Do Movie Reviewers Rate at the Extremes?

AUTHOR
Thomas Cochran, Will Young, Sawarj Patil

Introduction

Online movie ratings influence how people decide what to watch. Intuitively, it can feel like people either love a movie (5 stars) or hate it (1 star), and that moderate ratings are rare.

In this project we use individual ratings from the MovieLens dataset to investigate the question:

Do individual reviewers tend to rate movies at the extremes (<=1 or >= 4.5 stars), or do most ratings fall in the middle?

We treat ratings of <= 1 and >= 4.5 (on a 0.5 to 5 star scale) as extreme. Our main statistical question is:

Is the proportion of extreme ratings greater than 20% of all ratings?

If more than one in five ratings are extreme, we interpret that as evidence of meaningful polarization in user rating behavior.

Data

The MovieLens dataset is provided by GroupLens, a research group in the Department of Computer Science and Engineering at the University of Minnesota. We use two CSV files:

ratings.csv is individual ratings

movies.csv is movie titles and genres

```r
# load packages for the whole document
library(readr)    # for read_csv()
library(dplyr)    # for glimpse()
```

```

Attaching package: 'dplyr'


The following objects are masked from 'package:stats':

    filter, lag


The following objects are masked from 'package:base':

    intersect, setdiff, setequal, union
```

```r
library(tidyverse)
```

```
Warning: package 'tidyverse' was built under R version 4.4.3
```

```
Warning: package 'ggplot2' was built under R version 4.4.3
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   4.0.1     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.2
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts
to become errors
```

```r
#Read data from CSV files in the working directory

ratings <- read_csv("C:/Users/tcochran33/Desktop/LargeData/ratings.csv")
```

```
Rows: 33832162 Columns: 4
-- Column specification --------------------------------------------------------
Delimiter: ","
dbl (4): userId, movieId, rating, timestamp

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
movies  <- read_csv("C:/Users/tcochran33/Desktop/LargeData/movies.csv")
```

```
Rows: 86537 Columns: 3
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): title, genres
dbl (1): movieId

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
glimpse(ratings)
```

```
Rows: 33,832,162
Columns: 4
$ userId    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ movieId   <dbl> 1, 110, 158, 260, 356, 381, 596, 1036, 1049, 1066, 1196, 120~
$ rating    <dbl> 4.0, 4.0, 4.0, 4.5, 5.0, 3.5, 4.0, 5.0, 3.0, 4.0, 3.5, 3.5, ~
$ timestamp <dbl> 1225734739, 1225865086, 1225733503, 1225735204, 1225735119, ~
```

```r
glimpse(movies)
```

```
Rows: 86,537
Columns: 3
$ movieId <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,~
$ title   <chr> "Toy Story (1995)", "Jumanji (1995)", "Grumpier Old Men (1995)~
$ genres  <chr> "Adventure|Animation|Children|Comedy|Fantasy", "Adventure|Chil~
```

Each row of ratings is a single user’s rating of a movie:

- userId
- movieId
- rating (0.5 to 5, in 0.5 star increments)
- timestamp

Each row of movies contains:

- movieId
- title
- genres

We join these so that each rating has its corresponding movie title.

```r
ratings_full <- ratings %>%
inner_join(movies, by = "movieId")

glimpse(ratings_full)
```

```
Rows: 33,832,162
Columns: 6
$ userId    <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ~
$ movieId   <dbl> 1, 110, 158, 260, 356, 381, 596, 1036, 1049, 1066, 1196, 120~
$ rating    <dbl> 4.0, 4.0, 4.0, 4.5, 5.0, 3.5, 4.0, 5.0, 3.0, 4.0, 3.5, 3.5, ~
$ timestamp <dbl> 1225734739, 1225865086, 1225733503, 1225735204, 1225735119, ~
$ title     <chr> "Toy Story (1995)", "Braveheart (1995)", "Casper (1995)", "S~
$ genres    <chr> "Adventure|Animation|Children|Comedy|Fantasy", "Action|Drama~
```

Filtering to Movies with at Least 30 Ratings

To focus on movies with a reasonable amount of feedback, we keep only movies that have at least 30 ratings.

```
ratings_30plus <- ratings_full %>%
group_by(movieId) %>%
filter(n() >= 30) %>%
ungroup()

nrow(ratings_full)      # total number of ratings
```

```
[1] 33832162
```

```
nrow(ratings_30plus)    # ratings for movies with >= 30 ratings
```

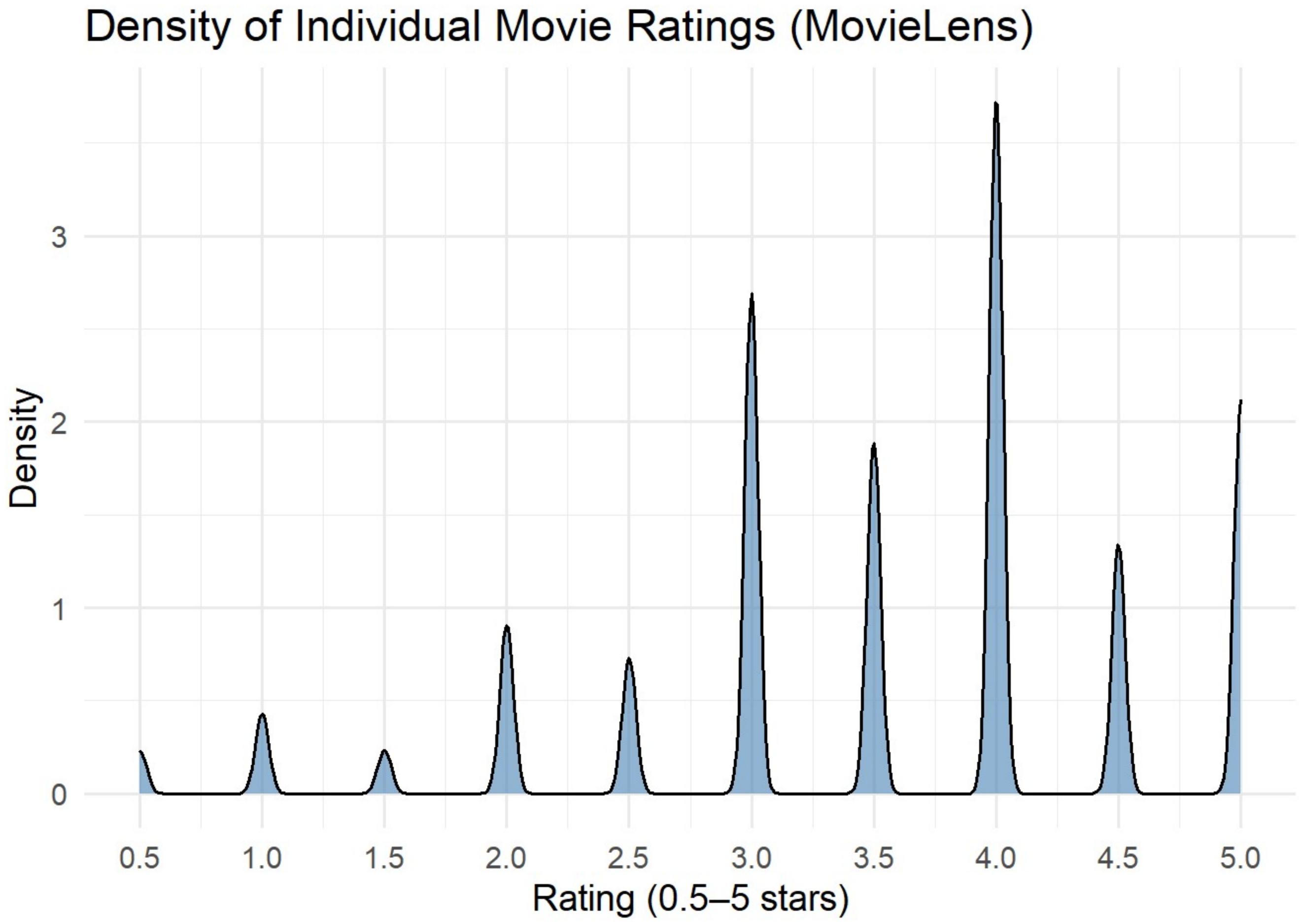
```
[1] 33471612
```

Descriptive Statistics

Distribution of Individual Ratings

We first examine the distribution of all individual ratings in our filtered data set.

```
ggplot(ratings_30plus, aes(x = rating)) +
geom_density(fill = "steelblue", alpha = 0.6, adjust = 1.3) +
scale_x_continuous(breaks = seq(0.5, 5, by = 0.5)) +
labs(
title = "Density of Individual Movie Ratings (MovieLens)",
x = "Rating (0.5–5 stars)",
y = "Density"
) +
theme_minimal(base_size = 14)
```



Density of individual movie ratings (movies with ≥ 30 ratings).

Categorizing Ratings as Low, Middle, or High

We classify each rating as low extreme, middle, or high extreme:

- **Low extreme**: rating <= 1
- **High extreme**: rating >= 4.5
- **Middle**: 1 < rating < 4.5

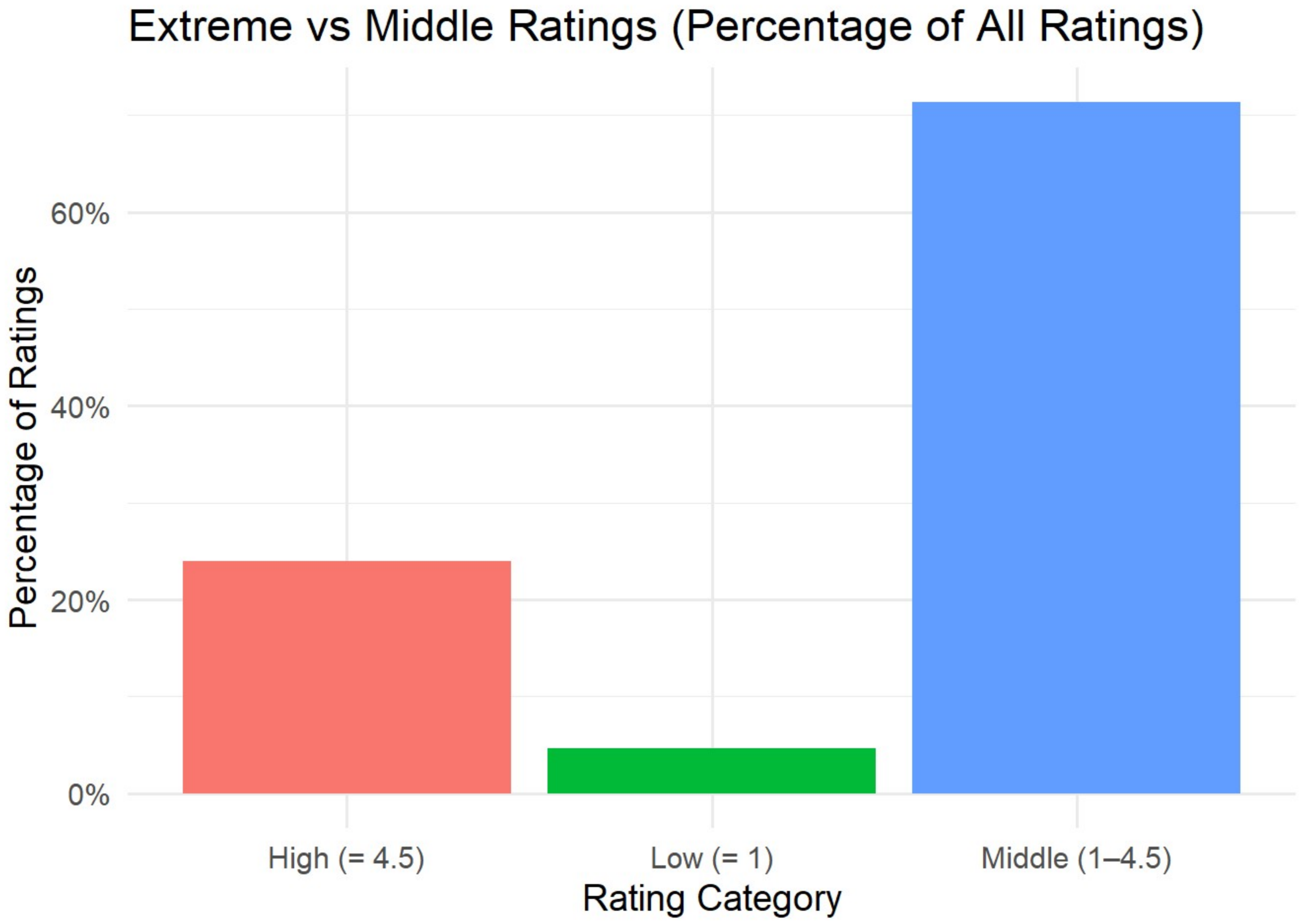
```
ratings_extreme <- ratings_30plus %>%
mutate(
extreme = case_when(
rating <= 1   ~ "Low (≤ 1)",
rating >= 4.5 ~ "High (≥ 4.5)",
TRUE          ~ "Middle (1–4.5)"
)
)
```

```r
extreme_counts <- ratings_extreme %>%
count(extreme) %>%
mutate(prop = n / sum(n))

extreme_counts
```

```
# A tibble: 3 x 3
  extreme                n   prop
  <chr>              <int>  <dbl>
1 High (= 4.5)     8037731 0.240
2 Low (= 1)        1541499 0.0461
3 Middle (1-4.5) 23892382 0.714
```

```r
# Plot percentages instead of counts
ggplot(extreme_counts, aes(x = extreme, y = prop, fill = extreme)) +
  geom_col() +
  scale_y_continuous(labels = scales::percent_format(accuracy = 1)) +
  labs(
    title = "Extreme vs Middle Ratings (Percentage of All Ratings)",
    x = "Rating Category",
    y = "Percentage of Ratings"
  ) +
  theme_minimal(base_size = 14) +
  theme(legend.position = "none")
```



We see the proportions of low, middle, and high ratings, which provides context for our hypothesis tests.

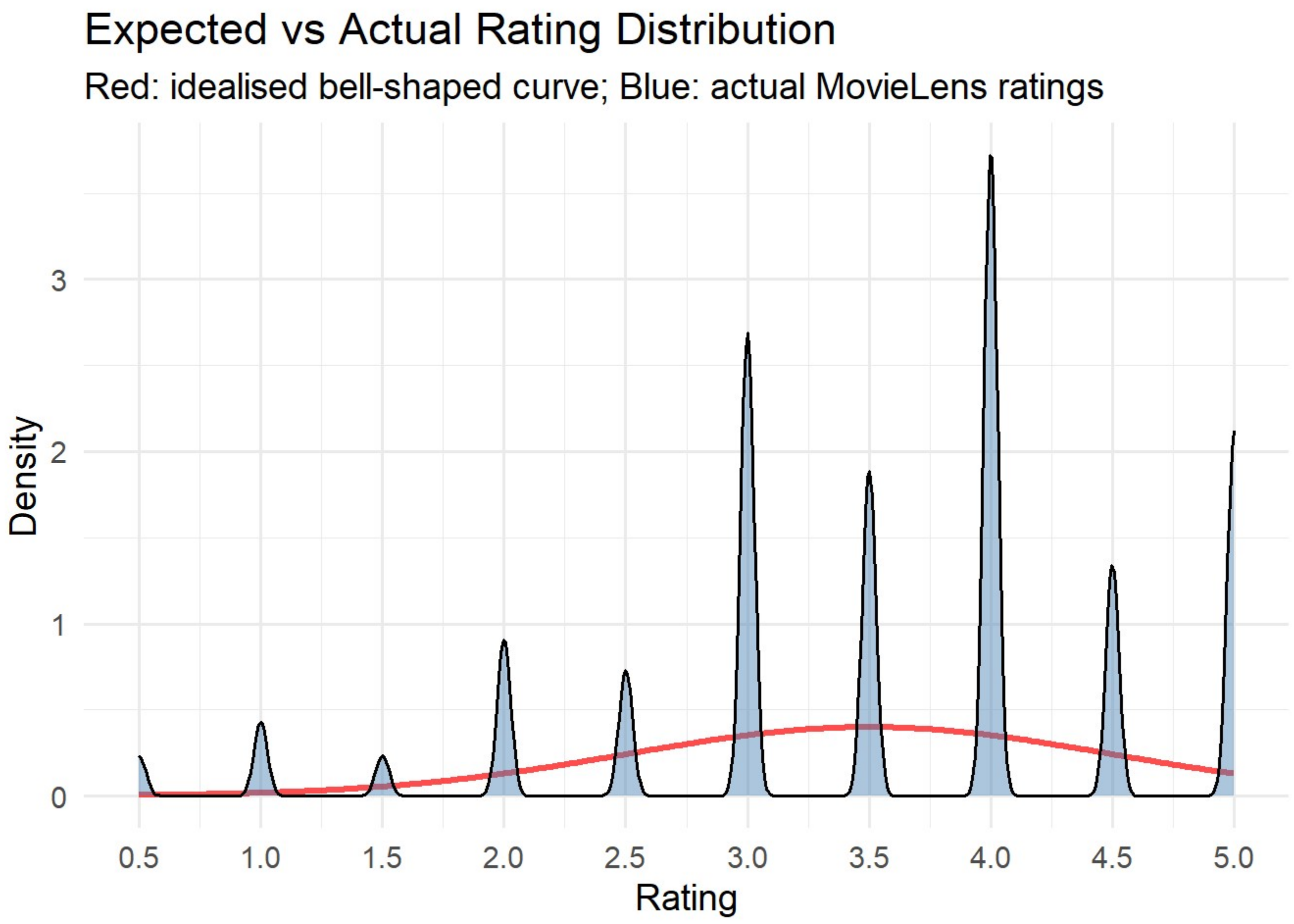
Expected vs Actual Distribution

In non-polarized rating systems, we expect scores to form a bell-shaped curve that is centered at about 3.5 stars with few ratings in the extremes. However our graph shows that the actual MovieLens ratings (in blue) look much different from the ideal. Instead of a smooth curve there's sharp spikes at each half-star with especially tall peaks at 3.0 and 4.0. This concentration of ratings on the right side indicates both a preference for round scores and a strong left skew or inflation in ratings. Overall, users don't spread ratings evenly over the scale and cluster around what would be seen as good or great which supports our conclusion about positive bias in user reviews.

```r
theoretical <- tibble(
rating  = seq(0.5, 5, by = 0.01),
density = dnorm(rating, mean = 3.5, sd = 1)
)

ggplot() +
geom_line(data = theoretical,
aes(x = rating, y = density),
color = "red", size = 1.2, alpha = 0.7) +
geom_density(data = ratings_30plus,
aes(x = rating),
fill = "steelblue", alpha = 0.45, adjust = 1.3) +
scale_x_continuous(breaks = seq(0.5, 5, by = 0.5)) +
labs(
title = "Expected vs Actual Rating Distribution",
subtitle = "Red: idealised bell-shaped curve; Blue: actual MovieLens ratings",
x = "Rating",
y = "Density"
) +
theme_minimal(base_size = 14)
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```



Theoretical bell-shaped rating distribution (red) vs actual MovieLens ratings (blue).

Hypothesis Testing: Are Extreme Ratings Too Common?

Let:

- p = true proportion of all ratings that are extreme

We test whether extreme ratings are more frequent than 20% of all ratings:

- **Null hypothesis** H_0 : p <= 0.20
- **Alternative hypothesis** H_A : p > 0.20

We choose **20%** as a practical cutoff for a moderate polarization. In a typical uni modal system, we would expect a much smaller share of ratings at the extremes. If more than one in five ratings are extreme, we see that as evidence that users lean toward very strong opinions.

One-sample Proportion Test (All Extremes Combined)

```
# Count all extreme ratings (low + high)

extreme_count <- ratings_extreme %>%
filter(extreme != "Middle (1-4.5)") %>%
nrow()

total_ratings <- nrow(ratings_extreme)

extreme_prop <- extreme_count / total_ratings
extreme_prop
```

```
[1] 0.2861897
```

```
prop_test_all <- prop.test(
x = extreme_count,
n = total_ratings,
p = 0.20,             # null value p0 = 0.20
alternative = "greater",
correct = FALSE
)

prop_test_all
```

```
	1-sample proportions test without continuity correction
```

```
data:  extreme_count out of total_ratings, null probability 0.2
X-squared = 1554058, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is greater than 0.2
95 percent confidence interval:
 0.2860612 1.0000000
sample estimates:
        p 
0.2861897 
```

The sample proportion of extreme ratings is:

```
round(extreme_prop, 3)
```

```
[1] 0.286
```

Separate Tests for Low and High Extremes

To understand which side drives the result, we separately examine low and high extremes.

```
low_count <- ratings_30plus %>%
filter(rating <= 1) %>%
nrow()

high_count <- ratings_30plus %>%
filter(rating >= 4.5) %>%
nrow()

low_prop  <- low_count  / total_ratings
high_prop <- high_count / total_ratings

c(low_prop  = low_prop,
high_prop = high_prop)
```

```
  low_prop  high_prop 
0.04605392 0.24013576 
```

Test for Low Extreme Ratings (≤ 1 star)

```
prop_test_low <- prop.test(
x = low_count,
n = total_ratings,
p = 0.20,
alternative = "greater",
correct = FALSE
)

prop_test_low
```

```
	1-sample proportions test without continuity correction

data:  low_count out of total_ratings, null probability 0.2
X-squared = 4957856, df = 1, p-value = 1
alternative hypothesis: true p is greater than 0.2
95 percent confidence interval:
 0.04599437 1.00000000
sample estimates:
         p 
0.04605392 
```

Test for High Extreme Ratings (≥ 4.5 stars)

```
prop_test_high <- prop.test(
x = high_count,
n = total_ratings,
p = 0.20,
alternative = "greater",
correct = FALSE
)

prop_test_high
```

```
	1-sample proportions test without continuity correction

data:  high_count out of total_ratings, null probability 0.2
X-squared = 336992, df = 1, p-value < 2.2e-16
alternative hypothesis: true p is greater than 0.2
95 percent confidence interval:
 0.2400143 1.0000000
sample estimates:
        p 
0.2401358 
```

Results and Interpretation

Based on the combined test:

- The observed proportion of extreme ratings is about **28.6%**

- The p-value from the sample proportion test with $p_0 = 0.20$ is extremely small, so we reject H_0 and conclude that **more than 20% of all ratings are extreme.**

From the separate tests:

- The proportion of low extreme ratings (<=1 star) is **4.6%**, and the test for low extremes does *not* show evidence that this exceeds 20%. Very low ratings are **rare.**

- The proportion of high extreme ratings (>= 4.5 stars) is about **24%**, and the test shows strong evidence that this is **greater** than 20%. Very high ratings are **common.**

In summary, extreme ratings are more frequent than expected in a non-polarized system, but this is driven almost entirely by very positive scores, not by very negative ones.

Discussion

Review inflation from user created reviews can cause films that should be considered some of the greatest of all time to become statistically indistinguishable from your every day sequel.

- ~29% of reviews were extreme compared to our benchmark of 20%.
- Positivity Bias: quicker to praise a movie rather than hate it.
- Selection Bias: People who enjoy a movie may be more likely to rate it. Reviewers may interpret the meaning of 5 stars differently between each other.
- Over 33,000 reviewers giving more than 33,000,000 reviews.
- Some reviewers give mostly 4-5 star ratings while others may use the whole scale, making averages harder to compute.
- A larger threshold, such as 1-10 stars could possibly help with exact percentages.
- Our data is only from one website, and does not include data from sites such as Fandango or IMDB. This is a limitation on our data.

At first, we had problems when we realized our data set was too small and did not meet the threshold. We searched for a while and finally found MovieLens which solved all our issues.

Conclusion

Overall, when analyzing more than 33 million reviews from over 33,000 reviewers we can conclude that user ratings are inflated rather than just being truly polarized. Approximately 29% of all ratings from MovieLens were extreme relative to our benchmark of 20%. This high amount of extreme rating was primarily driven by very high scores, consistent with both selection and positivity bias when it comes to ratings and who does them. The possibility of users not having a set definition for what is really defined as “5-stars” along with most reviewers not using the lower half of the rating system can cause average rating not to properly distinguish what is considered an exceptional movie versus your every day indie film. The results produced from this test imply that raw user star ratings should be taken with caution along with how further exploration into alternative scaling methods may help us correct for inflation and better separate the good from great when it comes to movies.