Exploratory Data Analysis of Youtube Statistics

Submitted in partial fulfillment of the requirements

of the degree of

Bachelor of Engineering (B.E.)

By

Aayushi Gandhi - 60003170002
Abheet Shaju - 60003170003
Ansh Shah - 60003170008

Project Guide

Prof. Purva Raut

1. Problem Definition

Video has become an integral part of Internet users' online experience, and no site hosts more videos than YouTube. The site boasts more than 3 billion video views per day, making it an excellent place for marketers to find consumers. Its reach is global, too; 70 percent of its traffic originates outside of the United States, making it more than a place to find only American customers.YouTube is the home of the viral video, the term for a video that spreads quickly to a large audience on the Internet. Marketers can use other social media avenues, such as Facebook or Twitter, to direct consumers to YouTube videos as a way of trying to get the videos to catch fire with the public. Especially popular videos make it to the YouTube homepage, further strengthening traffic.

In addition to brand channels, YouTube offers other options for marketers attempting to reach the site's many users. These opportunities include buying advertising on the home page or buying advertising that runs beside certain videos. Marketers can also use keywords so that, when certain search terms are used on YouTube, it will trigger their videos to appear as an option for users. Marketers then pay based on how many users choose to watch their video.

Youtube, has become a major platform for organisations to market their products and services. However, before entering any market, a thorough market analysis should be conducted. We here are analysing the statistics of videos that are trending on Youtube to get an idea about the types of content that users like to view.

2. Motivation:

Videos provide a more engaging way to tell a brand story and connect with the audience. Utilizing YouTube to promote business comes with tons of benefits. Users are likely to reach new audiences, and since the service is free, it's a budget-friendly way to expand reach.

Possible uses for this dataset could include:

- Sentiment analysis in a variety of forms
- Categorising YouTube videos based on their comments and statistics.
- Training ML algorithms like RNNs to generate their own YouTube comments.
- Analysing what factors affect how popular a YouTube video will be.
- Statistical analysis over time.

3. Description of the Dataset

This dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the US, GB, DE, CA, FR and IN regions (USA, Great Britain, Denmark, Canada, France, and India respectively), with up to 200 listed trending videos per day.

It also includes data from RU, MX, KR and JP regions (Russia, Mexico, South Korea, and Japan respectively) over the same time period.

Each region's data is in a separate file. Data includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count.

The data also includes a category_id field, which varies between regions. Dataset can be accessed from : https://www.kaggle.com/datasnaek/youtube-new

4. Pre-processing of Data

The dataset was divided into four parts, each containing the data from different regions. The four datasets had to be merged to form one single dataset. Post this, a few columns in the dataset were reformatted for better analysis. This was performed using the **rbind** function.

(Fill for Missing Columns) The name of the **rbind R function** stands for row-**bind**. The **rbind function** can be used to **combine** several vectors, matrices and/or data frames by rows.

- The column **trending_date** was converted to a YYYY-MM-DD format.
- The column **publish_time** was converted to a YYYY-MM-DD format.
- The column **dif_days** was converted to a YYYY-MM-DD format.

5. Features

- Categorising YouTube videos based on their comments and statistics.
- A clear visual representation showing the correlation between the number of likes and the number of views.
- Identify the top trending channel and top trending videos and display in the form of bigrams.
- Analysing what factors affect how popular a YouTube video will be.
- A clear visual representation of distribution of attributes like "views", "comments" and "likes" in the form of histograms and scatter plots.
- Sentiment analysis in a variety of forms.
- Generating a Word Cloud displaying the most popular or common words used in trending videos.
- Studying the relation of time between releasing a video and it becoming trending on Youtube.

Moving on to the demo

To begin with, we install and import all the required libraries.

Next, we import the six datasets from six different regions and merge them to form one single data frame ie. videos. Further we manipulate a few columns by reformatting them.

Next, we plot the correlation matrix using the corrplot function. The columns category_id, views, likes, dislikes and comment_count are taken under consideration. The visualisation is shown in Fig 12.
Videos with the most number of likes and most number of views are selected using an order by clause. The visualisations are shown in Fig 13 and Fig 14 respectively.
Videos with the most number of dislikes and most number of comments are selected using an order by clause. The visualisations are shown in Fig 15 and Fig 16 respectively.

We use ggplot to generate a bar chart to represent the top countries based on the maximum number of views and maximum number of likes. We consider the video as a dataframe. We consider all the rows and group them by location. Then we reorder them by the maximum number of views, and fill on the basis of location. The fill is unique for each country. We do this for all the countries and it displays the max views from each country. The same is done for the total number of likes.

6. Algorithms

a. Exploratory Data Analysis:

Exploratory data analysis (**EDA**) is the very first step in a data project.

EDA is an iterative approach that includes:

- Generating questions about our data
- Searching for the answers by using visualization, transformation, and modeling of our data.
- Using the lessons that we learn in order to refine our set of questions or to generate a new set of questions.

EDA consists of univariate (1-variable) and bivariate (2-variables) analysis.

- Step 1 - First approach to data
- Step 2 - Analyzing categorical variables
- Step 3 - Analyzing numerical variables
- Step 4 - Analyzing numerical and categorical at the same time

b. *Sentiment Analysis:*

Sentiment analysis is the computational task of automatically determining what feelings a writer is expressing in text. Sentiment is often framed as a binary distinction (positive vs. negative), but it can also be a more fine-grained entity.

Some examples of applications for sentiment analysis include:

- Analyzing the social media discussion around a certain topic
- Determining whether reviews are positive or negative

It can be useful to quickly summarize some qualities of text, especially if you have so much text that a human reader cannot analyze all of it.

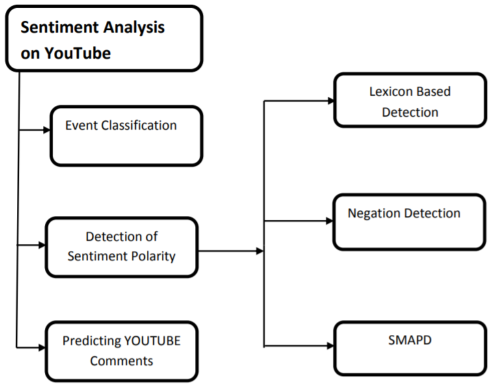

Sentiment Analysis on YouTube
Event Classification
Detection of Sentiment Polarity
Predicting YOUTUBE Comments
Lexicon Based Detection
Negation Detection
SMAPD

Fig.1. Sentiment Analysis on YOUTUBE

This function loads text and calculates sentiment of each sentence. It classifies sentences into 6 categories: Positive, Negative, Very Positive, Very Negative Sarcasm and Neutral.

Value: A vector containing sentiment of each sentence.

Examples:

- calculate_sentiment("This is good")
- calculate_sentiment(c("This is good","This is bad"))

c. *Visualization through Word Cloud:*

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

The reasons one should use word clouds to present the text data are:

- Word clouds add simplicity and clarity. The most used keywords stand out better
- Word clouds are a potent communication tool. They are easy to understand, to be shared, and are impactful.
- Word clouds are visually more engaging than a table data.

Steps to generate a word cloud:

Step 1: Create a Text File

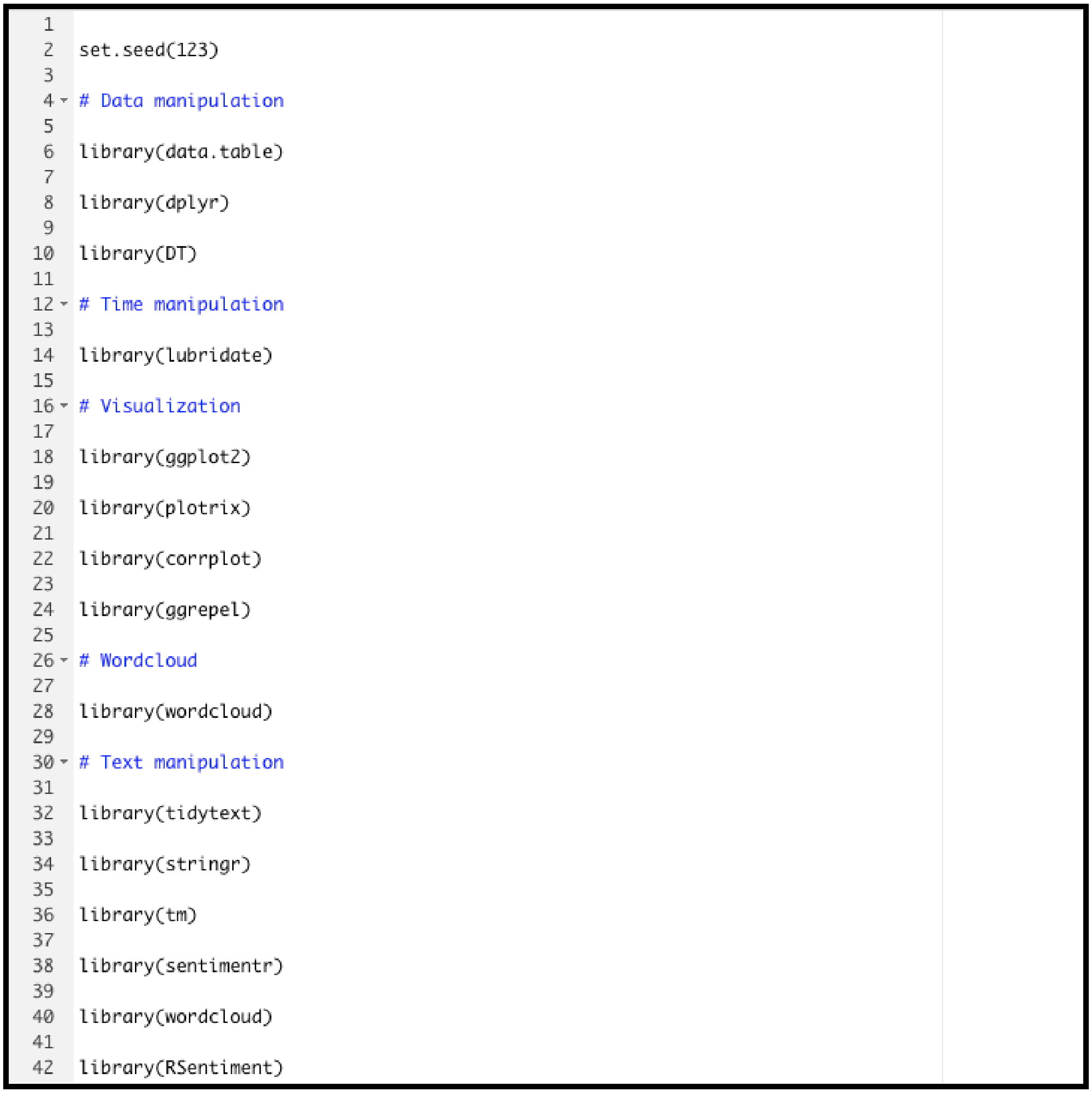
Step 2: Install and Load the Required Packages

Step 3: Text Mining

Step 4: Build a term-document Matrix

Step 4: Build a term-document Matrix

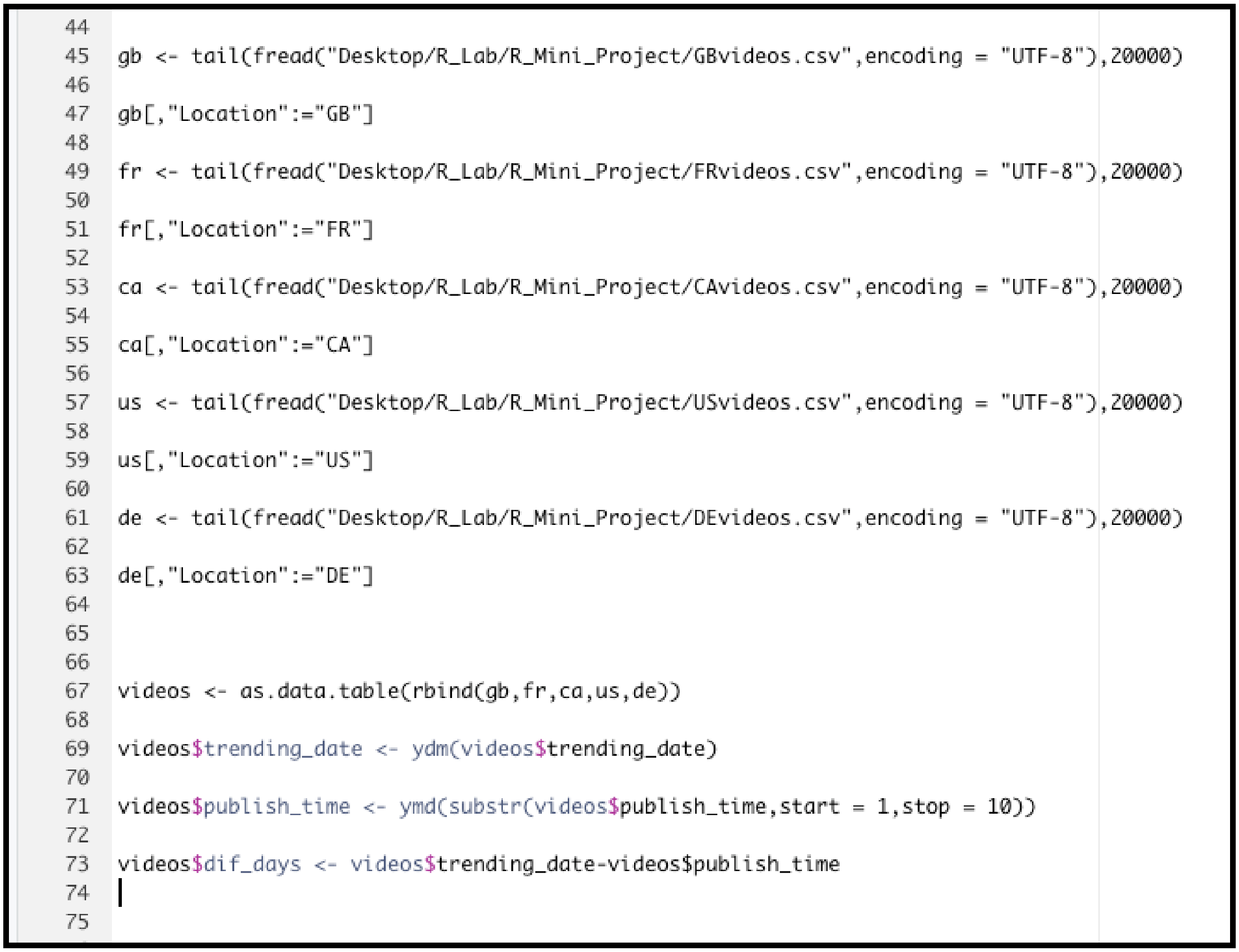
7. Code



```r

set.seed(123)

# Data manipulation

library(data.table)

library(dplyr)

library(DT)

# Time manipulation

library(lubridate)

# Visualization

library(ggplot2)

library(plotrix)

library(corrplot)

library(ggrepel)

# Wordcloud

library(wordcloud)

# Text manipulation

library(tidytext)

library(stringr)

library(tm)

library(sentimentr)

library(wordcloud)

library(RSentiment)
```

Fig 1. Importing Libraries

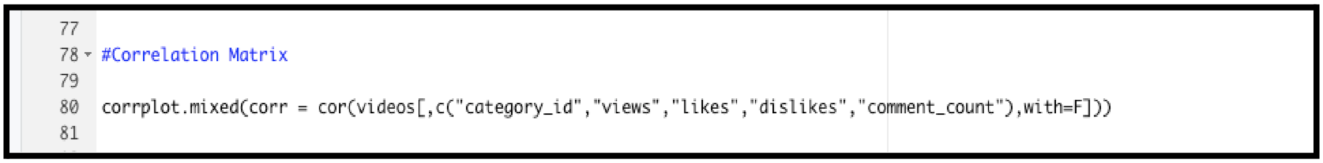
To begin with, we install and import all the required libraries.



```

gb <- tail(fread("Desktop/R_Lab/R_Mini_Project/GBvideos.csv",encoding = "UTF-8"),20000)

gb[,"Location":="GB"]

fr <- tail(fread("Desktop/R_Lab/R_Mini_Project/FRvideos.csv",encoding = "UTF-8"),20000)

fr[,"Location":="FR"]

ca <- tail(fread("Desktop/R_Lab/R_Mini_Project/CAvideos.csv",encoding = "UTF-8"),20000)

ca[,"Location":="CA"]

us <- tail(fread("Desktop/R_Lab/R_Mini_Project/USvideos.csv",encoding = "UTF-8"),20000)

us[,"Location":="US"]

de <- tail(fread("Desktop/R_Lab/R_Mini_Project/DEvideos.csv",encoding = "UTF-8"),20000)

de[,"Location":="DE"]



videos <- as.data.table(rbind(gb,fr,ca,us,de))

videos$trending_date <- ydm(videos$trending_date)

videos$publish_time <- ymd(substr(videos$publish_time,start = 1,stop = 10))

videos$dif_days <- videos$trending_date-videos$publish_time


```

Fig 2. Importing the dataset

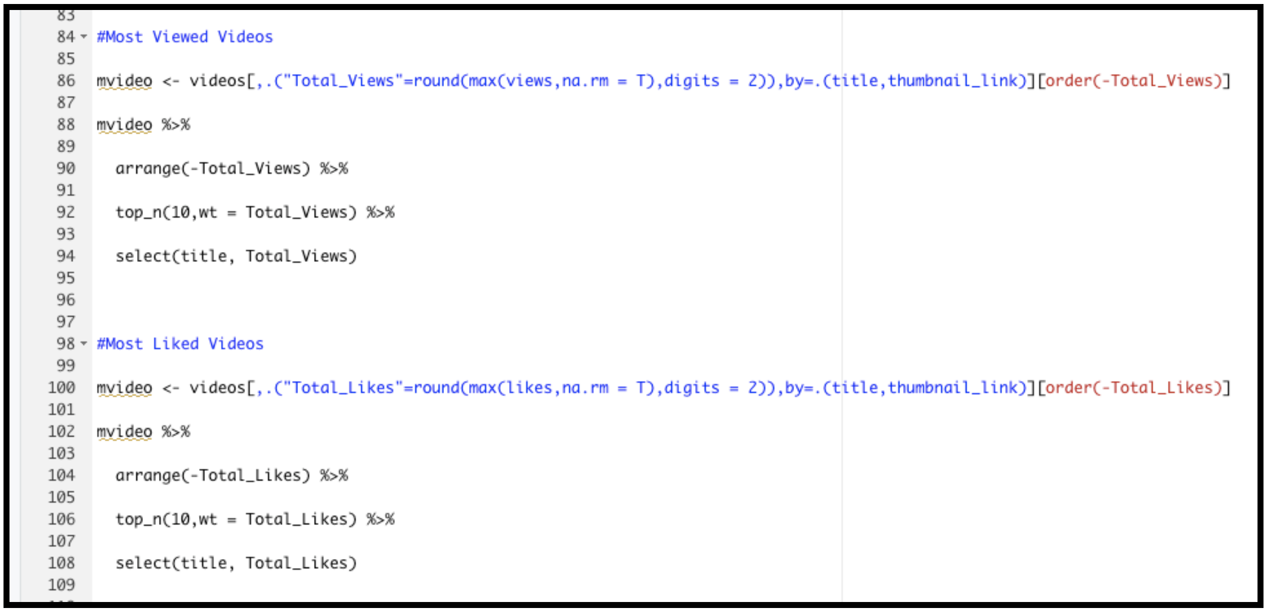
Next, we import the six datasets from six different regions and merge them to form one single data frame ie. videos. Further we manipulate a few columns by reformatting them.



```

#Correlation Matrix

corrplot.mixed(corr = cor(videos[,c("category_id","views","likes","dislikes","comment_count"),with=F]))

```

Fig 2. Correlation Matrix

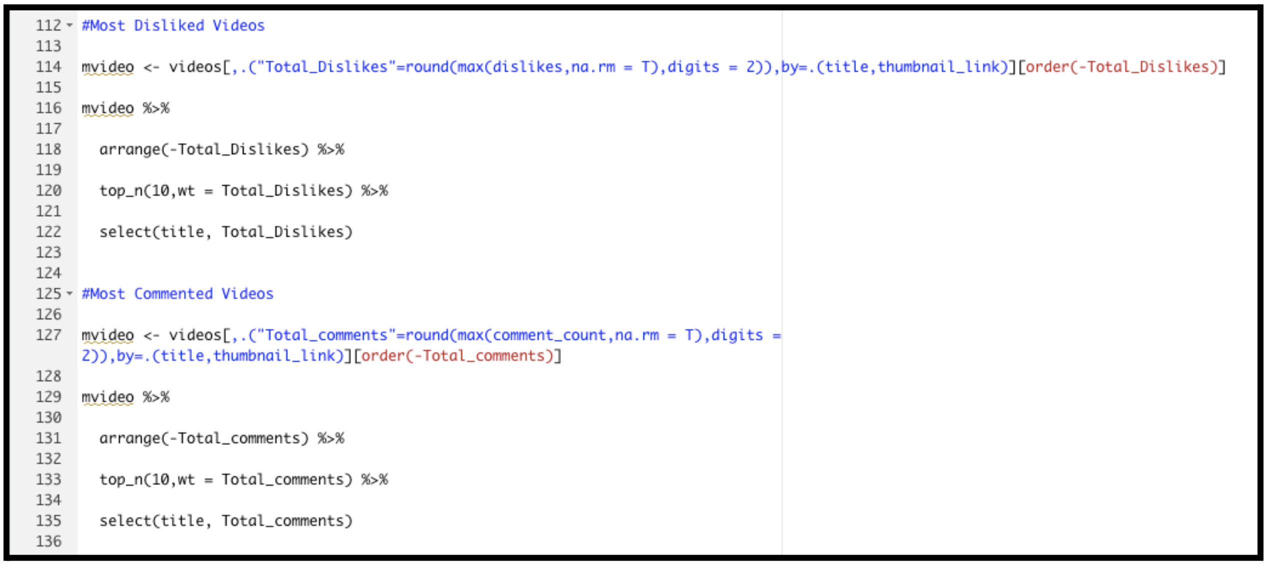
Next, we plot the correlation matrix using the corrplot function. The columns category_id, views, likes, dislikes and comment_count are taken under consideration. The visualisation is shown in Fig 12.



```
#Most Viewed Videos

mvideo <- videos[,.("Total_Views"=round(max(views,na.rm = T),digits = 2)),by=.(title,thumbnail_link)][order(-Total_Views)]

mvideo %>%

  arrange(-Total_Views) %>%

  top_n(10,wt = Total_Views) %>%

  select(title, Total_Views)



#Most Liked Videos

mvideo <- videos[,.("Total_Likes"=round(max(likes,na.rm = T),digits = 2)),by=.(title,thumbnail_link)][order(-Total_Likes)]

mvideo %>%

  arrange(-Total_Likes) %>%

  top_n(10,wt = Total_Likes) %>%

  select(title, Total_Likes)

```

Fig 4. Most Liked and Viewed Videos

Videos with the most number of likes and most number of views are selected using an order by clause. The visualisations are shown in Fig 13 and Fig 14 respectively.

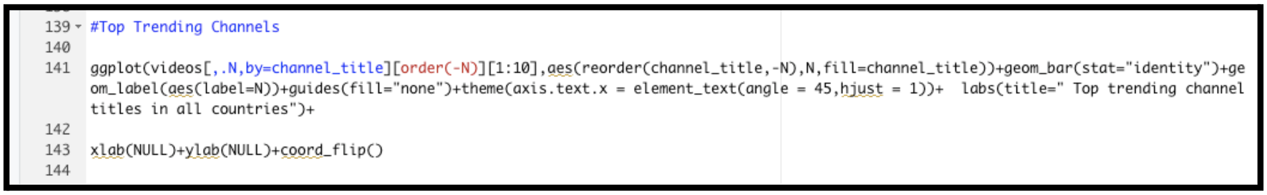


```
#Most Disliked Videos

mvideo <- videos[,.("Total_Dislikes"=round(max(dislikes,na.rm = T),digits = 2)),by=.(title,thumbnail_link)][order(-Total_Dislikes)]

mvideo %>%

  arrange(-Total_Dislikes) %>%

  top_n(10,wt = Total_Dislikes) %>%

  select(title, Total_Dislikes)


#Most Commented Videos

mvideo <- videos[,.("Total_comments"=round(max(comment_count,na.rm = T),digits = 2)),by=.(title,thumbnail_link)][order(-Total_comments)]

mvideo %>%

  arrange(-Total_comments) %>%

  top_n(10,wt = Total_comments) %>%

  select(title, Total_comments)

```

Fig 5. Most Disliked and Commented Videos

Videos with the most number of dislikes and most number of comments are selected using an order by clause. The visualisations are shown in Fig 15 and Fig 16 respectively.

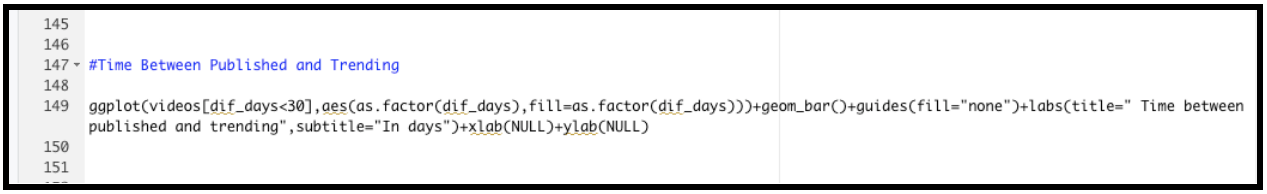
We use ggplot to generate a bar chart to represent the top countries based on the maximum number of views and maximum number of likes. We consider the video as a dataframe. We consider all the rows and group them by location. Then we reorder them by the maximum number of views, and fill on the basis of location. The fill is unique for each country. We do this for all the countries and it displays the max views from each country. The same is done for the total number of likes.



```
#Top Trending Channels

ggplot(videos[,.N,by=channel_title][order(-N)][1:10],aes(reorder(channel_title,-N),N,fill=channel_title))+geom_bar(stat="identity")+geom_label(aes(label=N))+guides(fill="none")+theme(axis.text.x = element_text(angle = 45,hjust = 1))+  labs(title=" Top trending channel titles in all countries")+

xlab(NULL)+ylab(NULL)+coord_flip()

```

Fig 6. Top Trending Channels

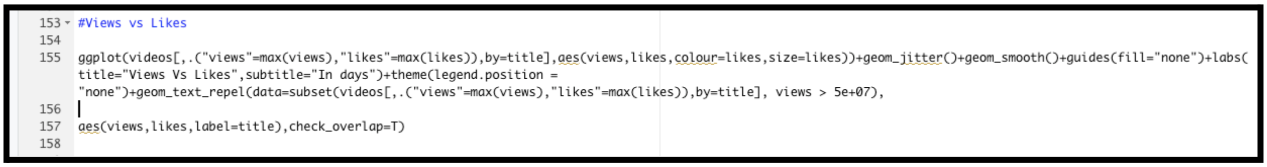
We use the ggplot function to plot a bigram of top trending channels.The visualisation is shown in Fig 17.



```


#Time Between Published and Trending

ggplot(videos[dif_days<30],aes(as.factor(dif_days),fill=as.factor(dif_days)))+geom_bar()+guides(fill="none")+labs(title=" Time between published and trending",subtitle="In days")+xlab(NULL)+ylab(NULL)


```

Fig 6. Time Diff between Published and Trending

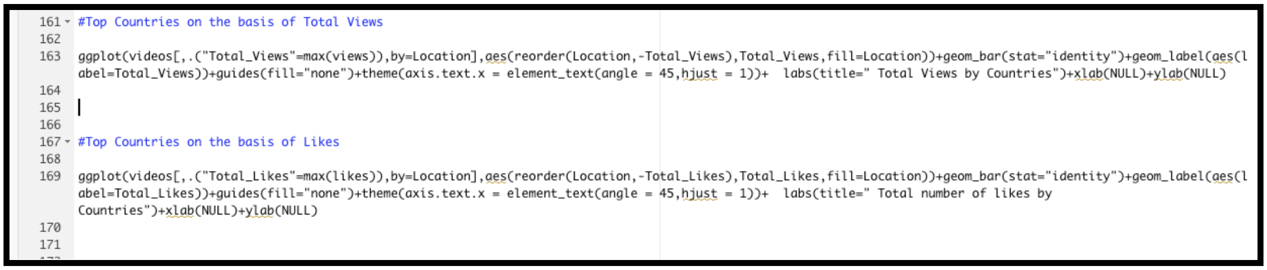
We use the ggplot function to plot a bigram of time elapsed between the published and trending status of the videos.The visualisation is shown in Fig 18.



```
#Views vs Likes

ggplot(videos[,.("views"=max(views),"likes"=max(likes)),by=title],aes(views,likes,colour=likes,size=likes))+geom_jitter()+geom_smooth()+guides(fill="none")+labs(
title="Views Vs Likes",subtitle="In days")+theme(legend.position =
"none")+geom_text_repel(data=subset(videos[,.("views"=max(views),"likes"=max(likes)),by=title], views > 5e+07),

aes(views,likes,label=title),check_overlap=T)

```

Fig 7. Views and Likes

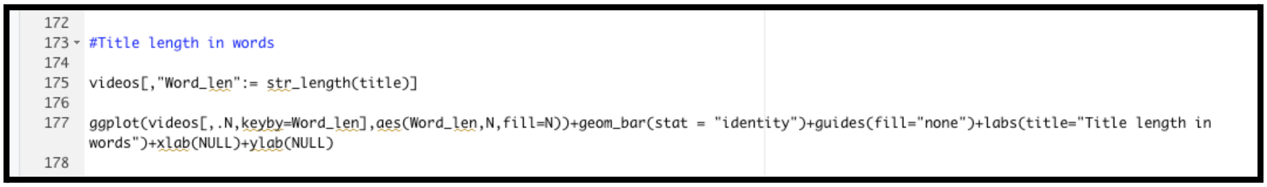
Here, we use ggplot to plot a scatter plot between the number of likes and number of views to get an idea about the correlation between the two for trending videos. The visualisation is shown in Fig 19.



```
#Top Countries on the basis of Total Views

ggplot(videos[,.("Total_Views"=max(views)),by=Location],aes(reorder(Location,-Total_Views),Total_Views,fill=Location))+geom_bar(stat="identity")+geom_label(aes(label=Total_Views))+guides(fill="none")+theme(axis.text.x = element_text(angle = 45,hjust = 1))+  labs(title=" Total Views by Countries")+xlab(NULL)+ylab(NULL)




#Top Countries on the basis of Likes

ggplot(videos[,.("Total_Likes"=max(likes)),by=Location],aes(reorder(Location,-Total_Likes),Total_Likes,fill=Location))+geom_bar(stat="identity")+geom_label(aes(label=Total_Likes))+guides(fill="none")+theme(axis.text.x = element_text(angle = 45,hjust = 1))+  labs(title=" Total number of likes by Countries")+xlab(NULL)+ylab(NULL)
```

Fig 8. Top Countries based on Views and Likes

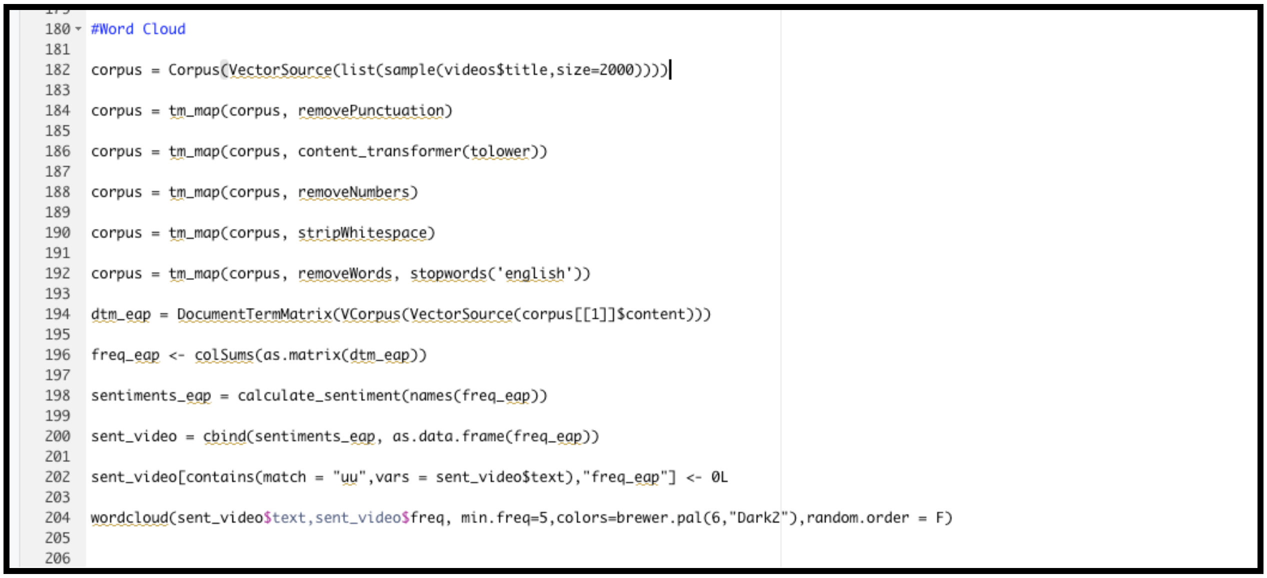
We use ggplot to generate a bar chart to represent the top countries based on the maximum number of views and maximum number of likes. We consider the video as a dataframe. We consider all the rows and group them by location. Then we reorder them by the maximum number of views, and fill on the basis of location. The fill is unique for each country. We do this for all the countries and it displays the max views from each country. The visualisations are shown in Fig 20 and Fig 21 respectively.



```

#Title length in words

videos[,"Word_len":= str_length(title)]

ggplot(videos[,.N,keyby=Word_len],aes(Word_len,N,fill=N))+geom_bar(stat = "identity")+guides(fill="none")+labs(title="Title length in words")+xlab(NULL)+ylab(NULL)
```

Fig 9. Bar Chart of Title Length

We've used ggplot to plot a bar chart for the length of title for videos. The visualization is shown in Fig 22.



```
#Word Cloud

corpus = Corpus(VectorSource(list(sample(videos$title,size=2000))))

corpus = tm_map(corpus, removePunctuation)

corpus = tm_map(corpus, content_transformer(tolower))

corpus = tm_map(corpus, removeNumbers)

corpus = tm_map(corpus, stripWhitespace)

corpus = tm_map(corpus, removeWords, stopwords('english'))

dtm_eap = DocumentTermMatrix(VCorpus(VectorSource(corpus[[1]]$content)))

freq_eap <- colSums(as.matrix(dtm_eap))

sentiments_eap = calculate_sentiment(names(freq_eap))

sent_video = cbind(sentiments_eap, as.data.frame(freq_eap))

sent_video[contains(match = "uu",vars = sent_video$text),"freq_eap"] <- 0L

wordcloud(sent_video$text,sent_video$freq, min.freq=5,colors=brewer.pal(6,"Dark2"),random.order = F)
```

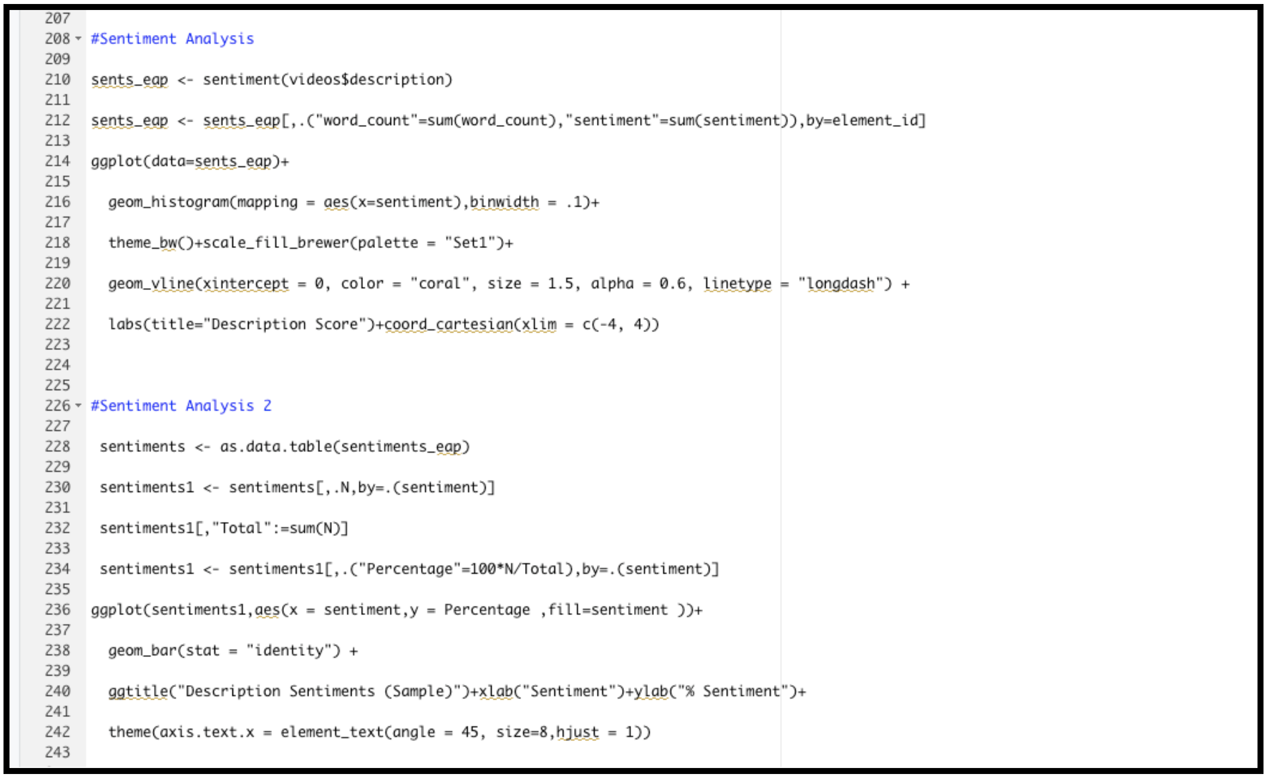
Fig 10. Word Cloud

Here we process the video titles before generating the word cloud. We remove all the punctuations, convert it to lower text, remove all the numbers, strip whitespaces and finally remove all the stopwords. Further a Document Term Matrix is generated and finally we use the wordcloud function to generate the word cloud. The visualization is shown in Fig 23.

Word Cloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.

The reasons one should use word clouds to present the text data are:

- Word clouds add simplicity and clarity. The most used keywords stand out better
- Word clouds are a potent communication tool. They are easy to understand, to be shared, and are impactful.
- Word clouds are visually more engaging than a table data.



```
#Sentiment Analysis

sents_eap <- sentiment(videos$description)

sents_eap <- sents_eap[,.("word_count"=sum(word_count),"sentiment"=sum(sentiment)),by=element_id]

ggplot(data=sents_eap)+

  geom_histogram(mapping = aes(x=sentiment),binwidth = .1)+

  theme_bw()+scale_fill_brewer(palette = "Set1")+

  geom_vline(xintercept = 0, color = "coral", size = 1.5, alpha = 0.6, linetype = "longdash") +

  labs(title="Description Score")+coord_cartesian(xlim = c(-4, 4))



#Sentiment Analysis 2

 sentiments <- as.data.table(sentiments_eap)

 sentiments1 <- sentiments[,.N,by=.(sentiment)]

 sentiments1[,"Total":=sum(N)]

 sentiments1 <- sentiments1[,.("Percentage"=100*N/Total),by=.(sentiment)]

ggplot(sentiments1,aes(x = sentiment,y = Percentage ,fill=sentiment ))+

  geom_bar(stat = "identity") +

  ggtitle("Description Sentiments (Sample)")+xlab("Sentiment")+ylab("% Sentiment")+

  theme(axis.text.x = element_text(angle = 45, size=8,hjust = 1))

```

Fig 11. Sentiment Analysis

We use the RSentiment library to perform sentiment analysis and generate two visualizations.The visualisations are shown in Fig 24 and Fig 25 respectively.

Sentiment analysis is the computational task of automatically determining what feelings a writer is expressing in text. Sentiment is often framed as a binary distinction (positive vs. negative), but it can also be a more fine-grained entity.

Some examples of applications for sentiment analysis include:

- Analyzing the social media discussion around a certain topic
- Determining whether reviews are positive or negative

It can be useful to quickly summarize some qualities of text, especially if you have so much text that a human reader cannot analyze all of it.

8. Visualization

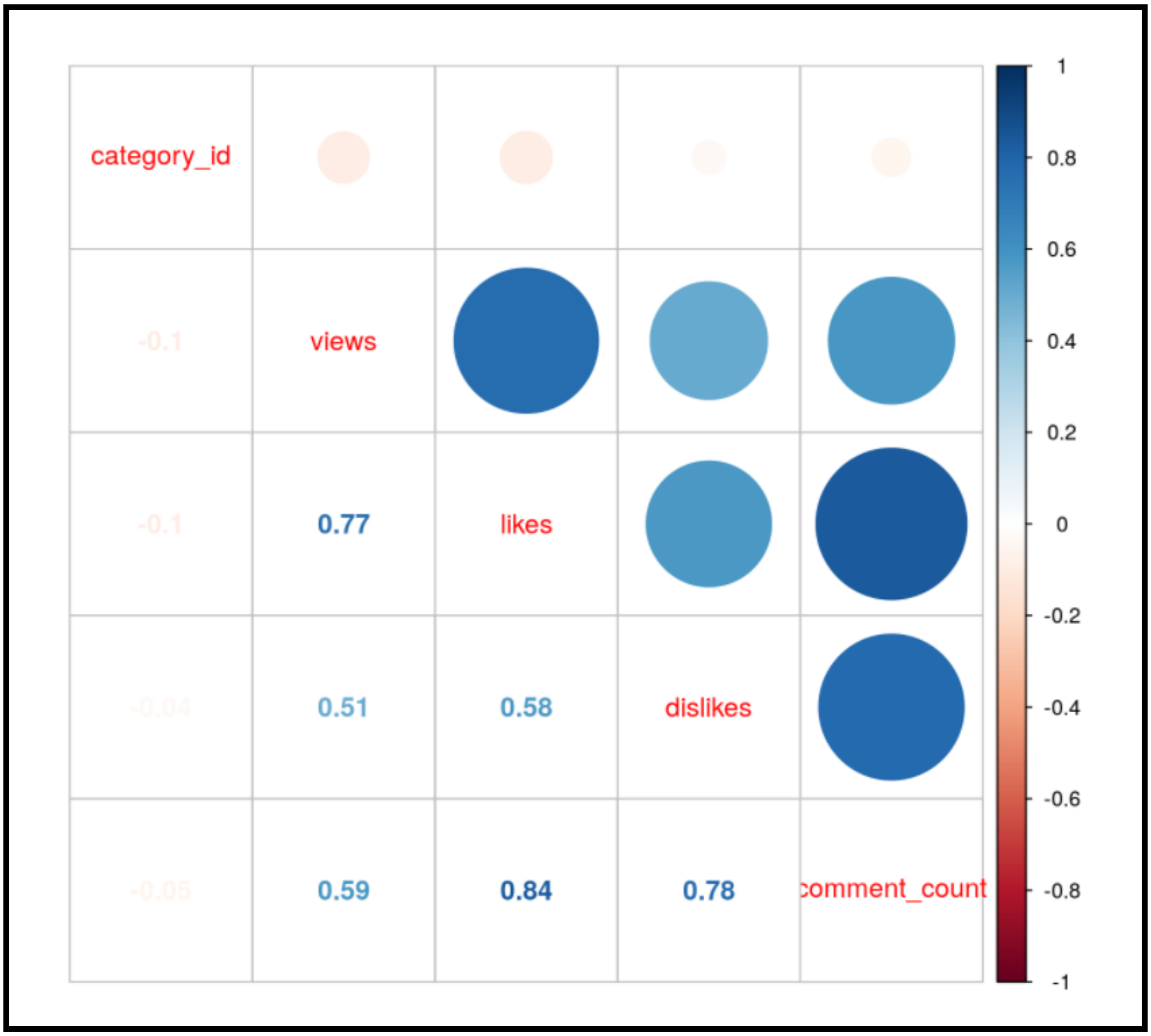
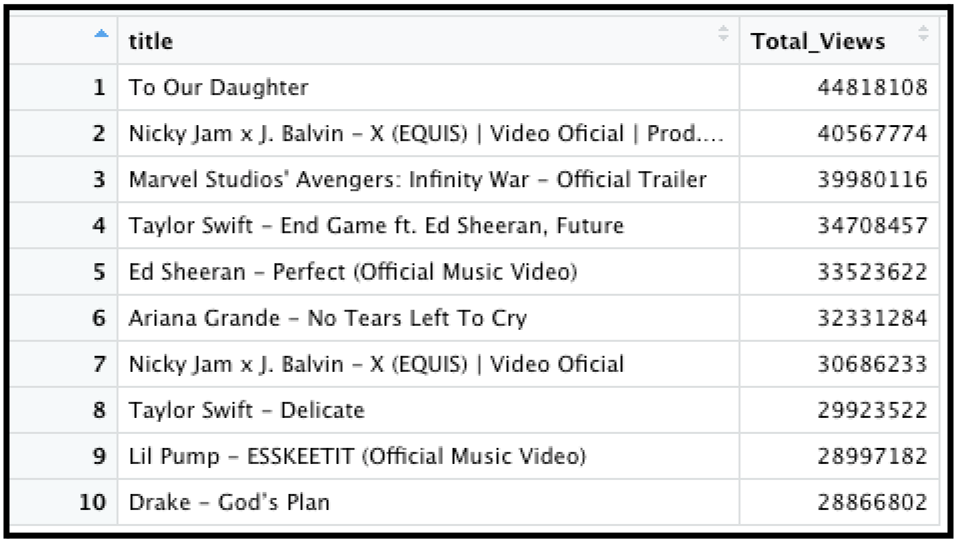
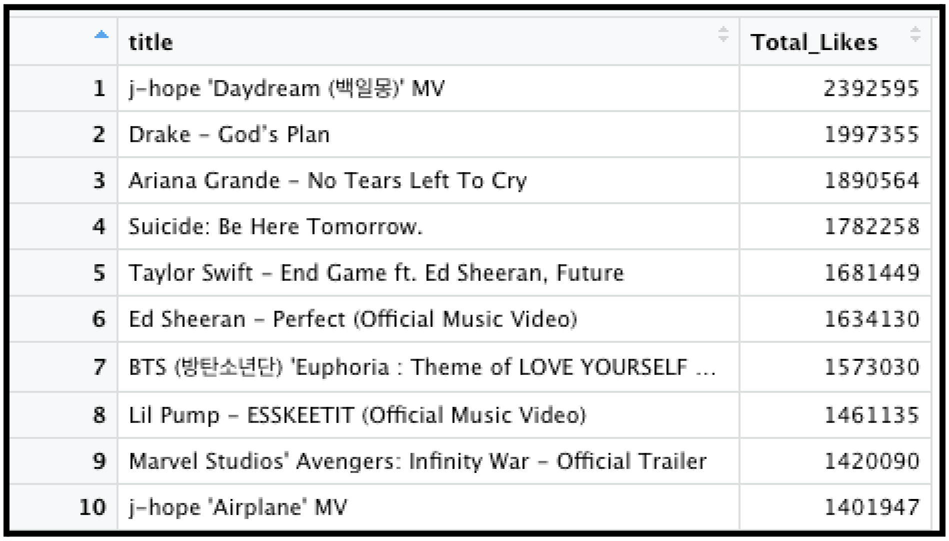


Fig 12. Correlation matrix between views and likes



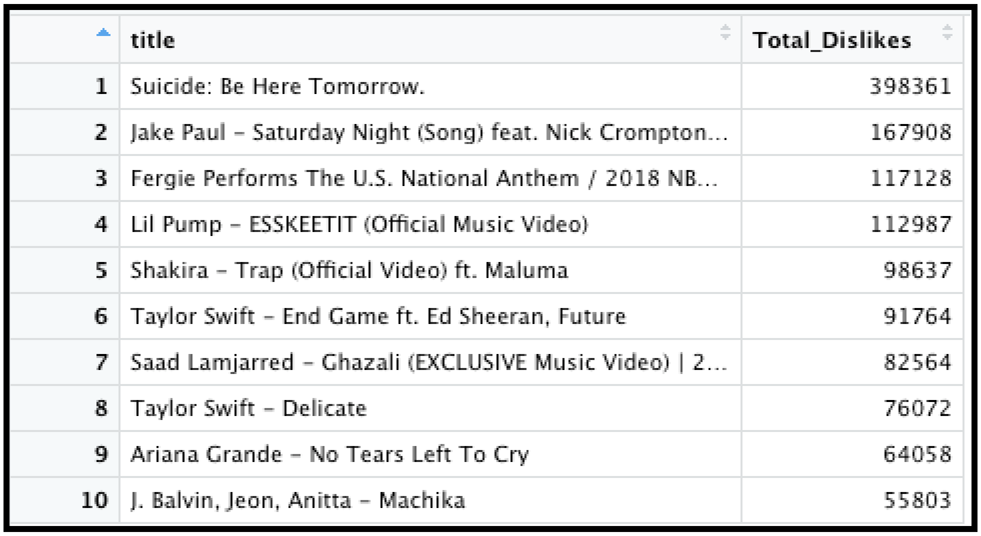
	title	Total_Views
1	To Our Daughter	44818108
2	Nicky Jam x J. Balvin - X (EQUIS) | Video Oficial | Prod.…	40567774
3	Marvel Studios' Avengers: Infinity War - Official Trailer	39980116
4	Taylor Swift - End Game ft. Ed Sheeran, Future	34708457
5	Ed Sheeran - Perfect (Official Music Video)	33523622
6	Ariana Grande - No Tears Left To Cry	32331284
7	Nicky Jam x J. Balvin - X (EQUIS) | Video Oficial	30686233
8	Taylor Swift - Delicate	29923522
9	Lil Pump - ESSKEETIT (Official Music Video)	28997182
10	Drake - God's Plan	28866802

Fig 13. Top 10 Videos based on views



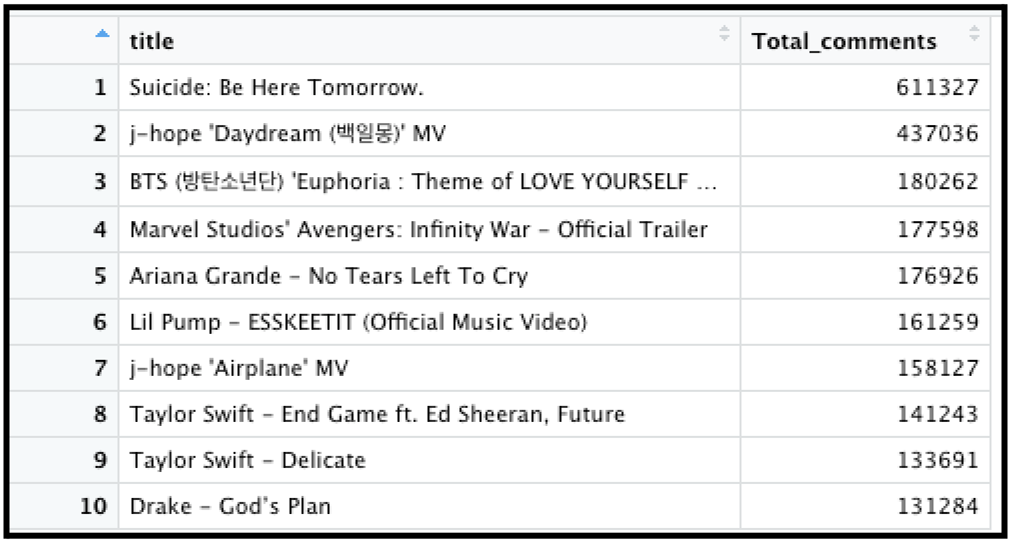
	title	Total_Likes
1	j-hope 'Daydream (백일몽)' MV	2392595
2	Drake - God's Plan	1997355
3	Ariana Grande - No Tears Left To Cry	1890564
4	Suicide: Be Here Tomorrow.	1782258
5	Taylor Swift - End Game ft. Ed Sheeran, Future	1681449
6	Ed Sheeran - Perfect (Official Music Video)	1634130
7	BTS (방탄소년단) 'Euphoria : Theme of LOVE YOURSELF …	1573030
8	Lil Pump - ESSKEETIT (Official Music Video)	1461135
9	Marvel Studios' Avengers: Infinity War - Official Trailer	1420090
10	j-hope 'Airplane' MV	1401947

Fig 14. Top 10 Videos based on likes



	title	Total_Dislikes
1	Suicide: Be Here Tomorrow.	398361
2	Jake Paul - Saturday Night (Song) feat. Nick Crompton…	167908
3	Fergie Performs The U.S. National Anthem / 2018 NB…	117128
4	Lil Pump - ESSKEETIT (Official Music Video)	112987
5	Shakira - Trap (Official Video) ft. Maluma	98637
6	Taylor Swift - End Game ft. Ed Sheeran, Future	91764
7	Saad Lamjarred - Ghazali (EXCLUSIVE Music Video) | 2…	82564
8	Taylor Swift - Delicate	76072
9	Ariana Grande - No Tears Left To Cry	64058
10	J. Balvin, Jeon, Anitta - Machika	55803

Fig 15. Top 10 Videos based on dislikes



	title	Total_comments
1	Suicide: Be Here Tomorrow.	611327
2	j-hope 'Daydream (백일몽)' MV	437036
3	BTS (방탄소년단) 'Euphoria : Theme of LOVE YOURSELF …	180262
4	Marvel Studios' Avengers: Infinity War - Official Trailer	177598
5	Ariana Grande - No Tears Left To Cry	176926
6	Lil Pump - ESSKEETIT (Official Music Video)	161259
7	j-hope 'Airplane' MV	158127
8	Taylor Swift - End Game ft. Ed Sheeran, Future	141243
9	Taylor Swift - Delicate	133691
10	Drake - God's Plan	131284

Fig 16. Top 10 Videos based on comments

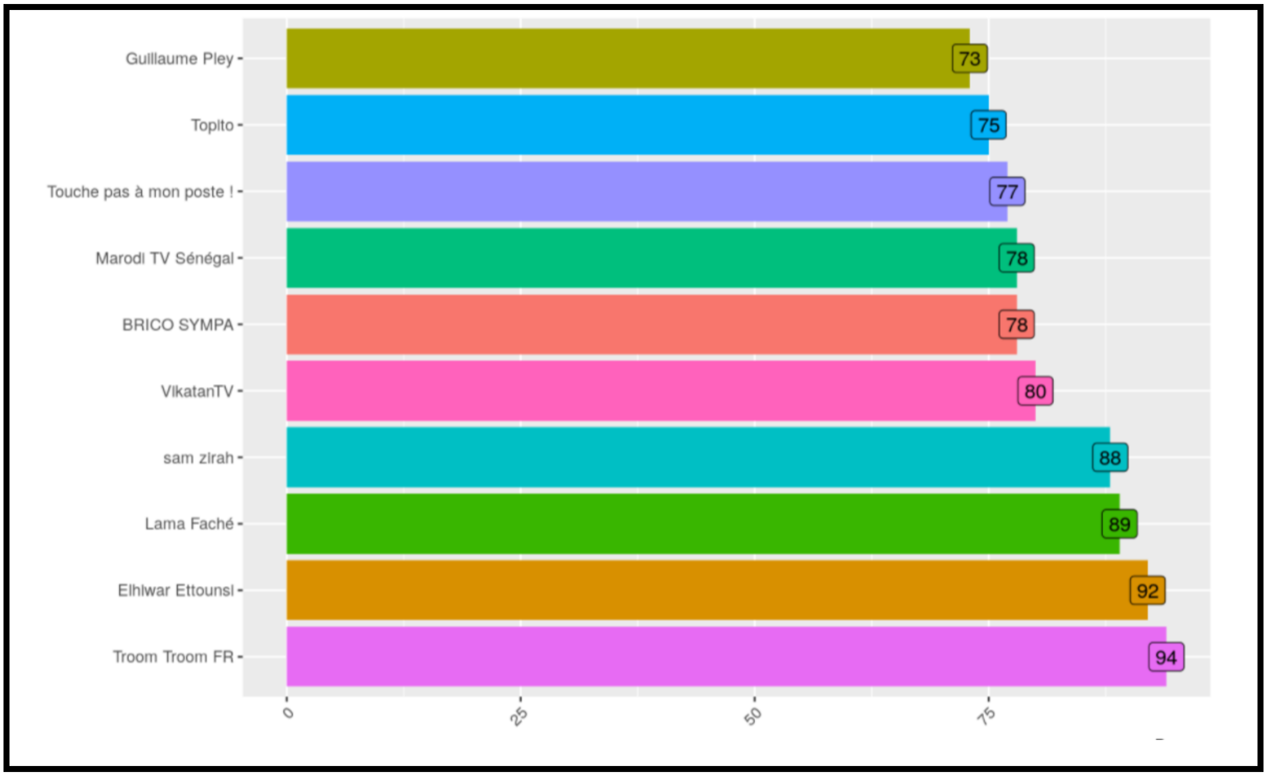


Fig 17. Top trending channels in all countries

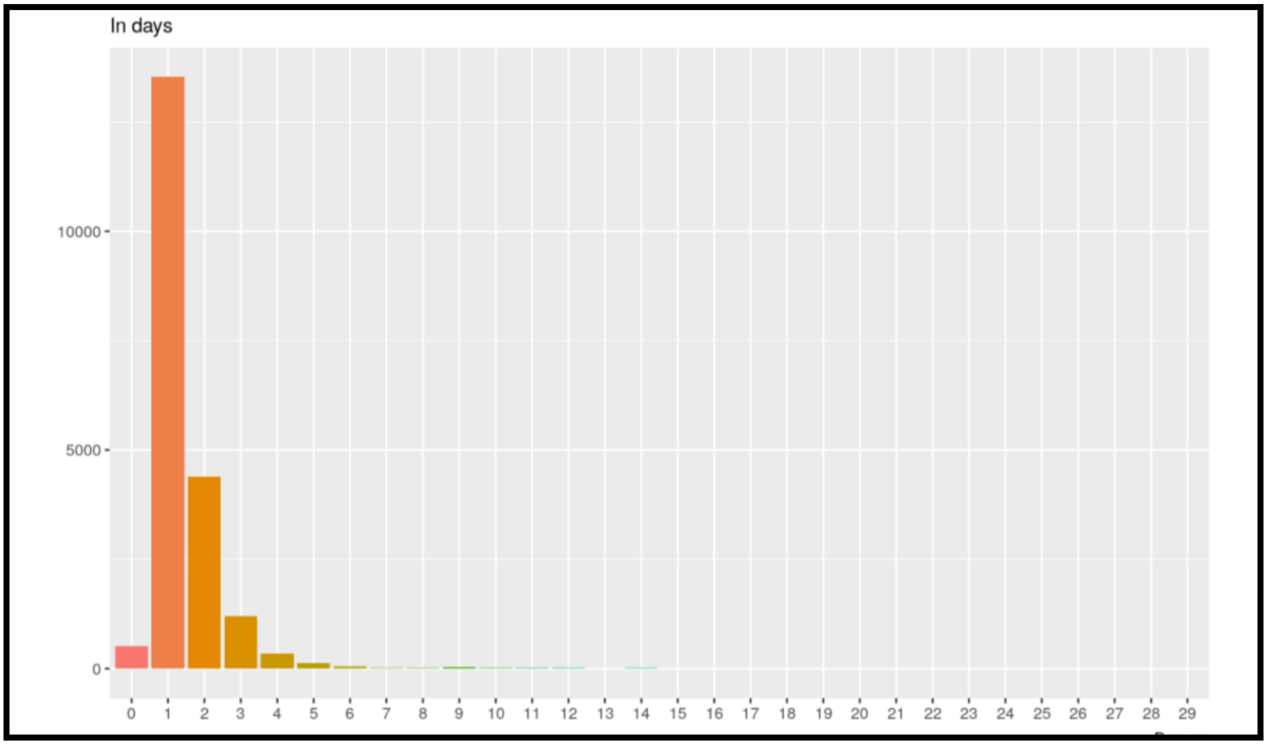


Fig 18. Time between published and trending

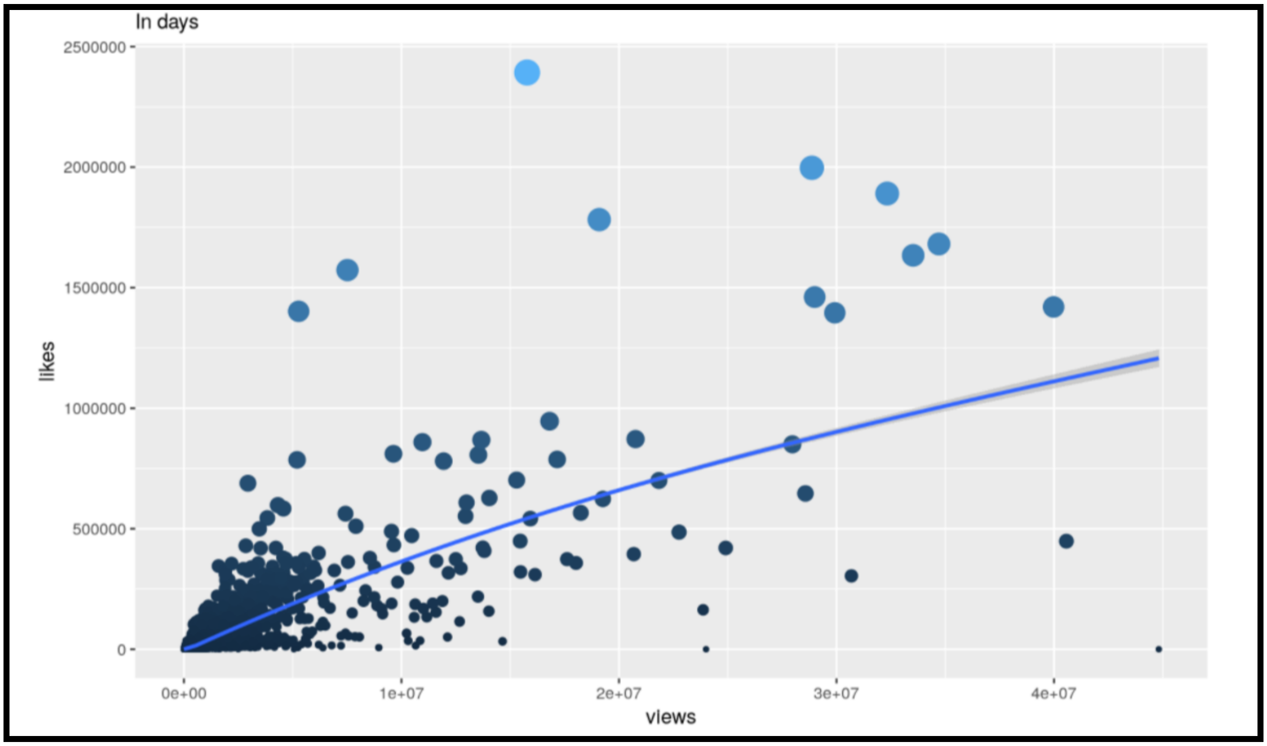


Fig 19. Views vs Likes

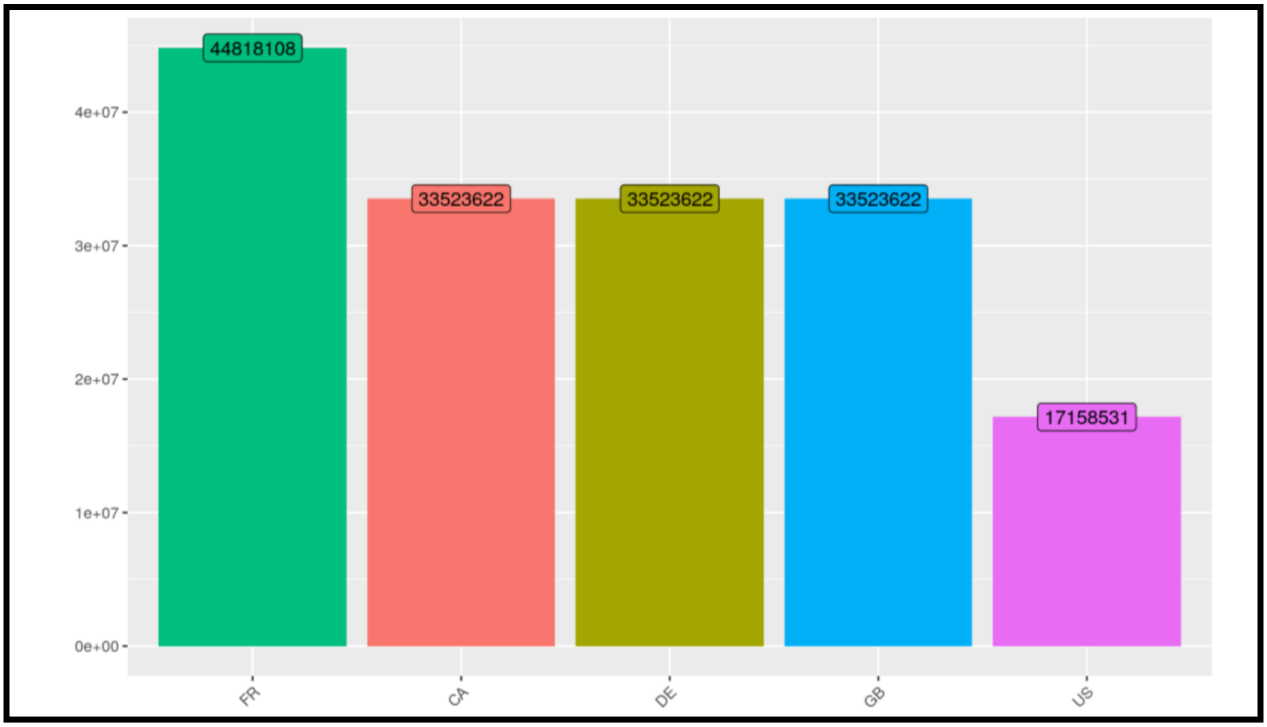


Fig 20. Total Views by Countries

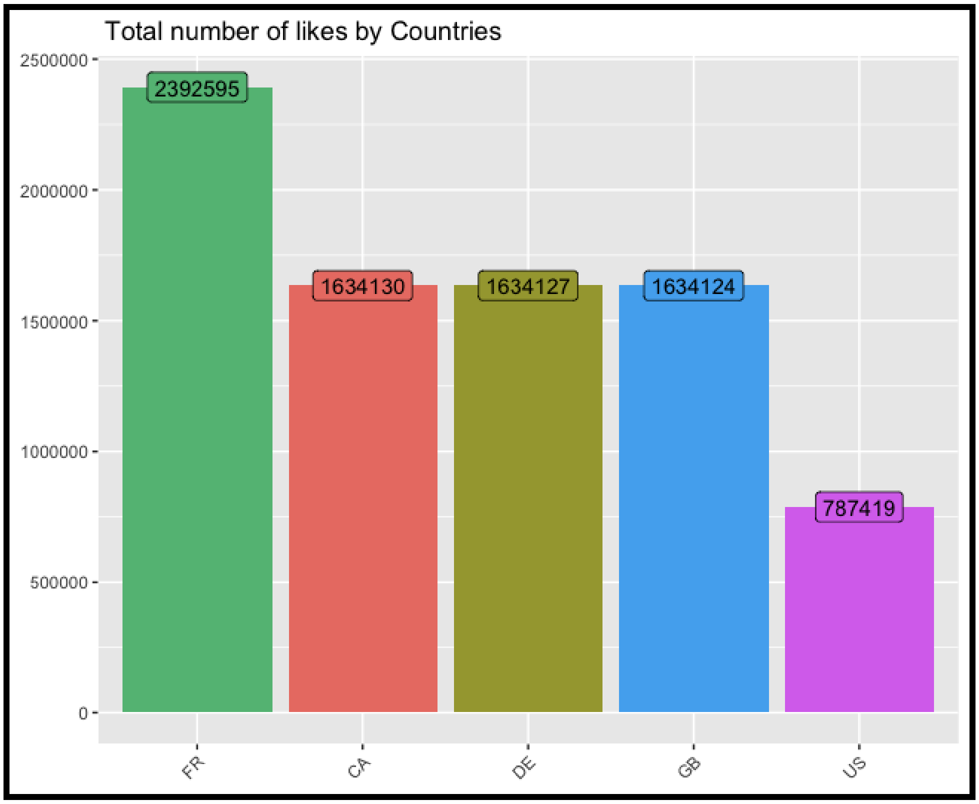


Fig 21. Total Likes by Countries

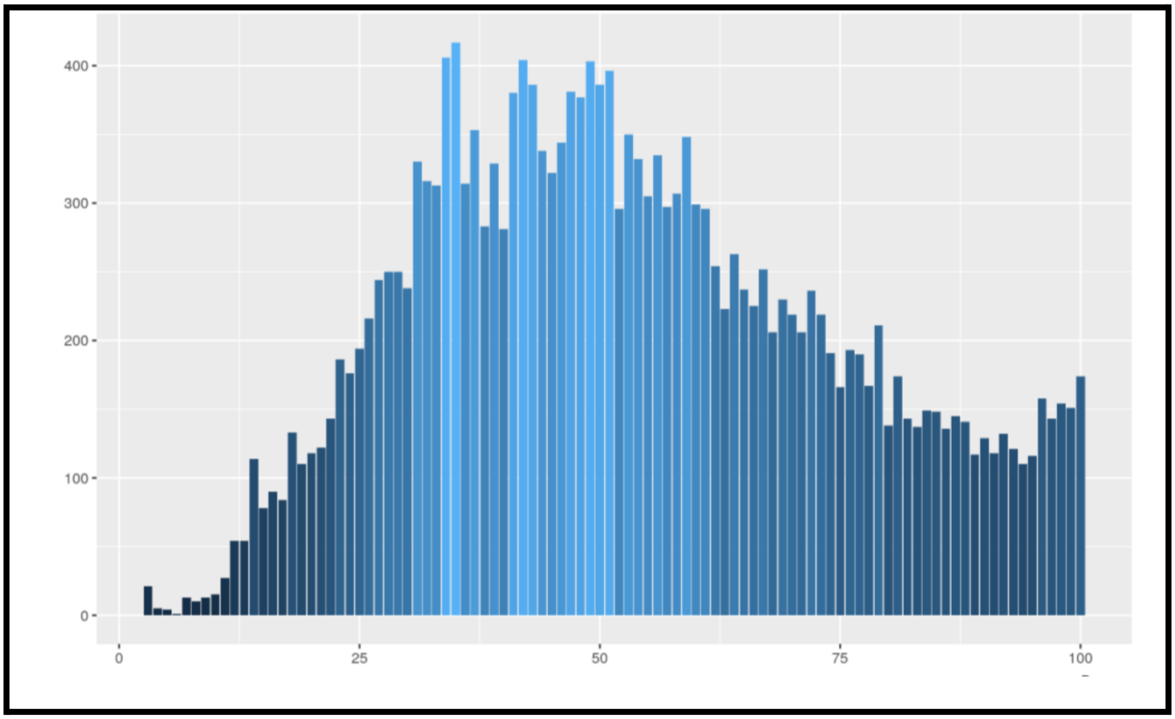


Fig 22. Title length in words

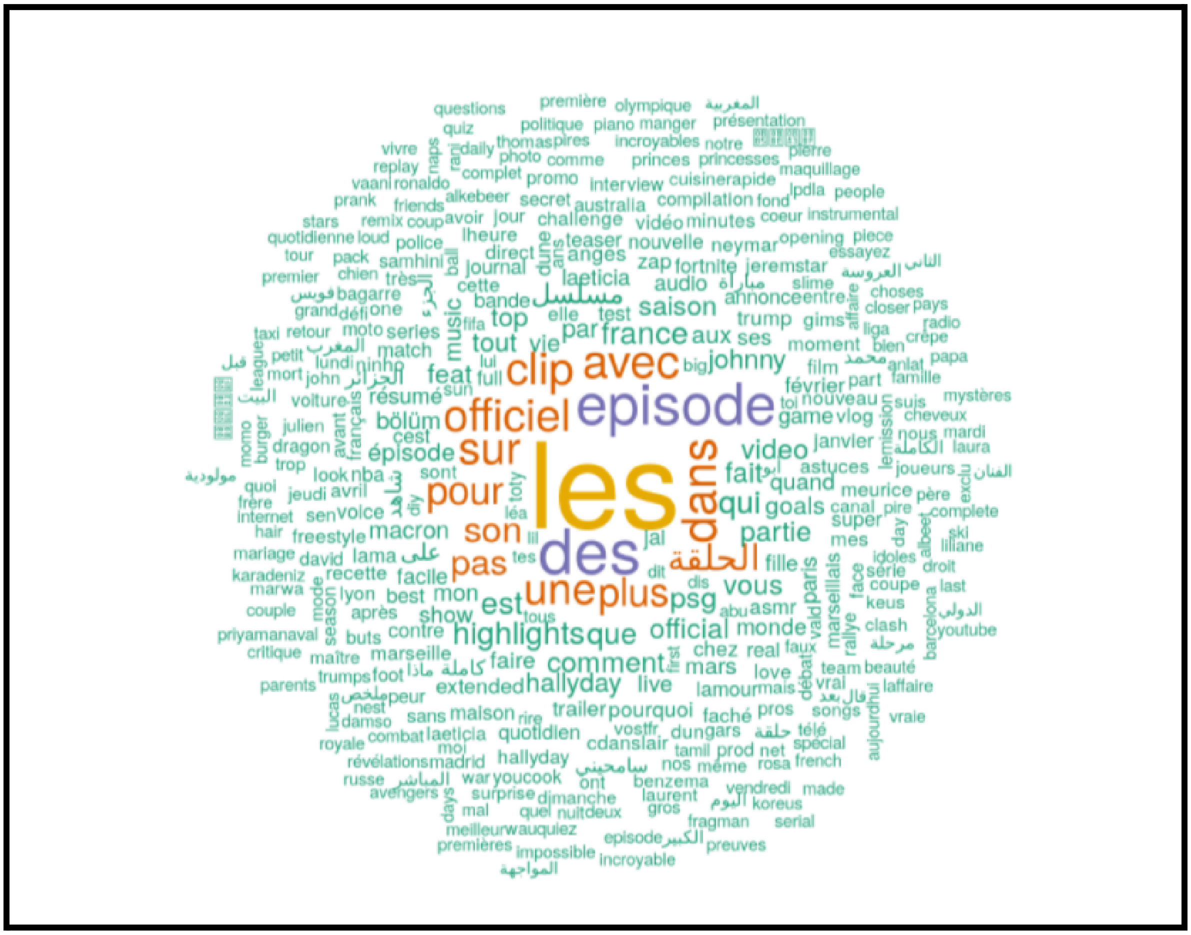


Fig 23: Video title word cloud

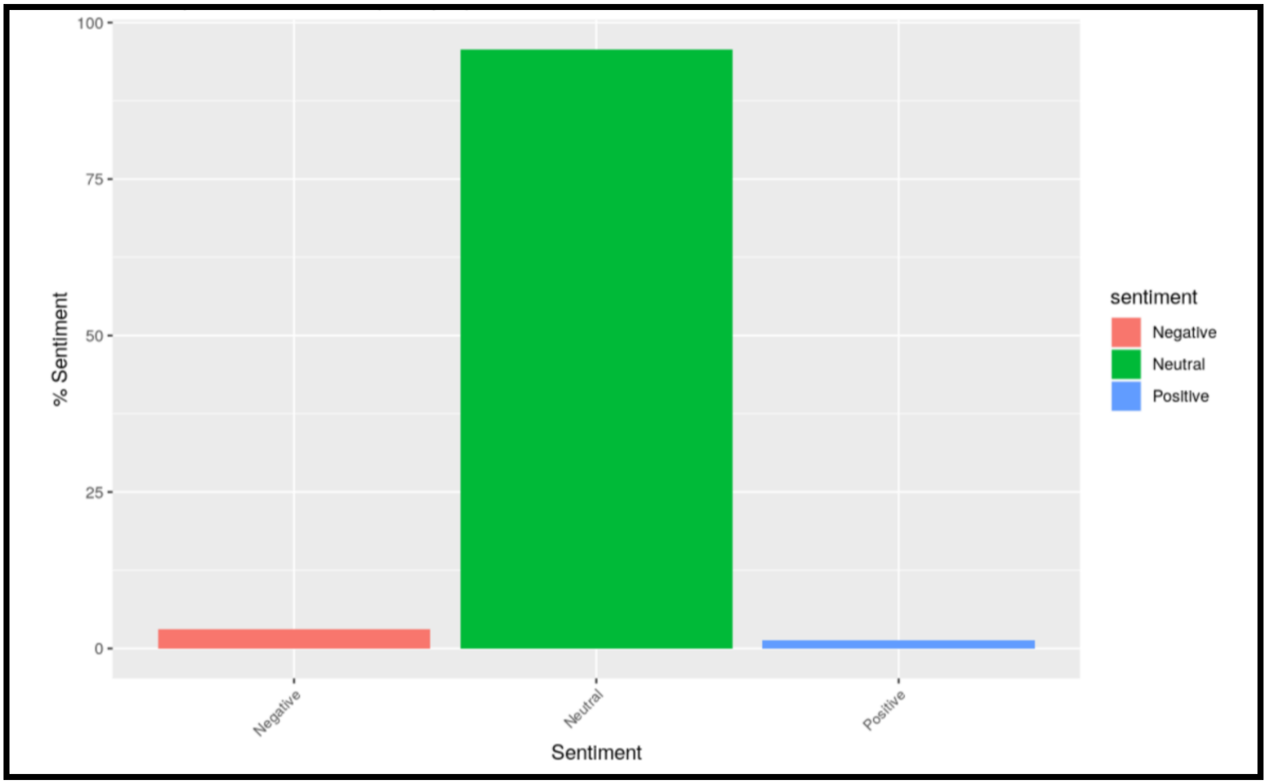


Fig 24. Description Sentiments

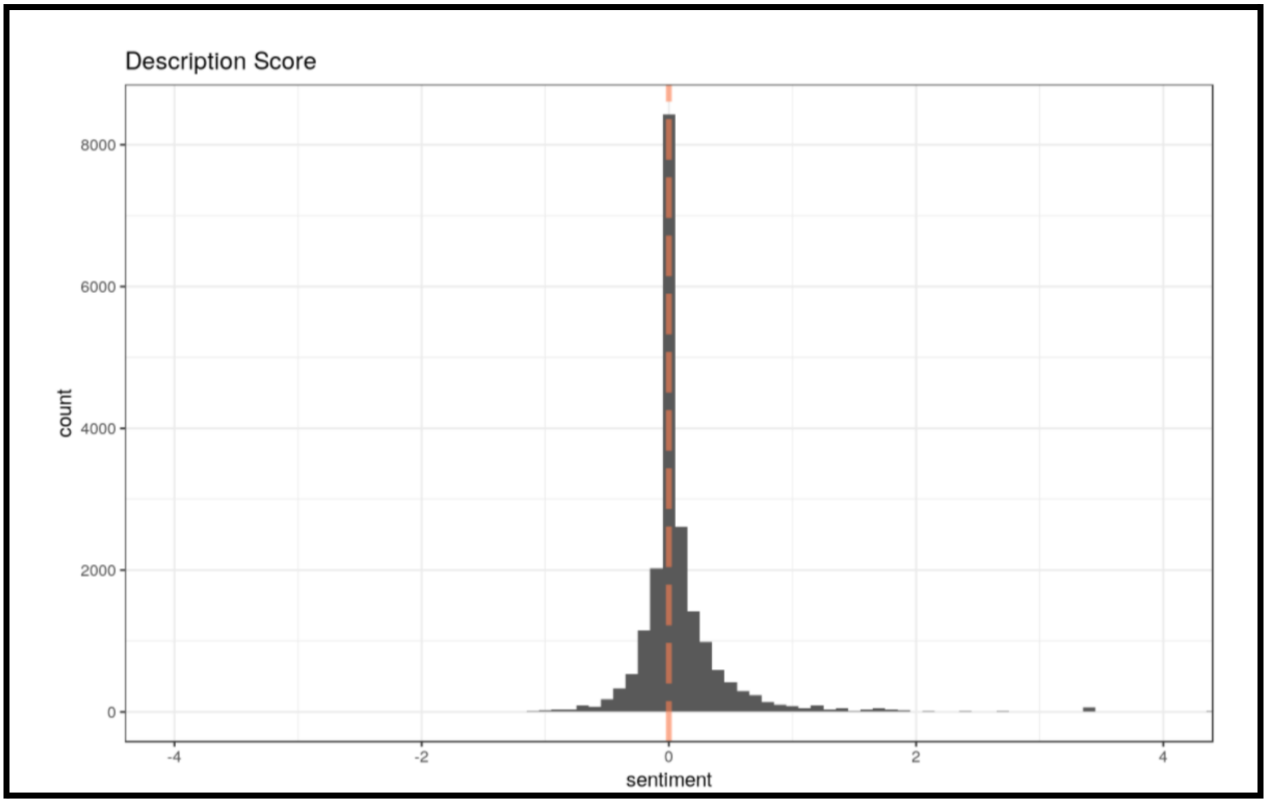


Fig 25. Sentiment Analysis

9. Conclusion

Exploratory data analysis (EDA) is an approach to analyzing data sets to summarize their main characteristics, often with visual methods. This step is very important especially when we arrive at modeling the data in order to apply Machine learning. Plotting in EDA consists of Histograms, Box plot, Scatter plot, and many more. It often takes much time to explore the data. Through the process of EDA, we can ask to define the problem statement or definition on our data set which is very important.

Our aim is to produce a scientific knowledge preprocessing analysis operating solely with the dataset US Videos. This step is important for all data processing exercises and that we wish to emphasize it. Before building theories from knowledge we'd like to grasp key knowledge attributes, like missing values, distinctive counts, outliers, and time-series trends.

The present scope of the project includes analyzing the statistics for a channel/category as well as read counts, likes, dislikes, country wise read etc. By distinguishing classifying/categorizing the polarity of the words, sentiment analysis and generating word cloud will be performed for a selected video. This might tell a user's perspective towards a specific product or a given subject. By Exploiting Sentiment Analysis, we are able to verify if the overall perspective of individuals is positive, negative or neutral towards a selected subject/video.