YOUTUBE VIDEO ANALYSIS

Abhilash Antony

We have a YouTube video data set of the videos uploaded at the last week of the month of May, 2024. Each row in the table contains information about a single video, including the video ID, title, description, publish date, channel title, channel ID, category, tags, duration, definition, caption, view count, likes, dislikes, comment count, and favourites count.

Here are some specific details about the various features in the dataset:

- Video IDs: The video IDs appear to be alphanumeric strings, likely unique identifiers assigned by YouTube to each video.
- Titles: The titles of the videos are included in the dataset.
- Descriptions: The descriptions of the videos are also included, which can provide some context about the content of the video.
- Publish Date: The date the video was published on YouTube is included.
- Channel Title: The title of the channel that uploaded the video is listed.
- Channel ID: There is also a channel ID included, which could be a unique identifier for the channel.
- Category: The category of the video is listed. This could be a general category like "music" or "entertainment," or a more specific category like "video games" or "comedy."
- Tags: The tags associated with the video are also included. Tags are keywords that creators can add to their videos to help people find them.
- Duration: The duration of the video is listed in minutes.
- Definition: It appears there is a data point for video definition, but it is not clear from this sample what the values mean ("TRUE" or "FALSE").
- Caption: There is also a data point for caption, but it is not clear from this sample what the values mean ("TRUE" or "FALSE").
- View Count: The number of times the video has been viewed is listed.
- Likes: The number of likes the video has received is listed.
- Dislikes: The number of dislikes the video has received is listed.
- Comment Count: The number of comments on the video is listed.
- Favorites Count: The number of times the video has been added to a user's favorites list is listed.

This dataset can be used for understanding what kind of content is popular on YouTube. By analyzing the views, likes, dislikes, and comments on different videos, we can get a sense of what kind of content resonates with viewers. This information can be valuable for content creators, who can use it to inform their video strategy.

```python
# import the libraries
import pandas as pd
import matplotlib.pyplot as plt

# import the data
df = pd.read_csv("/content/trending_videos.csv")
```

Performing initial analysis,

```python
df.head()
```

video_id	title	description	published_at	channel_id	channel_title	category_id	tags	duration	definition	caption	view_count	like_count	dislike_count	favorite_count	comment_count
22tVWwmTie8	Eminem - Houdini [Official Music Video]	Eminem - Houdini\nListen: https://eminem.lnk.t...	2024-05-31T04:00:02Z	UC20vb-R_px4CguHzzBPhoyQ	EminemVEVO	10	['Eminem', 'Houdini', 'Hip', 'Hop', 'エミネム', '에...	PT4M57S	hd	True	14736971	1306831	0	0	105793
Kf86x8F9M90	College Football 25 | Gameplay Deep Dive	Bring Glory Home. Pre-order EA SPORTS College ...	2024-05-31T14:55:06Z	UCT4wAMwETXqDf-U_DVuqabA	EA SPORTS College	20	['college football', 'college football 25', 'c...	PT4M52S	hd	False	1079642	50259	0	0	6936
mfz-Ztki88s	ILLEGAL builds in LEGO...	50+ secret ways to build in Lego you probably ...	2024-05-31T15:30:38Z	UCUU3GdGuQshZFRGnxAPBf_w	TD BRICKS	24	['lego', 'lego set', 'lego sets', 'lego movie'...	PT9M7S	hd	True	1064281	24723	0	0	2690
VGnOpZhsPk4	ATEEZ(에이티즈) - 'WORK' Official MV	[GOLDEN HOUR : Part.1]\nRelease Date: 2024. 5....	2024-05-31T04:00:01Z	UCQdq-lqPEq_yZ_wP_kuVB9Q	KQ ENTERTAINMENT	10	['KQ', '케이큐']	PT3M15S	hd	True	11742765	338559	0	0	28919
m-4ZM3jxhdE	State of Play | May 30, 2024	State of Play is back! Tune in live for update...	2024-05-30T22:00:12Z	UC-2Y8dQb0S6DtpxNgAKoJKA	PlayStation	20	['PlayStation', 'PS5', 'video games', 'next ge...	PT35M32S	hd	True	1672973	52456	0	0	8292

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 16 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   video_id        200 non-null    object
 1   title           200 non-null    object
 2   description     196 non-null    object
 3   published_at    200 non-null    object
 4   channel_id      200 non-null    object
 5   channel_title   200 non-null    object
 6   category_id     200 non-null    int64
 7   tags            200 non-null    object
 8   duration        200 non-null    object
 9   definition      200 non-null    object
 10  caption         200 non-null    bool
 11  view_count      200 non-null    int64
 12  like_count      200 non-null    int64
 13  dislike_count   200 non-null    int64
 14  favorite_count  200 non-null    int64
 15  comment_count   200 non-null    int64
dtypes: bool(1), int64(6), object(9)
memory usage: 23.8+ KB
```

```python
# Check for missing values
print(df.isnull().sum())
```

```
video_id          0
title             0
description       4
published_at      0
channel_id        0
channel_title     0
category_id       0
tags              0
duration          0
definition        0
caption           0
view_count        0
like_count        0
dislike_count     0
favorite_count    0
comment_count     0
```

We see here that the variable description has 4 missing values. Usually we impute or remove them. Here, we remove the tuples with missing values.

```
# Drop the missing values
df = df.dropna()
```

```
# Get descriptive statistics
df.describe()
```

	category_id	view_count	like_count	dislike_count	favorite_count	comment_count
count	200.000000	2.000000e+02	2.000000e+02	200.0	200.0	200.000000
mean	18.835000	2.296781e+06	9.129304e+04	0.0	0.0	8131.505000
std	6.585943	5.992482e+06	2.397322e+05	0.0	0.0	28670.786143
min	1.000000	5.526100e+04	1.430000e+02	0.0	0.0	0.000000
25%	17.000000	3.462905e+05	1.472700e+04	0.0	0.0	1010.000000
50%	20.000000	7.330895e+05	2.795400e+04	0.0	0.0	2046.000000
75%	24.000000	1.386557e+06	6.148650e+04	0.0	0.0	4197.000000
max	28.000000	6.643700e+07	2.535500e+06	0.0	0.0	279003.000000

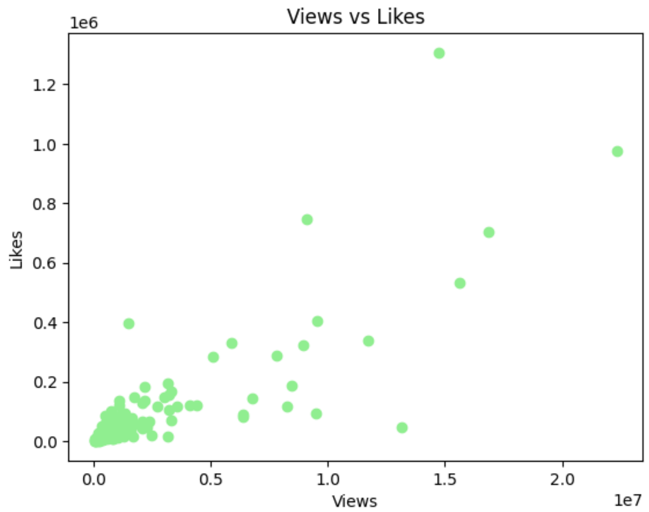
```
# find the channel_id that published the most videos
most_videos_channel = df['channel_id'].value_counts().idxmax()
print(f"Channel with the most videos: {most_videos_channel}")
```

Channel with the most videos: UCWJ2lWNubArHWmf3FIHbfcQ

```
# find the range of date our data spans
min_date = df['published_at'].min()
max_date = df['published_at'].max()
print(f"Range of published dates: {min_date} to {max_date}")
```

Range of published dates: 2024-05-04T03:06:01Z to 2024-05-31T15:30:38Z

```
# the relationship between views and likes
plt.scatter(df['view_count'], df['like_count'], color='lightgreen')
plt.title('Views vs Likes')
plt.xlabel('Views')
plt.ylabel('Likes')
plt.show()
```

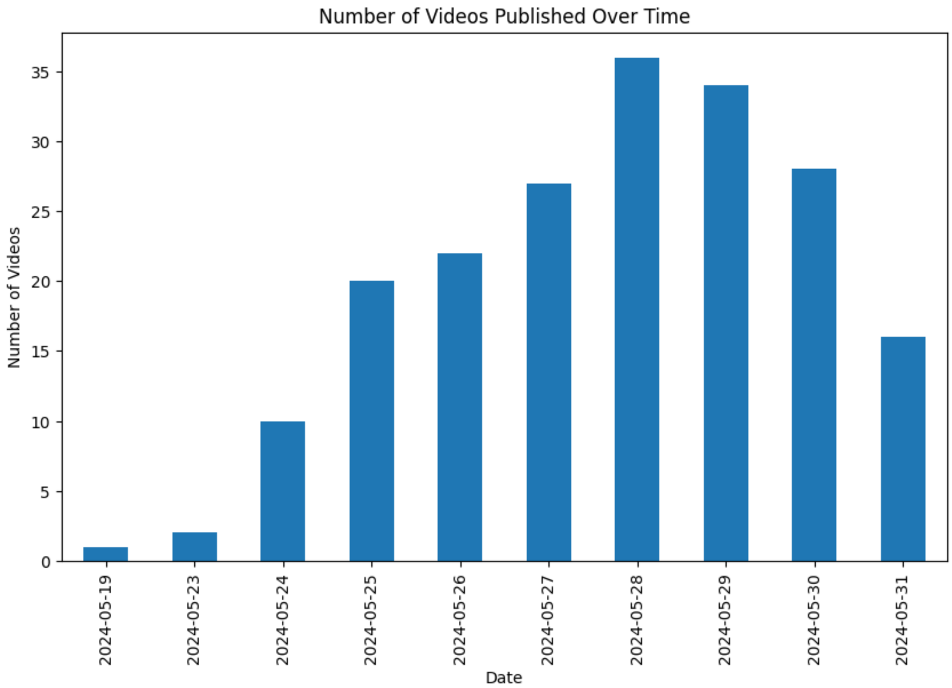


The plot above shows a positive correlation between the Views and Likes. That is, as the number of views to the video increases, the possibility of getting more likes also increases.

Here, all the numerical values are in a consistent format as all of them are counts. So, we opt not to perform any standardization procedures.

Now, let us look at the distribution of videos from 2024-05-04 to 2024-05-31.

```
# Create a new column with the date only
df['date'] = pd.to_datetime(df['published_at']).dt.date

# Group the data by date and count the number of videos
grouped_data = df.groupby('date')['video_id'].count()

# Plot the data
grouped_data.plot(kind='bar', figsize=(10, 6))
plt.title('Number of Videos Published Over Time')
plt.xlabel('Date')
plt.ylabel('Number of Videos')
plt.show()
```



Here, we see that most of the videos were uploaded at the end of the week. There can be other potential factors affecting this. But that would need further data.

Another thing that affects the popularity of YouTube videos is the tags that are used along with the videos. Now, let's find out the tags that are most used in the given data.

```
# Convert the tags column to a list of strings
df['tags'] = df['tags'].str.strip('[]').str.strip("'").str.split("',
'")

# Flatten the list of lists into a single list
```

```
all_tags = [tag for sublist in df['tags'].tolist() for tag in sublist]

# Count the occurrences of each tag
tag_counts = {}
for tag in all_tags:
  if tag not in tag_counts:
    tag_counts[tag] = 0
  tag_counts[tag] += 1

# Sort the tags by their counts
sorted_tags = sorted(tag_counts.items(), key=lambda item: item[1],
reverse=True)

# Print the most used tags
print("Most used tags:")
for tag, count in sorted_tags[:10]:
  print(f"{tag}: {count}")
```

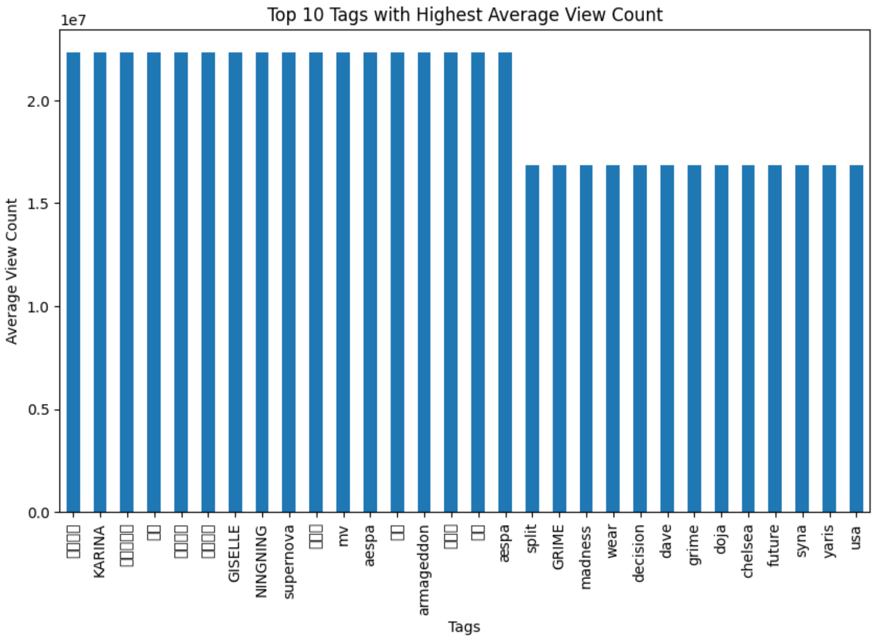
```
Most used tags:
the: 40
vs: 37
and: 34
man: 34
2: 33
of: 33
new: 29
pokemon: 29
Fortnite: 29
minecraft: 28
```

This seems to be misleading as we are not sure of the stochastic behaviour of our data. There is a possibility that the data might be skewed to videos for kids as the most used tags seem to relate to kids' videos.

Now, lets look at how tags are related to the view count.

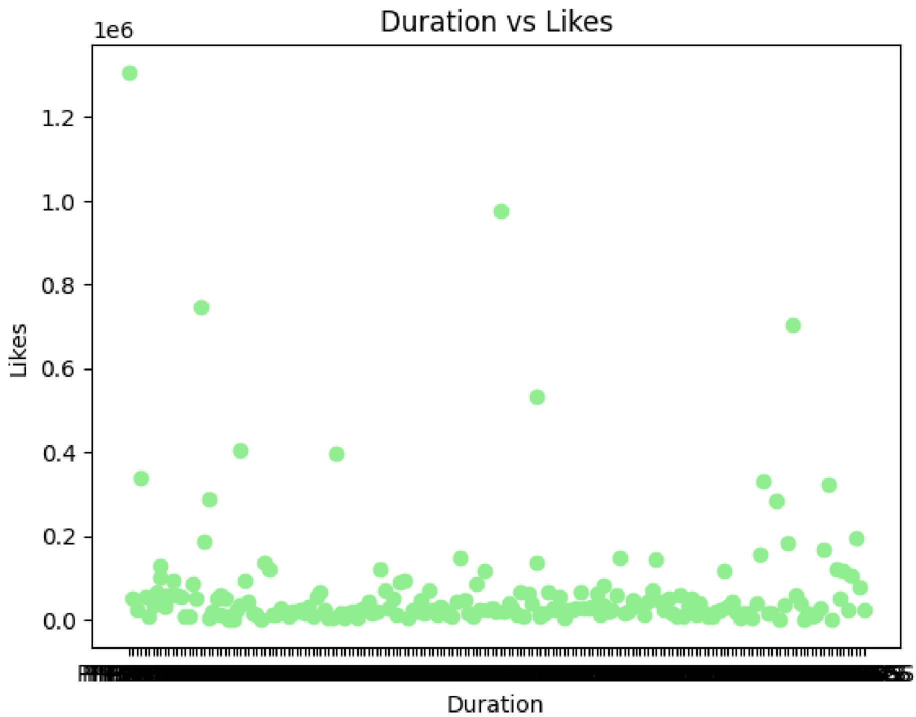
```
# Explode the tags column to get individual tag-video pairs
df_exploded = df.explode('tags')

# Group the exploded data by tags and calculate the average view count
avg_views_by_tag = df_exploded.groupby('tags')['view_count'].mean()

# Sort the tags by average view count
sorted_tags_by_views = avg_views_by_tag.sort_values(ascending=False)

# Plot the top 10 tags with the highest average view count
plt.figure(figsize=(10, 6))
sorted_tags_by_views[:20].plot(kind='bar')
plt.title('Top 10 Tags with Highest Average View Count')
plt.xlabel('Tags')
plt.ylabel('Average View Count')
```

```
plt.show()
```



Now, we look into the relationship between the video duration and the number of likes the video received.

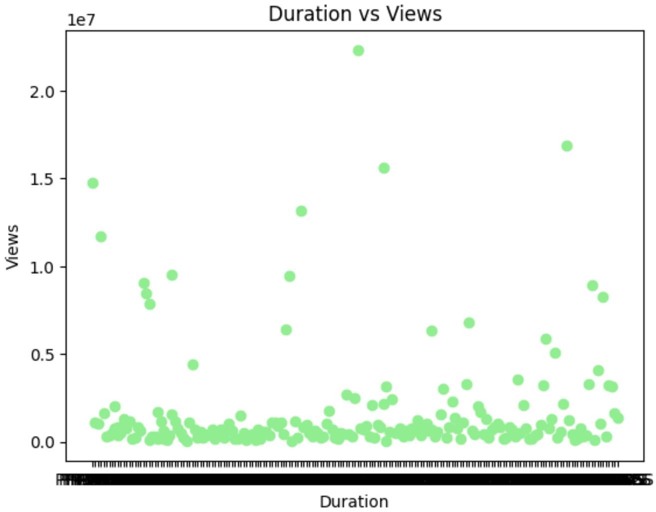
```
plt.scatter(df['duration'], df['like_count'], color='lightgreen')
plt.title('Duration vs Likes')
plt.xlabel('Duration')
plt.ylabel('Likes')
plt.show()
```



It's evident that the videos with the least length are the most likely to be watched. However, the association of duration with the number of likes seems vague.

Lets also peek into the relationship between the duration and number of views.

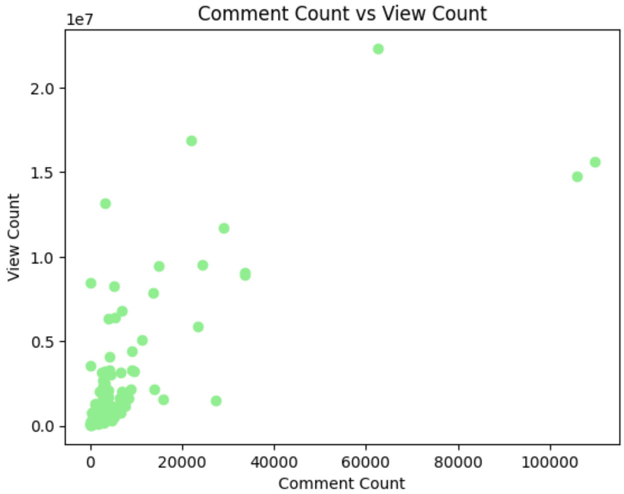
```
plt.scatter(df['duration'], df['view_count'], color='lightgreen')
plt.title('Duration vs Views')
plt.xlabel('Duration')
plt.ylabel('Views')
plt.show()
```



It seems that videos with moderate duration are more likely to be viewed more frequently.

Now, looking at the number of comments, with respect to the number of views.

```
plt.scatter(df['comment_count'], df['view_count'], color='lightgreen')
plt.title('Comment Count vs View Count')
plt.xlabel('Comment Count')
plt.ylabel('View Count')
plt.show()
```



In contrast to my expectations, videos with lesser view counts get fewer comments. And a few videos with more views have more comments.

Conclusions…

This exploratory data analysis (EDA) provided insights into a YouTube video dataset for videos uploaded during the last week of May 2024. Here's a summary of the key findings:

- **Video characteristics:** The dataset includes video IDs, titles, descriptions (with some missing values), publish dates, channel titles and IDs, categories, tags, durations, definitions (unclear meaning), captions (unclear meaning), view counts, likes, dislikes, comment counts, and favorites counts.
- **Content popularity:** There might be a positive correlation between views and likes, suggesting viewers tend to like videos they watch more.
- **Temporal distribution:** The number of videos uploaded increased towards the end of the analyzed week. Further investigation is required to determine if this is a consistent pattern.
- **Tag analysis:** "Most used tags" might be misleading due to potential skewness towards a specific category (e.g., kids' videos) requiring further exploration. There's a possibility of identifying tags associated with higher average view counts.
- **Video duration and engagement:** There seems to be a weak association between video duration and likes/dislikes. Videos with moderate duration might be more likely to receive views. There's a need for further analysis to solidify these observations.
- **Comments and views:** Contrary to expectations, videos with fewer views have fewer comments, suggesting a low level of interaction for less popular content.

However, there are some limitations:

- The analysis is based on data for a single week, limiting the generalizability of the findings.
- The meaning of "definition" and "caption" data points is unclear and needs clarification.
- Further investigation is required to confirm the observed trends and understand the underlying reasons.

This analysis has applications to future prospects:

- Analyze data from a longer period to understand seasonal or long-term trends.
- Investigate the content of video descriptions (after dealing with missing values) to gain insights into video topics, and NLP can be applied for further investigation.
- Explore sentiment analysis of comments to understand audience reception.

This initial exploration provides a foundation for further analysis of YouTube video data. By delving deeper, you can gain valuable insights into user preferences and content creation strategies for the platform. These insights provide a comprehensive understanding of the trends and patterns in YouTube video data. Content creators can leverage these findings to optimize video uploads, tagging, and content curation strategies to enhance viewer engagement and popularity.