

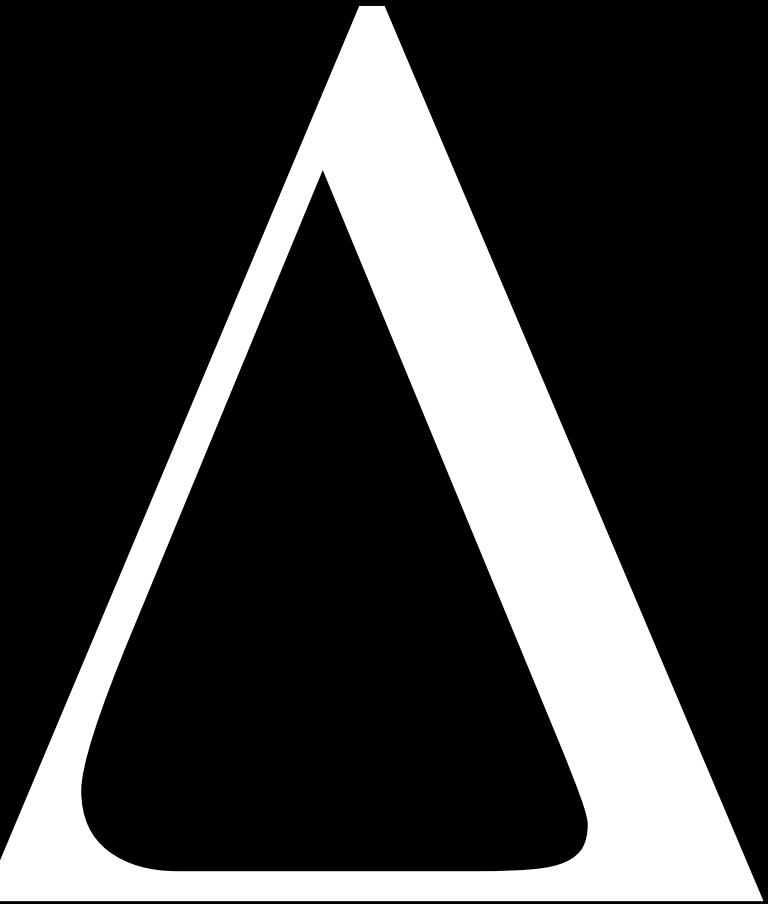
Team Delta

Luz del Cármén Vázquez

Liliana Vicario

Carlos Flores

Sergio Mejía

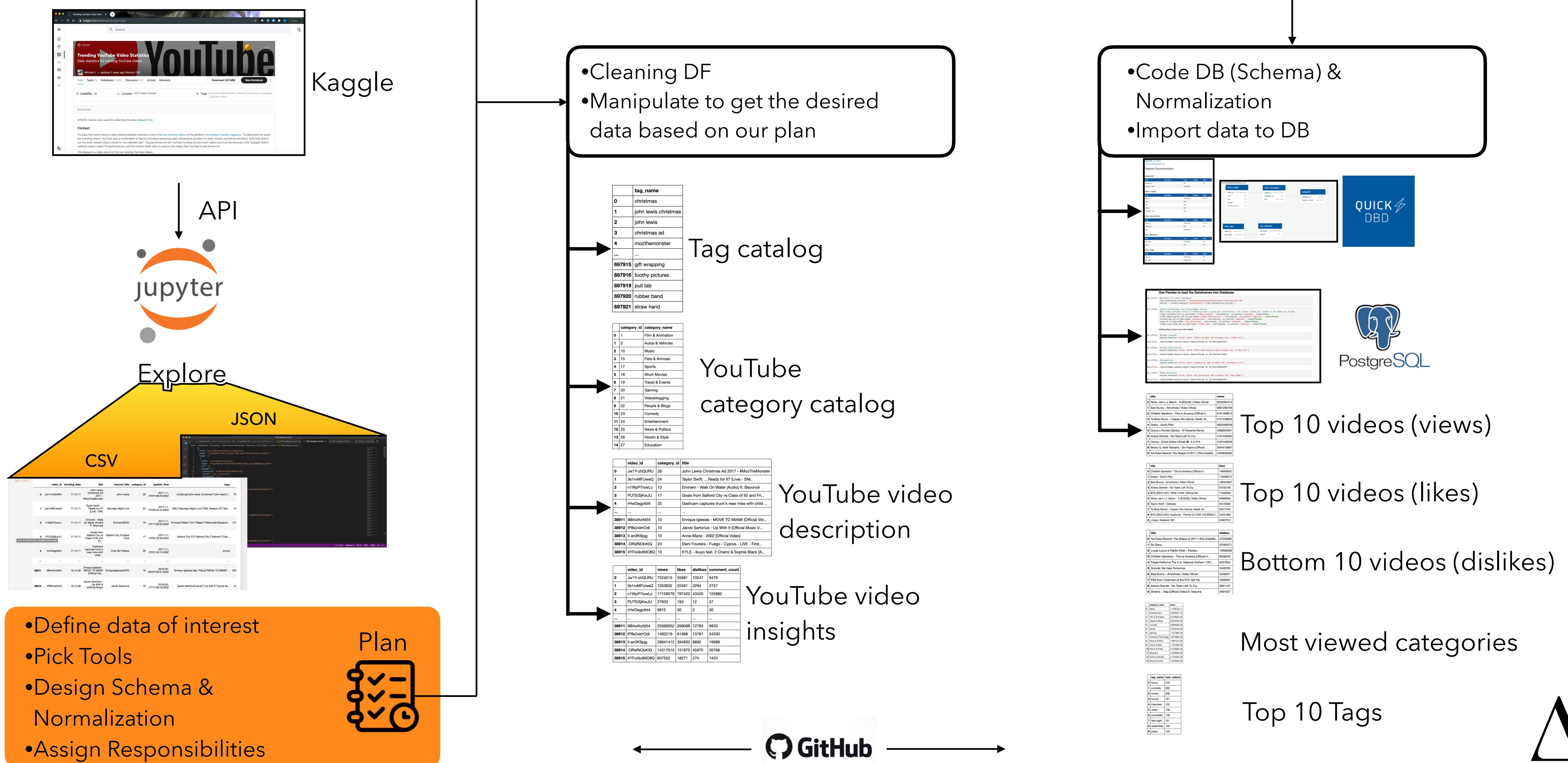


The Challenge & The Dataset

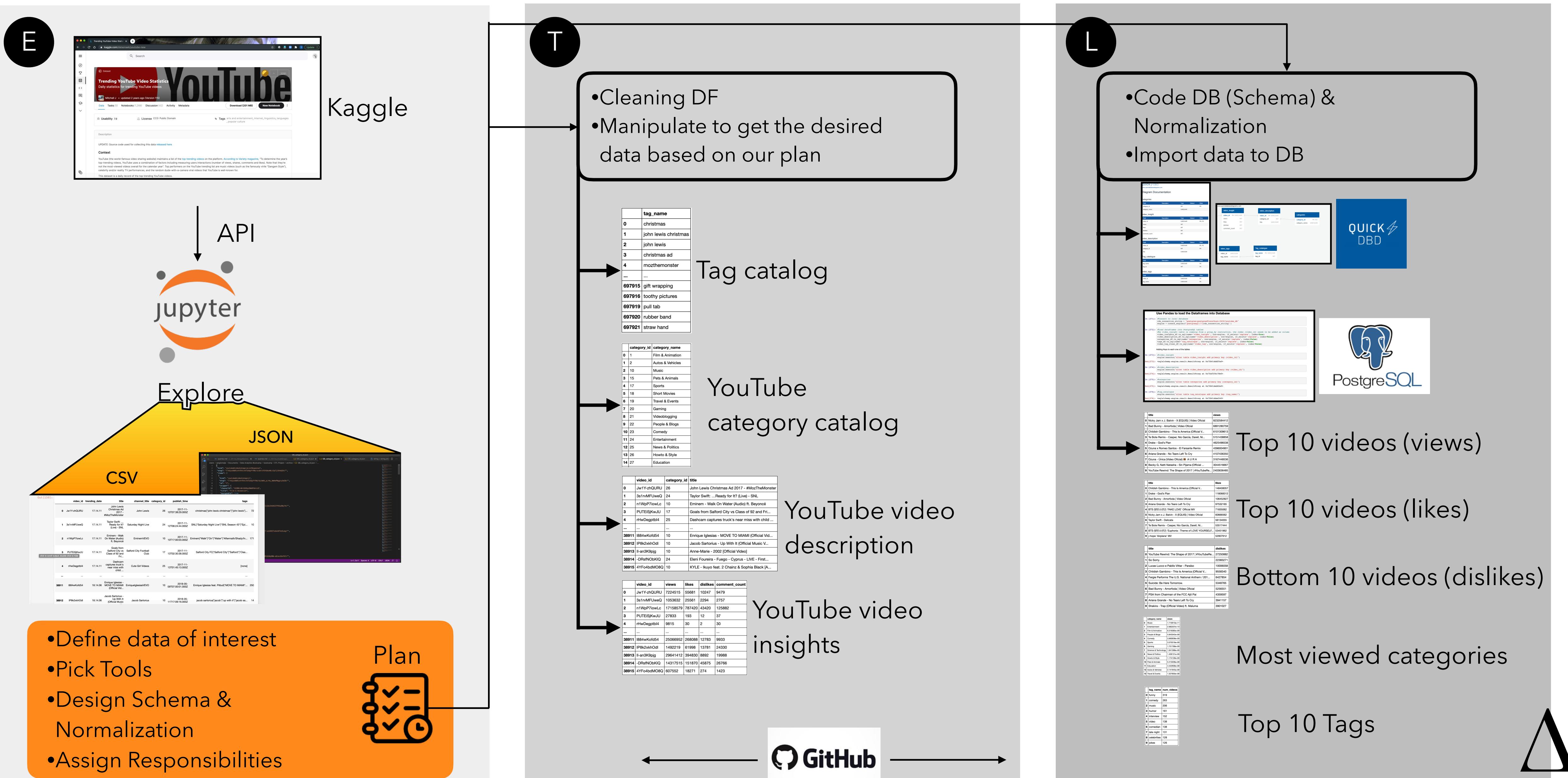
- As part of the ETL Challenge, we decided to work on a Youtube Dataset found in Kaggle.
- The dataset is called 'Trending YouTube Video Statistics':
 - YouTube maintains a list of the top trending videos on the platform
 - In order to determine which videos are trending, YouTube uses a combination of factors including user interactions (number of views, shares, comments and likes)
 - The dataset includes two different files for each country served:
 - CSV File containing video id, name, comments, tags, etc.
 - JSON File containing video category id and category title
- Originally we downloaded the files, however, Kaggle has an API that can be used to extract the data ... (Which we used)
- You can find the dataset here:
 - <https://www.kaggle.com/datasnaek/youtube-new>
 - Project goal: Transform the given datasets to simplify user consumption through queries based on tags, likes, dislikes, views & category id.



Our Process



Our Process



Challenges and Learning Experiences

- The biggest challenge we had as encoding and a changing set from Kaggle:
 - Encoding proved to be extremely hard to solve, initially we tried to work with Mexico's dataset, but special characters were extremely hard to work with (á, Á, é, É, í, Í, ó, Ó, ú, Ú, ñ, Ñ, ¡, ¿) were different throughout the dataset, and it also included characters in Portuguese, Korean, etc. considering the time, we decided to move to UK's Dataset.
 - Kaggle has an API and the dataset number keeps growing (Dataset from India was added yesterday and changed our project)
- We decided to use PostgreSQL because despite of the fact that one of the files has a json structure (and therefore made us consider use MongoDB) our info is structured, so we figured SQL would be best to use.
- Using Pandas to fully program Postgres was great (Kudos to the team), that helped us deliver a fully integrated Jupyter Notebook



Detailed Report

EXTRACT

Step by Step details: Extraction from Kaggle

```
In [1]: #Set up command lines with terminal tool
# This project need data set in Kaggle for this is necessary to install kaggle
!pip install kaggle
#Also reference in:
#https://github.com/Kaggle/kaggle-api documents and
#and https://www.youtube.com/watch?v=gkEbaMgvLs8
```

```
In [2]: # Kaggle requires to ensure kaggle.json (API token file) is in the location ~/.kaggle/kaggle.json to use the API
# You have to create an account or and API token in your account. This will download a fresh authentication token onto your machine (kaggle.json)
#Copy your token to downloads to the folder of kaggle (~./kaggle)
!cp -'your path download file kaggle.json' -'new path ~./kaggle'

cp: illegal option -- y
usage: cp [-R [-H | -L | -P]] [-fi | -n] [-apvXc] source_file target_file
        cp [-R [-H | -L | -P]] [-fi | -n] [-apvXc] source_file ... target_directory
```

```
In [3]: #Check file kaggle is the ~./kaggle path
!cd {'new path ~./kaggle'} && ls
```

Dependencies

```
In [4]: #Other dependencies
import pandas as pd
import os
from os import walk
from sqlalchemy import create_engine
```

```
In [5]: #path to download
path_download=os.path.join('Resources')
```

Import dataset from Kaggle.com

```
In [6]: #this line uses the kaggle.json of your account and authenticate
from kaggle.api.kaggle_api_extended import KaggleApi
api=KaggleApi()
api.authenticate()
```

```
In [7]: # Download all files of a dataset
# Signature: dataset_download_files(dataset, path=None, force=False, quiet=True, unzip=False)
api.dataset_download_files('datasnaek/youtube-new', path=path_download, force=False, quiet=True, unzip=True)
#A simple file:#Signature: dataset_download_file(dataset, file_name, path=None, force=False, quiet=True)
```



Step by Step details:
Exploration and
Dataframe
formation in Pandas

Forming Data Frames in Pandas

```
In [8]: #files in the data set:  
files=[]  
for (dirpath, dirnames, filenames) in walk(path_download):  
    files.append(filenames)  
    print(filenames)  
  
['IN_category_id.json', 'MXvideos.csv', 'RU_category_id.json', 'JP_category_id.json', 'FR_category_id.json', 'INvideos.csv',  
'GB_category_id.json', 'DEvideos.csv', 'JPvideos.csv', 'KR_category_id.json', 'KRvideos.csv', 'CAvideos.csv', 'MX_category_id.  
json', 'RUvideos.csv', 'FRvideos.csv', 'CA_category_id.json', 'US_category_id.json', 'DE_category_id.json', 'USvideos.csv', 'G  
Bvideos.csv']  
  
In [9]: #Number of files of data base in kaggle  
len(files[0])  
  
Out[9]: 20  
  
In [10]: #Show the first file  
files[0][0]  
  
Out[10]: 'IN_category_id.json'  
  
In [11]: #Print all name files  
for file in range(len(files[0])):  
    print(files)  
  
[['IN_category_id.json', 'MXvideos.csv', 'RU_category_id.json', 'JP_category_id.json', 'FR_category_id.json', 'INvideos.csv',  
'GB_category_id.json', 'DEvideos.csv', 'JPvideos.csv', 'KR_category_id.json', 'KRvideos.csv', 'CAvideos.csv', 'MX_category_id.  
json', 'RUvideos.csv', 'FRvideos.csv', 'CA_category_id.json', 'US_category_id.json', 'DE_category_id.json', 'USvideos.csv', 'G  
Bvideos.csv']]  
[['IN_category_id.json', 'MXvideos.csv', 'RU_category_id.json', 'JP_category_id.json', 'FR_category_id.json', 'INvideos.csv',  
'GB_category_id.json', 'DEvideos.csv', 'JPvideos.csv', 'KR_category_id.json', 'KRvideos.csv', 'CAvideos.csv', 'MX_category_id.  
json', 'RUvideos.csv', 'FRvideos.csv', 'CA_category_id.json', 'US_category_id.json', 'DE_category_id.json', 'USvideos.csv', 'G  
Bvideos.csv']]
```



Step by Step details: Automating Extraction

```
In [12]: #Creating Country value, to know the position of GB videos csv file in array result from Kaggle results
#Setting Country as zero by default
Country = 0
```

```
In [13]: # read files in pandas
files_csv=[] #Create frame for csv files
files_json=[] #Create frame fo json files

for file in range(len(files[0])):
    file_path=f'{path_download}/{files[0][file]}'
    try:
        files_json.append(pd.read_json(file_path))
        print(f'{files[0][file]} with json')
    except:
        try:
            files_csv.append(pd.read_csv(file_path, encoding='utf-8-sig'))
            print(f'{files[0][file]} with encoding utf-8')
        except:
            files_csv.append(pd.read_csv(file_path, encoding='latin1'))
            print(f'{files[0][file]} with encoding latin')

#Assigning length of files_csv if GB csv file found
if files[0][file].find('GBvideos.csv')>-1:
    Country = len(files_csv)-1
```

IN_category_id.json with json
MXvideos.csv with encoding latin
RU_category_id.json with json
JP_category_id.json with json
FR_category_id.json with json
INvideos.csv with encoding utf-8
GB_category_id.json with json
DEvideos.csv with encoding utf-8
JPvideos.csv with encoding latin
KR_category_id.json with json
KRvideos.csv with encoding latin
CAvideos.csv with encoding utf-8
MX_category_id.json with json
RUvideos.csv with encoding latin
FRvideos.csv with encoding utf-8
CA_category_id.json with json
US_category_id.json with json
DE_category_id.json with json
USvideos.csv with encoding utf-8
GBvideos.csv with encoding utf-8



Step by Step details: JSON

```
In [15]: files_json[0].head(1)
```

| Out[15]: | kind | etag | items |
|----------|-----------------------------------|---|--|
| 0 | youtube#videoCategoryListResponse | "XI7nbFXulYBipL0ayR_gDh3eu1k/1v2mrzYSYG6onNLt2... | {"kind": "youtube#videoCategory", "etag": "'XI...'"} |

```
In [16]: print(len(files_csv))
```

```
10
```

```
In [17]: files_csv[0].head(1)
```

| Out[17]: | video_id | trending_date | title | channel_title | category_id | publish_time | tags | views | likes | dislikes | comment_count | thumb... |
|----------|-------------|---------------|-------------------------------|-----------------|-------------|--------------------------|---|--------|-------|----------|---------------|---|
| 0 | SbOwzAl9ZfQ | 17.14.11 | Capítulo 12 MasterChef 2017 | MasterChef 2017 | 24 | 2017-11-13T06:06:22.000Z | MasterChef Junior 2017 "TV Azteca" "recetas" "... | 310130 | 4182 | 361 | 1836 | https://i... |

```
In [28]: files_json[Country]['items'][0]
```

```
Out[28]: {'kind': 'youtube#videoCategory',
          'etag': '"ld9biNPKjAjgjV7EZ4EKeEGrhao/Xy1mB4_yLrHy_BmKmPBggty2mZQ"',
          'id': '1',
          'snippet': {'channelId': 'UCBR8-60-B28hp2BmDPdntcQ',
                      'title': 'Film & Animation',
                      'assignable': True}}
```



Step by Step

details:

CSV

| | 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 |
|-------|-------------|---------------|------------------------------------|---------------|-------------|--------------------------|---|---------|--------|----------|---------------|---|-------------------|------------------|------------------------|---------------------------------|
| 8522 | --uXQQ0h-yQ | 17.26.12 | Paul McCartney - Pipes of Peace | PonteFractus | 24 | 2008-11-24T17:59:52.000Z | ww1 "great" "war" "trenches" "tommies" | 26299 | 234 | 3 | 11 | https://i.ytimg.com/vi/--uXQQ0h-yQ/default.jpg | False | False | False | Paul McCartney - Pipes of Peace |
| 9984 | --uXQQ0h-yQ | 18.02.01 | Paul McCartney - Pipes of Peace | PonteFractus | 24 | 2008-11-24T17:59:52.000Z | ww1 "great" "war" "trenches" "tommies" | 28725 | 253 | 3 | 12 | https://i.ytimg.com/vi/--uXQQ0h-yQ/default.jpg | False | False | False | Paul McCartney - Pipes of Peace |
| 8327 | --uXQQ0h-yQ | 17.25.12 | Paul McCartney - Pipes of Peace | PonteFractus | 24 | 2008-11-24T17:59:52.000Z | ww1 "great" "war" "trenches" "tommies" | 24723 | 216 | 3 | 10 | https://i.ytimg.com/vi/--uXQQ0h-yQ/default.jpg | False | False | False | Paul McCartney - Pipes of Peace |
| 9145 | --uXQQ0h-yQ | 17.29.12 | Paul McCartney - Pipes of Peace | PonteFractus | 24 | 2008-11-24T17:59:52.000Z | ww1 "great" "war" "trenches" "tommies" | 27689 | 243 | 3 | 11 | https://i.ytimg.com/vi/--uXQQ0h-yQ/default.jpg | False | False | False | Paul McCartney - Pipes of Peace |
| 8938 | --uXQQ0h-yQ | 17.28.12 | Paul McCartney - Pipes of Peace | PonteFractus | 24 | 2008-11-24T17:59:52.000Z | ww1 "great" "war" "trenches" "tommies" | 27413 | 241 | 3 | 11 | https://i.ytimg.com/vi/--uXQQ0h-yQ/default.jpg | False | False | False | Paul McCartney - Pipes of Peace |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| 37050 | zxwfDlhJlpw | 18.03.06 | kanye west / charlamagne interview | Kanye West | 22 | 2018-05-01T15:57:06.000Z | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... | 8579665 | 167841 | 19615 | 48621 | https://i.ytimg.com/vi/zxwfDlhJlpw/default.jpg | False | False | False | NaN |
| 35864 | zxwfDlhJlpw | 18.27.05 | kanye west / charlamagne interview | Kanye West | 22 | 2018-05-01T15:57:06.000Z | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... | 8473031 | 166781 | 19493 | 48494 | https://i.ytimg.com/vi/zxwfDlhJlpw/default.jpg | False | False | False | NaN |
| 36035 | zxwfDlhJlpw | 18.28.05 | kanye west / charlamagne interview | Kanye West | 22 | 2018-05-01T15:57:06.000Z | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... | 8486581 | 166889 | 19506 | 48506 | https://i.ytimg.com/vi/zxwfDlhJlpw/default.jpg | False | False | False | NaN |
| 37215 | zxwfDlhJlpw | 18.04.06 | kanye west / charlamagne interview | Kanye West | 22 | 2018-05-01T15:57:06.000Z | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... | 8596449 | 168026 | 19640 | 48633 | https://i.ytimg.com/vi/zxwfDlhJlpw/default.jpg | False | False | False | NaN |
| 34034 | zxwfDlhJlpw | 18.16.05 | kanye west / charlamagne interview | Kanye West | 22 | 2018-05-01T15:57:06.000Z | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... | 8295147 | 165136 | 19295 | 48115 | https://i.ytimg.com/vi/zxwfDlhJlpw/default.jpg | False | False | False | NaN |



Step by Step details: Tags

```
In [46]: pd.DataFrame(files_csv[Country][ 'tags' ].value_counts())
```

```
Out[46]:
```

[none]

James Corden|"The Late Late Show"|"Colbert"|"late night"|"late night show"|"Stephen Colbert"|"Comedy"|"monologue"|"comedian"|"impressions"|"celebrities"|"Show"|"Corden"|"joke"|"jokes"|"funny"|"funny video"|"funny videos"|"humor"|"celebrity"|"celeb"|"hollywood"|"famous"

BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"

ABC|"americanidol"|"idol"|"american idol"|"ryan"|"seacrest"|"ryan seacrest"|"katy"|"perry"|"katy perry"|"luke"|"bryan"|"luke bryan"|"lionel"|"richie"|"lionel richie"|"XVI"|"television"|"ad"|"spring"|"2018"|"music"|"reality"|"competition"|"song"|"sing"|"audition"|"auditions"|"performance"|"live"|"fox"|"AI"|"hollywood"|"contestant"|"week"|"hometown audition"

The Late Show|"Stephen Colbert"|"Colbert"|"Late Show"|"celebrities"|"late night"|"talk show"|"skits"|"bit"|"monologue"|"The Late Late Show"|"Late Late Show"|"letterman"|"comedian"|"impressions"|"CBS"|"joke"|"jokes"|"funny"|"funny video"|"funny videos"|"humor"|"celebrity"|"celeb"|"hollywood"|"famous"|"James Cord

...

Demi|"Lovato"|"addiction"|"live"|"interview"|"memoir"|"mental"|"health"|"Dianna"|"de"|"la"|"Garza"

jimmy|"kimmel"|"live"|"late"|"night"|"talk"|"show"|"funny"|"comedic"|"comedy"|"clip"|"comedian"|"charles"|"barkely"|"philadelphia"|"eagles"|"super"|"bowl"|"los"|"barkley"|"philadelphia eagles"|"super bowl"

avengers infinity war best scenes|"thor gets his hammer scene"|"avengers infinity scenes"|"avengers infinity war"|"thor scenes"|"Thor get its axe"

rudy mancuso maia mitchell magic official music video|"rudy"|"mancuso"|"maia"|"mitchell"|"magic"|"official"|"music"|"video"|"racist superman"|"domino"|"new"|"Bieber"|"Alesso"|"King Bach"|"lelepons"|"hannahstocking"|"rudymancuso"|"inanna"|"anwar"|"sarkis"|"shots"|"shotsstudios"|"alesso"|"anitta"|"brazil"

tim|"westwood"|"tv"|"freestyle"|"crib"|"sessions"|"capital"|"xtra"|"rap"|"hip"|"hop"|"grime"|"throwback"|"tbt"|"bryson tiller"|"R&B"|"trapsoul"|"God Tiller"|"dont"|"khaled"|"true to self"|"set it off"|"sza"

3122 rows × 1 columns



TRANSFORM

Step by Step details: Catalog Creation

Transform activities

Transforming extracted data to get the following DataFrames:

- Tag catalog
- YouTube Video category catalog
- YouTube Video description (video id, category id, video name, link)
- YouTube Video insights (likes, dislikes, # of views, category id)
- Tag - video id normalization relationship table: containing all video ids and their corresponding tag ids

Tag catalog & Tag normalization table

Tag Normalization table

```
In [56]: #We create a df with video ID and tag reference  
video_tag_df = files_csv[Country][['video_id','tags']]  
video_tag_df.sort_values(by='video_id')
```

Out[56]:

| | video_id | tags |
|-------|-------------|---|
| 8522 | --uXQQ0h-yQ | ww1 "great" "war" "trenches" "tommies" |
| 9984 | --uXQQ0h-yQ | ww1 "great" "war" "trenches" "tommies" |
| 8327 | --uXQQ0h-yQ | ww1 "great" "war" "trenches" "tommies" |
| 9145 | --uXQQ0h-yQ | ww1 "great" "war" "trenches" "tommies" |
| 8938 | --uXQQ0h-yQ | ww1 "great" "war" "trenches" "tommies" |
| ... | ... | ... |
| 37050 | zxwfDIhJIpw | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... |
| 35864 | zxwfDIhJIpw | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... |
| 36035 | zxwfDIhJIpw | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... |
| 37215 | zxwfDIhJIpw | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... |
| 34034 | zxwfDIhJIpw | Kanye West "YEEZY" "Kanye" "Charlamagne" "The ... |

38916 rows × 2 columns

Data cleaning: removing duplicates, NaN records and '[none]' value



Step by Step details:
Exploration and
Dataframe
formation in Pandas

Tag catalog

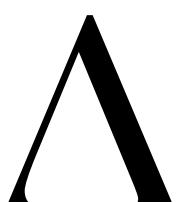
```
In [65]: #Creating Tag catalog using tags list
tags_df = pd.DataFrame({
    'tag_name': tags
})

#Removing duplicated records
tags_df.drop_duplicates(keep='first', inplace=True)
#Removing nan records
tags_df = tags_df.dropna()
tags_df
```

Out[65]:

| | tag_name |
|-------|----------------------|
| 0 | christmas |
| 1 | john lewis christmas |
| 2 | john lewis |
| 3 | christmas ad |
| 4 | mozthemonster |
| ... | ... |
| 64760 | gift wrapping |
| 64761 | toothy pictures |
| 64764 | pull tab |
| 64765 | rubber band |
| 64766 | straw hand |

31881 rows × 1 columns



Step by Step details:
Exploration and
Dataframe
formation in Pandas

YouTube video category catalog

In [66]:

```
#To create YouTube video category, we need to iterate on items value from JSON file
#Categories are stored in array, which will be used to create YouTube video category DataFrame
categories_ids = []
categories = []

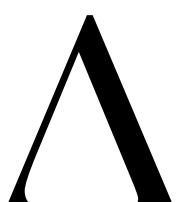
for category in files_json[Country]['items']:
    #print(category['snippet']['title'])
    categories_ids.append(category['id'])
    categories.append(category['snippet']['title'])

categories_df = pd.DataFrame( {
    'category_id': categories_ids,
    'category_name': categories
})

categories_df
```

Out[66]:

| | category_id | category_name |
|----|-------------|----------------------|
| 0 | 1 | Film & Animation |
| 1 | 2 | Autos & Vehicles |
| 2 | 10 | Music |
| 3 | 15 | Pets & Animals |
| 4 | 17 | Sports |
| 5 | 18 | Short Movies |
| 6 | 19 | Travel & Events |
| 7 | 20 | Gaming |
| 8 | 21 | Videoblogging |
| 9 | 22 | People & Blogs |
| 10 | 23 | Comedy |
| 11 | 24 | Entertainment |
| 12 | 25 | News & Politics |
| 13 | 26 | Howto & Style |
| 14 | 27 | Education |
| 15 | 28 | Science & Technology |



LOAD

Step by Step details:
Exploration and
Dataframe
formation in Pandas

Use Pandas to load the Dataframes into Database

```
In [73]: #Connect to local database
rds_connection_string = "postgres:***@localhost:5432/youtube_db"
engine = create_engine(f'postgresql:///{rds_connection_string}')

In [74]: #Load DataFrames into PostgreSQL tables
#As video_insight table is comming from a group_by instruction, the index (video_id) needs to be added as column
video_insights_df.to_sql(name='video_insight', con=engine, if_exists='replace', index=True)
video_description_df.to_sql(name='video_description', con=engine, if_exists='replace', index=False)
categories_df.to_sql(name='categories', con=engine, if_exists='replace', index=False)
tags_df.to_sql(name='tag_catalogue', con=engine, if_exists='replace', index=False)
video_tag_clean_df.to_sql(name='video_tag', con=engine, if_exists='replace', index=False)
```

Adding Keys to each one of the tables

```
In [75]: #video_insight
engine.execute("alter table video_insight add primary key (video_id)")

Out[75]: <sqlalchemy.engine.result.ResultProxy at 0x7f9f25524190>
```

```
In [76]: #video_description
engine.execute("alter table video_description add primary key (video_id)")

Out[76]: <sqlalchemy.engine.result.ResultProxy at 0x7f9f25524370>
```

```
In [77]: #categories
engine.execute("alter table categories add primary key (category_id)")

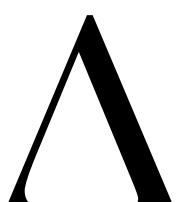
Out[77]: <sqlalchemy.engine.result.ResultProxy at 0x7f9f24a967c0>
```

```
In [78]: #tag_catalogue
engine.execute("alter table tag_catalogue add primary key (tag_name)")

Out[78]: <sqlalchemy.engine.result.ResultProxy at 0x7f9f24a966a0>
```

```
In [79]: #video_tag
engine.execute("alter table video_tag add primary key (video_id,tag_name)")

Out[79]: <sqlalchemy.engine.result.ResultProxy at 0x7f9f24a96bb0>
```



Step by Step details:
Exploration and
Dataframe
formation in Pandas

Queries

Using SQLAlchemy to explore data

```
In [80]: import sqlalchemy
from sqlalchemy.ext.automap import automap_base
from sqlalchemy.orm import Session
from sqlalchemy import create_engine, func, Column, Integer, String, Float, Table
from sqlalchemy.sql.expression import cast
```

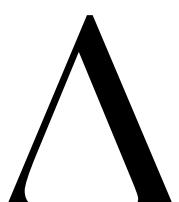
```
In [81]: Base = automap_base()
Base.prepare(engine, reflect=True)
```

```
In [82]: # We can view all of the classes that automap found
Base.classes.keys()
```

```
Out[82]: ['video_tag',
'video_insight',
'video_description',
'categories',
'tag_catalogue']
```

```
In [83]: #References to each one of the tables
Video_Insight = Base.classes.video_insight
Video_Description = Base.classes.video_description
Categories = Base.classes.categories
Tag_Catalogue = Base.classes.tag_catalogue
Video_Tag = Base.classes.video_tag
```

```
In [84]: # Create our session (link) from Python to the DB
session = Session(bind=engine)
```



Step by Step details:
Exploration and
Dataframe
formation in Pandas

Top 10 videos (more views)

It shows the top 10 videos in Great Britain

```
In [85]: #Getting the order by desc of 'views' column
top_videos = (session.query(Video_Description.title, Video_Insight.views).join(Video_Description,
                                         Video_Insight.video_id == Video_Description.video_id)
                                         .order_by(Video_Insight.views.desc()).limit(10))

#Convert it to DataFrame to display results
top_videos_df = pd.read_sql(top_videos.statement, engine )
top_videos_df
```

Out[85]:

| | title | views |
|---|---|------------|
| 0 | Nicky Jam x J. Balvin - X (EQUIS) Video Oficial | 8232584412 |
| 1 | Bad Bunny - Amorfoda Video Oficial | 6891280759 |
| 2 | Childish Gambino - This Is America (Official V... | 6101309613 |
| 3 | Te Bote Remix - Casper, Nio García, Darell, Ni... | 5151438858 |
| 4 | Drake - God's Plan | 4625496538 |
| 5 | Ozuna x Romeo Santos - El Farsante Remix | 4396004901 |
| 6 | Ariana Grande - No Tears Left To Cry | 4107436350 |
| 7 | Ozuna - Única (Video Oficial) 🐻 A U R A | 3167448536 |
| 8 | Becky G, Natti Natasha - Sin Pijama (Official ... | 3044516667 |
| 9 | YouTube Rewind: The Shape of 2017 #YouTubeRe... | 2403636485 |



Step by Step details:

Likes

| | title | likes |
|---|---|-----------|
| 0 | Childish Gambino - This Is America (Official V... | 148408057 |
| 1 | Drake - God's Plan | 118068312 |
| 2 | Bad Bunny - Amorfoda Video Oficial | 106452827 |
| 3 | Ariana Grande - No Tears Left To Cry | 97532165 |
| 4 | BTS (방탄소년단) 'FAKE LOVE' Official MV | 71835062 |
| 5 | Nicky Jam x J. Balvin - X (EQUIS) Video Oficial | 60666562 |
| 6 | Taylor Swift - Delicate | 56134055 |
| 7 | Te Bote Remix - Casper, Nio García, Darell, Ni... | 53517444 |
| 8 | BTS (방탄소년단) 'Euphoria : Theme of LOVE YOURSELF...' MV | 53451962 |
| 9 | j-hope 'Airplane' MV | 52907912 |

Dislikes

| | title | dislikes |
|---|--|----------|
| 0 | YouTube Rewind: The Shape of 2017 #YouTubeRewind | 27250882 |
| 1 | So Sorry. | 22383271 |
| 2 | Lucas Lucco e Pabllo Vittar - Paraíso | 10006058 |
| 3 | Childish Gambino - This Is America (Official V... | 9558340 |
| 4 | Fergie Performs The U.S. National Anthem / 201... | 6427854 |
| 5 | Suicide: Be Here Tomorrow. | 5348765 |
| 6 | Bad Bunny - Amorfoda Video Oficial | 5256551 |
| 7 | PSA from Chairman of the FCC Ajit Pai | 4359597 |
| 8 | Ariana Grande - No Tears Left To Cry | 3941157 |
| 9 | Shakira - Trap (Official Video) ft. Maluma | 3901027 |

Most Viewed Categories

| | category_name | views |
|----|----------------------|--------------|
| 0 | Music | 1.710913e+11 |
| 1 | Entertainment | 2.982207e+10 |
| 2 | Film & Animation | 8.319385e+09 |
| 3 | People & Blogs | 5.843342e+09 |
| 4 | Comedy | 3.960938e+09 |
| 5 | Sports | 3.370319e+09 |
| 6 | Gaming | 1.751768e+09 |
| 7 | Science & Technology | 1.641388e+09 |
| 8 | News & Politics | 1.409131e+09 |
| 9 | Howto & Style | 1.174128e+09 |
| 10 | Pets & Animals | 5.413206e+08 |
| 11 | Education | 4.442906e+08 |
| 12 | Autos & Vehicles | 2.141845e+08 |
| 13 | Travel & Events | 1.327850e+08 |



Step by Step details:

Top Ten Tags

| | tag_name | num_videos |
|---|-------------|------------|
| 0 | funny | 319 |
| 1 | comedy | 263 |
| 2 | music | 206 |
| 3 | humor | 161 |
| 4 | interview | 152 |
| 5 | video | 138 |
| 6 | comedian | 138 |
| 7 | late night | 131 |
| 8 | celebrities | 128 |
| 9 | jokes | 126 |

Videos with Random Chosen Tags

| | title | tag_name |
|-----|---|------------|
| 0 | JRE MMA Show #11 with John Danaher | comedian |
| 1 | Hawaii's Weekend Missile Mishap | comedian |
| 2 | Amy Schumer Is Ready to Host the Best SNL Yet | comedian |
| 3 | Christina Aguilera Carpool Karaoke - Extended Cut | comedian |
| 4 | New Hulu Show - SNL | comedian |
| ... | ... | ... |
| 416 | Tony Awards The Musical Starring Sara Bareille... | late night |
| 417 | Niecy Nash Has A Prince Story | late night |
| 418 | Shawn Mendes: Perfectly Wrong #LateLateShow | late night |
| 419 | Hilary Duff Is Having a Baby Girl and Her Son ... | late night |
| 420 | Drop the Mic w/ Ashton Kutcher & Sean Diddy Combs | late night |

