Analysis of ott platforms

Alekhya | Dipali | Gifty | Megha

03/03/2021

1.OVERVIEW

Hulu is an American subscription video on demand service fully. In 2010, Hulu became the first streaming service to add “Plus” to its name when it launched a subscription service, In 2017, the company launched Hulu with Live TV—an over-the-top IPTV service featuring linear television channels. As of the third quarter of 2020, Hulu had 36.6 million subscribers.

Amazon Prime Video, or simply Prime Video, is an American subscription video on-demand over-the-top streaming and rental service of Amazon.com, Inc., offered as a part of Amazon’s Prime subscription December 14, 2016, Prime Video became worldwide (except for Mainland China, Cuba, Iran, North Korea, Syria).

Netflix, is an American over-the-top content platform and production company headquartered in Los Gatos, California.In January 2021, Netflix reached 203.7 million subscribers.It is available worldwide except in the following: mainland China (due to local restrictions), Syria, North Korea, and Crimea (due to US sanctions)

The Walt Disney Company, commonly known as Disney is an American diversified multinational mass media and entertainment conglomerate headquartered at the Walt Disney Studios complex in Burbank, California.

2.OBJECTIVES

we will be performing the following steps to accomplish the project objectives:

Performing Exploratory Data Analysis and Generating Insights.

1) Visualization of a pie chart for proportion of each genre.
2) Visualizations for no. of movies/shows released by the years released [1990-2000]
3) Visualizations for most rated movies on IMDB based on country.
4) Select the movies with the highest IMDb ratings.
5) Visualize the no of movies based on IMDB.
6) Visualizations for no of movies and ratings based on rotten tomatoes.
7) No of movies present in all OTT platforms (Netflix, prime, Hulu, Disney)
8) Find movies with long runtime in overall.
9) Total movies based on genres and language overall.
10) Find the proportion directors who made most movies.

11) Most rated movies on IMDB based on following languages.
12) Movie Duration in following 12 Countries.
13) To display top 20 movies in Netflix, Hulu, Disney, Prime video.

3.PACKAGES REQUIRED

The following packages have been used for the analysis:

ggplot2: Create Elegant Data Visualization Using the Grammar of Graphics

lazyeval: Lazy (Non-Standard) Evaluation provides a full implementation of LISP style 'quasiquotation', making it easier to generate code with other code.

mosaic: Project MOSAIC Statistics and Mathematics Teaching Utilities.

statisticalModeling: Provides graphics and other functions that evaluate and display models across many different kinds of model architecture.

dplyr: dplyr is a grammar of data manipulation, providing a consistent set of verbs that help you solve the most common data manipulation challenges

tidyverse: The 'tidyverse' is a set of packages that work in harmony because they share common data representations and 'API' design.

readxl: Read Excel Files

treemap: TreeMap Visualization

reshape2: Flexibly Reshape Data

stringi: Character String Processing Facilities

stringr: Simple, Consistent Wrappers for Common String Operations

```
library(ggplot2)
library(lazyeval)
library(mosaic)

## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order
to add
## additional features.  The original behavior of these functions should not
be affected by this.

##
## Attaching package: 'mosaic'
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(statisticalModeling)
```

```
##
## Attaching package: 'statisticalModeling'
```

```
## The following objects are masked from 'package:ggformula':
##
##     gf_abline, gf_bar, gf_boxplot, gf_counts, gf_density,
##     gf_density_2d, gf_frame, gf_freqpoly, gf_hex, gf_histogram,
##     gf_hline, gf_jitter, gf_line, gf_path, gf_point, gf_text
```

```
library(dplyr)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.0 --
```

```
## v tibble  3.0.5     v purrr   0.3.4
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x mosaic::count()            masks dplyr::count()
## x purrr::cross()             masks mosaic::cross()
## x mosaic::do()               masks dplyr::do()
## x tidyr::expand()            masks Matrix::expand()
## x dplyr::filter()            masks stats::filter()
## x ggstance::geom_errorbarh() masks ggplot2::geom_errorbarh()
## x purrr::is_atomic()         masks lazyeval::is_atomic()
## x purrr::is_formula()        masks lazyeval::is_formula()
## x dplyr::lag()               masks stats::lag()
## x tidyr::pack()              masks Matrix::pack()
## x mosaic::stat()             masks ggplot2::stat()
## x mosaic::tally()            masks dplyr::tally()
## x tidyr::unpack()            masks Matrix::unpack()
```

```
library(readxl)
library(treemap)
```

```
## Warning: package 'treemap' was built under R version 4.0.4
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
```

```
## The following object is masked from 'package:tidyr':
##
##     smiths
```

4.LOADING DATASET

```
data <-read.csv("MoviesOnStreamingPlatforms.csv")
colnames(data)
```

```
##  [1] "X"               "ID"              "Title"           "Year"
##  [5] "Age"             "IMDb"            "Rotten.Tomatoes" "Netflix"
##  [9] "Hulu"            "Prime.Video"     "Disney."         "Type"
## [13] "Directors"       "Genres"          "Country"         "Language"
## [17] "Runtime"
```

In the dataset there are 16744 observations of 17 following variables describing the ott platforms and genres:

X: Index value for every movie
ID: Unique ID for every movie
Title: Title of the movie
Year: Actual Release year of the movie
Age: Age restriction for the movie
IMDb: TV Rating of the movie
Rotten. Tomatoes: TV Rating of the movie
Netflix: OTT platform Hulu: OTT platform Prime Video: OTT platform Disney: OTT platform Type: Identifier, Movie Directors: Director of the Movie Genres: Action, Adventure, Sci-Fi, Thriller Country: Country where the movie was produced Language: The Movie language Runtime: Duration of the movie

5.DATA CLEANING

with help of summary would help us spot any anomalies like negative values. It would also indicate the fields with missing values and their counts.

```
summary(data)
```

```
##        X               ID            Title                Year     
##  Min.   :    0   Min.   :    1   Length:16744       Min.   :1902  
##  1st Qu.: 4186   1st Qu.: 4187   Class :character   1st Qu.:2000  
##  Median : 8372   Median : 8372   Mode  :character   Median :2012  
##  Mean   : 8372   Mean   : 8372                      Mean   :2003  
##  3rd Qu.:12557   3rd Qu.:12558                      3rd Qu.:2016  
##  Max.   :16743   Max.   :16744                      Max.   :2020  
##                                                                   
##      Age                 IMDb       Rotten.Tomatoes       Netflix      
##  Length:16744       Min.   :0.000   Length:16744       Min.   :0.0000  
##  Class :character   1st Qu.:5.100   Class :character   1st Qu.:0.0000  
##  Mode  :character   Median :6.100   Mode  :character   Median :0.0000  
##                     Mean   :5.903                      Mean   :0.2126  
##                     3rd Qu.:6.900                      3rd Qu.:0.0000  
##                     Max.   :9.300                      Max.   :1.0000  
##                     NA's   :571                                        
##       Hulu          Prime.Video        Disney.             Type  
##  Min.   :0.00000   Min.   :0.0000   Min.   :0.00000   Min.   :0  
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0  
##  Median :0.00000   Median :1.0000   Median :0.00000   Median :0  
##  Mean   :0.05393   Mean   :0.7378   Mean   :0.03368   Mean   :0  
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.:0.00000   3rd Qu.:0  
##  Max.   :1.00000   Max.   :1.0000   Max.   :1.00000   Max.   :0  
##                                                                  
##   Directors            Genres            Country            Language        
##  Length:16744       Length:16744       Length:16744       Length:16744      
##  Class :character   Class :character   Class :character   Class :character  
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character  
##                                                                             
##                                                                             
##                                                                             
##                                                                             
##     Runtime       
##  Min.   :   1.00  
##  1st Qu.:  82.00  
##  Median :  92.00  
##  Mean   :  93.41  
##  3rd Qu.: 104.00  
##  Max.   :1256.00  
##  NA's   :592      
```

There are NA’s value we are not removing them cause it will impact our analysis.

Deletion of unnecessary columns:

Few of the column like X we wouldn’t be needing for analysis because these contain index values. Lets get rid of the these column.

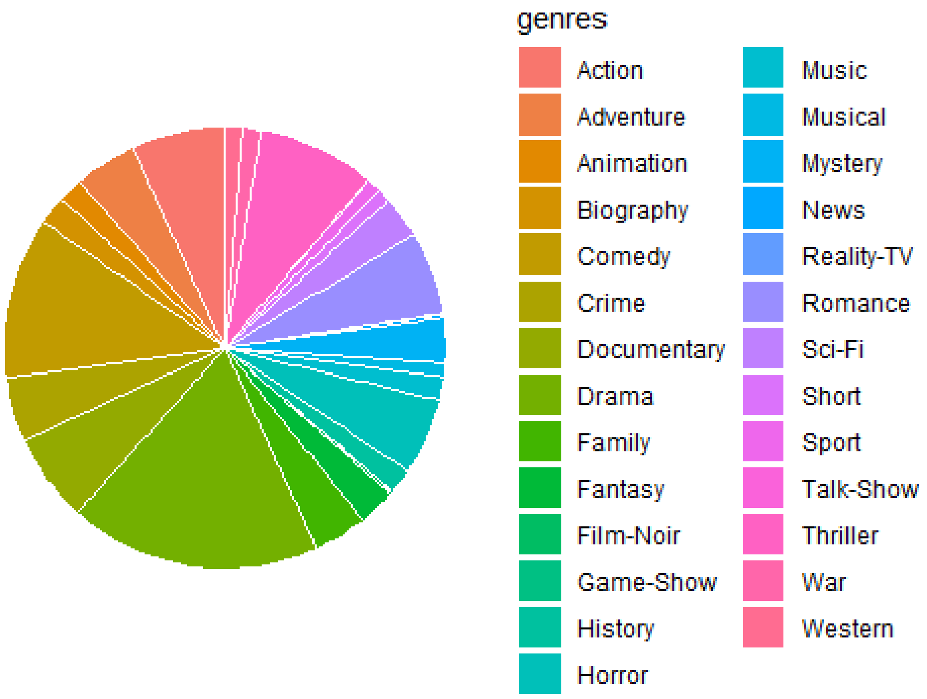
```
data_clean <- data %>% select(-X)
```

Checking final dimensions of cleaned dataset:

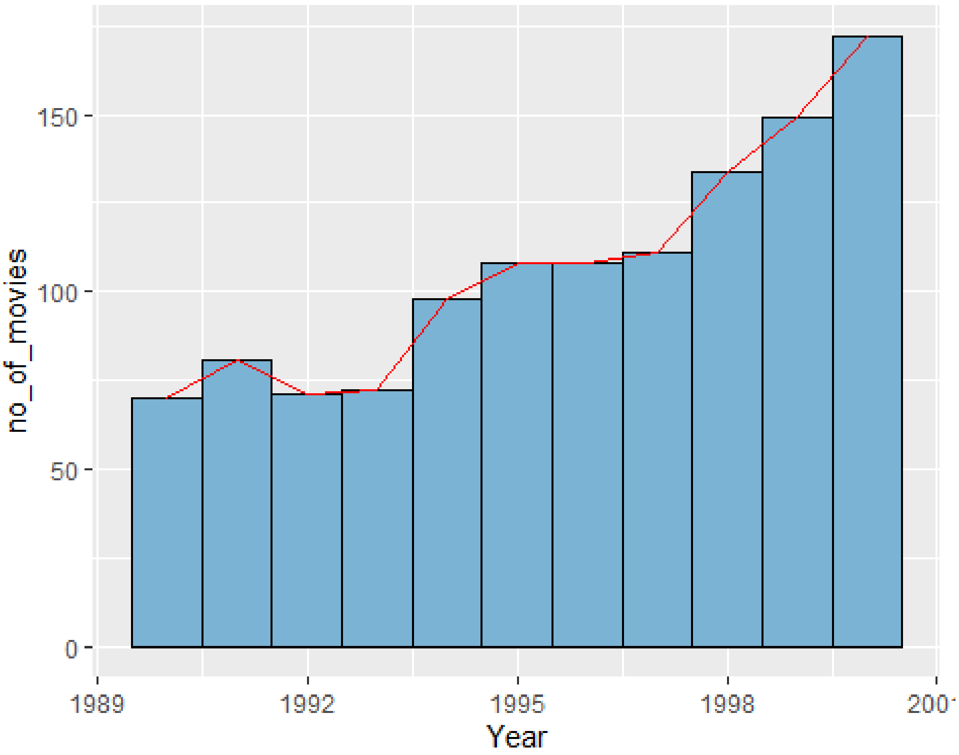
```
dim(data_clean)

## [1] 16744    16
```

6.EXPLORATORY DATA ANALYSIS AND GENERATING INSIGHTS.

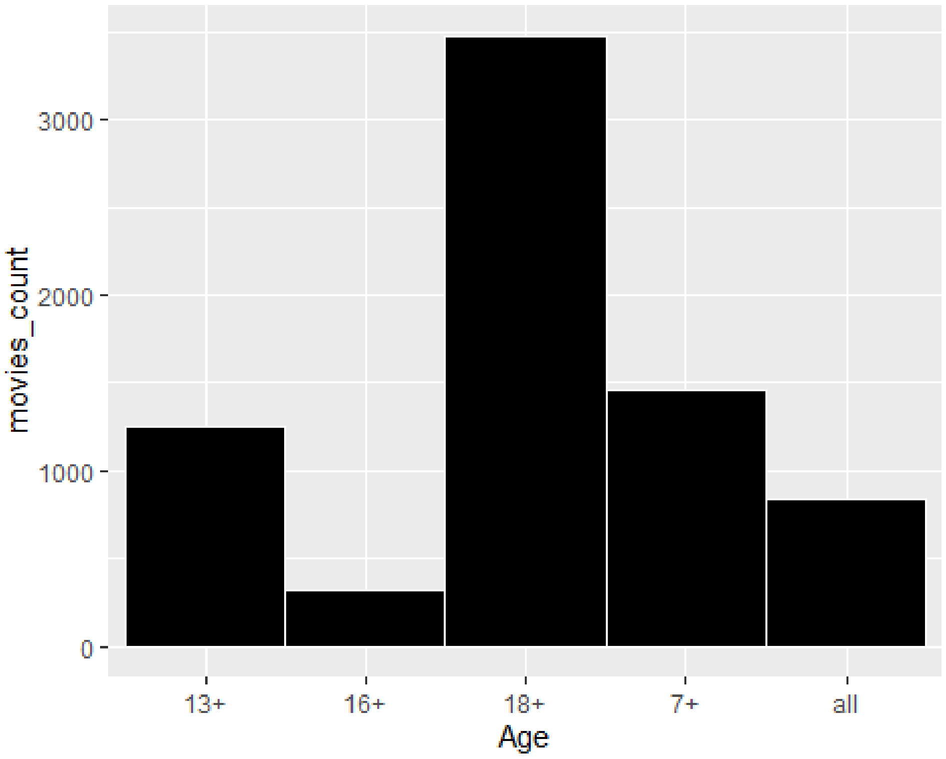
```
# Make a pie chart and show the proportion for each genre

ott <- distinct(data_clean,Title,Country,Year, .keep_all= TRUE)



# the column genre has multiple values against each movie so first we will
count them and make the pie chart
g <- str_split(ott$Genres, ",")
ott_genres <- data.frame(ID = rep(ott$ID, sapply(g, length)), genres =
unlist(g))
ott_genres$genres <- as.character(gsub(",","",ott_genres$genres))

df_by_genres_full <- ott_genres %>% group_by(genres) %>% summarise(count =
n()) %>%
  arrange(desc(count))

# Compute the position of labels
df_by_genres_full <- df_by_genres_full %>%
  arrange(desc(genres)) %>%
  mutate(prop = count / sum(df_by_genres_full$count) *100) %>%
  mutate(ypos = cumsum(prop)- 0.5*prop )
df_by_genres_full <- df_by_genres_full[-nrow(df_by_genres_full), ]

# Basic pie chart using ggplot
ggplot(df_by_genres_full, aes(x="", y=prop, fill=genres)) +
  geom_bar(stat="identity", width=1, color="white") +
  coord_polar("y", start=0) +
  theme_void()
```



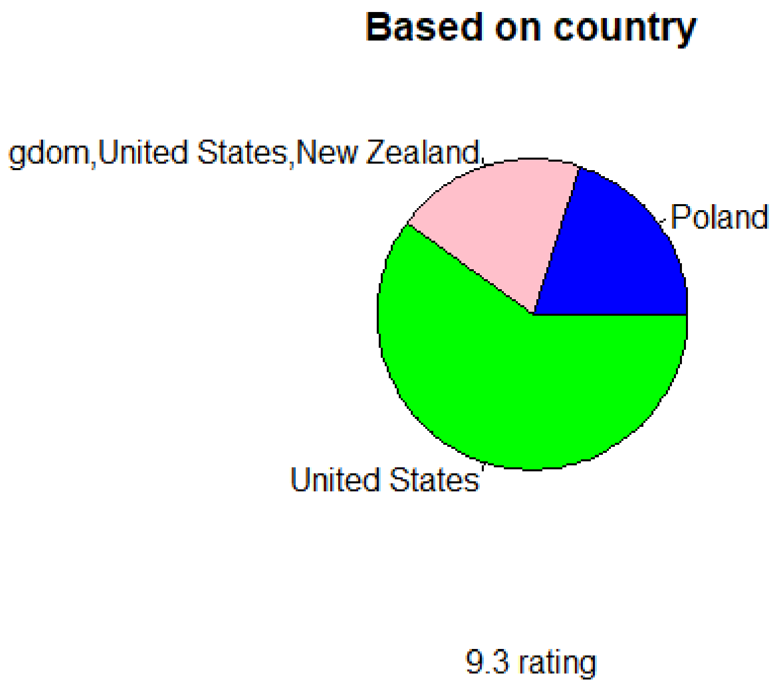
```
#Visualizations for no.of movies/shows released by the years released [1990-
2000]
movies_year <- ott %>% group_by(Year) %>% arrange(desc(Year)) %>%
filter(Year>=1990 & Year<=2000) %>%  summarise(no_of_movies = n())

# visualization using line and bar chart
ggplot(data = movies_year, aes(x=Year, y = no_of_movies)) +
  geom_bar(stat = 'identity', width = 1, color = "black", fill = "#7bb3d4") +
  geom_line(stat = 'identity', color = "red")
```



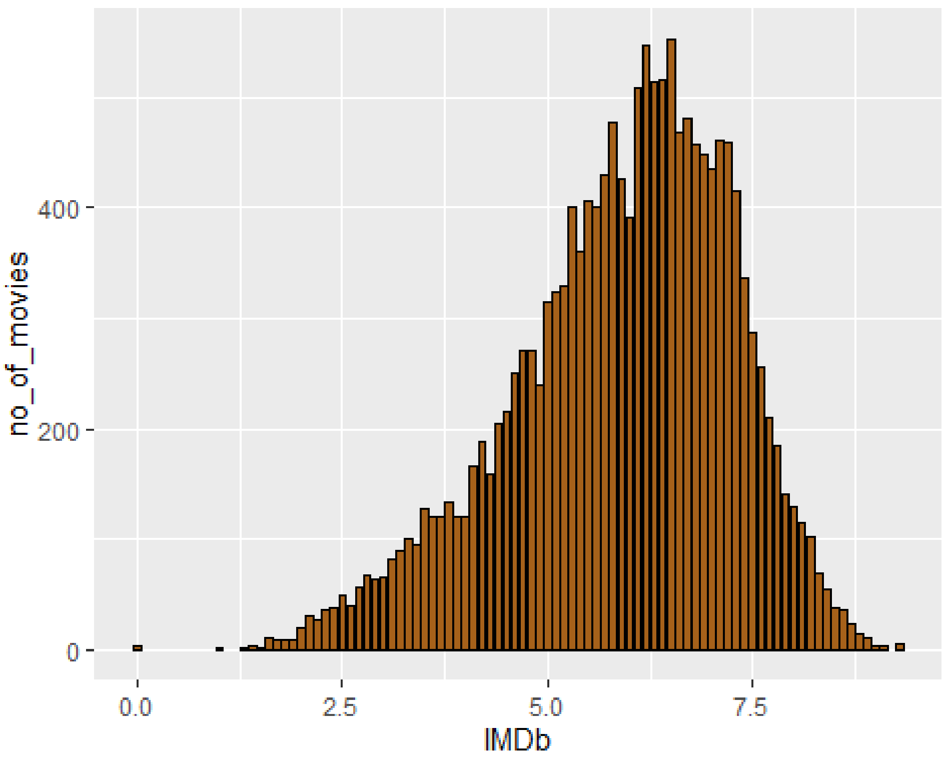
```
#Visualizations for no.of movies released vs age category

movies_age <- ott %>% group_by(Age) %>% arrange(desc(Age)) %>%
summarise(movies_count = n()) %>%
  filter(Age!="")


ggplot(data = movies_age, aes(x=Age, y = movies_count)) +
  geom_bar(stat = 'identity', width = 1, color = "white", fill = "black")
```



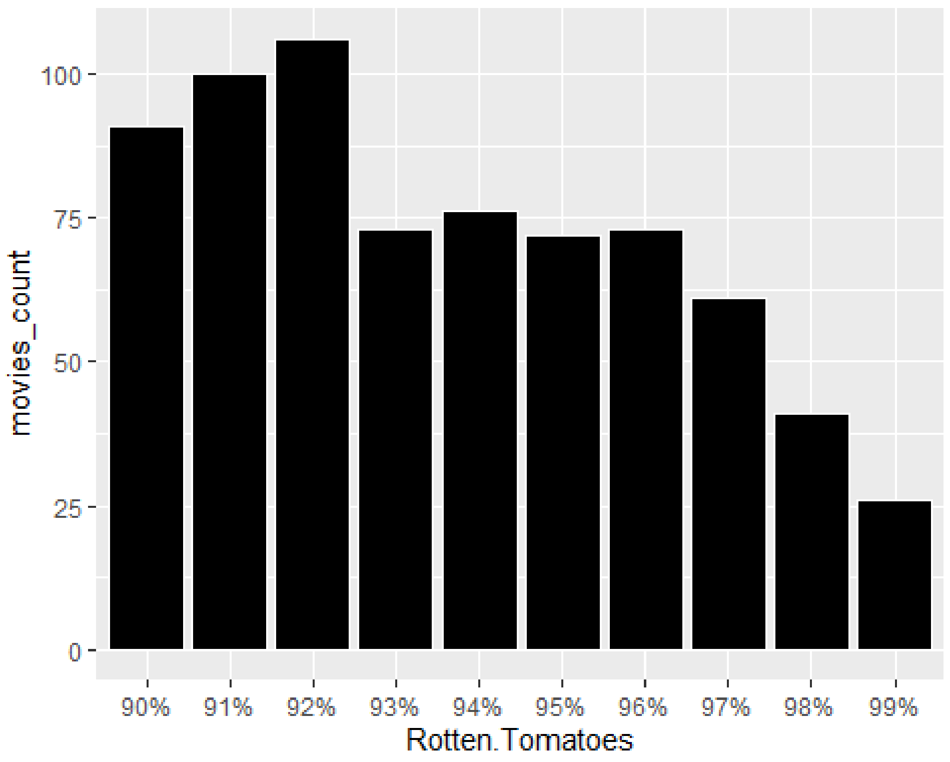
```
#Visualizations for most rated movies on imdb based on country.

d1<- data_clean%>%select(Title,IMDb,Country)%>%filter(Country
!="")%>%slice_max(IMDb,n=1)
d2= data_clean%>%select(Title,IMDb,Country)

pie(xtabs(~d1$Country),main= "Based on country",xlab="9.3 rating",
col=c("blue", "pink", "green", "purple" , "orange"))
```



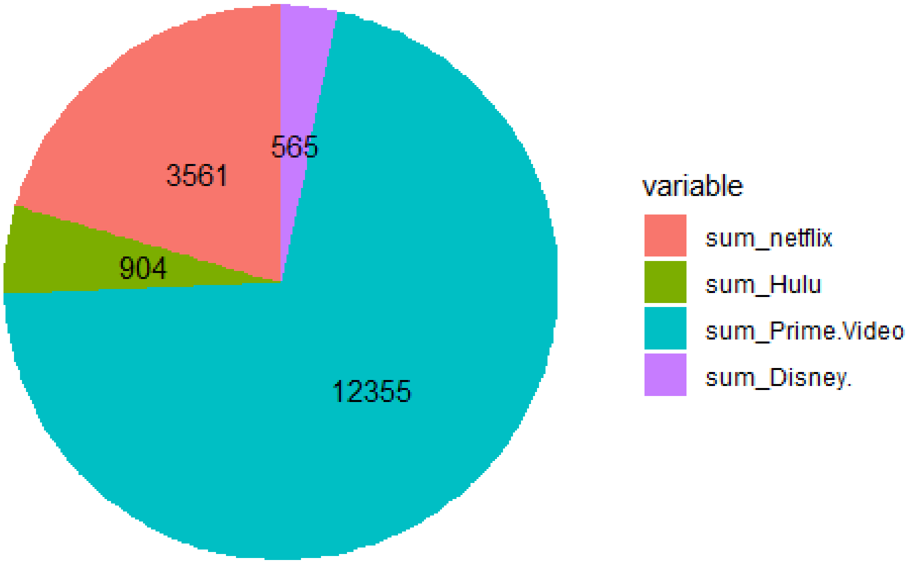
```
#select the movies with the highest IMDb ratings.
movies<-data_clean%>%select(Title,IMDb)%>%slice_max(IMDb,n=1)

# visualize the no of movies based on imdb
movie3 <- data_clean %>% group_by(IMDb) %>% arrange(desc(IMDb)) %>%
summarise(no_of_movies = n())
movie3<- movie3[-nrow(movie3),]

ggplot(data = movie3, aes(x=IMDb, y = no_of_movies)) +
  geom_bar(stat = 'identity', color = "black", fill = "#A6611A")
```



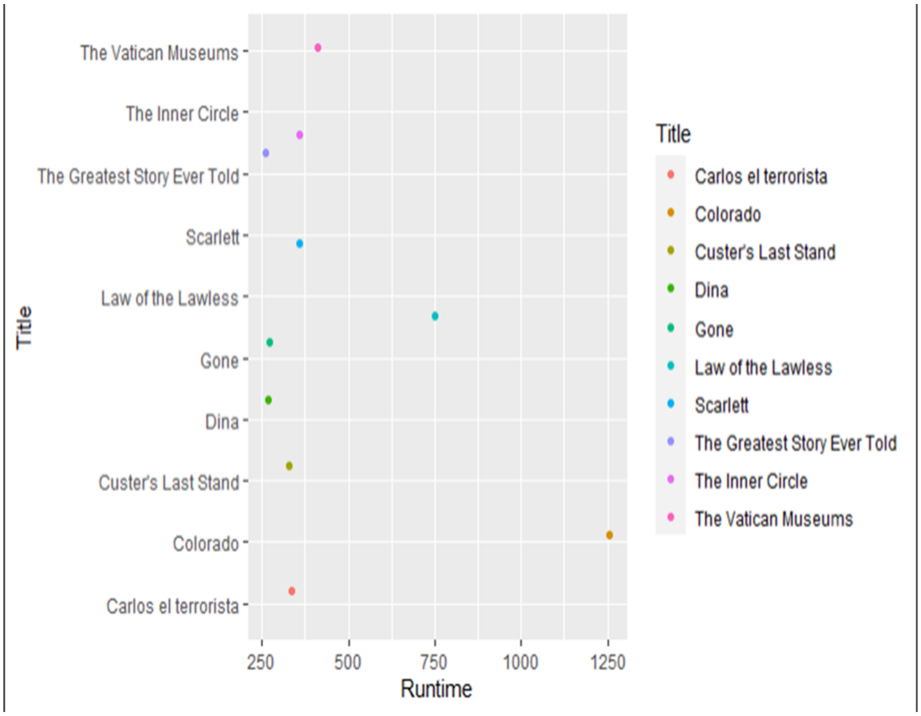
```
#Visualizations for no of movies based on rotten tomatoes
movie1<- data_clean%>%group_by(Rotten.Tomatoes)%>%
summarise(movies_count=n())%>% arrange(desc(Rotten.Tomatoes)) %>% head(10)

ggplot(data=movie1,aes(x=Rotten.Tomatoes,y=movies_count))+ geom_bar(stat =
"identity",color="white",fill="black")
```



```
#No of movies present in all ott platforms (Netflix, prime, hulu, Disney)

movie_data <- data_clean  %>% select(Title,Netflix,Hulu,Prime.Video,Disney.)
%>%
  summarise(sum_netflix = sum(Netflix,na.rn=TRUE),
            sum_Hulu = sum(Hulu,na.rn=TRUE),
            sum_Prime.Video = sum(Prime.Video,na.rn=TRUE),
            sum_Disney. = sum(Disney.,na.rn=TRUE))


#melt is used to change columns to rows vice-verse
molted=melt(movie_data)

## No id variables; using all as measure variables

ggplot(molted, aes(x="", y=value, fill=variable)) +
  geom_bar(width = 3, stat = "identity") +
  coord_polar("y", start=0) +
  geom_text(aes(label=value),position=position_stack(vjust = 0.5))+
  theme_void()
```

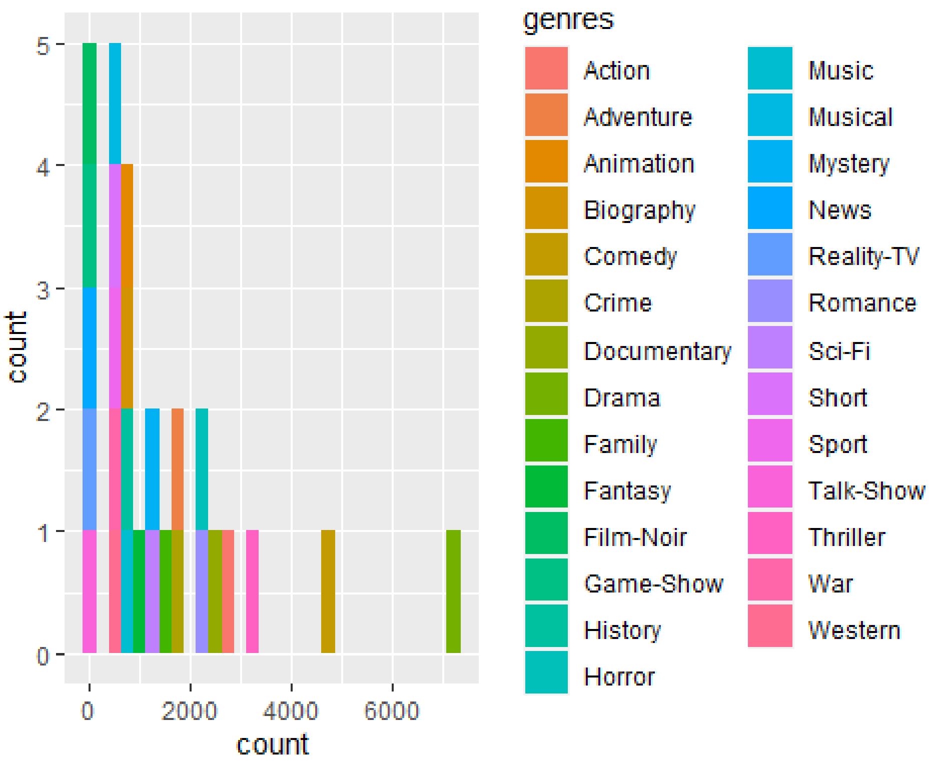


```
#Find movies with long runtime in overall.

movie_runtime <-data_clean %>%
  select(Title,Runtime) %>%
  arrange(desc(Runtime)) %>% head(10)

ggplot(data=movie_runtime,aes(x=Runtime,y=Title,col=Title))+geom_jitter()
```

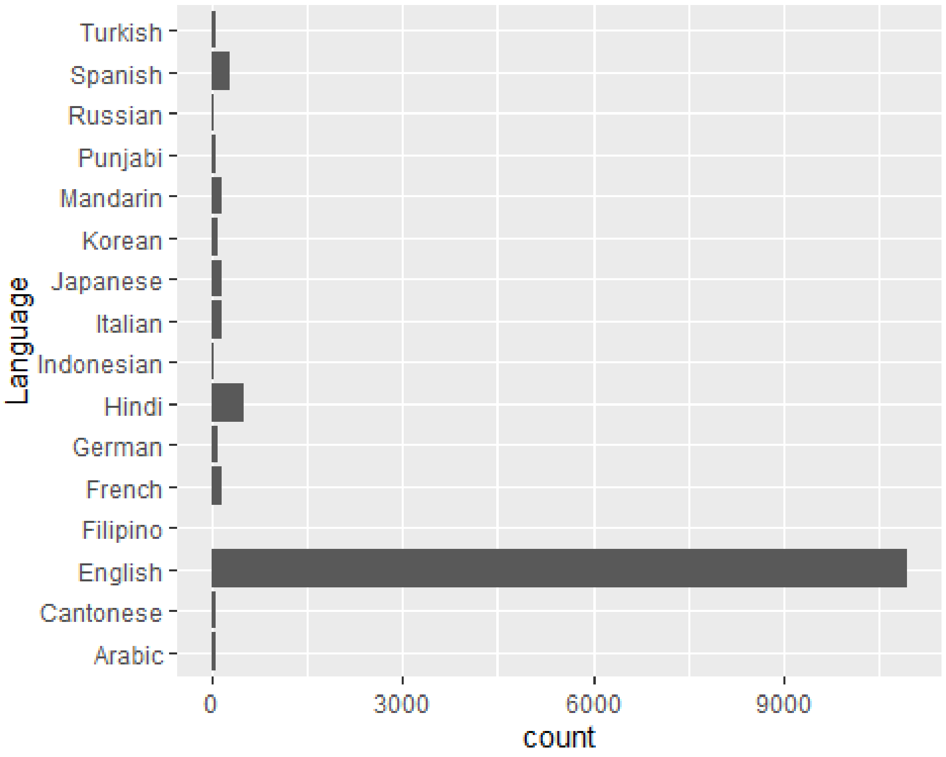


```
# Total movies based on genre:
ott <- distinct(data_clean,Title,Genres,Language, .keep_all= TRUE)
g <- str_split(ott$Genres, ",")
ott_genres <- data.frame(ID = rep(ott$ID, sapply(g, length)), genres =
unlist(g))
ott_genres $ genres <- as.character(gsub(",","",ott_genres$genres))
df_by_genres_full <- ott_genres %>% group_by(genres) %>% summarise(count =
n()) %>% arrange(desc(count)) %>% filter(genres != "")
ggplot(data = df_by_genres_full,aes(x=count,fill=genres)) +geom_histogram()
```

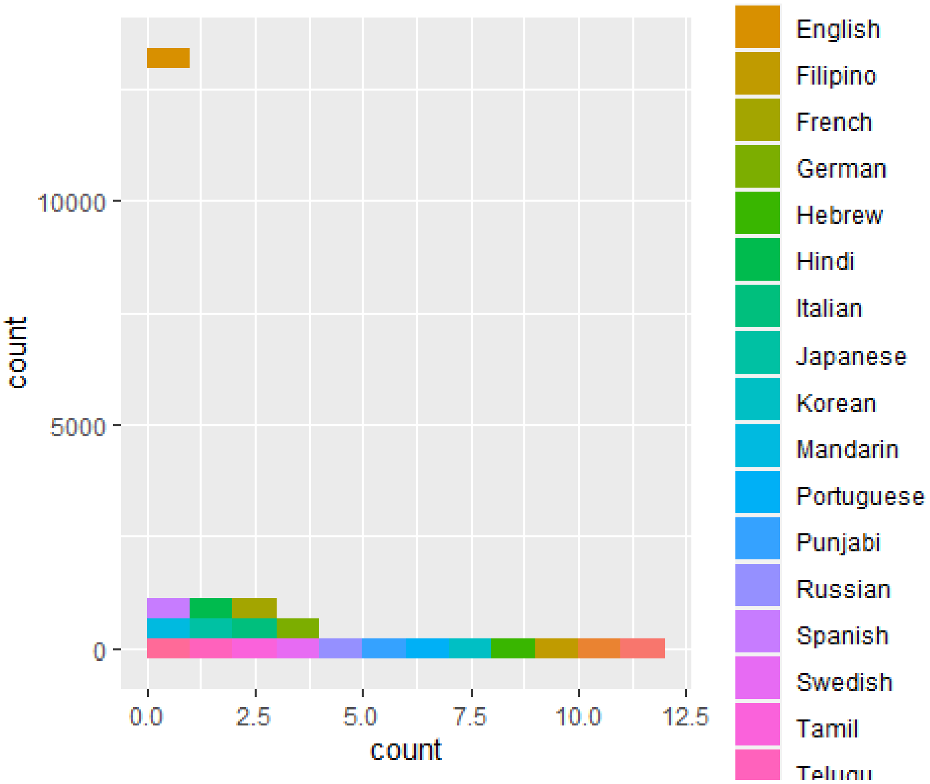
```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



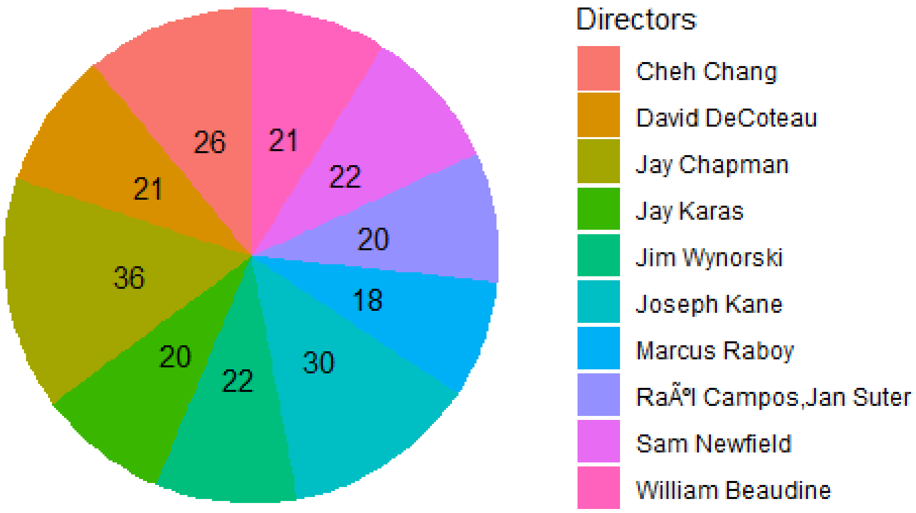
```
# Total number of movies based on languages:-

# We can either print some particular languages:
Total_movies <- data_clean %>% select(Title,Genres,Language)
tmovie_subset <- Total_movies[Total_movies$Language %in%
                                c("English", "Hindi","Spanish" ,"French"
,"Others",
                                  "German","Japanese","Arabic"  ,
"Mandarin","Italian",
                                  "Turkish", "Cantonese","Russian","Tamil
",
                                  "Punjabi"," Portuguese","Indonesian","
Malayalam",
                                  "Filipino","Korean"),]

ggplot(data = tmovie_subset,aes(y=Language)) +geom_bar()
```



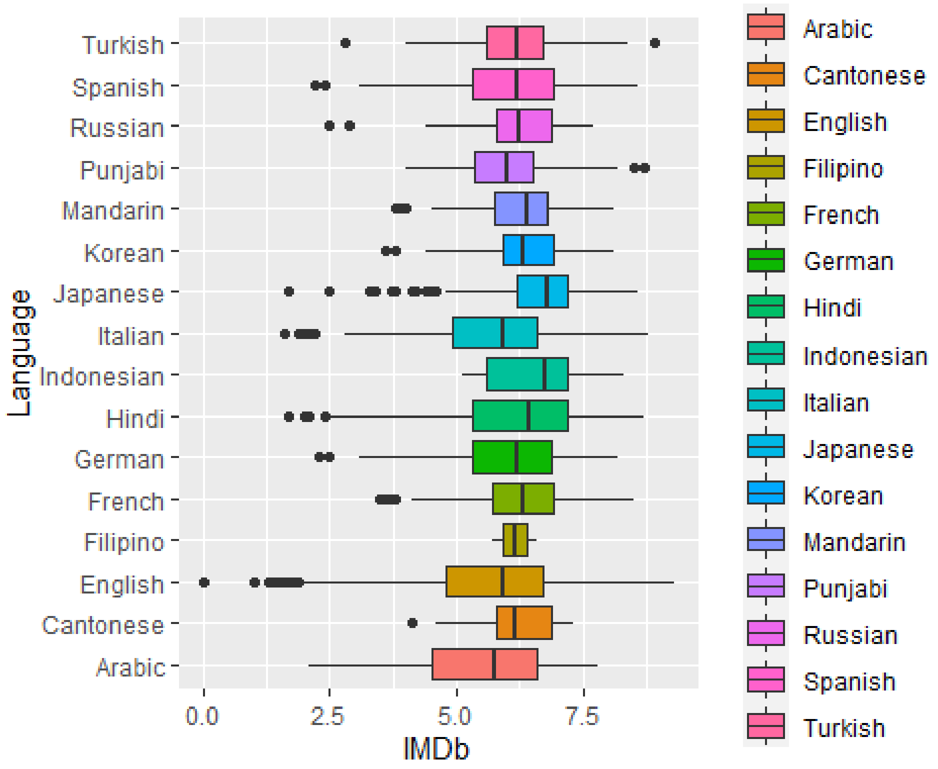
```
# or we can print all:
l<- str_split(ott$Language,",")
ott_language <- data.frame(ID = rep(ott$ID, sapply(l, length)), language =
unlist(l))
ott_language$language <- as.character(gsub(",","",ott_language$language))
df_by_language_full <- ott_language %>% group_by(language) %>%
summarise(count = n()) %>%
  arrange(desc(count)) %>% filter(language != "")

f2 <-head(df_by_language_full,20)
ggplot(data = f2,aes(y=count,fill=language) )+geom_histogram()

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



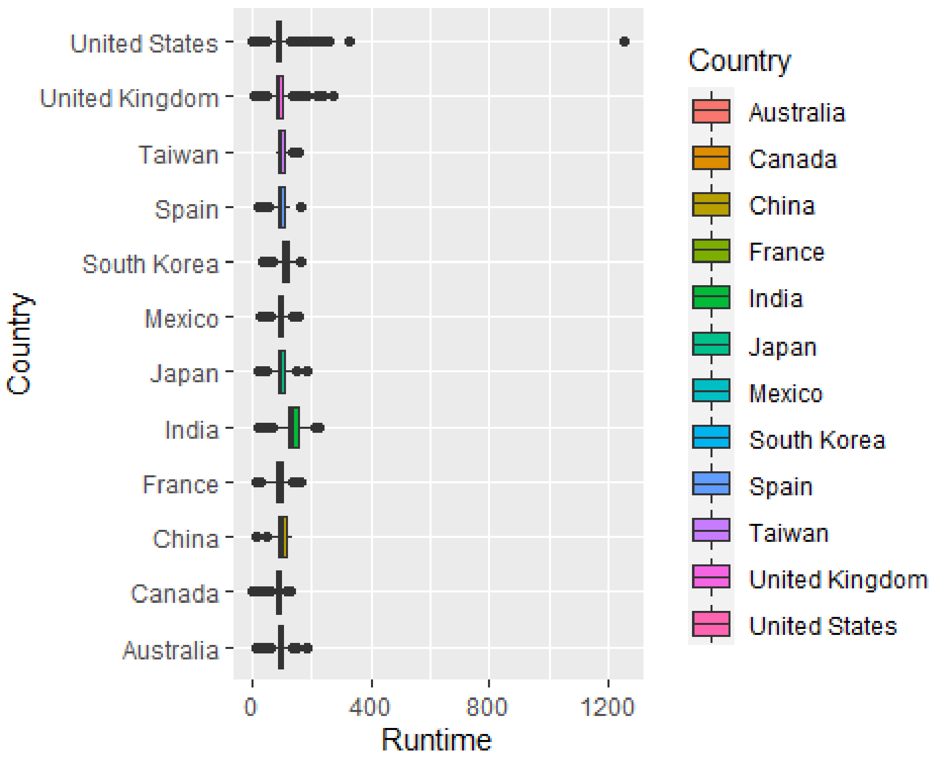
```r
#Find the proportion directors who made most movies

proportion_of_directions <- data_clean %>% group_by(Directors)  %>%
summarise(movie_count = n()) %>%
  arrange(desc(movie_count)) %>% filter(Directors!="")

f1 <- head(proportion_of_directions,10)


#visualization

ggplot(f1, aes(x="", y=movie_count, fill=Directors)) +
  geom_bar(width = 3, stat = "identity") +
  coord_polar("y", start=0) +
  geom_text(aes(label=movie_count),position=position_stack(vjust = 0.5))+
  theme_void()
```



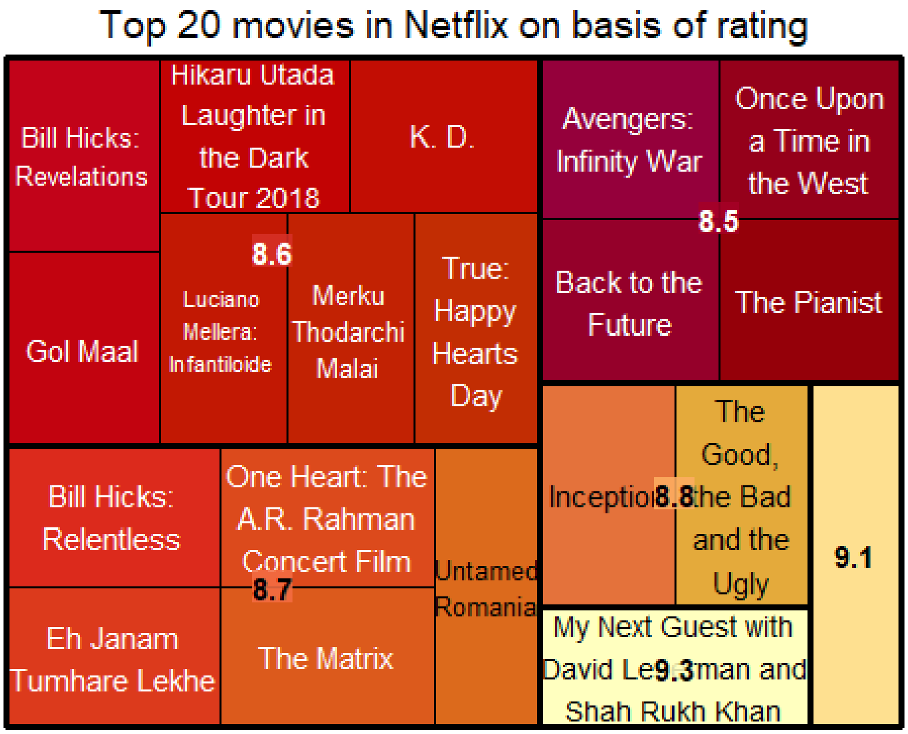
```
#Most rated movies on imdb based on following languages
language_rating <- data_clean %>%
  select(Language,Title,IMDb) %>%
  filter(Language !="", IMDb !="",Title != "") %>%
   arrange(desc(IMDb))

language_subset <- language_rating[language_rating$Language %in%
                                     c("English", "Hindi","Spanish" ,"French"
,"Others",
                                       "German","Japanese","Arabic" ,
"Mandarin","Italian",
                                       "Turkish",
"Cantonese","Russian","Tamil  ",
                                       "Punjabi"," Portuguese","Indonesian","
Malayalam",
                                       "Filipino","Korean"),]

ggplot(data = language_subset,aes(y=Language,x=IMDb,fill=Language))
+geom_boxplot()
```



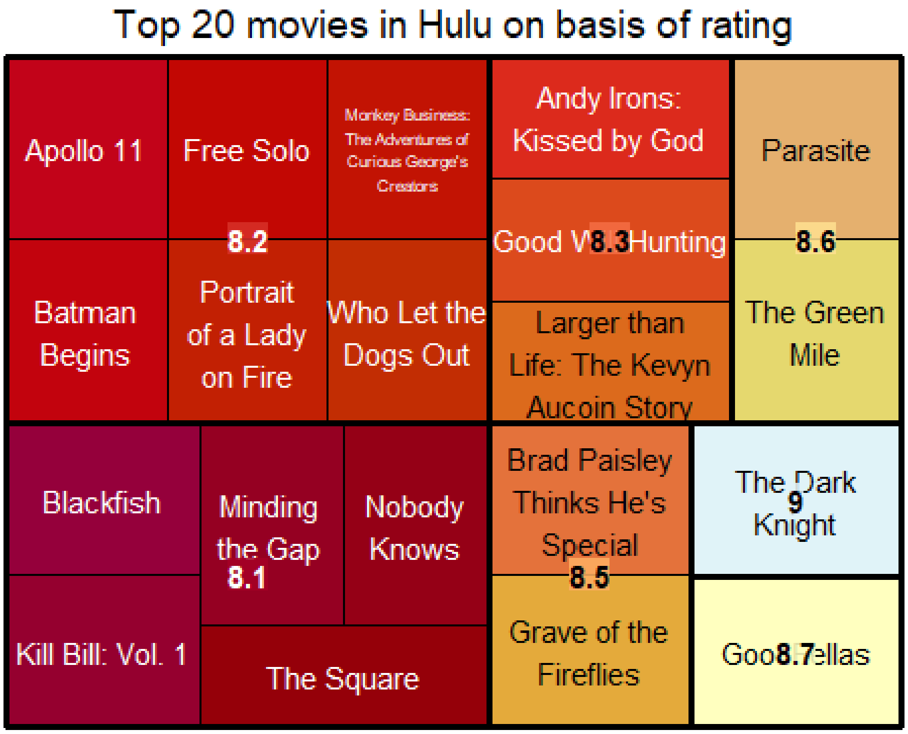
```
#Movie Duration in following 12 Countries.


movie_d <- data_clean %>%
  select(Title,Country,Runtime) %>%
  arrange(desc(Runtime))

movie_duration <- na.omit(movie_d)

duration_subset<- movie_duration[movie_duration$Country %in%
                                        c("United States", "India", "United
Kingdom",
                                          "Canada", "France", "Japan",
"Spain", "South Korea",
                                          "Mexico", "Australia", "China",
"Taiwan"),]


ggplot(data = duration_subset,aes(y=Country,x=Runtime,fill=Country))
+geom_boxplot()
```



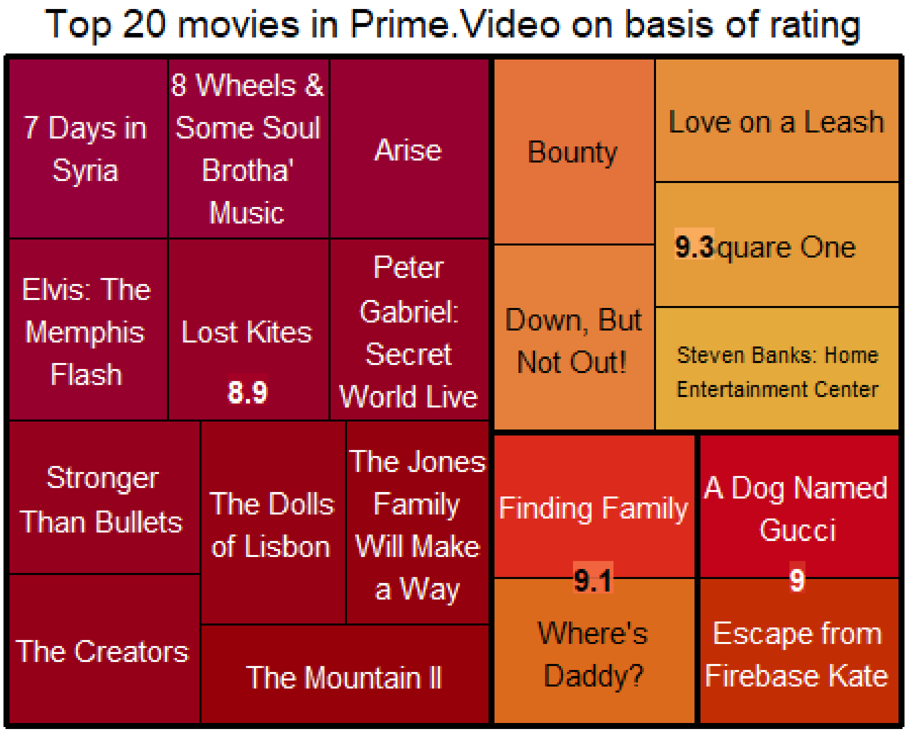
```
#count of movies based on countries
c<- str_split(ott$Country, ",")
ott_country <- data.frame(ID = rep(ott$ID, sapply(c, length)), Country =
unlist(c))
ott_country$Country <- as.character(gsub(",","",ott_country$Country))

c_tab <- ott_country %>% group_by(Country) %>% summarise(count = n()) %>%
  arrange(desc(count)) %>% filter(Country != "")

#To display top 20 movies in netflix,hulu,disney,primevideo



display_movies <- data_clean %>%
select(Title,IMDb,Genres,Netflix,Hulu,Prime.Video,Disney.) %>%
  arrange(desc(IMDb))



netflix_movie <- display_movies %>% filter(Netflix == 1) %>% head(20)
treemap(netflix_movie, index = c("IMDb", "Title"), vSize = "IMDb",palette =
"RdYlBu" , title="Top 20 movies in Netflix on basis of rating")
```


Top 20 movies in Netflix on basis of rating

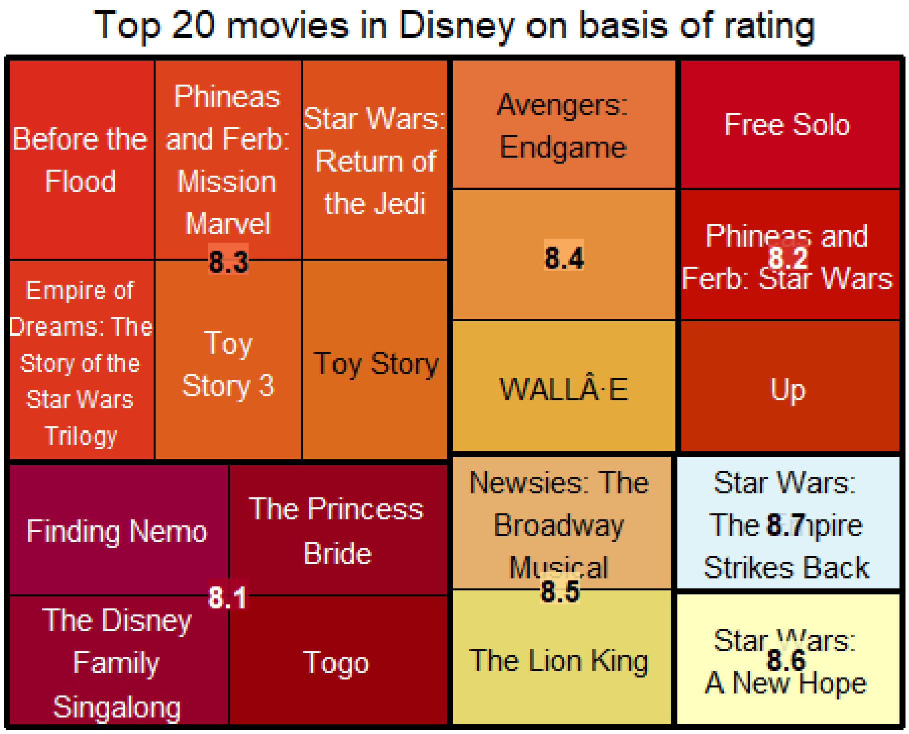
```
hulu_movie<-display_movies %>% filter(Hulu==1)%>% head(20)
treemap(hulu_movie, index = c("IMDb", "Title"), vSize = "IMDb",palette =
"RdYlBu" , title="Top 20 movies in Hulu on basis of rating")
```


Top 20 movies in Hulu on basis of rating

```
prime_movie<- display_movies %>% filter(Prime.Video == 1)%>% head(20)
treemap(prime_movie, index = c("IMDb", "Title"), vSize = "IMDb",palette =
"RdYlBu" , title="Top 20 movies in Prime.Video on basis of rating")
```



```
disney_movie <- display_movies %>% filter(Disney.==1)%>% head(20)
treemap(disney_movie, index = c("IMDb", "Title"), vSize = "IMDb",palette =
"RdYlBu" , title="Top 20 movies in Disney on basis of rating")
```



7.SUMMARY

7.1. Problem Statement

```
    The analysis was intended to understand the evolution of ott platforms
and characterisitcs.
    To analyze the highest movie rating ,ott rating depending upon the
geners,country and language.
```

7.2. Methodology

```
   * Finding the proportion of geners followed by number of movies relased in
particular year.
   * Analysis of IMDb rating of movies based on country ,language and geners.
   * Similarly, we have analyzed rotten tomaotes rating.
   * This was followed by finding proportions of directors and overall based
ratings.
```

7.3. Insights

```
   * Drama has the highest proportion for genre.
   * No.of movies released by the years released [1990-2000] there was growth
in the nuber of movies.
```

```
   * Age 18+ has the highest count of relased movies followed by the highest
imbd rating based on
     country in which is 9.3 is the highest rating with Poland,UN,NewZealand.
   * Visualization of no of movies based on imdb and rotten tomatoes.
   * Prime.Video has the highest no of movies compared to netflix,hulu and
disney and visualizing
     count of movies based on geners in which drama is highest and english
languages is the highest.
   * Jay Chapman has made most movies and most rated movies on imdb based on
languages is
     Square One with 9.3 rating which is english.
   * Colorado movie has the highest runtime 1256 which is from United States.
   * To display top 20 movies in all platforms on basis of ratings.
```

7.4. Limitations

```
  * Even though there are millions of movies that exist on various platforms,
we only had about             16744 data size for our analysis, and hence we
couldn’t obtain a full picture of the features of         movies on all
platforms.
  * Also, the analysis could be strengthened by incorporating user related
features like
    their demographical attributes, user history etc.
```

Reference links:

https://www.kaggle.com/siddharth2000/simple-seaborn-plots
https://www.kaggle.com/rickyrick/ott-platforms-movie-analysis
https://rpubs.com/phone_thit_htun/netflix_dataviz
https://rpubs.com/bhasinrl/spotify__data__analysis