Thinkful Data Science Fundamentals Capstone Analysis

Youtube Trending Video Statistics

Describing the Dataset

This dataset is the daily record from the top trending YouTube videos. Top 200 trending videos of a given day. Original Data was collected during 14th November 2017 & 5th March 2018(though, data for January 10th & 11th of 2017 is missing).

This dataset is an improved version of a series of parent datasets:

- The original dataset was Trending Youtube Video Statistics and Comments, which was collected using Youtube's API, and contained files for different countries and files for comments. These were linked by the "unique_video_id" field.
- A subsequent dataset was structurally improved and named Trending Youtube Video Statistics, it still was based off one file per country, with the difference that now the comment files were now integrated into each country's file.
- Finnally, this dataset YouTube Trending Video Statistics with Subscriber is a fork off the US data only. It was further improved in minor ways and was added a "Subscriber" field, by automatically gathering data for each video's subscribers using the author's own Python scripts.

Summary Statistics

Let's explore this data.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

```
from google.colab import drive
drive.mount('/content/gdrive')
```

```
Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).
```

```
import os
print(os.listdir())# we need to know the filename
```

```
['sample_data', '.config', 'gdrive']
```

```
# import the file using pandas read_csv function
usvids = pd.read_csv('gdrive/My Drive/Data Science/JupyterNotebooks/DataSets/youtube-trend-with-subscriber/USvideos_modified.csv', index_col='video_id')
# let's see the columns
usvids.head(10)
```

	last_trending_date	publish_date	publish_hour	category_id	channel_title	views	likes	dislikes	comment_count	comments_
video_id										
2kyS6SvSYSE	2017-11-20	2017-11-13	17	22	CaseyNeistat	2564903	96321	7972	22149	False
1ZAPwfrtAFY	2017-11-20	2017-11-13	7	24	LastWeekTonight	6109402	151250	11508	19820	False
5qpjK5DgCt4	2017-11-20	2017-11-12	19	23	Rudy Mancuso	5315471	187303	7278	9990	False
puqaWrEC7tY	2017-11-20	2017-11-13	11	24	Good Mythical Morning	913268	16729	1386	2988	False
d380meD0W0M	2017-11-19	2017-11-12	18	24	nigahiga	2819118	153395	2416	20573	False
gHZ1Qz0KiKM	2017-11-20	2017-11-13	19	28	iJustine	1038365	22594	2798	3142	False

	last_trending_date	publish_date	publish_hour	category_id	channel_title	views	likes	dislikes	comment_count	comments_
video_id										
39idVpFF7NQ	2017-11-18	2017-11-12	5	24	Saturday Night Live	2688797	19042	3059	2689	False
nc99ccSXST0	2017-11-19	2017-11-12	21	28	CrazyRussianHacker	1251577	28951	1146	2606	False
jr9QtXwC9vc	2017-11-15	2017-11-13	14	1	20th Century Fox	2671756	12699	505	1010	False
TUmyygCMMGA	2017-11-15	2017-11-13	13	25	Vox	635985	20721	2417	4111	False

10 rows × 22 columns

```
# Let's see how many rows and columns we have,
# and how many of them have distinct values.

# Total number of rows and columns.
print(usvids.shape)
# Are there any duplicates?
print("video_id: "+ str(usvids.index.nunique()))
# Number of unique values for each column.
print(usvids.nunique())
```

```
(4547, 22)
video_id: 4547
last_trending_date              110
publish_date                    211
publish_hour                     24
category_id                      16
channel_title                  1905
views                          4532
likes                          3949
dislikes                       1842
comment_count                  2645
comments_disabled                 2
ratings_disabled                  2
tag_appeared_in_title_count      18
tag_appeared_in_title             2
title                          4540
tags                           4190
description                    4415
trend_day_count                  14
trend.publish.diff              127
trend_tag_highest               111
trend_tag_total                1256
tags_count                       65
subscriber                     1831
dtype: int64
```

Observations from the unique number of values for each column:

- There are no video duplicates, since the number of unique video ID's is the same as the total number of videos.
- The publish date has 211 unique values from a total of 4547 videos. This means that many videos were published on the same dates. It would be interesting to see if certain dates or seasons were more popular than others for publishing videos. Is there a correlation between publish date and views or days trending?
- The last trending date has 110 unique values, which is even less than the unique values in the publish date. This means there were similar dates after which many of these videos stopped trending. This would be another interesting variable to plot. Are there dates of the year where people simply watched less videos, thus causing these to stop trending?

```
#Looking for missing values and type of our data
usvids.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 4547 entries, 2kyS6SvSYSE to Eouvsy8JdLU
Data columns (total 22 columns):
last_trending_date            4547 non-null object
publish_date                  4547 non-null object
publish_hour                  4547 non-null int64
category_id                   4547 non-null int64
channel_title                 4547 non-null object
```

```
views                          4547 non-null int64
likes                          4547 non-null int64
dislikes                       4547 non-null int64
comment_count                  4547 non-null int64
comments_disabled              4547 non-null bool
ratings_disabled               4547 non-null bool
tag_appeared_in_title_count    4547 non-null int64
tag_appeared_in_title          4547 non-null bool
title                          4547 non-null object
tags                           4339 non-null object
description                    4458 non-null object
trend_day_count                4547 non-null int64
trend.publish.diff             4547 non-null int64
trend_tag_highest              4547 non-null int64
trend_tag_total                4547 non-null int64
tags_count                     4547 non-null int64
subscriber                     4525 non-null float64
dtypes: bool(3), float64(1), int64(12), object(6)
memory usage: 723.8+ KB
```

Observations from info on missing values:

For the most part the data is complete. Only a small fraction of videos have no tags or description. Could there be a relation between the number of tags and the number of views? If there is a relation, are there any other features that these videos with missing values have in common?

```
# Let's get a statistical summary of each numerical column.
usvids.describe()
```

	publish_hour	category_id	views	likes	dislikes	comment_count	tag_appeared_in_title_count	trend_day_count	trend.publish.
count	4547.000000	4547.000000	4.547000e+03	4.547000e+03	4.547000e+03	4.547000e+03	4547.000000	4547.000000	4547.000000
mean	13.503189	20.416538	1.265665e+06	3.919696e+04	2.616788e+03	4.938788e+03	2.961293	4.830658	34.429954
std	6.548420	7.309226	4.526133e+06	1.419793e+05	3.662803e+04	3.110122e+04	2.482547	2.614707	247.514298
min	0.000000	1.000000	5.590000e+02	0.000000e+00	0.000000e+00	0.000000e+00	0.000000	1.000000	0.000000
25%	8.000000	17.000000	9.089650e+04	1.486000e+03	7.600000e+01	2.260000e+02	1.000000	3.000000	5.000000
50%	15.000000	24.000000	3.188400e+05	7.397000e+03	2.910000e+02	8.540000e+02	3.000000	5.000000	6.000000
75%	18.000000	25.000000	1.006673e+06	2.557550e+04	1.023000e+03	2.862500e+03	4.000000	7.000000	7.000000
max	23.000000	43.000000	1.493761e+08	3.093544e+06	1.674420e+06	1.361580e+06	18.000000	14.000000	4215.000000

Observations from Statistical Description:

- Tag Appeared in Title Count. Most of the trending videos have an average of 2.9 tags included in their title. The 75th percentile of videos have 4 tags in title. The maximum number of tags in a title was 18, a big jump from the 75th percentile. This is clearly an outlier and would be interesting to see which video this is and if there are other similar deviations. Did this video get more views/likes? Are tags in title a good predictor to views & likes?
- Trend Publish Difference. In average, a video takes 34 days to trend from the date it is published. Except that we have a major outlier of 4215 days. This means there was a trending video that was published over 10 years ago. This might be affecting the mean and the standard deviation of 247. If we removed this outlier, what would be the mean and standard deviation?
- Tags Count. Most trending videos had an average of 19 tags total. Some had 0 tags and at least one had 69 tags. This is another outlier, since the 75th percentile is at only 29 tags. It's a jump of twice the number of tags in 3/4 of all the trending videos.

Analytical Questions:

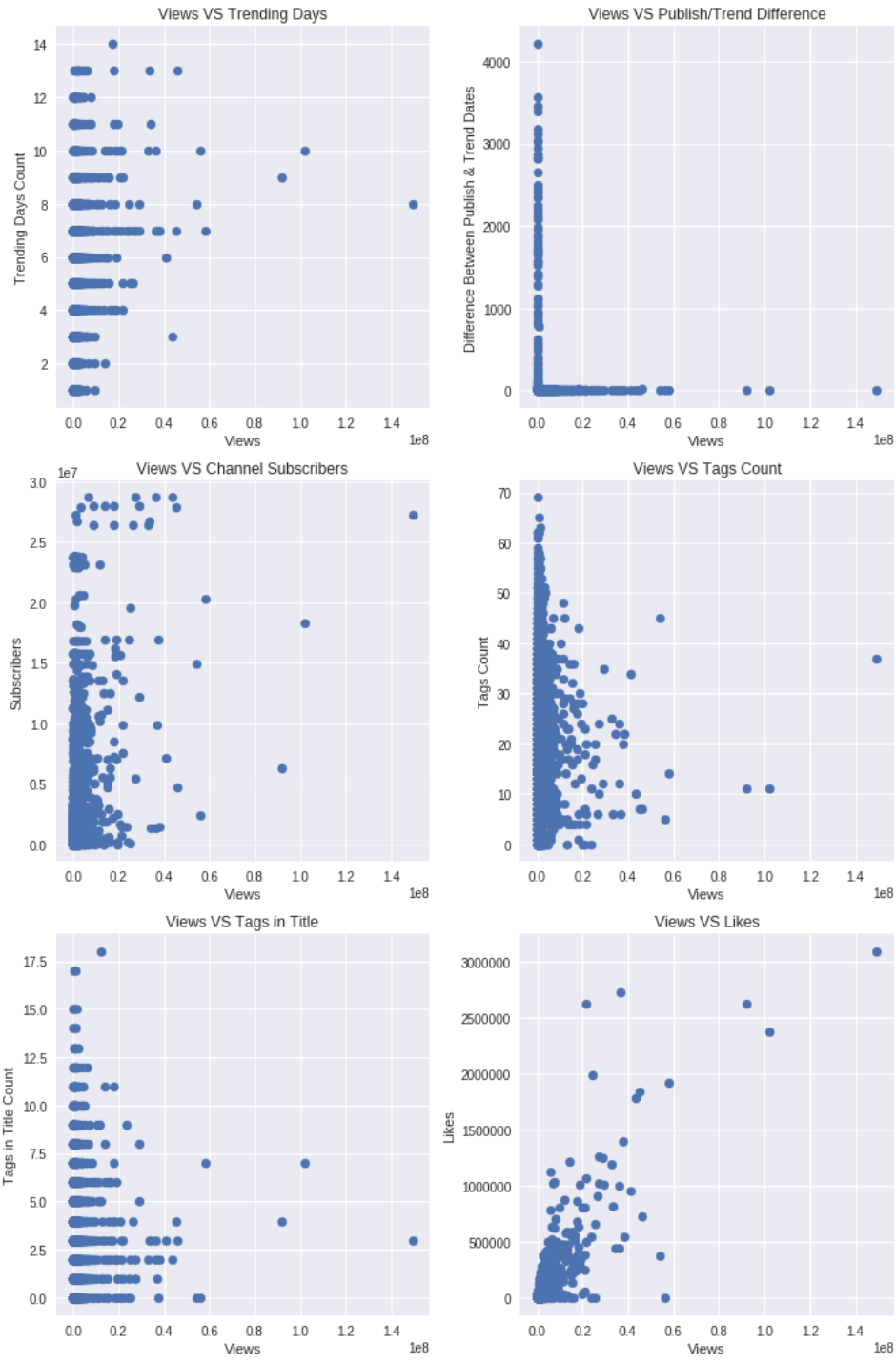
Question 1: What variable has the strongest correlation with the number of views in this dataset?

Visualizing this will help us gain a better understanding of this data, and might clarify some of the questions raised in the previous sections.

Let's plot the meaningful variables against the number of views...

```
plt.figure(figsize=(10,15))

plt.subplot(3,2,1)
plt.scatter(usvids.views, usvids.trend_day_count)
plt.xlabel('Views')
plt.ylabel('Trending Days Count')
plt.title('Views VS Trending Days')

plt.subplot(3,2,2)
plt.scatter(usvids.views, usvids['trend.publish.diff'])
plt.xlabel('Views')
plt.ylabel('Difference Between Publish & Trend Dates')
plt.title('Views VS Publish/Trend Difference')

plt.subplot(3,2,3)
plt.scatter(usvids.views, usvids['subscriber'])
plt.xlabel('Views')
plt.ylabel('Subscribers')
```

```
plt.title('Views VS Channel Subscribers')

plt.subplot(3,2,4)
plt.scatter(usvids.views, usvids['tags_count'])
plt.xlabel('Views')
plt.ylabel('Tags Count')
plt.title('Views VS Tags Count')

plt.subplot(3,2,5)
plt.scatter(usvids.views, usvids['tag_appeared_in_title_count'])
plt.xlabel('Views')
plt.ylabel('Tags in Title Count')
plt.title('Views VS Tags in Title')

plt.subplot(3,2,6)
plt.scatter(usvids.views, usvids['likes'])
plt.xlabel('Views')
plt.ylabel('Likes')
plt.title('Views VS Likes')

plt.tight_layout()
plt.show()
```



Analysis based on Visual Summary

Views VS Trending Days.

The videos that had the least number of trending days also had the least number of views. Although some videos also trended from 12-14 days and still had less views than some videos which only trended 10 days. Up to a certain upper limit, more trending days correlates with more views per video.

Views VS Publish/Trend Difference

This plot is very interesting because it shows that although many videos trended after years of being published, these old-bloomers only reached a small fraction of views compared to the videos which trended sooner. It would be interesting to get a closer view in both directions and determine if there is a cutoff time after which a trending video will not get as many views. According to this graph, the days-to-trend variable is a good predictor of whether a videos will get many views. I.e. The videos that get the most views all trend very soon after their publish date.

Views VS Channel Subscribers

Trending videos get views regardless of the number of subscribers to their channel. There is a very weak correlation between the most viewed videos having the most subscribers, however.

Views VS Tags Count

The videos with the most views have a count of 5-30 tags. More tags than that doesn't help a video get more views. This answers our previous question about the outlier with 69 tags. It is clear that the video with the most tags has very few views.

Views VS Tags in Title

The most watched videos had 3-7 tags in their title. Videos with more than 7 tags in title had the fewest views, and videos with no tags in their title still managed to get a decent amount of views. More tags in title means more views untill you reach 7 tags or more.

Views VS Likes

The most watched videos also had the most likes. This makes sense because people can only hit the 'Like' button once they've started seeing a video. This variable has the strongest correlation with views of all the ones examined here.

Question 2: What's the biggest different factor that sets apart the segment of videos whose ratings & comments were disabled by their publisher?

Specifically...

- Do these videos get more views/tags?
- Do they trend for more or less days?
- Do they get more comments?
- Do they take longer to trend, from day of publishing?

```
# We'll divide out dataset by wether the videos had their ratings disabled or not.
# We'll get a statistical summary of the relevant columns.
usvids.groupby('ratings_disabled').describe()
[['views','tag_appeared_in_title_count','trend_day_count','trend.publish.diff','comment_count']]
```

	views								tag_appeared_in_title_count		...	trend.publish.d	
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max
ratings_disabled													
False	4522.0	1.254778e+06	4.459278e+06	559.0	91338.75	319203.5	1007402.0	149376127.0	4522.0	2.966387	...	7.0	4215.(
True	25.0	3.234856e+06	1.142095e+07	2093.0	24737.00	139068.0	839375.0	56111957.0	25.0	2.040000	...	8.0	409.0

2 rows × 40 columns

```
# We'll divide out dataset by wether the videos had their comments disabled or not.
# We'll get a statistical summary of the relevant columns.
usvids.groupby('comments_disabled').describe()
[['views','likes','dislikes','tag_appeared_in_title_count','trend_day_count','trend.publish.diff']]
```

	views								likes		...	trend_day_count	
	count	mean	std	min	25%	50%	75%	max	count	mean	...	75%	max
comments_disabled													
False	4471.0	1.258650e+06	4.481281e+06	559.0	92535.5	320792.0	1010413.0	149376127.0	4471.0	39701.713711	...	7.0	14.0
True	76.0	1.678329e+06	6.682966e+06	748.0	23921.5	149300.0	862810.5	56111957.0	76.0	9502.815789	...	7.0	13.0

2 rows × 48 columns

Observations:

Based on the dataframes above...

Videos with their ratings disabled had, in average: Twice the number of views, less tags appearing in their title, and trended for more days than their counterparts. But they took slightly longer to trend and had considerably less comments than their counterparts.

Videos with their comments disabled had, in average: More views than their counterparts, only by a small margin. Four-times less likes than their counterparts. Surprisingly, they also have less dislikes. Less tags appeared in their titles, they trended for longer, and trended way quicker from their publishing date, than their counterparts.

Analysis:

Videos with their ratings disabled.

The most surprising fact is that these videos had an average number of views more than twice the average views for all other videos.

ratings enabled: 1,254,778 average views

ratings disabled: 3,234,856 average views

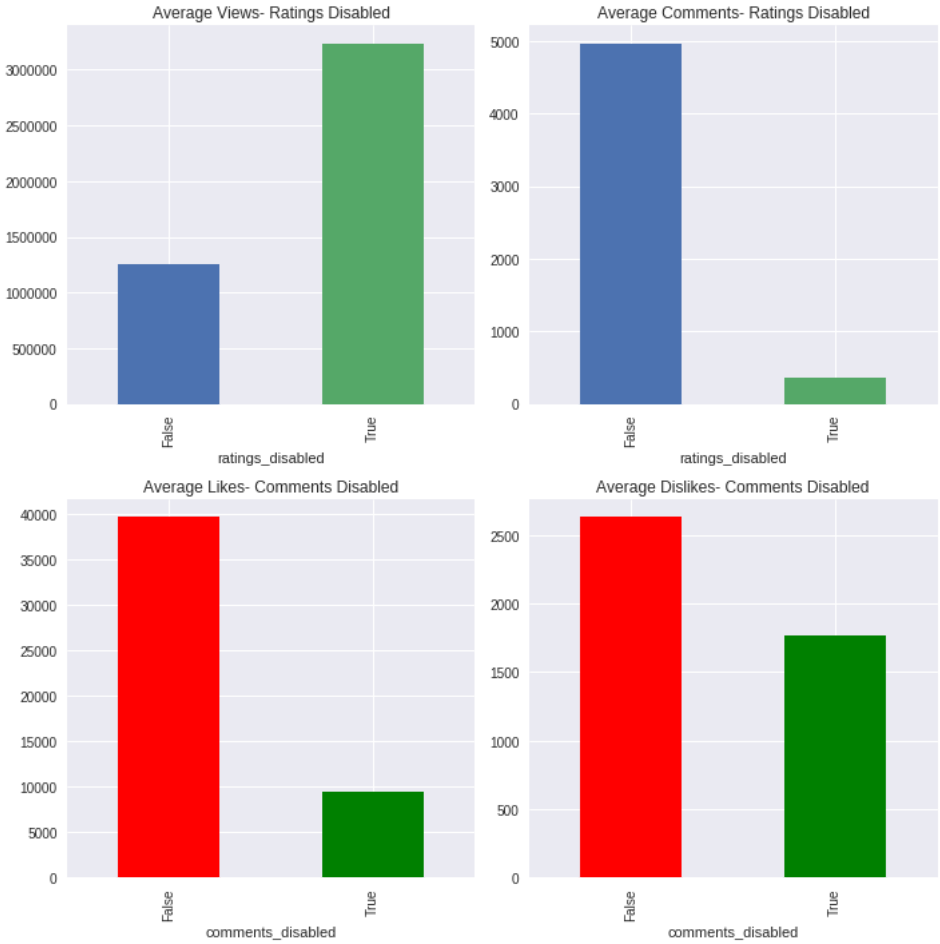
However, the sample size of this category is only 25 videos, compared to 4522 for the rest. Therefore, there is a high risk of bias since this average is coming from a very small sample. The standard deviation is also small, but the small count might also be making this calculation less trustworthy.

To deal with this uncertainty, we could apply some statistical analysis. On one hand, the higher number of views could be due to a concrete difference between these two groups of videos. This would suggest that there is a commonality in the videos which got their ratings disabled, which usually results in a larger view count. On the other hand, the higher number of views could be due to the mere coincidence, therefore not having statistical significance. A good way to investigate this uncertainty would be by applying the principles of the Central Limit Theorem and performing a T-Test and looking at the P-Value, to assess the likelihood that this mean difference would be the result of an actual difference in the population.

Videos with their comments disabled.

I was expecting to see these videos have more dislikes, since controversial content could draw more negative attention and this could be a good reason for publishers to block comments. Maybe there are other reasons for them doing this. However, these videos also got less likes by 4:1. So for some weird reason, people are watching these videos at a higher than average rate than their counterparts, but they are not engaging with them as much. (If we measure engagement by the number of likes, dislikes or comments). This can also be said about the videos with ratings disabled, which had significantly less comments. Contrary to my expectation, it seems like disabling ratings results in less comments, and disabling comments results in less ratings. Perhaps people are habituated to have both engagement avenues at their disposal and being denied either of them leads them to engaging less with the one that's left available. Could we prove this statistically?

```
plt.figure(figsize=(20,20))

plt.subplot(2,2,1)
usvids.groupby('ratings_disabled').agg(np.mean)['views'].plot(kind='bar',figsize=(10,10))
plt.title('Average Views- Ratings Disabled')

plt.subplot(2,2,2)
usvids.groupby('ratings_disabled').agg(np.mean)['comment_count'].plot(kind='bar',figsize=(10,10))
plt.title('Average Comments- Ratings Disabled')

plt.subplot(2,2,3)
usvids.groupby('comments_disabled').agg(np.mean)['likes'].plot(kind='bar',figsize=(10,10),color=['red','green'])
plt.title('Average Likes- Comments Disabled')

plt.subplot(2,2,4)
usvids.groupby('comments_disabled').agg(np.mean)['dislikes'].plot(kind='bar',figsize=(10,10),color=['red','green'])
plt.title('Average Dislikes- Comments Disabled')

plt.tight_layout()
plt.show()

```



Question 2.2: Do videos with ratings disabled statistically different from their counterparts? I.e. Do they really get twice the number of views?

```
# Perform T-Test and find P-Value
# Apply the natural logarithm to normalize the distributions
rat_dis = np.log(usvids[usvids.ratings_disabled == True].views)
rat_en = np.log(usvids[usvids.ratings_disabled == False].views)
from scipy.stats import ttest_ind
ttest_ind(rat_en, rat_dis, equal_var=False)
```

```
Ttest_indResult(statistic=1.4094641992052217, pvalue=0.17145653730660942)
```

T-Test Analysis:

The t-value of mean views between videos with disabled and enabled ratings is 1.40. This tells us that the difference between the means is 1.40 times greater than the combined standard error of the samples. Values closer to zero indicate the difference is most likely coincidental. At 1.40, their difference is mildly beyond the standard error, which is very inconclusive. However, we have a p-value of 0.171. It is conventionally accepted that a p-value of 0.05 is the cutoff for statistical significance, with higher values suggesting the null hypothesis. At 0.171, the p-value is bordering the margin for significance, but it is not close enough to be conclusive.

Conclusion: We can conclude that although our samples had a large difference between their mean views, there is no conclusive difference between these videos. However, the T and P-Values do suggest a strong trend in that direction.

Understanding the T-test in the Context of this Dataset:

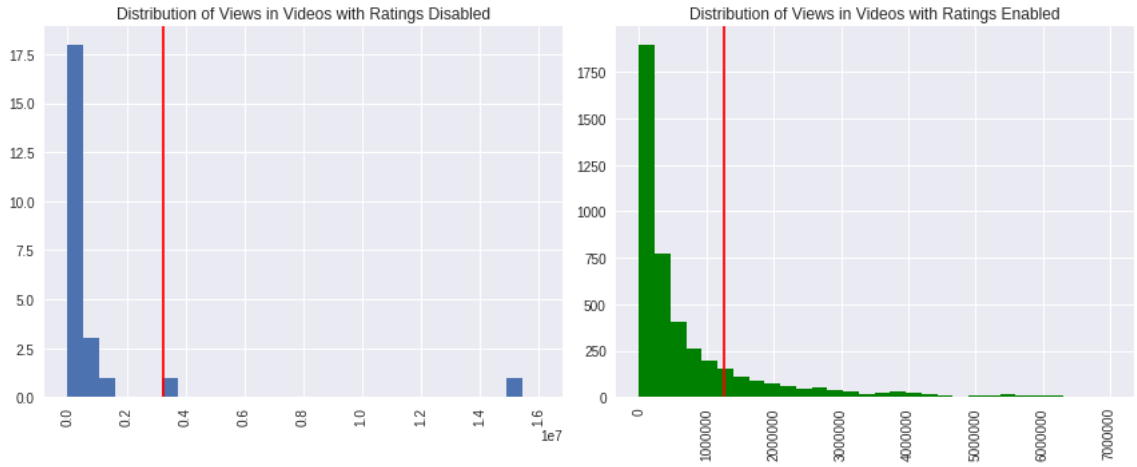
The application of the T-Value test to these samples could serve as an inference for the total population of videos in Youtube. Namely, this would give us a hint to determine if the rest of the videos in Youtube that aren't in this dataset would follow the assumption that: "Videos with ratings disabled usually get twice the number of views as videos with ratings enabled."

However, the limitations of this calculation would also be biased by the fact that this sample only contains the Youtube videos that Youtube determined were trending during a given year. Therefore, Youtube's total video population might be vastly different than this sample of trending videos. In other words, this dataset might not be representative of the whole of Youtube's content.

Question 2.3: What can the distributions of these groups tell us about their mean differences?

```
plt.figure(figsize=(12,5))

plt.subplot(1,2,1)
plt.hist(usvids[usvids.ratings_disabled].views,range=(1,16000000),bins=30)
plt.axvline(np.mean(usvids[usvids.ratings_disabled].views),color='red')
plt.title('Distribution of Views in Videos with Ratings Disabled')
plt.xticks(rotation=90)

plt.subplot(1,2,2)
plt.hist(usvids[~usvids.ratings_disabled].views,range=(1,7000000),color='green',bins=30)
```

```
plt.title('Distribution of Views in Videos with Ratings Enabled')
plt.axvline(np.mean(usvids[~usvids.ratings_disabled].views),color='red')
plt.xticks(rotation=90)


plt.tight_layout()
plt.show()
```

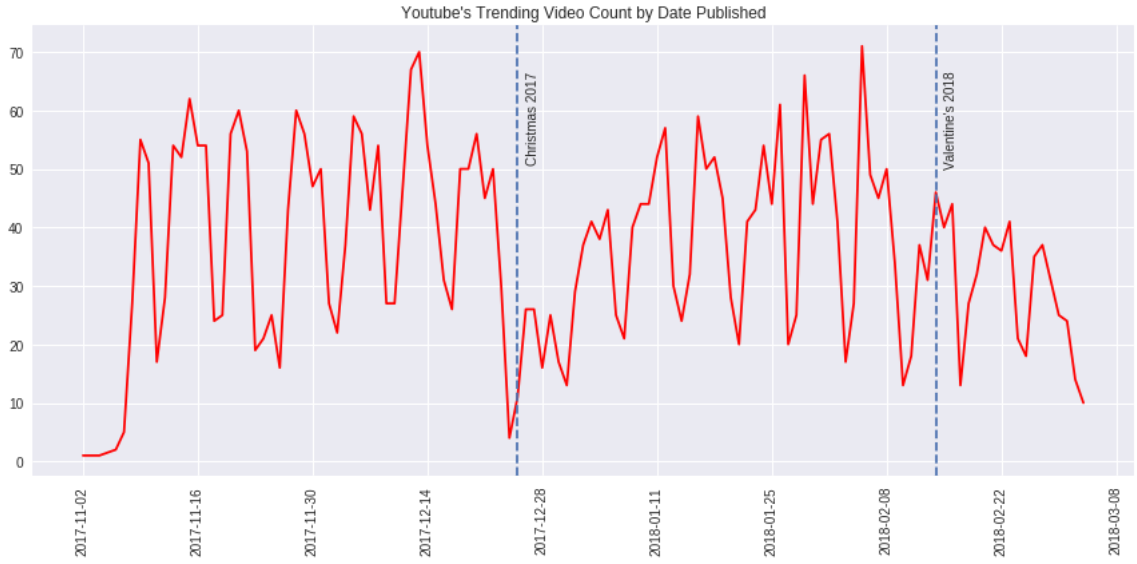


Answer:

Clearly there are two outliers in the videos with ratings disabled, one of which is skewing the mean views to a point far above the rest. Without the outlier, the mean would surely correlate more with the other segment of videos with ratings enabled.

Question 3: What were the most popular dates to publish videos?

```
usvids.publish_date = pd.to_datetime(usvids.publish_date)
# Let's plot these dates
pop_dates = usvids['publish_date'].value_counts().sort_index()
pop_dates = pop_dates[pop_dates.index > '2017-11-01']

plt.figure(figsize=(12,6))

plt.plot(pop_dates.index,pop_dates.values, color='red')
plt.xticks(rotation=90)
plt.title('Youtube\'s Trending Video Count by Date Published')
plt.axvline('2017-12-25',linestyle='dashed')
plt.axvline('2018-02-14',linestyle='dashed')
plt.text('2018-02-15',65,"Valentine's 2018",rotation=90)
plt.text('2017-12-26',65,"Christmas 2017",rotation=90)

plt.tight_layout()
plt.show()
```




```
# Let's see the most recurring publishing dates
print(usvids['publish_date'].value_counts().head(10))

```

```
2018-02-05    71
2017-12-13    70
2017-12-12    67
2018-01-29    66
2017-11-15    62
2018-01-26    61
2017-11-28    60
2017-11-21    60
2018-01-16    59
2017-12-05    59
Name: publish_date, dtype: int64
```

Answer:

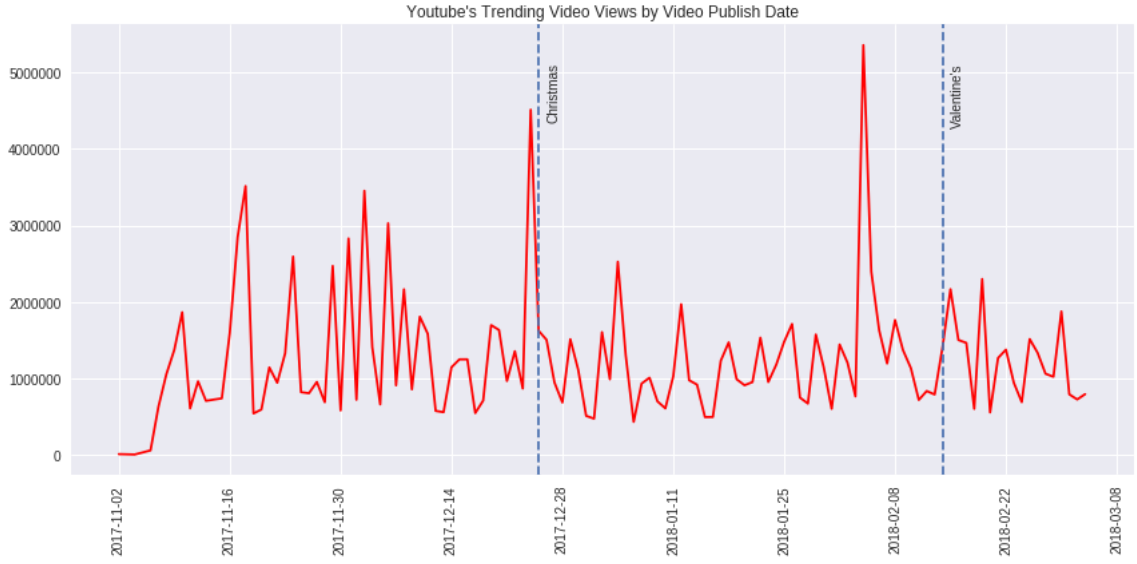
- The most videos published in the same date were from February 5th, 2018. Could this have been in anticipation to Valentine's day?
- The second and third most popular days for video publishers were about two weeks before Christmas. Could we verify if these videos had a seasonal theme?
- The 2 most repeated last trending dates are a few days after Valentine's Day and the Christmas/New Year's holiday combo. These could be the same large number of videos that were published before these dates.

Question 3.2: Are the videos from the most popular publishing dates getting more views than the videos from low upload rate seasons?

Hypothesis:

Based on the timeline above, there are two well-defined seasons in which more videos are uploaded to Youtube. They are both followed by periods of low upload rates. If we analyze the views attributed to videos based on their publish date, we might have a better understanding of this trend. If people publish more videos in certain seasons than others, perhaps this correlates with when people are watchin more videos. After all, views are what video publishers are after. If we assume that the trending videos in this dataset were created by professional Youtubers who are trying to maximize their channel's views and profits, then we should also assume that the periods when they are more active are also the periods when they are getting more views as reward for their work.

```
#Let's visualize views by publish date of trending videos
dates_views = usvids.groupby('publish_date').agg(np.mean).sort_values('views',ascending=False).views.sort_index()
dates_views = dates_views[dates_views.index > '2017-11-01']
plt.figure(figsize=(12,6))

plt.plot(dates_views.index,dates_views.values, color='red')
plt.xticks(rotation=90)
plt.title('Youtube\'s Trending Video Views by Video Publish Date')
plt.axvline('2017-12-25',linestyle='dashed')
plt.axvline('2018-02-14',linestyle='dashed')
plt.text('2018-02-15',5000000,"Valentine's",rotation=90)
plt.text('2017-12-26',5000000,"Christmas",rotation=90)


plt.tight_layout()
plt.show()
```



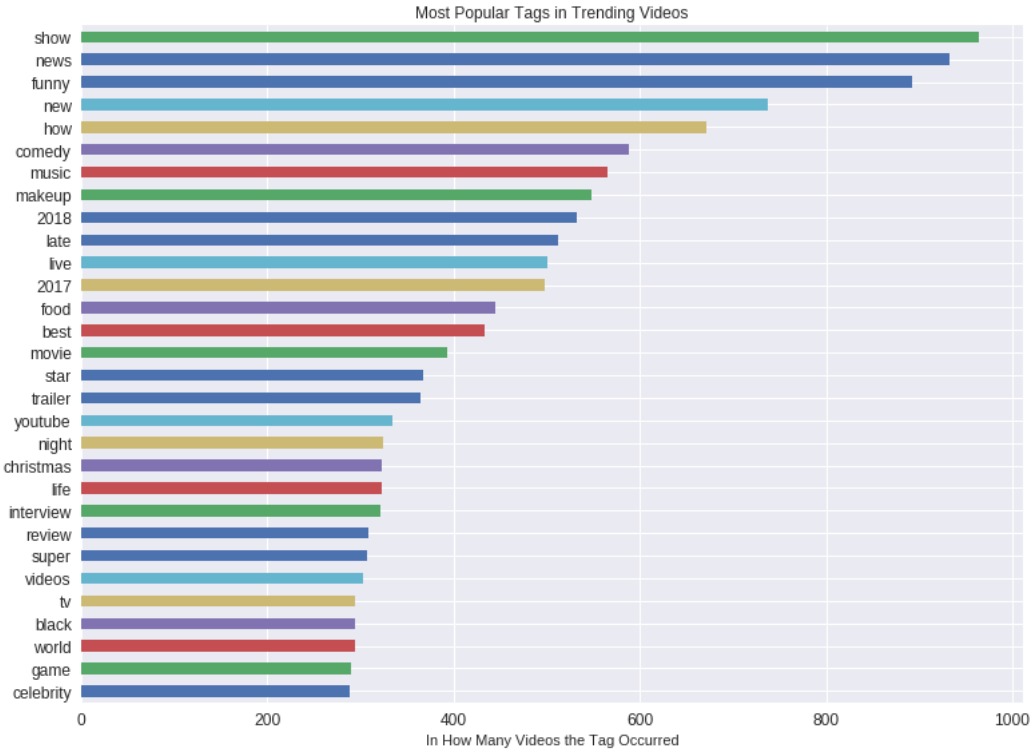

```
# let's list the publishing dates with the highest number of views
dates_views.sort_values(ascending=False).head()
```

```
publish_date
2018-02-04    5.360935e+06
2017-12-24    4.514713e+06
2017-11-18    3.515628e+06
2017-12-03    3.453279e+06
2017-12-06    3.028438e+06
Name: views, dtype: float64
```

Answer:
Indeed, the videos published during the high-upload rate periods received a higher mean number of views. Trending videos published on the 5th of Feb, 2018 received an average of 5.3 million views. Videos published on December 24th, 2017 received 4.5 million views in average.

Question 4: What are the Most Popular Tags in Trending Videos?

```
# separate each word in the tags column and add them onto a list of strings
# first split by '|' and send to a list.
tags = usvids.tags.str.split('|').tolist()
# then get rid of anything that isn't a list
tags = [x for x in tags if type(x) == list]

# that gave us a list of lists (of strings), so we must separate the items in each
tags2 = []
tags3 = []
for item in tags:
    for string in item:
        # get rid of numbers and other types
        if type(string) == str:
            tags2.append(string)

def meaningless(x):
    words = ['to','the','a','of','and','on','in','for','is','&','with','you','video']
    return x in words

# now let's split these strings by the spaces between words
for multiple in tags2:
    singles = multiple.split()
    # then let's add these cleaned tags to the final list
    for tag in singles:
        # now let's make everything lowercase and get rid of spaces
        tag = tag.strip()
        tag = tag.lower()
        # now let's remove the meaningless tags
        if not meaningless(tag):
            tags3.append(tag)

# let's bring that into a dataframe
tagsdf = pd.DataFrame(tags3,columns=['tags'])
# then count the values
tagcounts = tagsdf.tags.value_counts()

# now preparing a bar chart representing the top values
tagcountslice = tagcounts[:30].sort_values()
tagcountslice.plot(kind='barh',title='Most Popular Tags in Trending Videos',grid=True,fontsize=12,figsize=(11,8))
plt.xlabel('In How Many Videos the Tag Occurred')

plt.tight_layout()
plt.show()
```



Question 4.1: Which tags received the most views?

To answer this question, for each tag we will count the views in every video where it appears. Since videos usually have more than one tag, their views will count toward each of the tags.

```
# clean raw tags for each video and append them to a new list
# make another list with the views of each video
cleantagslist = []
tagsviews = []
count = 0
for rawtags in usvids.tags:
    try:
        cleantags = " ".join(" ".join(" ".join(" ".join(rawtags.split('|')).split()).split('(')).split(')')).strip().lower()
        cleantagslist.append(cleantags)

        count += 1
        tagsviews.append(usvids.views[count-1])
    except:
        ValueError
# let's show the cleaned tags for the first 5 videos
cleantagslist[:5]
```

```
['shantell martin',
 'last week tonight trump presidency last week tonight donald trump john oliver trump donald trump',
 "racist superman rudy mancuso king bach racist superman love rudy mancuso poo bear black white official music video iphone x by pineapple lelepons hannahstocking rudymancuso inanna anwar sarkis shots shotsstudios alesso anitta brazil getting my driver's license lele pons",
 'rhett and link gmm good mythical morning rhett and link good mythical morning good mythical morning rhett and link mythical morning season 12 nickelback lyrics nickelback lyrics real or fake nickelback nickelback songs nickelback song rhett link nickelback gmm nickelback lyrics  website category  nickelback  musical group  rock music lyrics chad kroeger music  industry mythical gmm challenge comedy funny the betrayal the betrayal act iii how you remind me',
 'ryan higa higatv nigahiga i dare you idy rhpc dares no truth comments comedy funny stupid fail']
```

```
# create a dataframe containing each video's cleaned tags and views
cleantagsdf = pd.DataFrame(columns=['tags','views'])
cleantagsdf['tags'] = cleantagslist
cleantagsdf['views'] = tagsviews
# now we have those cleaned tags in a dataframe along with their video views
cleantagsdf.head()
```

	tags	views
0	shantell martin	2564903
1	last week tonight trump presidency last week t...	6109402
2	racist superman rudy mancuso king bach racist ...	5315471
3	rhett and link gmm good mythical morning rhett...	913268

	tags	views
4	ryan higa higatv nigahiga i dare you idy rhpc ...	2819118

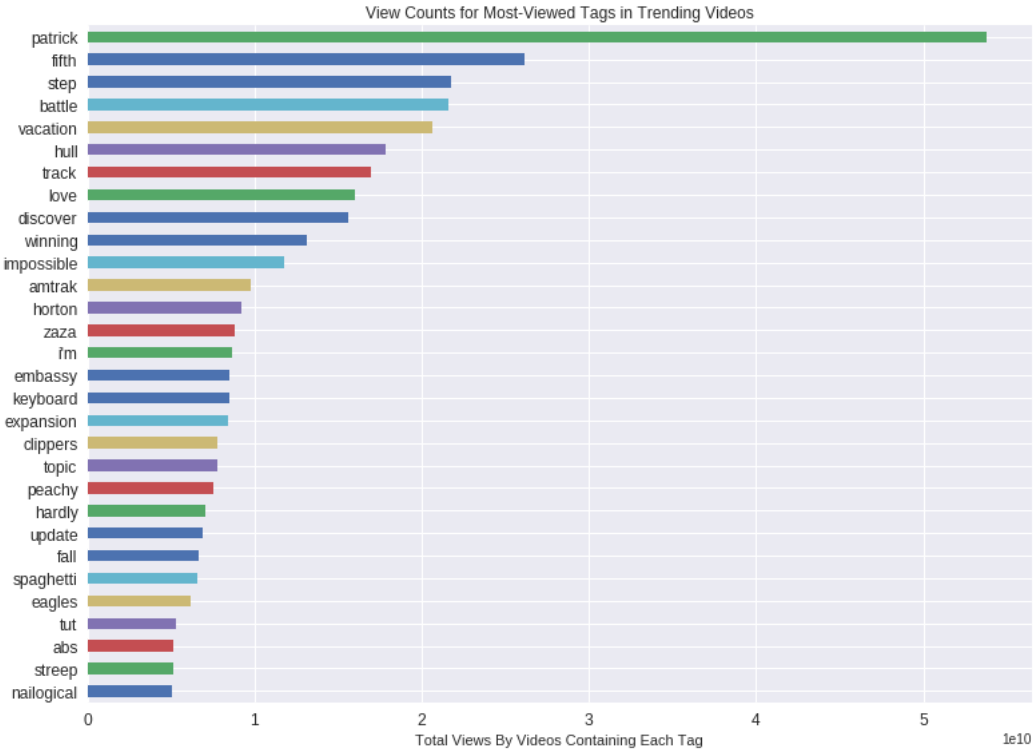
```
# make a list of unique tags. no repeated tags. no meaningless words
df = pd.DataFrame(" ".join(cleantagslist).split(),columns=['tags'])
uniquetagslist = df.tags.value_counts().keys()
uniquetagslist = [tag for tag in uniquetagslist if not meaningless(tag)]

# make a dataframe with each unique tag as the index and zeros on the 'views' column
# we will use this dataframe to count the views for each unique tag
tagsviewsdf = pd.DataFrame(index=uniquetagslist,columns=['views'])
tagsviewsdf = tagsviewsdf.views.fillna(0)
tagsviewsdf = pd.DataFrame(tagsviewsdf)

# show the dataframe where we'll count the views for each tag
tagsviewsdf.head()
```

	views
show	0
news	0
funny	0
new	0
how	0

```
# count the views for each unique tag and add them to above's dataframe
for unique in uniquetagslist:
    index = 0
    for tag in cleantagsdf.tags:
        index += 1
        if unique in tag:
            tagsviewsdf.views[index-1] += cleantagsdf.views[index-1]

# show the first tags along with their view count
tagsviewsdf.head()
```

	views
show	128245150
news	745347044
funny	1647796010
new	277633472
how	377761812

```
# Now creating a bar chart of the top-viewed tags along with their view counts
tagsviewslice = tagsviewsdf.views.sort_values(ascending=False)[:30].sort_values()
tagsviewslice.plot(kind='barh',title='View Counts for Most-Viewed Tags in Trending Videos',grid=True,fontsize=12,figsize=(11,8))
plt.xlabel('Total Views By Videos Containing Each Tag')

plt.tight_layout()
plt.show()
```



Observations:

This was the most surprising answer to find in this report. Some of these other tags are completely unknown terms to me. But they are insightful because they don't represent what youtubers think is popular. Instead, they represent what people actually watch the most in Youtube.

Further Research Proposal

This analysis has perhaps raised more questions than the ones it answered. In fact, each one of the questions analyzed could be further explored to indefinite lengths. Here are some ideas for further research regarding this data.

1: Create a Dataset that contains an even balance of all kinds of Youtube videos.

Scrapping a dataset like this would allow to make calculations that could better describe youtube's content as a whole, instead of studying only those videos who have been carefully crafted to be famous.

2: Include a column that includes the comments section of each video and analyze viewer's emotions with NLP.

3: Use Machine Learning to predict number of views based on video tags and uploader demographics. (Where they're from, subscribers, profile, other videos they have, etc.)