Stats Final Project

Aparna Devi Akula

R Markdown

Doing analysis on YouTube data seems very interesting to me, so I decide to consider YouTube data. This dataset is a daily record of the top trending YouTube videos. This dataset includes several months of data on daily trending YouTube videos. Data is included for the USA, Great Britain, Germany, Canada, and France with up to 200 listed trending videos per day. But for my analysis I considered only USA dataset. This data is in a separate file. Data includes the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count. So with the data I am doing visualizations that tells us how views, likes,dislikes,comment_count,trending date and some other variables are related to each other. I also developed two models to find out which model is efficient for this particular dataset and what is the accuracy of the predicted models.

Declaring the required packages

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse
1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.3     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------
tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(lubridate)
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
library(dplyr)
library(tinytex)
```

Importing data

```
youtube = read_csv("C:/Users/SIRIDEVA/Documents/UCB/STAT 5000/USvideos.csv")

## Rows: 6440 Columns: 15

## -- Column specification --------------------------------------------------
------
## Delimiter: ","
## chr  (6): video_id, trending_date, title, channel_title, tags, description
## dbl  (8): category_id, views, likes, dislikes, comment_count,
comments_disab...
## dttm (1): publish_time

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this
message.

glimpse(youtube)

## Rows: 6,440
## Columns: 15
## $ video_id               <chr> "7C2z4GqqS5E", "VYOjWnS4cMY",
"ffxKSjUwKdU", "F~
## $ trending_date          <chr> "18.01.06", "18.02.06", "18.14.05",
"17.14.12",~
## $ title                  <chr> "BTS
(<U+BC29><U+D0C4><U+C18C><U+B144><U+B2E8>) 'FAKE LOVE' Official MV", "Ch~
## $ channel_title          <chr> "ibighit", "ChildishGambinoVEVO",
"ArianaGrande~
## $ category_id            <dbl> 10, 10, 10, 24, 10, 10, 10, 24, 10, 10, 10,
22,~
## $ publish_time           <dttm> 2018-05-18 09:00:02, 2018-05-06 04:00:07,
2018~
## $ tags                   <chr>
"BIGHIT|\"<U+BE45><U+D788><U+D2B8>\"|\"<U+BC29><U+D0C4><U+C18C><U+B144><U+B2E
8>\"|\"BTS\"|\"BAN~
## $ views                  <dbl> 123010920, 225211923, 148689896, 149376127,
368~
## $ likes                  <dbl> 5613827, 5023450, 3094021, 3093544,
2729292, 27~
## $ dislikes               <dbl> 206892, 343541, 129502, 1643059, 47896,
29341, ~
## $ comment_count          <dbl> 1228655, 517232, 242039, 810698, 546100,
371864~
## $ comments_disabled      <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,~
## $ ratings_disabled       <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,~
## $ video_error_or_removed <dbl> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0,~
```

```
## $ description            <chr> "BTS
(<U+BC29><U+D0C4><U+C18C><U+B144><U+B2E8>) 'FAKE LOVE' Official MVDirect~
```

The dataset obtained is clear and there is no need to clean the data, so considering the original dataset itself.

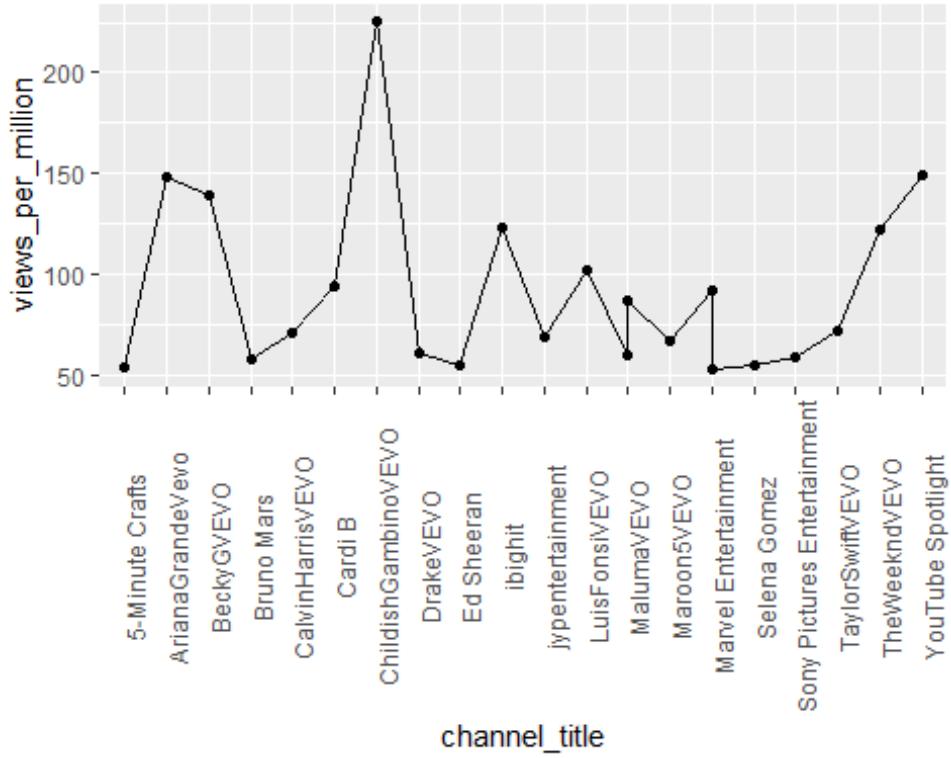
Rearranging data as per convenience

```
youtube <- youtube %>% mutate(views_per_million = views/1000000)
youtube <- youtube %>% mutate(likes_per_million = likes/1000000)
youtube <- youtube %>% mutate(dislikes_per_million = dislikes/1000000)
```

Converting view,likes and dislikes to views_per_million,likes_per_million and dislikes_per_million respectively as per convenience.

How many channels have the videos greater 50 million views?

```
youtube %>% filter(views_per_million>50) %>%
  ggplot(aes(x=channel_title,y=views_per_million)) +
geom_point(aes(x=channel_title,y=views_per_million)) +
geom_line(aes(group=1))+ theme(legend.position = "bottom", axis.text.x =
element_text(angle = 90))
```



20 channels have the videos which have more than 50 million views. Two channels each have two videos that have more than 50 million views.

Which video has the highest and lowest likes?

```
youtube %>% filter(likes==0 | likes==5613827 )
```

```
## # A tibble: 31 x 18
##    video_id    trending_date title channel_title category_id publish_time
##    <chr>       <chr>         <chr> <chr>                <dbl> <dttm>
##  1 7C2z4GqqS5E 18.01.06      BTS ~ ibighit                 10 2018-05-18
09:00:02
##  2 wRGldR_SQAA 17.14.11      Appl~ Steve Kovach            22 2017-11-09
18:01:04
##  3 Kn5UgGQukYQ 17.21.11      Brea~ hudsonunions~            1 2016-10-14
21:14:51
##  4 A_mlvG_nRsg 17.21.11      Kell~ Rob Andretti            17 2017-10-28
11:15:14
##  5 tj3EKWwIulQ 17.26.11      Sara~ HARRY                   24 2017-11-22
19:00:16
##  6 nx1R-eHSkfM 17.30.11      The ~ Snapchat                10 2017-11-29
14:00:03
##  7 aLSG3178eD4 17.30.11      Tell~ Define Ameri~           29 2017-11-28
13:00:35
##  8 l5S5DM564O8 17.05.12      Ariz~ Pac-12 Netwo~           17 2017-12-04
18:32:36
##  9 PRHVfIbeGpQ 17.14.12      How ~ American Bri~           25 2017-12-11
14:54:32
## 10 kjkw7rKhPfc 18.12.01      PK I~ PK Inventor             28 2017-02-10
22:20:03
## # ... with 21 more rows, and 12 more variables: tags <chr>, views <dbl>,
## #   likes <dbl>, dislikes <dbl>, comment_count <dbl>, comments_disabled
<dbl>,
## #   ratings_disabled <dbl>, video_error_or_removed <dbl>, description
<chr>,
## #   views_per_million <dbl>, likes_per_million <dbl>,
## #   dislikes_per_million <dbl>
```

“BTS (ë°©íƒ„ì†Œë…„ë‹¨) ‘FAKE LOVE’ Official MV” has the highest likes with 5.61 million and there are 30 videos have zero likes.

Which video has the highest and lowest dislikes?

```
youtube %>% filter(dislikes==0 | dislikes==1674420 )
```

```
## # A tibble: 101 x 18
##    video_id    trending_date title channel_title category_id publish_time
##    <chr>       <chr>         <chr> <chr>                <dbl> <dttm>
##  1 QwZT7T-TXT0 18.09.01      So S~ Logan Paul V~           24 2018-01-02
16:42:21
##  2 Q6Usd3_fbq8 17.14.11      Impr~ Nahre Sol               10 2017-11-10
04:08:31
##  3 2XLw9U7Z1CM 17.24.12      Kate~ LIVEKellyand~           24 2017-12-21
17:25:38
##  4 aHjS9YBXzXU 18.05.02      Magi~ Joshua Levin            28 2013-10-20
14:23:54
##  5 TTio2AvTMKk 18.01.02      Insi~ YBF Chic                24 2018-01-26
20:46:07
```

```
##  6 YqH4eWR7jDQ 17.14.11      The ~ Stewart Brand          27 2012-06-10
19:24:38
##  7 1x77e4XvqZ4 18.09.01      RC J~ Mike H                 22 2015-01-08
21:02:08
##  8 yirvgC-kMq0 17.29.12      Hugh~ PeopleTV               24 2017-12-21
14:41:56
##  9 c_KdAO6MqiM 18.12.01      TimR~ GrDrtube               27 2009-09-29
16:27:28
## 10 RxzZ_OXQApM 17.03.12      Caro~ Maggie Smith~          10 2017-11-28
20:10:12
## # ... with 91 more rows, and 12 more variables: tags <chr>, views <dbl>,
## #   likes <dbl>, dislikes <dbl>, comment_count <dbl>, comments_disabled
<dbl>,
## #   ratings_disabled <dbl>, video_error_or_removed <dbl>, description
<chr>,
## #   views_per_million <dbl>, likes_per_million <dbl>,
## #   dislikes_per_million <dbl>
```

"So Sorry" video has highest number of dislikes with 1.6 million, and there are 100 videos which have zero dislikes

The videos which have zero likes and zero dislikes.

```
youtube %>% filter(likes==0 & dislikes==0) %>% select(title)
```

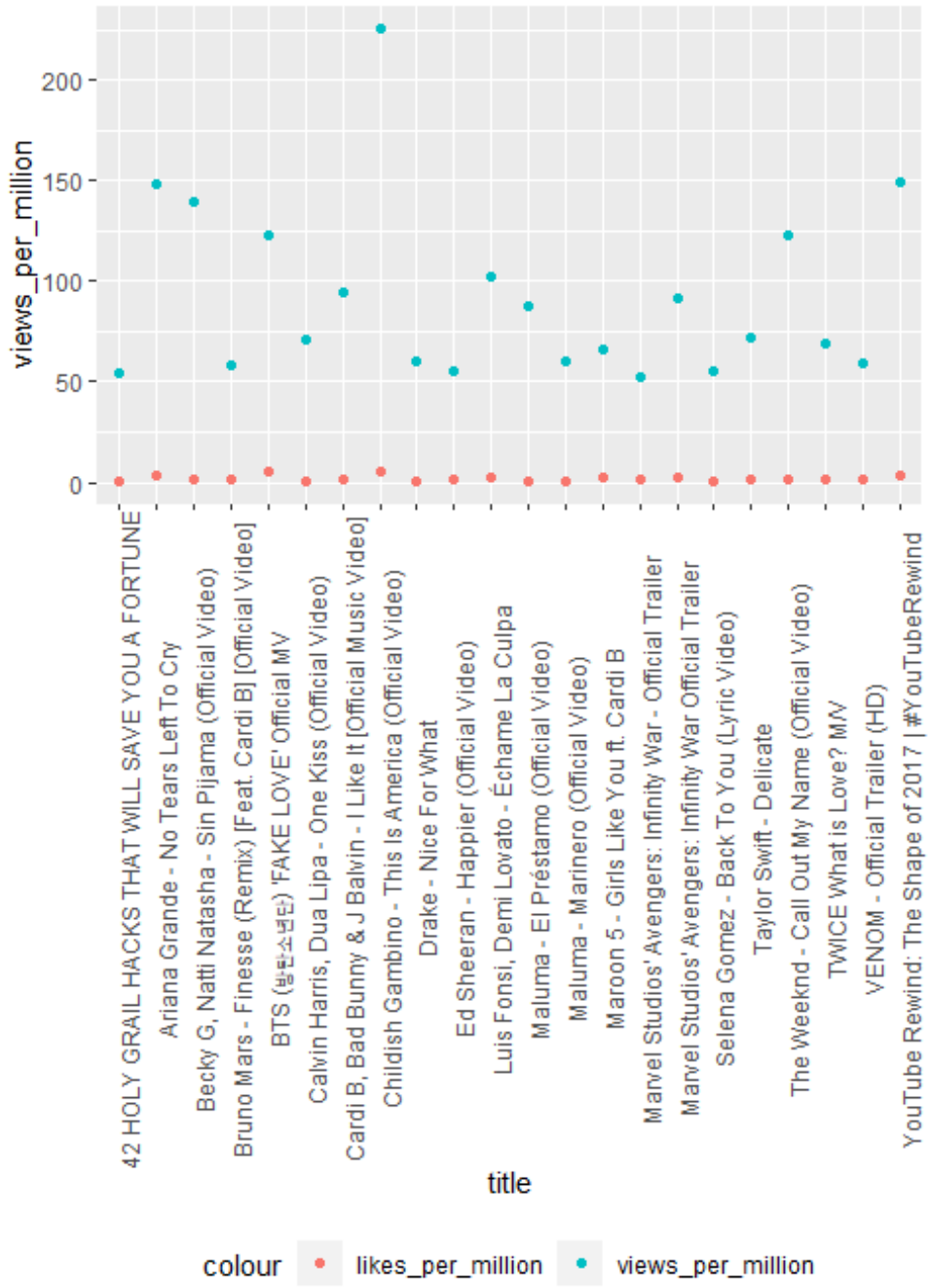
```
## # A tibble: 29 x 1
##    title
##    <chr>
##  1 Apple Clips sample
##  2 Breaking Bad's Bryan Cranston on Meeting Charles Manson
##  3 Sarah Michelle Gellar Vs. Sarah Michelle Prinze
##  4 The New Snapchat in 60 Seconds
##  5 Tell Hollywood to stand with immigrants
##  6 Arizona State introductory press conference of Herm Edwards
##  7 How to Write-In for the Alabama Special Election, Roll Tide!
##  8 PK Inventor ASM V1 0
##  9 RAPID EYE MOVEMENT<U+23AA>Teaser Trailer
## 10 RAPID EYE MOVEMENT<U+23AA>Official Teaser Trailer
## # ... with 19 more rows
```

There are 29 videos which have no likes and no dislikes.

To the videos which have more than 50 million views,how are their corresponding views and likes?

```
youtube_above50 <- youtube %>% filter(views_per_million>50)
youtube_above50 %>% ggplot() +
      geom_point(aes(x=title,y=views_per_million,color =
"views_per_million")) +
      geom_point(aes(x=title,y=likes_per_million,color =
"likes_per_million")) +
```

```
      theme(legend.position = "bottom", axis.text.x = element_text(angle =
90))
```



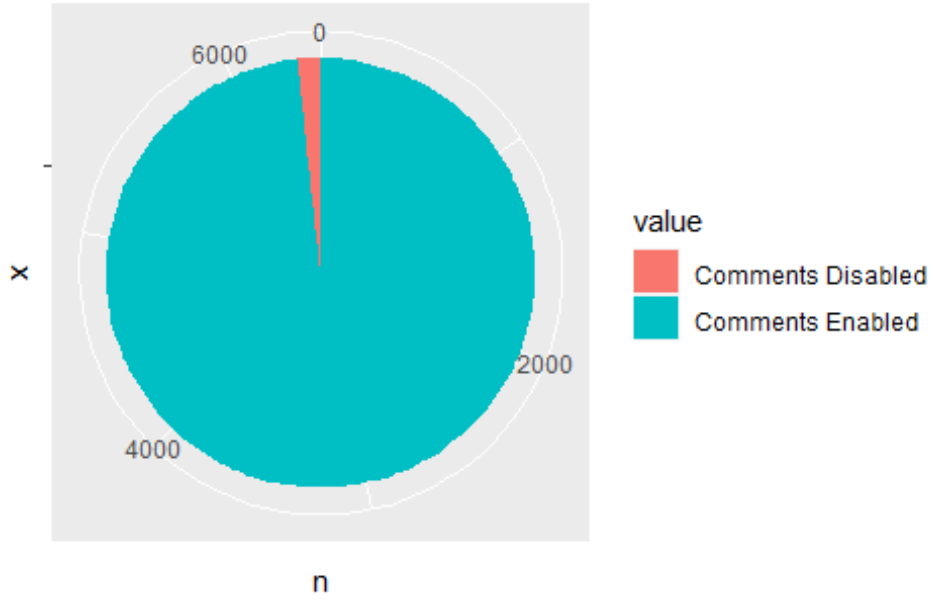
The video “Childish Gambino - This Is America (Official Video)” has 225 million views but it has 5 million likes only, where as “BTS (방탄소년단) ‘FAKE LOVE’ Official MV” video has 5.6 million likes which is highest likes but the views are 123 million.

```
youtube_above50 <- youtube %>% filter(views_per_million>50)
```

Filtering and creating a list with the views greater for 50 million

How many videos have disabled commenting?

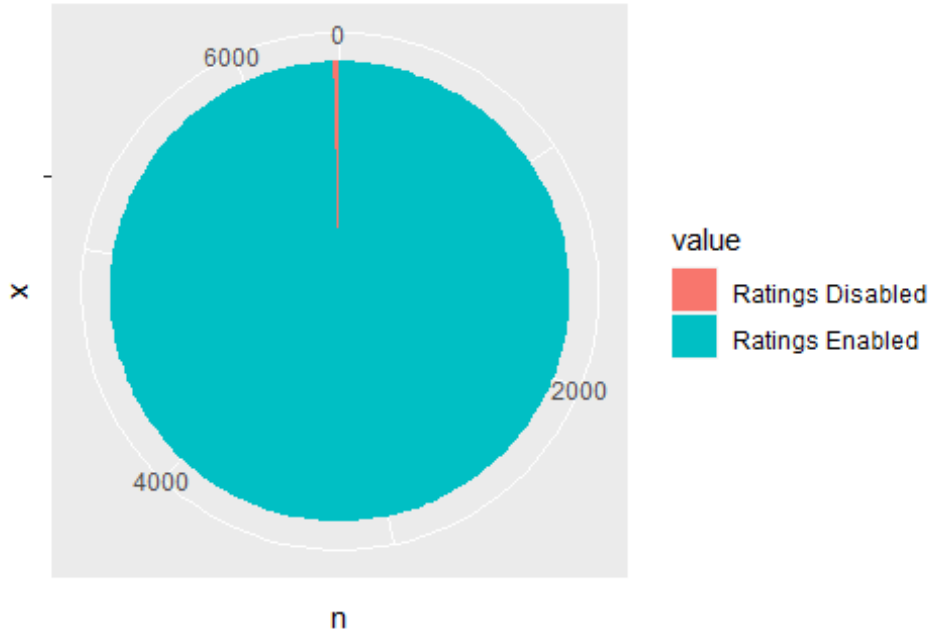
```
youtube_comments <- as_tibble(c("Comments Enabled","Comments Disabled"))
youtube_comments <- youtube_comments %>% add_column(n = c(6329,111))
ggplot(youtube_comments, aes(x="", y = n,fill=value)) +
  geom_bar(stat="identity", width=1) +coord_fixed() +
  coord_polar("y",start = 0)

## Coordinate system already present. Adding new coordinate system, which
will replace the existing one.
```



Nearly 111 videos have disabled the comments out of 6440 videos

How many videos have disabled ratings?

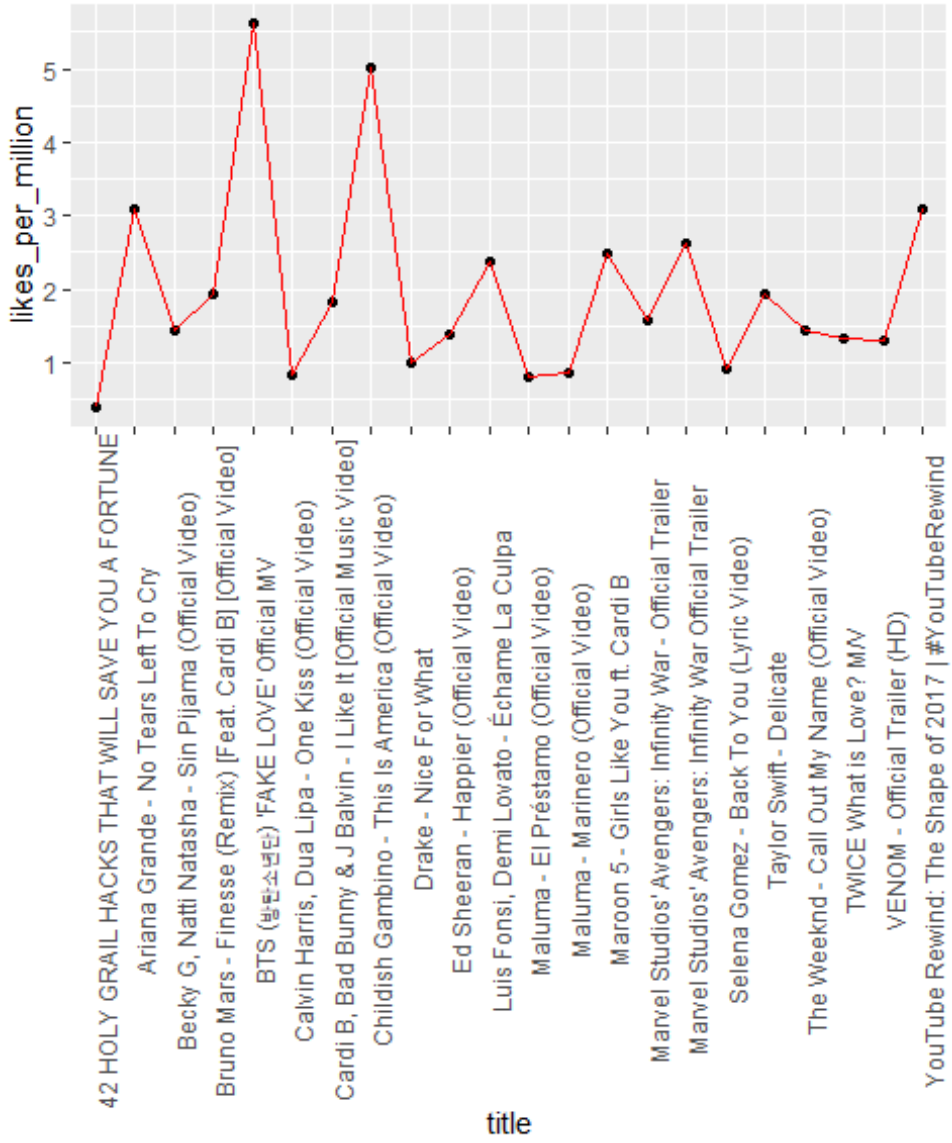
```
youtube_ratings <- as_tibble(c("Ratings Enabled","Ratings Disabled"))
youtube_ratings <- youtube_ratings %>% add_column(n = c(6410,28))
ggplot(youtube_ratings, aes(x="", y = n,fill=value)) +
  geom_bar(stat="identity", width=1) +coord_fixed() +
  coord_polar("y",start = 0)

## Coordinate system already present. Adding new coordinate system, which
will replace the existing one.
```



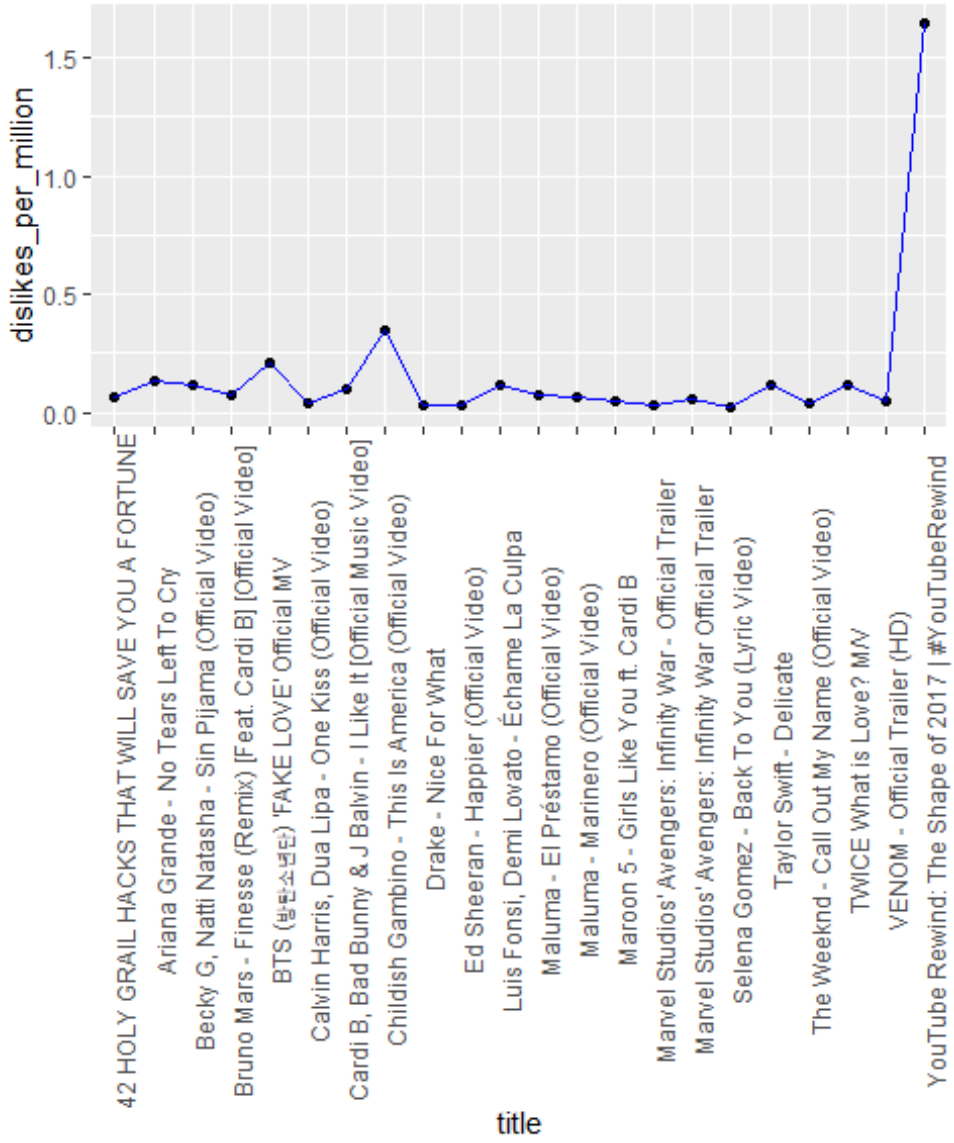
About 28 videos ratings have been disabled and for 6410 videos ratings are enabled

Comparing likes and dislikes for the videos which have more than 50 million views.

```
p1 <- youtube_above50 %>% ggplot(aes(x=title,y=likes_per_million)) +
  geom_point(aes(x=title,y=likes_per_million))+
  geom_line(aes(group=1),color = "red")+
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90))

p2 <- youtube_above50 %>% ggplot(aes(x=title,y=dislikes_per_million))+
  geom_point(aes(x=title,y=dislikes_per_million)) +
  geom_line(aes(group=1),color = "blue") +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90))

plot(p1)
```


likes_per_million
5
4
3
2
1
42 HOLY GRAIL HACKS THAT WILL SAVE YOU A FORTUNE
Ariana Grande - No Tears Left To Cry
Becky G, Natti Natasha - Sin Pijama (Official Video)
Bruno Mars - Finesse (Remix) [Feat. Cardi B] [Official Video]
BTS (방탄소년단) 'FAKE LOVE' Official MV
Calvin Harris, Dua Lipa - One Kiss (Official Video)
Cardi B, Bad Bunny & J Balvin - I Like It [Official Music Video]
Childish Gambino - This Is America (Official Video)
Drake - Nice For What
Ed Sheeran - Happier (Official Video)
Luis Fonsi, Demi Lovato - Échame La Culpa
Maluma - El Préstamo (Official Video)
Maluma - Marinero (Official Video)
Maroon 5 - Girls Like You ft. Cardi B
Marvel Studios' Avengers: Infinity War - Official Trailer
Marvel Studios' Avengers: Infinity War Official Trailer
Selena Gomez - Back To You (Lyric Video)
Taylor Swift - Delicate
The Weeknd - Call Out My Name (Official Video)
TWICE What is Love? M/V
VENOM - Official Trailer (HD)
YouTube Rewind: The Shape of 2017 | #YouTubeRewind
title

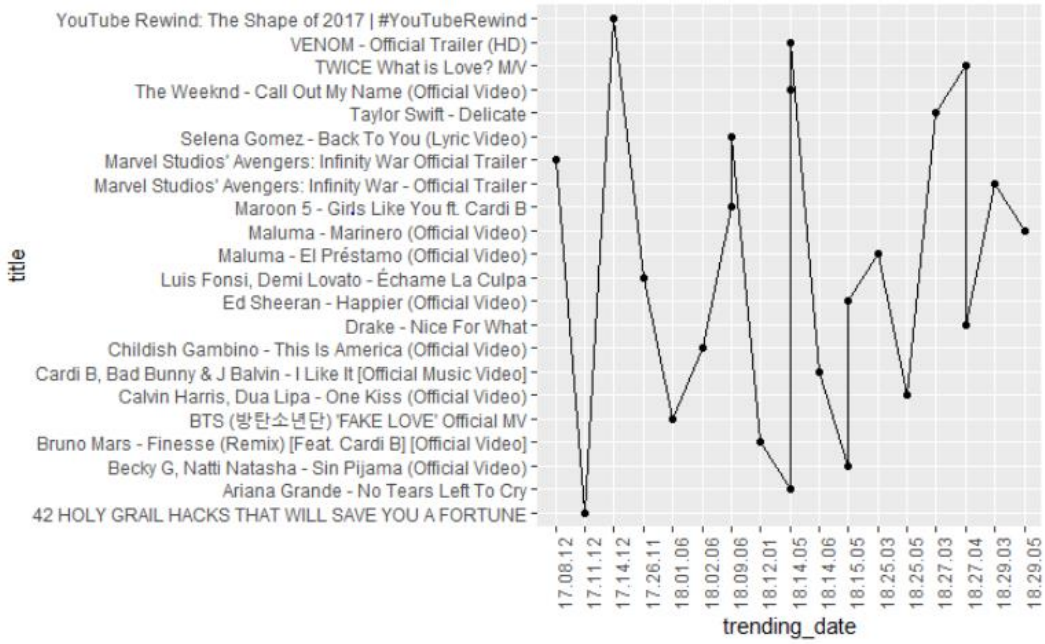
```
plot(p2)
```



"BTS (방탄소년단) 'FAKE LOVE' Official MV" video has highest likes which is 5.6 million and 0.2 million dislikes. "YouTube Rewind:The Shape of 2017|#YouTubeRewind" has highest dislikes as 1.6 million. We can say that with the increase of likes there is simultaneous increase in dislikes

Which video got most trended? (videos that has greater than 50 million views)

```
youtube_above50 %>%  ggplot(aes(x=trending_date,y=title))+
  geom_point(aes(x=trending_date,y=title)) +
  geom_line(aes(group=1)) + theme(legend.position = "bottom", axis.text.x =
element_text(angle = 90))
```



"YouTube Rewind:The Shape of 2017|#YouTubeRewind" is the video that got most trended, in the videos that has greater than 50 million views. Based on the above two visualizations we can say that the video which is most trending is the video which got the highest number of dislike.(videos that has greater than 50 million views.)

Converting publish date and trending date to time format

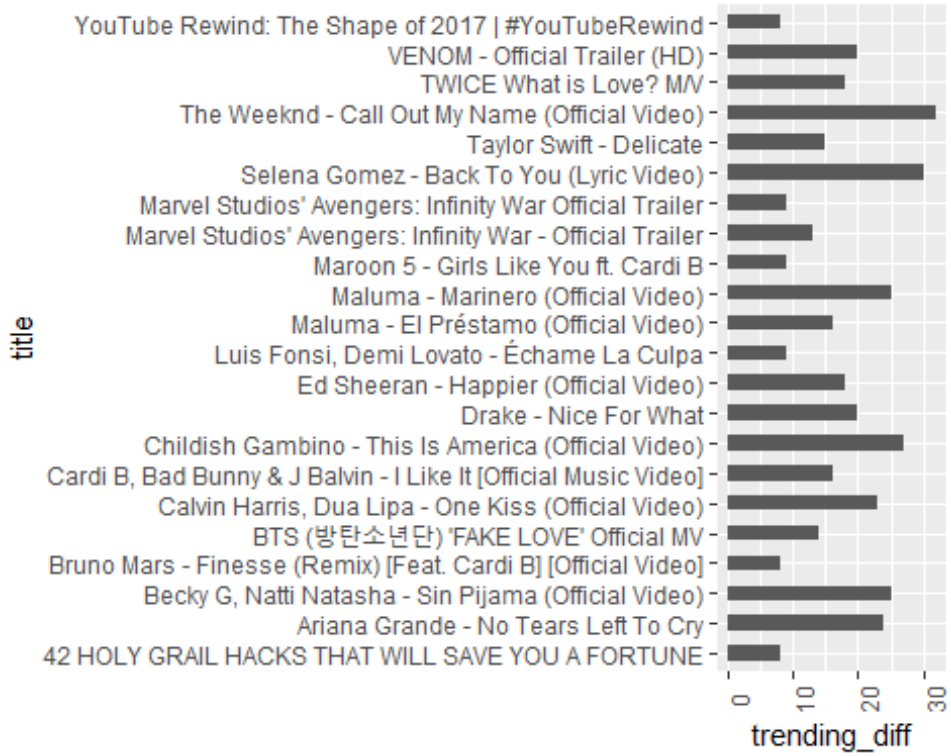
```
youtube <- youtube %>% add_column(trendingdate =
as.Date(youtube$trending_date,format="%y.%d.%m"))
youtube <- youtube %>% add_column(publish_date =
as.Date(youtube$publish_time,format="%y-%d-%m"))
```

Finding trend difference

```
youtube_above50 <- youtube %>% filter(views_per_million>50)
youtube_above50 <- youtube_above50 %>% mutate(trending_diff = trendingdate -
publish_date)
```

How many days after the publish date did the videos start trending?(for videos that has greater than 50 million views)

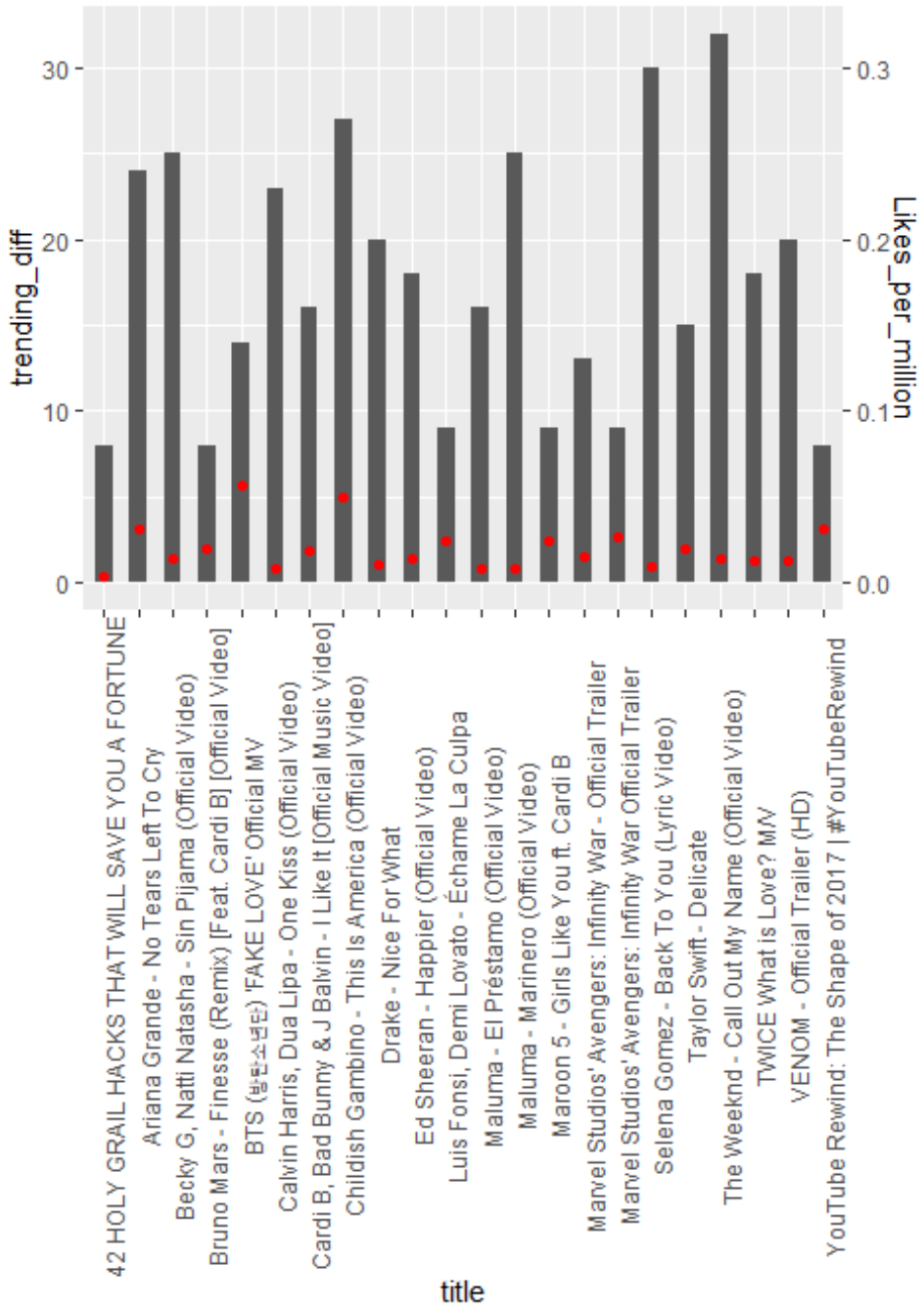
```
youtube_above50 %>% ggplot(aes(x=trending_diff,y=title))+
  geom_bar(stat="identity", width=0.5) + theme(legend.position = "bottom",
axis.text.x = element_text(angle = 90))

## Don't know how to automatically pick scale for object of type difftime.
Defaulting to continuous.
```



"YouTube Rewind: The Shape of 2017 | #YouTubeRewind by YouTube Spotlight", "Bruno Mars - Finesse (Remix) [Feat. Cardi B] [Official Video] by Bruno Mars" and "42 HOLY GRAIL HACKS THAT WILL SAVE YOU A FORTUNE by 5-Minute Crafts" – these 3 videos took less time to get trending, after 8 days from the publish date these videos started to trend.

How trending date and publish date effect likes_per_million? (for videos that has greater than 50 million views)

```
ggplot(youtube_above50) +
geom_bar(aes(y=trending_diff,x=title),stat="identity", width=0.5) +
  geom_point(aes(y=likes_per_million, x=title),stat="identity",color="red") +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90)) +
  scale_y_continuous(sec.axis=sec_axis(~.*0.01,name="Likes_per_million"))
```



Trending_diff doesn't really effect anything on the likes, its just the increase in the number of views per video.

Linear Model

```
model <- lm(views~likes+dislikes+comment_count, data=youtube)
summary(model)

##
## Call:
## lm(formula = views ~ likes + dislikes + comment_count, data = youtube)
```

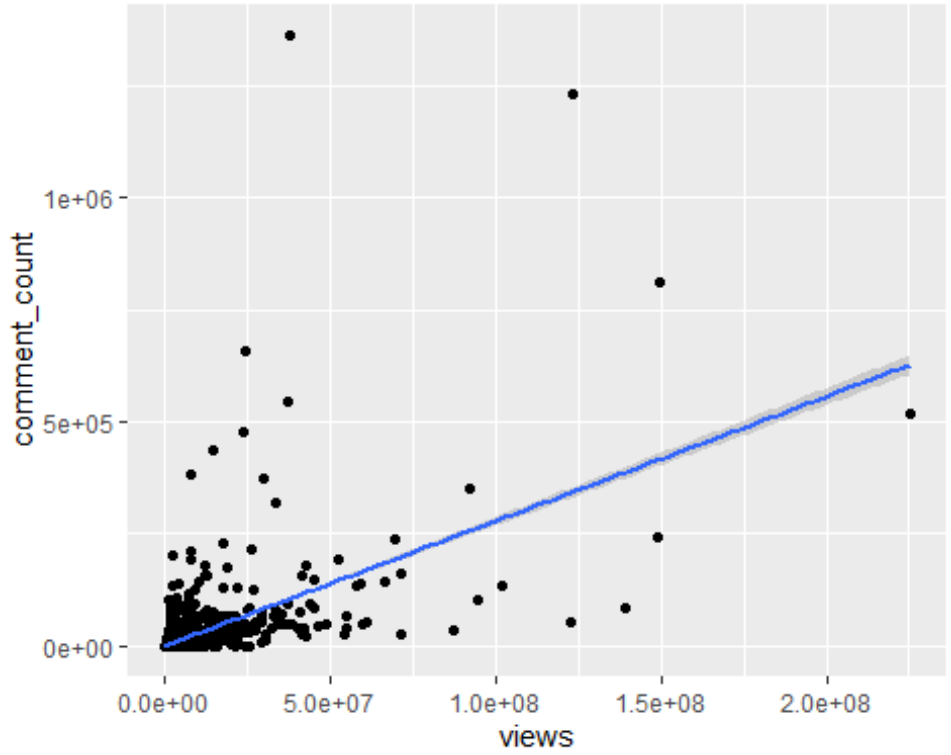
```
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -44194822   -304388   -177052    122235  82163532
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)    2.203e+05  4.311e+04   5.111  3.3e-07 ***
## likes          3.949e+01  3.531e-01 111.859  < 2e-16 ***
## dislikes       8.346e+01  2.033e+00  41.065  < 2e-16 ***
## comment_count -1.113e+02  2.709e+00 -41.105  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3323000 on 6436 degrees of freedom
## Multiple R-squared:  0.7738, Adjusted R-squared:  0.7737
## F-statistic:  7339 on 3 and 6436 DF,  p-value: < 2.2e-16
```

```
coefficients(model)
```

```
##   (Intercept)         likes      dislikes comment_count
##  220333.44800      39.49356      83.46331    -111.34019
```

```
ggplot(data =
youtube,aes(x=views,y=comment_count))+geom_point()+geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```



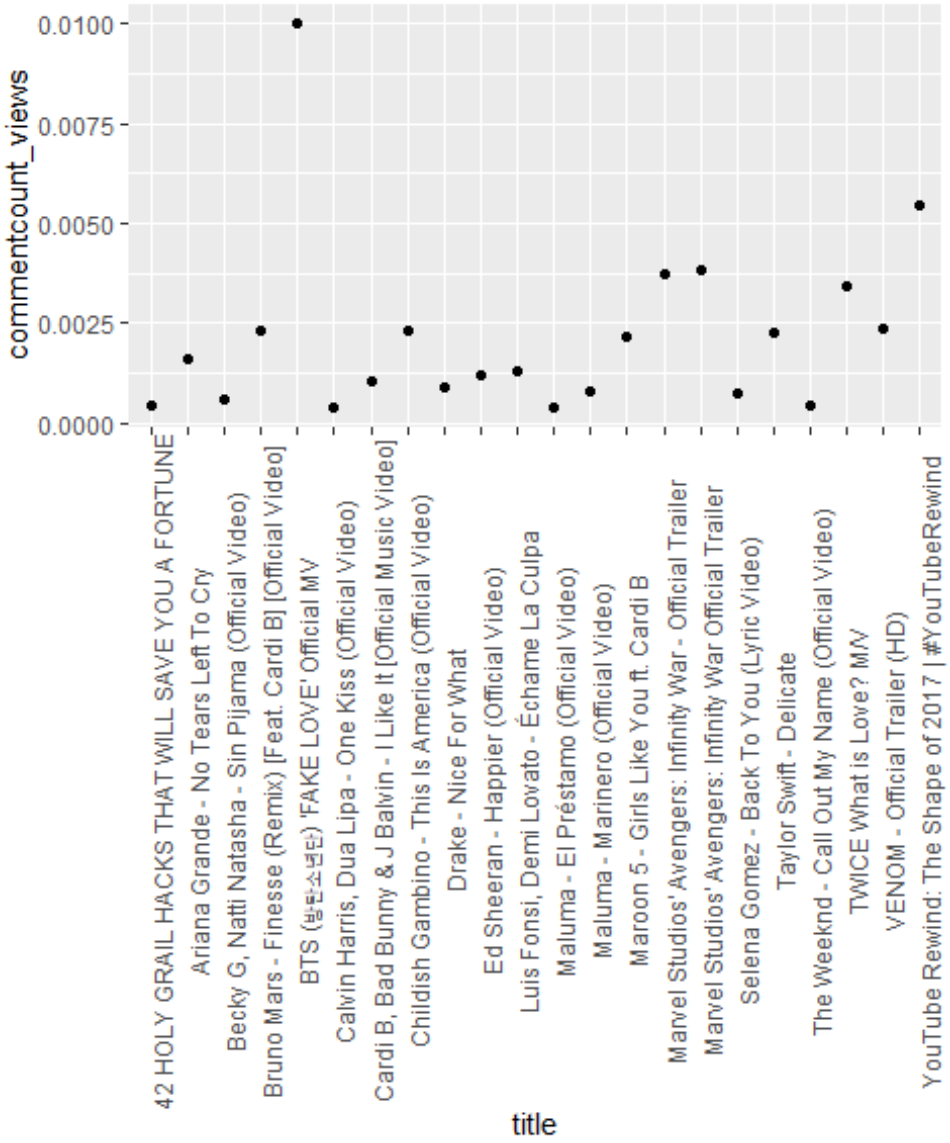
It can be clearly seen from the linear model that views are co-related with likes,dislike and comment count. and with the increase in the number of views there will an increase in the comment count

Providing some more insights based on the linear model.

```
youtube_above50 <- youtube_above50 %>% mutate(commentcount_views =
comment_count/views)
```

Comparing Commentcount with respective to views

```
youtube_above50 %>%
ggplot(aes(x=title,y=commentcount_views))+geom_point()+geom_smooth() +
  theme(legend.position = "bottom", axis.text.x = element_text(angle = 90))

## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



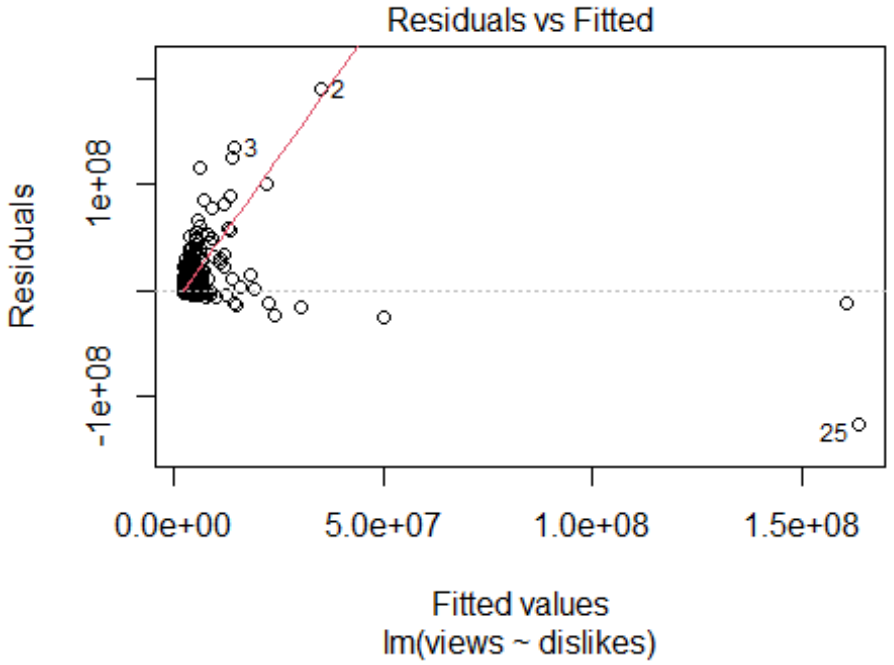
I did a calculation commentcount_views which is comment_count/views, if the ratio is equal to 1, we can determine that the number of views to the video is equal to the number of comments that video have. From the plot we can see that BTS is in the ratio of 0.01, this means “BTS’FAKE LOVE’ Official MV” is the only video among the videos that have more than 50 million views, that has its number of views nearer to the comment count.

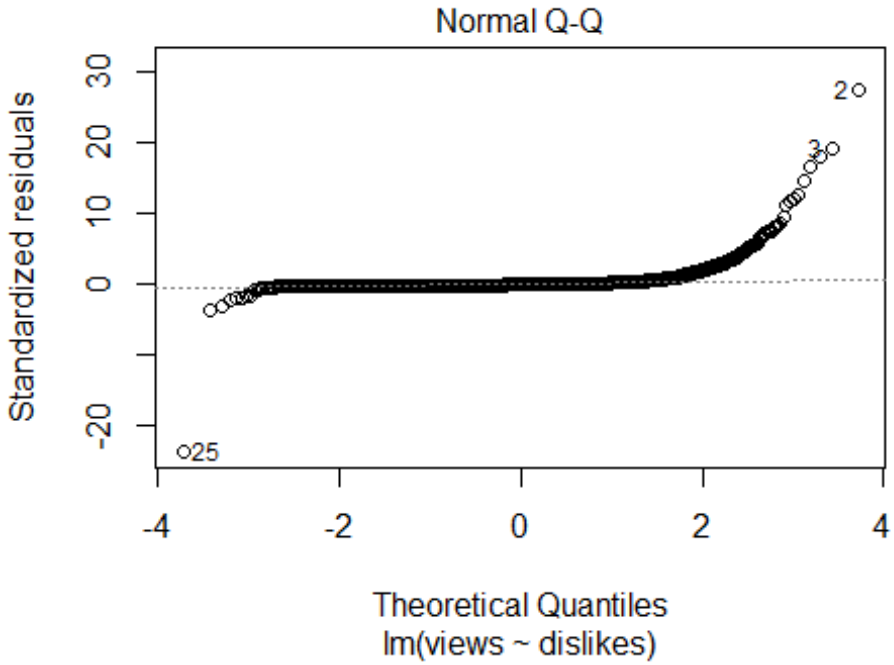
Dividing the data into test data and train data

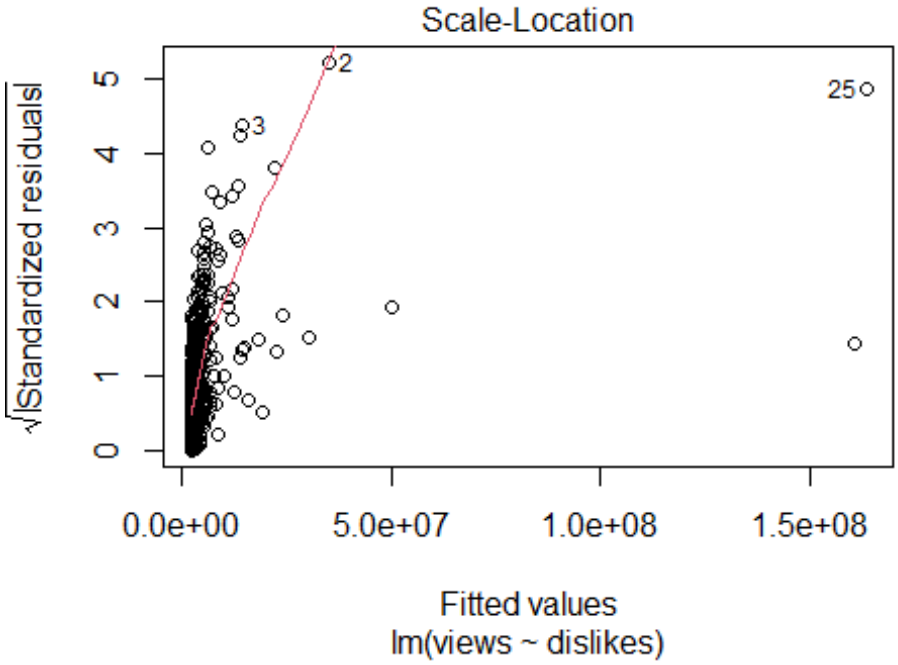
```
train_data <- youtube[0:5000,]
test_data <- youtube[5001:6440,]
```

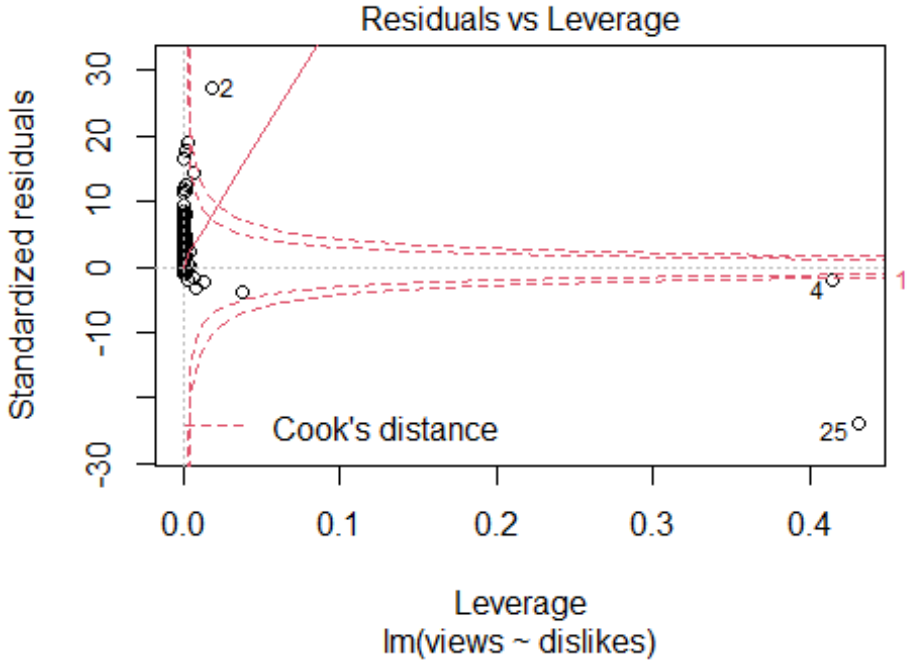
Anova Model

```
model3 <- lm(views~dislikes,data=train_data)
anova(model3)

## Analysis of Variance Table
##
## Response: views
##             Df     Sum Sq    Mean Sq F value    Pr(>F)
## dislikes     1 6.0336e+16 6.0336e+16  1221.1 < 2.2e-16 ***
## Residuals 4998 2.4696e+17 4.9413e+13
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

plot(model3)
```


Residuals vs Fitted
Residuals
1e+08
-1e+08
0.0e+00
5.0e+07
1.0e+08
1.5e+08
Fitted values
lm(views ~ dislikes)
2
3
25


Normal Q-Q
Standardized residuals
30
20
10
0
-20
-4
-2
0
2
4
Theoretical Quantiles
lm(views ~ dislikes)
2
3
25





This technique is used to answer the hypothesis while analyzing multiple groups of data. This is done using views and dislikes.

Developing a Linear Model based on the train data based on views,tredingdate,likes and dislike.

```
model2 <- lm(views~trendingdate+likes+dislikes,data=train_data)
summary(model2)
```

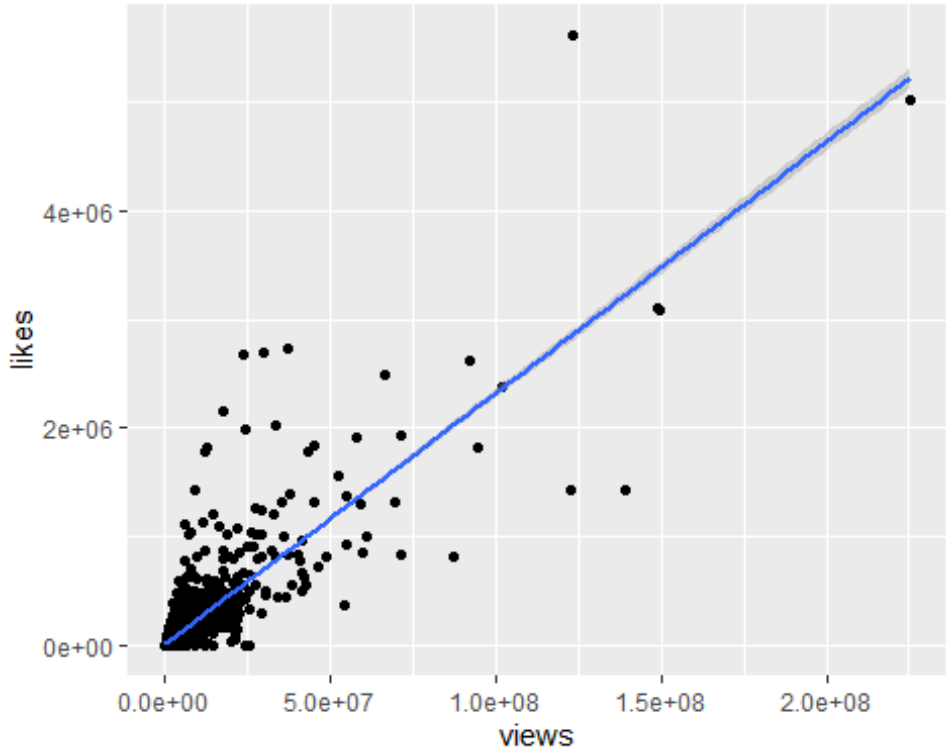
```
##
## Call:
## lm(formula = views ~ trendingdate + likes + dislikes, data = train_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -53792970   -573026   -143182    237701  94927304
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.088e+08  1.673e+07  -6.501 8.78e-11 ***
## trendingdate  6.206e+03  9.518e+02   6.520 7.73e-11 ***
## likes         2.846e+01  3.039e-01  93.656  < 2e-16 ***
## dislikes      2.480e+01  1.814e+00  13.673  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4193000 on 4996 degrees of freedom
## Multiple R-squared:  0.7142, Adjusted R-squared:  0.7141
## F-statistic:  4162 on 3 and 4996 DF,  p-value: < 2.2e-16
```

```
coefficients(model2)
```

```
##   (Intercept)  trendingdate         likes      dislikes
## -1.087609e+08  6.205584e+03  2.845938e+01  2.480465e+01
```

```
ggplot(data =
train_data,aes(x=views,y=likes))+geom_point()+geom_smooth(method = 'lm')
```

```
## `geom_smooth()` using formula 'y ~ x'
```



It can be clearly seen from the linear model that views are co-related with trending date,likes and dislikes. and with the increase in the number of views there will an increase in the number of likes.

Calculating predicted views based on the test data using the linear model that is developed for train data.

```
predicted <- data.frame(predict = predict(model2, newdata = test_data))
glimpse(predicted)
```

```
## Rows: 1,440
## Columns: 1
## $ predict <dbl> 421853.23, -199342.82, 569495.85, -110404.53, 224552.62,
17656~
```

Calculating error value based on the predicted data we derived.

```
error <- data.frame(e_value = (predicted$predict -
test_data$views)/predicted$predict * 100)
mean(error$e_value)
```

```
## [1] 35.82877
```

We can see the error as 35% so there is only 65% accuracy in the predicated data, which is low.

Developing Generalized Linear Model using the train data

```
glm_model <- glm(views~trendingdate+likes+dislikes,data=train_data)
summary(glm_model)
```

```
##
## Call:
## glm(formula = views ~ trendingdate + likes + dislikes, data = train_data)
##
## Deviance Residuals:
##       Min         1Q     Median         3Q        Max
## -53792970    -573026    -143182     237701   94927304
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.088e+08  1.673e+07  -6.501 8.78e-11 ***
## trendingdate  6.206e+03  9.518e+02   6.520 7.73e-11 ***
## likes         2.846e+01  3.039e-01  93.656  < 2e-16 ***
## dislikes      2.480e+01  1.814e+00  13.673  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 1.757782e+13)
##
##     Null deviance: 3.0730e+17  on 4999  degrees of freedom
## Residual deviance: 8.7819e+16  on 4996  degrees of freedom
## AIC: 166684
##
## Number of Fisher Scoring iterations: 2
```

It can be determined from the Generalized Linear Model that views are co-related with trending date, likes and dislikes.

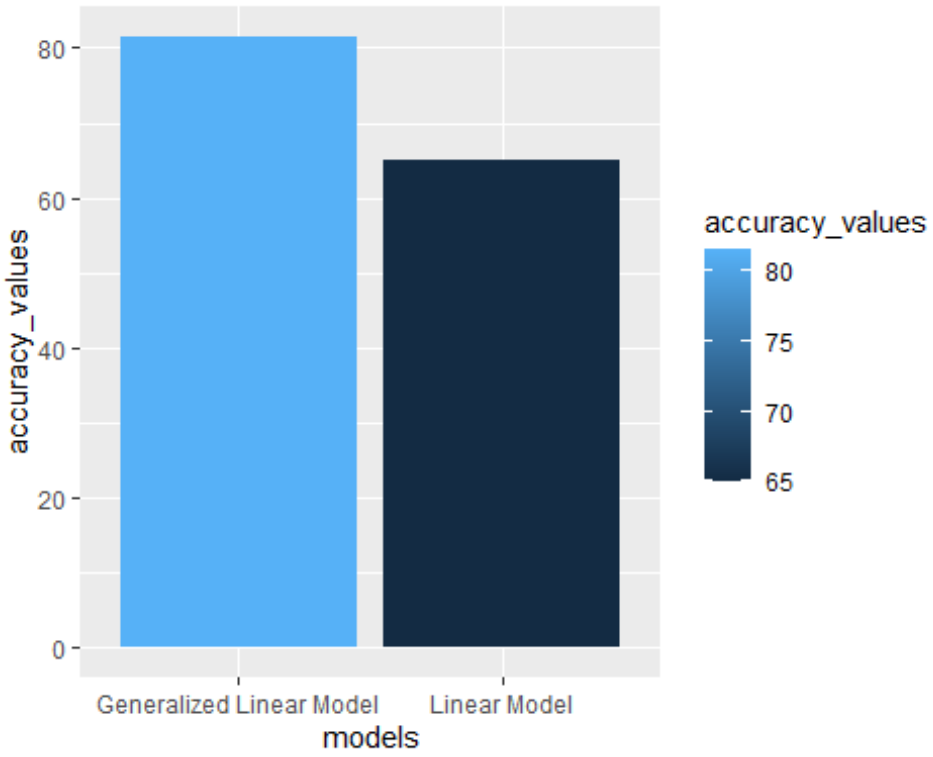
How much accuracy be derived using the Generalized Linear Model

```
library(performance)

## Warning: package 'performance' was built under R version 4.1.2

performance_accuracy(glm_model)

## # Accuracy of Model Predictions
##
## Accuracy: 82.23%
##       SE: 7.41%-points
##   Method: Correlation between observed and predicted
```

We can see that the accuracy is 83.62% meaning there is only 16.38% of error.

Comparing the two models developed on views ~ trendingdate + likes + dislikes

```
models <- c('Linear Model','Generalized Linear Model')
accuracy_values <- c(65,81.51)
ggplot()+ geom_bar(aes(x = models, y = accuracy_values, fill =
accuracy_values),stat = 'identity')
```



From the bar chart above, it can be concluded that Generalized Linear Model is more efficient than Linear Model for the opted data.

References:-

https://www.kaggle.com/datasnaek/youtube-new/data?select=USvideos.csv
https://stat412612.netlify.app/ https://stackoverflow.com/questions/28320228/put-one-line-chart-and-bar-chart-in-one-plot-in-r-not-ggplot https://www.educba.com/anova-in-r/ http://web.stanford.edu/class/stats306a/RforGLM.pdf
https://www.educba.com/bar-charts-in-r/ https://r4ds.had.co.nz/model-basics.html