

Movie Data Analysis Project

This project involves analyzing movie ratings, genres, and runtime data using R. Below is a summary of the steps involved in the project.

Contents

1. 1. Installing Necessary Libraries
2. 2. Loading the Necessary Libraries
3. 3. Reading the Data
4. 4. Joining Datasets
5. 5. Exploring the Data
6. 6. Data Cleaning
7. 7. Frequency Distribution Analysis
8. 8. Movie Ratings Analysis
9. 9. Bayesian Rating Analysis
10. 10. Genre and Average Rating Correlation

1. Installing Necessary Libraries

```
``r
install.packages('readr')
install.packages('dplyr')
install.packages('ggplot2')
install.packages('tidyr')
install.packages('plotly')
``
```

2. Loading the Necessary Libraries

```
``r
library(readr)
library(dplyr)
library(ggplot2)
library(tidyr)
library(plotly)
``
```

3. Reading the Data

```
``r
ratings <- read_tsv('raw_data/title.ratings.tsv')
basics <- read_tsv('raw_data/title.basics.tsv')
``
```

4. Joining Datasets

```
```r
joined_data <- full_join(basics, ratings)
View(joined_data)
```
```

5. Exploring the Data

```
```r
test <- joined_data
```
```

6. Data Cleaning

6.1 Counting Missing Values

```
```r
sum(is.na(test))
na_counts <- lapply(test, function(test) sum(is.na(test)))
```
```

6.2 Identifying Outliers

```
```r
boxplot(test, col = 'lightblue', main = 'Distribution by Column', xlab = 'Variables', ylab =
'Values')
```
```

6.3 Removing NA Values

```
```r
test <- na.omit(test)
```
```

6.4 Removing Duplicates

```
```r
total_duplicates <- sum(duplicated(test))
duplicates <- test %>%
 group_by(across(everything())) %>%
 filter(n() > 1) %>%
 ungroup()
test <- test[!duplicated(test),]
sum(duplicated(test))
```
```

6.5 Cleaning Columns

```
```r
cleaned_v2 <- cleaned_v1 %>%
 select(-tconst, -originalTitle, -isAdult, -endYear)
```
```

```
View(cleaned_v2)
...
```

7. Frequency Distribution Analysis

7.1 Frequency Table of `titleType`

```
```r
View(fdtQl(cleaned_v2$titleType))
...
```

### 7.2 Selecting Movies Only

```
```r
only_movies = filter(cleaned_v2, titleType == 'movie' | titleType == 'short' | titleType ==
'tvMovie')
...
```

8. Movie Ratings Analysis

8.1 Top 100 Movies

```
```r
sorted_by_num_votes <- only_movies %>% arrange(desc(numVotes))
top_100_best_rated_movies <- sorted_by_num_votes[1:100,]
top_100_best_rated_movies_according_to_rating <- top_100_best_rated_movies %>%
arrange(desc(averageRating))
...
```

### 8.2 Average Rating by Genre

```
```r
ggplot(average_rating_of_movies, aes(x=genres, y=avg_rating, fill=avg_rating)) +
geom_bar(stat = 'identity') +
  scale_fill_gradient(low = 'blue', high = 'red')
...
```

8.3 Genre Analysis

```
```r
ggplot(genre_analysis, aes(x = genres, y = avg_rating, fill = genres)) +
 geom_bar(stat = 'identity', show.legend = FALSE)
...
```

### 8.4 Average Rating by Genre Over the Years

```
```r
plot <- average_ratings_by_genre_year %>%
  ggplot(aes(startYear, avg_rating, color = genres)) +
  geom_line() +
  facet_wrap(~genres) +
```

```

  theme(legend.position = 'none')
ggplotly(plot) %>% layout(hovermode = 'x unified')
...

```

8.5 Correlation of Average Rating and Length Over the Years

```

```r
ggplot(genre_data_scaled, aes(x = startYear, y = ScaledValue, color = Factor, linetype =
Factor)) +
 geom_line(aes(group = Factor, genres), size = 1.2) +
 facet_wrap(~genres)
...

```

### 8.6 Number of Movies Released Per Year

```

```r
ggplot(number_of_movies_per_year, aes(x=startYear, y=count)) +
  geom_line() +
  facet_wrap(~)
...

```

8.7 Number of Movies by Genre

```

```r
ggplot(number_of_movies_each_year, aes(x=startYear, y=count)) +
 geom_line(aes(group = genres, color = genres), size=0.5) +
 facet_wrap(~genres, scales='free_y')
...

```

## 9. Bayesian Rating Analysis

```

```r
C <- mean(horror_movies$rating, na.rm = TRUE)
m <- 100
horror_movies <- horror_movies %>%
  mutate(adjusted_rating = (C * m + horror_movies$averageRating *
horror_movies$numVotes) / (horror_movies$numVotes + m))
...

```

10. Genre and Average Rating Correlation

```

```r
grouped_by_genre$runtimeMinutes <-
as.numeric(as.character(grouped_by_genre$runtimeMinutes))
...

```