Research project

Final Report

Team: Khrystyna Korets, Bozhena Sai, Sofiia Sydorchuk

Data Set: IMDB Movies Dataset

Kaggle: https://www.kaggle.com/datasets/ashishjangra27/imdb-movies-dataset

Goal of the project

Our team aims to study the main factors that affect the success of films. We will use data analysis on an existing dataset to find patterns and relationships between important aspects like audience ratings, box office earnings, and other key factors. Based on our findings, we will give recommendations to film studios on how to make successful films with high ratings and strong financial performance.

Dataset Information

Data set that our team chose for this research project is having the data of 2.5 Million movies/series listed on the official website of IMDB. IMDB is one of the main sources which people use to judge the movie and it‘s rating plays an important role for a lot of people watching a movie. This data set contain next information (columns) about every movie (data types changed from original data set during data cleaning):

Field	Description	Data type
id	Movie ID on IMDb platform	<chr>
name	The title of the movie	<chr>
year	Year of release	<chr>
rating	Average audience rating	<dbl>
certificate	Age rating of the movie (suitable audience age)	<dbl>
duration	Length of the movie	<int> (in minutes)
genre	Genre(s) the movie belongs to	<chr> (comma separated)
votes	Number of votes received on IMDb	<dbl>
gross_income	Total income from all sources before tax deductions	<dbl>
directors_name	Name of the director(s)	<chr>
stars_name	List of movie stars who	<chr> (comma separated)

Field	Description	Data type
	starred in the movie	
`description`	Short description or synopsis of the movie	`<chr>`

Analysis Objectives

Install packages and libraries for analysis.

```r
library(tidyverse)

## ── Attaching core tidyverse packages ──────────────────────── tidyverse
2.0.0 ──
## ✔ dplyr     1.1.4     ✔ readr     2.1.5
## ✔ forcats   1.0.0     ✔ stringr   1.5.1
## ✔ ggplot2   3.5.1     ✔ tibble    3.2.1
## ✔ lubridate 1.9.3     ✔ tidyr     1.3.1
## ✔ purrr     1.0.2
## ── Conflicts ──────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all
conflicts to become errors

library(ggplot2)
library(dplyr)
library(tidytext)
library(wordcloud)

## Loading required package: RColorBrewer
```

Data Cleaning

The data cleaning started by removing rows with missing or incorrect values. Any rows where gross_income was 0, directors_name was “nm0000000”, or rating was 11.0 were deleted. These values didn’t make sense for analysis and could affect results.

Then, the certificate column was cleaned up by turning its text values into numbers. For example, certificates like “G” or “TV-G” were changed to 1, “6+” became 6, and so on. Anything that didn’t fit into a known group was set to 0.

The gross_income and votes columns had commas in their values, so those were removed, and the data was converted into numbers. The duration column, which had extra text, was cleaned to keep only the numbers and was turned into integers.

```r
movies <- read.csv("movies.csv")
movies <- movies %>% filter(gross_income != 0)
movies <- movies %>% filter(directors_name != "nm0000000")
movies <- movies %>% filter(rating != 11.0)
#String tags to integers
movies <- movies %>%
  mutate(certificate = case_when(
    certificate %in% c("G", "TV-G", "TV-Y", "E", "K-A", "Passed", "Approved")
~ 1,
    certificate %in% c("6+") ~ 6,
    certificate %in% c("12", "PG-12") ~ 12,
    certificate %in% c("PG-13", "TV-13", "MA-13", "M/PG", "M") ~ 13,
    certificate %in% c("R", "TV-MA", "MA-17", "R-15", "T") ~ 17,
    certificate %in% c("NC-17", "R-18", "18", "AO") ~ 18,
    certificate %in% c("PG", "TV-PG", "GP") ~ 1,
    TRUE ~ 0
  ))
movies <- movies %>%
  mutate(
    gross_income = gsub(",", "", gross_income),
    gross_income = as.numeric(gross_income)
  )
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `gross_income = as.numeric(gross_income)`.
## Caused by warning:
## ! NAs introduced by coercion
```

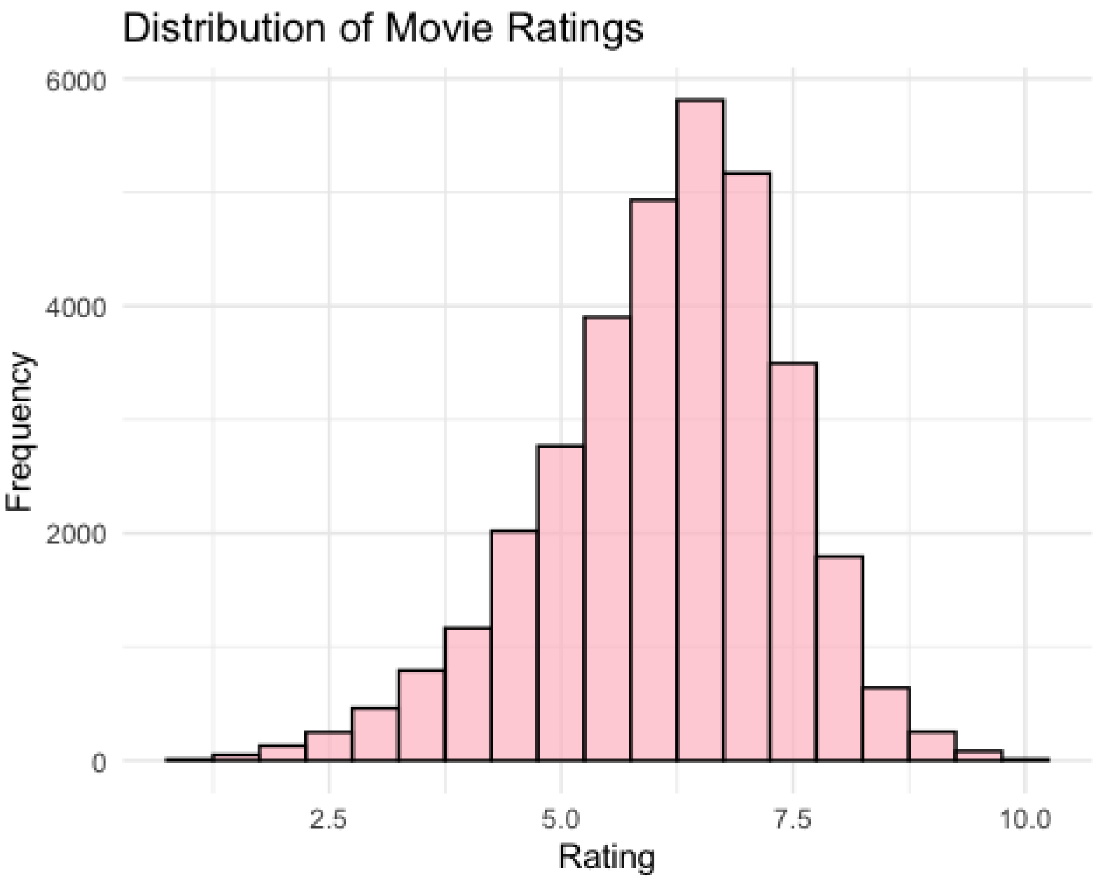
```
movies <- movies %>%
  mutate(
    votes = gsub(",", "", votes),
    votes = as.numeric(votes)
  )

movies <- movies %>%
  mutate(duration = as.integer(gsub("[^0-9]", "", duration)))
head(movies)
```

```
##           id                     name   year rating certificate duration
## 1  tt0068646            The Godfather (1972)    9.2          17      175
## 2  tt5113044 Minions: The Rise of Gru (2022)    7.3           1       87
## 3  tt7657566        Death on the Nile (2022)    6.3          13      127
## 4  tt8115900             The Bad Guys (2022)    6.9           1      100
## 5 tt10809742                Collision (2022)    3.8          17       99
## 6  tt0071562    The Godfather Part II (1974)    9.0          17      202
##                           genre   votes gross_income
## 1                  Crime, Drama 1798749    134966411
## 2 Animation, Adventure, Comedy    1220    134966411
## 3         Crime, Drama, Mystery  121063    134966411
## 4 Animation, Adventure, Comedy   23090    134966411
## 5        Crime, Drama, Thriller     656    134966411
## 6                 Crime, Drama 1239038     57300000
##                    directors_id                           directors_name
## 1                     nm0000338                     Francis Ford Coppola
## 2 nm0049633,nm1556070,nm3646390 Kyle Balda,Brad Ableson,Jonathan del Val
## 3                     nm0000110                          Kenneth Branagh
## 4                     nm2010048                           Pierre Perifel
## 5                     nm1926494                         Fabien Martorell
## 6                     nm0000338                     Francis Ford Coppola
##                                    stars_id
## 1  nm0000008,nm0000199,nm0001001,nm0000473
## 2  nm0136797,nm1853544,nm0000273,nm0378245
## 3  nm5290643,nm0000906,nm0000110,nm1258970
## 4  nm0005377,nm0549505,nm5377144,nm0732497
## 5 nm1339181,nm0456810,nm12651040,nm7357127
## 6  nm0000199,nm0000134,nm0000380,nm0000473
##                                                  stars_name
## 1          Marlon Brando,Al Pacino,James Caan,Diane Keaton
## 2   Steve Carell,Pierre Coffin,Alan Arkin,Taraji P. Henson
## 3 Tom Bateman,Annette Bening,Kenneth Branagh,Russell Brand
## 4         Sam Rockwell,Marc Maron,Awkwafina,Craig Robinson
## 5   Tessa Jubber,Langley Kirkwood,Zoey Sneedon,Bonko Khoza
## 6      Al Pacino,Robert De Niro,Robert Duvall,Diane Keaton
##
description
## 1                The aging patriarch of an organized crime dynasty in
postwar New York City transfers control of his clandestine empire to his
reluctant youngest son.
## 2
The untold story of one twelve-year-old's dream to become the world's
greatest supervillain.
## 3
While on vacation on the Nile, Hercule Poirot must investigate the murder of
a young heiress.
## 4                                              Several reformed yet
misunderstood criminal animals attempt to become good, with some disastrous
results along the way.
## 5
Freedom always comes at a price.
## 6 The early life and career of Vito Corleone in 1920s New York City is
portrayed, while his son, Michael, expands and tightens his grip on the
family crime syndicate.
```

Data Visualization

During data visualization, we wanted to explore trends and patterns in the movie industry. By analyzing ratings, box office earnings, genres, and other factors, we aimed to uncover how different elements contribute to a film’s success. Visuals helped us spot interesting tendencies, like how budgets affect performance, which genres are most popular, and how audience preferences have changed over time.

```
ggplot(movies, aes(x = rating)) +
  geom_histogram(binwidth = 0.5, fill = "pink", color = "black", alpha = 0.7)
+
  labs(title = "Distribution of Movie Ratings",
       x = "Rating",
       y = "Frequency") +
  theme_minimal()
```

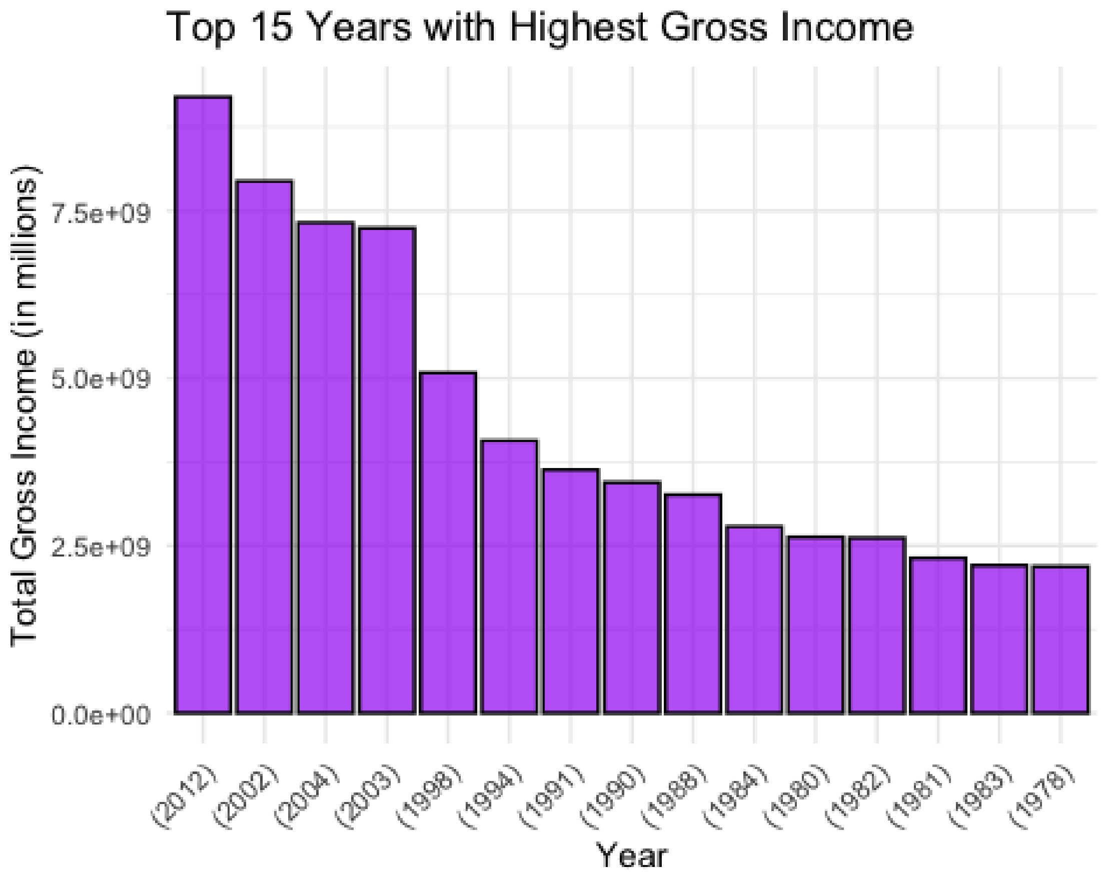


```
mean_rating <- mean(movies$rating)
mean_rating

## [1] 6.136386
```

The plot reveals the distribution of ratings across all movies in the dataset. It shows that the majority of movies (nearly 4,500) have ratings around 6 out of 10. While the distribution of ratings appears to follow a roughly normal shape, it is clear that more movies have ratings below the overall mean (which is 6.15 points).

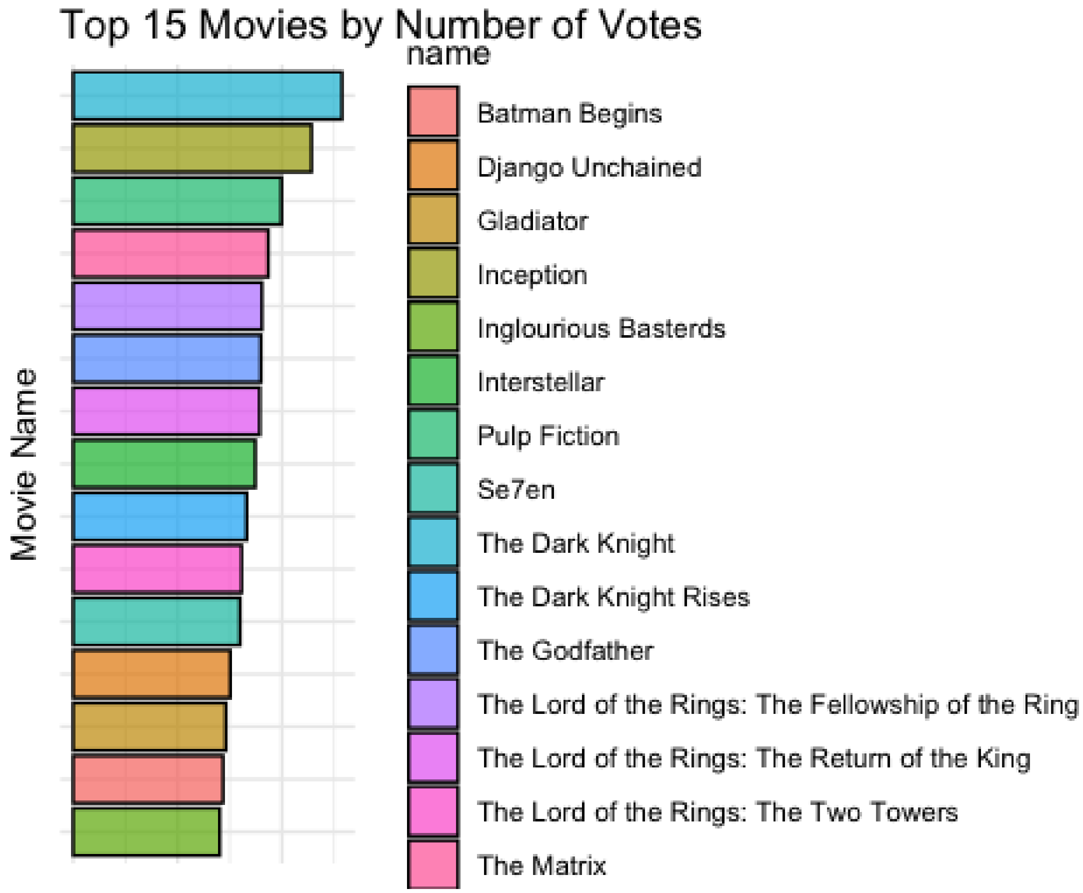
```
total_gross_income_per_year <- movies %>%
  group_by(year) %>%
  summarise(TotalGrossIncome = sum(gross_income)) %>%
  arrange(desc(TotalGrossIncome))

top_15_years <- head(total_gross_income_per_year, 15)
```

```r
ggplot(top_15_years, aes(x = reorder(year, -TotalGrossIncome), y =
TotalGrossIncome)) +
  geom_bar(stat = "identity", fill = "purple", color = "black", alpha = 0.7)
+
  labs(title = "Top 15 Years with Highest Gross Income",
       x = "Year",
       y = "Total Gross Income (in millions)") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



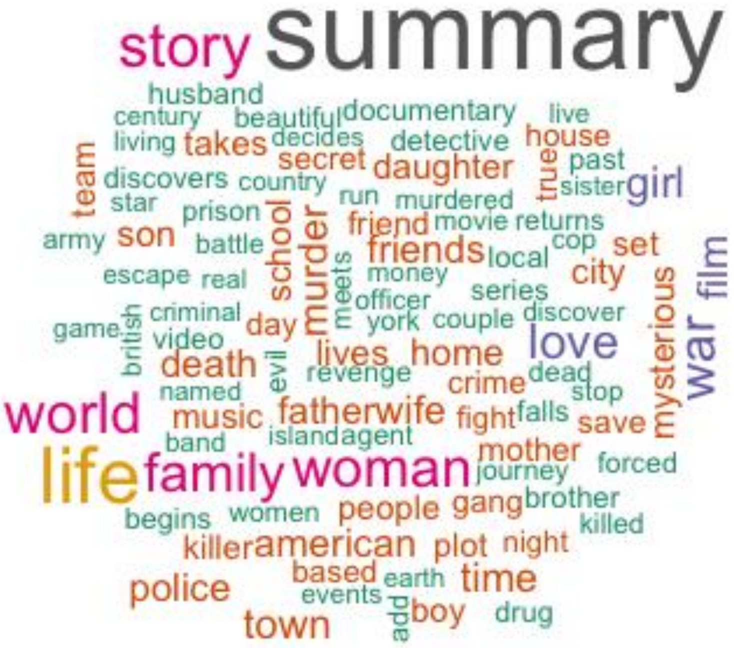
The most successful year in the movie industry was 2019, with nearly $15 billion in earnings, significantly outpacing subsequent years. For example, in 2022, earnings were slightly over $11 billion with 4 billion difference.

```r
top_movies_by_votes <- movies %>%
  arrange(desc(votes)) %>%
  head(15)

ggplot(top_movies_by_votes, aes(x = reorder(name, votes), y = votes, fill =
name)) +
  geom_bar(stat = "identity", color = "black", alpha = 0.7) +
  labs(title = "Top 15 Movies by Number of Votes",
       x = "Movie Name",
       y = "Number of Votes") +
  coord_flip() +
  theme_minimal() +
  theme(axis.title.x = element_blank(),
        axis.text.x = element_blank(),
        axis.text.y = element_blank())
```



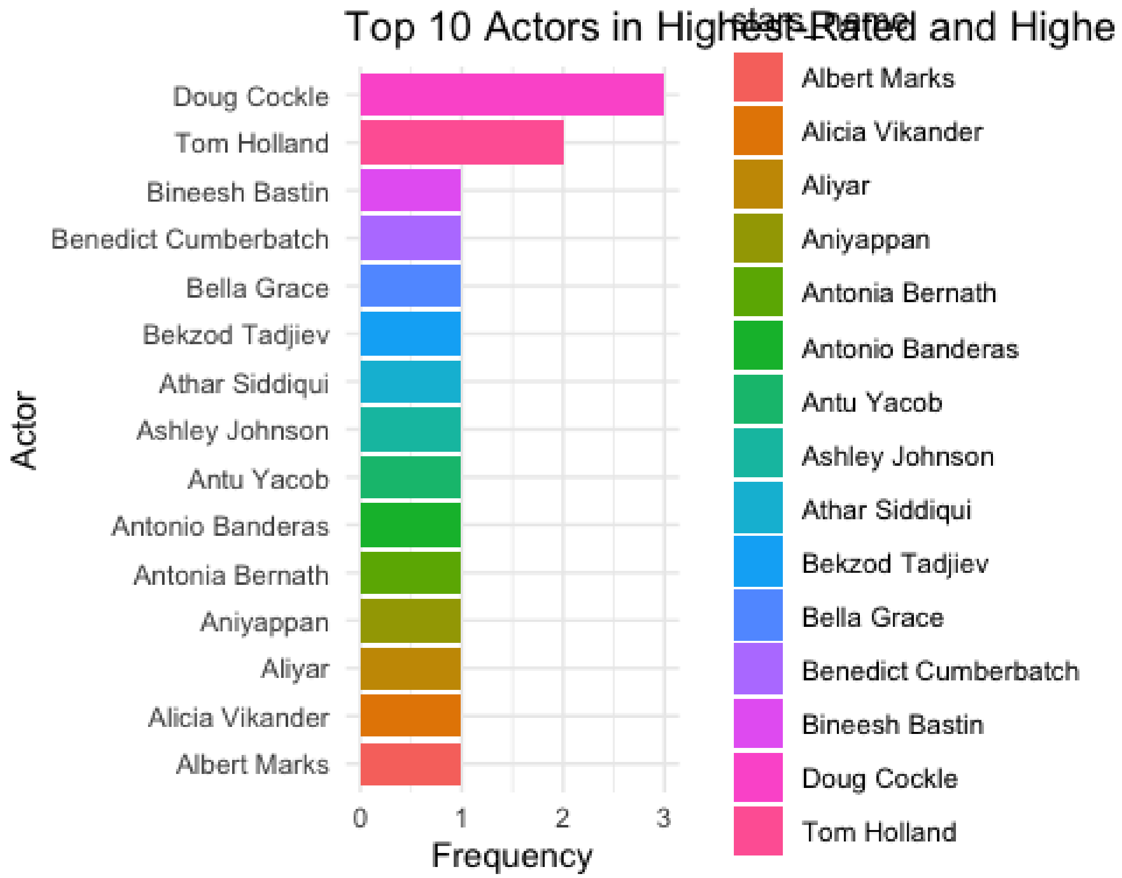
The most-voted movie in our dataset is *Batman Begins*. Also making the top five are *Django Unchained*, *Gladiator*, *Inception*, and *Inglourious Basterds*.

```
word_counts <- movies %>%
  unnest_tokens(word, description) %>%
  anti_join(stop_words, by = join_by(word)) %>%
  count(word, sort = TRUE)

top_words <- word_counts %>% head(100)
wordcloud(words = top_words$word, freq = top_words$n, min.freq = 1,
          scale = c(3,0.5), colors = brewer.pal(8, "Dark2"))
```



There you might see top 100 words used in movie descriptions by their frequency.

```
top_movies <- movies %>%
  filter(!is.na(rating), !is.na(gross_income)) %>%
  slice_max(rating, n = 15) %>%
  bind_rows(slice_max(movies, gross_income, n = 10))

top_actors <- top_movies %>%
  separate_rows(stars_name, sep = ",") %>%
  count(stars_name, sort = TRUE)

top_actors <- top_actors %>% head(15)

ggplot(top_actors, aes(x = reorder(stars_name, n), y = n, fill = stars_name))
+
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(title = "Top 10 Actors in Highest-Rated and Highest-Grossing Movies",
       x = "Actor", y = "Frequency") +
  theme_minimal()
```



This plot showing us top of most successful actors and actress's by number of highest-rated and highest-grossing movies they starred in. The most successful is Doug Cockle. The first woman in rating is Ashley Johnson who placed only 8-th place.

Hypothesis test

1. Movies in popular genres tend to have higher gross income.

In this analysis, we are testing the relationship between a movie's genre and its gross income.

Null Hypothesis (H_0):

There is **no relationship** between a movie's genre and its gross income.
$\mathbf{H_0}$: The mean gross income is the same across all genres.

Alternative Hypothesis (H_1):

Movies in **popular genres** tend to have higher gross income compared to those in less popular genres.
$\mathbf{H_1}$: The mean gross income differs across genres.

To test this hypothesis we use **ANOVA test**.

Key Concepts of ANOVA (Analysis of Variance)

In summary, ANOVA compares the **variance** within groups and between groups to assess whether there are significant differences in the means. The key steps are:

1. **Calculate** the total, between-group, and within-group sums of squares.

$$\text{SST},\text{SSB},\text{SSW}$$

2. **Compute** the mean squares by dividing the sums of squares by their respective degrees of freedom.

$$\text{MSB} = \frac{\text{SSB}}{\text{dfB}}, \quad \text{MSW} = \frac{\text{SSW}}{\text{dfW}}$$

3. **Determine** the F-statistic by comparing the mean square between groups to the mean square within groups.

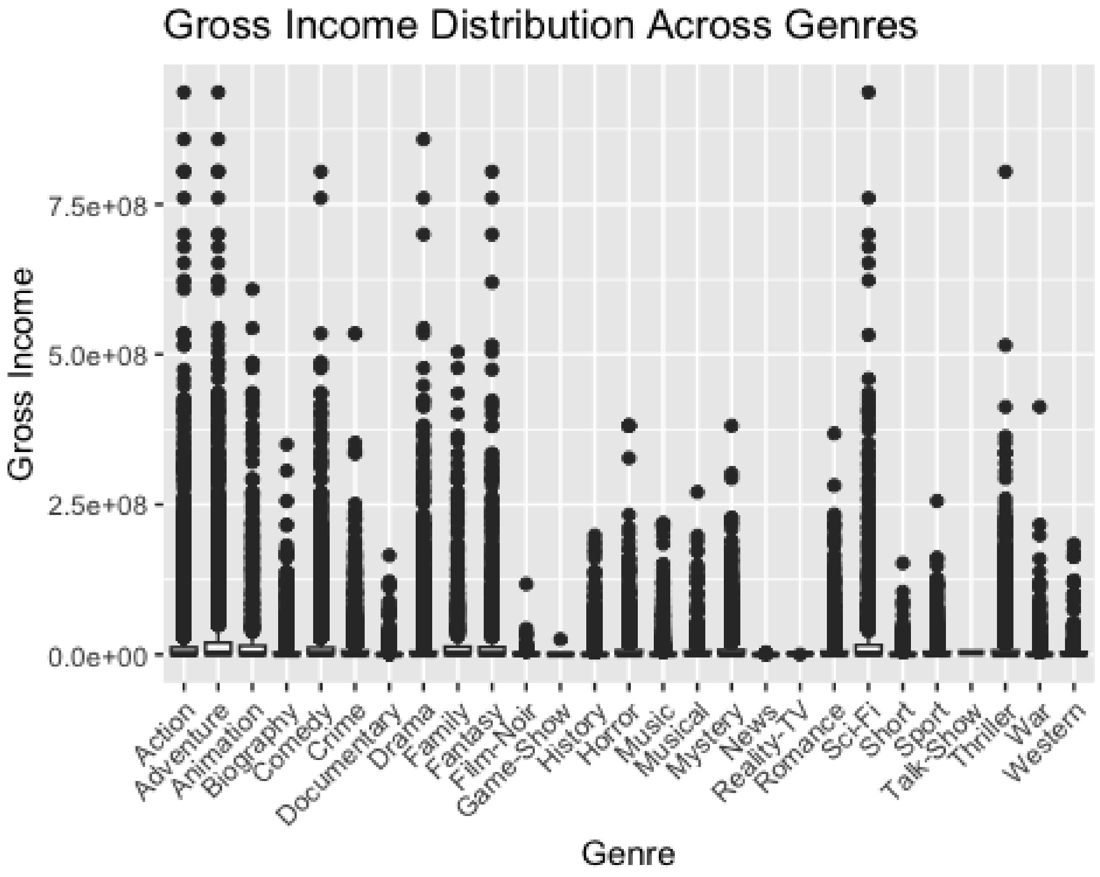
$$F = \frac{\text{MSB}}{\text{MSW}}$$

4. **Evaluate** the p-value to decide whether to reject the null hypothesis.

R Implementation

```
movies_data_separated <- movies %>%
  # Split genres by commas and gather them into separate rows
  separate_rows(genre, sep = ",") %>%
  # Trim any leading/trailing spaces
  mutate(genre = str_trim(genre))

# Step 2: Perform One-way ANOVA
anova_result <- aov(gross_income ~ genre, data = movies_data_separated)

# Summarize the ANOVA result
summary(anova_result)

##                Df    Sum Sq   Mean Sq F value Pr(>F)
## genre          26 3.946e+18 1.518e+17   99.19 <2e-16 ***
## Residuals   84770 1.297e+20 1.530e+15
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 241 observations deleted due to missingness

# Step 3: Visualize the gross income distribution across genres
ggplot(movies_data_separated, aes(x = genre, y = gross_income)) +
  geom_boxplot() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(title = "Gross Income Distribution Across Genres",
       x = "Genre",
       y = "Gross Income")
```

```
## Warning: Removed 241 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Test conclusion

Results:

From the ANOVA test:

- $F(26,65283) = 86.3$’
- $p < 2 \times 10^{-16}$

Conclusion:

The **null hypothesis (H_0)** is **rejected** based on the extremely low p-value ($p < 2 \times 10^{-16}$), which indicates that the observed differences in gross income across genres are highly unlikely to have occurred by chance.

Implications:

- The **alternative hypothesis (H_1)** is supported, suggesting that there is a significant relationship between a movie’s genre and its gross income. Movies in certain genres tend to have higher gross income compared to others.

- The large F-value further reinforces that the variation in gross income across genres is substantial and noteworthy.

```
genre_means <- movies_data_separated %>%
  group_by(genre) %>%                   # Group by genre
  summarise(mean_gross_income = mean(gross_income, na.rm = TRUE)) %>%  #
Calculate mean gross income for each genre
  arrange(desc(mean_gross_income))      # Sort genres by mean gross income in
descending order

# View the results
genre_means

## # A tibble: 27 × 2
##    genre     mean_gross_income
##    <chr>                 <dbl>
##  1 Adventure         28616928.
##  2 Animation         26526279.
##  3 Sci-Fi            24800343.
##  4 Action            19167190.
##  5 Fantasy           18932935.
##  6 Family            17468625.
##  7 Comedy            16058916.
##  8 Thriller          10672185.
##  9 Mystery           10431741.
## 10 Drama              9179852.
## # ℹ 17 more rows
```

Here we might see what movies genres have most gross income. The most grossing genre is the sci-fy, the least - history.

2. Movies directed by a specific director are significantly higher than the the average rating of all movies.

Here we will try to find out whether director has impact on rating of movies

Null Hypothesis (H_0):

The mean ratings of movies directed by a specific director (e.g., Christopher Nolan) are equal to the population mean.

Alternative Hypothesis (H_1):

The mean ratings of movies directed by the specific director are significantly higher than the population mean.

For this hypothesis we will use *t-test*

Key Concepts of t-test

A t-test is a statistical test used to compare the means of two groups or a sample mean with a known population mean to determine if there is a significant difference between them. The key steps are:

1. For each director, calculate the average rating and average gross income of their movies to choose the top director.

2. Calculate the mean rating of the selected director’s movies.

3. Calculate the mean rating of all movies in the population.

4. Use the one-sample t-test formula to calculate the t-statistic: $t = \frac{\bar{x}-\mu}{s/\sqrt{n}}$

5. Evaluate the p-value to decide whether to reject the null hypothesis.

R Implementation

```
# Separate the directors by commas and create separate lines for each
movies_separated <- movies %>%
  separate_rows(directors_name, sep = ",") %>%
  mutate(directors_name = str_trim(directors_name))  # Remove unnecessary
spaces

# Now we can continue with the analysis for each director
top_directors <- movies_separated %>%
  group_by(directors_name) %>%
  summarize(
    avg_rating = mean(rating, na.rm = TRUE),
    avg_gross_income = mean(gross_income, na.rm = TRUE)
  ) %>%
  filter(
    avg_rating >= quantile(movies_separated$rating, 0.9, na.rm = TRUE) &
    avg_gross_income >= quantile(movies_separated$gross_income, 0.9, na.rm =
TRUE)
  ) %>%
  arrange(desc(avg_rating), desc(avg_gross_income))

print(top_directors)

## # A tibble: 93 × 3
##    directors_name      avg_rating avg_gross_income
##    <chr>                    <dbl>            <dbl>
##  1 Cory Barlog               9.4         62438154
##  2 Neil Druckmann            9.28        45318003.
##  3 Venu Udugula              9.2        100830111
##  4 Mathijs de Jonge          9.15        27909356
##  5 Robert Darryl Purdy       8.85        91989082
##  6 Ryan Smith                8.85        40303438
##  7 Luke Smith                8.8         43585753
##  8 Madhavan                  8.7        107325195
##  9 America Young             8.7         43869350
## 10 Morimasa Sato             8.7         43869350
## # ℹ 83 more rows
```

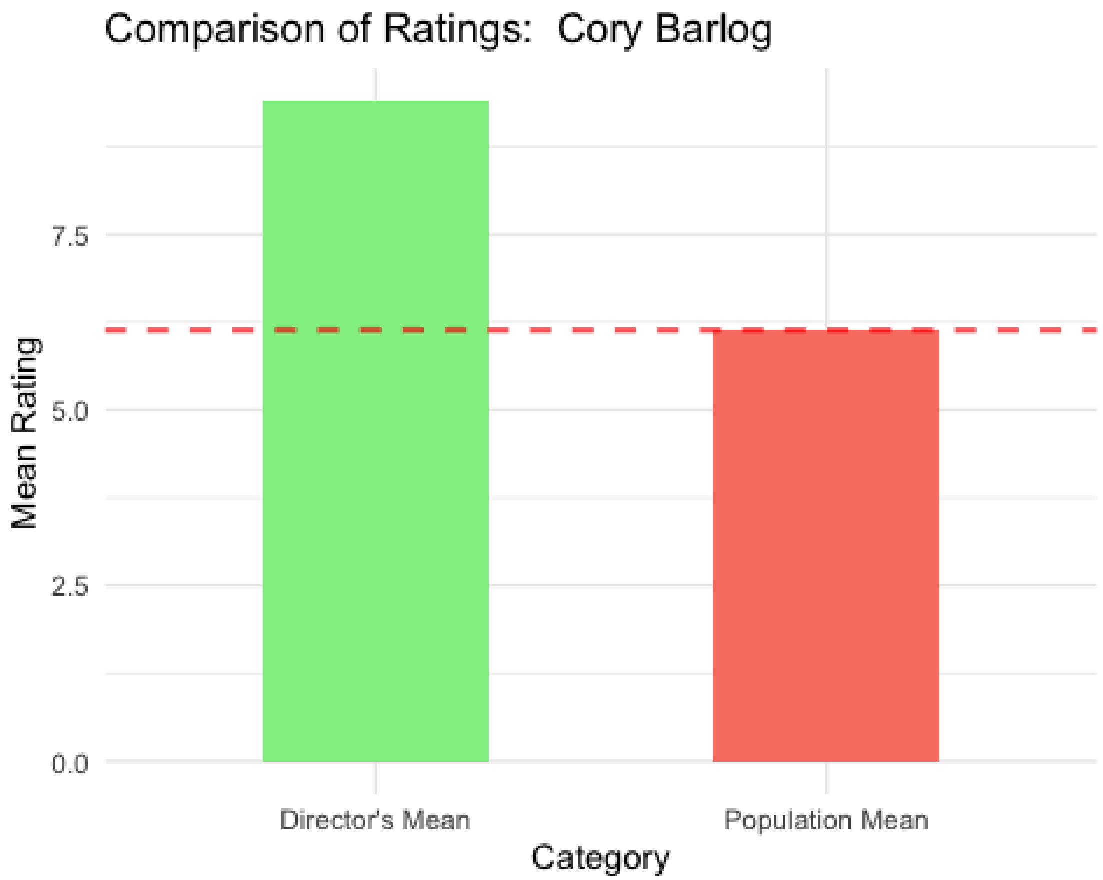
```
director <- head(top_directors$directors_name, 1)

director_movies <- movies %>%
  filter(directors_name == director)

mean_director_rating <- mean(director_movies$rating, na.rm = TRUE)
mean_population_rating <- mean(movies$rating, na.rm = TRUE)

t_test_result <- t.test(director_movies$rating, mu = mean_population_rating,
alternative = "greater")

director
```

```
## [1] "Cory Barlog"
```

```
t_test_result
```

```
## 
##  One Sample t-test
## 
## data:  director_movies$rating
## t = 16.318, df = 1, p-value = 0.01948
## alternative hypothesis: true mean is greater than 6.136386
## 95 percent confidence interval:
##  8.13725      Inf
## sample estimates:
## mean of x
##       9.4
```

```
ratings_data <- data.frame(
  Category = c("Population Mean", "Director's Mean"),
  Mean_Rating = c(mean_population_rating, mean_director_rating)
)

# Plot the bar chart
bar_plot <- ggplot(ratings_data, aes(x = Category, y = Mean_Rating, fill =
Category)) +
  geom_bar(stat = "identity", width = 0.5) +
  geom_hline(yintercept = mean_population_rating, linetype = "dashed", color
= "red", linewidth = 0.8, alpha = 0.7) +
  labs(
    title = paste("Comparison of Ratings: ", director),
    x = "Category",
    y = "Mean Rating"
  ) +
  theme_minimal() +
  scale_fill_manual(values = c("lightgreen", "salmon")) +
  theme(legend.position = "none")

print(bar_plot)
```



Test conclusion

The p-value (0.01948) is less than the typical significance level ($\alpha = 0.05$). This means we reject the null hypothesis (H_0) and accept the alternative hypothesis (H_1).

There is sufficient evidence to conclude that the mean rating of movies directed by the specific director is significantly higher than the population mean rating at a 95% confidence level.

3. How gross income changes from year to year

We are examining whether the gross income of movies is significantly influenced by the year of release.

Null Hypothesis (H_0):

The gross income of movies does not change significantly with the year of release.

Alternative Hypothesis (H_1):

The gross income of movies changes significantly with the year of release.

Here we use **ANOVA test**, because it is appropriate since it tests the differences in the means of a continuous variable (gross income) across multiple groups (genres or years).

Key Concepts of ANOVA (Analysis of Variance)

1. **Between-Groups Variation (SSB):**
   Measures how the mean gross income differs between years. A large SSB indicates a significant difference in gross income between years.

2. **Within-Groups Variation (SSW):**
   Measures the variability of gross income within each year. A small SSW suggests that movies within the same year have similar gross incomes.

3. **Total Variation (SST):**
   The total variation in gross income across all movies. The relationship is: $SST = SSB + SSW$

4. **Mean Square Between Groups (MSB):**
   $MSB = \frac{SSB}{df_B}, \quad df_B = k - 1$ where k is the number of years.

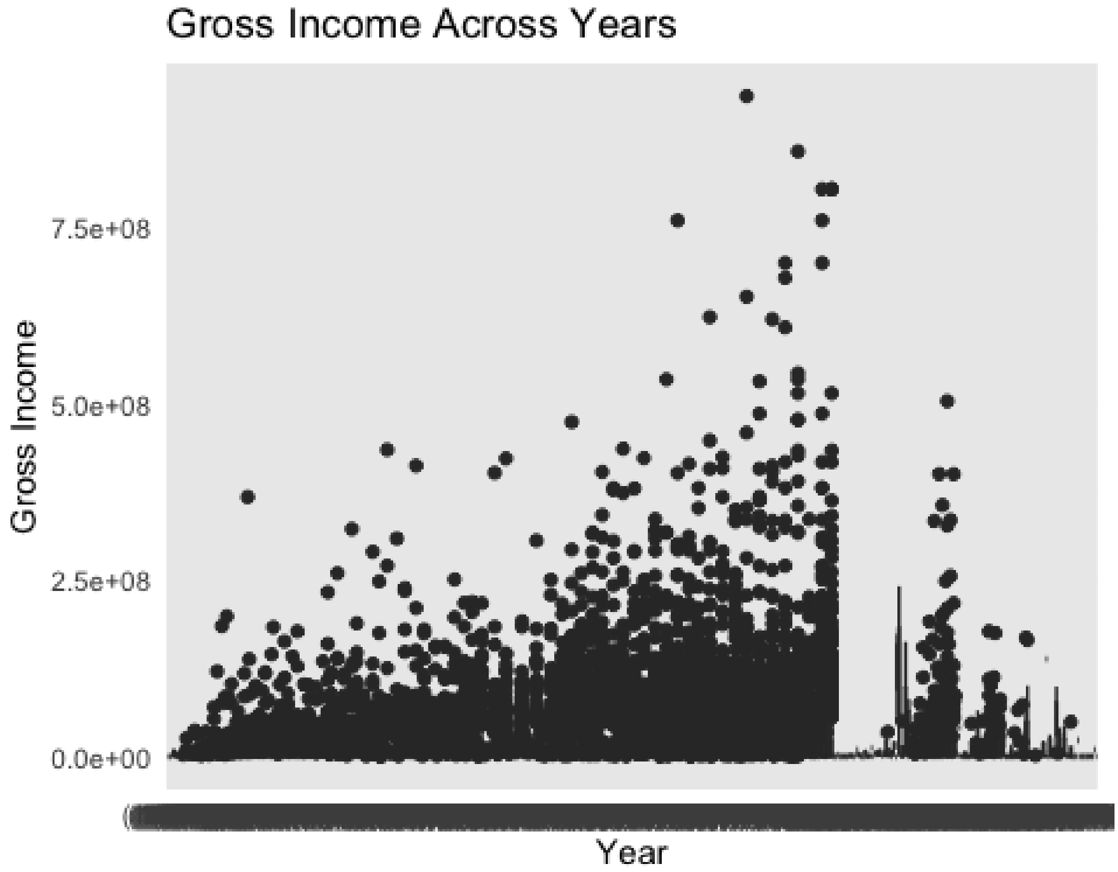
5. **Mean Square Within Groups (MSW):**
   $MSW = \frac{SSW}{df_W}, \quad df_W = n - k$ where n is the total number of movies.

6. **F-Statistic:**
   $F = \frac{MSB}{MSW}$ The F-statistic compares the variation between groups (years) to the variation within groups (movies in the same year).

7. Using the F-statistic and the degrees of freedom, determine the **p-value** from the F-distribution.

R Implementation

```
# Ensure 'year' is treated as a categorical variable
movies$year <- as.factor(movies$year)

# Perform ANOVA
anova_result <- aov(gross_income ~ year, data = movies)

# Summary of ANOVA
summary(anova_result)

##                Df    Sum Sq   Mean Sq F value Pr(>F)    
## year          575 1.388e+18 2.415e+15   1.736 <2e-16 ***
## Residuals   33034 4.595e+19 1.391e+15                   
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 106 observations deleted due to missingness

#visualization of gross income distribution across years
ggplot(movies, aes(x = year, y = gross_income)) +
  geom_boxplot() +
  labs(title = "Gross Income Across Years", x = "Year", y = "Gross Income") +
  theme_minimal()

## Warning: Removed 106 rows containing non-finite outside the scale range
## (`stat_boxplot()`).
```



Test conclusion

The p-value associated with the “year” factor is < 2e-16, which is extremely small (much smaller than the typical significance level of 0.05). This means that the gross income of movies does change significantly over the years, so we reject the null hypothesis.

Conclusions

In this study, we examined various factors that contribute to the success of films, including audience ratings, box office earnings, genre, and director influence. Our analysis revealed that most films tend to have ratings around 6 out of 10, with a slight skew toward lower ratings. This suggests that achieving high ratings is challenging and may be an important factor for success. The year 2019 stood out as the most successful year for the movie industry in terms of box office earnings, with a notable decline in subsequent years. This suggests that while the industry can experience periods of peak performance, earnings fluctuate over time.

We also found that certain genres, such as sci-fi, have a stronger correlation with higher earnings, while others, like historical films, tend to have lower gross incomes. This highlights the significant role that genre selection plays in a film's financial success. Furthermore, our analysis showed that movies directed by certain individuals tend to receive higher ratings compared to the average, indicating that experienced or popular directors have a notable impact on a film's success. The significant changes in box office earnings over the years suggest that market trends and audience preferences evolve, making it crucial for film studios to stay attuned to these shifts.

In conclusion, the success of films is influenced by a combination of factors, including genre, director, and timing of release. Understanding these elements can help film studios make more informed decisions when producing and marketing films, ultimately increasing the likelihood of achieving high ratings and strong financial performance.