An Analysis of Factors Influencing High IMDB Ratings

Group 8

1 Data Description

Source: IMDB film database

Description of variables:

- `film_id`: Unique identifier
- `year`: Year of release
- `length`: Duration (minutes)
- `budget`: Production budget (in $10 million)
- `votes`: Number of viewer votes
- `genre`: Genre of the film
- `rating`: IMDB score from 0–10

Total observations: 2,847 films

Objective of the analysis: To determine which factors of films are associated with an IMDB rating above 7 by using a Generalised Linear Model (GLM).

2 Data Preparing & Cleaning

2.1 Data Cleaning

```
# Load dataset
raw_data <- read.csv("dataset08.csv")

# Preview the structure of the dataset
glimpse(raw_data)
```

```
Rows: 2,847
Columns: 7
$ film_id <int> 5993, 37190, 43646, 28476, 23975, 50170, 56142, 2287, 17822, 5~
$ year    <int> 1943, 1961, 1987, 1976, 1982, 1936, 1932, 1967, 1983, 2003, 19~
$ length  <int> 65, 87, 79, NA, 88, NA, 75, 100, 82, 15, 86, 96, 150, 86, 102,~
$ budget  <dbl> 15.5, 12.3, 16.4, 12.2, 12.5, 7.0, 12.0, 12.2, 13.4, 13.9, 11.~
$ votes   <int> 42, 6, 161, 5, 97, 146, 14, 8, 141, 20, 121, 119, 5, 14, 48, 1~
$ genre   <chr> "Action", "Drama", "Action", "Documentary", "Action", "Drama",~
$ rating  <dbl> 7.6, 6.0, 7.5, 8.0, 3.5, 4.4, 4.5, 8.4, 3.5, 7.8, 8.2, 2.9, 4.~
```

```
# Remove rows that have missing values in 'length'variable
clean_data <- raw_data %>%
  filter(!is.na(length))

# Convert 'genre' to factor for categorical analysis
clean_data$genre <- as.factor(clean_data$genre)

# Define a function to create new binary response variable 'rating_above7'
rating_rank <- function(rating_column, threshold = 7){
  ifelse(rating_column > threshold, 1, 0)
}
#check the range of 'year' variable
range(clean_data$year) #we can see that range is between 1898 and 2005
```

```
[1] 1898 2005
```

```
# Mutate new variables : binary outcome 'rating_above_7' & 'decade_group'
clean_data <- clean_data %>%
  mutate(
```

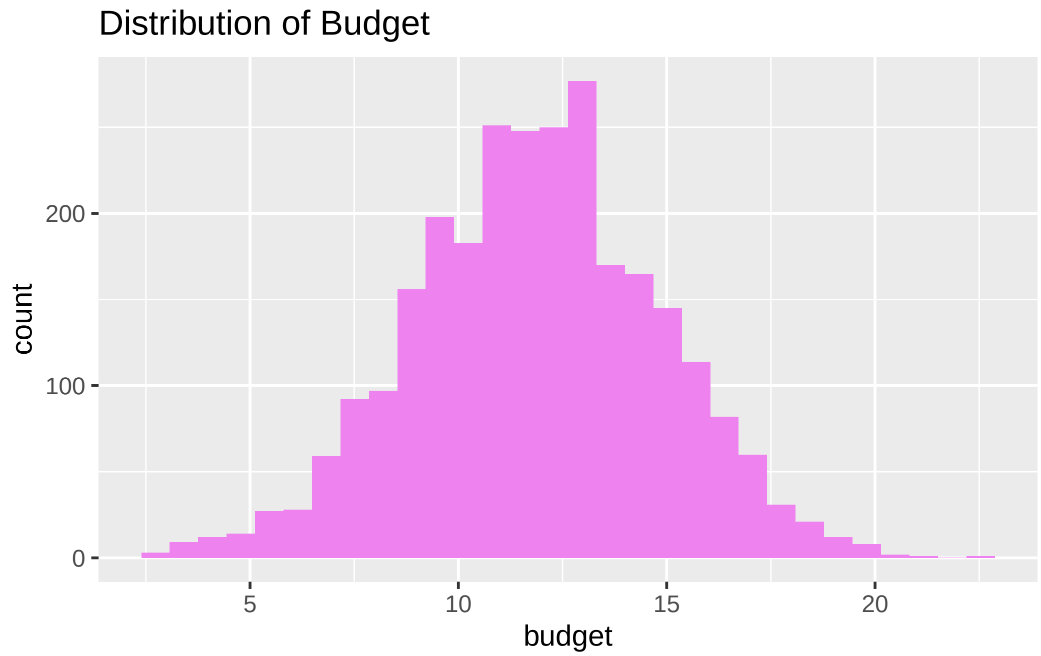
```
    rating_above_7 = rating_rank(rating),
    decade_group = cut(year,
                       breaks = c(1890, 1930, 1940, 1950, 1960, 1970, 1980, 1990, 2000, 2010
                       labels = c("1890s-1920s", "1930s", "1940s", "1950s", "1960s", "1970s"
                       right=FALSE)
  )
#check the missing values in 'budget' & 'votes' variables
sum(is.na(clean_data$budget)) # 0 missing values
```

```
[1] 0
```

```
sum(is.na(clean_data$votes)) # 0 missing values
```

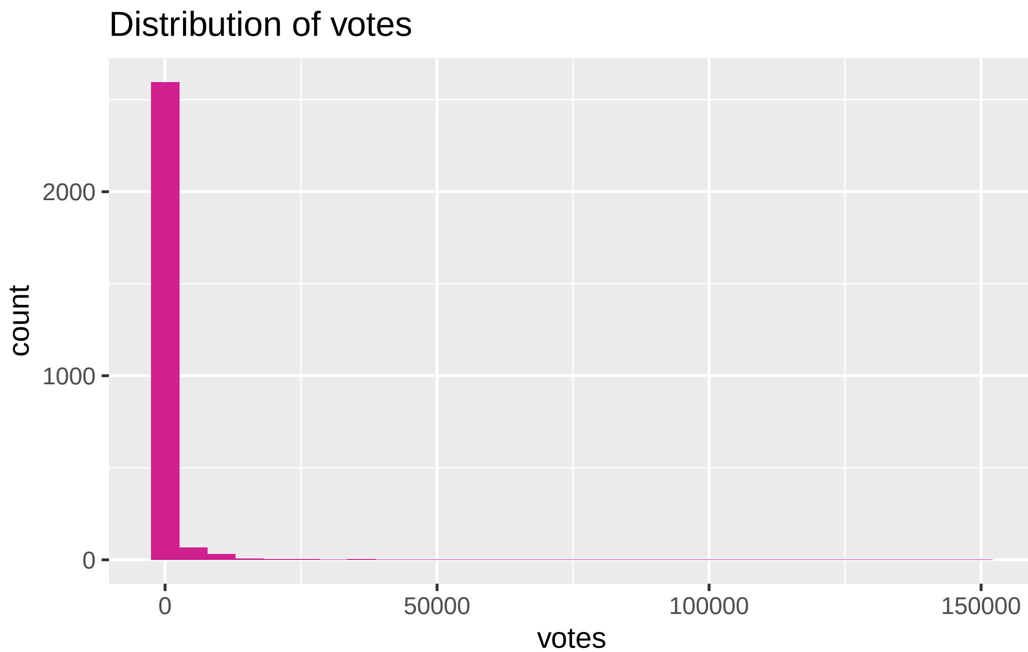
```
[1] 0
```

```
#Visualize the distribution of 'budget'
#If distribution is heavily skewed, log-transformation might be needed
ggplot(clean_data, aes(x = budget)) +
  geom_histogram(bins = 30, fill = "violet") +
  labs(title = "Distribution of Budget")
```



```
#Interpretation:
#The 'budget' variable appears approximately normally distributed.

#Visualize the distribution of 'votes'
ggplot(clean_data, aes(x = votes)) +
  geom_histogram(bins = 30, fill = "violetred") +
  labs(title = "Distribution of votes")
```




```
#Interpretation:
#The 'votes' variable is highly right-skewed.
#A log-transformation should be applied before using this variable in modelling.
```

2.2 Train-Test Splitting

```
set.seed(69)
# From this part, we split into 60/40
# A larger test set (40%) allows for more reliable model evaluation
train_data_index <- sample(seq_len(nrow(clean_data)), size = 0.6 * nrow(clean_data))
```

```
train_data <- clean_data[train_data_index, ]
test_data <- clean_data[-train_data_index, ]
```

3 Exploratory Data Analysis (EDA)

4 Statistical Modelling

5 Model Diagnostics

6 Conclusions