Capstone Project Movie Ratings

HarvardX Data Science Certificate

Erik Carrion

2023-09-18

Introduction

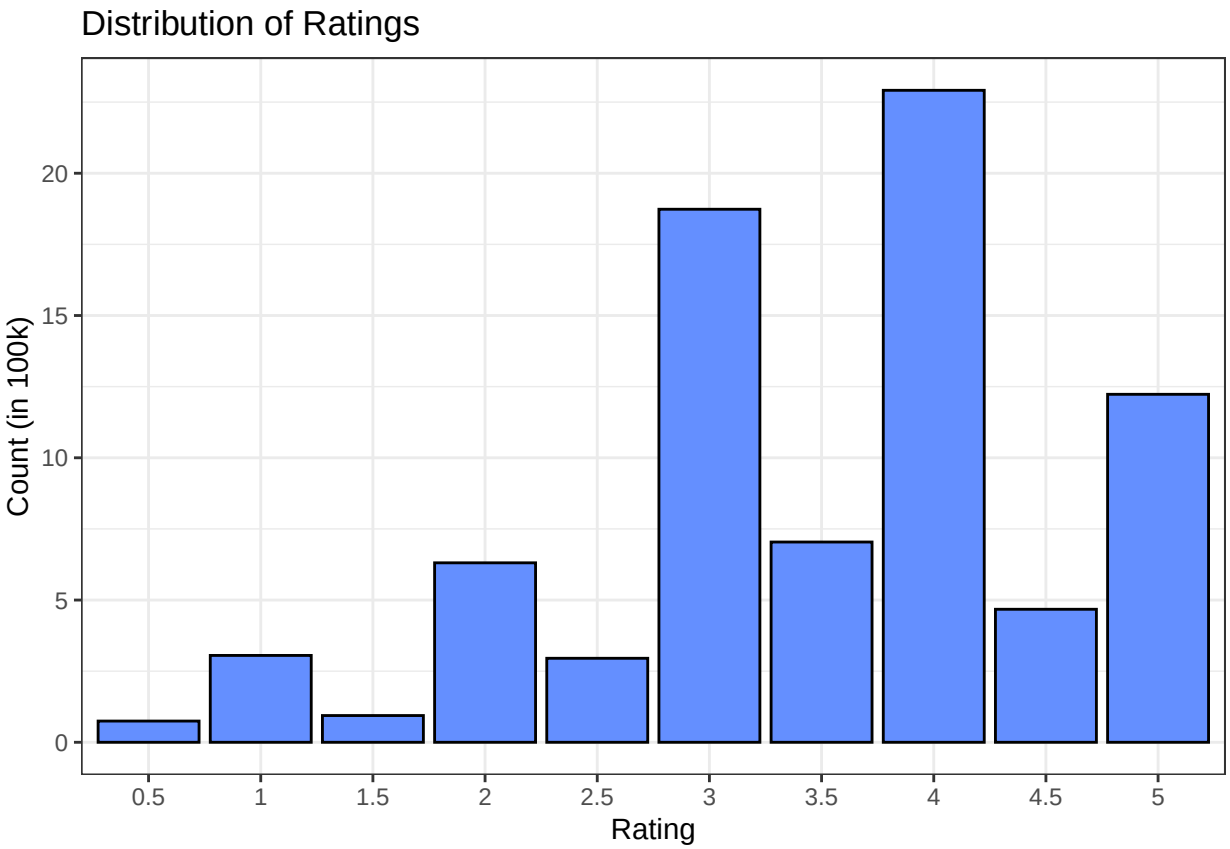
The purpose of this project is to develop a predictive model for movie ratings using the Movielens 10M dataset which has a combined 10 million observations across 12,145 unique movies and 69,892 unique user id's. Our final model takes the following form:$R = \mu_{..} + c * [b^*_m + b^*_m + f(t)] + \epsilon$. The model is composed of 4 parts: the grand mean of all ratings, $\mu_{..}$, the regularized effect for users and movies (b^*_m & b^*_u), and a smooth functions of time, $f(t)$ The 2nd half of the model, $f(t)$, is a smooth function of time. Finally, c is a scale factor s.t. $c > 0$.

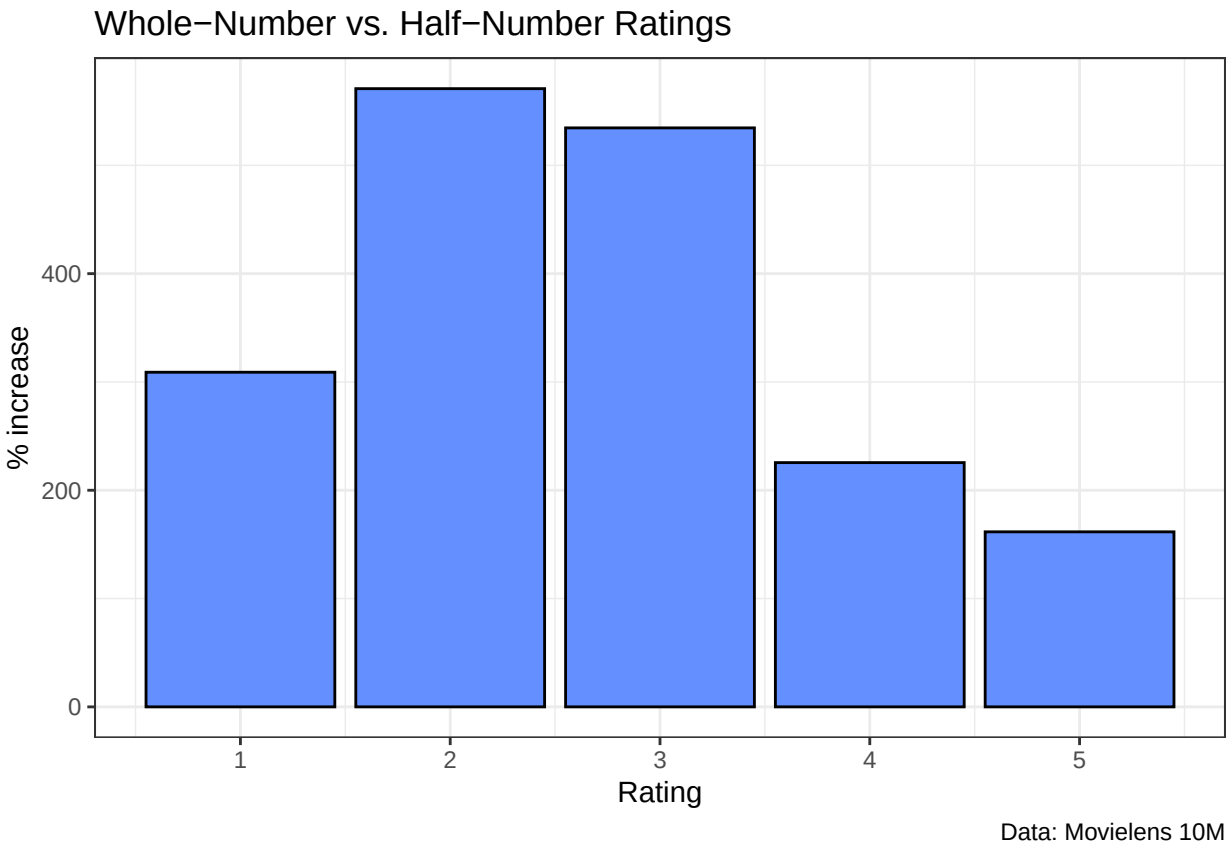
We commence our discussion with an exploratory analysis of the dataset before moving on to model development and optimization. Finally, we'll conclude with a summary of our results.

Exploratory Analysis

Rating Distribution & Summary Statistics

The distribution of the raw ratings data is asymmetric and left-tailed. In general, users appear generous in their ratings, with 82% of ratings being a 3 or better. Whole number ratings are far more prevalent, outnumbering half-number ratings 3.8:1. The magnitude of this imbalance is visualized below. We see there are 309% more movies rated a 1 than a 0.5 and 568% more movies rated a 2 than a 1.5.


Distribution of Ratings
Count (in 100k)
0
5
10
15
20
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Rating

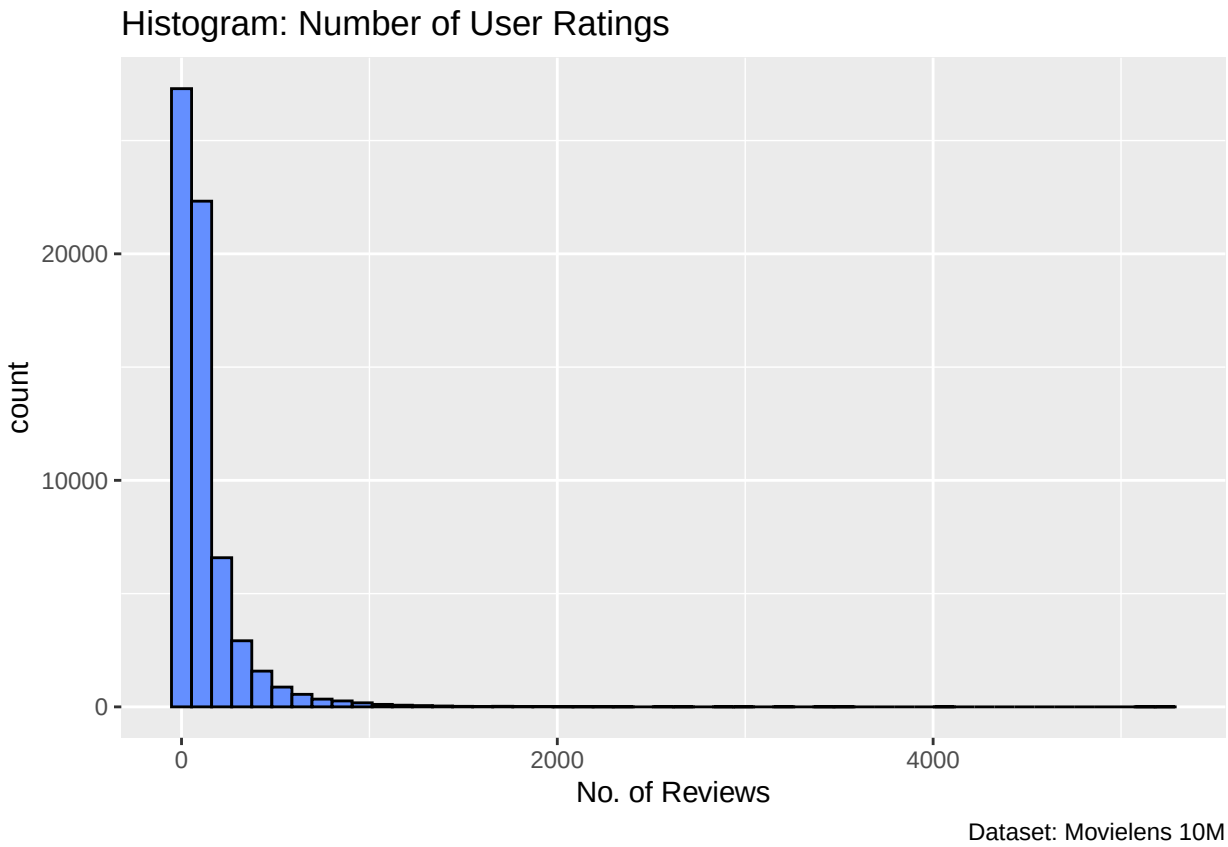


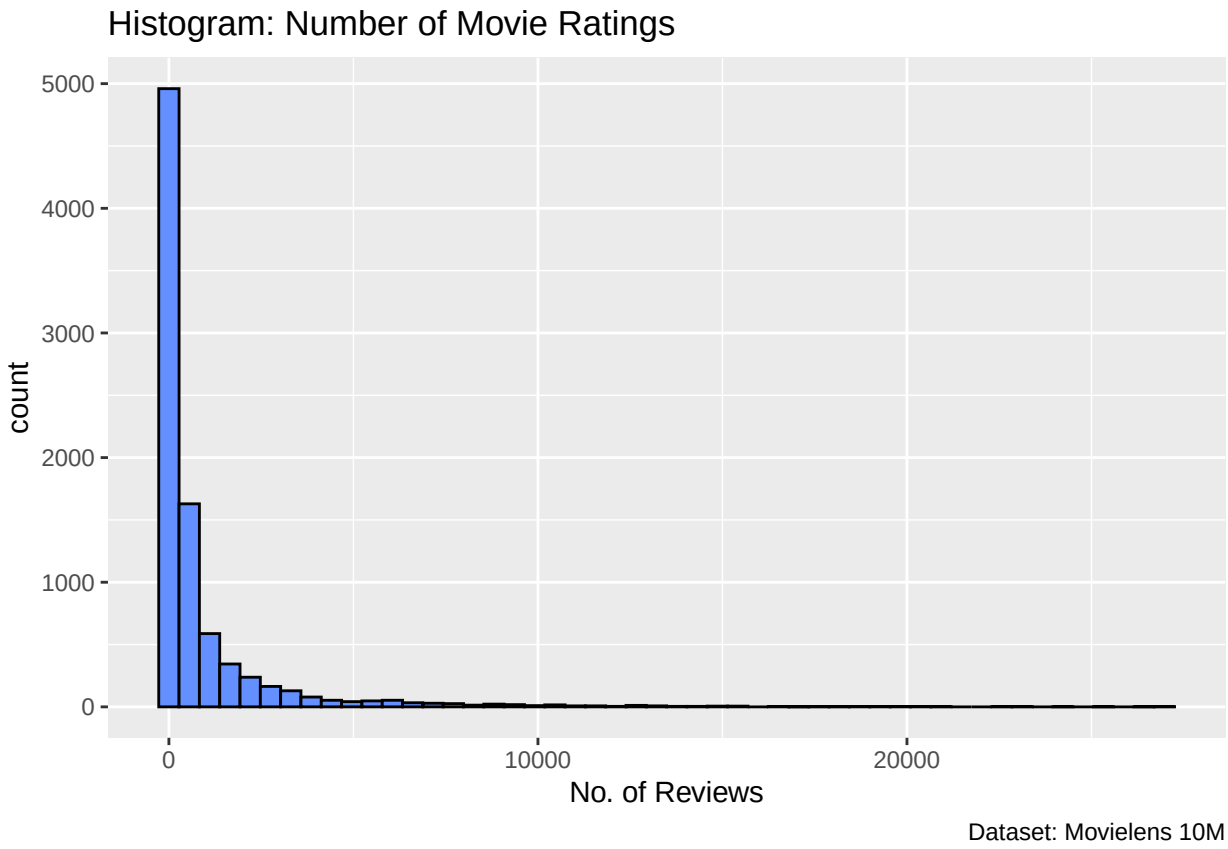

```
## Percentage of movies rated 3 or better:       0.86

## Ratio of whole-number to half-number ratings: 3.87
```

Distribution of Reviews - Users & Movies

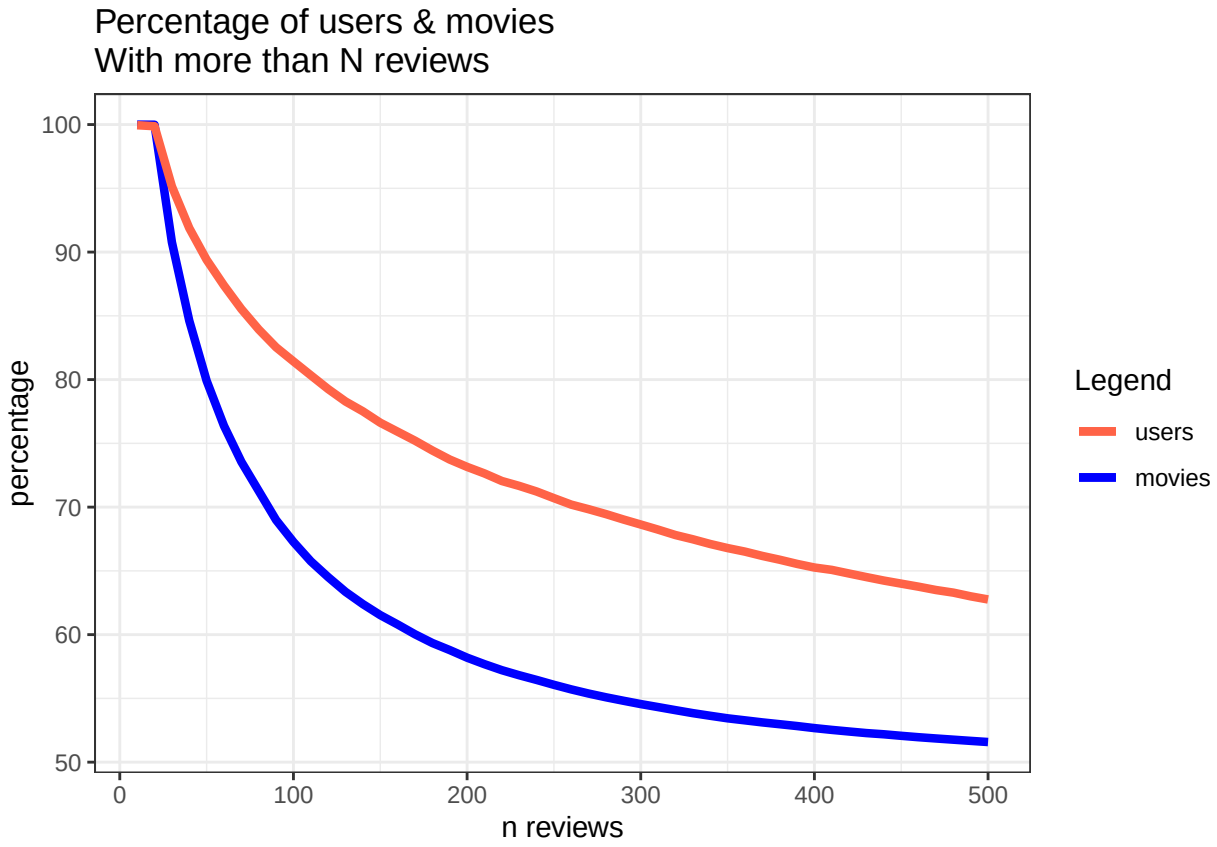
Both users and movies exhibit highly right tailed distributions. Lots of users and movies have relatively few reviews while very few users and movies have significant numbers of reviews.


Histogram: Number of User Ratings
count
20000
10000
0
0
2000
4000
No. of Reviews
Dataset: Movielens 10M



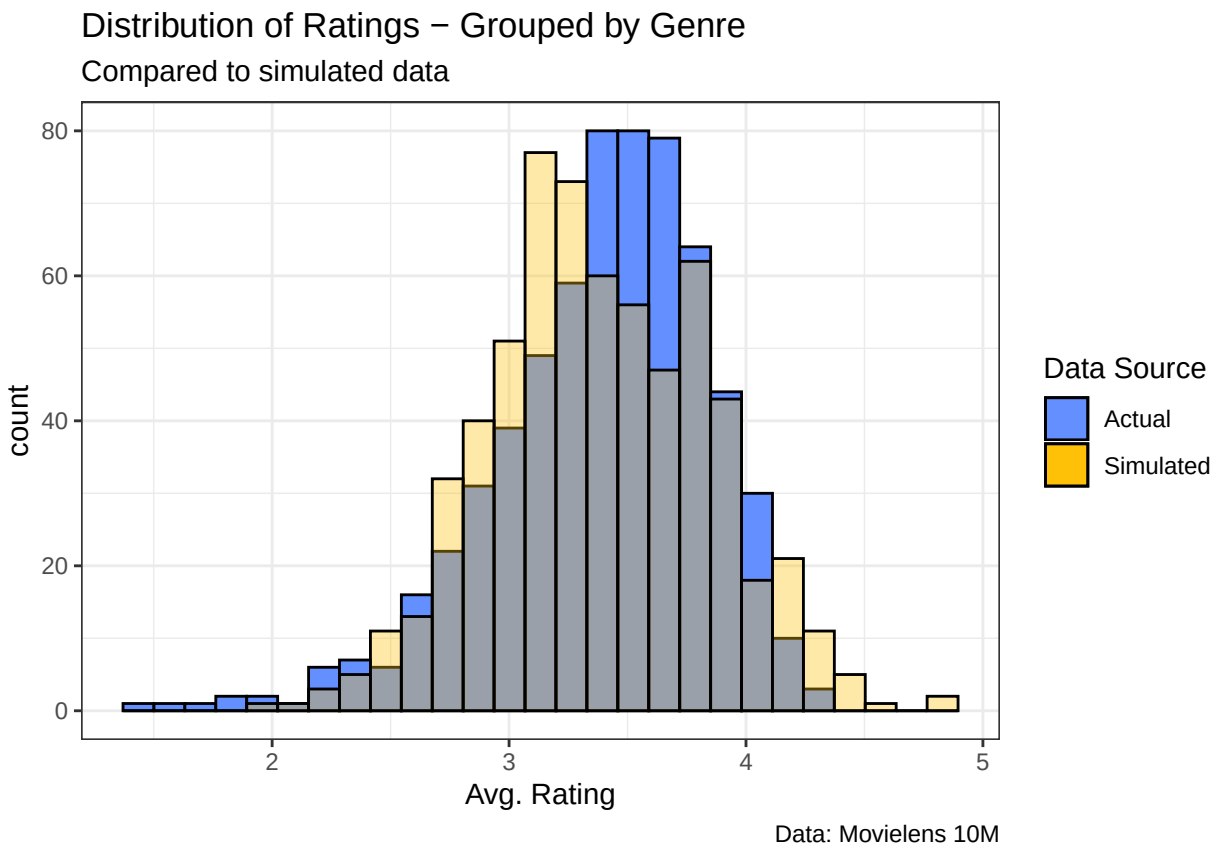
For our model to be accurate, we want each user and movie to have enough ratings in order to make an accurate estimate of their respective effects. If a user only has 2 observations, we're unlikely to make a good estimate of his or her effect. Likewise for movies.

Requiring a minimum number of observations has to balanced against data loss. As we increase the minimum number of reviews, we see a sharp loss in both users and movies from the dataset. These lost users and movies make up an ever growing proportion of the training set. We therefore have to balance our minimum number of reviews against user, movie, and data loss.



Genres

After limiting genres to a maximum of 4 genres, there are 684 distinct genres. In comparison to a normal distribution, average ratings by genre are left skewed, peaks higher, and is more tightly clustered about its mean than we expect under normality.



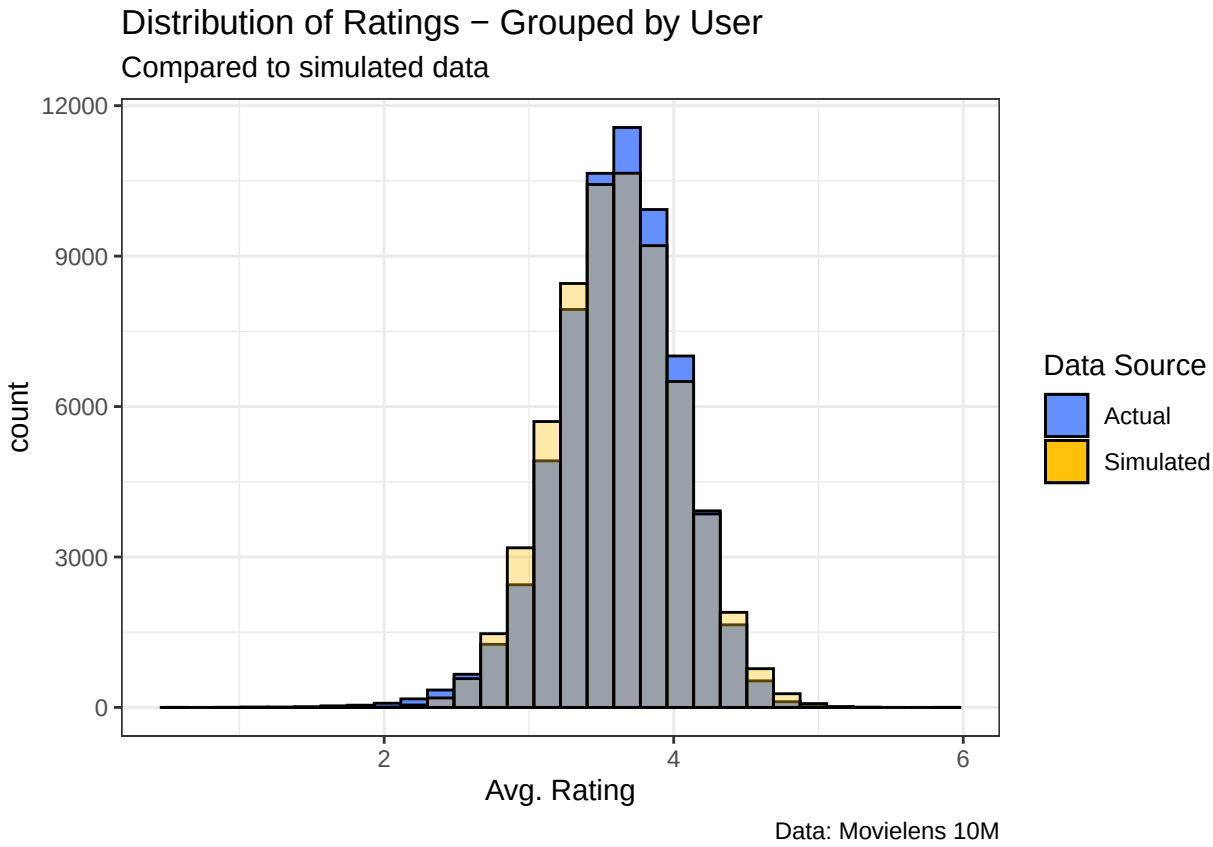
Further, we see a longer and taller left tail and a slimmer and slightly shorter right tail. This leads us to the ratings themselves and the worst and best genres. Looking at the bottom 5, most have an element of horror while in the top 5 animation, action, and adventure are prevalent.

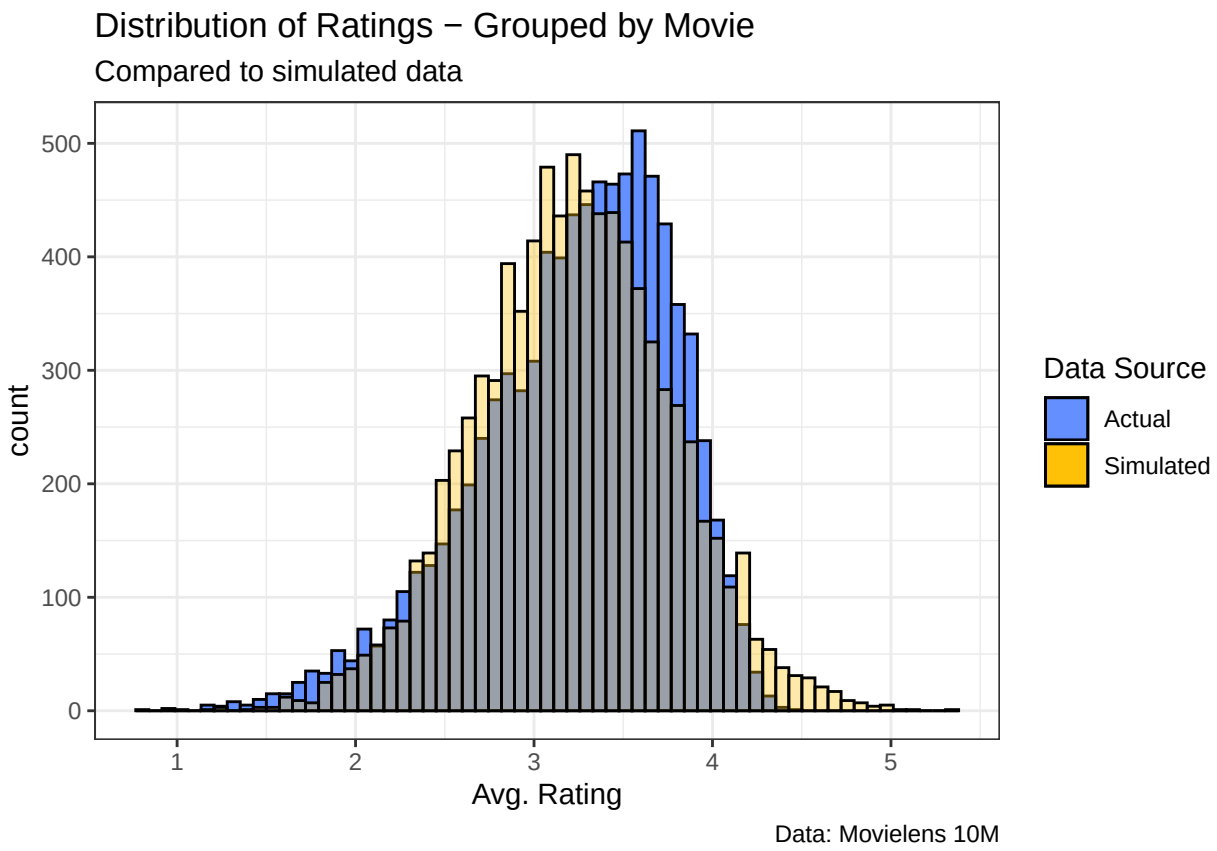
```
## # A tibble: 5 x 2
##   genre                           rating
##   <chr>                            <dbl>
## 1 Documentary|Horror                1.48
## 2 Action|Horror|Mystery|Thriller    1.60
## 3 Adventure|Drama|Horror|Sci-Fi     1.73
## 4 Action|Children|Comedy            1.87
## 5 Action|Adventure|Children         1.89
```

```
## # A tibble: 5 x 2
##   genre                             rating
##   <chr>                              <dbl>
## 1 Drama|Film-Noir|Romance             4.31
## 2 Action|Crime|Drama|IMAX             4.29
## 3 Animation|Children|Comedy|Crime     4.28
## 4 Film-Noir|Mystery                   4.24
## 5 Film-Noir|Romance|Thriller          4.23
```

Users & Movies

The distribution of average ratings for users and movies is approximately normal with the distribution for users more so than for movies. The distribution of mean rating by movieID bears similarities to that for genre.



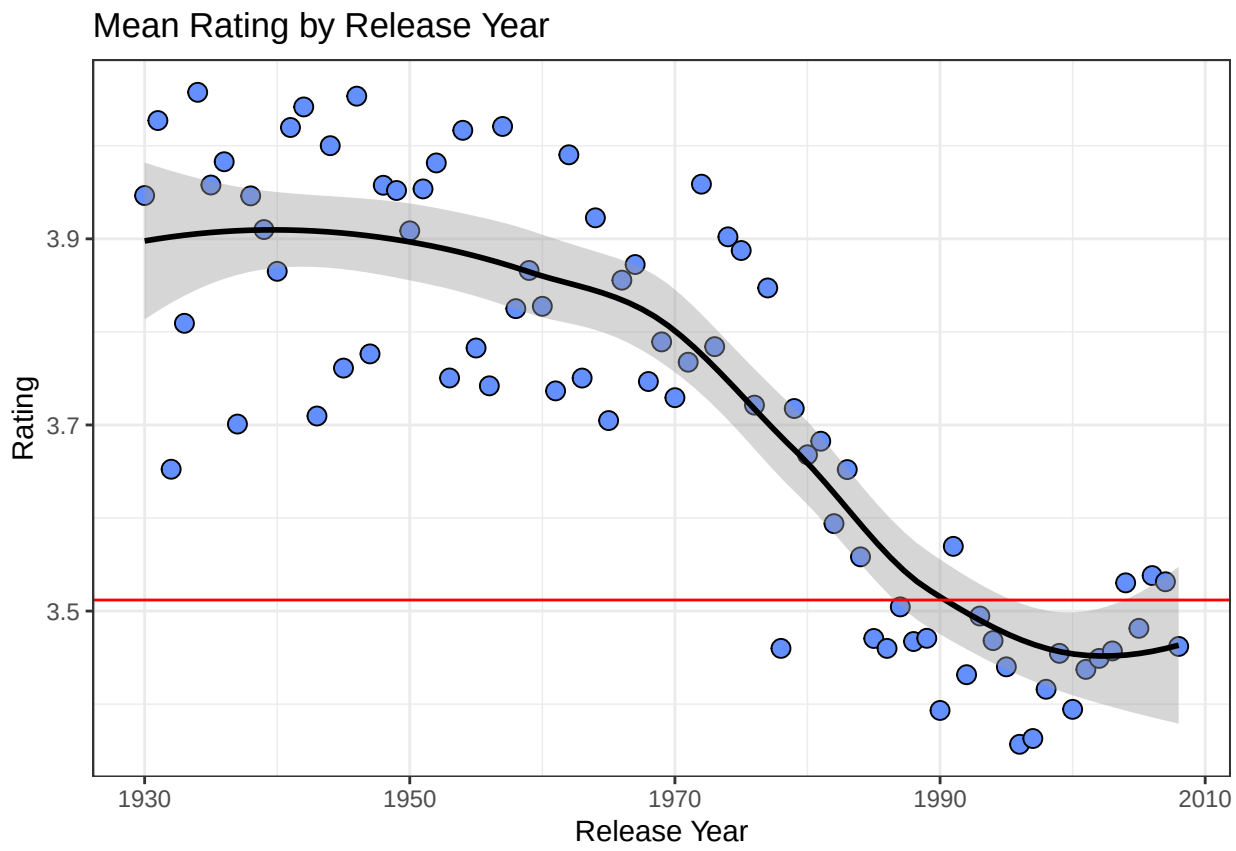


Ratings Over Time

Aside from release year, we can extract the year, month, day, week, and hour of review from the timestamp, allowing us to consider ratings from across a number of different facets.

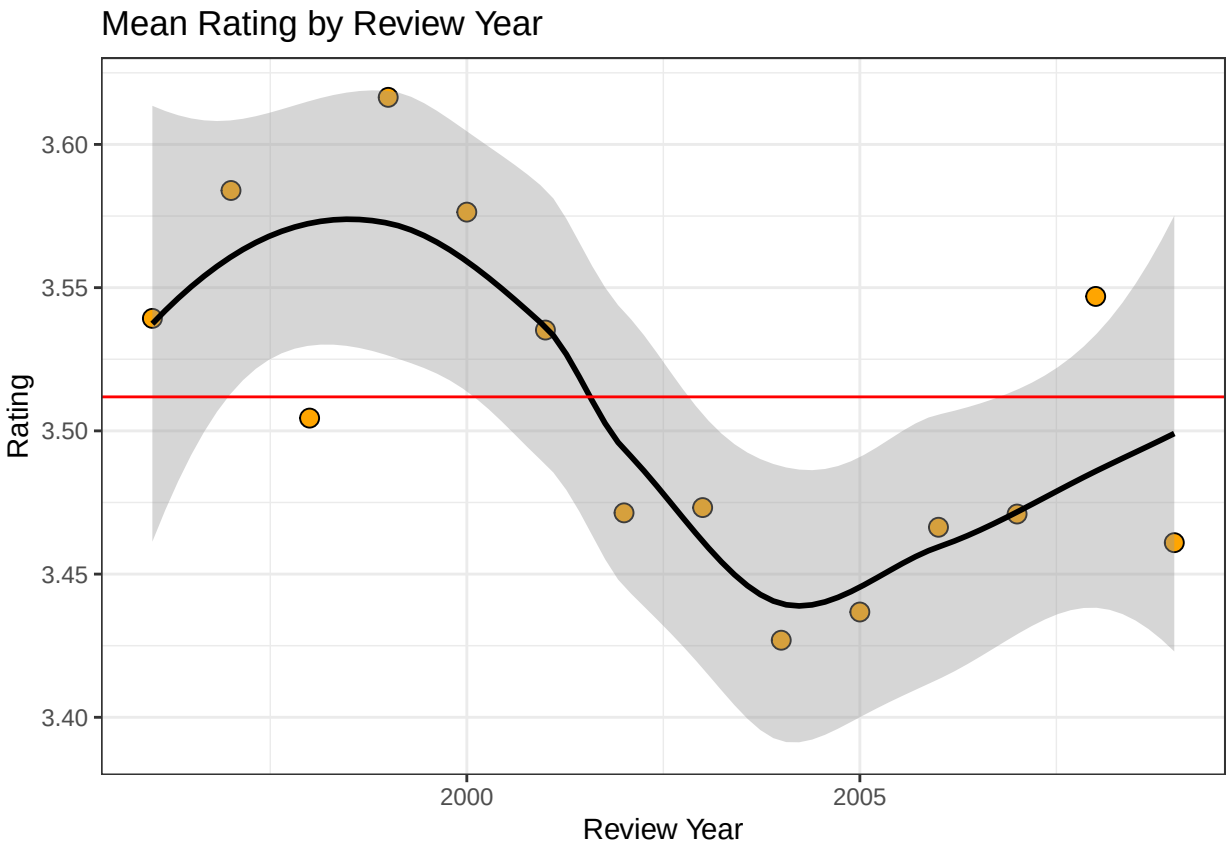
Time Effect: Release Year

First, we consider how the mean rating changes by release year. Immediately, we see a clear non-linear relationship between the average rating and the year of release. We see that for movies made prior to 1990, ratings were generally above the mean, whereas afterwards we see a decline. Does this imply that movies released before 1990 were really that much better?



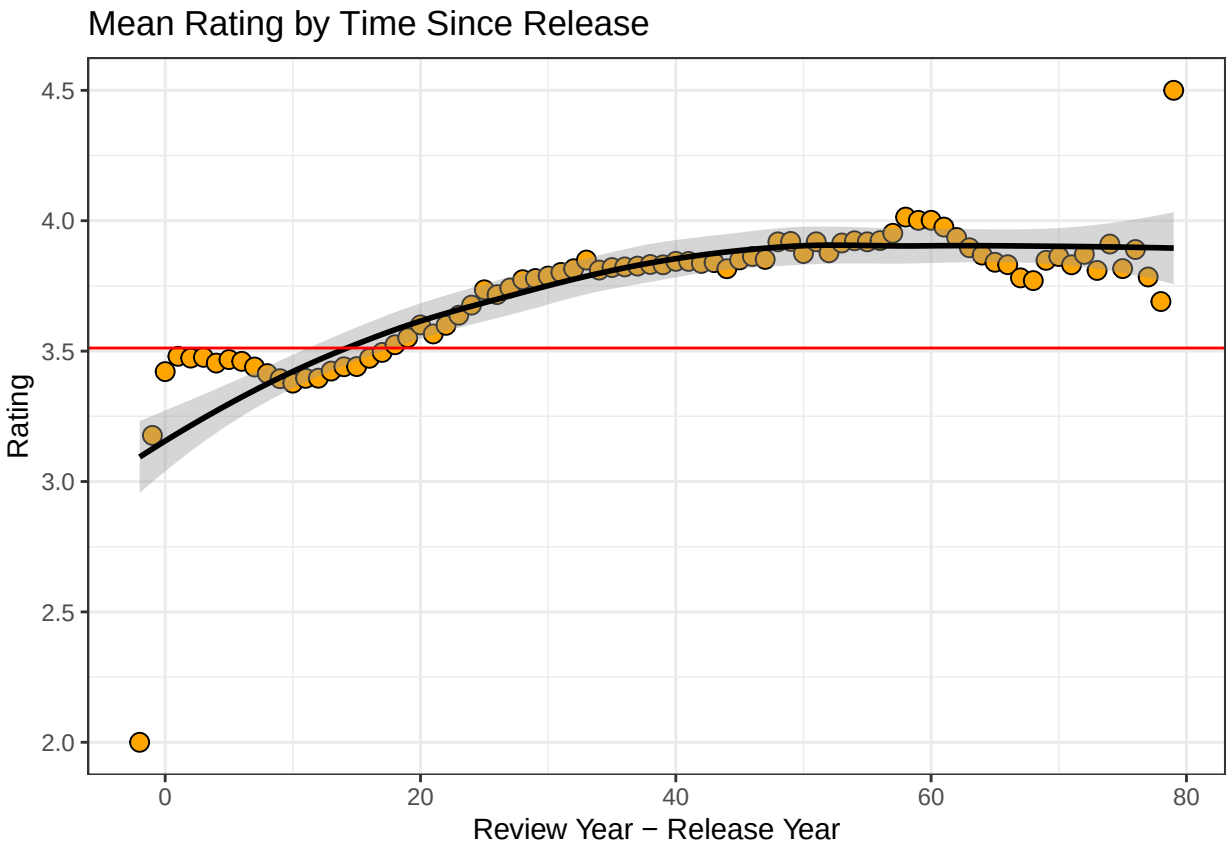
Time Effect: Review Year

Moving on to year of review, after we filter out the 1 review entered in 1995, we see the following relationship. Again, there is clearly a non-linear relationship between the average rating and the year the rating was submitted.



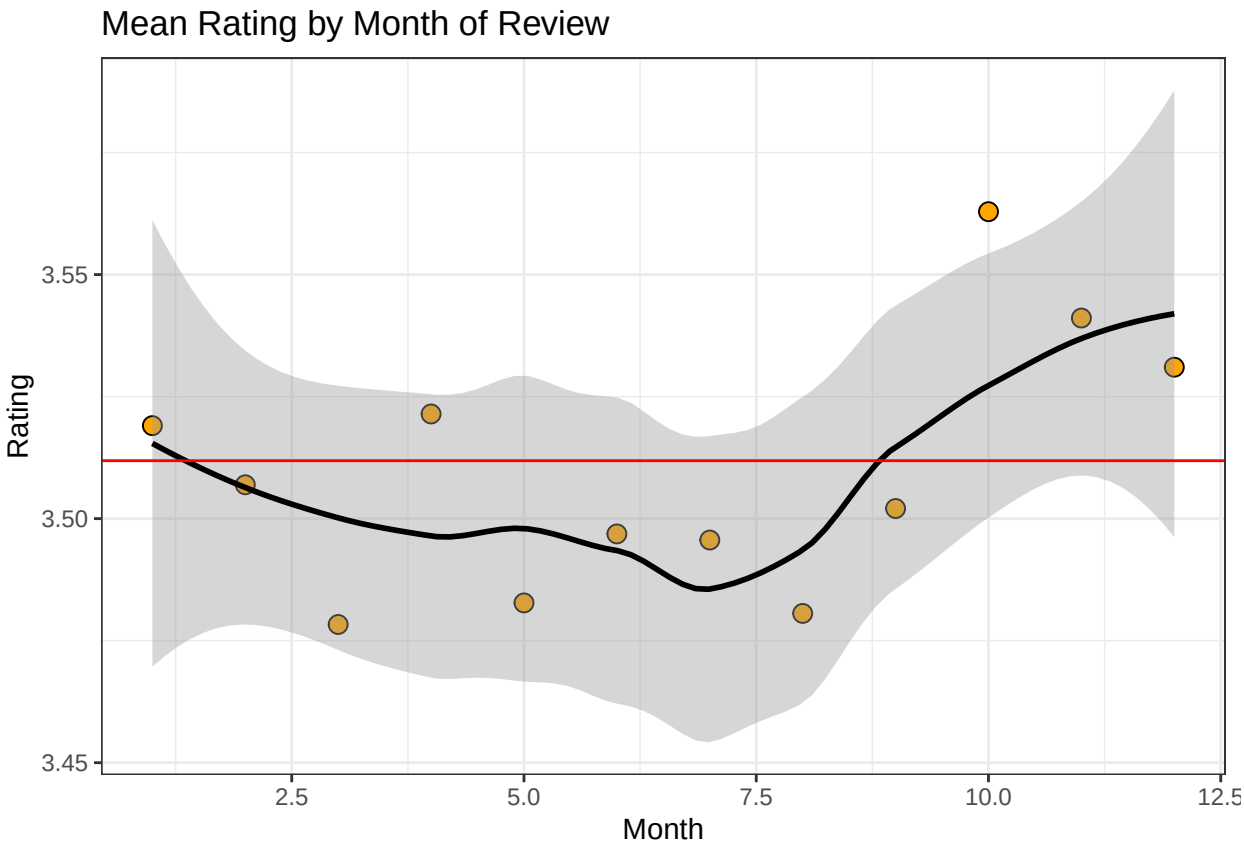
Time Effect: Years Since Release

Differences play an important role in analysis. In our case, we can ask if a pattern emerges when we consider the difference in time between a movie's release and a rating's submission. We see there is a parabolic effect with the average rating peaking between 50 and 60 years from the year of release.



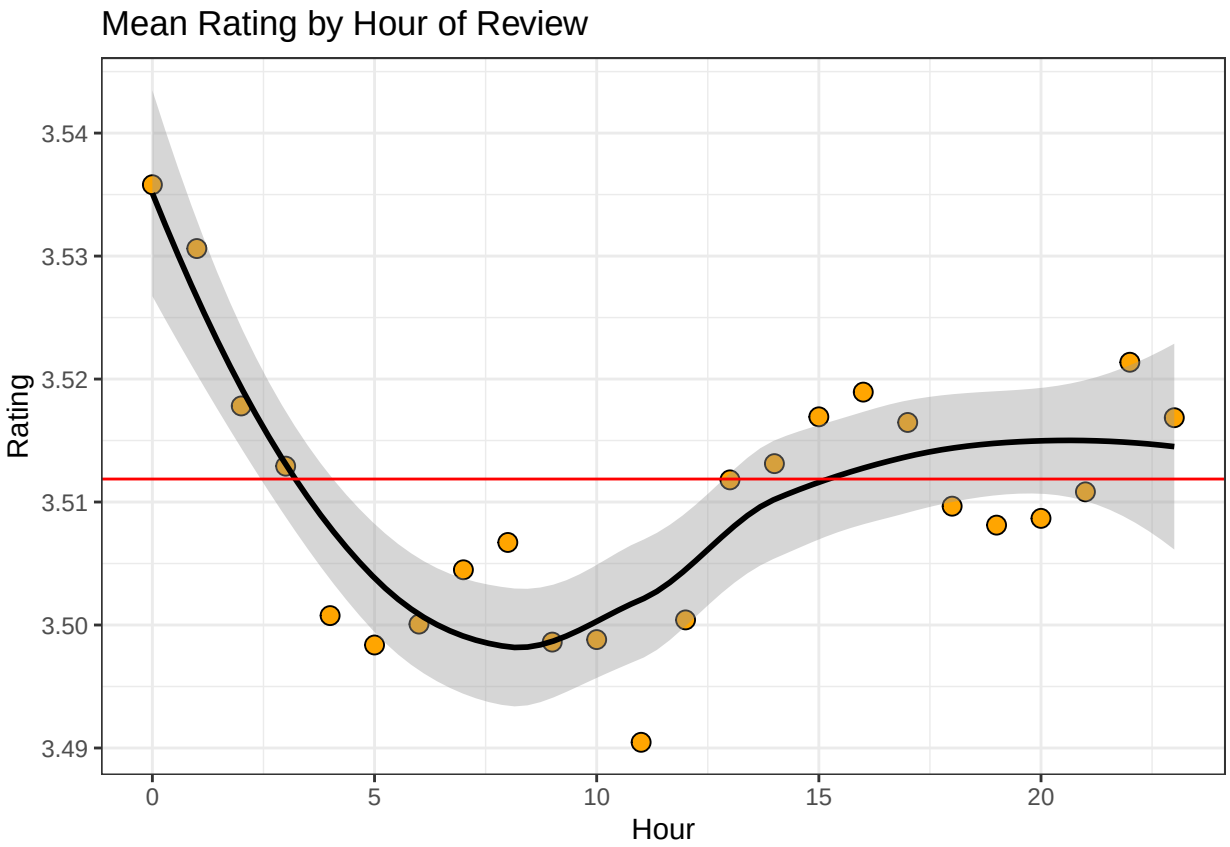
Time Effect: Review Month

When it comes to the month a review was made, there seems to be a non-linear effect present, but the confidence bands suggest we can't conclude it's significant.



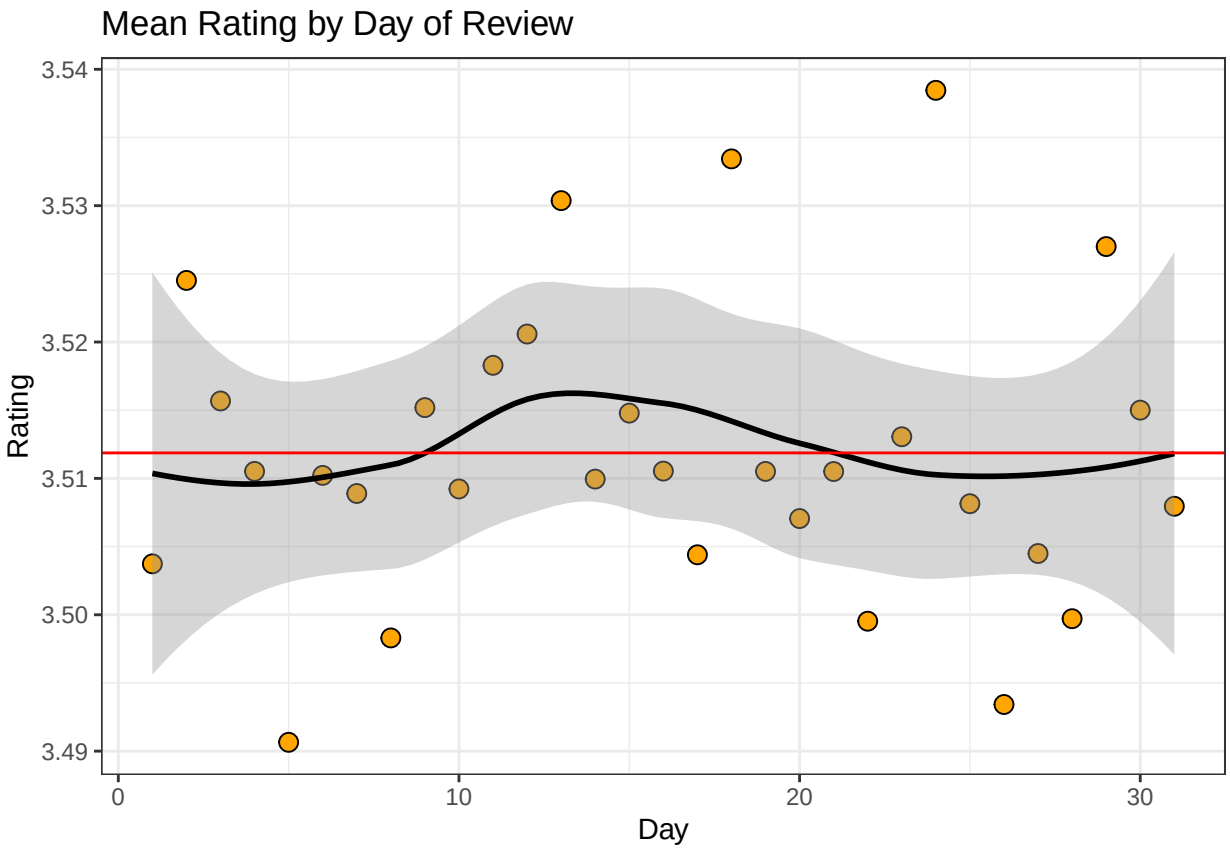
Time Effect: Review Hour

The hour a review is submitted displays a non-linear relationship with confidence bands that suggest it's effect is significantly different from 0.



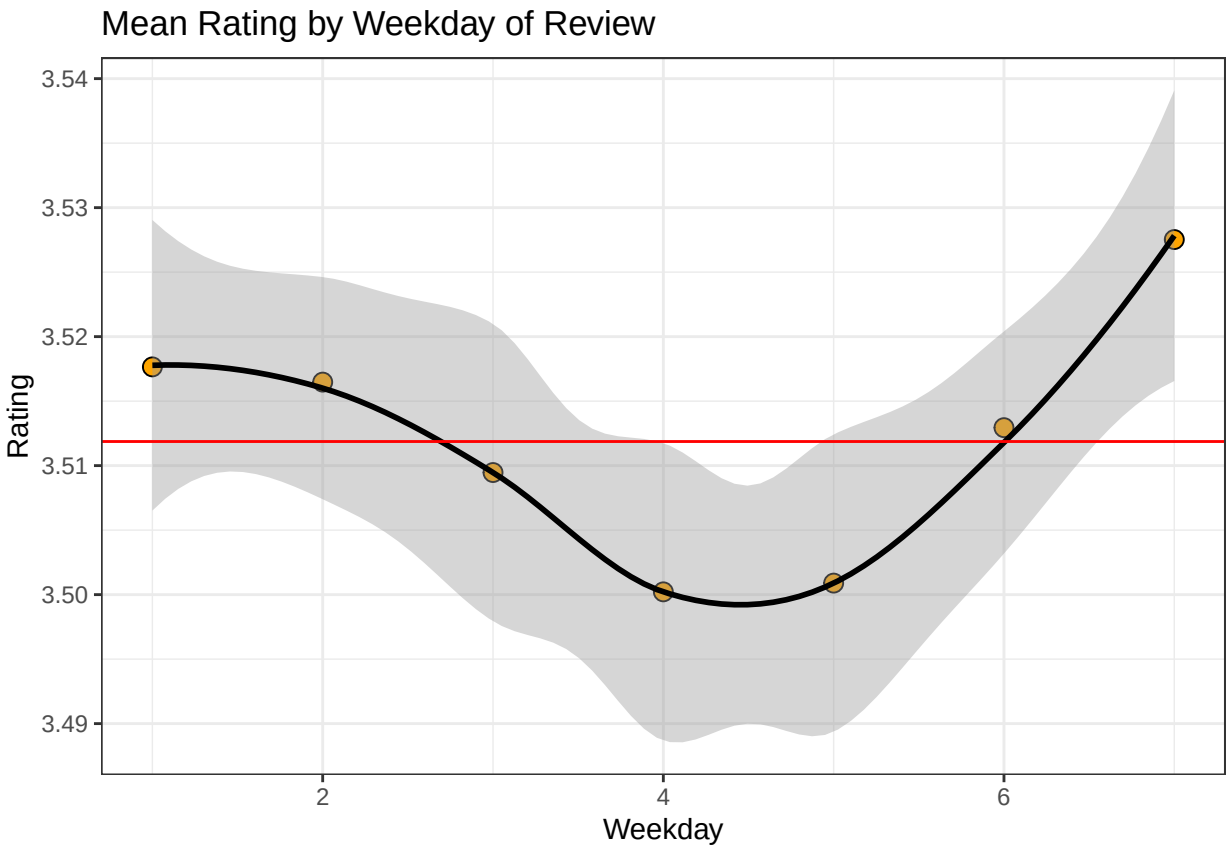
Time Effect: Review Day

Looking at which day of the month a review is submitted, we see a similar effect to that of review month. There exists some non-linearity, but we can't say for sure that the effect is significantly different from the average.



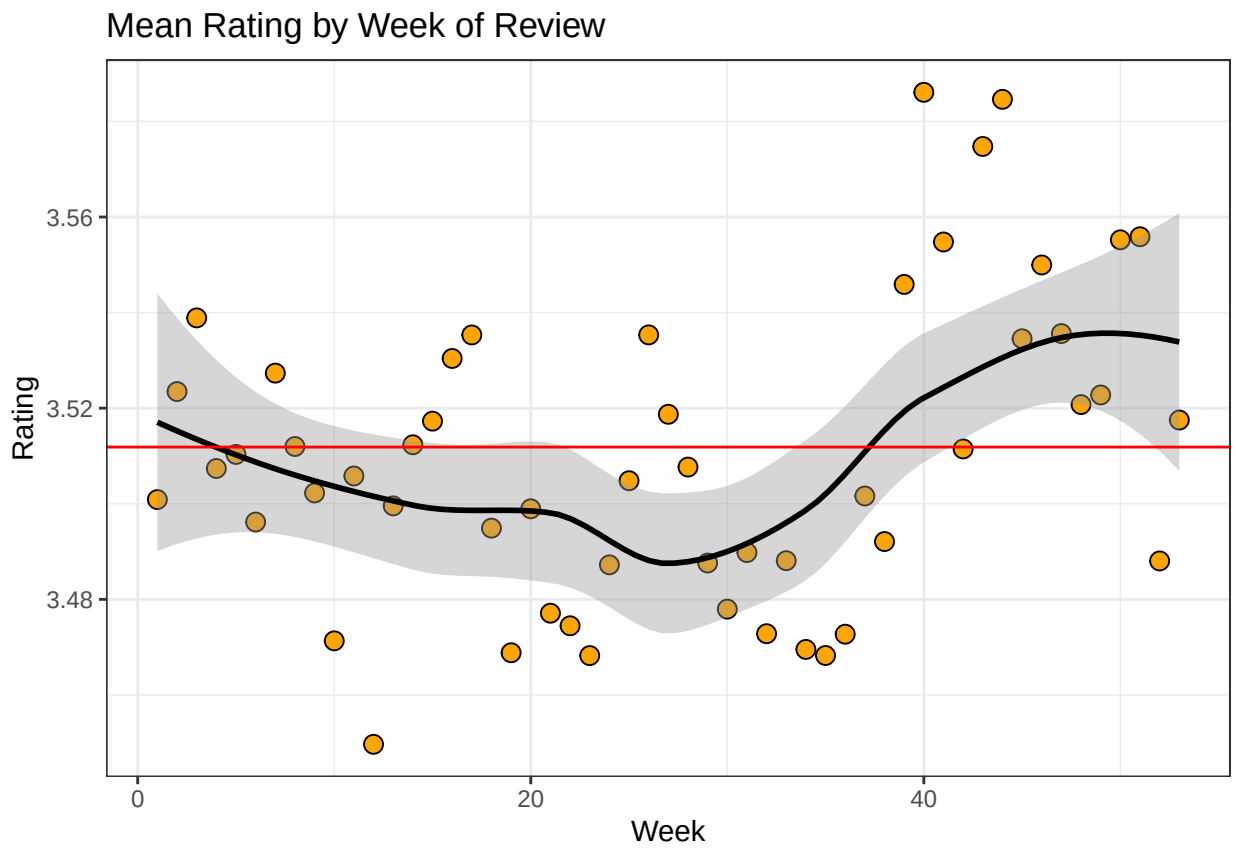
Time Effect: Review Weekday

As with the day of the review, we can't conclude that weekday has an effect significantly different from the mean rating.



Time Effect: Review Week

The week a rating is submitted exhibits non-linearities, but our confidence that it's significantly different from the average only occurs at 2 isolated subsets of the domain.



Time Effect: Summary

In summary, we see that release year, review year, the difference between them, and review hour exhibit non-linear relationships to the average ratings, while the other time related variables are inconclusive.

Modeling User & Movie Effects

The simplest model we can have is the model where the rating is equal to a constant plus some error. The constant that minimizes squared error loss is the overall average.

We improve on the base model by including the conditional user and movie effect allowing our model to take on the form *residual = rating - mu - movie_effect/mu - user_effect/movie_effect, mu.*

User & Movie Effects: Methodology

To arrive at the final effects, we first investigate the effect of requiring users and movies to have a minimum number of reviews, the effect of normalizing the response vector, and the effect of regularization. Since normalization and centering differ by a constant, normalization is appropriate.

Model Entry Criteria: Release Year

The earliest released movie in the data set was released in 1915 during the Silent Era of Film. Since then, Hollywood has had 5 major eras:

1. 1911 - 1927: Silent Era
2. 1927 - 1930: Rise of the Talkies
3. 1930 - 1948: Golden Age
4. 1948 - 1965: Fall of the Studio System

5. 1965 - 1983: New Hollywood
6. 1975 - present: Blockbuster Age

We are limiting entry into the model to movies released on or after 1930 so as to coincide with the commencement of Hollywood's 'Golden Age'. Movies made prior to 1930 comprise .1% of all the data, so their exclusion is not expected to make a material impact on model performance.

Model Entry Criteria: Review Year

In all of the training data there are only 2 ratings submitted in 1995. As such, we'll also limit entry to those ratings submitted starting in 1996.

Search Grid

Regularization

A movie's effect on rating is measured as its average rating across all users and time and vice versa for users. Regularization wishes to account for the number of reviews attributable to each user or movie. The parameter of interest is lambda.

Number of Reviews

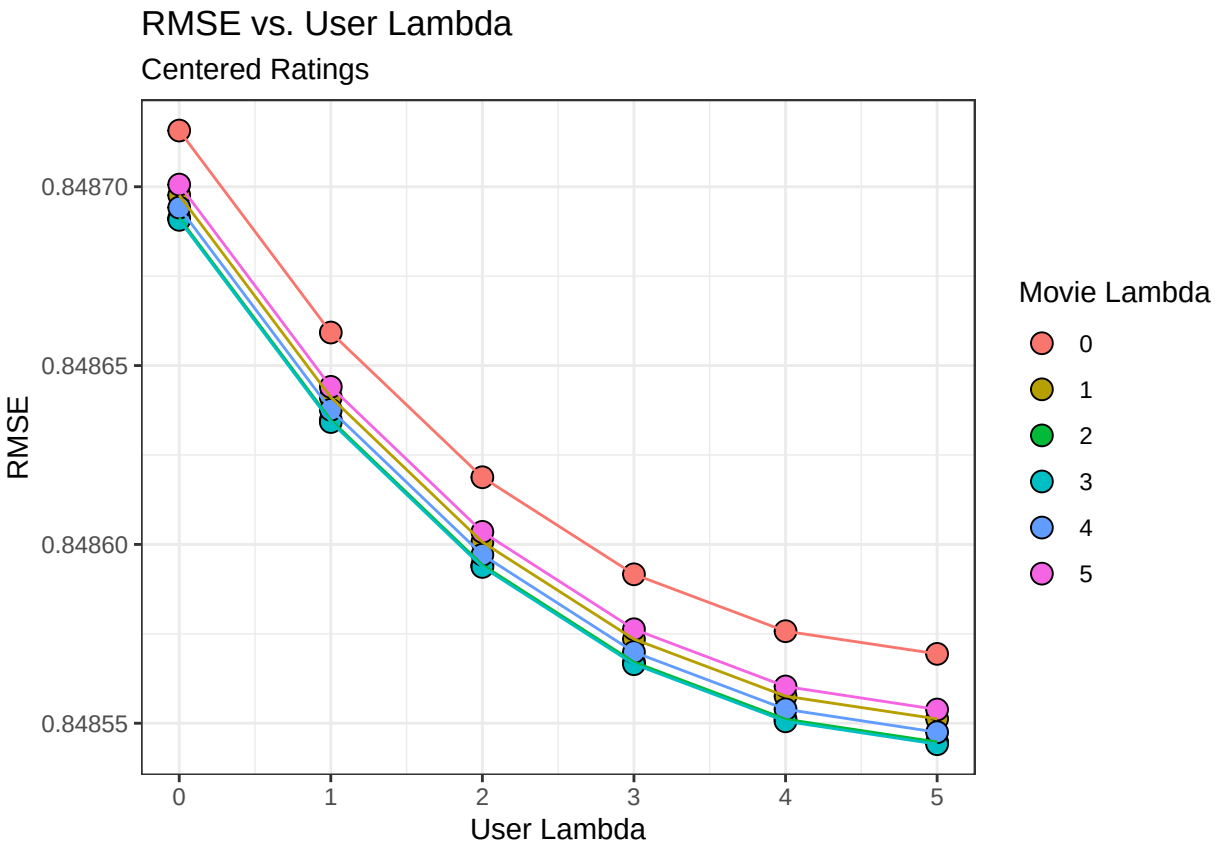
Regularization is one approach to accounting for the number of ratings a user or movie has. Another approach is to limit entry into the model to those users and movies with a minimum number of reviews.

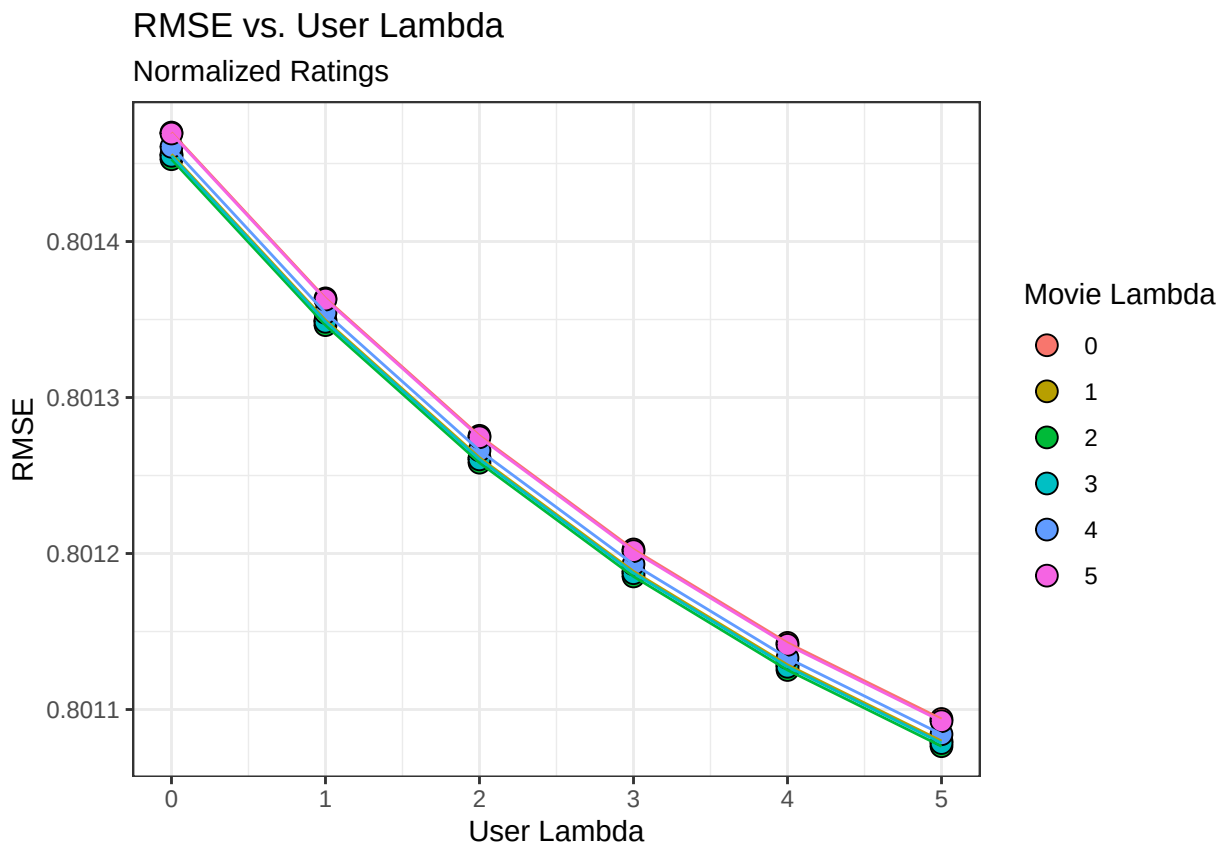
Grid Specification

We perform a grid search to identify the optimal values for lambda and the minimum number of reviews.

To test a sufficiently large range of parameter values, in a computationally efficient amount of time, we optimize each individually. By searching separately, we don't have to search the entire model space.

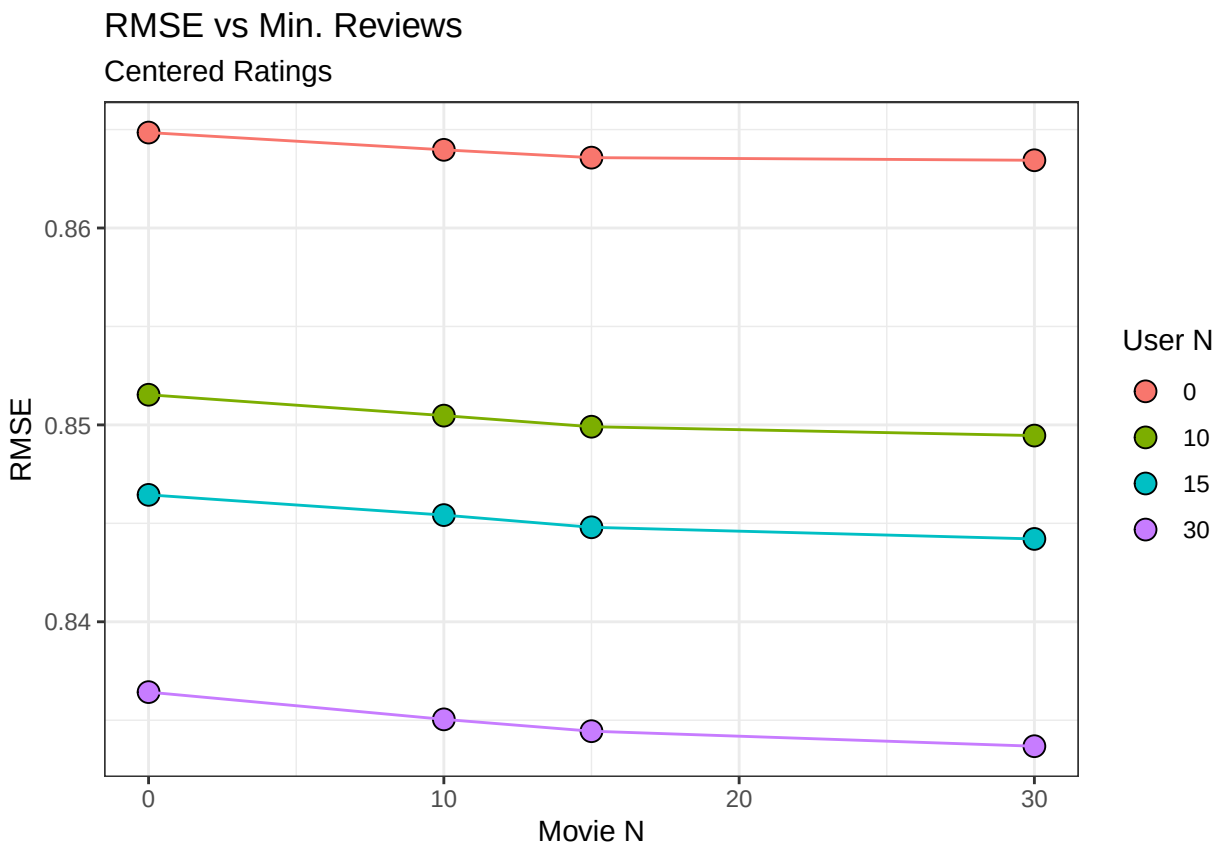
Regularization

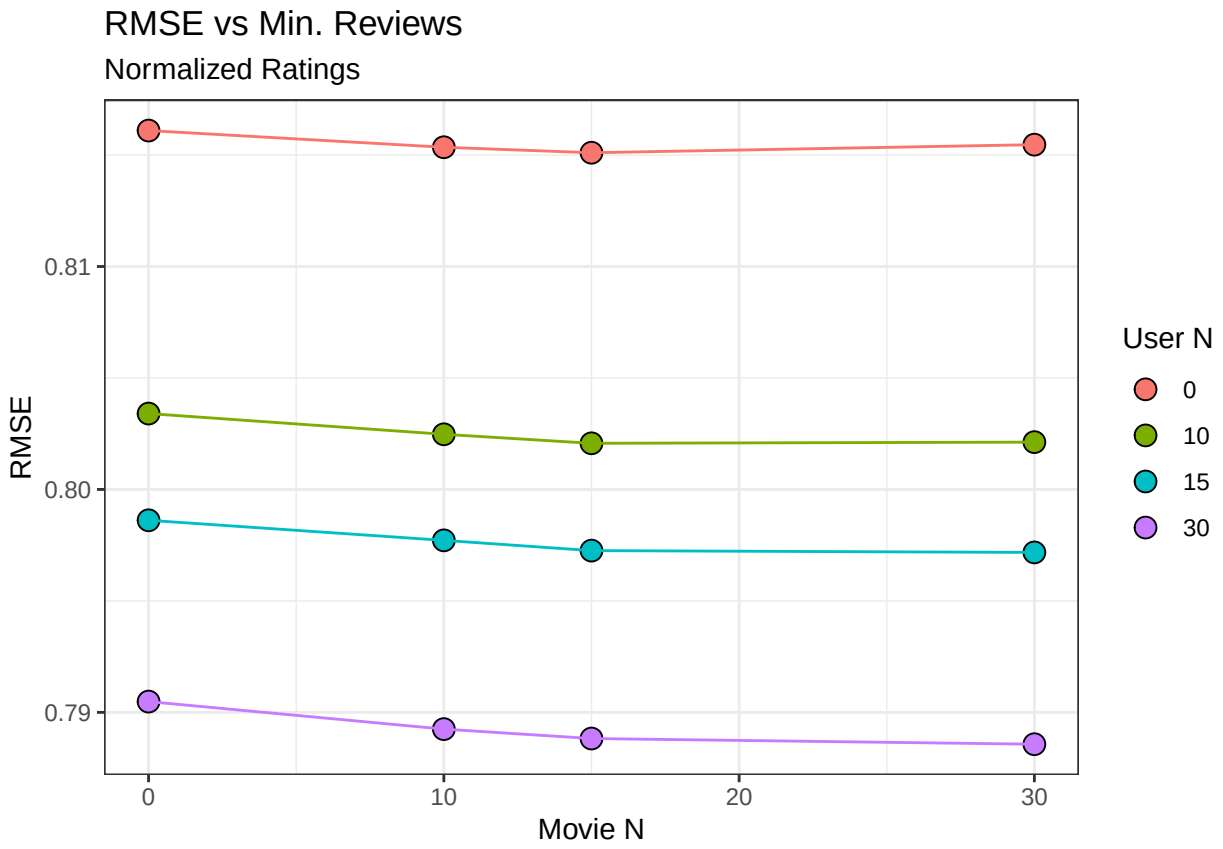

RMSE vs. User Lambda
Centered Ratings
RMSE
0.84870
0.84865
0.84860
0.84855
0
1
2
3
4
5
User Lambda
Movie Lambda
0
1
2
3
4
5



From the plot above, we note that performance is better under regularization than centering and that in either case, we arrive at a minimum test error when lambda = 5. Since regularization has a positive effect on model performance, we must account for it when validating the minimum number of reviews.

Minimum Number of Reviews

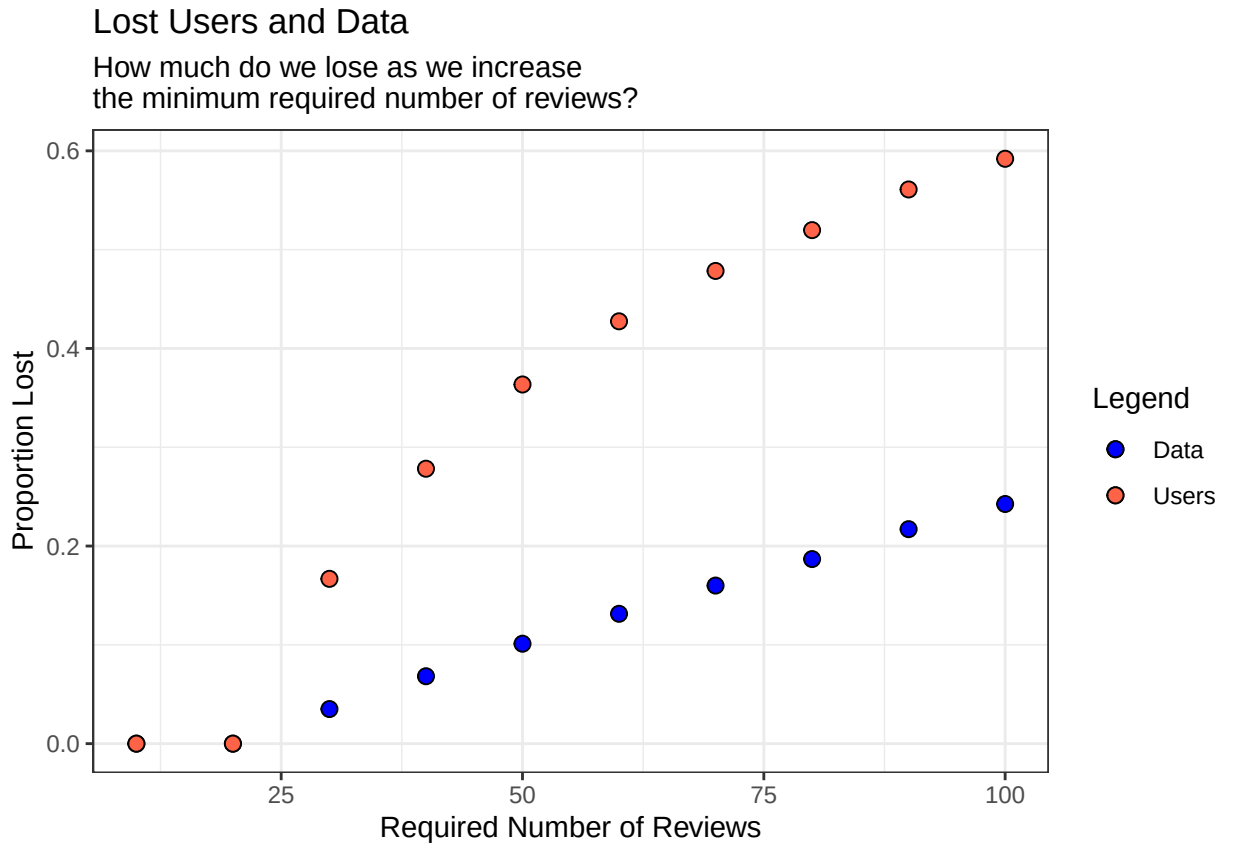

RMSE vs Min. Reviews
Centered Ratings
RMSE
0.86
0.85
0.84
0
10
20
30
Movie N
User N
0
10
15
30



We see that setting a minimum number of reviews for movies doesn't have a significant impact on the test error in either the centered or normalized contexts. After 20 reviews, there's not much of a difference in test error.

Given the above, we'll normalize the ratings & require users and movies to have a minimum of 20 movies before proceeding to calculate the user and movie effects. Before doing so, it's worth considering the data we lose by requiring a minimum number of movies.

Data Loss



Above, we saw that test error beings accelerating upwards past a minimum of 20 reviews. Here we see a correspondence - after a minimum of 20 reviews, the the rate at which users and data is lost accelerates. At a minimum of 20, we lose 5% of users and less than 1% of the data.

Final Movie & User Effect

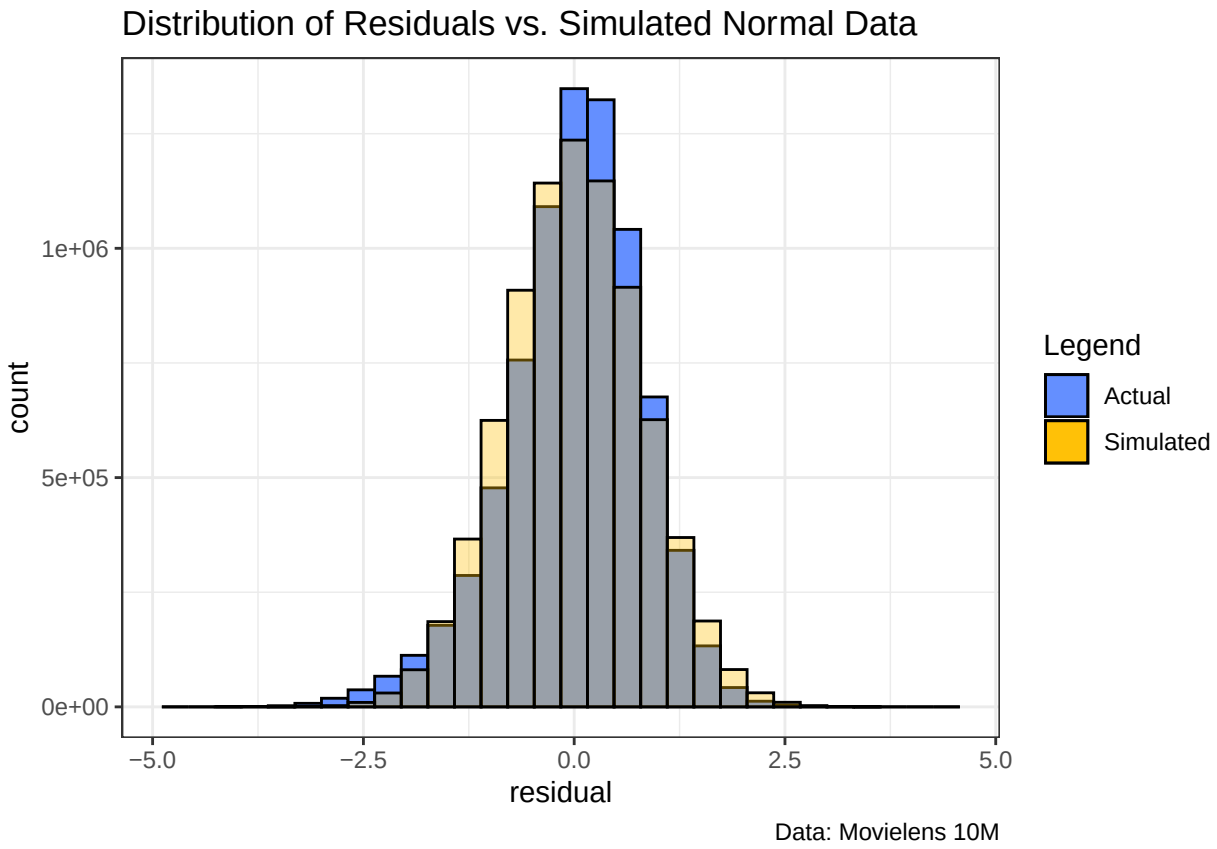
After removing the user and movie effects from the normalized ratings, we get a train error of 0.8061 and a test error of 0.8154. Looking at the distribution of residuals, we see their distribution is more highly peaked and more highly concentrated about its mean than a normal distribution. A Shapiro-Wilk test confirms non-normality.

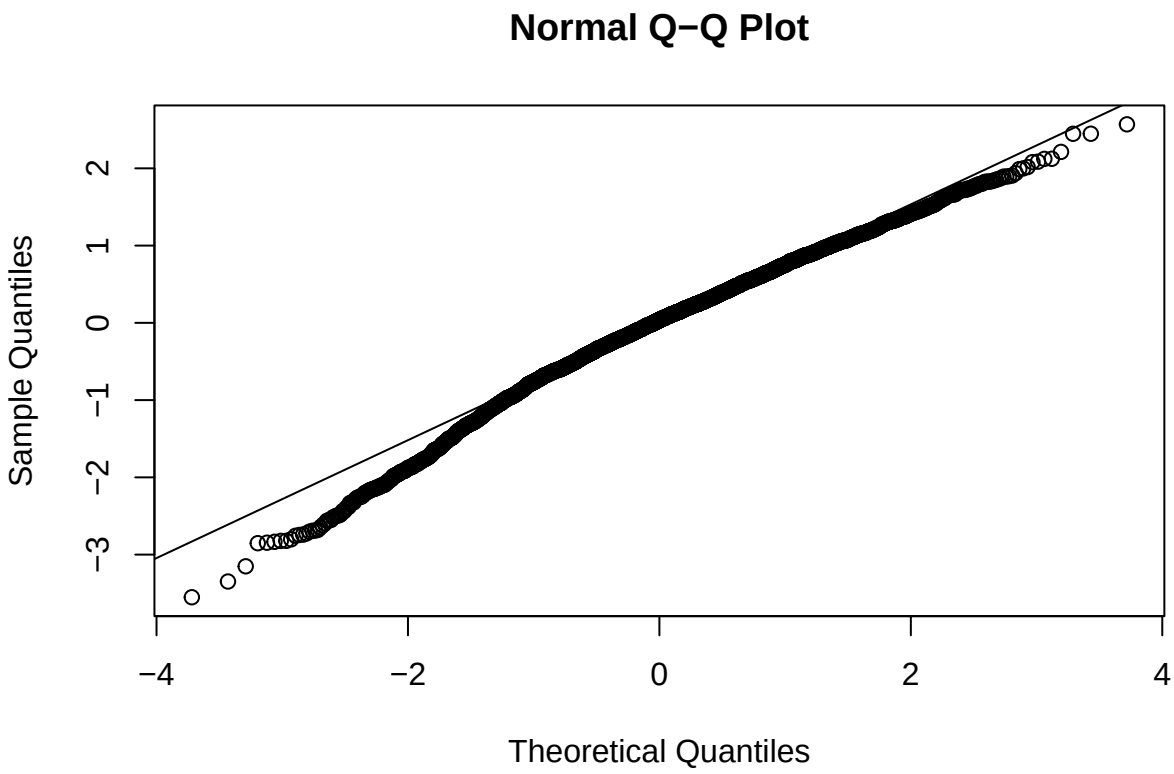
```
## [1] 0.8067081
```

```
## [1] 0.7932881
```

```
##
##  Shapiro-Wilk normality test
##
## data:  samp
## W = 0.9876, p-value < 2.2e-16
```

Further, they exhibit some skew, affecting how the residuals behave in the tails. The normal QQ plot confirms this visually, reflecting the behavior in the histogram above.


Distribution of Residuals vs. Simulated Normal Data
count
1e+06
5e+05
0e+00
−5.0
−2.5
0.0
2.5
5.0
residual
Legend
Actual
Simulated
Data: Movielens 10M



Feature Engineering

Engineering new features allows us to gain further insight and expose more intricate relationships among the variables than we can view at the current resolution. In the current setting we will encode two types of information: information about the release year and information about movie popularity.

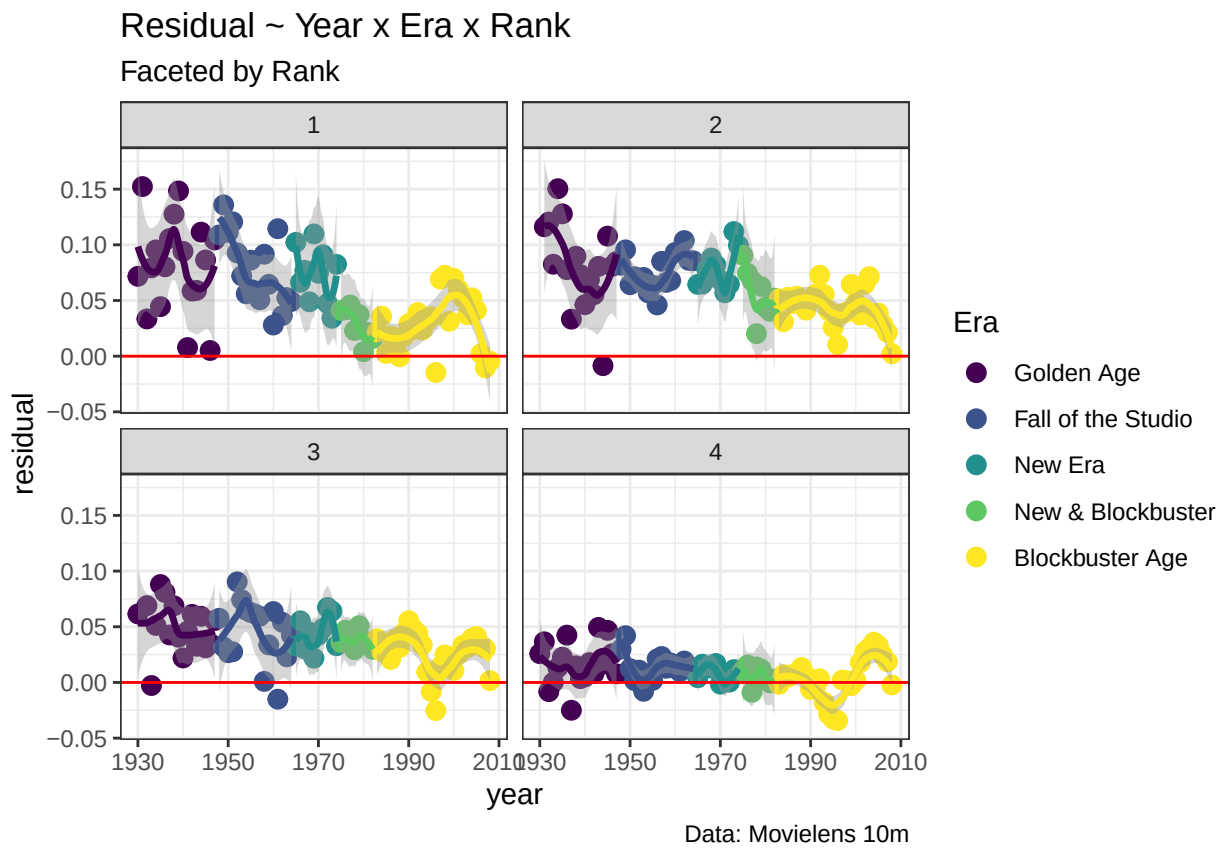
Movie Rank

We assign a movie to 1 of 4 strata according to how its number of reviews compares to other movies. Specifically, we'll employ the 25th, 50th, and 75th percentiles as cutoffs.

The values of those quantiles for the training data are N = c(58, 174, and 678), so we'll assign a movie's rank accordingly and investigate if it exhibits a relationship to time in any way.

Movie Era

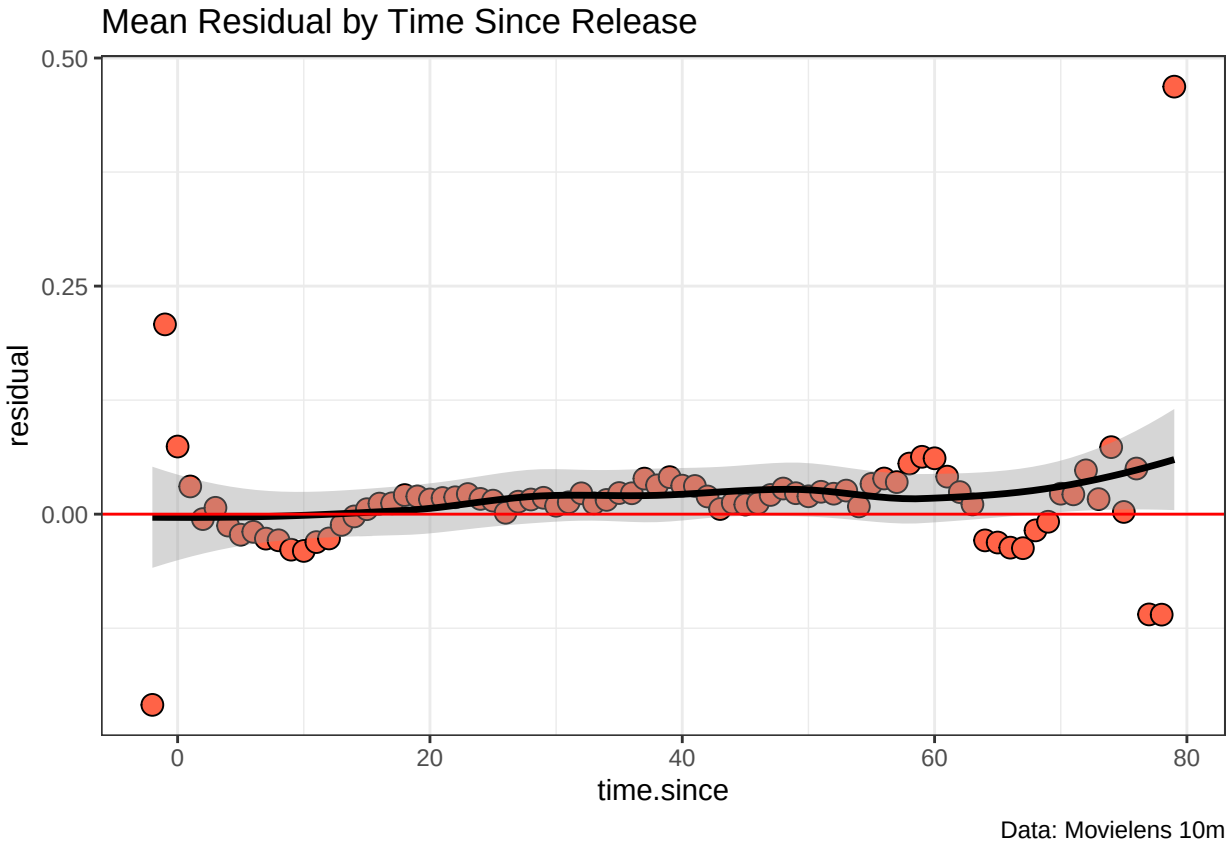
Above, we saw a non-linear relationship between release year and the average residual. We also discussed the various eras Hollywood has experienced since 1930. To account for the era in which a movie is made, we create an era variable according to the time frame we discussed above. Doing so will allow us to expose a non-linear relationship between the release year and the residual.



There's a clear interaction between era and rank across release years. As we go from rank to rank, the non-linear relationship between era and time changes. As such, a function of time that takes the era into account should also take the movie's rank into account as well.

Changing Relationships

Above, we noted a non-linear relationship between the average rating and the difference between the year of review and the year of release. After adjusting for the user and movie effects, that relationship no longer holds.



Estimating The Time Effect

Rank, Era, & Year Effect

Above, we observed an interactive effect between the release year, the era, and the movie's rank. In this context, the era variable effectively acts as the knots, allowing us to visualize the non-linear relationship between rank, year, and residual.

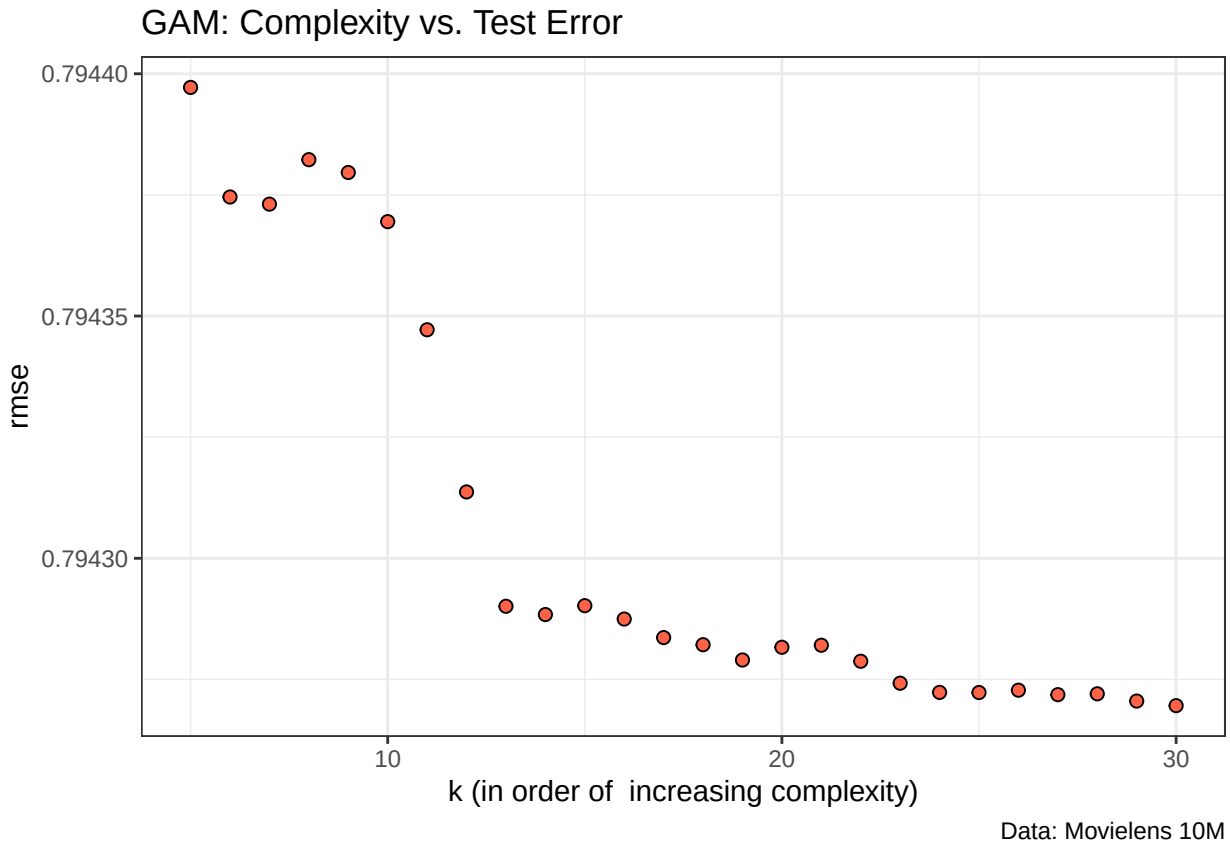
We consider two models: loess & gam. For each we vary their respective complexity parameters and evaluate their performance. In both cases we compare two methodologies. In the first, we split the data by rank and fit a different model on each subset. In the second, we use the entire data set and model the interactions directly.

The primary benefit of splitting the data by rank is computational efficiency. On average, splitting the data yields a 93.47% decrease in fitting time at the cost of a 0.13% increase in the test error.

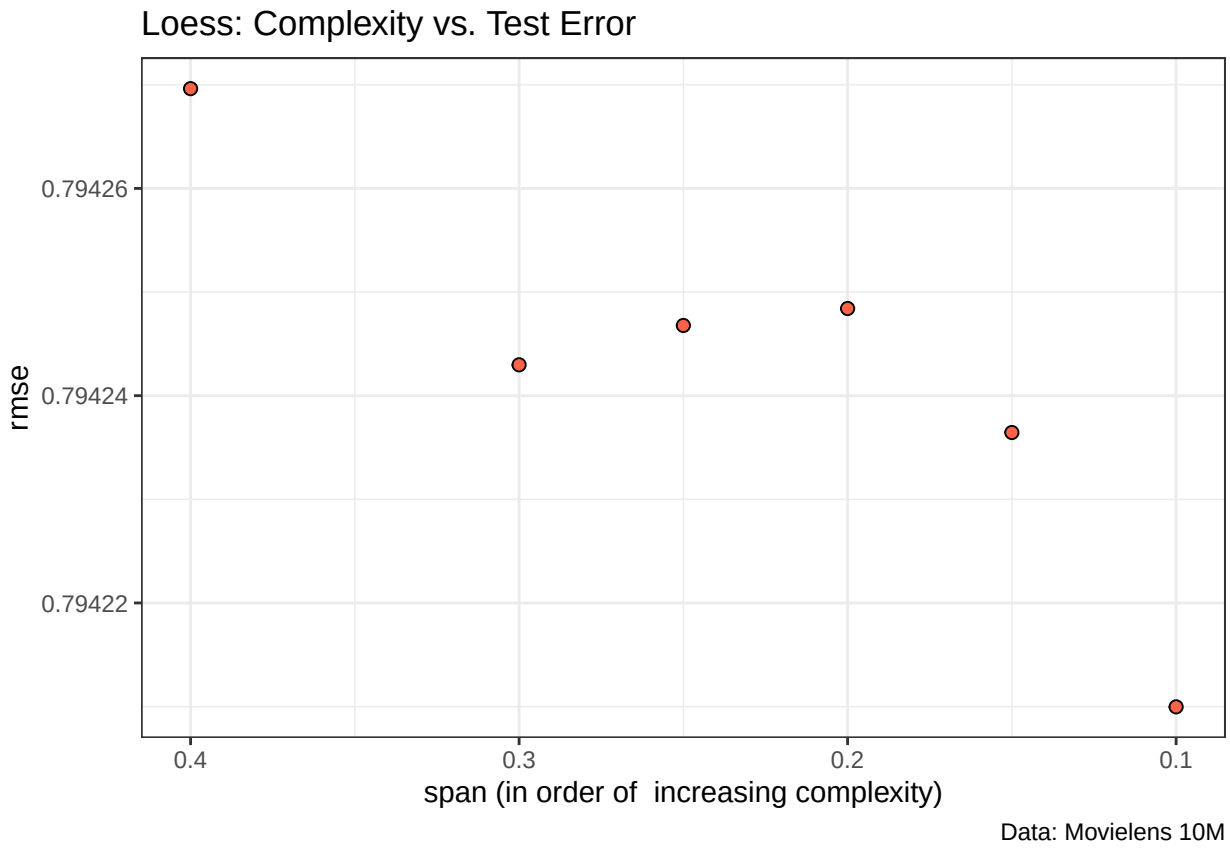
Looking at the split data, we see that loess and gam both perform comparably with loess outperforming gam by a marginal difference.

```
##   rank  gam.rmse loess.rmse    difference
## 1    1 0.7969258  0.7969884 -6.264031e-05
## 2    2 0.7933788  0.7933058  7.292583e-05
## 3    3 0.7908473  0.7907067  1.406170e-04
## 4    4 0.7960722  0.7959685  1.036421e-04
```

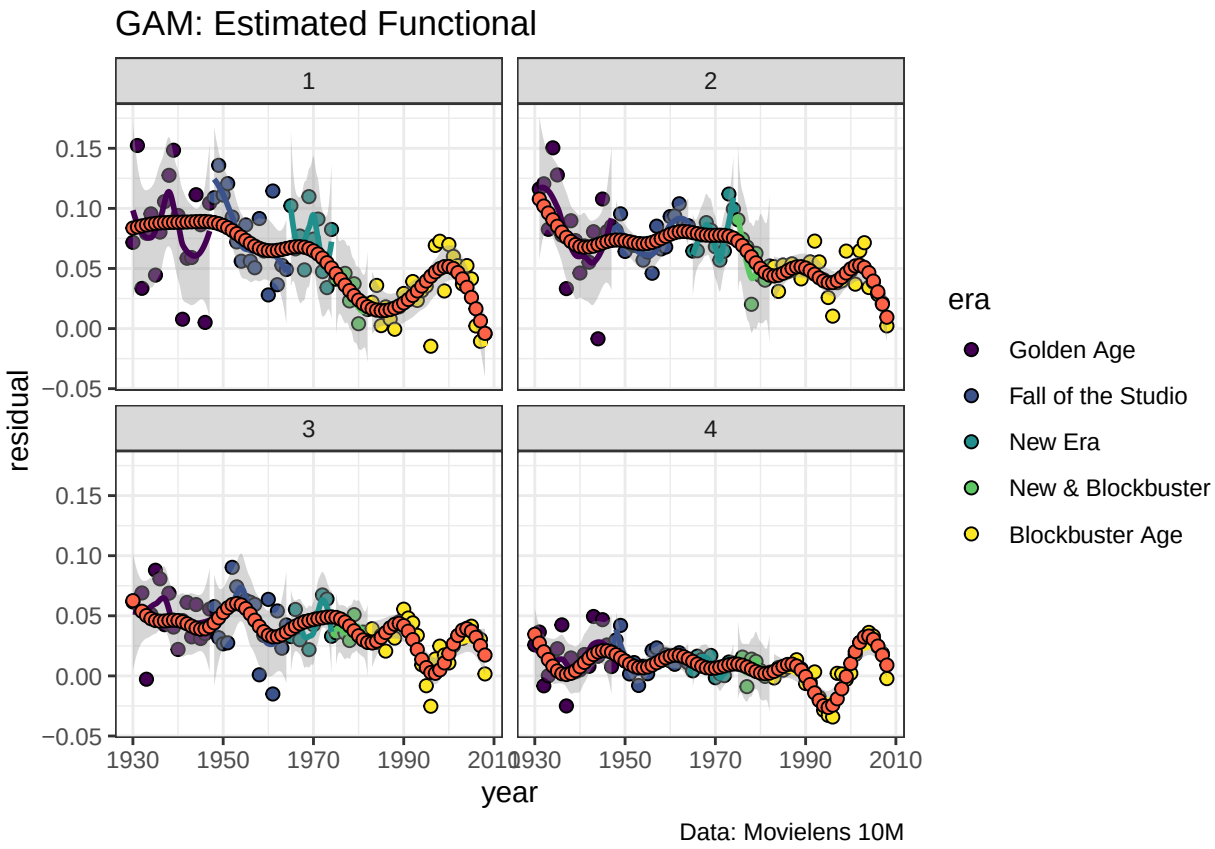
Comparing complexity to test error, for both gam and loess we see a reduction in the test error with the GAMs leveling off after a K of about 13 or 14, with negligible decreases in test error thereafter.

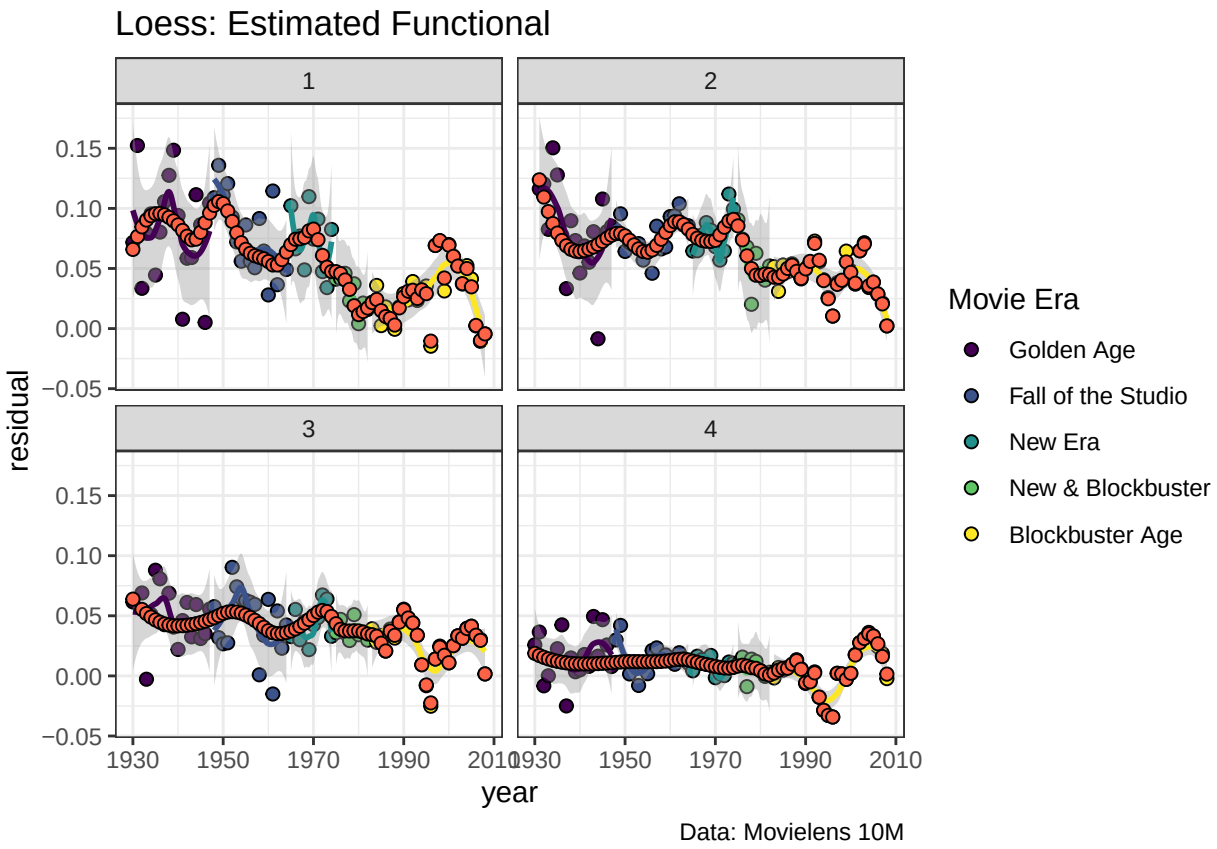


For loess, as the complexity increases, we see an almost cubic relationship emerge between complexity and test error.



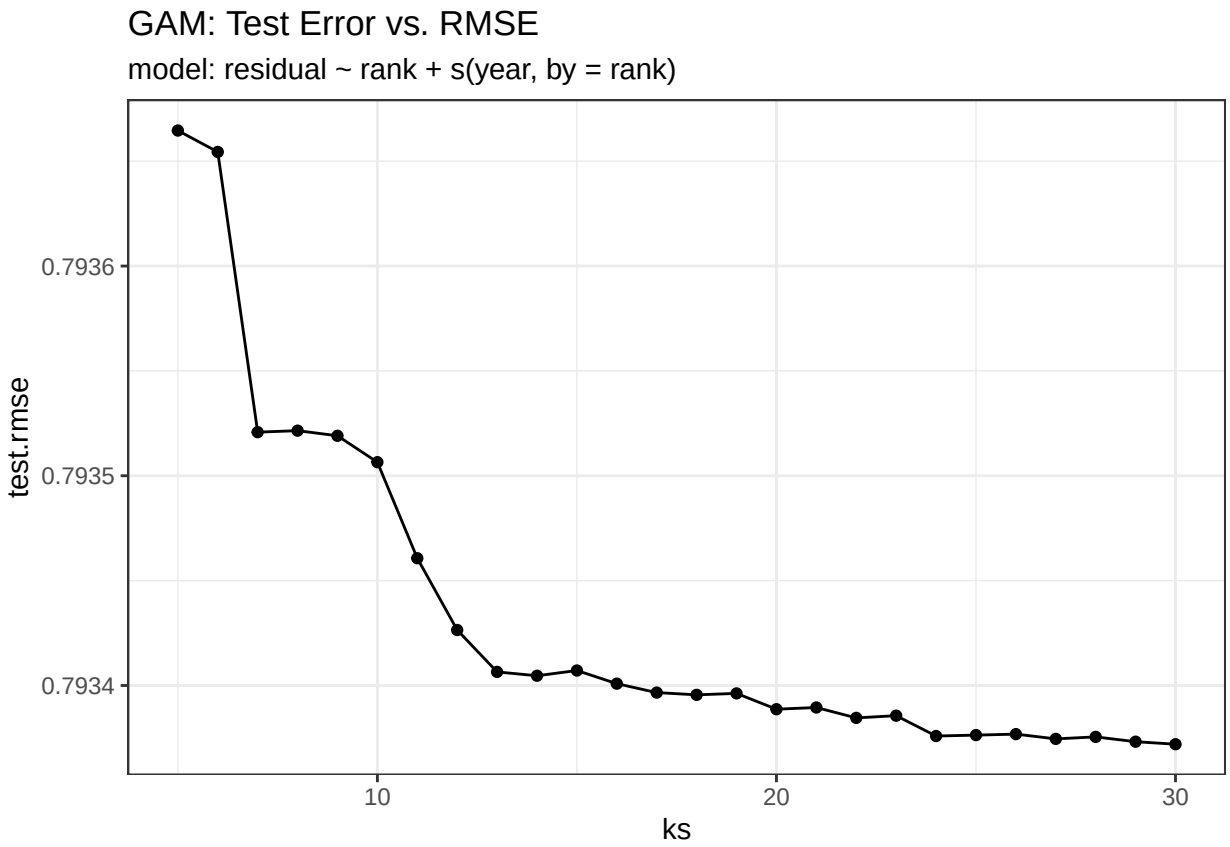
Comparing the estimated functional using the optimal complexity parameters yields the following two graphs. Underneath we have the original plot with each model's estimate of the conditional expectation. Comparing the two, we see that the GAM fit a much smoother function than loess does, despite the fact that loess has a slight analytic edge.


GAM: Estimated Functional
1
2
3
4
residual
0.15
0.10
0.05
0.00
−0.05
1930
1950
1970
1990
2010
year
era
Golden Age
Fall of the Studio
New Era
New & Blockbuster
Blockbuster Age
Data: Movielens 10M

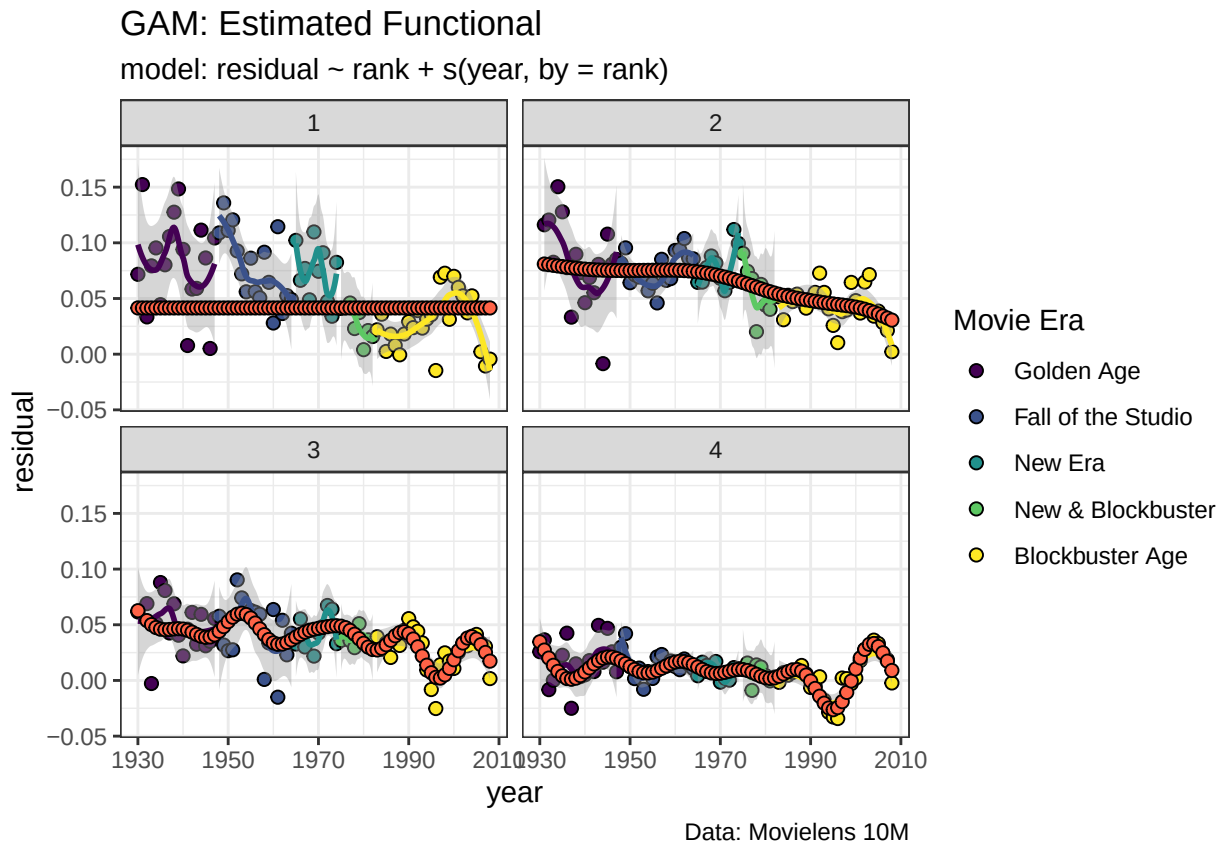


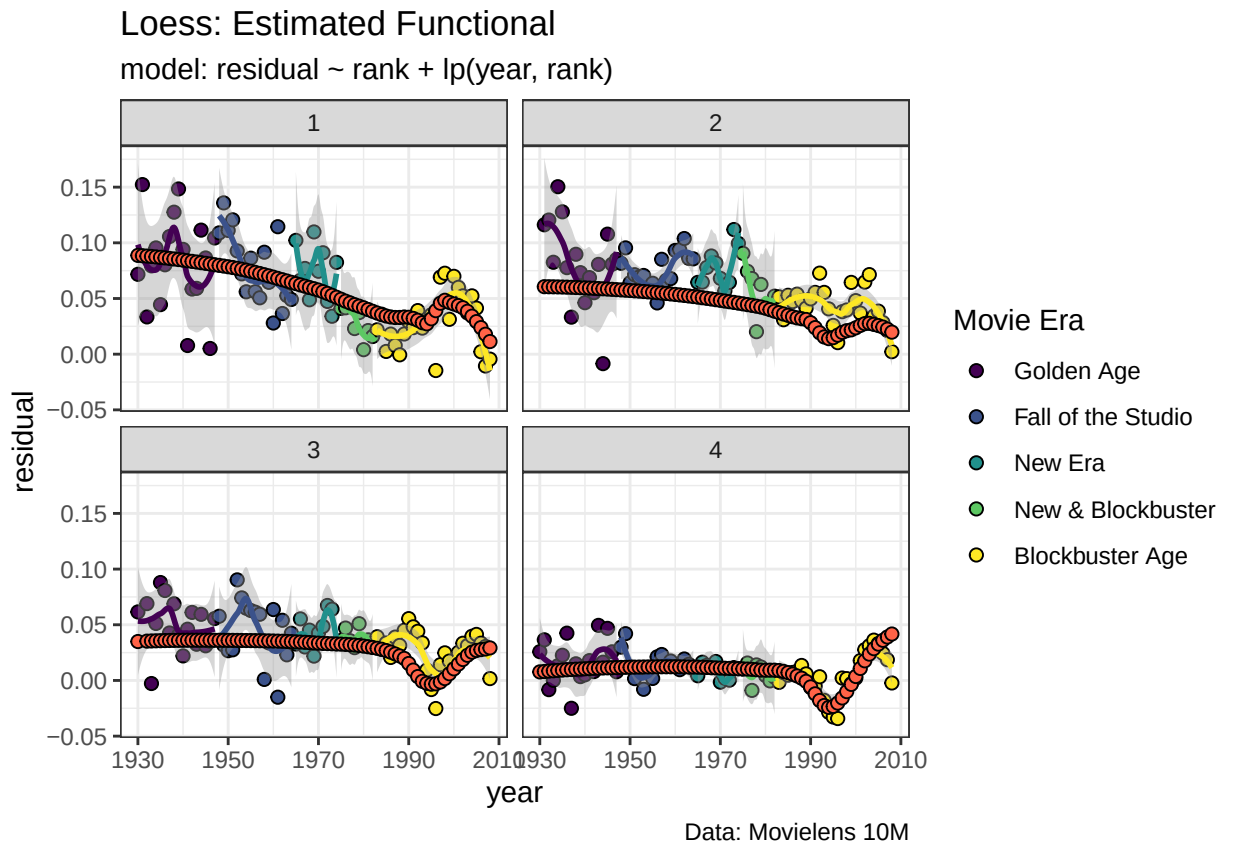
Combining the estimates of both models in a single ensemble will produce a model with more robust estimates of the true functional. Before fitting the model we have a train error of 0.8067081. Whether using the ensemble or either of its individual contributors, we still see a drop in the train error down to an average of 0.8063648.

```
##      model      rmse
## 1 residual 0.8067081
## 2 ensemble 0.8063648
## 3      gam 0.8063935
## 4    loess 0.8063553
```



Comparing to the models fit on the entire data set we have the following set of estimated conditional expectations for GAM and Loess. For both, using the entire data set and fitting a more complicated model resulted in much more linear expectations for ranks 1 and 2 when compared to the functionals estimated using the split data.


GAM: Estimated Functional
model: residual ~ rank + s(year, by = rank)
1
2
3
4
residual
0.15
0.10
0.05
0.00
−0.05
1930
1950
1970
1990
2010
year
Movie Era
Golden Age
Fall of the Studio
New Era
New & Blockbuster
Blockbuster Age
Data: Movielens 10M



Analytically, we get better train performance from using the split data versus the full data set so, when we train on all of edx, we'll split the data by rank first.

```
##   model      rmse  data
## 1   gam 0.8063995  full
## 2 loess 0.8064285  full
## 3   gam 0.8063935 split
## 4 loess 0.8063553 split
```

After fitting the model on the entire training set, the error is reduced by a good amount. However, after predicting on the test set, we get an *increase* in the test error over not estimating a time effect.

```
##   residual      rmse
## 1      rsd 0.8075284
## 2  rsd.ens 0.8071830
## 3  rsd.gam 0.8072108
## 4  rsd.loe 0.8071739
```

```
##   residual      rmse
## 1      rsd 0.7951653
## 2  rsd.ens 0.7957203
## 3  rsd.gam 0.7958073
## 4  rsd.loe 0.7956741
```

Conclusion

After validating model entry criteria and the optimal level of regularization, we estimated the user and movie effects, allowing us to then estimate the effect of time. By placing knots at the boundaries of Hollywood Eras allowed us to uncover a non-linear relationship between release year and residual as faceted by movie rank.

Though our models are able to capture the non-linearities present in the data, they do not necessarily aid in our understanding as the normalized response vector, adjusted for users and movies, yields a test rmse of 0.7951653 while the estimation of the time effect by ensemble actually increases the test rmse by a marginal amount.

Within the current model are a number of latent variables such as cast, director, budget, gross revenue, etc., all of which factor into a movie's popularity but were not included within this analysis. Attempts were made to source this information by scraping the web, but the disparity of sources, lack of complete data, and other issues prevented their inclusion within the model. A more sophisticated model would seek to include those variables.