Youtube sentiment Analysis

Importing necessary libraries

In [120]:
```
import pandas as pd # Pandas for analyzing, cleaning, exploring and manipulating the data
import numpy as np # Numpy to work with arrays
import matplotlib.pyplot as plt # Data visualization library
import seaborn as sns # advance data visualization
import pandoc

import warnings
from warnings import filterwarnings
filterwarnings('ignore')
```

In [3]:
```
data = pd.read_csv(r'UScomments.csv', error_bad_lines = False)
```

```
Skipping line 41589: expected 4 fields, saw 11
Skipping line 51628: expected 4 fields, saw 7
Skipping line 114465: expected 4 fields, saw 5

Skipping line 142496: expected 4 fields, saw 8
Skipping line 189732: expected 4 fields, saw 6
Skipping line 245218: expected 4 fields, saw 7

Skipping line 388430: expected 4 fields, saw 5
```

- r/R used to create raw string The r before the string denotes a raw string literal in Python. This means that backslashes within the string are treated as literal backslashes, and not as escape characters. In this context, it ensures that the file path is interpreted correctly, though in this specific case, it isn't strictly necessary since there are no backslashes in the string.

- error_bad_line is used to Handle Errors. If the file contains rows that do not conform to the expected structure (e.g., a row has too many or too few columns), those rows will be skipped instead of causing the function to throw an error.

In [4]:
```
data.head()
```

Out[4]:

	video_id	comment_text	likes	replies
0	XpVt6Z1Gjjo	Logan Paul it's yo big day !!!!!	4	0
1	XpVt6Z1Gjjo	I've been following you from the start of your...	3	0
2	XpVt6Z1Gjjo	Say hi to Kong and maverick for me	3	0
3	XpVt6Z1Gjjo	MY FAN . attendance	3	0
4	XpVt6Z1Gjjo	trending 😊	3	0

- Data.head() prints top 5 rows of the DataFrame and name_of_df.tail() prints bottom 5 values of the rows you can change the number of rows to be seen by adding the number between the function for eg - name_of_df.head(10) will show use top 10 rows.

In [5]:
```
data.isnull().sum()
```

Out[5]:
```
video_id         0
comment_text    25
likes            0
replies          0
dtype: int64
```

- name_of_df.isnull().sum() is used to check the null values in the data frame in each coulmn

In [6]:
```
data.dropna(inplace=True)
```

- name_of_df.dropna(inplace = True) this function is used to drop the null values. 'inplace = True' is used for permanent change

In [7]:
```
data.isnull().sum()
```

Out[7]:
```
video_id        0
comment_text    0
likes           0
replies         0
dtype: int64
```

In [8]:
```
data.shape
```

Out[8]:
```
(691375, 4)
```

- name_of_df.shape -> is used to check number of rows and columns in a DataFrame

Data Processing

In [10]:
```
# !pip install textblob
```

In [6]:
```
from textblob import TextBlob
```

In [7]:
```
data.head(6)
```

Out[7]:

	video_id	comment_text	likes	replies
0	XpVt6Z1Gjjo	Logan Paul it's yo big day ‼️‼️‼️	4	0
1	XpVt6Z1Gjjo	I've been following you from the start of your...	3	0
2	XpVt6Z1Gjjo	Say hi to Kong and maverick for me	3	0
3	XpVt6Z1Gjjo	MY FAN . attendance	3	0
4	XpVt6Z1Gjjo	trending 😉	3	0
5	XpVt6Z1Gjjo	#1 on trending AYYEEEEE	3	0

In [8]:
```
TextBlob("Logan Paul it's yo big day ‼️‼️‼️").sentiment.polarity
```

Out[8]:
```
0.0
```

In [9]:
```
polarity = []

for comment in data['comment_text']:
    try:
        polarity.append(TextBlob(comment).sentiment.polarity)
    except:
        polarity.append(0)
```

In [10]:
```
len(polarity)
```

Out[10]:
```
691400
```

In [11]:
```
data['polarity'] = polarity
```

In [12]:
```
data.head()
```

Out[12]:

	video_id	comment_text	likes	replies	polarity
0	XpVt6Z1Gjjo	Logan Paul it's yo big day ‼️‼️‼️	4	0	0.0
1	XpVt6Z1Gjjo	I've been following you from the start of your...	3	0	0.0
2	XpVt6Z1Gjjo	Say hi to Kong and maverick for me	3	0	0.0
3	XpVt6Z1Gjjo	MY FAN . attendance	3	0	0.0
4	XpVt6Z1Gjjo	trending 😉	3	0	0.0

In [20]:
```
print(data['polarity'].unique())
```

```
[ 0.          0.8        -0.13571429 ...  0.38350313 -0.03787879
 -0.1155303 ]
```

- THe above line of codes is used to give a polarity to a sentences i.e sentiment to a sentence -1 polarity is for negative sentiment and 1 is for positive sentiment. To gice polarity to each sentence we used Textbblob library and its inbuilt functions

Word Cloud

In [19]:
```
# !pip install wordcloud
```

In [21]:
```
from wordcloud import WordCloud, STOPWORDS
```

In [22]:
```
len(set(STOPWORDS))
```

Out[22]:
```
192
```

- removing stops words. Stop words are common words that are often filtered out before processing textual data in various natural language processing (NLP) tasks. These words are considered to be of little value in terms of the overall meaning and context of the text. Common stop words include articles, prepositions, conjunctions, and pronouns such as "a," "an," "the," "and," "or," "but," "is," "in," "on," "at," etc.

In [22]:
```
type(data['comment_text'])
```

Out[22]:
```
pandas.core.series.Series
```

Wordcloud for positive words.

In [23]:
```
positive_comments = data[data['polarity'] == 1]
```

In [24]:
```
total_comments_positive = ' '.join(positive_comments['comment_text'])
```

In [118]:
```
# total_comments_positive
```

In [26]:
```
wordcloud = WordCloud(stopwords = set(STOPWORDS)).generate(total_comments_positive)
```

In [27]:
```
plt.imshow(wordcloud)
plt.axis('off')
```

Out[27]:
```
(-0.5, 399.5, 199.5, -0.5)
```

Wordcloud for negative comments

In [28]:
```
negative_comments = data[data['polarity']== -1]
```

In [29]:
```
total_comments_negative = ' '.join(negative_comments['comment_text'])
```

In [30]:
```
wordcloud2 =  WordCloud(stopwords = set(STOPWORDS)).generate(total_comments_negative)
```

In [31]:
```
plt.imshow(wordcloud2)
plt.axis('off')
```

Out[31]:
```
(-0.5, 399.5, 199.5, -0.5)
```

Emoji Analysis

In [32]:
```
# !pip install emoji==2.2.0
```

In [33]:
```
import emoji
```

In [34]:
```
emoji.__version__
```

Out[34]:
```
'2.2.0'
```

In [35]:
```
data['comment_text'].head()
```

Out[35]:
```
0                    Logan Paul it's yo big day ‼️‼️‼️
1    I've been following you from the start of your...
2                  Say hi to Kong and maverick for me
3                                MY FAN . attendance
4                                          trending 😉
Name: comment_text, dtype: object
```

In [36]:
```
comment = ' trending 😉'
```

In [37]:
```
[char for char in comment if char in emoji.EMOJI_DATA]
```

Out[37]:
```
['😉']
```

In [38]:
```
emoji_list = []

for comment in data['comment_text'].dropna():
    for char in comment:
        if char in emoji.EMOJI_DATA:
            emoji_list.append(char)
```

In [39]:
```
emoji_list[0:10]
```

Out[39]:
```
['‼', '‼', '‼', '😉', '😭', '👍', '', '', '😍', '💋']
```

In [40]:
```
from collections import Counter
```

In [41]:
```
Counter(emoji_list).most_common(10)
```

Out[41]:
```
[('😂', 36987),
 ('😍', 33453),
 ('❤', 31119),
 ('🔥', 8694),
 ('😭', 8398),
 ('👏', 5719),
 ('😘', 5545),
 ('👍', 5476),
 ('💖', 5359),
 ('💕', 5147)]
```

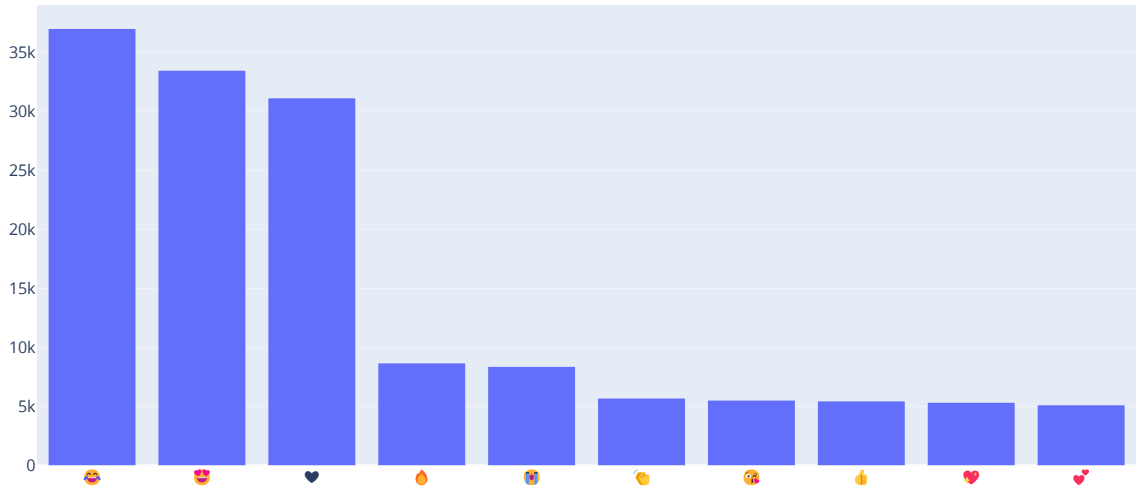
In [42]:
```
frequency = [Counter(emoji_list).most_common(10)[i][1] for i in range(10)]
```

In [43]:
```
emojis = [Counter(emoji_list).most_common(10)[i][0] for i in range(10)]
```

In [44]:
```
import plotly.graph_objs as go
from plotly.offline import iplot
```

In [45]:
```
trace =go.Bar(x=emojis, y=frequency)
```

In [46]:
```
iplot([trace])
```



Youtube Comments

In [48]:
```
import os
```

- The 'os' module in Python provides a way of using operating system dependent functionality like reading or writing to the file system, handling directories, executing system commands, and more. It is part of the standard library, so it comes with Python and does not need to be installed separately.

In [53]:
```
files = os.listdir(r'C:\Users\Atharva\Desktop\aaaaaaa\Study\Data Analysis Course\additional_data')
```

- The os.listdir function in Python is used to list all files and directories in a specified directory.

In [54]:
```
files
```

Out[54]:
```
['CAvideos.csv',
 'CA_category_id.json',
 'DEvideos.csv',
 'DE_category_id.json',
 'FRvideos.csv',
 'FR_category_id.json',
 'GBvideos.csv',
 'GB_category_id.json',
 'INvideos.csv',
 'IN_category_id.json',
 'JPvideos.csv',
 'JP_category_id.json',
 'KRvideos.csv',
 'KR_category_id.json',
 'MXvideos.csv',
 'MX_category_id.json',
 'RUvideos.csv',
 'RU_category_id.json',
 'USvideos.csv',
 'US_category_id.json',
 'YTdc.sqlite']
```

In [55]:
```
files_csv = [file for file in files if '.csv' in file]
```

In [56]:
```
files_csv
```

Out[56]:
```
['CAvideos.csv',
 'DEvideos.csv',
 'FRvideos.csv',
 'GBvideos.csv',
 'INvideos.csv',
 'JPvideos.csv',
 'KRvideos.csv',
 'MXvideos.csv',
 'RUvideos.csv',
 'USvideos.csv']
```

In [57]:
```
full_df = pd.DataFrame() # Creating a empty data drame to concat all the dataframes
path = r'C:\Users\Atharva\Desktop\aaaaaaa\Study\Data Analysis Course\additional_data'

for file in files_csv:
    current_df = pd.read_csv(path+'/'+file, encoding='iso-8859-1', error_bad_lines = False)

    full_df = pd.concat([full_df , current_df], ignore_index = True)
```

- 'ignore_index = True' ignore the existing row indices of the DataFrames and to reindex the resulting DataFrame. When ignore_index=True, the resulting DataFrame will have a new integer index that ranges from 0 to n-1, where n is the total number of rows in the concatenated DataFrame.

In [58]:
```
full_df.shape
```

Out[58]:
```
(375942, 16)
```

In [60]:
```
full_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 375942 entries, 0 to 375941
Data columns (total 16 columns):
 #   Column                  Non-Null Count   Dtype
---  ------                  --------------   -----
 0   video_id                375942 non-null  object
 1   trending_date           375942 non-null  object
 2   title                   375942 non-null  object
 3   channel_title           375942 non-null  object
 4   category_id             375942 non-null  int64
 5   publish_time            375942 non-null  object
 6   tags                    375942 non-null  object
 7   views                   375942 non-null  int64
 8   likes                   375942 non-null  int64
 9   dislikes                375942 non-null  int64
 10  comment_count           375942 non-null  int64
 11  thumbnail_link          375942 non-null  object
 12  comments_disabled       375942 non-null  bool
 13  ratings_disabled        375942 non-null  bool
 14  video_error_or_removed  375942 non-null  bool
 15  description             356464 non-null  object
dtypes: bool(3), int64(5), object(8)
memory usage: 38.4+ MB
```

- The df.info() method in pandas is used to get a concise summary of a DataFrame. This method provides important details about the DataFrame, including the index dtype and column dtypes, non-null values, and memory usage. It is particularly useful for quickly understanding the structure and quality of your data.

In [61]:
```
full_df.describe()
```

Out[61]:

	category_id	views	likes	dislikes	comment_count
count	375942.000000	3.759420e+05	3.759420e+05	3.759420e+05	3.759420e+05
mean	20.232302	1.326568e+06	3.788431e+04	2.126107e+03	4.253775e+03
std	7.132413	7.098568e+06	1.654131e+05	2.248437e+04	2.545876e+04
min	1.000000	1.170000e+02	0.000000e+00	0.000000e+00	0.000000e+00
25%	17.000000	4.697800e+04	6.690000e+02	4.100000e+01	1.090000e+02
50%	23.000000	1.773705e+05	3.446000e+03	1.790000e+02	5.110000e+02
75%	24.000000	6.476792e+05	1.747650e+04	7.490000e+02	2.011000e+03
max	44.000000	4.245389e+08	5.613827e+06	1.944971e+06	1.626501e+06

- The df.describe() function in pandas is used to generate descriptive statistics of a DataFrame. It provides a summary of the central tendency, dispersion, and shape of a dataset's distribution, excluding NaN values. This function is particularly useful for quickly getting an overview of numeric data in a DataFrame.

In [62]:
```
full_df.duplicated().sum()
```

Out[62]:
```
36417
```

df.duplicated() function is used to check duplicated values in the data frames

In [63]:
```
full_df[full_df.duplicated()].shape
```

Out[63]:
```
(36417, 16)
```

In [64]:
```
full_df = full_df.drop_duplicates()
```

df.drop_duplicated() is used to remove the duplicated values from the data frame

In [65]:
```
full_df.shape
```

Out[65]:
```
(339525, 16)
```

In [67]:
```
path = r'C:\Users\Atharva\Desktop\aaaaaaa\Study\Data Analysis Course\additional_data'
full_df.to_csv(f'{path}\youtube_sample.csv', index = False)
```

df.to_csv exports the created dataframe to our desired path in csv format

In [69]:
```
full_df.to_json(f'{path}\youtube_sample.json')
```

df.to_json exports the created dataframe to our desired path in json format

In [70]:
```
from sqlalchemy import create_engine
```

In [71]:
```
engine = create_engine(f'sqlite:///{path}\YTdc.sqlite')
```

In [63]:
```
# full_df[0:1000].to_sql('Users', con = engine, if_exists = 'append')
```

In [72]:
```
full_df = pd.read_csv(r'additional_data\youtube_sample.csv')
```

In [73]:
```
full_df.head()
```

Out[73]:

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	l
0	n1WpP7iowLc	17.14.11	Eminem - Walk On Water (Audio) ft. BeyoncÃ©	EminemVEVO	10	2017-11-10T17:00:03.000Z	Eminem|"Walk"|"On"|"Water"|"Aftermath/Shady/In...	17158579	787
1	0dBIkQ4Mz1M	17.14.11	PLUSH - Bad Unboxing Fan Mail	iDubbbzTV	23	2017-11-13T17:00:00.000Z	plush|"bad unboxing"|"unboxing"|"fan mail"|"id...	1014651	127
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	racist superman|"rudy"|"mancuso"|"king"|"bach"...	3191434	146
			I Dare						

Below we will extract category title using data manipluation

In [74]:
```
full_df['category_id'].unique()
```

Out[74]:
```
array([10, 23, 24, 25, 22, 26,  1, 28, 20, 17, 29, 15, 19,  2, 27, 43, 30,
       44], dtype=int64)
```

As we can see above there are videos with category id but without thier names and its hard to understand the category of the video by thier ids.

In [77]:
```
json_df = pd.read_json(fr'{path}\US_category_id.json')
```

in the above cell we took one jsaon dataframe to extract category name from the dictionary you can take any other category data frame present in the data files

In [79]:
```
json_df
```

Out[79]:

	kind	etag	items
0	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
1	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
2	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
3	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
4	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
5	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
6	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
7	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
8	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
9	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
10	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
11	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
12	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
13	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
14	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
15	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
16	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
17	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
18	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
19	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
20	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
21	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
22	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
23	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
24	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
25	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
26	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
27	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
28	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
29	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
30	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...
31	youtube#videoCategoryListResponse	"m2yskBQFythfE4irbTIeOgYYfBU/S730Ilt-Fi-emsQJv...	{'kind': 'youtube#videoCategory', 'etag': '"m2...

In [9]:
```
json_df['items'][1]
```

Out[9]:
```
{'kind': 'youtube#videoCategory',
 'etag': '"m2yskBQFythfE4irbTIeOgYYfBU/UZ1oLIIz2dxIhO45ZTFR3a3NyTA"',
 'id': '2',
 'snippet': {'channelId': 'UCBR8-60-B28hp2BmDPdntcQ',
  'title': 'Autos & Vehicles',
  'assignable': True}}
```

- we can see that the category name is in the items column in our dataframe and is in dictionary. to access the dictonary we will have to manipulate it and extract our desired category name

In [82]:
```
cat_dict = {} # creating dictionary to store category name and id

for item in json_df['items'].values:
    cat_dict[ int(item["id"])] = item['snippet']['title']

cat_dict
```

Out[82]:
```
{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',
 29: 'Nonprofits & Activism',
 30: 'Movies',
 31: 'Anime/Animation',
 32: 'Action/Adventure',
 33: 'Classics',
 34: 'Comedy',
 35: 'Documentary',
 36: 'Drama',
 37: 'Family',
 38: 'Foreign',
 39: 'Horror',
 40: 'Sci-Fi/Fantasy',
 41: 'Thriller',
 42: 'Shorts',
 43: 'Shows',
 44: 'Trailers'}
```

In [83]:
```
full_df['category_name'] = full_df['category_id'].map(cat_dict) # Creating a new column and mapping the title to its desired
```

In [87]:
```
full_df[['category_id','category_name']].head()
```

Out[87]:

	category_id	category_name
0	10	Music
1	23	Comedy
2	23	Comedy
3	24	Entertainment
4	10	Music

Now its easier for us to understand the type/category of the video and we successfully were able to extract the category name from the table using data manipulation

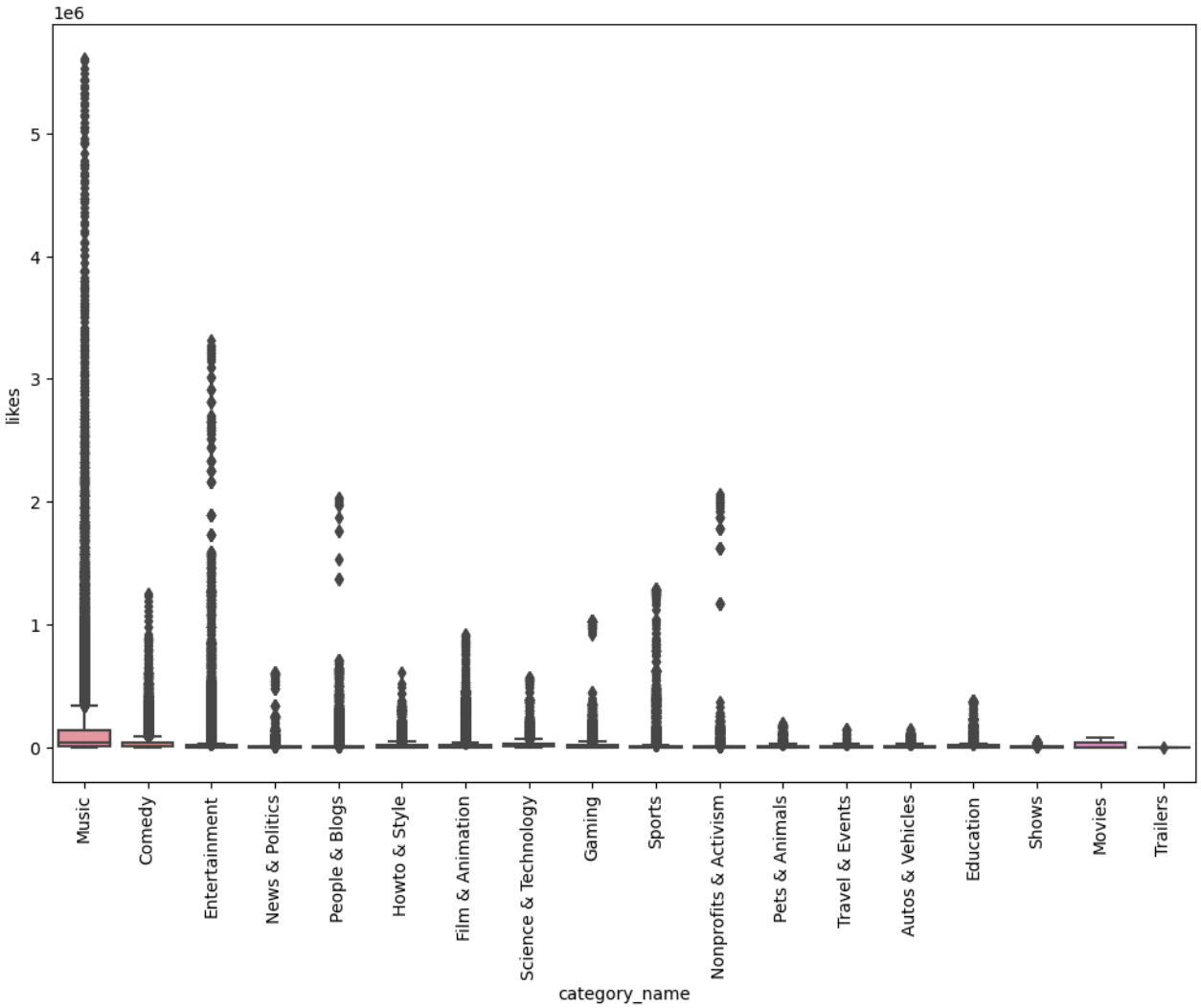
Data visualization

In [88]:
```
full_df['likes'].describe()
```

Out[88]:
```
count    3.395250e+05
mean     3.454187e+04
std      1.528666e+05
min      0.000000e+00
25%      6.040000e+02
50%      3.083000e+03
75%      1.542400e+04
max      5.613827e+06
Name: likes, dtype: float64
```

In [89]:
```
plt.figure(figsize=(12,8))
sns.boxplot(x='category_name', y='likes', data = full_df)
plt.xticks(rotation='vertical')
```

Out[89]:
```
(array([ 0,  1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14, 15, 16,
        17]),
 [Text(0, 0, 'Music'),
  Text(1, 0, 'Comedy'),
  Text(2, 0, 'Entertainment'),
  Text(3, 0, 'News & Politics'),
  Text(4, 0, 'People & Blogs'),
  Text(5, 0, 'Howto & Style'),
  Text(6, 0, 'Film & Animation'),
  Text(7, 0, 'Science & Technology'),
  Text(8, 0, 'Gaming'),
  Text(9, 0, 'Sports'),
  Text(10, 0, 'Nonprofits & Activism'),
  Text(11, 0, 'Pets & Animals'),
  Text(12, 0, 'Travel & Events'),
  Text(13, 0, 'Autos & Vehicles'),
  Text(14, 0, 'Education'),
  Text(15, 0, 'Shows'),
  Text(16, 0, 'Movies'),
  Text(17, 0, 'Trailers')])
```

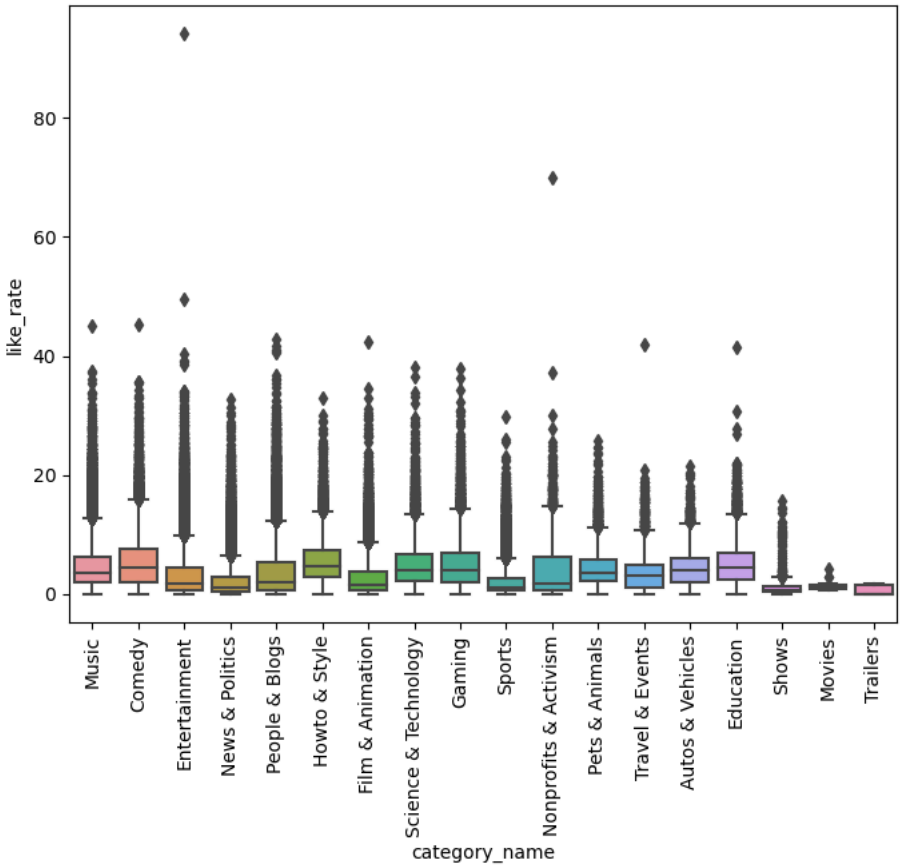


In [93]:
```
full_df['like_rate'] =  (full_df['likes']/full_df['views']) * 100
full_df['dislike_rate'] = (full_df['dislikes']/full_df['views']) * 100
full_df['comment_count_rate'] = (full_df['comment_count']/full_df['views']) * 100
```

In [94]:
```
full_df.columns
```

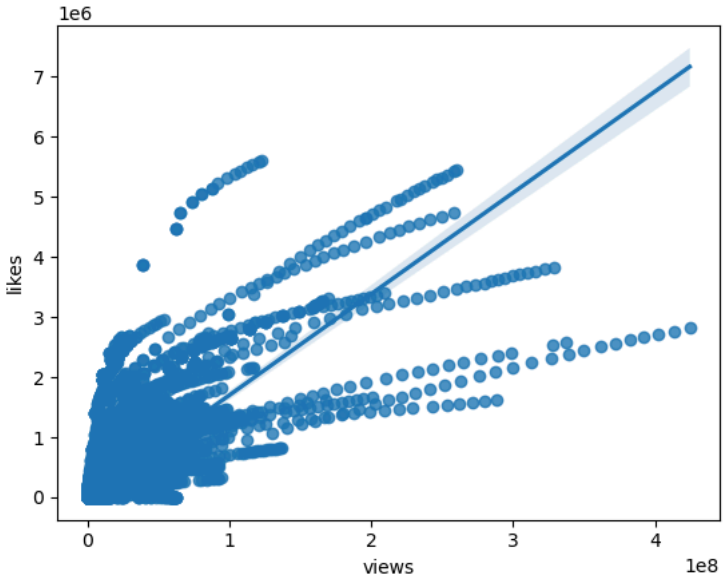
Out[94]:
```
Index(['video_id', 'trending_date', 'title', 'channel_title', 'category_id',
       'publish_time', 'tags', 'views', 'likes', 'dislikes', 'comment_count',
       'thumbnail_link', 'comments_disabled', 'ratings_disabled',
       'video_error_or_removed', 'description', 'category_name', 'like_rate',
       'dislike_rate', 'comment_count_rate'],
      dtype='object')
```

In [95]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='category_name', y='like_rate', data = full_df)
plt.xticks(rotation='vertical')
plt.show()
```



In [96]:
```
sns.regplot(x='views',y='likes', data= full_df)
```

Out[96]:
```
<Axes: xlabel='views', ylabel='likes'>
```



In [97]:
```
full_df.columns
```

Out[97]:
```
Index(['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', 'category_name', 'like_rate',
       'dislike_rate', 'comment_count_rate'],
      dtype='object')
```

In [98]:
```
full_df[['views', 'likes', 'dislikes']].corr()
```

Out[98]:

	views	likes	dislikes
views	1.000000	0.779531	0.405428
likes	0.779531	1.000000	0.451809
dislikes	0.405428	0.451809	1.000000

In [99]:
```
sns.heatmap(full_df[['views', 'likes', 'dislikes']].corr(), annot = True)
```

Out[99]: <Axes: >

In [100]:
```
full_df.head(6)
```

Out[100]:

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes
0	n1WpP7iowLc	17.14.11	Eminem - Walk On Water (Audio) ft. BeyoncÃ©	EminemVEVO	10	2017-11-10T17:00:03.000Z	Eminem|"Walk"|"On"|"Water"|"Aftermath/Shady/In...	17158579	787425
1	0dBIkQ4Mz1M	17.14.11	PLUSH - Bad Unboxing Fan Mail	iDubbbzTV	23	2017-11-13T17:00:00.000Z	plush|"bad unboxing"|"unboxing"|"fan mail"|"id...	1014651	127794
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	racist superman|"rudy"|"mancuso"|"king"|"bach"...	3191434	146035
3	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga	24	2017-11-12T18:01:41.000Z	ryan|"higa"|"higatv"|"nigahiga"|"i dare you"|"...	2095828	132239
4	2Vv-BfVoq4g	17.14.11	Ed Sheeran - Perfect (Official Music Video)	Ed Sheeran	10	2017-11-09T11:04:14.000Z	edsheeran|"ed sheeran"|"acoustic"|"live"|"cove...	33523622	1634130
5	0yIWz1XEeyc	17.14.11	Jake Paul Says Alissa Violet CHEATED with LOGA...	DramaAlert	25	2017-11-13T07:37:51.000Z	#DramaAlert|"Drama"|"Alert"|"DramaAlert"|"keem...	1309699	103755

In [101]:
```
full_df['channel_title'].value_counts()
```

Out[101]:
```
The Late Show with Stephen Colbert    710
WWE                                   643
Late Night with Seth Meyers           592
TheEllenShow                          555
Jimmy Kimmel Live                     528
                                     ...
Daas                                    1
YT Industries                           1
BTLV Le mÃ©dia complÃ©mentaire           1
Quem Sabia ?                            1
Jessi Osorno                            1
Name: channel_title, Length: 37824, dtype: int64
```

In [102]:
```
top_20_channels = full_df.groupby(['channel_title']).size().sort_values(ascending = False).reset_index().head(20)
```

In [103]:
```
top_20_channels
```

Out[103]:

	channel_title	0
0	The Late Show with Stephen Colbert	710
1	WWE	643
2	Late Night with Seth Meyers	592
3	TheEllenShow	555
4	Jimmy Kimmel Live	528
5	PewDiePie	511
6	The Tonight Show Starring Jimmy Fallon	509
7	CNN	500
8	The Late Late Show with James Corden	453
9	ESPN	452
10	FBE	439
11	VikatanTV	435
12	Netflix	410
13	SET India	405
14	MLG Highlights	382
15	BuzzFeedVideo	361
16	SMTOWN	359
17	Ã□ukur	356
18	Marvel Entertainment	352
19	SAB TV	351

In [104]:
```
top_20_channels.rename(columns = {0:'total_videos'}, inplace = True)
```

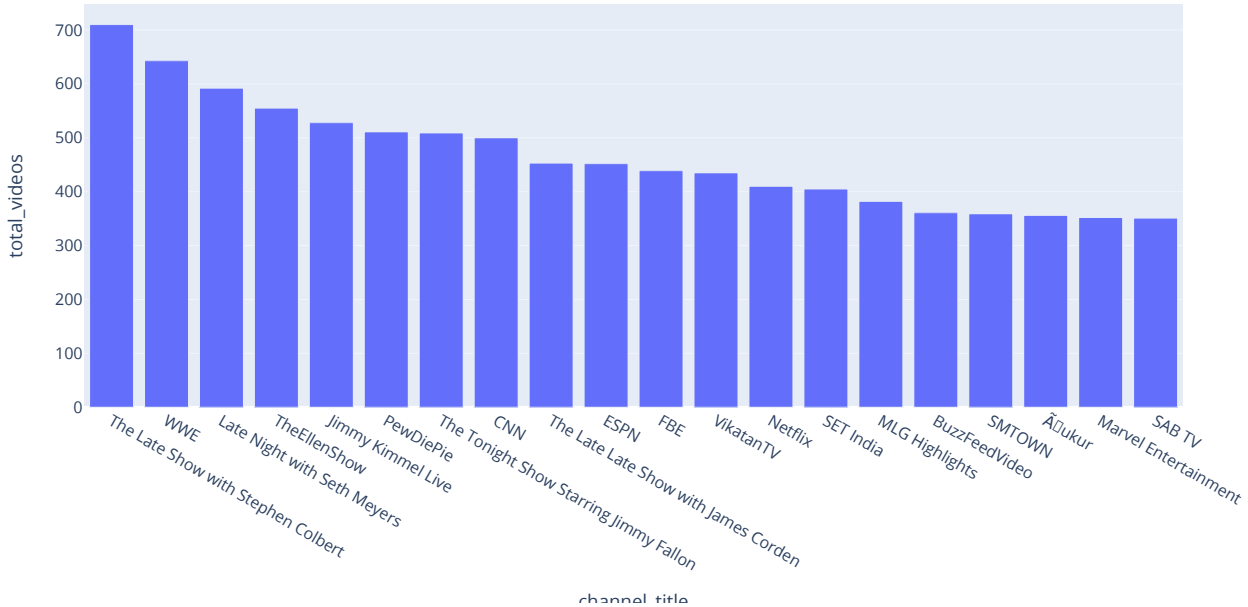
In [105]:
```
top_20_channels
```

Out[105]:

	channel_title	total_videos
0	The Late Show with Stephen Colbert	710
1	WWE	643
2	Late Night with Seth Meyers	592
3	TheEllenShow	555
4	Jimmy Kimmel Live	528
5	PewDiePie	511
6	The Tonight Show Starring Jimmy Fallon	509
7	CNN	500
8	The Late Late Show with James Corden	453
9	ESPN	452
10	FBE	439
11	VikatanTV	435
12	Netflix	410
13	SET India	405
14	MLG Highlights	382
15	BuzzFeedVideo	361
16	SMTOWN	359
17	Ã□ukur	356
18	Marvel Entertainment	352
19	SAB TV	351

In [106]:
```
import plotly.express as px
```

In [107]:
```
px.bar(data_frame=top_20_channels[0:20], x = 'channel_title', y='total_videos')
```



To check if adding punctuation in the title helps to increase views or likes

In [108]:
```
full_df['title'][0]
```

Out[108]:
```
'Eminem - Walk On Water (Audio) ft. BeyoncÃ©'
```

In [109]:
```
import string
```

In [110]:
```
string.punctuation
```

Out[110]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [111]:
```
len([char for char in full_df['title'][0] if char in string.punctuation])
```

Out[111]:
```
4
```

In [112]:
```
def puncuation_count(text):
    return len([char for char in text if char in string.punctuation])
```

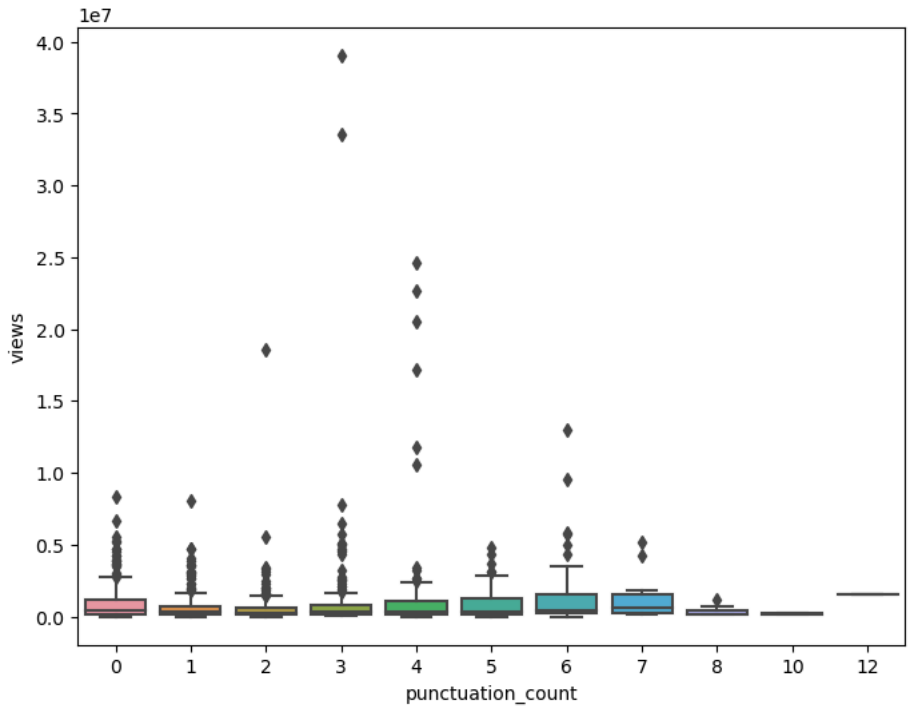
In [113]:
```
full_df['punctuation_count'] = full_df['title'].apply(puncuation_count)
```

In [114]:
```
# full_df.drop(columns = 'punctuation_count', inplace = True)
```

In [115]:
```
full_df['punctuation_count']
```

Out[115]:
```
0         4
1         1
2         3
3         3
4         3
         ..
339520    0
339521    1
339522    3
339523    0
339524    1
Name: punctuation_count, Length: 339525, dtype: int64
```

In [116]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='punctuation_count', y='views', data = full_df[0:1000])
plt.show()
```



In [117]:
```
plt.figure(figsize=(8,6))
sns.boxplot(x='punctuation_count', y='likes', data = full_df[0:1000])
plt.show()
```