Dataquest Guided Project: Investigating Fandango Movie Ratings

Cindy Zhang

11/1/2020

Contents

Introduction

This is my solution to Dataquest's Guided Project from the first Probability and Statistics course, which investigates whether Fandango's rating system has changed after Walter Hickey's analysis.

More details such as the RMD and csv files can be found in the repository in GitHub. More details about the survey response variables can be found here for 2014-15 and here (https://github.com/mircealex/Movie_ratings_2016_17/blob/master/README.md) for 2016-17.

Findings

Step 1: Understanding the Data

First step as always is loading the data into R; in this project, there were two separate csv files for movies released in 2014 & 2015 and 2016 & 2017, respectively. I then made two new dataframes that isolated the columns of relevant variables.

```
fandango_score_before <- data.frame(read_csv("fandango_prior.csv"))
fandango_score_after <- data.frame(read_csv("fandango_after.csv"))
```

```
fandango_score_before_slice <- fandango_score_before %>%
  select(FILM, Fandango_Stars, Fandango_Ratingvalue, Fandango_votes, Fandango_Difference)
fandango_score_after_slice <- fandango_score_after %>%
  select(movie, year, fandango)
```

Since the goal of this project is to determine whether there has been any change in Fandango's rating system after Hickey's original analysis, the population of interest would be all movies released between 2014 and 2017.

The README.md files describe the sample selection process for each dataset as follows:

- 2014-2015: "every film that has a Rotten Tomatoes rating, a RT User rating, a Metacritic score, a Metacritic User score, and IMDb score, and at least 30 fan reviews on Fandango. The data from Fandango was pulled on Aug. 24, 2015."
- 2016-2017: "214 of the most popular movies from 2016 and 2017"

This leads me to conclude that the sampling for both datasets was not random as not all movies had an equal chance to be included in the two samples.

Step 2: Changing the Goal of our Analysis & Isolating the Samples We Need

As a result, the populations of interest needs to be changed to popular movies in 2015 and 2016, which is more representative of the two samples. This analysis will be defining "popular" the same way Hickey defines it in his analysis- 30 fan ratings or more.

I checked each sample separately if they contained popular movies, or movies with over 30 fan ratings on Fandango's website. Checking the 2014-15 sample was straightforward and only required I filter by `Fandango_votes`.

```
fandango_score_before %>%
  select(Fandango_votes) %>%
  filter(Fandango_votes < 30)
```

```
## [1] Fandango_votes
## <0 rows> (or 0-length row.names)
```

Checking the 2016-17 sample was not as straightforward as this dataset does not provide information about fan ratings. To quickly check if the sample contains enough popular movies to representative, I randomly sampled 10 movies from the dataset and manually verified the number of user ratings for each movie in the sample.

```
set.seed(1)
sample_n(fandango_score_after, size=10)
```

```
##                       movie year metascore imdb tmeter audience fandango
## 1            Hands of Stone 2016        54  6.6     45       54      4.0
## 2           The Bye Bye Man 2017        37  3.8     23       27      3.0
## 3       Our Kind of Traitor 2016        57  6.2     71       51      3.5
## 4   The Autopsy of Jane Doe 2016        64  6.8     84       71      4.5
## 5             Dirty Grandpa 2016        18  6.0     11       45      3.5
## 6                   Arsenal 2017        25  4.0      4       22      3.5
```

```
## 7  The Light Between Oceans 2016        60  7.2     59       62      4.0
## 8                   Exposed 2016        23  4.2      5       13      2.5
## 9              Jason Bourne 2016        58  6.7     56       57      4.0
## 10            Before I Fall 2017        58  6.5     66       65      3.5
##    n_metascore n_imdb n_tmeter n_audience nr_metascore nr_imdb nr_tmeter
## 1         2.70   3.30     2.25       2.70          2.5     3.5       2.0
## 2         1.85   1.90     1.15       1.35          2.0     2.0       1.0
## 3         2.85   3.10     3.55       2.55          3.0     3.0       3.5
## 4         3.20   3.40     4.20       3.55          3.0     3.5       4.0
## 5         0.90   3.00     0.55       2.25          1.0     3.0       0.5
## 6         1.25   2.00     0.20       1.10          1.0     2.0       0.0
## 7         3.00   3.60     2.95       3.10          3.0     3.5       3.0
## 8         1.15   2.10     0.25       0.65          1.0     2.0       0.0
## 9         2.90   3.35     2.80       2.85          3.0     3.5       3.0
## 10        2.90   3.25     3.30       3.25          3.0     3.0       3.5
##    nr_audience
## 1          2.5
## 2          1.5
## 3          2.5
## 4          3.5
## 5          2.0
## 6          1.0
## 7          3.0
## 8          0.5
## 9          3.0
## 10         3.0
```

Movie	Number of User Ratings
Hands of Stone	5,279
The Bye Bye Man	7,274
Our Kind of Traitor	7,272
The Autopsy of Jane Done	12,301
Dirty Grandpa	30,295
Arsenal	284
The Light Between Oceans	13,692
Exposed	1,216
Jason Bourne	56,869
Before I Fall	9,236

Finally, I created two more datasets isolating only the sample points that belong to my populations of interest that exclude movies not released in either 2015 or 2016.

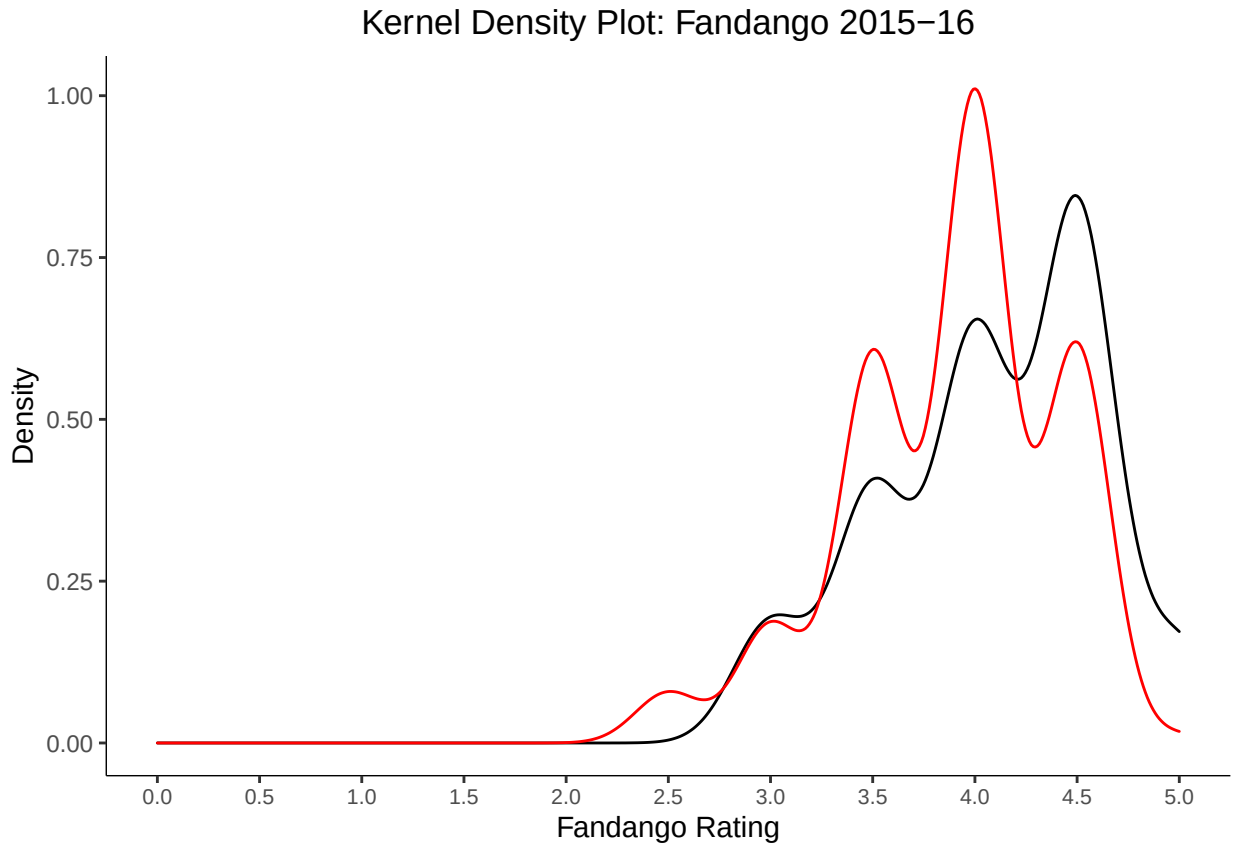
```
fandango_2015 <- fandango_score_before %>%
  separate(FILM, c("Film", "Year"), sep = "\\(") %>%
  mutate(Year = str_sub(Year, 1, 4)) %>%
  filter(Year == 2015)
```

```
fandango_2016 <- fandango_score_after %>%
  filter(year == 2016)
```

Step 3: Comparing Distribution Shapes for 2015 and 2016

I generated two kernel density plots in one graph to compare each sample's distribution shape.

```
ggplot(data=fandango_2015, aes(x=Fandango_Stars)) +
  geom_density() +
  geom_density(data=fandango_2016, aes(x=fandango), color = "red") +
  labs(title = "Kernel Density Plot: Fandango 2015-16", x="Fandango Rating", y="Density")+
  scale_x_continuous(breaks=seq(0,5, by=0.5), limits = c(0,5))+
  theme(panel.background=element_rect(fill="white"), axis.line = element_line(size=0.25, colour = "blac
```



Both sample's density plots are left-skewed, with most ratings in the sample falling in the higher range of 0 to 5. While their shapes are similar, there is a clear difference between the two- the bulk of ratings in the 2015 sample fall between 4 and 4.5 while the bulk of ratings in the 2016 sample fall between 3.5 and 4. This suggests that movies in 2016 were rated lower compared to 2015.

Step 4: Comparing Relative Frequencies

Since the two samples have different numbers of movies, it makes more sense to compare the two samples with relative frequencies, specifically percentages.

```
freq_dist_15 <- fandango_2015 %>%
  group_by(Fandango_Stars) %>%
  summarize(Freq=n()) %>%
  mutate(Percentage = Freq / nrow(fandango_2015)*100)
```

```
freq_dist_16 <- fandango_2016 %>%
  group_by(fandango) %>%
  summarize(Freq=n()) %>%
  mutate(Percentage = Freq / nrow(fandango_2016)*100)
freq_dist_15
```

```
## # A tibble: 5 x 3
##   Fandango_Stars  Freq Percentage
##            <dbl> <int>      <dbl>
## 1            3      11       8.53
## 2            3.5    23      17.8
## 3            4      37      28.7
## 4            4.5    49      38.0
## 5            5       9       6.98
```

```
freq_dist_16
```

```
## # A tibble: 6 x 3
##   fandango  Freq Percentage
##      <dbl> <int>      <dbl>
## 1      2.5     6      3.14
## 2      3      14      7.33
## 3      3.5    46     24.1
## 4      4      77     40.3
## 5      4.5    47     24.6
## 6      5       1      0.524
```

From examining the two frequency tables, I can tell that there is a difference between the two distribution, but the direction is not as obvious as it was looking at a visualization.

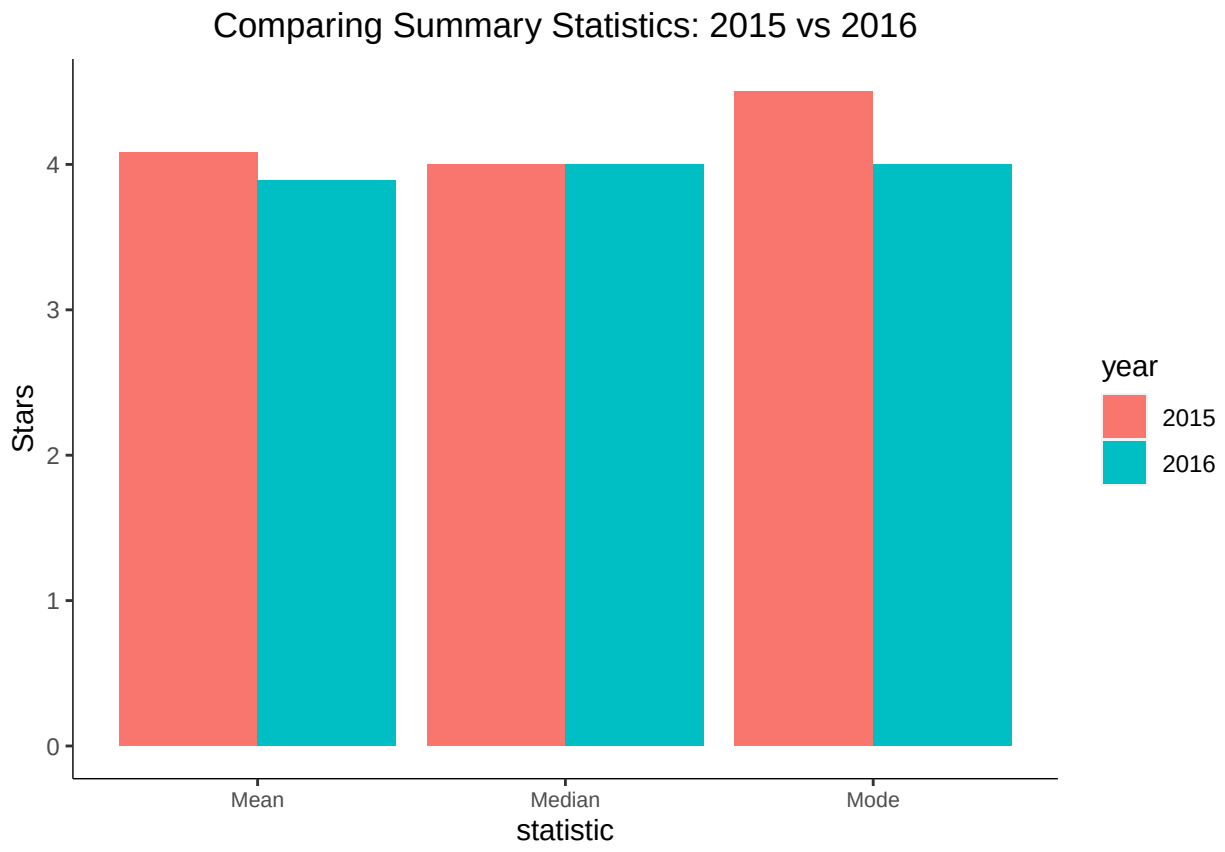
Step 5: Determining the Direction of the Change

Finally, I generated summary statistics, specifically mean, median, and mode (the most frequently occurring value found in a series of numbers), for both samples and plotted them on a graph.

```
mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}
summary_2015 <- fandango_2015 %>%
  summarize(year = "2015", Mean = mean(Fandango_Stars), Median = median(Fandango_Stars), Mode = mode(Fa
summary_2016 <- fandango_2016 %>%
  summarize(year = "2016", Mean = mean(fandango), Median = median(fandango), Mode = mode(fandango))
```

```
summary_df <- bind_rows(summary_2015, summary_2016)
summary_df <- summary_df %>%
  gather(key = "statistic", value = "value", - year)
```

```
ggplot(data=summary_df,
       aes(x=statistic, y=value, fill=year))+
  geom_bar(position="dodge", stat="identity")+
  labs(title="Comparing Summary Statistics: 2015 vs 2016", y="Stars")+
  theme(panel.background=element_rect(fill="white"), axis.line = element_line(size=0.25, colour = "blac
```



There is almost no change in median between the 2015 and 2016 samples. However, the 2015 sample mean is slightly higher than the 2016 sample mean; the 2015 sample mode is even higher than the 2016 sample mode. This corresponds with the trend of 2015 sample movies being rated higher than 2016 sample movies.

Conclusion

Based on the findings of this analysis, it appears that Fandango fixed the apparent bug that caused teh biased rounding seen in the 2015 sample, as movies in the 2016 sample aren't rated nearly as high.

Thanks to everyone who's been reading so far and stay tuned for more projects!