Regional Habits of Trending YouTube Videos

Cesar F. Pardede

14 DEC 2020

Abstract

With over 2 billion monthly users and 500 hours of content uploaded every minute, YouTube undoubtedly falls into the regime of big data [6]. In this paper, we examine a particular subset of videos on YouTube - the "trending" category. Trending videos are videos which are considered by YouTube to have the *potential* to be massively popular and amass a large number of views quickly [4]. These videos are then promoted by YouTube itself on a trending page. Which videos are considered trending is constantly changing as videos are uploaded, view counts are updated, and potentially popular videos are identified. Although trending videos represent a very small sample among the massive amount of content being uploaded or already on YouTube, the trending videos selected by YouTube must reveal something about the viewers of a given region. We want to find out whether there are any discernible differences in viewing habits or online behavior of viewers around the globe based on the trending videos worldwide and within any region.

Introduction

Since the number of trending and non-trending videos grows everyday, there is virtually no limit to the amount of data that can be analyzed. Similarly, with such rich data sets and recent developments in data science, there are an unlimited number of methods and techniques we could try to apply to extract some information. We narrow the scope of this paper to trying to determine the different viewing habits of consumers between regions.

Our data set of interest covers the top 200 trending videos on YouTube from 14 November 2017 to 14 June 2018 for 10 different regions: Canada (CA), Germany (DE), Great Britain (GB), India (IN), South Korea (KR), Mexico (MX), Russia (RU), and the United States (US); and can be obtained at https://www.kaggle.com/datasnaek. Each region has their own data file, and we combine them together to create one global data set. There are

approximately 380,000 total records, with approximately 40,000 observations per region (JP only has around 20,000 for some reason). There are 16 variables including: trending date, video ID, title, channel, category ID, publishing time and date, tags, views, likes, dislikes, comment count, thumbnail link, comments disabled, ratings disabled, video error or removed, and description.

We explain the variables briefly. Most of the variable types are clear, but we note that the variables comments disabled, ratings disabled, video error or removed are boolean, video ID is a randomly generated alphanumeric string, category ID is an integer encoding different categories of videos (Film & Animation, Autos & Vehicles, Music, etc.), tags are one long continuous string, and thumbnail link is a URL to a thumbnail photo. We include the category ID key in Appendix A. In addition to these variables, we separate publishing time and date into two separate variables, and define a region variable indicating which file the record was read from.

Before we begin exploring our data, we explore previous efforts to perform statistical analysis on a set of trending YouTube videos. In all but one of our reviews, the data our predecessors analyze is not identical to our own, though the variables remain. There are many ways to obtain YouTube trending data, including navigating to the Kaggle repository [5], or interfacing directly with the YouTube API. We chose to take our data from the Kaggle repository for convenience in a class project, but our predecessors may have chosen to draw directly from the YouTube API to have the most competitive or current data available at the time. Regardless, we feel their analysis is valuable and worth review. Their results, discussions, and conclusions will inform our own decisions and interpretations of our data and results.

Related Work

I. Barjasteh, Y. Liu, H. Radha

The earliest work we could find related to trending YouTube videos was conducted by Barjesteh et al. in 2014 [3]. In their analysis, they treat trending videos as a time series, and track the status of 8,000 trending and non-trending videos over a period of nine months. Their analysis was concerned with learning more about the four following topics:

1. the lifecycle of a trending video,

2. comparison between trending and non-trending videos,

3. the profiles of channels with trending videos, and

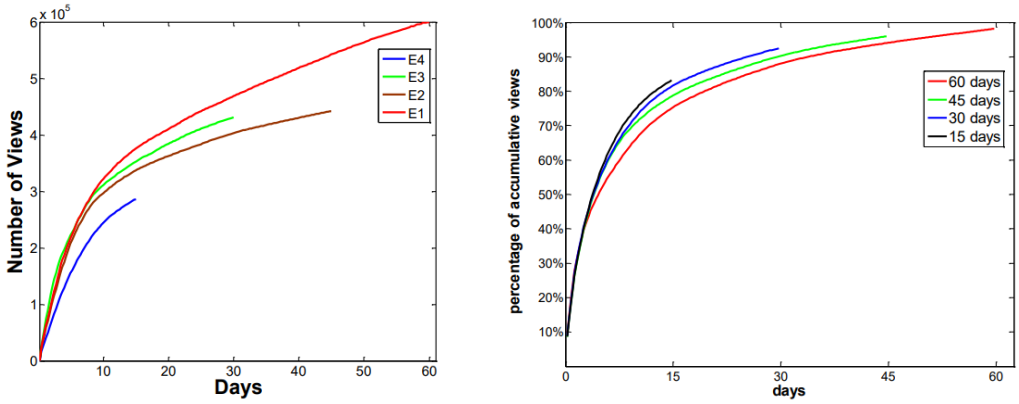

Figure 1: Figures from [3] show the rate of accumulation of views for four different sets of trending videos.

4. the directional relationship between categories of trending videos (i.e. does viewing a video from one category *cause* one to view videos in another category?)

In their approach, Barjasteh et al. obtained an initial sample of 4,000 trending videos between May 2012 and July 2012. Then, over a course of four weeks, they obtained another sample of 2,000 trending and 2,000 non-trending videos for a total of 8,000 data points (6,000 trending, 2,000 non-trending) over a course of four months. By analyzing the rate of accumulated views for sets of trending videos added to the data set at different points in time found a common pattern among all trending videos. Trending videos, as expected, gain a lot of views very quickly, but the speed at which they gain views starts to slow down after about 10 days. Actually, around 75% of the views obtained in the first 60 days are gained in the first 15 days. Beyond the 10-15 day mark, videos will accumulate views at a much slower rate, possibly due to approaching saturation of the target audience. Figure 1 (taken from [3]) shows number of views over time for (left), and the percentage of views within 60 days for trending videos collected at four different times (right).

In comparing differences between trending and non-trending videos, Barjasteh et al. do not uncover much that we don't already know such as the fact that trending videos get more views, get more comments, and get more views and comments over a longer period of time. However, they also showed that trending and non-trending videos share a similar distribution for duration, which implies the length of a video is not a significant factor in whether a video will be trending or not. When comparing the differences between channels of trending videos, and channels hosting non-trending videos, Barjasteh et al. codify what we already suspected: channels with more subscribers and more total views are more likely to be trending, which makes sense as the number of subscribers and views indicate a channel's popularity and audience. Although there was a difference between the distribution of genders for trending and non-trending videos, it was not indicated as a significant difference.

One of their most interesting results is discovering the tendency of viewers to watch a video from another category given the category of the video they are currently watching, this

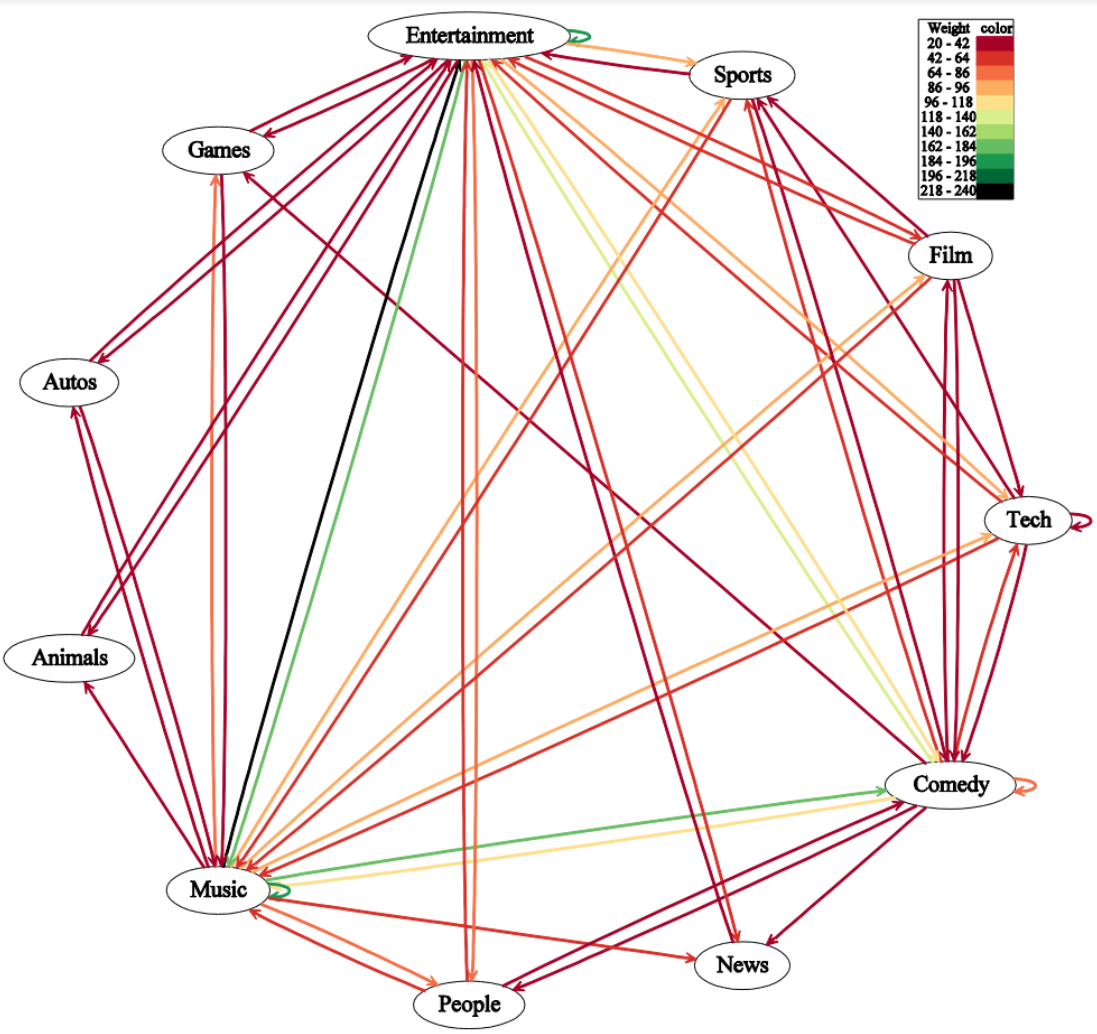

Figure 2: Directional relationships of trending videos and their categories.

is called a directional relationship. A directional relationship can be thought of as a one-way correlation in that if some stochastic process X causes another Y, it is not necessarily true that Y also causes X. In the way it is described in [3], if taking into account past samples of X improves the predictability of Y, then X is considered to have caused Y. Strong directional relationships indicate a strong tendency to move from category X to category Y after viewing a trending video in category X. Using the Granger Causality metric, Barjasteh et al. reveal directional relationships between all pairs of categories, the strongest directional relationships are illustrated in Figure 2, which is taken from [3]. The many relationships to the music, entertainment, and comedy categories (including reflexive relationships) indicate that regardless of which category a viewer is currently watching, there is a high tendency to watch a video from one of those categories next.

Barjasteh et al. provide a unique analysis that I could not find in subsequent efforts by incorporating non-trending videos, channel data, and directional relationships. The addition of

these supplementary data really bolstered their analysis and provided a richer understanding of viewing habits.

A. Alyousfi and K. Maclever

Subsequent efforts by A. Alyousfi and K. Maclever stick strictly to the same variables we have available, but is comprised of data from different time frames, or from a single region rather than the entire global set. In their analysis, they provide some statistics specific to their particular data set. A. Alyousfi published analyses of trending YouTube videos twice - once in October 2018 and again in July 2020. Their analysis in October 2018 detailed trending videos between 14 November 2017 to 14 June 2018 in the US; their analysis in July 2020 covers all trending videos for the year 2019. In both analyses, the entertainment category had the most trending videos, and 'official' was one of the most popular words in the title of trending videos [1, 2]. Interestingly, the most popular day of the week to upload trending videos changed from Fridays in October 2018 to Tuesdays in July 2020. K. Maclver performs a similar analysis for Canadian trending videos from November 2017 to 14 June 2018. He also found the entertainment category had the most trending videos, but added on that videos in the music category have the highest monthly average number of views [4].

Overall, while interesting, A. Alyousfi and K. Maclever analyses don't dig in to the underlying structure of the data like Barjasteh et al. They reveal some of the surface level statistics of trending YouTube videos, but I don't feel they contribute to a deeper understanding of the viewing habits or behaviors of consumers or society as a whole.

Exploratory Data Analysis

Recall our chosen data set and the scope of our endeavor, mentioned in the introduction. Our goal is to perform some analysis somewhere in scale between Barjasteh et al. analysis and A. Alyousfi and K. Maclever analyses. Our chosen data set is a very rich data set, and has a lot of potential for analysis. No doubt there are a lot of interesting statistics we could pull out, but we want to keep the focus on discovering deeper relationships than the surface level analyses provided by A. Alyousfi and K. Maclever. At the same time, the data we have is already cumbersome to work with on our personal computer, and we have a limited time on this project so it will not be as thorough or exploratory as Barjasteh et al. analysis.

When I initially chose this data set, I considered examining it to try to discover the common properties among trending videos to see how one could engineer a trending video and monetize their success. The issue with this approach is something I learned about called "survivorship bias". According to Wikipedia, "Survivorship bias or survival bias is the logical error of concentrating on the people or things that made it past some selection process and overlooking those that did not, typically because of their lack of visibility."

Without examining non-trending videos, it would be difficult to determine the differences between trending and non-trending videos. While Barjasteh et al. did include non-trending videos in their analysis, I feel the only notable results was that distribution of video length between trending and non-trending videos was similar. Their analysis showed trending videos get more views and comments than non-trending videos, but they don't show what makes a video trending rather than non-trending. Thus we began our exploratory data analysis looking for the differences in trends and viewing habits between different regions instead.

In exploring our data, we discover there exist videos with as few as 117 views, 0 likes, 0 dislikes, and 0 comments, which indicates to us that trending videos are selected by something other than just these variables. Potentially, other factors not included in this data set might determine which videos are trending, including the properties of the channel uploading the video, the popular "hashtags" at the time, the number of shares a video has, or the potential reach of a video through the YouTube recommendation system. All these things are external to our data set, so it's not something we could test right now, but it's interesting to think about and a potential project to tackle in the future.

In agreement with A. Alyousfi and K. Maclever analyses, the entertainment category has the most trending videos globally, with twice as many trending videos as the next most trending category (people & blogs).

Our exploratory analysis revealed 95% of videos go trending within 14 days of being published, with 50% of videos starting to trend within 1 day of being published. There was one incredible outlier that starting trending on 5 Feb. 2018, 4,215 days after being published. That video was the popular "Budweiser - Original Whazzup?" commercial for Budweiser beer originally airing from 1999 to 2002. The longest trending video by title was Mission: Impossible - Fallout (2018) - Official Trailer which trended globally for 100 days. By video ID, the longest trending video was Childish Gambino - This Is America (Official Video) for 92 days. We make the distinction between videos trending by title and ID, because videos can have multiple ID when they are removed and re-uploaded under the same name.

We show the most notable videos (those with the most views, likes, dislikes, and comments) and the channel that uploaded them for each region and globally in Figure 3.

Notice that Canada, Germany, France, South Korea, and Mexico have the same preferences for most viewed, most liked, most disliked, and most commented video. Interestingly, "YouTube Rewind: The Shape of 2017" was the most viewed, liked, disliked, and commented video in India, while it was the most disliked video by almost every other region except for Great Britain, Japan, and the US. Similarly, "'FAKE LOVE' Official MV by BTS" was the most viewed, liked, disliked, and commented video in Japan, along with being the most liked video in every other region except India. The most watched video globally is "Nicky Jam x J. Balvin - X (EQUIS)", although it is the most viewed video for only GB, it had over 420 million views at the time of this data collection; almost double the second most viewed video globally, "Childish Gambino - This Is America", with 225 million views. Similarly, "So Sorry." only appears as the most disliked and commented on video for GB and the US, but

	Views	Likes	Dislikes	Comments
CA	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)
DE	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)
FR	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)
GB	Nicky Jam x J. Balvin - X (EQUIS) (NickyJamTV)	'FAKE LOVE' Official MV by BTS (ibighit)	So Sorry. (Logan Paul Vlogs)	So Sorry. (Logan Paul Vlogs)
IN	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)
JP	'FAKE LOVE' Official MV by BTS (ibighit)	'FAKE LOVE' Official MV by BTS (ibighit)	'FAKE LOVE' Official MV by BTS (ibighit)	'FAKE LOVE' Official MV by BTS (ibighit)
KR	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)
MX	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)
RU	'FAKE LOVE' Official MV by BTS (ibighit)	'FAKE LOVE' Official MV by BTS (ibighit)	YouTube Rewind: The Shape of 2017 (Youtube Spotlight)	'FAKE LOVE' Official MV by BTS (ibighit)
US	Childish Gambino - This Is America	'FAKE LOVE' Official MV by BTS (ibighit)	So Sorry. (Logan Paul Vlogs)	So Sorry. (Logan Paul Vlogs)

	Views	Likes	Dislikes	Comments
Global	Nicky Jam x J. Balvin - X (EQUIS) (NickyJamTV)	'FAKE LOVE' Official MV by BTS (ibighit)	So Sorry. (Logan Paul Vlogs)	So Sorry. (Logan Paul Vlogs)
	424,538,912	5,613,827	1,944,971	1,626,501

Figure 3: Videos with the most views, likes, dislikes, and comments for each region.

	Views	Likes	Dislikes	Comments
Minimum	117	0	0	0
Median	177,370	3,446	179	511
Mean	1,326,568	37,884	2,126	4,254
95%	4,479,059	162,042	6,561	16,263

Figure 4: Quantiles of views, likes, dislikes, and comment count.

is the most disliked and commented video globally, because of the larger audiences in those regions.

To get an idea of the biases and variances of views, likes, dislikes, and comment count, we create a table of quantiles in Figure 4. We notice that there are large differences between the medians and means of each variable with the means being much larger than the medians. This indicates the presence of outliers skewing the mean views, likes, dislikes, and comment count higher. Consider the most viewed, most liked, most disliked, and most commented videos in Figure 3 and compare them to the 95% of views, likes, dislikes, and comments. Again, we see a huge difference between the 95 percentile and the maximums, confirming the presence of outliers.

My intuition tells me the median number of views, likes, dislikes, and comments seem small for trending videos. Most videos I've encountered on the trending page already have millions of views, or at least several hundred thousand. It's hard for me personally to believe that half of all trending videos have less than 177,000 views. If we know that the data covers the top 200 trending videos each day, then that means the trending video with the median number of views would be the 100th trending video each day. That's a long way to scroll, which I doubt most users actually get to. However, it also helps to explain the massive leverage outliers have. A popular video landing on the first page of trending videos gets a good amount of exposure, leading to more views and keeping it near the top of the list for trending videos, leading to even more exposure until its target audience has been saturated and it stops accumulating views.

We plot the data in Figure 5 to see if we can easily glean information through visual inspection. Unfortunately, our plots don't reveal anything; the outliers are so extreme, being orders of magnitude greater than the rest of the data, that they can hardly be visualized in the histograms squishing the rest of the data points into one bar near 0.

Because of the magnitude of the outliers, we do a log transformation on the number of views, likes, dislikes, and comment counts and plot the results in Figure 6. Interestingly, the distribution of the log-transformation of variables all resemble normal distributions. We are not sure how to interpret this, what does it mean for a log-transformation to have a normal distribution? A regular distribution implies data is generated around some mean parameter with dispersion determined by another parameter. If a log-transformation of data has a normal distribution, does that imply the data is generated by some underlying

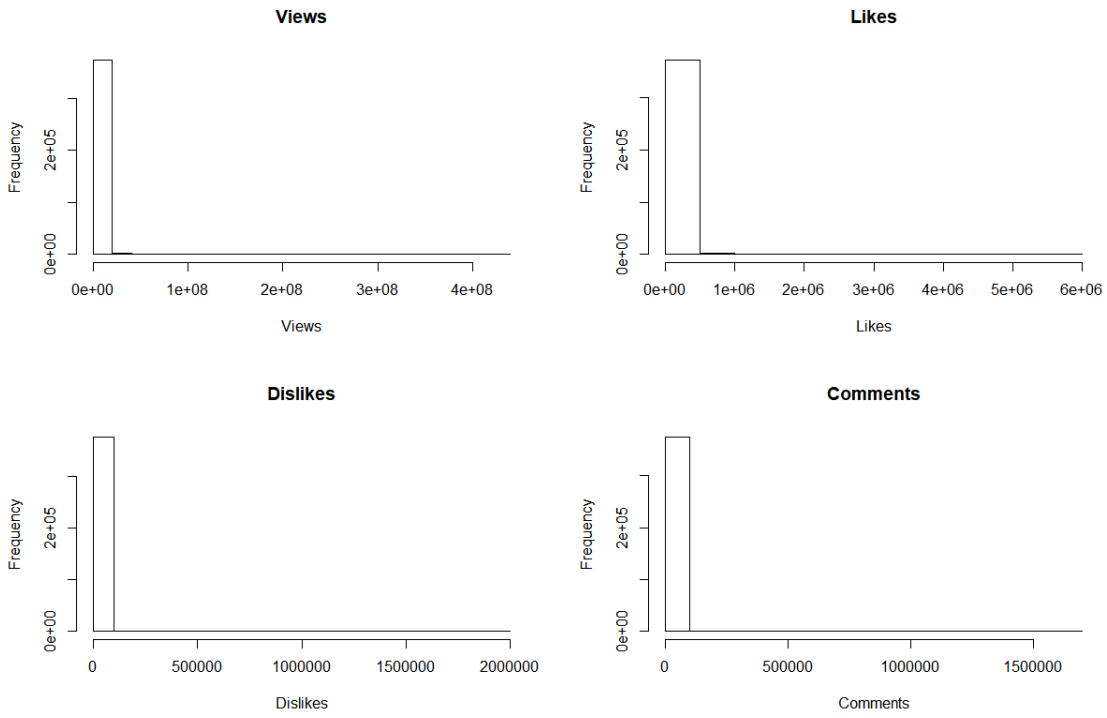

Figure 5: Histogram of variables.

mean magnitude and dispersion. That is, if data generated from a Gaussian distribution has equation $y = \mu + \epsilon$ where $\epsilon \sim N(0, 1)$, then does data from a log-transformation have equation $y = e^{\mu+\epsilon}$? This would explain how the data is normally distributed after taking the log of values.

Continuing on, we plot the data together; first as their raw values, then as their log values. In Figure 7, we show one example of a pair of variables plotted together, plotting any of the other pairs of variables together will produce similar plots because they were all similarly distributed after log transformations. Since the log-transformed values each had a univariate normal distribution, then plotting the log-transformed values together will generate something that looks like a bivariate normal distribution which we see on the right side of Figure 7. Thus, plotting more than two log-transformed variables together will result in a multivariate normal distribution of observations.

This leads us to think that clustering may not be an appropriate method to analyze the data using views, likes, dislikes, and comments variables. I mean, sure, you could perform clustering on the data, but from our exploration we see there really is only one cluster of data in the form of a multivariate normal distribution. Clustering would indiscriminately and artificially partition that multivariate normal distribution around whichever centroids happen to be defined, and would not provide anymore insight about the consumer habits of YouTube viewers.

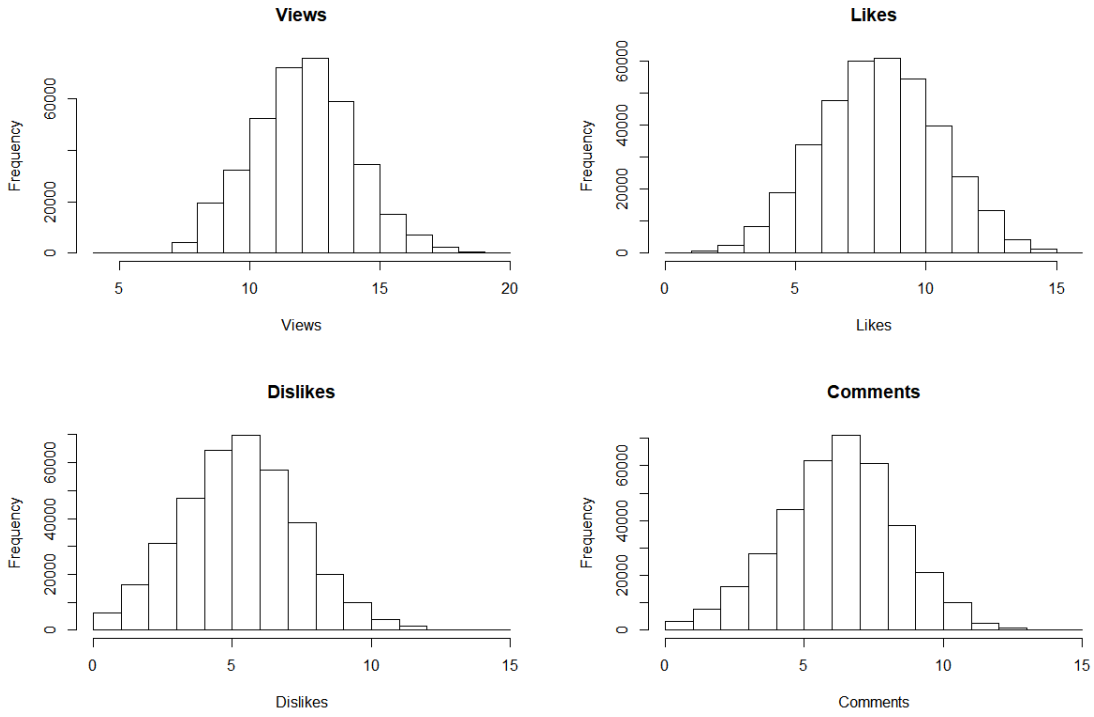

Figure 6: Histogram of variables after a log transformation

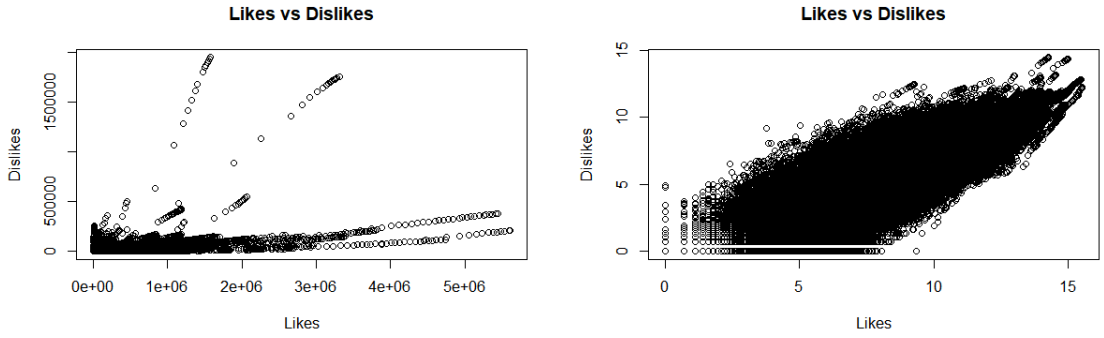

Figure 7: L: Raw number of likes plotted with dislikes. R: Log-transformed number of likes plotted with log number of dislikes

PCA and Other Methods

The perceived multivariate normal structure of the log-transformed data gives us an idea to try principal component analysis (PCA). Since the shape of a multivariate normal is ellipsoidal and, intuitively, PCA performs a change of basis to fit data to an ellipsoid with the axes corresponding to variations in the data.

We also attempted to generate SVM and neural network models without success.

In the case of SVMs, we thought to implement a support vector machine (SVM) on the data set to see if any kind of higher-dimensional classification structure might exist and become apparent, but we never found success in getting one to work effectively. Due to the sheer size of the data set, modeling an SVM was an intractable task on our home computer. We attempted analysis on subsets of the global data, splitting it up by region or category, then trying to train an SVM. We could train some SVM's, but every kernel except a linear kernel would result in a ridiculously large number of support vectors (around 39,000 support vectors for a data set with 40,000 observations). The linear SVM we were able to generate had 14,000 support vectors - still a lot but something that might be possible to work with. However, when we went to make predictions using our linear SVM, we could not make predictions, and we could not plot. Of course, since SVMs are highly parameter-dependent we tried different parameters, but each training cycle would take 5-10 minutes just to see if a model could be fitted, if it didn't encounter an error along the way. We received numerous errors trying to fit an SVM, but one of the most astounding ones said "Error: cannot allocate vector of size 10.0 Gb". Due to the time investment it would take to tune the many parameters of an SVM, and the frustration of copious errors and possible hardware limitations, we chose to forego any further attempts at generating an SVM.

For neural networks, we would have wanted to perform textual processing and analysis of titles, tags, and descriptions along with views, likes, dislikes, and comment count to try to train a neural network and find the fine relationships between our numeric and text variables. Unfortunately, we were not able to implement a neural network, partially due to errors and possible hardware limitations. Among our errors was a code reading "W tensorflow/core/framework/cpu_allocator_impl.cc:81] Allocation of 373492832 exceeds 10% of free system memory." This leads us to believe that part of the problem we encountered with training a neural network as our hardware. Frankly, we also simply did not have as much time as we wanted to be able to sit down and figure things out. This is still a skill we want to develop, and we may try it again in the future on a smaller, more manageable data set but we have no neural network to turn in for this assignment except scratchwork code.

Standard Deviations			
1.6872024	0.8450078	0.6075770	0.2648774

Loadings	PC1	PC2	PC3	PC4
Views	-0.4760740	-0.5631718	-0.5676547	-0.3660044
Likes	-0.5393249	-0.3537354	0.3733230	0.6668057
Dislikes	-0.4449097	0.6905684	-0.4947726	0.2834973
Comments	-0.5334206	0.2842957	0.5418470	-0.5839865

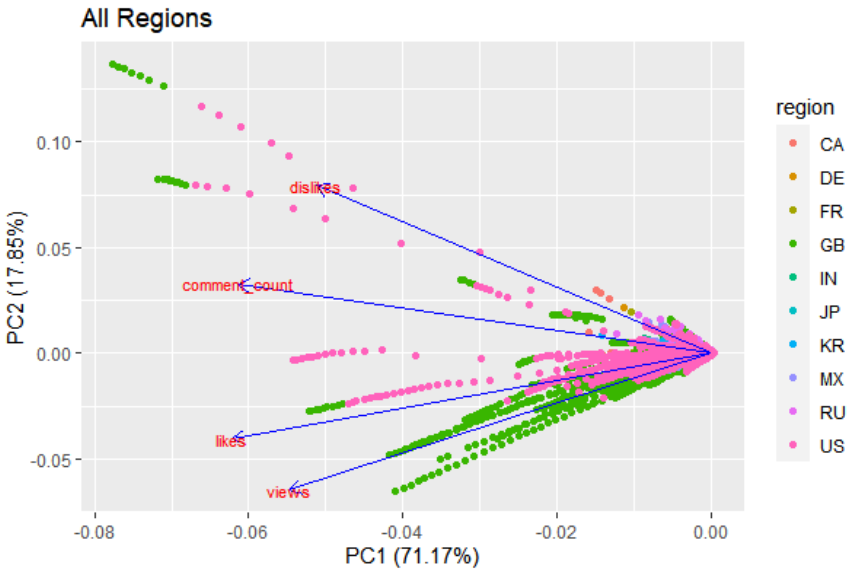

Figure 8: Top: Standard deviations and loadings for PCA of global trending videos. Bottom: Visualization of principal components.

Results

We perform PCA on the global set of trending videos and get the standard deviations and loadings found in Figure 8 (top) and visualize the loadings of our variables on the principal components in Figure 8 (bottom).

We see that between PC1 and PC2, roughly 89% of the variation in the data set is accounted for. With the new loadings along the PC1 and PC2, we see that, globally, likes and views are more related to each other than any other variables, and dislikes and comment counts are more related to each other than any other variables. Intuitively, this mean views and likes will trend together - an increase in views suggests an increase in likes more than it suggests an increase in dislikes or comments. Similarly, dislikes and comment counts will trend together as well which suggests viewers are more vocal about videos they dislike.

We perform PCA on the regional data sets, and plot the loadings for each variables on PC1 and PC2 in Figure 9 to see the different behaviors between regions. We notice that similarly to the global PCA, around 90% of the variation is accounted for with just two principal components. We do see that likes and views do strongly move together in the same direction for most of the regional plots, with the exception of JP. In JP, likes and comment counts trend together more strongly than any other two variables. This suggests that viewers in JP are more likely to comment on videos they enjoy than other regions.

Another thing to note is that in 7/10 regional plots, views and comment counts have very similar loadings, in some plots the relationship between views and comments is stronger than any other relationship. Yet, views and comments have a large gap between them in the global PCA plot. This is because of the leverage provided by the 3/10 regional plots that don't share the same relationship between views and comments - GB, IN, and US all have loadings such that comments trend distinctly from views. Its their larger viewership which heavily influences the trends in the global PCA plot, as explained in the EDA where British and American tastes dominated the most viewed, disliked, and commented on videos. The trending behaviors of GB and US viewers mask the more common behaviors found in other viewing regions.

In 8/10 PCA plots, the most different variables in the PC1 and PC2 chart are likes and dislikes. Intuitively, this makes sense; videos with a large number of likes are likely to be different from videos with a large number of dislikes. Since these PCA plots do a good job of visualizing the variation among variables within a data set, loading likes and dislikes as differently as possible accounts for as much variation as possible. However, there are two major exceptions, and they are once again GB and US. Their most differently loaded variables were views and dislikes. Although likes are still strongly related to views in both plots, the largest difference among loadings is still, for some reason, between views and dislikes, which might suggest viewers in these regions are a little more stingy with their likes, and are less likely to like a video than viewers in other regions.

Conclusions

Principal component analysis is the only method we were able to fully implement, although we feel we did gain some valuable insight into the different viewing habits of viewers in other countries. With just two principal components, we are able to account for roughly 90% of the variation in the data, both on the global data set and for each regional subset. PCA reveals that the viewing habits of British and American viewers of trending YouTube videos behave differently from viewers from other regions. Namely, the data suggest GB and US viewers are ruder, and more prone to comment on a video they dislike than other viewers, implying that comments left on a disliked video are negative comments. In contrast, viewers from JP are kinder and are more likely to comment on videos they like, implying comments left on a liked video are positive. Despite the minority behavior from GB and US, their activity has a large influence on global YouTube behavior. All throughout our analysis, from the EDA to

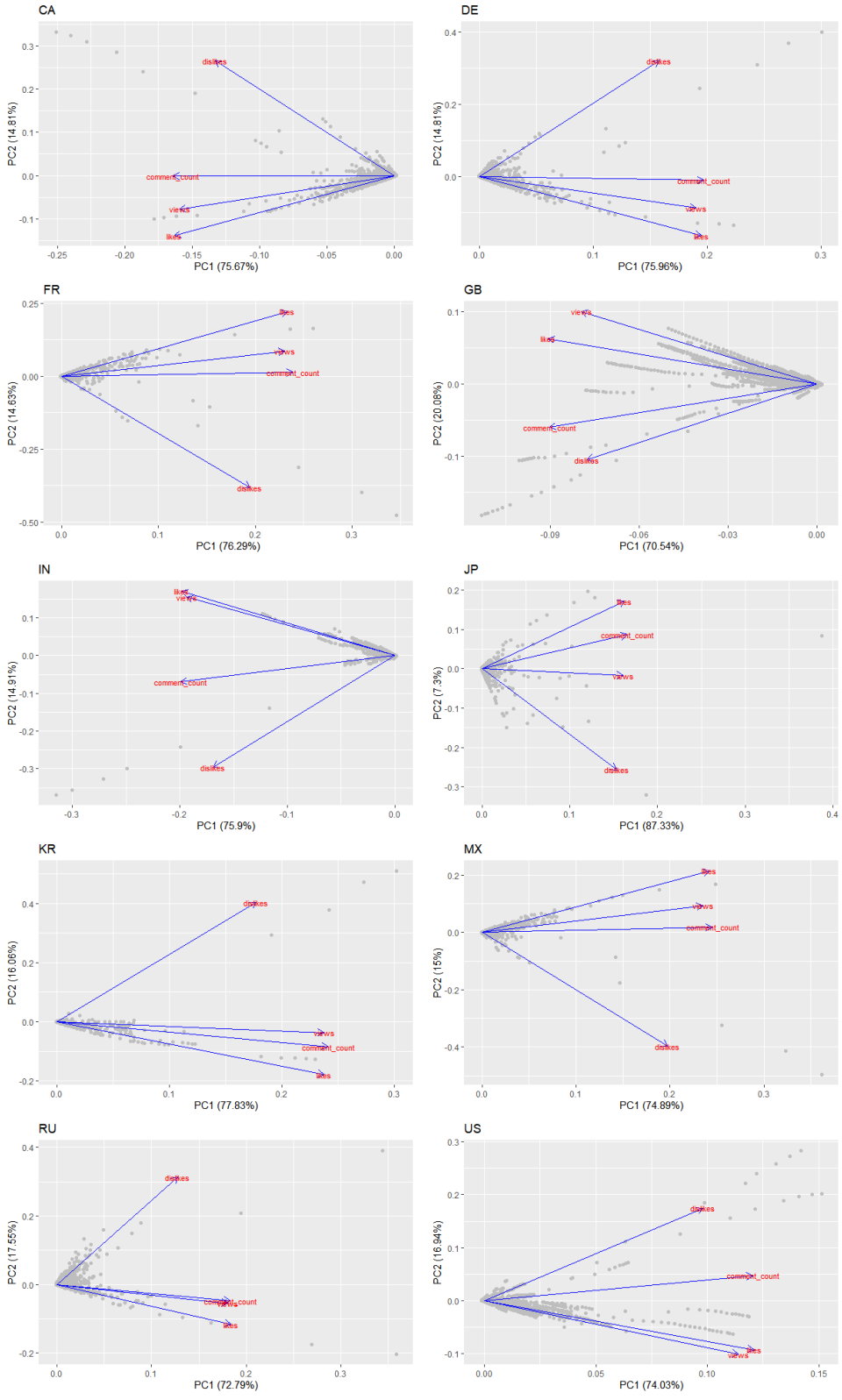

Figure 9: Regional PCAs

the PCA, GB and US observations have had incredible leverage on the summary data and statistics.

We would have wanted to build a neural network if we had more time. But now we wonder neural network is the best method for textual analysis and processing on such a large data set. Neural networks can grow prohibitively large as the number of neurons and layers grow. The first layer would have to be pretty large to accommodate the many possible words that could appear in a title, tag, or description, and include the rest of our counted variables (views, likes, dislikes, comments) too. That would only be the first layer. Then there would have had to have been many different connections between the first layer and hidden layers or output layer.

The more we think about it, the more difficult we realize implementing a neural network would have been, and the more we doubt whether our computer could have performed it. This leads us to ask what other methods are available for text processing. If we had more time to work on this project, future work would including researching alternate methods for text processing and analysis. We defaulted to neural networks, because it was the only one we knew.

I think I might have bit off more than I could chew with this data set. Surprisingly, there was not as much literature review of trending YouTube videos as I expected. I thought it would be a popular topic in the (mis)information age. The volume of data was difficult to work with, it would take minutes, over 20 minutes at some points, to fail at creating a model. In hindsight, it might have been a better idea to work with just a regional subset of trending YouTube videos for all of my analysis, or to have chosen another more established and well-studied data set altogether. It was a challenge, and I struggled with it, but there is no denying it is a very rich data set, and I had a lot of fun exploring it.

References

1. A. Alyousfi, "Analysis of YouTube Trending Videos of 2019 (US)", https://ammar-alyousfi.com/2020/youtube-trending-videos-analysis-2019-us, Published 06 Jul 2020,, Accessed 13 Dec 2020

2. A. Alyousfi, "YouTube Trending Videos Analysis", https://www.kaggle.com/ammar111/youtube-trending-videos-analysis, Published 02 Oct 2018, Accessed 13 Dec 2020

3. I. Barjasteh, Y. Liu, H. Radha, "Trending Videos: Measurement and Analysis", https://arxiv.org/ftp/ Submitted 26 Sep 2014, Accessed 13 Dec 2020

4. K. Maclver , "Youtube Trending Videos Analysis", https://medium.com/@kmacver/youtube-trending-videos-analysis-5d0c248c6a83, Published 06 Nov 2019, Accessed 13 Dec 2020

5. Mitchell J., "Trending YouTube Video Statistics", https://www.kaggle.com/datasnaek, Updated 02 Jun 2019, Accessed 13 Dec 2020

6. YouTube Official Blog, https://blog.youtube/press/, Accessed 13 DEC 2020

Appendix A: Category ID key

1. Film & Animation
2. Autos & Vehicles
10. Music
15. Pets & Animals
17. Sports
18. Short Movies
19. Travel & Events
20. Gaming
21. Videoblogging
22. People & Blogs
22. People & Blogs
23. Comedy
24. Entertainment
25. News & Politics
26. Howto & Style
27. Education
28. Science & Technology
29. Nonprofits & Activism

Additional categories (reserved for commercial uploads)
30. Movies
31. Anime/Animation
32. Action/Adventure
33. Classics
34. Comedy
35. Documentary
36. Drama
37. Family
44. Trailers
38. Foreign
39. Horror
40. Sci-Fi/Fantasy
41. Thriller
42. Shorts
43. Shows

Appendix: Pardede_533_Final.R

```
#### Cesar Pardede ####
# Final Project
# Math 533
# 2020-11-19

library(lubridate)
library(chron)
library(e1071)
library(plyr)

#### Analysis of YouTube Data ####
youtube_CA = read.csv('Fall 2020/Math 533/Final/archive/
   CAvideos.csv', stringsAsFactors = F)
youtube_DE = read.csv('Fall 2020/Math 533/Final/archive/
   DEvideos.csv', stringsAsFactors = F)
youtube_FR = read.csv('Fall 2020/Math 533/Final/archive/
   FRvideos.csv', stringsAsFactors = F)
youtube_GB = read.csv('Fall 2020/Math 533/Final/archive/
   GBvideos.csv', stringsAsFactors = F)
youtube_IN = read.csv('Fall 2020/Math 533/Final/archive/
   INvideos.csv', stringsAsFactors = F)
youtube_JP = read.csv('Fall 2020/Math 533/Final/archive/
   JPvideos.csv', stringsAsFactors = F)
youtube_KR = read.csv('Fall 2020/Math 533/Final/archive/
   KRvideos.csv', stringsAsFactors = F)
youtube_MX = read.csv('Fall 2020/Math 533/Final/archive/
   MXvideos.csv', stringsAsFactors = F)
youtube_RU = read.csv('Fall 2020/Math 533/Final/archive/
   RUvideos.csv', stringsAsFactors = F)
youtube_US = read.csv('Fall 2020/Math 533/Final/archive/
   USvideos.csv', stringsAsFactors = F)
youtube_list = list(youtube_CA, youtube_DE, youtube_FR,
   youtube_GB, youtube_JP, youtube_KR, youtube_MX, youtube_RU,
   youtube_US)

# add region column
youtube_CA['region'] = 'CA'
youtube_DE['region'] = 'DE'
youtube_FR['region'] = 'FR'
youtube_GB['region'] = 'GB'
youtube_IN['region'] = 'IN'
youtube_JP['region'] = 'JP'
```

```
youtube_KR['region'] = 'KR'
youtube_MX['region'] = 'MX'
youtube_RU['region'] = 'RU'
youtube_US['region'] = 'US'
regions = c('CA', 'DE', 'FR', 'GB', 'IN', 'JP', 'KR', 'MX', 'RU
   ', 'US')
youtube = rbind(youtube_CA, youtube_DE, youtube_FR, youtube_GB,
    youtube_IN, youtube_JP, youtube_KR, youtube_MX, youtube_RU,
    youtube_US)
youtube = data.frame(youtube)
youtube[, 'region'] = as.factor(youtube[, 'region'])

## DATA CLEANING ##

# formatting true and false entries
tf_columns = c("comments_disabled", "ratings_disabled", "
   video_error_or_removed")
for (columns in tf_columns){
        youtube[, columns] = gsub('FALSE', 'False', youtube[,
           columns])
        youtube[, columns] = gsub('TRUE', 'True', youtube[,
           columns])
        youtube[, columns] = as.factor(youtube[, columns])
}

# read category_id as factors
youtube[, 'category_id'] = as.factor(youtube[, 'category_id'])
cat_key = rep(NA, 44)
cat_key[1] = 'Film & Animation'
cat_key[2] = 'Autos & vehicles'
cat_key[10] = 'Music'
cat_key[15] = 'Pets'
cat_key[17] = 'Sports'
cat_key[18] = 'Short Movies'
cat_key[19] = 'Travel & Events'
cat_key[20] = 'Gaming'
cat_key[21] = 'Videoblogging'
cat_key[22] = 'People & Blogs'
cat_key[23] = 'Comedy'
cat_key[24] = 'Entertainments'
cat_key[25] = 'News & Politics'
cat_key[26] = 'Howto & Style'
cat_key[27] = 'Education'
cat_key[28] = 'Science & Technology'
cat_key[29] = 'Nonprofits & Activism'
```

```
cat_key[30] = 'Movies'
cat_key[31] = 'Anime/Animation'
cat_key[32] = 'Action/Adventure'
cat_key[33] = 'Classics'
cat_key[34] = 'Comedy'
cat_key[35] = 'Documentary'
cat_key[36] = 'Drama'
cat_key[37] = 'Family'
cat_key[38] = 'Foreign'
cat_key[39] = 'Horror'
cat_key[40] = 'Sci-Fi/Fantasy'
cat_key[41] = 'Thriller'
cat_key[42] = 'Shorts'
cat_key[43] = 'Shows'
cat_key[44] = 'Trailers'

# extract publish date, edit publish time
publish_raw = ymd_hms(youtube[, "publish_time"])
youtube[, 'publish_date'] = as.Date(publish_raw)
youtube[, 'publish_time'] = times(gsub(".* ", '', publish_raw))

# format trending date and video id
youtube[, "trending_date"] = as.Date(ydm(youtube[, "
   trending_date"]))
youtube[, "video_id"] = as.character(youtube[, "video_id"])

# drop unneeded columns, rearrange columns
youtube = subset(youtube, select = -c(thumbnail_link)) # keep
   video_id to make querying videos easier
reorder_columns = c("trending_date", "video_id", "title", "
   channel_title", "category_id", "publish_date", "publish_time
   ", "tags",
"views", "likes", "dislikes", "comment_count", "
   comments_disabled", "ratings_disabled", "
   video_error_or_removed",
"region","description")
youtube = youtube[, reorder_columns]

#### EXPLORATORY DATA ANALYSIS (EDA) ####

# get quick summaries
summary(youtube)
i = 1
summaries = list()
for(r in regions){
```

```
        summaries[[i]] = summary(youtube[which(youtube['region
           '] == r), ])
        i = i + 1
}
summaries

# most viewed, liked, disliked, and commented vidoes from each
   region
i = 1
region_yt_max = list()
for (r in regions){
        yt = youtube[youtube[, 'region'] == r, ]
        region_max_index = apply(yt[, c('views', 'likes', '
           dislikes', 'comment_count')], 2, which.max)
        region_max_title = yt[region_max_index, 'title']
        region_max_channel = yt[region_max_index, '
           channel_title']
        region_max_count = apply(yt[, c('views', 'likes', '
           dislikes', 'comment_count')], 2, max)
        region_yt_max[[i]] = list(region = r,
        'most viewed' = c(
        'title' = region_max_title[1], 'channel' =
           region_max_channel[1], region_max_count[1]),
        'most liked' = c(
        'title' = region_max_title[2], 'channel' =
           region_max_channel[2], region_max_count[2]),
        'most disliked' = c(
        'title' = region_max_title[3], 'channel' =
           region_max_channel[3], region_max_count[3]),
        'most commented' = c(
        'title' = region_max_title[4], 'channel' =
           region_max_channel[4], region_max_count[4]))
        i = i + 1
}
region_yt_max

# most viewed, liked, disliked, and commented vidoes globally
max_index = apply(youtube[, c('views', 'likes', 'dislikes', '
   comment_count')], 2, which.max)
max_title = youtube[max_index, 'title']
max_channel = youtube[max_index, 'channel_title']
max_count = apply(youtube[, c('views', 'likes', 'dislikes', '
   comment_count')], 2, max)
yt_max = list(region = 'Global',
'most viewed' = c(
```

```
'title' = max_title[1], 'channel' = max_channel[1], max_count
   [1]),
'most liked' = c(
'title' = max_title[2], 'channel' = max_channel[2], max_count
   [2]),
'most disliked' = c(
'title' = max_title[3], 'channel' = max_channel[3], max_count
   [3]),
'most commented' = c(
'title' = max_title[4], 'channel' = max_channel[4], max_count
   [4]))
yt_max

# many metrics are approximately lognormal? what does it mean?
# par(mfrow = c(2, 2))
# hist(youtube[, "views"], main = 'Views', xlab = 'Views')
# hist(youtube[, "likes"], main = 'Likes', xlab = 'Likes')
# hist(youtube[, "dislikes"], main = 'Dislikes', xlab = '
   Dislikes')
# hist(youtube[, "comment_count"], main = 'Comments', xlab = '
   Comments')
#
# hist(log(youtube[, "views"]), main = 'Views', xlab = 'Views')
# hist(log(youtube[, "likes"]), main = 'Likes', xlab = 'Likes')
# hist(log(youtube[, "dislikes"]), main = 'Dislikes', xlab = '
   Dislikes')
# hist(log(youtube[, "comment_count"]), main = 'Comments', xlab
    = 'Comments')
# par(mfrow = c(1,1))

# trending quantiles
trending_CI = apply(youtube[, c('views', 'likes', 'dislikes', '
   comment_count')], 2, quantile, probs = c(0, 0.50, 0.95))
trending_CI
views_CI = quantile(youtube[, 'views'], probs = c(0, 0.50,
   0.95))
views_CI

# how many days to trend after publishing?
days_to_trend = as.numeric(youtube[, "trending_date"] - youtube
   [, "publish_date"])
days_CI = quantile(days_to_trend, probs = c(0, 0.50, 0.95))
whazzup = list(video = youtube[which.max(days_to_trend), ],
days = max(days_to_trend))
days_CI
```

```
# the whazzup video resurfaced after 4215 days to trend for 1
   day on 05 February
# in the US - corresponding with the Philadelphia Eagles win
   over the New England
# Patriots in Super Bowl LII the day before. I suspect this is
   a cyclical occurrence.
dim(youtube[youtube['title'] == 'Budweiser - Original Whazzup?
   ad', ])[1]

## MOST POPULAR VIDEOS PER CATEGORY PER REGION??? ##
cat_max = list()
i = 1
for (r in regions){
        yt = youtube[which(youtube['region'] == r), ]
        for (category in which(!is.na(cat_key))){
                yt_cat = yt[which(yt['category_id'] == category
                   ), ]
                max_view_cat = which.max(yt_cat[, 'views'])
                cat_max[[i]] = list('region' = r,
                'category' = cat_key[category],
                'category id' = category,
                'video id' = yt[max_view_cat, "video_id"],
                yt_cat[max_view_cat, c('title', 'channel_title
                   ', 'views', 'likes', 'dislikes', '
                   comment_count')])
                i = i + 1
        }
}
# cat_max

trending_counts = list()
i = 1
for (r in regions){
        yt = youtube[which(youtube['region'] == r), ]

        counts_title = count(yt, c('title'))
        counts_by_id = count(yt[which(yt['video_id'] != '#NAME
           ?'), ], c('video_id'))
        max_counts_title = as.character(counts_title[which.max(
           counts_title[,'freq']), 'title'])
        max_counts_id = counts_by_id[which.max(counts_by_id[,'
           freq']), 'video_id']
```

```
        trending_title = yt[which(yt[, 'title'] ==
           max_counts_title), c("trending_date", "video_id", "
           title", "channel_title", "publish_date",
        "views", "likes", "dislikes", "comment_count")]
        trending_by_id = yt[which(yt[, 'video_id'] ==
           max_counts_id), c("trending_date", "video_id", "
           title", "channel_title", "publish_date",
        "views", "likes", "dislikes", "comment_count")]
        trending_counts[[i]] = list('region' = r,
        'most trending by title' = trending_title[1, ], '#
           trending by title' = dim(trending_title)[1],
        'channel names' = unique(trending_title[, "
           channel_title"]),
        'most trending by ID' = trending_by_id[1, ], '#
           trending by ID ' = dim(trending_by_id)[1]
        )
        i = i + 1
}
trending_counts

global_trending_counts = list()
counts_title = count(youtube, c('title'))
counts_by_id = count(youtube[which(youtube['video_id'] != '#
   NAME?'), ], c('video_id'))
max_counts_title = as.character(counts_title[which.max(
   counts_title[,'freq']), 'title'])
max_counts_id = as.character(counts_by_id[which.max(
   counts_by_id[,'freq']), 'video_id'])

trending_title = youtube[which(youtube[, 'title'] ==
   max_counts_title), c("trending_date", "video_id", "title", "
   channel_title", "publish_date",
"views", "likes", "dislikes", "comment_count", 'region')]
trending_by_id = youtube[which(youtube[, 'video_id'] ==
   max_counts_id), c("trending_date", "video_id", "title", "
   channel_title", "publish_date",
"views", "likes", "dislikes", "comment_count", 'region')]
global_trending_counts = list('most trending by title' =
   trending_title[1, ], '# trending by title' = dim(
   trending_title)[1],
'channel names' = unique(trending_title[, "channel_title"]),
'most trending by ID' = trending_by_id[1, ], '# trending by ID
   ' = dim(trending_by_id)[1])
global_trending_counts
```

```
# par(mfrow = c(3, 1))
par(mfrow = c(1, 1))
hist(youtube[,'views'], main = 'Views', xlab = 'Views')
hist(youtube[,'likes'], main = 'Likes', xlab = 'Likes')
hist(youtube[,'dislikes'], main = 'Dislikes', xlab = 'Dislikes
   ')
hist(youtube[,'comment_count'], main = 'Comments', xlab = '
   Comments')

hist(log(youtube[,'views']), main = 'Views', xlab = 'Views')
hist(log(youtube[,'likes']), main = 'Likes', xlab = 'Likes')
hist(log(youtube[,'dislikes']), main = 'Dislikes', xlab = '
   Dislikes')
hist(log(youtube[,'comment_count']), main = 'Comments', xlab =
   'Comments')
par(mfrow = c(1, 1))

plot(youtube[,'views'], youtube[,'likes'], main = 'Views vs
   Likes', xlab = 'Views', ylab = 'Likes')
plot(youtube[,'views'], youtube[,'dislikes'], main = 'Views vs
   Dislikes', xlab = 'Views', ylab = 'Dislikes')
plot(youtube[,'likes'], youtube[,'dislikes'], main = 'Likes vs
   Dislikes', xlab = 'Likes', ylab = 'Dislikes')

plot(log(youtube[,'views']), log(youtube[,'likes']), main = '
   Views vs Likes', xlab = 'Views', ylab = 'Likes')
plot(log(youtube[,'views']), log(youtube[,'dislikes']), main =
   'Views vs Dislikes', xlab = 'Views', ylab = 'Dislikes')
plot(log(youtube[,'likes']), log(youtube[,'dislikes']), main =
   'Likes vs Dislikes', xlab = 'Likes', ylab = 'Dislikes')

# top 3 dislikes/likes
youtube_dislikes = youtube[order(youtube[,'dislikes'],
   decreasing = T),]
unique(youtube_dislikes[, c('title', "channel_title")])[1:3, ]

youtube_likes = youtube[order(youtube[,'likes'], decreasing = T
   ),]
unique(youtube_likes[, c('title', "channel_title")])[1:3, ]

youtube[which(youtube[,'likes'] == sort(youtube[,'likes'],
   decreasing = T)[1]), ]
youtube[which(youtube[,'likes'] == sort(youtube[,'likes'],
   decreasing = T)[2]), ]
```

```
#### STATISTICAL LEARNING AND MODELING ####
n = dim(youtube)[1]
library(ggfortify)
pca_youtube = youtube[, c("views", "likes", "dislikes", "
   comment_count")]
pca = prcomp(pca_youtube, scale. = T)
autoplot(pca, data = youtube, colour = 'region', loadings =
   TRUE, loadings.colour = 'blue',
loadings.label = TRUE, loadings.label.size = 3, main = 'All
   Regions')
pca
# autoplot(pcl, data = pca_logtube) # fails because pcl fails

# for some reason looping through region names doesn't display
   plots. manually do pca for all regions.
r = 'CA'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'DE'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'FR'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
```

```
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'GB'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'IN'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'JP'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'KR'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
```

```
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'MX'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'RU'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
r = 'US'
youtube_region = youtube[which(youtube['region'] == r), ]
pca_youtube = youtube_region[, c("views", "likes", "dislikes",
   "comment_count")]
pca = prcomp(pca_youtube, scale. = T)
pca
autoplot(pca, data = youtube_region, colour = 'grey', loadings
   = TRUE,
loadings.colour = 'blue', loadings.label = TRUE, loadings.label
   .size = 3,
main = r)
```