Project III Process Book

David Diciurcio
Kenny Lei

Project II Work

3/7/2013
Brainstorming Session - David
Today, Kenny and I met up to start working on brainstorming project ideas. We had discussed various ideas in the past week, but our meeting tonight was to finalize ideas and submit the project proposal. We started off making a list of data sources we'd be interested in mining. Naturally, we chose to focus on our hobbies and websites we use on a daily basis. From here, we gathered a simple list:

Websites: Facebook, GMail, Amazon, Google Reader, The Verge, Ars Technica, Engadget, CNET, Youtube,

Hobbies: Tech, Video games, Movies, Android, Basketball, International Events

After going through the websites, we narrowed down the list to possibly parsing through the New York Times, some sports data, The Verge, or Amazon, or a combination of Rotten Tomatoes and Metacritic. In the end, I think we ended up deciding to work with movies, only because there's a lot of different data we can acquire and ideally, we'd love to include a rudimentary recommendation engine (visualized, of course). We also put together a very brief list of data we can analyze and compare:

Rating (critic and user), Sales, Genre, Studio, Producer, Actors, Director, Date, Comments, Social Network Participation

Parsing through the data shouldn't be too bad. Kenny and I both feel comfortable with Pattern and Google Refine (though Microsoft Excel is a stronger data refining source since we're both comfortable with the application). We both have experience with web development and we're both comfortable with jQuery, so we're looking forward to the project.

3/7/2013

Project II Proposal - Kenny

After brainstorming with David, we have come up with this project proposal. It will be centered on Movies as the topic.

Title:

Movie Trends and Relationships

Research Questions and Hypotheses:

What are the trends and relationships that exists with movies? We believe that there will exist trends in genre popularity over time as well as an increased relationship between critic ratings and the box office success of movies. We hope to also find a way of determining similarity between movies.

Motivation:

Movies are a very popular form of entertainment and enjoyed by everyone. However, it is always very difficult to find the right movie to watch. We can explore movies of the same genre or by a specific director/actor, but that is often not enough to find the right now. As Pandora was created to find similar music tastes, we wanted something that could provide movies that are similar and would be enjoyed. With the vast amount of accessible data, it seemed feasible to find relationships among movies as well as look at trends of movies.

Data:

Movie data is fortunately very abundant and accessible. Sites like metacritic.com and rottentomatoes.com provide movie data from the past decade. To construct visualizations, important movie data are title, producer(s), director(s), actor(s), studio, box office, critic rating, user rating, genre, release date, user comments, etc. All this information is abundant and easily accessible through both websites. These data points will help visualize trends and relationships among movies.

Visualization:

We aim to visualize trends in movie data over time and find relationships. Trends can consist of the popularity of genres, the movie career of a certain actor/producer. Trends will be visualized on a timeline. To visualize relationships, a complex graph with nodes and edges will link movies together based on similarity with options to filter and visualize close-up relations. The first priority is to identify trends, and it would be ideal to visualize the relationships of movies.

3/14/2013
Project II Final Proposal - David and Kenny

Title:
Movie Trends and Relationships

Research Questions and Hypotheses:
What are the trends and relationships that exists with movies? We believe that there will exist trends in genre popularity over time as well as an increased relationship between critic ratings and the box office success of movies. We hope to also find a way of determining similarity between movies.

Motivation:
Movies are a very popular form of entertainment and enjoyed by everyone. However, it is always very difficult to find the right movie to watch. We can explore movies of the same genre or by a specific director/actor, but that is often not enough to find the right now. As Pandora was created to find similar music tastes, we wanted something that could provide movies that are similar and would be enjoyed. With the vast amount of accessible data, it seemed feasible to find relationships among movies as well as look at trends of movies.

Data Source and Technical Process:
Movie data is fortunately very abundant and accessible. Sites like metacritic.com and rottentomatoes.com provide movie data from the past decade. To construct visualizations, important movie data are title, producer(s), director(s), actor(s), studio, box office, critic rating, user rating, genre, release date, user comments, etc. All this information is abundant and easily accessible through both websites. These data points will help visualize trends and relationships among movies.

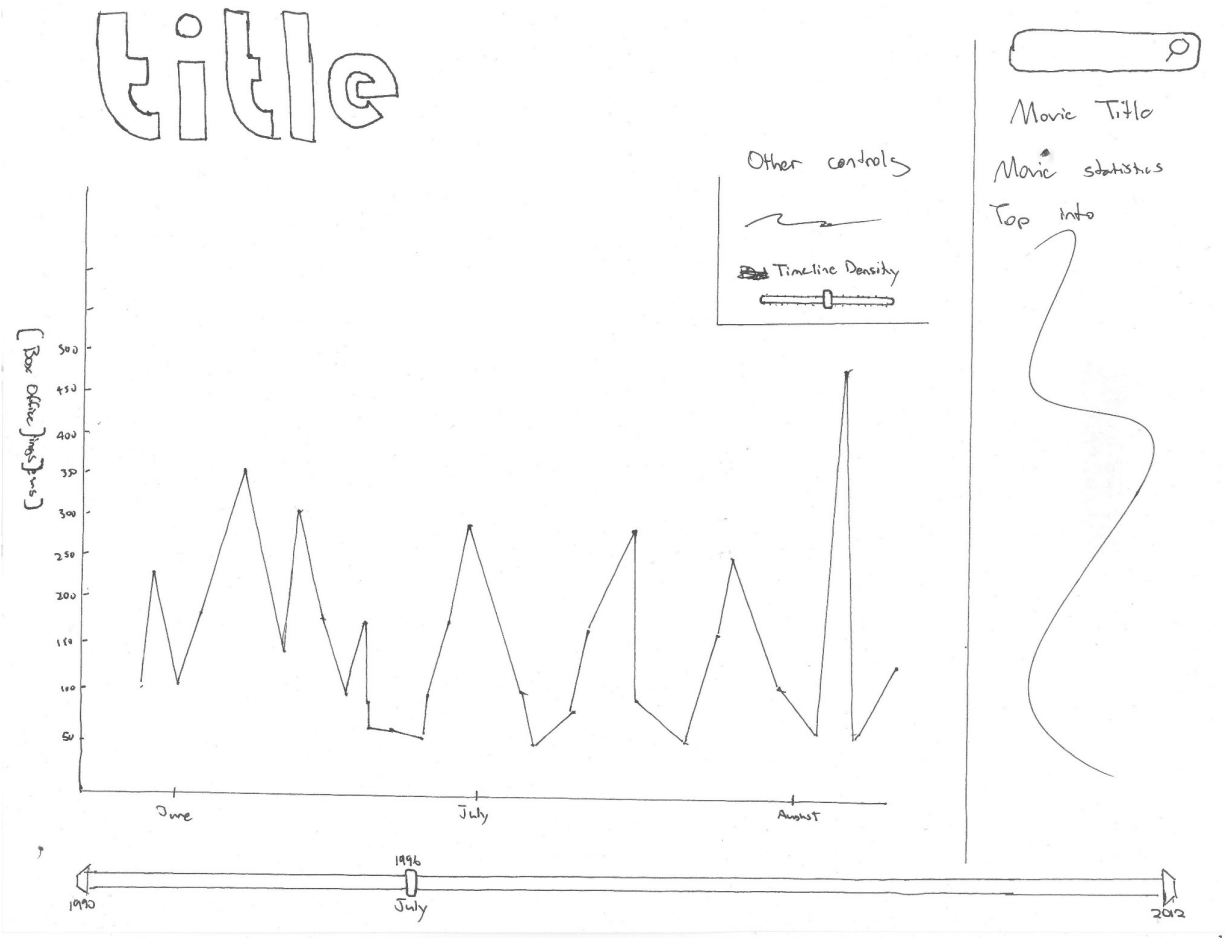
Since both metacritic.com and rottentomatoes.com do not use client-side scripts to load movie data, the Pattern library will be suitable for web scraping. If we notice any data that is unaccessible in the future, we will switch to Selenium for web scraping. It seems like most information are accessible through the DOM, and there should not be any problem with web scraping.

Visualization and Technical Process:
We aim to visualize trends in movie data over time and find relationships. Trends can consist of the popularity of genres, the movie career of a certain actor/producer. Trends will be visualized on a timeline. To visualize relationships, a complex graph with nodes and edges will link movies

together based on similarity with options to filter and visualize close-up relations. The first priority is to identify trends, and it would be ideal to visualize the relationships of movies.

For the web visualization, we will be using D3 with HTML/CSS to create an interactive web visualization. D3 will be used to incorporate features like time detail/granularity and movie feature filters.

Sketches:



The search bar will be the most powerful filter in this visualization, as you can choose to focus on a single movie, the movie career of an actor, the timeline of a movie franchise, etc. A toolbox, preferably above the graph or in the bottom right, would give the user fine-grain controls over the visualization. For example, maybe we would decrease the amount of data or have multiple graphs in the same area. Options would be presented. This visualization would be easier to put together and more simple, something that honestly appeals to both Kenny and I.

This graph was definitely more “out-there.” We were thinking what if we had bubble that through some parameter, would expand with the magnitude of that parameter. In the example above, box office revenues would dictate the area of the circle. Large column on the left and right would act as too that would allow the user to scroll through different dates and within these column, you could set the years. In the bottom right, we’d have controls and tools that would allow us to filter the data. Most likely, hovering on a movie would provide extra information. Most interestingly, we’d have a “similar movies” area in the bottom left area of your visualization. This would simply display movies which surpass some threshold of similarity. The list would be scrollable. The visualization would be super cool to implement, but it honestly seems like more work and more glamorous than it would be useful. Pretty graphs are great and all, but this class is on visualization data well, not how-to-be-showy-101.

(Belated) Spring Break Updated
Project 2: Putting together data scraping code - David

Over spring break, I decided to put together some code that would scrape the data from IMDb. Originally, we had decided to use metacritic or rotten tomatoes, but unfortunately, using either site (or even both) would've taken a lot more time and given us less data. Mainly, IMDb has information about budgets and links to the metacritic score, as well as loads of information about actors and other factors. This code wasn't particularly difficult to put together, as I springboarded from homework 1. However, the website fundamentally changed between writing the code and homework 1. This just meant that I had to review all my functions to make sure that pattern was accessing the correct elements. I added metascore, budget, box office us, box office world, mpaa.

Essentially, the code goes to a large list of movies offered on a year by year basis. The movies are ranked by some proprietary, non public rating, "Moviemeter." Luckily we could load each respective list by slightly altering the URL, and then we go through a certain number of pages, again by altering the URL, and load information about every movie. For the sake of time, we ended up loading 100 movies per year for 11 years. Our intention was actually to only do 10 years worth of data, but we realized at the last second that we set it up for 11 years.

One of the larger issues I went through was pulling budget info. Most of the information was on a separate page, so I'd have to look for a URL on each respective film's page. Then I'd have to use regular expressions in order to find the specific box office information and budgeting. After this, I finally had all the functions I needed, but when I started running into the code, I realized it would crash anytime any element was missing. At first, I tried individually identifying all the errors and addressing them, but to me, it seemed a fruitless endeavor. Seeing as the major issue was try to access elements from empty arrays, we needed an easy to stop the code from crashing. Rather than checking to see whether every array was empty, I simply wrapped all functions in try/except statements. If the function tried to access an empty array, it would return an empty string, signifying that the element did not exist.

Kenny ran the code for us because his computer is a little faster and he has a more secure ethernet connection. For him it took a couple hours to parse through all the websites. After failing to run a couple times, he had to change pattern's timeout variable to 30 in order to prevent connection timeouts that had us waste a few hours total waiting for the code to finish running. We were considering simply starting where we left off and merging the datasets, but we felt that it'd be best to have our scrapper be able to complete one full of scrapping without running into any problems.

Here is the link to our original data set:
https://www.dropbox.com/s/0e69gryoxfqoepb/project2_original.csv

Project 2 TF Feedback
3/23/13

Hi David and Kenny,

Your possible parameters are:

critic rating
user rating
genre
year
box office gross
producer
director
actor
studio

I think you need to decide what your primary research question is. Is it, what is the change in genre popularity over time? If so, a line graph makes the most sense, with time (release date) as the x axis, and different lines for different genres, with popularity measured by box office or critic rating (or both, possibly with filtering or switching capacity) on the y axis. You'd have to decide what time is: all movies in a genre for a month, year, etc. Is it, how are movies similar? If so, maybe a visualization like http://mbostock.github.com/d3/talk/20111116/iris-splom.html is better, where the boxes showing the strongest trends might be most predictive of similarity. Your different parameters could be release date, box office gross, critic rating, etc (x and y axes), and maybe could be color coded by genre, studio, actor, director (with an "independent" colored category for the smaller ones) - allow the user to choose. You might be able to come up with a way to combine the timeline and the similary prediction: if you had both side by side, maybe selections on the timeline (like a different line/genre or year) would change the similarity/trend analysis.

The search bar is a cool idea, so is the year filter.

Then you could incorporate your other parameters, or show breakdowns, in zoom-ins/filtering/brushing and linking when you click on the dots or lines. These are my thoughts - in the end the choice is yours (and of course what I just discussed is not necessarily the ultimate solution).

Please do still read my commentary about your initial proposal (copied below). Also, please feel free to email at any time for questions or feedback, or to come to my office hours Tuesdays 8-10p.

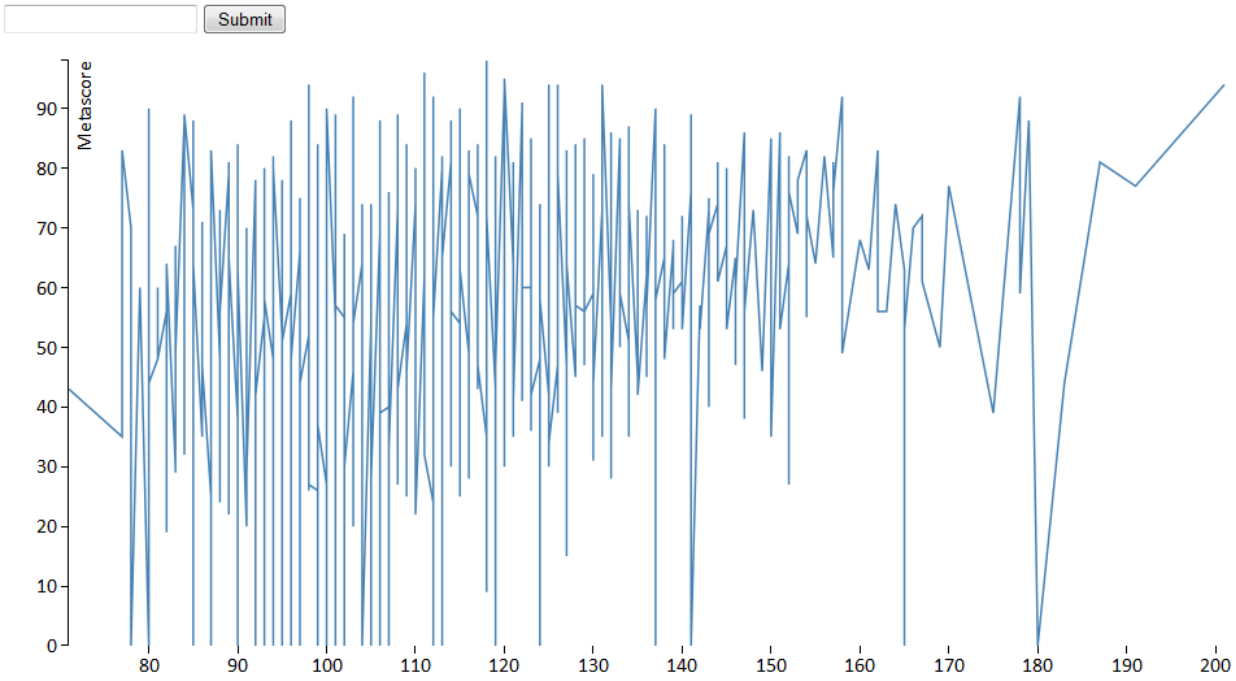
Best,
Lila

Scraping Issues - Kenny
3/28/13

So David asked me to go over his scrapping code earlier, and at first, I assumed he must've made a big mistake. About half the entries he was scrapping for returned blank, and in general, the code seemed rather finicky. We exchanged a couple of messages, where David admitted that his code wasn't perfect, so I assumed that I needed to fix something in it. After hours of going through it, I couldn't determine why my code still wasn't picking up the missing entries. Lucky for us, David and I decided to meetup and get food, when we discovered that something specific to my computer was causing the code to not function properly. Running David's original code on his computer would pick up 95% of the information, while on my code, it would pick up maybe 50%. We hypothesized that there might be an issue with pattern's implementation in Windows, but I will just start using a virtualized Ubuntu for python. At least this will fix the problem.

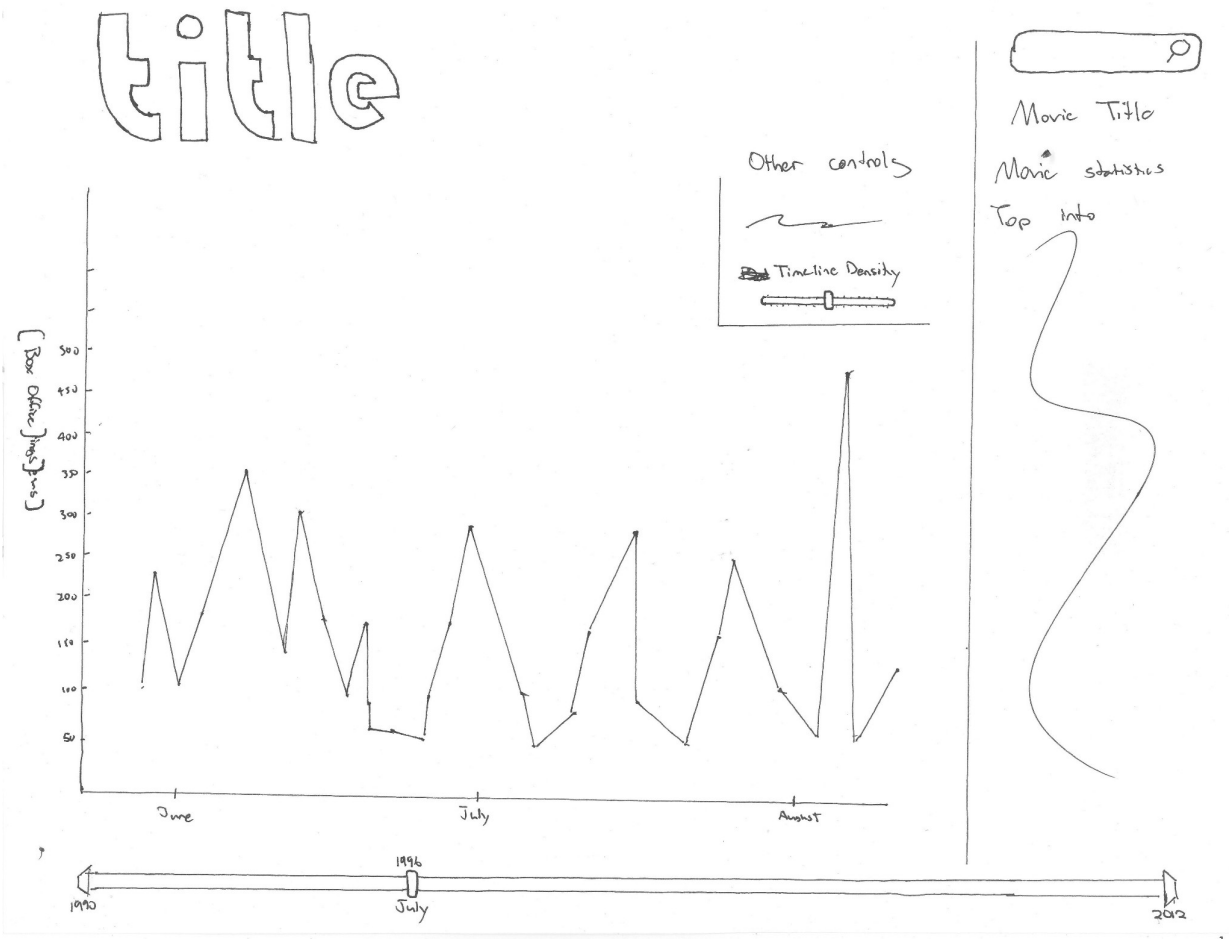
Scraping Update - Kenny
3/29/13

So we were about to start visualizing the data when we realized that we forgot to scrape one of the most important elements - date! David is currently modifying the code while I start to play around in D3, just to get something to show up. Below I have copied just the first visualization - this is time of film vs metascore. Already from here, we can point out a few issues we're going to need to address: one, the graph spikes downward anytime we're missing a data point. Secondly, the data seems rather tight as of now. Maybe we'll find some way to start expanding it (given enough time).



Planned Visualization
3/31/13

So Kenny and I are leaning towards making our visualization closer to something like this. We're going to have one major visualization area (in this example, its a line graph) and that area will be able to have different x axes. Depending on the x axis, different y axes options will be available. This will obviously work for cases where the two data sets we're comparing are both numerical, however we might need to expand for other data. For example, maybe bar graphs for how many movies have a certain MPAA rating. To be honest, the search bar itself would be nice, but that might be something for a different project. When you select a point on the line graph, you should get a little hover-over giving additional information for that movie. I understand we need to implement some sort of linking mechanism so maybe we'll highlight top information in a line graph or gray out certain information. The time scroll bar would be awesome to implement, so hopefully that shouldn't be too difficult or time consuming.



Cleaning up the Data
3/31/13

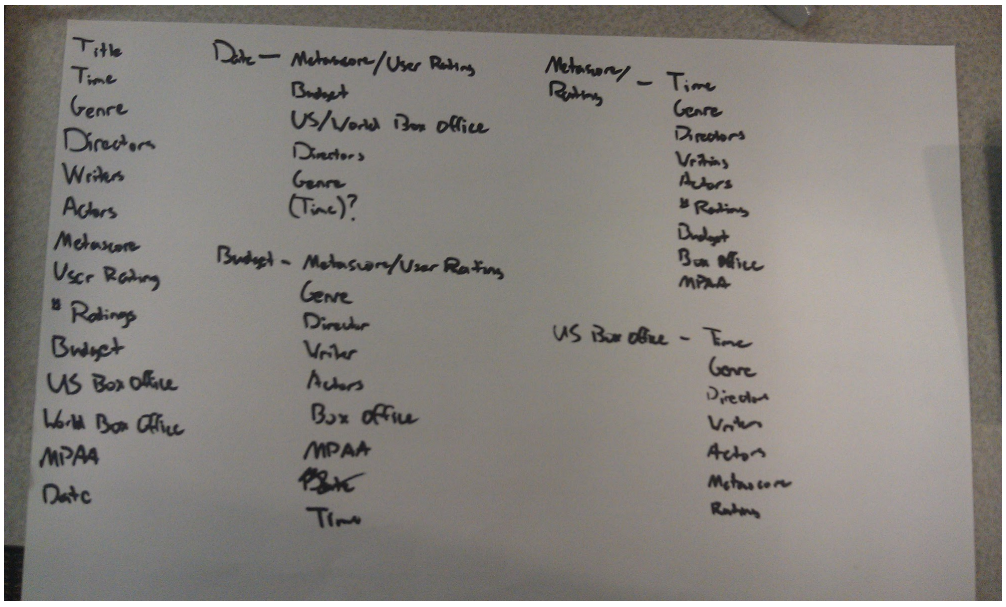
So at this point, we have our data, but we're trying to decide what will cause problems and what needs to be changed. First things first, I started cleaning the obvious stuff: removing commas, removing dollars signs, converting things from strings to ints. At this stage, I stuck to Google Refine for most of these batch operations. During this process, I realized that we weren't really getting a lot of useful information from World Box Office - more often than not, the information was coming back empty. From here, we decided to simply remove the column itself. Had we known the information would've been less helpful, we wouldn't have scraped it in the first place.

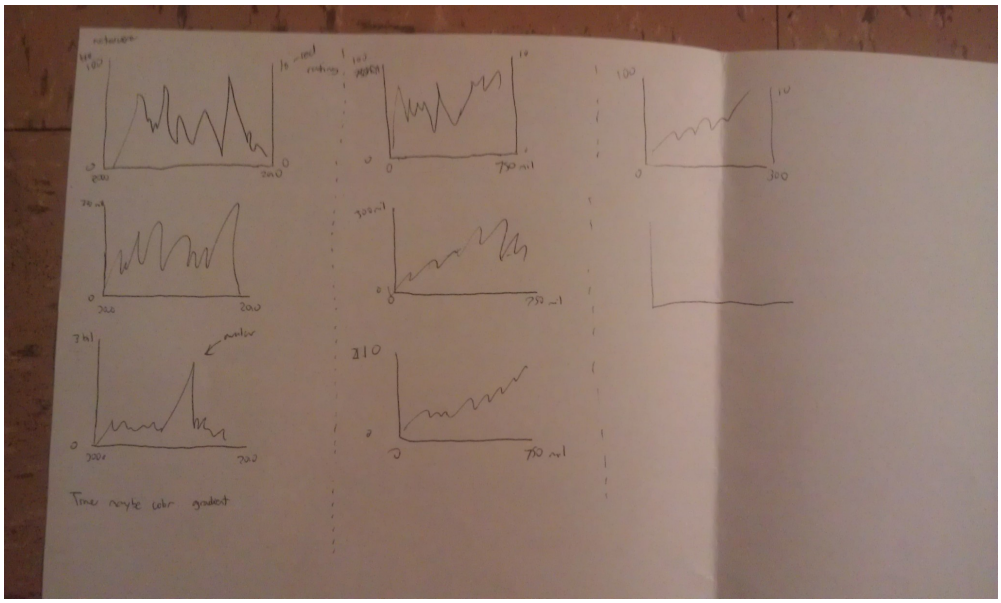
At this stage, the data was significantly more organized and usable. However, Kenny and I decided that we didn't want our code to be having to clean the data at all, so we were more liberal in removing data. I opened up the csv in Excel and started removing rows that were lacking multiple entries. Usually if the row was missing date or metascore, it'd often be missing other entries. In the end, we decided to remove any incomplete row. At the end of the process, we still ended up with over 900 movies, so we were certainly not short of data.

Here are the links to the different data sets as we prepared them
https://www.dropbox.com/s/yl9qiyhpjfqwwpy/project2-csv.csv
https://www.dropbox.com/s/rl6zxymztdvxs9k/project2-update1.csv
https://www.dropbox.com/s/dq7edhbaskwpmw4/project2-update2.csv

Project Planning
4/2/13

So here's a rough organization of how we could break down our data. At least in terms of line graphs, we will have the data, budget, rating and box office records as factors for the x axis. Perhaps buttons or a dropdown menu will allow us to switch between the different choices. A similar options will be available for the y axes. I'm going to do a brief sketch/test of all of these and see if they make sense. I will post this later. Maybe then I'll also try to put together an alternate graph for non numerical information. Our primary focus is lines graphs but maybe we could get a combination line/area graph in there as well? Again, this will need to be figured out more so as a factor of time.





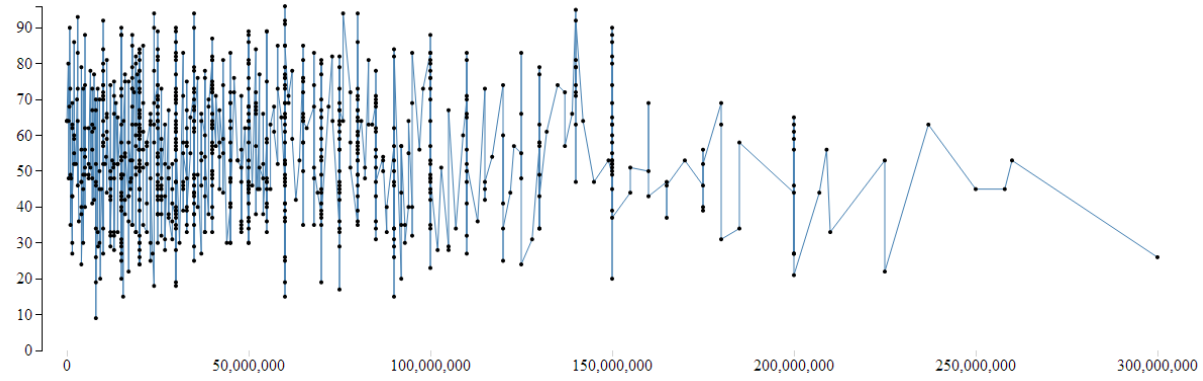
Sharing the code
4/2/13

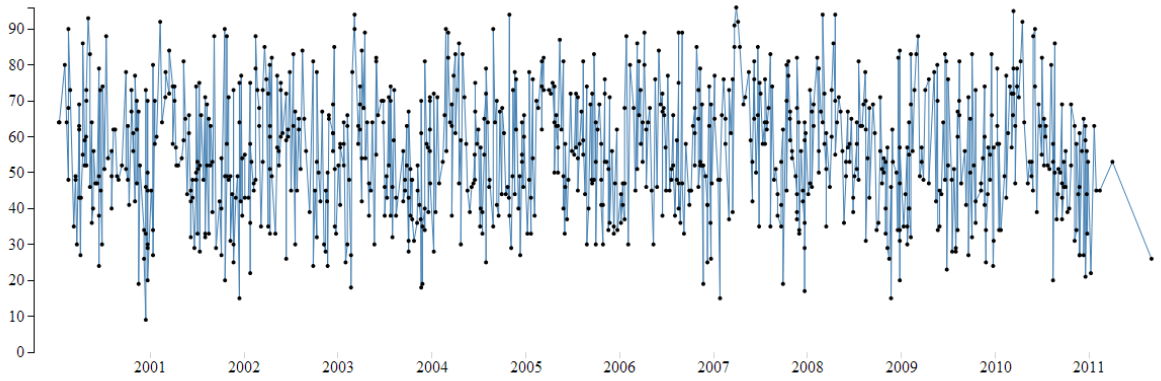
For the most part, Kenny and I have avoided requiring git or something. We usually work at different times, so dropbox has been satisfying our needs. However, as we start to work together more, it's becoming annoying to think about saving at the appropriate time. Maybe for project 3, we'll set up a git repo.

Setting up a graph

Earlier today, I pulled together a functioning graph with budget on the x axis and metascore at the y axis. It was a joint effort between Kenny and I. Kenny's code laid the foundation, and I tried to improve upon it. Though being his code and it not being particularly well commented at the time, I half rewrote (not necessarily for the better). It ended up helping me understand D3 way better as I followed his code as well as another example online. He was having issues getting points to show at every data point, so through modifying the code, we made it significantly more workable. During the process, I ended up running into the most obnoxious of problems - when javascript tries to sort an array of numbers and one of the values is not a number, sort will place the NaN in a seemingly arbitrary place in the "sorted" array. This caused me frustration for over an hour, because I couldn't see where my NaNs were coming from. In the end, I discovered that the data included two blank rows at the end, and it was failing to read both of those. After removing those two rows, the code worked perfectly and we had our first full graph up and running.

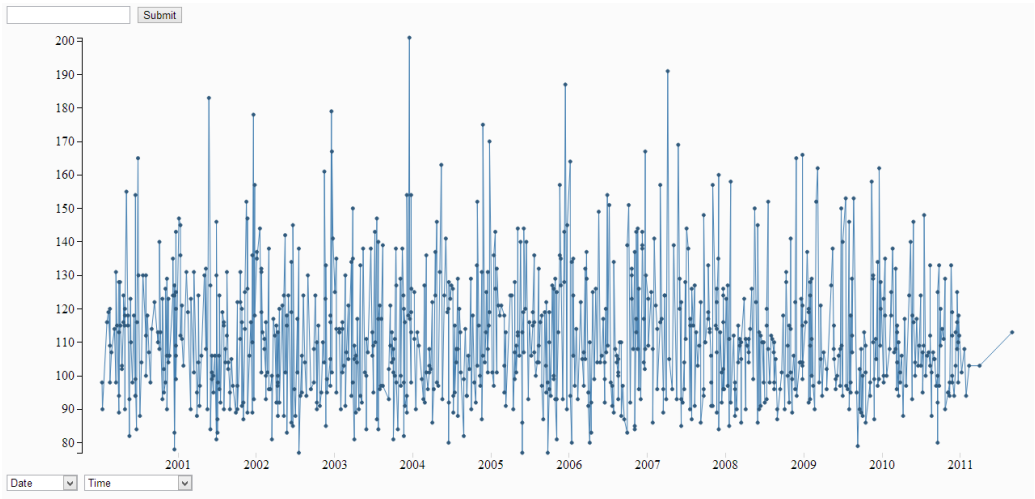
Our next goal was to get date on the x axis. I stupidly avoided taking advantage of how well D3 handles dates, and tried to implement a complicated system that would plot them anyway. Kenny, seeing my frustration, near instantly reminded me of time format parsing and fixed our next major issues in a couple of minutes. After playing around for a bit, date was working as were all of our other x axes - metascore, box office, and budget.



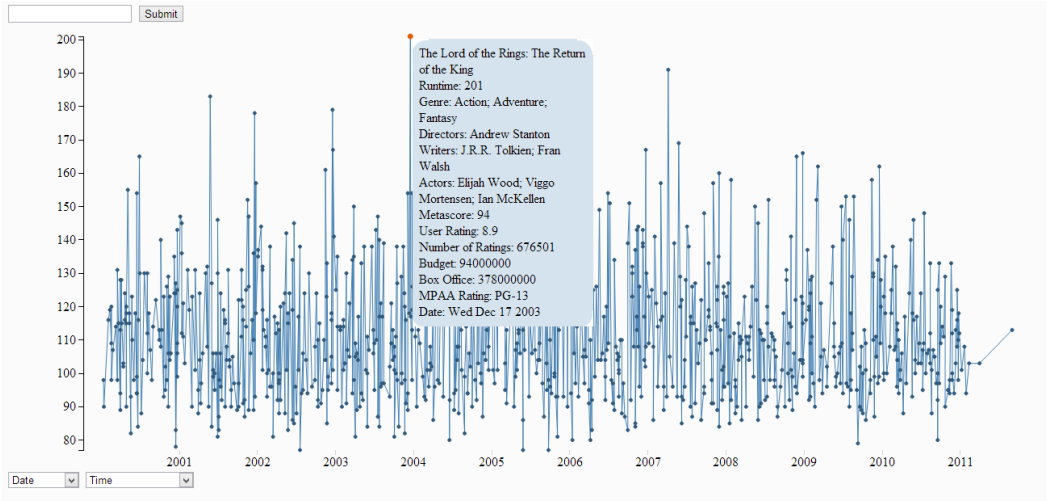


Requirements Complete

So earlier today Kenny and I technically completed the coding portions of our project. I mention technically because we're still hoping to improve the design of the website and add some extra functionality. We have interactive highlighting and multiple views. We have a details-on-demand hover that provides information above every film. As I speak, Kenny is currently working on adding a data slider and then, if we *still* have time, we might try to implement a search function. I am taking the role of cleaning up our design and taking care of the process book. Here's an image of the original finished project (I mentioned original because I expect more finished products).



Obviously you can see some of our more logical decisions. We updated the color of our points that goes way more in line with our design. We go with a simplistic gray to dark blue theme, with orange used for selecting points and high contrast.



That orange point will stay highlighted even when you change the x and y axes. Also you can see

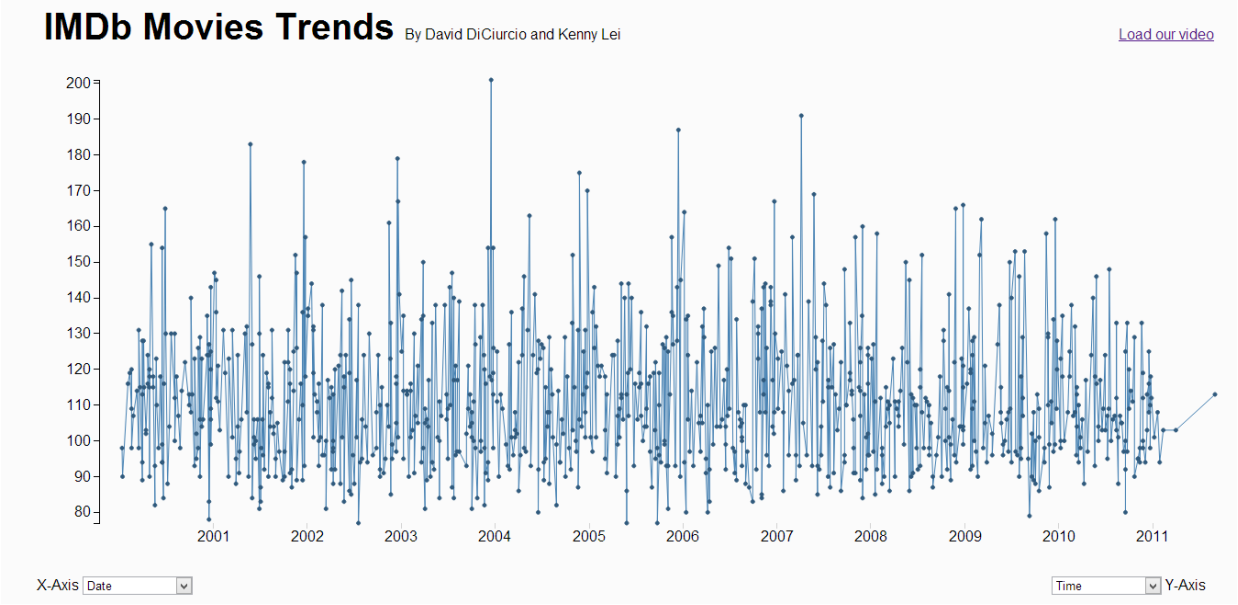
here an example of our hoverview. It uses only css/html and no images, so it is great and fast! On top, you can see our still obsolete search bar. Again, it's a secondary feature and we will take it out if we can't have it working by midnight. Getting more specific, let me point your attention to the axes. Both are calculated depending on the parameters of the data. For the most part, we let D3 handle this, but there are a couple of cases when we specifically address data. For example, we try to maintain data integrity by always starting at 0, but the in the case of film runtime, we thought it'd be better to start at the minimum runtime, otherwise, half our chart would be designated to showing blank space.

Range Slider and Search

In addition to the current state of the visualization, we attempted to incorporate more interactivity and data refinement. Data on the x-axis was sometimes too condensed and made it difficult to discern data points from each other. A solution to this design issue was to incorporate a range slider along the x-axis of the graph. We looked at some examples incorporated D3 and range sliders and found this example (http://bl.ocks.org/dem42/3878029) to be the most useful for our needs. This example used the jQuery UI range slider element, which uses external css and js libraries for its functionality. While implementing the range slider, we encountered difficulty in creating a slider when the x-axis variable was anything other than a pure number. For example, several bugs arose where the jQuery UI slider simply would not function with movie release dates. We were able to get the graph to have variable range and therefore be able to expand and zoom in on portions of the graph. However, although range of data can be simply changed for an appended path, filtering the circles to be displayed in the graph proved extremely difficult to incorporate. There was no simple way to have the circles transform in a similar fashion to the line. We decided from the design point-of-view to exclude the range slider from our final product since the partially-functionally slider did not have any usability benefits. The circles were vital for providing details-on-demand, which is essentially the purpose of expanding and zooming in on portions of the data.

Another feature we initially wanted to implement was a search bar to filter movies displayed based on actors, directors, genre, etc. However, this feature was slightly complicated with our complex, dynamically changing view. Essentially when any form of interactivity occurred with the visualization, we were forced to redraw the entire svg, which takes a lot of runtime. We experimented with existing search algorithms, but the time it took to effectively filter data was too slow for modern web standards. We were unable to come up with any better solution on our own with the time given for the project, so search was, also, unfortunately excluded from this project.

Final Submission



As you can see above, our final changes proved to be more evolutionary than revolutionary. We chose to add a clean title in the top left, replacing the search bar that we chose not to implement. The axes are controlled by two dropdown menus which each take up a corner of the visualization. Finally, we include a link to our screencast in the top right. Not shown here is our description of our visualization. From here I'll go into some of the in-depth analysis portions of the process book.

Describe the intent and functionality of your visualizations. What did you learn about the data?

When Kenny and I originally met up, we decided we wanted to visualize a hobby of ours. Kenny liked basketball while I enjoyed tech news. Between us, we both enjoyed playing video games and watching movies. We were inspired by Gonda Felix's work, where he found trends in video games. Part of us wanted to copy what he did, but in the end, we chose to work with movies instead. Now for us, we had more features we wanted to implement, things that were not imperative for the project. Search and a scroll bar for fine tuning the x axis were planned and worked on to some extent, but not finished on time. Kenny and I made a call to submit a project that was more refined before we submitted something with more features.

Of course, our visualization offers the user to view trends in movie viewing. For example, after looking at the visualization, it becomes clear that there are certain seasons and off seasons for movies. Just to pull an example, we see that during months like September, October and March, the films that are released having noticeably smaller budgets. We also see that the idea of summer and winter blockbusters is very much alive. Many of the top grossing films were released around christmas or the summer.

One of my favorites features completely surprised me. In our original planning stages, we forgot we needing to include linking/brushing so late in the process, we realized we needed to implement some feature to highlight selected movies. This is a fantastic feature. It's really interesting to be able to notice that Lord of the Rings films are some of the highest rated films, but compared to other movies, did not have crazy production costs. These relationships do not jump out at the viewer without the context of the surrounding data. Fifty million on a movie seems small to me (for whatever reason) but I realized that it's a very average amount for a producer to spend.

As with any visualization, our goal is to facilitate the consumption and comprehension of a lot of information in a small set of time. From there, the user can draw conclusions. Perhaps our visualization isn't perfect yet, but this was a product of a relatively short amount of time. As mentioned above, given more time, we very much would've included more features and probably will come our next project.

Visualization Integrity:
When displaying data points in the line graph, we did our best to maintain visual integrity, in which data represented the values that they appeared to be. Axes and data range were taken into account to reduce confusion and bring out the existence of actual trends and relationships among the data.

Visual Encoding:
Here I'll cover very minor points that would not be covered elsewhere. For example, we spent some time playing with the size and the colors of the points. We obviously needed something with contrast and it had to be the appropriate side. From some example code, our plot was blue,

so we decided to adopt a blue theme at that point. We then settled on a radius of 2 - enough to make the point noticeable, but not enough to make the graph look weird or not fluid. When the user hovers over the point, it will increase its size to a size of 3, to indicate to the user that a data point has been selected to be viewed. A tooltip appears on hover over a point, presenting data details about the point. Clicked points were made noticeable from the rest of the points with a separate coloring to bright orange.

End of Project II Work

Lila's Feedback on Project 2:

Hi David and Kenny,

Congratulations on finishing project II! Please use the following feedback to inform your homework 7 and project III. You can do a different project or even change groups for project III - but I think you have a good thing going.

Screencast
Your video is good. It does a good job of showing how the visualization works, but doesn't do a great job of selling why your visualization is interesting/important.

Process book
Your process book is great. You did a really great job of using your process book to brainstorm (what it's best for!) and document and justify your choices. It would be good to host it online and include a link to it on your web page.

Technical aspects/full features
I found no bugs in your code or on your page. You implemented brushing and linking, multiple views, details on demand, but as far as I can tell, no drill down/filtering - I know you had difficulty with this.

Design
I love that your visualization works even in a small browser window, is simple, and I really love your brushing and linking. The entire thing is clean and has great color choice. Here are some suggestions:

• Incorporate drill down/filtering. Your original ideas were a search bar or a year slider - might you still be able to do those? Filtering was an important requirement in the spec.

• Put less information in the tooltips. One thing you could do is have a details div on the side (or above or whatever) with all that information that changes with your selection, and maybe just have the movie title with the relevant x and y axis values in the tooltip. It's too much to read, and the text should be separated out by at least bolding or something.

• Think about the axes a little more. I think being able to choose the axes is really cool, but are drop downs the best selection mechanism? I also think that both dropdowns being on the x axis (instead of one on the y axis) is unintuitive. You could have a navbar type thing next to each axis (here is not a beautiful example, but gives an idea of what I mean).

• Move your descriptive information up, I think, to below the title, or even just have an info button with a popup with that information. Make sure all the spacing and padding on the web page is adequate/nice-looking too.

• Give your visualization context. I believe your main goal is examining the relationship between important characteristics for different movies and movies overall - including a trend line for each will help show that relationship.

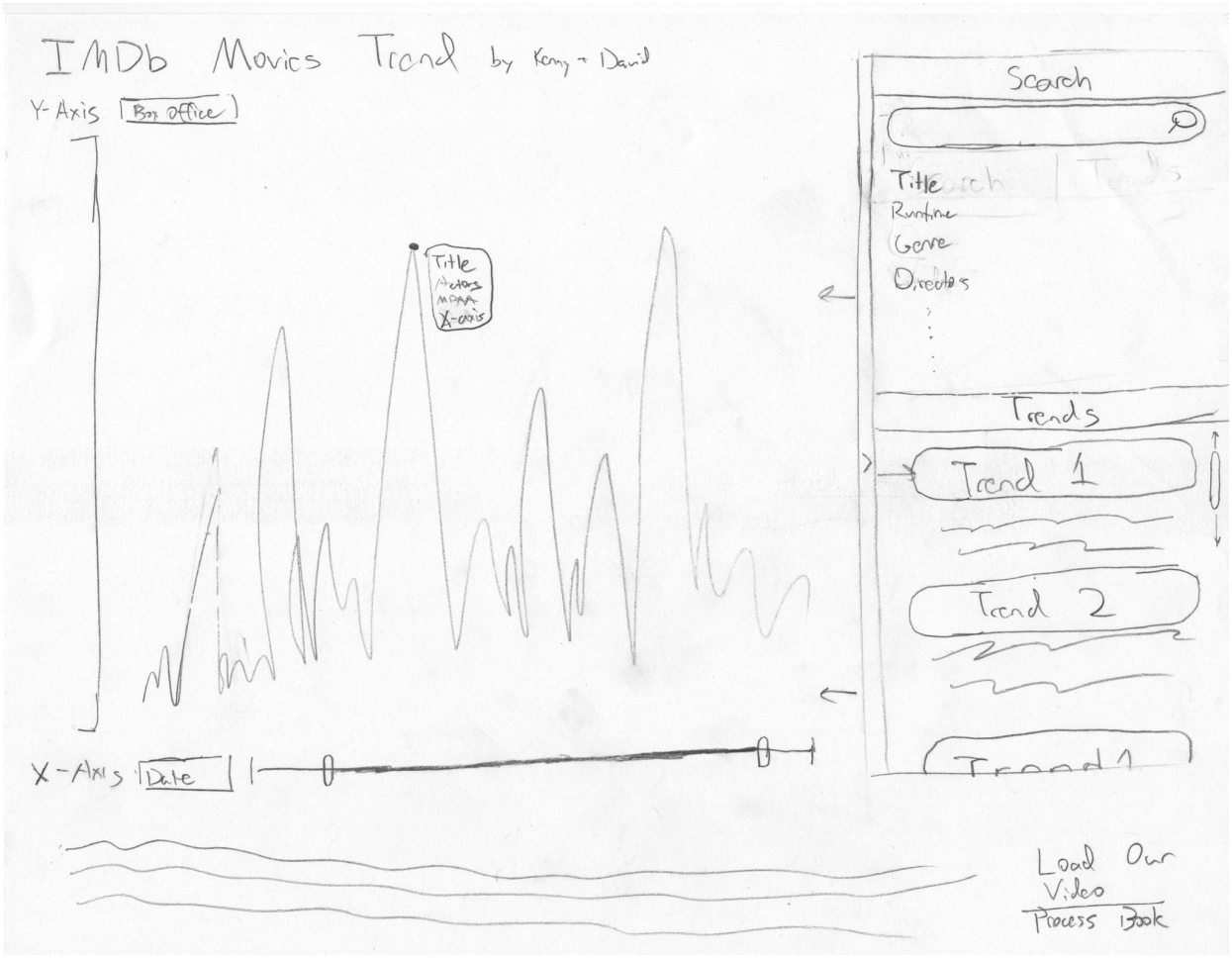
Final grade: 3.5

These are all suggestions - it's up to you to decide what works best, and to come up with more for project III. Read Project III and Homework 7. I'll be in contact later today to offer meeting times this week.

Best,
Lila

Project III Meeting

Our project III meeting went somewhat unsmoothly with Lila. We had to push back our initial meeting and we didn't have our sketches and updated javascript on hand, so we sent them to Lila later. Overall, we decided that our visualization was clean and straightforward, but somewhat on the simple side. Also, we need to figure out a good way to tell a story - we're thinking of displaying different movie trends decade by decade. That wouldn't be terribly difficult to handle as long as we break up it up by decade and hardcode in the trends themselves.



Here's a sketch of a possible project III layout. On the bottom we've added an x-axis slider and on the side, we have useful tools such as search and the movies trends. When you click on one trend, a description of that trend should appear. This meshes in the story telling aspect as we'll be offering the evolution of film in the second half of the 20th century.

Possible Trend Topics

Movie Trends:

50's

Pro America

Grand Films

Early Activism

Disney

60's

Family films most successful

Nuclear fear

Innovative films

70's

New Hollywood

More sex and blood - Last House on the Left

Controversy - Straw Dogs, A Clockwork Orange, The French Connection, Dirty Harry

Auteur Theory – Taxi Driver, Godfather, Nashville, Annie Hall, Manhattan, Badlands, Days of Heaven, Chinatown

Martial Arts – Enter the Dragon, Snake in the Eagle's Shadow, Drunken Master

80's

Lucas-Speilberg:

Sequels – Two Star Wars, three Jaws, three Indiana Jones

Blockbusters - ET

Martial Arts (Jackie Chan) – Project A, Wheels on Meals, Police Story, Armour of God, Project A Part II, Policy Story 2, Dragons Froever

90's

Re-rise of family animated films – Beauty and the Beast, Aladdin, Lion King, Toy Story

Rise of independent – Sex, Lies and Videotape, Reservoir Dogs, Pulp Fiction

00's

Increase in Documentaries – March of the Penguins, Bowling for Columbine, Fahrenheit 9/11, Voices of Iraq

Epic film – Lords of the Rings, Harry Potter, Gladiator

Globalization – Crouching Tiger, Hidden Dragon, Amelie, Lagaan, Spirited Away, City of God, The Passion of the Christ, Apocalypto, Slumdog Millionaire, Inglourious Bastards

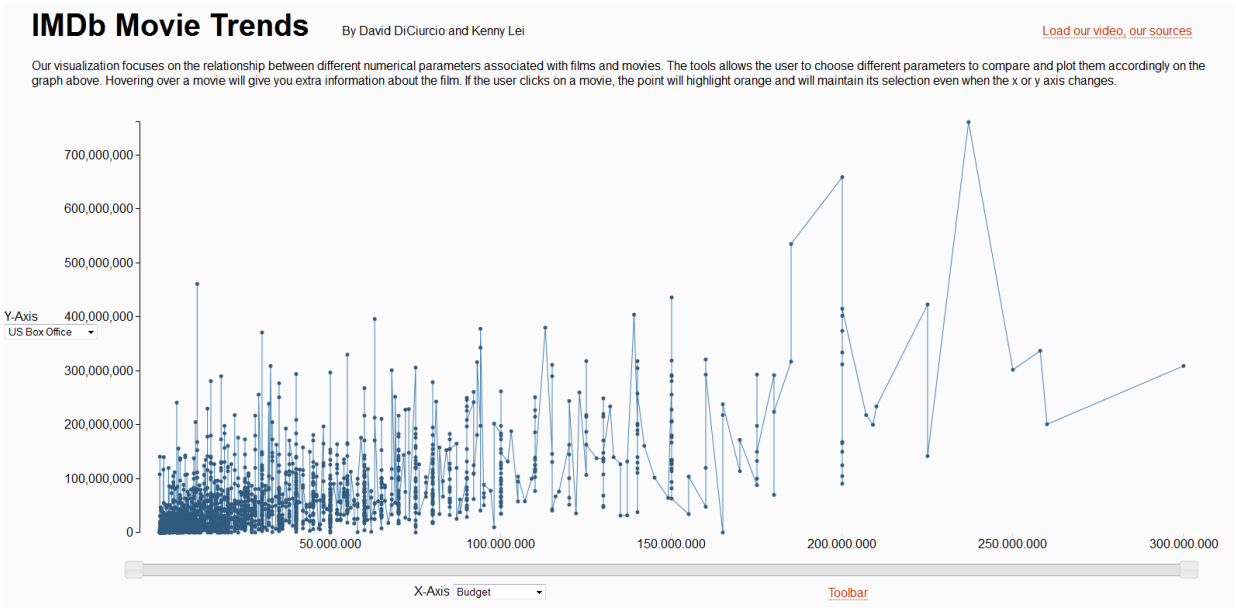
Data Cleaning

Essentially, data cleaning went similarly as to how it went the last time. But we did run into a few more interesting problems. We started the mining process going as far back as 1950 and had it run until 2010. To save time, we decide to implement a function that would prevent writing to our csv if the movie was missing values - unforutnately, this resulted in the loss of 5000 out of our 6000 movies. Of course, this was not acceptable. As it turned, the MPAA wasn't commissioned until 1968, so nothing before then had a rating, throwing out all that information. And through the rest of the data, certain values here and there were missing. Essentially, we updated our d3 to just handle blank entries by not graphing them at all.

(update)

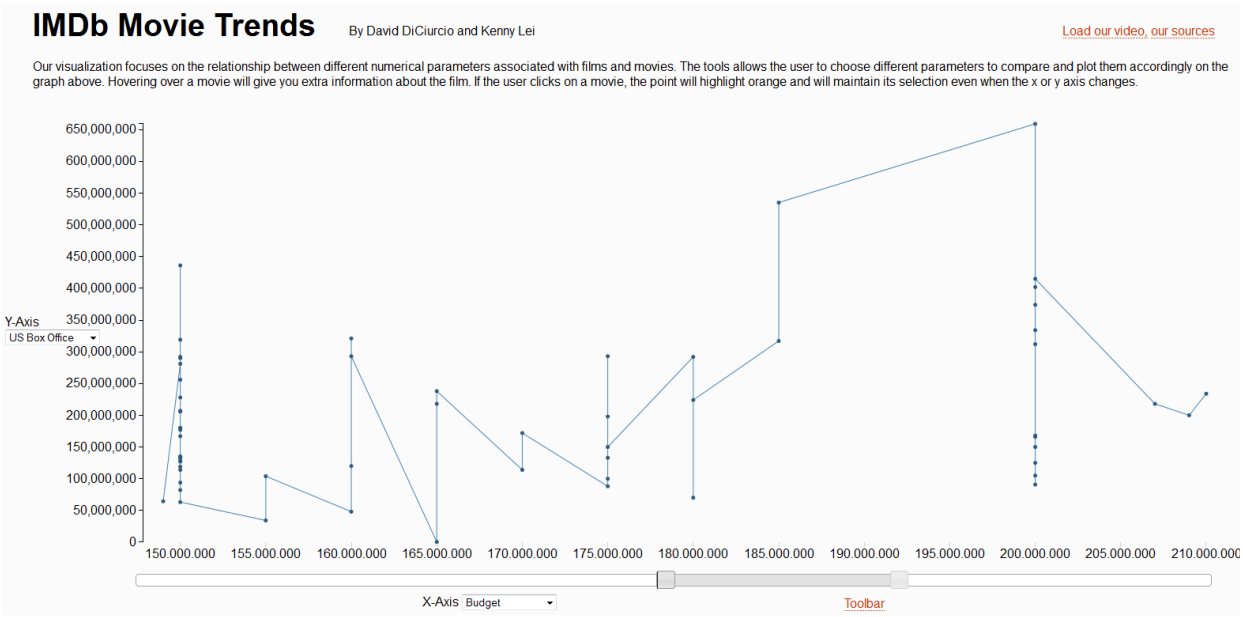
We decided to take out the 1950-1970. Although we had basic trends prepared for that era, the amount of information the code was running through was causing our website to require a ridiculous amount of memory and not run smoothly. Also, even though we added the slider bar (explained below) it was just a much more fluid experience to start off with less information. In the end, it was a joint decision to save time and just make things work better.

Slider



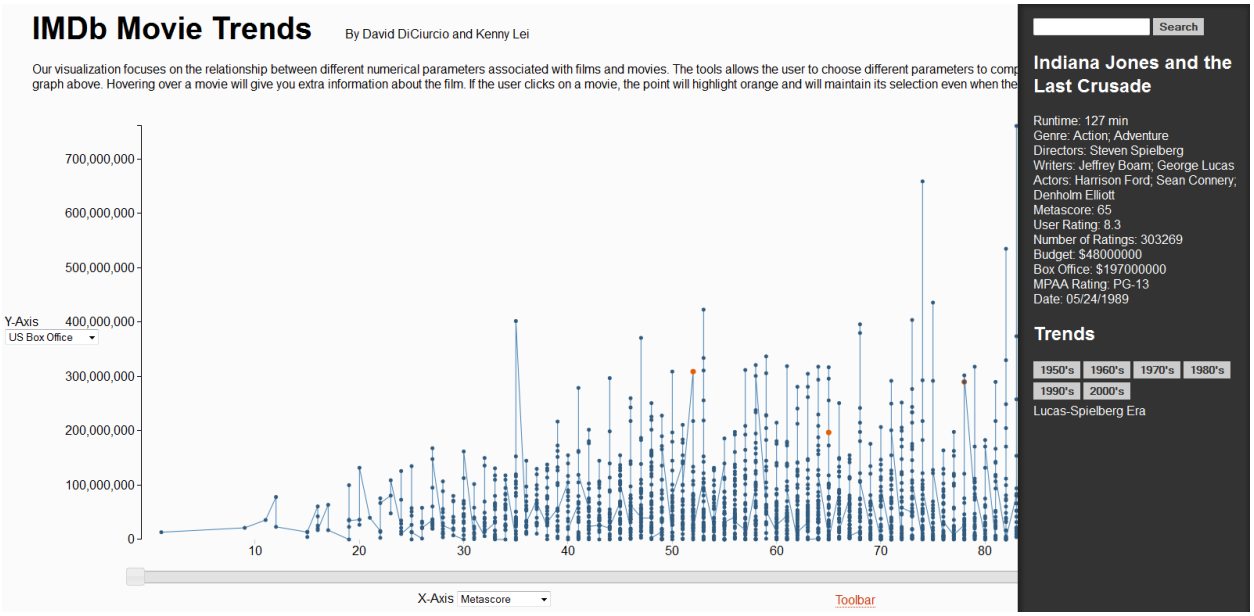
A functionality we did not have time to implement in Project II, was a slider for the x-axis. Essentially the slider helps filter the data and allows the viewer to zoom in on a portion of data in interest. For example, if we only wanted to see movies that had a budget between $150 and $200 million, we would drag the sliders and filter down the movies displayed.

This new functionality was implemented using a jQuery UI slider. The code had to be adapted to make the slider change the positioning of the line graph as well as the circle data points, essentially requiring the SVG to be redrawn every time the slider is moved. Surprisingly, the time it takes to redraw the SVG is just as smooth as changing the domain of a basic line graph. Even with the new data set that contains are six times more movies scraped from IMDb, there is very insignificant lag in the slider functionality. Unfortunately, D3 and the slider and dates never got along too well, so we disabled the slider for dates. We tried to compensate for this by having the movie trends buttons filter the dates.

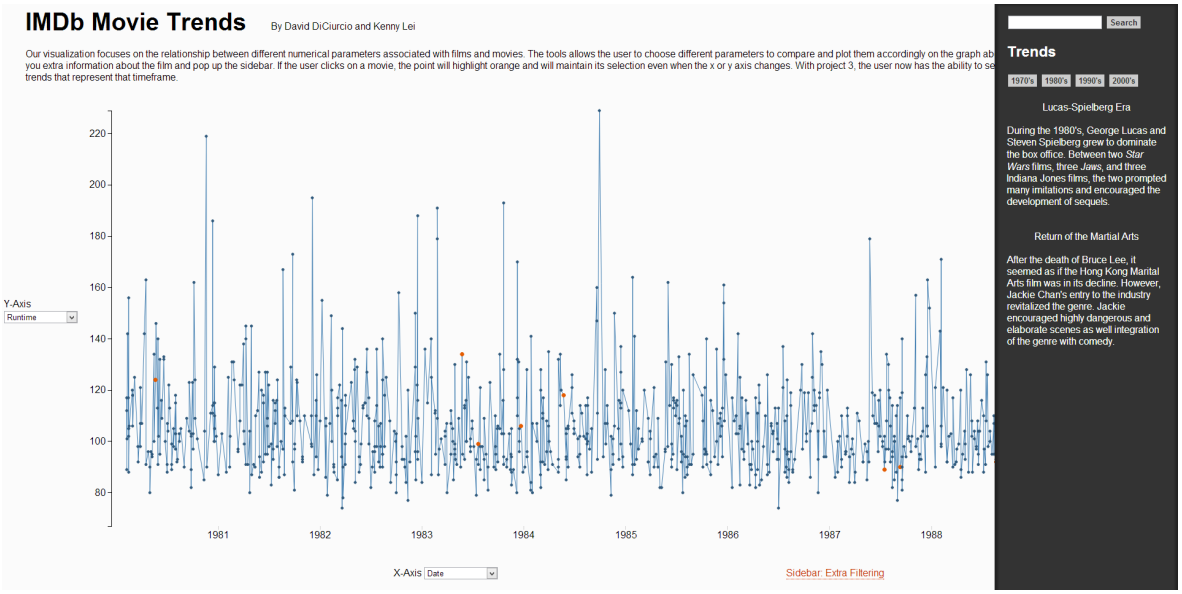

IMDb Movie Trends
By David DiCiurcio and Kenny Lei
Load our video, our sources
Our visualization focuses on the relationship between different numerical parameters associated with films and movies. The tools allows the user to choose different parameters to compare and plot them accordingly on the graph above. Hovering over a movie will give you extra information about the film. If the user clicks on a movie, the point will highlight orange and will maintain its selection even when the x or y axis changes.
650,000,000
600,000,000
550,000,000
500,000,000
450,000,000
400,000,000
350,000,000
300,000,000
250,000,000
200,000,000
150,000,000
100,000,000
50,000,000
0
Y-Axis
US Box Office
150.000.000
155.000.000
160.000.000
165.000.000
170.000.000
175.000.000
180.000.000
185.000.000
190.000.000
195.000.000
200.000.000
205.000.000
210.000.000
X-Axis
Budget
Toolbar

Sidebar

The biggest change from Project II to Project is the addition of a sidebar, which is both dynamic and nonobtrusive. The sidebar contributes to functionality in providing more in-depth information about movies that a hover-over does not provide. The sidebar is activated by either a link or clicking on a movie data point for more information. The sidebar animates in from the right-hand side and disappears by simply clicking anywhere outside of the sidebar.



To be honest, the sidebar really is one our favorite elements that we added. It looks elegant and provides a nicer way to offer the user extra information about the films they select. Similar the sidebar is the user's portal into viewing movie trends.



Quick Comment

Going over Lila's comments, I just wanted to point out why we did something and didn't do others. First off, a trend line wouldn't have been particularly useful in the grand scheme of the visualization I think. I think it would be an oversimplification of the situation, whereas just giving the user the necessary data and highlighting what is important allows the user to decide what is important. Also, we kept the dropdowns. I think this just ends up being an archaic design decision, rather than hugely telling. Perhaps with more time, we would've crafted a better selected mechanism, but it allowed us to focus on our pretty sidebar :3

Search

Unfortunately, up till the last minute, search was never working as we wanted it. Idealistically, when you search for Johny Depp, it would only shows movies that includes Johny Depp and change the slide appropriately, but it simply never functioned that way. It might have something to do with how our sidebar is technically loaded from a separate html page or the refreshing of the sidebar. We're not sure. Just we're sad that it never worked properly, so we had to take it out at the last hour on the last day.

Trends (Storytelling)

One of the most important additions to our visualization was inclusion of movie trends. Through research, we picked up some of the biggest patterns in films during each decade, and briefly summarized them for the user. We also organized the data in such a way that it was easier for the user to analyze it. For example, the visualization will highlight some examples of the trend in the visualization, making it easier to see how those movies fared in the greater context of the decade and if they want, the past 40 years. The sidebar is also intelligent enough to expand when necessary, in order to accommodate the extra information.

Describe the intent and functionality of your visualizations. What did you learn about the data?

With the IMDb Movie Trends, we are able to explore how movies have progressed and changed over the past few decades, as genres fade and decline, as well as how directors/actors come into light. With the ability to filter and zoom in on data while viewing different variables, rich data about movies can be found and extracted from this visualization that may have been tougher by sifting through raw data. This visualization brings into light the trends of movies.