Movies

Jordan Ledbetter

2023-03-03

Uploading the dataset

```
library(dplyr)
```

```
##
## 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
```

```
library(tidyr)
library(ggplot2)
library(ggthemes)
library(ggplot2movies)
data(movies)
```

The range of years in movie production

```
min_year <- min(movies$year) # oldest movie
max_year <- max(movies$year) # most recent movie
range(movies$year) # range of movies
```

```
## [1] 1893 2005
```

```
max_year - min_year # find the span of years in the dataset
```

```
## [1] 112
```

The oldest movie in the dataset was released in 1893, and the most recent movie in the dataset was released in 2005. The range of years of production of the movies in the dataset is 112 years.

The proportion of movies that have their budget included in the dataset and the proportion of movies that do not have their budget included in the dataset

```
budget <- sum(!is.na(movies$budget)) # find the sum of all movies with a budget
budget / count(movies) # find proportion of all movies with a budget within the whole dataset
```

```
##            n
## 1 0.08870858
```

```
no_budget <- sum(is.na(movies$budget)) # find the sum of all movies without a budget
no_budget / count(movies) # find proportion of all movies without a budget within the whole dataset
```

```
##           n
## 1 0.9112914
```

There are a total of 5,215 movies in the dataset that have included their buget, and there are a total of 53,573 movies that did not include their budget in the dataset. We can take the total number of movies for each subset and divide it by the total number of movies to find the proportions. After performing these calculations, approximately 0.088 of the movies in the dataset included their budget and approximately 0.911 did not include their budget.

The Top 5 Most Expensive Movies

```
most_expensive <- movies |>
  filter(!is.na(movies$budget)) |> # filter out the movies that do not include their budget
  select(title, budget) |> # return only these columns
  arrange(desc(budget)) |> # arrange movies in decreasing order of budget
  head(5) # return first five movies at the top of the list

most_expensive
```

```
## # A tibble: 5 x 2
##   title                                budget
##   <chr>                                 <int>
## 1 Spider-Man 2                       200000000
## 2 Titanic                            200000000
## 3 Troy                               185000000
## 4 Terminator 3: Rise of the Machines 175000000
## 5 Waterworld                         175000000
```

A tibble of the five most expensive movies in the dataset, in decreasing order.

The Top 5 Longest Movies

```
longest_movies <- movies |>
  select(title, length) |> # return only these columns
  arrange(desc(length)) |> # arrange movies in decreasing order of length
  head(5) # return top five movies in the list

longest_movies
```

```
## # A tibble: 5 x 2
##   title                                                length
##   <chr>                                                 <int>
## 1 Cure for Insomnia, The                                 5220
## 2 Longest Most Meaningless Movie in the World, The       2880
## 3 Four Stars                                             1100
## 4 Resan                                                   873
## 5 Out 1                                                   773
```

A tibble of the five longest movies in the dataset, in decreasing order.

The Shortest and Longest Short Film (in Minutes)

```
short_short_films <- movies |>
  filter(movies$Short == 1) |> # filter in movies that are classified as short movies
  select(title, length) |> # return these columns only
  arrange(length) |> # arrange in increasing order by length
  head(1) # return the film at the top of the list

sum(movies$Short == 1 & movies$length == 1) # number of short movies that are one minute long
```

```
## [1] 165
```

```
short_short_films # return the shortest short film
```

```
## # A tibble: 1 x 2
##   title               length
##   <chr>                <int>
## 1 17 Seconds to Sophie      1
```

```
long_short_films <- movies |>
  filter(movies$Short == 1) |> # filter in movies that are classified as short movies
  select(title, length) |> # return these columns only
  arrange(desc(length)) |> # arrange in decreasing order by length
  head(1) # return the film at the top of the list

long_short_films # return the longest short film
```

```
## # A tibble: 1 x 2
##   title                length
##   <chr>                 <int>
## 1 10 jaar leuven kort     240
```

```
short_short_films$length # length of the shortest short film
```

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

```
long_short_films$length # length of the longest short film
```

```
## [1] 240
```

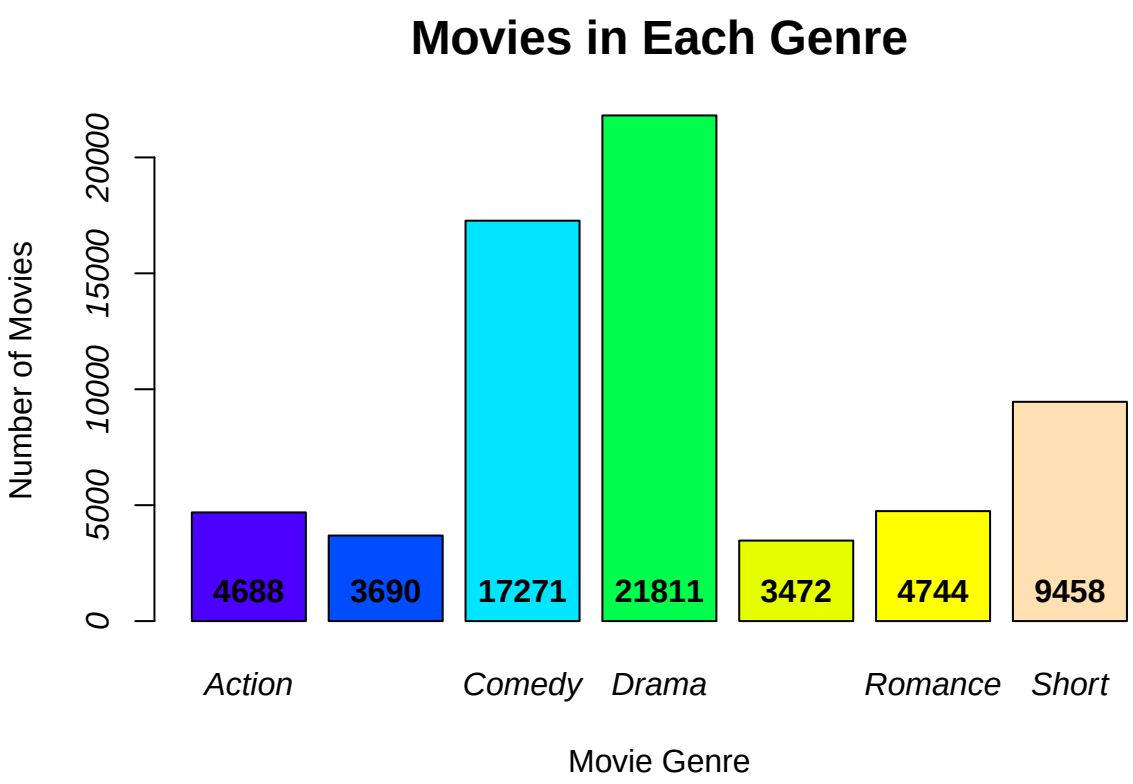
There are 165 short movies in the dataset that are tied for the shortest movie, with 1 minute; however, the one that appears at the top of the list, due to the dataset being in alphabetically order, is '17 Seconds to Sophie'. The longest short movie in the dataset is '10 jaar leuven kort' at 240 minutes long.

Number of Movies of Each Genre

```
movie_genres <- movies |>
  select(Action, Animation, Comedy, Drama, Documentary, Romance, Short) |> # select columns of genres
  colSums() # sum the number of movies in each genre

# create a barplot to show number of movies in each genre
genre_plot <- barplot(movie_genres,
        main = "Movies in Each Genre",
        xlab = "Movie Genre",
        ylab = "Number of Movies",
        font = 3,
        cex.main = 1.5,
        col = topo.colors(length(movie_genres)))

# add text to each bar to showcase the number of movies per genre
text (genre_plot, 0, movie_genres,
      cex = 1, pos = 3, font = 2)
```



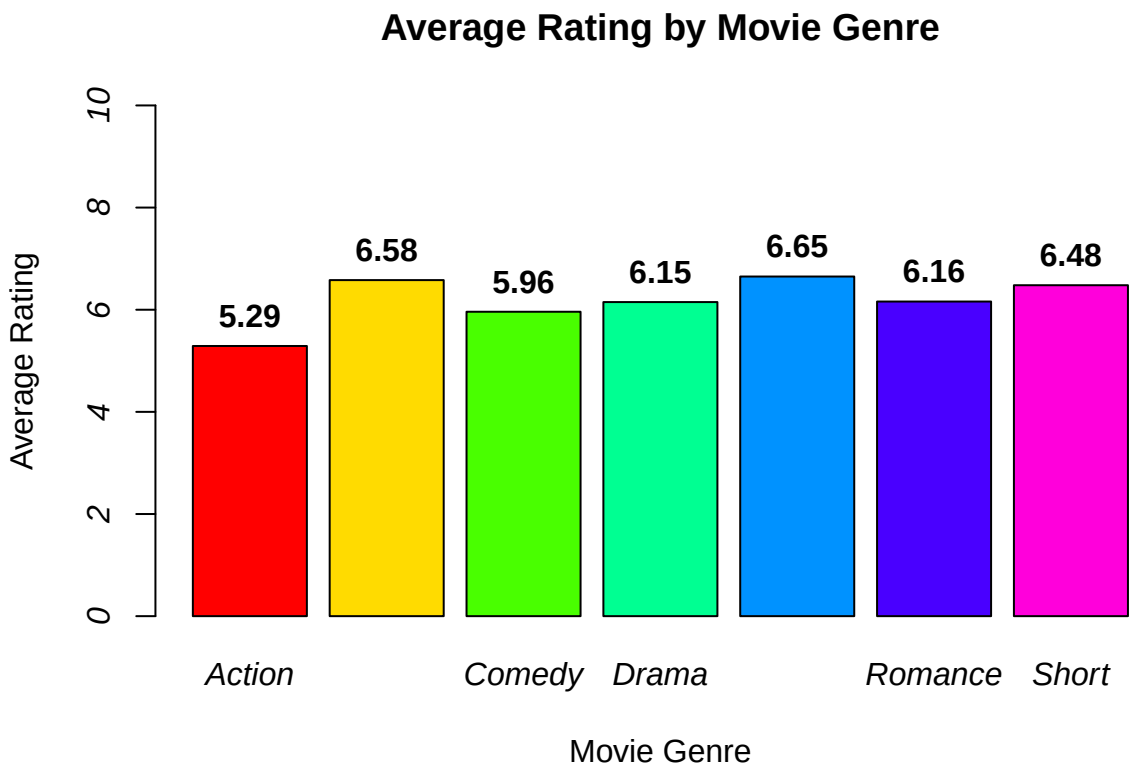
This barplot shows the number of movies that are in each genre. Note that some movies are apart of more than one genre.

Average Rating of All Movies Within Each Genre

```
# computes the average rating for each genre
avg_rating <- sapply(movies[,18:24], function(x) round(mean(movies$rating[x == 1]),2))

# create a barplot for average ratings by genre
avg_rating_plot <- barplot(avg_rating,
                           main = "Average Rating by Movie Genre",
                           xlab = "Movie Genre",
                           ylab = "Average Rating",
                           font = 3,
                           ylim = c(0,10),
                           col = rainbow(length(avg_rating)))

# add text to each bar to showcase the average rating per genre
text(x = avg_rating_plot, y = avg_rating,
     labels = avg_rating,
     cex = 1, pos = 3, font = 2)
```



Barplot of the average rating of movies by genre. Movies were rated on a scale of 1 to 10.

Average Rating of All Movies Within Each Genre

Produced between 2000-2005

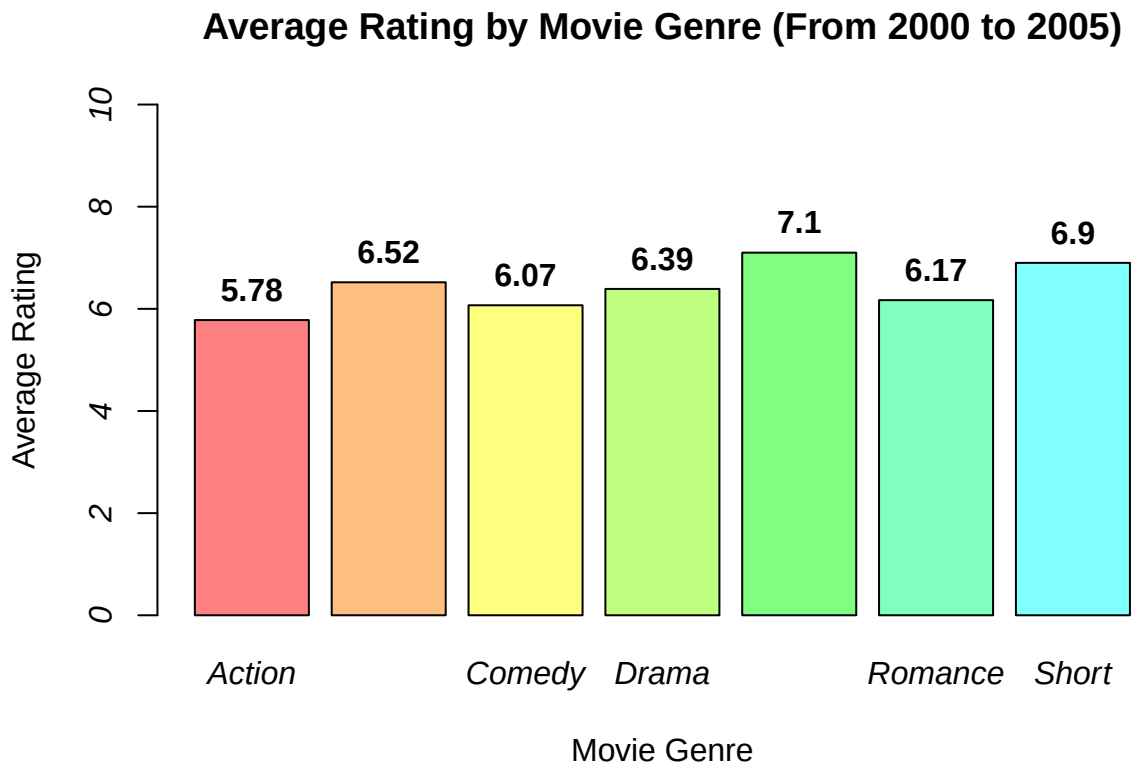
```
# subsets movies that were produced between 2000-2005
movies_2000s <- subset(movies, movies$year == c(2000:2005))

# computes the average rating of movies in the 2000s by genre
avg_rating_2000s <- sapply(movies_2000s[,18:24], function(x) round(mean(movies_2000s$rating[x == 1]),2))

# creates a barplot for average ratings by genre
avg_rating_plot_2000s <- barplot(avg_rating_2000s,
                          main = "Average Rating by Movie Genre (From 2000 to 2005)",
                          xlab = "Movie Genre",
                          ylab = "Average Rating",
                          font = 3,
                          ylim = c(0,10),
                          col = hsv(seq(0,1 - 1/12,length.out = 12), 0.5 , 1))

# add text to each bar to showcase the average rating per genre
text(x = avg_rating_plot_2000s, y = avg_rating_2000s,
```

```
    labels = avg_rating_2000s,
    cex = 1, pos = 3, font = 2)
```



Barplot of the average rating of movies by genre from 2000 to 2005. Movies were rated on a scale of 1 to 10.

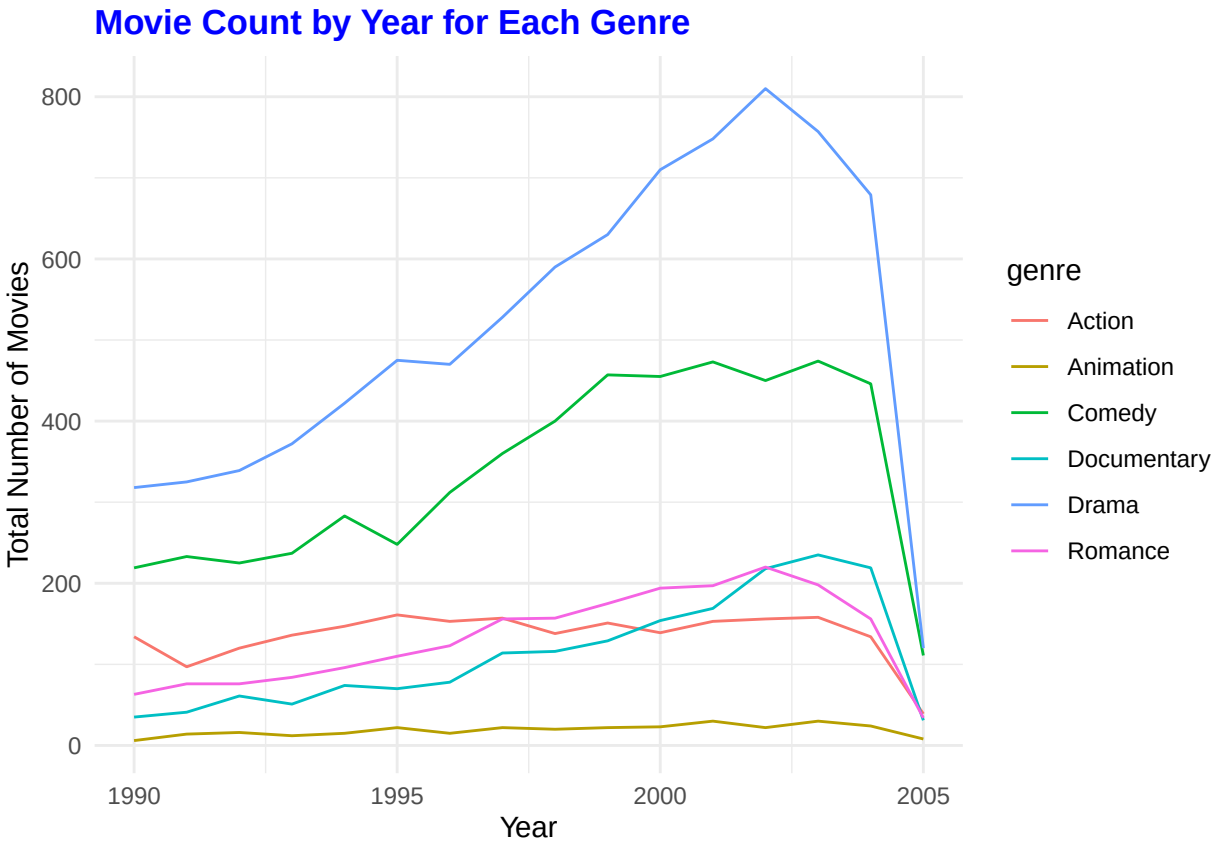
Number of Movies Produced By Year, from 1990 to 2005, For Each Genre

```
movies_produced <- movies |>
  filter(year >= 1990 & Short == 0) |> # filters movies that were produced between 1990 and 2005, and f
  pivot_longer(cols = c(18:23), names_to = 'genre', values_to = 'count') |> # creates new table with to
  select(year, genre, count) |> # columms
  group_by(genre, year) |>
  summarise(total = sum(count))
```

```
## `summarise()` has grouped output by 'genre'. You can override using the
## `.groups` argument.
```

```
# creates a line plot with a line for each genre
ggplot(movies_produced, aes(x = year, y = total, color = genre)) +
```

```
  geom_line() +
  labs(title = "Movie Count by Year for Each Genre",
       x = "Year", y = "Total Number of Movies") +
  theme_minimal() +
  theme(plot.title = element_text(face = "bold", colour = "blue"))
```



The line plot shown above shows the amount of movies produced over time by genre. Each line in the plot represents a different genre and can be determined in the legend provided. According to the plot, drama movies are the most common and have increased significantly overtime. On the other hand, animated movies are the least common in the dataset and have been produced relatively constant overtime.

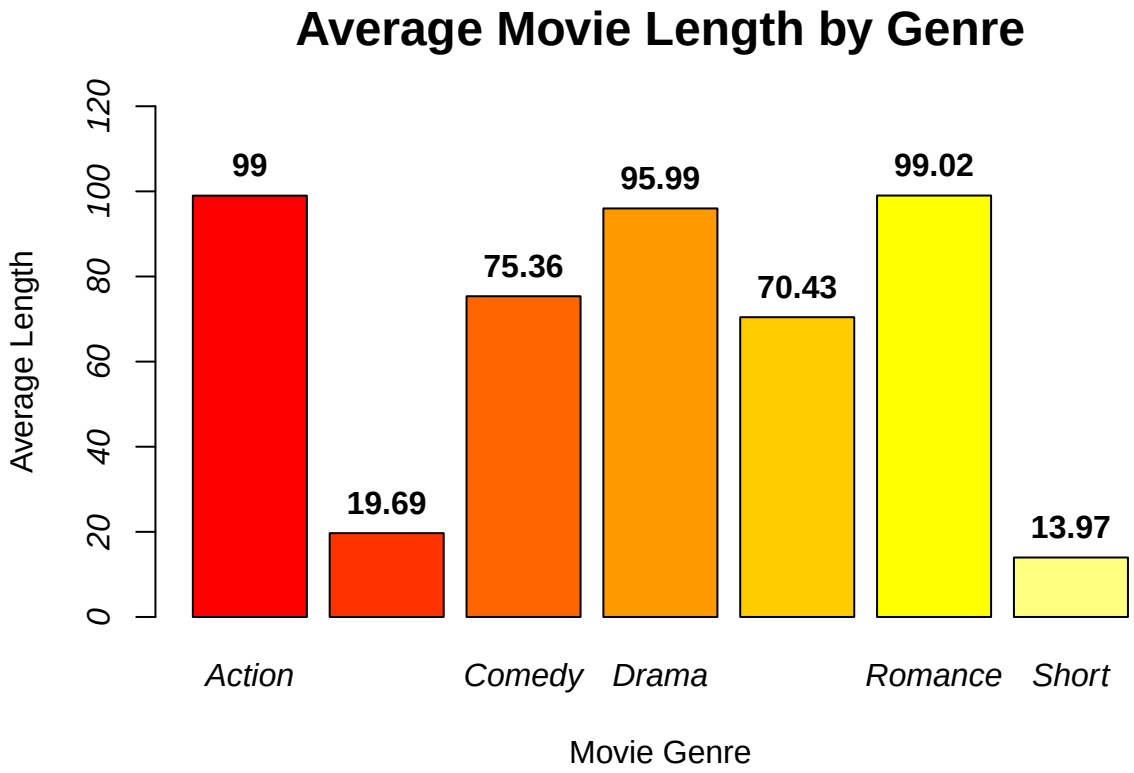
Average Length of All Movies Within Each Genre

```
# computes the average length of movies by genre
avg_length <- sapply(movies[,18:24], function(x) round(mean(movies$length[x == 1]),2))

# creates a barplot for average movie length by genre
avg_length_plot <- barplot(avg_length,
                           main = "Average Movie Length by Genre",
                           xlab = "Movie Genre",
                           ylab = "Average Length",
```

```
                          ylim = c(0,120),
                          font = 3,
                          cex.main = 1.5,
                          col = heat.colors(length(avg_length)))

# add text to each bar to showcase the average movie length per genre
text(x = avg_length_plot, y = avg_length,
     labels = avg_length,
     cex = 1, pos = 3, font = 2)
```

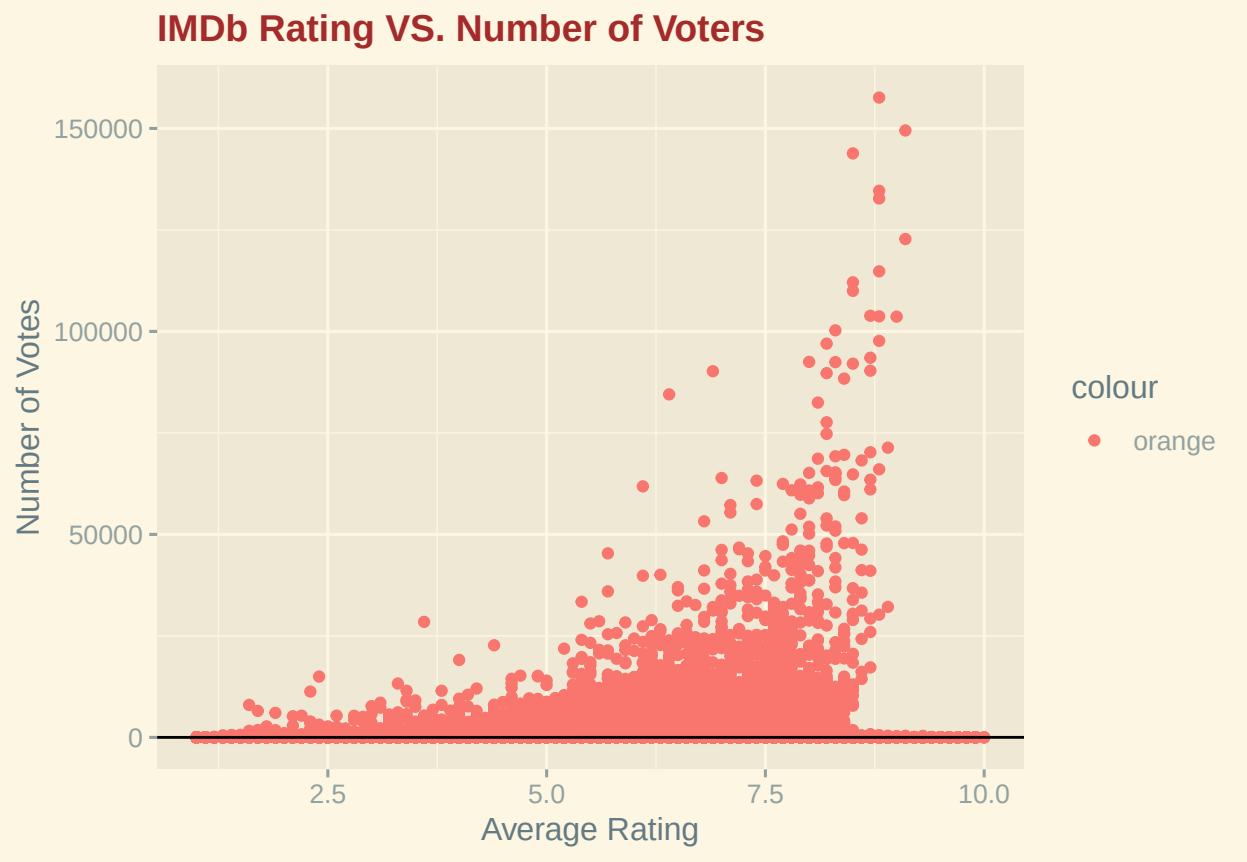


The barplot is arranged by genre and shows the mean length of movies for each category. According to the barplot shown above, the movie genres that had the longest runtimes were Romance, Action, and Drama. The movie genres with the shortest runtimes were Animations and Shorts.

Relationship between Average Rating and IMDb Voters

```
ggplot(data = movies, aes(x = rating, y = votes, color = 'orange')) +
   geom_point() +
   geom_abline() +
   labs(title = "IMDb Rating VS. Number of Voters",
         x = "Average Rating", y = "Number of Votes") +
```

```
  theme_solarized_2() +
  theme(plot.title = element_text(face = "bold", colour = "brown"))
```



The scatterplot shows the relationship between the number of voters who rated each movie and the average rating for each movie. According to the scatterplot, there seems to be a moderate relationship between average rating of a movie and how many people voted for the movie. There are very few votes for movies with low ratings, and there is a significant increase in the number of votes where the mean movie rating is between 6 and 9. The plot then decreases significantly as the average rating reaches 10.

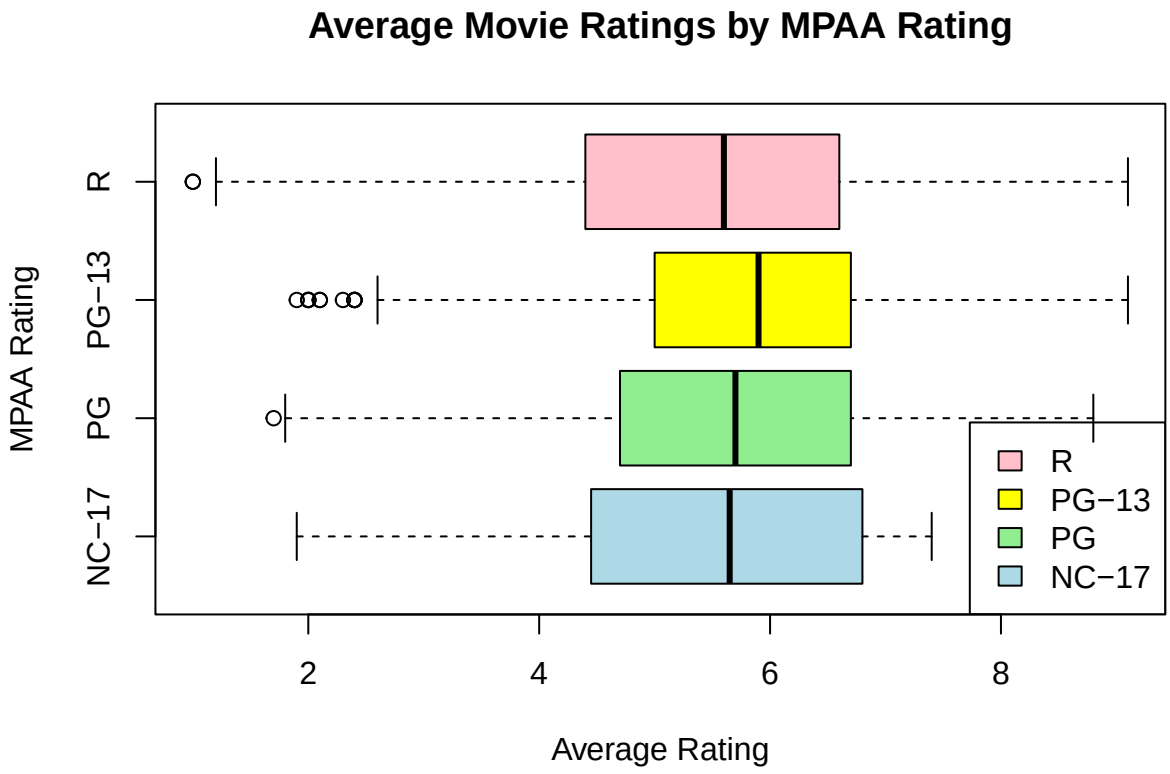
Relationship between MPAA Rating and Average Rating

```
# subsets the data to contain movies with MPAA ratings
mpaa_rating <- subset(movies, movies$mpaa == 'PG' | movies$mpaa == 'PG-13' | movies$mpaa == 'NC-17' | m
# create variable 'mpaa' to contain the MPAA ratings from the mpaa_rating dataset
mpaa <- mpaa_rating$mpaa

# creates a boxplot to show the relationship between MPAA ratings and IMDb ratings
boxplot(rating~mpaa, data = mpaa_rating, horizontal = TRUE,
        col = c('lightblue','lightgreen','yellow','pink'),
        main = "Average Movie Ratings by MPAA Rating",
        xlab = "Average Rating",
```

```
        ylab = "MPAA Rating")

# creates a legend to organize MPAA ratings
legend("bottomright", legend = c('R','PG-13','PG','NC-17'),
       fil = c('pink','yellow','lightgreen','lightblue'))
```



The boxplot is arranged by MPAA rating and shows the mean rating of movies for each category. According to the boxplot shown above, there were no significant differents in IMDb ratings for different MPAA ratings. In fact, all MPAA ratings had a very similar mean values.