IMBD TV Visualization & Sentiment Analysis

Neftali Lemus

2023-01-15

Description: The following visualization project involves the **50,000 IMDB TV and Web Series** Kaggle data set that can be freely downloaded through the link below. This project looks to visualize and understand the distribution trends among IMDB TV's top shows. One of the most popular uses for data of this type are recommendation systems which involve meaningful item or product recommendations to a collection of users.

https://www.kaggle.com/datasets/muralidharbhusal/50000-imdb-tv-and-web-series

1 About the Dataset

1.1 Context

Scrapped from IMDb, the dataset is a collection of top 50,000 TV shows worldwide based on their popularity.

1.2 Content

The data contains 7 columns and 50,000 rows.

1. Series Title : The name of the TV show.
2. Release Year : The Year the show was released in.
3. Runtime : The runtime of single episode of the Show.
4. Genre : The genre of the show.
5. Rating : The rating the specific show has received from users in IMDB.
6. Cast : The leading stars of the show.
7. Synopsis : Background and summary of the story of the show.

1.3 Acknowledgement

The dataset is prepared by scraping the IMDb's website but is not endorsed by IMDb.

2 Raw Dataset

```
setwd("C:/Users/domin/OneDrive/Escritorio/Github Files")
tvseries <- read.csv("TV Series.csv")
library(tidyverse)
library(ggplot2)
dim(tvseries)
```

```
## [1] 50000     7
```

```
str(tvseries)
```

```
## 'data.frame':    50000 obs. of  7 variables:
##  $ Series.Title: chr  "Wednesday" "Yellowstone" "The White Lotus" "1923" ...
##  $ Release.Year: chr  "(2022â\200" )" "(2018â\200" )" "(2021â\200"2023)" "(2022â\200"2023)" ...
##  $ Runtime     : chr  "45 min" "60 min" "60 min" "60 min" ...
##  $ Genre       : chr  "Comedy, Crime, Fantasy" "Drama, Western" "Comedy, Drama" "Drama, Western" ...
##  $ Rating      : chr  "8.2" "8.7" "7.9" "8.6" ...
##  $ Cast        : chr  "Jenna Ortega, Hunter Doohan, Percy Hynes White, Emma Myers" "Kevin Costner, L
##  $ Synopsis    : chr  "Follows Wednesday Addams' years as a student, when she attempts to master her
```

```
sum(is.na(tvseries))
```

```
## [1] 0
```

First Impressions:

- There are no initial NA values.
- There are only character variables.
- Release.year's format is messy.
- Multiple genre's combined as single character strings.
- A proper data transformation is necessary to understand its distributions.

3 Data Cleaning

3.1 Duplicates Check

```
# Series.Title Frequencies
length(which(table(tvseries$Series.Title) > 1))
```

```
## [1] 362
```

The output shows that 362 title entries have a frequency greater than one. We can adress this issue by using the duplicate() function to identify and remove repeated elements in order to avoid inaccurate results. An example of this follows:

```
repeated_instances <- which(tvseries$Series.Title == "Breaking Bad")
breaking_bad_instances <- tvseries[repeated_instances,]
dim(breaking_bad_instances)
```

```
## [1] 801   7
```

Question: Are these 801 instances equal?

```
head(breaking_bad_instances,3)
```

```
##       Series.Title  Release.Year Runtime                  Genre Rating
## 22    Breaking Bad (2008â\200"2013)  49 min Crime, Drama, Thriller    9.5
## 10022 Breaking Bad (2008â\200"2013)  49 min Crime, Drama, Thriller    9.5
## 10072 Breaking Bad (2008â\200"2013)  49 min Crime, Drama, Thriller    9.5
##                                                       Cast
## 22    Bryan Cranston, Aaron Paul, Anna Gunn, Betsy Brandt
## 10022 Bryan Cranston, Aaron Paul, Anna Gunn, Betsy Brandt
## 10072 Bryan Cranston, Aaron Paul, Anna Gunn, Betsy Brandt
##
## 22    A chemistry teacher diagnosed with inoperable lung cancer turns to manufacturing and selling m
## 10022 A chemistry teacher diagnosed with inoperable lung cancer turns to manufacturing and selling m
## 10072 A chemistry teacher diagnosed with inoperable lung cancer turns to manufacturing and selling m
```

Yes, all Breaking Bad entries contain the same information across all column variables.

3.2 Removal of Duplicates

Since duplicates were found in the data, we can proceed with the *duplicate* function which returns a logical vector where TRUE elements correspond to duplicate series titles.

```
# Number of duplicates
sum(duplicated(tvseries$Series.Title))
```

```
## [1] 40356
```

```
# Extracting non duplicates
tvseries2 <- tvseries[!duplicated(tvseries$Series.Title), ]
dim(tvseries2)
```

```
## [1] 9644    7
```

```
sum(which(table(tvseries2$Series.Title) > 1))
```

```
## [1] 0
```

The new IMDB TV data set has shrunk to 9,644 unique observations.

Note: The duplicate() function keeps the first instance of a record and removed the rest, which is why *His Dark Materials* rating of 9.7 was dropped. The first instance for this show had a rating of 7.8

3.3 String Extraction and Data Type Transformations of 'Release.Year', 'Runtime' and 'Rating'

As seen in the beginning, Release.Year has a messy character format that is hard to visualize as some of the shows lack and end date. Many are still in production which is why extracting the release year only is the best way to understand its distribution.

```
head(tvseries2$Release.Year) # Note: Incomplete observations
```

```
## [1] "(2022â\200" )"    "(2018â\200" )"    "(2021â\200"2023)" "(2022â\200"2023)"
## [5] "(2018â\200" )"    "(2022â\200" )"
```

```
# Extracting release year only
tvseries2$Release.Year <- substr(tvseries2$Release.Year, start = 2, stop = 5)
# Coercing character to numeric
tvseries2$Release.Year <- as.numeric(tvseries2$Release.Year)
```

```
## Warning: NAs introduced by coercion
```

Runtime and Rating are also character strings that can be more useful as a numeric variables.

```
#Extracting "Runtime" minutes and coercing them to numeric
tvseries2$Runtime <- as.numeric(substr(tvseries2$Runtime, start = 1 , stop = 3 ))
```

```
## Warning: NAs introduced by coercion
```

```
# Converting Rating to numeric
tvseries2$Rating <- as.numeric(tvseries2$Rating)
```

```
## Warning: NAs introduced by coercion
```

3.4 Transformed Dataset

```
str(tvseries2)
```

```
## 'data.frame':    9644 obs. of  7 variables:
##  $ Series.Title: chr  "Wednesday" "Yellowstone" "The White Lotus" "1923" ...
##  $ Release.Year: num  2022 2018 2021 2022 2018 ...
##  $ Runtime     : num  45 60 60 60 60 40 50 55 NA NA ...
##  $ Genre       : chr  "Comedy, Crime, Fantasy" "Drama, Western" "Comedy, Drama" "Drama, Western" ...
##  $ Rating      : num  8.2 8.7 7.9 8.6 8 8.3 7.7 7.5 5.3 NA ...
##  $ Cast        : chr  "Jenna Ortega, Hunter Doohan, Percy Hynes White, Emma Myers" "Kevin Costner, Lu
##  $ Synopsis    : chr  "Follows Wednesday Addams' years as a student, when she attempts to master her
```

```
dim(tvseries2)
```

```
## [1] 9644    7
```

```
sum(is.na(tvseries2))
```

```
## [1] 2498
```

Notes:

- NA's introduced by coercion, there is no need for imputation/deletion as ggplot and base R have ways to deal with missing values while plotting. We can update missing entries but for the sake of simplicity I will opt to use complete observations only.
- Release.Year, Runtime, and Rating are now numeric.
- The data set has significantly reduced dimensions.
- Since Genre and Synopsis are single character strings with multiple elements, its best to break them down individually and store them in an object for analysis.

3.5 2.5 Subset of Analysis: Top 100 Rated Shows

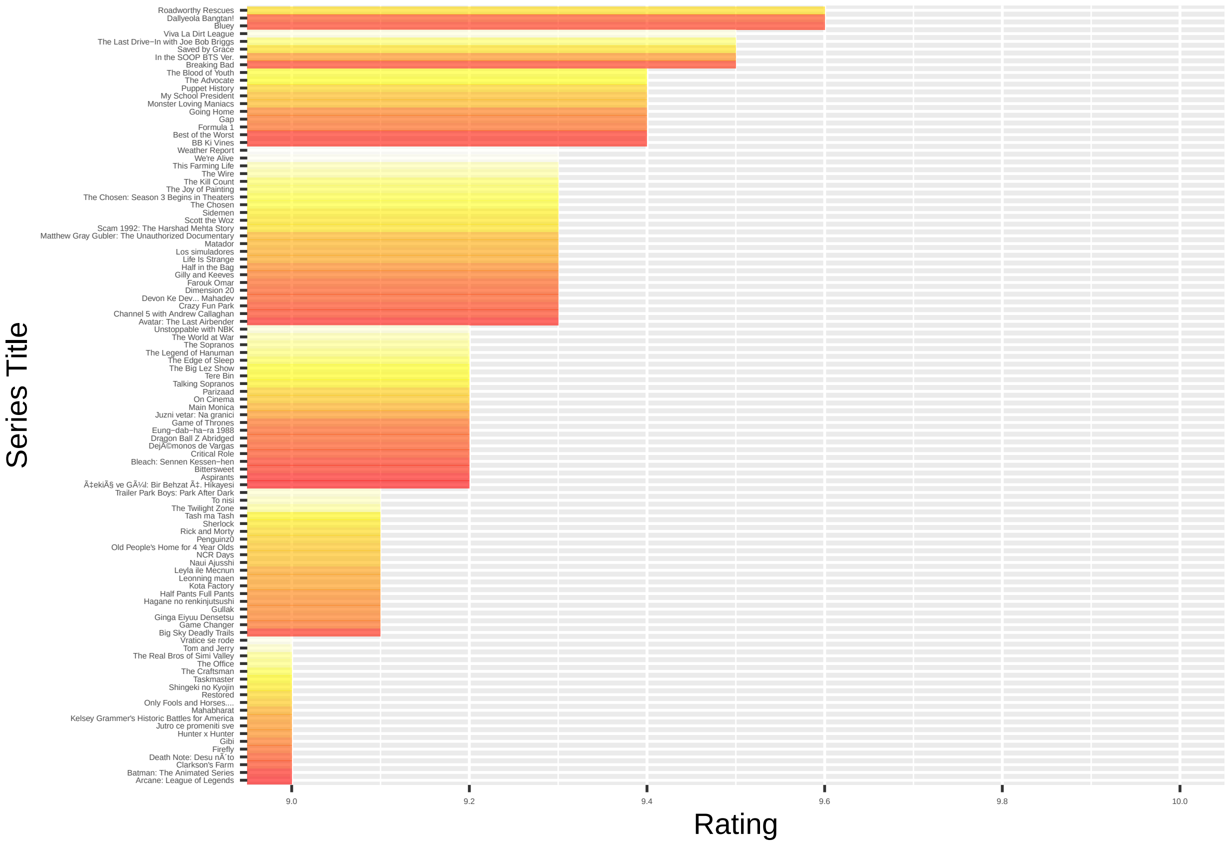
Due to the large amount of titles I have opted to focus on the top 100 shows based on ratings, "top100" will be our last subset fore analysis from here on.

```r
# Lets rank the top 100 by rating
tvseries2r <- tvseries2[order(tvseries2$Rating, decreasing = T),]
top100 <- head(tvseries2r, 100)
```

4 Explanatory Data Analysis

4.1 Ordered Ratings Bar Plot

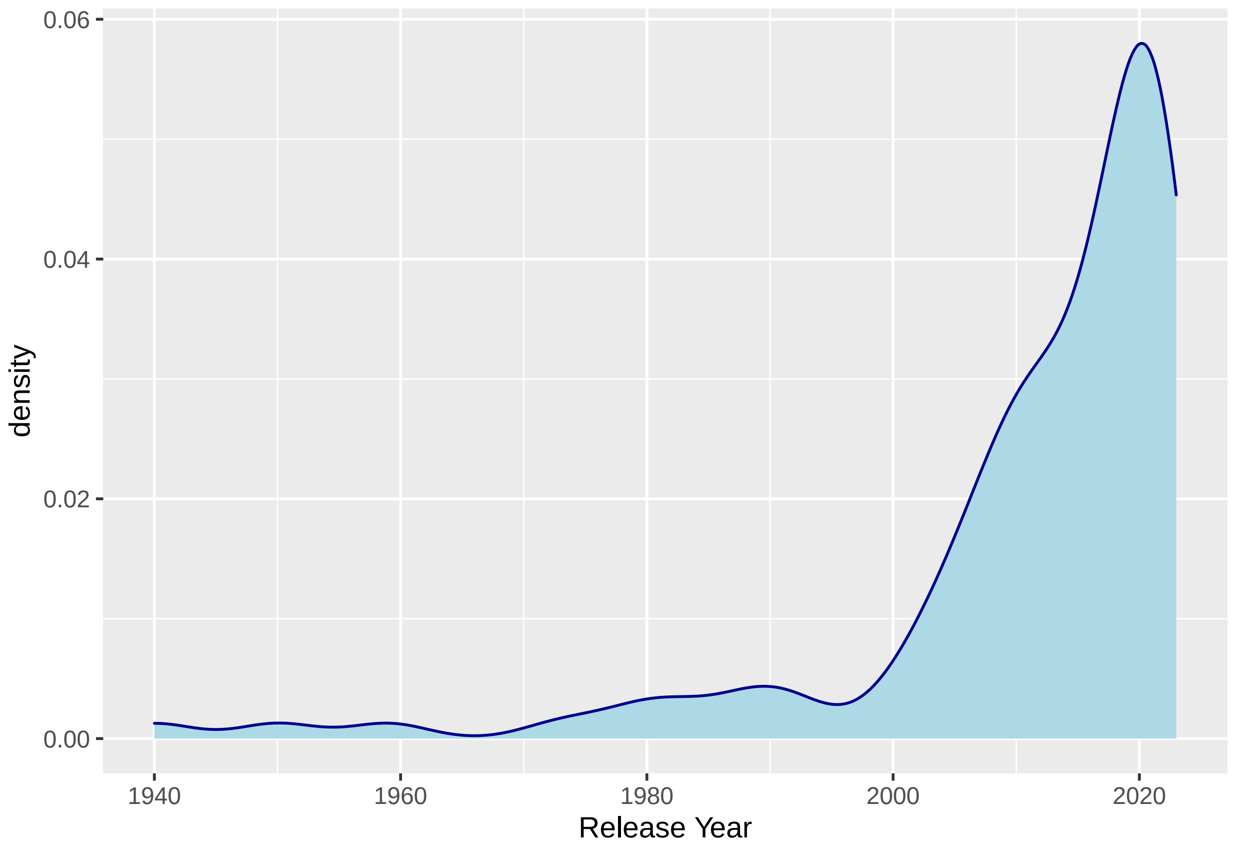
```r
library(forcats)
df_means = top100 %>%
  group_by(Series.Title) %>%
  summarize(Rating = mean(Rating))
df_means %>%
  ggplot(aes(x = fct_reorder(Series.Title, Rating), y = Rating)) +
  geom_col( fill = heat.colors(100, alpha = .6), width = 1) +
  coord_flip(ylim = c(9,10), ) +
  scale_y_continuous(breaks=seq(9,10,0.2)) +
  xlab("Series Title") +
  theme(axis.text = element_text(size = 3))
```



- As we can see by the previous bar graph, IMDB's top 100 series have a rating of 9 or higher.

4.2 Release Year Density Line

```
# Density Line (Release.Year) #Note: NA's removed by default
ggplot(top100, aes(x = Release.Year)) +
  geom_density(color="darkblue", fill="lightblue", na.rm = T) +
  xlab("Release Year")
```



```
# Summary of "Release.Year"
summary(top100$Release.Year)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's 
##    1940    2009    2017    2011    2021    2023       2
```

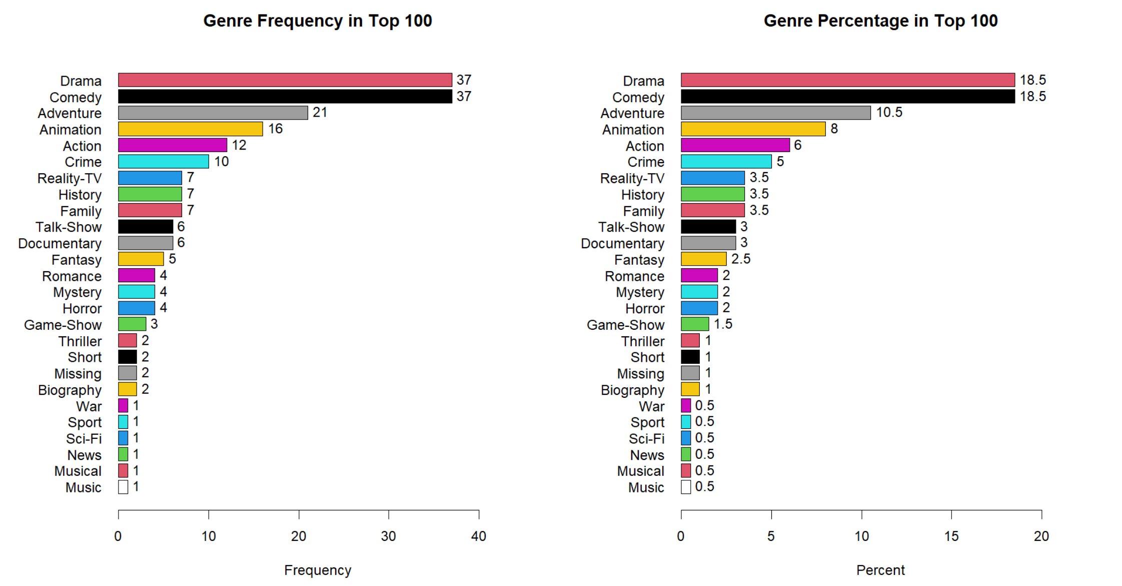
- The density plot shows the distribution of the release year of the top 100 shows. The data is left skewed, meaning that most of the show's release dates concentrate on the far right side of the graph. We can see a sharp increase in shows starting in the 2000's and a slight drop between 2019-2020 which was due to the impact that Covid-19 lock down had in the TV industry.

4.3 Genre Frequency Table

Furthermore, we can also visualize the different genres in 'top100'. The frequency charts were made using the "epiDisplay" package.There are multiple genres listed per 'Series.Title' and its best to string split them in order to have them grouped accordingly.

```
# string splitting and coercing the top100 genres into a factor
genres <- factor(unlist(strsplit(top100$Genre[1:100],split = c(", "))),
                 exclude = c("****")) #Note: "****" missing entries removed
table(genres)
```

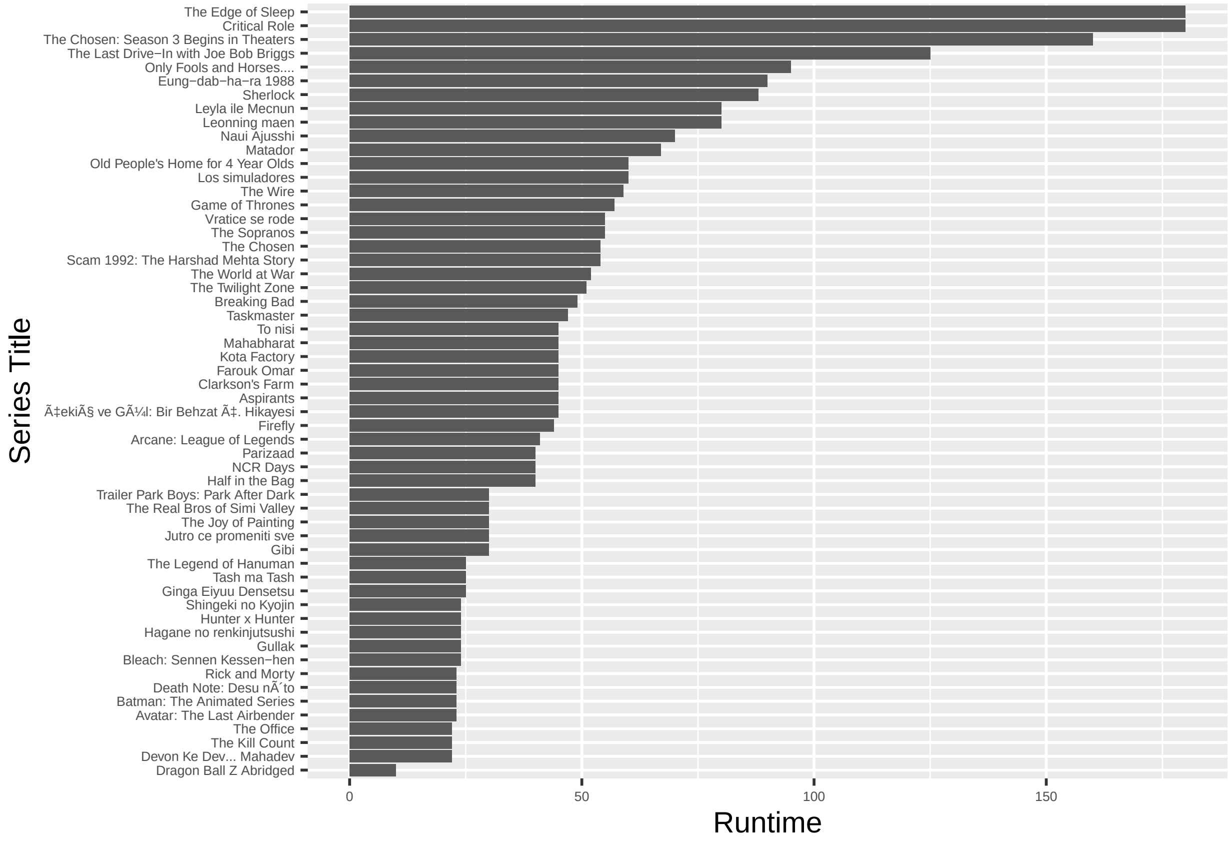
```
## genres
##      Action   Adventure   Animation   Biography      Comedy       Crime
##          12          21          16           2          37          10
## Documentary       Drama      Family     Fantasy   Game-Show     History
##           6          37           7           5           3           7
##      Horror       Music     Musical     Mystery        News  Reality-TV
##           4           1           1           4           1           7
##     Romance      Sci-Fi       Short       Sport   Talk-Show    Thriller
##           4           1           2           1           6           2
##         War
##           1
```



- Drama, Comedy and Adventure are the top three genres mentioned in the ‘Genre’ variable.
- Out of a total of 198 entries, Drama and Comedy are mentioned 37 times each, accounting for an estimate of 18.5% of the total data respectively while Adventure accounts for 10.5% of the data with 21 mentions.

4.4 Runtimes Barchart

```
#Removing NA Running Times
top100_modified <-top100[-which(is.na(top100$Runtime)),]
#Ordering Runtimes plot by lenght
library(forcats)
df_means2 = top100_modified %>%
  group_by(Series.Title) %>%
  summarize(Runtime = mean(Runtime))
df_means2 %>%
  ggplot( aes(y = Runtime, x = fct_reorder(Series.Title, Runtime)) +
  geom_col(na.rm = T) +
  coord_flip() +
  theme(axis.text = element_text(size = 5)) +
  xlab("Series Title")
```



Note : Only complete 'Runtime' observations were used for this plot.

5 Sentiment Analysis

For the next part of my IMDB TV I will conduct a quick sentiment analysis in order to understand the structure of synopsis.

```
library(tidyverse)
library(textdata)
library(tidytext)
library(dplyr)
library(stringr)
library(tibble)
library(ggplot2)
```

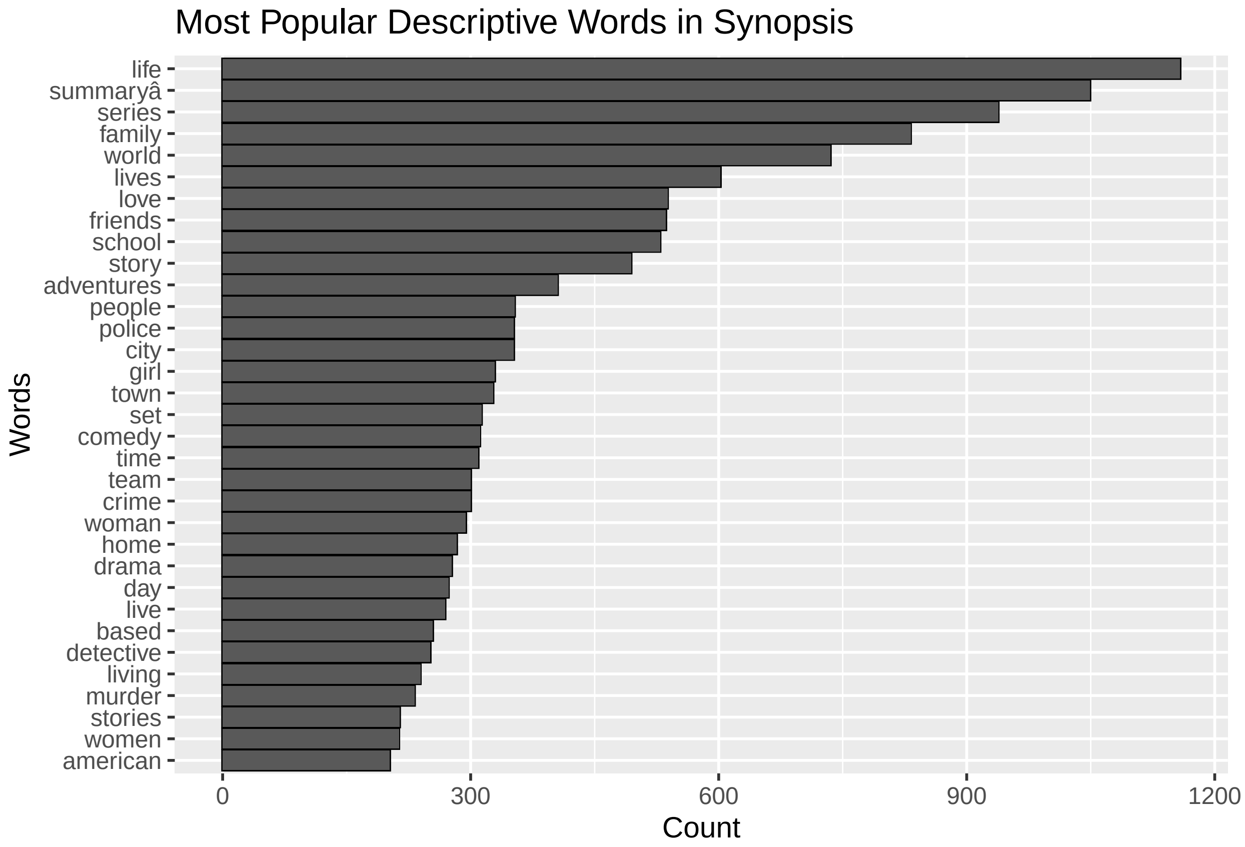
5.1 Frequency of Most Popular Words in Sypnosis

```
# Separate descriptions into words & mutate "word"
data_by_word <- tvseries2r %>%
  mutate(linenumber = row_number()) %>%
  unnest_tokens(word, "Synopsis")
```

```
#Stop Words: This removes meaningless words
#EX.(I, a, me, you, about, to, etc.)
data("stop_words")

#Remove stop words
clean_data <- data_by_word %>%
  anti_join(stop_words)

#Count Most Common Words
a <- clean_data %>%
  count(word,sort = TRUE)
head(a)
```

```
##        word    n
## 1      life 1158
## 2         â 1061
## 3 summaryâ 1049
## 4    series  938
## 5    family  832
## 6     world  735
```

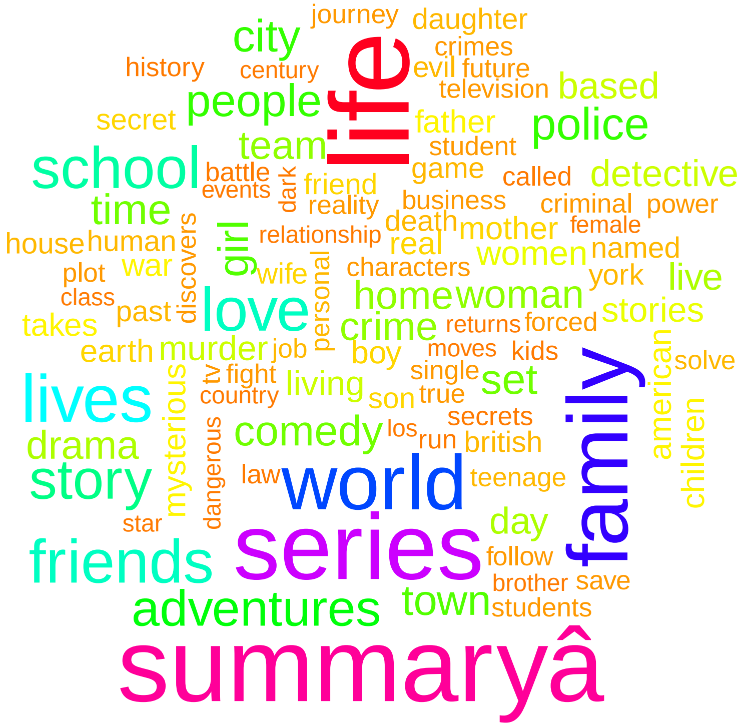
```
#Graph the top 30 words
p <- clean_data %>%
  count(word, sort = TRUE) %>%
  filter(n>200, word != "â") %>%
  mutate(word = reorder(word,n)) %>%
  ggplot(aes(word,n)) +
  geom_col(colour = "black") +
  xlab("Words") +
  ylab("Count") +
  ggtitle("Most Popular Descriptive Words in Synopsis") +
  coord_flip() +
  geom_bar(stat="identity")

p
```



- Life, summary, series, family, world and lives are the top words used in ‘Synopsis’ with over 600 occurrences each.

5.2 Sypnosis Word Cloud

```r
library(wordcloud)
# Word Cloud
library(wordcloud)
clean_data %>%
  filter(word != "â",) %>%
  anti_join(stop_words) %>%
  count(word) %>%
  with(wordcloud(word,n, max.words = 100, scale = c(3.5,0.5), colors = rainbow(50)))
```



- The word cloud shows with greater detail the most popular words employed in 'Sypnosis' where life, family, series and summary are frequently used to describe IMDB TV's highest rated shows.

6 Ideas for Further Analysis

- Merging public reviews with our data set would help understand why these shows are popular.
- Analyse ratings between genres.
- Break down cast and look into the most active actors among the top shows.
- Is there a relationship between a show's cast and its rating?
- Since most of the top shows are recent,do they owe their popularity to the mass media exposure or has the quality of TV shows increased with time?
- What are some of the most important factors that contribute to a show's rating?
- How can a genre recommendation system be built upon the remaining variables?