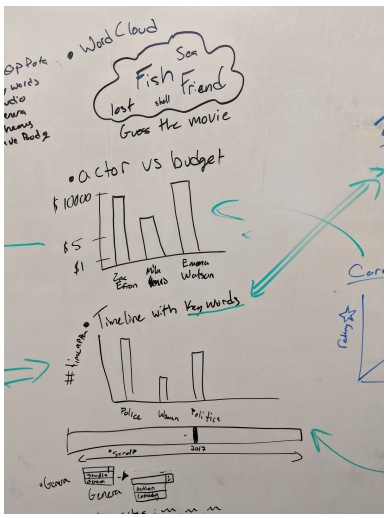


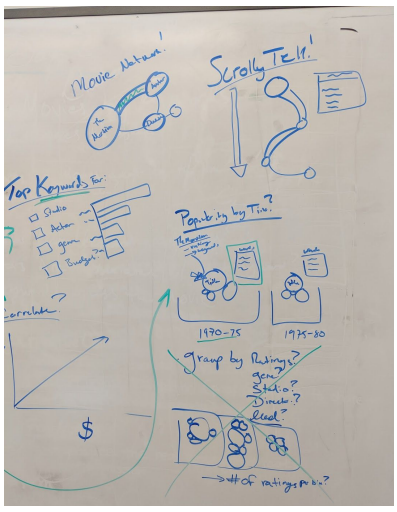


2. Filter Ideas.

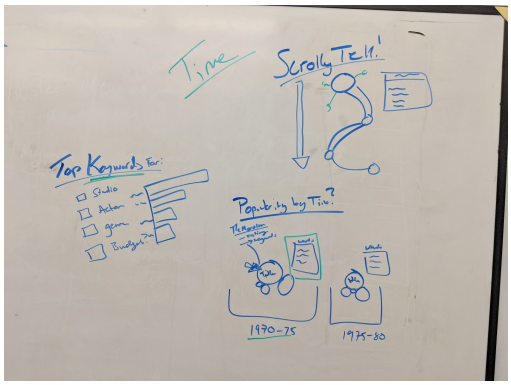


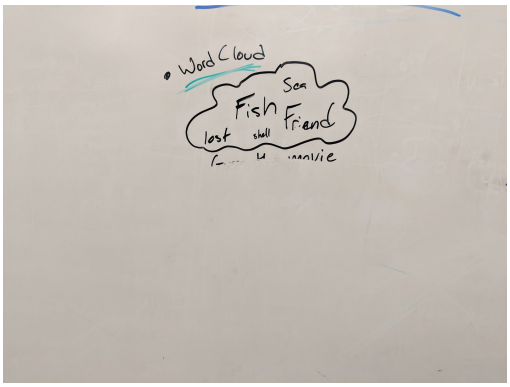


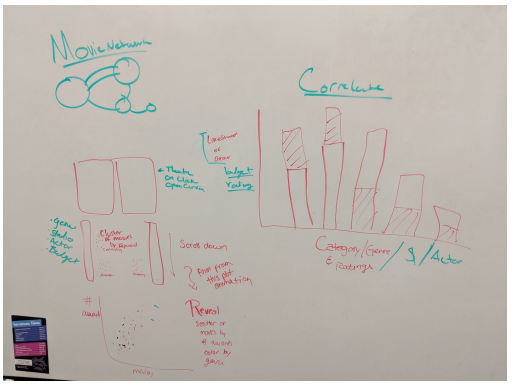




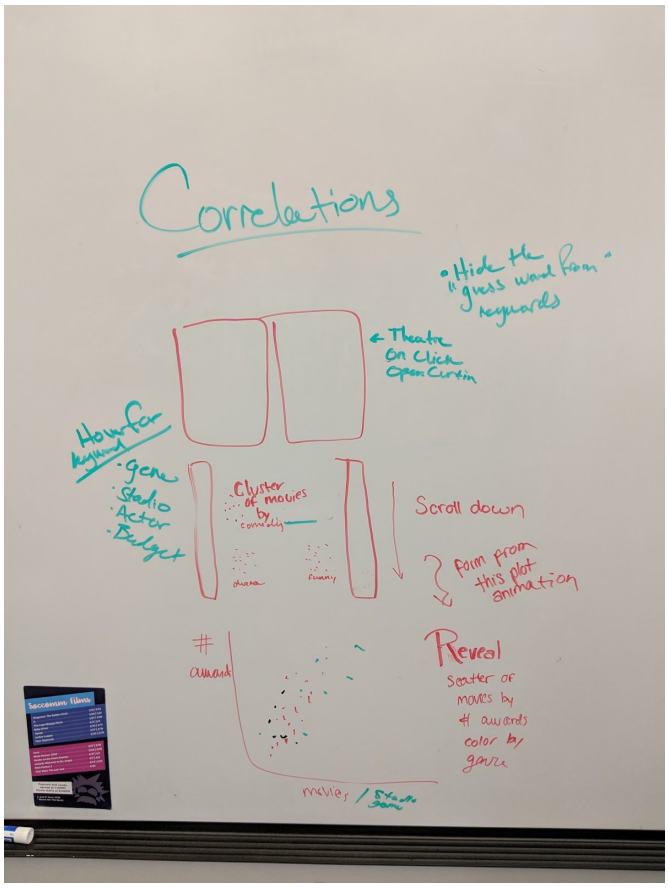
3. Categorize.

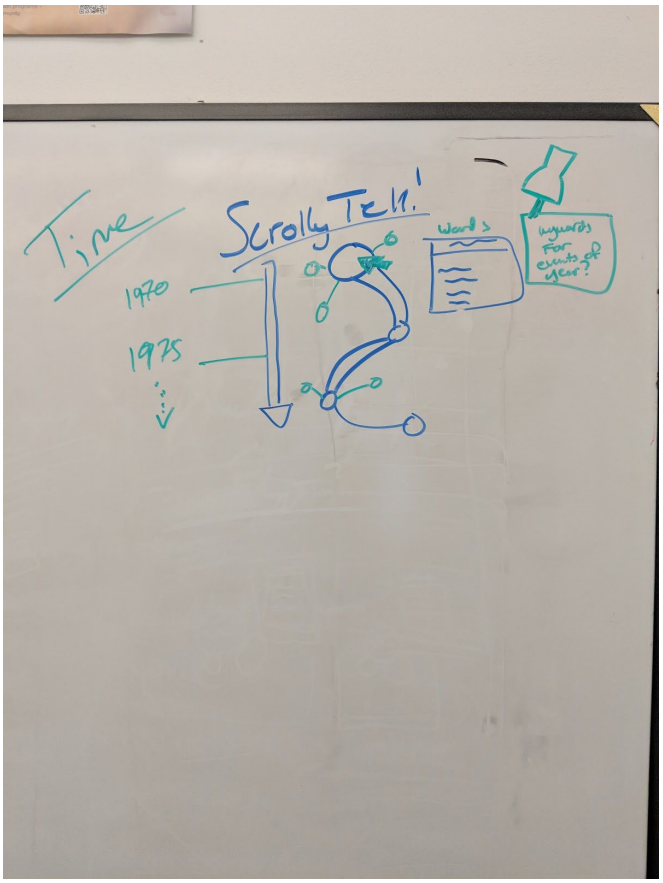




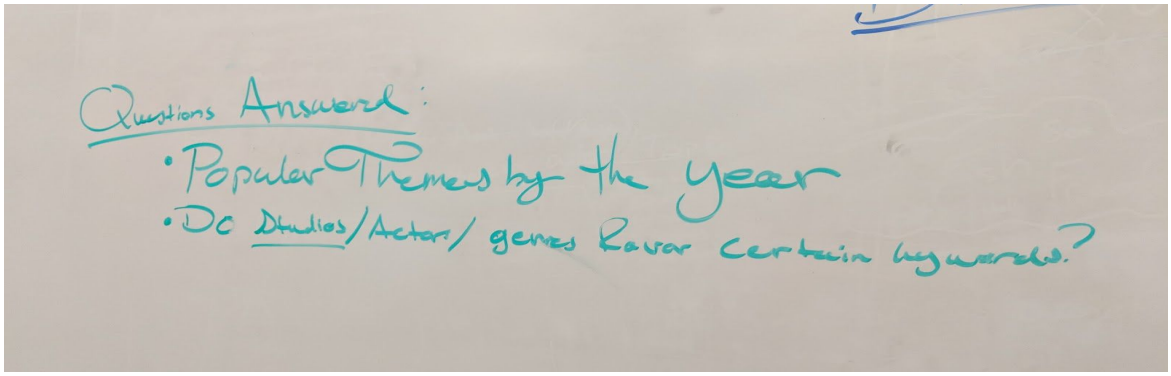


4. Combine & Refine.



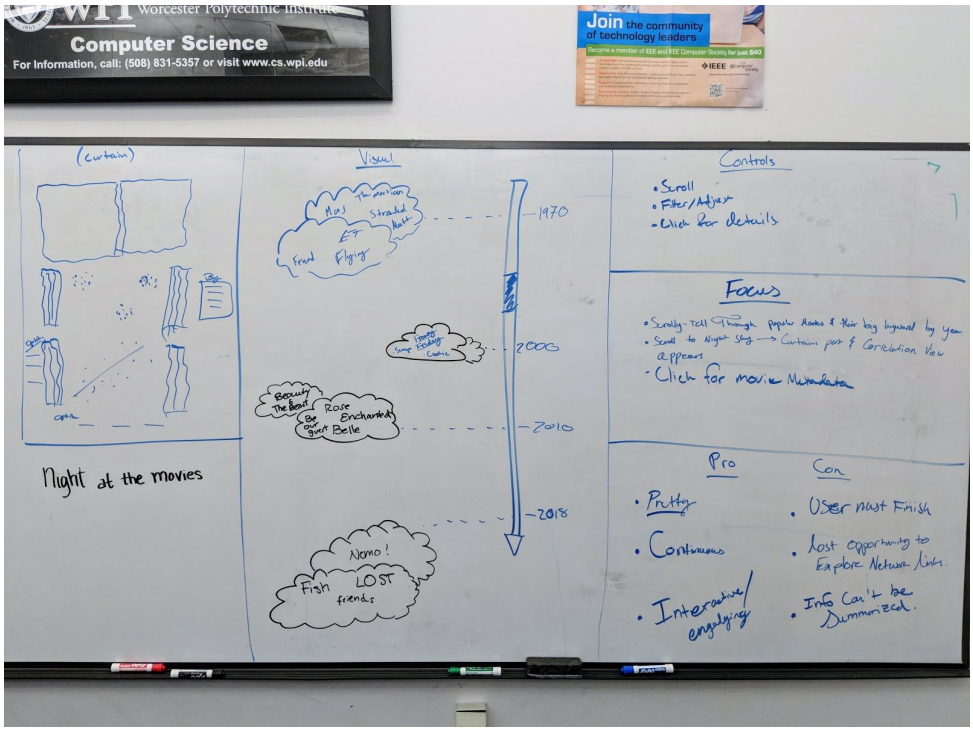


5. Questions (Partially pictured)

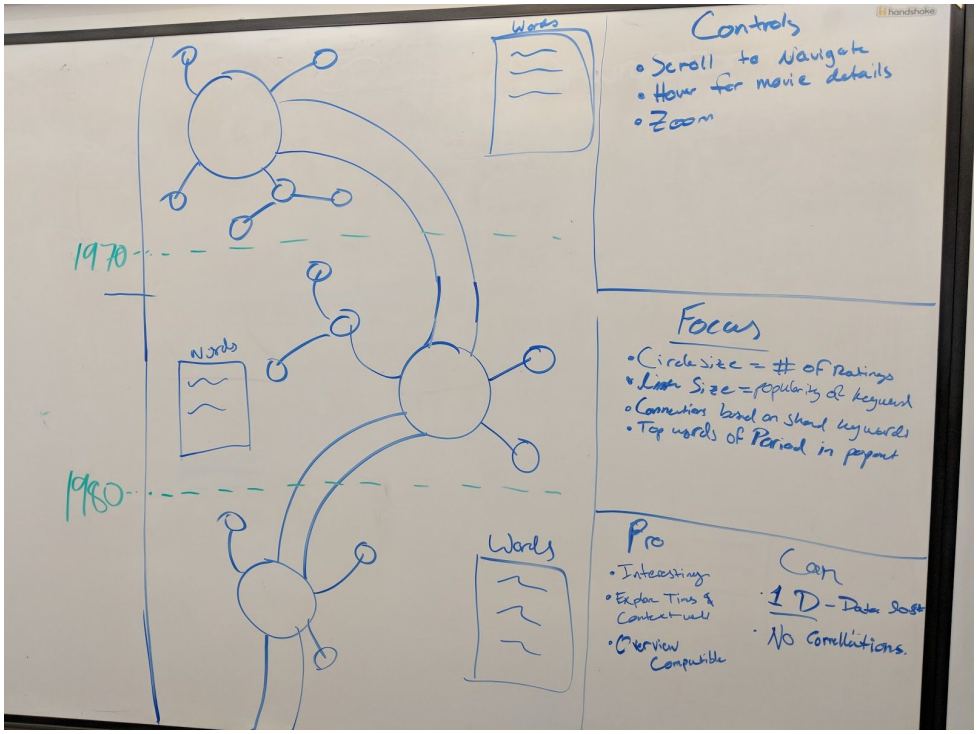


Five Design Sheets: Initial Designs

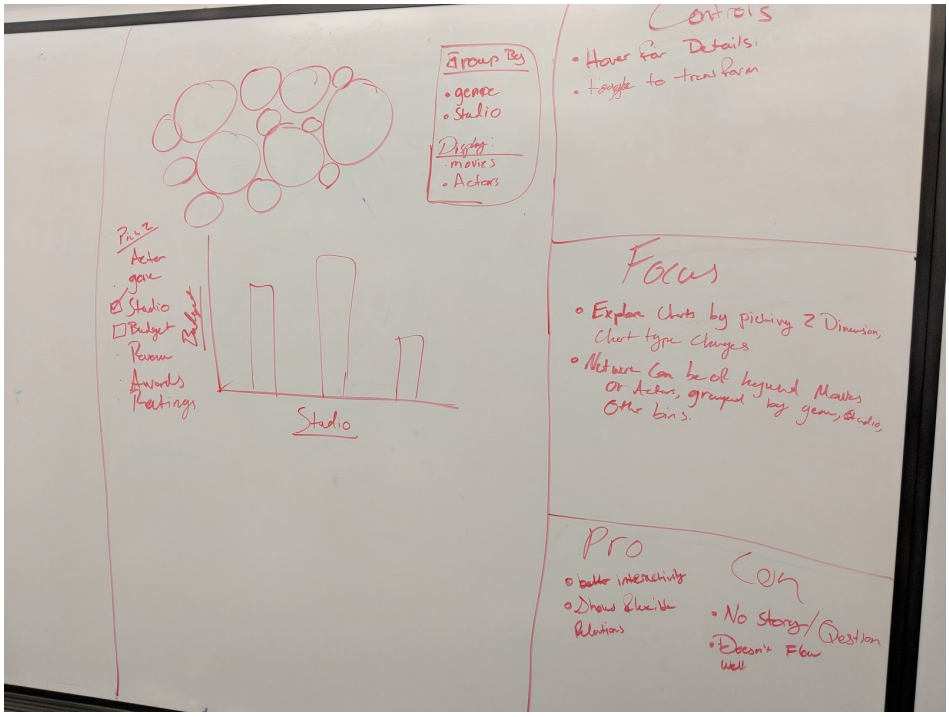
Design 1:



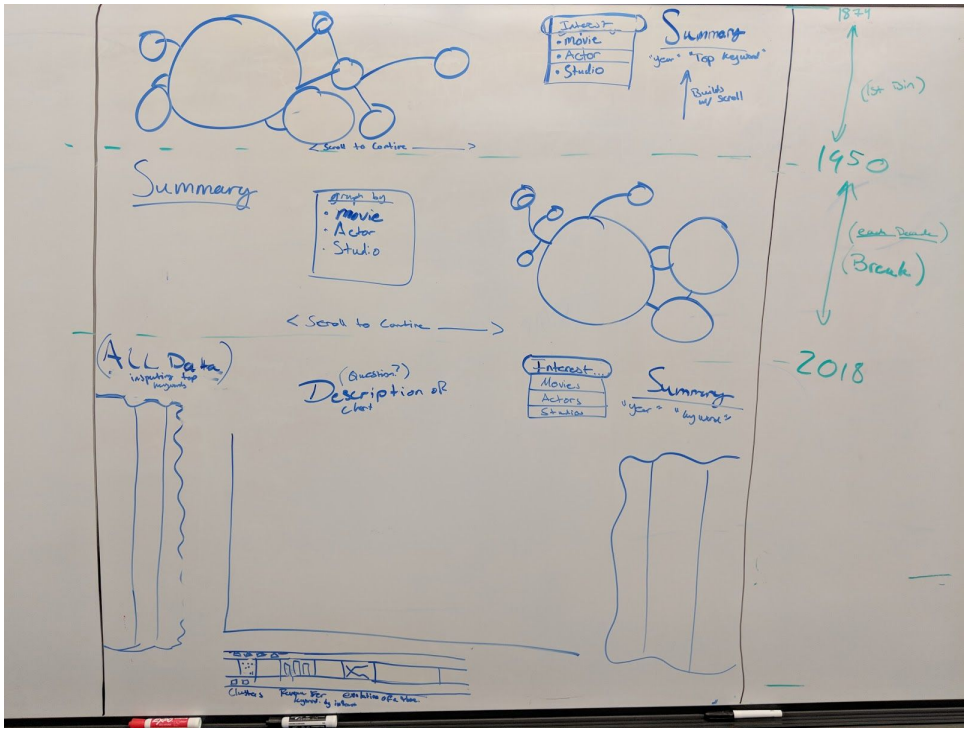
Design 2:



Design 3:



Final Design



Data Design / Algorithms:

This section discusses how to process the data and translate that into our visualizations. It should be noted most of these descriptions are preprocessing steps. What is served from the data server is described in the Data Server Design section.

Global Data:

- Top Keywords hashmap: bins to list of words for bin.
- Bin data map: Bin is mapped to its backing data structures to use in the bottom half, as in: Metadata map, Actor Maps, and Studio maps.

Top Half of Final Design's Data Processing:

i. Filter movie metadata for the bin, joining with Cast, Crew, and Keywords.
ii. For each bin, append its top 3 keywords to the Top Keywords HashMap.
iii. Node size correlates to movie rating, for movies, and to the frequency an actor or studio used a Top Keyword for the bin, for actors/studio networks.
    1. Movie Node Size: Rating * Number of ratings.
    2. Filter the Movies for the bin to only those with Top Keywords in them.
        a. Actor Nodes: Look at the top 3 actors from the movie. You will make 3 maps: ActorID-to-Frequency, ActorID-to-ActorName, ActorID-to-MoviesTheyHave. The last mapping's movies are defined as: { MovieName, MovieID, MovieRevenue, MovieKeywords }.
        b. Studio Nodes: Make the same maps as actors, but this time use the "produce_companies" field instead of cast.
iv. Links for Nodes are shared keywords, the more keywords they share the thicker the lines. Can click a line to see a tooltip of shared terms

Movies:

1. Prepare the output hashmap, which is a movieID mapped to a list of neighbors
    a. A neighbor looks like { id, name, [Shared words] }
2. Reduce filtered list down to list of movie ID, name, and keywords. This is the movies list.
3. While movies is not empty:
    a. Movie = Dequeue(movies)
    b. Add Movie key in output, with empty list
    c. For other in movies:
        i. If other is NOT in output keylist:
            1. Make list of shared words in Movie and Other.

2. If list not empty, build a neighbor object for other and append to Movie's output entry.

Actors:

- Same as movies code, but the movies queue is of Actors. Reduce the actors keyword map to be just the actor and the unique keywords its movies have.

Studios:

- Same methodology as Actors. Reduce the studios keyword map to be just studio and unique keywords its movies have.

Bottom Half of Final Design's Data Processing:

○ Clusters

Inspects relation of interests to each keyword.

Movies:

On hover show Name, studios, top 3 "Leading" actors.

1. Using global keyword hashmap, create a cluster map with keys as all terms in the global keyword map (ie the values).
2. Iterate over all movies (Can merge movie keyword list for all bins). For each of its keywords in the cluster mapping, add an object to each mapping's list: { Movie Name, ID, Studios, top 3 actors }

Actors:

On hover show Name. Use the actor's keyword list mapping built earlier (merged across all bins & reduced to actor-keywords) and repeat steps in Movies (only record actor name and ID, not studios or movies)

Studios:

On hover show Name. Use the studio's keyword list mapping built earlier (merged across all bins & reduced to studio-keywords) and repeat steps in Movies (only record studio name and ID, not studios or movies)

- ## Bar graph

  Inspecting actor and studio profitability from a keyword. Movies are grayed out for this vis.

  ### Studios:

  i. Reduce the studio keyword map (joined from all bins): merge the movies list for each studio such that the keyword frequency is stored and the sum of revenue earned from movies with that keyword.
  ii. Reduce the mapping further, finding studio with max frequency for keyword and remove its mapping from all other studios. Repeat for all keywords in mapping until unique keywords exist only.
  iii. Plot stacked bars: size of stack item = revenue for keyword, each stack-item = a keyword it owns.

  ### Actors:

  i. Repeat studio but with actor keyword map.

- ## Multi-line graph

  Inspects the popularity of a THEME changing over time, with respect to each interest. A theme is defined as a bin's top 3 keywords combined. Thus there are 8 themes if there are 8 bins. 'Popularity' is defined as frequency of the theme's keywords for movies, # of actors associated with the theme's keywords for actors, and # of studios associated with the keyword for studios. Actors and studios are double counted if they have 2+ keywords from a Theme.

  ### Shared for all:

  1. X-Axis ticks are the final year from each bin.
  2. Y-Axis Frequency of keywords in theme, the scale of which adapts when interests change (See Dylan's Assignment 4).

  ### Movies:

  1. Start with the Bin -> Keywords mapping to fetch theme of each year
  2. Y Value for bin is found by looking up the Movie Metadata entry in the Global Bin map, counting frequency of keyword from all movies (keywords can be double counted). Repeat for all Themes across each bin.

Actors:

1. Start with the Bin -> Keywords mapping to fetch theme of each year
2. Y Value for bin is found by looking up the Actor-Keyword entry in the Global Bin map. Count the number of actors who are in a movie with a keyword. This is repeated for each keyword in the Theme, meaning a unique actor will be counted at MOST 3 times total. Repeat for all Themes across each bin.

Studios:

- Same as actors, but use the Studio-Keyword list.

Data Server Design:

All data served from the server is from serialized files. These files were computed ahead of time in the preprocessing step.

Endpoints (Top half):

- /bin:
    - /summary: Returns the top three keywords for that bin, to display in the summary.
    - /interest:
        - /size: Return mapping of node IDs to ToolTip information & node value (deterministic of size).
        - /network: Return a list of links and the values making that link (to determine size of link and tooltip information)

  Where bin is one of < 1950, 1960, 1970, 1980, 1990, 2000, 2010, 2020
  Where interest is one of actors, studios, movies.

Endpoints (Bottom half):

- /clusters
    - /interest: Returns the JSON form the cluster map described in the processing step.

  Where interest is one of actors, movies, studios
- /bar
    - /actors: Return the Actor to Keyword+Revenue mapping
    - /studios: Return the Studio to Keyword+Revenue mapping
- /line
    - /interest: Returns a mapping of FinalYearOfBin to {Theme: [Keywords], Values: {{FinalYearOfBin, value}] }

  Where interest is one of actors, movies, studios

Other Final Design Notes:

- Any dependencies. E.g. this could be software libraries that the tool would be built upon, or aspects such as that it must be compatible with a current tool.
    - D3js, Heroku, NodeJS, ExpressJS
- Estimates of cost or time to build, or man-months of effort
    - I hope less than 3 weeks.
- Must Haves:
    - Scrolly-telly: After scrolling a certain amount, a progress bar displayed at the bottom fills up and the next decade's data is swapped into the DOM, updating it.
        i. Smooth transitioning

    - Network of graphs by time period
        i. Running summary of keywords by time period.
        ii. Current decade displayed in background on page.
        iii. ToolTip when hover on node.
    - Toggle between different datasets (Movies, Actors vs Studios)
    - Bar Chart
- Optional Features:
    - React, Movie theatre curtains, Cluster Graph, Multi-line graph

Schedule

Team Tasks:

1. Define data shape as fake jsons (Team activity) **(2/9)**
2. Network Graph
    a. Using dummy data, make the movie network for one decade. (Static sized bubbles that are connected)
    b. Then thickness of network connections
    c. Size of Bubbles **(2 / 15 - Aura)**
    d. ToolTip for Bubbles that shows information regarding the movie
    e. ToolTip for Network that shows the keywords
    f. Dropdown that shows interest
        i. Swaps data and visualizes it successfully **(2 / 20 - Aura)**
3. Scrolly-telly & Summary
    a. Top with bare bones Summary ~ three top keywords
    b. When you scroll through, it updates to ADD the keywords
    c. Swap networks **(2 / 20 - Kenedi & Dylan)**
    d. After final network swap to bar chart
    e. Maintain the state for current decade displayed after barchart
    f. Maintain the state for summary (JSON backed) **(2 / 23 - Kenedi & Dylan)**
4. Bar Graph
    a. Using dummy data make bar graph for movies
    b. Then add interactions/tooltip
    c. Adding the swap of interest **(2 / 15 - Kenedi)**
5. Data Preprocess / Server*
    a. Serialize JSONs for serving **(2 / 15 - Dylan)**
    b. Create REST server
    c. Add endpoints, serving the respective JSONs.
    d. Deploy to Heroku. **(2 / 16 - Dylan)**
6. Pretty!
    a. Animations
    b. Styling
    c. Color scales are appropriate **(2 / 26)**