Lab2

Shreyash Singh

2024-01-29

Importing the Data and the library

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v readr     2.1.5
## v ggplot2   3.4.4     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
```

```
## -- 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 error
```

```
library(ggplot2)
library(ggplot2movies)
data(movies)
```

Soln 1. Range of years of production of the movies of this data set

```
range <- range(movies$year)
cat(" Range of years of production\n","The oldest movie was produced in:",
    range[1],"\n","The most recent movie was produced in:", range[2],"\n",
    "Time gap between the two:", range[2]-range[1], "years.")
```

```
##  Range of years of production
##  The oldest movie was produced in: 1893
##  The most recent movie was produced in: 2005
##  Time gap between the two: 112 years.
```

Soln 2. Budget Information for the movies of data set

```
hasBudget <- sum(!is.na(movies$budget)) / nrow(movies)
top5 <- head(movies[order(movies$budget, decreasing = TRUE),c("title", "budget")], 5)
for (i in 1:nrow(top5)) {
  if (i==1) {
    cat(" Proportion of movies with budget included:", hasBudget, "\n",
        "Proportion of movies without budget information:", 1 - hasBudget,"\n",
        "Top 5 most expensive movies:\n",
        sprintf("%-40s %-15s\n","Title", "Budget"))
  }
  cat(sprintf("%-40s %-15s\n", top5[i, "title"], top5[i, "budget"]))
}
```

```
##  Proportion of movies with budget included: 0.08870858
##  Proportion of movies without budget information: 0.9112914
##  Top 5 most expensive movies:
##  Title                                     Budget
## Spider-Man 2                              200000000
## Titanic                                   200000000
## Troy                                      185000000
## Terminator 3: Rise of the Machines        175000000
## Waterworld                                175000000
```

Soln 3. Top 5 Longest Movies

```
top5 <- head(movies[order(movies$length, decreasing = TRUE), c("title", "length")], 5)
for (i in 1:nrow(top5)) {
  if(i ==1){
    cat("Top 5 longest movies:\n",sprintf("%-50s %-10s\n", "Title", "Length(minutes)"))
  }
  cat(sprintf("%-50s %-10s\n", top5[i, "title"], top5[i, "length"]))
}
```

```
## Top 5 longest movies:
##  Title                                              Length(minutes)
## Cure for Insomnia, The                             5220
## Longest Most Meaningless Movie in the World, The   2880
## Four Stars                                         1100
## Resan                                              873
## Out 1                                              773
```

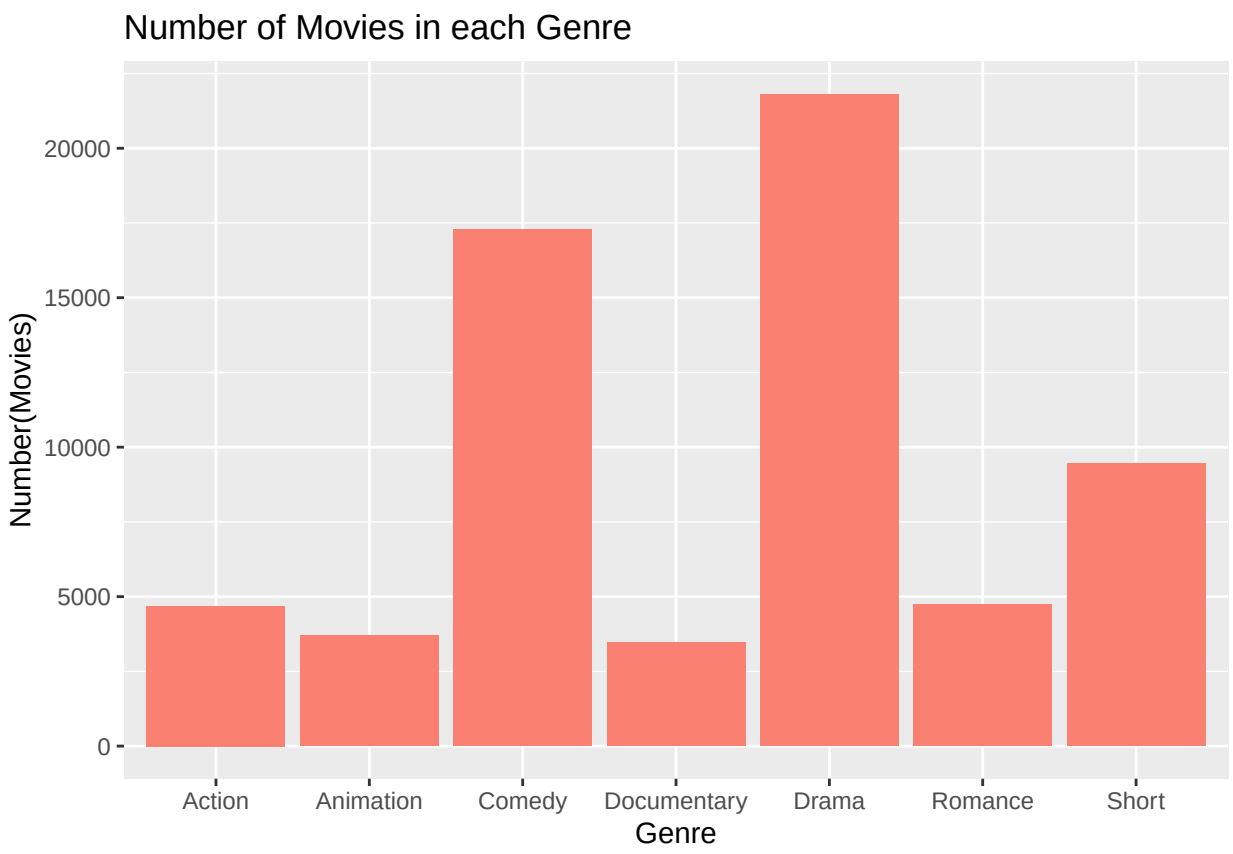
Soln 4. The Shortest and Longest Movies

```
shortMovies <- movies[movies$Short == 1, ]
shortest <- head(shortMovies[order(shortMovies$length, decreasing = FALSE), ], 1)
longest <- head(shortMovies[order(shortMovies$length, decreasing = TRUE), ], 1)
cat(" Shortest short movie:", shortest$title, "(", shortest$length, "minutes )\n",
  "Longest short movie:", longest$title, "(", longest$length, "minutes )\n")
```

```
##  Shortest short movie: 17 Seconds to Sophie ( 1 minutes )
##  Longest short movie: 10 jaar leuven kort ( 240 minutes )
```

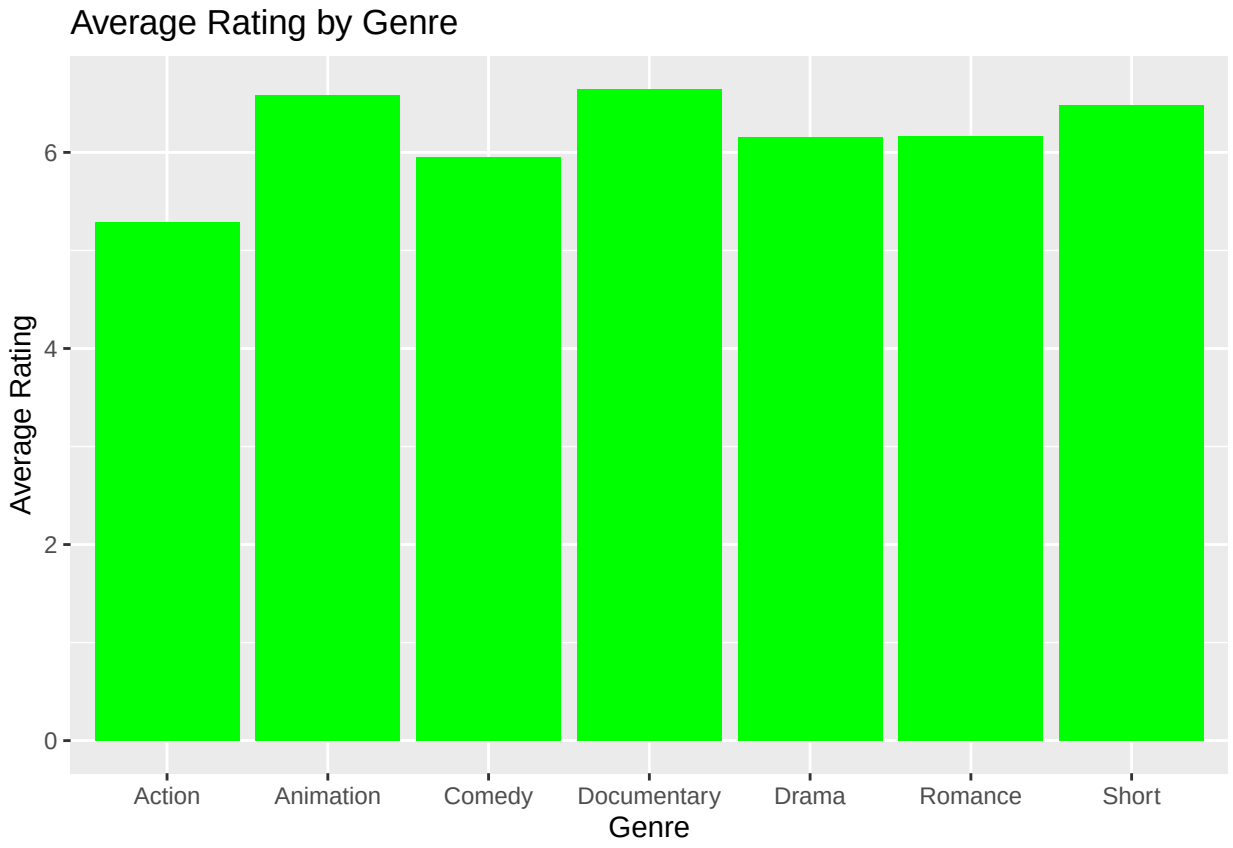
Soln 5. The Shortest and Longest Movies

```
genres <- c("Action", "Animation", "Comedy", "Drama", "Documentary", "Romance", "Short")
nums <- apply(movies[genres], 2, sum)
df = data.frame(genres = names(nums), count = nums)
ggplot(df, aes(y = count, x = genres)) +
geom_bar(stat = "identity", fill="salmon") +
labs(title = "Number of Movies in each Genre", x = "Genre", y = "Number(Movies)")
```



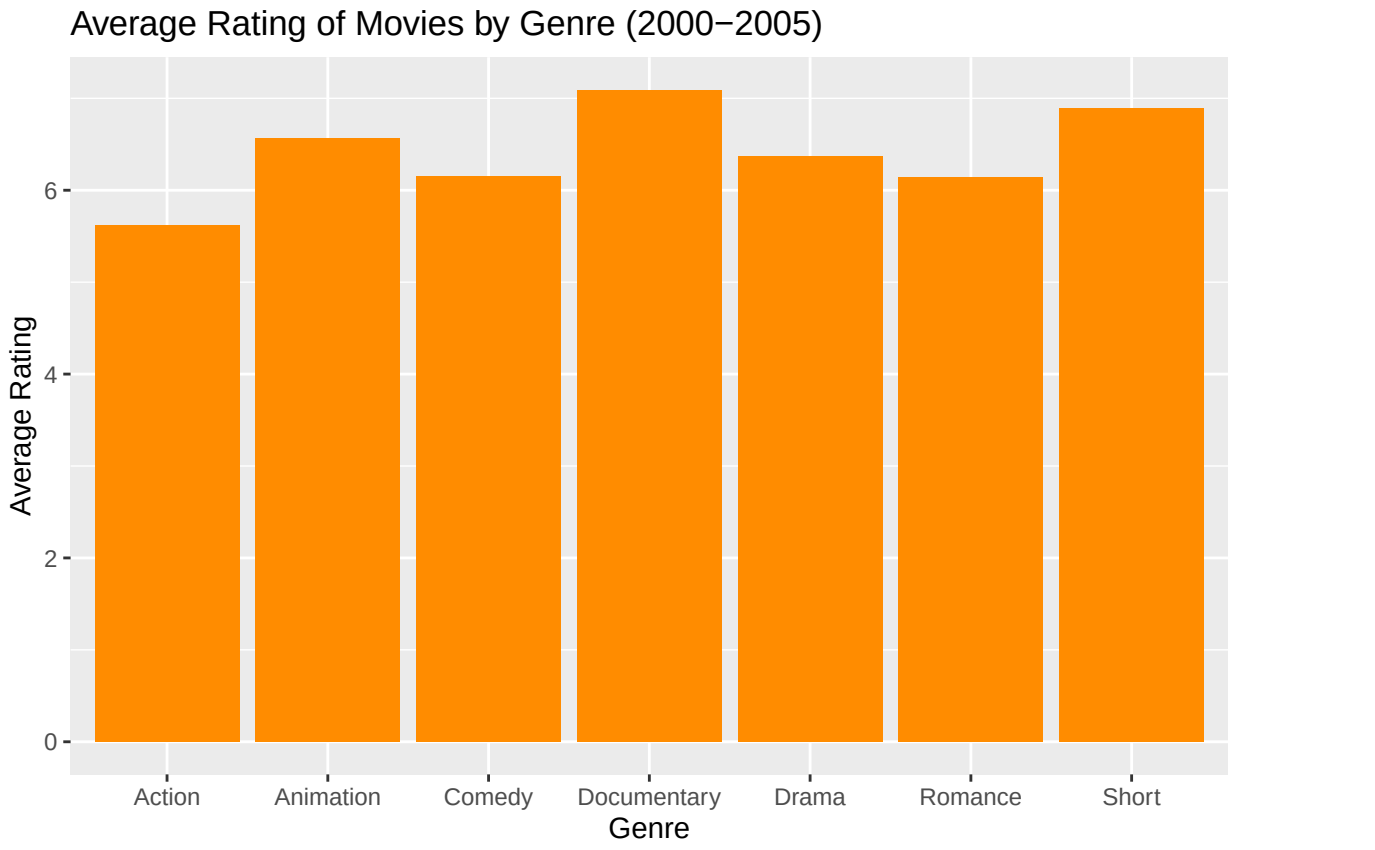
Soln 6. The average rating of all movies within each genre

```r
avgRating <- data.frame(genre = character(), rating = numeric())

for (genre in genres) {
  # Calculate average rating for each genre
  rating <- mean(movies$rating[movies[, genre] == 1], na.rm = TRUE)
  avgRating  <- rbind(avgRating, data.frame(genre = genre, rating = rating))
}
ggplot(avgRating, aes(x = genres, y = rating)) +
geom_bar(stat = "identity", fill = "green") +
labs(title = "Average Rating by Genre", x = "Genre", y = "Average Rating")
```



Soln 7. The average rating of all movies within each genre that were produced in the years 2000-2005?

```
# Create average rating for each genre as subset
avgRating_2000_2005 <- data.frame(genre = character(), rating = numeric())

for (genre in genres) {
  rating <- mean(movies$rating[movies[, genre] == 1 & movies$year >= 2000 & movies$year <= 2005], na.rm
  avgRating_2000_2005 <- rbind(avgRating_2000_2005, data.frame(genre = genre, rating = rating))
}

ggplot(avgRating_2000_2005, aes(x = genre, y = rating)) +
  geom_bar(stat = "identity", fill = "darkorange") +
  labs(title = "Average Rating of Movies by Genre (2000-2005)", x = "Genre", y = "Average Rating")
```

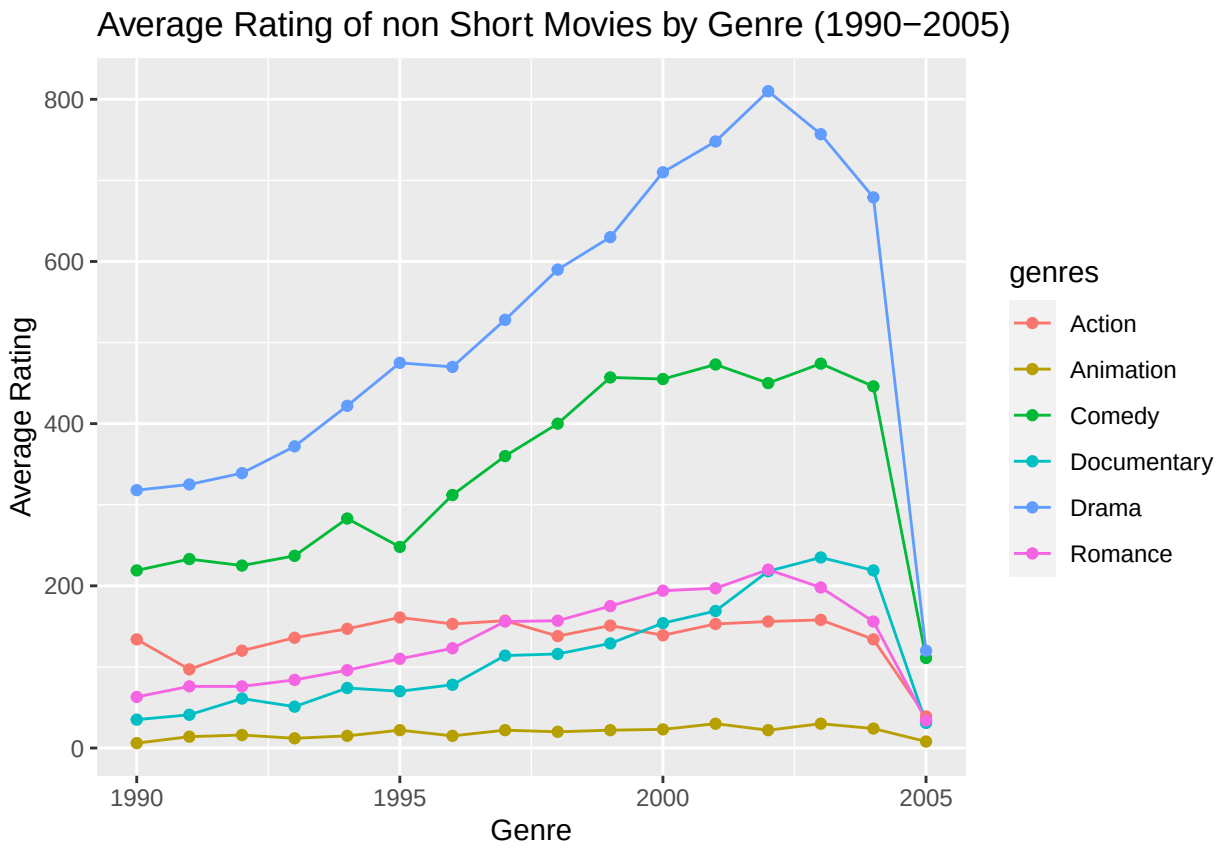


Soln 8. The average rating of all movies within each genre that were produced in the years 2000-2005?

```
movies_1990 <- movies[movies$year <= max(movies$year) & movies$year >= 1990 & movies$Short == 0,]
movies_1990 <- movies_1990 %>% pivot_longer(cols = c(Action, Animation, Comedy,
                                                     Drama, Documentary, Romance),
                                            names_to = "genres",
                                            values_to = "present")
numMovies <- movies_1990 %>%
  filter(present == 1) %>%
  group_by(genres, year) %>%
  summarise(nums = n())
```

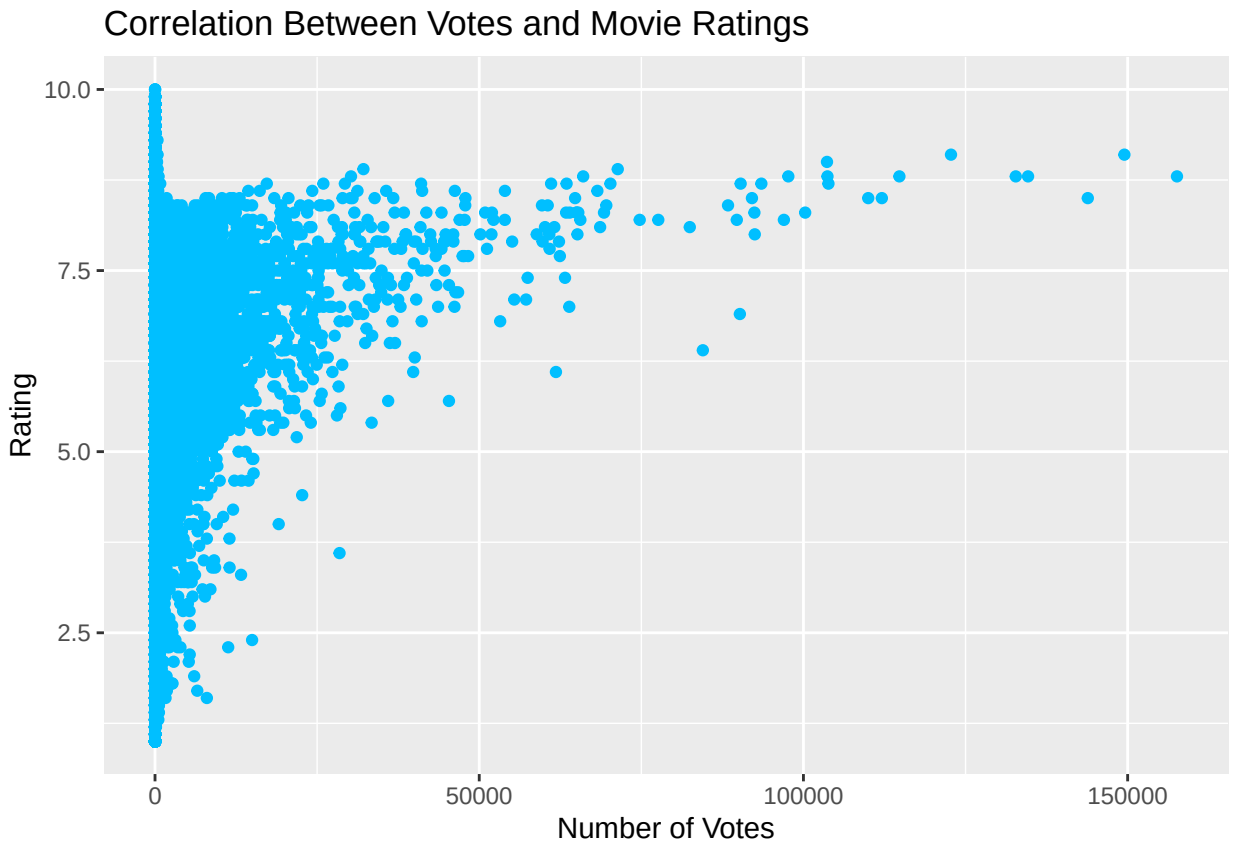
```
## `summarise()` has grouped output by 'genres'. You can override using the
## `.groups` argument.
```

```
ggplot(numMovies, aes(x = year, y = nums, group = genres, color = genres)) +
  geom_line(linewidth = 0.5) +
  labs(title = "Average Rating of non Short Movies by Genre (1990-2005)",
       x = "Genre", y = "Average Rating") +
  geom_point()
```



Soln 9. 3 questions of my choice related to the dataset

```
#Question 1: Is there any correlation between the user votes and movie rating?

#Soln 9.1: Using a scatterplot
ggplot(movies, aes(x = votes, y = rating)) +
  geom_point(color = "deepskyblue") +
  labs(title = "Correlation Between Votes and Movie Ratings",
       x = "Number of Votes", y = "Rating")
```




```
#Answer: Looking at the plot we can see the votes tend to increase as the rating increases
```

```r
#Question 2: How many movies have a rating of 9.5 in different genre?

#Soln 9.2:

high_rating_movies <- subset(movies, rating >= 9.5)

genre_distribution <- colSums(high_rating_movies[,
          c("Action", "Animation", "Comedy", "Drama", "Documentary", "Romance", "Short")])

cat("Genre distribution of movies with rating of 9.5 or above:\n")
```

```
## Genre distribution of movies with rating of 9.5 or above:
```

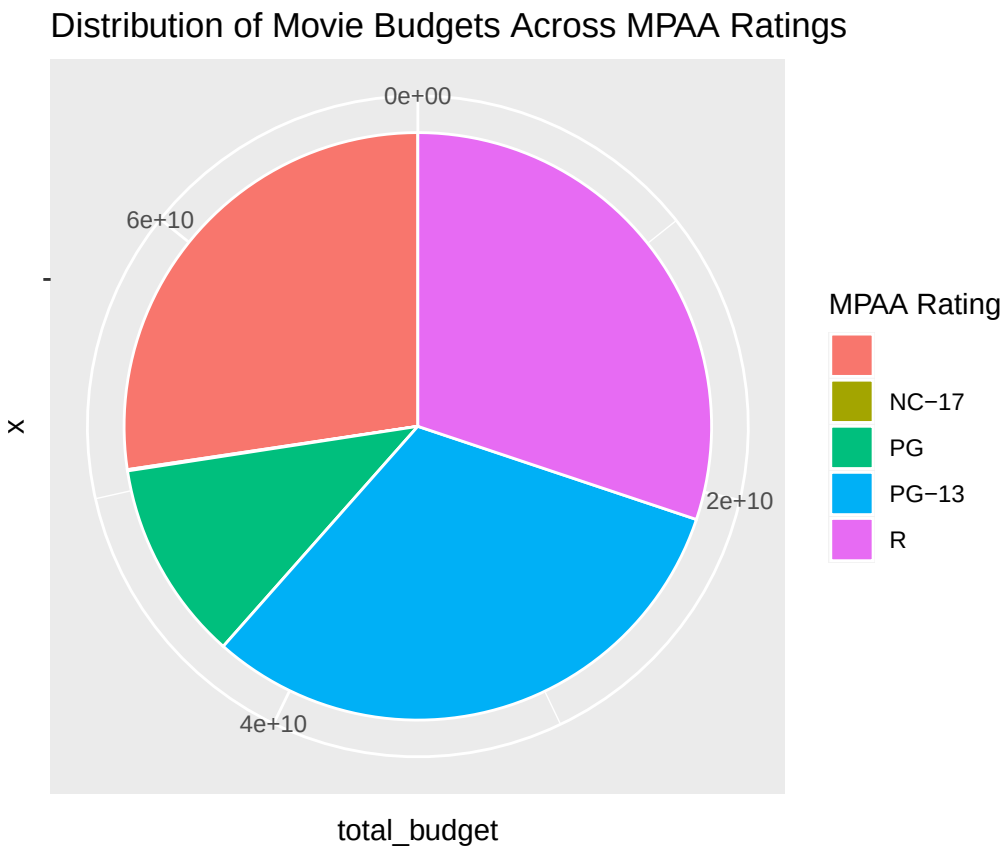
```r
print(genre_distribution)
```

```
##      Action   Animation      Comedy       Drama Documentary     Romance
##          12          14          59          91          49          16
##       Short
##         136
```

```r
#Question 3: What is the distribution of movie budgets across different MPAA rating?

#Soln 9.3: Using a tableview

withBudget <- movies[!is.na(movies$budget), ]
tot_bud <- withBudget %>%
  group_by(mpaa) %>%
  summarise(total_budget = sum(budget))

tot_bud$percent <- sprintf("%.2f%%", (tot_bud$total_budget / sum(tot_bud$total_budget)) * 100)
print(tot_bud)
```

```
## # A tibble: 5 x 3
##   mpaa    total_budget percent
##   <chr>          <dbl> <chr>
## 1 ""       19135024991 27.36%
## 2 "NC-17"     48637000 0.07%
## 3 "PG"      7728300000 11.05%
## 4 "PG-13"  21955784000 31.39%
## 5 "R"      21078510606 30.14%
```

```r
#Visual Representation:
ggplot(tot_bud, aes(x = "", y = total_budget, fill = mpaa, label = percent)) +
  geom_bar(stat = "identity", width = 1, color = "white") +
  coord_polar("y") +
  labs(title = "Distribution of Movie Budgets Across MPAA Ratings",
       fill = "MPAA Rating")
```



```
# the "" here in "salmon" color is the set of unrated movies
```