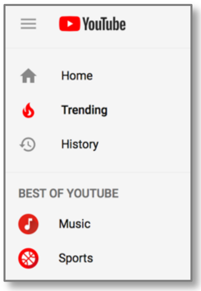
An Analysis of Trending YouTube Video Statistics

Jon Gimpel and Terukazu Nakano
March 30, 2018

Final Project for winter quarter Introduction to Data Analysis course, DBDA.X404.(2), with professor Pramod Gupta in the Database and Data Analytics department at UCSC Silicon Valley Extension.

Introduction

YouTube has become a hobby and business for individuals as well as an adjunct business or promotion venue for many corporations. By looking at the data provided by Google (YouTube) for the Trending section of its platform, this analysis intends to give insight and aid to those commencing or growing a YouTube venture. YouTube does not publish exactly how it selects videos for its Trending list, and one way to gain insight to the list is through this data.



See: www.youtube.com/feed/trending

Dataset Information

Gathered by the YouTube API, the dataset for this analysis includes just over four months of recent data on daily Trending YouTube videos. The authors obtained the data from Kaggle:

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

The specific dataset used was dated March 20, 2018, for the Trending videos between November 14, 2017 and March 20, 2018 (36.4MB).

Data for the United States, Great Britain, Germany, Canada, and France were provided in separate CSV files for the 125 days, with up to 200 listed Trending videos per day. There are 16 variables such as video title, channel title, publish date, tags, views, likes and dislikes, comment count, and description. The data also includes a category_id number that varies by region. An associated structured JSON file translates the category_id to a category name.

YouTube defines what determines if a video is ranked on Trending as follows:

> Trending aims to surface videos that:
>
> - Are appealing to a wide range of viewers
> - Are not misleading, clickbaity or sensational
> - Capture the breadth of what's happening on YouTube and in the world
> - Ideally, are surprising or novel
>
> Trending aims to balance all of these considerations. To achieve this, Trending considers many signals, including (but not limited to):
>
> - View count
> - The rate of growth in views
> - Where views are coming from (including outside of YouTube)
> - The age of the video
>
> … the video with the highest view count on a given day may not be #1 on Trending, and videos with more views may be shown below videos with fewer views… The Trending system tries to choose videos that will be most relevant to our viewers and most reflective of the broad content on the platform.
>
> YouTube does not accept payment for placement on Trending. We do not include views from YouTube ads in selecting videos for Trending. YouTube does not favor specific creators.

See: support.google.com/youtube/answer/7239739

Methods

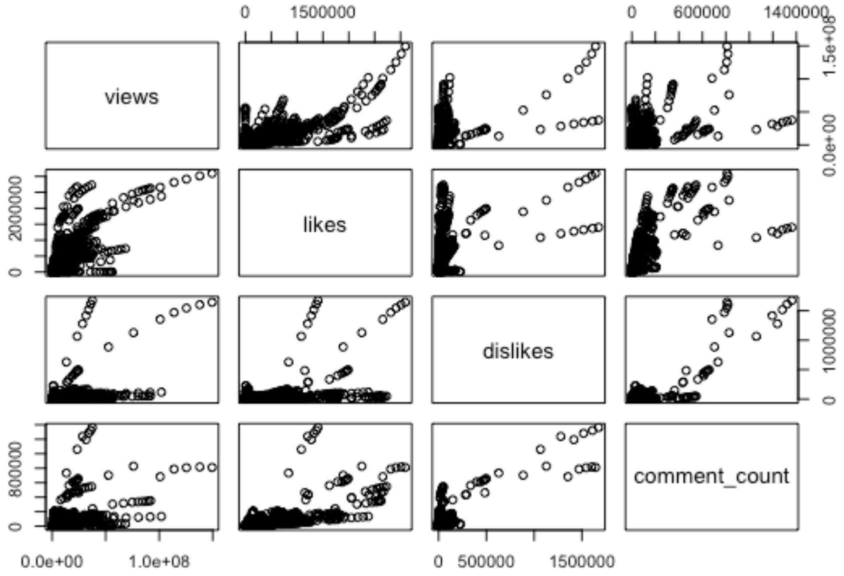
The R programming language was used for the analysis of the data. Packages included dplyr, ggplot2, wordcloud, RColorBrewer, data.table, lubridate, and corrplot.

Overview of Data and Variables

The data set is clean, with no NA fields. This is a list of the variables (US data set):

```
> str(data)
'data.frame':24951 obs. of  16 variables:
 $ video_id              : Factor w/ 4881 levels "__-22AJoFxY",
 $ trending_date         : Factor w/ 125 levels "17.01.12",
 $ title                 : Factor w/ 4946 levels "'Bachelor' Finale: Worst..",
 $ channel_title         : Factor w/ 1972 levels "12 News","1MILLION Dance..",
 $ category_id           : int  22 24 23 24 24 28...
 $ publish_time          : Factor w/ 4812 levels "2006-07-23T08:24:11.000Z",
 $ tags                  : Factor w/ 4652 levels "“Developer Update | Year..",
 $ views                 : int  748374 2418783 3191434 343168 2095731 119180..
 $ likes                 : int  57527 97185 146033 10172 132235 9763...
 $ dislikes              : int  2966 6146 5339 666 1989 511...
 $ comment_count         : int  15954 12703 8181 2146 17518 1434...
 $ thumbnail_link        : Factor w/ 4881 levels "https://i.ytimg.com/vi..",
 $ comments_disabled     : Factor w/ 2 levels "False","True": 1 1 1 1 1 1...
 $ ratings_disabled      : Factor w/ 2 levels "False","True": 1 1 1 1 1 1...
 $ video_error_or_removed: Factor w/ 2 levels "False","True": 1 1 1 1 1 1...
 $ description           : Factor w/ 5099 levels ""," ","- Charities -\\..",
```

For the US, videos were listed for an average of 24,951/4881 = 5.1 days. The least number of views when listed was 459 and the largest was 149,376,127. The median was 380,393 views. Only 433 (1.7%) had comment disabled and 152 (0.6%) had ratings disabled.



The repetition of particular videos in the Trending list is evident in these correlation plots.

To see the average of the difference between trending date and publishing date, we need to create a unique video list and then convert both date forms to the same type:

```
> library(dplyr)
> data.distinct <- data %>% distinct(video_id,.keep_all=TRUE)
> data.distinct$trending_date <- ydm(data.distinct$trending_date)
> data.distinct$publish_time <- ymd(substr(data.distinct$publish_time,start =
1,stop = 10))
> data.distinct$dif_days <- data.distinct$trending_date-
data.distinct$publish_time
> mean(data.distinct$dif_days)
[1] 28.6007
```

So, in general, it takes four weeks for YouTube videos to become Trending.

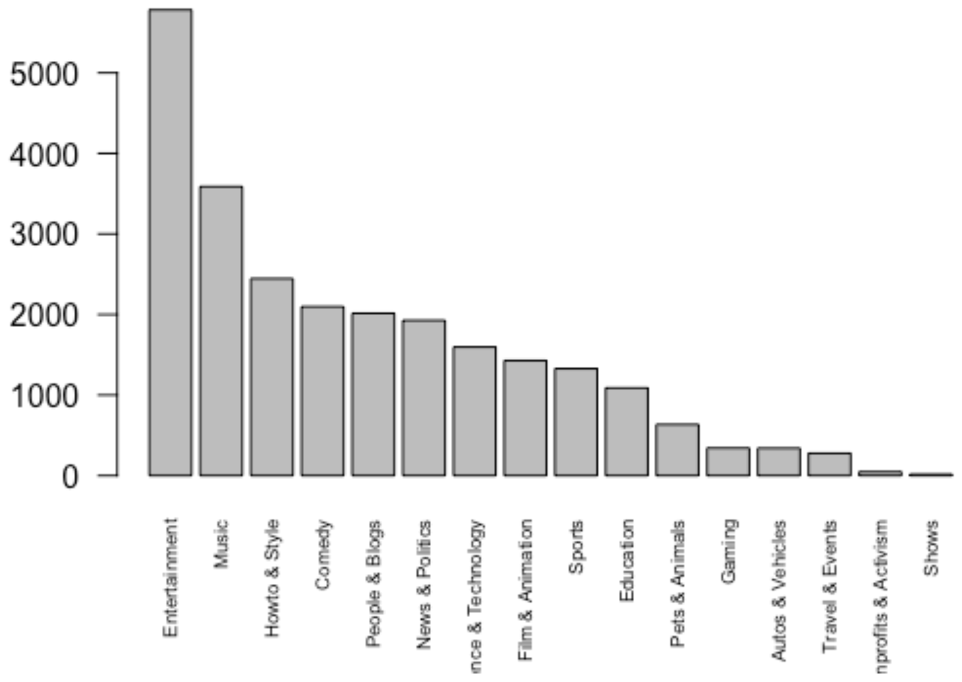
Categories Most Frequently in Trending List, US

To determine the YouTube video categories that appear most frequently in the US Trending list, we'll use the table function.

```
> table(data$category_id)
   1    2   10   15   17   19   20   22   23   24   25   26   27   28   29   43 
1428  340 3588  634 1328  277  342 2015 2097 5782 1927 2445 1088 1596   48   16 
```

Adding the category names from the US JSON file, sorting by decreasing, and then plotting:

```
data$category_name <- ifelse(data$category_id==1,"Film & Animation",
    ifelse(dataCategoryFrequencySorted$category_id==2,"Autos & Vehicles",
    ifelse(dataCategoryFrequencySorted$category_id==10,"Music",
    ifelse(dataCategoryFrequencySorted$category_id==15,"Pets & Animals",
    ifelse(dataCategoryFrequencySorted$category_id==17,"Sports",
    ifelse(dataCategoryFrequencySorted$category_id==18,"Short Movies",
    ifelse(dataCategoryFrequencySorted$category_id==19,"Travel & Events",
    ifelse(dataCategoryFrequencySorted$category_id==20,"Gaming",
    ifelse(dataCategoryFrequencySorted$category_id==21,"Videoblogging",
    ifelse(dataCategoryFrequencySorted$category_id==22,"People & Blogs",
    ifelse(dataCategoryFrequencySorted$category_id==23,"Comedy",
    ifelse(dataCategoryFrequencySorted$category_id==24,"Entertainment",
    ifelse(dataCategoryFrequencySorted$category_id==25,"News & Politics",
    ifelse(dataCategoryFrequencySorted$category_id==26,"Howto & Style",
    ifelse(dataCategoryFrequencySorted$category_id==27,"Education",
    ifelse(dataCategoryFrequencySorted$category_id==28,"Science & Technology",
    ifelse(dataCategoryFrequencySorted$category_id==29,"Nonprofits & Activism",
    ifelse(dataCategoryFrequencySorted$category_id==43,"Shows", NA
        ))))))))))))))))))
> barplot(sort(table(data$category_name),decreasing = T),las=2,cex.names = .6)
```



In the US, *Entertainment* videos are the most comonly trending, followed by *Music*.

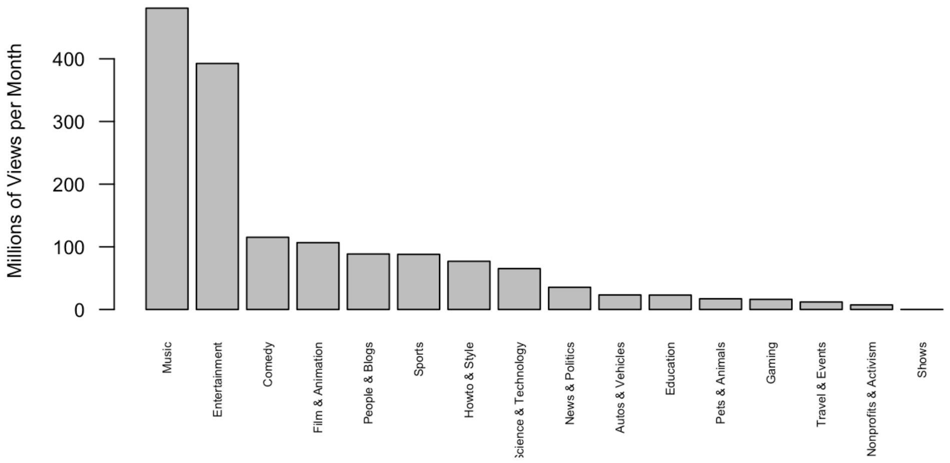
Views by Category, US

To estimate total views by category, one must be careful not to recount the videos that are listed over multiple days. In fact, most videos are listed over multiple days: using the str() function, we see that there are only 4881 unique videos in the 24,951 observations (US). First we'll order data so that each video listed over multiple days is sorted by descended views (reverse chronological order). Then we'll remove the repeated appearances of the video except for the first, most-viewed listing, and then aggregate and sort:

```
> dataSortDupes <- data[order(data$video_id, -data$views ), ]
> dataSortDupesRemove <- dataSortDupes[ !duplicated(dataSortDupes$video_id), ]
> dataViewsByCat <- aggregate(views ~ category_id, dataSortDupesRemove, sum)
> dataViewsByCatSorted <- dataViewsByCat[order(-dataViewsByCat$views),]
```

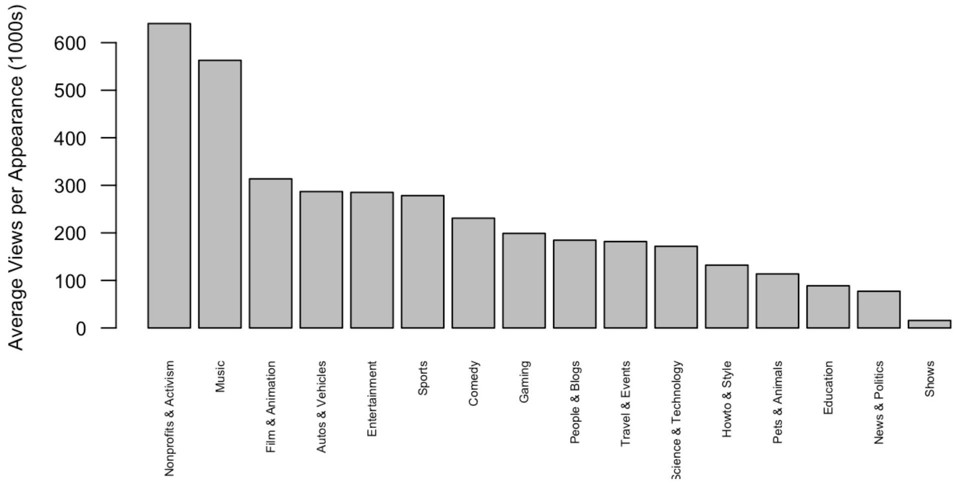
Adding back the category names and then plotting:

```
> barplot(dataViewsByCatSorted$views/1000000/4.2, names.arg
=dataViewsByCatSorted$category_name, ylab="Millions of Views per Month", las=2,
cex.names=.5, cex.axis=.8, cex.lab=.8)
```



Comparing the *frequency* and *views* plots, we see differences. *Howto & Style*, for example, appears third in the frequency of appearance, but seventh in the total views. It appears to be a category that YouTube promotes beyond the merits of views alone. Let's look at the ratio of views to frequency of appearance for all the categories. After merging and sorting:

```
> barplot(dataCategoryMerge$views/dataCategoryMerge$freq/1000, names.arg
=dataCategoryMerge$category_name, ylab="Average Views per Appearance (1000s)",
las=2, cex.names=.5, cex.axis=.8, cex.lab=.8)
```



The categories with lower values in this graph are categories where it appears that YouTube is promoting videos with fewer views in general in order to satisfy its stated desires to appeal to a wide range of viewers, capture the breadth of what's happening on YouTube, etc.

And vise versa, the larger values show categories where YouTube is limiting the number of appearances. *Music* is the second most frequently listed category and it gets the most views as well. *Nonprofits & Activism* is unexpected. YouTube very infrequently puts videos in this category in its Trending list, but when it does, it is for videos that are very popular.

Note: The *Shows* category is an outlier. The category is poorly defined in Google documentation. It includes just over a dozen shows for only one channel, CNET.

Most Popular US Channels

To determine the most popular channels in the Trending list, we'll look at how often a channel appears in the Trending list.

There are 1972 channels represented in the list for the US during this time period. The most popular channel appeared 122 of the 125 days. The median number of days a channel appears is 7. These are the 20 most frequently Trending channels:

```
> dataChannelFrequencySorted <- dataChannelFrequency[order(-
dataChannelFrequency$freq),]
> head(dataChannelFrequencySorted,20)
555                                    ESPN  122
1892                                    Vox  121
1217                                    NBA  120
1231                                Netflix  120
591                          First We Feast  119
1729 The Tonight Show Starring Jimmy Fallon  117
1242                                    NFL  115
1795                              Tom Scott  114
1745                           TheEllenShow  112
1699     The Late Show with Stephen Colbert  111
1941                                    WWE  109
996                             Life Noggin  107
784                                 INSIDER  106
854                       Jimmy Kimmel Live  105
969             Late Night with Seth Meyers  105
369                                     CNN  104
1504                         Screen Junkies  104
1492                    Saturday Night Live   96
676                         Great Big Story   94
1075                       Marques Brownlee   92
```

So while a few big names had a video listed almost every day, no channels were listed more than the number of days for the data set. Thus, each channel is probably limited to one Trending video per day. It is also interesting to see that of the top 20 channels, most are from major media corporations for whom YouTube is a secondary business opportunity. There are five or six channels that seem to have been born on YouTube (see highlights). Of these, most are in the *Science & Technology* category.

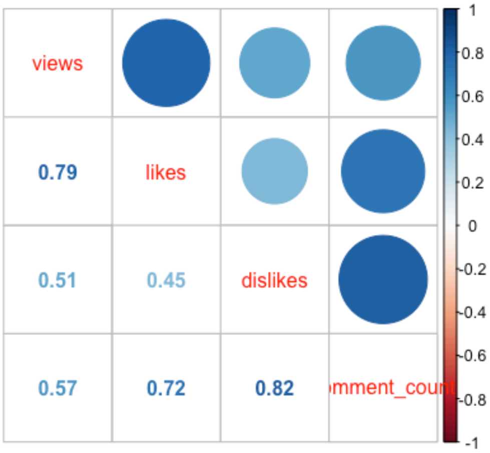
Correlation between Views, Comments, Likes, and Dislikes, US

To determine correlation between views, comments, likes, and dislikes in the list of the US, the cor() function is the best way.

```
> cor(data[,c("views","likes","dislikes","comment_count")])
                  views     likes  dislikes comment_count
views         1.0000000 0.7947606 0.5120405     0.5746443
likes         0.7947606 1.0000000 0.4461225     0.7231761
dislikes      0.5120405 0.4461225 1.0000000     0.8163151
comment_count 0.5746443 0.7231761 0.8163151     1.0000000
```

And visualizing this correlation by corrplot.mixed():

```
> corrplot.mixed(corr=cor(data[,c("views","likes","dislikes","comment_count")]))
```



The strongest correlation is between comments and dislikes. Views and likes are also highly correlated.

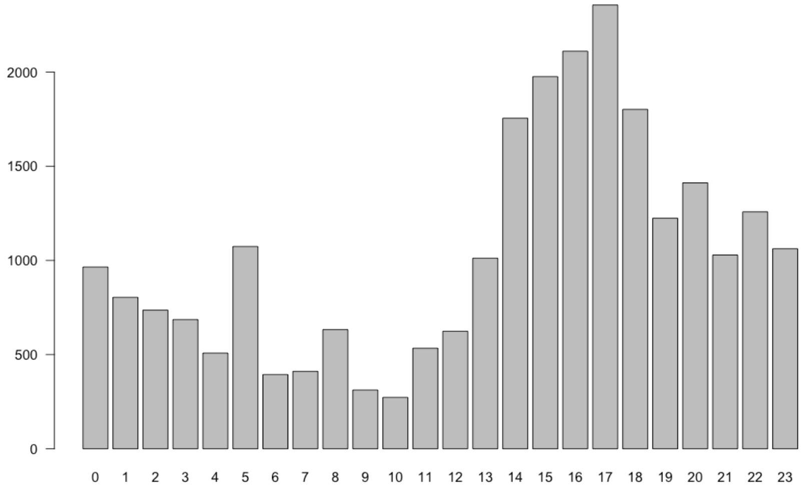
Video Publishing Time and Day of Week, US

To determine the most common publishing time in the Trending list, we first need to convert the values of the column publish_time:

```
> data.publishtime <- as.POSIXlt(data$publish_time, format = "%Y-%m-
%dT%H:%M:%S.000Z")
```

Then plotting the frequency of publishing hour:

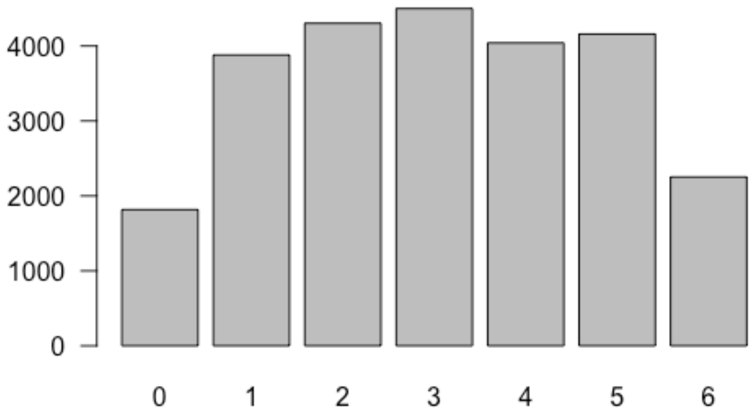
```
> barplot(table(data.publishtime$hour), las = 1)
```



The vertical axis shows the frequency and horizontal one shows the time of publishing with UTC time zone. In this chart, 5:00pm UTC is the most frequent time and 10:00am UTC is the least frequent time. This chart will shift depending on your time zones. For PST, this chart needs to shift to the left by 8 hours. People publish most frequently at 9:00am PST.

Plotting by day of week:

```
> barplot(table(data.publishtime$wday), las = 1)
```

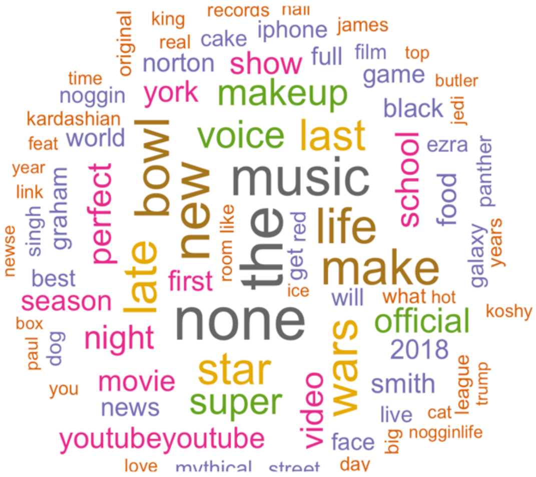


The values of the horizontal axis represent days of week, with 0 to 6 meaning Sunday to Saturday. We found that weekdays are more common than weekends in publishing time on the Trending list.

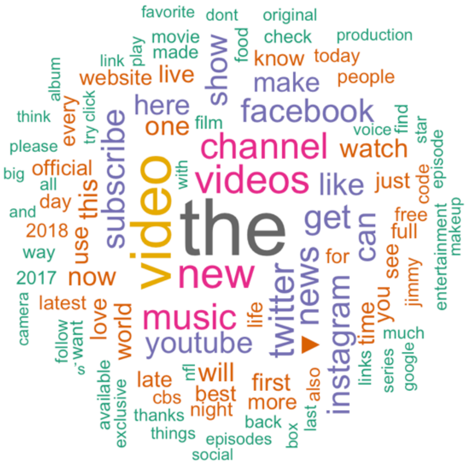
Common Keywords in Tags, Description, and Title, US

A good way to present keywords to a creative audience, the individuals that would be writing their own text for descriptions is visually with a "word cloud". In R, this is facilitated by a package called wordcloud. Colors can be added with the RColorBrewer package.

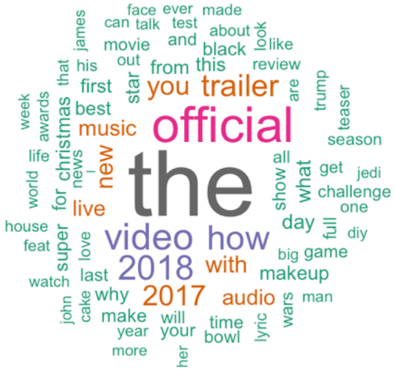
```
> install.packages("wordcloud"); install.packages("RColorBrewer")
> library("wordcloud"); library("RColorBrewer")
> wordcloud(words=data$tags, max.words=80, random.order=FALSE, rot.per=0.35,
use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
> wordcloud(words=data$description, max.words=100, random.order=FALSE,
rot.per=0.40, use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
> wordcloud(words=data$title, max.words=80, random.order=FALSE, rot.per=0.35,
use.r.layout=FALSE, colors=brewer.pal(8, "Dark2"))
```



Tags



Video Description



Video Title

It is interesting to see how many keywords in the video descriptions are used in promoting other social media presences for the channels (twitter, facebook, instagram). YouTube does not seem to discourage this, and it may even prefer to list videos that attract users from other places on the internet.

Differences between Countries

To streamline the process, for this comparison we'll use the complete data set. That is, we will not subtract duplicate listings of the same video over multiple days as we did in the *Views by Category, US* section. First, add a location column in data sets for each country.

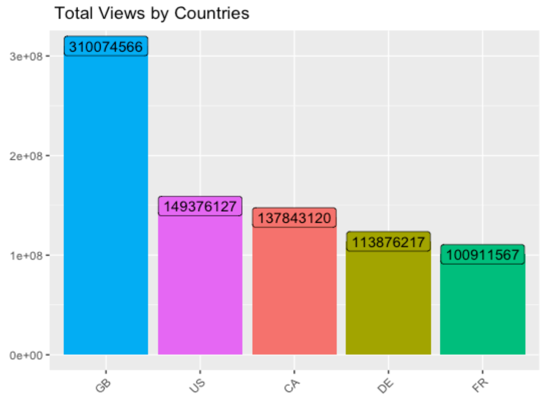
```
> data$location <- c(rep("US",24951))
> dataGB$location <- c(rep("GB",24948))
> dataDE$location <- c(rep("DE",24946))
> dataCA$location <- c(rep("CA",24922))
> dataFR$location <- c(rep("FR",24903))
```

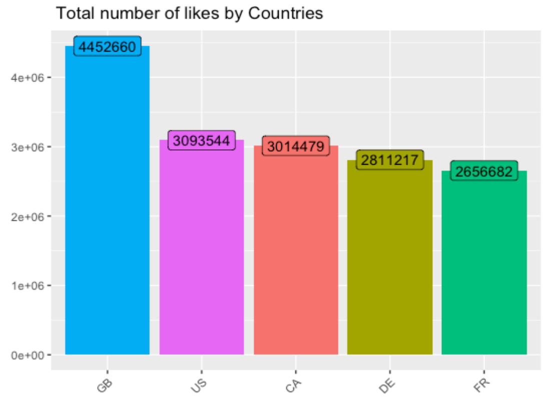
Note that each country has approximately equal number of observations, meaning that YouTube promotes about the same number of videos each day in its Trending list. Now, combining data sets:

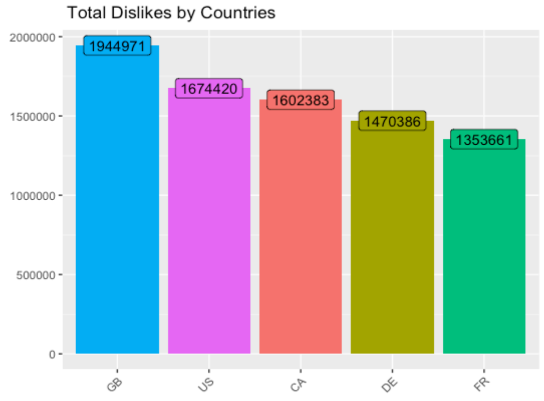
```
> videos <- as.data.table(rbind(data,dataGB,dataDE,dataCA,dataFR))
> nrow(videos)
[1] 124670
```

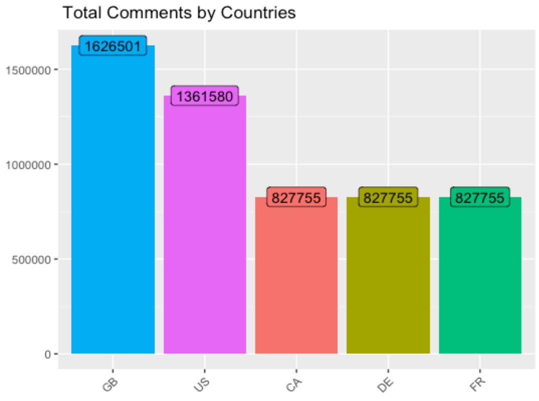
Then plotting views, likes, dislikes, and comments respectively:

```
> 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)
> 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)
> ggplot(videos[,.("Total_Dislikes"=max(dislikes)),by=location],
aes(reorder(location,-Total_Dislikes),Total_Dislikes,fill=location))+
geom_bar(stat="identity")+ geom_label(aes(label=Total_Dislikes))+
guides(fill="none")+ theme(axis.text.x = element_text(angle = 45,hjust = 1))+
labs(title=" Total Dislikes by Countries")+ xlab(NULL)+ ylab(NULL)
> ggplot(videos[,.("Total_Comments"=max(comment_count)),by=location],
aes(reorder(location,-Total_Comments),Total_Comments,fill=location))+
geom_bar(stat="identity")+ geom_label(aes(label=Total_Comments))+
guides(fill="none")+ theme(axis.text.x = element_text(angle = 45,hjust = 1))+
labs(title=" Total Comments by Countries")+ xlab(NULL)+ ylab(NULL)
```









Note that YouTube is not providing data based on views only from that country. It is providing the total views, likes, dislikes, etc, for the video from all regions. This is evident in the fact that Great Britain shows higher “views” than the US. Let’s examine this more by comparing the frequency of Trending appearances by category for the US and Great Britain:

First we’ll make the US data the same size as GB and combine the US/GB category names.

```
> data.us <- data[1:24948,]
> data.compare.usgb <- cbind(data.us$category_name,data.gb$category_name)
```

Convert the table data into matrix data.

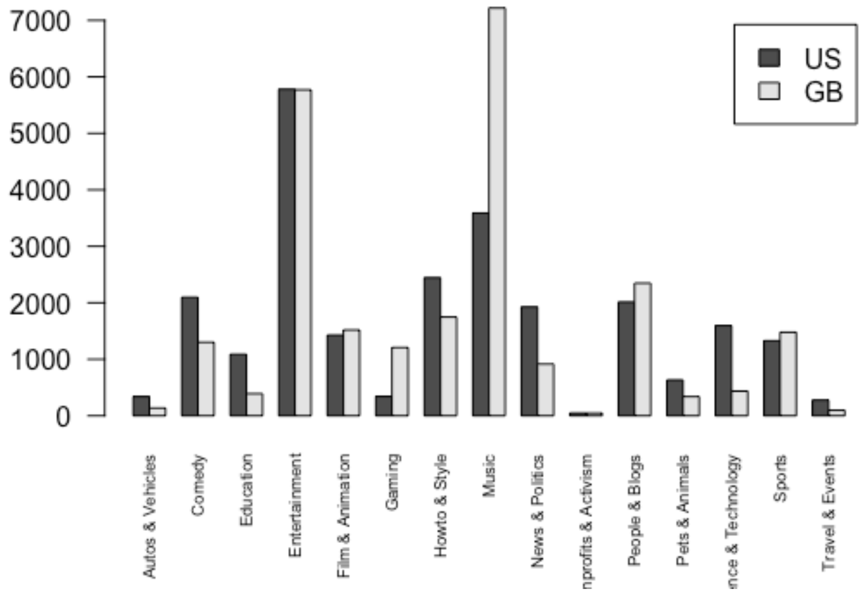
```
> table.us <- as.matrix(table(data.compare.usgb[,1]))
> table.gb <- as.matrix(table(data.compare.usgb[,2]))
```

Then, remove the outliner category called *Shows* because GB does not have the category.

```
> table.us <- as.matrix(table.us[c(-14),])
```

Transpose these matrix and plot.

```
> table.us <- t(table.us)
> table.gb <- t(table.gb)
> rownames(table.compare.usgb)<-c("US","GB")
> barplot(table.compare.usgb,beside = T,las=2, legend=T, cex.names = .6)
```



The larger number of view, likes, etc. for Great Britain is due to it have a higher number of *Music* videos in its Trending list. As we saw in the *Average Views per Appearance* analysis for the US, *Music* category videos have a lot more views/listing than other categories.

Conclusion

We had limited information about the Trending list in YouTube, so we could not do some analyses such as compare data before and after a video was listed as Trending. The data is the same as what's shown on the YouTube video pages, but in a clean tabular format. That said, our analysis led us to a few general conclusions:

- There are about 200 Trending videos, and they trend for an average of 5.1 days.
- *Music* and *Entertainment* are the most popular Trending categories, and often large corporations host those types of channels.
- YouTube promotes certain videos and categories of videos with fewer views in order to satisfy its stated desires to appeal to a wide range of viewers, etc.
- Publishing is most frequent during weekdays, with 9:00am PST being the peak time.
- The strongest correlation is between comments and dislikes, but views and likes are also highly correlated.
- YouTube seems OK with links out to other sites.
- Music is especially popular on YouTube in Great Britain.

In addition to the references provided already in this document, the authors would like to thank the people that posted analytic ideas on websites such as kaggle.com and stackoverflow.com, as well as the creators of the R packages we used.