

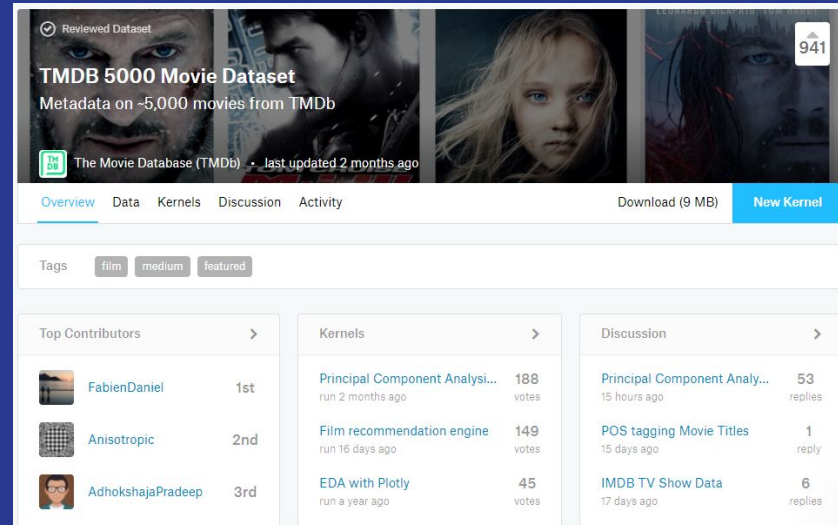
Movie Study

Jonah Hill From Moneyball



What we wanted to do

- Looked at movie data of 5000 movies (at times) from Kaggle and look for trends and correlations between gender representation of actors and movie success, as measured by revenue, and also the budget of the movie.
- Perform a sentiment analysis on the way production studios sells their movie based on positive/negative words and see if there is a correlation with the above metrics.



The screenshot shows the Kaggle dataset page for 'TMDB 5000 Movie Dataset'. The dataset is described as 'Metadata on ~5,000 movies from TMDB' and was last updated 2 months ago. It has 941 votes. The page includes tabs for Overview, Data, Kernels, Discussion, and Activity. There are tags for 'film', 'medium', and 'featured'. The 'Top Contributors' section lists FabienDaniel (1st), Anisotropic (2nd), and AdhokshajaPradeep (3rd). The 'Kernels' section lists 'Principal Component Analysis' (188 votes), 'Film recommendation engine' (149 votes), and 'EDA with Plotly' (45 votes). The 'Discussion' section lists 'Principal Component Analysis' (53 replies), 'POS tagging Movie Titles' (1 reply), and 'IMDB TV Show Data' (6 replies).

Top Contributors	Kernels	Discussion
FabienDaniel 1st	Principal Component Analysis 188 votes run 2 months ago	Principal Component Analysis 53 replies 15 hours ago
Anisotropic 2nd	Film recommendation engine 149 votes run 16 days ago	POS tagging Movie Titles 1 reply 15 days ago
AdhokshajaPradeep 3rd	EDA with Plotly 45 votes run a year ago	IMDB TV Show Data 6 replies 17 days ago

Tools

Comprehensive Data Analysis

Use what we learned about data visualization, wrangling, and text mining in particular to provide insights into this large data set provided from Kaggle.

2 Sets of Data

Credits/casting data

-Data available included movie name, cast, and crew data

-cast and crew data in JSON format

Success Metrics

-Data on budget, box office, language, ratings, title, company provided overview, “popularity index”, runtime, release date

-Also had data on genre, production, studio, and keywords in JSON format

	movie_id	title	cast	crew
1	19995	Avatar	[{"cast_id": 242, "character": "Jake Sully", "credit_i...	[{"credit_id": "52fe48009251416c750aca23", "dep...
2	285	Pirates of the Caribbean: At World's End	[{"cast_id": 4, "character": "Captain Jack Sparrow"...	[{"credit_id": "52fe4232c3a36847f800b579", "dep...
3	206647	Spectre	[{"cast_id": 1, "character": "James Bond", "credit_i...	[{"credit_id": "54805967c3a36829b5002c41", "dep...
4	49026	The Dark Knight Rises	[{"cast_id": 2, "character": "Bruce Wayne / Batma...	[{"credit_id": "52fe4781c3a36847f81398c3", "depa...
5	49529	John Carter	[{"cast_id": 5, "character": "John Carter", "credit_i...	[{"credit_id": "52fe479ac3a36847f813eaa3", "dep...
6	559	Spider-Man 3	[{"cast_id": 30, "character": "Peter Parker / Spider...	[{"credit_id": "52fe4252c3a36847f80151a5", "dep...
7	38757	Tangled	[{"cast_id": 34, "character": "Flynn Rider (voice)", ...	[{"credit_id": "52fe46db9251416c91062101", "dep...
8	99861	Avengers: Age of Ultron	[{"cast_id": 76, "character": "Tony Stark / Iron Man...	[{"credit_id": "55d5f7d4c3a3683e7e0016eb", "dep...
9	767	Harry Potter and the Half-Blood Prince	[{"cast_id": 3, "character": "Harry Potter", "credit_...	[{"credit_id": "52fe4273c3a36847f801fab1", "depa...
10	209112	Batman v Superman: Dawn of Justice	[{"cast_id": 18, "character": "Bruce Wayne / Batm...	[{"credit_id": "553bf23692514135c8002886", "dep...

```

f<- function(n,k){
  tmdb_5000_credits%>%
  slice(n)%>% #one slice at a time
  dplyr::select(cast)%>%
  map_df(fromJSON)%>% #taking text from cast and converting it to data frame
  filter(order%in% c(0:9))%>%
  head(k) #how many cast from each movie, some typos repeat order, need to get around this
  #instead use head, which gives just the first however many, since this is in same order as order anyway
  #we filter it so we only have movies with the order of 6, this means we filter the sets with at least 6
  credits, because some movies have less
}

newfun<-function(n,k){
  data <- data.frame()
  for (i in 1:n) {
    data<-rbind(data, f(i,k)) # add to the main data frame
  }
  return(data)
}

descrip<-newfun(500,5) #description of gender

g<- function(n){ #extracting the titles from the movies

  for(j in 1:n){
    datay<-tmdb_5000_movies%>%
    slice(j)%>%
    dplyr::select(title)
  }
  return(datay)
}

newg<-function(n,k){# copying the amount of titles, n movies, k characters per movie
  data<-data.frame()
  for (j in 1:n){
    data <- rbind(data,g(j))
    data[rep(seq_len(nrow(data)), each=k),]
  }
}

#titles<-newg(500,5)

tots<-cbind(titles,descrip)

```

Sample Code

Text Mining

- Used the positive negative word list and did a sentiment analysis on the overview of the movies
- Compared positive, neutral, and negative analyses with a $\log(\text{revenue})$
- Could make new list of our own words

cast
<chr>

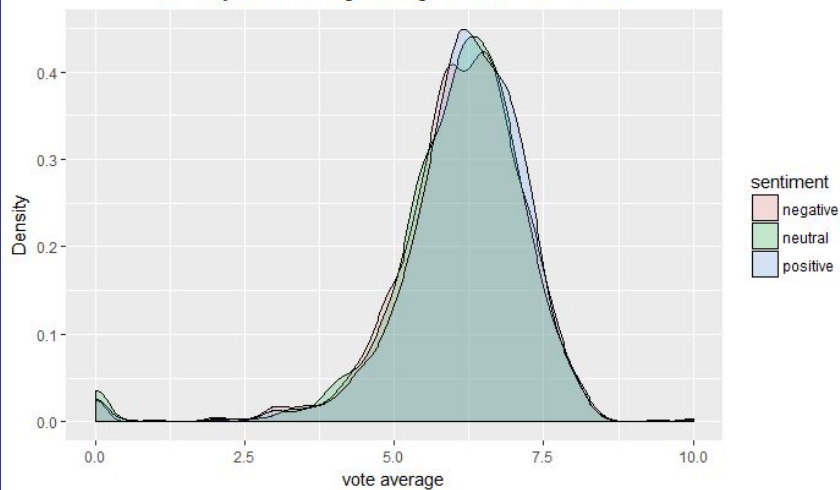
```
[{"cast_id": 242, "character": "Jake Sully", "credit_id": "5602a8a7c3a3685532001c9a", "gender": 2, "id": 65731, "name": "Sam Worthington", "order": 0}, {"cast_id": 3, "character": "Neytiri", "order": 1}, {"cast_id": 4, "character": "Captain Jack Sparrow", "credit_id": "52fe4232c3a36847f800b50d", "gender": 2, "id": 85, "name": "Johnny Depp", "order": 0}, {"cast_id": 5, "character": "Will Turn...", "order": 1}, {"cast_id": 1, "character": "James Bond", "credit_id": "52fe4d22c3a368484e1d8d6b", "gender": 2, "id": 8784, "name": "Daniel Craig", "order": 0}, {"cast_id": 14, "character": "Blofeld", "credit...", "order": 1}, {"cast_id": 2, "character": "Bruce Wayne / Batman", "credit_id": "52fe4781c3a36847f8139869", "gender": 2, "id": 3894, "name": "Christian Bale", "order": 0}, {"cast_id": 8, "character": "Alfre...", "order": 1}, {"cast_id": 5, "character": "John Carter", "credit_id": "52fe479ac3a36847f813ea75", "gender": 2, "id": 60900, "name": "Taylor Kitsch", "order": 0}, {"cast_id": 20, "character": "Dejah Thoris", "order": 1}
```



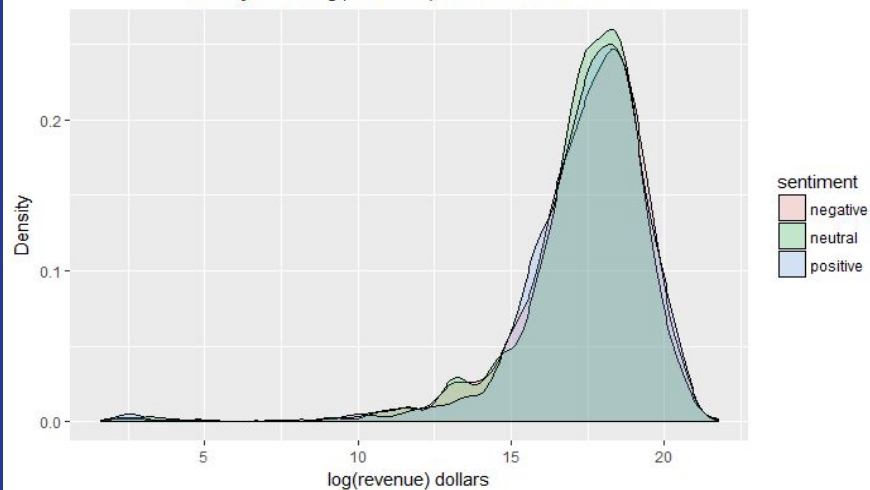
	cast_id <int>	character <chr>	credit_id <chr>	gender <int>	id <int>	name <chr>
1	242	Jake Sully	5602a8a7c3a3685532001c9a	2	65731	Sam Worthington
2	3	Neytiri	52fe48009251416c750ac9cb	1	8691	Zoe Saldana
3	25	Dr. Grace Augustine	52fe48009251416c750aca39	1	10205	Sigourney Weaver
4	4	Col. Quaritch	52fe48009251416c750ac9cf	2	32747	Stephen Lang
5	5	Trudy Chacon	52fe48009251416c750ac9d3	1	17647	Michelle Rodriguez

Sentiment analysis

Density of of average rating based on sentiment



Density of of log(revenue) based on sentiment

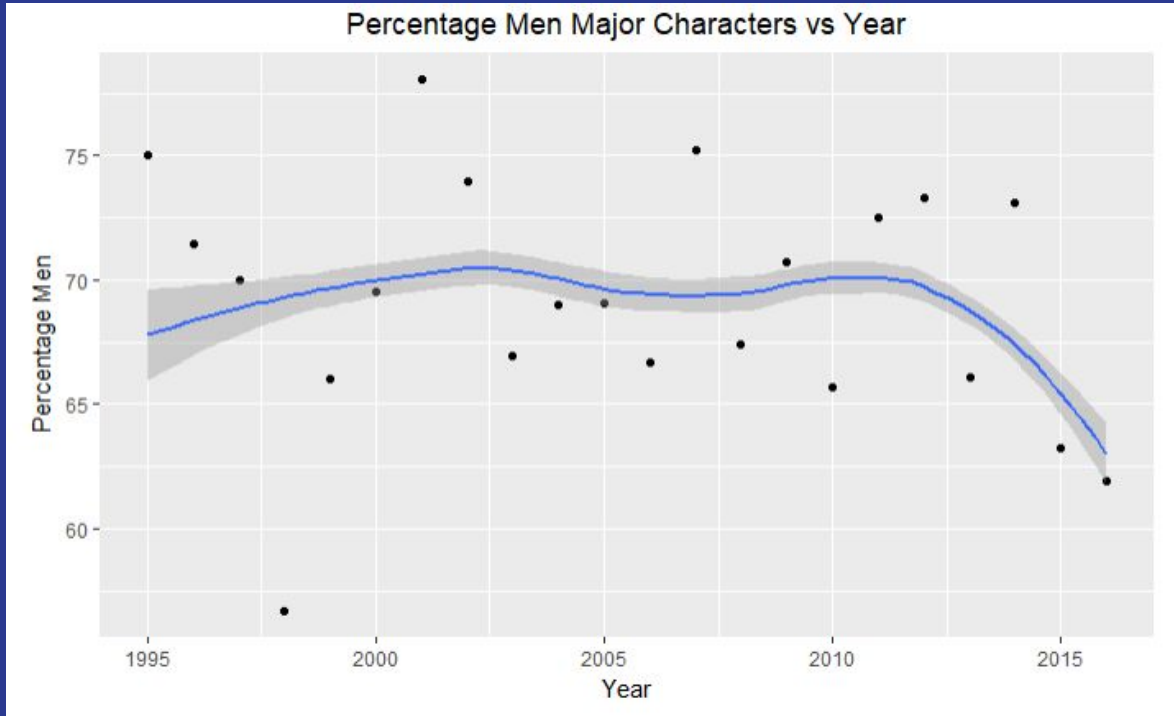


Wordcloud



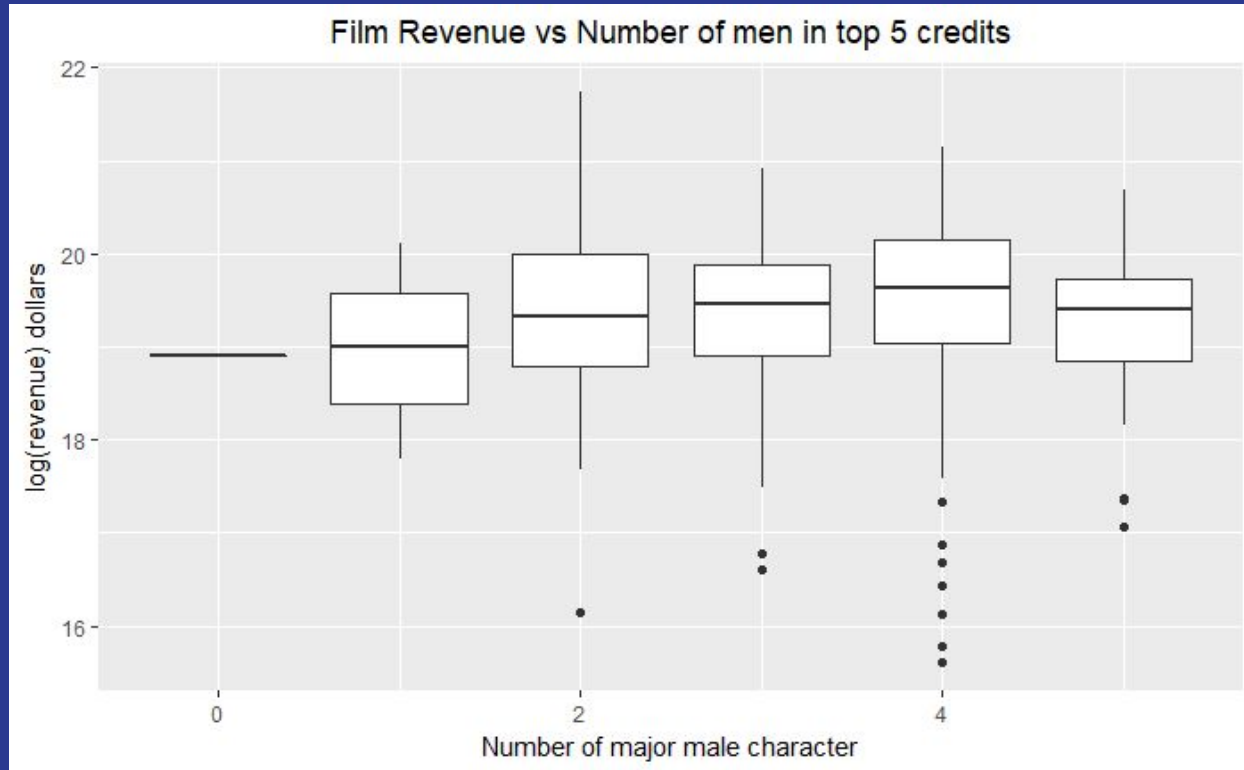
- Wordcloud of the most common words used in the overview of movies
- More or less expected

Male Major characters over time

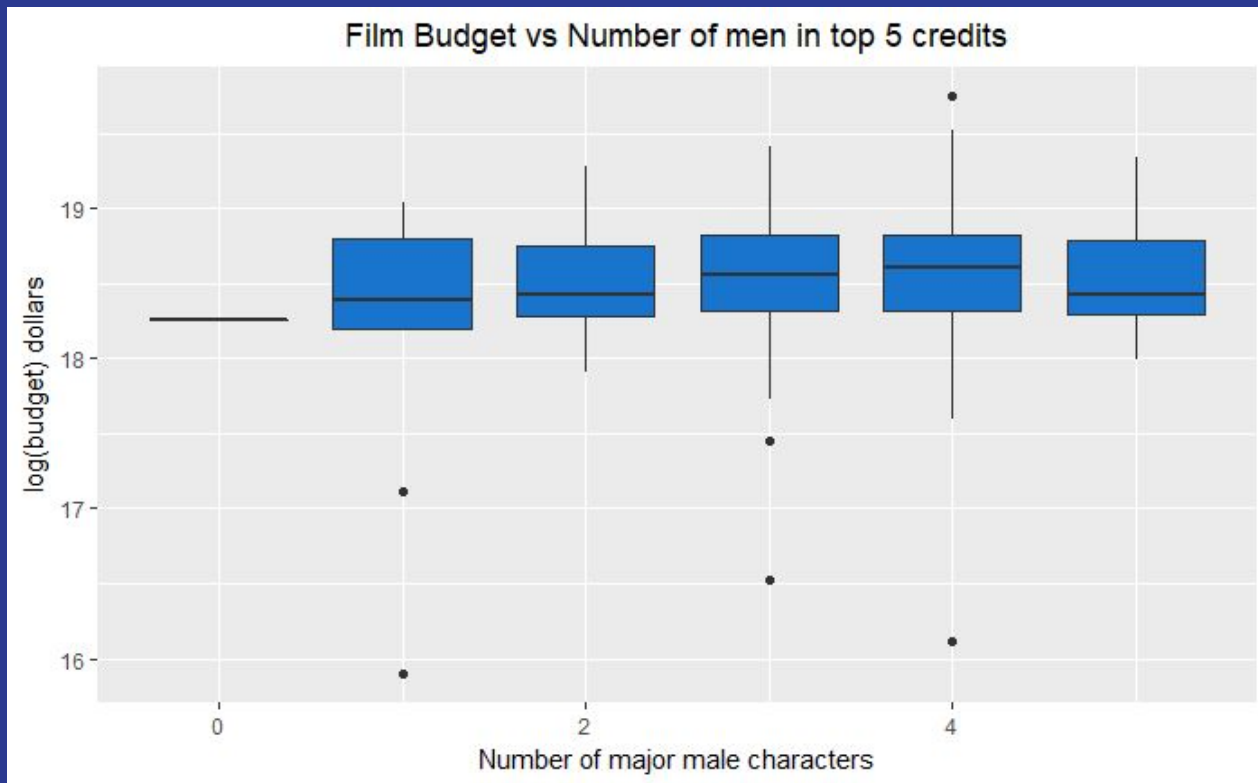


- 500 movies subset
- Mean percentage for major male characters for each year
- Interesting decline to note

Film revenue vs Number of men in top 5 credits-Notable outliers

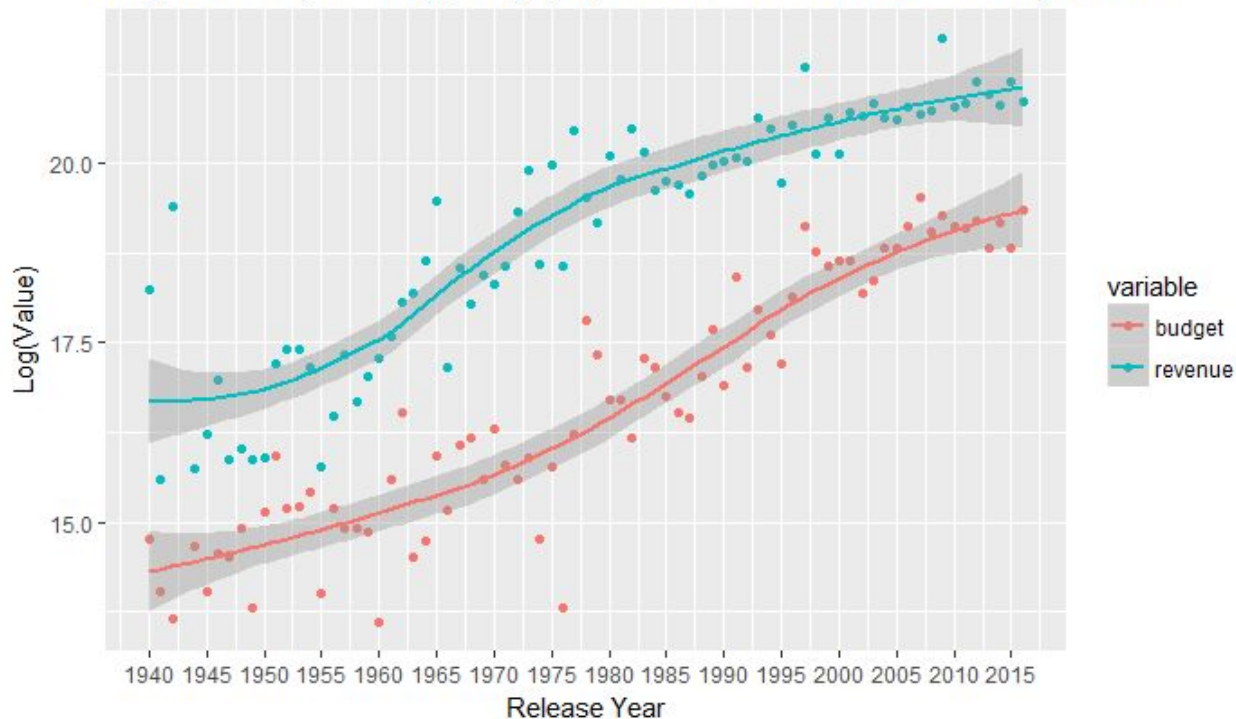


Budget

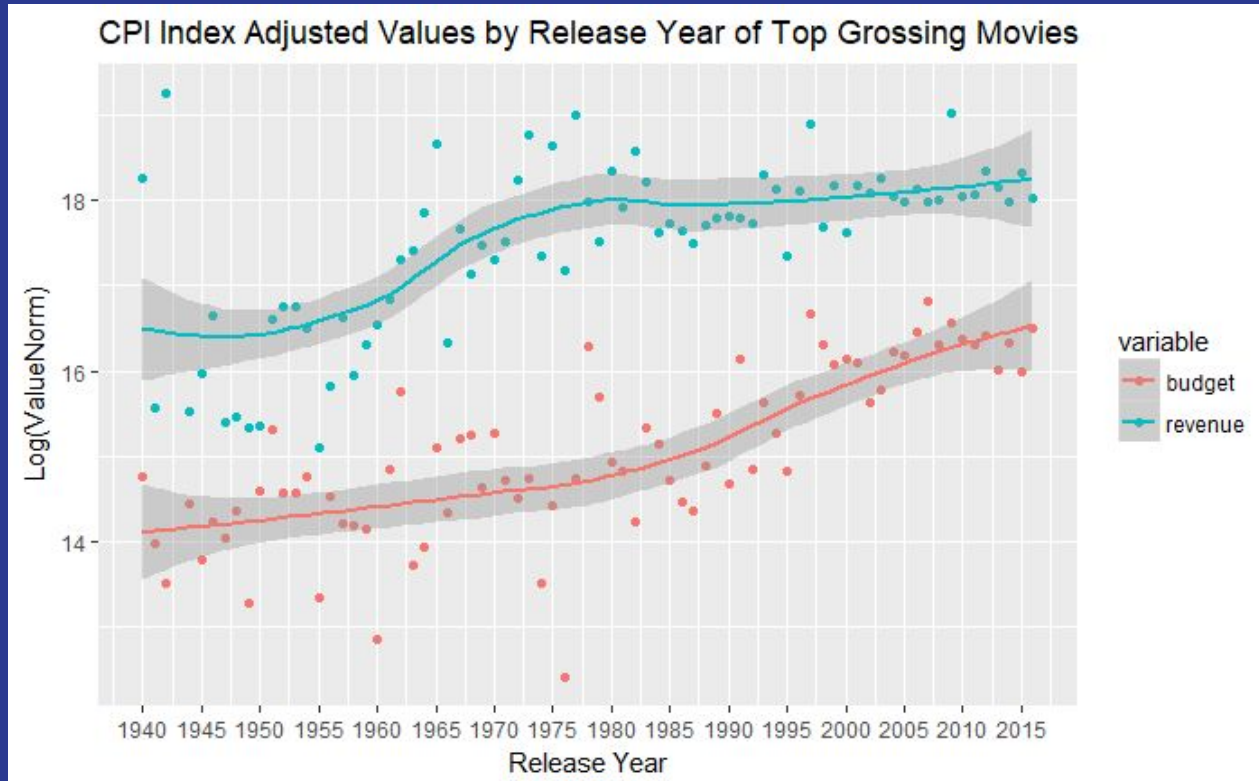


Budget and Revenue

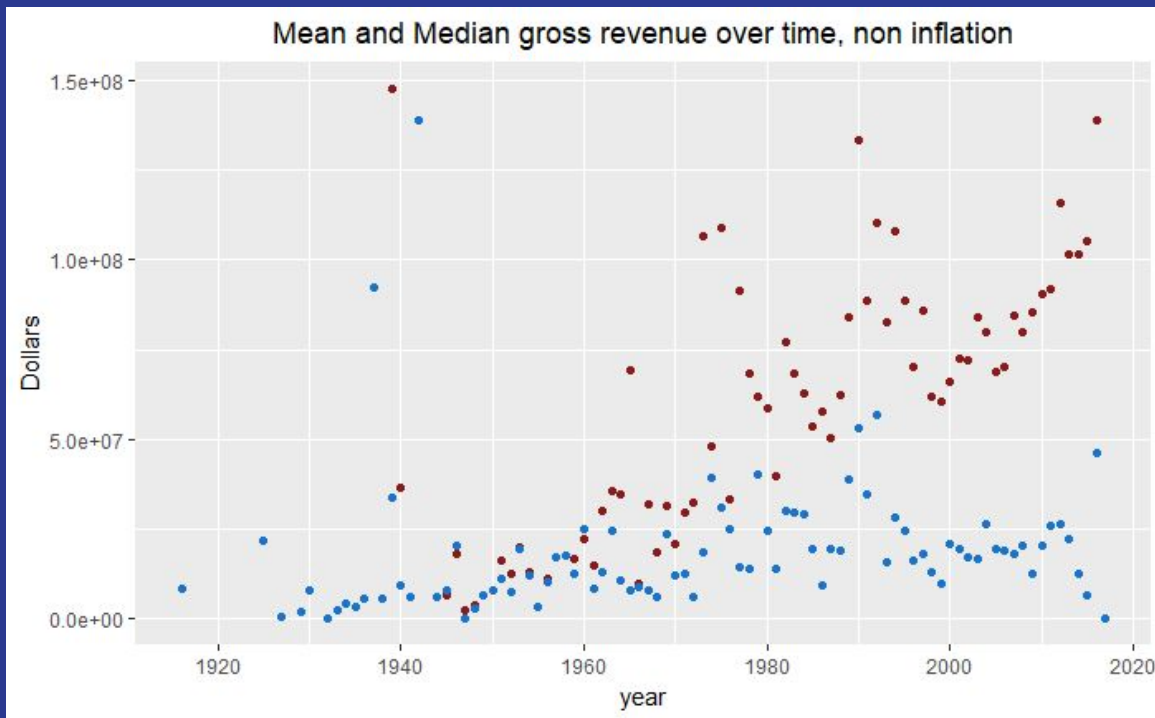
Log(Revenue) and Log(Budget) by Release Year of Top Grossing Movies



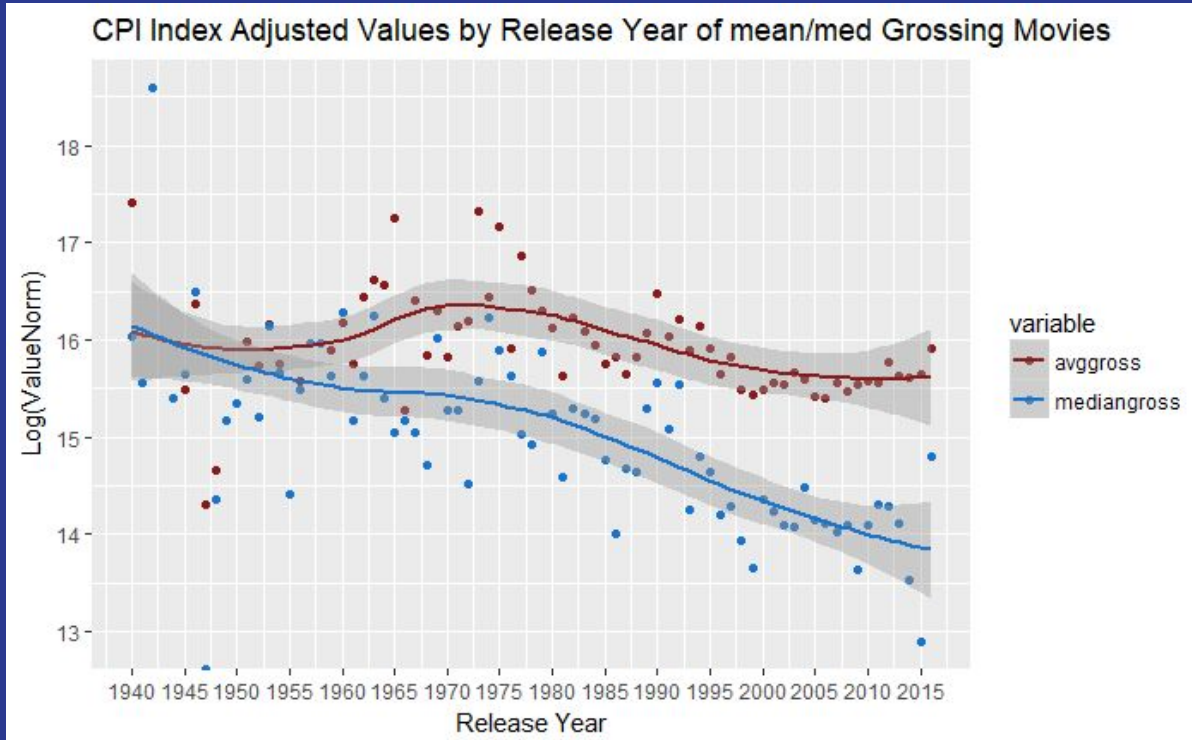
Budget and Revenue Adjusted



No adjustment



Mean and Median



- Potential issues include that we have less data for older movies
- More new “indie” movies

Regression Model

-Ran a linear multivariate regression with revenue as a dependent variable. For independent, we used budget, percentage of major characters who were men, runtime, and average rating of movie. Didn't adjust for inflation because the budget and revenue were at same time

-Mixed success.

Results

```
call:
lm(formula = revenue ~ pctmen + budget + runtime + vote_average,
    data = tots3)
```

Residuals:

	Min	1Q	Median	3Q	Max
	-665685569	-135573066	-23370857	101812988	1900222213

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	-9.351e+08	8.785e+07	-10.645	<2e-16	***
pctmen	-2.057e+07	5.703e+07	-0.361	0.719	
budget	3.203e+00	2.124e-01	15.082	<2e-16	***
runtime	6.996e+05	5.007e+05	1.397	0.163	
vote_average	1.332e+08	1.371e+07	9.714	<2e-16	***

Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

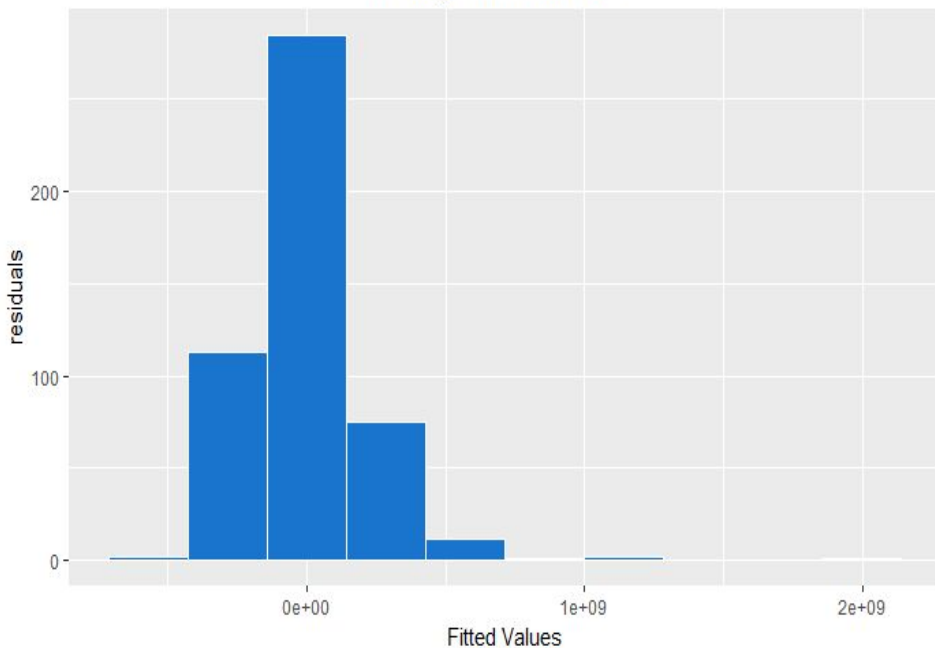
Residual standard error: 2.26e+08 on 486 degrees of freedom

Multiple R-squared: 0.4748, Adjusted R-squared: 0.4704

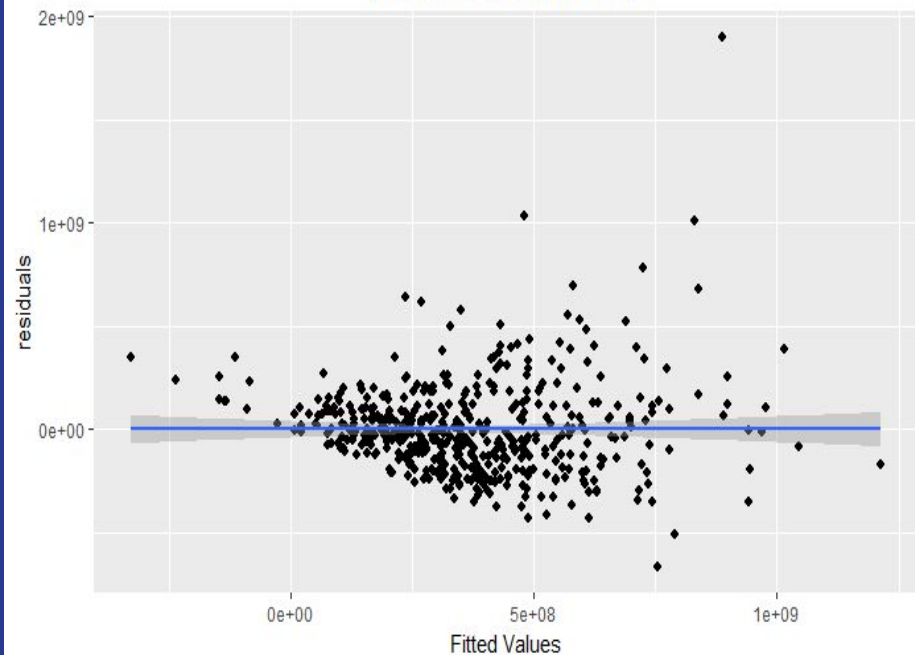
F-statistic: 109.8 on 4 and 486 DF, p-value: < 2.2e-16

Questionable Fit

Histogram of residuals



Residuals vs fitted values



Challenges

Challenge 1

The data set was big

-This was a large data set and it took a lot of run time. In part due to using for loops, so we should search for better solution

Challenge 2

Data wasn't perfect

-The data had mistakes that gravely cost us
ex) The order of importance for cast repeated terms, making it difficult to match size

Challenge 3

Working with JSON form

- Difficulties due to formatting of the data, which was somewhat foreign to us.

Example Error

```
Adding missing grouping variables: `release_date`
```

```
Error: lexical error: invalid char in json text. NA (right here) -----^
```

Errors with the json format inhibited us from performing as much analysis as we wanted on the cast list.



stackoverflow

Extremely large runtimes for this dataset also restricted us from analysis that we would have wanted to run

Conclusions

- Big data is hard to work with
- It's hard to know what is in the data
- Stack overflow is your best friend
- Top grossing movies have been rising, but the median gross of movies has fallen, more indie films?