Trending Youtube Statistics

Xinyi Wang / Jennie

12/15/2019

The potential reader for this paper towards who is interested in content strategy through contemporary social media.

My research question:

Which kind of video is the most popular program in each country? Do they have some common preferences?

The purpose & interests

This visualization is created based on my personal interests. I'm enjoying browsing the recommended Youtube Videos because the algorithm knows my interests very well. It recommends videos by my viewing history. From that point, I start to be curious about the others' interests based on their Youtube Dataset.

In this project, I will explore the most popular videos in other countries. My goal is using Rstudio to grape detailed key information as much as possible and visualize it. I selected three countries, France, Germany, and the UK because those three countries have a similar population that would make my results more reliable.

Implications & Importance

People's preference is influenced by their country deeply. Their habits, interests can reflect the condition of their country. As the development of social media increasing incredibly, people start to use Youtube a lot to watch videos instead of television shows. In this sense, tracing their data is a direct way to know their thoughts. A country is built by the government, culture, industry, etc. It's not easy to know the real condition by chasing the news only but trending data is pretty real. People only spend their time on their interests. Therefore, trending YouTube data could tell the users' characteristics of a certain country. It's important to their cultural development.

A description of the dataset

This data is from Youtube API. https://www.kaggle.com/datasnaek/youtube-new

Variables:

video_id / trending_date / title / channel_title / category_id / publish_time / tags / views / likes / dislikes / comment_count / thumbnail_link / comments_disabled / ratings_disabled / video_error_or_removed / description.

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

Limitation:

The biggest limitation of this dataset is it doesn't include the complete dataset by years. Because of that limitation, I can't make a prediction trending by years. It also doesn't have all the countries. In order to find answers to my question, I will choose countries considerably. For instance, 1000 people like music videos in India have a different meaning with 1000 people like music videos in Japan. Therefore, according to the population all over the world, I choose UK (0.857%), Germany (1.07%) and France (0.865%) as a group to analyze.

Methods

> **Project Proposal feedback:**
> "I'm still not quite sure what you plan to do with the data - that is, what your methods will be. You've defined what each of the methods means, but not how you will apply these methods to this data, or what each method will tell you about your data. What variable will you apply confidence intervals and why - what will that tell you about your project? Why are you doing text analysis? What are you looking for? How will it help you answer your research questions? Please revise your methods section with more details about *how* you will use each method to answer your research questions."

My database has two forms to represent variables, numbers, and letters. For this reason, I divided my methods into two parts. The first part is the visual charts and the other is text analysis. The number shows the total count of 'likes', 'dislike', 'view', 'comments'. The best way to analyze is by using 'ggthemes' package to make charts. The other is titles. The titles are related to catalog that reveals people's concern on Youtube videos. Doing text analysis of titles can provide more detailed information on popularity videos than the catalog.

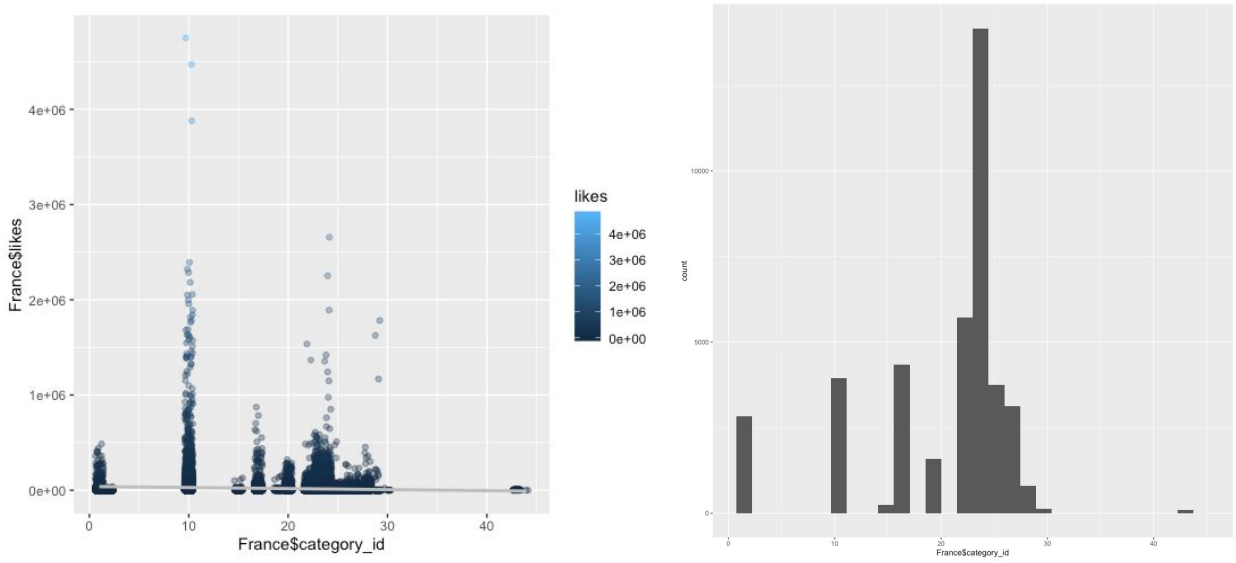
First, I use 'tidyverse' to clean my data. After I import the CSV file. I select the column by '$' and unify words by 'tolower' function. Then use 'removePunctuation' and 'stripWhitespace' to remove the useless space. I also need to remove stopwords by 'removeWords' function. 'Stopwords' means the words don't contain specific meanings, like 'on', 'and', etc. There is a tricky detail that different languages using different words represent 'stopwords'. For this project, English's 'stopwords' is 'en'; French's 'stopwords' is 'fr'; German's 'stopwords' is 'de'. Finally, I use 'tibble' to count frequent words on the title. 'Table' is also a good function to list data based on its value. Finally, export the CSV file by 'write.table'. Plus, 'Merge' is a useful function to combine 100 top frequent words of each country.

Before creating charts, I use ‘glimpse’ function to see how many columns and rows I have. Knowing data variables helps me to think about the value of the X-axis and Y-axis. I installed ‘Ggplot’, it has many libraries to help me create charts. I usually use ‘geom_histogram’ and ‘ggplot’ library to create charts. I tried ‘data.frame’ function to display the common words of three countries. It’s too many on bar chart display so I put the CSV file into Tableau tree chart because it’s better to show 100 words frequency in one square.

One obstacle that I met is I can’t make a prediction of the YouTube video trending preference because of the lack of data by years. I made a regression of catalog variables and the results totally didn’t make any sense. I gave up this direction.

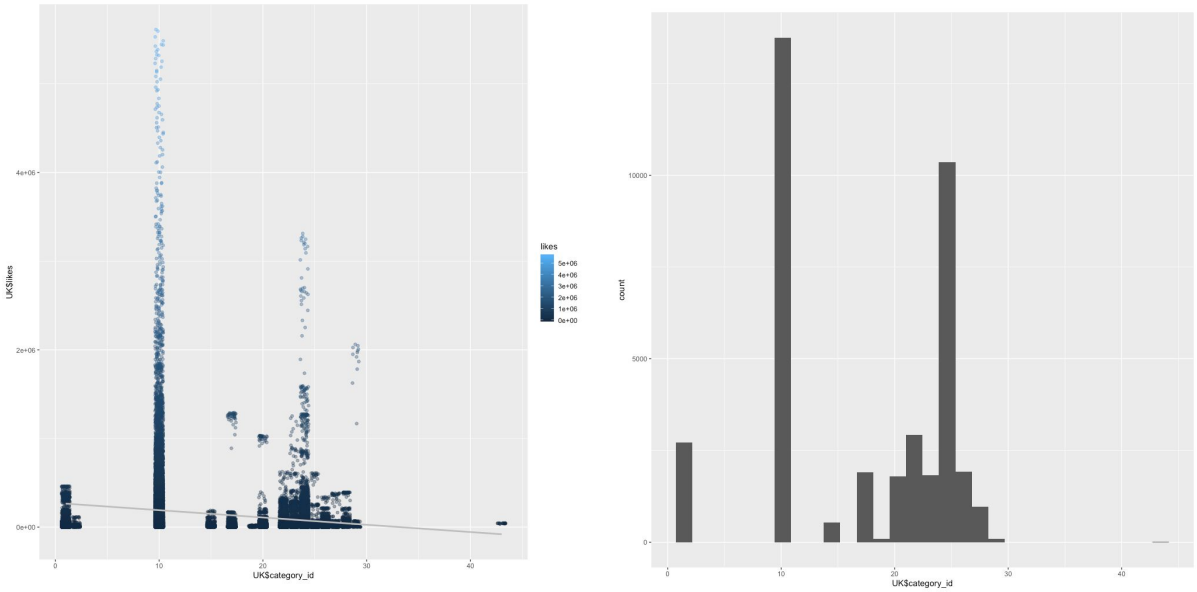
Results

France



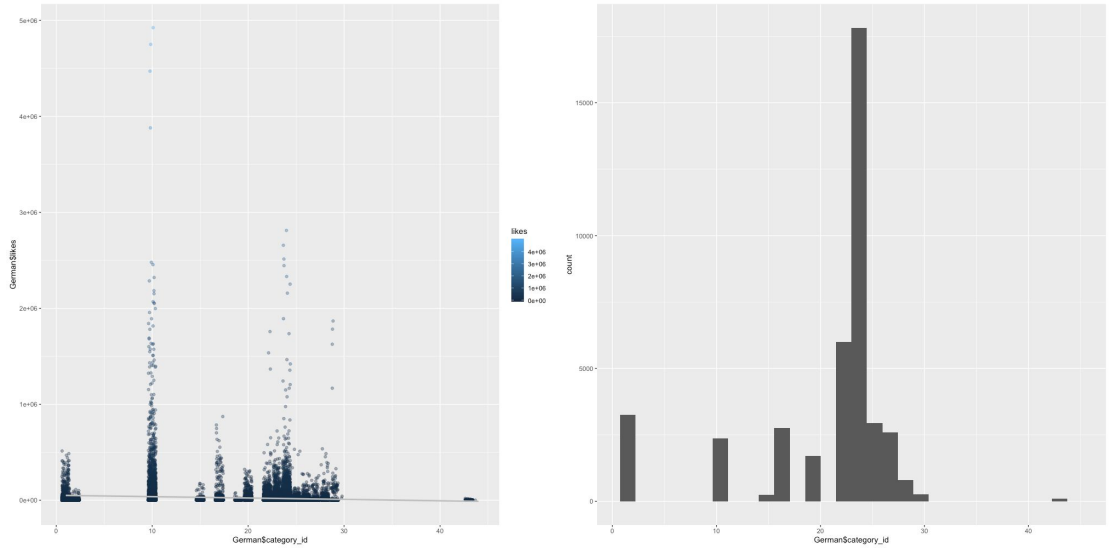
Overall Comedy videos (catalog ID: 23) have the largest number of the whole trending videos in France. The second large count is entertainment(catalog ID: 24). Music videos got the most satisfying rate than others. Even french people watch entertainment videos a lot, they didn’t click on ‘like’ button that much.

The United Kingdom

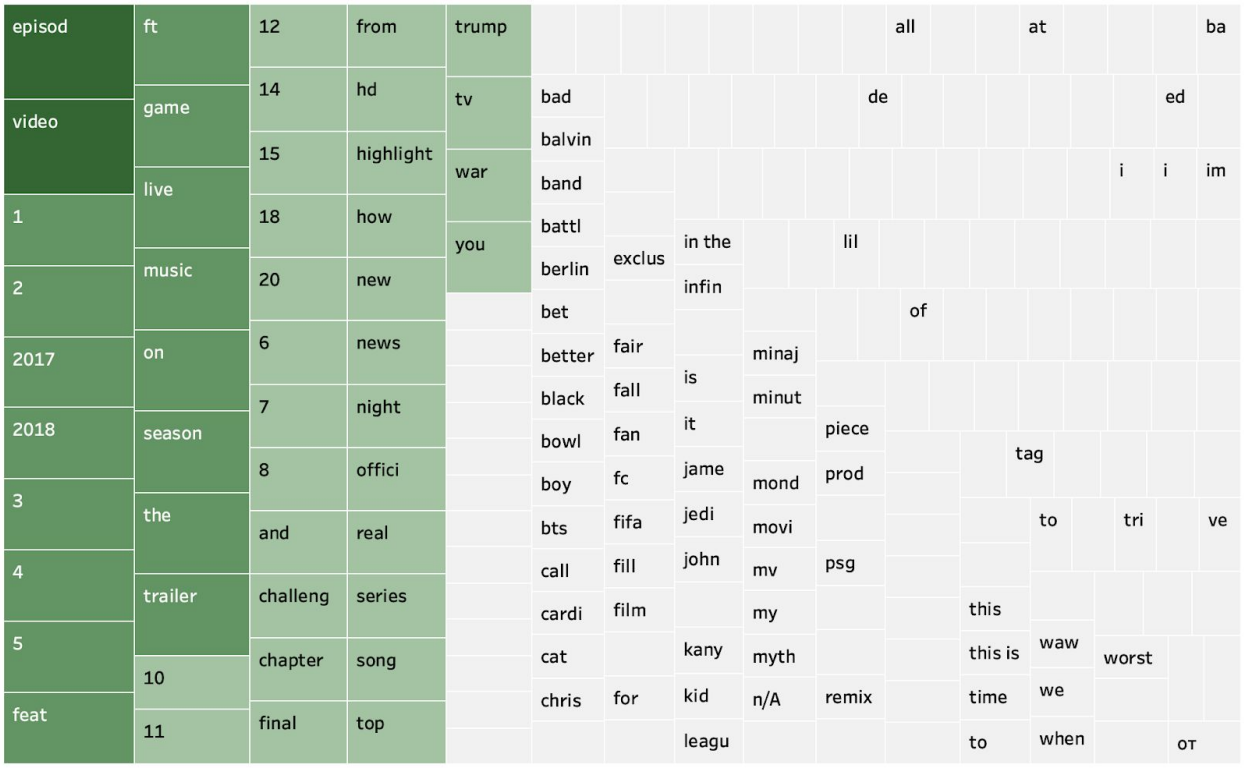


From the ‘ggplot’ chart, I can know that England people love music a lot. It has the largest number of comment counts and likes. Same as France’s situation, entertainment(catalog ID: 24) is the second popular catalog on ‘ggplot’ charts.

Germany



People like entertainment more than music. Overall, entertainment (catalog ID: 24) video is the No.1 for likes, and views. One more interesting finding is german also enjoys music (catalog ID: 10). They didn't have the strongest preference like England people love music more than other catalogs.



I use R to calculate the overlap number of the top 100 frequent titles and put them into the tree chart. The four colors represent how many times it appears in the CSV file. The degree is from four to one. If the words appear three times, it means people in three countries both like them. 'Episod' and 'video' have a four times appearance. '1, 2, 2017, 2018, 3, 4, 5, feat, ft, game, live, music, on, season, the, trailer' have three times appearance that means those words exist on each country's frequent words list. The words in rest light green blocks mean they are in the two countries' top title list. People make decisions with their knowledge and culture. It's interesting to find common humanity interests by arranging and visualizing their data.

Reflection

From the visual charts in three countries, entertainment and music are the most popular catalog. That means people using Youtube videos for having fun as the main purpose. I used to make a hypothesis that people's interests in Youtube video depending on their humanity preference more than cultural influence. It doesn't make sense, like the case of the UK. Obviously, music is quite a popular and long history in England so that many people love music. However, three countries, they both love music and entertainment with different degrees and they even have some common topic through all the trending videos. Plus, it's hard to define what is the 'humanity' preference. Overall, I finally answer my research question that I know each country's most popular catalog Youtube video and show the common title words in a treemap.

Appendix

Charts:

```
install.packages("ggthemes")
install.packages("gutenbergr")
install.packages("tm")
install.packages("topicmodels")

library(ggthemes)
library(tidyverse)
library(dplyr)
library(lubridate)

library(tidytext)
library(tm)
library(stringr)
library(topicmodels)
library(gutenbergr)
library(tidyverse)
library(dplyr)
library(data.table)

France <- read.csv("Downloads/R_YoutubeData/France.csv", header=TRUE)
class(France)

dim(France)
names(France)
str(France)
glimpse(France)
```

```
summary(France)
head(France)
tail(France)

sum(is.na(France))


UK <- read.csv("Downloads/R_YoutubeData/UK.csv", header=TRUE)
class(UK)

UK_title <- UK$title

UK_title1 <- tolower(UK_title)

UK_title2 <- removePunctuation(UK_title1)
UK_title2


stopwords("en")

stopwords("fr")

uk_new2 <- removeWords(uk_new, stopwords("en"))
uk_new2

nvec <- unlist(strsplit(uk_new2, split=" "))
stem_text <- stemDocument(nvec)
stem_text

print(stem_text)

want_to_see5 = tibble(text = stem_text) %>%
 unnest_tokens(word, text) %>%    # split words
 anti_join(stop_words) %>%    # take out "a", "an", "the", etc.
 count(word, sort = TRUE)
want_to_see5

write.table(want_to_see5 , file = "mtcars.txt", sep = "\t",
        row.names = TRUE, col.names = NA)

write.csv(want_to_see5, file = "want_to_see5.csv")
write.csv2(want_to_see5,"want_to_see5.csv")

----------

FR <- read.csv("Downloads/R_YoutubeData/France.csv", header=TRUE)
class(FR)

FR_title <- FR$title

FR_title1 <- tolower(FR_title)

FR_title2 <- removePunctuation(FR_title1)
FR_title2

FR_new <- stripWhitespace(FR_title2 )
FR_new
```

```
stopwords("fr")

FR_new2 <- removeWords(FR_new, stopwords("fr"))
FR_new2


nvec2 <- unlist(strsplit(FR_new2, split=" "))
stem_text2 <- stemDocument(nvec2)
stem_text2

print(stem_text2)

want_to_see6 = tibble(text = stem_text2) %>%
 unnest_tokens(word, text) %>%    # split words
 anti_join(stop_words) %>%    # take out "a", "an", "the", etc.
 count(word, sort = TRUE)
want_to_see6

write.table(want_to_see6 , file = "mtcars.txt", sep = "\t",
         row.names = TRUE, col.names = NA)

write.csv(want_to_see6, file = "want_to_see6.csv")
write.csv2(want_to_see6,"want_to_see6.csv")

-----------

German <- read.csv("Downloads/R_YoutubeData/German.csv", header=TRUE)
class(German)

German_title <- German$title

German_title1 <- tolower(German_title)

German_title2 <- removePunctuation(German_title1)
German_title2

German_new <- stripWhitespace(German_title2 )
German_new

stopwords("de")

German_new2 <- removeWords(German_new, stopwords("en"))
German_new2

nvec3 <- unlist(strsplit(German_new2, split=" "))
stem_text3 <- stemDocument(nvec3)
stem_text3

print(stem_text3)

want_to_see7 = tibble(text = stem_text3) %>%
 unnest_tokens(word, text) %>%    # split words
 anti_join(stop_words) %>%    # take out "a", "an", "the", etc.
 count(word, sort = TRUE)
want_to_see7
```

```
write.table(want_to_see7 , file = "mtcars.txt", sep = "\t",
          row.names = TRUE, col.names = NA)

write.csv(want_to_see7, file = "want_to_see7.csv")
write.csv2(want_to_see7,"want_to_see7.csv")


install.packages("tm")

want_to_see7_non <- tm_map("want_to_see7",removeNumbers)


---------

install.packages("tidytext")

library(tidytext)
library(dplyr)

install.packages("tm")

want_to_see6_non <- tm_map("want_to_see6",removeNumbers)


France <- read.csv("Downloads/R_YoutubeData/France.csv", header=TRUE)
class(France)

popular_tv_F <- France$category_id

install.packages("tidytext")
glimpse(popular_tv)

library(tidytext)
library(dplyr)

f <- table(popular_tv_F)
f

German <- read.csv("Downloads/R_YoutubeData/German.csv", header=TRUE)
class(German)

popular_tv_G <- German$category_id

G <- table(popular_tv_G)
G

UK <- read.csv("Downloads/R_YoutubeData/UK.csv", header=TRUE)
class(UK)

UK_title <- UK$title

UK_title1 <- tolower(UK_title)

UK_title2 <- removePunctuation(UK_title1)

popular_tv_U <- UK$category_id
```

```
U <- table(popular_tv_U)
U

France_title <- France$title

France_title1 <- tolower(France_title)

France_title2 <- removePunctuation(France_title1)

popular_tv <- France$category_id
install.packages("tidytext")

library(tidytext)
library(dplyr)

count_popular_tv = data_frame(text = popular_tv) %>%
 anti_join(stop_words) %>%    # take out "a", "an", "the", etc.
 count(word, sort = TRUE)

ggplot(iris, aes(x ="category_id", y ="likes" )) +
 geom_point()

category_id <- ggplot(France, aes(x=France$category_id))
category_id + geom_histogram()
stat_bin()
bins = 30

German <- read.csv("Downloads/R_YoutubeData/German.csv", header=TRUE)
class(German)

category_id <- ggplot(German, aes(x=German$category_id))
category_id + geom_histogram()

UK <- read.csv("Downloads/R_YoutubeData/UK.csv", header=TRUE)
class(UK)

category_id <- ggplot(UK, aes(x=UK$category_id))
category_id + geom_histogram()

category_id
glimpse(category_id)


ggplot(UK, aes(x=UK$category_id, y=UK$likes)) +
 geom_jitter(alpha=.4)

ggplot(UK, aes(x=UK$category_id, y=UK$likes, color=likes)) + geom_jitter(alpha=.4)+
 stat_smooth(method = "lm", se=FALSE, col="grey")

ggplot(German, aes(x=German$category_id, y=German$likes, color=likes)) + geom_jitter(alpha=.4)+
 stat_smooth(method = "lm", se=FALSE, col="grey")

ggplot(France, aes(x=France$category_id, y=France$likes, color=likes)) + geom_jitter(alpha=.4)+
 stat_smooth(method = "lm", se=FALSE, col="grey")

install.packages("qdapTools")
```

```
w1 <- read.csv("Downloads/F_100toptitile_trans.csv", header=TRUE)
head(w1)
w2 <- read.csv("Downloads/UK_100toptitile_trans.csv", header=TRUE)
w3 <- read.csv("Downloads/G_100toptitile_trans.csv", header=TRUE)

myfulldata = merge(w1, w2)


all_w <- read.csv("Downloads/ccccc.csv", header = TRUE)

library(tidytext)
library(dplyr)
install.packages("tibble")

all_w

text_wordcounts <- all_w %>% count(UK, sort = TRUE)
text_wordcounts

write.table(text_wordcounts, file = "mtcars.txt", sep = "\t",
            row.names = TRUE, col.names = NA)

write.csv(text_wordcounts, file = "text_wordcounts.csv")
write.csv2(text_wordcounts,"text_wordcounts.csv")


library(ggplot2)

allww <- read.csv("Downloads/text_wordcounts.csv", header = TRUE)

 # Create data
 data <- data.frame(
   name=c("episod","video","feat","ft","game","live","music","on","season","the","trailer") ,
   value=c(3,12,5,18,45)
 )

# Barplot
ggplot(data, aes(x=name, y=value)) +
 geom_bar(stat = "identity") +
 coord_flip()

# Create data
data <- data.frame(
 name=c("text_wordcounts$UK") ,
 value=c("text_wordcounts$n")
)
```

Text analysis:

```
install.packages("gutenbergr")
install.packages("tm")
install.packages("topicmodels")
install.packages("tidyverse")

library(tidyverse)
library(tm)
library(tidytext)
```

```
library(stringr)

library(topicmodels)
library(gutenbergr)

library(data.table)
library(dplyr)


UK <- read.csv("Downloads/R_YoutubeData/UK.csv", header=TRUE)
class(France)

UK_title <- UK$title

UK_title1 <- tolower(UK_title)

UK_title2 <- removePunctuation(UK_title)


stripWhitespace(UK_title2)

stopwords("en")

install.packages("SnowballC")

uk_new <- removeWords(UK_title2, stopwords("en"))
U <- table(uk_new)
U

uu <- table(U)
uu


#UK_title4 <- unlist(strsplit(UK_title3, split=" "))
#stem_text <- stemDocument(UK_title4)

UK_title2
#stem_text_cleaned <- tm_map(stem_text, removeNumbers)
#stem_text_cleaned

#unique_indexes <- unique(stem_text$i)
#stem_text <- stem_text[unique_indexes]
#stem_text

#broom::tidy
#library(broom)

#stem_text_tidy <- tidy(stem_text)
#stem_text_tidy
#help("Deprecated")

install.packages("tidytext")

library(tidytext)
library(dplyr)

want_to_see4= data_frame(text = UK_title2) %>%
```

```
 unnest_tokens(word, text) %>%
 anti_join(stop_words) %>%
 count(word, sort = TRUE)


German <- read.csv("Downloads/R_YoutubeData/German.csv", header=TRUE)
class(German)

German_title <- German$title

German_title1 <- tolower(German_title)

German_title2 <- removePunctuation(German_title1)
-----------------------------------------------------------------
install.packages("tidytext")

library(tidytext)
library(dplyr)

want_to_see3 = data_frame(text = German_title2) %>%
 unnest_tokens(word, text) %>%    # split words
 anti_join(stop_words) %>%    # take out "a", "an", "the", etc.
 count(word, sort = TRUE)
-----------------------------------------------------------------

France <- read.csv("Downloads/R_YoutubeData/France.csv", header=TRUE)
class(France)

France_title <- France$title

France_title1 <- tolower(France_title)

France_title2 <- removePunctuation(France_title1)

#stem_text_cleaned <- tm_map(stem_text, removeNumbers)
#stem_text_cleaned

install.packages("tidytext")

library(tidytext)
library(dplyr)

want_to_see2 = data_frame(text = France_title2) %>%
 unnest_tokens(word, text) %>%    # split words
 anti_join(stop_words) %>%    # take out "a", "an", "the", etc.
 count(word, sort = TRUE)


want_to_see2
```

Reference

https://www.kaggle.com/datasnaek/youtube-new