Final Project - 650

Vikas Rayala & Vikas Reddy B

Netflix Analysis

Data Cleaning & Processing :

```
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     2.1.4
v ggplot2   3.4.3     v stringr   1.5.0
v lubridate 1.9.2     v tibble    3.2.1
v purrr     1.0.2     v tidyr     1.3.0

-- 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 becor
```

```r
library(lubridate)
library(readr)
library(wordcloud)
```

```
Loading required package: RColorBrewer
```

```r
library(tm)
```

```
Loading required package: NLP

Attaching package: 'NLP'

The following object is masked from 'package:ggplot2':

    annotate
```

```r
#Load data
netflix_data <- read_csv("~/Stat 650/netflix_titles.csv")
```

```
Rows: 8807 Columns: 12
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (11): show_id, type, title, director, cast, country, date_added, rating,...
dbl  (1): release_year

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```r
#About data
dim(netflix_data)
```

```
[1] 8807   12
```

```r
names(netflix_data)
```

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

```r
head(netflix_data)
```

```
# A tibble: 6 x 12
  show_id type    title    director cast  country date_added release_year rating
  <chr>   <chr>   <chr>    <chr>    <chr> <chr>   <chr>             <dbl> <chr>
1 s1      Movie   Dick Jo~ Kirsten~ <NA>  United~ September~         2020 PG-13
2 s2      TV Show Blood &~ <NA>     Ama ~ South ~ September~         2021 TV-MA
3 s3      TV Show Ganglan~ Julien ~ Sami~ <NA>    September~         2021 TV-MA
4 s4      TV Show Jailbir~ <NA>     <NA>  <NA>    September~         2021 TV-MA
5 s5      TV Show Kota Fa~ <NA>     Mayu~ India   September~         2021 TV-MA
6 s6      TV Show Midnigh~ Mike Fl~ Kate~ <NA>    September~         2021 TV-MA
# i 3 more variables: duration <chr>, listed_in <chr>, description <chr>
```

```r
str(netflix_data)
```

```
spc_tbl_ [8,807 x 12] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
 $ show_id     : chr [1:8807] "s1" "s2" "s3" "s4" ...
 $ type        : chr [1:8807] "Movie" "TV Show" "TV Show" "TV Show" ...
 $ title       : chr [1:8807] "Dick Johnson Is Dead" "Blood & Water" "Ganglands" "Jailbirds N
 $ director    : chr [1:8807] "Kirsten Johnson" NA "Julien Leclercq" NA ...
 $ cast        : chr [1:8807] NA "Ama Qamata, Khosi Ngema, Gail Mabalane, Thabang Molaba, Di
 $ country     : chr [1:8807] "United States" "South Africa" NA NA ...
 $ date_added  : chr [1:8807] "September 25, 2021" "September 24, 2021" "September 24, 2021"
 $ release_year: num [1:8807] 2020 2021 2021 2021 2021 ...
 $ rating      : chr [1:8807] "PG-13" "TV-MA" "TV-MA" "TV-MA" ...
 $ duration    : chr [1:8807] "90 min" "2 Seasons" "1 Season" "1 Season" ...
 $ listed_in   : chr [1:8807] "Documentaries" "International TV Shows, TV Dramas, TV Mysteri
 $ description : chr [1:8807] "As her father nears the end of his life, filmmaker Kirsten Jo
 - attr(*, "spec")=
  .. cols(
  ..   show_id = col_character(),
  ..   type = col_character(),
  ..   title = col_character(),
  ..   director = col_character(),
  ..   cast = col_character(),
  ..   country = col_character(),
  ..   date_added = col_character(),
  ..   release_year = col_double(),
  ..   rating = col_character(),
  ..   duration = col_character(),
```

```
  ..   listed_in = col_character(),
  ..   description = col_character()
  .. )
 - attr(*, "problems")=<externalptr>
```

```
#Summary Stats
summary(netflix_data)
```

```
  show_id              type              title             director
Length:8807        Length:8807        Length:8807        Length:8807
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character



    cast              country           date_added         release_year
Length:8807        Length:8807        Length:8807        Min.   :1925
Class :character   Class :character   Class :character   1st Qu.:2013
Mode  :character   Mode  :character   Mode  :character   Median :2017
                                                         Mean   :2014
                                                         3rd Qu.:2019
                                                         Max.   :2021
   rating            duration          listed_in          description
Length:8807        Length:8807        Length:8807        Length:8807
Class :character   Class :character   Class :character   Class :character
Mode  :character   Mode  :character   Mode  :character   Mode  :character
```

```
#Mutating date to date format
netflix_data <- netflix_data %>%
  mutate(date_added = mdy(date_added))
```

```
#Grouping by type of shows
netflix_data %>%
 group_by(type) %>%
 summarize(count = n())
```

```
# A tibble: 2 x 2
```

```
  type    count
  <chr>   <int>
1 Movie    6131
2 TV Show  2676
```

cleaning data:

```
#Printing Null Values for each variable
colSums(is.na(netflix_data))
```

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

```
#Omiting Null Values & saving data in different name
netflix <- netflix_data %>%
                filter(
                        show_id != "n/a",
                        type != "n/a",
                        title != "n/a",
                        director != "n/a",
                        cast != "n/a",
                        country != "n/a")
head(netflix)
```

```
# A tibble: 6 x 12
  show_id type    title    director cast  country date_added release_year rating
  <chr>   <chr>   <chr>    <chr>    <chr> <chr>   <date>             <dbl> <chr>
1 s8      Movie   Sankofa  Haile G~ Kofi~ United~ 2021-09-24         1993 TV-MA
2 s9      TV Show The Gre~ Andy De~ Mel ~ United~ 2021-09-24         2021 TV-14
3 s10     Movie   The Sta~ Theodor~ Meli~ United~ 2021-09-24         2021 PG-13
4 s13     Movie   Je Suis~ Christi~ Luna~ German~ 2021-09-23         2021 TV-MA
5 s25     Movie   Jeans    S. Shan~ Pras~ India   2021-09-21         1998 TV-14
6 s28     Movie   Grown U~ Dennis ~ Adam~ United~ 2021-09-20         2010 PG-13
# i 3 more variables: duration <chr>, listed_in <chr>, description <chr>
```

```r
#Grouping By Rating
netflix_data %>%
  group_by(rating) %>%
  summarise(num_ratings = n())
```

```
# A tibble: 18 x 2
   rating   num_ratings
   <chr>          <int>
 1 66 min             1
 2 74 min             1
 3 84 min             1
 4 G                 41
 5 NC-17              3
 6 NR                80
 7 PG               287
 8 PG-13            490
 9 R                799
10 TV-14           2160
11 TV-G             220
12 TV-MA           3207
13 TV-PG            863
14 TV-Y             307
15 TV-Y7            334
16 TV-Y7-FV           6
17 UR                 3
18 <NA>               4
```

```r
#Checking distinct values for title & Show_id
n_distinct(netflix_data$show_id)
```

```
[1] 8807
```

```r
n_distinct(netflix_data$title)
```

```
[1] 8807
```

```r
#Anlysing
#Movies by country
```

```r
movies_world <- netflix_data %>%
  group_by(type) %>%
  group_by(country) %>%
  summarise(num_movies_country = n()) %>%
  arrange(desc(num_movies_country)) %>%
  slice(1:20)

head(movies_world)
```

```
# A tibble: 6 x 2
  country        num_movies_country
  <chr>                       <int>
1 United States                2818
2 India                         972
3 <NA>                          831
4 United Kingdom                419
5 Japan                         245
6 South Korea                   199
```

```r
#Produced content by year
netflix_data %>%
  group_by(release_year) %>%
  summarise(year_produce = n()) %>%
  arrange(desc(year_produce)) %>%
  slice(1:10)
```

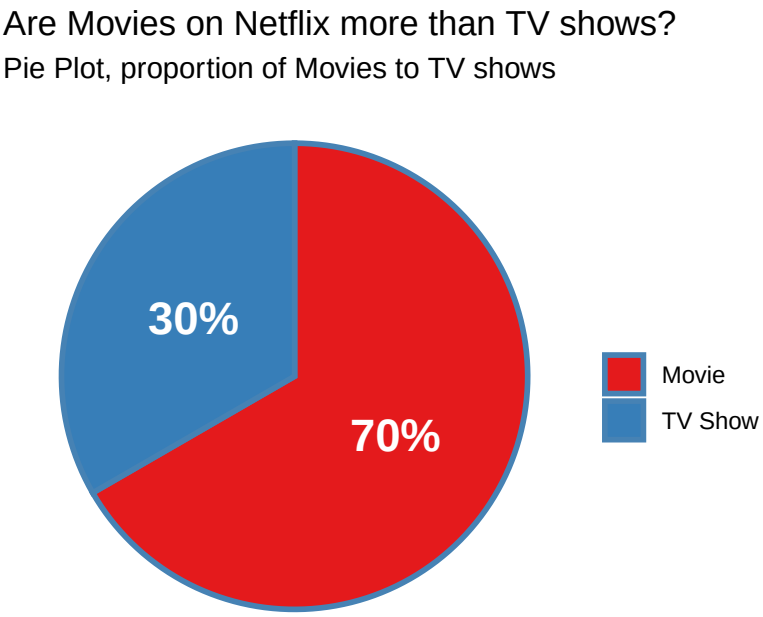
```
# A tibble: 10 x 2
   release_year year_produce
          <dbl>        <int>
 1         2018         1147
 2         2017         1032
 3         2019         1030
 4         2020          953
 5         2016          902
 6         2021          592
 7         2015          560
 8         2014          352
 9         2013          288
10         2012          237
```

EDA:

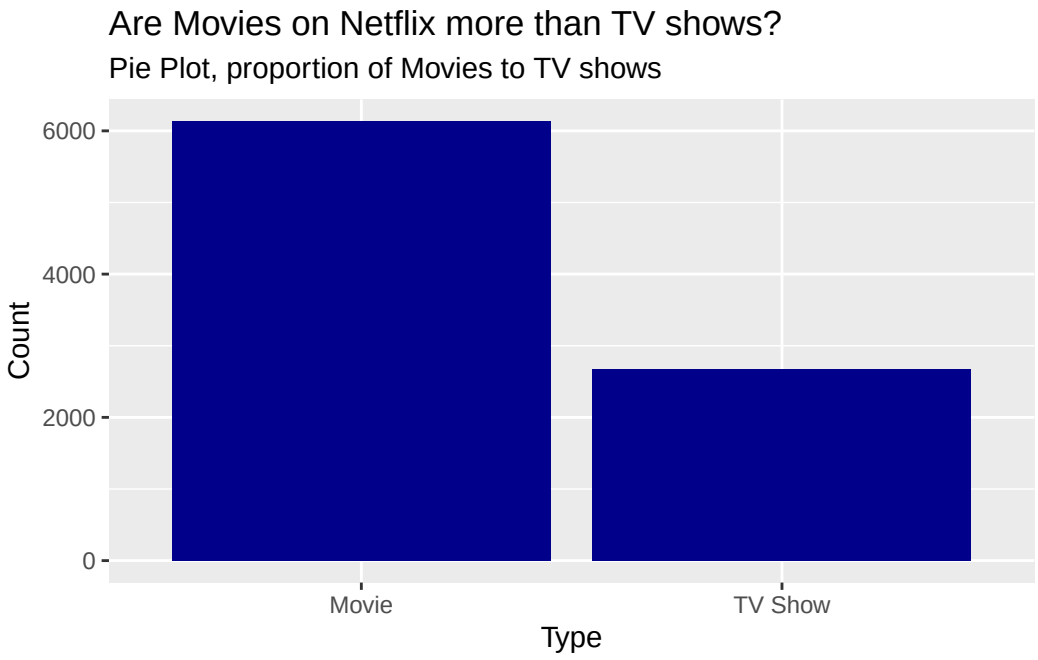
1. Are Movies on Netflix more than TV shows?

```
# With Null values data.
netflix_data %>% count(type, sort = T) %>%

  mutate(prop = paste0(round(n / sum(n) * 100, 0), "%")) %>%
  ggplot(aes(x = "", y = prop, fill = type)) +
  geom_bar(
    stat = "identity",
    width = 1,
    color = "steelblue",
    size = 1
  ) +
  coord_polar("y", start = 0) +
  geom_text(
    aes(y = prop, label = prop),
    position = position_stack(vjust = 0.5),
    size = 6,
    col = "white",
    fontface = "bold"
  ) +
  scale_fill_manual (values = c('#e41a1c', '#377eb8')) +
  theme_void() +
  labs(
    title = "Are Movies on Netflix more than TV shows?",
    subtitle = "Pie Plot, proportion of Movies to TV shows",
    fill = ""
  )
```

```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
```

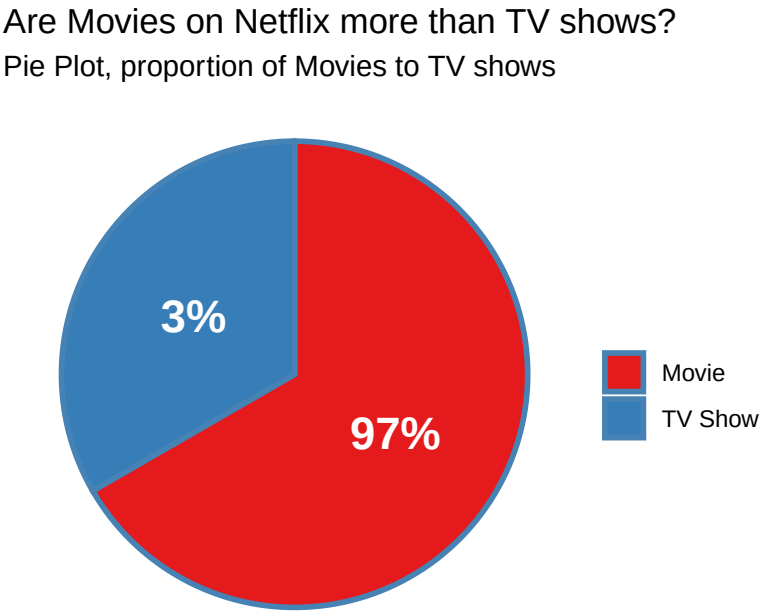



```
ggplot(data=netflix_data, aes(x=type))+geom_bar(fill = "dark blue")+
labs(
    title = "Are Movies on Netflix more than TV shows?",
    subtitle = "Pie Plot, proportion of Movies to TV shows",
    fill = ""
  ) +
xlab("Type")+
ylab("Count")
```

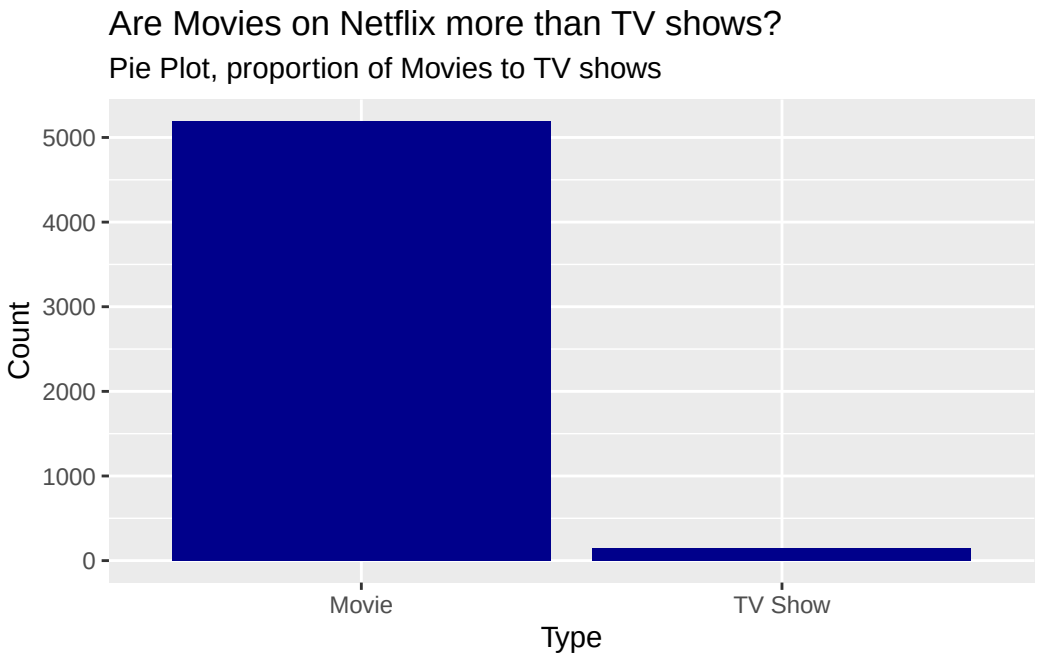



```
# Without Null Values data

netflix %>% count(type, sort = T) %>%

  mutate(prop = paste0(round(n / sum(n) * 100, 0), "%")) %>%
  ggplot(aes(x = "", y = prop, fill = type)) +
  geom_bar(
    stat = "identity",
    width = 1,
    color = "steelblue",
    size = 1
  ) +
  coord_polar("y", start = 0) +
  geom_text(
    aes(y = prop, label = prop),
    position = position_stack(vjust = 0.5),
    size = 6,
    col = "white",
    fontface = "bold"
  ) +
  scale_fill_manual (values = c('#e41a1c', '#377eb8')) +
  theme_void() +
```

```
  labs(
    title = "Are Movies on Netflix more than TV shows?",
    subtitle = "Pie Plot, proportion of Movies to TV shows",
    fill = ""
  )
```

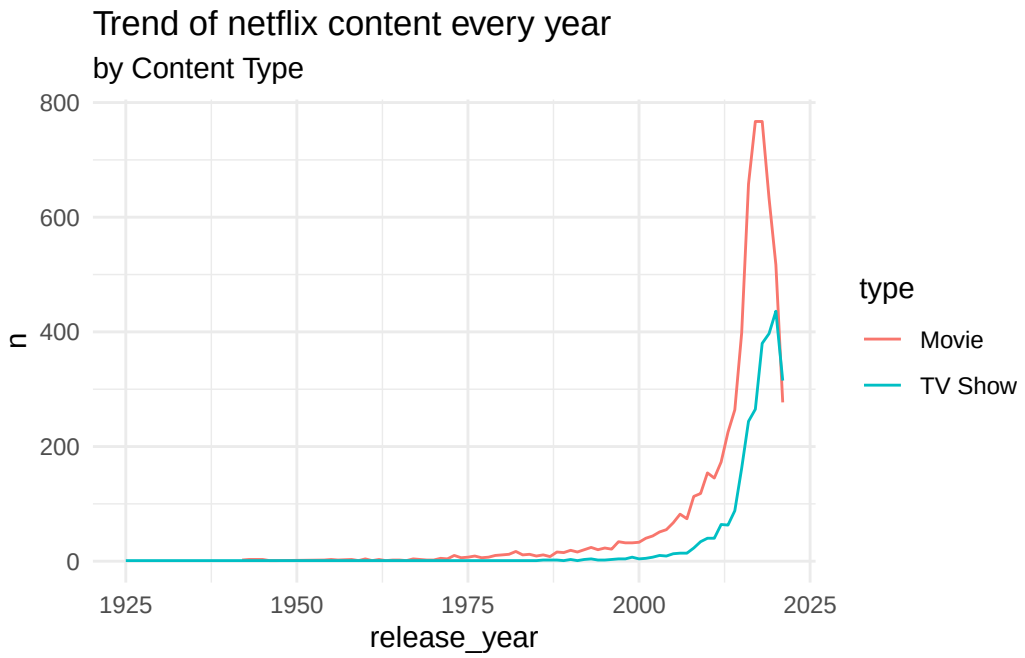



```
# Histogram
ggplot(data=netflix, aes(x=type))+geom_bar(fill = "dark blue")+
labs(
    title = "Are Movies on Netflix more than TV shows?",
    subtitle = "Pie Plot, proportion of Movies to TV shows",
    fill = ""
  ) +
xlab("Type")+
ylab("Count")
```



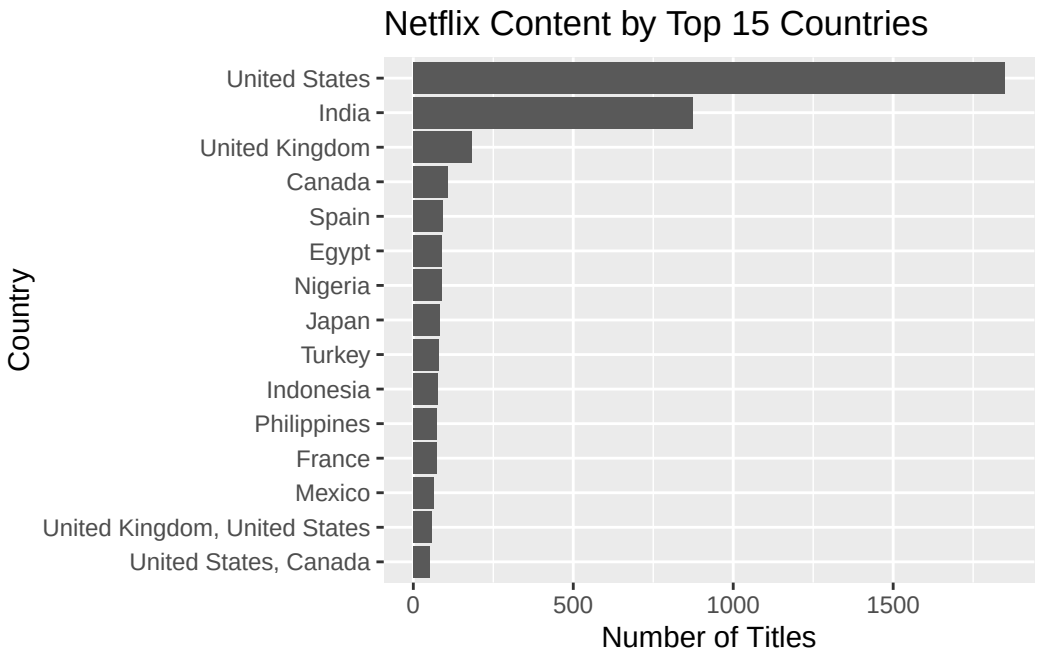
What is the trend of content over the years?

```
netflix_data %>%
 filter(release_year != 2022) %>%
 group_by(type, release_year) %>%
 count() %>%
 ggplot() + geom_line(aes(x = release_year, y = n, group = type, color = type)) +
  labs(title = 'Trend of netflix content every year',
       subtitle = 'by Content Type') +
theme_minimal()
```

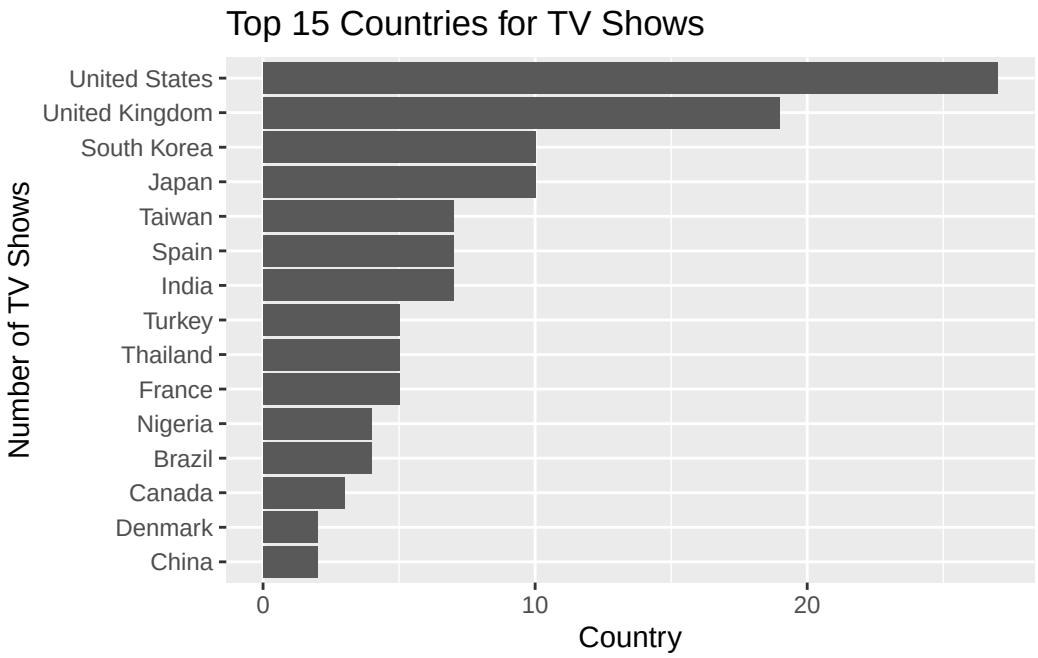


What are the top countries for content in netflix?

```r
# Filter to top 15 countries by count
top_countries <- netflix %>%
  count(country) %>%
  top_n(15, wt = n) %>%
  arrange(desc(n))

# Bar plot
ggplot(top_countries, aes(x = reorder(country, n), y = n)) +
  geom_col() +
  labs(
    title = "Netflix Content by Top 15 Countries",
    x = "Country",
    y = "Number of Titles"
  ) +
  coord_flip()
```

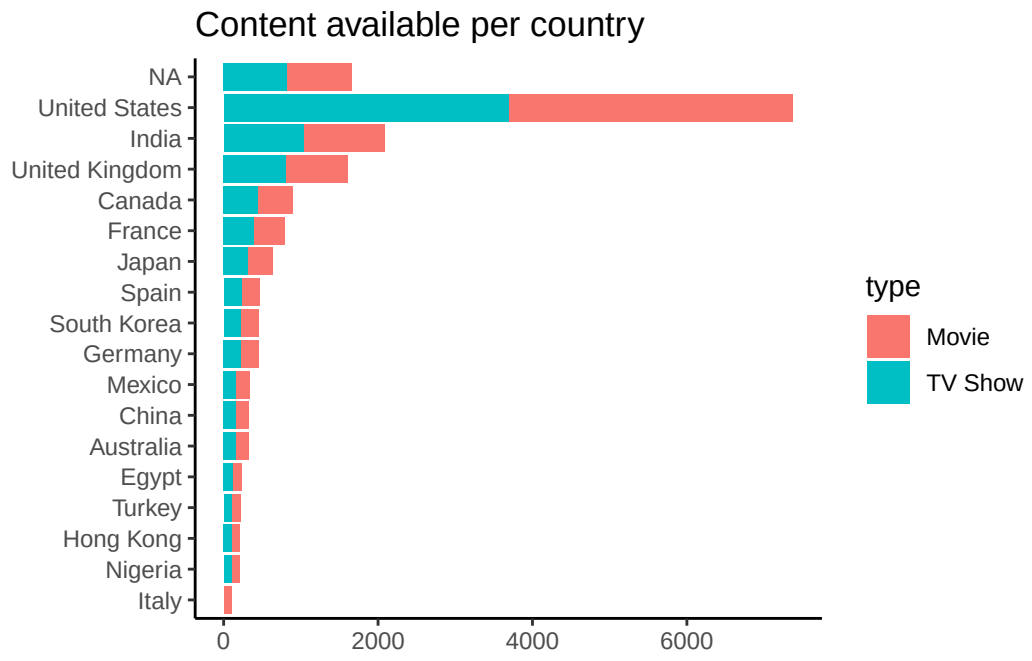



```r
# Top 15 countries for TV shows
netflix %>%
  filter(type == "TV Show") %>%
  count(country, sort = TRUE) %>%
  head(15) %>%
  ggplot(aes(x = reorder(country, n), y = n)) +
  geom_col() +
  coord_flip() +
  labs(title = "Top 15 Countries for TV Shows",
       x = "Number of TV Shows",
       y = "Country")
```




```
df_country <- netflix_data %>%
    mutate(country = strsplit(as.character(country), ",")) %>%
    unnest(country) %>%
      mutate(country = trimws(country, which = c("left")))#eliminate space on the left sid

 df_country <- df_country %>%
  group_by(country)%>%
    add_tally()

 df_country <- df_country%>%
   select(country,n,type) %>%
   unique()
 df_country_top5 <- df_country[order(-df_country$n),]



 df_country_top5 <- df_country_top5[1:35,]

 ggplot(df_country_top5, aes(x = reorder(country, n), y = n, fill = type))+
   geom_bar(stat = "identity")+
   coord_flip()+
   theme_classic()+
```

```
  theme(axis.title.x = element_blank(),
        axis.title.y = element_blank())+
  labs(title="Content available per country", x = "Amount of content")
```



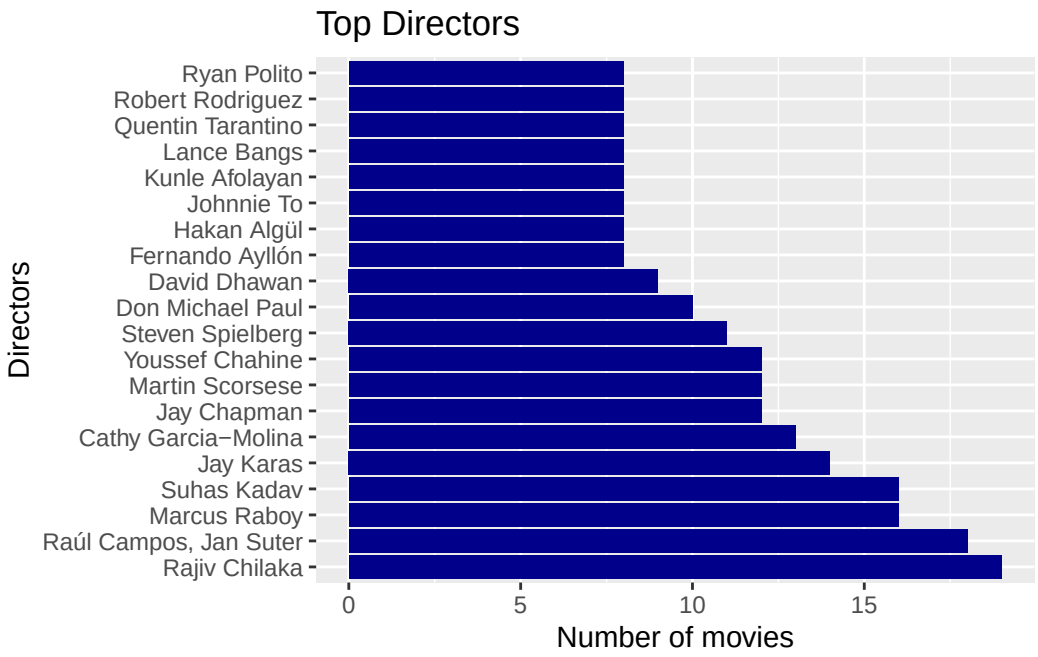
who are the TOP DIRECTORS for netflix movies and TV shows?

```
  directors <- netflix_data %>%
  group_by(director)%>%
  filter(director!="")%>%
  summarize(number = n())%>%
  arrange(desc(number))%>%
  slice(1:20)
  head(directors)
```

```
# A tibble: 6 x 2
  director                number
  <chr>                    <int>
1 Rajiv Chilaka               19
2 Raúl Campos, Jan Suter      18
3 Marcus Raboy                16
4 Suhas Kadav                 16
```

```
5 Jay Karas                  14
6 Cathy Garcia-Molina        13
```

```
ggplot(data=directors, aes(x=reorder(director, - number), y=number)) +
       geom_col(fill='dark blue') +
labs(title = "Top Directors") +
xlab("Directors")+
ylab("Number of movies")+
coord_flip()
```



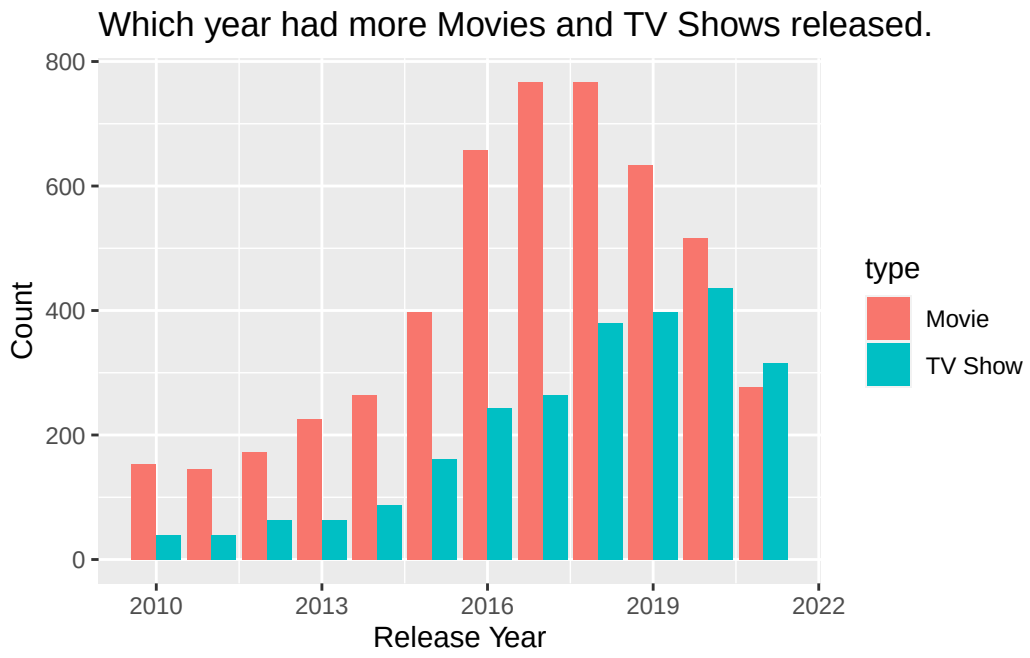
Which year had more Movies and TV Shows released?

```
netflix_years <- netflix_data%>%
filter(release_year>=2010)%>%
group_by(type)%>%
arrange()

head(netflix_years)
```

```
# A tibble: 6 x 12
# Groups:   type [2]
```

```
  show_id type    title    director cast  country date_added release_year rating
  <chr>   <chr>   <chr>    <chr>    <chr> <chr>   <date>             <dbl> <chr>
1 s1      Movie   Dick Jo~ Kirsten~ <NA>  United~ 2021-09-25          2020 PG-13
2 s2      TV Show Blood &~ <NA>     Ama ~ South ~ 2021-09-24          2021 TV-MA
3 s3      TV Show Ganglan~ Julien ~ Sami~ <NA>    2021-09-24          2021 TV-MA
4 s4      TV Show Jailbir~ <NA>     <NA>  <NA>    2021-09-24          2021 TV-MA
5 s5      TV Show Kota Fa~ <NA>     Mayu~ India   2021-09-24          2021 TV-MA
6 s6      TV Show Midnigh~ Mike Fl~ Kate~ <NA>    2021-09-24          2021 TV-MA
# i 3 more variables: duration <chr>, listed_in <chr>, description <chr>
```

```
ggplot(data=netflix_years, aes(x=release_year,fill=type))+geom_bar(position=position_dodge
labs(title = "Which year had more Movies and TV Shows released.") +
xlab("Release Year")+
ylab("Count")
```



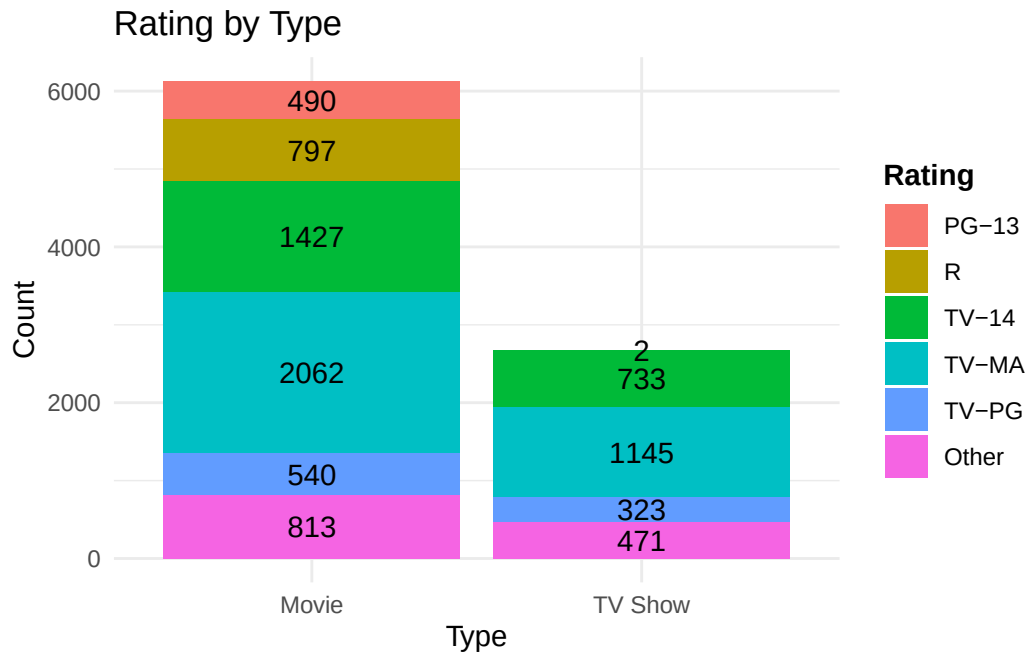
What are the ratings for different type of content?

```
library(ggplot2)
library(dplyr)

# Filter and transform data
```

```r
filtered_data <- netflix_data %>%
  select(rating, type) %>%
  filter(!is.na(rating)) %>%
  mutate(rating = fct_lump(rating, 5)) %>%
  group_by(rating, type) %>%
  summarise(Count = n()) %>%
  arrange(Count)
```

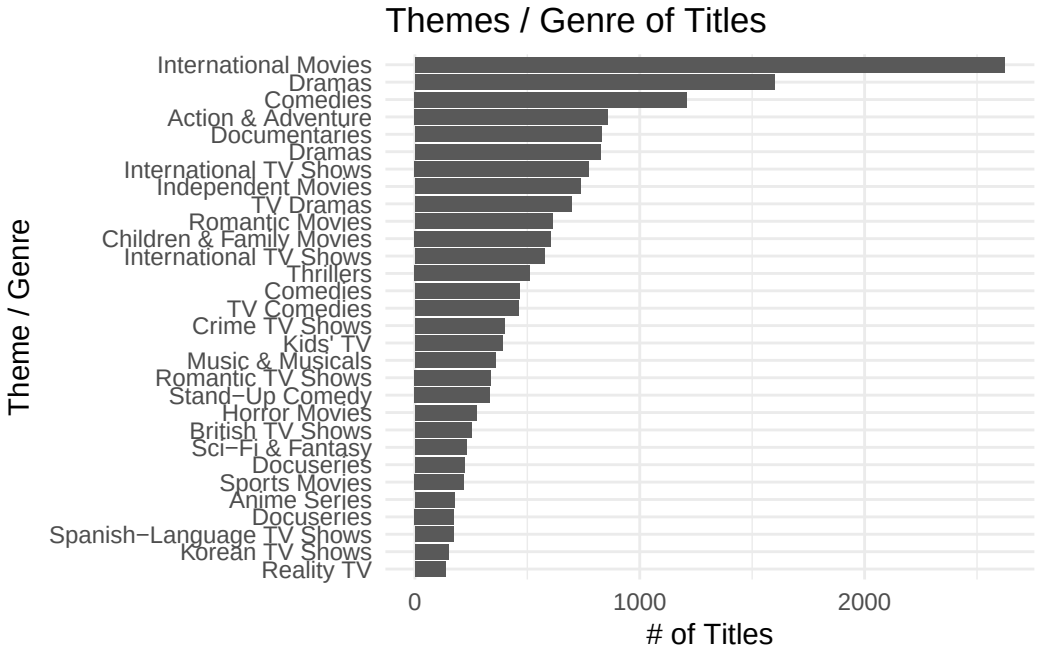
```
`summarise()` has grouped output by 'rating'. You can override using the
`.groups` argument.
```

```r
# Create bar plot using ggplot2
ggplot(filtered_data, aes(x = type, y = Count, fill = rating, label = Count)) +
  geom_bar(stat = "identity") +
  geom_text(position = position_stack(vjust = 0.5), size = 4) +
  theme_minimal() +
  labs(title = "Rating by Type",
       y = "Count",
       x = "Type") +
  theme(legend.title = element_text(face = "bold")) +
  guides(fill = guide_legend(title = "Rating"))
```



what are top & bottom genres of content on Netflix

```
netflix_data %>%
 select(listed_in) %>%
 mutate(listed_in = str_split(listed_in,',')) %>%
 unnest(listed_in) %>%
 group_by(listed_in) %>%
 count() %>%
 arrange(desc(n)) %>%
 head(30) %>%
 ggplot() + geom_col(aes(y = reorder(listed_in,n), x = n)) +
 labs(title = 'Themes / Genre of Titles',
     x = '# of Titles',
     y = 'Theme / Genre') +
theme_minimal()
```



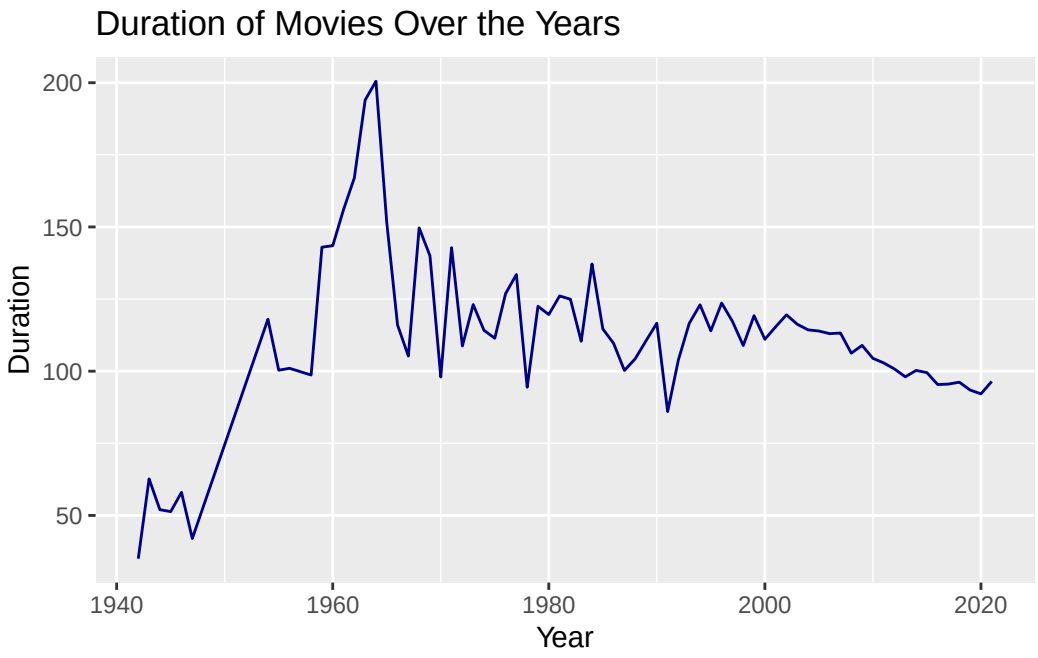

```
netflix_data %>%
 tail(20) %>%
 select('listed_in') %>%
 mutate(listed_in = str_split(listed_in, ',')) %>%
 unnest(listed_in) %>%
```

```
  group_by(listed_in) %>%
  count()
```

```
# A tibble: 21 x 2
# Groups:   listed_in [21]
   listed_in                    n
   <chr>                    <int>
 1 " Comedies"                  3
 2 " Dramas"                    4
 3 " Horror Movies"             1
 4 " Independent Movies"        2
 5 " International Movies"      8
 6 " Kids' TV"                  1
 7 " Korean TV Shows"           1
 8 " Music & Musicals"          1
 9 " Romantic Movies"           1
10 " Romantic TV Shows"         1
# i 11 more rows
```

Show how the time series plot for duration of movies

```
netflix_data$duration<-gsub("min","",as.character(netflix_data$duration))
netflix_data%>%
filter(type == "Movie")%>%
filter(duration != "")%>%
group_by(release_year)%>%
summarize(avg_duration = mean(as.numeric(as.character(duration), na.rm = TRUE)))%>%
ggplot(aes(x=release_year, y = avg_duration)) +geom_line(col = 'dark blue') +
labs(title = 'Duration of Movies Over the Years') +
xlab('Year')+
ylab('Duration')
```



what are Most frequent words in description variable For Movies?

```
library(tidytext)
desc_words_m <- netflix_data %>% select(type, show_id, description) %>%
  filter(type == "Movie") %>%
    unnest_tokens(word, description) %>%
    anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```
count_word <- desc_words_m %>%
   count(word, sort = TRUE)


wordcloud(words = count_word$word,
          freq = count_word$n,
          min.freq = 80,
          max.words = nrow(count_word),
          random.order = FALSE,
          rot.per = 0.1,
          colors = brewer.pal(8, "Dark2"))
```

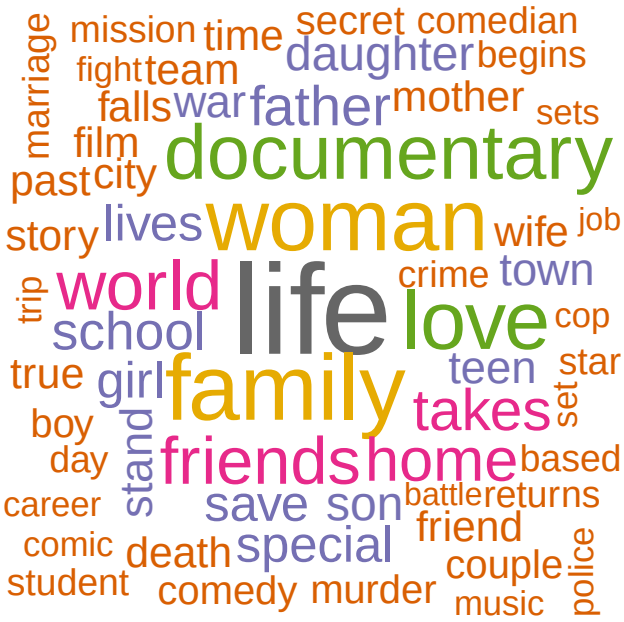
```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : mysterious could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : american could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : brother could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : journey could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : women could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : college could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : relationship could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : meets could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : night could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : childhood could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : drug could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : romance could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : series could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : stories could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : live could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : run could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
80, : village could not be fit on page. It will not be plotted.
```



what are Most frequent words in description variable For shows?

```
desc_words_tv <- netflix_data %>% select(type, show_id, description) %>%
  filter(type == "TV Show") %>%
    unnest_tokens(word, description) %>%
    anti_join(stop_words)
```

```
Joining with `by = join_by(word)`
```

```r
count_word <- desc_words_tv %>%
   count(word, sort = TRUE)


wordcloud(words = count_word$word,
          freq = count_word$n,
          min.freq = 30,
          max.words = nrow(count_word),
          random.order = FALSE,
          rot.per = 0.1,
          colors = brewer.pal(8, "Dark2"))
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : detective could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : daughter could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : journey could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : friendship could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : magical could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : explore could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : solve could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : students could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : returns could not be fit on page. It will not be plotted.
```

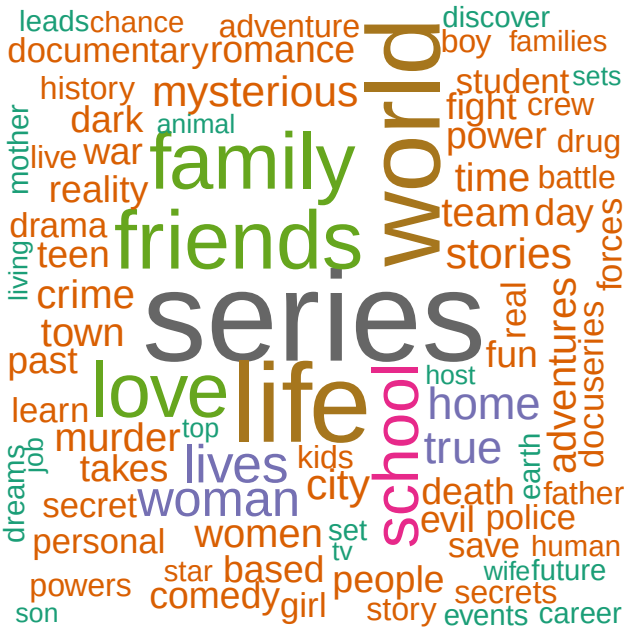
```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : college could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : powerful could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : food could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : game could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : music could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : social could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : special could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : animated could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : challenges could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : join could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : navigate could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : share could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : wealthy could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : competition could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : cop could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : single could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : american could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : business could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : explores could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : survive could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : crimes could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : culture could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : follow could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : husband could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : meet could not be fit on page. It will not be plotted.

Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : mission could not be fit on page. It will not be plotted.
```

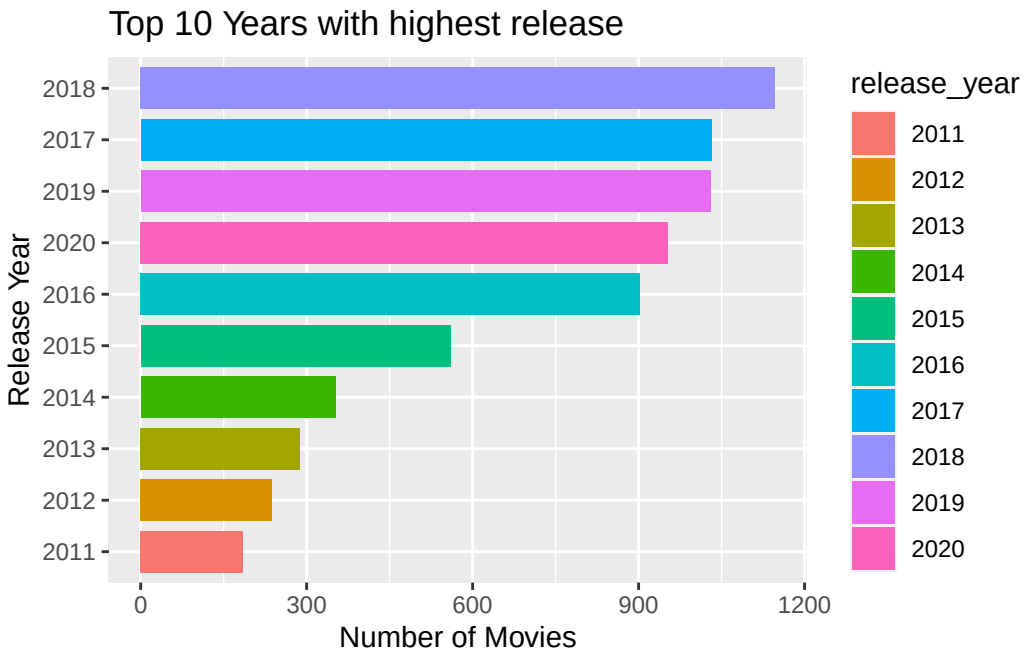
```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : quest could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : struggles could not be fit on page. It will not be plotted.
```

```
Warning in wordcloud(words = count_word$word, freq = count_word$n, min.freq =
30, : win could not be fit on page. It will not be plotted.
```




```
netflix_data %>% #removed 2021 because the year has not yet ended
  filter(release_year != 2021)  %>% #transform the release year into characters
  transform(release_year = as.character(release_year)) %>%
  group_by(release_year) %>%
  summarize(no_of_movies = n()) %>%
  arrange(desc(release_year)) %>%
  head(10) %>% #plot a bar plot of each year against the no of movies released each year
  ggplot(aes(x = reorder(release_year, no_of_movies), y = no_of_movies, fill= release_year
  geom_bar(stat = "identity", width = 0.8) +
  xlab("Release Year") +
  ylab("Number of Movies") +
  ggtitle("Top 10 Years with highest release") +
  coord_flip()
```


Top 10 Years with highest release
Release Year
2018
2017
2019
2020
2016
2015
2014
2013
2012
2011
0
300
600
900
1200
Number of Movies
release_year
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020