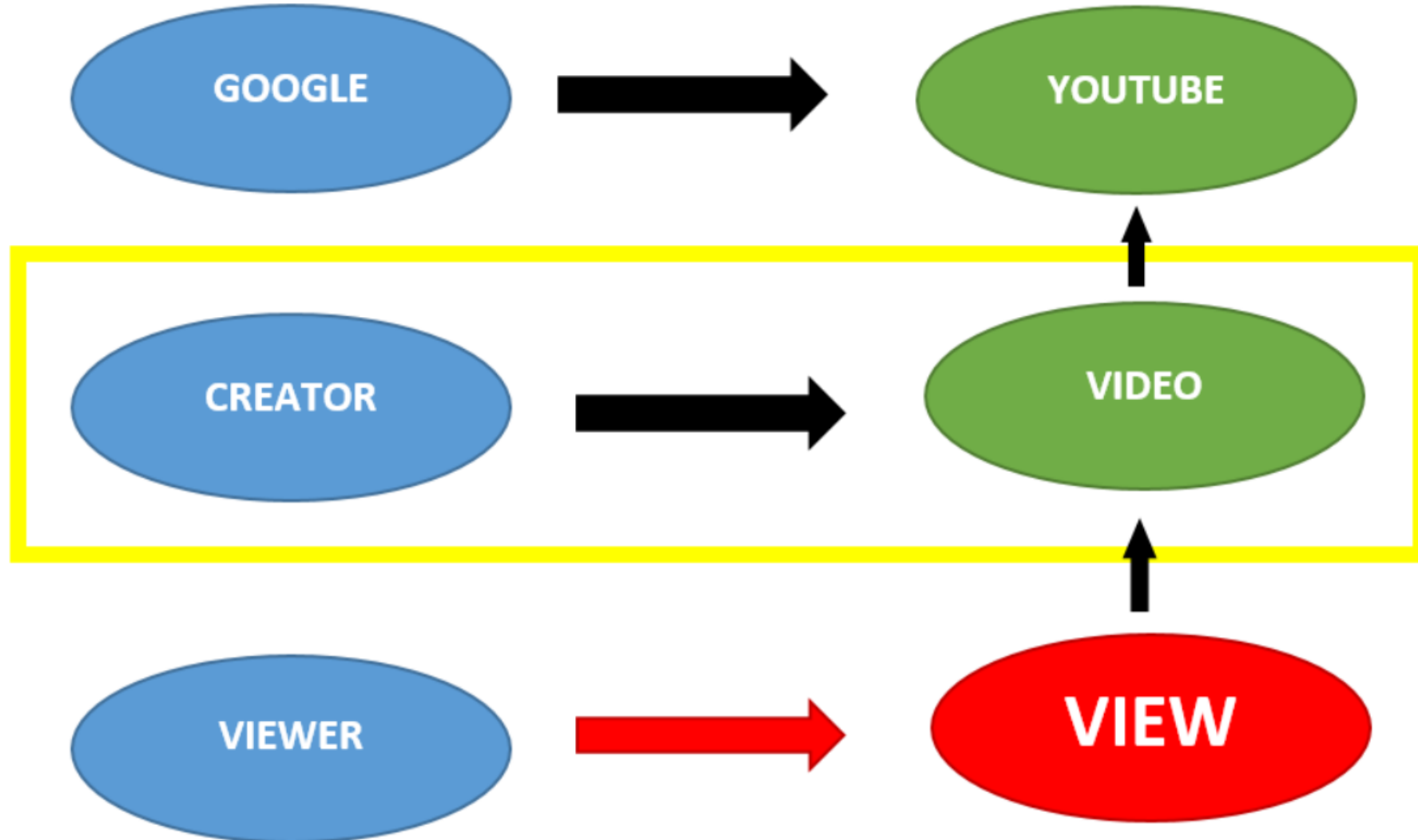


"Trends on Trends" - An Analysis of the Youtube Creator's influence on video trendability.

This project will explore how the Youtube creator affects the path from viewer to view, within the Youtube platform (excluding outside factors).

```
In [169]: from IPython.display import Image  
Image(filename='Youtube Relationship - Diagram.PNG')
```

Out[169]:



DATA BACKGROUND & PREVIEW

What data is available?

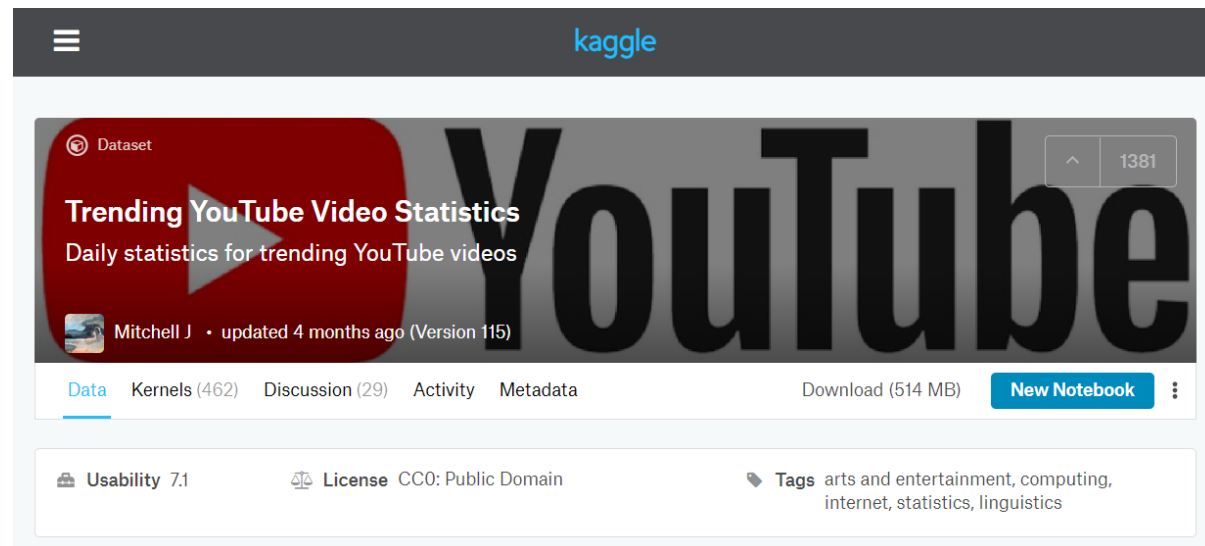
- Source: <https://www.kaggle.com/datasnaek/youtube-new>
- "Daily Record of the Top Trending Youtube Videos"
- Up to 200 trending videos per day
- Two dataframes with the same columns (19).
 - 'df' should contain one line per video with final numbers
 - "df_all_timestamps" should contain snapshots throughout the day.

Limitations

- No data on the viewers or path to arrive to video.
- Only 1 property of channel to be attributed to Youtube creator. More information on creator could lead to further insights.

Unanswered Questions

- What defines "trending"?
 - What is the metric? Assumption: Unknown internal measure (Top X videos by views / time_unit)
 - Dynamic - Constantly changes throughout day (e.g. Jan 3 - 184, Jan 4 - 828, Jan 5 - 875).
 - Does it change by viewer's geographical region?
- What is "country"?



QUESTIONS TO EXPLORE

WHAT?

- Is there a correlation between CONTENT properties inputted by the creator and trendability?
- Is there a correlation between the creator's publishing TIMELINE and trendability?

WHY?

- For Creators, are there any opportunities and or suggested behavior to encourage trendability?
- For Youtube, is it worthwhile to spend additional resources on the Youtube Creator? Are there any suggested changes to the existing platform?

EFFECT OF CONTENT

The Youtube Creator has a lot of control over the properties of the channel and video. The following properties within dataframe are manually entered by the creator: Title, Description, Thumbnail, Tags, Category, Comments Disabled, and Ratings Disabled. This section will explore if the content properties from the creator have any correlation with trendability.

Upload video (beta) Draft - not visible to viewers 95% of video processed

1 Basic info 2 Advanced settings 3 Preview & publish

Title (required)
TBA

Description (required) ?
Add description

Thumbnail

Custom thumbnail
(2MB or less) ?

Video URL
https://youtu.be/M6xCDEc_xwE

Filename
Bear.wmv

Upload video (beta) Draft - not visible to viewers 100% of video processed

1 Basic info 2 Advanced settings 3 Preview & publish

Playlists
Add this video to one or more playlists

End screen
Add an end screen to this video

Cards
Add info cards to this video

Language and subtitles
Make your content accessible to everyone

Tags
Provide descriptive keywords to help correct search mistakes

Other

Category
People & Blogs

Video location
Help viewers find your video when they search for a specific lo...

Recording date
Include the date your video was created

License and rights ownership
Choose a copyright license

Additional settings
Additional settings and information

Comments and ratings
All comments allowed, Ratings viewable

REQUIRED

- Title
- Description

REQUIRED & PRE-FILLED

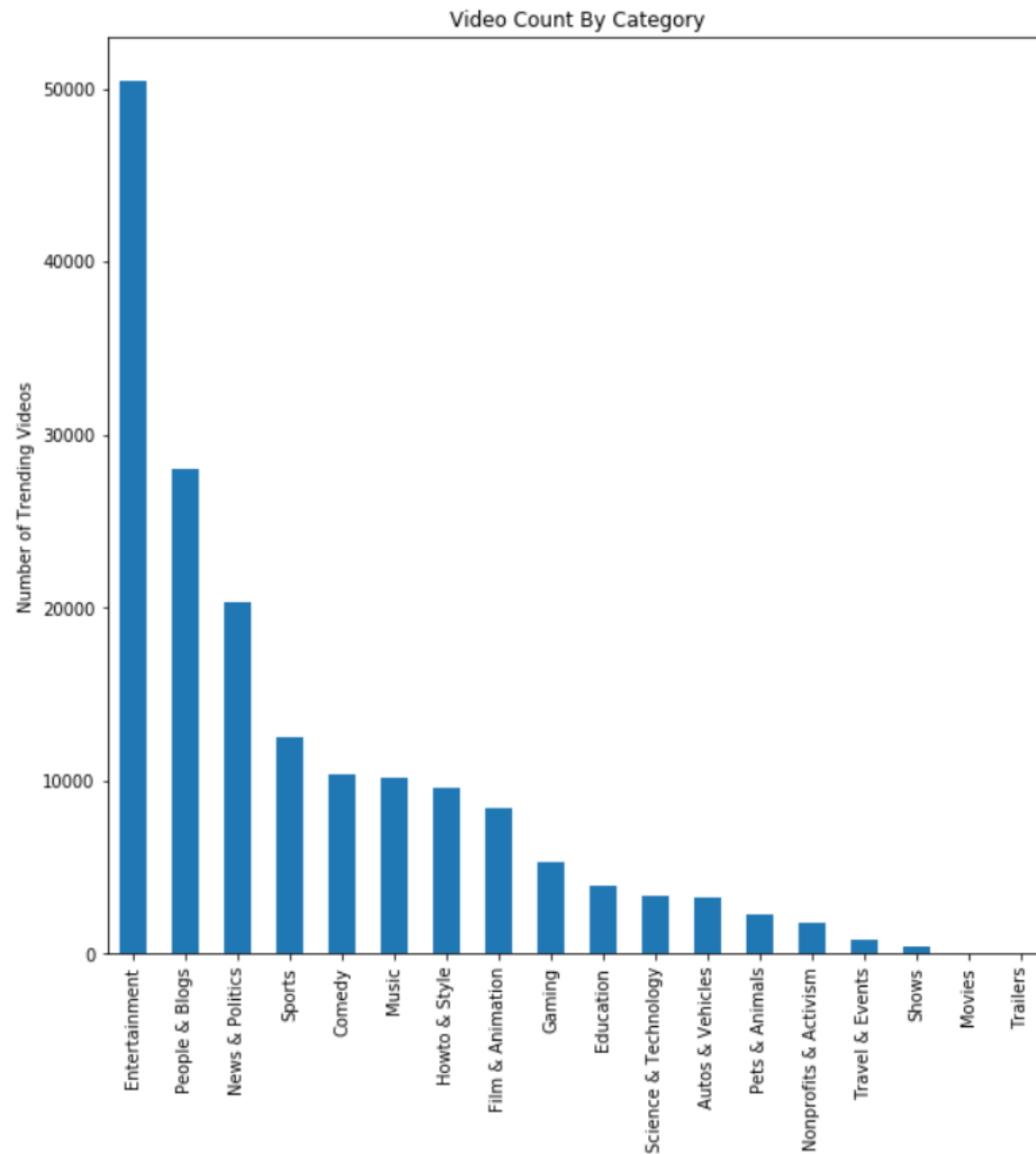
- Category
- Allow Comments
- Users can view ratings

... CATEGORY ...

```
category_count = df['category'].value_counts()  
print(category_count)
```

```
Entertainment      50468  
People & Blogs      28045  
News & Politics     20327  
Sports             12561  
Comedy             10358  
Music              10145  
Howto & Style       9580  
Film & Animation    8460  
Gaming             5318  
Education          3897  
Science & Technology 3404  
Autos & Vehicles    3274  
Pets & Animals      2271  
Nonprofits & Activism 1779  
Travel & Events     812  
Shows              435  
Movies             12  
Trailers           3  
Name: category, dtype: int64
```

```
fig = plt.figure(figsize=(10,10))  
ax = fig.gca()  
category_count.plot(kind = 'bar', ax = ax)  
ax.set_title('Video Count By Category') # Give the plot a main title  
ax.set_xlabel('category') # Set text for the x axis  
ax.set_ylabel('Number of Trending Videos') # Set text for y axis
```



```
In [82]: # Is Entertainment category more popular OR is it used very leniently, based on US?
# Qualitative Assessment - Are the Top 5 US videos properly categorized into Entertainment?
```

```
entertainment = df['category'] == 'Entertainment'
US = df['country'] == 'US'
df[entertainment & US].head()
```

Out[86]:

	video_id	trending_date	title	channel_title	category_id	category	publish_date	publish_time	tags
322	zYWt2mnalP8	2017-11-14	How To Do Thanksgiving Makeup That Has Nothing...	Sailor J	24 Entertainment Howto and Style		2017-11-08	00:03:47	Thanksgiving "Tutorial "Makeup"
666	cOc3tsFWoRs	2017-11-14	Jason Momoa & Lisa Bonet: Love at First Sight	The Late Late Show with James Corden	24 Entertainment Comedy		2017-11-10	09:35:00	James Corden "The Late Late Show "Colbert "I...
672	F-j_luaYfw	2017-11-14	I Miss the Old Taylor	Wong Fu Productions	24 Entertainment		2017-11-08	21:05:20	swift "swiftie "reputation "ready for... taylor
674	Y6eKxjMA9ek	2017-11-14	Dropping And CATCHING A Paper Airplane At 2,00...	Tucker Gott	24 Entertainment		2017-11-09	16:00:09	paramotor "tucker "gott "tucker gott "param...
675	zy0b9e40tK8	2017-11-14	Dark Official Trailer [HD] Netflix	Netflix	24 Entertainment Movies OR Trailers		2017-11-09	09:00:07	Netflix "Baran Bo Odar "Jantje Frieze "DARK"...

- Category is subjectively chosen by the creator.
- Although 'Entertainment' is the most popular category, a closer looks shows that it is used broadly instead of better matching categories
- With this in mind, is a better solution to make categories more descriptive for Youtube OR should the creator play around this?
- If 'Entertainment' videos trend better and assuming category is used in suggestion algorithms, the Youtube creator should consider if he/she would prefer to use a wider category that may reach more people but might not fit what the viewer is looking for OR if they prefer to select the category that better matches their video, targets less viewers, is less competitive, and perhaps it is more relevant to what the viewer is looking for.

... DESCRIPTION LENGTH ...

```
In [104]: df['description_length'] = None  
df['description_length'] = df['description'].str.len()
```

```
In [110]: df.head()
```

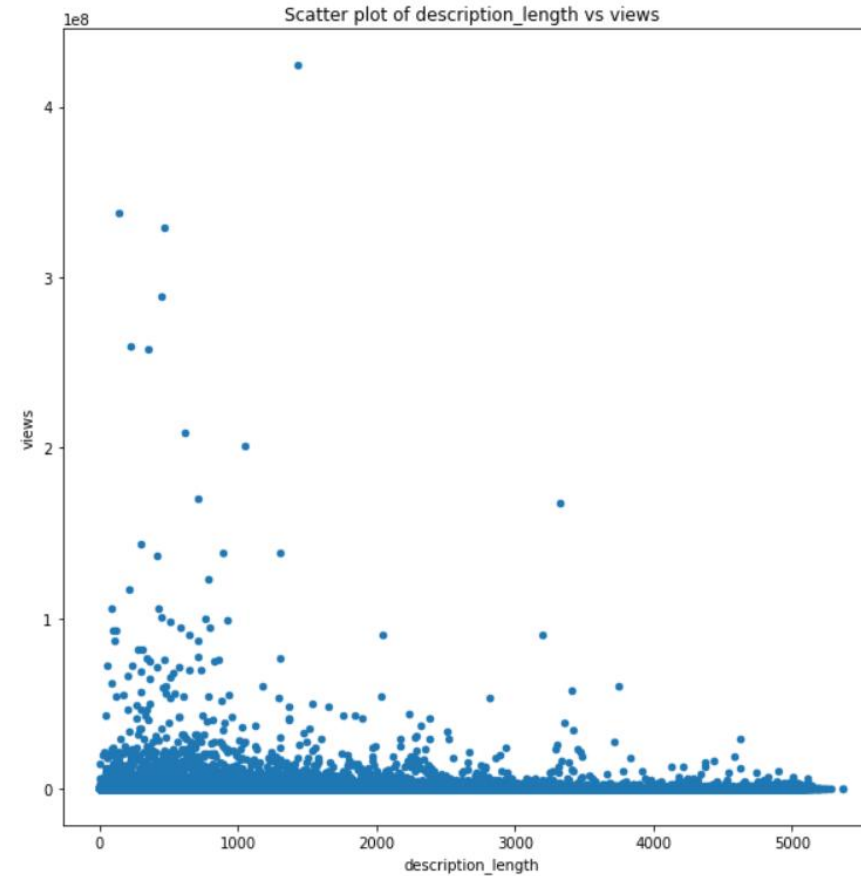
```
Out[110]:
```

log_comment_count	tag_count	publish_day	trending_day	Time_Live	Days_Live	description_length
5.375278	44	Sunday	Tuesday	1 days 06:01:03	1	288

```
fig = plt.figure(figsize=(10, 10))  
ax = fig.gca()  
df.plot(kind = 'scatter', x = 'description_length', y = 'views', ax = ax)  
ax.set_title('Scatter plot of description_length vs views')  
ax.set_xlabel('description_length')  
ax.set_ylabel('views')
```

```
df['description_length'].describe()
```

```
count    171149.000000  
mean       800.058656  
std        835.311361  
min         1.000000  
25%        225.000000  
50%        536.000000  
75%       1065.000000  
max       5369.000000  
Name: description_length, dtype: float64
```



- Description length count does not correlate with views but we can see patterns on how long are the descriptions 'trending' views tend to have.
- Assuming description factor into Youtube search algorithms, we can make recommendations on description_length.
- The average description_length of trending videos is 536 characters and it should be kept within 225 and 536 (the range followed by 50% of trending videos).

... TAG COUNT ...

```
In [23]: # every video has tags
```

```
no_tags = df['tags'] == None
df[no_tags]
```

```
Out[23]:
```

video_id	trending_date	title	channel_title	cate
----------	---------------	-------	---------------	------

```
# make new column called tag_count
```

```
df['tag_count'] = None
```

```
# put data into the column
```

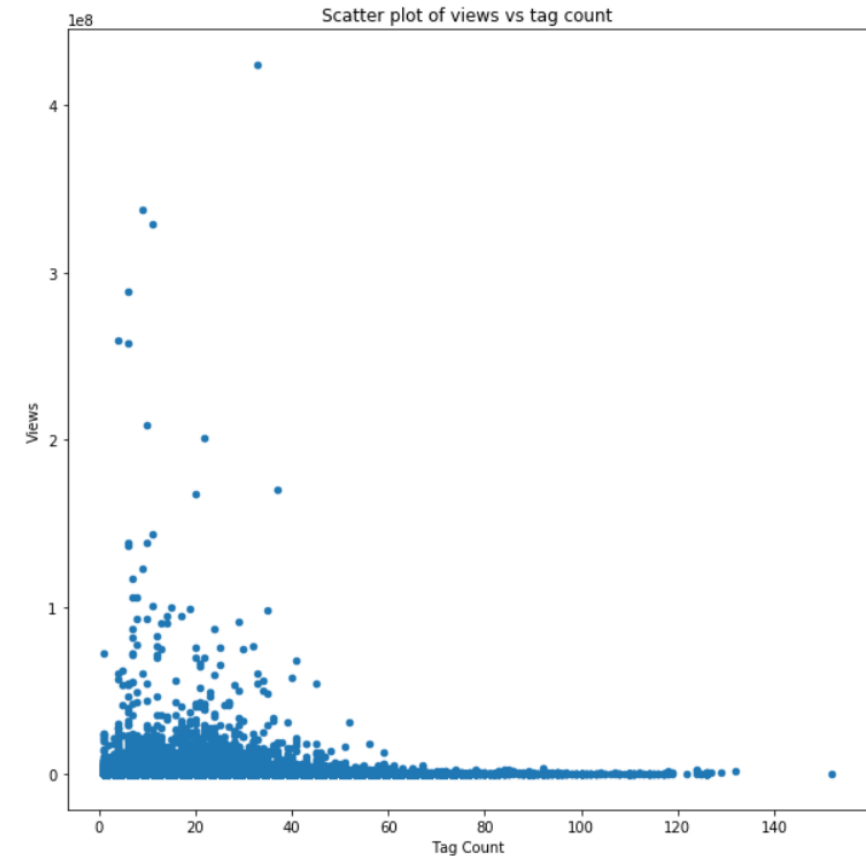
```
df['tag_count'] = df['tags'].str.split(r'|')
df.loc[:, 'tag_count'] = df['tag_count'].apply(lambda x: len(x))
```

t	tag_count
---	-----------

3	44
---	----

```
df['tag_count'].describe()
```

count	171149.000000
mean	17.148222
std	13.284787
min	1.000000
25%	7.000000
50%	15.000000
75%	25.000000
max	152.000000
Name: tag_count, dtype: float64	



```
fig = plt.figure(figsize=(10, 10))
ax = fig.gca()
df.plot(kind = 'scatter', x = 'tag_count', y = 'views', ax = ax)
ax.set_title('Scatter plot of views vs tag count')
ax.set_xlabel('Tag Count')
ax.set_ylabel('Views')
```

- Tag count does not directly correlate with views but we can see patterns on how many tags 'trending' views tend to have.
- Assuming tags factor into Youtube search algorithms, we can make recommendations on tag_count.
- The average tag count of trending videos is 17 and it should be kept within 7 and 25 (the range followed by 50% of trending videos). Every trending video has at least 1 so it should be not left blank.

... OPENNESS TO FEEDBACK (COMMENTS & RATINGS) ...

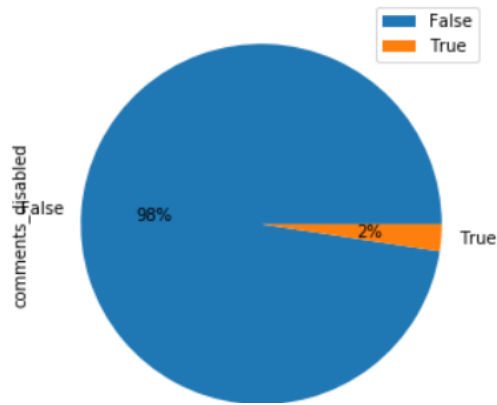
```
# open to comments? - count
```

```
comments_disabled_count = df['comments_disabled'].value_counts()
print(comments_disabled_count)
```

```
False    167027
True       4122
Name: comments_disabled, dtype: int64
```

```
# open to comments? - proportion
```

```
df2 = pd.DataFrame(comments_disabled_count)
plot = df2.plot.pie(y='comments_disabled', figsize=(5, 5), autopct='%1.0f%%')
```



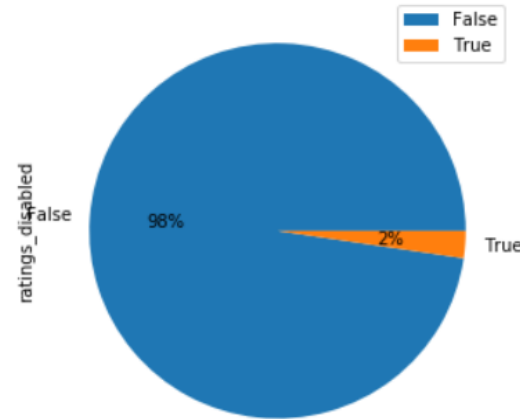
```
# open to ratings? - count
```

```
ratings_disabled_count = df['ratings_disabled'].value_counts()
print(ratings_disabled_count)
```

```
False    167159
True       3990
Name: ratings_disabled, dtype: int64
```

```
# open to ratings? - proportion
```

```
df3 = pd.DataFrame(ratings_disabled_count)
plot = df3.plot.pie(y='ratings_disabled', figsize=(5, 5), autopct='%1.0f%%')
```



```
# What is the overlap? Closed to both comments are ratings.
```

```
df4 = df[(df.ratings_disabled == True) & (df.comments_disabled == True)]
len(df4)
```

```
1153
```

```
# closed videos / total videos
# as a percentage
```

```
(len(df4)/len(df))*100
```

```
0.6736819963891113
```

- The grand majority of videos are open to feedback with 98% open to comments and 98% open to ratings
- Less than 1 percent (0.67%) are closed to both ratings and comments.
- For Youtube creators: it is important to note that trending videos overwhelming tend to be open to feedback and if trendability/popularity is the goal, their videos should follow this pattern.

EFFECT OF TIME

Some video properties stem from the Youtube Creator's actions: particularly publish time and frequency of uploads. This section will explore possibly opportunities related to the timeline of video uploads.

PROCESSING DONE

Publish

Publish on Sat Sep 28 2019



Click "Publish" to make your video live.

Draft saved.

Draft saved.

Basic info

Translations

Advanced settings

Butterfly

Description

Tags (e.g., albert einstein, flying pig, mashup)



New! Premieres

Make it a moment! Build the hype and get fans excited about your next video.

- Schedule your Premiere
- Share your watch page URL with your fans
- Chat with fans before and during the Premiere
- Watch the Premiere with fans in real time

[Learn more](#)

Premiere ☐

Public



New! Premieres

Make it a moment! Build the hype and get fans excited about your next video.

- Schedule your Premiere
- Share your watch page URL with your fans
- Chat with fans before and during the Premiere
- Watch the Premiere with fans in real time

[Learn more](#)

Premiere ☐

Scheduled

Sep 28, 2019

6:30 AM

(GMT-0400) Local Time

... PUBLISH DATE ...

```
df['publish_day'] = None
```

```
df['publish_day'] = df['publish_timestamp'].dt.dayofweek
```

```
df.head(50)
```

country	publish_timestamp	trending_timestamp	log_views	log_likes	log_dislikes	log_comment_count	tag_count	publish_day
FR	2017-11-12 17:58:57	2017-11-14	9.877554	7.375882	3.178054	5.375278	44	6

```
publish_day_count = df['publish_day'].value_counts()  
print(publish_day_count)
```

```
4    26978  
3    25624  
0    24920  
1    23992  
2    23868  
6    23233  
5    22534  
Name: publish_day, dtype: int64
```

```
def dayofweek(row):  
    if row['publish_day'] == 0:  
        return 'Monday'  
    elif row['publish_day'] == 1:  
        return 'Tuesday'  
    elif row['publish_day'] == 2:  
        return 'Wednesday'  
    elif row['publish_day'] == 3:  
        return 'Thursday'  
    elif row['publish_day'] == 4:  
        return 'Friday'  
    elif row['publish_day'] == 5:  
        return 'Saturday'  
    elif row['publish_day'] == 6:  
        return 'Sunday'  
  
df.loc[:, 'publish_day'] = df.apply(dayofweek, axis=1)  
df.head()
```

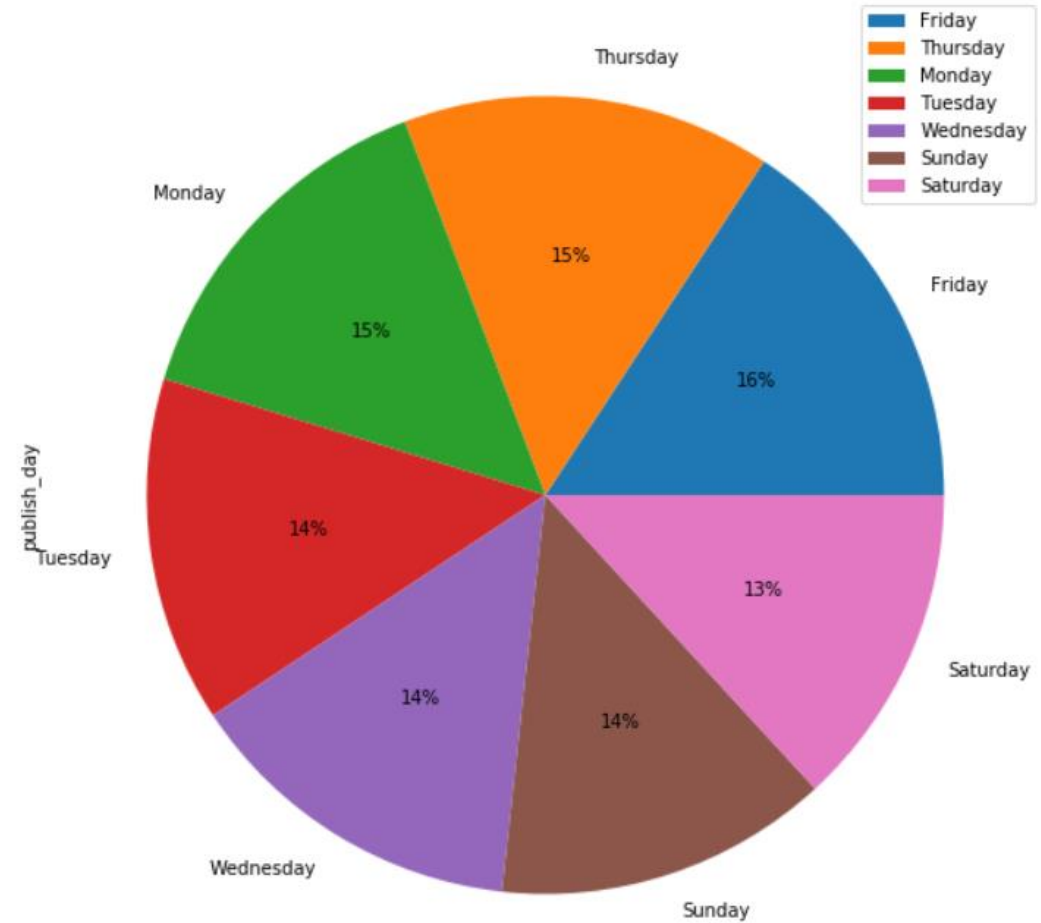
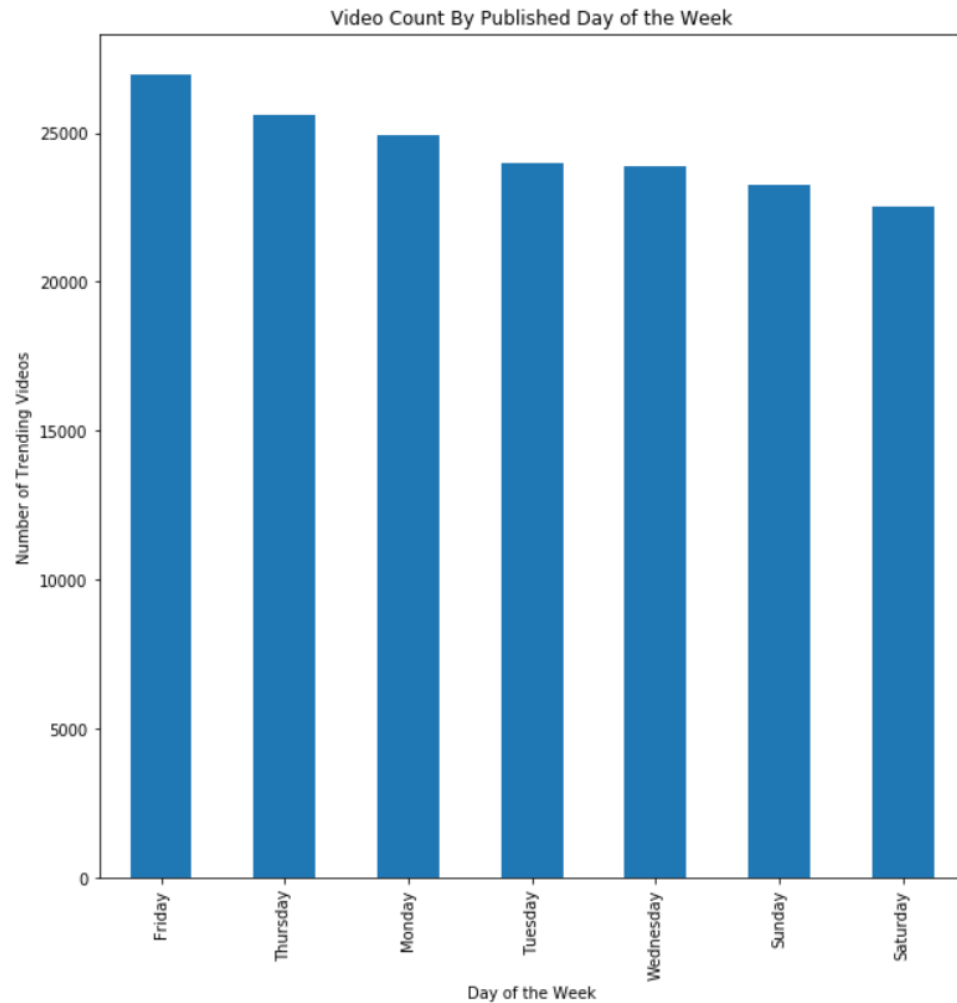
country	publish_timestamp	trending_timestamp	log_views	log_likes	log_dislikes	log_comment_count	tag_count	publish_day
FR	2017-11-12 17:58:57	2017-11-14	9.877554	7.375882	3.178054	5.375278	44	Sunday

```
publish_day_count = df['publish_day'].value_counts()  
print(publish_day_count)
```

```
Friday      26978  
Thursday    25624  
Monday      24920  
Tuesday     23992  
Wednesday   23868  
Sunday      23233  
Saturday    22534  
Name: publish_day, dtype: int64
```

```
fig = plt.figure(figsize=(10,10))  
ax = fig.gca()  
publish_day_count.plot(kind = 'bar', ax = ax)  
ax.set_title('Video Count By Published Day of the Week')  
ax.set_xlabel('Day of the Week')  
ax.set_ylabel('Number of Trending Videos')
```

```
df5 = pd.DataFrame(publish_day_count)  
plot = df5.plot.pie(subplots = True, figsize=(10, 10), autopct='%1.0f%%')
```



- Publish date is generated when the Creator uploads the video and hence in his control.
- The day of the week is pretty evenly split so the difference between days is not drastic. However... when placed in video count order, there seems to be a preference and/or strategic decision to publish on weekdays (Saturday and Sunday had the least amount)
- Friday is the most popular day to publish, perhaps getting ahead of the weekend?

... DAYS ON THE PLATFORM (PUBLISH DATE vs TRENDING DATE) ...

```
df['Time_Live'] = df['trending_timestamp'] - df['publish_timestamp']
```

```
df.head(100)
```

_timestamp	log_views	log_likes	log_dislikes	log_comment_count	tag_count	publish_day	trending_day	Time_Live
2017-11-14	9.877554	7.375882	3.178054	5.375278	44	Sunday	Tuesday	1 days 06:01:03

```
df.info()
```

```
Time_Live      171149 non-null timedelta64[ns]
```

```
df["Days_Live"] = None
```

```
df["Days_Live"] = df["Time_Live"].dt.days
```

```
df.head(100)
```

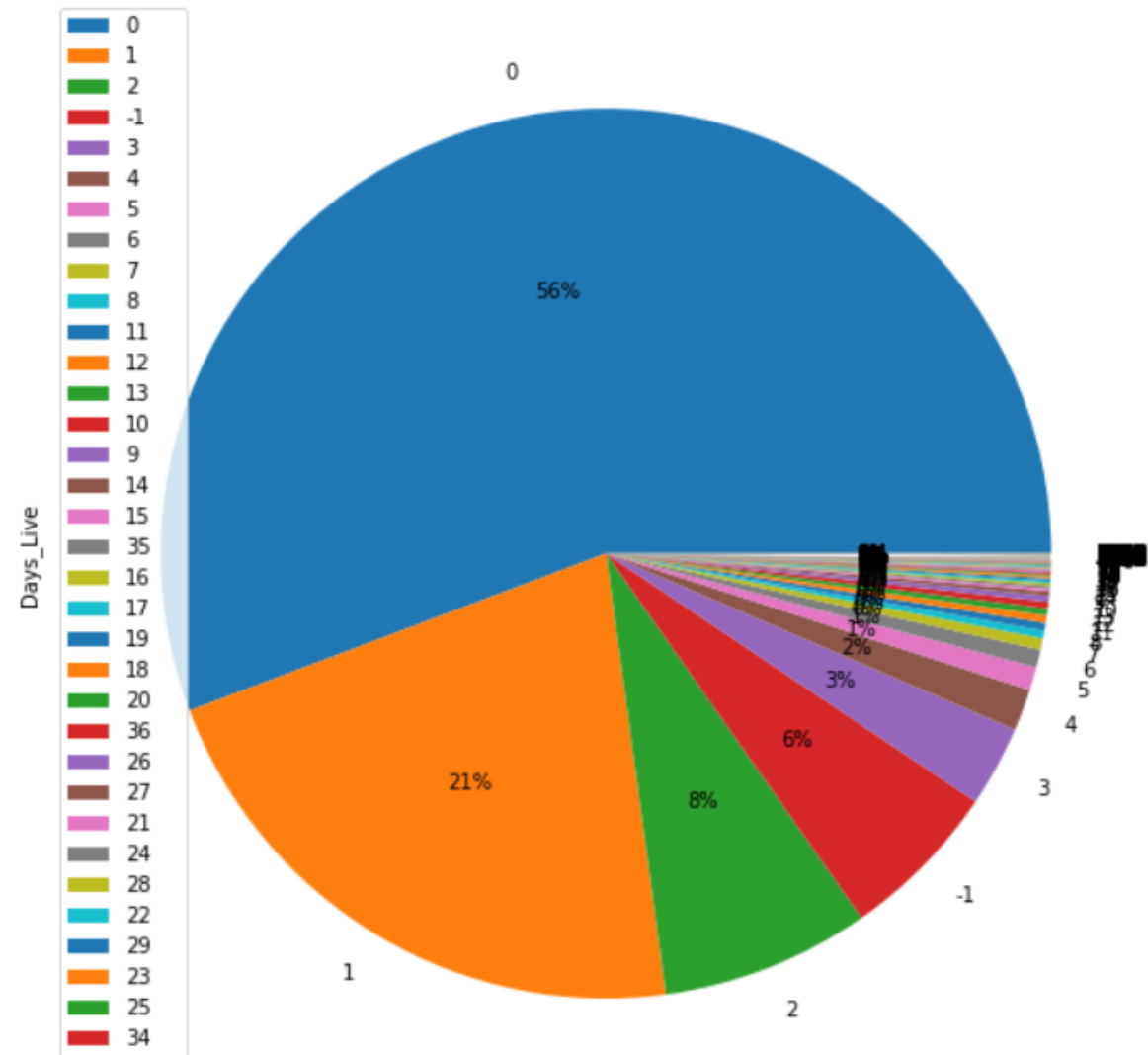
_timestamp	log_views	log_likes	log_dislikes	log_comment_count	tag_count	publish_day	trending_day	Time_Live	Days_Live
2017-11-14	9.877554	7.375882	3.178054	5.375278	44	Sunday	Tuesday	1 days 06:01:03	1

```
days_live_count = df['Days_Live'].value_counts()
print(days_live_count)
```

```
0      95410
1      36631
2      12972
-1     10024
3       5021
...
1356      1
77         1
333        1
461         1
698         1
```

Name: Days_Live, Length: 394, dtype: int64

```
df7 = pd.DataFrame(days_live_count)
plot = df7.plot.pie(subplots = True, figsize=(10, 10), autopct='%1.0f%%')
```



- The overwhelming majority of trending videos have been on the platform for less than a day 56%, with 85% for less than 3 days.
- This shows a clear correlation between the publish date and trending day. That videos tend to trend within days if not hours from the date published.
- In this perhaps an indication of viewer's attention span?

... MULTIPLE TRENDERS ...

```
channel_count_series = df['channel_title'].value_counts()
print(channel_count_series)
```

```
SET India                253
SAB TV                   248
VikatanTV               240
The Late Show with Stephen Colbert  230
Анатолий Шарий          224
...
MCDAVO                   1
Technical Patel          1
Shizarium                1
Deejay Maquina           1
Гельмут Вайссвальд       1
Name: channel_title, Length: 33965, dtype: int64
```

```
channel_count_series.describe()
```

```
count    33965.000000
mean       5.038981
std       12.047872
min        1.000000
25%        1.000000
50%        1.000000
75%        4.000000
max       253.000000
Name: channel_title, dtype: float64
```

```
multiple_trender = channel_count['VideoCount'] > 1
channel_count[multiple_trender]
```

	channel_title	VideoCount
0	SET India	253
1	SAB TV	248
2	VikatanTV	240
3	The Late Show with Stephen Colbert	230
4	Анатолий Шарий	224
...
16399	Sólo Ellas 4	2
16400	Коллекция Рецептов	2
16401	اعلانات مسلسل تركي	2
16402	메이리우 Korean Sister	2
16403	Mehmed Bir Cihan Fatihi	2

16404 rows × 2 columns

```
# From the channels with trending videos, almost 50% are concurrently trending more than one video
```

```
# From Previous calculations
Total_VideoCount_DF = 171149      # df.describe()
Total_ChannelCount_DF = 33965     # channel_count_series.describe()
Multiple_TrenderCount_Df = 16404  # multiple_trender view
```

```
Multiple_TrenderCount_Df/Total_ChannelCount_DF
```

0.48296776093036947

- If you aggregate the total number of videos trending by channel, almost 50% of those channels had more than one video trending. With the top 25% of channels, having at least 4 trending videos.
- Viewers tend to see videos from the same channel OR presumably, this property feeds into suggestion algorithms and viewers that see 1 video from a channel may be suggested to see another.
- This presents an opportunity for the Youtube Creator to have more than one video trending within a short timespan and for Youtube to encourage more trending videos.
- If a Creator achieves trending with 1 video, two things may happen: there is a potential correlation that either one of his/her existing videos can increase in viewership OR if the creator posts another video, that new video may benefit from the popularity of the previous.
- Assuming the creator does not proactively see this connection, Youtube (who is most likely already suggesting to viewers to see videos from the same channel) may now encourage via prompts or incentives more video uploads in a short period of time from the same creator to build on momentum

CONCLUSION

QUESTIONS TO EXPLORE

WHAT?

- Is there a correlation between CONTENT properties inputted by the creator and trendability?
 - YES!
 - Category "Entertainment" is by far the most frequently used by trending videos
 - The vast majority of videos (98%) are open to comments and ratings
 - The following content suggestions are in line with most trending videos:
 - Description length should be within 225 and 536 characters
 - Tag count should be within 7 and 25 tags and should not be left blank (as every trending video had at least 1)
- Is there a correlation between the creator's publishing TIMELINE and trendability?
 - YES!
 - The vast majority of trending videos (85%) achieved trending within 3 days of being published with 56% within the day.
 - Creators slightly favor weekdays over weekends to upload videos.

WHY?

- For Creators, are there any opportunities and or suggested behavior to encourage trendability?
 - YES!
 - A creator looking for trendability should use the upload content properties and timeline of past trending videos as a guideline.
- For Youtube, is it worthwhile to spend additional resources on the Youtube Creator? Are there any suggested changes to the existing platform?
 - YES!
 - Creators are the lifeline of Youtube. It is important they have the tools to do well and are guided by Youtube. Any support in training and or product development from Youtube to the creators can positively contribute to the continued success and usage of the platform.

WHAT ARE POTENTIAL NEXT STEPS FOR PROJECT?

- Qualitative Analysis: Particularly on Tags and Description
- Closer look at qualities provided for Multiple Trenders