24 hours after the video was posted on YouTube in the USA (Top Video) in 2020-2023

Worawich Seubpala

2023-09-07

Content

YouTube maintains a list of the top-trending videos on the platform. According to Variety magazine, to determine the year's top-trending videos, there are up to 200 listed trending videos per day.

The dataset includes information about the video title, channel title, publish time, tags, views, likes and dislikes, description, and comment count. The main file has the category_id, but the category details are in a separate JSON file. This dataset includes several months (and counting) of data on daily trending YouTube videos in the USA. The data has 224,788 rows and 16 columns.

data updated on 08/09/2023 extracted the data from Kaggle

Objective

Understanding real-time people's behaviors that consume content and top-performing YouTubers in the USA

Step

1. Load the data
2. Data Preparation
   2.1 Gather
   2.2 Cleaning and Transformation
3. Exploratory Analysis
   3.1 Descriptive Statistics
   3.2 Finding insight by question
4. Conclusion and Recommendations

1.Load the data from CSV and JSON file

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.2     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.2     v tibble    3.2.1
```

```
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(jsonlite)
```

```
##
## Attaching package: 'jsonlite'
##
## The following object is masked from 'package:purrr':
##
##     flatten
```

```
library(ggbeeswarm)
library(scales)
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:purrr':
##
##     discard
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
youtube <- read_csv("US_youtube_trending_data.csv")
```

```
## Warning: One or more parsing issues, call `problems()` on your data frame for details,
## e.g.:
##   dat <- vroom(...)
##   problems(dat)
```

```
## Rows: 224788 Columns: 16
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (7): video_id, title, channelId, channelTitle, tags, thumbnail_link, de...
## dbl  (5): categoryId, view_count, likes, dislikes, comment_count
## lgl  (2): comments_disabled, ratings_disabled
## dttm (2): publishedAt, trending_date
##
## 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.
```

```
head(youtube)
```

```
## # A tibble: 6 x 16
##   video_id    title        publishedAt         channelId channelTitle categoryId
##   <chr>       <chr>        <dttm>              <chr>     <chr>              <dbl>
## 1 3C66w5Z0ixs I ASKED HER~ 2020-08-11 19:20:14 UCvtRTOM~ Brawadis             22
## 2 M9Pmf9AB4Mo Apex Legend~ 2020-08-11 17:00:10 UC0ZV6M2~ Apex Legends         20
## 3 J78aPJ3VyNs I left yout~ 2020-08-11 16:34:06 UCYzPXpr~ jackseptice~         24
## 4 kXLn3HkpjaA XXL 2020 Fr~ 2020-08-11 16:38:55 UCbg_UMj~ XXL                   10
```

```
## 5 VIUo6yapDbc Ultimate DI~ 2020-08-11 15:10:05 UCDVPcEb~ Mr. Kate             26
## 6 w-aidBdvZo8 I Haven't B~ 2020-08-11 20:00:04 UC5zJwsF~ Professor L~         24
## # i 10 more variables: trending_date <dttm>, tags <chr>, view_count <dbl>,
## #   likes <dbl>, dislikes <dbl>, comment_count <dbl>, thumbnail_link <chr>,
## #   comments_disabled <lgl>, ratings_disabled <lgl>, description <chr>
```

```
category <- data.frame(fromJSON("US_category_id.json"))
head(category)
```

```
##                                kind                         etag
## 1 youtube#videoCategoryListResponse HIrK3n45Uw2IYz9_U2-gK1OsXvo
## 2 youtube#videoCategoryListResponse HIrK3n45Uw2IYz9_U2-gK1OsXvo
## 3 youtube#videoCategoryListResponse HIrK3n45Uw2IYz9_U2-gK1OsXvo
## 4 youtube#videoCategoryListResponse HIrK3n45Uw2IYz9_U2-gK1OsXvo
## 5 youtube#videoCategoryListResponse HIrK3n45Uw2IYz9_U2-gK1OsXvo
## 6 youtube#videoCategoryListResponse HIrK3n45Uw2IYz9_U2-gK1OsXvo
##              items.kind                  items.etag items.id
## 1 youtube#videoCategory IfWa37JGcqZs-jZeAyFGkbeh6bc        1
## 2 youtube#videoCategory 5XGylIs7zkjHh5940dsT5862m1Y        2
## 3 youtube#videoCategory HCjFMARbBeWjpm6PDfReCOMOZGA       10
## 4 youtube#videoCategory ra8H7xyAfmE2FewsDabE3TUSq10       15
## 5 youtube#videoCategory 7mqChSJogdF3hSIL-88BfDE-W8M       17
## 6 youtube#videoCategory 0Z6uGkj97NgjD-X3pkA-nL18Hqk       18
##   items.snippet.title items.snippet.assignable  items.snippet.channelId
## 1    Film & Animation                     TRUE UCBR8-60-B28hp2BmDPdntcQ
## 2    Autos & Vehicles                     TRUE UCBR8-60-B28hp2BmDPdntcQ
## 3               Music                     TRUE UCBR8-60-B28hp2BmDPdntcQ
## 4      Pets & Animals                     TRUE UCBR8-60-B28hp2BmDPdntcQ
## 5              Sports                     TRUE UCBR8-60-B28hp2BmDPdntcQ
## 6        Short Movies                    FALSE UCBR8-60-B28hp2BmDPdntcQ
```

2.Data Preparation

```
# Extract the ID and name column
item_id <- category$items.id
category_name <- category$items.snippet$title

new_category <- data.frame(item_id,
                           category_name)
```

Extract data from a JSON file containing the category_id details and only the category names used for analysis.

2.1 Gather the data

```
# Change the type of primary key for joining
youtube$categoryId <- as.character(youtube$categoryId)

full_df <- youtube %>%
  inner_join(new_category, by = c("categoryId" = "item_id" ))

colnames(full_df) <- tolower(colnames(full_df))
```

Changing the data type of the category_id column of the main file (CSV) to character type for the same

type with item_id of the JSON file to join the data, then use it for further classification analysis and change the column names to lowercase for easy extraction of data.

2.2 Cleaning and Transformation

```
# Filter the data within 24 hours after upload a video
final_df <- full_df %>%
  select(video_id,
         publishedat,
         channeltitle,
         title,
         category_name,
         tags,
         view_count,
         likes,
         dislikes,
         comment_count) %>%
  group_by(video_id) %>%
  filter(view_count == min(view_count) & view_count != 0)

glimpse(final_df)
```

```
## Rows: 40,521
## Columns: 10
## Groups: video_id [40,480]
## $ video_id      <chr> "3C66w5Z0ixs", "M9Pmf9AB4Mo", "J78aPJ3VyNs", "kXLn3Hkpja~
## $ publishedat   <dttm> 2020-08-11 19:20:14, 2020-08-11 17:00:10, 2020-08-11 16~
## $ channeltitle  <chr> "Brawadis", "Apex Legends", "jacksepticeye", "XXL", "Mr.~
## $ title         <chr> "I ASKED HER TO BE MY GIRLFRIEND...", "Apex Legends | St~
## $ category_name <chr> "People & Blogs", "Gaming", "Entertainment", "Music", "H~
## $ tags          <chr> "brawadis|prank|basketball|skits|ghost|funny videos|vlog~
## $ view_count    <dbl> 1514614, 2381688, 2038853, 496771, 1123889, 949491, 4704~
## $ likes         <dbl> 156908, 146739, 353787, 23251, 45802, 77487, 47990, 8919~
## $ dislikes      <dbl> 5855, 2794, 2628, 1856, 964, 746, 440, 854, 2158, 4373, ~
## $ comment_count <dbl> 35313, 16549, 40221, 7647, 2196, 7506, 4558, 6455, 6613,~
```

Filtering the data that needs to be used in preparing the report takes video data that has been published within 24 hours to explore the real-time behavior of users who come to use the YouTube platform, including those that do not take videos that no one views.

```
# Convert the data to date type
final_df$date <- as.Date(final_df$publishedat)
final_df$publishedat <- ymd_hms(final_df$publishedat)
```

```
## Warning: 9 failed to parse.
```

```
final_df$hour <- hour(final_df$publishedat)

# Remove unused columns
final_df$publishedat <- NULL
final_df$video_id <- NULL
```

Creating a new column from published data

1. year/month/day
2. The time to upload(hour)

The latest data removes the published and video_id columns because it was not used in the analysis.

```
# Preview the record that have missing values
mean(complete.cases(final_df))
```

```
## [1] 0.9997779
```

```
final_df %>%
  filter(rowSums(is.na(.)) > 0)
```

```
## # A tibble: 9 x 10
##   channeltitle        title       category_name tags  view_count  likes dislikes
##   <chr>               <chr>       <chr>         <chr>      <dbl>  <dbl>    <dbl>
## 1 JDPantojaVEVO       JD Pantoja~ Music         Pant~    1591043 411154    10508
## 2 Netflix Film Club   BLACKPINK ~ Entertainment Blac~    1195279 204827     1227
## 3 JerryRigEverything  All Electr~ Howto & Style Humm~     565030  38198      802
## 4 LiamPayneVEVO       Liam Payne~ Music         Liam~    1439543 307472     2349
## 5 Scott The Woz       Nintendo S~ Gaming        Nint~     884257  73694      604
## 6 Benny Soliven       SERVING AL~ People & Blo~ benn~     333344  22909      300
## 7 The Rich Eisen Show "It Was Sp~ Sports        Rich~     219811   3747       96
## 8 FeidVEVO            Feid - Si ~ Music         Feid~    3104692 294007        0
## 9 RomeoSantosVEVO     Romeo Sant~ Music         Rome~     973108  56382        0
## # i 3 more variables: comment_count <dbl>, date <date>, hour <int>
```

```
# Replace the missing values by "No information"
final_df$hour <- as.character(final_df$hour)

final_df <- final_df%>%
  mutate(hour = ifelse(is.na(hour), "No information", hour))
```

Checking the missing values, the result is 0.9998 (the result equals 1, which means no missing values) for the data that has missing values.

Then extract the records with missing values from the exploration.

The result is 9 records in the hours column.

Therefore, we think that the data set does not have much of a problem. and the number of missing values is very small. So we replace the value with "No information".

3.Exploratory Analysis

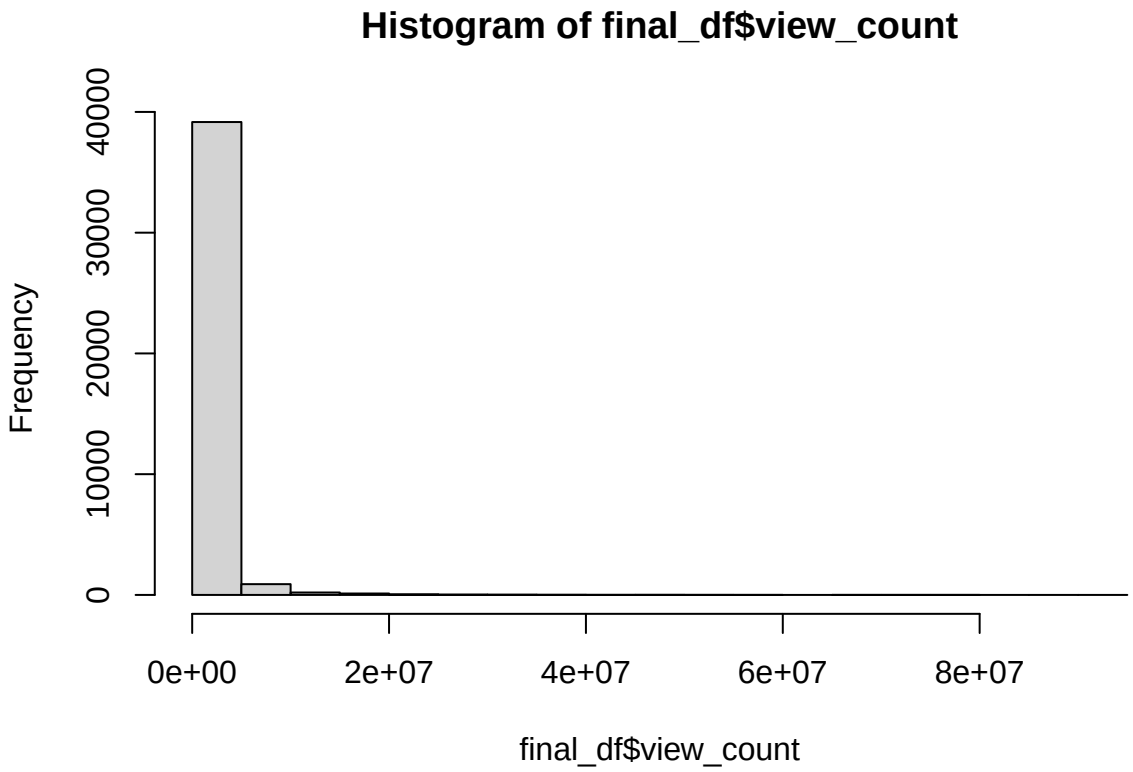
3.1 Descriptive Statistics

```
summary(final_df$view_count)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##     2658   313984   578722  1235340  1211072 91463891
```

Histrogram

```
hist(final_df$view_count)
```




```
mean(final_df$view_count)
```

```
## [1] 1235340
```

```
median(final_df$view_count)
```

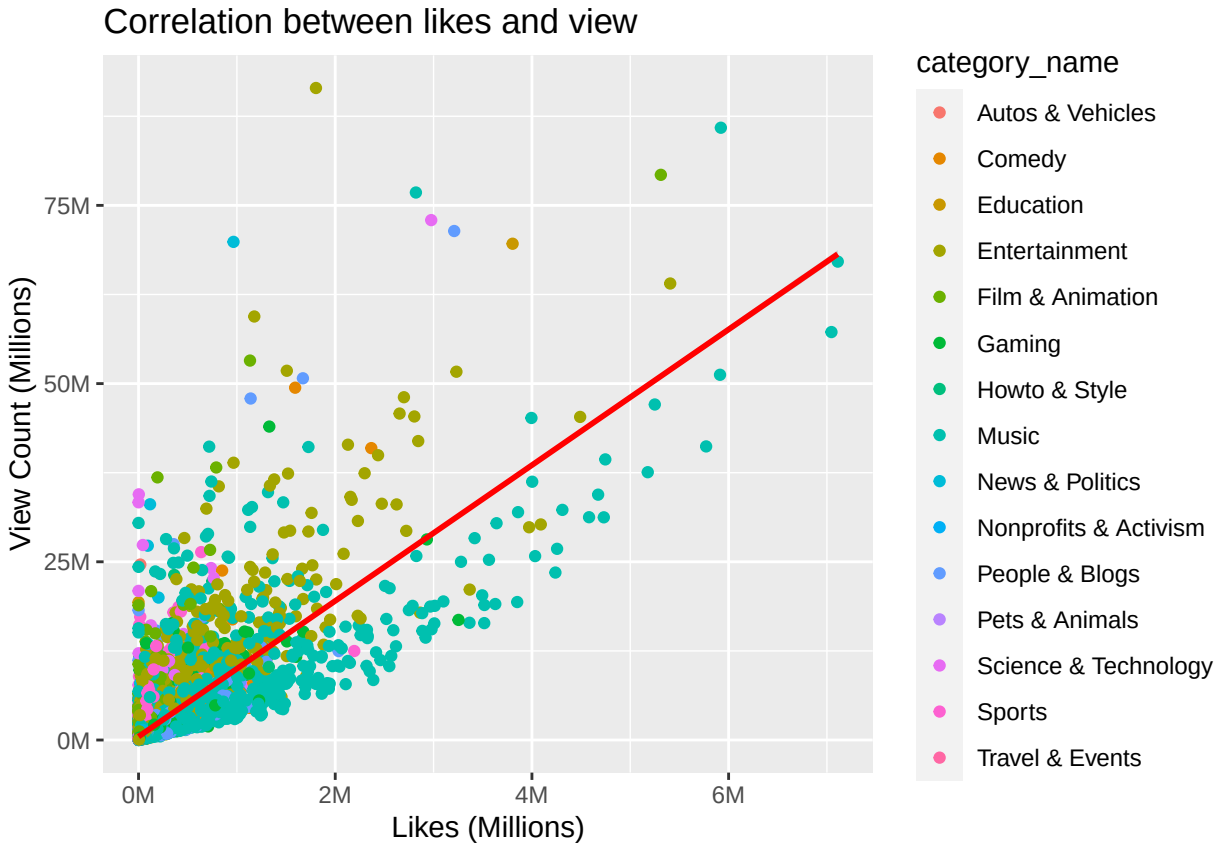
```
## [1] 578722
```

The video view data is right-skewed, meaning there are only a few video clips that receive very high views, causing the overall average view to suffer too much variation. I will use the median value in specifying the central tendency instead of the mean.

Scatter plot

```
ggplot(final_df, aes(likes, view_count, col = category_name)) +
  geom_point()+
  geom_smooth(method = "lm", col = "red")+
  scale_x_continuous(labels = scales::comma_format(scale = 1e-6, suffix = "M")) +
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6, suffix = "M")) +
  labs(title =  "Correlation between likes and view",
       x = "Likes (Millions)", y = "View Count (Millions)")
```

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



The correlation between like and view that has a relationship is in the same direction and linear.

```
cor(final_df[ ,c("likes", "comment_count", "view_count")])
```

```
##                    likes comment_count view_count
## likes          1.0000000     0.6925059  0.7682558
## comment_count 0.6925059     1.0000000  0.4910043
## view_count    0.7682558     0.4910043  1.0000000
```

The correlation between like and view is 0.768

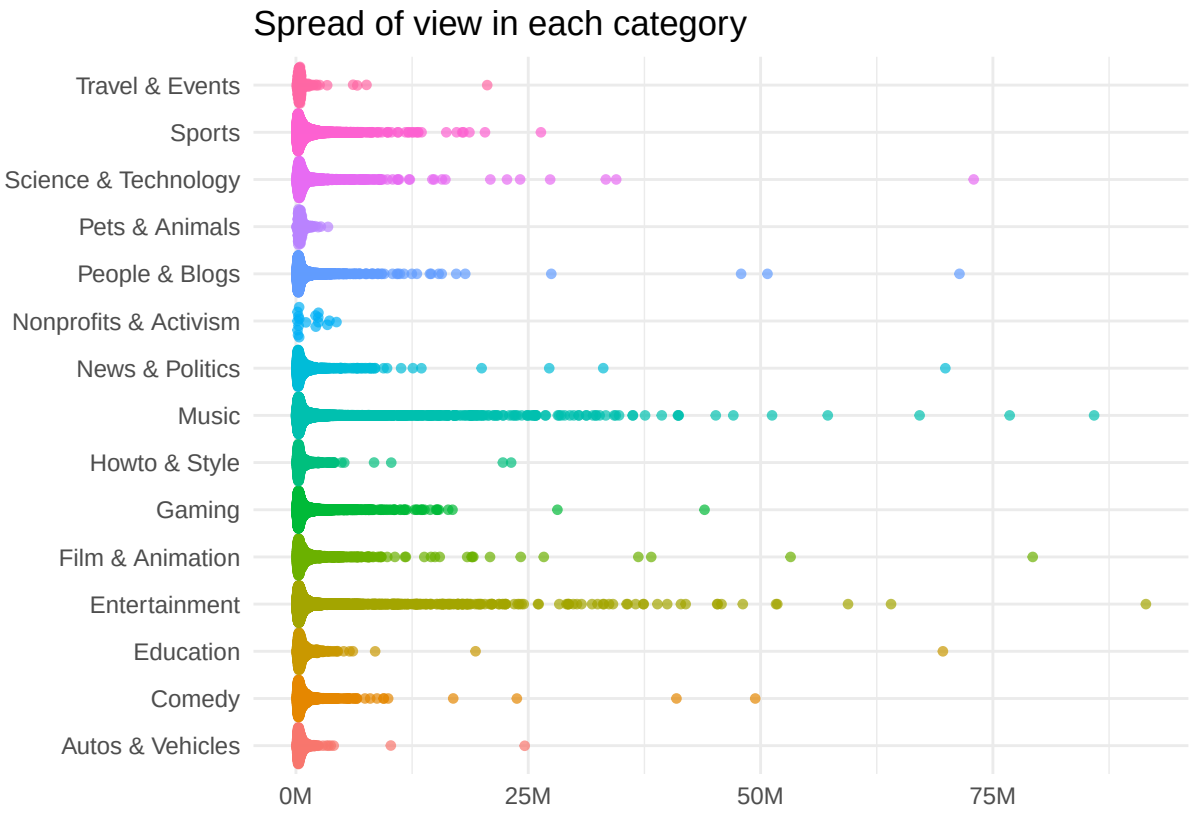
The correlation between comment and view is 0.491

The conclusion is that the number of "likes" on a clip has a more positive relationship with the number of views than comment.

Beeswarm plot

```
ggplot(final_df,
       aes(x = category_name,
           y = view_count,
           color = category_name)) +
  geom_quasirandom(alpha = 0.7,
                   size = 1.2) +
  coord_flip()+
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6, suffix = "M")) +
  labs(title = "Spread of view in each category",
       x = "",
       y = "") +
```

```
  theme_minimal() +
  theme(legend.position = "none")
```




```
# Whisker calculate for detecting outlier
q3 <- quantile(final_df$view_count, .75)
q1 <- quantile(final_df$view_count, .25)

IQR <- q3-q1

upper_outlier <- q3 + 1.5 * IQR

print(upper_outlier)
```

```
##     75%
## 2556704
```

There is a wide distribution of views. If your clip receives approximately 2.5 million views or more, it is considered to be in the minority (upper outliers), and that clip has a chance of being the top real-time performance in Top Video.

calculation upper outlier is Quartile3 + 1.5*IQR (IQR = Quartile3-Quartile1)

3.2 Finding insight by question

1.Which videos are the 10 most popular?

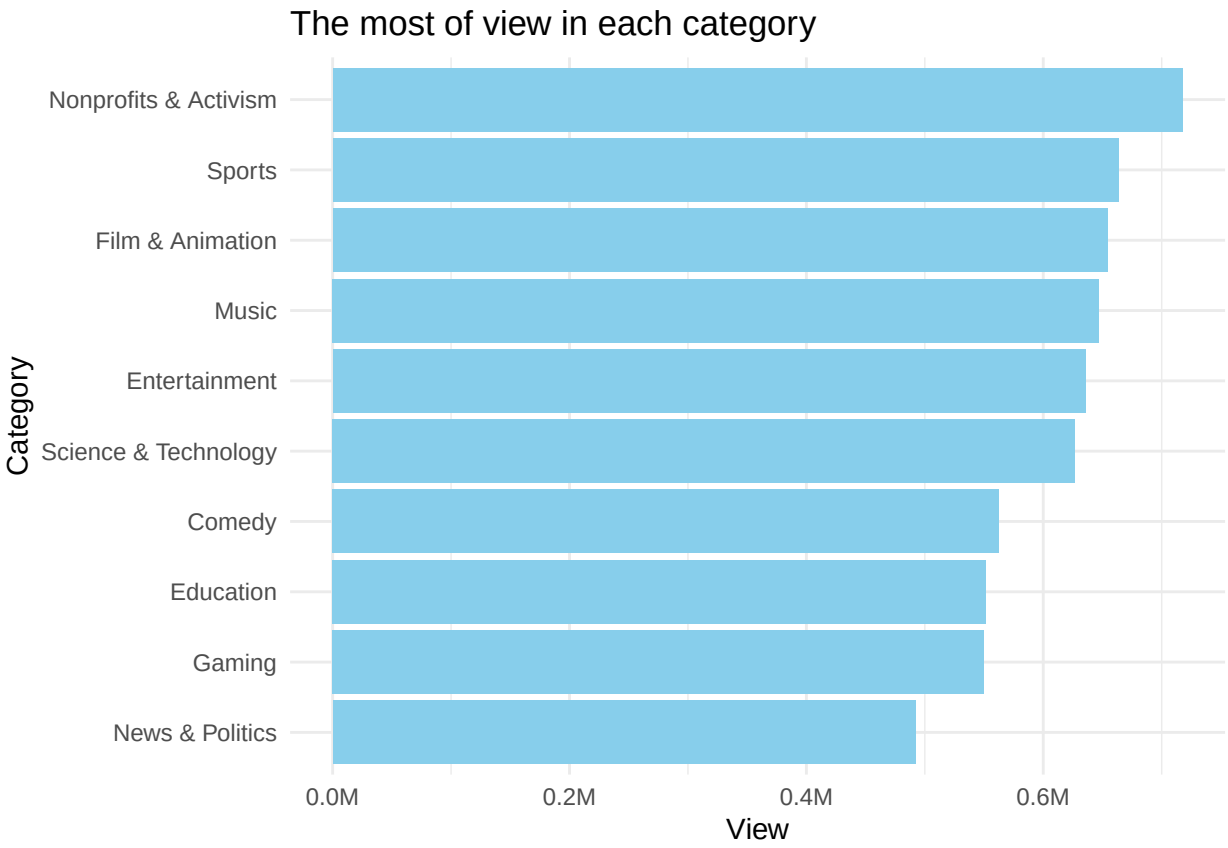
```
# Total view
final_df %>%
  group_by(category_name) %>%
```

```
  summarise(total_view = sum(view_count)) %>%
  arrange(desc(total_view)) %>%
  head(10)
```

```
## # A tibble: 10 x 2
##    category_name         total_view
##    <chr>                      <dbl>
##  1 Entertainment        12069058843
##  2 Music                11315545989
##  3 Gaming                7988231920
##  4 Sports                5724842296
##  5 People & Blogs        3369594646
##  6 Film & Animation      2248904860
##  7 Comedy                1803868900
##  8 Science & Technology  1753910017
##  9 News & Politics       1416041052
## 10 Education              879531622
```

```
# Average view
final_df %>%
  group_by(category_name) %>%
  summarise(median_view = median(view_count)) %>%
  top_n(10) %>%
  ggplot(aes(reorder(category_name, median_view), median_view))+
  geom_col(fill = "skyblue")+
  coord_flip()+
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6, suffix = "M")) +
  labs(title = "The most of view in each category",
       x = "Category", y = "View")+
  theme_minimal()
```

```
## Selecting by median_view
```



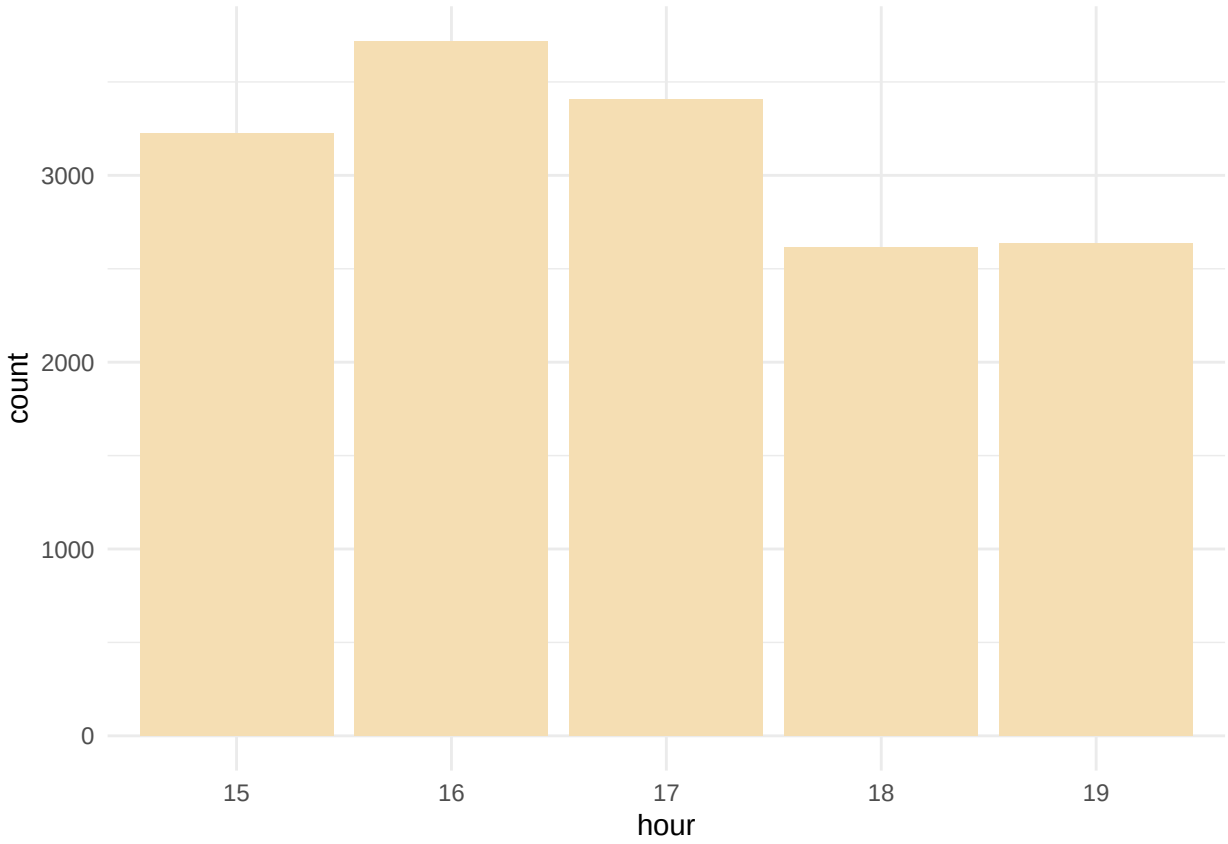
The video category with the most views is entertainment with 12,000 million views, with music next and education having the fewest views.

But there may also be a factor in the number of people who upload videos, meaning that there may be fewer people who make educational content compared to people who make entertainment or music.

So let's try to measure the performance of the video by averaging the number of viewers of each clip. I can see that entertainment music is not on top 2, but it became a category about Nonprofits & Activism instead, and education was not the last.

2.When do YouTubers upload videos most often?

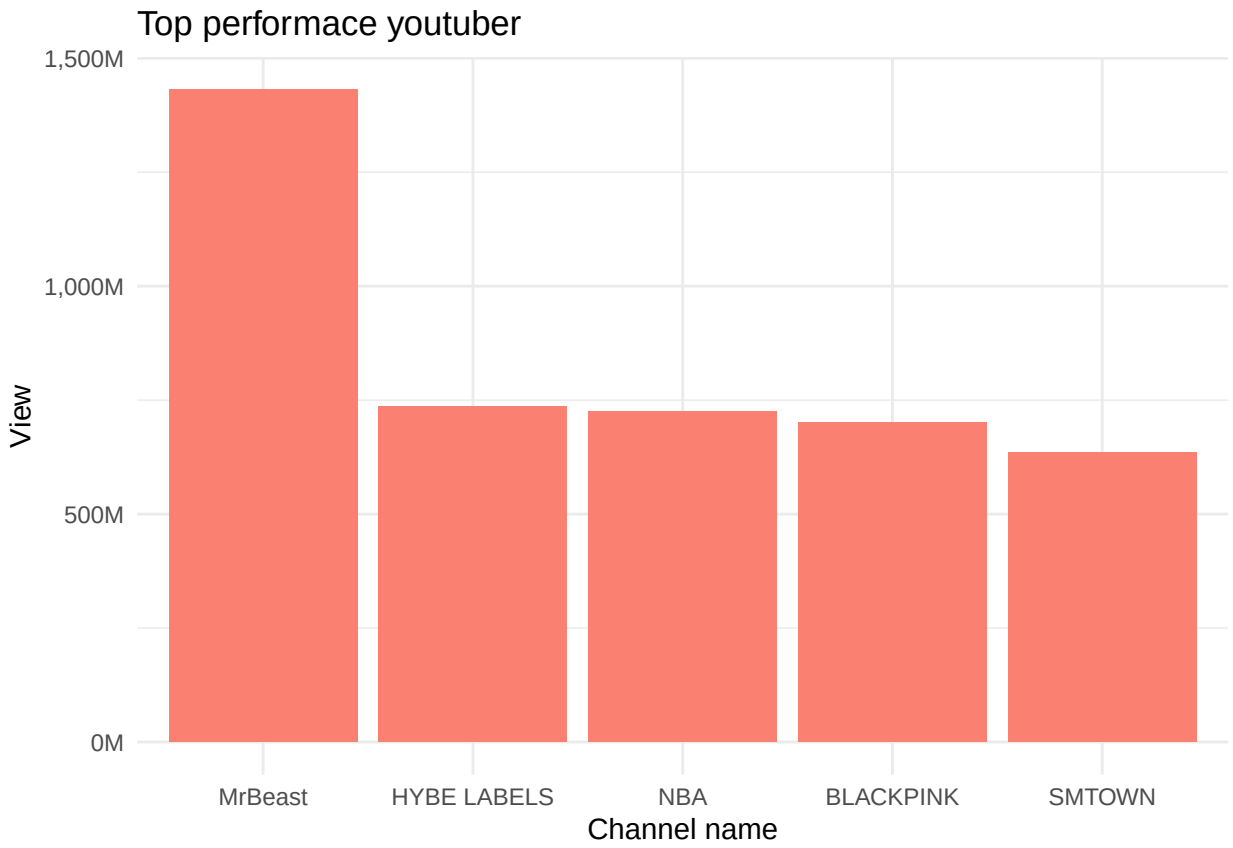
```
final_df %>%
  group_by(hour) %>%
  summarise(count = n()) %>%
  arrange(desc(count)) %>%
  head(5)%>%
  ggplot(aes(x = hour, y = count))+
  geom_col(fill = "wheat")+
  theme_minimal()
```



It's pretty clear that the owner of the clip usually posts during the afternoon (3 p.m.–7 p.m.).

3.Which YouTube channel has the most views?

```
final_df %>%
  group_by(channeltitle)%>%
  summarise(view = sum(view_count))%>%
  arrange(desc(view))%>%
  head(5) %>%
  ggplot(aes(reorder(channeltitle,-view), view))+
  geom_col(fill = "salmon")+
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6, suffix = "M"))+
  labs(title = "Top performace youtuber",
       x = "Channel name", y = "View")+
  theme_minimal()
```



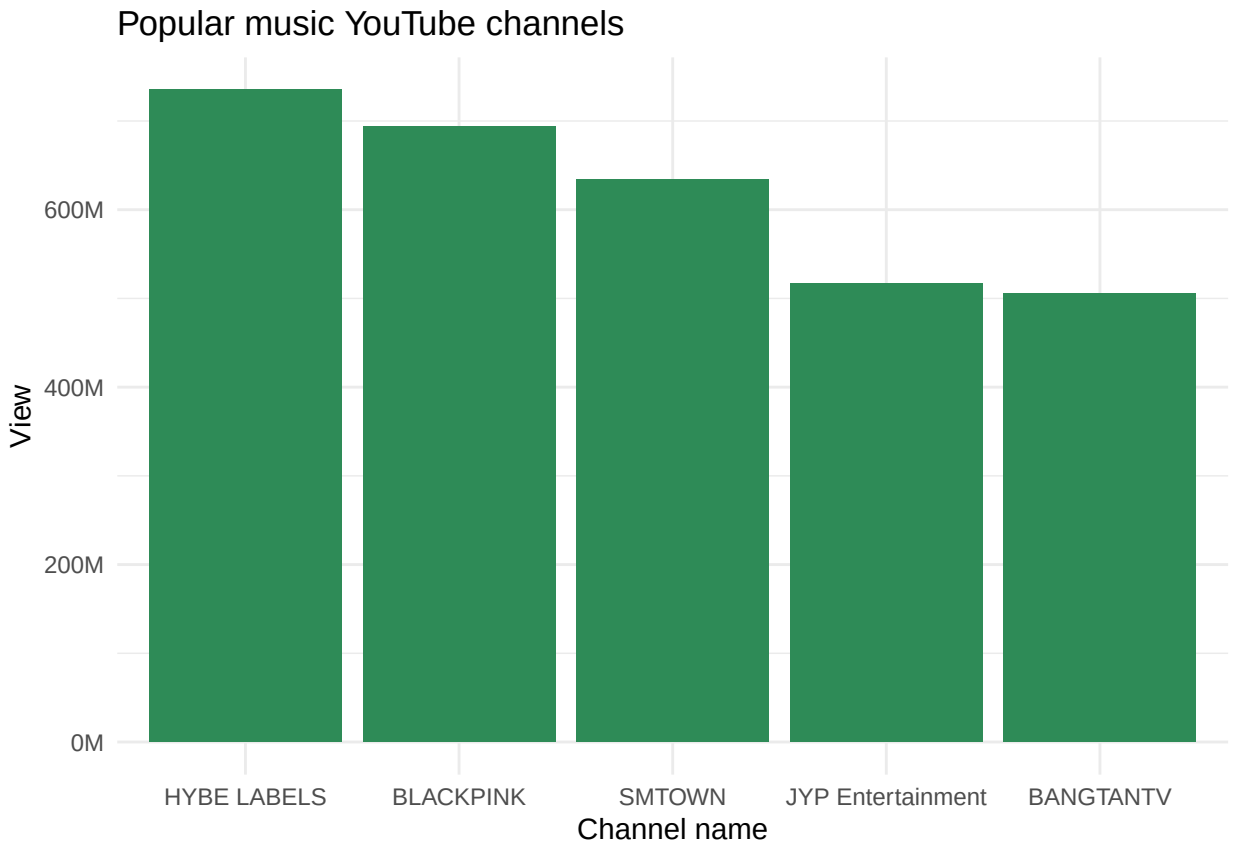
The most popular Youtuber is MrBreast. This channel makes entertainment content. And their channels are different in the dataset because they have a lot of viewers, produce few videos, and focus on quality.

Followed by a channel from the Korean artist record label HYBE.

Labels has very famous artists like BTS and New Jeans, which means that America likes and follows the work of Korean artists quite a bit.

4.Which YouTube music channel is the most popular?

```
final_df %>%
  filter(category_name == "Music") %>%
  group_by(channeltitle)%>%
  summarise(view = sum(view_count))%>%
  arrange(desc(view))%>%
  head(5) %>%
  ggplot(aes(reorder(channeltitle,-view), view))+
  geom_col(fill = "seagreen")+
  scale_y_continuous(labels = scales::comma_format(scale = 1e-6, suffix = "M"))+
  labs(title = "Popular music YouTube channels",
       x = "Channel name", y = "View")+
  theme_minimal()
```



The first one with the most views will be HYBE LABELS mentioned earlier, followed by Blackpink.

5.Which brand of mobile phone content will people watch between the iPhone and Samsung?

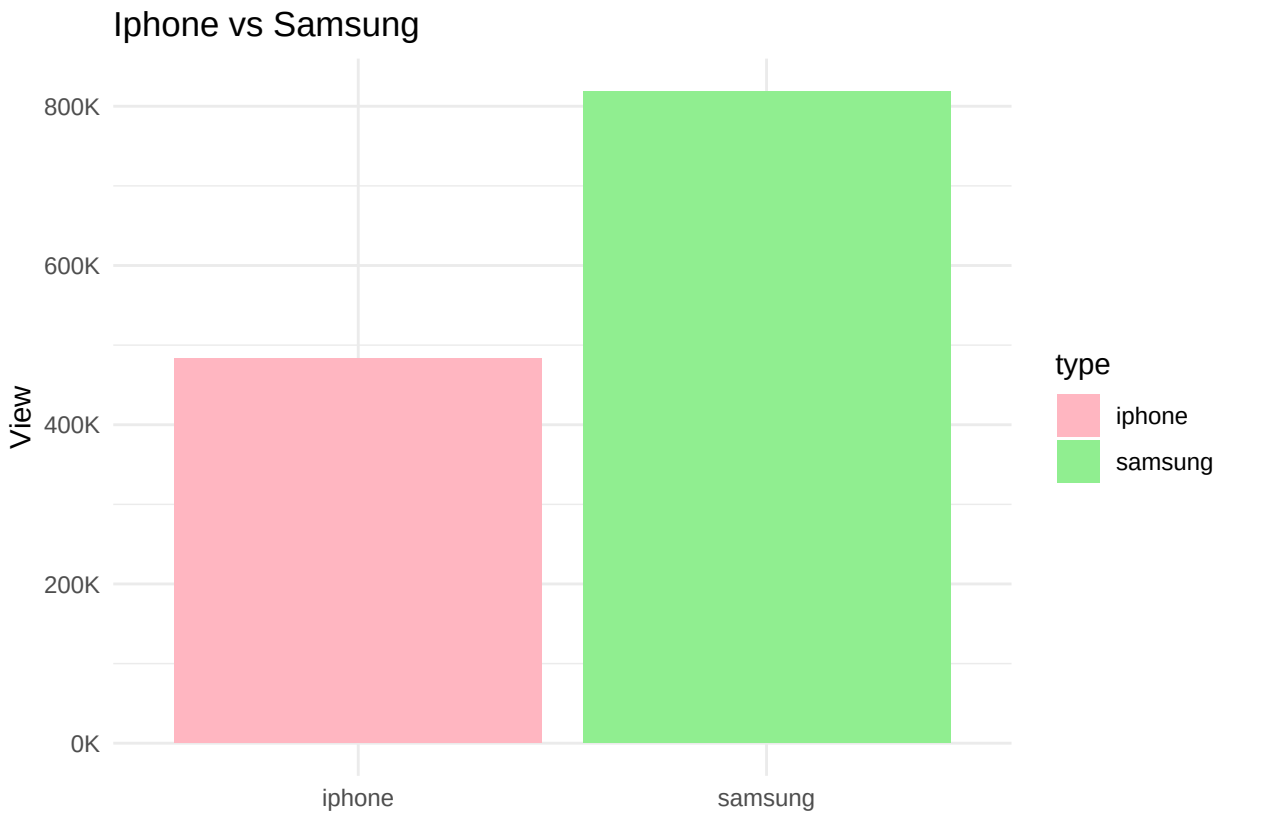
```
phone <- final_df %>%
  mutate(type = ifelse(grepl("samsung", tags, ), "samsung",
                       ifelse(grepl("iphone",tags), "iphone", "others" ))
  )

# Total view
phone %>%
  filter(type %in% c("samsung","iphone"))%>%
  group_by(type) %>%
  summarise(total_view = sum(view_count))
```

```
## # A tibble: 2 x 2
##   type    total_view
##   <chr>        <dbl>
## 1 iphone   183084834
## 2 samsung   54839394
```

```
# Average view
phone %>%
  filter(type %in% c("samsung","iphone")) %>%
  group_by(type) %>%
  summarise(view = median(view_count)) %>%
  ggplot(aes(type, view)) +
  geom_col(aes(fill = type))+
```

```
scale_fill_manual(values = c("lightpink","lightgreen"))+
theme_minimal() +
scale_y_continuous(labels = scales::comma_format(scale = 1e-3, suffix = "K"))+
labs(title = "Iphone vs Samsung",
     x = "", y = "View")+
theme_minimal()
```



This case is the same as the first case in that it's true that the number of people viewing content about iPhones is the highest. But it might be because there are more people making content about the iPhone than Samsung, but when looking at the average views per video, Samsung has more viewers.

4.Conclusion and Recommendations

-Most of the content on YouTube is related to entertainment and music, but if we were to measure the average number of views per video, it would be content about nonprofits & activism.

-Within 24 hours of posting a video with more than 25 million views, your clip is likely to be the top real-time performance in the top video.

-The more likes a video has, the more views it tends to have.

-The central tendency of view is the median, not the mean.

-The highest number of videos posted was from 4:00 p.m. - 5:00 p.m.

-The most popular Youtuber is MrBreast

-The most popular music channel is HYBE LABELS