Netflix

陈思蓉 & 许小如

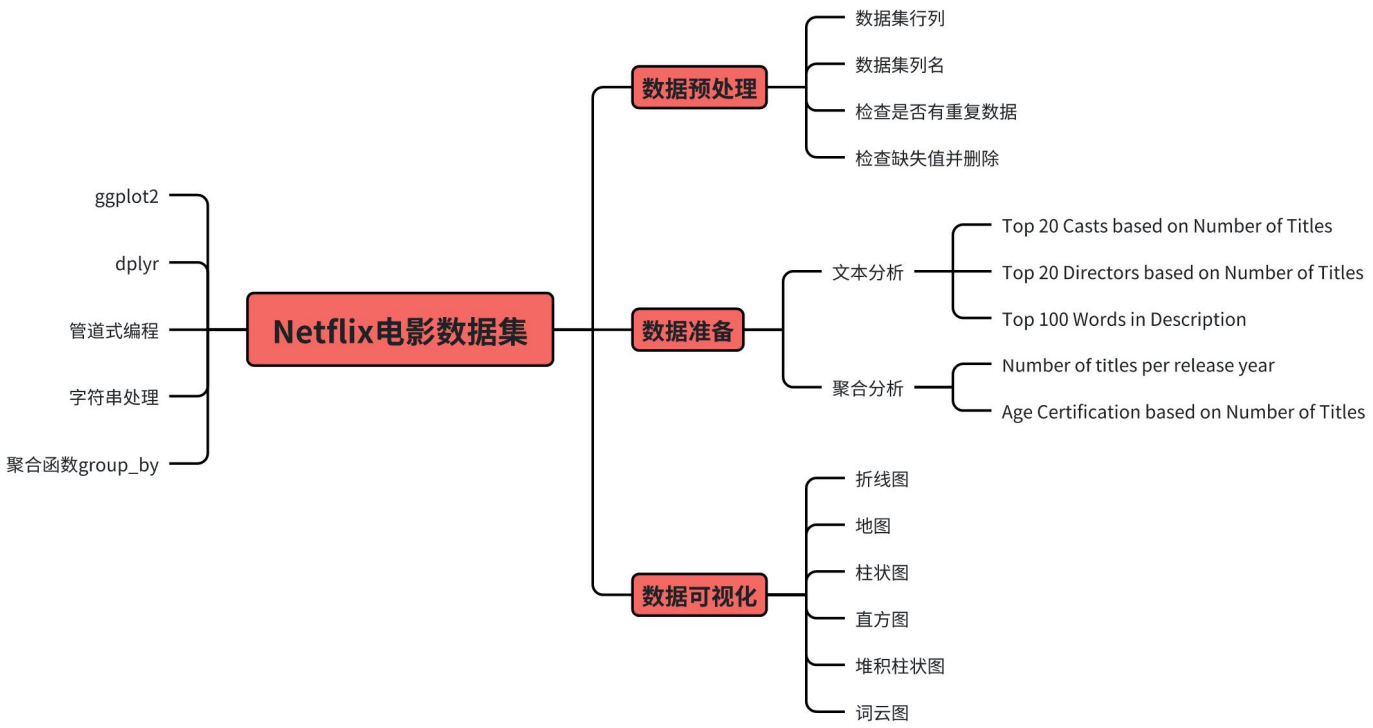
2024-03-25

Introduction to Datasets

With the rise and development of video streaming services such as Netflix, Hulu, Amazon Prime, etc., people's demand and viewing habits for movies and TV programs have changed. As one of the world's largest video streaming service providers, Netflix has a vast library of movies and TV programs, covering a variety of different types and styles of content.

This dataset is derived from Kaggle, which is a Netflix movie and TV show dataset, with a total of 8807 entries and 12 variable.

Mind map



Mind map

Data preprocessing

Installation package

```
library(dplyr)
library(rnaturalearth)
library(rnaturalearthdata)
library(sf)
library(ggplot2)
library(RColorBrewer)
library(wordcloud2)
library(wordcloud)
library(jiebaR)
library(tm)
```

Loading the dataset

```
df <- read.csv("C:\\Users\\Administrator\\Desktop\\应用线性数据集\\netflix.csv",encoding =
         "UTF-8",na.strings = "")

#print the name of columns
colname=colnames(df)
print(colname)
```

```
##  [1] "show_id"      "type"         "title"        "director"     "cast"
##  [6] "country"      "date_added"   "release_year" "rating"       "duration"
## [11] "listed_in"    "description"
```

Shape of Dataset

```
ncol(df)
```

```
## [1] 12
```

```
nrow(df)
```

```
## [1] 8807
```

Check for null values

Check if there are any null values in the data.

```
for(i in 1:ncol(df)){
  print(paste(colname[i],length(which(is.na(df[,i])))))
}
```

```
## [1] "show_id 0"
## [1] "type 0"
## [1] "title 0"
## [1] "director 2634"
## [1] "cast 825"
## [1] "country 831"
## [1] "date_added 10"
## [1] "release_year 0"
## [1] "rating 4"
## [1] "duration 3"
## [1] "listed_in 0"
## [1] "description 0"
```

Drop NULL values

Delete null values in data.

```
dfcopy = df[-which(is.na(df['director'])),]
dfcopy = dfcopy[-which(is.na(dfcopy['cast'])),]
dfcopy = dfcopy[-which(is.na(dfcopy['country'])),]
dfcopy = dfcopy[-which(is.na(dfcopy['rating'])),]
dfcopy = dfcopy[-which(is.na(dfcopy['duration'])),]
```

Check for duplicate values

Check if there are duplicate values in the data, and the result is none.

```
duplicated_rows <- df[duplicated(df), ]
any(duplicated_rows)
```

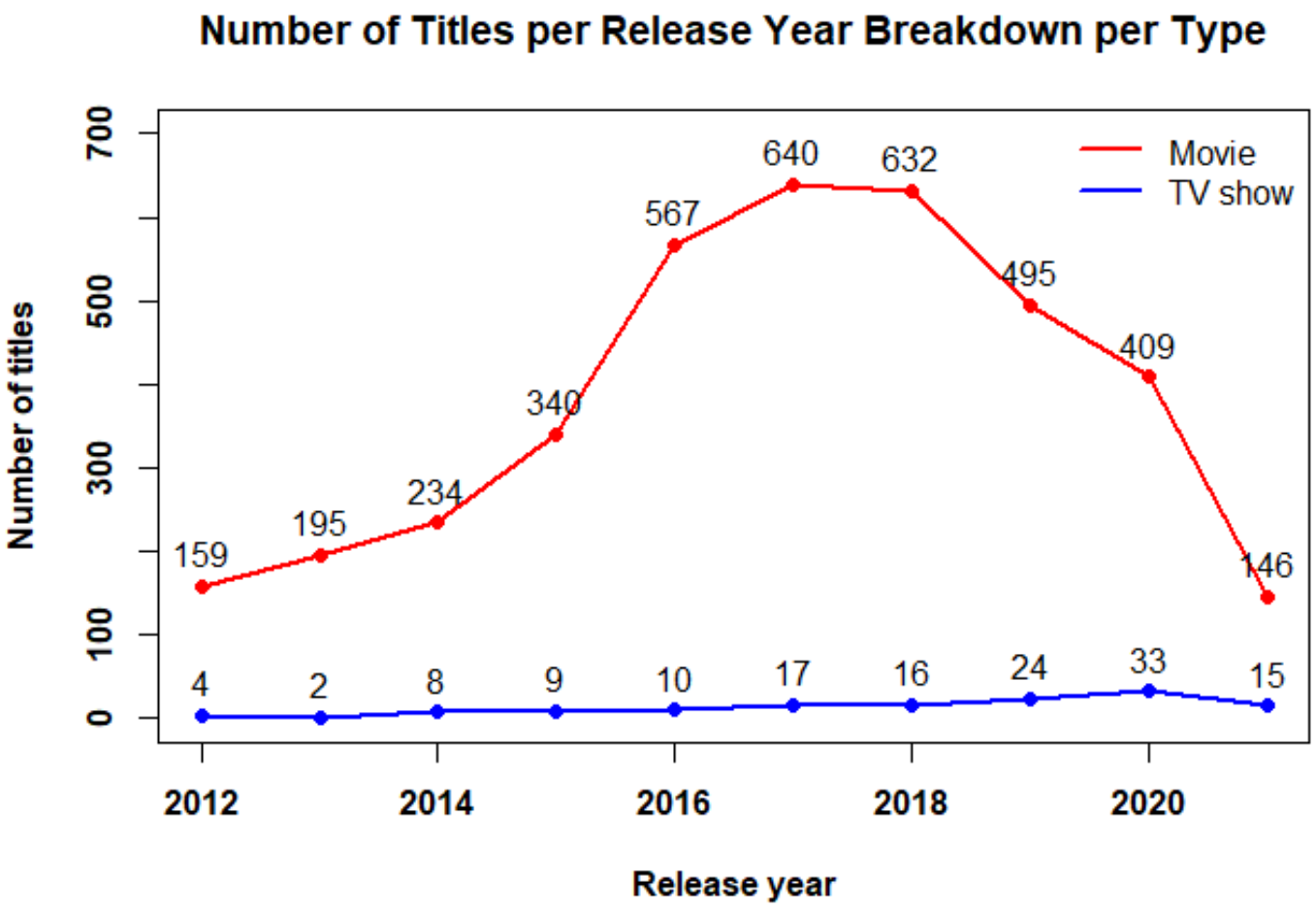
```
## [1] FALSE
```

Data Visualization

1.Number of titles per release year

```
juhe <- dfcopy%>% group_by(release_year,type)
titles <- summarise(juhe,count=n())
```

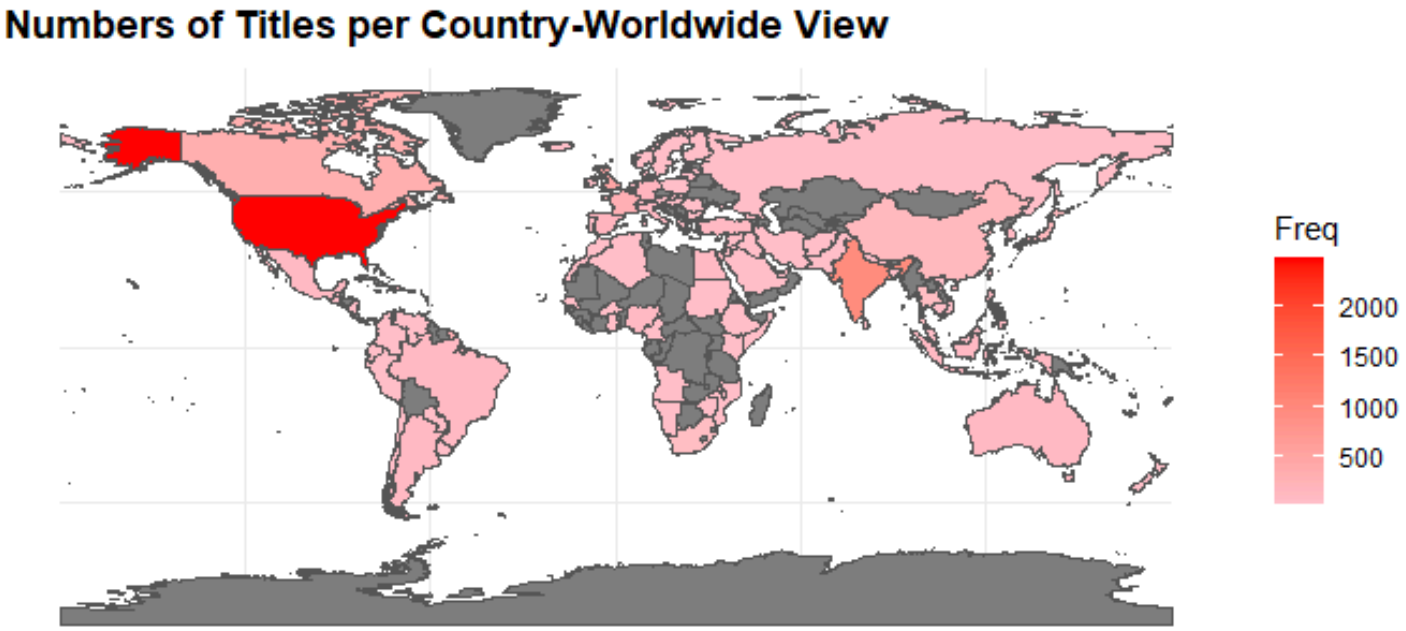
```r
titles <- titles[which(titles$release_year>2011 & titles$release_year<2022),]
titlesmovie <- titles[which(titles$type=='Movie'),]
titlestv <- titles[which(titles$type=='TV Show'),]



#draw
par(font.main = 2, font.lab = 2, font.axis = 2)
plot(titlesmovie$release_year, titlesmovie$count, type = "l", col = "red", ylim = c(0,
        700), xlab = "Release year", ylab = "Number of titles",lwd = 2)
lines(titlestv$release_year, titlestv$count, col = "blue",lwd = 2)

#add points
points(titlesmovie$release_year, titlesmovie$count, col = "red", pch = 16)
points(titlestv$release_year, titlestv$count, col = "blue", pch = 16)

#add legend
legend("topright", legend = c("Movie", "TV show"), col = c("red", "blue"), lwd = 2, bty =
        "n")

#add value
for (i in 1:length(titlesmovie$release_year)) {
  text(titlesmovie$release_year[i], titlesmovie$count[i], labels = titlesmovie$count[i],
        pos = 3)
  text(titlesmovie$release_year[i], titlestv$count[i], labels = titlestv$count[i], pos = 3)
}
# add title
title(main = "Number of Titles per Release Year Breakdown per Type")
```



2.Numbers of Titles per Country-Worldwide View

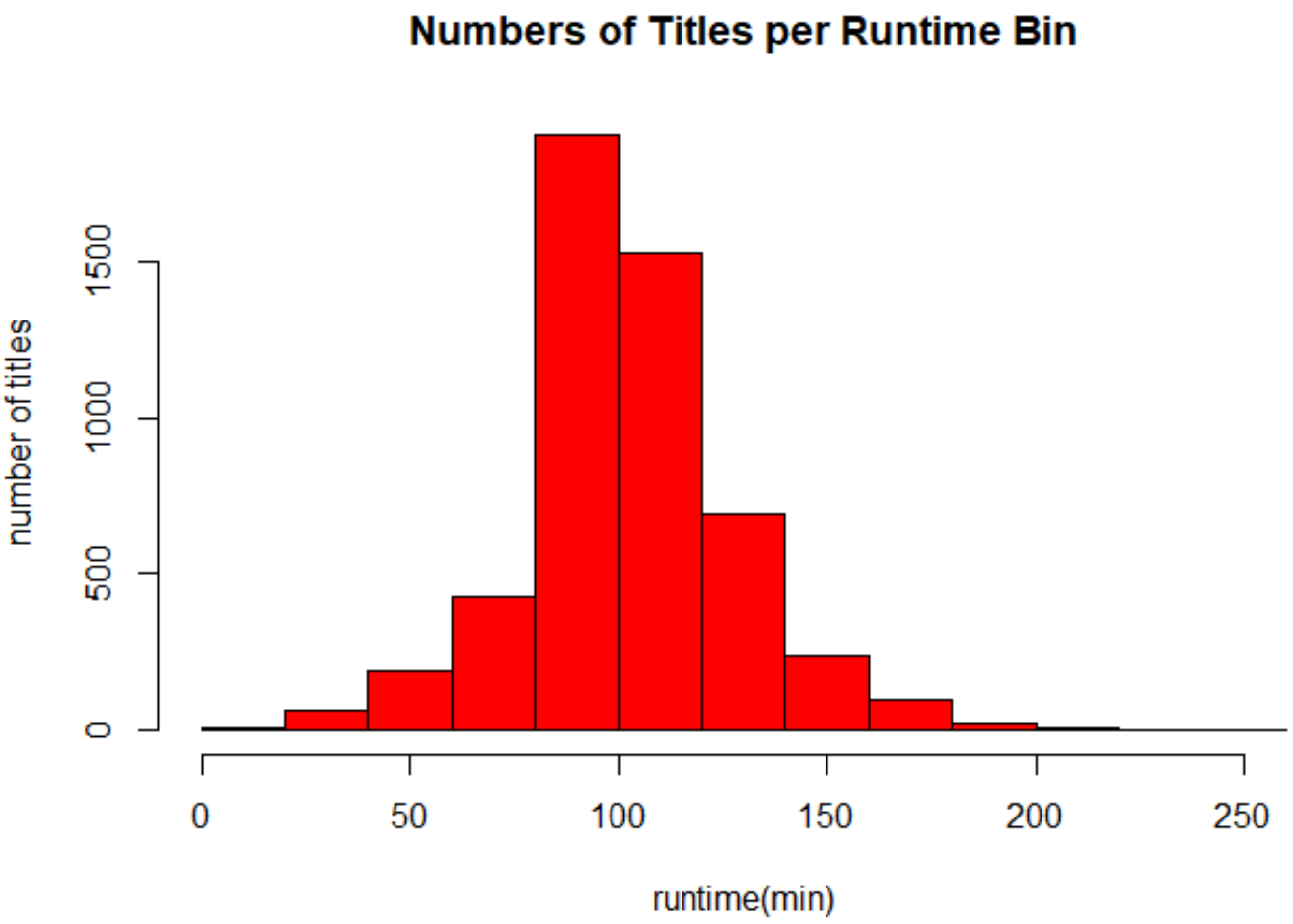
```
#country
countries <- unlist(strsplit(dfcopy$country, ", "))
countries <- trimws(countries)
countries[countries == "United States"] <- "United States of America"
frequency_table <- table(countries)

# Obtaining World Map Data
world_map <- ne_countries(scale = "medium", returnclass = "sf")

# Create a data frame containing columns of country names and numerical values
data <- data.frame(frequency_table)

# Merge data with world map data
world_map_data <- merge(world_map, data, by.x = "admin", by.y = "countries", all.x = TRUE)

# Draw a map
ggplot() +
  geom_sf(data = world_map_data, aes(fill = Freq)) +
  scale_fill_gradient(low = "pink", high = "red") +
  theme_minimal()+
```

```
  labs(title = "Numbers of Titles per Country-Worldwide View") +
  theme(plot.title = element_text(face = "bold"))
```



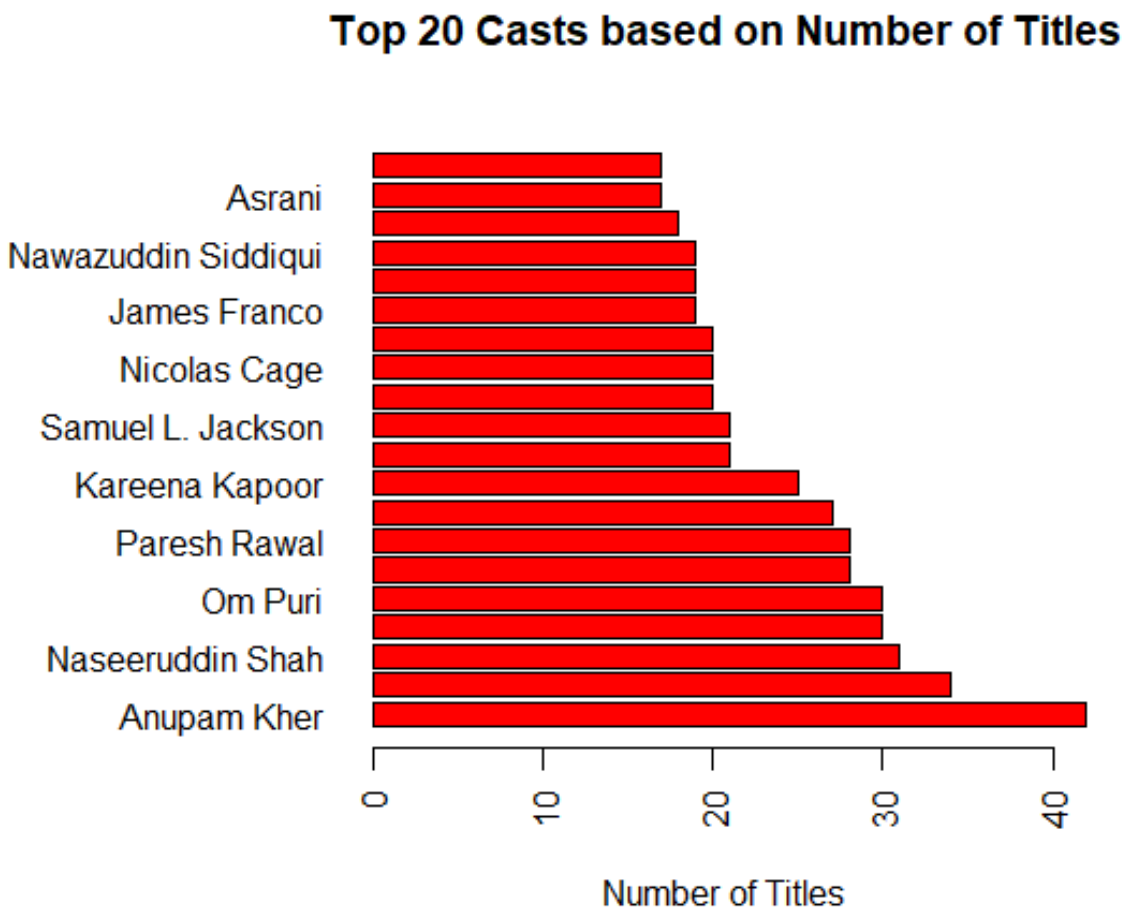
3.Number of titles per runtime bin(Movie)

```
dfcopy_movie <-dfcopy[which(dfcopy$type=='Movie'),]
dfcopy_movie$runtime <- sub(" min","",dfcopy_movie$duration)
dfcopy_movie$runtime <- as.numeric(dfcopy_movie$runtime)

hist(dfcopy_movie$runtime,
     breaks=9,
     col="red",
     xlab="runtime(min)",
     ylab="number of titles",
     main="Numbers of Titles per Runtime Bin")
```



4.Top 20 Casts based on Number of Titles

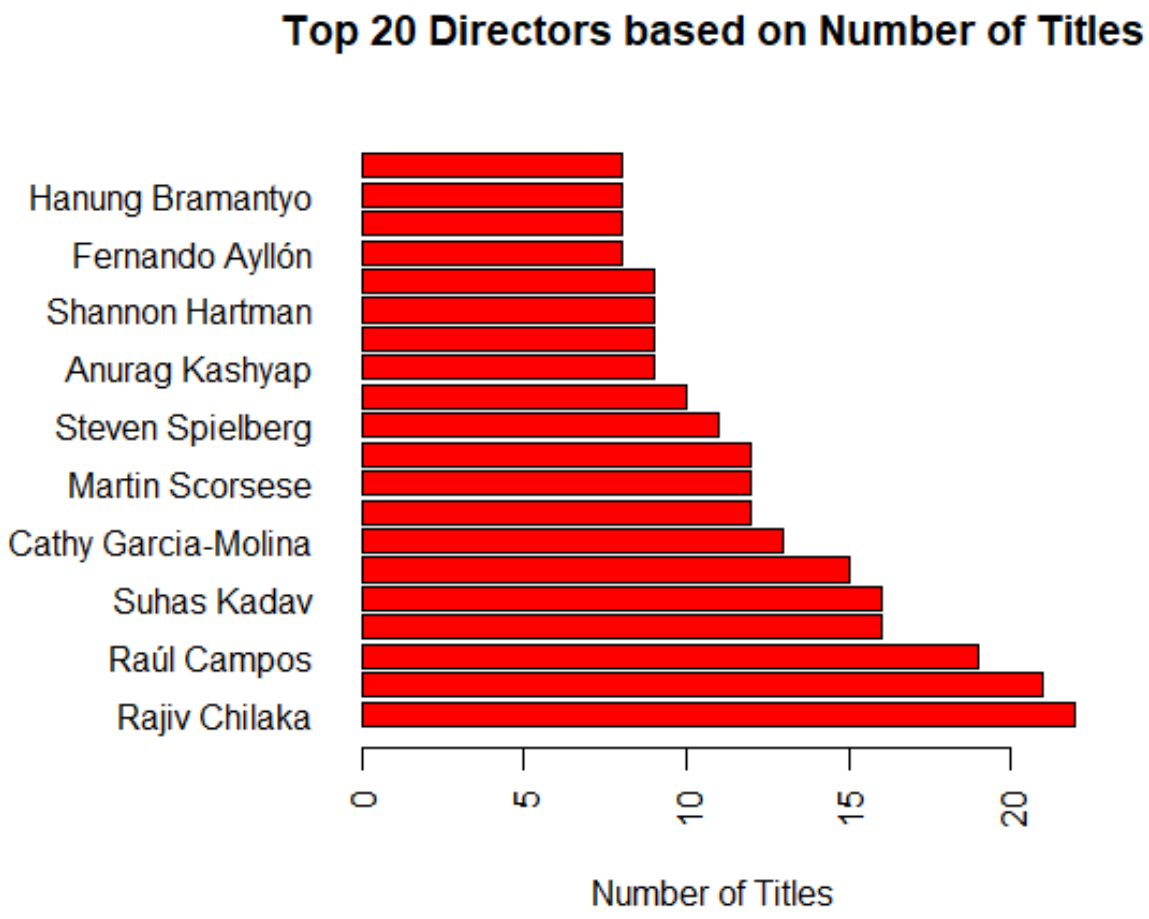
```
casts <- unlist(strsplit(dfcopy$cast, ", "))
casts <- trimws(casts)
casts <- data.frame(table(casts))
casts <- casts[order(casts$Freq, decreasing = TRUE),]
casts20 <- casts[1:20,]

par(mai = c(1, 2, 1, 1.5))
# Draw a bar chart and display Freq values at the right end of the column
barplot(casts20$Freq, names.arg = casts20$casts, xlab = "Number of Titles", col = "red",
        border = "black", horiz = TRUE, las = 2)

# Add title
title(main = "Top 20 Casts based on Number of Titles")
```



The chart presents the top 20 actors who participated in the movie in the form of a horizontal bar chart. We can see that the top three "model workers" are Anupam Kher, Naseeruddin Shah, and Om Puri.

5.Top 20 Casts based on Number of Titles

```
directors<-unlist(strsplit(df$director,", "))
directors <- trimws(directors)
directors <- data.frame(table(directors))
directors <- directors[order(directors$Freq, decreasing = TRUE),]
directors20 <- directors[1:20,]

par(mai = c(1, 2, 1, 1.5))
# Draw a bar chart and display Freq values at the right end of the column
barplot(directors20$Freq, names.arg = directors20$directors, xlab = "Number of Titles", col
        = "red", border = "black", horiz = TRUE, las = 2)

# Add title
title(main = "Top 20 Directors based on Number of Titles")
```



The top model in the directing industry is Rajiv Chilaka, who has directed many animations. If you are interested, click on this link.

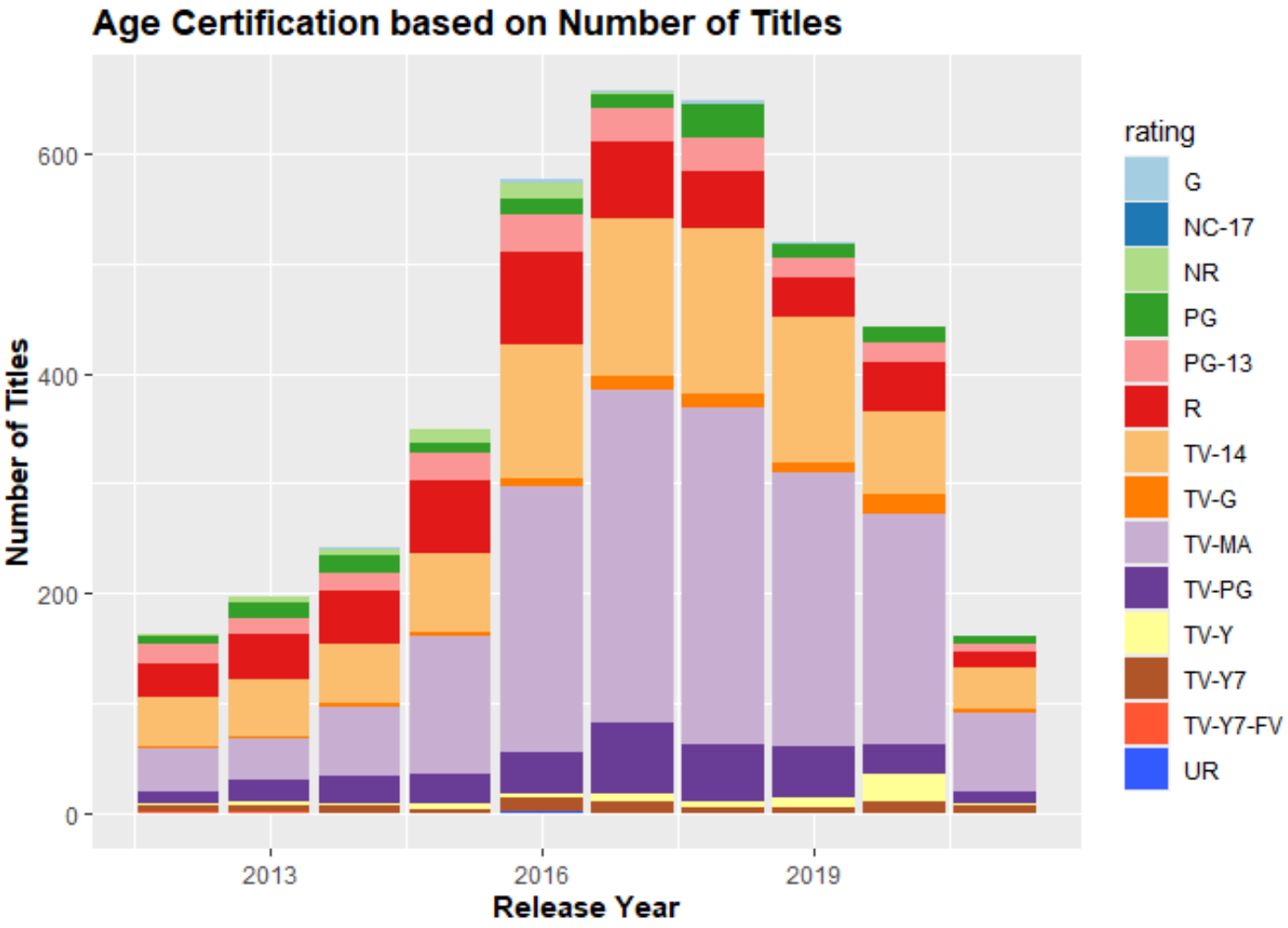
6.release_year-rating-titles

```
rating <- dfcopy%>%group_by(release_year,rating)
rating_sum <- summarise(rating,count=n())
```

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

```
rating_sum <- rating_sum[which(rating_sum$release_year>2011 &
         rating_sum$release_year<2022),]

# Get 12 colors from the Paired palette
paired_colors <- brewer.pal(12, "Paired")

# Customize six additional colors
custom_colors <- c("#FF5733", "#335EFF", "#33FF57", "#FF33C7", "#FFD700", "#9400D3")

# Merge Paired palette colors and custom colors
all_colors <- c(paired_colors, custom_colors[1:6])
```

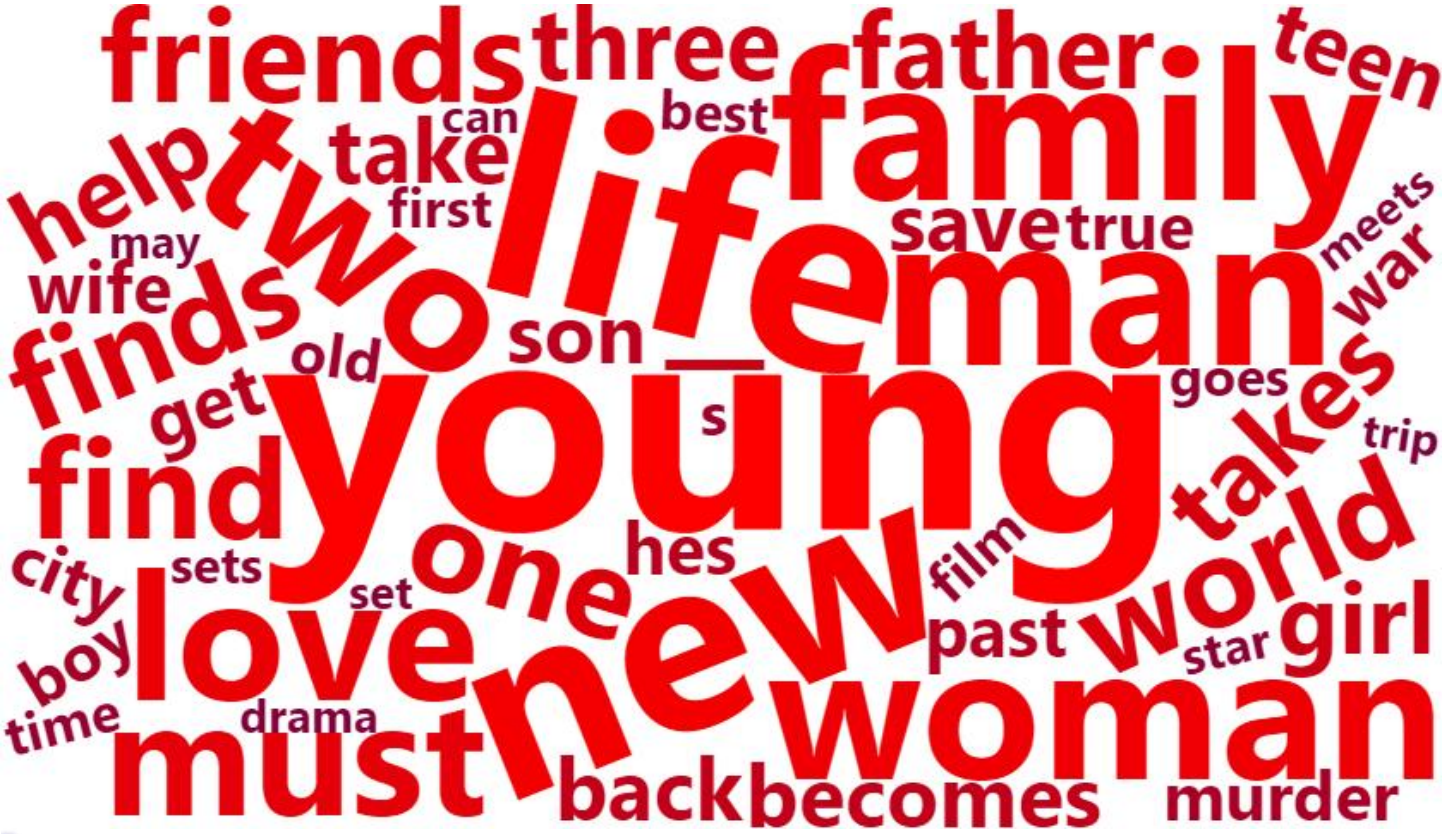
```
# Draw stacked bar charts based on the rating_sum dataset, using 18 different colors
ggplot(data = rating_sum, aes(x = release_year, y = count, fill = rating)) +
  geom_bar(stat = "identity", position = "stack") +

  # Set axis labels and titles
  labs(x = "Release Year",
       y = "Number of Titles",
       title = "Age Certification based on Number of Titles") +

  # Fill with 18 different colors
  scale_fill_manual(values = all_colors) +

  # Specify label style
  theme(axis.title.x = element_text(face = "bold"),
        axis.title.y = element_text(face = "bold"),
        plot.title = element_text(face = "bold"))
```



According to the stacked bar chart, it can be found that “purple” accounts for a large proportion in each column, so there are more films with a rating of “TV-MA”.

7.Draw word cloud diagram with “description”

```r
# Extract description text data
text <- dfcopy$description

# Create a text corpus object
corpus <- Corpus(VectorSource(text))

# Preprocess text
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, removeWords, stopwords("english"))
for(i in 1:length(corpus)){
  corpus[[i]]$content <- trimws(corpus[[i]]$content)
  corpus[[i]]$content <- gsub("  "," ",corpus[[i]]$content)
  corpus[[i]]$content <- gsub("   "," ",corpus[[i]]$content)
  corpus[[i]]$content <- gsub("  "," ",corpus[[i]]$content)
}

# Tokenize the sentences (split into words)
corpus <- tm_map(corpus, content_transformer(strsplit), split = " ")
#corpus <- sub(corpus,'c(','')

#Create word frequency matrix
corpus1<-unlist(corpus)
table_corpus1<-table(corpus1)
table_corpus2<-data.frame(table_corpus1)
table_corpus2<-table_corpus2[order(table_corpus2$Freq,decreasing = TRUE),]
table_corpus2<-table_corpus2[-which(table_corpus2$corpus1=='-'),]
table_corpus3<-table_corpus2[1:100,]
#wordcloud2(data=table_corpus3)
custom_colors <- colorRampPalette(c("#FF0000", "navy"))(100)

# Generate a word cloud map and specify a custom color palette
wordcloud2(data = table_corpus3, color = custom_colors)
```



The word cloud chart shows the top 100 words that appear most frequently in evaluations. According to words such as “young,” “life,” and “love,” it can be seen that the film may involve youthful love.