YouTube Analysis

Anna Charchyan

2024-04-23

```r
library(ggplot2)
library(dplyr)
library(tidyr)
library(psych)
library(patchwork)
library(ggplot2)
library(ggthemes)
library(hrbrthemes)
library(lubridate)
library(forecast)
library(Rtsne)
library(stats)
library(cluster)
library(factoextra)
library(tidymodels)
library(corrplot)
library(forcats)
```

```r
#loading the data data set
yt_data_us19 <- read.csv("2019-YT-trending-videos-US_111.csv", header = TRUE)
yt_data_gb19 <- read.csv("2019-YT-trending-videos-GB_111.csv", header = TRUE)
yt_data_us <- read.csv("USvideos_111.csv", header = TRUE)
yt_data_gb <- read.csv("GBvideos_111.csv", header = TRUE)
```

#Data Cleaning and Pre Processsing

```r
yt_data_gb19 <- yt_data_gb19 %>%
  rename(
    publish_time = publishedAt,
    channel_title = channelTitle,
    views = view_count
  ) %>%
  select(-channelId)

yt_data_us19 <- yt_data_us19 %>%
  rename(
    publish_time = publishedAt,
    channel_title = channelTitle,
    views = view_count
  )%>%
  select(-channelId)
```

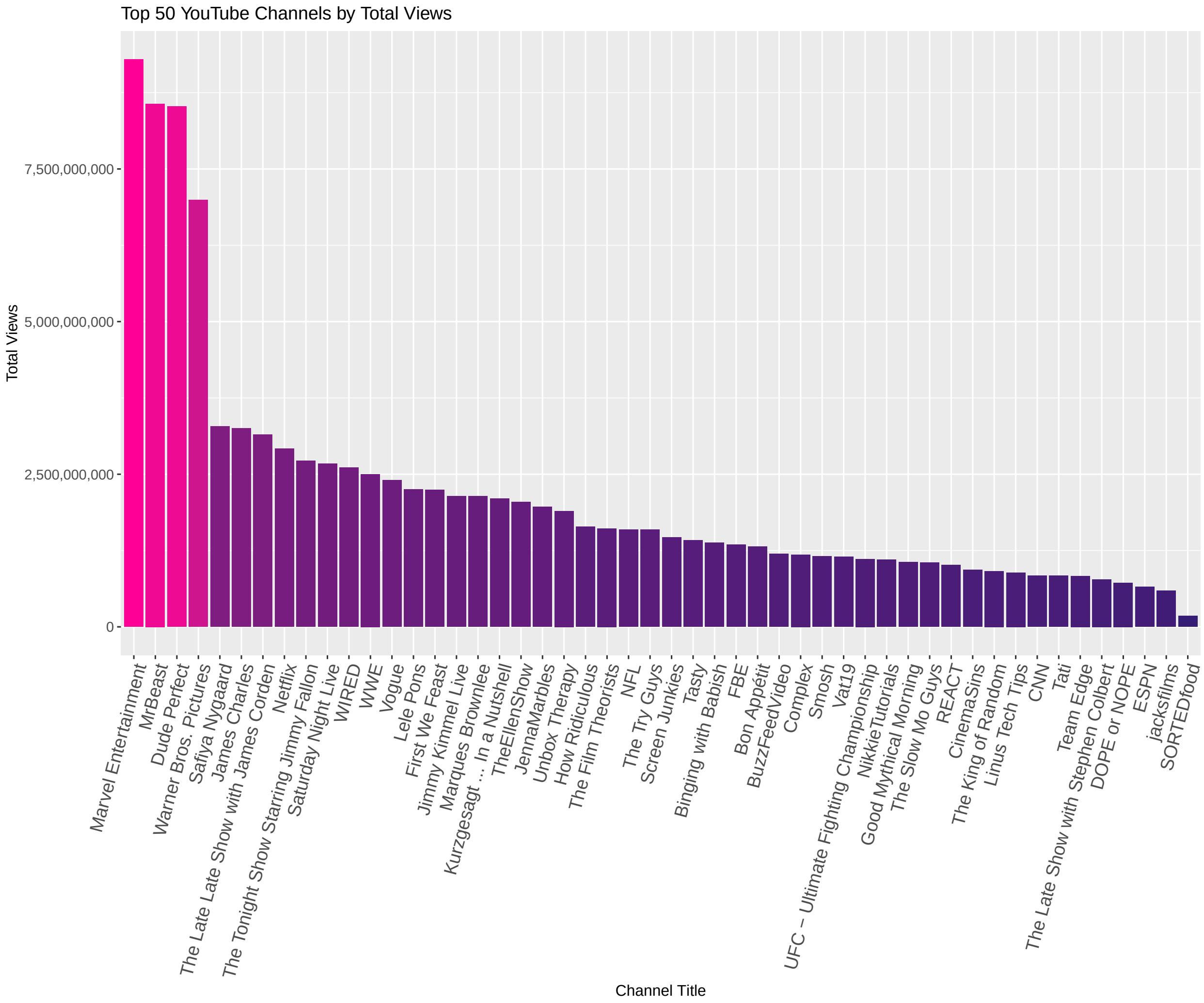
```r
yt_data_us <- rename(yt_data_us, category = category_id)

# Create a named vector with the mappings
category_mapping <- c(
  `1` = "Film & Animation",
  `2` = "Autos & Vehicles",
  `10` = "Music",
  `17` = "Sports",
  `18` = "Short Movies",
  `19` = "Travel & Events",
  `20` = "Gaming",
  `21` = "Videoblogging",
  `22` = "People & Blogs",
  `23` = "Comedy",
  `24` = "Entertainment",
  `25` = "News & Politics",
  `26` = "Howto & Style",
  `27` = "Education",
  `28` = "Science & Technology",
  `29` = "Nonprofits & Activism",
  `30` = "Movies",
  `31` = "Anime/Animation",
  `32` = "Action/Adventure",
  `33` = "Classics",
  `34` = "Comedy",
  `35` = "Documentary",
  `36` = "Drama",
  `37` = "Family",
  `38` = "Foreign",
  `39` = "Horror",
  `40` = "Sci-Fi/Fantasy",
  `41` = "Thriller",
  `42` = "Shorts",
  `43` = "Shows",
  `44` = "Trailers"
)
# Replace the numerical values with category names
yt_data_us$category <- category_mapping[as.character(yt_data_us$category)]

yt_data_us <- yt_data_us %>%
    rename(views = view_count,
           publish_time = publish_at)

yt_data_gb <- rename(yt_data_gb, category = category_id)

# Create a named vector with the mappings
category_mapping <- c(
  `1` = "Film & Animation",
  `2` = "Autos & Vehicles",
  `10` = "Music",
  `17` = "Sports",
  `18` = "Short Movies",
  `19` = "Travel & Events",
  `20` = "Gaming",
  `21` = "Videoblogging",
  `22` = "People & Blogs",
  `23` = "Comedy",
  `24` = "Entertainment",
  `25` = "News & Politics",
  `26` = "Howto & Style",
  `27` = "Education",
  `28` = "Science & Technology",
  `29` = "Nonprofits & Activism",
  `30` = "Movies",
  `31` = "Anime/Animation",
  `32` = "Action/Adventure",
  `33` = "Classics",
  `34` = "Comedy",
  `35` = "Documentary",
  `36` = "Drama",
  `37` = "Family",
  `38` = "Foreign",
  `39` = "Horror",
  `40` = "Sci-Fi/Fantasy",
  `41` = "Thriller",
  `42` = "Shorts",
  `43` = "Shows",
  `44` = "Trailers"
)

# Replace the numerical values with category names
yt_data_gb$category <- category_mapping[as.character(yt_data_gb$category)]
```

```
combined_yt<-bind_rows(yt_data_gb, yt_data_us, yt_data_gb19, yt_data_us19)
```

In this part, we combine the datasets.

```
channel_stats <- combined_yt%>%group_by(channel_title) %>%summarise(trending_count = n(), total_views =
```

```
# Reorder channel_title by total_views in decreasing order
channel_stats$channel_title <- fct_reorder(channel_stats$channel_title, channel_stats$total_views, .desc

# Create a bar plot with gradient fill
ggplot(channel_stats, aes(x = channel_title, y = total_views, fill = total_views)) +
  geom_bar(stat = "identity") +
  labs(x = "Channel Title", y = "Total Views", title = "Top 50 YouTube Channels by Total Views") +
  scale_y_continuous(labels = scales::comma) + # Format Y axis labels to display standard numbers
  scale_fill_gradient(low = "#351c75", high = "#ff0096") +  # Add gradient fill
  theme(axis.text.y = element_text(size = 10),  # Adjust y-axis label size
        axis.text.x = element_text(angle = 75, hjust = 1, size = 13),  # Rotate x labels for better vis
        legend.position = "none")  # Hide the legend
```



The bar plot above shows the Top 50 Youtube Channels by Total views, Marvel entertainment and MrBeast have highest total views in the data.

Here we are interested in VEVO videos, we group the filtered data by the channel_title. For each group, it calculates several summary statistics, including the count of distinct video_ids, the total number of trending days, and the cumulative sums of views, likes, dislikes, and comments

```
main_yt<-combined_yt %>% filter(grepl("VEVO", channel_title)) %>%group_by(channel_title) %>% summarise(
```

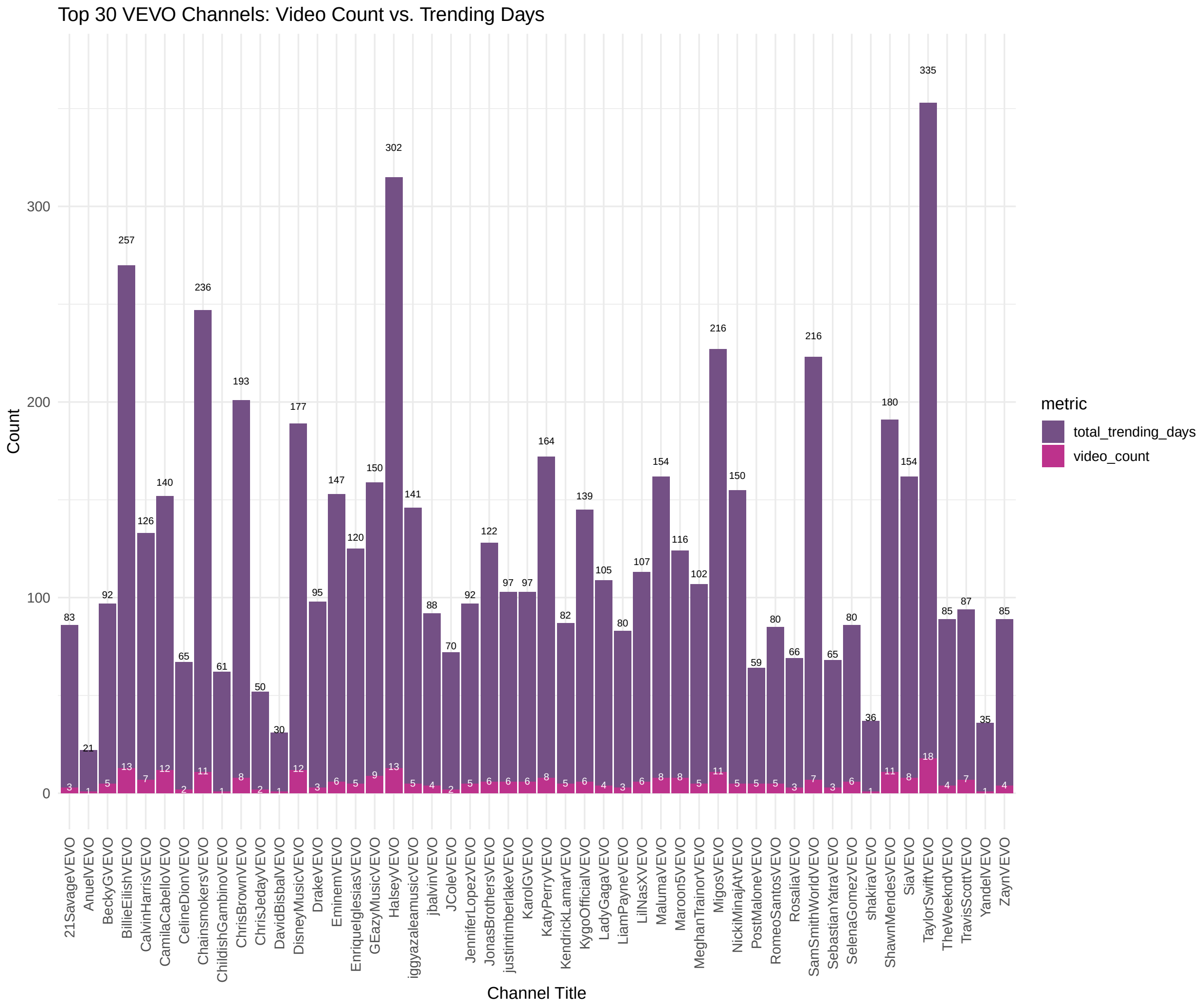
```
vevo_data <- combined_yt %>%
  filter(grepl("VEVO", channel_title)) %>%
  group_by(channel_title) %>%
  summarise(
    video_count = n_distinct(video_id),        # Count unique video IDs for number of songs
    total_trending_days = n(),                 # Count total entries as total trending days
    total_views = sum(views, na.rm = TRUE)     # Total views might help to select top channels
  ) %>%
  arrange(desc(total_views)) %>%
  slice_max(order_by = total_views, n = 50)
```

```r
vevo_data_long <- vevo_data %>%
  pivot_longer(
    cols = c(video_count, total_trending_days),
    names_to = "metric",
    values_to = "value"
  )
```

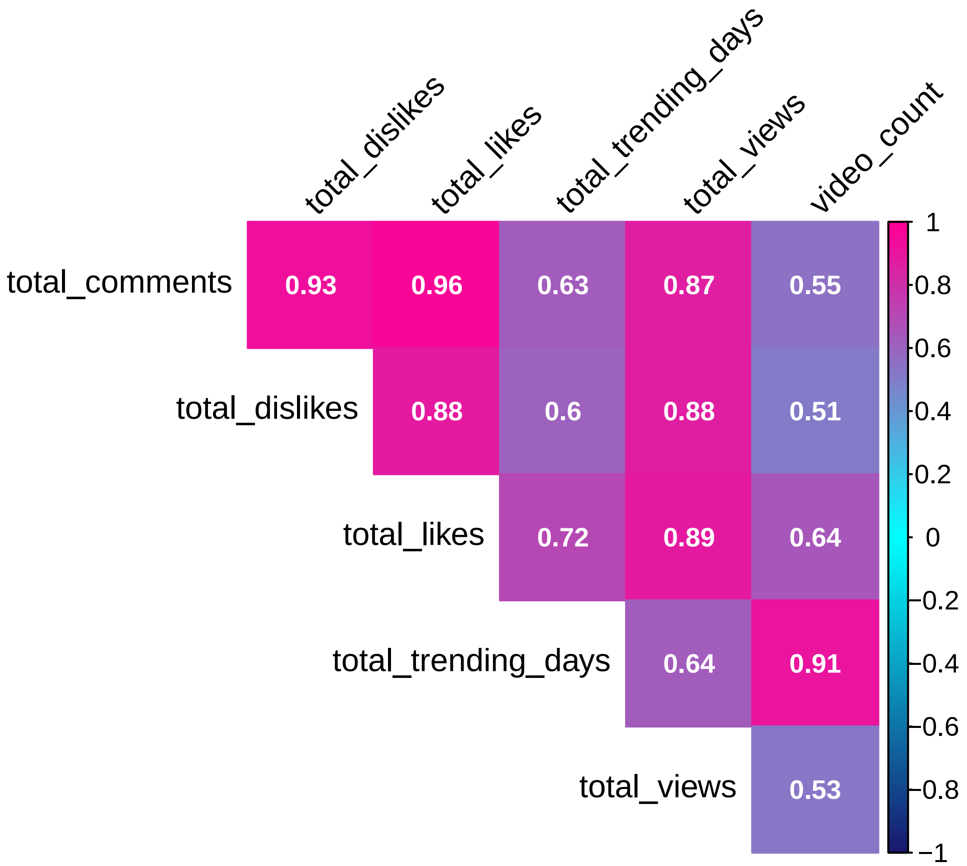
```r
vevo_artists <- combined_yt %>%
  filter(grepl("VEVO", channel_title)) %>%
  distinct(channel_title) %>%  # Get distinct channel titles
  summarise(number_of_vevo_artists = n_distinct(channel_title))  # Count distinct artists

# View the number of unique VEVO artists
print(vevo_artists)
```

```
##   number_of_vevo_artists
## 1                    367
```

```r
# Create a stacked bar plot with custom colors and labels on each segment
ggplot(vevo_data_long, aes(x = channel_title, y = value, fill = metric)) +
  geom_bar(stat = "identity") +  # Use geom_bar with stat="identity" for pre-summarized data

  # Add text labels above each segment with conditional coloring
  geom_text(aes(label = value, y = value, color = metric),  # Add color aesthetic within geom_text
            position = position_stack(vjust = 1.05),  # Position labels just above the bars
            size = 2.5,  # Set text size
            show.legend = FALSE) +  # Hide the legend for text colors

  labs(x = "Channel Title", y = "Count", title = "Top 30 VEVO Channels: Video Count vs. Trending Days")
  scale_fill_manual(values = c("video_count" = "#bd328c", "total_trending_days" = "#745085")) +  # Cust
  scale_color_manual(values = c("video_count" = "white", "total_trending_days" = "black")) +  # Custom

  # Improve X-axis labels readability and adjust theme settings
  theme_minimal(base_size = 12) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1, vjust = 0.5),
        plot.title = element_text(size = 14),
        axis.title = element_text(size = 12),
        plot.background = element_rect(fill = "white", color = "white"),
        panel.background = element_rect(fill = "white", color = "white"),
        legend.position = "right")
```

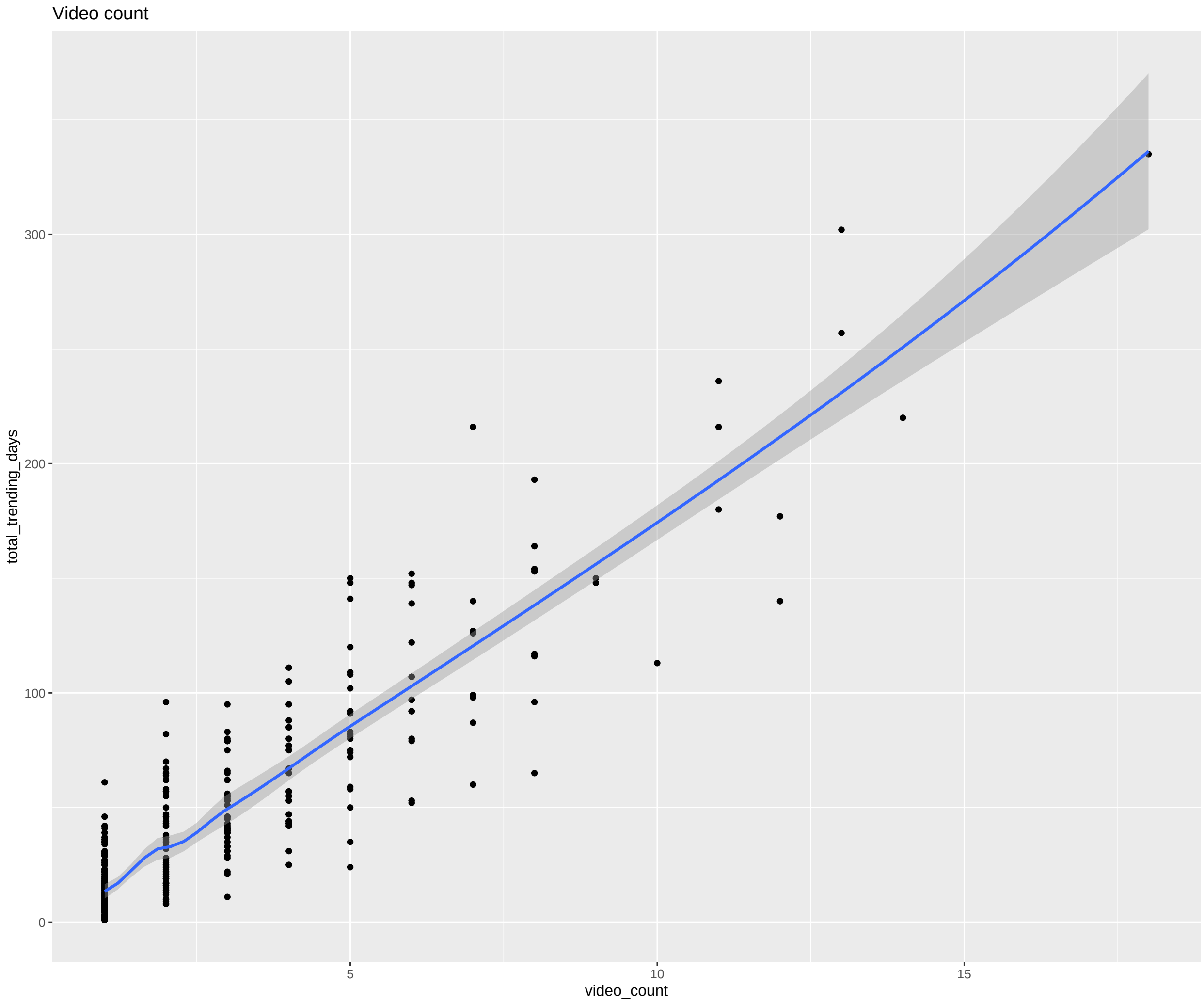


```r
# Compute the correlation matrix for specified variables in the dataset
correlation_matrix <- cor(main_yt[, c('video_count', 'total_trending_days', 'total_views',
                                      'total_likes', 'total_dislikes', 'total_comments')])

# Plotting the lower triangle of the correlation matrix
corrplot(correlation_matrix,
         type = "upper",  # Display only the lower triangle
         method = "color",  # Use color to fill squares
         order = "alphabet",
         addCoef.col = "white",  # White color for coefficients
         diag = FALSE,
         tl.srt = 45,  # Rotate text labels
         tl.col = "black",
         col = colorRampPalette(c("midnightblue","cyan", "#ff0096"))(100), # Color gradient
         cl.pos = 'r', # Position the color legend on the right
         number.cex = 0.8, # Adjust the size of coefficient text
         cl.cex = 0.8) # Adjust the size of the color legend text
```



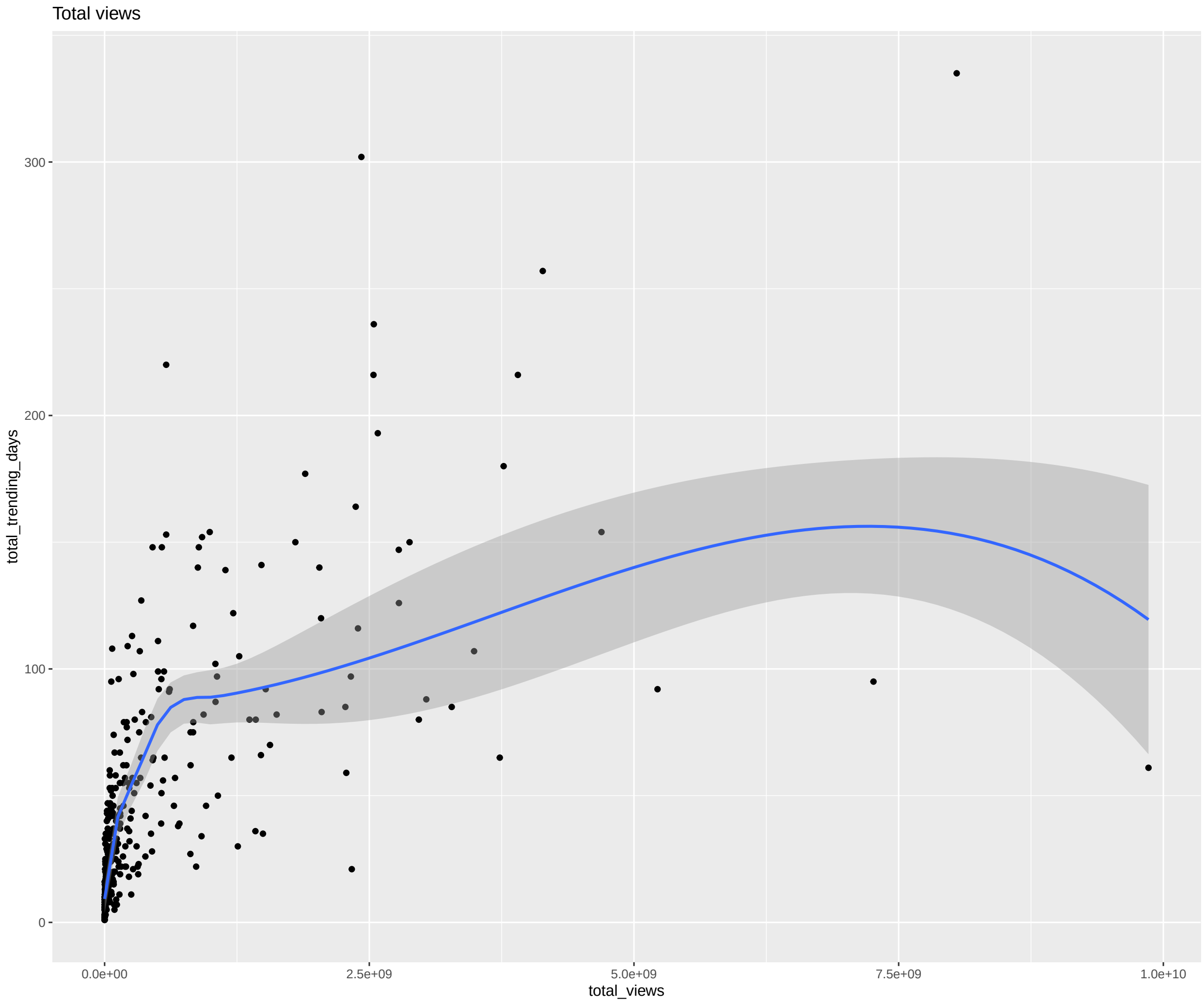
```
c1<-ggplot(data=main_yt)+geom_point(mapping=aes(x=video_count, y=total_trending_days))+geom_smooth(mappi
c2<-ggplot(data=main_yt)+geom_point(mapping=aes(x=total_views, y=total_trending_days))+geom_smooth(mappi
c3<-ggplot(data=main_yt)+geom_point(mapping=aes(x=total_likes, y=total_trending_days))+geom_smooth(mappi
c4<-ggplot(data=main_yt)+geom_point(mapping=aes(x=total_dislikes, y=total_trending_days))+geom_smooth(ma
c5<-ggplot(data=main_yt)+geom_point(mapping=aes(x=total_comments, y=total_trending_days))+geom_smooth(ma

c1
```

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



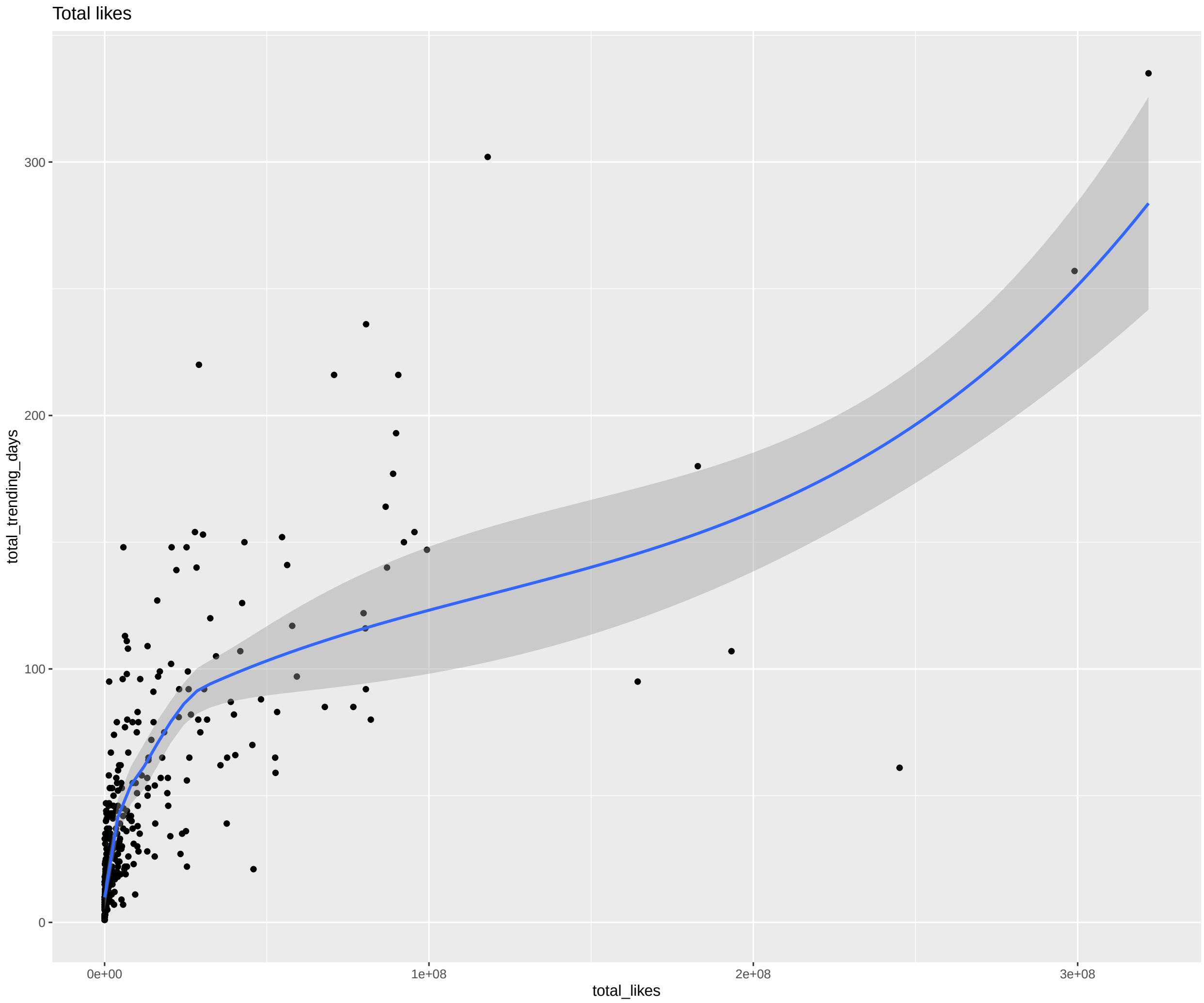
```
c2
```

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



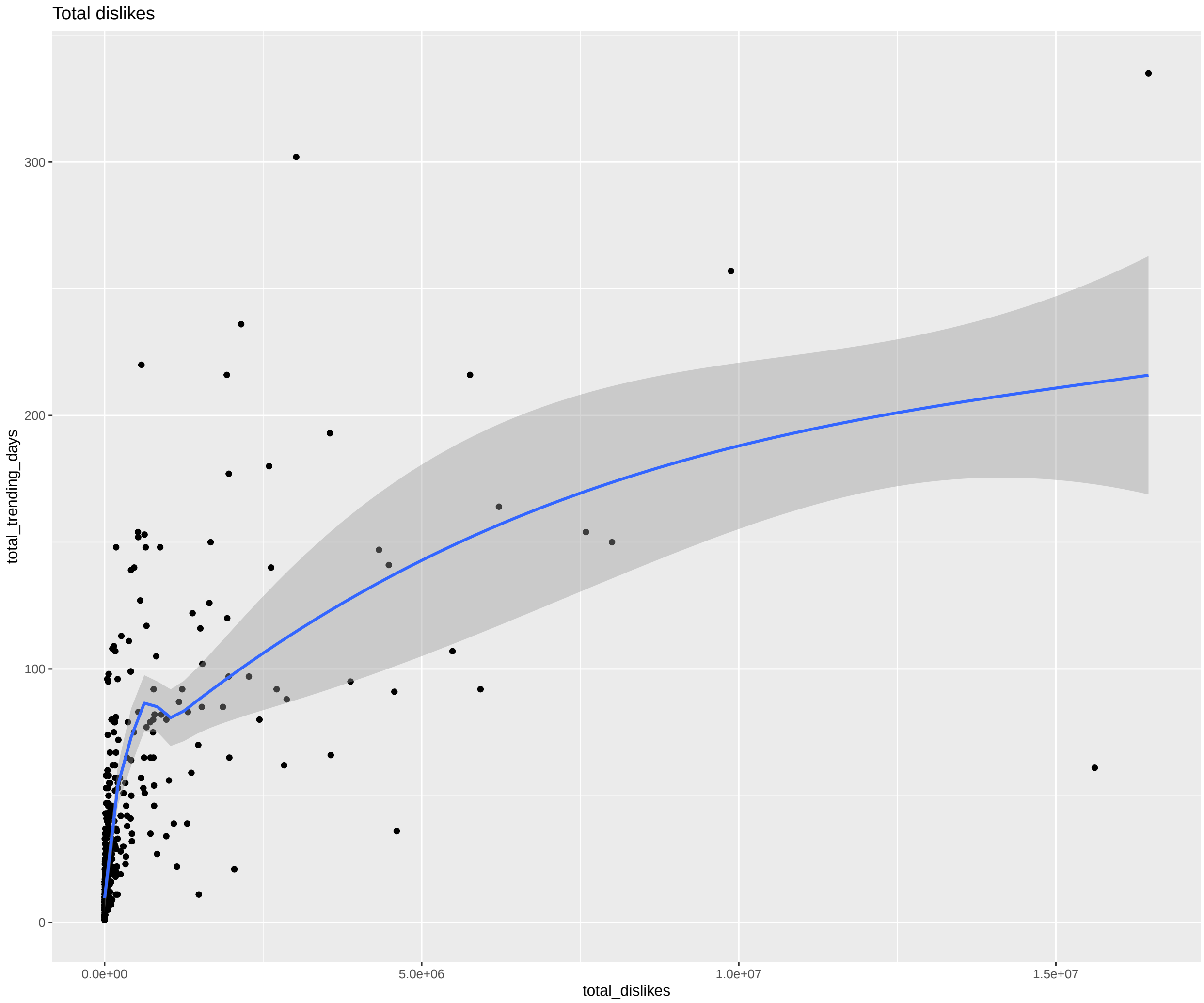
```
c3
```

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



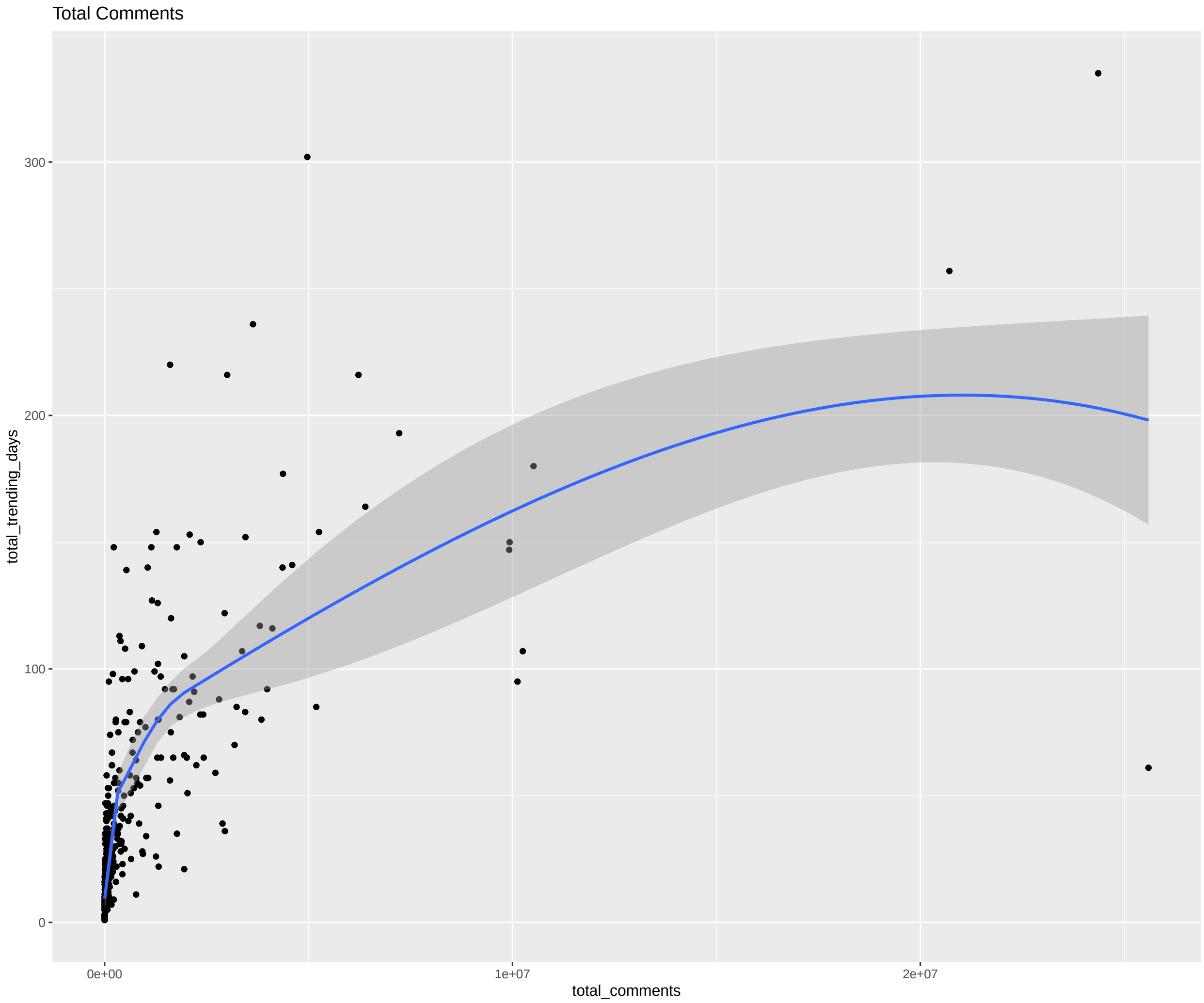
```
c4
```

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



```
c5
```

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



```
cor(main_yt$video_count, main_yt$total_trending_days)
```

```
## [1] 0.9091936
```

```
cor(main_yt$total_views, main_yt$total_trending_days)
```

```
## [1] 0.637624
```

```
cor(main_yt$total_likes, main_yt$total_trending_days)
```

```
## [1] 0.7152225
```

```
cor(main_yt$total_dislikes, main_yt$total_trending_days)
```

```
## [1] 0.6031573
```

```
cor(main_yt$total_comments, main_yt$total_trending_days)
```

```
## [1] 0.6323473
```

The correlation between video_count and total_trending_days is 0.909, indicating a strong positive linear relationship. This suggests that as the number of videos increases, the total number of days those videos trend on YouTube also tends to increase significantly.

The correlation between total_views and total_trending_days is 0.638, indicating a moderate positive linear relationship. This suggests that there is a tendency for videos with more views to trend for a longer duration

The correlation between total_likes and total_trending_days is 0.715, indicating a moderate positive linear relationship. This suggests that videos with more likes tend to trend for a longer duration on YouTube

The correlation between total_dislikes and total_trending_days is 0.603, indicating a moderate positive linear relationship. This suggests that videos with more dislikes tend to trend for a longer duration

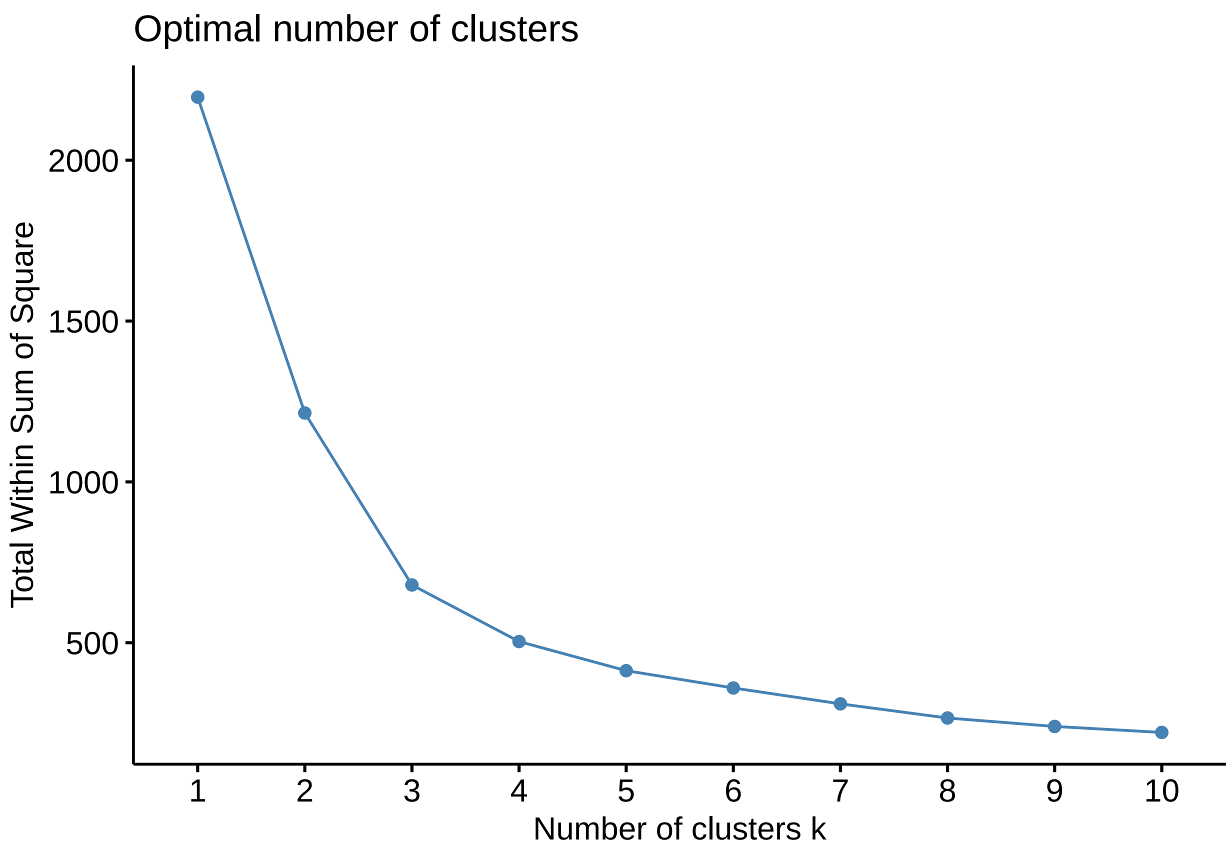
Finally, the correlation between total_comments and total_trending_days is 0.632, indicating a moderate positive linear relationship. This suggests that videos with more comments tend to trend for a longer duration.

#K-Means Clustering

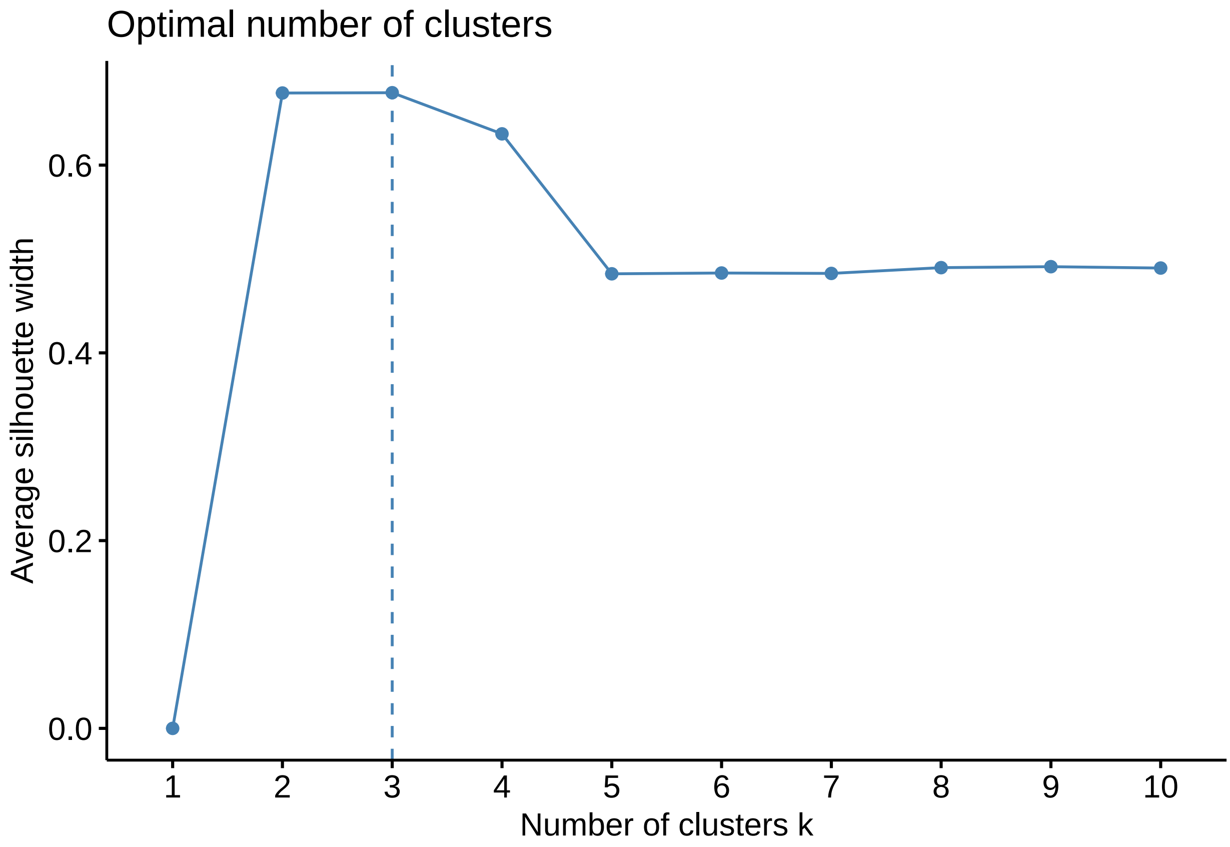
```
main_ytt<-unique(main_yt)
numeric_features2 <- main_ytt[, sapply(main_ytt, is.numeric)]
scaled_features2 <- scale(numeric_features2)
```

In this part, we select only the numeric columns from data. Finally, we scale the numeric features in using z-score normalization.

```
fviz_nbclust(scaled_features2, FUN = hcut, method = "wss")
```



```
fviz_nbclust(scaled_features2, FUN = hcut, method = "silhouette")
```



From the Elbow method and Silhouette method, three clusters were clearly identified, hence we will be selecting three clusters as our optimal clusters.

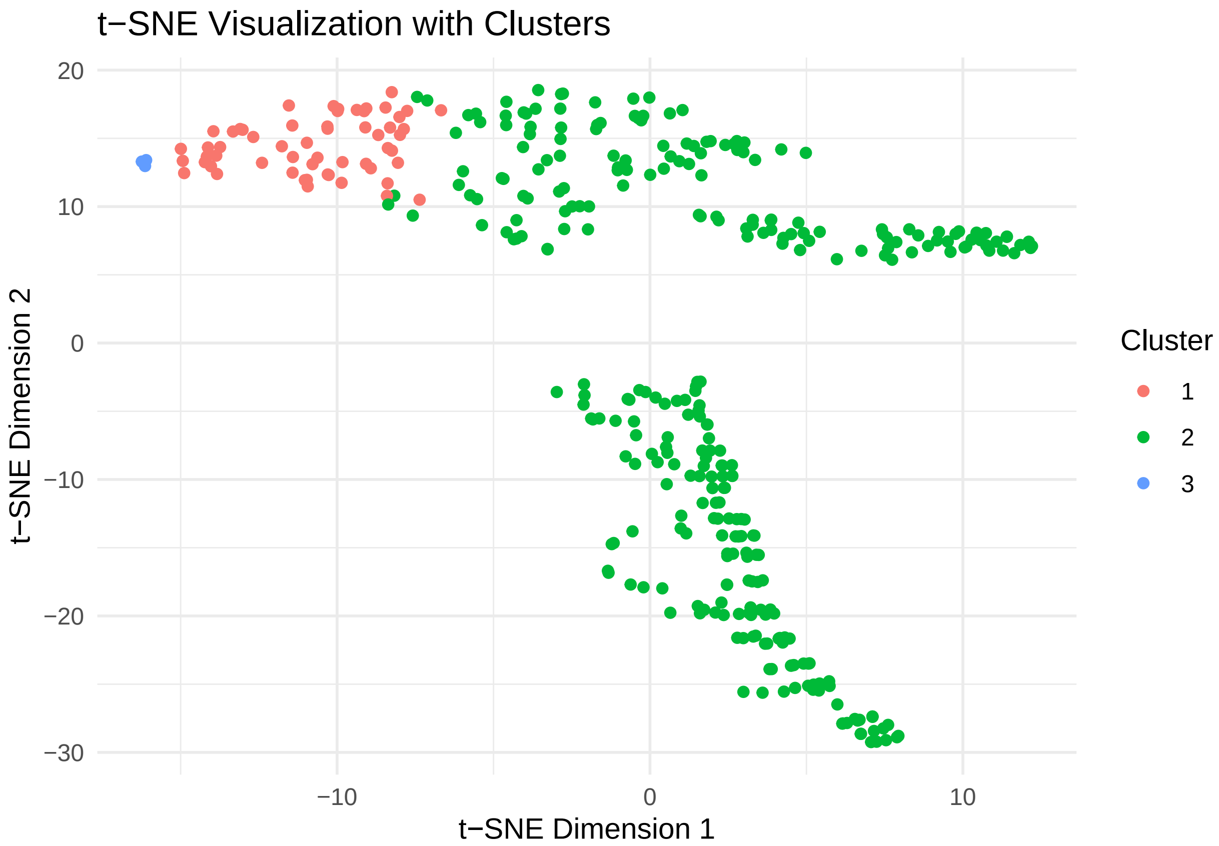
```
optimal_k=3
kmeans_model <- kmeans(scaled_features2, centers = optimal_k)
main_ytt$cluster <- as.factor(kmeans_model$cluster)
```

visualizing the Clustering

```
tsne_result <- Rtsne::Rtsne(as.matrix(scaled_features2), dims = 2, perplexity = 30, theta = 0.5, max_ite
```

```
## Read the 367 x 6 data matrix successfully!
## Using no_dims = 2, perplexity = 30.000000, and theta = 0.500000
## Computing input similarities...
## Building tree...
## Done in 0.01 seconds (sparsity = 0.347096)!
## Learning embedding...
## Iteration 50: error is 51.360079 (50 iterations in 0.02 seconds)
## Iteration 100: error is 49.159781 (50 iterations in 0.02 seconds)
## Iteration 150: error is 48.883582 (50 iterations in 0.02 seconds)
## Iteration 200: error is 48.832599 (50 iterations in 0.02 seconds)
## Iteration 250: error is 48.821894 (50 iterations in 0.02 seconds)
## Iteration 300: error is 0.281522 (50 iterations in 0.02 seconds)
## Iteration 350: error is 0.247289 (50 iterations in 0.02 seconds)
## Iteration 400: error is 0.237471 (50 iterations in 0.02 seconds)
## Iteration 450: error is 0.232158 (50 iterations in 0.02 seconds)
## Iteration 500: error is 0.227993 (50 iterations in 0.02 seconds)
## Iteration 550: error is 0.226175 (50 iterations in 0.02 seconds)
## Iteration 600: error is 0.225519 (50 iterations in 0.02 seconds)
## Iteration 650: error is 0.224608 (50 iterations in 0.02 seconds)
## Iteration 700: error is 0.224116 (50 iterations in 0.02 seconds)
## Iteration 750: error is 0.223683 (50 iterations in 0.02 seconds)
## Iteration 800: error is 0.222960 (50 iterations in 0.02 seconds)
## Iteration 850: error is 0.222796 (50 iterations in 0.02 seconds)
## Iteration 900: error is 0.222224 (50 iterations in 0.02 seconds)
## Iteration 950: error is 0.221900 (50 iterations in 0.02 seconds)
## Iteration 1000: error is 0.220705 (50 iterations in 0.02 seconds)
## Fitting performed in 0.43 seconds.
```

```
ggplot(main_ytt, aes(x = tsne_result$Y[, 1], y = tsne_result$Y[, 2], color = cluster)) +geom_point() +s
    theme_minimal()
```



Here we created a scatter plot using ggplot2, visualizing the t-SNE dimensions as points colored by their assigned cluster labels from the cluster variable in the dataset. We can see that the clusters are well partitions using K-means clustering.

Regression Analysis

```
model <- lm(total_trending_days ~ video_count+total_views +total_likes + total_dislikes + total_comment
summary(model)
```

```
##
## Call:
## lm(formula = total_trending_days ~ video_count + total_views +
##     total_likes + total_dislikes + total_comments + cluster,
##     data = main_ytt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.909  -8.547  -2.790   7.010  83.903
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     1.267e+01  5.481e+00   2.312  0.02137 *
## video_count     1.382e+01  6.242e-01  22.134  < 2e-16 ***
## total_views     2.288e-09  2.411e-09   0.949  0.34318
## total_likes     1.227e-07  1.282e-07   0.957  0.33907
## total_dislikes  1.945e-06  1.752e-06   1.110  0.26769
## total_comments  1.490e-06  2.050e-06   0.727  0.46766
## cluster2       -1.306e+01  4.740e+00  -2.756  0.00615 **
## cluster3       -5.692e+01  2.095e+01  -2.718  0.00689 **
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 18.05 on 359 degrees of freedom
## Multiple R-squared:  0.8696, Adjusted R-squared:  0.8671
## F-statistic: 342.1 on 7 and 359 DF,  p-value: < 2.2e-16
```

```
# Building the regression model with the cluster factor
model_refined <- lm(total_trending_days ~ video_count + cluster, data = main_ytt)

# Displaying the summary of the refined model
summary(model_refined)
```

```
##
## Call:
## lm(formula = total_trending_days ~ video_count + cluster, data = main_ytt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.152  -9.653  -3.653   8.347  87.195
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  31.1073     4.6654   6.668 9.69e-11 ***
## video_count  14.1306     0.5868  24.080  < 2e-16 ***
## cluster2    -30.5853     3.9300  -7.783 7.48e-14 ***
## cluster3     35.8331    11.4635   3.126  0.00192 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 18.95 on 363 degrees of freedom
## Multiple R-squared:  0.8547, Adjusted R-squared:  0.8535
## F-statistic: 711.6 on 3 and 363 DF,  p-value: < 2.2e-16
```

Video Count: For every one-unit increase in video_count, total_trending_days is expected to increase by approximately 13.82 units. Hence the video count plays a major role on trending days.

Total Views, Likes, Dislikes, and Comments: These coefficients are not statistically significant at the 5% level of significance (p > 0.05), suggesting that there is insufficient evidence to conclude that these predictors have a significant linear relationship with total_trending_days.

Clusters 2 and 3: Cluster 2 has a negative coefficient, indicating that it is associated with a decrease in total_trending_days compared to Cluster 1. Similarly, Cluster 3 also has a negative coefficient, suggesting a larger decrease in total_trending_days compared to Cluster 1.

The R-squared value (0.8696) indicates that approximately 86.96% of the variance in total_trending_days is explained by the predictors in the model.

A large F-statistic (342.1) with a very low p-value (< 2.2e-16) suggests that the model as a whole is statistically significant and provides a better fit to the data than a model with no predictors.