CIS5200 Term Project Tutorial

Authors: Snehil Sarkar; Sapan Shah; Sai Sridhar Karri; Kaushik N Adhikari;

Instructor: Jongwook Woo

Date: 12/13/2023

Lab Tutorial

Snehil Sarkar(ssarkar4@calstatela.edu)

Sapan Shah(sshah82@calstatela.edu)

Sai Sridhar Karri(skarri2@calstatela.edu)

Kaushik N Adhikari(kadhika3@calstatela.edu)

12/13/2023

YouTube Trending Videos Dataset Analysis

Objectives:

YouTube Trending Videos dataset includes several years (and counting) of data on daily trending YouTube videos (2020-2023. Data is included for India, USA, Great Britain, Germany, Canada, France, Russia, Brazil, Mexico, South Korea, and Japan, respectively. Each country's data is in a separate file. In this tutorial,

- We have analyzed the whole dataset & conclude the top-10 trending videos worldwide.
- Figured out the most trending video of individual countries based on the engagement score.
- Conducted year-on-year analysis to find the most viewed channel in the USA and identify content preferences and marketing opportunities.
- Extracted the most frequent tags used while uploading videos to understand the prevalent themes or topics in YouTube videos in India.
- Analyzed the most viewed category of videos in all countries and identified the public interest every year.

Introduction:

In this tutorial using the dataset, you'll learn how to use HADOOP CLUSTER to:

- Download and upload CSV files to HDFS.
- Create Hive tables in HDFS.
- Create Hive queries to analyze data.
- Use Tableau to visualize the analyzed data.
- Use Excel 3D Map for 3D visualization.

Platform Spec:

- CLUSTER VERSION: Hadoop 3.1.2 CPU Speed
- CLUSTER NODES: 5 (2 master nodes, 3 worker nodes)
- MEMORY SIZE: Memory used 480.15 GB and Memory Remaining 54.29GB
- CPU SPEED: 1995.312 MHz

To Get the cluster details, execute the below commands:

- To know the CLUSTER VERSION: hdfs version

```
Last login: Tue Dec 12 19:40:51 2023 from 172.56.233.166
-bash-4.2$ hdfs version
Hadoop 3.1.2
```

- To know the CLUSTER NODES: yarn node –list –all

```
-bash-4.2$ yarn node -list -all
23/12/13 00:36:07 INFO client.RMProxy: Connecting to ResourceManager at bigdaimn0.sub03291929060.trainingvcn.oraclevcn.com/10.1.0.180:8050
23/12/13 00:36:07 INFO client.AHSProxy: Connecting to Application History server at bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com/10.1.0.59:10200
Total Nodes:3
         Node-Id             Node-State Node-Http-Address       Number-of-Running-Containers
bigdaiwn2.sub03291929060.trainingvcn.oraclevcn.com:45454               RUNNING bigdaiwn2.sub03291929060.trainingvcn.oraclevcn.com:8042
bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:45454               RUNNING bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:8042
bigdaiwn1.sub03291929060.trainingvcn.oraclevcn.com:45454               RUNNING bigdaiwn1.sub03291929060.trainingvcn.oraclevcn.com:8042
-bash-4.2$
```

- To know the MEMORY SIZE: hdfs dfsadmin -report

```
-bash-4.2$ hdfs dfsadmin -report
Configured Capacity: 575990573877 (536.43 GB)
Present Capacity: 573846456636 (534.44 GB)
DFS Remaining: 58293031133 (54.29 GB)
DFS Used: 515553425503 (480.15 GB)
DFS Used%: 89.84%
Replicated Blocks:
```

- To know the CPU SPEED: lscpu | grep ‘MHz”

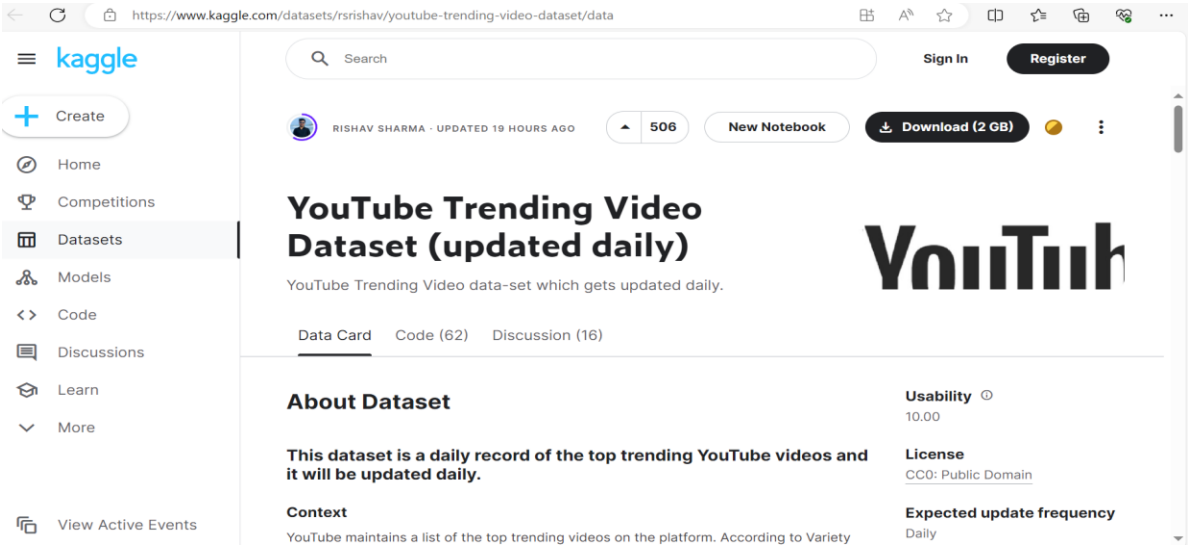
```
-bash-4.2$ lscpu | grep 'MHz'
CPU MHz:               1995.312
```

Dataset Details:

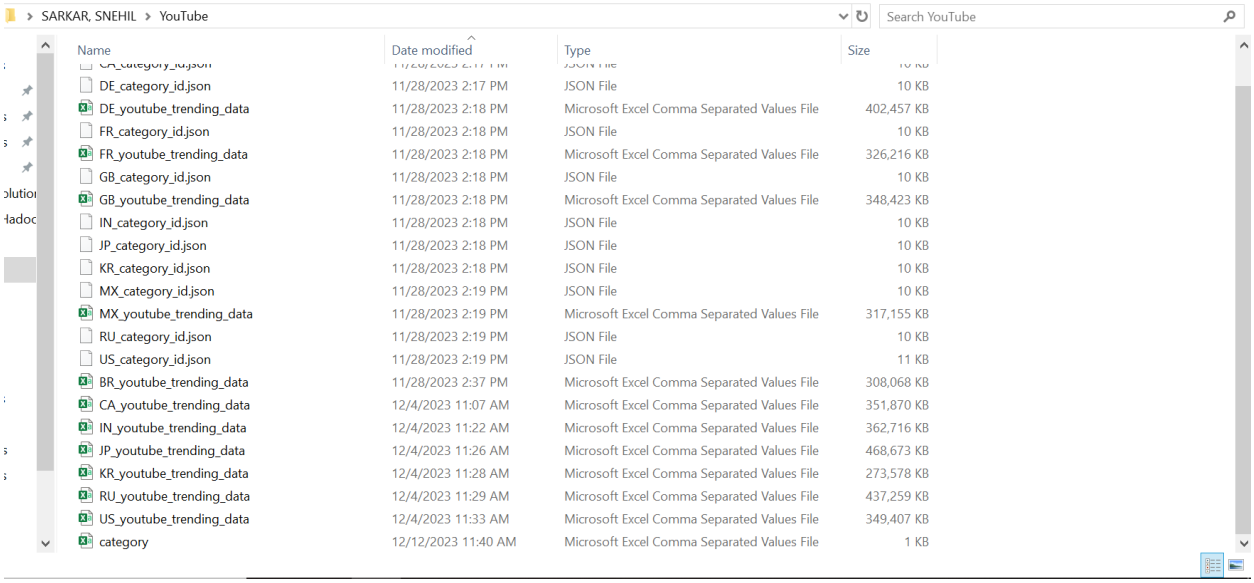
1. Dataset Name: YouTube Trending Videos Dataset
2. Dataset Link: [YouTube Trending Video Dataset (updated daily) (kaggle.com)]
3. Total Size of File: 3.76 GB
4. Size Of File Used: 2.43 GB
5. Format: CSV
6. Countries Used: India, USA, Canada, Russia, Brazil, South Korea, and Japan.

Step 1: Download Dataset:

1. Open the link [YouTube Trending Video Dataset (updated daily) (kaggle.com)](https://www.kaggle.com/datasets/rsrishav/youtube-trending-video-dataset/data) & download the dataset from Kaggle.



2. Unzip the downloaded file and give the folder name as YouTube.



SARKAR, SNEHIL › YouTube

Search YouTube

Name	Date modified	Type	Size
DE_category_id.json	11/28/2023 2:17 PM	JSON File	10 KB
DE_youtube_trending_data	11/28/2023 2:18 PM	Microsoft Excel Comma Separated Values File	402,457 KB
FR_category_id.json	11/28/2023 2:18 PM	JSON File	10 KB
FR_youtube_trending_data	11/28/2023 2:18 PM	Microsoft Excel Comma Separated Values File	326,216 KB
GB_category_id.json	11/28/2023 2:18 PM	JSON File	10 KB
GB_youtube_trending_data	11/28/2023 2:18 PM	Microsoft Excel Comma Separated Values File	348,423 KB
IN_category_id.json	11/28/2023 2:18 PM	JSON File	10 KB
JP_category_id.json	11/28/2023 2:18 PM	JSON File	10 KB
KR_category_id.json	11/28/2023 2:18 PM	JSON File	10 KB
MX_category_id.json	11/28/2023 2:19 PM	JSON File	10 KB
MX_youtube_trending_data	11/28/2023 2:19 PM	Microsoft Excel Comma Separated Values File	317,155 KB
RU_category_id.json	11/28/2023 2:19 PM	JSON File	10 KB
US_category_id.json	11/28/2023 2:19 PM	JSON File	11 KB
BR_youtube_trending_data	11/28/2023 2:37 PM	Microsoft Excel Comma Separated Values File	308,068 KB
CA_youtube_trending_data	12/4/2023 11:07 AM	Microsoft Excel Comma Separated Values File	351,870 KB
IN_youtube_trending_data	12/4/2023 11:22 AM	Microsoft Excel Comma Separated Values File	362,716 KB
JP_youtube_trending_data	12/4/2023 11:26 AM	Microsoft Excel Comma Separated Values File	468,673 KB
KR_youtube_trending_data	12/4/2023 11:28 AM	Microsoft Excel Comma Separated Values File	273,578 KB
RU_youtube_trending_data	12/4/2023 11:29 AM	Microsoft Excel Comma Separated Values File	437,259 KB
US_youtube_trending_data	12/4/2023 11:33 AM	Microsoft Excel Comma Separated Values File	349,407 KB
category	12/12/2023 11:40 AM	Microsoft Excel Comma Separated Values File	1 KB

3. Now, you must transfer the seven countries (India, USA, Canada, Russia, Brazil, South Korea, and Japan) .csv and the category .csv file from the local machine to HADOOP Cluster, which is shown in next section.

Step 2: Upload Files to Hadoop File system (HDFS)

1. Upload the file BR_youtube_trending_data.csv to Linux server using the below command.

```
scp C:/Users/ssarkar4/YouTube/BR_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4
```

```
AD+ssarkar4@STU-PF2Y9ZX7 MINGW64 ~
$ scp C:/Users/ssarkar4/YouTube/BR_youtube_trending_data.csv ssarkar4@129.153.66.218:/home/ssarkar4
ssarkar4@129.153.66.218's password:
BR_youtube_trending_data.csv                                    100%  301MB   1.3MB/s   03:53
```

2. Similarly, upload the files for remaining six countries and the category.csv using the below code:

```
scp C:/Users/ssarkar4/YouTube/CA_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4

scp C:/Users/ssarkar4/YouTube/IN_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4

scp C:/Users/ssarkar4/YouTube/JP_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4

scp C:/Users/ssarkar4/YouTube/KR_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4

scp C:/Users/ssarkar4/YouTube/US_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4

scp C:/Users/ssarkar4/YouTube/RU_youtube_trending_data.csv
ssarkar4@129.153.66.218:/home/ssarkar4

scp C:/Users/ssarkar4/YouTube/category.csv ssarkar4@129.153.66.218:/home/ssarkar4
```

3. Now, create a main directory YouTube as shown below.

```
drwxr-xr-x   - ssarkar4 hdfs          0 2023-11-30 02:35 tmp
-bash-4.2$ hdfs dfs -mkdir YouTube
```

4. Create subdirectories within YouTube for all the seven countries and category as shown below,

```
-bash-4.2$ hdfs dfs -ls YouTube/CA
-bash-4.2$ hdfs dfs -ls YouTube/IN
-bash-4.2$ hdfs dfs -ls YouTube/BR
-bash-4.2$ hdfs dfs -ls YouTube/JP
-bash-4.2$ hdfs dfs -ls YouTube/KR
-bash-4.2$ hdfs dfs -ls YouTube/US
-bash-4.2$ hdfs dfs -ls YouTube/RU
-bash-4.2$ hdfs dfs -ls YouTube/category
```

5. Use the below command to check if the subdirectories are created or not.

```
-bash-4.2$ hdfs dfs -ls YouTube
```

```
drwxr-xr-x   - ssarkar4 hdfs          0 2023-11-28 22:59 YouTube/BR
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-04 19:56 YouTube/CA
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-04 20:24 YouTube/IN
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-04 20:52 YouTube/JP
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-04 21:03 YouTube/KR
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-04 21:10 YouTube/RU
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-04 21:09 YouTube/US
drwxr-xr-x   - ssarkar4 hdfs          0 2023-12-12 19:50 YouTube/category
```

6. Put the respective files in subdirectories & check if they exist or not.

```
hdfs dfs -put BR_youtube_trending_data.csv YouTube/BR

hdfs dfs -put CA_youtube_trending_data.csv YouTube/CA

hdfs dfs -put IN_youtube_trending_data.csv YouTube/IN

hdfs dfs -put JP_youtube_trending_data.csv YouTube/JP

hdfs dfs -put KR_youtube_trending_data.csv YouTube/KR

hdfs dfs -put US_youtube_trending_data.csv YouTube/US

hdfs dfs -put RU_youtube_trending_data.csv YouTube/RU

hdfs dfs -put category.csv YouTube/category
```

To check use the commands as shown below,

```
-bash-4.2$ hdfs dfs -ls YouTube/RU
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  447752282 2023-12-04 21:10 YouTube/RU/RU_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/US
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  357792502 2023-12-04 21:09 YouTube/US/US_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/IN
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  371421031 2023-12-04 20:24 YouTube/IN/IN_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/CA
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  360314130 2023-12-04 19:56 YouTube/CA/CA_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/BR
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  315460886 2023-11-28 22:59 YouTube/BR/BR_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/JP
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  479920557 2023-12-04 20:52 YouTube/JP/JP_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/KR
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs  280143719 2023-12-04 21:03 YouTube/KR/KR_youtube_trending_data.csv
-bash-4.2$ hdfs dfs -ls YouTube/category
Found 1 items
-rw-r--r--   3 ssarkar4 hdfs        498 2023-12-12 19:50 YouTube/category/category.csv
-bash-4.2$
```

Step 3: Create Hive Tables in HDFS

1. Setting up the Beeline environment to run the hive queries.
   Instructions: Open another Gitbash session to get into beeline Command Line Interface using the following command to run hive queries.
   Command: beeline

```
-bash-4.2$ beeline
SLF4J: Class path contains multiple SLF4J bindings.
SLF4J: Found binding in [jar:file:/usr/odh/1.1.7/hive/lib/log4j-slf4j-impl-2.17.1.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: Found binding in [jar:file:/usr/odh/1.1.7/hadoop/lib/slf4j-log4j12-1.7.25.jar!/org/slf4j/impl/StaticLoggerBinder.class]
SLF4J: See http://www.slf4j.org/codes.html#multiple_bindings for an explanation.
SLF4J: Actual binding is of type [org.apache.logging.slf4j.Log4jLoggerFactory]
Connecting to jdbc:hive2://bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com:2181,bigdaimn0.sub03291929060.trainingvcn.oraclevcn.com:2181,bigdaiwn0.su
vcn.oraclevcn.com:2181/default;password=ssarkar4;serviceDiscoveryMode=zooKeeper;user=ssarkar4;zooKeeperNamespace=hiveserver2
23/12/14 08:44:43 [main-EventThread]: ERROR imps.EnsembleTracker: Invalid config event received: {server.1=bigdaimn0.sub03291929060.trainingvcn.oracleve
icipant, version=0, server.3=bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:2888:3888:participant, server.2=bigdaiun0.sub03291929060.trainingvcn.or
:participant}
23/12/14 08:44:43 [main-EventThread]: ERROR imps.EnsembleTracker: Invalid config event received: {server.1=bigdaimn0.sub03291929060.trainingvcn.oracleve
icipant, version=0, server.3=bigdaiwn0.sub03291929060.trainingvcn.oraclevcn.com:2888:3888:participant, server.2=bigdaiun0.sub03291929060.trainingvcn.or
:participant}
23/12/14 08:44:43 [main]: INFO jdbc.HiveConnection: Connected to bigdaiun0.sub03291929060.trainingvcn.oraclevcn.com:10010
Connected to: Apache Hive (version 3.1.2)
Driver: Hive JDBC (version 3.1.2)
Transaction isolation: TRANSACTION_REPEATABLE_READ
Beeline version 3.1.2 by Apache Hive
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> |
```

2. Now you must create your database with your username to separate your tables from other users. For example, the user **ssarkar4** should run the following:

```
Beeline version 3.1.2 by Apache Hive
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE DATABASE IF NOT EXISTS ssarkar4;
```

3. Check for your database and use your database for creating tables.

```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
| group5                                   |
| hwoo                                     |
| information_schema                       |
| jarias68                                 |
| jbarret9                                 |
| jdiaz294                                 |
| jmarias                                  |
| jsanch369                                |
| jtam2                                    |
| jtang7                                   |
| jwoo5                                    |
| kadhika3                                 |
| kbhanda3                                 |
| lgreena                                  |
| lvelasq                                  |
| malmeid5                                 |
| mazad3                                   |
| mmiran64                                 |
| mnunez82                                 |
| mpere110                                 |
| mroman                                   |
| mvan10                                   |
| nkwok2                                   |
| sadirep                                  |
| sannam                                   |
| sapan                                    |
| seab                                     |
| skarri2                                  |
| ssarkar4                                 |
| synn                                     |
| sys                                      |
| tyu3                                     |
| uguijar                                  |
| vphanvo                                  |
| wfung2                                   |
| wgiron2                                  |
| xcolin                                   |
| ypolshy                                  |
| zpatel6                                  |
+------------------------------------------+
99 rows selected (0.135 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai>
```

```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;
INFO  : Compiling command(queryId=hive_20231214085558_e3a0e105-19ac-4532-b0ab-0704229089ed): use ssarkar4
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20231214085558_e3a0e105-19ac-4532-b0ab-0704229089ed); Time taken: 0.025 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20231214085558_e3a0e105-19ac-4532-b0ab-0704229089ed): use ssarkar4
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20231214085558_e3a0e105-19ac-4532-b0ab-0704229089ed); Time taken: 0.207 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
No rows affected (0.243 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> |
```

4. Create table for all the seven countries India, USA, Canada, Russia, Brazil, Korea, and Japan.

   Example for one country is shown below,

   TABLE: BRAZIL

```
DROP TABLE IF EXISTS Brazil;

CREATE EXTERNAL TABLE IF NOT EXISTS Brazil(
Video_id STRING,
Title STRING,
Published STRING,
Channel_id STRING,
Channel_title STRING,
category INT,
Trending STRING,
Tags STRING,
View_count INT,
Likes INT,
Dislikes INT,
Comments INT,
Thumbnail STRING,
Comment_des STRING,
Ratings_des STRING,
Description STRING
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/ssarkar4/YouTube/BR'
TBLPROPERTIES ('skip.header.line.count'='1');
```

```
No rows affected (0.243 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> DROP TABLE IF EXISTS Brazil;
INFO  : Compiling command(queryId=hive_20231214090141_250c73dd-86ad-4757-a831-0f6f39dccd4a): DROP TABLE IF EXISTS Brazil
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20231214090141_250c73dd-86ad-4757-a831-0f6f39dccd4a); Time taken: 0.042 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20231214090141_250c73dd-86ad-4757-a831-0f6f39dccd4a): DROP TABLE IF EXISTS Brazil
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20231214090141_250c73dd-86ad-4757-a831-0f6f39dccd4a); Time taken: 0.363 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
No rows affected (0.419 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai>
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE EXTERNAL TABLE IF NOT EXISTS Brazil(
. . . . . .> Video_id STRING,
. . . . . .> Title STRING,
. . . . . .> Published STRING,
. . . . . .> Channel_id STRING,
. . . . . .> Channel_title STRING,
. . . . . .> category INT,
. . . . . .> Trending STRING,
. . . . . .> Tags STRING,
. . . . . .> View_count INT,
. . . . . .> Likes INT,
. . . . . .> Dislikes INT,
. . . . . .> Comments INT,
. . . . . .> Thumbnail STRING,
. . . . . .> Comment_des STRING,
. . . . . .> Ratings_des STRING,
. . . . . .> Description STRING
. . . . . .> )
. . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE LOCATION '/user/ssarkar4/YouTube/BR'
. . . . . .> TBLPROPERTIES ('skip.header.line.count'='1');
```

Note: Similarly create table for countries India, USA, Canada, Russia, Korea, and Japan.

5. Perform data Cleaning by createing new clean table for each countries. Thereby, retaining the required data column only,removing irregular NULL data and formatting the data:

   Example for one country is shown below,

   TABLE: BRCleanedTable for Brazil

```
DROP TABLE IF EXISTS BRCleanedTable;
CREATE TABLE BRCleanedTable AS
SELECT
  video_id,
  title,
  DATE_FORMAT(FROM_UNIXTIME(UNIX_TIMESTAMP(Published, 'yyyy-MM-dd\'T\'HH:mm:ss\'Z\'')), 'yyyy-MM-dd') AS published_date,
  channel_id,
  channel_title,
  category,
  DATE_FORMAT(FROM_UNIXTIME(UNIX_TIMESTAMP(Trending, 'yyyy-MM-dd\'T\'HH:mm:ss\'Z\'')), 'yyyy-MM-dd') AS trending_date,
  tags,
  view_count,
  likes,
  dislikes,
  Comments
FROM Brazil
WHERE Published IS NOT NULL
  AND Trending IS NOT NULL;
```

```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> DROP TABLE IF EXISTS BRCleanedTable;
INFO  : Compiling command(queryId=hive_20231214091935_d76319ba-64bc-46d8-b587-b27109a380d6): DROP TABLE IF EXISTS BRCleanedTable
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20231214091935_d76319ba-64bc-46d8-b587-b27109a380d6); Time taken: 0.037 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20231214091935_d76319ba-64bc-46d8-b587-b27109a380d6): DROP TABLE IF EXISTS BRCleanedTable
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20231214091935_d76319ba-64bc-46d8-b587-b27109a380d6); Time taken: 0.362 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
No rows affected (0.415 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE TABLE BRCleanedTable AS
. . . . . .> SELECT
. . . . . .>   video_id,
. . . . . .>   title,
. . . . . .>   DATE_FORMAT(FROM_UNIXTIME(UNIX_TIMESTAMP(Published, 'yyyy-MM-dd\'T\'HH:mm:ss\'Z\'')), 'yyyy-MM-dd') AS published_date,
. . . . . .>   channel_id,
. . . . . .>   channel_title,
. . . . . .>   category,
. . . . . .>   DATE_FORMAT(FROM_UNIXTIME(UNIX_TIMESTAMP(Trending, 'yyyy-MM-dd\'T\'HH:mm:ss\'Z\'')), 'yyyy-MM-dd') AS trending_date,
. . . . . .>   tags,
. . . . . .>   view_count,
. . . . . .>   likes,
. . . . . .>   dislikes,
. . . . . .>   Comments
. . . . . .> FROM Brazil
. . . . . .> WHERE Published IS NOT NULL
. . . . . .>   AND Trending IS NOT NULL;
```

Note: Similarly create table for countries India, USA, Canada, Russia, Korea, and Japan.

6. Create a **categories** table and show the table content;

```
DROP TABLE IF EXISTS categories;
CREATE EXTERNAL TABLE IF NOT EXISTS categories (
category_id INT,
category_name STRING
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION '/user/ssarkar4//YouTube/category'
TBLPROPERTIES ('skip.header.line.count'='1');
```

```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> DROP TABLE IF EXISTS categories;
INFO  : Compiling command(queryId=hive_20231214093317_f3894c87-99c0-40ec-9afc-7589dccad622): DROP TABLE IF EXISTS categories
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
INFO  : Returning Hive schema: Schema(fieldSchemas:null, properties:null)
INFO  : Completed compiling command(queryId=hive_20231214093317_f3894c87-99c0-40ec-9afc-7589dccad622); Time taken: 0.18 seconds
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Executing command(queryId=hive_20231214093317_f3894c87-99c0-40ec-9afc-7589dccad622): DROP TABLE IF EXISTS categories
INFO  : Starting task [Stage-0:DDL] in serial mode
INFO  : Completed executing command(queryId=hive_20231214093317_f3894c87-99c0-40ec-9afc-7589dccad622); Time taken: 0.36 seconds
INFO  : OK
INFO  : Concurrency mode is disabled, not creating a lock manager
No rows affected (0.555 seconds)
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> CREATE EXTERNAL TABLE IF NOT EXISTS categories (
. . . . . .>     category_id INT,
. . . . . .>     category_name STRING
. . . . . .> )
. . . . . .> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
. . . . . .> STORED AS TEXTFILE LOCATION '/user/ssarkar4//YouTube/category'
. . . . . .> TBLPROPERTIES ('skip.header.line.count'='1');
```

```
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> select * from categories;
INFO  : Compiling command(queryId=hive_20231214093503_0fdd89a0-8333-4722-b342-533ca1b44213): select * from categories
INFO  : Concurrency mode is disabled, not creating a lock manager
INFO  : Semantic Analysis Completed (retrial = false)
```

categories.category_id	categories.category_name
1	Film & Animation
2	Autos & Vehicles
10	Music
15	Pets & Animals
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
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

7. Check if all the tables are created;

```
0: jdbc: hive2://bigdaiun0.sub03291929060.trai> show tables;
```

Created tables are highlighted in BLUE as shown below.

tab_name
brazil
brcleanedtable
cacleanedtable
canada
categories
drivers
incleanedtable
india
japan
jpcleanedtable
korea
krcleanedtable
products
ratings
rucleanedtable
russia
top10
top10_trending_videos
top_tags_frequency
top_trending_video_country_wise
top_viewed_categories
top_viewed_channels
top_viewed_channels_by_year
truck_events
tweets_top_countries
tweetsbi
usa
uscleanedtable

Step 4: Create Hive Queries to analyze data.

1. **Description:** Analyze the whole dataset & conclude the top-10 trending videos worldwide.

   Instructions:

   In this tutorial, created table using database "ssarkar4"
   Note: Please use your own database;

   0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
   0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;

   The following Hive statement creates a Hive table **top10_trending_videos** using data from the BRCleanedTable, INCleanedTable, USCleanedTable, RUCleanedTable, JPCleanedTable, KRCleanedTable, and CACleanedTable tables.

```
-- Drop the table if it exists
DROP TABLE IF EXISTS top10_trending_videos;

-- Create the table
CREATE TABLE top10_trending_videos
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION 'YouTube/top10_trending_videos/'
AS
SELECT TRENDING_VIDEOS, popularity_score
FROM (
```

```
SELECT title AS TRENDING_VIDEOS,
      popularity_score,
      ROW_NUMBER() OVER (PARTITION BY title ORDER BY popularity_score
DESC) as row_num
FROM (
    -- Data from INCleanedTable
    SELECT title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM INCleanedTable
    GROUP BY title

    UNION ALL
    -- Data from USCleanedTable
    SELECT title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM USCleanedTable
    GROUP BY title

    UNION ALL
    -- Data from RUCleanedTable
    SELECT title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM RUCleanedTable
    GROUP BY title

    UNION ALL
    -- Data from JPCleanedTable
    SELECT title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM JPCleanedTable
    GROUP BY title

    UNION ALL
    -- Data from KRCleanedTable
    SELECT title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM KRCleanedTable
    GROUP BY title

    UNION ALL
```

```
        -- Data from CACleanedTable
        SELECT title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM CACleanedTable
        GROUP BY title

        UNION ALL

        -- Data from BRCleanedTable
        SELECT title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM BRCleanedTable
        GROUP BY title
    ) AS AllData
) AS TrendingVideos
WHERE row_num = 1
ORDER BY popularity_score DESC
LIMIT 10;
```

The **top10_trending_videos** table consists of top-10 trending videos of worldwide ordered in the descending order of their engagement score(popularity_score).

```
--show table contents--

SELECT * FROM top10_trending_videos;
```

top10_trending_videos.trending_videos	top10_trending_videos.popularity_score
Turn into orbeez - Tutorial #Shorts	5836161791
BLACKPINK - ‘Pink Venom’ M/V	5156228086
7 Days Stranded At Sea	4519753431
JISOO - ‘꽃 (FLOWER)’ M/V	3670874902
World's Most Dangerous Trap!	3542360504
BTS (방탄소년단) 'Butter' Official MV	3450633936
BLACKPINK - ‘Shut Down’ M/V	3409167584
정국 (Jung Kook) 'Seven (feat. Latto)' Official MV	3222180349
Rihanna’s FULL Apple Music Super Bowl LVII Halftime Show	2910440706
BTS (방탄소년단) 'Permission to Dance' Official MV	2405014859

Get the file on Linux from HDFS: Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS

```
hdfs dfs -get YouTube/top10_trending_videos/000000_0 top10_trending_videos.csv
```

Copy the file to local PC: Open another terminal with git bash to download the file to local PC. Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password and then the file will be downloaded.

```
scp ssarkar4@129.153.66.218:/home/ssarkar4/top10_trending_videos.csv top10_trending_videos.csv
```

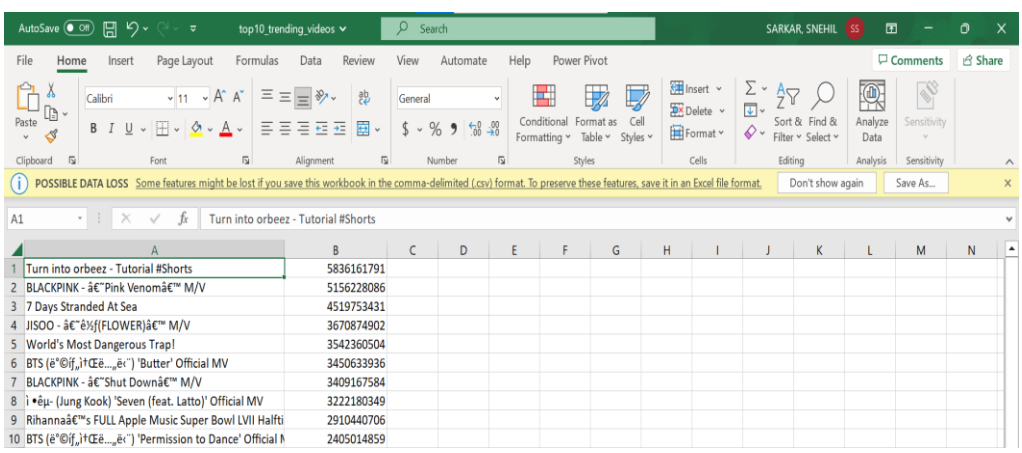
Run the following command to check if files are present:

```
ls -al
```

```
-rw-r--r--   1 ssarkar4 ssarkar4       504 Dec 13 07:18 top10_trending_videos.cs
v
```

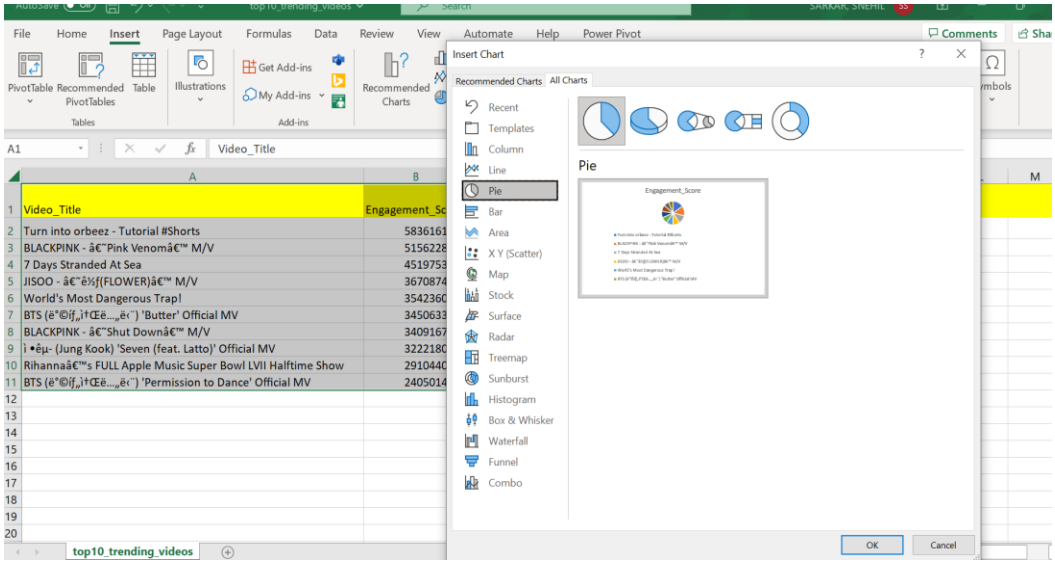
Visualization:

i. Open the top10_trending_videos.csv file from the location /home/ssarkar4/…and check if all the content of table top10_trending_videos is present in it.

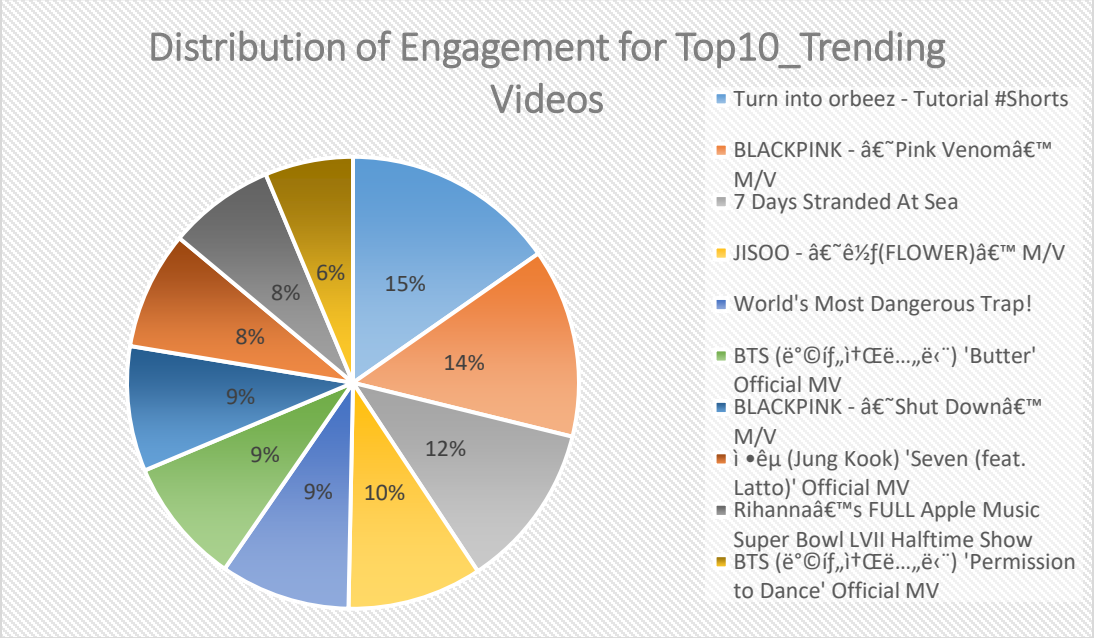


	A	B
1	Turn into orbeez - Tutorial #Shorts	5836161791
2	BLACKPINK - â€˜Pink Venomâ€™ M/V	5156228086
3	7 Days Stranded At Sea	4519753431
4	JISOO - â€˜ë½ƒ(FLOWER)â€™ M/V	3670874902
5	World's Most Dangerous Trap!	3542360504
6	BTS (ë°©íƒ„ì†Œë…„ë¨) 'Butter' Official MV	3450633936
7	BLACKPINK - â€˜Shut Downâ€™ M/V	3409167584
8	ì •êµ- (Jung Kook) 'Seven (feat. Latto)' Official MV	3222180349
9	Rihannaâ€™s FULL Apple Music Super Bowl LVII Halfti	2910440706
10	BTS (ë°©íƒ„ì†Œë…„ë¨) 'Permission to Dance' Official M	2405014859

ii. Insert a row header with column name as “VideoTitle” & “Engagement Score” respectively. Click on Save As top10_trending_videos.xlnx format . Select all the data inside it and click on Insert. Then, Select the pie chart to show distribution engagement.



iii. Click ok to plot the Distribution of Engagement graph of Top-10 trending videos worldwide and rename the title as shown in below figure:



2. **Description:** Figure out the most trending video of individual countries based on the engagement score.

Instructions:

In this tutorial, created table using database "ssarkar4"
Note: Please use your own database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;

The following Hive statement creates a Hive table **top_trending_video_country_wise** using data from the BRCleanedTable, INCleanedTable, USCleanedTable, RUCleanedTable, JPCleanedTable, KRCleanedTable, and CACleanedTable tables.

```
-- Drop the table if it exists
DROP TABLE IF EXISTS top_trending_video_country_wise;

-- Create the table
CREATE TABLE top_trending_video_country_wise
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION 'YouTube/top_trending_video_country_wise/'
AS
SELECT country, title AS TRENDING_VIDEO, popularity_score
FROM (
  SELECT country, title, popularity_score,
      ROW_NUMBER() OVER (PARTITION BY country ORDER BY popularity_score
DESC) AS video_rank
  FROM (
    -- Union of aggregated data from multiple country tables
    SELECT 'Brazil' AS country, title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM BRCleanedTable
    GROUP BY title

    UNION ALL

    SELECT 'India' AS country, title,
        SUM(view_count + likes + dislikes + Comments) AS popularity_score
    FROM INCleanedTable
    GROUP BY title
```

```
        UNION ALL

        SELECT 'United States' AS country, title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM USCleanedTable
        GROUP BY title

        UNION ALL

        SELECT 'Russia' AS country, title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM RUCleanedTable
        GROUP BY title

        UNION ALL

        SELECT 'Japan' AS country, title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM JPCleanedTable
        GROUP BY title

        UNION ALL

        SELECT 'South Korea' AS country, title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM KRCleanedTable
        GROUP BY title

        UNION ALL

        SELECT 'Canada' AS country, title,
            SUM(view_count + likes + dislikes + Comments) AS popularity_score
        FROM CACleanedTable
        GROUP BY title
    ) AS CountryData
) AS RankedVideos
WHERE video_rank = 1;
```

The **top_trending_video_country_wise** table consists of trending videos of each country.

```
--show table contents--

SELECT * FROM top_trending_video_country_wise;
```

top_trending_video_country_wise.country	top_trending_video_country_wise.trending_video	top_trending_video_country_wise.popularity_score
Canada	BTS (방탄소년단) 'Dynamite' Official MV	1541806680
India	BTS (방탄소년단) 'Dynamite' Official MV	2086190763
Russia	BLACKPINK - ‘Pink Venom’ M/V	645047944
Brazil	BLACKPINK - ‘Pink Venom’ M/V	1935558240
Japan	BTS (방탄소년단) 'Butter' Official MV	2814750333
South Korea	BLACKPINK - ‘Pink Venom’ M/V	4344025045
United States	Turn into orbeez - Tutorial #Shorts	5836161791

Get the file on Linux from HDFS: Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS

```
hdfs dfs -getmerge YouTube/top_trending_video_country_wise/
top_trending_video_country_wise.csv
```

Copy the file to local PC: Open another terminal with git bash to download the file to local PC. Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password and then the file will be downloaded.

```
scp ssarkar4@129.153.66.218:/home/ssarkar4/top_trending_video_country_wise.csv
top_trending_video_country_wise.csv
```

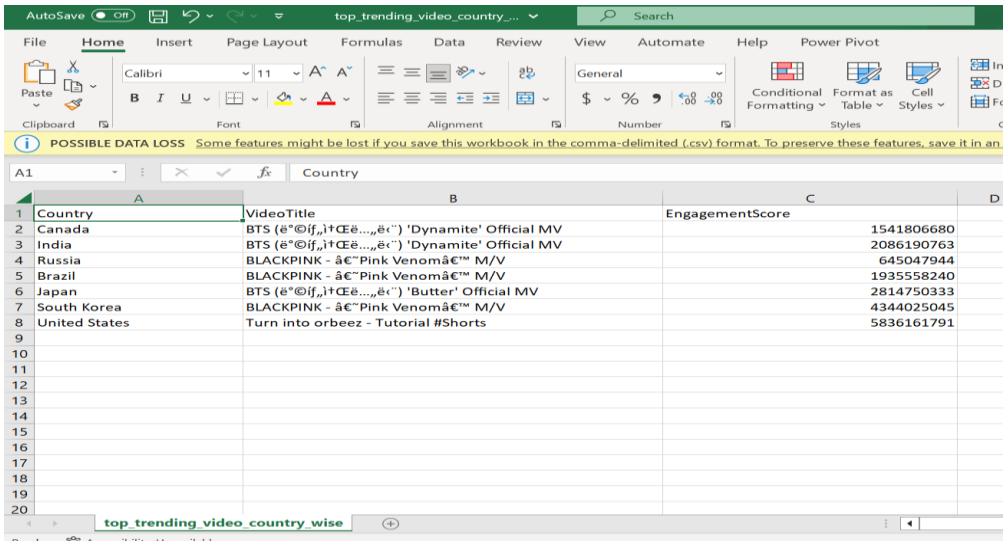
Run the following command to check if files are present:

```
ls -al
```

```
-rw-r--r--   1 ssarkar4 ssarkar4       403 Dec  9 19:49 top_trending_video_count
ry_wise.csv
```

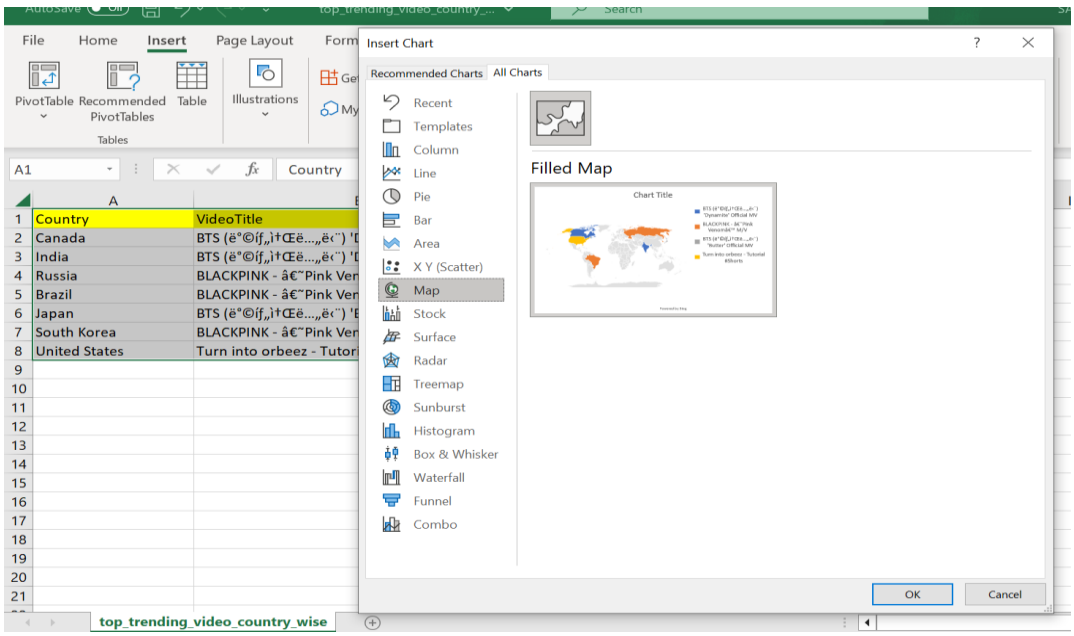
Visualization:

i. Open the top_trending_video_country_wise.csv file from the location /home/ssarkar4/… and check if all the contents of top_trending_video_country_wise table is present in it.

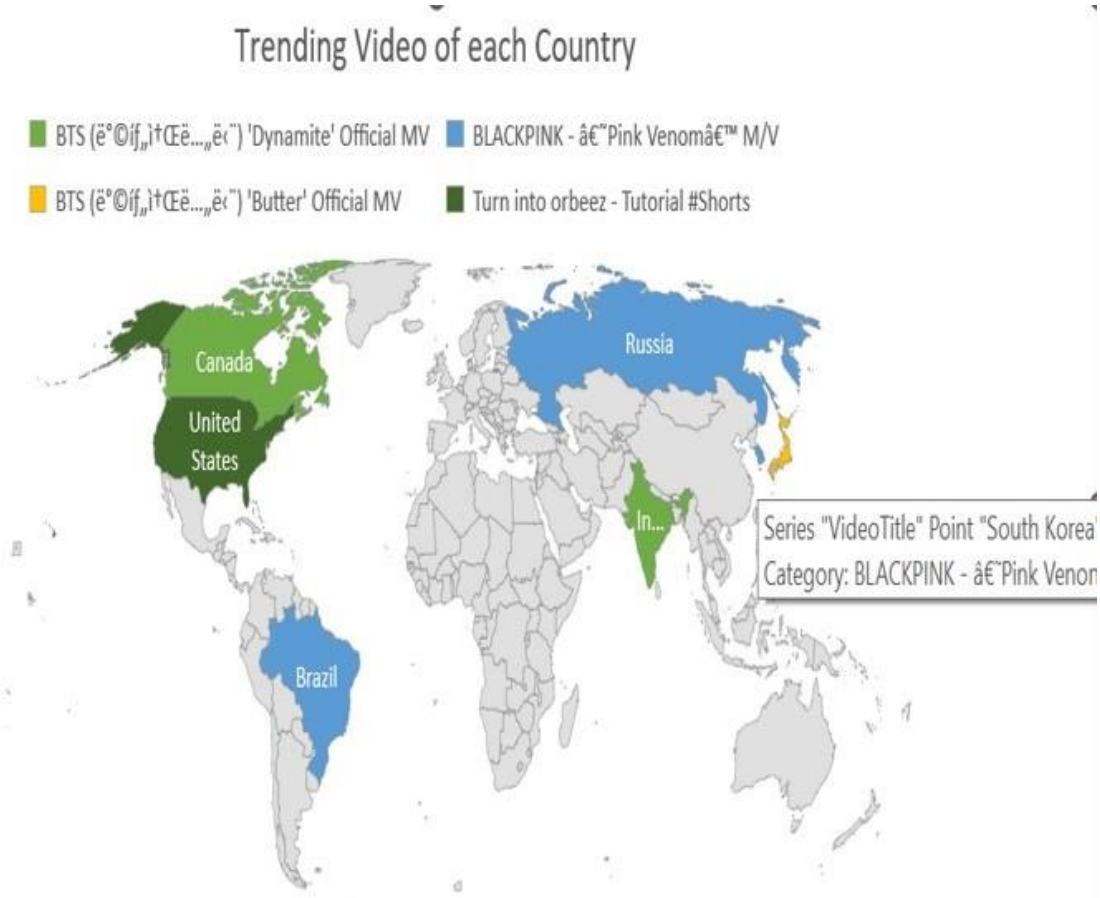


Country	VideoTitle	EngagementScore
Canada	BTS (ë°©íƒ„ì†Œë…„ë‹¨) 'Dynamite' Official MV	1541806680
India	BTS (ë°©íƒ„ì†Œë…„ë‹¨) 'Dynamite' Official MV	2086190763
Russia	BLACKPINK - â€˜Pink Venomâ€™ M/V	645047944
Brazil	BLACKPINK - â€˜Pink Venomâ€™ M/V	1935558240
Japan	BTS (ë°©íƒ„ì†Œë…„ë‹¨) 'Butter' Official MV	2814750333
South Korea	BLACKPINK - â€˜Pink Venomâ€™ M/V	4344025045
United States	Turn into orbeez - Tutorial #Shorts	5836161791

ii. Insert a row header with column name as “Country” “VideoTitle” & “Engagement Score” respectively. ”. Click on Save As top_trending_video_country_wise.xlnx excel format.Select all the data inside it and click on Insert. Then, Select the map chart to show the trending videos of each country.



iii. Click ok to plot the World Map Chart with the top trending videos of each country and rename the title as shown in below figure:



3. **Description:** Conducted year-on-year analysis to find the most viewed channel in the USA and identify content preferences and marketing opportunities.

   A. **Top-20 Most Viewed Channels in the USA**

Instructions:

In this tutorial, created table using database "ssarkar4"
Note: Please use your own database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;

The following Hive statement creates a Hive table **top_viewed_channels** using data from the USCleanedTable.

```
DROP TABLE IF EXISTS top_viewed_channels;

-- Create the top_viewed_channels table by selecting from the existing table
CREATE TABLE top_viewed_channels
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION 'YouTube/top_viewed_channels/'
AS
  SELECT
    Channel_id,
    Channel_title,
    SUM(view_count) AS total_views
  FROM USCleanedTable  -- Replace with the actual name of your existing table
  GROUP BY Channel_id, Channel_title
  ORDER BY total_views DESC
  LIMIT 20;  -- Adjust the limit to get the top N most viewed channels
```

The **top_viewed_channels** table consists of top-20 most viewed channels in the USA for the year span(2020-2023).

```
--show table contents--

select channel_title,total_views from top_viewed_channels;
```

channel_title	total_views
MrBeast	18963233126
BLACKPINK	18309354732
HYBE LABELS	12121949880
SMTOWN	8325187030
JYP Entertainment	7681590634
DaFuq!?Boom!	7358621115
Marvel Entertainment	6148647386
NFL	6046723443
FFUNTV	5030654339
Sony Pictures Entertainment	4697323088
BANGTANTV	4208701142
Dude Perfect	4174630089
America's Got Talent	3863611252
Warner Bros. Pictures	3828133452
Bizarrap	3476385302
Big Hit Labels	3378335085
Bad Bunny	3261616438
MrBeast Gaming	3060170674
Apple	3056798758
Mark Rober	2953322378

Get the file on Linux from HDFS: Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS

```
hdfs dfs -get YouTube/top_viewed_channels/000000_0 top_viewed_channels.csv
```

Copy the file to local PC: Open another terminal with git bash to download the file to local PC. Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password and then the file will be downloaded.

```
scp ssarkar4@129.153.66.218:/home/ssarkar4/top_viewed_channels.csv top_viewed_channels.csv
```

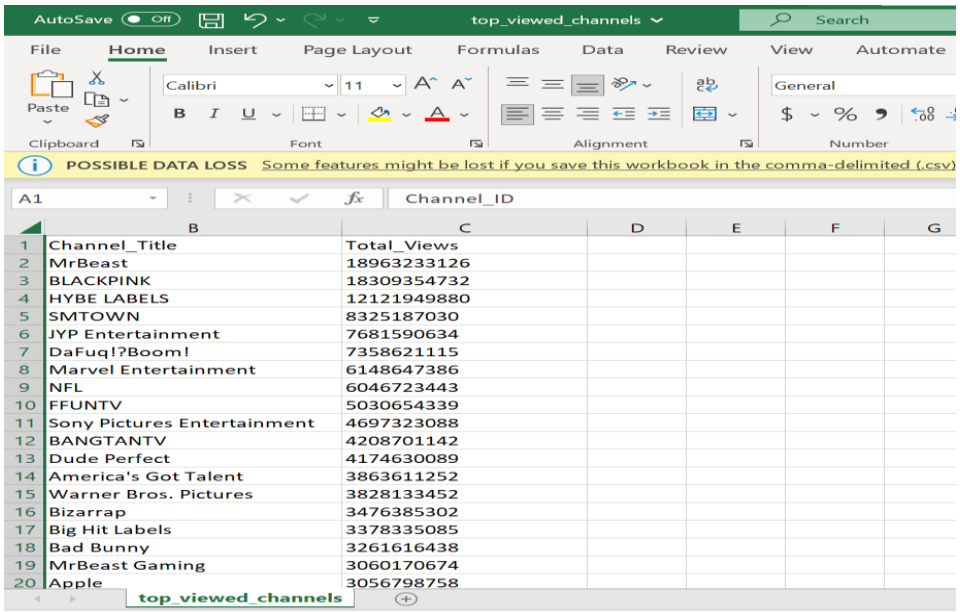
Run the following command to check if files are present:

```
ls -al
```

```
-rw-r--r--   1 ssarkar4 ssarkar4      983 Dec  7 21:24 top_viewed_channels.csv
```

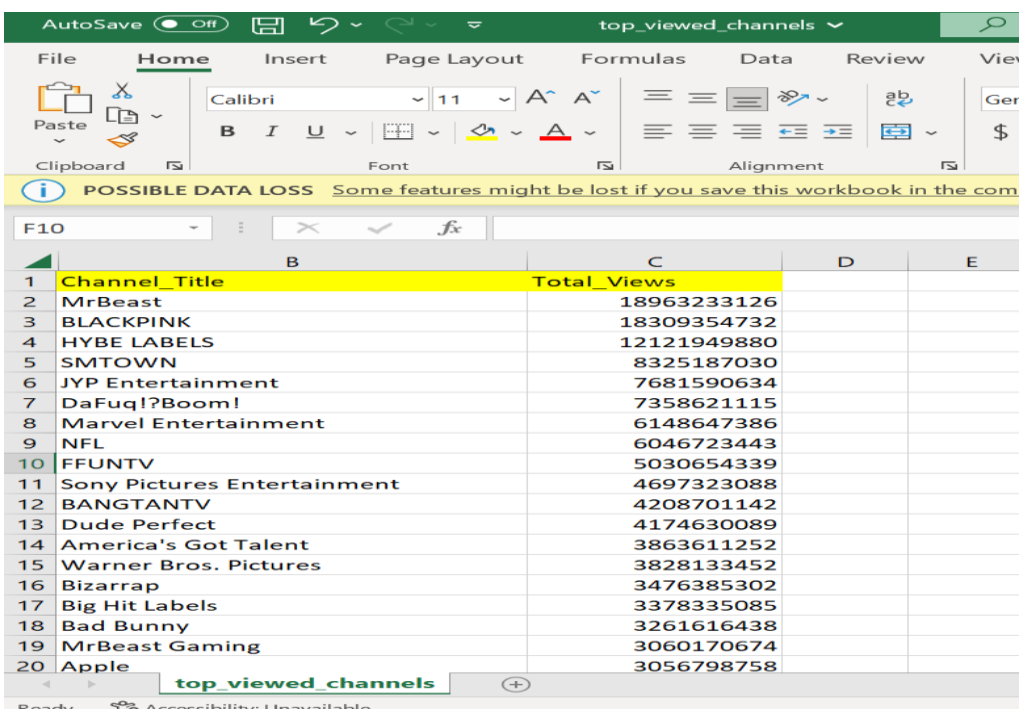
Visualization:

i. Open the top_viewed_channels.csv file from the location /home/ssarkar4/… and check if the content matches with the table **top_viewed_channels.**



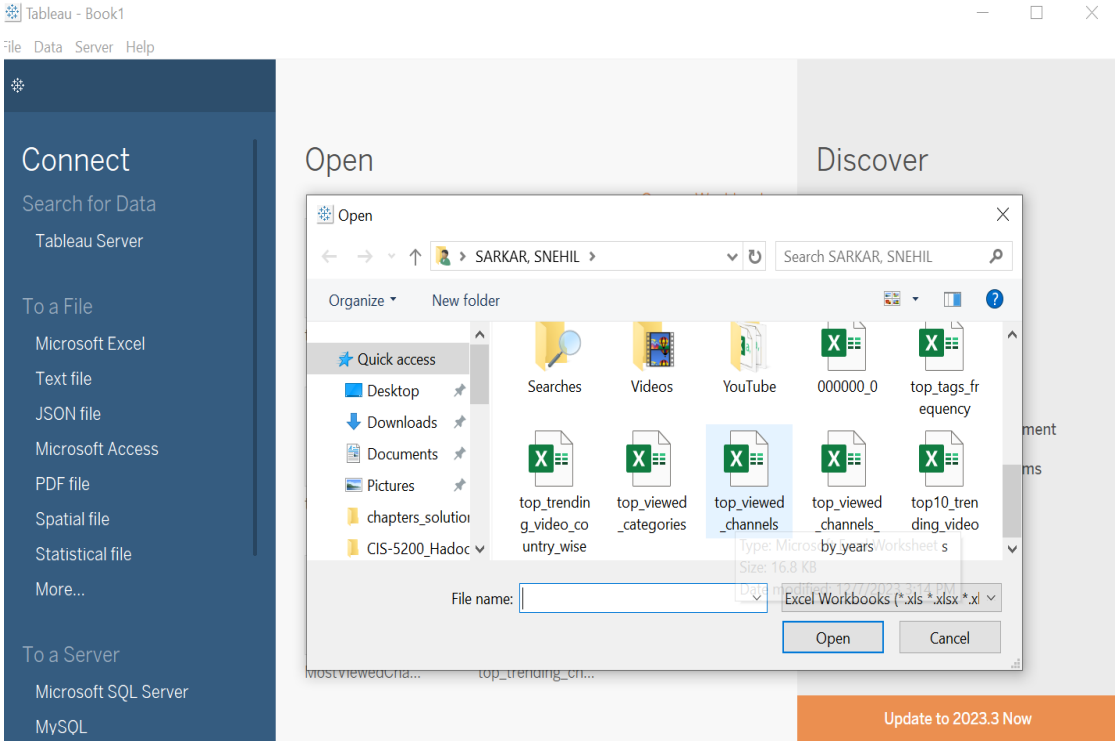
	B	C
1	Channel_Title	Total_Views
2	MrBeast	18963233126
3	BLACKPINK	18309354732
4	HYBE LABELS	12121949880
5	SMTOWN	8325187030
6	JYP Entertainment	7681590634
7	DaFuq!?Boom!	7358621115
8	Marvel Entertainment	6148647386
9	NFL	6046723443
10	FFUNTV	5030654339
11	Sony Pictures Entertainment	4697323088
12	BANGTANTV	4208701142
13	Dude Perfect	4174630089
14	America's Got Talent	3863611252
15	Warner Bros. Pictures	3828133452
16	Bizarrap	3476385302
17	Big Hit Labels	3378335085
18	Bad Bunny	3261616438
19	MrBeast Gaming	3060170674
20	Apple	3056798758

ii. Insert a header row with column name as “Channel_Title” & “Total_Views”. Click on Save As top_viewed_channels.xlnx excel format.

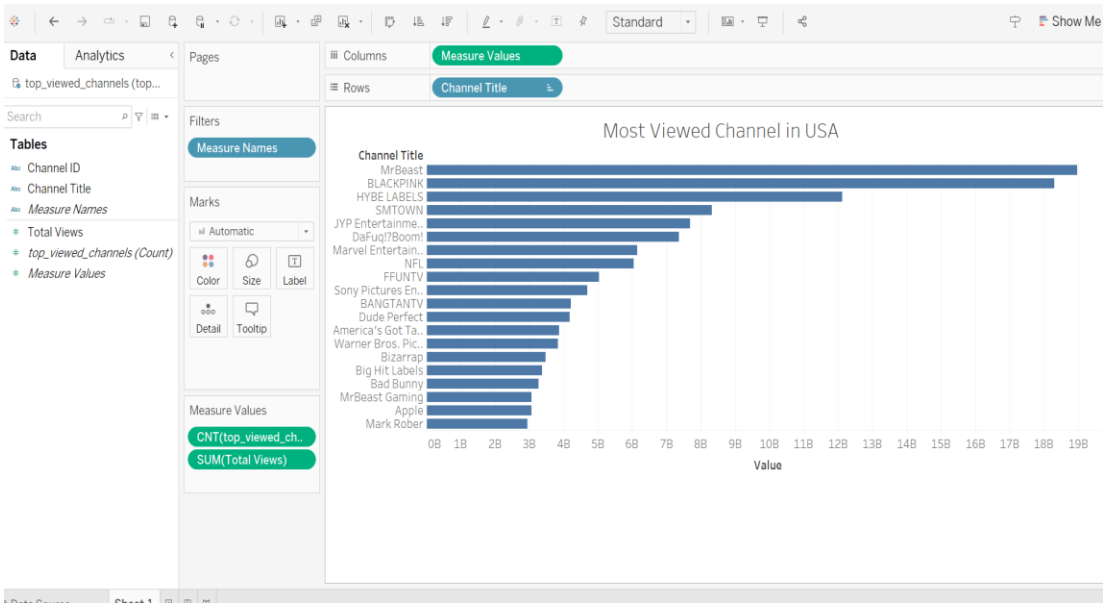


	B	C
1	Channel_Title	Total_Views
2	MrBeast	18963233126
3	BLACKPINK	18309354732
4	HYBE LABELS	12121949880
5	SMTOWN	8325187030
6	JYP Entertainment	7681590634
7	DaFuq!?Boom!	7358621115
8	Marvel Entertainment	6148647386
9	NFL	6046723443
10	FFUNTV	5030654339
11	Sony Pictures Entertainment	4697323088
12	BANGTANTV	4208701142
13	Dude Perfect	4174630089
14	America's Got Talent	3863611252
15	Warner Bros. Pictures	3828133452
16	Bizarrap	3476385302
17	Big Hit Labels	3378335085
18	Bad Bunny	3261616438
19	MrBeast Gaming	3060170674
20	Apple	3056798758

iii. Open Tableau and load the top_viewed_channels.xlnx excel file to plot Visual representation of the Most Viewed Channel of the USA:



iv. Click on sheets and in the rows, field select "Channel Title" and in the column field select "measured value" which is the total view count and rename the title of the sheet as "Most Viewed channel in USA".



B. Top Viewed Channels in the USA Year Wise

Instructions:

In this tutorial, created a table using the database “ssarkar4”.
Note: Please use your database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;

The following Hive statement creates a Hive table **top_viewed_channels_by_years** using data from the USCleanedTable.

```
DROP TABLE top_viewed_channels_by_year;

CREATE TABLE top_viewed_channels_by_year
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'YouTube/top_viewed_channels_by_years/'
AS
SELECT
   Channel_title,
   total_views,
   year
FROM (
   SELECT
      Channel_title,
      SUM(view_count) AS total_views,
      YEAR(published_date) AS year,
      ROW_NUMBER() OVER (PARTITION BY YEAR(published_date) ORDER BY
SUM(view_count) DESC) AS rnk
   FROM USCleanedTable
   WHERE published_date IS NOT NULL
   GROUP BY Channel_title, YEAR(published_date)
) ranked
WHERE rnk = 1
ORDER BY year;
```

The **top_viewed_channels_by_years** table consists of the most viewed channels in the USA for each year.

```
--show table contents--

SELECT * FROM top_viewed_channels_by_year;
```

top_viewed_channels_by_year.channel_title	top_viewed_channels_by_year.total_views	top_viewed_channels_by_year.year
Big Hit Labels	3288794031	2020
FFUNTV	5030654339	2021
BLACKPINK	8810490280	2022
MrBeast	12984742500	2023

Get the file on Linux from HDFS: Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS

```
hdfs dfs -get YouTube/top_viewed_channels_by_years/000000_0
top_viewed_channels_by_years.csv
```

Copy the file to local PC: Open another terminal with git bash to download the file to local PC. Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password and then the file will be downloaded.

```
scp ssarkar4@129.153.66.218:/home/ssarkar4/top_viewed_channels_by_years.csv
top_viewed_channels_by_years.csv
```

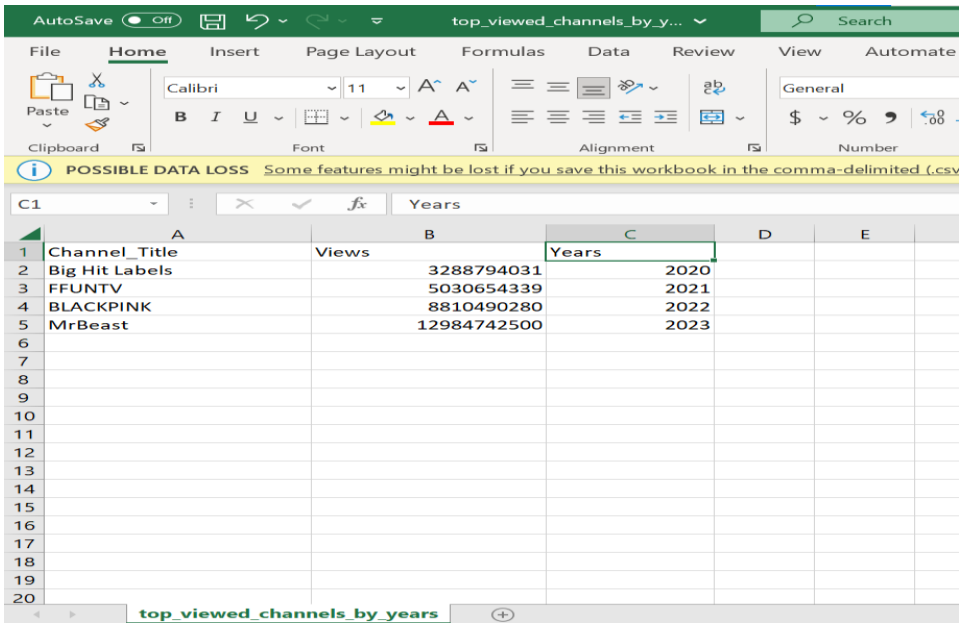
Run the following command to check if files are present:

```
ls -al
```

```
-rw-r--r--   1 ssarkar4 ssarkar4       105 Dec  7 23:44 top_viewed_channels_by_y
ears.csv
```

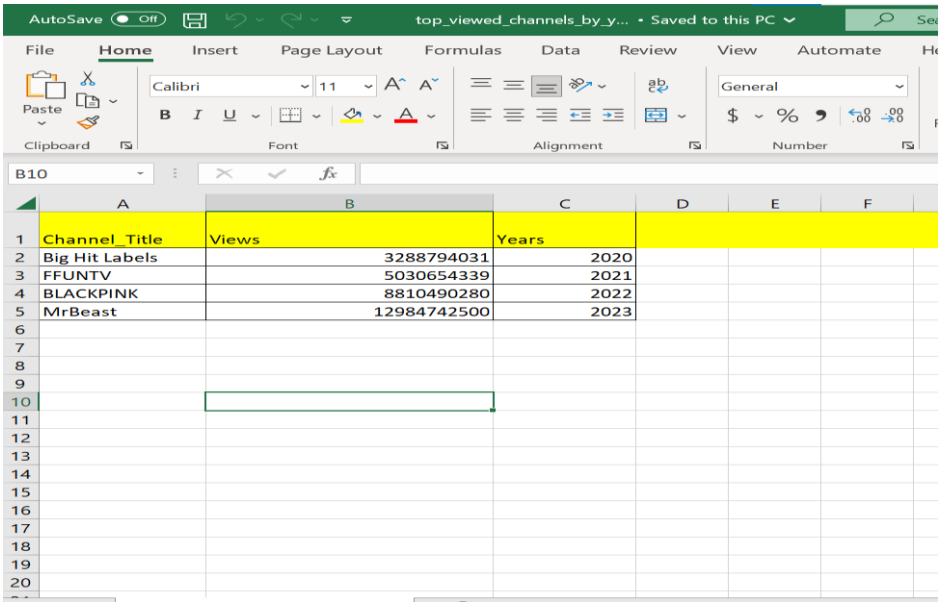
Visualization:

i. Open the top_viewed_channels_by_years.csv file from the location /home/ssarkar4/… and check if the content matches with the table **top_viewed_channels_by_years**.



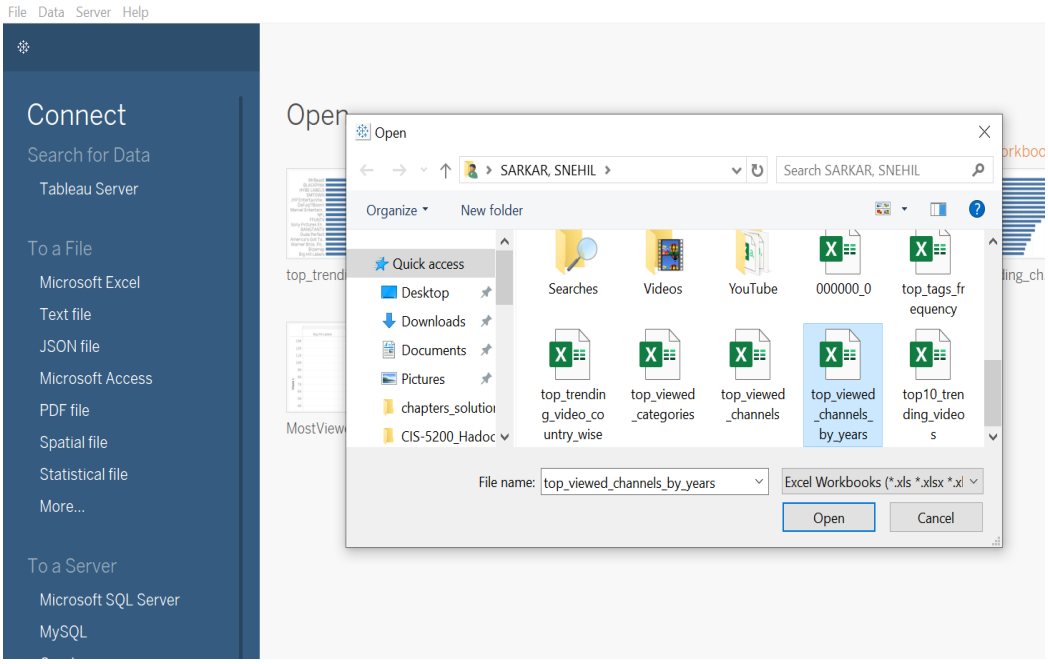
Channel_Title	Views	Years
Big Hit Labels	3288794031	2020
FFUNTV	5030654339	2021
BLACKPINK	8810490280	2022
MrBeast	12984742500	2023

ii. Insert a header row with column names as “Channel_Title” ,“Views” & “Years respectively. Click on Save As top_viewed_channels_by_years.xlnx excel format.

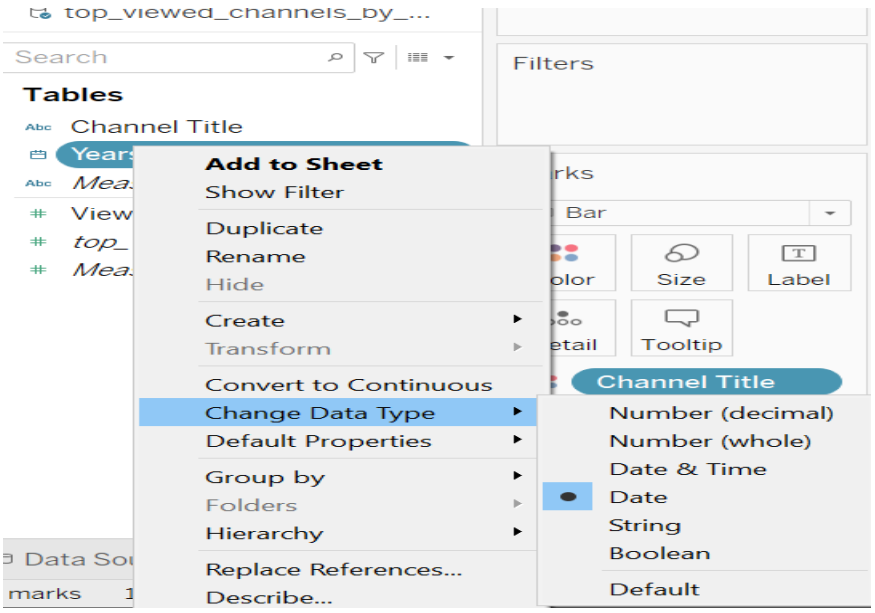


Channel_Title	Views	Years
Big Hit Labels	3288794031	2020
FFUNTV	5030654339	2021
BLACKPINK	8810490280	2022
MrBeast	12984742500	2023

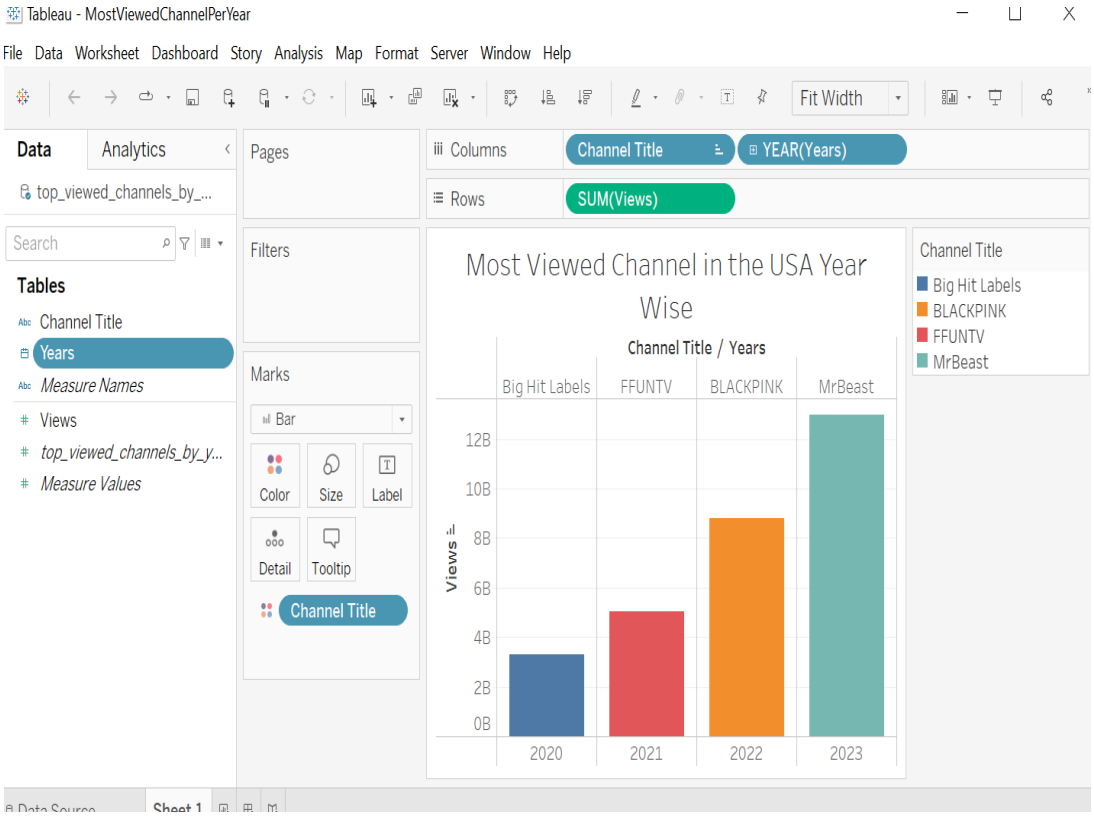
iii. Open Tableau and load the top_viewed_channels_by_years.xlnx excel file to plot a visual representation of the Topmost Viewed Channel of the USA Year Wise.



iv. Click on Sheets and then change the Data Type of Years to "Date."



v. In the columns field select “Channel Title” and “Years” and in the rows field select “Views” which is the total view count and rename the title of the sheet as “Most Viewed Channel in the USA Year Wise”.



4. **Description:** Extracted the most frequent tags used while uploading videos to understand the prevalent themes or topics in YouTube videos in India.

Instructions:

In this tutorial, created a table using the database “ssarkar4”.
Note: Please use your database;

0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;

The following Hive statement creates a Hive table **top_tags_frequency** using data from the INCleanedTable.

```
DROP TABLE IF EXISTS top_tags_frequency;

-- Drop the table if it already exists
DROP TABLE IF EXISTS top_tags_frequency;

-- Create the table
CREATE TABLE top_tags_frequency
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE
LOCATION 'YouTube/top_tags_frequency/'
AS
SELECT
   words_in_tags,
   COUNT(words_in_tags) AS frequency
FROM (
   SELECT
      EXPLODE(SPLIT(LOWER(CONCAT(tags, '|')), '[^a-zA-Z0-9]+')) AS words_in_tags
   FROM
      INCleanedTable
) expanded_tags
WHERE
   LENGTH(words_in_tags) > 2 AND words_in_tags IS NOT NULL AND words_in_tags !=
'none'
GROUP BY
   words_in_tags
ORDER BY
   frequency DESC
LIMIT 20;
```

The **top_tags_frequency** table consists of the top 20 Most searched/used words as Video Tags in India. Words with size>2 is taken into considerations.

```
--show table contents--

SELECT * FROM top_tags_frequency;
```

top_tags_frequency.words_in_tags	top_tags_frequency.frequency
new	153378
song	118190
songs	107368
comedy	100717
video	98463
latest	90618
tamil	62124
videos	56297
episode	55598
funny	52791
punjabi	52097
serial	51401
movie	50378
telugu	49818
hindi	45705
vlogs	43325
news	37730
2021	37650
vlog	35335
india	34339

Get the file on Linux from HDFS: Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS

```
hdfs dfs -get YouTube/top_tags_frequency/000000_0 top_tags_frequency.csv
```

Copy the file to local PC: Open another terminal with git bash to download the file to local PC. Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password and then the file will be downloaded.

```
scp ssarkar4@129.153.66.218:/home/ssarkar4/top_tags_frequency.csv top_tags_frequency.csv
```

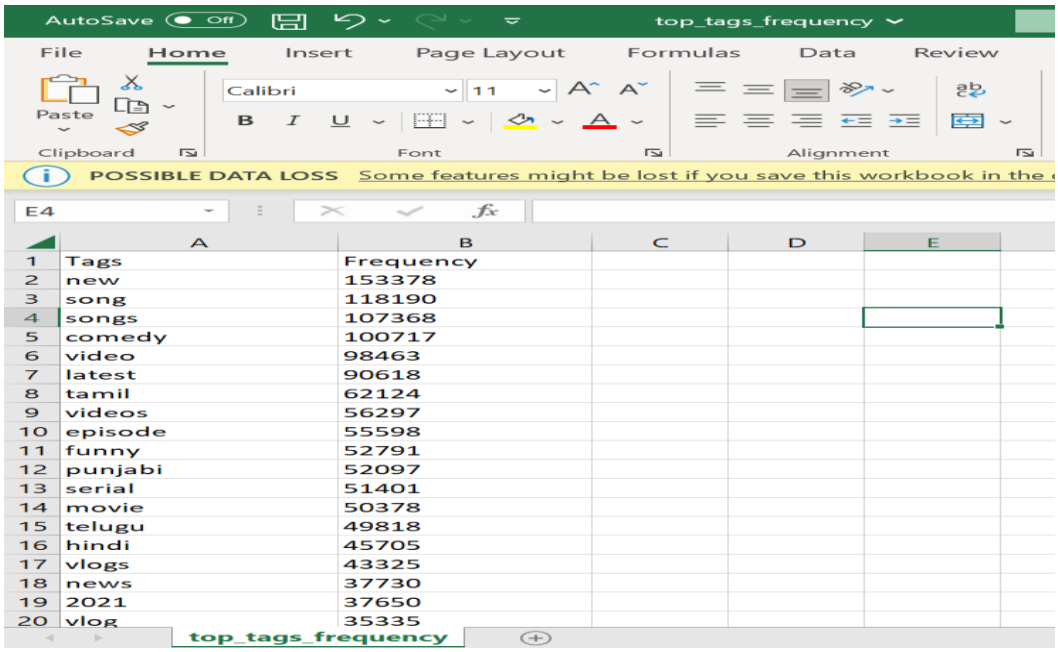
Run the following command to check if files are present:

```
ls -al
```

```
-rw-r--r--   1 ssarkar4 ssarkar4      247 Dec  8 23:42 top_tags_frequency.csv
```

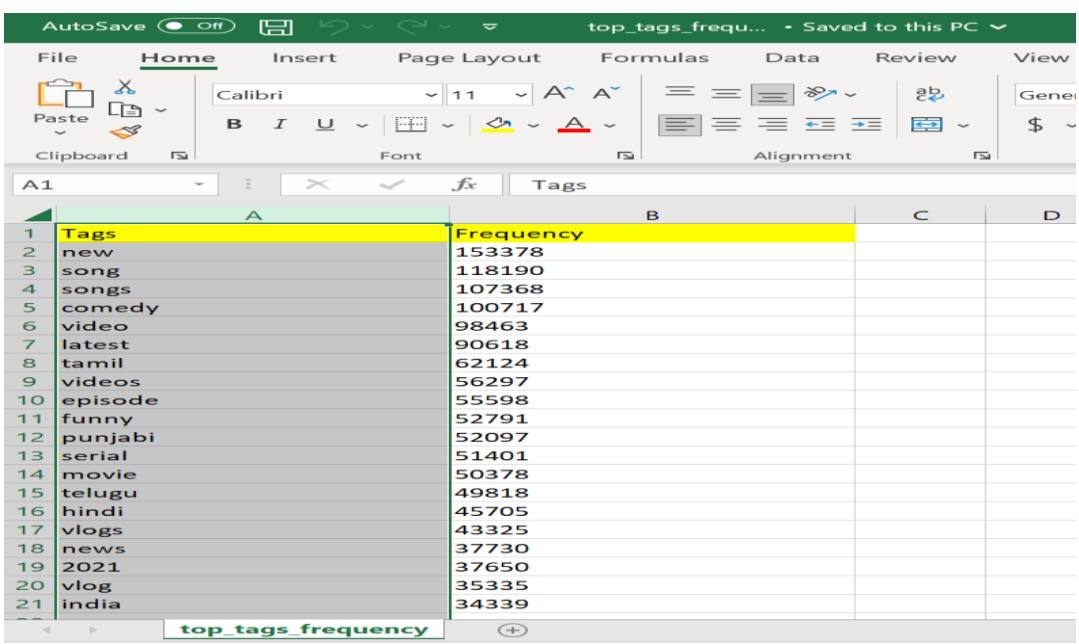
Visualization:

i. Open the top_tags_frequency.csv file from the location /home/ssarkar4/… and check if the content matches with the table **top_tags_frequency**.



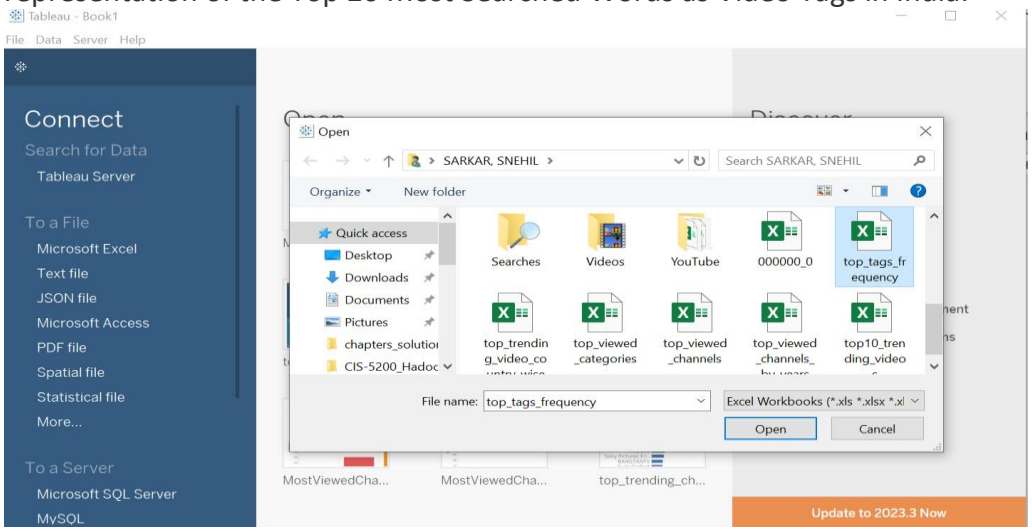
	A	B	C	D	E
1	Tags	Frequency			
2	new	153378			
3	song	118190			
4	songs	107368			
5	comedy	100717			
6	video	98463			
7	latest	90618			
8	tamil	62124			
9	videos	56297			
10	episode	55598			
11	funny	52791			
12	punjabi	52097			
13	serial	51401			
14	movie	50378			
15	telugu	49818			
16	hindi	45705			
17	vlogs	43325			
18	news	37730			
19	2021	37650			
20	vlog	35335			

ii. Insert a header row with column names as “Tags” & “Frequency” respectively. Click on Save As top_tags_frequency.xlnx excel format.

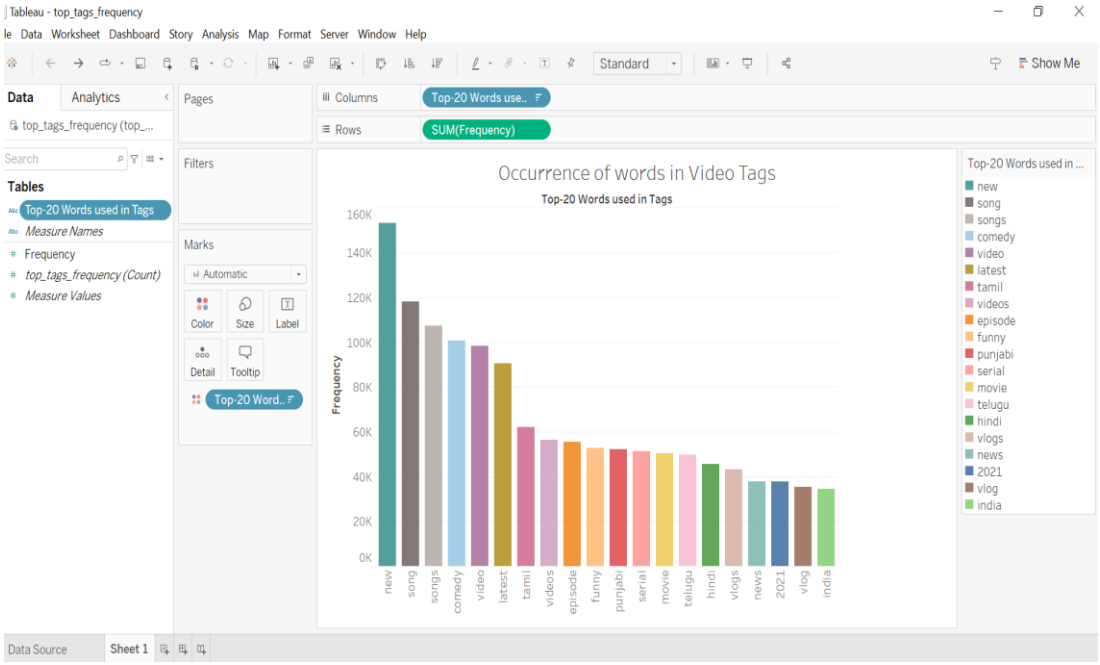


	A	B	C	D
1	Tags	Frequency		
2	new	153378		
3	song	118190		
4	songs	107368		
5	comedy	100717		
6	video	98463		
7	latest	90618		
8	tamil	62124		
9	videos	56297		
10	episode	55598		
11	funny	52791		
12	punjabi	52097		
13	serial	51401		
14	movie	50378		
15	telugu	49818		
16	hindi	45705		
17	vlogs	43325		
18	news	37730		
19	2021	37650		
20	vlog	35335		
21	india	34339		

iii. Open Tableau and select the top_tags_frequency.xlnx file to plot Visual representation of the Top 20 Most Searched Words as Video Tags in India.



iv. Click on Sheets and for the columns field select "Top 20 Words used in Tags" and in the rows field, field select "SUM(frequency)" which is the total view count, and rename the title of the sheet as "Occurrence of Words in Video Tags" for India.



5. **Description:** Analyzed the most viewed category of videos in all countries and identified the public interest every year.

   Instructions:

   In this tutorial, created a table using the database "ssarkar4".
   Note: Please use your database;

   0: jdbc:hive2://bigdaiun0.sub03291929060.trai> show database;
   0: jdbc:hive2://bigdaiun0.sub03291929060.trai> use ssarkar4;

   The following Hive statement creates a Hive table **top_viewed_categories** using data from the the BRCleanedTable, INCleanedTable, USCleanedTable, RUCleanedTable, JPCleanedTable, KRCleanedTable, and CACleanedTable tables.

```
-- Drop the table if it exists
DROP TABLE IF EXISTS top_viewed_categories;

-- Create the top_viewed_categories table by selecting from the existing CTE
CREATE TABLE top_viewed_categories
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
STORED AS TEXTFILE LOCATION 'YouTube/top_viewed_categories/'
AS
WITH RankedCategories AS (
  SELECT
    country,
    YEAR(published_date) AS year,
    category,
    COUNT(*) AS category_occurrences,
    ROW_NUMBER() OVER (PARTITION BY country, YEAR(published_date) ORDER
BY COUNT(*) DESC) AS category_rank
  FROM (
    SELECT 'India' AS country, * FROM INCleanedTable
    UNION ALL
    SELECT 'USA' AS country, * FROM USCleanedTable
    UNION ALL
    SELECT 'Russia' AS country, * FROM RUCleanedTable
    UNION ALL
    SELECT 'Brazil' AS country, * FROM BRCleanedTable
    UNION ALL
    SELECT 'Japan' AS country, * FROM JPCleanedTable
    UNION ALL
    SELECT 'South Korea' AS country, * FROM KRCleanedTable
    UNION ALL
```

```
    SELECT 'Canada' AS country, * FROM CACleanedTable
  ) AS CombinedTables
  WHERE category IS NOT NULL
  GROUP BY country, YEAR(published_date), category
)

SELECT
  country,
  year,
  category_name,
  category,
  category_occurrences
FROM RankedCategories rc
JOIN categories c ON rc.category = c.category_id
WHERE category_rank = 1
ORDER BY country, year;
```

The **top_viewed_categories** table consists of the most-watched category of videos in each country year-wise.

```
--show table contents--

SELECT * FROM top_viewed_categories;
```

top_viewed_categories.country	top_viewed_categories.year	top_viewed_categories.category_name	top_viewed_categories.category	top_viewed_categories.category_occurrences
Brazil	2020	Music	10	6976
Brazil	2021	Entertainment	24	18336
Brazil	2022	Entertainment	24	15782
Brazil	2023	Entertainment	24	12633
Canada	2020	Entertainment	24	5364
Canada	2021	Entertainment	24	14501
Canada	2022	Gaming	20	14884
Canada	2023	Entertainment	24	11964
India	2020	Entertainment	24	9529
India	2021	Entertainment	24	24812
India	2022	Entertainment	24	23144
India	2023	Entertainment	24	21311
Japan	2020	Entertainment	24	10223
Japan	2021	Entertainment	24	24268
Japan	2022	Entertainment	24	23526
Japan	2023	Entertainment	24	20160
Russia	2020	Entertainment	24	4005
Russia	2021	Entertainment	24	10298
Russia	2022	Gaming	20	11720
Russia	2023	Gaming	20	8894

```
| South Korea | 2020 | Entertainment | 24 | 7071
| South Korea | 2021 | Entertainment | 24 | 22744
| South Korea | 2022 | Entertainment | 24 | 21634
| South Korea | 2023 | Entertainment | 24 | 16993
| USA | 2020 | Music | 10 | 5705
| USA | 2021 | Entertainment | 24 | 14082
| USA | 2022 | Gaming | 20 | 14641
| USA | 2023 | Gaming | 20 | 11933
```

Get the file on Linux from HDFS: Switch on to first git-bash terminal to execute following command to download the output file(s) to Linux from HDFS

```
hdfs dfs -get YouTube/top_viewed_categories/000000_0 top_viewed_categories.csv
```

Copy the file to local PC: Open another terminal with git bash to download the file to local PC. Run the following command to copy the combined files to the local PC. You will be prompted for your credentials. Provide your password and then the file will be downloaded.

```
scp ssarkar4@129.153.66.218:/home/ssarkar4/top_viewed_categories.csv top_viewed_categories.csv
```

Run the following command to check if files are present:

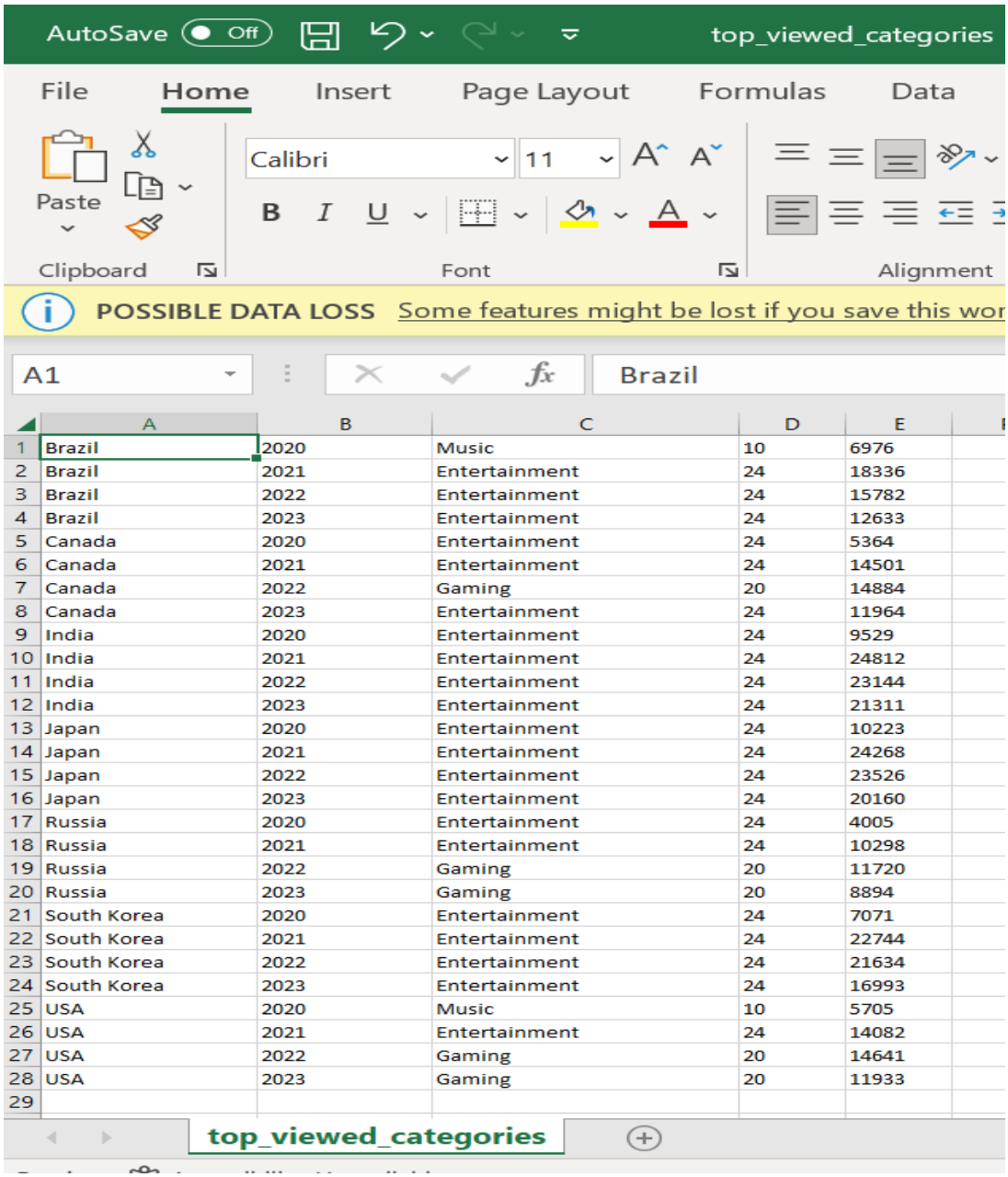
```
ls -al
```

```
-rw-r--r--   1 ssarkar4 ssarkar4       922 Dec 12 20:07 top_viewed_categories.csv
```

```
-bash-4.2$ ls -al
total 2551644
drwx------   7 ssarkar4 ssarkar4      4096 Dec 14 07:49 .
drwxr-xr-x. 92 root     root          4096 Nov  1 03:34 ..
-rw-------   1 ssarkar4 ssarkar4     19024 Dec 15 02:05 .bash_history
drwxr-xr-x   2 root     root            40 Nov  1 03:34 .beeline
-rw-r--r--   1 ssarkar4 ssarkar4 315460886 Dec 14 07:53 BR_youtube_trending_data.csv
drwxrwxr-x   3 ssarkar4 ssarkar4        18 Nov  2 01:47 .cache
-rw-r--r--   1 ssarkar4 ssarkar4       498 Dec 12 19:48 category.csv
-rw-r--r--   1 ssarkar4 ssarkar4 360314130 Dec  4 19:56 CA_youtube_trending_data.csv
drwxrwxr-x   3 ssarkar4 ssarkar4        18 Nov  2 01:47 .config
-rw-r--r--   1 ssarkar4 ssarkar4 371421031 Dec  4 19:37 IN_youtube_trending_data.csv
-rw-r--r--   1 ssarkar4 ssarkar4 479920557 Dec  4 19:38 JP_youtube_trending_data.csv
-rw-r--r--   1 ssarkar4 ssarkar4 280143719 Dec  4 19:39 KR_youtube_trending_data.csv
-rw-r--r--   1 ssarkar4 ssarkar4 447752282 Dec  4 19:41 RU_youtube_trending_data.csv
drwx------   2 ssarkar4 ssarkar4        25 Nov 16 05:17 .ssh
-rw-r--r--   1 ssarkar4 ssarkar4       504 Dec 13 07:18 top10_trending_videos.csv
-rw-r--r--   1 ssarkar4 ssarkar4       247 Dec  8 23:42 top_tags_frequency.csv
-rw-r--r--   1 ssarkar4 ssarkar4       403 Dec  9 19:49 top_trending_video_country_wise.csv
-rw-r--r--   1 ssarkar4 ssarkar4        12 Dec  9 19:49 .top_trending_video_country_wise.csv.crc
-rw-r--r--   1 ssarkar4 ssarkar4       922 Dec 12 20:07 top_viewed_categories.csv
-rw-r--r--   1 ssarkar4 ssarkar4       105 Dec  7 23:44 top_viewed_channels_by_years.csv
-rw-r--r--   1 ssarkar4 ssarkar4       983 Dec  7 21:24 top_viewed_channels.csv
-rw-r--r--   1 ssarkar4 ssarkar4 357792502 Dec  4 19:40 US_youtube_trending_data.csv
```

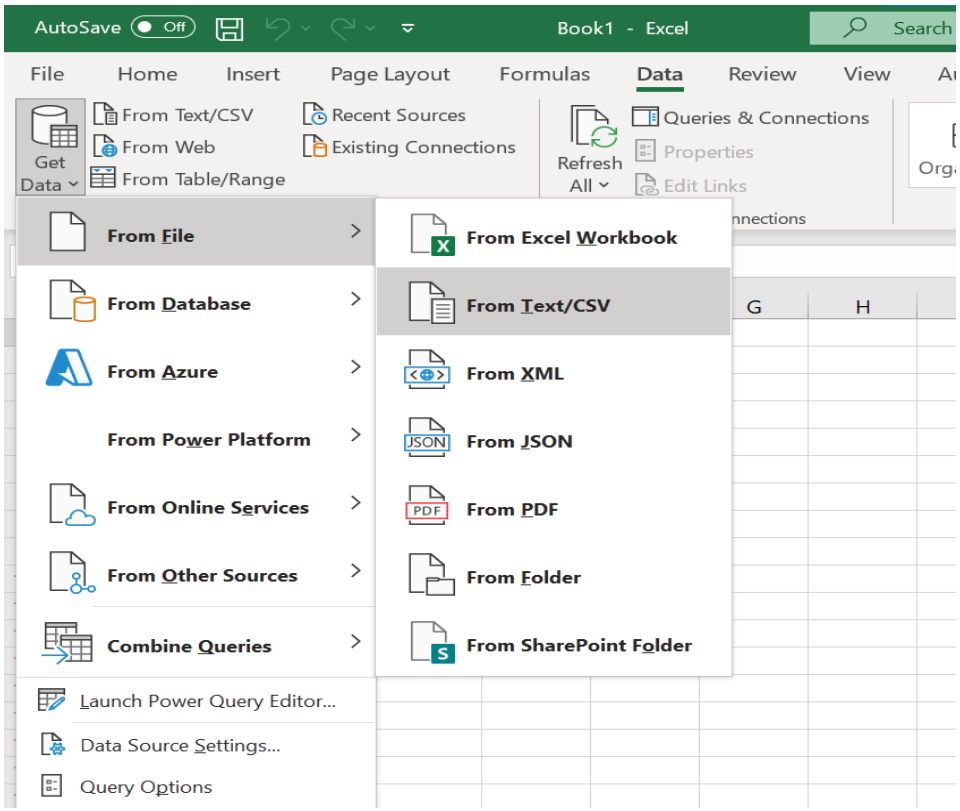
Visualization:

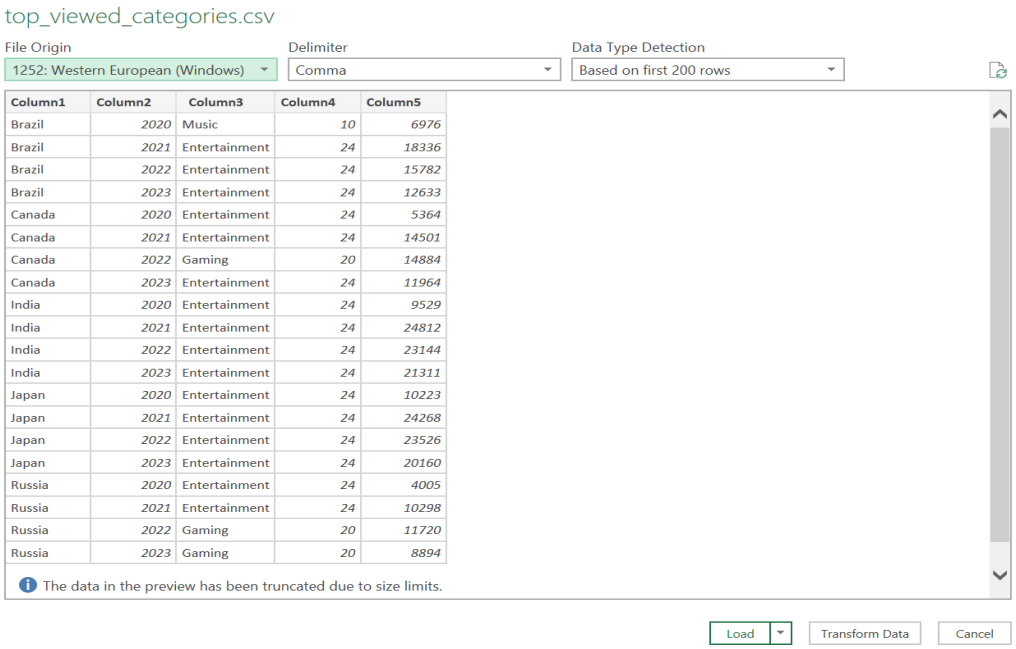
i. Open the top_viewed_categories.csv file from the location /home/ssarkar4/… and check if the content matches with the table **top_viewed_categories**.



	A	B	C	D	E
1	Brazil	2020	Music	10	6976
2	Brazil	2021	Entertainment	24	18336
3	Brazil	2022	Entertainment	24	15782
4	Brazil	2023	Entertainment	24	12633
5	Canada	2020	Entertainment	24	5364
6	Canada	2021	Entertainment	24	14501
7	Canada	2022	Gaming	20	14884
8	Canada	2023	Entertainment	24	11964
9	India	2020	Entertainment	24	9529
10	India	2021	Entertainment	24	24812
11	India	2022	Entertainment	24	23144
12	India	2023	Entertainment	24	21311
13	Japan	2020	Entertainment	24	10223
14	Japan	2021	Entertainment	24	24268
15	Japan	2022	Entertainment	24	23526
16	Japan	2023	Entertainment	24	20160
17	Russia	2020	Entertainment	24	4005
18	Russia	2021	Entertainment	24	10298
19	Russia	2022	Gaming	20	11720
20	Russia	2023	Gaming	20	8894
21	South Korea	2020	Entertainment	24	7071
22	South Korea	2021	Entertainment	24	22744
23	South Korea	2022	Entertainment	24	21634
24	South Korea	2023	Entertainment	24	16993
25	USA	2020	Music	10	5705
26	USA	2021	Entertainment	24	14082
27	USA	2022	Gaming	20	14641
28	USA	2023	Gaming	20	11933
29					

ii. Open a new Excel sheet. Click on Get Data from csv and select top_viewed_categories.csv as shown below:

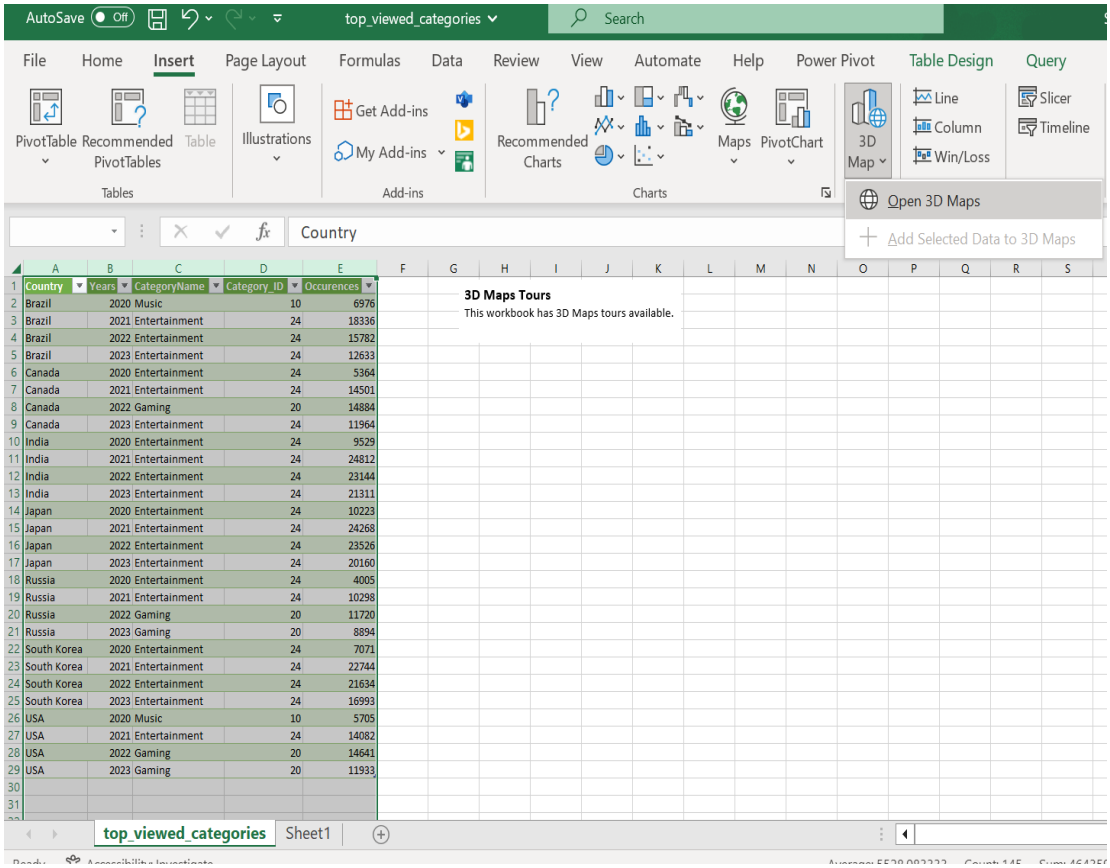




Column1	Column2	Column3	Column4	Column5
Brazil	2020	Music	10	6976
Brazil	2021	Entertainment	24	18336
Brazil	2022	Entertainment	24	15782
Brazil	2023	Entertainment	24	12633
Canada	2020	Entertainment	24	5364
Canada	2021	Entertainment	24	14501
Canada	2022	Gaming	20	14884
Canada	2023	Entertainment	24	11964
India	2020	Entertainment	24	9529
India	2021	Entertainment	24	24812
India	2022	Entertainment	24	23144
India	2023	Entertainment	24	21311
Japan	2020	Entertainment	24	10223
Japan	2021	Entertainment	24	24268
Japan	2022	Entertainment	24	23526
Japan	2023	Entertainment	24	20160
Russia	2020	Entertainment	24	4005
Russia	2021	Entertainment	24	10298
Russia	2022	Gaming	20	11720
Russia	2023	Gaming	20	8894

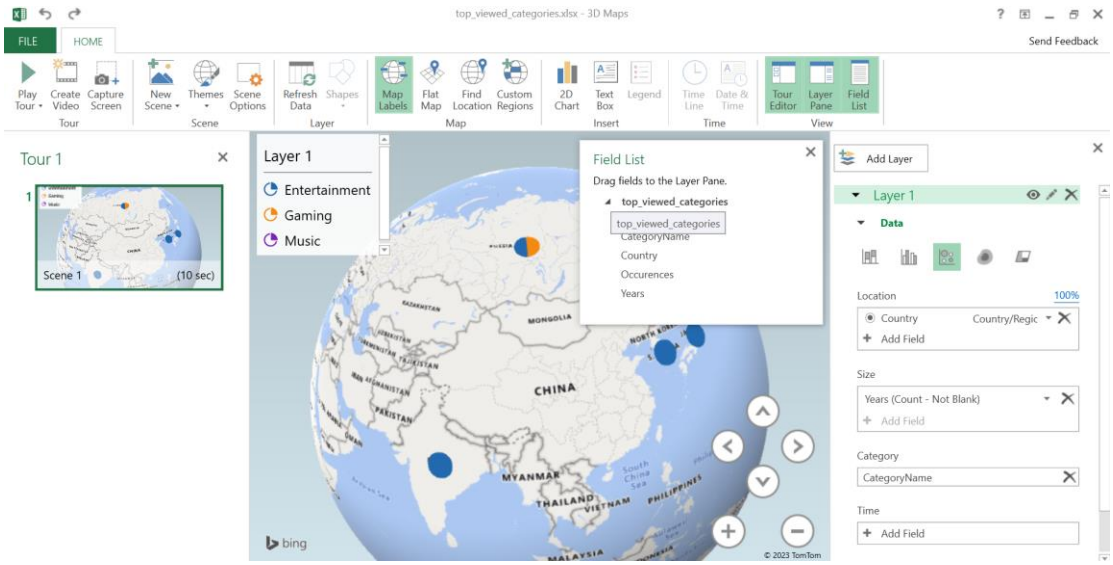
iii. After loading the data, rename the column as “Country”, “Years”, “CategoryName”, “Category_ID” and “Occurrences”.

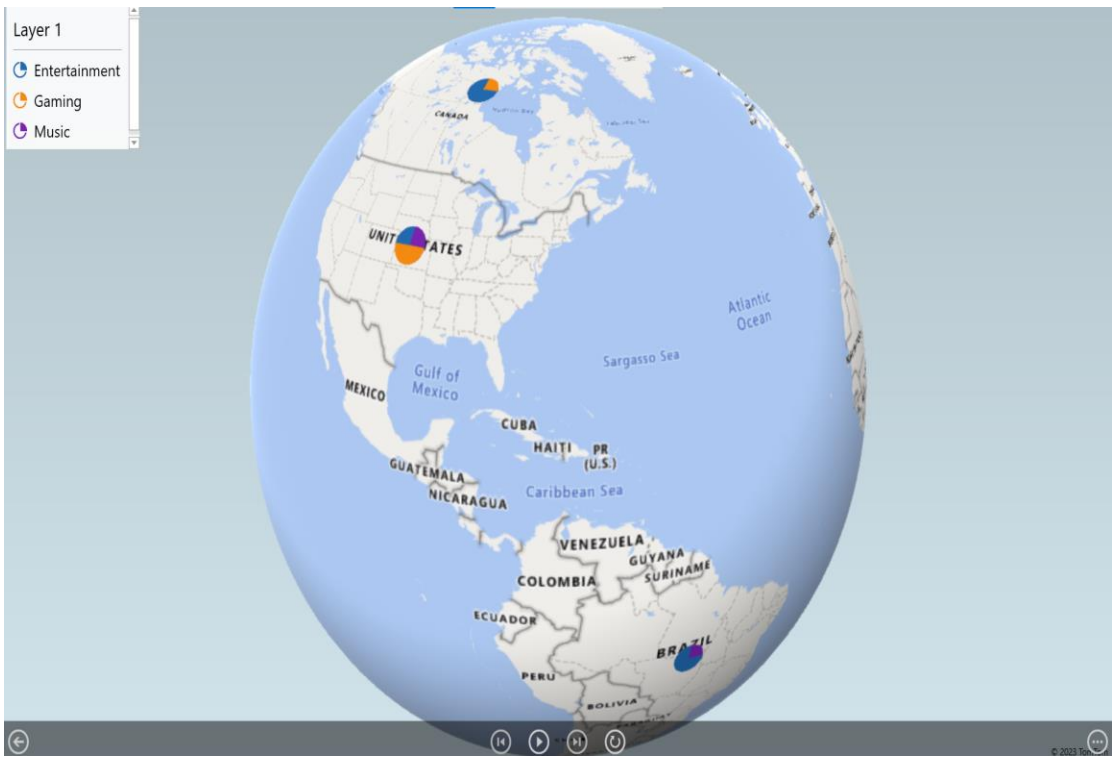
Then, save the file as top_viewed_categories.xlnx excel format. Select all the column, click on Insert, and then click on Open 3D Map.



Country	Years	CategoryName	Category_ID	Occurences
Brazil	2020	Music	10	6976
Brazil	2021	Entertainment	24	18336
Brazil	2022	Entertainment	24	15782
Brazil	2023	Entertainment	24	12633
Canada	2020	Entertainment	24	5364
Canada	2021	Entertainment	24	14501
Canada	2022	Gaming	20	14884
Canada	2023	Entertainment	24	11964
India	2020	Entertainment	24	9529
India	2021	Entertainment	24	24812
India	2022	Entertainment	24	23144
India	2023	Entertainment	24	21311
Japan	2020	Entertainment	24	10223
Japan	2021	Entertainment	24	24268
Japan	2022	Entertainment	24	23526
Japan	2023	Entertainment	24	20160
Russia	2020	Entertainment	24	4005
Russia	2021	Entertainment	24	10298
Russia	2022	Gaming	20	11720
Russia	2023	Gaming	20	8894
South Korea	2020	Entertainment	24	7071
South Korea	2021	Entertainment	24	22744
South Korea	2022	Entertainment	24	21634
South Korea	2023	Entertainment	24	16993
USA	2020	Music	10	5705
USA	2021	Entertainment	24	14082
USA	2022	Gaming	20	14641
USA	2023	Gaming	20	11933

iv. Select Country (Country/Region) in Location, CategoryName in category field and Year(Count Not Blank) in Size. This will provide us the visualization of different trending category of videos worldwide, along with the number of years the category is trending.





References

1. URL of the data source: [YouTube Trending Video Dataset (updated daily) (kaggle.com)]
2. URL of the GitHub: https://github.com/ssarkar4/YouTube_TrendingVideos_DataSet