Mini Project 2

PSTAT 100: Spring 2024 (Instructor: Ethan P. Marzban)

AARTI GARAYE (aartigaraye)

May 12, 2024

Introduction

The Internet Movie Database (IMdB) is a leading source of film- and media-related statistics. In this mini-project, we will explore various aspects about films (e.g. runtimes, ratings, etc.) and compare and contrast across several different characteristics.

Data Overview

There are two main data files associated with this project: one named basics.csv and the other named ratings.csv.

The basics data frame contains the following variables:

tconst (string) - alphanumeric unique identifier of the title

titleType (string) – the type/format of the title (e.g. movie, short, tvseries, tvepisode, video, etc)

primaryTitle (string) – the more popular title / the title used by the filmmakers on promotional materials at the point of release

originalTitle (string) - original title, in the original language

isAdult (boolean) - 0: non-adult title; 1: adult title

startYear (YYYY) – represents the release year of a title. In the case of TV Series, it is the series start year

endYear (YYYY) – TV Series end year. ' ' for all other title types

runtimeMinutes – primary runtime of the title, in minutes

genres (string array) – includes up to three genres associated with the title

The ratings dataframe contains the following variables:

tconst (string) - alphanumeric unique identifier of the title

averageRating – weighted average of all the individual user ratings

numVotes - number of votes the title has received

Part I: Data Preprocessing

Loading all the necessary libraries so that we can work on efficiently processing the data present in the different files. Some of the libraries are also loaded to work on tests and graphs.

```
library(readr)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.3     v purrr     1.0.2
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.0     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.0
-- 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 e
```

```
library(dplyr)
library(ggplot2)
```

Reading the csv files.

```
unzip("/home/jovyan/100-sp24/Mini_Projects/MP02/data/basics.csv.zip", exdir = "data")
basics <- read.csv("data/basics.csv", stringsAsFactors = FALSE)
ratings <- read.csv("/home/jovyan/100-sp24/Mini_Projects/MP02/data/ratings.csv")
```

Now we should split the rows to make the data easier to sort later on in the project

```
basics %>%
  mutate(genres = strsplit(as.character(genres), ",")) %>%
  unnest(genres)
```

```
# A tibble: 1,356,877 x 9
   tconst    titleType primaryTitle      originalTitle isAdult startYear endYear
   <chr>     <chr>     <chr>             <chr>           <int> <chr>     <chr>
 1 tt0000009 movie     Miss Jerry        Miss Jerry          0 1894      "\\N"
 2 tt0000147 movie     The Corbett-Fitz~ The Corbett-~       0 1897      "\\N"
 3 tt0000147 movie     The Corbett-Fitz~ The Corbett-~       0 1897      "\\N"
 4 tt0000147 movie     The Corbett-Fitz~ The Corbett-~       0 1897      "\\N"
 5 tt0000502 movie     Bohemios          Bohemios            0 1905      "\\N"
 6 tt0000574 movie     The Story of the~ The Story of~       0 1906      "\\N"
 7 tt0000574 movie     The Story of the~ The Story of~       0 1906      "\\N"
 8 tt0000574 movie     The Story of the~ The Story of~       0 1906      "\\N"
 9 tt0000591 movie     The Prodigal Son  L'enfant pro~       0 1907      "\\N"
10 tt0000615 movie     Robbery Under Ar~ Robbery Unde~       0 1907      "\\N"
# i 1,356,867 more rows
# i 2 more variables: runtimeMinutes <chr>, genres <chr>
```

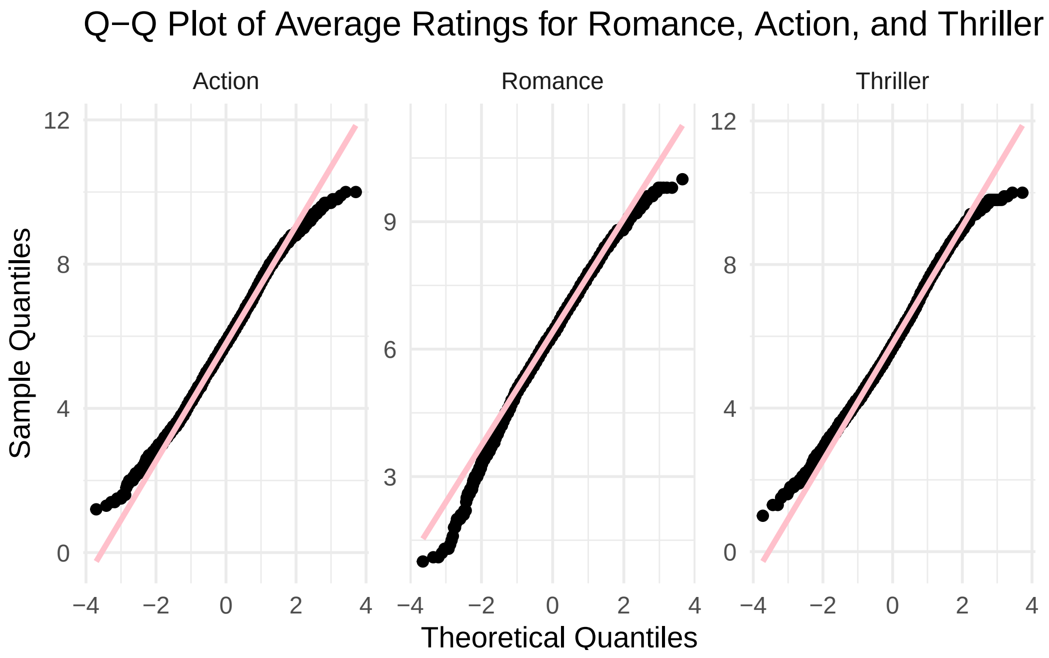
Part II: Report Questions

a)

To check if there appears to a significant relationship of average ratings across some genres, more than two, we can use the AVNOVA test for variance. If the variance is more, it would seem like there is a significant different between the average ratings because they are more widely spread apart from the average mean.

However, before proceeding we need to check normality in the data at least approximately and the homogeneity of variances. To do that, we could see the distribution by plotting a graph to check if it approximately takes the normal bell-curve or not.

```
merged_df <- merge(basics, ratings, by = "tconst")

genres_of_interest <- c("Romance", "Action", "Thriller")
filtered_df <- merged_df %>% filter(genres %in% genres_of_interest)

filtered_df %>%
  ggplot(aes(sample = averageRating)) +
  geom_qq() +
  geom_qq_line(linewidth = 1, col = "pink") +
  facet_wrap(~genres, scales = "free") +
  theme_minimal() +
  ylab("Sample Quantiles") +
  xlab("Theoretical Quantiles") +
  ggtitle("Q-Q Plot of Average Ratings for Romance, Action, and Thriller Genres")
```



From the facet wrap of different graphs of different genres, we can safely assume that they approximately follow a normal distribution. Now that we have assured visually ourselves, we can proceed with the ANOVA tests to analyze the variances across the three different genres.

Before conducting the ANOVA test, we'll also need to check the assumption of homogeneity of variances, which means that the variances of the data in different groups should be approximately equal. Since we haven't taught how to do this, we will just assume that varainces across the genres are approximately equal. So let's proceed with our anova testing

```r
anova_result <- aov(averageRating ~ genres, data = filtered_df)
summary(anova_result)
```

```
               Df Sum Sq Mean Sq F value Pr(>F)    
genres          2    706   352.8     152 <2e-16 ***
Residuals   13652  31692     2.3                   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test will provide us with the F-statistic and p-value. A significant p-value (typically < 0.05) indicates that there is a significant difference in average ratings across the genres.

In this test our p-value is 2e-16 which is significantly less than 0.05, therefore, we can reject out null hypothesis and confidently say that there is a significant diffference in average ratings across the romance, action, and thriller genres.

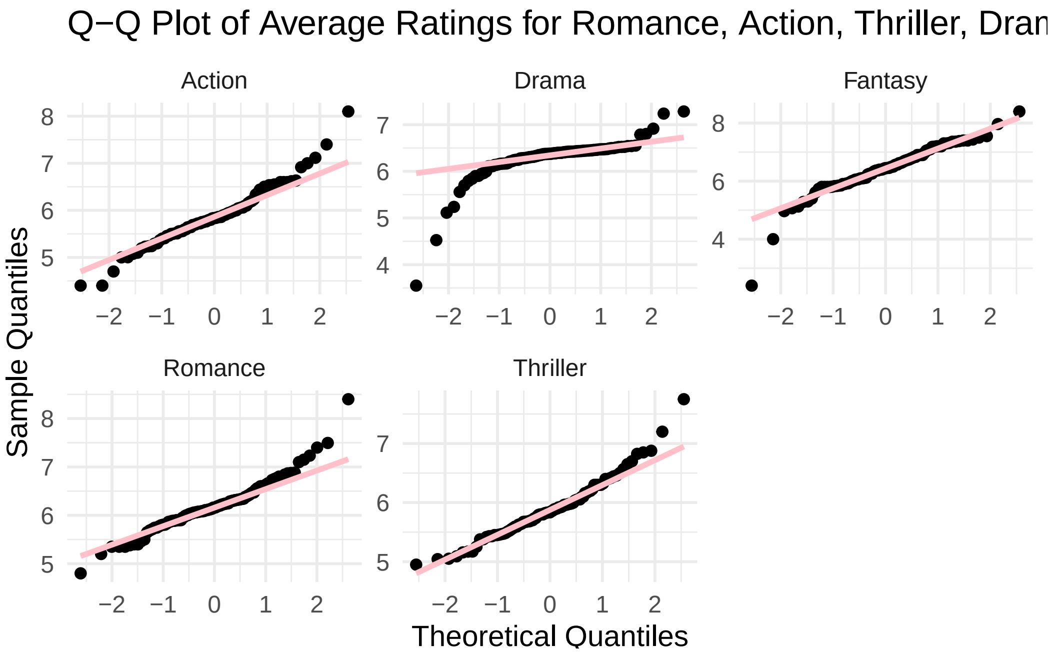
b)

Now, to address the question of if there is a significance difference between the five genres I selected — romance, action, thriller, drama, and fantasy — in their average rating over the years, we gotta perform the same tests. So, let's proceed with checking if the approximately follow a normal distribution

```r
selected_genres <- c("Romance", "Action", "Thriller", "Drama", "Fantasy")
filtered_genre_df <- merged_df %>% filter(genres %in% selected_genres)

average_ratings_years <- filtered_genre_df %>%
  group_by(genres, startYear) %>%
  summarise(avg_rating = mean(averageRating, na.rm = TRUE))
```

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

```r
average_ratings_years %>%
  ggplot(aes(sample = avg_rating)) +
  geom_qq() +
  geom_qq_line(linewidth = 1, col = "pink") +
  facet_wrap(~genres, scales = "free") +
  theme_minimal() +
  ylab("Sample Quantiles") +
  xlab("Theoretical Quantiles") +
  ggtitle("Q-Q Plot of Average Ratings for Romance, Action, Thriller, Drama, and Fantasy Genr
```



From these graphs we can tell that these genres almost approximately follow a normal distribution, so now we can proceed with our ANOVA testing to check if there is any significant differences between average ratings over the years for these genres.

```
result <- aov(avg_rating ~ genres + Error(startYear/genres), data = average_ratings_years)
```

```
Warning in aov(avg_rating ~ genres + Error(startYear/genres), data =
average_ratings_years): Error() model is singular
```

```
summary(result)
```

```
Error: startYear
           Df Sum Sq Mean Sq F value Pr(>F)
genres      4   5.26  1.3156   2.506 0.0459 *
Residuals 116  60.91  0.5251
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Error: startYear:genres
           Df Sum Sq Mean Sq F value Pr(>F)
genres      4  27.10   6.776   28.72 <2e-16 ***
Residuals 383  90.35   0.236
```

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

We are introducing the Error term in our aov because if we don't then it would consider each year as an individual observation rather than considering startYear as a repeated measure within genres. Since we are interested in comparing average ratings within genres over years and startYear is a repeated measure (each genre has multiple observations for different years), a repeated measures ANOVA is more appropriate. The Error() term in aov() allows us to specify the within-subject factor (startYear) and the between-subject factor (genres), indicating that startYear is a repeated measure within each level of genres.

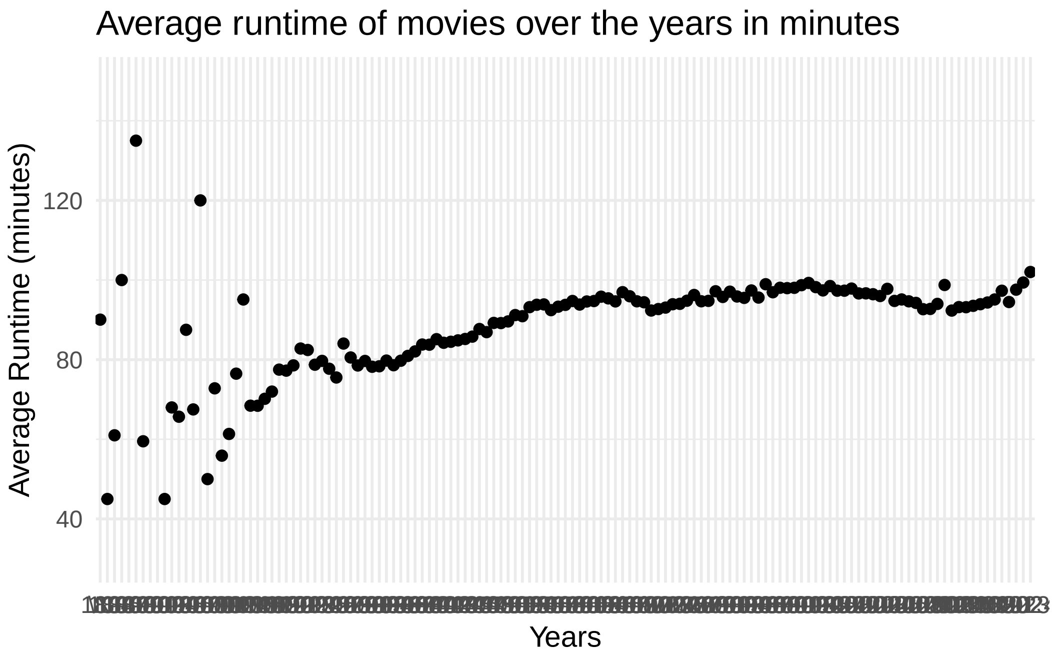
Since the p-value is less than 0.05 we can safely say that we reject the null hypothesis and that there is a significant difference between the average rating over the five genres — romance, action, thriller, drama, and fantasy — over the years.

c)

We can address the question of run time of movies changing over the time visually by using a line graph. However, we need to filter the dataframes so we can have it sorted by movies and their runtimes over the years.

```r
movies_runtime <- merged_df %>%
  filter(titleType == "movie")

movies_runtime$runtimeMinutes[movies_runtime$runtimeMinutes == "\\N"] <- NA
movies_runtime$runtimeMinutes <- as.numeric(movies_runtime$runtimeMinutes, na.rm = TRUE)

avg_runtime <- movies_runtime %>%
  group_by(startYear) %>%
  summarise(average_runtime = mean(runtimeMinutes, na.rm = TRUE))

avg_runtime %>%
  ggplot(aes(x = as.factor(startYear), y = average_runtime)) +
  geom_point() +
  theme_minimal() +
  ylab("Average Runtime (minutes)") +
  xlab("Years") +
  ylim(30,150) +
  ggtitle("Average runtime of movies over the years in minutes")
```

```
Warning: Removed 3 rows containing missing values or values outside the scale range
(`geom_point()`).
```



From the scatterplot we can see that the average run time over the years have slowing been converging between 90-120 minutes which is 1:30-2:00 hours for a movie which seems reasonable. Something interesting is that in early years, there were movies that were less than 80 minutes, some even less than 60 minutes which is less than an hour. Rest assured, we don't see those anymore.

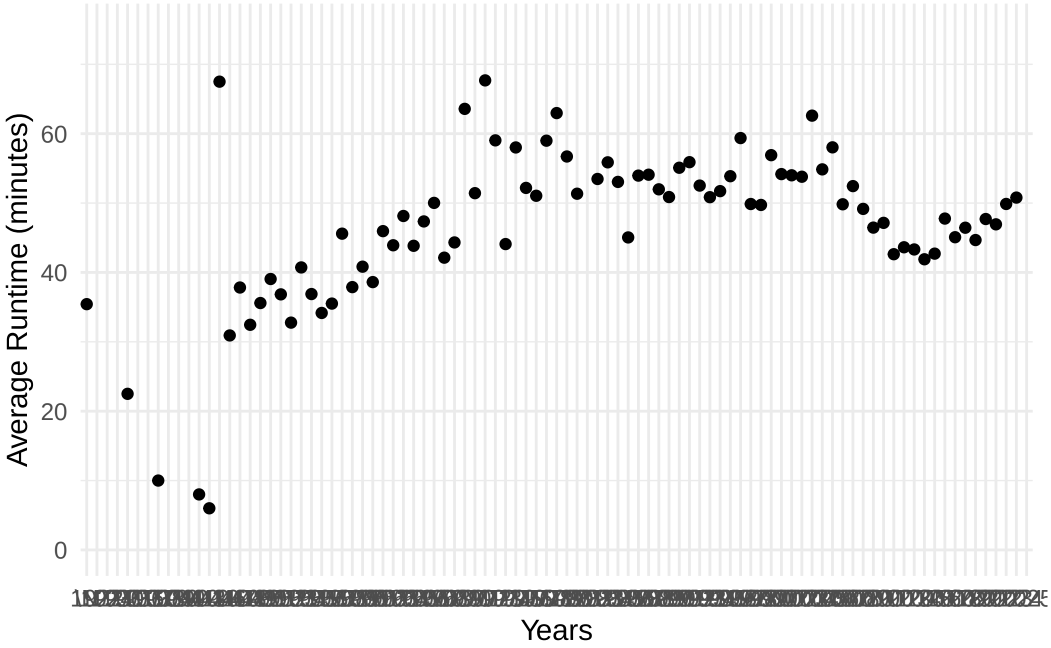
d)

Let's do the similar thing we did in the previous problem where we can examine the average runtime of each TV series episode over the time using a scatterplot.

```r
tv_series <- merged_df %>%
  filter(titleType == "tvSeries")

tv_series$runtimeMinutes[tv_series$runtimeMinutes == "\\N"] <- NA
tv_series$runtimeMinutes <- as.numeric(tv_series$runtimeMinutes, na.rm = TRUE)

avg_runtime_series <- tv_series %>%
  group_by(startYear) %>%
  summarise(average_runtime_series = mean(runtimeMinutes, na.rm = TRUE))

avg_runtime_series %>%
  ggplot(aes(x = as.factor(startYear), y = average_runtime_series)) +
  geom_point() +
  theme_minimal() +
  ylab("Average Runtime (minutes)") +
  xlab("Years") +
  ylim(0,75)
```

```
Warning: Removed 10 rows containing missing values or values outside the scale range
```

```
(`geom_point()`).
```



```r
  ggtitle("Average runtime of TV Series over the years in minutes")
```

```
$title
[1] "Average runtime of TV Series over the years in minutes"

attr(,"class")
[1] "labels"
```

From the graph above, we can tell that slowly the average runtime of each tv series episode is convering to approximately 50 minutes as years go on. Compared to the years in 1990s, it seems that the runtime of each episode have gone longer but compared to 2000s, it has gone shorter.

e)

For my own question, I would like to see whose average runtime is more romance movies vs average runtime of Mystery movies and see if there is a significant difference in their runtimes.

```r
romance_mystery <- merged_df %>%
  filter(genres %in% c("Romance", "Mystery"))

romance_mystery$runtimeMinutes[romance_mystery$runtimeMinutes == "\\N"] <- NA
romance_mystery$runtimeMinutes <- as.numeric(romance_mystery$runtimeMinutes, na.rm = TRUE)

average_runtime_by_genre <- romance_mystery %>%
  group_by(genres) %>%
  summarize(avg_runtime = mean(runtimeMinutes, na.rm = TRUE))

print(average_runtime_by_genre)
```

```
# A tibble: 2 x 2
  genres  avg_runtime
  <chr>         <dbl>
1 Mystery        83.9
2 Romance        97.4
```

From the table above, we can see that romance movies have a longer runtime than the mystery movies. One reason could be that romance movies are usually watched with comfort people and we like to spend time with our comfort people so we are okay spending more time watching romance movies than mystery movies.

Acknowledgments

Data dictionaries were taken from the official IMdB Documentation, and datasets were sourced from https://datasets.imdbws.com/.