Table of Contents

Introduction：

This file contains all the codes I used to perform the analysis and all the outputs. Techniques I used included merging data, cleaning data, exploring data from different perspectives, bigram, wordcloud, and clustering.

Part 1: Data preparation

1. Data import

Import four data files

```
> library(readr)
> usv <- read_csv("Desktop/Hulu project/USvideos.csv")
> usc <- read_csv("Desktop/Hulu project/UScomments.csv")
> gbv <- read_csv("Desktop/Hulu project/GBvideos.csv")
> gbc <- read_csv("Desktop/Hulu project/GBcomments.csv")
```

2. Merge data and add location

```
> uscombine <- merge(x = usv, y = usc, by = "video_id", all.x = TRUE)
> gbcombine <- merge(x = gbv, y = gbc, by = "video_id", all.x = TRUE)
> uscombine$country = "US"
> gbcombine$country = "GB"
> allin = rbind(uscombine,gbcombine)
```

rename columns as there are “likes” columns in both the video and the comment datasets

```
> colnames(allin)[colnames(allin)=="likes.x"] <- "likes_video"
> colnames(allin)[colnames(allin)=="likes.y"] <- "likes_comment"
```

3. Summarize data

Summary and look at the structure of the data

```
> summary(allin)
```

There are some missing values, and the data type of the date column is inaccurate.

4. Missing values

Further check missing values and their importance on our analysis

```
> sum(is.na(allin))
[1] 7439
> sum(is.na(allin$date))
[1] 6559
> sum(is.na(allin$comment_text))
[1] 301
> sum(is.na(allin$likes.y))
[1] 257
> sum(is.na(allin$replies))
[1] 322
```

The number of missing values is very small portion of our data (0.18%), and some missing values are in the same row. Also, for text and date missing values, we cannot simply replace them with mode or mean values. Therefore, we can remove missing values for better analysis.

remove missing values

```
> all <- allin[complete.cases(allin), ]
> sum(is.na(all))
[1] 0
```

5. Correct data type of the date column

Change data type of the date column

```
> all$date <- as.Date(as.character(all$date), format='%d.%m')
> all$date <- format(all$date, "%m-%d")
```

Limit date from September 13th to September 30th

```
> all <- all[all$date >= "09-13" & all$date <= "09-30",]
```

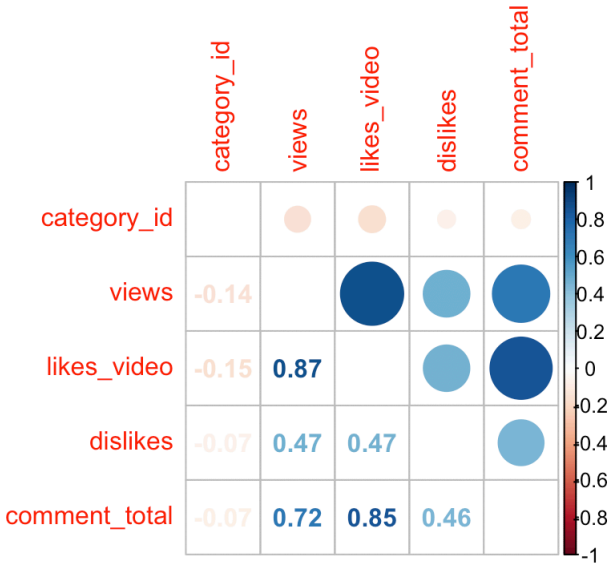
Part 2: Exploratory data analysis

1. Correlation among category id, views, likes, dislikes, and total comments

```
> library(corrplot)
> corrplot.mixed(corr =cor(all[,c("category_id", "views","likes_video","dislikes", "comment_total")]),tl.pos = "lt")
```

From the correlation plot below, we can see that the number of likes and views are highly correlated. Total comments are also correlated with number of views and likes. People are more likely to leave comments when they like the videos compared to dislikes.

Output:



2. Top trending channels/videos/categories based on # comments

Top 10 most popular/least popular trending channels in US

subset data to exclude specific comment information

```
> newdf = unique(all[,c("title","channel_title", "category_id",
"views","likes_video","dislikes","comment_total","date","country")])
```

as metrics are running sums instead of daily totals, we first select maximum comments in total for each video, and then sum these maximum total comments in the channel dimension

```
> channel_us1 <- newdf %>% select(title,channel_title,comment_total,country) %>% filter(country == "US") %>%
group_by(title, channel_title) %>% summarise(n = max(comment_total))
```

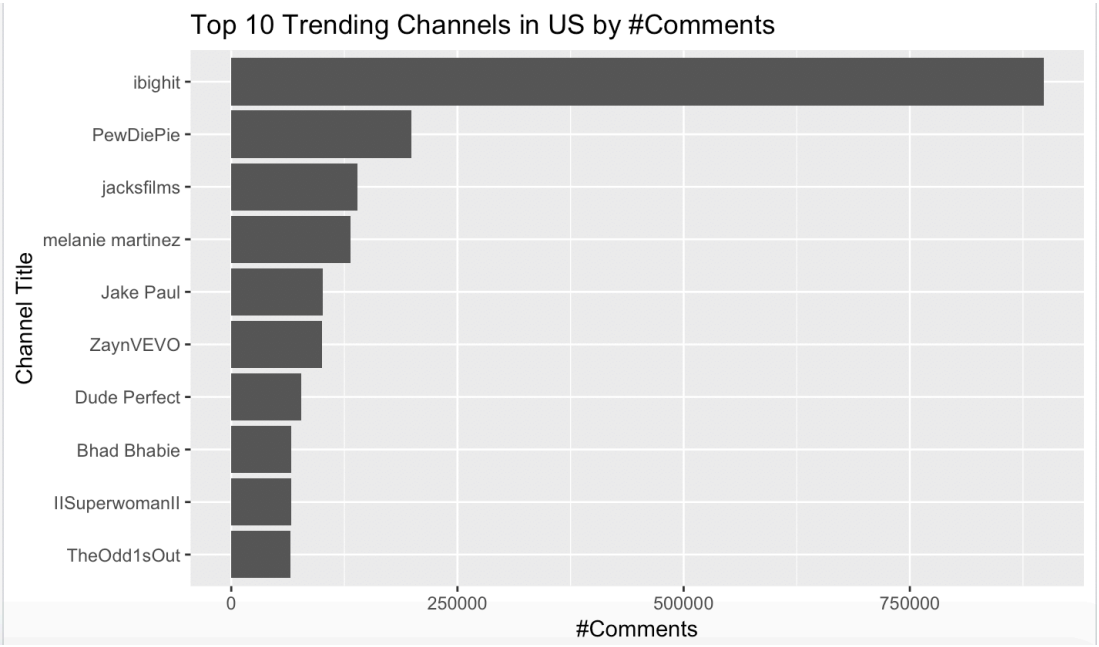
```
> channel_us <- channel_us1 %>% group_by(channel_title) %>% summarise(n_of_comments = sum(n)) %>%
arrange(desc(n_of_comments)) %>% head(10)
> View(channel_us)
```

Output:

	channel_title	n_of_comments
1	ibighit	897821
2	PewDiePie	199406
3	jacksfilms	139379
4	melanie martinez	132061
5	Jake Paul	101387
6	ZaynVEVO	100661
7	Dude Perfect	77570
8	Bhad Bhabie	66509
9	IISuperwomanII	65906
10	TheOdd1sOut	65764

```
> library(ggplot2)
> channel_us$channel_title <- factor(channel_us$channel_title, levels =
channel_us$channel_title[order(channel_us$n_of_comments)])
> p_channel_us <-ggplot(data=channel_us, aes(x=channel_title, y=n_of_comments)) +geom_bar(stat = "identity") +
xlab("Channel Title") + ylab("#Comments") + ggtitle("Top 10 Trending Channels in US by #Comments")+ coord_flip()
> p_channel_us
```

Output:



we can also have a look at 10 least popular channels in US

```
> channel_us_t <- channel_us1 %>% group_by(channel_title) %>% summarise(n_of_comments = sum(n)) %>%
arrange(desc(n_of_comments)) %>% tail(10)
> View(channel_us_t)
```

Output:

	channel_title	n_of_comments
1	madison.com	4
2	Video Detective	4
3	Xposure 365 TV	4
4	Adam Sifounakis	3
5	Cosmic Book News	3
6	Malhar Takle	3
7	МИР УДИВЛЯЕТ v2.0	3
8	Fathom Events	2
9	Rad Universe	2
10	Vertical Entertainment LA	1

Top 10 most popular/least popular trending channels in US

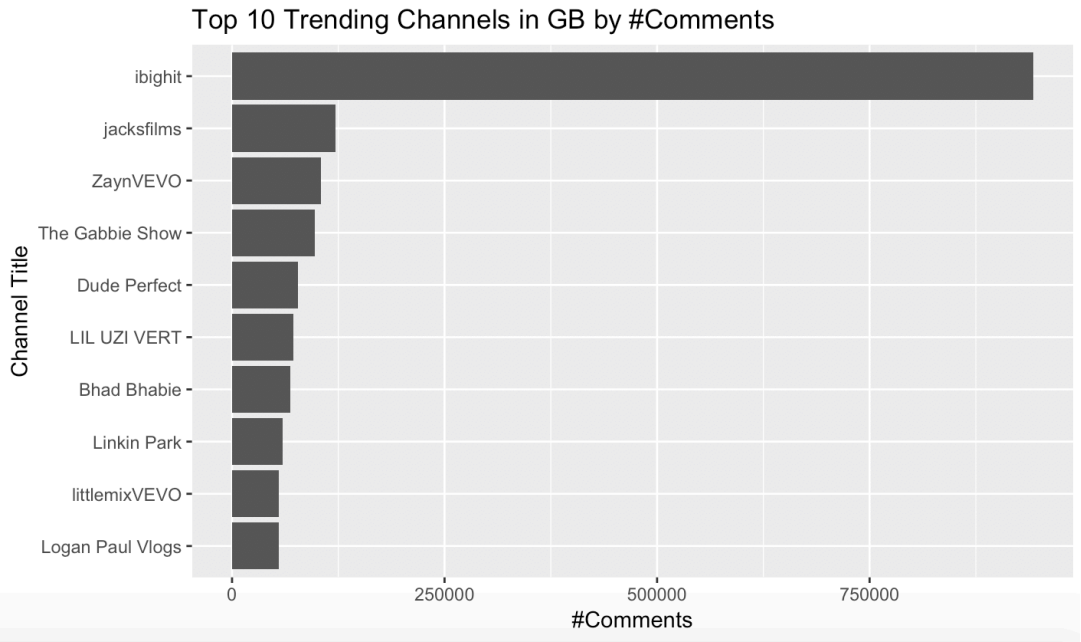
```
> channel_gb1 <- newdf %>% select(title,channel_title,comment_total,country) %>% filter(country == "GB") %>%
group_by(title, channel_title) %>% summarise(n = max(comment_total))
> channel_gb <- channel_gb1 %>% group_by(channel_title) %>% summarise(n_of_comments = sum(n)) %>%
arrange(desc(n_of_comments)) %>% head(10)
> View(channel_gb)
```

Output:

	channel_title	n_of_comments
1	ibighit	942617
2	jacksfilms	121442
3	ZaynVEVO	104931
4	The Gabbie Show	97434
5	Dude Perfect	77562
6	LIL UZI VERT	72329
7	Bhad Bhabie	68640
8	Linkin Park	59392
9	littlemixVEVO	55008
10	Logan Paul Vlogs	54666

```
> channel_gb$channel_title <- factor(channel_gb$channel_title, levels =
channel_gb$channel_title[order(channel_gb$n_of_comments)])
> p_channel_gb <-ggplot(data=channel_gb, aes(x=channel_title, y=n_of_comments)) +geom_bar(stat = "identity") +
xlab("Channel Title") + ylab("#Comments") + ggtitle("Top 10 Trending Channels in GB by #Comments")+ coord_flip()
> p_channel_gb
```

Output:



Ibighit, ZaynVEVO, Bhad Bhabie, jacksfilms and Dude Perfect are top popular channels in both the United States and Great Britain.

have a look at the top 10 least popular channels in GB

```
> channel_gb_t <- channel_gb1 %>% group_by(channel_title) %>% summarise(n_of_comments = sum(n)) %>%
arrange(desc(n_of_comments)) %>% tail(10)
> View(channel_gb_t)
```

Output:

	channel_title	n_of_comments
1	Mean Girls on Broadway	8
2	XQ America	8
3	John Plunkett	7
4	My Football Views	7
5	The Paramount Vault	7
6	SaintsFan1964	5
7	y2ksale	5
8	James HansonTv	3
9	PoeticConvict	3
10	ArrowFilmsUK	1

Top 10 most popular/least popular trending videos in US

we select maximum total comments for each video, and there is no need to aggregate.

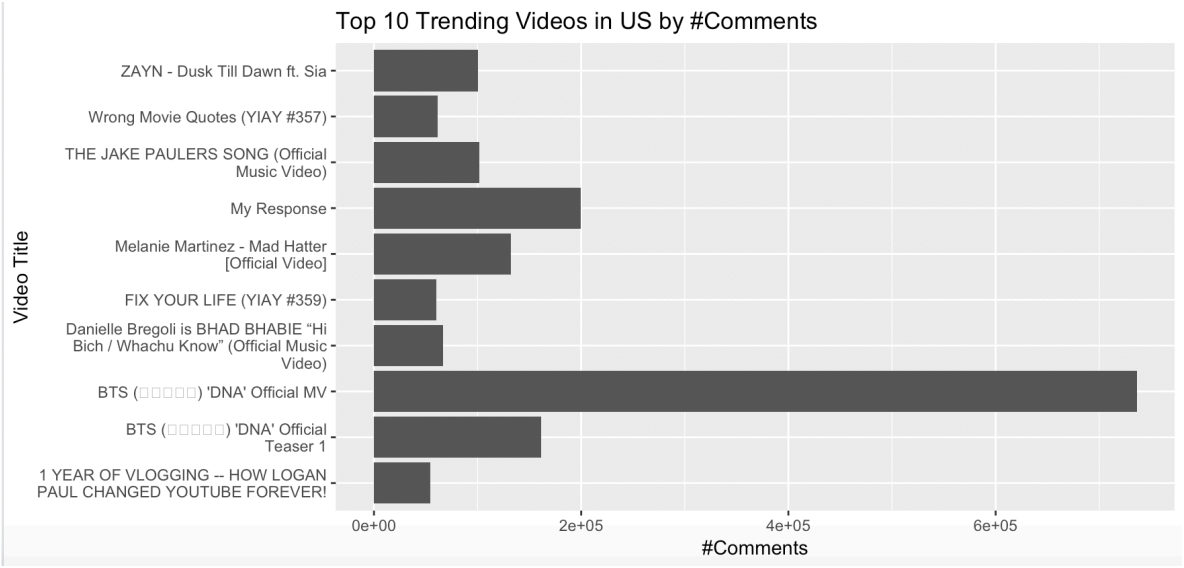
```
> video_us <- newdf %>% filter(country == "US") %>% group_by(title) %>% summarise(n_of_comments=
max(comment_total)) %>% arrange(desc(n_of_comments)) %>% head(10)
> View(video_us)
```

Output:

	title	n_of_comments
1	BTS (방탄소년단) 'DNA' Official MV	736179
2	My Response	199406
3	BTS (방탄소년단) 'DNA' Official Teaser 1	161642
4	Melanie Martinez - Mad Hatter [Official Video]	132061
5	THE JAKE PAULERS SONG (Official Music Video)	101387
6	ZAYN - Dusk Till Dawn ft. Sia	100661
7	Danielle Bregoli is BHAD BHABIE "Hi Bich / Whachu Kn...	66509
8	Wrong Movie Quotes (YIAY #357)	61179
9	FIX YOUR LIFE (YIAY #359)	60491
10	1 YEAR OF VLOGGING -- HOW LOGAN PAUL CHANGE...	54671

```
> p_video_us <-ggplot(data=video_us, aes(stringr::str_wrap(title, 35), y=n_of_comments)) + geom_bar(stat = "identity") +
xlab("Video Title") + ylab("#Comments") + ggtitle("Top 10 Trending Videos in US by #Comments")+ coord_flip()
> p_video_us
```

Output:



Top 10 most popular/least popular trending videos in GB

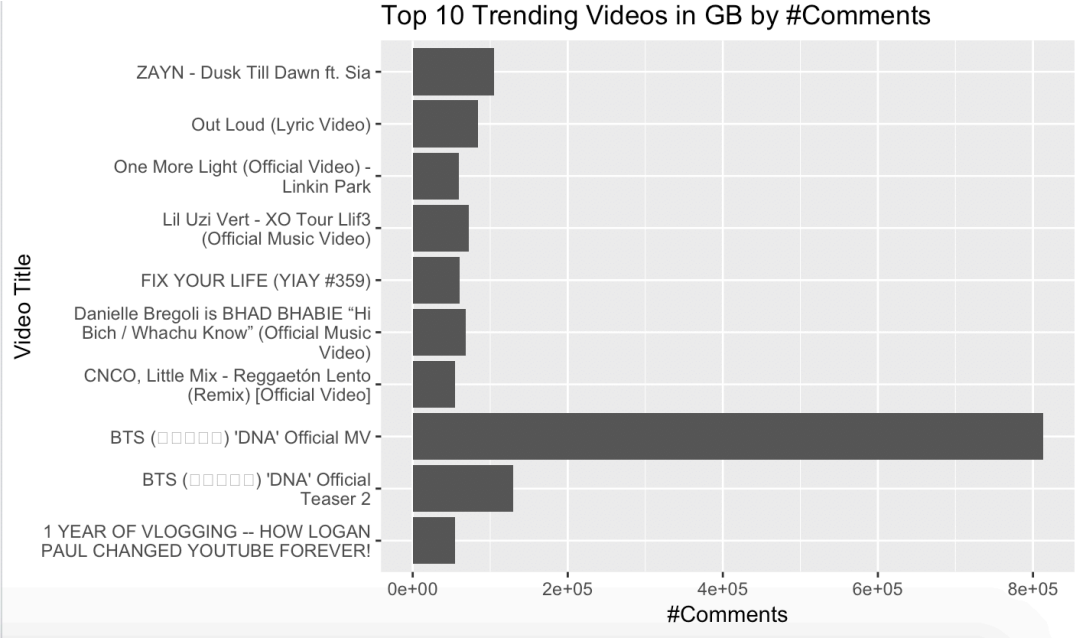
```
> video_gb <- newdf %>% filter(country == "GB") %>% group_by(title) %>% summarise(n_of_comments=
max(comment_total)) %>% arrange(desc(n_of_comments)) %>% head(10)
> View(video_gb)
```

Output:

	title	n_of_comments
1	BTS (방탄소년단) 'DNA' Official MV	813322
2	BTS (방탄소년단) 'DNA' Official Teaser 2	129295
3	ZAYN - Dusk Till Dawn ft. Sia	104931
4	Out Loud (Lyric Video)	83992
5	Lil Uzi Vert - XO Tour Llif3 (Official Music Video)	72329
6	Danielle Bregoli is BHAD BHABIE "Hi Bich / Whachu Kn...	68640
7	FIX YOUR LIFE (YIAY #359)	60484
8	One More Light (Official Video) - Linkin Park	59392
9	CNCO, Little Mix - Reggaetón Lento (Remix) [Official ...	55008
10	1 YEAR OF VLOGGING -- HOW LOGAN PAUL CHANGE...	54666

```
> p_video_gb <-ggplot(data=video_gb, aes(stringr::str_wrap(title, 35), y=n_of_comments)) + geom_bar(stat = "identity") +
xlab("Video Title") + ylab("#Comments") + ggtitle("Top 10 Trending Videos in GB by #Comments")+ coord_flip()
> p_video_gb
```

Output:



Category Ranking in US/GB

For US:

First, we select maximum comments in total for each video, and then sum these maximum total comments in the category dimension.

```
> category_us1 <- newdf %>% select(title,category_id,comment_total,country) %>% filter(country == "US") %>% group_by(title) %>% summarise(category_id = max(category_id),n = max(comment_total))
> category_us <- category_us1 %>% group_by(category_id) %>% summarise(n_of_comments = sum(n)) %>% arrange(desc(n_of_comments))
> View(category_us)
```

Output:

	category_id	n_of_comments
1	10	1775623
2	22	697893
3	24	651983
4	23	616683
5	26	472507
6	28	275653
7	1	172836
8	25	159575
9	17	155132
10	27	115913
11	15	57553
12	20	53148
13	2	36957
14	19	17842
15	29	70

For GB:

```
> category_gb1 <- newdf %>% select(title,category_id,comment_total,country) %>% filter(country == "GB") %>%
group_by(title) %>% summarise(category_id = max(category_id),n = max(comment_total))
> category_gb <- category_gb1 %>% group_by(category_id) %>% summarise(n_of_comments = sum(n)) %>%
arrange(desc(n_of_comments))
> View(category_gb)
```

Output:

	category_id	n_of_comments
1	10	1826604
2	24	530902
3	26	398455
4	22	391190
5	23	358146
6	17	139307
7	20	132979
8	1	124478
9	28	77582
10	27	65494
11	25	55852
12	15	51821
13	2	13076
14	19	2042
15	29	28

3. Top trending channels/videos/categories based on # views

Top 10 trending channels in US

```
> channel_us_views1 <- newdf %>% select(title,channel_title,views,country) %>% filter(country == "US") %>% group_by(title, channel_title) %>% summarise(n = max(views))
> channel_us_views <- channel_us_views1 %>% group_by(channel_title) %>% summarise(n_of_views = sum(n)) %>% arrange(desc(n_of_views)) %>% head(10)
> View(channel_us_views)
```

Output:

	channel_title	n_of_views
1	ibighit	56536494
2	ZaynVEVO	36323498
3	shakiraVEVO	32136948
4	SamSmithWorldVEVO	28906707
5	Dude Perfect	24317807
6	Jimmy Kimmel Live	18973161
7	Primitive Technology	15579127
8	BuzzFeedVideo	15412064
9	Brave Wilderness	13198292
10	Safiya Nygaard	12241650

Top 10 trending channels in GB

```
> channel_gb_views1 <- newdf %>% select(title,channel_title,views,country) %>% filter(country == "GB") %>%
group_by(title, channel_title) %>% summarise(n = max(views))
> channel_gb_views <- channel_gb_views1 %>% group_by(channel_title) %>% summarise(n_of_views = sum(n)) %>%
arrange(desc(n_of_views)) %>% head(10)
> View(channel_gb_views)
```

Output:

	channel_title	n_of_views
1	ibighit	68986323
2	SamSmithWorldVEVO	47265652
3	YandelVEVO	42486342
4	ZaynVEVO	41959549
5	shakiraVEVO	38144440
6	Dude Perfect	25586700
7	Warner Music	23699605
8	Jimmy Kimmel Live	18674841
9	LIL UZI VERT	17521230
10	littlemixVEVO	16529782

Top 10 trending videos in US

```
> video_us_views <-newdf %>% filter(country == "US") %>% group_by(title) %>% summarise(n_of_views =
max(views)) %>% arrange(desc(n_of_views)) %>% head(10)
> View(video_us_views)
```

Output:

	title	n_of_views
1	BTS (방탄소년단) 'DNA' Official MV	41500672
2	ZAYN – Dusk Till Dawn ft. Sia	36323498
3	Shakira – Perro Fiel (Official Video) ft. Nicky Jam	32136948
4	Primitive Technology: Mud Bricks	15579127
5	BTS (방탄소년단) 'DNA' Official Teaser 1	15035822
6	Sam Smith – Too Good At Goodbyes (Official Video)	14077967
7	Celebrities Read Mean Tweets #11	13740234
8	Sam Smith – Too Good At Goodbyes (Official Audio)	13728070
9	Nerf Bow Trick Shots | Dude Perfect	13686054
10	Danielle Bregoli is BHAD BHABIE "Hi Bich / Whachu Kn...	12115465

Top 10 trending videos in GB

```
> video_gb_views <-newdf %>% filter(country == "GB") %>% group_by(title) %>% summarise(n_of_views = max(views)) %>% arrange(desc(n_of_views)) %>% head(10)
> View(video_gb_views)
```

Output:

	title	n_of_views
1	BTS (방탄소년단) 'DNA' Official MV	58961407
2	Yandel – Como Antes (Official Video) ft. Wisin	42486342
3	ZAYN – Dusk Till Dawn ft. Sia	41959549
4	Shakira – Perro Fiel (Official Video) ft. Nicky Jam	38144440
5	Sam Smith – Too Good At Goodbyes (Official Video)	29080061
6	De La Ghetto, Daddy Yankee, Ozuna & Chris Jeday – L...	23699605
7	Lil Uzi Vert – XO Tour Llif3 (Official Music Video)	17521230
8	Sam Smith – Too Good At Goodbyes (Official Audio)	17103567
9	CNCO, Little Mix – Reggaetón Lento (Remix) [Official ...	16529782
10	Primitive Technology: Mud Bricks	15877131

Category ranking in US/GB

For US:

```
> category_us_views1 <- newdf %>% select(title,category_id,views,country) %>% filter(country == "US") %>% group_by(title) %>% summarise(category_id = max(category_id),n = max(views))
> category_us_views <- category_us_views1 %>% group_by(category_id) %>% summarise(n_of_views = sum(n)) %>% arrange(desc(n_of_views))
```

```
> View(category_us_views)
```

Output:

	category_id	n_of_views
1	10	311759608
2	24	200791393
3	23	129400666
4	22	127095223
5	26	79715118
6	28	64537294
7	1	57569892
8	17	45742864
9	25	45289090
10	27	25109098
11	15	16735270
12	2	15127875
13	20	10298349
14	19	5956390
15	29	81060

For GB:

```
> category_gb_views1 <- newdf %>% select(title,category_id,views,country) %>% filter(country == "GB") %>%
group_by(title) %>% summarise(category_id = max(category_id),n = max(views))
> category_gb_views <- category_gb_views1 %>% group_by(category_id) %>% summarise(n_of_views = sum(n)) %>%
arrange(desc(n_of_views))
> View(category_gb_views)
```

Output:

	category_id	n_of_views
1	10	461843901
2	24	150589492
3	22	105794976
4	23	80552696
5	26	68037845
6	1	52472622
7	17	50736644
8	28	29529647
9	25	24481229
10	20	23403106
11	15	14857879
12	27	13094024
13	2	4091704
14	19	564343
15	29	131068

3. Top trending channels/videos/categories based on # likes and # dislikes

Top 10 trending channels in US

to consider both likes and dislikes, we use popularity to measure, which is the difference between the number of likes and dislikes.

```
> channel_us_likes1 <- newdf %>% select(title,channel_title,likes_video,dislikes,country) %>% filter(country == "US") %>% group_by(title,channel_title) %>% summarise(popularity = max(likes_video)-max(dislikes))
> channel_us_likes <- channel_us_likes1 %>% group_by(channel_title) %>% summarise(n_of_popularities = sum(popularity)) %>% arrange(desc(n_of_popularities)) %>% head(10)
> View(channel_us_likes)
```

Output:

	channel_title	n_of_popularities
1	ibighit	2976182
2	ZaynVEVO	1403634
3	SamSmithWorldVEVO	989119
4	melanie martinez	909081
5	Dude Perfect	796365
6	Liza Koshy	749837
7	NiallHoranVEVO	740971
8	IISuperwomanII	692098
9	PewDiePie	648824
10	Linkin Park	618590

Top 10 trending channels in GB

```
> channel_gb_likes1 <- newdf %>% select(title,channel_title,likes_video,dislikes,country) %>% filter(country == "GB") %>% group_by(title,channel_title) %>% summarise(popularity = max(likes_video)-max(dislikes))
> channel_gb_likes <- channel_gb_likes1 %>% group_by(channel_title) %>% summarise(n_of_popularities = sum(popularity)) %>% arrange(desc(n_of_popularities)) %>% head(10)
> View(channel_gb_likes)
```

Output:

	channel_title	n_of_popularities
1	ibighit	2906474
2	ZaynVEVO	1472061
3	SamSmithWorldVEVO	1187825
4	Dude Perfect	809847
5	Liza Koshy	769728
6	Linkin Park	692136
7	littlemixVEVO	665092
8	YandelVEVO	664394
9	The Gabbie Show	635304
10	shakiraVEVO	576788

Top 10 trending videos in US

```
> video_us_likes <-newdf %>% filter(country == "US") %>% group_by(title) %>% summarise(n_of_popularities = max(likes_video)-max(dislikes)) %>% arrange(desc(n_of_popularities)) %>% head(10)
> View(video_us_likes)
```

Output:

	title	n_of_popularities
1	BTS (방탄소년단) 'DNA' Official MV	1932290
2	ZAYN – Dusk Till Dawn ft. Sia	1403634
3	BTS (방탄소년단) 'DNA' Official Teaser 1	1043892
4	Melanie Martinez – Mad Hatter [Official Video]	909081
5	My Response	648824
6	One More Light (Official Video) – Linkin Park	618590
7	Shakira – Perro Fiel (Official Video) ft. Nicky Jam	534791
8	Sam Smith – Too Good At Goodbyes (Official Audio)	462039
9	Ed Sheeran – Perfect [Official Lyric Video]	456566
10	Sam Smith – Too Good At Goodbyes (Official Video)	443269

Top 10 trending videos in GB

```
> video_gb_likes <-newdf %>% filter(country == "GB") %>% group_by(title) %>% summarise(n_of_popularities = max(likes_video)-max(dislikes)) %>% arrange(desc(n_of_popularities)) %>% head(10)
> View(video_gb_likes)
```

Output:

	title	n_of_popularities
1	BTS (방탄소년단) 'DNA' Official MV	2193653
2	ZAYN – Dusk Till Dawn ft. Sia	1472061
3	BTS (방탄소년단) 'DNA' Official Teaser 2	712821
4	One More Light (Official Video) – Linkin Park	692136
5	CNCO, Little Mix – Reggaetón Lento (Remix) [Official …	665092
6	Yandel – Como Antes (Official Video) ft. Wisin	664394
7	Sam Smith – Too Good At Goodbyes (Official Video)	600650
8	Shakira – Perro Fiel (Official Video) ft. Nicky Jam	576788
9	Sam Smith – Too Good At Goodbyes (Official Audio)	503490
10	Out Loud (Lyric Video)	492262

Category Ranking in US/GB

For US:

```
> category_us_likes1 <- newdf %>% select(title,category_id,likes_video,dislikes,country) %>% filter(country == "US") %>% group_by(title) %>% summarise(category_id = max(category_id),popularity = max(likes_video)-max(dislikes))
> category_us_likes <- category_us_likes1 %>% group_by(category_id) %>% summarise(n_of_popularities = sum(popularity)) %>% arrange(desc(n_of_popularities))
> View(category_us_likes)
```

Output:

	category_id	n_of_popularities
1	10	13896172
2	23	5315799
3	24	4705154
4	22	4197728
5	26	4089206
6	28	1522776
7	1	1222464
8	17	1087481
9	27	665506
10	15	368015
11	20	297627
12	25	296214
13	2	273897
14	19	161992
15	29	626

For GB:

```
> category_gb_likes1 <- newdf %>% select(title,category_id,likes_video,dislikes,country) %>% filter(country == "GB") %>% group_by(title) %>% summarise(category_id = max(category_id),popularity = max(likes_video)-max(dislikes))
> category_gb_likes <- category_gb_likes1 %>% group_by(category_id) %>% summarise(n_of_popularities = sum(popularity)) %>% arrange(desc(n_of_popularities))
> View(category_gb_likes)
```

Output:

	category_id	n_of_popularities
1	10	14929756
2	24	3938442
3	26	3521709
4	22	3348123
5	23	3238592
6	17	1201543
7	1	960098
8	20	752255
9	28	500958
10	27	415376
11	15	317865
12	25	171392
13	2	53300
14	19	18886
15	29	266

4. Top 15 trending videos with most trending days

In US:

```
> library(lubridate)
> trending_us <- newdf[newdf$country == "US",]
> trending_us <- trending_us %>% group_by(title) %>% summarise(max_date = max(date),min_date = min(date))
> trending_us$max_date <- as.Date(trending_us$max_date, format='%m-%d')
> trending_us$min_date <- as.Date(trending_us$min_date, format='%m-%d')
> trending_us$trending_days <- difftime(trending_us$max_date,trending_us$min_date,unit = "days")
> top_trending_us <- trending_us %>% arrange(desc(trending_days)) %>% head(15)
> View(top_trending_us)
```

Output:

	title	max_date	min_date	trending_days
1	1 YEAR OF VLOGGING -- HOW LOGAN PAUL CHANGE...	2019-09-19	2019-09-13	6
2	Apple iPhone X first look	2019-09-19	2019-09-13	6
3	Bella Hadid Roughs Up Security Roughing Up Female ...	2019-09-20	2019-09-14	6
4	COOK OFF! (2017 Movie) - Official Trailer	2019-09-26	2019-09-20	6
5	Everything Wrong With The LEGO Batman Movie	2019-09-26	2019-09-20	6
6	Foo Fighters Carpool Karaoke	2019-09-27	2019-09-21	6
7	Hello, world. Meet our baby girl: Alexis Olympia Ohan...	2019-09-20	2019-09-14	6
8	iPhone X (parody)	2019-09-19	2019-09-13	6
9	My Response	2019-09-19	2019-09-13	6
10	Shakira - Perro Fiel (Official Video) ft. Nicky Jam	2019-09-22	2019-09-16	6
11	The Insane Full House Theory That Might Be True	2019-09-22	2019-09-16	6
12	TOMB RAIDER - Official Trailer #1	2019-09-26	2019-09-20	6
13	100% PURE UNCUT HOLO *not drugz* *for nails*	2019-09-22	2019-09-17	5
14	22 Injured In Explosion On London Subway Train At P...	2019-09-21	2019-09-16	5
15	26 Facts about Libraries - mental_floss List Show Ep. ...	2019-09-26	2019-09-21	5

In GB:

```
> trending_gb <- newdf[newdf$country == "GB",]
> trending_gb <- trending_gb %>% group_by(title) %>% summarise(max_date = max(date),min_date = min(date))
> trending_gb$max_date <- as.Date(trending_gb$max_date, format='%m-%d')
> trending_gb$min_date <- as.Date(trending_gb$min_date, format='%m-%d')
> trending_gb$trending_days <- difftime(trending_gb$max_date,trending_gb$min_date,unit = "days")
> top_trending_gb <- trending_gb %>% arrange(desc(trending_days)) %>% head(15)
> View(top_trending_gb)
```

Output:

	title	max_date	min_date	trending_days
1	Avicii - Lonely Together ft. Rita Ora	2019-09-27	2019-09-19	8
2	Bill Skarsgård's Demonic "IT Smile - CONAN on TBS	2019-09-22	2019-09-14	8
3	Burning device filmed on tube carriage at Parsons Gre...	2019-09-24	2019-09-16	8
4	CNCO, Little Mix - Reggaetón Lento (Remix) [Official ...	2019-09-26	2019-09-18	8
5	Fergie - You Already Know ft. Nicki Minaj	2019-09-21	2019-09-13	8
6	Gucci Mane - Curve feat The Weeknd [Official Audio]	2019-09-22	2019-09-14	8
7	Hello, world. Meet our baby girl: Alexis Olympia Ohan...	2019-09-22	2019-09-14	8
8	iPhone X (parody)	2019-09-22	2019-09-14	8
9	Lana Del Rey - White Mustang (Official Video)	2019-09-22	2019-09-14	8
10	Official US Find Your Voice Trailer - Disney/Pixar's Coco	2019-09-22	2019-09-14	8
11	PUBG AIRSOFT - REAL LIFE BATTLEGROUNDS!	2019-09-22	2019-09-14	8
12	Rick and Morty: Has Rick Changed?	2019-09-25	2019-09-17	8
13	Sam Smith - Too Good At Goodbyes (Official Video)	2019-09-27	2019-09-19	8
14	SOY PABLO Extended Trailer -- A Bad Lip Reading of ...	2019-09-22	2019-09-14	8
15	SPIKED by a Sea Urchin?	2019-09-22	2019-09-14	8

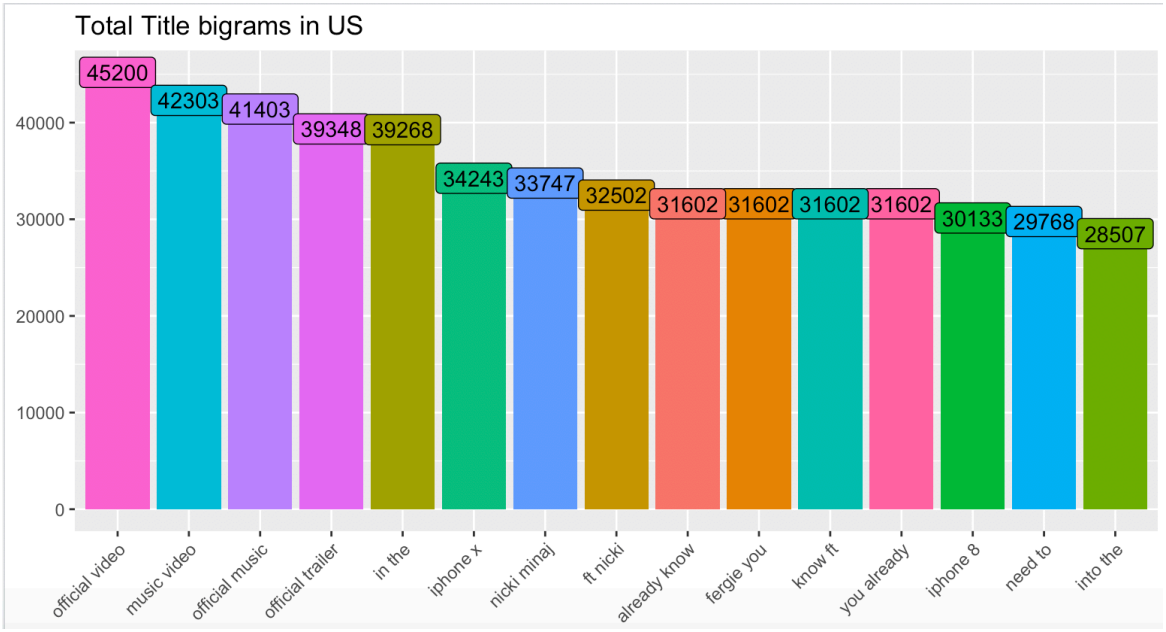
Part 3: Text data analysis

1. Most Common words from videos/tags - Bigram

Bigram for video titles in US

```
> worddata_us <- all[all$country == "US",]
> library(tidytext)
> titleBigram_us <- unnest_tokens(worddata_us, bigram, title, token = "ngrams", n = 2)
> library(data.table)
> titleBigram_us <- as.data.table(titleBigram_us)
> p_titleBigram_us <- ggplot(titleBigram_us[,.N,by=bigram][order(-N)][1:15],aes(reorder(bigram,-N),N,fill=bigram))+geom_bar(stat="identity")+ geom_label(aes(label=N))+guides(fill="none")+ theme(axis.text.x = element_text(angle = 45,hjust = 1))+  labs(title="Total Title bigrams in US")+  xlab(NULL)+ylab(NULL)
> p_titleBigram_us
```

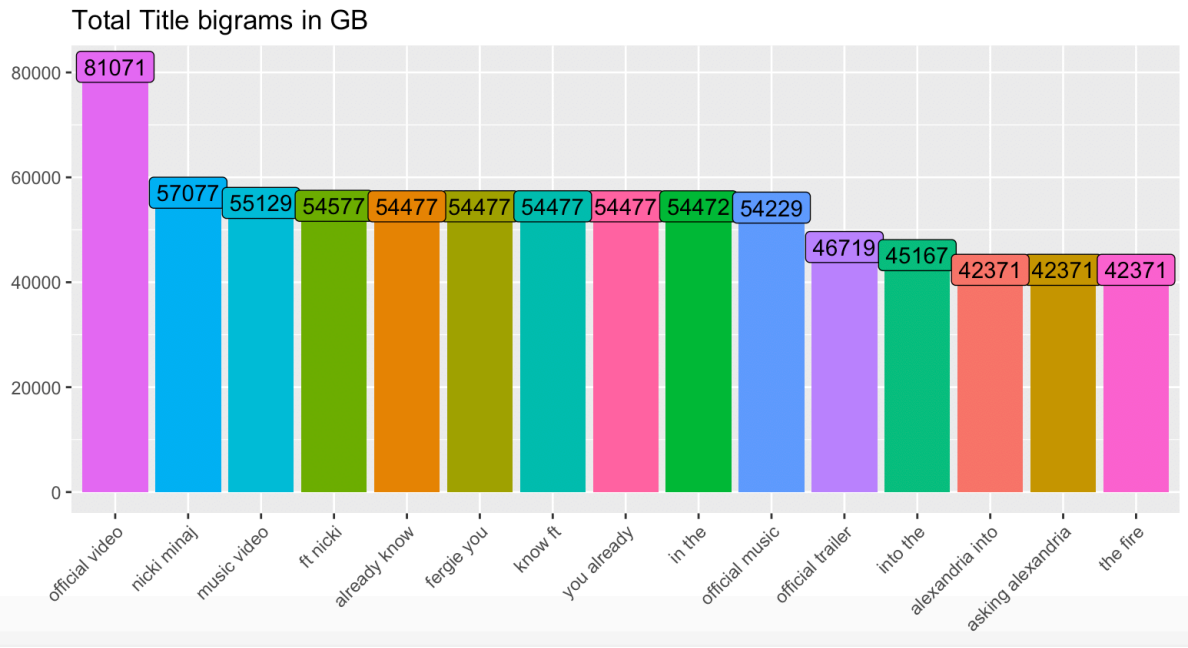
Output:



Bigram for video titles in GB

```
> worddata_gb <- all[all$country == "GB",]
> titleBigram_gb <- unnest_tokens(worddata_gb, bigram, title, token = "ngrams", n = 2)
> titleBigram_gb <- as.data.table(titleBigram_gb)
> p_titleBigram_gb <- ggplot(titleBigram_gb[,.N,by=bigram][order(-N)][1:15],aes(reorder(bigram,-N),N,fill=bigram))+geom_bar(stat="identity")+ geom_label(aes(label=N))+guides(fill="none")+ theme(axis.text.x = element_text(angle = 45,hjust = 1))+  labs(title="Total Title bigrams in GB")+  xlab(NULL)+ylab(NULL)
> p_titleBigram_gb
```

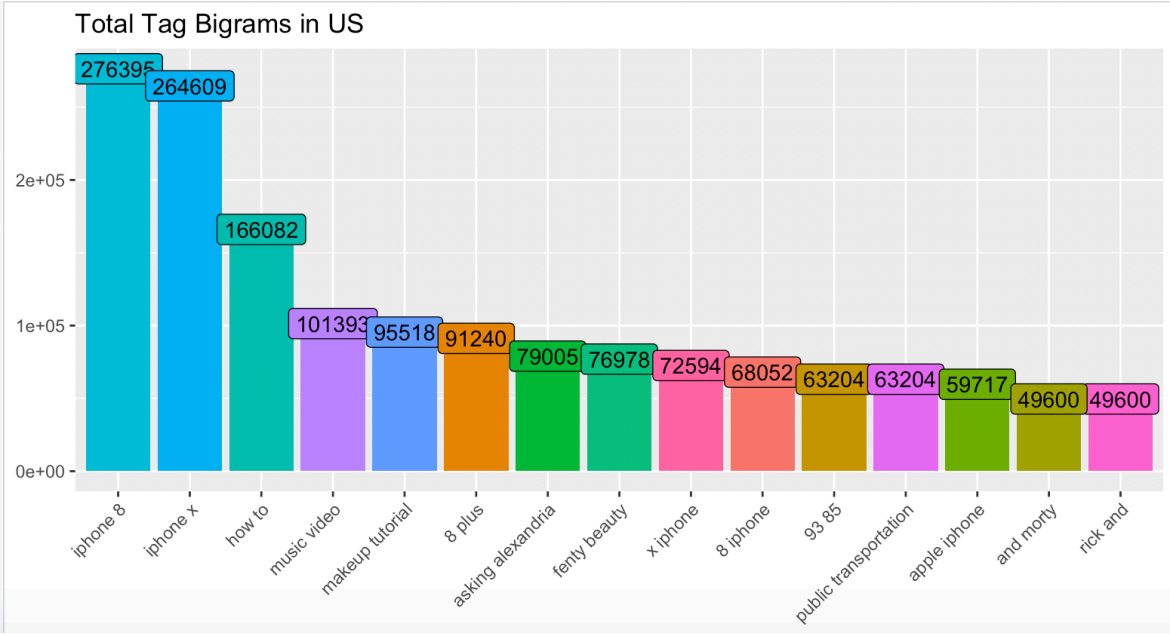
Output:



Bigram for tags in US

```
> tagBigram_us <- unnest_tokens(worddata_us, bigram, tags, token = "ngrams", n = 2)
> tagBigram_us <- as.data.table(tagBigram_us)
> p_tagBigram_us <- ggplot(tagBigram_us[,.N,by=bigram][order(-N)][1:15],aes(reorder(bigram,-
N),N,fill=bigram))+geom_bar(stat="identity")+ geom_label(aes(label=N))+guides(fill="none")+ theme(axis.text.x =
element_text(angle = 45,hjust = 1))+  labs(title="Total Tag Bigrams in US")+  xlab(NULL)+ylab(NULL)
> p_tagBigram_us
```

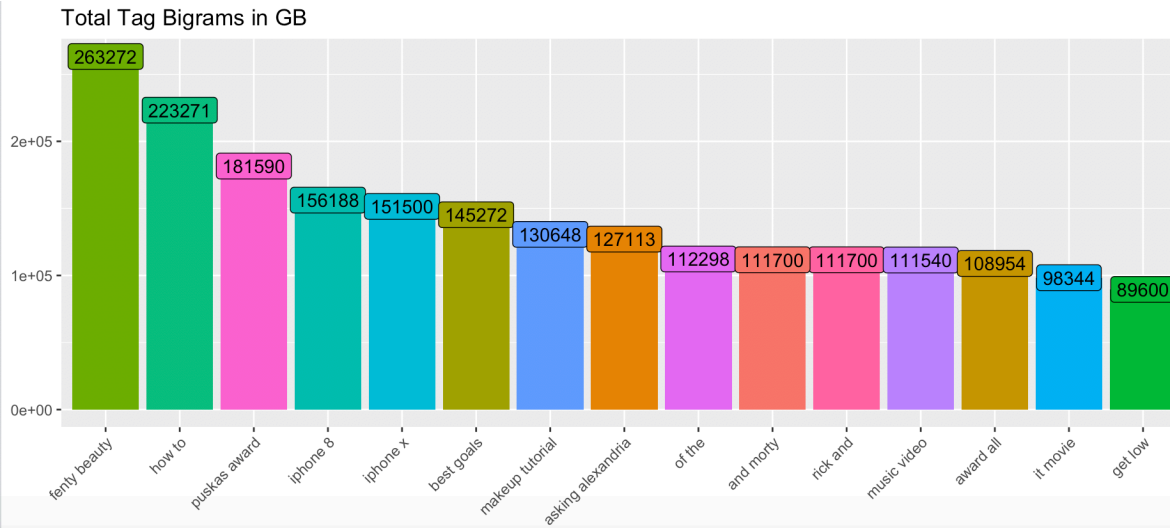
Output:



Bigram for tags in GB

```
> tagBigram_gb <- unnest_tokens(worddata_gb, bigram, tags, token = "ngrams", n = 2)
> tagBigram_gb <- as.data.table(tagBigram_gb)
> p_tagBigram_gb <- ggplot(tagBigram_gb[,.N,by=bigram][order(-N)][1:15],aes(reorder(bigram,-N),N,fill=bigram))+geom_bar(stat="identity")+ geom_label(aes(label=N))+guides(fill="none")+ theme(axis.text.x = element_text(angle = 45,hjust = 1))+  labs(title="Total Tag Bigrams in GB")+  xlab(NULL)+ylab(NULL)
> p_tagBigram_gb
```

Output:



2. Most Common words from videos/tags - Wordcloud

Wordcloud for video titles in US

```
> library(corpus)
> library(tm)
> library(wordcloud)
> worddata_us <- all[all$country == "US",]
> titles_us <- c(worddata_us['title'])
> corpus_title_us <-Corpus(VectorSource(list(sample(worddata_us$title,size=2000))))
> corpus_title_us_new <-tm_map(corpus_title_us,tolower)
> corpus_title_us_new <-tm_map(corpus_title_us_new,removeWords,stopwords("english"))
> corpus_title_us_new <-tm_map(corpus_title_us_new,removePunctuation)
> corpus_title_us_new <-tm_map(corpus_title_us_new,stripWhitespace)
> corpus_title_us_new <-tm_map(corpus_title_us_new,removeNumbers)
> tdm_title_us = TermDocumentMatrix(corpus_title_us_new)
> tdm_title_us = as.matrix(tdm_title_us)
> v_us <- sort(rowSums(tdm_title_us),decreasing=TRUE)
> d_us <- data.frame(word = names(v_us),freq=v_us)
> set.seed(1234)
> wordcloud(words = d_us$word, freq = d_us$freq, min.freq = 10,max.words=200, random.order=FALSE, rot.per=0.2,
colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud for video titles in GB

```
> worddata_gb <- all[all$country == "GB",]
> titles_gb <- c(worddata_gb['title'])
> corpus_title_gb <-Corpus(VectorSource(list(sample(worddata_gb$title,size=2000))))
> corpus_title_gb_new <-tm_map(corpus_title_gb,tolower)
> corpus_title_gb_new <-tm_map(corpus_title_gb_new,removeWords,stopwords("english"))
> corpus_title_gb_new <-tm_map(corpus_title_gb_new,removePunctuation)
> corpus_title_gb_new <-tm_map(corpus_title_gb_new,stripWhitespace)
> corpus_title_gb_new <-tm_map(corpus_title_gb_new,removeNumbers)
> tdm_title_gb = TermDocumentMatrix(corpus_title_gb_new)
> tdm_title_gb = as.matrix(tdm_title_gb)
> v_gb <- sort(rowSums(tdm_title_gb),decreasing=TRUE)
> d_gb <- data.frame(word = names(v_gb),freq=v_gb)
> set.seed(1234)
> wordcloud(words = d_gb$word, freq = d_gb$freq, min.freq = 10,max.words=200, random.order=FALSE, rot.per=0.2,
colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud for tags in US

```
> tags_us <- c(worddata_us['tags'])
> corpus_tag_us <-Corpus(VectorSource(list(sample(worddata_us$tags,size=2000))))
> corpus_tag_us_new <-tm_map(corpus_tag_us,tolower)
> corpus_tag_us_new <-tm_map(corpus_tag_us_new,removeWords,stopwords("english"))
> corpus_tag_us_new <-tm_map(corpus_tag_us_new,removePunctuation)
> corpus_tag_us_new <-tm_map(corpus_tag_us_new,stripWhitespace)
> vcorpus_tag_us_new <-tm_map(corpus_tag_us_new,removeNumbers)
> tdm_tag_us = TermDocumentMatrix(corpus_tag_us_new)
> tdm_tag_us = as.matrix(tdm_tag_us)
> vv_us <- sort(rowSums(tdm_tag_us),decreasing=TRUE)
> dd_us <- data.frame(word = names(vv_us),freq=vv_us)
> set.seed(1234)
> wordcloud(words = dd_us$word, freq = dd_us$freq, min.freq = 10,max.words=200,  random.order=FALSE, rot.per=0.2,
colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud for tags in GB

```
> tags_gb <- c(worddata_gb['tags'])
> corpus_tag_gb <-Corpus(VectorSource(list(sample(worddata_gb$tags,size=2000))))
> corpus_tag_gb_new <-tm_map(corpus_tag_gb,tolower)
> corpus_tag_gb_new <-tm_map(corpus_tag_gb_new,removeWords,stopwords("english"))
> corpus_tag_gb_new <-tm_map(corpus_tag_gb_new,removePunctuation)
> corpus_tag_gb_new <-tm_map(corpus_tag_gb_new,stripWhitespace)
> corpus_tag_gb_new <-tm_map(corpus_tag_gb_new,removeNumbers)
> tdm_tag_gb = TermDocumentMatrix(corpus_tag_gb_new)
> tdm_tag_gb = as.matrix(tdm_tag_gb)
> vv_gb <- sort(rowSums(tdm_tag_gb),decreasing=TRUE)
> dd_gb <- data.frame(word = names(vv_gb),freq=vv_gb)
> set.seed(1234)
> wordcloud(words = dd_gb$word, freq = dd_gb$freq, min.freq = 10,max.words=200, random.order=FALSE, rot.per=0.2,
colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud for comments in US

```
> com_us <- c(worddata_us['comment_text'])
> corpus_com_us <-Corpus(VectorSource(list(sample(worddata_us$comment_text,size=2000))))
> corpus_com_us_new <-tm_map(corpus_com_us,tolower)
> corpus_com_us_new <-tm_map(corpus_com_us_new,removeWords,stopwords("english"))
> corpus_com_us_new <-tm_map(corpus_com_us_new,removePunctuation)
> corpus_com_us_new <-tm_map(corpus_com_us_new,stripWhitespace)
> corpus_com_us_new <-tm_map(corpus_com_us_new,removeNumbers)
> tdm_com_us = TermDocumentMatrix(corpus_com_us_new)
> tdm_com_us = as.matrix(tdm_com_us)
> vvv_us <- sort(rowSums(tdm_com_us),decreasing=TRUE)
> ddd_us <- data.frame(word = names(vvv_us),freq=vvv_us)
> set.seed(1234)
> wordcloud(words = ddd_us$word, freq = ddd_us$freq, min.freq = 10,max.words=200, random.order=FALSE,
rot.per=0.2, colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud for comments in GB

```
> com_gb <- c(worddata_gb['comment_text'])
> corpus_com_gb <-Corpus(VectorSource(list(sample(worddata_gb$comment_text,size=2000))))
> corpus_com_gb_new <-tm_map(corpus_com_gb,tolower)
> corpus_com_gb_new <-tm_map(corpus_com_gb_new,removeWords,stopwords("english"))
> corpus_com_gb_new <-tm_map(corpus_com_gb_new,removePunctuation)
> corpus_com_gb_new <-tm_map(corpus_com_gb_new,stripWhitespace)
> corpus_com_gb_new <-tm_map(corpus_com_gb_new,removeNumbers)
> tdm_com_gb = TermDocumentMatrix(corpus_com_gb_new)
> tdm_com_gb = as.matrix(tdm_com_gb)
> vvv_gb <- sort(rowSums(tdm_com_gb),decreasing=TRUE)
> ddd_gb <- data.frame(word = names(vvv_gb),freq=vvv_gb)
> set.seed(1234)
> wordcloud(words = ddd_gb$word, freq = ddd_gb$freq, min.freq = 10,max.words=200, random.order=FALSE,
rot.per=0.2, colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud – tags for top 10 trending videos in US

```
> vec = unlist(video_us_likes[,1])
> top_tags <- unique(all[all$title %in% vec,c(1:11,15)])
> top_tags_us <- top_tags[top_tags$country == "US",]
> toptag_us <- c(top_tags_us['tags'])
> corpus_toptag_us <-Corpus(VectorSource(toptag_us$tags))
> corpus_toptag_us_new <-tm_map(corpus_toptag_us,tolower)
> corpus_toptag_us_new <- tm_map(corpus_toptag_us_new,removeWords,stopwords("english"))
> corpus_toptag_us_new <-tm_map(corpus_toptag_us_new,removePunctuation)
> corpus_toptag_us_new <-tm_map(corpus_toptag_us_new,stripWhitespace)
> corpus_toptag_us_new <-tm_map(corpus_toptag_us_new,removeNumbers)
> tdm_toptag_us = TermDocumentMatrix(corpus_toptag_us_new)
> tdm_toptag_us = as.matrix(tdm_toptag_us)
> vvvv_us <- sort(rowSums(tdm_toptag_us),decreasing=TRUE)
> dddd_us <- data.frame(word = names(vvvv_us),freq=vvvv_us)
> set.seed(1234)
```

```
> wordcloud(words = dddd_us$word, freq = dddd_us$freq, min.freq = 1,max.words=200, random.order=FALSE,
rot.per=0.2, colors=brewer.pal(8, "Dark2"))
```

Output:

Wordcloud – tags for top 10 trending videos in GB

```
> toptags_gb <-top_tags[top_tags$country == "GB",]
> toptags_gb <- c(top_tags_gb['tags'])
> corpus_toptag_gb <-Corpus(VectorSource(toptags_gb$tags))
> corpus_toptag_gb_new <-tm_map(corpus_toptag_gb,tolower)
> corpus_toptag_gb_new <-tm_map(corpus_toptag_gb_new,removeWords,stopwords("english"))
> corpus_toptag_gb_new <-tm_map(corpus_toptag_gb_new,removePunctuation)
> corpus_toptag_gb_new <-tm_map(corpus_toptag_gb_new,stripWhitespace)
> corpus_toptag_gb_new <-tm_map(corpus_toptag_gb_new,removeNumbers)
> tdm_toptag_gb = TermDocumentMatrix(corpus_toptag_gb_new)
> tdm_toptag_gb = as.matrix(tdm_toptag_gb)
> vvvv_gb <- sort(rowSums(tdm_toptag_gb),decreasing=TRUE)
> dddd_gb <- data.frame(word = names(vvvv_gb),freq=vvvv_gb)
> set.seed(1234)
> wordcloud(words = dddd_gb$word, freq = dddd_gb$freq, min.freq = 1,max.words=200, random.order=FALSE,
rot.per=0.2, colors=brewer.pal(8, "Dark2"))
```

Output:

Part 4: Clustering

add three new variables, which are percentage of likes, dislikes, and comments on total views.

```
> newdf$Percentage_Likes <- round(100*(newdf$likes_video)/sum(as.numeric(newdf$views),na.rm = T),digits = 4)
> newdf$Percentage_Disikes <- round(100*(newdf$dislikes)/sum(as.numeric(newdf$views),na.rm = T),digits = 4)
> newdf$Percentage_Comments <- round(100*(newdf$comment_total)/sum(as.numeric(newdf$views),na.rm = T),digits = 4)
```

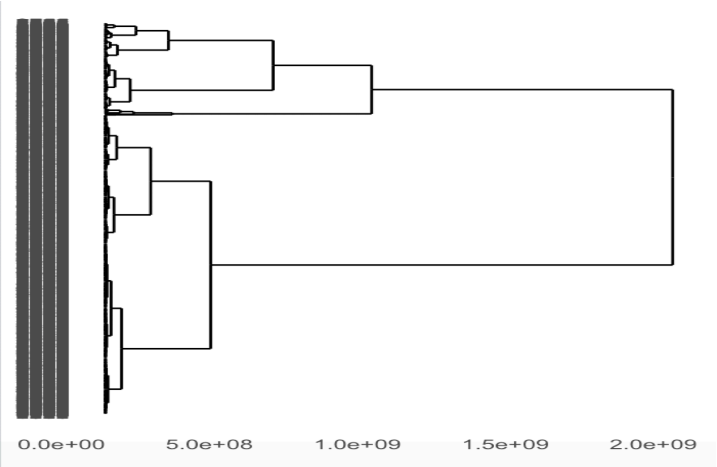
take a sample of the total dataset to do the clustering as the dataset is too large

```
> newdfsample <- newdf[sample(1:nrow(newdf),3000, replace=FALSE),]
```

plot a dendrogram to see the arrangement of the clusters

```
> sam <- dist(x = newdfsample)
> cluster <- hclust(sam,method = "ward.D")
> ggdendrogram(cluster,rotate = T)
```

Output:



as we can see from the dendrogram, there are 3 main clusters.

```
> cluster_cutree <- cutree(cluster,3)
> newdfsample$Cluster <- cluster_cutree
> View(newdfsample)
> cluster1 <- apply(newdfsample[newdfsample$Cluster==1,c(4:7,11:13)], 2, function(x) mean(x,na.rm=T))
> cluster2 <- apply(newdfsample[newdfsample$Cluster==2,c(4:7,11:13)], 2, function(x) mean(x,na.rm=T))
> cluster3 <- apply(newdfsample[newdfsample$Cluster==3,c(4:7,11:13)], 2, function(x) mean(x,na.rm=T))
> cluster_main <- as.matrix(rbind(cluster1,cluster2,cluster3))
> library(knitr)
> knitr::kable(t(cluster_main),digits=10)
```

Output:

	cluster1	cluster2	cluster3
views	2.768153e+05	2.565644e+06	1.669654e+07
likes_video	1.029528e+04	9.698118e+04	5.550225e+05
dislikes	3.467991e+02	5.278986e+03	2.579613e+04
comment_total	1.259571e+03	1.112207e+04	9.606436e+04
Percentage_Likes	1.292375e-04	1.256212e-03	7.184444e-03
Percentage_Disikes	6.972000e-07	5.727270e-05	3.333333e-04
Percentage_Comments	7.189500e-06	1.428788e-04	1.253333e-03