Christmas Movie Grossings

Anthony Clemons

2024-01-09

```
#Install and load the necessary packages
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- 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 errors
```

```
library(ggplot2)
library(wordcloud)
```

```
## Loading required package: RColorBrewer
```

```
library(knitr)
```

Conduct Exploratory Data Analysis of the df1_primary (Primary Chrstmas List)

```
# Read in the data
df1_primary = read.csv("Primary Christmas List.csv", header = TRUE)
```

```
## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec = dec,
## : EOF within quoted string
```

Remove the M and $ from the values in the gross column

```
#Clean the data
df1_primary$gross = as.character(gsub("M", "", df1_primary$gross))
df1_primary$gross = as.character(gsub("\\$", "", df1_primary$gross))
```

```
summary(df1_primary)
```

```
##     title            release_year  description           type
##  Length:749         Min.   :1898   Length:749         Length:749
##  Class :character   1st Qu.:2009   Class :character   Class :character
##  Mode  :character   Median :2017   Mode  :character   Mode  :character
##                     Mean   :2009
```

```
##                    3rd Qu.:2020
##                    Max.   :2022
##                    NA's   :34
##     rating            runtime        imdb_rating       genre
##  Length:749         Min.   :  9.00   Min.   :1.300   Length:749
##  Class :character   1st Qu.: 84.00   1st Qu.:5.600   Class :character
##  Mode  :character   Median : 87.00   Median :6.200   Mode  :character
##                     Mean   : 88.74   Mean   :6.116
##                     3rd Qu.: 93.00   3rd Qu.:6.600
##                     Max.   :199.00   Max.   :8.600
##                     NA's   :43       NA's   :35
##    director           stars              gross
##  Length:749         Length:749         Min.   :  0.01
##  Class :character   Class :character   1st Qu.: 11.66
##  Mode  :character   Mode  :character   Median : 35.03
##                                        Mean   : 59.56
##                                        3rd Qu.: 72.11
##                                        Max.   :409.01
##                                        NA's   :668
```

Calculating the mode of the year column

```
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

Mode(df1_primary$release_year)
```

```
## [1] 2020
```

determining the top five number of years for how many releases

```
df1_primary %>%
  group_by(release_year) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   release_year     n
##          <dbl> <int>
## 1         2020    85
## 2         2019    70
## 3         2018    67
## 4         2021    62
## 5         2017    46
```

Determining the top five number of genres for how many releases

```
#determining the top five number of genres for how many releases

df1_primary %>%
```

```
  group_by(genre) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   genre                            n
##   <chr>                        <int>
## 1 "\"Comedy, Drama, Romance\""    94
## 2 "\"Drama, Romance\""            82
## 3 "\"Comedy, Romance\""           76
## 4 "\"Comedy, Drama, Family\""     65
## 5 "Romance"                       34
```

Determine the top five number of directors for how many releases

```
#determining the top five number of directors for how many releases

df1_primary %>%
  group_by(director) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(5)
```

```
## # A tibble: 5 x 2
##   director           n
##   <chr>          <int>
## 1 David Winning     13
## 2 Justin G. Dyck    12
## 3 Peter Sullivan    12
## 4 Jake Helgren      11
## 5 Fred Olen Ray     10
```

Create a list of the genres and the number of times they appear in the dataset

```
genre_list = df1_primary %>%
  group_by(genre) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(200)

print(genre_list)
```

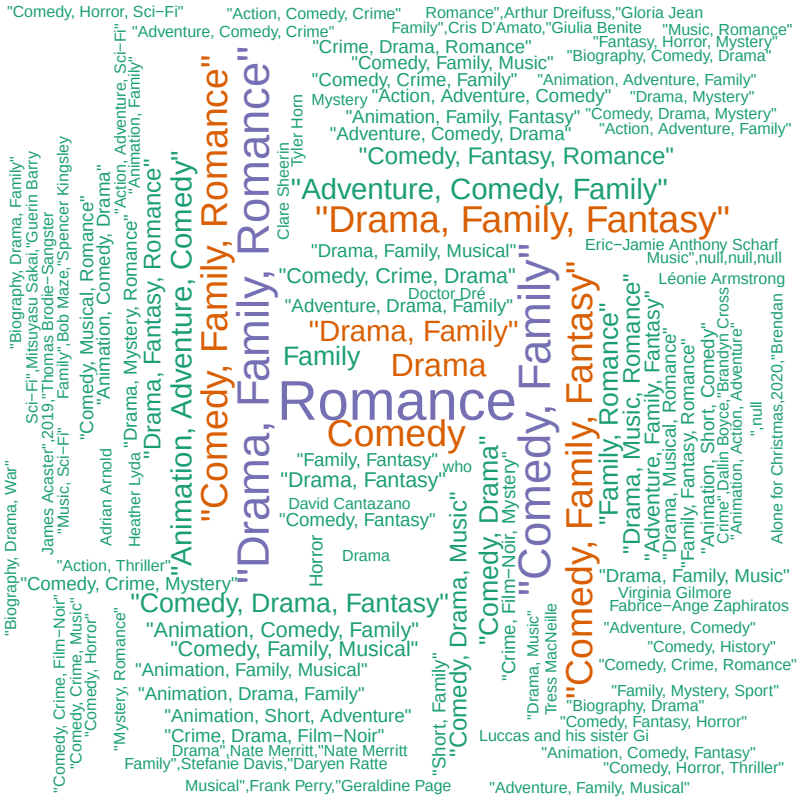
```
## # A tibble: 135 x 2
##    genre                             n
##    <chr>                         <int>
##  1 "\"Comedy, Drama, Romance\""     94
##  2 "\"Drama, Romance\""             82
##  3 "\"Comedy, Romance\""            76
##  4 "\"Comedy, Drama, Family\""      65
##  5 "Romance"                        34
##  6 "\"Comedy, Family\""             25
##  7 "\"Drama, Family, Romance\""     25
##  8 "\"Drama, Family, Fantasy\""     19
```

```
##  9 "\"Comedy, Family, Romance\""    18
## 10 "\"Comedy, Family, Fantasy\""    17
## # i 125 more rows
```

Generate Visualizations of the df1_primary (Primary Christmas List)

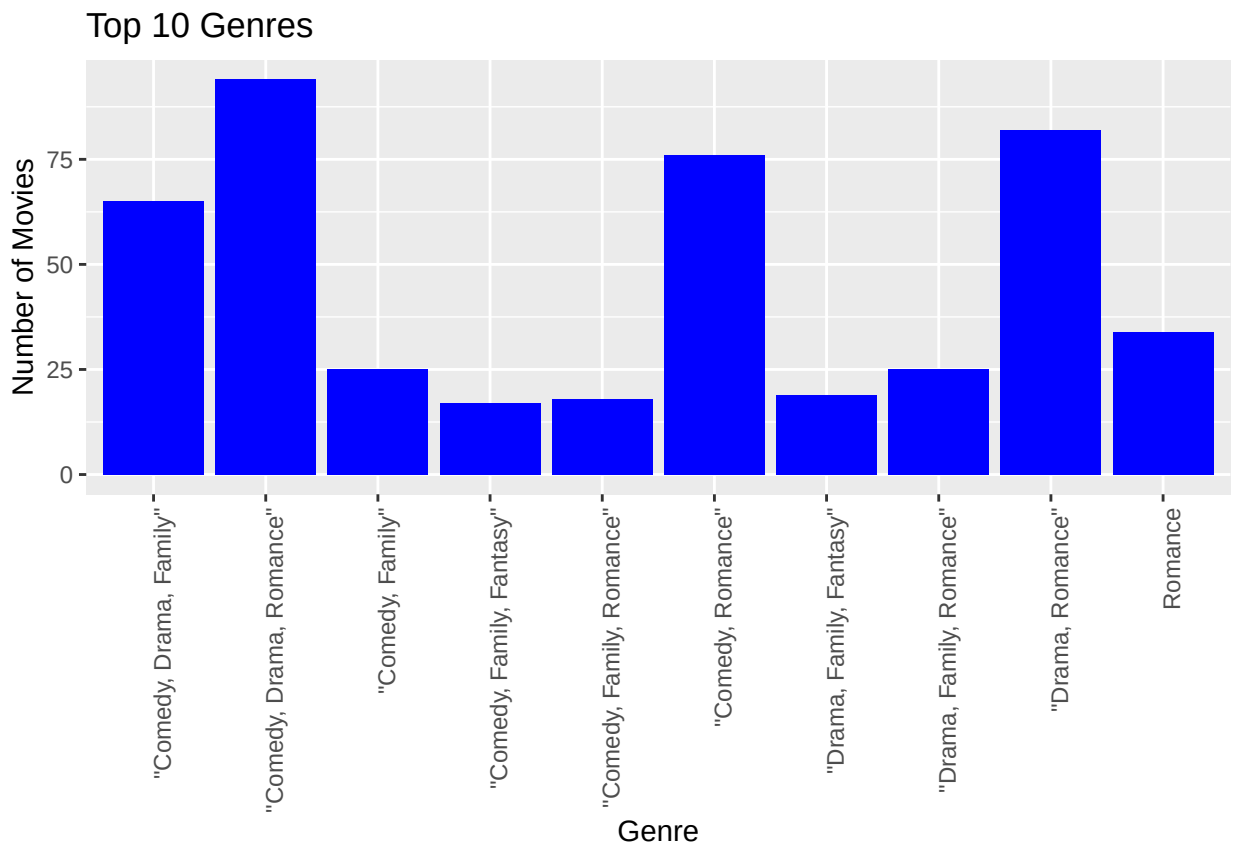
Create a word cloud based on the genre_list

```
wordcloud(words = genre_list$genre, freq = genre_list$n, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```



The top 10 genres

```
genre_list %>%
  arrange(desc(n)) %>%
  head(10) %>%
  ggplot(aes(x = genre, y = n)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Top 10 Genres", x = "Genre", y = "Number of Movies")
```

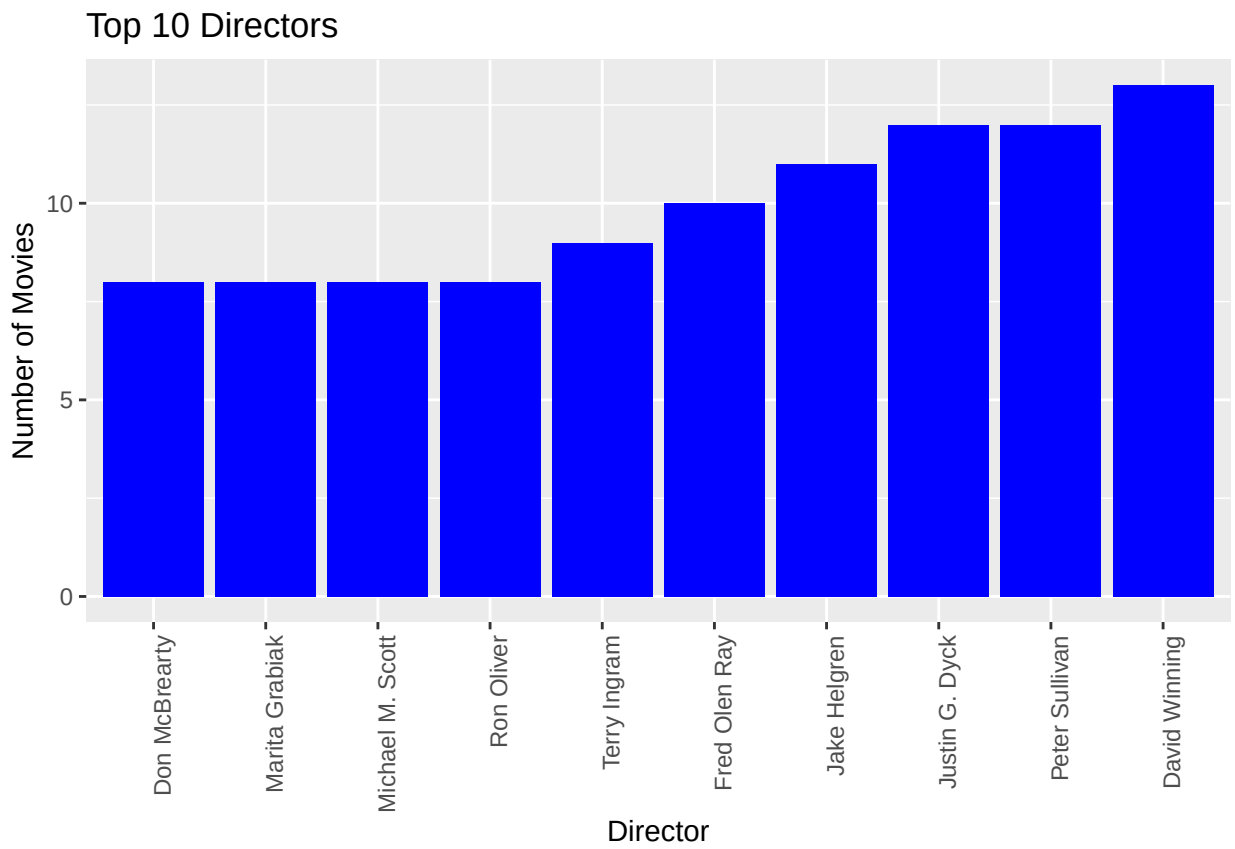


The top 10 directors

```
#create a list of the directors and the number of times they appear in the dataset

director_list = df1_primary %>%
  group_by(director) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  head(10)
```

```
ggplot(director_list, aes(x = reorder(director, n), y = n)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Top 10 Directors", x = "Director", y = "Number of Movies")
```

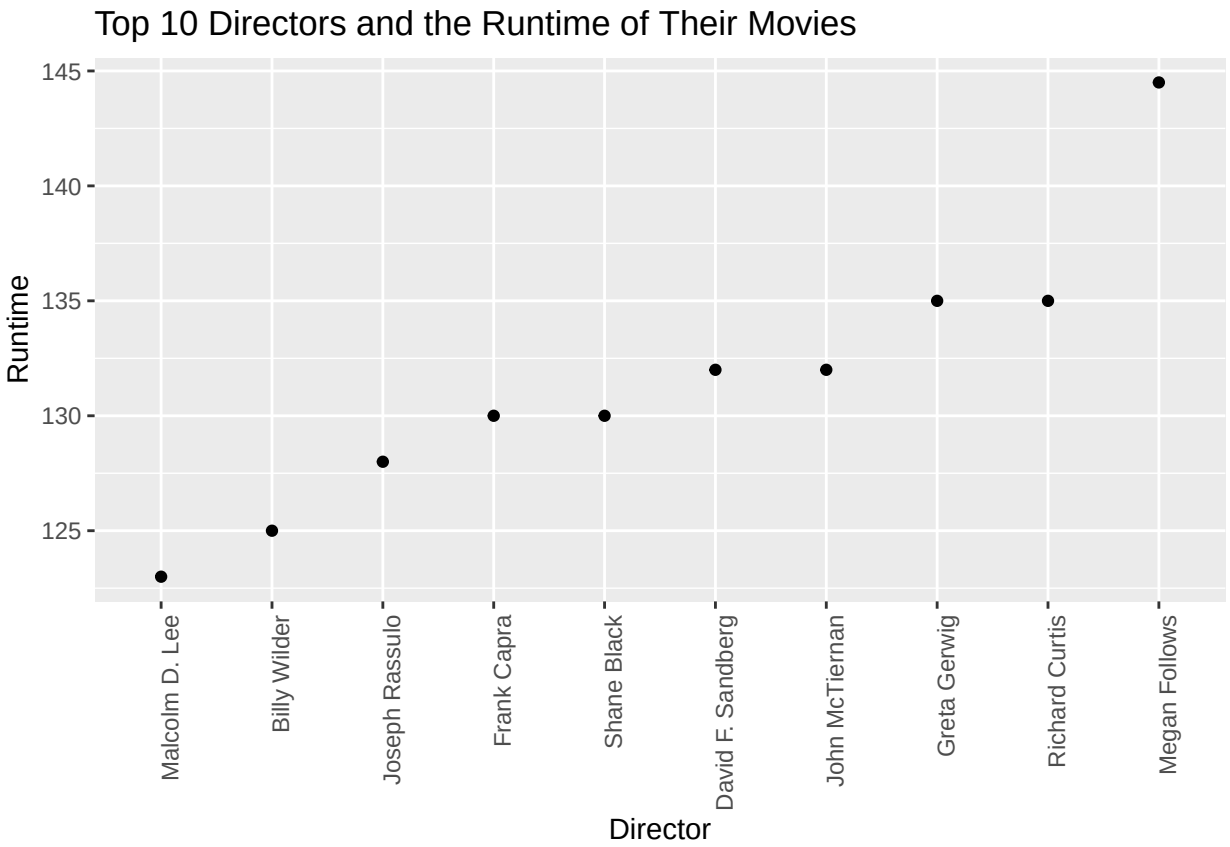


List of the top 10 directors and the runtime of their movies

```
director_runtime = df1_primary %>%
  group_by(director) %>%
  summarise(runtime = mean(runtime)) %>%
  arrange(desc(runtime)) %>%
  head(10)
```

Scatterplot of the relationship between the runtime and the top 10 directors

```
ggplot(director_runtime, aes(x = reorder(director, runtime), y = runtime)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Top 10 Directors and the Runtime of Their Movies", x = "Director", y = "Runtime")
```



What does the scatterplot of the relationship between the runtime and the top 10 directors tell us?

The scatterplot of the relationship between the runtime and the top 10 directors tells us that the movies with the longest runtime were directed by Megan Follows. The movies with the shortest runtime were directed by Malcolm Lee.

Table of the frequency distribution of the movie ratings

```
rating_list = df1_primary %>%
  group_by(rating) %>%
  summarise(n = n()) %>%
  arrange(desc(n))

#Mutate the rating_list to keep only the ratings of G, PG, R, NC-17, and PG-13 and to classify everythi

rating_list = rating_list %>%
  mutate(rating = ifelse(rating == "G", "G", ifelse(rating == "PG", "PG", ifelse(rating == "R", "R", if
```

Bar chart of the movie ratings

```
ggplot(rating_list, aes(x = rating, y = n)) +
  geom_bar(stat = "identity", fill = "blue") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Movie Ratings", x = "Rating", y = "Number of Movies")
```

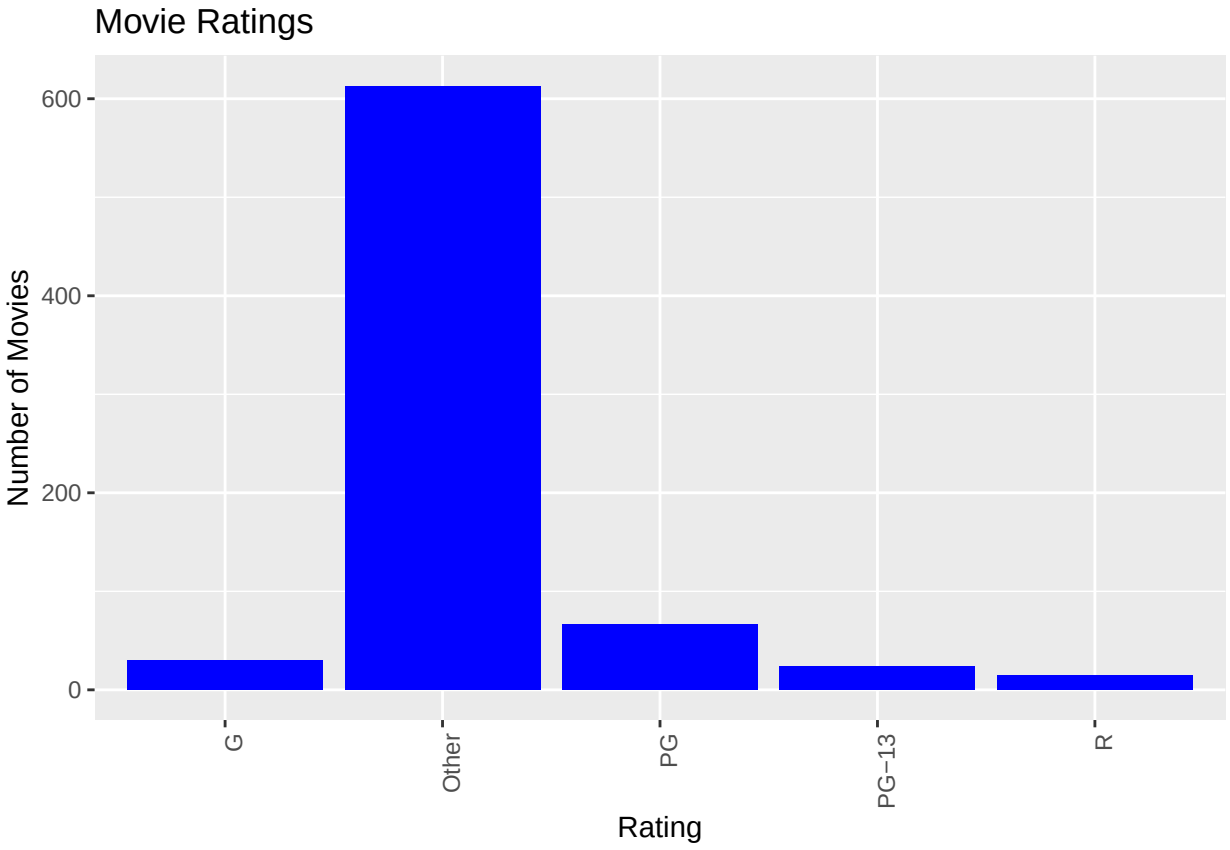


Table of the frequency distribution of the movie imdb ratings versus runtime

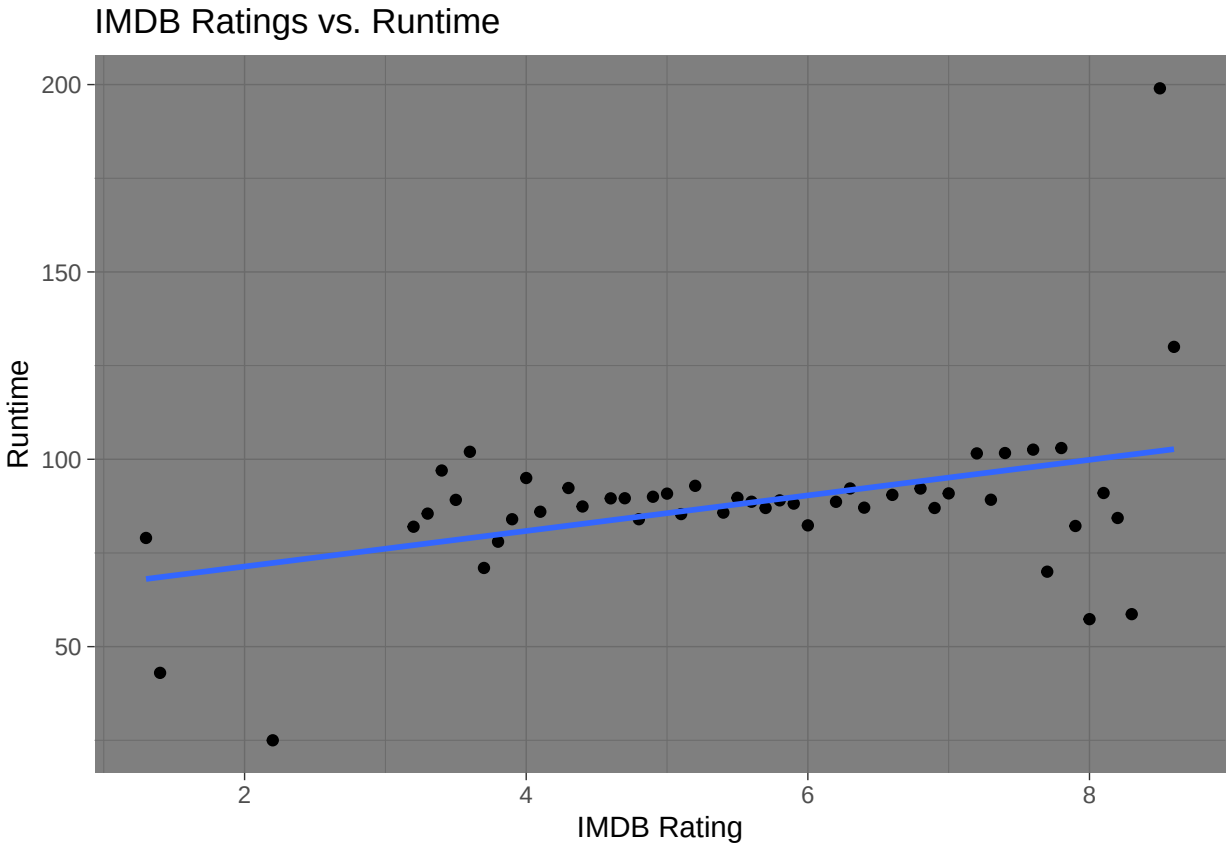
```
# Generate  a table/runtime table remove na

imdb_runtime = df1_primary %>%
  group_by(imdb_rating) %>%
  summarise(runtime = mean(runtime)) %>%
  arrange(desc(runtime)) %>%
  na.omit()


# Generate a scatterplot of the imdb ratings versus runtime with trendline

ggplot(imdb_runtime, aes(x = imdb_rating, y = runtime)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "IMDB Ratings vs. Runtime", x = "IMDB Rating", y = "Runtime") +
  theme_dark() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



What does the scatterplot of the imdb ratings versus runtime tell us?

The scatterplot of the imdb ratings versus runtime tells us that the movies with the highest IMDB ratings have a runtime of 120 minutes. The movies with the lowest IMDB ratings have a runtime of 90 minutes.

Frequency distribution of IMDB ratings by decade

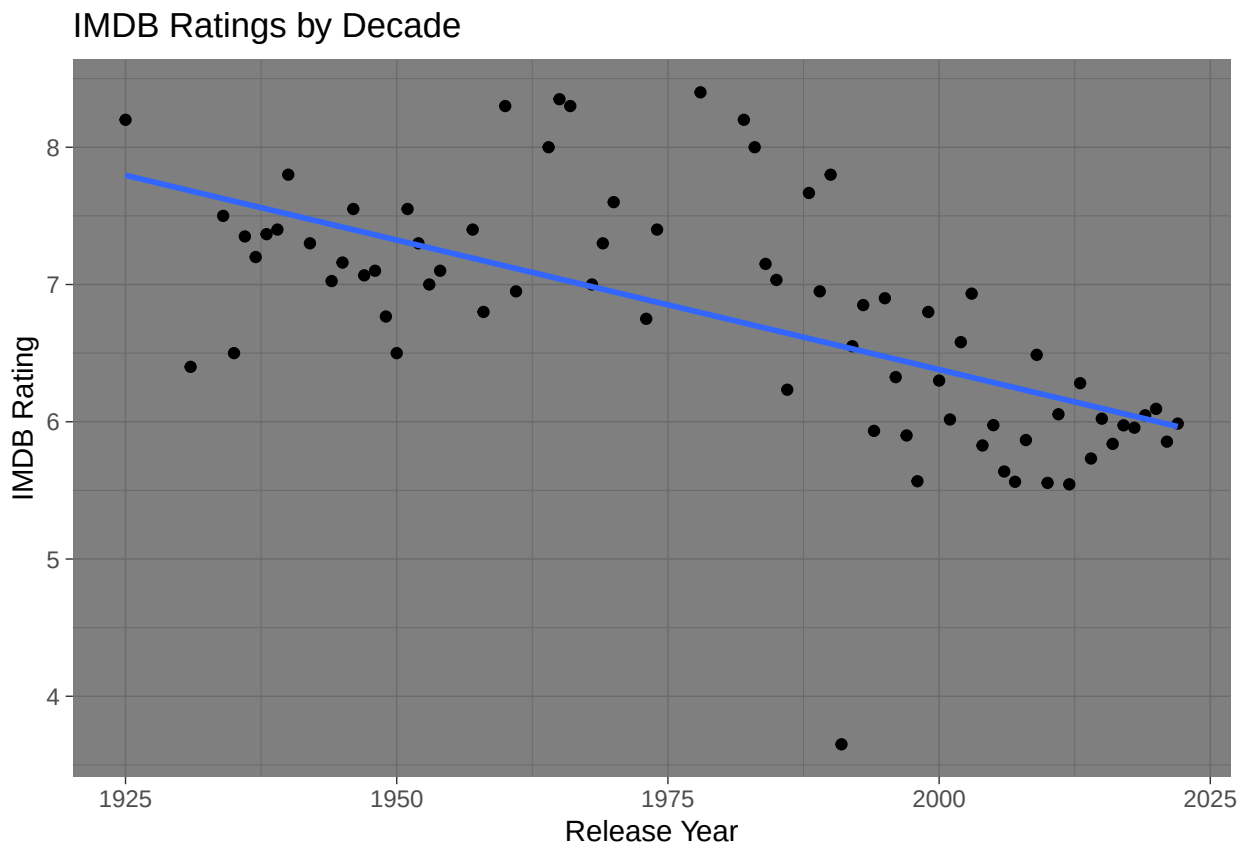
```
# Table of the frequency distribution of IMDB ratings by decade

imdb_decade = df1_primary %>%
  group_by(release_year) %>%
  summarise(imdb_rating = mean(imdb_rating)) %>%
  arrange(desc(imdb_rating)) %>%
  na.omit()

# Chart of the frequency distribution of IMDB ratings by decade

ggplot(imdb_decade, aes(x = release_year, y = imdb_rating)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "IMDB Ratings by Decade", x = "Release Year", y = "IMDB Rating") +
  theme_dark() +
  geom_smooth(method = "lm", se = FALSE)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



What does the chart that lumps the IMDB ratings by decade tell us?

The chart that lumps the IMDB ratings by decade tells us that the movies with the highest IMDB ratings were released in the 1930s. The movies with the lowest IMDB ratings were released in the 2010s.

trends in the release of christmas movies over time

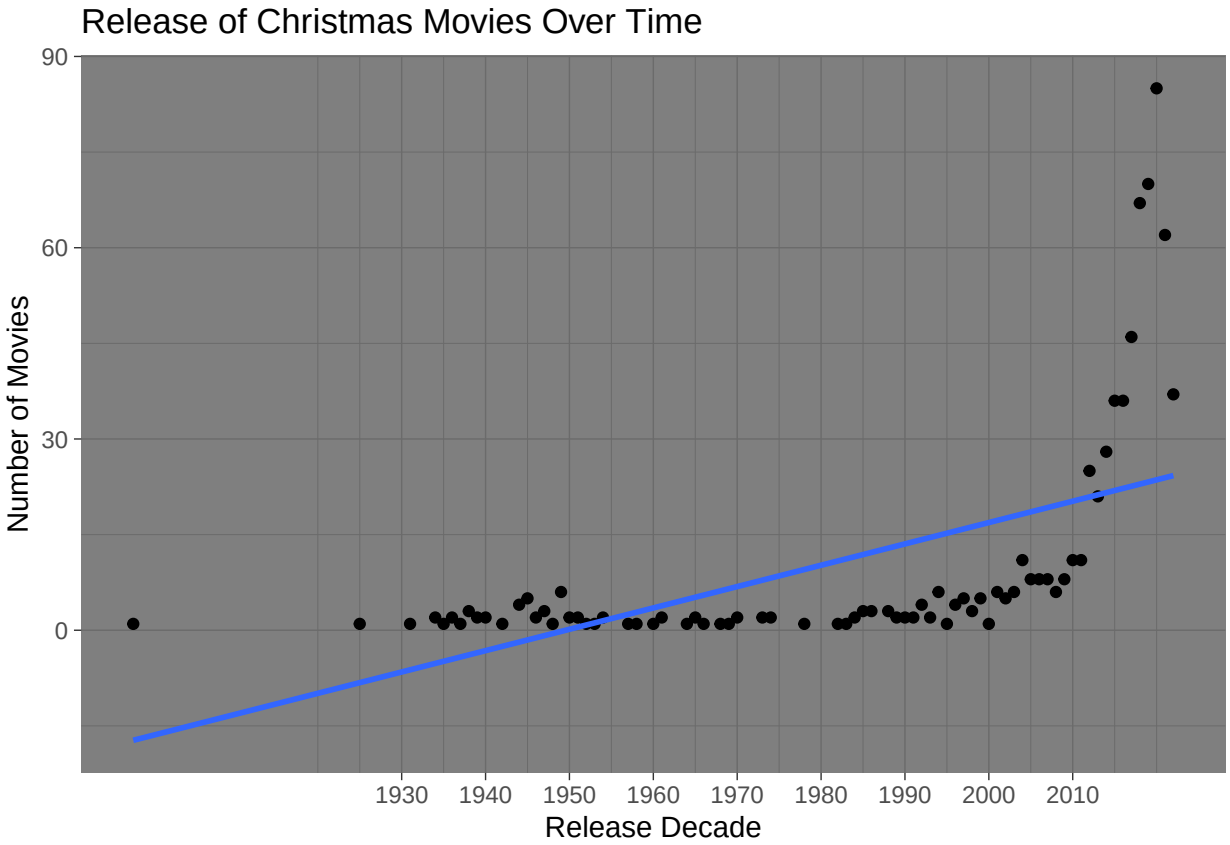
```
# Table of the frequency distribution of the release of christmas movies over time

release_decade = df1_primary %>%
  group_by(release_year) %>%
  summarise(n = n()) %>%
  arrange(desc(n)) %>%
  na.omit()

# Chart of the frequency distribution of the release of christmas movies over time with trendline and e

ggplot(release_decade, aes(x = release_year, y = n)) +
  geom_point() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  labs(title = "Release of Christmas Movies Over Time", x = "Release Decade", y = "Number of Movies") +
  theme_dark() +
  geom_smooth(method = "lm", se = FALSE) +
  scale_x_continuous(breaks = seq(1930, 2010, 10))
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



What does the chart of the release of Christmas movies over time tell us?

The chart of the release of Christmas movies over time tells us that the number of Christmas movies released has increased over time. The number of Christmas movies released has increased the most in the 2010s.