Data Visualisation Assignment 2

Mina Jamshidian

Table of contents:

How do YouTube videos get more likes in the three top categories in Europe?

1. Data

- ## Dataset description

The name of the dataset that has been used for this analysis is "Trending Youtube Video Statistics". This is a public dataset from Kaggle which was collected using the YouTube API.
This dataset has several CSV and JSON files with data that shows trending YouTube videos for several countries.
The purpose of this analysis is gathering the data of three big countries from Europe such as Great Britain (GBvideos.csv), Germany (DEvideos.csv) and France (FRvideos.csv) as a sample for Europe.
After gathering the final dataset was called “europe_data”, which has 16 columns with one extra column with the name of “category_name” which was exported from one of the JSON files.
The variables of final dataset “europe_data” are described as follows:

1. video_id
2. trending_date
3. title
4. channel_title
5. category_id
6. publish_time
7. tags
8. views
9. likes
10. dislikes
11. comment_count
12. thumbnail_link
13. comments_disabled
14. ratings_disabled
15. video_error_or_removed
16. description
17. category_name

- ## Loading data:

```
# Loading packages
library(readr)
library(lubridate)
library(stringr)
```

```
library(jsonlite)
library(dplyr)
library(ggplot2)
library(sqldf)
library(gsubfn)
library(proto)
library(RSQLite)
library(gridExtra)
library(RColorBrewer)
library(janeaustenr)
library(tidytext)
```

- Importing Data

```
GB_data<-read_csv('GBvideos.csv')
DE_data<-read_csv('DEvideos.csv')
FR_data<-read_csv('FRvideos.csv')
```

- Wrangling Data

1. Adding the new column: “country_name” to each country dataset(GB_data, DE_data, FR_data) that shows the name of each country. It would be used when we want to merge three datasets to the final dataset.
2. Extracting the category name from JSON file (US_category_id.json) then add it to a data frame (ct_df) with their id in two columns then add it to each country dataset as a new column (category_name) by using the left_join function.
3. Binding all three countries' dataset to the final dataset with the name of “europe_data” by using the rbind function. After that the dataset has around 27881 records.
4. The values of the “title” variable were transformed to the correct format for future analysing.
5. The columns of video_publish_month were added for analysing which was extracted from the publish_time variable.

```
# Adding the "country_name" column to each country dataset
GB_data$country_name<-'Great Britain'
DE_data$country_name<-'Germany'
FR_data$country_name<-'France'

# Extracting Category name from France JSON file and add it as a new variable(column) to
each country dataset
category<-read_json("US_category_id.json")

# Convert it to a dataframe with two columns which are the category id and category name
ct_df<-as.data.frame(NULL)
for(i in 1:32){
  ct_df[i,1]<-category$items[[i]]$id
  ct_df[i,2]<-category$items[[i]]$snippet$title
}

# Naming the column with appropriate name
colnames(ct_df)<-c('category_id','category_name')
```

```
# Adding the category_name column to each dataset by using left_join function
GB_data$category_id<-as.character(GB_data$category_id)
DE_data$category_id<-as.character(DE_data$category_id)
FR_data$category_id<-as.character(FR_data$category_id)

GB_data<-left_join(GB_data,ct_df,by='category_id')
DE_data<-left_join(DE_data,ct_df,by='category_id')
FR_data<-left_join(FR_data,ct_df,by='category_id')

# Binding the data from 3 countries in to Final dataset which is europe_data
europe_data<-rbind(GB_data,DE_data,FR_data)

# Transforming Title
video_title <- europe_data$title
video_title <- tolower(video_title)
video_title <- gsub("\\\\n", " ", video_title)
video_title <- gsub("http[^[:blank:]]+", "", video_title)
video_title <- gsub("www[^[:blank:]]+", "", video_title)
video_title <- gsub('[[:digit:]]+', "", video_title)
video_title <- gsub("[[:punct:]]+", "", video_title)
video_title <- gsub("\\s+"," ", video_title)

# Added as a new column to dataset
europe_data$transform_title<-video_title

# Extracting Just the publish month from the publish
europe_data$video_publish_month<-factor(month(europe_data$publish_time))
```

- Cleaning Data

After Checking for the missing value, the 5084 NA value for the “description” variable was found. Because for this analysing the “description” variable would not be used.it was removed from the dataset.

```
# Checking for NA values(Missing values)
sum(is.na(europe_data))

# Finding the missing values for each column
colSums(is.na(europe_data))

# Removing description column from the final dataset
europe_data<- select(europe_data, -description)
colSums(is.na(europe_data))
```

2. Story

Due to the increasing use of technology by people. People are starting to make money online.
One of the most popular online systems to make money is Youtube channel. Because it is free and working with that is easy for all of the world. So these days creating videos for youtube is a famous online job in the world. The one of the most popular conditions for earning money from there is that each video has a lot of viewers and gets more likes from viewers.
So in this project we want to analyse the likes of videos in Europe because a lot of people analysed the videos views before.

Therefore for this project the goal is searching about videos likes in Europe.
The first purpose of this project is finding three top video categories and working on that.

The figure 1 shows the most popular categories in Europe. The "Entertainment" with 28% is the first popular category, "Music" with 17% is the second one and the "People & Blogs" with 12% is the third one. Therefore these three groups would be analysed for this project.

- Finding three top categories in Europe

```
# Finding Tree Top categories in Europe
top_category <- sqldf("select category_name, count(likes) as likes, country_name from
europe_data group by category_name order by likes")

slices <- top_category$likes
lbls <- top_category$category_name
pct <- round(slices/sum(slices)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(slices,labels = lbls,cex=0.8, col=rainbow(length(lbls)),
    main="Finding Top Youtube Categories By Likes")
```

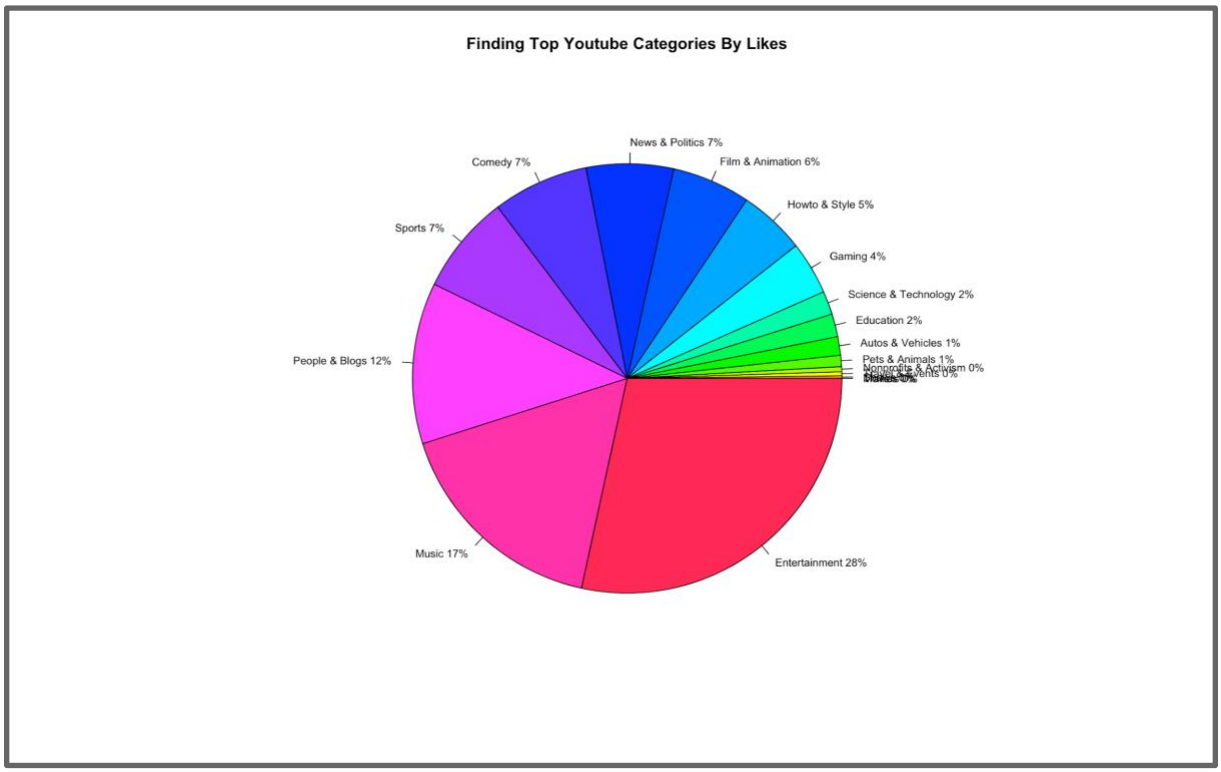

Figure 1

3. Data Visualization

- First visualization

Figure 2 gives information regarding top 10 YouTube channels by likes for three top categories (Entertainment, Music and People & Blogs) in Europe.

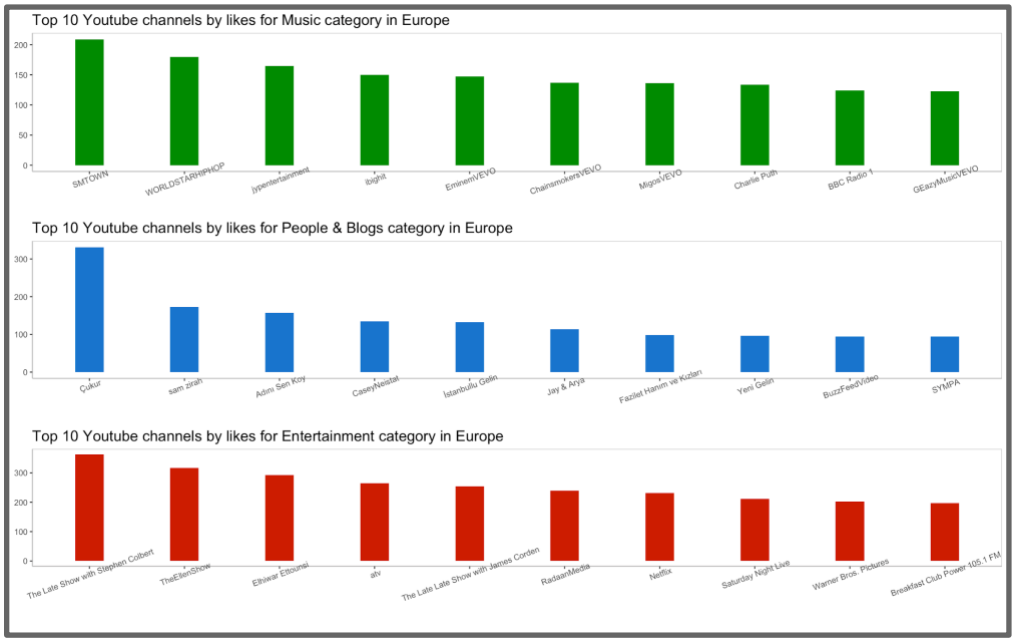

Figure 2

```
# First: Top 10 Youtube channels in Entertainment,Music and People & Blogs  categories by
likes in Europe

# Before
top_ch_m<-sqldf("select channel_title, count(likes) as likes, category_name from
europe_data where category_name='Music' group by channel_title")
top_ch_p<-sqldf("select channel_title, count(likes) as likes, category_name from
europe_data where category_name='People & Blogs' group by channel_title")
top_ch_e<-sqldf("select channel_title, count(likes) as likes, category_name from
europe_data where category_name='Entertainment' group by channel_title")

# Ordering the column with likes for each country
top_ch_m <- top_ch_m[with(top_ch_m, order(-likes)), ]
top_ch_p <- top_ch_p[with(top_ch_p, order(-likes)), ]
top_ch_e <- top_ch_e[with(top_ch_e, order(-likes)), ]

# Selecting top 10 channel fo reach country
top_ch_m10<- top_ch_m[1:10,]
top_ch_p10<- top_ch_p[1:10,]
top_ch_e10<- top_ch_e[1:10,]

# Binding 10 top channel for each country
top10_ch<-rbind(top_ch_m10,top_ch_p10,top_ch_e10)
top10_ch

# Bar Plot for Youtube channel
ggplot(data=top10_ch,aes(x=channel_title,y=likes))+
  labs(title='Top 10 Youtube channels by likes for each Category in Europe')+
  geom_bar(stat = 'identity',aes(fill=category_name))+
  scale_fill_brewer(palette="Set1")+
  theme(
```

```
    plot.title = element_text(size=16),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    axis.text.x = element_text(angle = 90),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )

# After (Final)

# Finding Top 10 youtube channel for each category separately
top10_ch_m =filter(top10_ch,top10_ch$category_name=='Music')
top10_ch_p =filter(top10_ch,top10_ch$category_name=='People & Blogs')
top10_ch_e =filter(top10_ch,top10_ch$category_name=='Entertainment')


m_plot<-ggplot(data =top10_ch_m )+
  labs(title='Top 10 Youtube channels by likes for Music category in Europe')+
  geom_bar(stat = 'identity',fill='green4',width=.3,aes(x=channel_title,y=likes))+
  scale_x_discrete(limits= top10_ch_m$channel_title)+
  theme(
    plot.title = element_text(size=16),
    axis.text.x = element_text(angle = 20),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )

p_plot<-ggplot(data =top10_ch_p)+
  labs(title='Top 10 Youtube channels by likes for People & Blogs category in Europe')+
  geom_bar(stat = 'identity',fill='dodgerblue3',width=.3,aes(x=channel_title,y=likes))+
  scale_x_discrete(limits= top10_ch_p$channel_title)+
  theme(
    plot.title = element_text(size=16),
    axis.text.x = element_text(angle = 20),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )

e_plot<-ggplot(data =top10_ch_e)+
  labs(title='Top 10 Youtube channels by likes for Entertainment category in Europe')+
  geom_bar(stat = 'identity',fill='red3',width=.3,aes(x=channel_title,y=likes))+
  scale_x_discrete(limits= top10_ch_e$channel_title)+
  theme(
    plot.title = element_text(size=16),
    axis.text.x = element_text(angle = 20),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )

# Putting 3 plot in one grid
grid.arrange(m_plot,p_plot,e_plot,nrow=3)
```

- ## Second visualization

  Figure 3 gives information regarding the best month for publishing videos to get the most likes for each three top categories in Europe.
  This visualization shows the videos in “March” and “December” get more likes for all categories.
  In overall, in the winter and spring the videos get more likes. And the Entertainment category gets the most likes.

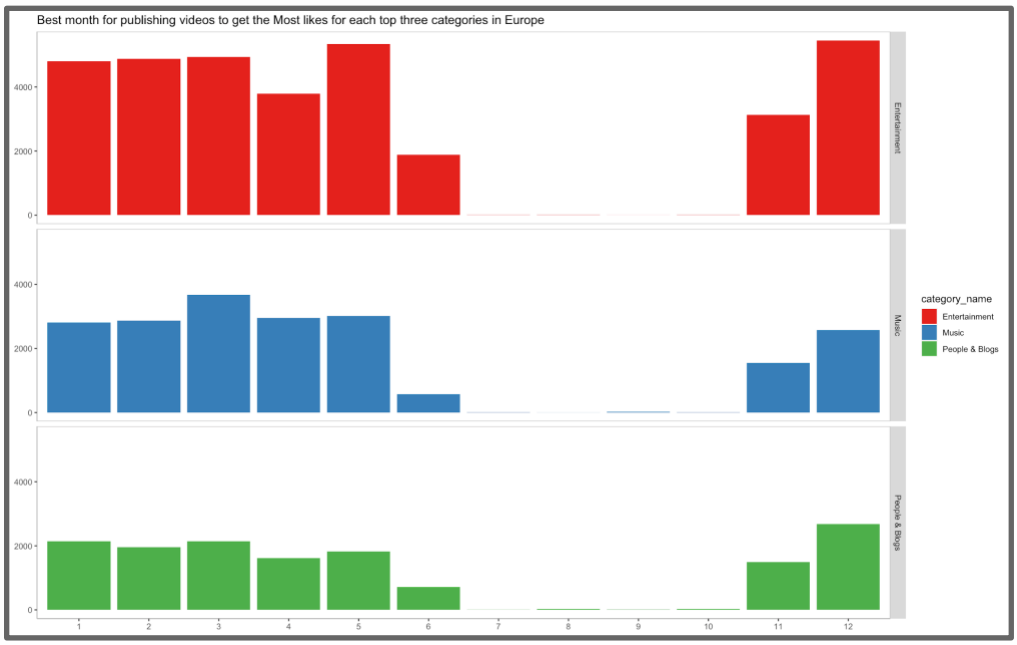

Figure 3

```
# Second: Best month for Publishing Videos to get the Most likes for top three categories
in Europe

publish_month_m<-sqldf("select video_publish_month,count(likes)as likes,category_name from
europe_data where category_name='Music' group by video_publish_month")
publish_month_c<-sqldf("select video_publish_month,count(likes)as likes,category_name from
europe_data where category_name='People & Blogs' group by video_publish_month")
publish_month_e<-sqldf("select video_publish_month,count(likes)as likes,category_name from
europe_data where category_name='Entertainment' group by video_publish_month")

# Binding three category to the new data (publish_month_cat)
publish_month_cat<-rbind(publish_month_m,publish_month_c,publish_month_e)

# Before
ggplot(data=publish_month_cat,aes(x=video_publish_month,y=likes))+
  scale_fill_brewer(palette="Set1")+
  labs(title='Best month for publishing videos to get the Most likes for top three
categories in Europe')+
  geom_bar(stat = 'identity',aes(fill=category_name))+
  theme(
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )


# After (Final)
ggplot(data=publish_month_cat,aes(x=video_publish_month,y=likes))+
```

```
  scale_fill_brewer(palette="Set1")+
  labs(title='Best month for publishing videos to get the Most likes for each top three
categories in Europe')+
  geom_bar(stat = 'identity',aes(fill=category_name))+
  facet_grid(category_name~.)+
  theme(
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
    )
```

- Third visualization

Figure 3 gives information regarding top 10 title bigram by likes for top three categories in Europe.
This diagram shows the title bigram for the Music categories is more important than other categories and the “official video” and “music video” title bigram leads to videos getting more like in this category. After the Music category, the bigram title in the Entertainment category lead more like by starting the word “official trailer”.
But the bigram title for the People & Blogs category is not that much important compared to Music and Entertainment categories.

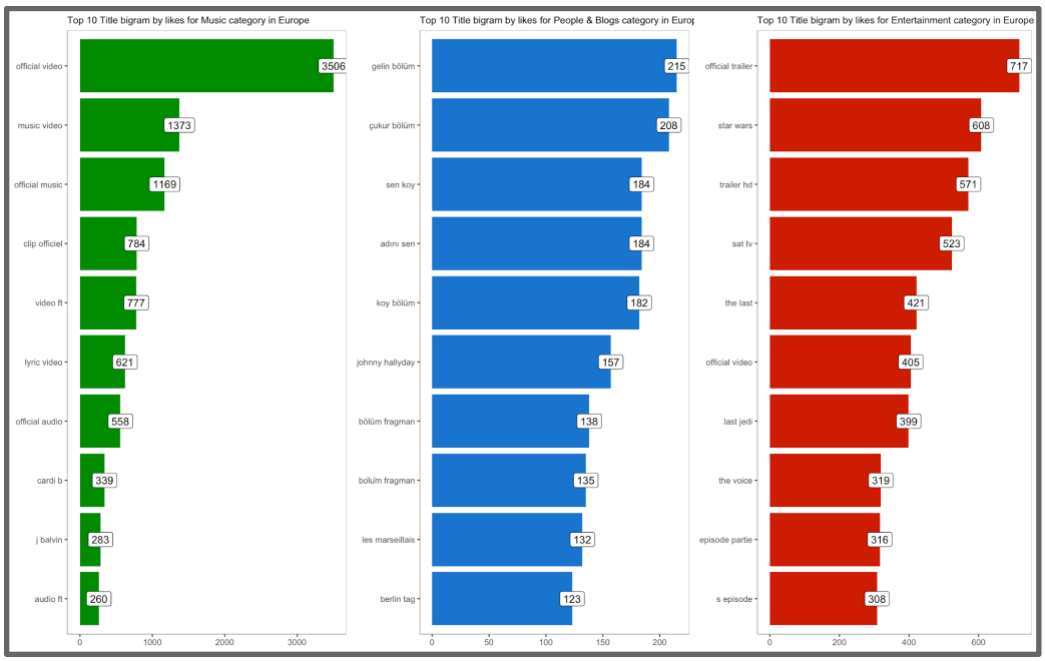

Figure 4

```
# Third: Top 10 Bigrams in Title for top three categories in Europe

# Before
europe_data_mpe<-
filter(europe_data,europe_data$category_name=="Music"|europe_data$category_name=="People &
Blogs"|europe_data$category_name=="Entertainment")
bigram_title<-unnest_tokens(europe_data_mpe, bigram, transform_title, token = "ngrams", n =
2)
bigram_df<-data.table(bigram_title)
top_bigram<-bigram_df%>%select(bigram)%>%group_by(bigram)%>%count()%>%arrange(desc(n))
%>%head(10)

ggplot(top_bigram, aes(x = reorder(bigram, n),y = n)) +geom_bar(stat="identity")+
  labs(title="Top 20 Title bigram by likes for three top categories in Europe")+
  guides(fill="none")+xlab(NULL)+ylab(NULL)+geom_label(aes(label = n))+coord_flip()+
  theme(
    plot.title = element_text(size=16),
    axis.text.x = element_text(angle = 20),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )

# After (Final)
europe_data_m<- europe_data_mpe %>%filter(category_name=="Music")
europe_data_p<- europe_data_mpe %>%filter(category_name=="People & Blogs")
europe_data_e<- europe_data_mpe %>%filter(category_name=="Entertainment")

bigram_title_m<-unnest_tokens(europe_data_m, bigram, transform_title, token = "ngrams", n =
2)
bigram_title_p<-unnest_tokens(europe_data_p, bigram, transform_title, token = "ngrams", n =
2)
bigram_title_e<-unnest_tokens(europe_data_e, bigram, transform_title, token = "ngrams", n =
2)

bigram_df_m<-data.table(bigram_title_m)
bigram_df_p<-data.table(bigram_title_p)
bigram_df_e<-data.table(bigram_title_e)

top_bigram_m <- bigram_df_m %>% select(bigram)%>%
group_by(bigram)%>%count()%>%arrange(desc(n))%>%head(10)
top_bigram_p <- bigram_df_p %>% select(bigram)%>%
group_by(bigram)%>%count()%>%arrange(desc(n))%>%head(10)
top_bigram_e <- bigram_df_e %>% select(bigram)%>%
group_by(bigram)%>%count()%>%arrange(desc(n))%>%head(10)


m<-ggplot(top_bigram_m, aes(x = reorder(bigram, n),y = n))
+geom_bar(stat="identity",fill='green4')+
  labs(title="Top 10 Title bigram by likes for Music category in Europe")+
  guides(fill="none")+xlab(NULL)+ylab(NULL)+geom_label(aes(label = n))+coord_flip()+
  theme(
    plot.title = element_text(size=10),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )
p<-ggplot(top_bigram_p, aes(x = reorder(bigram, n),y =
n))+geom_bar(stat="identity",fill='dodgerblue3')+
  labs(title="Top 10 Title bigram by likes for People & Blogs category in Europe")+
  guides(fill="none")+xlab(NULL)+ylab(NULL)+geom_label(aes(label = n))+coord_flip()+
  theme(
    plot.title = element_text(size=10),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
```

```
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )
e<-ggplot(top_bigram_e, aes(x = reorder(bigram, n),y = n)) +
  geom_bar(stat="identity",fill='red3')+
  labs(title="Top 10 Title bigram by likes for Entertainment category in Europe")+
  guides(fill="none")+xlab(NULL)+ylab(NULL)+geom_label(aes(label = n))+coord_flip()+
  theme(
    plot.title = element_text(size=10),
    axis.title.y = element_blank(),
    axis.title.x = element_blank(),
    panel.background = element_rect(fill = 'white', colour = 'gray'),
    panel.border = element_blank(),
    panel.grid.major = element_blank(),
    panel.grid.minor = element_blank(),
    axis.line = element_line(colour = "gray")
  )

grid.arrange(m,p,e,ncol=3)
```

4. Iterations

Iteration 1:

Before:

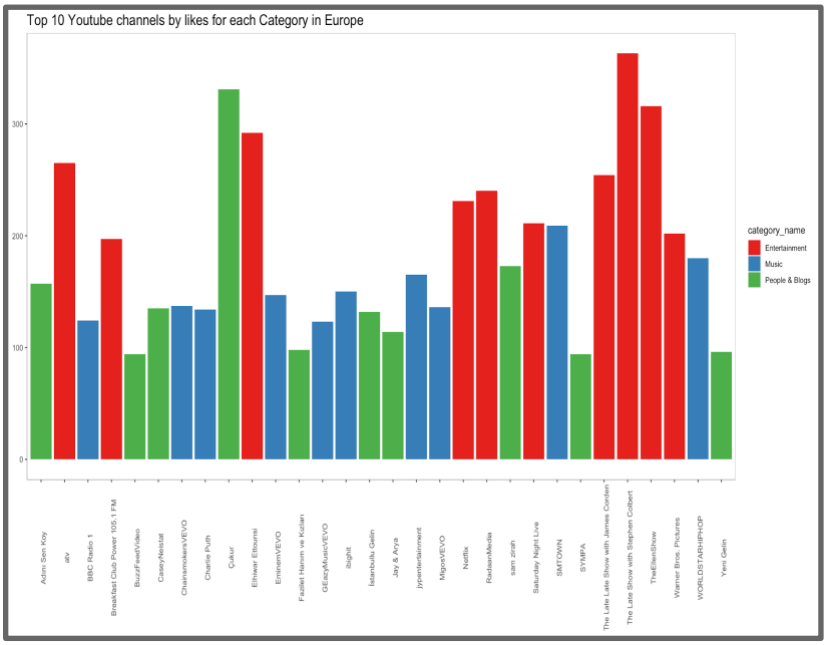

Figure 5

After:

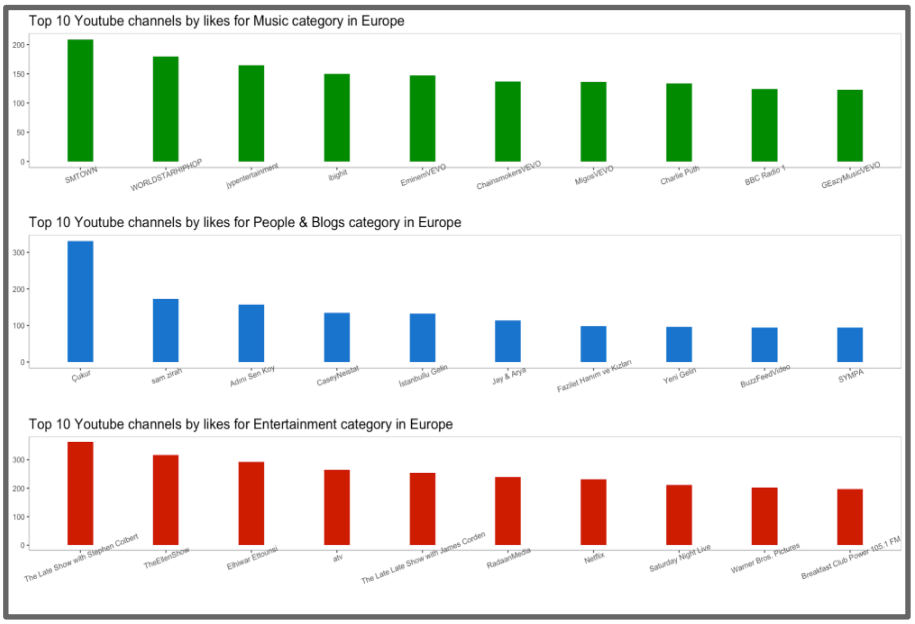

Figure 6

For finding the top youtube channel for each category is too difficult as it is obvious in Figure 5.
For removing this problem it separated for each group as Figure 6 shows.

Iteration 2:

Before:

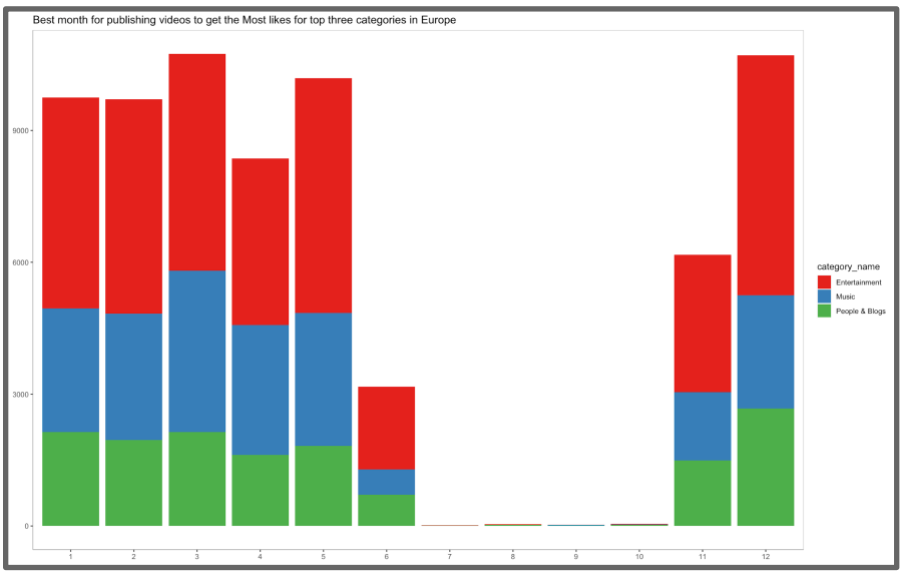

Figure 7

After:

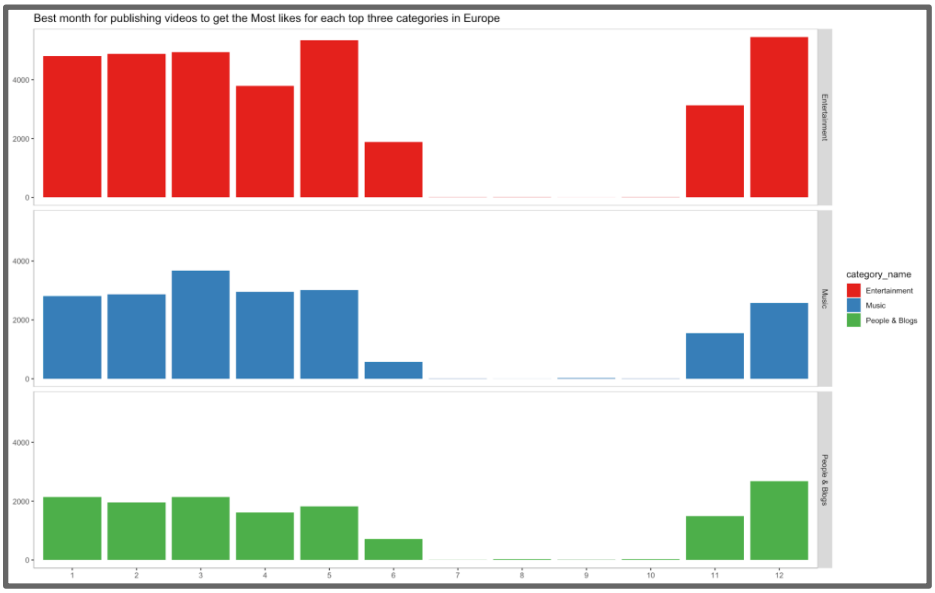

Figure 8

Finding the best publish month for each category to get more likes is too difficult as it is obvious in Figure 7 because we couldn't understand perfectly which category in which months has the highest likes number.
To treat this problem ,the categories would be separated for each group as it is obvious in Figure 8.
For example in the figure 7 shows “December” and “March” have the highest likes but in the Entertainment category in Figure 8 shows after “December” ,the “April” has the highest like.

Iteration 3:

Before:

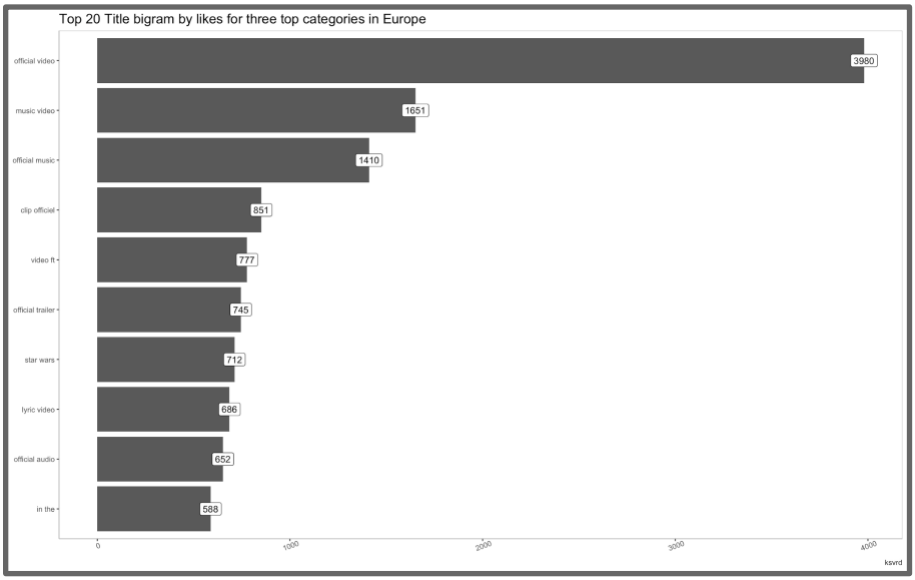

Figure 9

After:

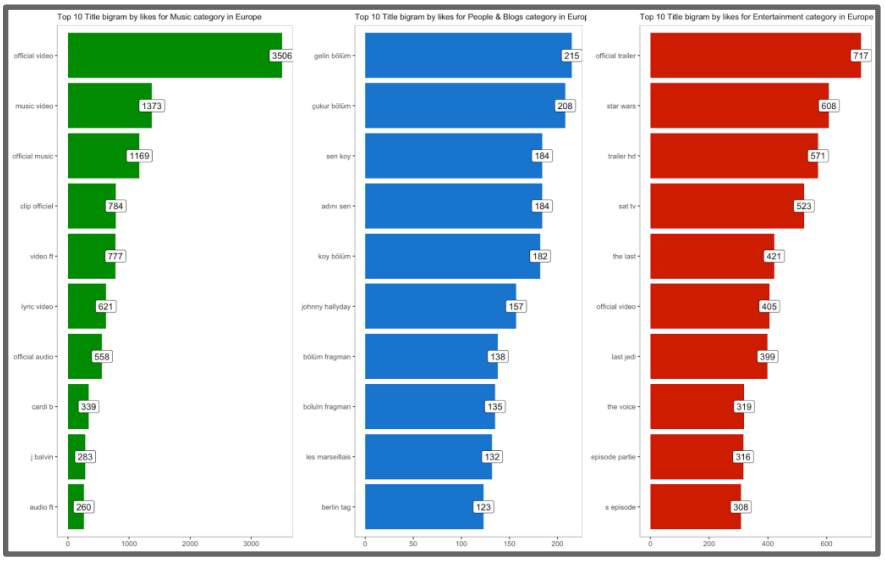

Figure 10

The Figure 9 shows the top bigram titles but it doesn't show these bigram titles are useful for which categories and make the likes for videos in each category. Therefore it was divided into each category to show which bigram title is useful or important for each category.