Experiment 10 (Mini-Project)

**Title:** To perform exploratory data analysis on trending video of youtube data and sentimental data analysis on the video.

**Description :**

### The dataset contains data about youTube trending videos. This database contains 18 attributes.

### YouTube (the world-famous video sharing website) maintains a list of the top trending videos on the platform. According to Variety magazine, “To determine the year’s top-trending videos, YouTube uses a combination of factors including measuring users interactions (number of views, shares, comments and likes). Note that they’re not the most-viewed videos overall for the calendar year”. Top performers on the YouTube trending list are music videos (such as the famously virile “Gangam Style”), celebrity and/or reality TV performances, and the random dude-with-a-camera viral videos that YouTube is well-known for. This dataset is a daily record of the top trending YouTube videos.

### This dataset includes several months (and counting) of data on daily trending YouTube videos. Data is included for the US, GB, DE, CA, FR, RU, MX, KR, JP and IN regions (USA, Great Britain, Germany, Canada, France, Russia, Mexico, South Korea, Japan and India respectively), with up to 200 listed trending videos per day.

### 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. To retrieve the categories for a specific video, find it in the associated JSON. One such file is included for each of the five regions in the dataset.

We have performed exploratory data analysis on this dataset. Then performed sentimental analysis and found out the amount of time between the uploaded\_time and the trending\_time.

**Code and Output:**

***Import Libraries:***

> library(tidyverse)

> library(dplyr)

> library(ggplot)

> library(ggcorrplot)

> library(sentimentr)

> library(wordcloud)

* Library dplyr is used for data manipulation.
* Library ggplot and ggcorrplot is used for data visualization.
* Library sentimentr and wordcloud is used for text manipulation.

***Import and Explore dataset***

no\_lines\_ca <- countLines("C:/Users/ANIKET/Documents/RLab/Dataset/CAvideos.csv")

no\_lines\_de <- countLines("Dataset/DEvideos.csv")

no\_lines\_fr <- countLines("Dataset/FRvideos.csv")

no\_lines\_gb <- countLines("Dataset/GBvideos.csv")

no\_lines\_in <- countLines("Dataset/INvideos.csv")

no\_lines\_jp <- countLines("Dataset/JPvideos.csv")

no\_lines\_kr <- countLines("Dataset/KRvideos.csv")

no\_lines\_mx <- countLines("Dataset/MXvideos.csv")

no\_lines\_ru <- countLines("Dataset/RUvideos.csv")

no\_lines\_us <- countLines("Dataset/USvideos.csv")

First we are counting number of data records (rows) in each csv file

Then we are storing data and no of line in a variable.

df\_ca <- tail(fread("Dataset/CAvideos.csv", encoding = "UTF-8"), no\_lines\_ca)

df\_de <- tail(fread("Dataset/DEvideos.csv", encoding = "UTF-8"), no\_lines\_de)

df\_fr <- tail(fread("Dataset/FRvideos.csv", encoding = "UTF-8"), no\_lines\_fr)

df\_gb <- tail(fread("Dataset/GBvideos.csv", encoding = "UTF-8"), no\_lines\_gb)

df\_in <- tail(fread("Dataset/INvideos.csv", encoding = "UTF-8"), no\_lines\_in)

df\_jp <- tail(fread("Dataset/JPvideos.csv", encoding = "UTF-8"), no\_lines\_jp)

df\_kr <- tail(fread("Dataset/KRvideos.csv", encoding = "UTF-8"), no\_lines\_kr)

df\_mx <- tail(fread("Dataset/MXvideos.csv", encoding = "UTF-8"), no\_lines\_mx)

df\_ru <- tail(fread("Dataset/RUvideos.csv", encoding = "UTF-8"), no\_lines\_ru)

df\_us <- tail(fread("Dataset/USvideos.csv", encoding = "UTF-8"), no\_lines\_us)

# add additional column "Country" to each data set

df\_ca$country <- c("Canada")

df\_de$country <- c("Germany")

df\_fr$country <- c("France")

df\_gb$country <- c("Great Britain")

df\_in$country <- c("India")

df\_jp$country <- c("Japan")

df\_kr$country <- c("South Korea")

df\_mx$country <- c("Mexico")

df\_ru$country <- c("Russia")

df\_us$country <- c("USA")

df\_all <- as.data.table(rbind(df\_ca, df\_de, df\_fr, df\_gb, df\_in, df\_jp, df\_kr, df\_mx, df\_ru, df\_us))

Now we will for ease add one column to know country of the trending video. And will bind all that rows in a single datset. It will have columns as mentioned below.

# reorder columns

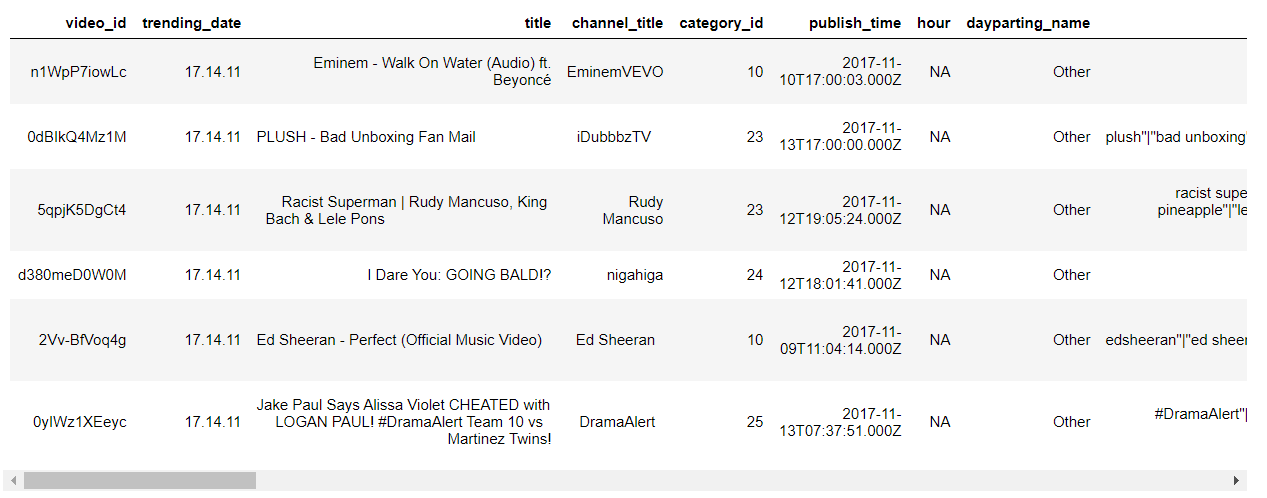
col\_order <- c("video\_id", "trending\_date", "title", "channel\_title", "category\_id", "publish\_time", "hour", "dayparting\_name",

"tags", "views", "likes", "dislikes", "comment\_count", "thumbnail\_link", "comments\_disabled",

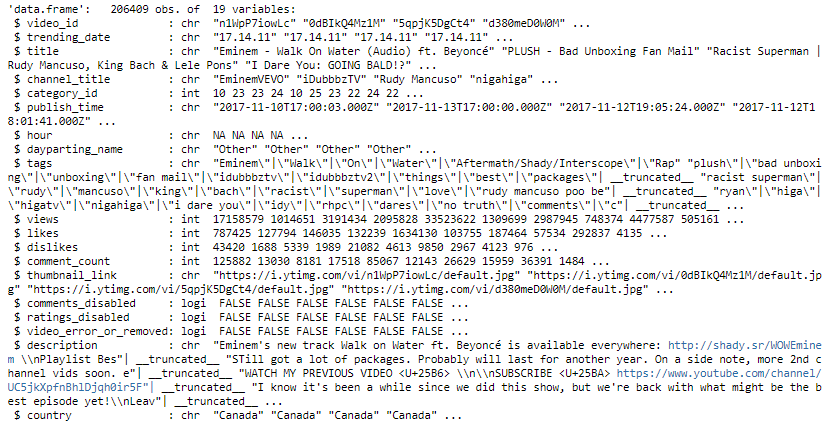
"ratings\_disabled", "video\_error\_or\_removed", "description", "country" )

df\_all <- df\_all[, col\_order]

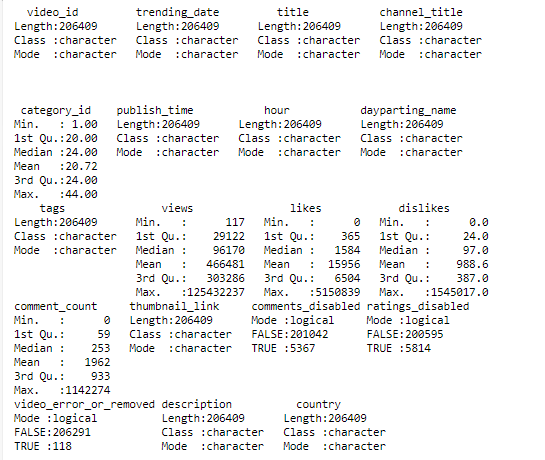
head(df\_all)



str(data) #GIVES INTERNAL STRUCTURE OF OBJECT



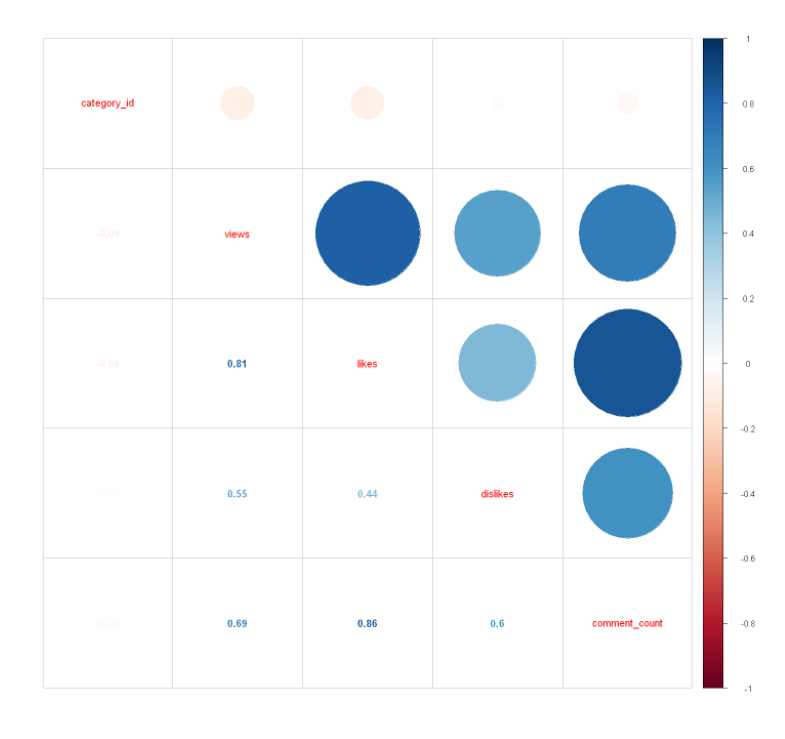
summary(data)



***Corrplot***

A correlation matrix is simply a table which displays the correlation. We will find correlation between views, likes, dislikes and comment count.

corrplot.mixed(corr = cor(df\_all[,c("category\_id","views","likes","dislikes","comment\_count")]))



***Frequency of tags using bar plot***

First we will use tag column from whole dataset and will clean the data by removing the unwanted symbols or garbage. Then we are converting all data into lower case to process easily. Then we will take frequency of each tag and save it in that tag using data frame.

Then we will barplot it to visualise it easily.

Encoding(df\_in$tags) <- "UTF-8"

text <- df\_in$tags

#memory size increasing

memory.limit(size=50000)

# Load the data as a corpus

docs <- Corpus(VectorSource(text))

toSpace <- content\_transformer(function (x , pattern ) gsub(pattern, " ", x))

docs <- tm\_map(docs, toSpace, "/")

docs <- tm\_map(docs, toSpace, "@")

docs <- tm\_map(docs, toSpace, "\\|")

# Convert the text to lower case

docs <- tm\_map(docs, content\_transformer(tolower))

# Remove numbers

docs <- tm\_map(docs, removeNumbers)

# Remove english common stopwords

docs <- tm\_map(docs, removeWords, stopwords("english"))

# Remove your own stop word

# specify your stopwords as a character vector

docs <- tm\_map(docs, removeWords, c(""))

# Remove punctuations

docs <- tm\_map(docs, removePunctuation)

# Eliminate extra white spaces

docs <- tm\_map(docs, stripWhitespace)

# Text stemming

docs <- tm\_map(docs, stemDocument)

dtm <- TermDocumentMatrix(docs)

m <- as.matrix(dtm)

v <- sort(rowSums(m),decreasing=TRUE)

d <- data.frame(word = names(v),freq=v)

# Increase bottom margin

par(mar=c(10,10,4,4))

barplot(d[1:10,]$freq, las = 2, names.arg = d[1:10,]$word,

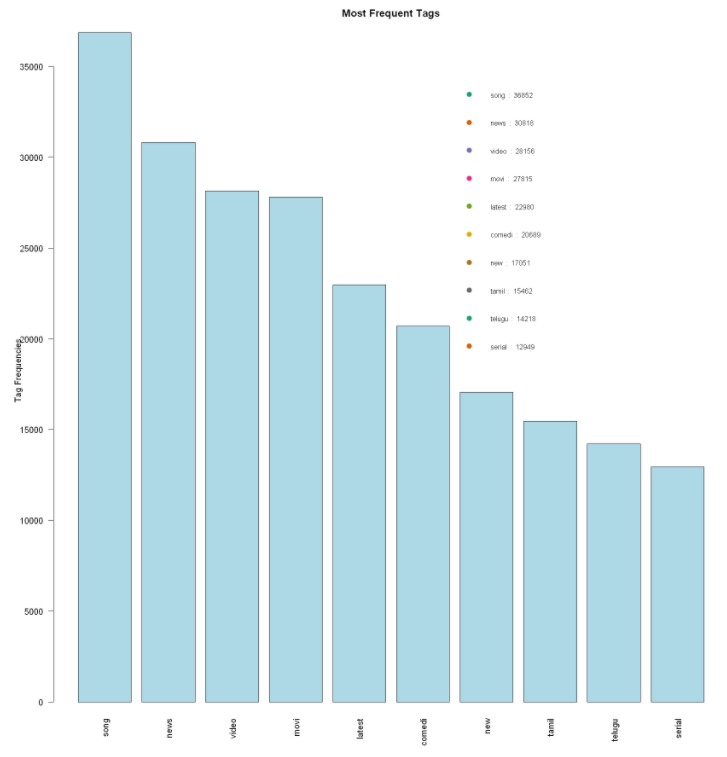
col ="lightblue", main ="Most Frequent Tags",

ylab = "Tag Frequencies")

legend("topright", legend = paste(d[1:10,]$word, " : ", d[1:10,]$freq) ,

col = brewer.pal(n = 10, name = "Dark2") ,

bty = "n", pch=20 , pt.cex = 2, cex = 0.8, horiz = FALSE, inset = c(0.05, 0.05))



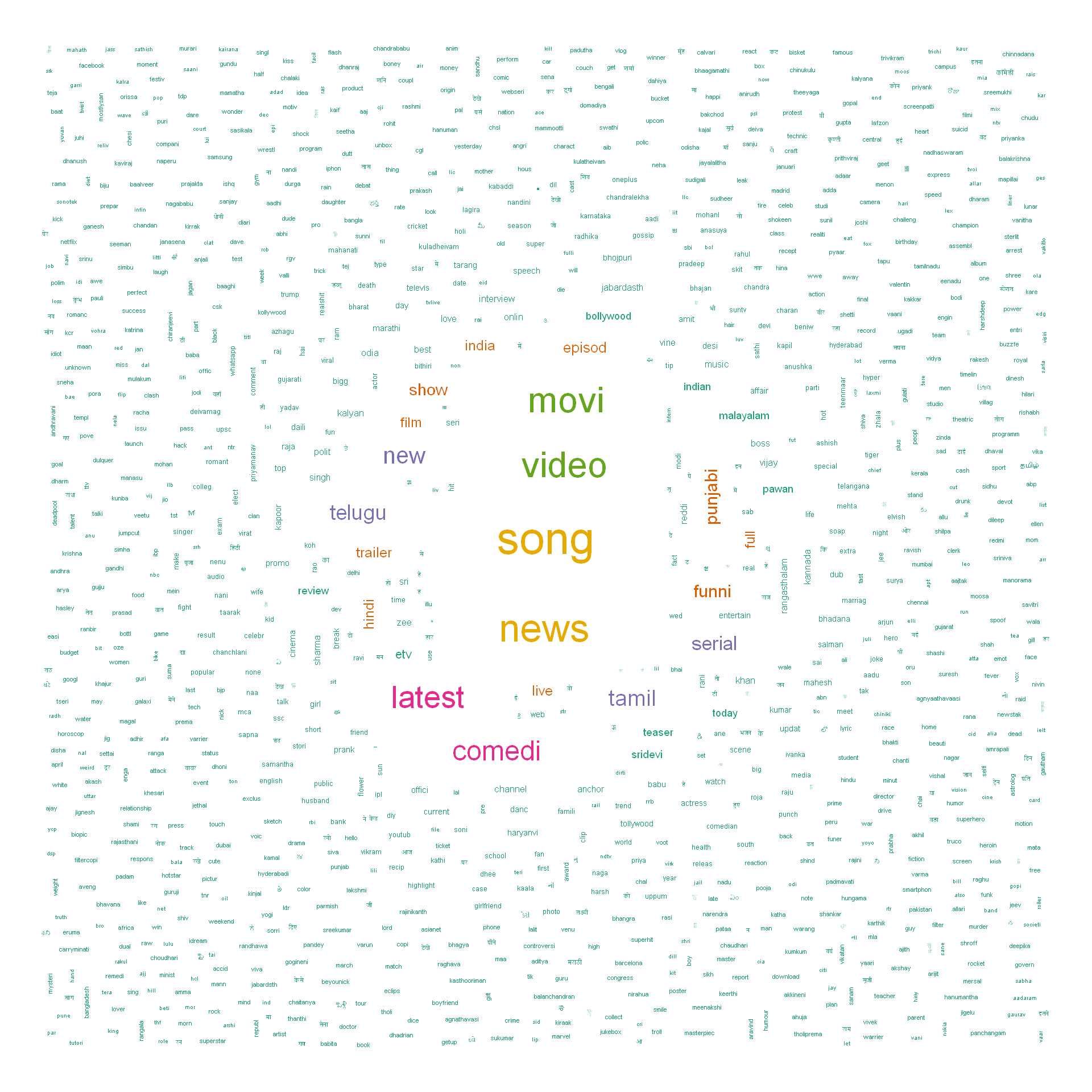
Here, we can most frequent song tag is used.

***WordCloud of Tag***

A wordcloud is a novelty visual representation of text data, typically used to depict keyword metadata (tags) or to visualize free form text. Tags are usually single words, and the importance of each tag is shown with font size or color. Below is wordcloud of tags (in india).

#Tags wordcloud

wordcloud(d$word,d$freq, min.freq=5,colors=brewer.pal(6,"Dark2"),random.order = F)

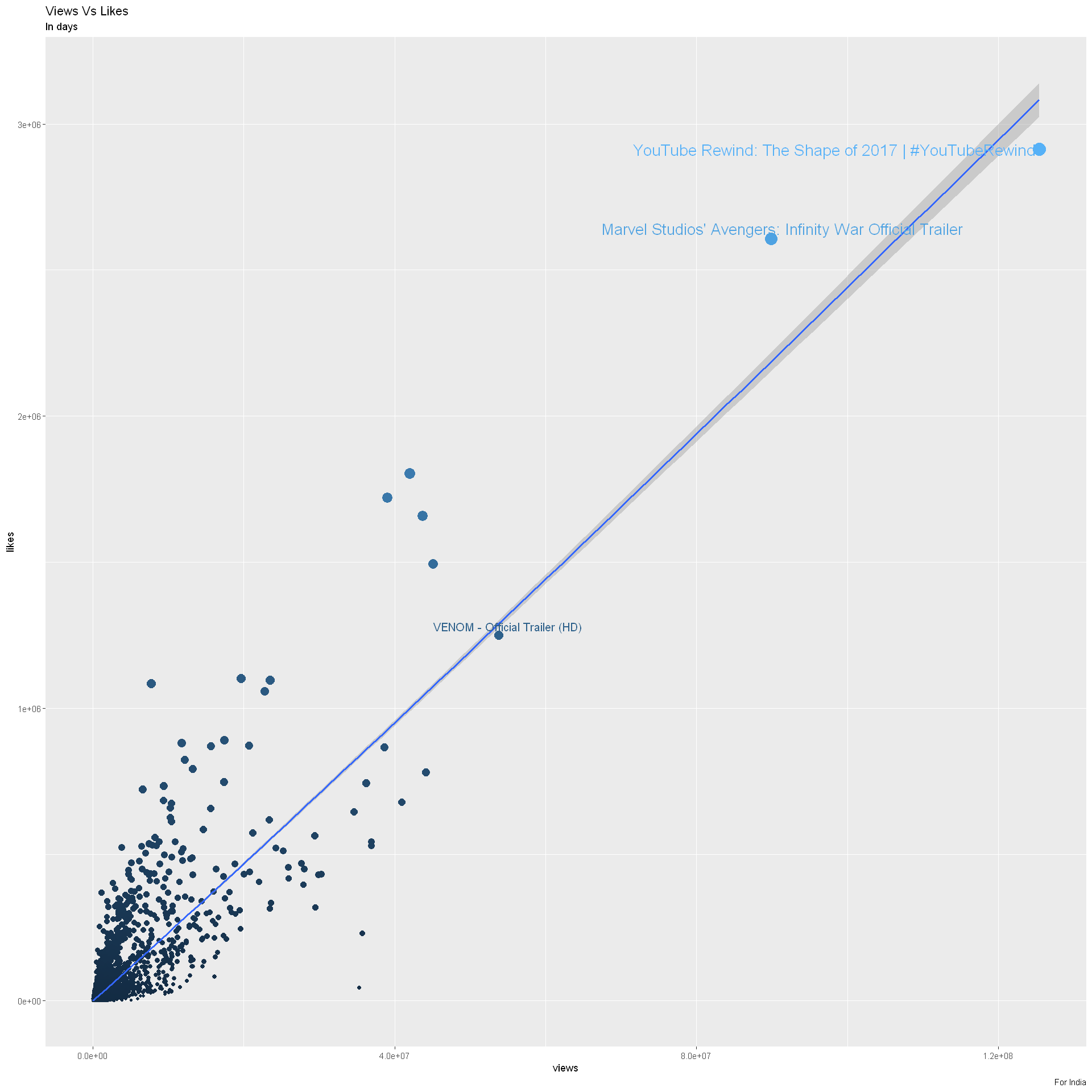
Here, we can see mostly news, song, video, movi, latest, etc such tag are trending frequently

***Views vs Likes***

##Views vs Likes

ggplot(df\_in[,.("views"=max(views),"likes"=max(likes)),by=title],aes(views,likes,colour=likes,size=likes))+geom\_jitter()+geom\_smooth()+guides(fill="none")+labs(caption="For India",title="Views Vs Likes",subtitle="In days")+theme(legend.position = "none")+geom\_text\_repel(data=subset(df\_in[,.("views"=max(views),"likes"=max(likes)),by=title], views > 5e+07),aes(views,likes,label=title),check\_overlap=T)

ggplot(df\_all[,.("views"=max(views),"likes"=max(likes)),by=title],aes(views,likes,colour=likes,size=likes))+geom\_jitter()+geom\_smooth()+guides(fill="none")+labs(caption="For countries",title="Views Vs Likes",subtitle="In days")+theme(legend.position = "none")+geom\_text\_repel(data=subset(df\_all[,.("views"=max(views),"likes"=max(likes)),by=title], views > 5e+07),aes(views,likes,label=title),check\_overlap=T)

******

***How much time passes between published video to get on trending?***

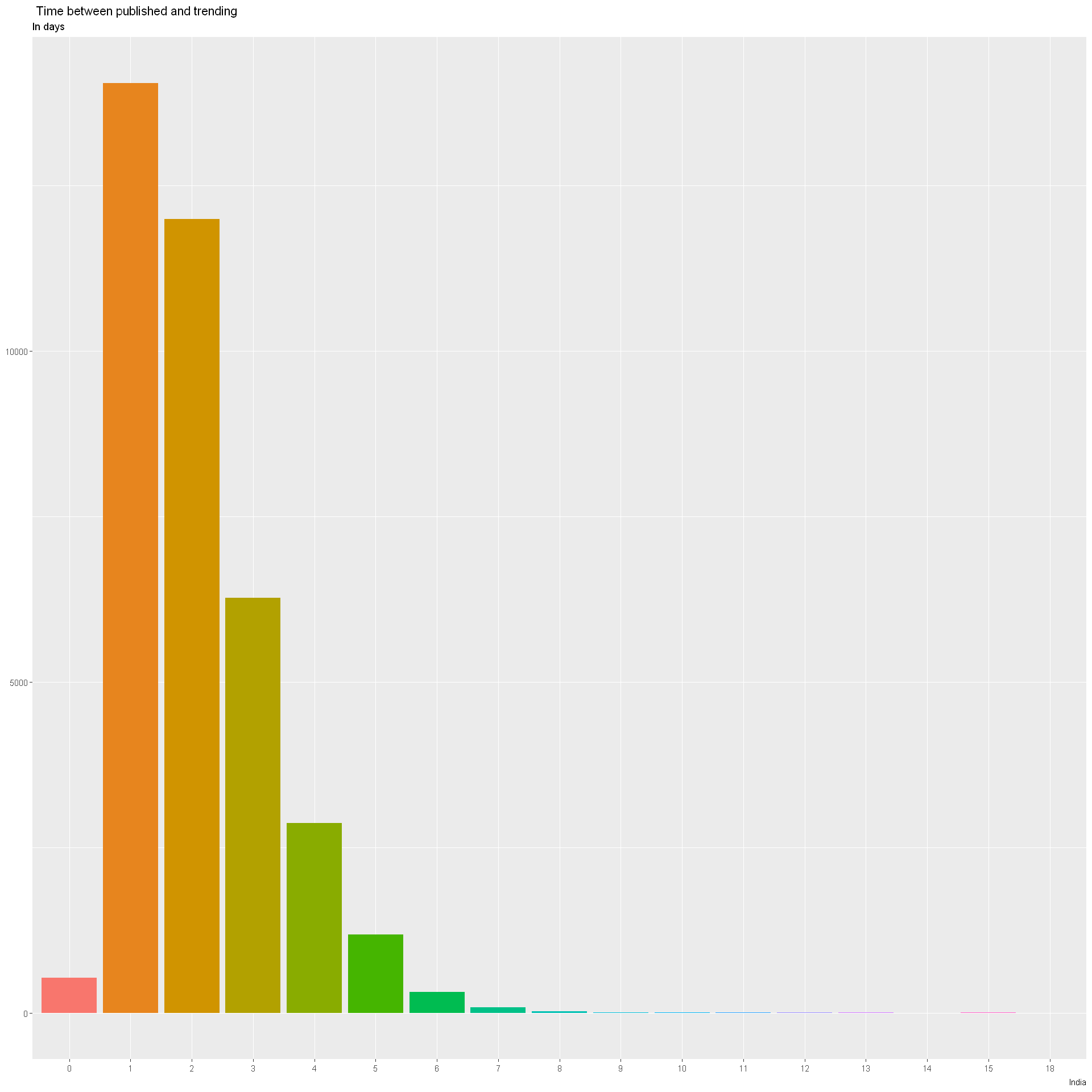
df\_in$trending\_date <- ydm(df\_in$trending\_date)

df\_in$publish\_time <- ymd(substr(df\_in$publish\_time,start = 1,stop = 10))

df\_in$dif\_days <- df\_in$trending\_date-df\_in$publish\_time

#How much time passes between published and trending? #ONLY FOR INDIA

ggplot(df\_in[dif\_days<30,],aes(as.factor(dif\_days),fill=as.factor(dif\_days)))+geom\_bar()+guides(fill="none")+labs(caption="India",title=" Time between published and trending",subtitle="In days")+xlab(NULL)+ylab(NULL)



Here, we can see mostly video comes in trending after one day after it is published and not on same day on it is published.

library(RSentiment)

corpus = Corpus(VectorSource(list(sample(df\_in$description,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))

sentiments <- as.data.table(sentiments\_eap)

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

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

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

***Sentiment Analysis on Description***

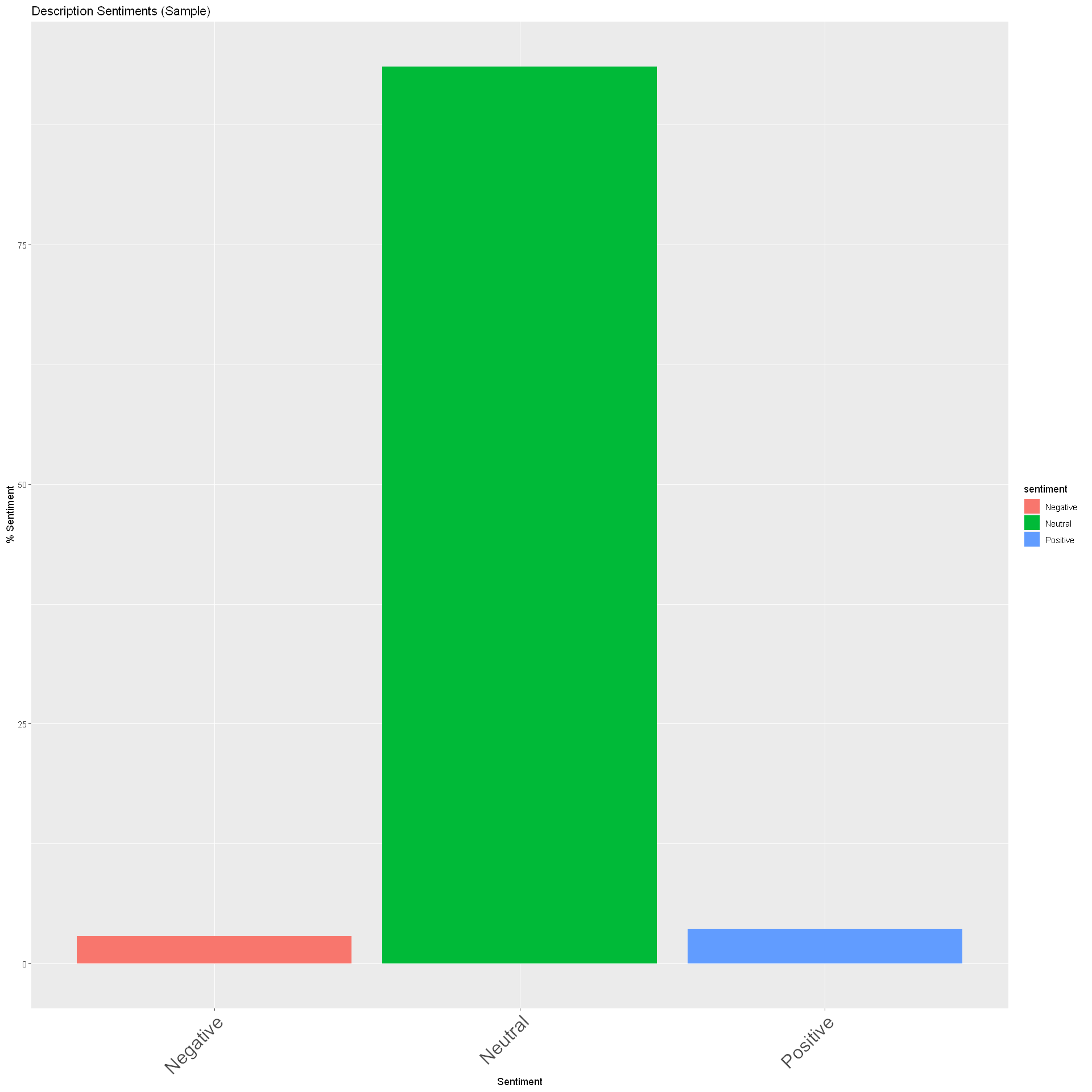
We are trying to find sentiment in description which tells us what type of video are trending.

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=20,hjust = 1))



Here we can see description is mostly Neutral.

Now will give score to description and try to find sentiment.

sents\_eap <- sentiment(df\_in$description)

sents\_eap <- sents\_eap[,.("word\_count"=sum(word\_count),"sentiment"=sum(sentiment)),by=element\_id]

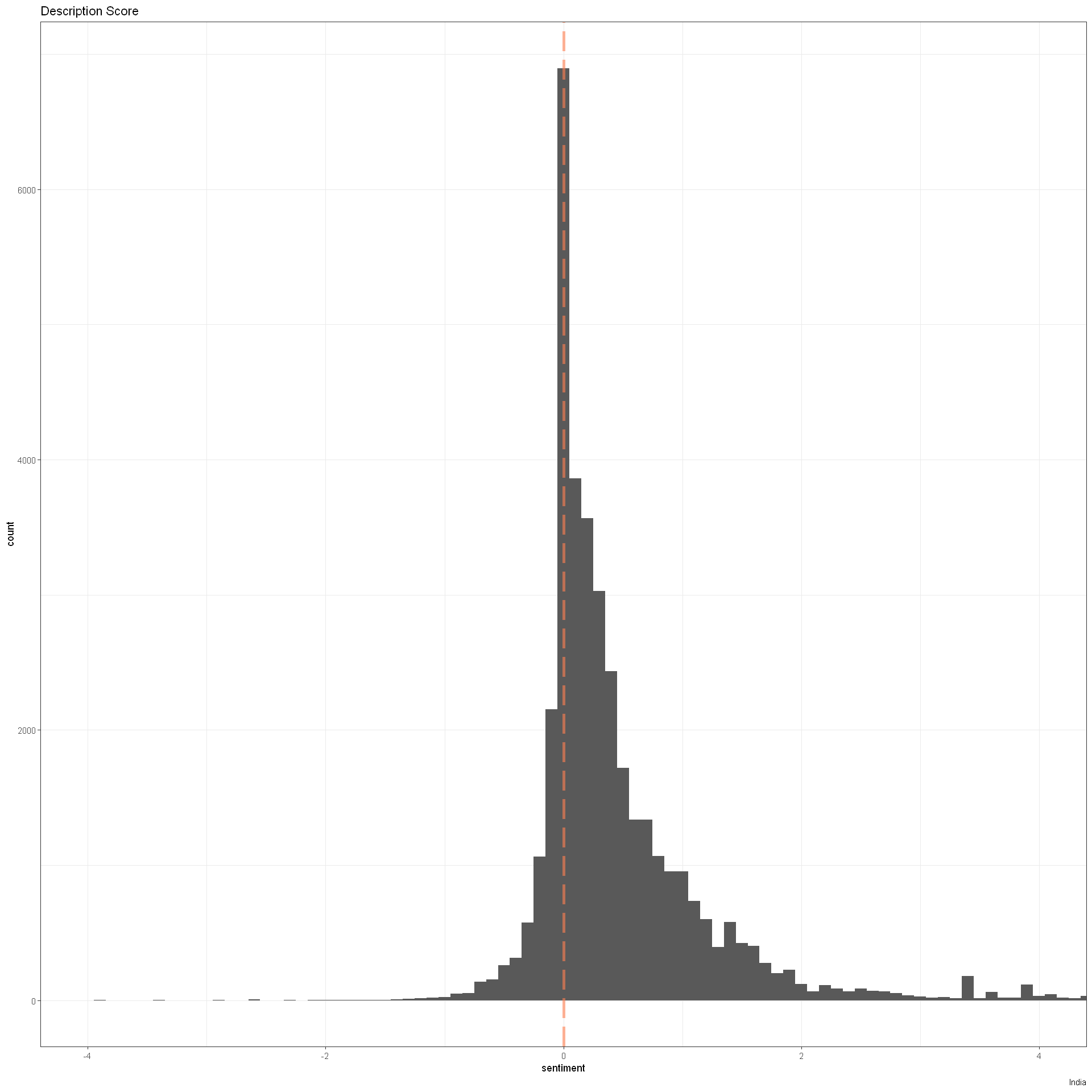
ggplot(data=sents\_eap)+

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",caption="India")+coord\_cartesian(xlim = c(-4, 4))



Clearly here we can see description is mostly positive than negative.

**Conclusion:**

1. “Song” tag videos are most frequent in trending section.
2. Trending videos has almost a linear relation in views and likes.
3. It takes more than a day to video get in trending section.
4. Positive Description let video to trend faster.