Youtube Analysis

David Medina Hernandez

5/11/2018

```r
library(syuzhet)
library(ggplot2)
library(tm)
library(wordcloud)
library(dplyr)
library(pcaPP)
```

```r
# EDA and Unsupervised Learning with youtube dataset from kaggle
# https://www.kaggle.com/datasnaek/youtube-new
utube_us <- read.csv("USvideos.csv",
                     encoding = "UTF-8", stringsAsFactors=FALSE,
                     na.strings=c("", "NA"))
```

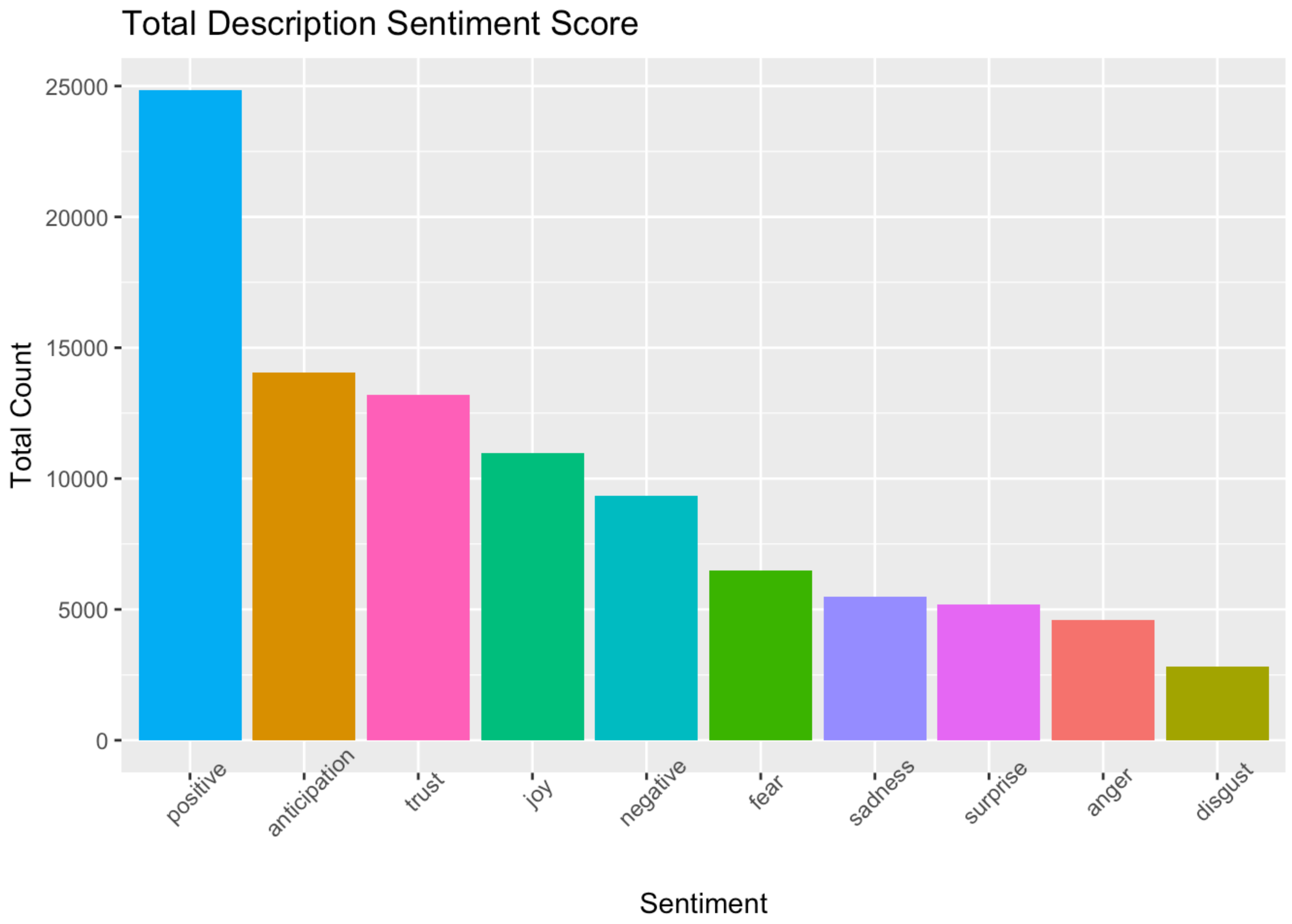
```r
utube_us$category_id <- factor(utube_us$category_id)
utube_us$video_id <- factor(utube_us$video_id)
utube_us$channel_title <- factor(utube_us$channel_title)
utube_us$comments_disabled <- factor(utube_us$comments_disabled)
utube_us$ratings_disabled <- factor(utube_us$ratings_disabled)
utube_us$video_error_or_removed <- factor(utube_us$video_error_or_removed)
utube_us$trending_date <- as.Date(utube_us$trending_date, format = '%y.%d.%m')
utube_us$publish_time <- as.Date(utube_us$publish_time, format = '%Y-%m-%d')
utube_us$pub_to_trend <- as.numeric(utube_us$trending_date - utube_us$publish_time)
```

```r
# description column has missing values
missing_per_col <- sapply(utube_us, function(x) sum(is.na(x)))
(total_missing <- sum(missing_per_col))
```

```
## [1] 448
```

```
# clean discription column
# exclude emojis
utube_us_nodup <- utube_us[!duplicated(utube_us$video_id), ]
utube_desc <- utube_us_nodup$description
utube_desc <- tolower(utube_desc)
# takes out "\\n"
utube_desc <- gsub("\\\\n", " ", utube_desc)
utube_desc <- gsub("http[^[:blank:]]+", "", utube_desc)
utube_desc <- gsub("www[^[:blank:]]+", "", utube_desc)
utube_desc <- gsub('[[:digit:]]+', "", utube_desc)
utube_desc <- gsub("[[:punct:]]+", "", utube_desc)
utube_desc <- gsub("\\s+"," ", utube_desc)
```

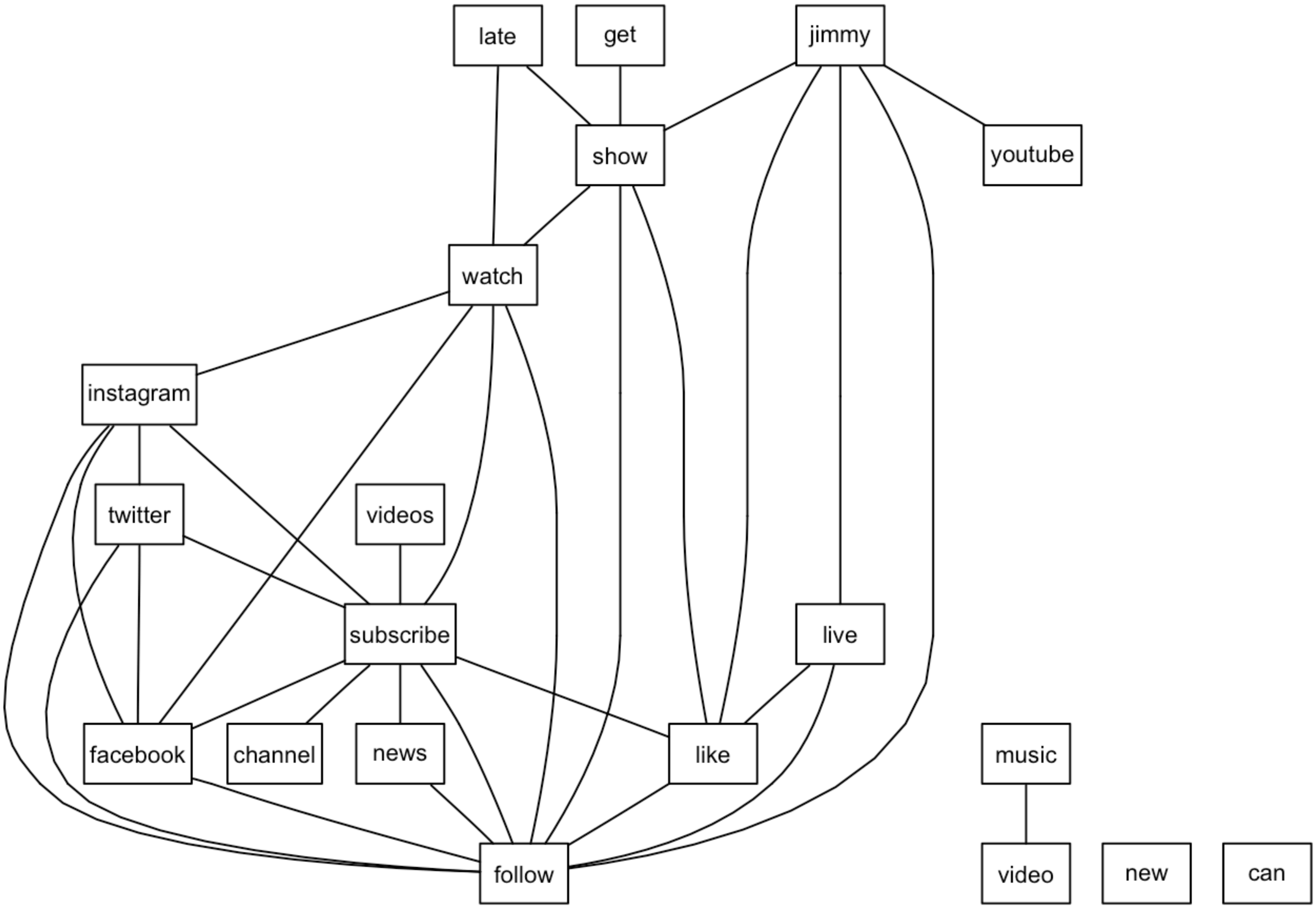
```
sentiment_desc <- get_nrc_sentiment(utube_desc)
sentiment_df_desc <- data.frame(feeling = names(colSums(sentiment_desc)), total = colSums(sentiment_desc), row.names = NULL)
ggplot(data = sentiment_df_desc,
       aes(x = reorder(feeling, -total, na.rm=TRUE), y = total)) +
  geom_bar(aes(fill = feeling), stat = "identity") +
  theme(legend.position = "none", axis.text.x = element_text(angle=45)) +
  xlab("Sentiment") + ylab("Total Count") + ggtitle("Total Description Sentiment Score")
```



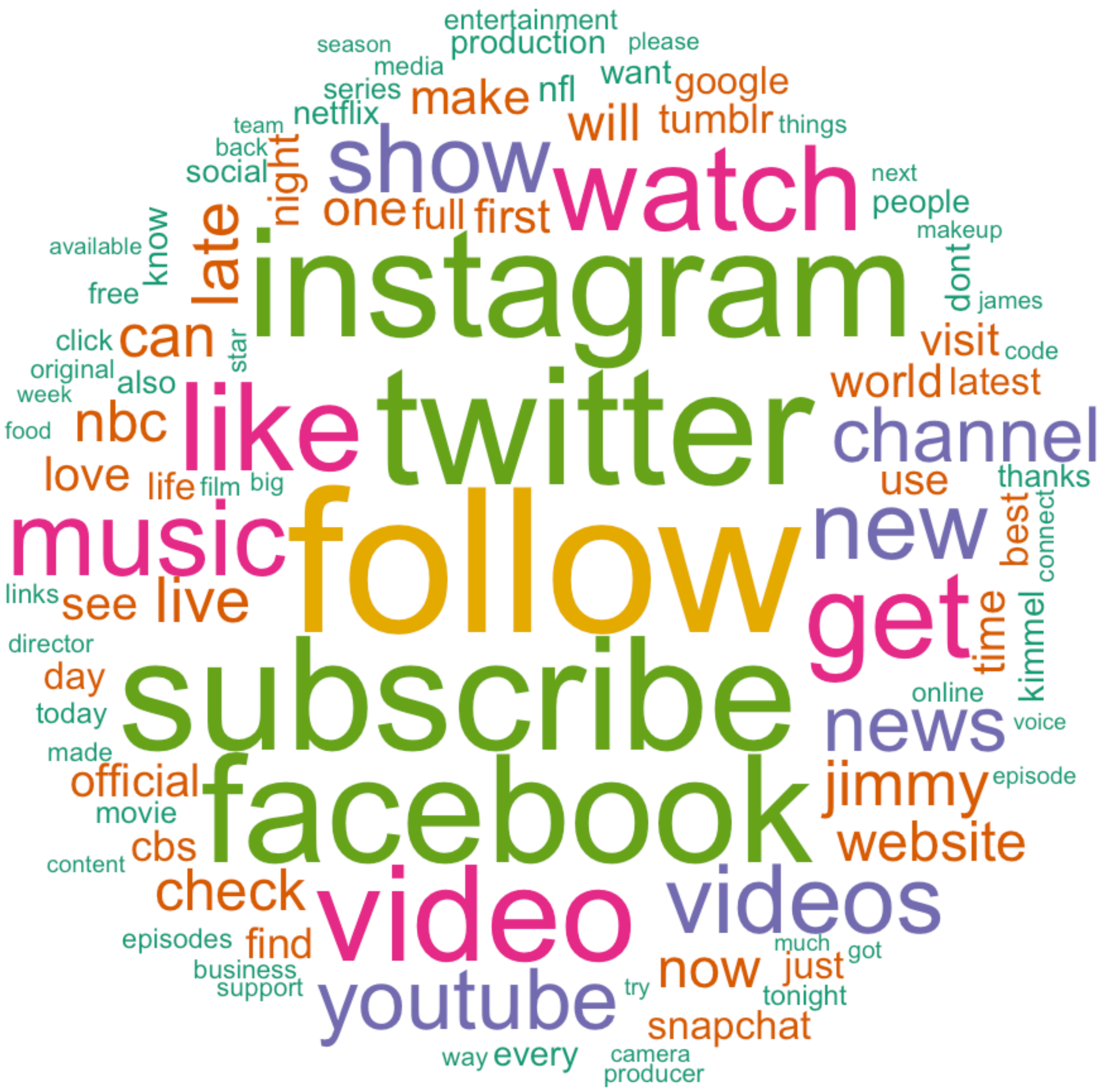
```
# text analysis
utube_desc_corpus <- Corpus(VectorSource(utube_desc))
corpus_desc <- tm_map(utube_desc_corpus, removeWords, stopwords("english"))
dtm_desc <- DocumentTermMatrix(corpus_desc)
dtm_mat_desc <- as.matrix(dtm_desc)
# most used words
freq_desc <- sort(colSums(dtm_mat_desc),decreasing = T)
freq_df_desc <- data.frame(word = names(freq_desc), freq = freq_desc, row.names = NUL
L)
head(freq_df_desc)
```

```
##        word freq
## 1    follow 3548
## 2   twitter 2874
## 3 subscribe 2770
## 4 instagram 2653
## 5  facebook 2631
## 6     video 2328
```

```
plot(dtm_desc, terms = findFreqTerms(dtm_desc, lowfreq = 1000),
     corThreshold = 0.25)
```

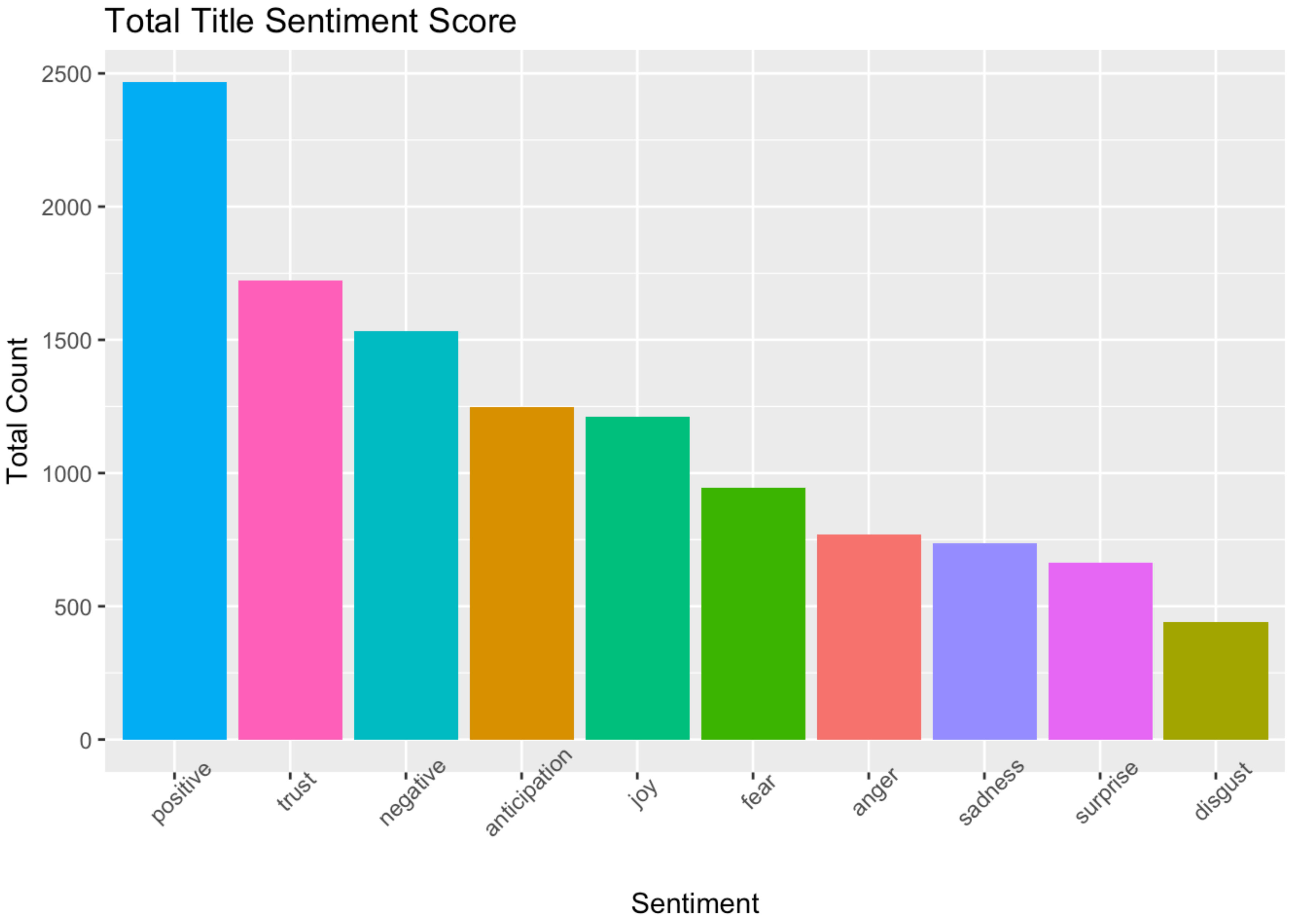


```
# word cloud
wordcloud(freq_df_desc$word, freq_df_desc$freq, max.words = 100, random.order = FALSE
,
          colors = brewer.pal(6, "Dark2"), scale = c(5, .1))
```



```
# clean title column
utube_title <- utube_us_nodup$title
utube_title <- tolower(utube_title)
utube_title <- gsub("\\\\n", " ", utube_title)
utube_title <- gsub("http[^[:blank:]]+", "", utube_title)
utube_title <- gsub("www[^[:blank:]]+", "", utube_title)
utube_title <- gsub('[[:digit:]]+', "", utube_title)
utube_title <- gsub("[[:punct:]]+", "", utube_title)
utube_title <- gsub("\\s+"," ", utube_title)
```

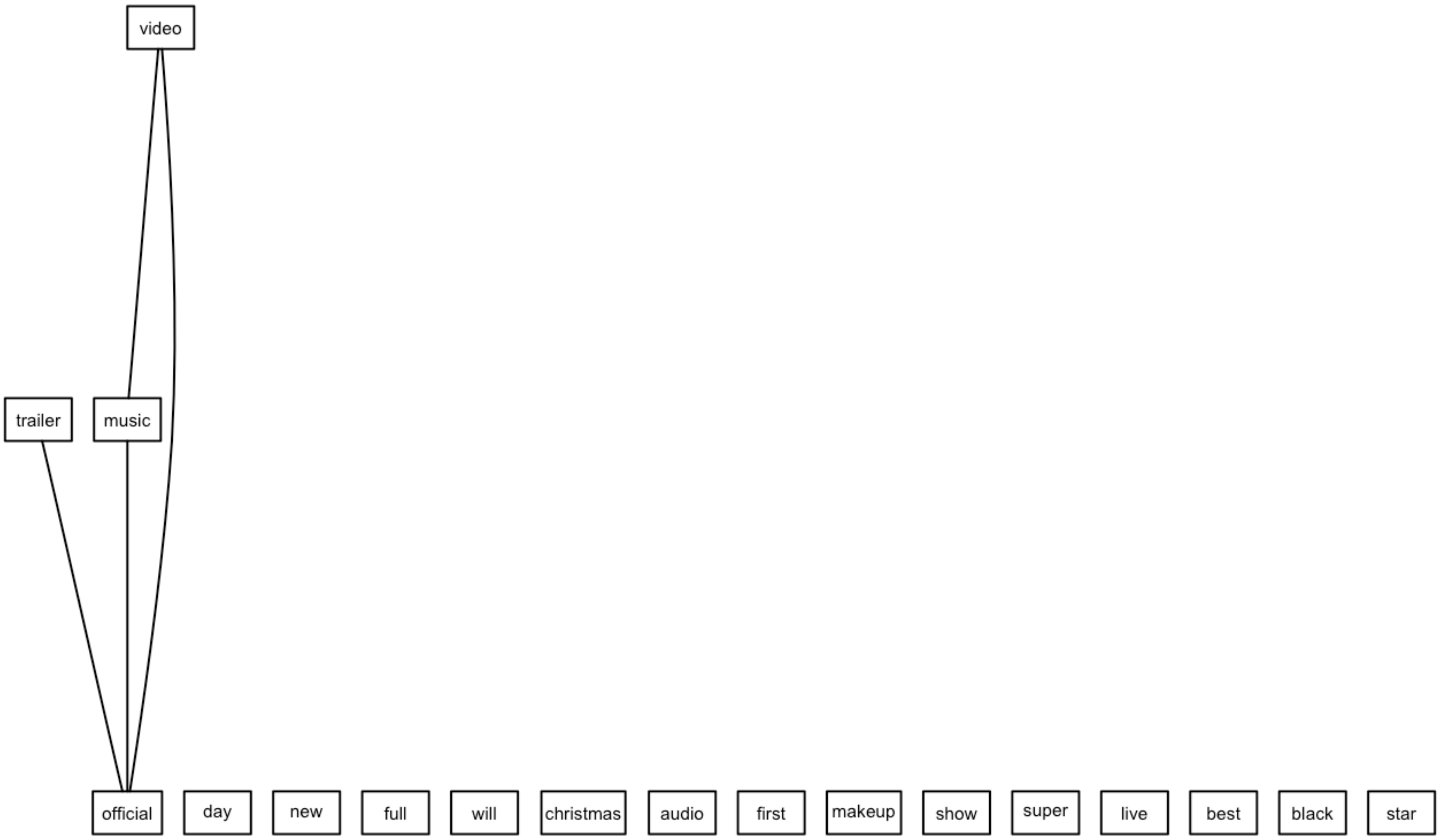
```
# sentient analysis
sentiment_title <- get_nrc_sentiment(utube_title)
sentiment_df_title <- data.frame(feeling = names(colSums(sentiment_title)), total = c
olSums(sentiment_title), row.names = NULL)
ggplot(data = sentiment_df_title, aes(x = reorder(feeling, -total, na.rm=TRUE), y = t
otal)) +
  geom_bar(aes(fill = feeling), stat = "identity") +
  theme(legend.position = "none", axis.text.x = element_text(angle=45)) +
  xlab("Sentiment") + ylab("Total Count") + ggtitle("Total Title Sentiment Score ")
```



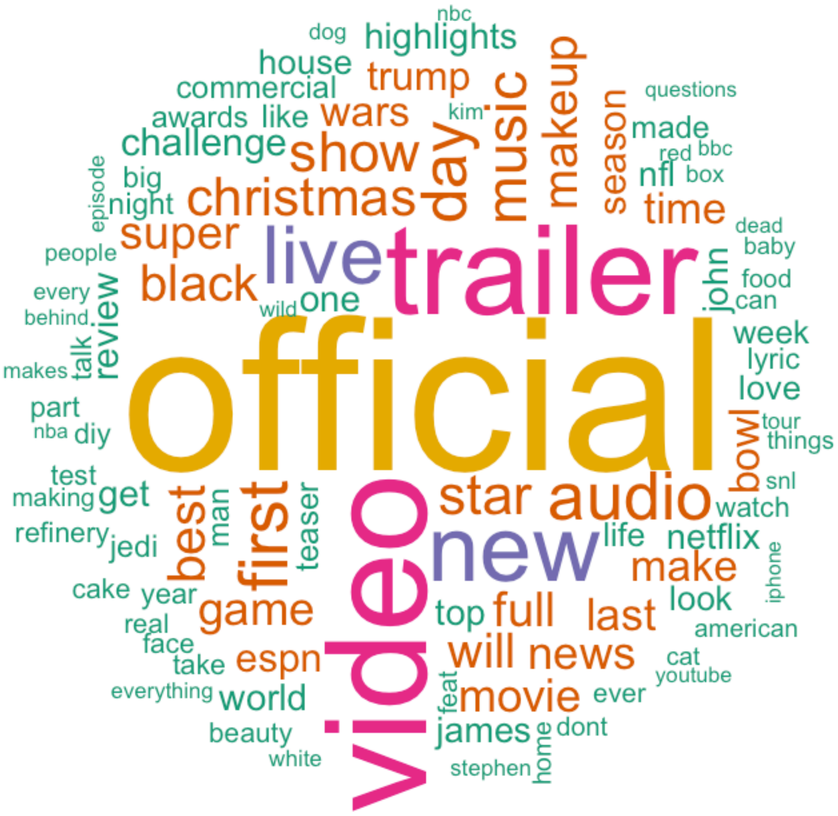
```
# text analysis
utube_title_corpus <- Corpus(VectorSource(utube_title))
corpus_title <- tm_map(utube_title_corpus, removeWords, stopwords("english"))
dtm_title <- DocumentTermMatrix(corpus_title)
dtm_mat_title <- as.matrix(dtm_title)
rownames(dtm_mat_title) <- utube_us_nodup$channel_title
freq_title <- sort(colSums(dtm_mat_title),decreasing = T)
freq_df_title <- data.frame(word = names(freq_title),
                            freq = freq_title, row.names = NULL)
head(freq_df_title)
```

```
##       word freq
## 1 official  338
## 2    video  222
## 3  trailer  195
## 4      new  145
## 5     live  124
## 6    first  104
```

```
# correlation between words
plot(dtm_title, terms = findFreqTerms(dtm_title, lowfreq = 70),
     corThreshold = 0.20)
```



```
# word cloud
wordcloud(freq_df_title$word, freq_df_title$freq, max.words = 100, random.order = FAL
SE,
          colors = brewer.pal(6, "Dark2"), scale = c(5, .1))
```

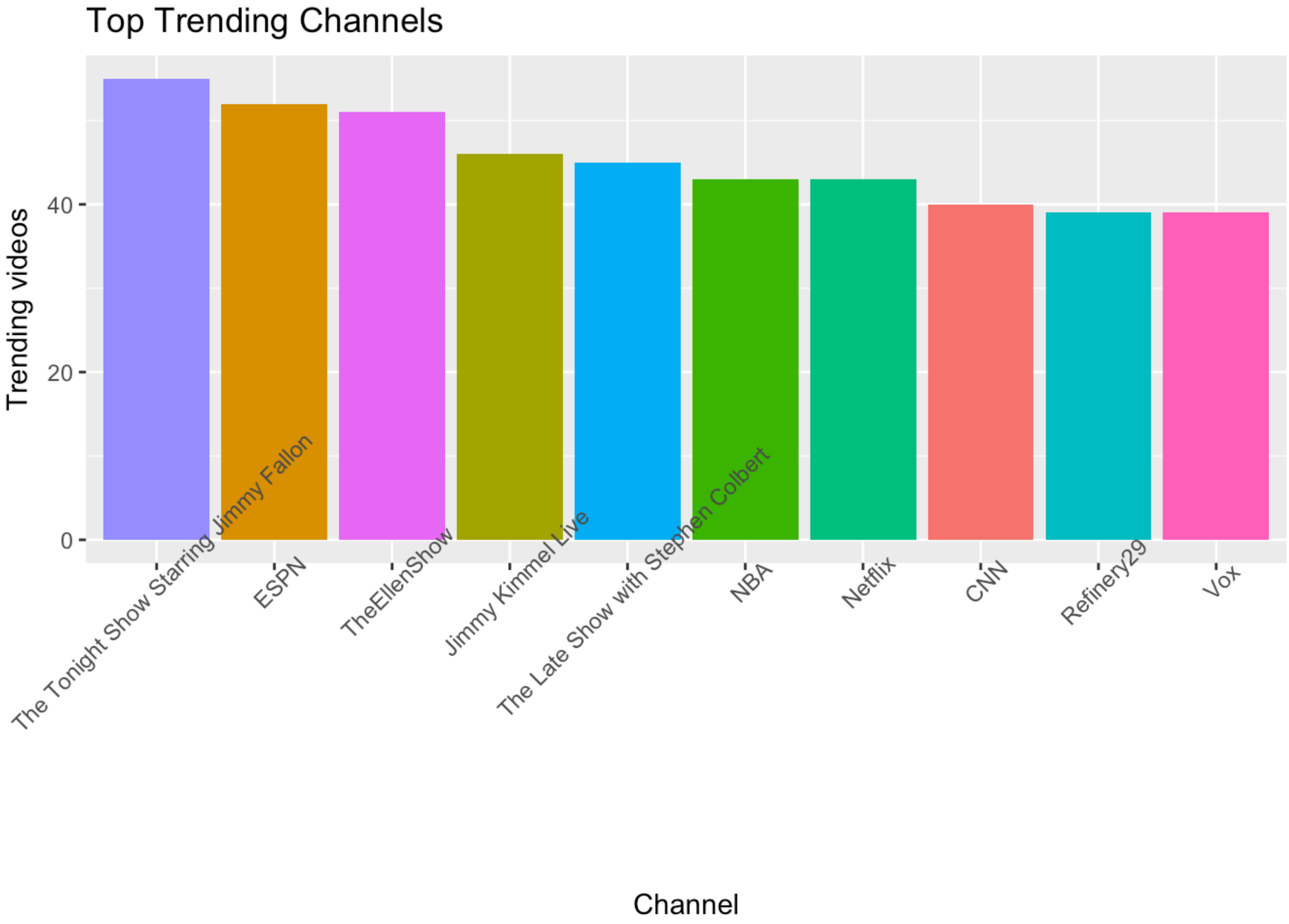


```
# kmeans with title words
# words associated with each other
set.seed(12345)
km_out <- kmeans(dtm_mat_title, centers = 4)
colnames(km_out$centers) <- colnames(dtm_mat_title)
names(head(sort(km_out$centers[3,], decreasing = TRUE), 16))
```

```
##  [1] "official"  "video"     "trailer"   "music"     "audio"
##  [6] "netflix"   "teaser"    "season"    "feat"      "lyric"
## [11] "black"     "hbo"       "christmas" "movie"     "theaters"
## [16] "mirror"
```

```r
# trending channels
library(dplyr)
trending_chan <- utube_us_nodup %>%
  group_by(channel_title) %>%
  summarise(num_trend_vids = length(channel_title)) %>%
  arrange(desc(num_trend_vids))

ggplot(data = trending_chan[1:10, ],
       aes(x = reorder(channel_title, -num_trend_vids, na.rm=TRUE),
           y = num_trend_vids, fill = channel_title)) +
  geom_bar(stat = "identity") +
  theme(legend.position="none",
        axis.text.x = element_text(angle=45)) +
  labs(x = "Channel", y = "Trending videos", title = "Top Trending Channels")
```



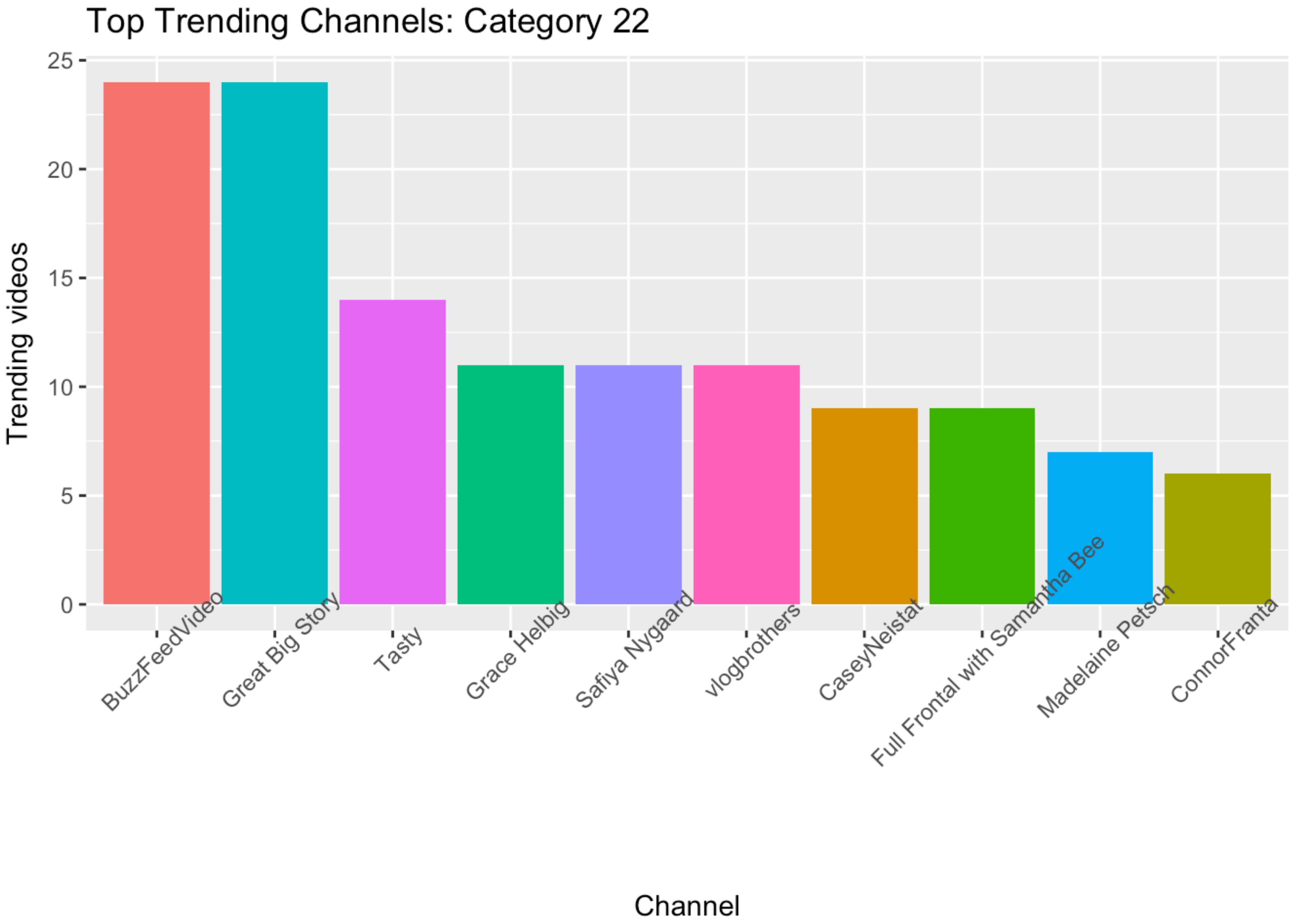
```r
# by category
trending_chan2 <- utube_us_nodup %>%
  group_by(category_id) %>%
  summarise(num_trend_vids = length(channel_title)) %>%
  arrange(desc(num_trend_vids))
head(trending_chan2)
```

```
## # A tibble: 6 x 2
##   category_id num_trend_vids
##   <fct>                <int>
## 1 24                    1231
## 2 10                     616
## 3 26                     468
## 4 25                     448
## 5 23                     422
## 6 22                     393
```

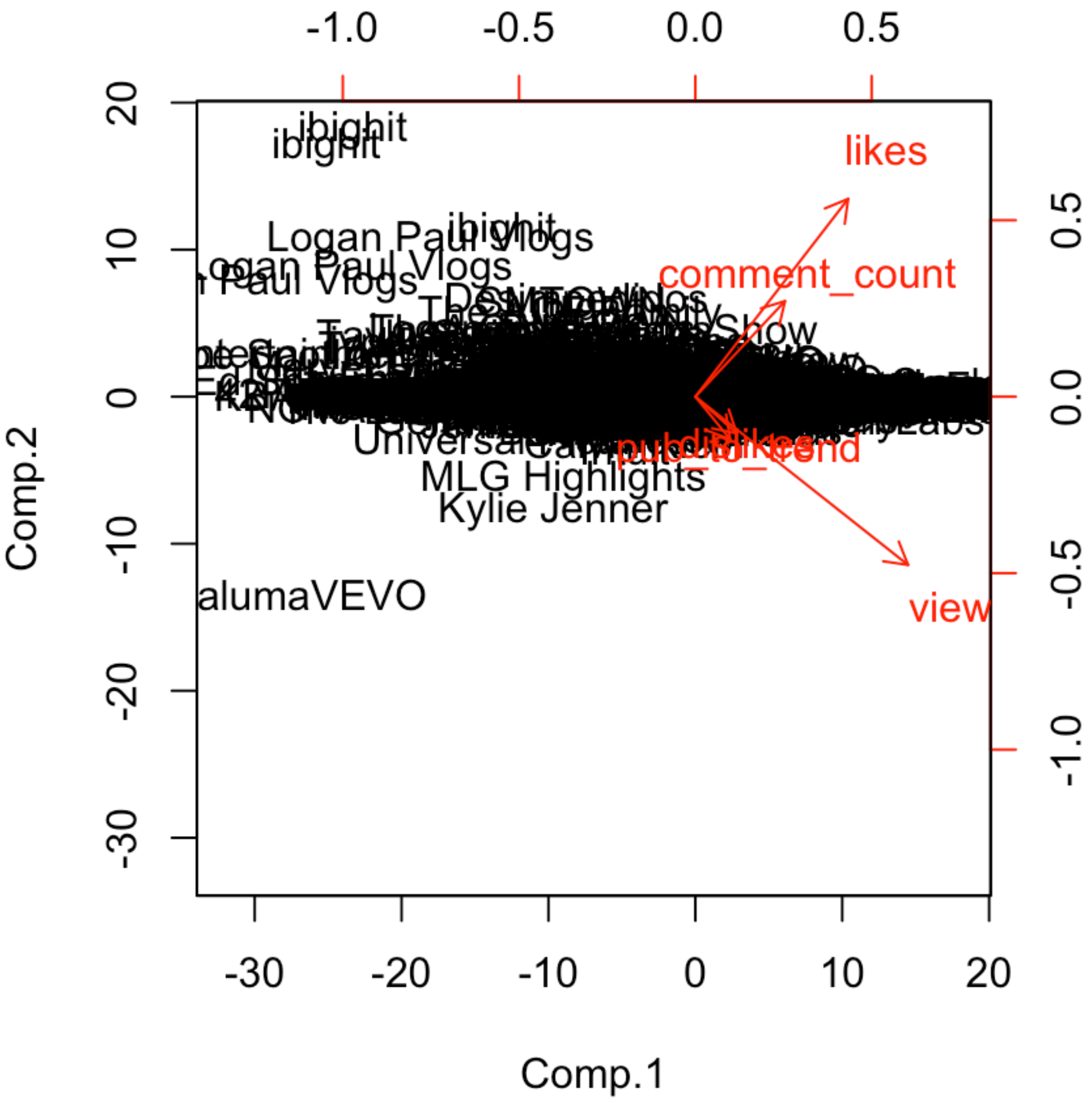
```
# channel by category
# dataframe that contains channel names and category ids
us_nodup_category <- data_frame(channel_title = utube_us_nodup$channel_title,
                                category_id = utube_us_nodup$category_id)

trending_chan3 <- merge(trending_chan, us_nodup_category, by = "channel_title")
trending_chan3 <- trending_chan3[!duplicated(trending_chan3$channel_title), ]
trending_chan3 <- arrange(trending_chan3, desc(trending_chan3$num_trend_vids))

ggplot(data = filter(trending_chan3, category_id == 22)[1:10, ],
       aes(x = reorder(channel_title, -num_trend_vids, na.rm=TRUE),
           y = num_trend_vids, fill = channel_title)) +
  geom_bar(stat = "identity") +
  theme(legend.position="none",
        axis.text.x = element_text(angle=45)) +
  labs(x = "Channel", y = "Trending videos",
       title = "Top Trending Channels: Category 22")
```



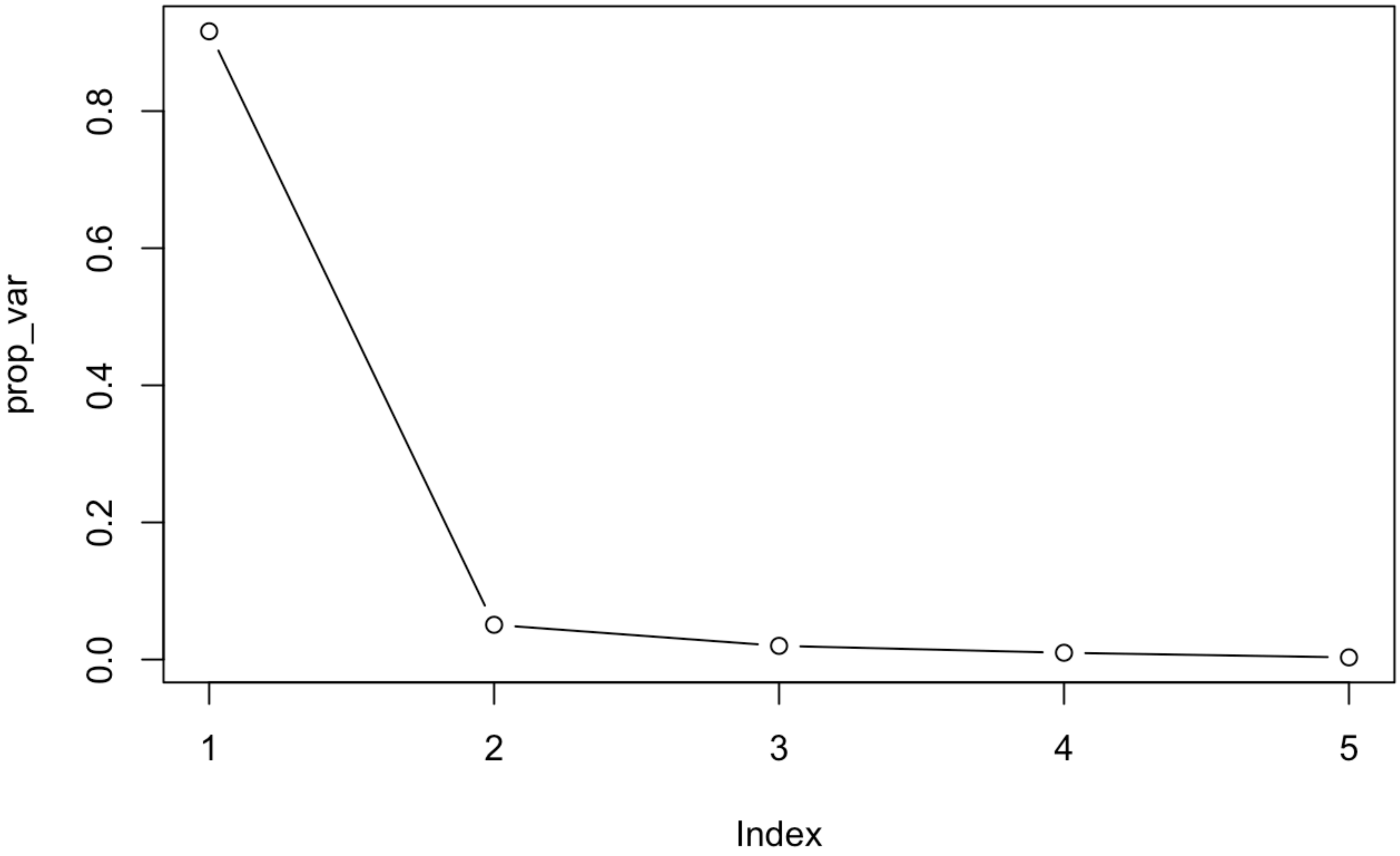
```
# PCA
# channels associated with each other
nodup_numeric <-select_if(utube_us_nodup, is.numeric)
pr_out <- PCAproj(nodup_numeric, scale = sd, k = 5)
rownames(pr_out$scores) <- utube_us_nodup$channel_title
biplot(pr_out, scale = 0)
```



```
pr_out$loadings
```

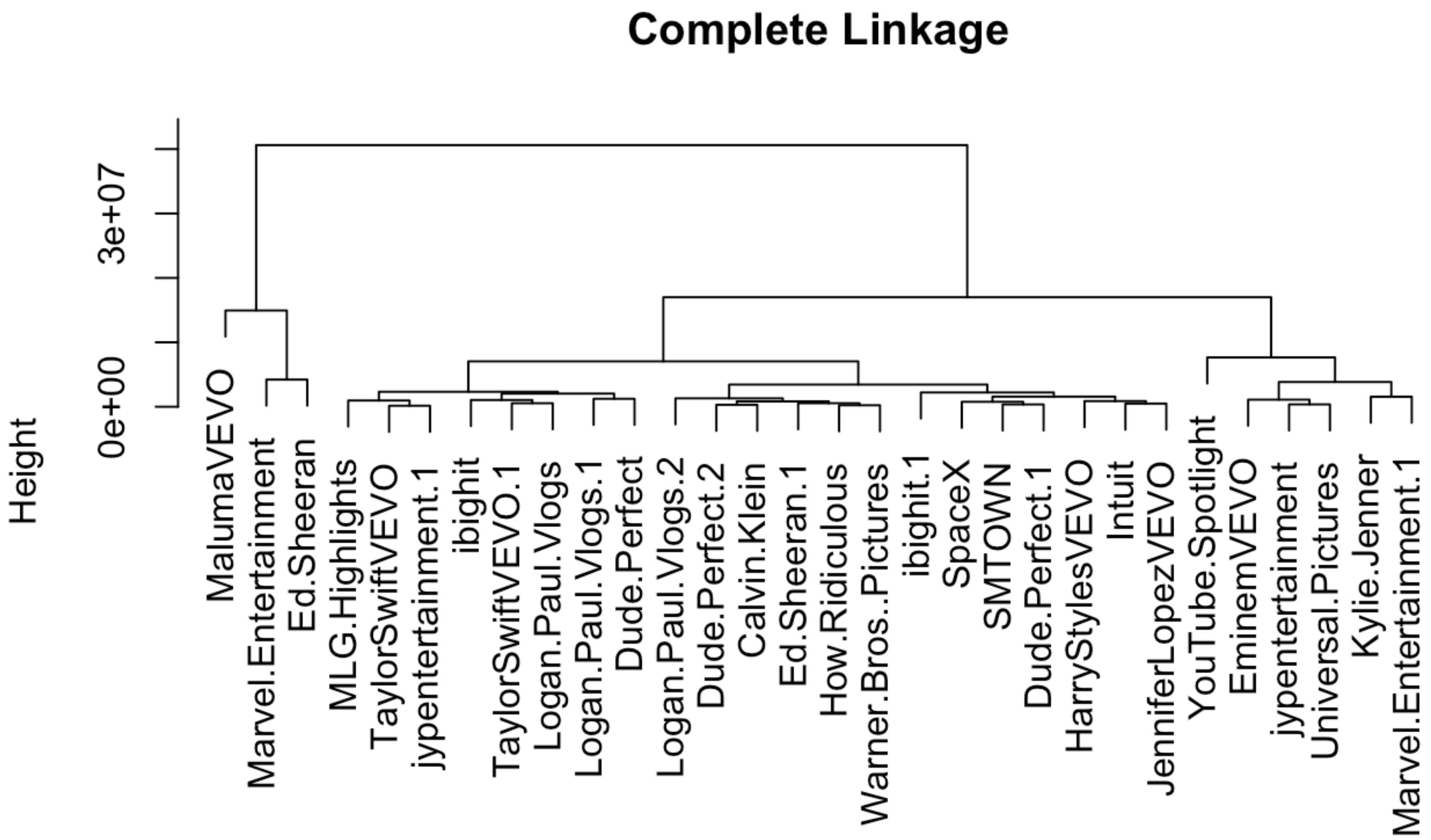
```
##
## Loadings:
##               Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## views          0.754 -0.596 -0.212 -0.134 -0.116
## likes          0.542  0.701 -0.310  0.326  0.113
## dislikes       0.147 -0.130  0.355         0.912
## comment_count  0.318  0.338  0.652 -0.556 -0.223
## pub_to_trend   0.123 -0.151  0.554  0.751 -0.303
##
##                Comp.1 Comp.2 Comp.3 Comp.4 Comp.5
## SS loadings       1.0    1.0    1.0    1.0    1.0
## Proportion Var    0.2    0.2    0.2    0.2    0.2
## Cumulative Var    0.2    0.4    0.6    0.8    1.0
```

```
prop_var <- (pr_out$sdev ^ 2) / (sum(pr_out$sdev ^ 2))
plot(prop_var, type='b')
```



```
# hierarchical clustering
# so that the channel names show up in the plot instead of numbers
top_views <- nodup_numeric %>%
  mutate(channel_title = utube_us_nodup$channel_title) %>%
  arrange(desc(views))
rownames(top_views) <- make.names(top_views$channel_title, unique = TRUE)
top_views <- select(top_views, -channel_title)

hc_complete <- hclust(dist(top_views[1:30, ]), method = "complete")
plot(hc_complete, main = "Complete Linkage", xlab = "", sub = "")
```



```
which(cutree(hc_complete, 4) == 3)
```

```
##      YouTube.Spotlight           Kylie.Jenner Marvel.Entertainment.1
##                      4                      5                      6
##       jypentertainment     Universal.Pictures             EminemVEVO
##                      7                      8                      9
```