DAT 560M Final Project

Group D5
Sicheng Liu 474278; Zehao Lyu 474275; Dominique Nie 474148; Lynna Xie 474887

1 - Description of the data

- About YouTube project
  - We are going to analyze YouTube dataset in this report. YouTube is a well-known American video-sharing platform. It allows users to upload, view, rate, share, report, comment on videos and subscribe to other users. Therefore, we are interested in getting to know more about most frequent words of description, popular categories, number of total likes among different countries, and sentiment analysis. By applying the knowledge of dealing and analyzing big data we learnt in class into our project, we mainly use Cloudera and Python to deal with the YouTube dataset, then present our result through Tableau.
- About YouTube dataset
  - Data collected from the (up to) 200 listed trending YouTube videos every day in 10 countries for 8 months after November 14th in 2017.
  - Mitchell collected this data. The total size of the data is 516 MB.
  - The link to data sources is https://github.com/mitchelljy/Trending-YouTube-Scraper.
  - Detailed information on each CSV:

File Name	Records	Size
CAvideos.csv	40882	62567 KB
FRvideos.csv	45091	50220 KB
JPvideos.csv	20524	28068 KB
MXvideos.csv	40452	44133 KB
USvideos.csv	40950	62542 KB
DEvideos.csv	40841	61563 KB
GBvideos.csv	38917	51967 KB
KRvideos.csv	34568	34020 KB
RUvideos.csv	40740	74481 KB
INvideos.csv	37353	58204 KB

  - There are 17 feature columns
    - video_id, trending_date, title, channel_title, category_id, publish_time, tags, views, likes, dislikes, comment_count, thumbnail_link, comments_disabled, ratings_disabled, video_error_or_removed, description

2 - Problem Statement

1. Take US as an example, which tags people use most frequently when they upload videos?

    - We use “tags” columns and separate all tags. Then we group the data and use it to make WordClouds and analyze result.
2. What is the trend of the five most popular categories in each country?
    - In this problem, our team first chose 5 categories that we are interested in. These 5 categories are 10 (music), 17 (sports), 20 (gaming), 24 (entertainment), and 27 (education). We mainly discuss and visualize the trends of these 5 categories in the ten countries, and we will focus more on the three English-speaking countries, Britain, Canada and the United States of America.
3. What is the difference among total likes for the same video_ids across various countries?
    - There are 9 same videos watched by all countries, 4 videos in category 10, 3 videos in category 24, 1 video each in categories 1 (film and animation) and 23 (comedy). We focus on analyzing the videos in category 24 as the results above show that it is the most popular category among 9 countries except in Britain.
4. What are the most frequent keywords in video descriptions for the three English-speaking countries (US, GB, CA)?
    - We process “description” column of each country in Python by deleting all the URLs following the video introduction and visualize the processed description words of the three English-speaking countries (US, GB, CA) via WordClouds.
5. How do total sentiment scores for the ten countries differ from each other?
    - We mainly use “tm” and “syuzhet” libraries in R. After several transformations of special characters, removal of numbers, stopwords, punctuation, and stripping of white space on the corpus, we manage to calculate the total sentiment score for each country.

3 - Why is this a big data?

Generally speaking, if we can’t analyze the data collected in time using traditional processes or tools, it is big data. Big data has three features: Volume, Velocity, Variety:

- **Volume:** Volume refers to the amount of data. The YouTube file totally has 380318 records. This dataset was collected from YouTube for nearly 6 months. Every day, 1 billion users of YouTube contribute information about videos, even though this data only collect up to 200 listed trending YouTube videos, it is definitely a huge amount.
- **Velocity:** Velocity refers to the speed of data processing. Every day 0.4 million hours of video are uploaded on YouTube. The flow of data is massive and continuous.
- **Variety:** Variety in Big Data refers to all the structured and unstructured data that has the possibility of getting generated either by humans or by machines. YouTube data generated from websites, information like views, likes, and dislikes are from a huge number of videos; comments, tags, and descriptions are from humans… Information on data comes from different forms and finally stored in several CSV files.

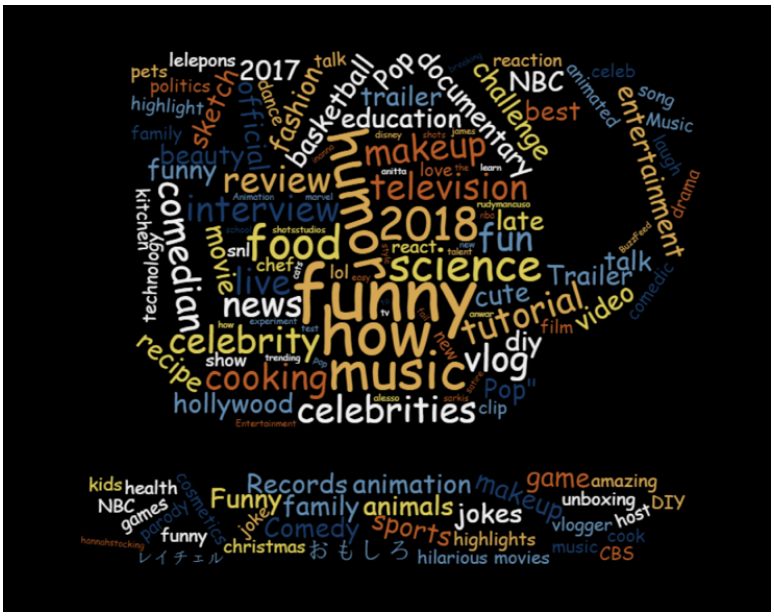
4 - Method & Results

1. Data cleaning
    - We cleaned the data using python. For example, the form of the trending_date column is not a standard form of date and time. Therefore, we created a function called change_date in Python. After we have applied the function to the column,

we then apply the to_datetime function built in pandas to change its type from string to date. This way the visualization step can be facilitated. Codes are shown in the appendix at the end.

2. Data loading and basic information
    - We input datasets into Cloudera and find some basic information:
        - Get data records and columns.
        - The dataset of US videos contains records of 206 days in 2018.
        - There are some words appear more frequently than others in “tag” column, like” funny”, “comedy”, “humor”, “music”.
3. Data organizing, preparing for analysis
    a. Categories
        - Our team use both Impala and python make sure that we get the same dataset. Both software shows same result. Codes are shown in appendix.

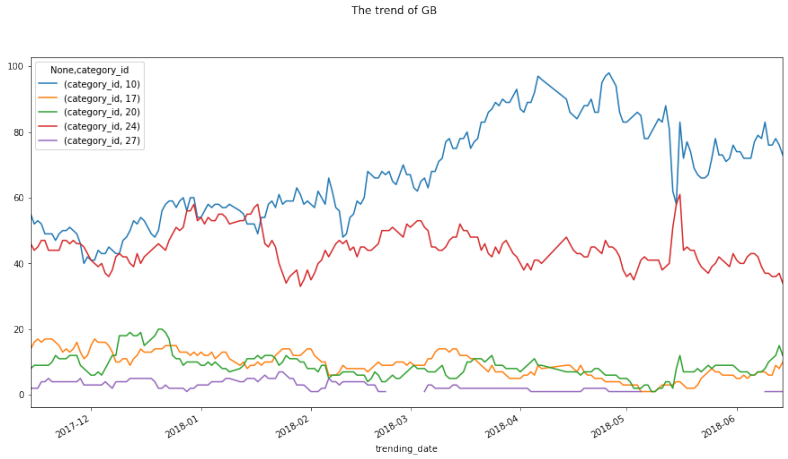
    b. Same videos
        - We first get the information of same video ids that appear in all countries by using for loop in Python. We get to know that there are 9 same video ids shown across 10 countries, then we write 10 new csv files which only include the information of those video ids. Next, we process our data using terminal in Cloudera, such as removing header and uploading the files into HDFS. Codes are shown in appendix. Then, we continue with our analysis mainly using Impala. After all things ready, we search for the data we want, sum of likes by grouping the video ids.
4. Data Visualization
    a. Tags
        - We use the sorted results of the tag to make a WordClouds image. From this image, we could find that “funny”, “how”, “music”, “humor”, “food”, “2018” are the most popular tag words. We also find an interesting thing that some Japanese words occur frequently like “おもしろ”(Omoshiro), which means “interesting”; and words “レイチェル”(Reicheru), who is the heroine of the famous cartoon “Angel of Death”.



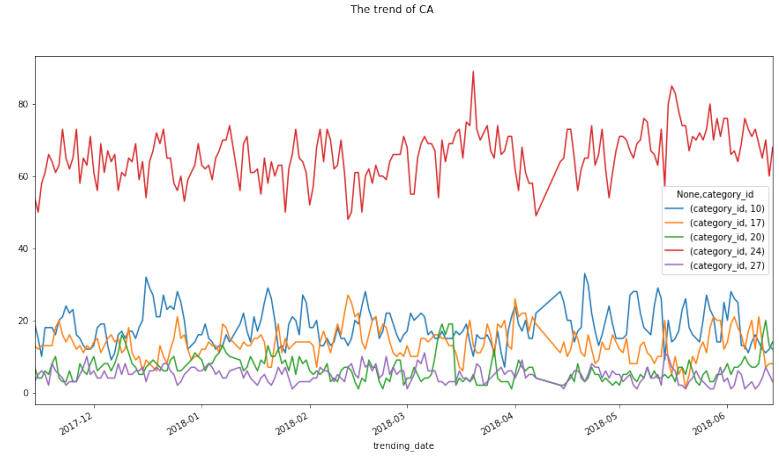
b. Categories

- After we retrieved the desired data, we can now have a more intuitive understanding of the data. Here is an overview of the five chosen categories in all 10 countries. We can see that for most countries, the most popular category is category 24, which is entertainment. However, in Britain, the most popular category is category 10, which is music. We can also see that on a certain date, the game category surpassed entertainment category in Japan, Korea and Russia. This suggests that there was a worldwide gaming event that day.

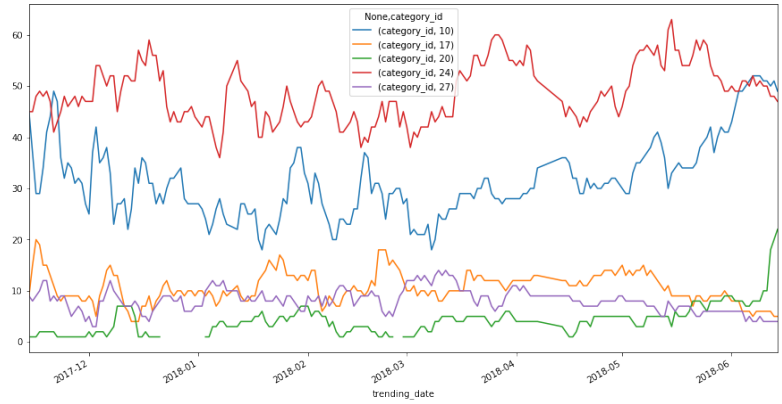
- Below are the graphs for Britain, Canada and the USA. In Britain, we can see that the British people really like music. The music category dominates the entertaining category most of the time. We can also see that the British people do not have as much interest in the other three categories as in music and entertainment.



- In Canada, the entertainment category dominates all other categories, and other categories are mostly the same. The number of popular videos of the remaining categories normally range from 0 to 20.



- In the USA, we can see that like most of the other countries, the music category has a very obvious upward trend after March 2018. Also, the gaming category soars after June 2018.

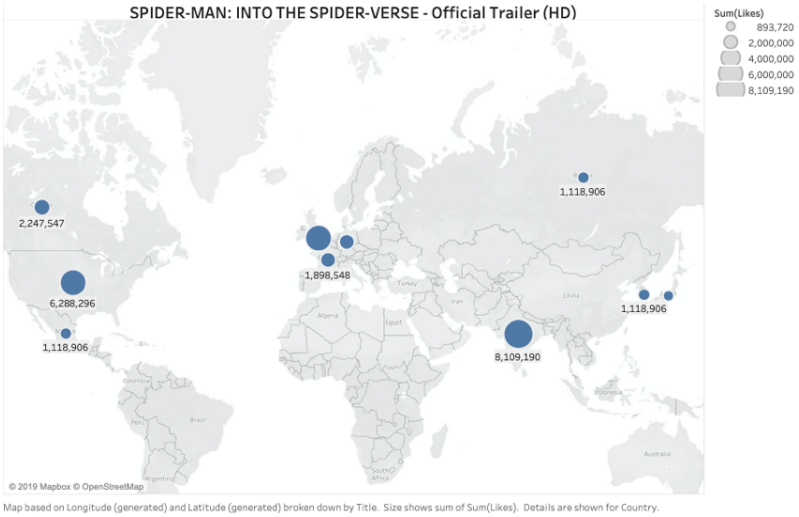


c. Total likes of same videos watched across 10 different countries

- Spider-man receives the greatest number of likes in India, which is 8,109,190, followed by the US and Canada. Looking at all other points in the graph above,

the total amount of likes is over 1 million, which means that the movie is popular around the world.

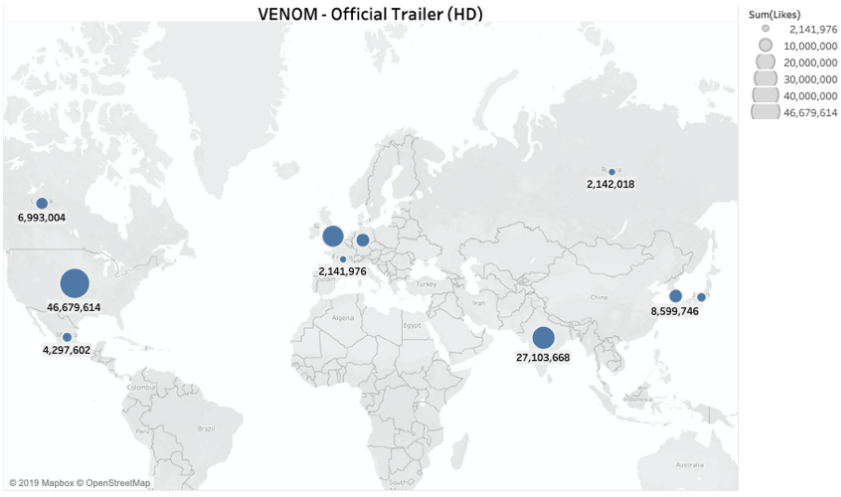
Sheet 1



Map based on Longitude (generated) and Latitude (generated) broken down by Title. Size shows sum of Sum(Likes). Details are shown for Country.

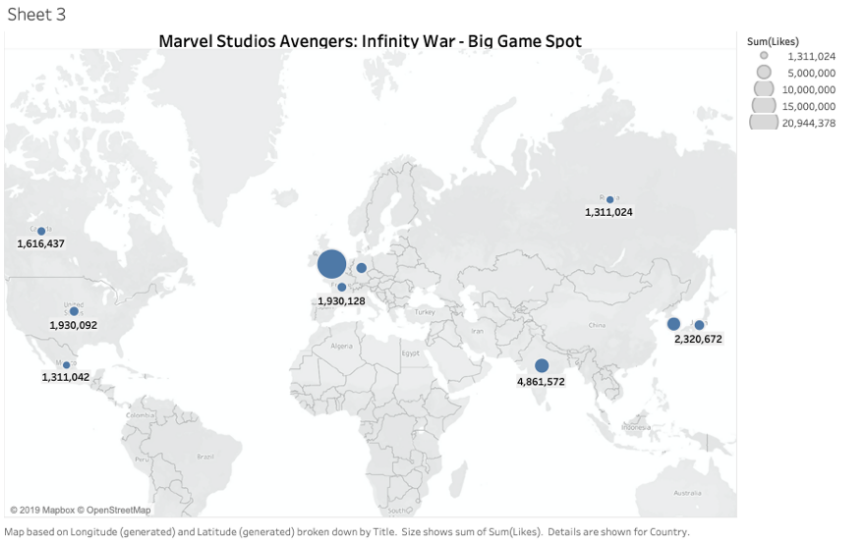
- By looking through the numbers, it clearly indicates that Venom is much more popular than the other two movies. Each country's total likes is over 2 million. The US has the highest amount of total likes, 46,689,614, followed by India with 27,103,668. France has the lowest amount of total likes.

Sheet 2



Map based on Longitude (generated) and Latitude (generated) broken down by Title. Size shows sum of Sum(Likes). Details are shown for Country.

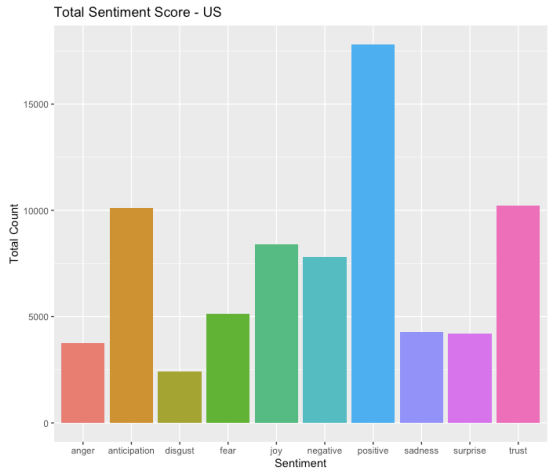
- Marvel Studios Avengers are most popular in England, which receives the highest amount of likes there. India, Japan and Korea share similar number of likes. The graph shows all countries have the number of likes over 1 million.

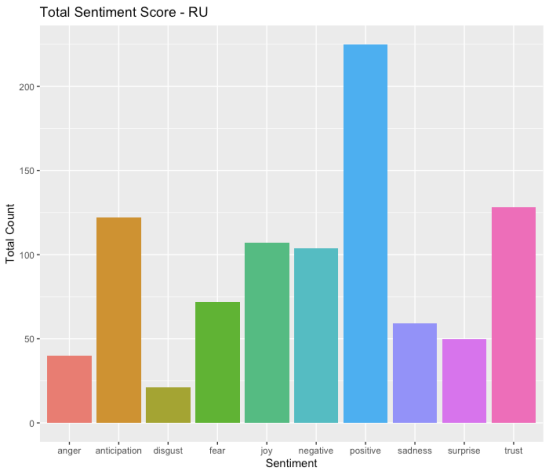


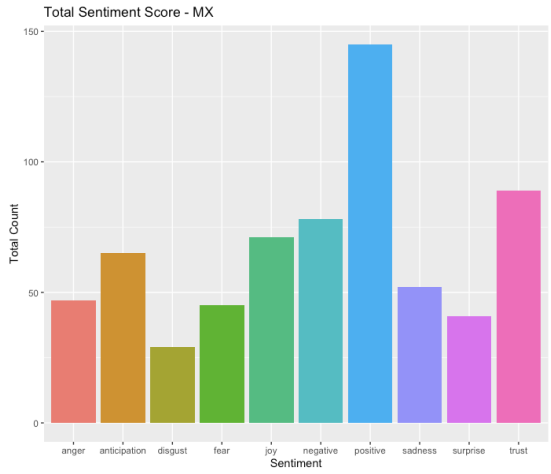
d. Description
   - From the graphs we can observe that the most frequent keywords that appear in the introduction are “DAY & LIFE”, “LATE & CAN”, “NEW & FREE” for US, GB, CA respectively.

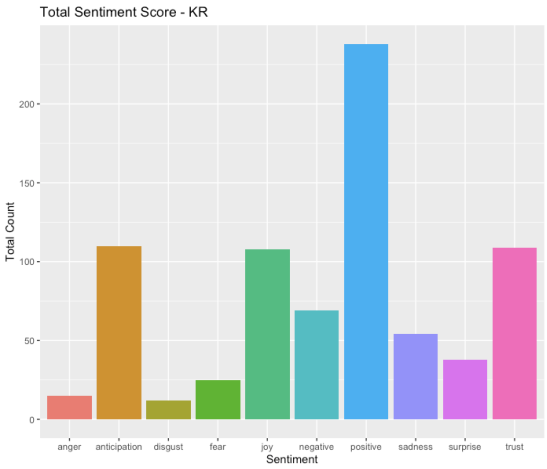
e. Sentiment Analysis
   - Sentiment analysis is a way to approach the emotional content of text programmatically and quantify the feeling or tone of written text, in which each case receives a numeric sentiment score (on a negative to positive scale). Here we utilize the processed video description and extract the first 10,000 lines of text for each country. From the graphs below, we can see

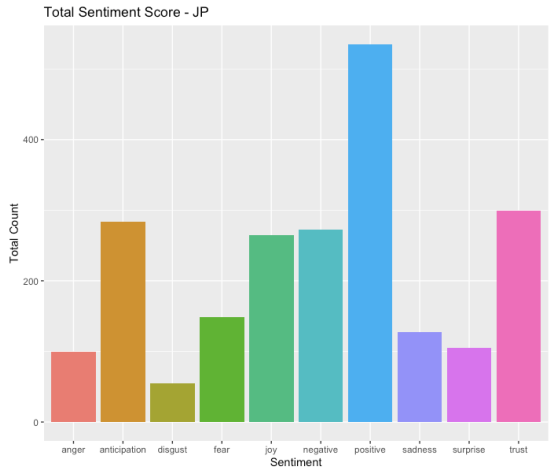
that overall, US, IN and GB achieve the highest total sentiment scores in all categories of emotions, while RU, MX and KR rank the lowest. The sentiment score distributions are similar in all the ten countries, with positive emotion scoring at the top, while disgust emotion scoring at the bottom. Notably, in MX and JP, their sentiment scores of “negative” are higher than those of “joy”, which means that video descriptions with negative emotion are more appealing to the audience than those with joy emotion.

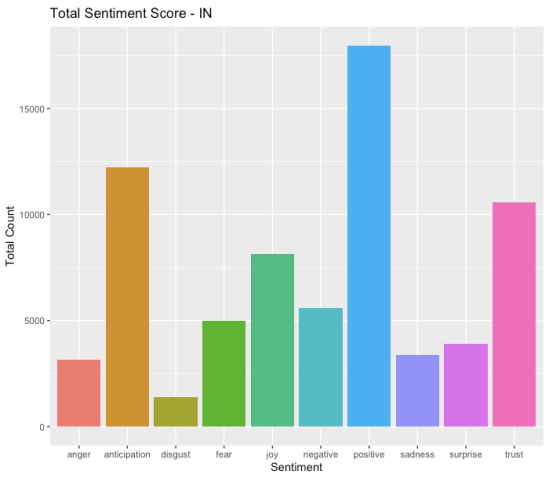


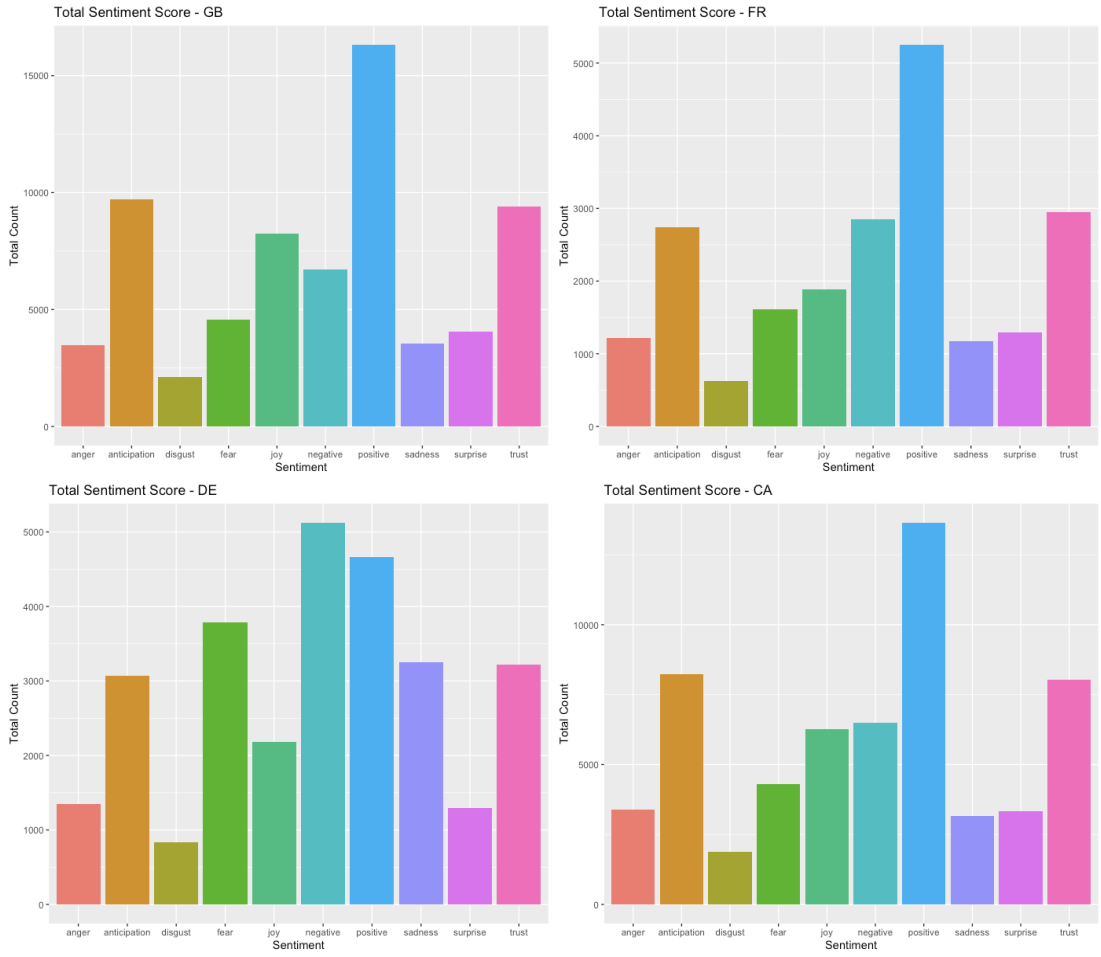












5 - Conclusion

- Answers of questions
  1. When uploading videos, people prefer to use tags: funny, humor, 2018, music, science, food as their video tags. We also find that people in the US are also very interested in Japanese content.
  2. Entertainment videos are the most popular category worldwide. In Britain, the most popular category is music. If there is a worldwide gaming event, the gaming category will surpass the entertainment category that day.
  3. Relatively, India receives most of the total likes in all movies as there are relatively much more people in the country. Then YouTube can focus on uploading entertainments related videos for Indian to receive higher number of likes. While Venom is significantly popular in US, then promoting similar movies in US would be a smart decision.
  4. The most frequently used words in description are “DAY & LIFE” for US, “LATE & CAN” for GB, and “NEW & FREE” for CA.
  5. In terms of scale, US, IN and GB have the highest total sentiment scores, while RU, MX and KR are among the lowest. In terms of sentiment score distribution, all the ten countries are similar, with “positive” emotion scoring highest, and “disgust” emotion scoring lowest. We are surprised to find that in MX and JP, sentiment scores of “negative” are higher than those of “joy”, from which we can infer that video descriptions with negative emotion can attract more audience than those with joy emotion.
- Suggestions

    - From the perspective of the YouTube platform: When YouTube is advertising, it can be comprehensively considered according to different national conditions. Cultural differences can cause viewers to react differently to different ads.
    - From the perspective of users: From the result, we could find that most people prefer to watch entertainment videos. However, YouTube is also a good platform to learn new knowledge and improve your study and career.
- Further improvements
    - From the perspective of the dataset: YouTube dataset only has 17 columns. If we could have a much larger dataset such as contain contents of comments, peak of video viewing, fans of youtuber, and number of times the video was shared… We could find more interesting results by analyzing the data.
    - From the perspective of methods: We have used methods including Impala, Python, R, Cloudera and Tableau to do data cleaning, data organization, data analysis and data visualization. However, because of variable limitation, we could not do some further research like using statistical models.

Codes

Data Cleaning

```
def change_date(x):
    new_list=x.split(".")
    year="20"+new_list[0]
    month=new_list[2]
    day=new_list[1]
    return year+"-"+month+"-"+day
```

Data loading and basic information

```
[cloudera@quickstart Desktop]$ wc -l INvideos.csv
37353 INvideos.csv
[cloudera@quickstart Desktop]$ head -1 USvideos.csv
video_id,trending_date,title,channel_title,category_id,publish_time,tags,views,l
ikes,dislikes,comment_count,thumbnail_link,comments_disabled,ratings_disabled,vi
deo_error_or_removed,description
```

```
[cloudera@quickstart Desktop]$ cut -d "," -f 2 USvideos.csv | sort | uniq -c | s
ort -n | wc -l
206
```

For Problem 1

```
[cloudera@quickstart Desktop]$ cut -d "," -f 7 < USvideos.csv| tr '|' '\n' | sor
t | uniq -c | sort -n -r | head -n 20
   3300 ""funny""
   2686 ""comedy""
   1588 ""30 CD 8F""
   1507 [none]
   1445 ""how to""
   1134 ""30 AC 30 A4 30 C9""
   1119 ""humor""
   1101 ""music""
   1061 ""2018""
   1055 ""food""
    991 ""science""
    892 ""news""
    889 ""vlog""
    887 ""makeup""
    864 ""review""
    826 ""tutorial""
    824 ""celebrity""
    809 ""live""
    786 ""comedian""
    758 ""interview""
```

For Problem 2

Impala:

```sql
SELECT trending_date,category_id,count(category_id)
FROM us_videos
WHERE category_id = 10 or category_id = 17 or category_id = 20 or category_id=24 or category_id=27
GROUP BY trending_date,category_id
ORDER BY trending_date,category_id;
```

Query History | Saved Queries | Results (1,011)

	trending_date	category_id	count(category_id)
1	17.01.12	10	25
2	17.01.12	17	9
3	17.01.12	20	2
4	17.01.12	24	47
5	17.01.12	27	5
6	17.02.12	10	37
7	17.02.12	17	8
8	17.02.12	20	1
9	17.02.12	24	47
10	17.02.12	27	3
11	17.03.12	10	42
12	17.03.12	17	5
13	17.03.12	20	2
14	17.03.12	24	54
15	17.03.12	27	3
16	17.04.12	10	35
17	17.04.12	17	9
18	17.04.12	20	2
19	17.04.12	24	54
20	17.04.12	27	8
21	17.05.12	10	36
22	17.05.12	17	11
23	17.05.12	20	2
24	17.05.12	24	52
25	17.05.12	27	8

Python:

In [1]:
```python
import pandas as pd
from matplotlib import pyplot as plt
```

In [34]:
```python
def change_date(x):
    new_list=x.split(".")
    year="20"+new_list[0]
    month=new_list[2]
    day=new_list[1]
    return year+"-"+month+"-"+day

def category_plot(nation,preferred_list):
    df=pd.read_csv(nation+"videos.csv",encoding="ISO-8859-1")
    df["trending_date"]=df["trending_date"].apply(change_date)
    df["trending_date"]=pd.to_datetime(df["trending_date"])
    df_category_count=df[df["category_id"].isin(preferred_list)].groupby(["trending_date","category_id"])["category_id"].count()
    df_category_count=pd.DataFrame(df_category_count)
    df_category_count.rename(columns={"category_id":"count"})
    fig, ax = plt.subplots(figsize=(15,8))
    fig.suptitle("The trend of %s"%nation)
    df_category_count.unstack().plot(ax=ax)

category_plot("US",[10,17,20,24,27])
```

In [26]:
```
def category_overall_count(nation,preferred_list):
    df=pd.read_csv(nation+"videos.csv",encoding="ISO-8859-1")
    df["trending_date"]=df["trending_date"].apply(change_date)
    df["trending_date"]=pd.to_datetime(df["trending_date"])
    df_category_count=df[df["category_id"].isin(preferred_list)].groupby(["trending_date","category_id"])["category_id"].count()
    df_category_count=pd.DataFrame(df_category_count)
    df_category_count.rename(columns={"category_id":"count"})
    return df_category_count

fig, ((ax0,ax1),(ax2,ax3),(ax4,ax5),(ax6,ax7),(ax8,ax9))=plt.subplots(nrows=5,ncols=2,figsize=(30,16),sharex=True)
list_preference=[10,17,20,24,27]
df1=category_overall_count("CA",list_preference)
df1.unstack().plot(ax=ax0)
ax0.set(title="Canada",xlabel="")
df2=category_overall_count("DE",list_preference)
df2.unstack().plot(ax=ax1)
ax1.set(title="Germany",xlabel="")
df3=category_overall_count("FR",list_preference)
df3.unstack().plot(ax=ax2)
ax2.set(title="France",xlabel="")
df4=category_overall_count("GB",list_preference)
df4.unstack().plot(ax=ax3)
ax3.set(title="Britain",xlabel="")
df5=category_overall_count("IN",list_preference)
df5.unstack().plot(ax=ax4)
ax4.set(title="India",xlabel="")
df6=category_overall_count("JP",list_preference)
df6.unstack().plot(ax=ax5)
ax5.set(title="Japan",xlabel="")
df7=category_overall_count("KR",list_preference)
df7.unstack().plot(ax=ax6)
ax6.set(title="Korea",xlabel="")
df8=category_overall_count("MX",list_preference)
df8.unstack().plot(ax=ax7)
ax7.set(title="Mexico",xlabel="")
df9=category_overall_count("RU",list_preference)
df9.unstack().plot(ax=ax8)
ax8.set(title="Russia")
df10=category_overall_count("US",list_preference)
df10.unstack().plot(ax=ax9)
ax9.set(title="America")
```

In [33]:
```
df=pd.read_csv("USvideos.csv",encoding="ISO-8859-1")
df["trending_date"]=df["trending_date"].apply(change_date)
df["trending_date"]=pd.to_datetime(df["trending_date"])
df_category_count=df[df["category_id"].isin(list_preference)].groupby(["trending_date","category_id"])["category_id"].count()
df_category_count=pd.DataFrame(df_category_count)
df_category_count.rename(columns={"category_id":"count"})
```

Out[33]:

		count
trending_date	category_id	
2017-11-14	10	45
	17	9
	20	1
	24	45
	27	9
2017-11-15	10	37
	17	15
	20	1
	24	45
	27	8
2017-11-16	10	29
	17	20
	20	1
	24	48
	27	9
2017-11-17	10	29
	17	19
	20	2
	24	49
	27	10

Cloudera:

```
CREATE TABLE US_videos (video_id STRING, trending_date STRING, title STRING, channel_title STRING,
                        category_id INT, publish_time STRING, tags STRING, views INT, likes INT, dislikes INT,
                        comment_count INT, thumbnail STRING, comments_disabled STRING, ratings_video STRING,
                        video_error_or_removed STRING, description STRING) ROW FORMAT DELIMITED FIELDS TERMINATED BY ',' STORED AS TEXTFILE;
```

```
LOAD DATA INPATH '/user/cloudera/US_videos.csv' INTO TABLE CA_videos;
```

```
SELECT trending_date,category_id,count(category_id)
FROM us_videos
WHERE category_id = 10 or category_id = 17 or category_id = 20 or category_id=24 or category_id=27
GROUP BY trending_date,category_id
ORDER BY trending_date,category_id;
```

For Problem 3

Python:

In [1]:
```
import pandas as pd
import numpy as np

df_CA=pd.read_csv("CAvideos.csv",encoding="ISO-8859-1")
df_GB=pd.read_csv("GBvideos.csv",encoding="ISO-8859-1")
df_US=pd.read_csv("USvideos.csv",encoding="ISO-8859-1")
df_DE=pd.read_csv("DEvideos.csv",encoding="ISO-8859-1")
df_FR=pd.read_csv("FRvideos.csv",encoding="ISO-8859-1")
df_IN=pd.read_csv("INvideos.csv",encoding="ISO-8859-1")
df_JP=pd.read_csv("JPvideos.csv",encoding="ISO-8859-1")
df_KR=pd.read_csv("KRvideos.csv",encoding="ISO-8859-1")
df_MX=pd.read_csv("MXvideos.csv",encoding="ISO-8859-1")
df_RU=pd.read_csv("RUvideos.csv",encoding="ISO-8859-1")
```

In [2]:
```
CA_video_id=df_CA.video_id
CA_video_id_unique=np.unique(CA_video_id)
GB_video_id=df_GB.video_id
GB_video_id_unique=np.unique(GB_video_id)
US_video_id=df_US.video_id
US_video_id_unique=np.unique(US_video_id)
DE_video_id=df_DE.video_id
DE_video_id_unique=np.unique(DE_video_id)
FR_video_id=df_FR.video_id
FR_video_id_unique=np.unique(FR_video_id)
IN_video_id=df_IN.video_id
IN_video_id_unique=np.unique(IN_video_id)
JP_video_id=df_JP.video_id
JP_video_id_unique=np.unique(JP_video_id)
KR_video_id=df_KR.video_id
KR_video_id_unique=np.unique(KR_video_id)
MX_video_id=df_MX.video_id
MX_video_id_unique=np.unique(MX_video_id)
RU_video_id=df_RU.video_id
RU_video_id_unique=np.unique(RU_video_id)
```

In [3]:
```
## Compare CA_video_id with GB_video_id
same_id_list=[]
for i in CA_video_id_unique:
    for j in GB_video_id_unique:
        if i==j:
            same_id_list.append(i)

same_id_list=np.unique(same_id_list)


## Compare same video_ids in CA_video_id and GB_video_id with US_video_id
same_id_list1=[]
for i in same_id_list:
    for j in US_video_id_unique:
        if i==j:
            same_id_list1.append(i)
same_id_list1=np.unique(same_id_list1)


## Compare same video_ids in CA_video_id, GB_video_id and US_video_id with DE_video_id
same_id_list2=[]
for i in same_id_list1:
    for j in DE_video_id_unique:
        if i==j:
            same_id_list2.append(i)
same_id_list2=np.unique(same_id_list2)


## Compare same video_ids in CA_video_id, GB_video_id, US_video_id and DE_video_id with FR_video_id
same_id_list3=[]
for i in same_id_list2:
    for j in FR_video_id_unique:
        if i==j:
            same_id_list3.append(i)
same_id_list3=np.unique(same_id_list3)
```

```
## Compare same video_ids in CA_video_id, GB_video_id, US_video_id, DE_video_id and FR_video_id with IN_video_id
same_id_list4=[]
for i in same_id_list3:
    for j in IN_video_id_unique:
        if i==j:
            same_id_list4.append(i)
same_id_list4=np.unique(same_id_list4)


## Compare same video_ids in CA_video_id, GB_video_id, US_video_id, DE_video_id, FR_video_id and IN_video_id \
## with JP_video_id
same_id_list5=[]
for i in same_id_list4:
    for j in JP_video_id_unique:
        if i==j:
            same_id_list5.append(i)
same_id_list5=np.unique(same_id_list5)


## Compare same video_ids in CA_video_id, GB_video_id, US_video_id, DE_video_id, FR_video_id, IN_video_id \
## and JP_video_id with KR_video_id
same_id_list6=[]
for i in same_id_list5:
    for j in KR_video_id_unique:
        if i==j:
            same_id_list6.append(i)
same_id_list6=np.unique(same_id_list6)


## Compare same video_ids in CA_video_id, GB_video_id, US_video_id, DE_video_id, FR_video_id, IN_video_id, \
## JP_video_id and KR_video_id \
## with MX_video_id
same_id_list7=[]
for i in same_id_list6:
    for j in MX_video_id_unique:
        if i==j:
            same_id_list7.append(i)
same_id_list7=np.unique(same_id_list7)
```

```
## Compare same video_ids in CA_video_id, GB_video_id, US_video_id, DE_video_id, FR_video_id, IN_video_id, \
## JP_video_id, KR_video_id and MX_video_id with RU_video_id
same_id_list8=[]
for i in same_id_list7:
    for j in RU_video_id_unique:
        if i==j:
            same_id_list8.append(i)
same_id_list8=np.unique(same_id_list8)
same_id_list8
```

Out[3]:
```
array(['DkeiKbqa02g', 'OrnpSe4OChM', 'XiHiW4N7-bo', 'g4Hbz2jLxvQ',
       'iWZmdoY1aTE', 'pVxOVlm_lE8', 'tCXGJQYZ9JA', 'tjA7nAHOAww',
       'u9Mv98Gr5pY'], dtype='<U11')
```

In [4]:
```
for i in ["CA","DE","GB","FR","IN","KR","JP","MX","RU","US"]:
    df=pd.read_csv(i+"videos.csv",encoding="ISO-8859-1")
    df=df[df["video_id"].isin(same_id_list8)]
    df.to_csv(i+"popular",index=False)
```

Cloudera:
sed 1d CApopular.csv > CApopular1.csv
sed 1d DEpopular.csv > DEpopular1.csv
sed 1d FRpopular.csv > FRpopular1.csv
sed 1d GBpopular.csv > GBpopular1.csv
sed 1d INpopular.csv > INpopular1.csv
sed 1d JPpopular.csv > JPpopular1.csv
sed 1d KRpopular.csv > KRpopular1.csv
sed 1d MXpopular.csv > MXpopular1.csv
sed 1d RUpopular.csv > RUpopular1.csv
sed 1d USpopular.csv > USpopular1.csv

hdfs dfs -put CApopular1.csv
hdfs dfs -put DEpopular1.csv
hdfs dfs -put FRpopular1.csv
hdfs dfs -put GBpopular1.csv
hdfs dfs -put INpopular1.csv
hdfs dfs -put JPpopular1.csv
hdfs dfs -put KRpopular1.csv
hdfs dfs -put MXpopular1.csv
hdfs dfs -put RUpopular1.csv
hdfs dfs -put USpopular1.csv

create table ca_popular (video_id string, trending_date string, title string,
channel_title string, category_id int, publish_time string, tags string, views int,
likes int, dislikes int, comment_count int, thumbnail string, comments_disabled string,
ratings_video string, video_error_or_removed string, description string) row format delimited
fields terminated by ',' stored as textfile;
…(repeated for 9 other countries)

select title, category_id, sum(likes) from ca_popular where category_id=24 group by
title,category_id;
…(repeated for 9 other countries)

For Problem 4

Python:

In [1]:
```
import pandas as pd
def Process_Data(filename):
    df = pd.read_csv(filename, header = None)
    for index, row in df.iterrows():
        head, sep, tail = str(row[0]).partition('http')
        row[0] = head
    df.to_csv(filename, index = False)
```

In [2]:
```
Process_Data("Youtube/CAvideos_desc.csv")
Process_Data("Youtube/USvideos_desc.csv")
Process_Data("Youtube/DEvideos_desc.csv")
Process_Data("Youtube/FRvideos_desc.csv")
Process_Data("Youtube/GBvideos_desc.csv")
Process_Data("Youtube/INvideos_desc.csv")
Process_Data("Youtube/RUvideos_desc.csv")
Process_Data("Youtube/JPvideos_desc.csv")
Process_Data("Youtube/KRvideos_desc.csv")
Process_Data("Youtube/MXvideos_desc.csv")
```

For Problem 5 (take the US as example)

R:

```r
rm(list = ls())
library(ggplot2)
library(lubridate)
library(scales)
library(reshape2)
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(stringr)
library(syuzhet)
library(dplyr)

#get the data
text <- readLines("/Users/niecong/Documents/WUSTL/DAT_560M/Youtube/USvideos_desc.csv",
                  n = 10000)

#convert the file to UTF-8 encoding
s2 <- iconv(text, "UTF-8", "ASCII", sub = "")
text <- s2

#let us create the corpus
docs <- Corpus(VectorSource(text))

#clean our chat data
trans <- content_transformer(function (x , pattern ) gsub(pattern, " ", x))
docs <- tm_map(docs, trans, "/")
docs <- tm_map(docs, trans, "@")
docs <- tm_map(docs, trans, "\\|")
docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, removeNumbers)
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, stripWhitespace)
docs <- tm_map(docs, stemDocument)

#create the document term matrix
dtm <- TermDocumentMatrix(docs)
mat <- as.matrix(dtm)
v <- sort(rowSums(mat),decreasing=TRUE)

#data frame
data <- data.frame(word = names(v),freq=v)

#generate the wordcloud
set.seed(1056)
wordcloud(words = data$word, freq = data$freq, min.freq = 1,
          max.words=200, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))

#fetch sentiment words from texts
Sentiment <- get_nrc_sentiment(text)
```

```r
head(Sentiment)
text <- cbind(text,Sentiment)

#count the sentiment words by category
TotalSentiment <- data.frame(colSums(text[,c(2:11)]))
names(TotalSentiment) <- "count"
TotalSentiment <- cbind("sentiment" = rownames(TotalSentiment), TotalSentiment)
rownames(TotalSentiment) <- NULL

#total sentiment score of all texts
ggplot(data = TotalSentiment, aes(x = sentiment, y = count)) +
    geom_bar(aes(fill = sentiment), stat = "identity") +
    theme(legend.position = "none") +
    xlab("Sentiment") + ylab("Total Count") + ggtitle("Total Sentiment Score - US")
```