IMDb Genre Analysis (2000–2020)

Saad Dahbani

Introduction

In this project, I explored how average IMDb ratings changed over time for different movie genres between 2000 and 2020. I used two raw IMDb datasets: `title.basics.tsv` and `title.ratings.tsv`. I cleaned the data, split multi-genre movies into separate rows, calculated average ratings for each genre and year, and visualized the results using `ggplot2`.

This analysis shows how genres like Drama consistently ranked higher, while others like Horror had lower average ratings. It was a hands-on exercise in data cleaning and visualization in R which I enjoyed.

```
# Load IMDb .tsv files and Replace \\N to na.
basics <- read_tsv("C:\\Users\\Saad\\Desktop\\imdb-genre-project\\title.basics.tsv.gz", na = "\\N")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 11589102 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (5): tconst, titleType, primaryTitle, originalTitle, genres
## dbl (4): isAdult, startYear, endYear, runtimeMinutes
##
## 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.
```

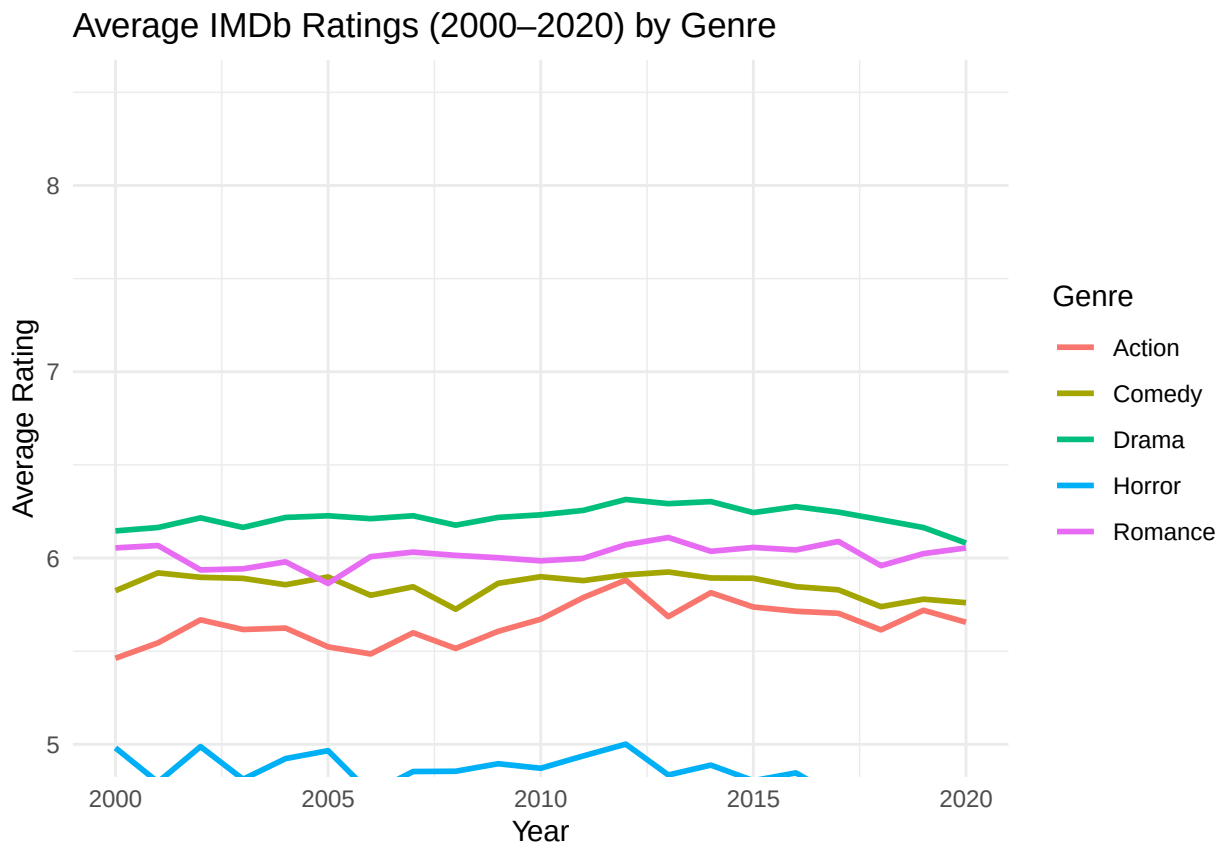
```
ratings <- read_tsv("C:\\Users\\Saad\\Desktop\\imdb-genre-project\\title.ratings.tsv.gz", na = "\\N")
```

```
## Rows: 1559241 Columns: 3
## -- Column specification --------------------------------------------------------
## Delimiter: "\t"
## chr (1): tconst
## dbl (2): averageRating, numVotes
##
## 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
# Clean the Data
movies_cleaned <- basics %>%
  filter(titleType == "movie") %>%
  filter(!is.na(startYear)) %>%
  mutate(startYear = as.integer(startYear)) %>%
  filter(startYear >= 2000, startYear <= 2020) %>%
  select(tconst, primaryTitle, startYear, genres)

# Join Ratings
movie_data <- movies_cleaned %>%
  inner_join(ratings, by = "tconst")

# Split Multi-Genre Movies
movie_data_cleaned <- movie_data %>%
  separate_rows(genres, sep = ",")

# Summarize Ratings by Genre & Year
summary_data <- movie_data_cleaned %>%
  group_by(genres, startYear) %>%
  summarise(avg_rating = mean(averageRating), .groups = "drop")

# Filter to Top Genres
top_genres <- c("Drama", "Comedy", "Action", "Horror", "Romance")

summary_data_filtered <- summary_data %>%
  filter(genres %in% top_genres)
```

Final Plot

```r
ggplot(data = summary_data_filtered, aes(x = startYear, y = avg_rating, color = genres)) +
  geom_line(linewidth = 1) +
  labs(
    title = "Average IMDb Ratings (2000-2020) by Genre",
    x = "Year",
    y = "Average Rating",
    color = "Genre"
  ) +
  theme_minimal() +
  coord_cartesian(ylim = c(5, 8.5))
```



Conclusion

This project gave me experience working with real-world messy data. I practiced using tools like dplyr for filtering, joining, and summarizing, and ggplot2 for creating clear, informative plots.

The final plot highlights interesting genre trends over the years, showing that Drama often earns the highest ratings, while Horror struggles to gain viewer appreciation. I'm proud of how far I've come learning R and excited to apply these skills to new datasets and challenges.