This Presentation is a Clickbait:

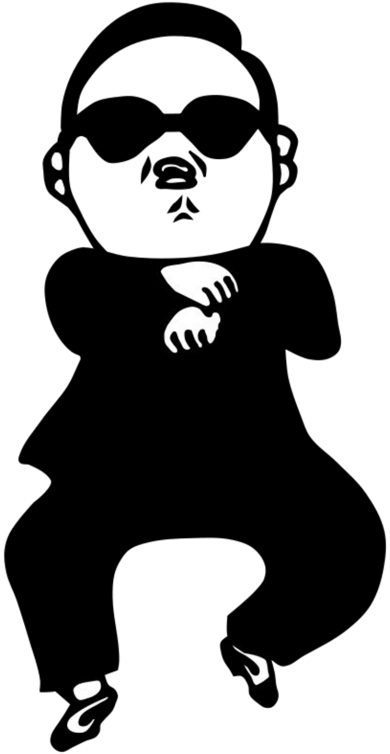
Analyzing Trending Videos From 11.14.17 - 06.14.18

Group 6: Mark Gu, Max Izotov, Sabikha Khatun, Jeongdae (JD) Kwak, Samuel Okunola

https://youtu.be/dQw4w9WgXcQ

Agenda

1. Reading & Cleaning the Data
2. Likes, Dislikes, Comments Relationships
3. Analysis of Trending Categories
4. How Reactions Affect What is Trending
5. Relationship of Dislikes and Removed Videos
6. Worldwide Youtube User Patterns
7. Q & A



Cleaning up Data

```
us_df = pd.read_csv("USvideos.csv", encoding="ISO-8859-1")

# reading the json file
    file = open("us_category_id.json")
    us_json = json.load(file)
# replacing the category id with the category name
    length = len(us_json["items"])
    us_df["category_id"] = us_df["category_id"].astype(str)
    for x in range(length):
        id_number = us_json["items"][x]["id"]
        category_name = us_json["items"][x]["snippet"]["title"]
        us_df["category_id"] = us_df["category_id"].replace({f"{id_number}": f"{category_name}"})
```

channel_title	category_id	publish_time	tags	views	likes	dislikes	comment_count
CaseyNeistat	22	2017-11-13T17:13:01.000Z	SHANtell martin	748374	57527	2966	15954
astWeekTonight	24	2017-11-13T07:30:00.000Z	last week tonight trump presidency|"last week ...	2418783	97185	6146	12703
Rudy Mancuso	23	2017-11-12T19:05:24.000Z	racist superman|"rudy"|"mancuso"|"king"|"bach"...	3191434	146033	5339	8181

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	views	likes
0	2kyS6SvSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	People & Blogs	2017-11-13T17:13:01.000Z	SHANtell martin	748374	57527
1	1ZAPwfrtAFY	17.14.11	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	Entertainment	2017-11-13T07:30:00.000Z	last week tonight trump presidency|"last week ...	2418783	97185

Translating Data from Foreign Language

Translating the CSV File

ru_df = pd.read_csv('resources/RUvideos.csv', encoding='utf-8')

encoding = ‘latin1’

encoding = ‘utf-16’ / ‘utf-32’ / ‘utf-64’

engine = ‘python’

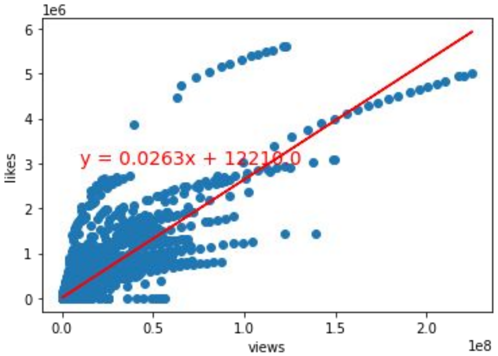
Unzipping the .csv.zip File

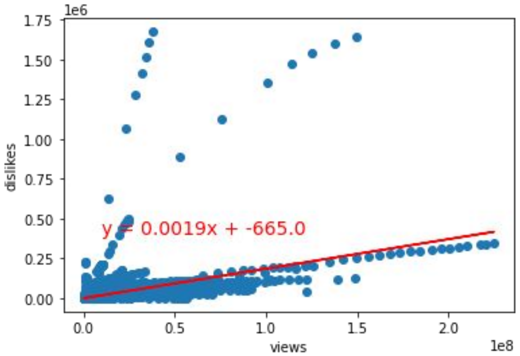
1. Unzip in properties on machine
2. !Unzip in python

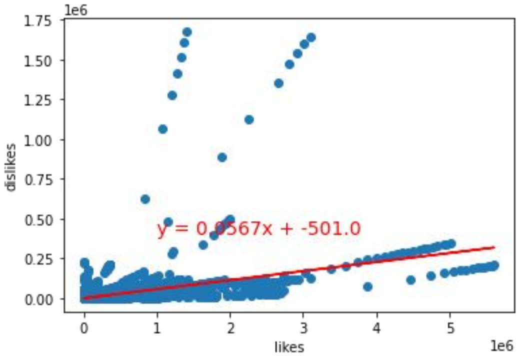
18.14.06	[ENG SUB] BTS PROM PARTY 2018 Intro + 2nd Gran...	DaisyxBTS 07
18.14.06	ОБЗОР ВАННОЙ КОМНАТЫ🛁/ ТУАЛЕТНОЙ КОМНАТЫ🚽 + ДЕК...	Ксюша Лебедева

Linear Regression: USA

Views, Likes and Dislikes

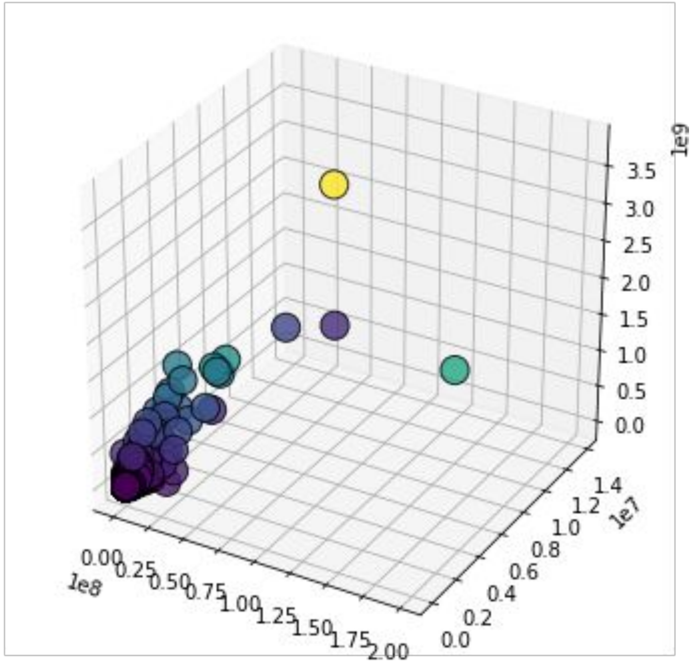






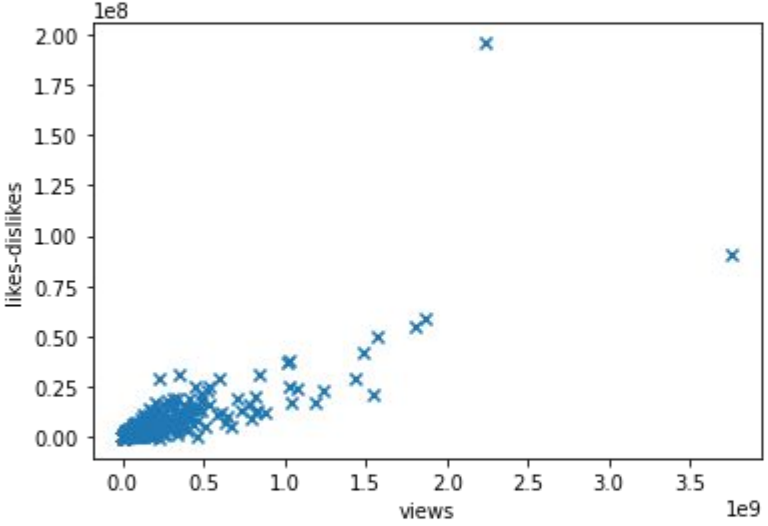
Dive into Channels: USA

channel_title	likes	dislikes	views
Logan Paul Vlogs	31545290	13847251	484356303
YouTube Spotlight	20173324	10924092	791388476
ChildishGambinoVEVO	96700818	6054434	3758488765
Call of Duty	11553594	5644083	315404711
ibighit	199247121	3467306	2235906679
jypentertainment	44900910	2482131	1486972132
TaylorSwiftVEVO	39292840	2127542	1010955662
ArianaGrandeVevo	52170970	1931230	1576959172
MalumaVEVO	23278380	1757948	1551515831
KatyPerryVEVO	8660466	1669622	273333649



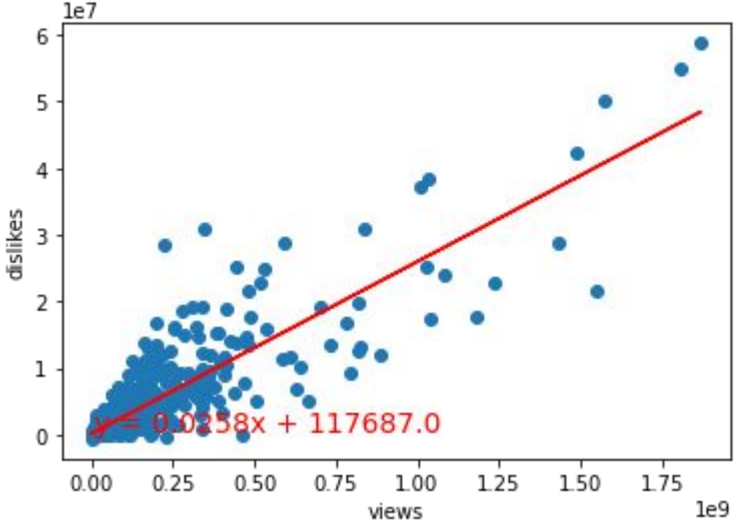
Views vs likes-Dislikes by Channel: USA

```
diff = us_channel_df['likes'] - us_channel_df['dislikes']
views = us_channel_df['views']
plt.scatter(views, diff, marker="x")
# plt.ylim(35,46)
plt.xlabel("views")
plt.ylabel("likes-dislikes")
plt.show()
```



Removing Outliers & Conclusion: USA

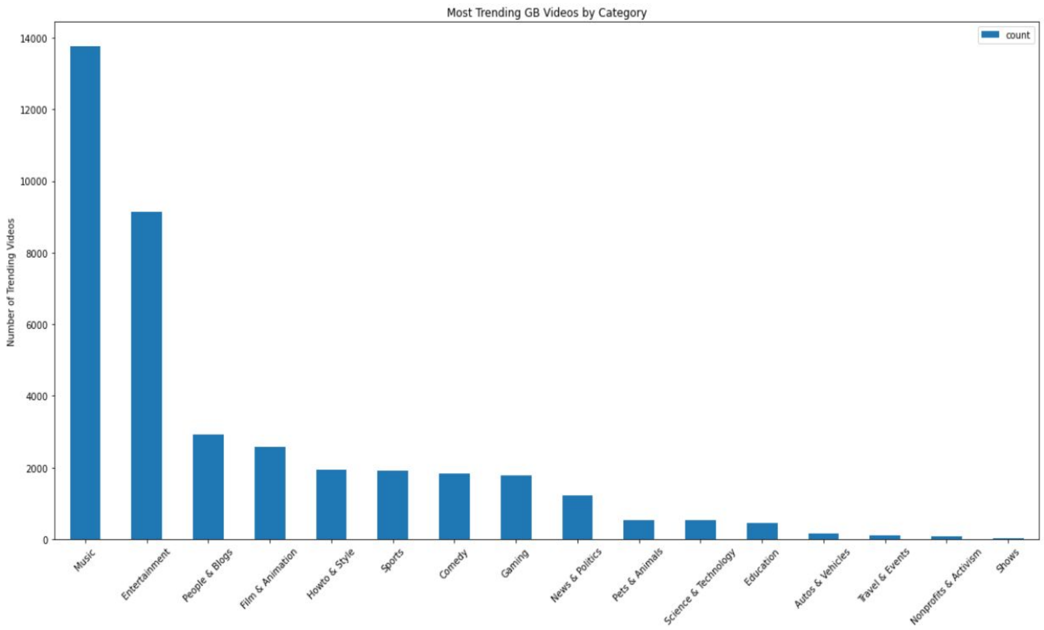
channel_title	likes	dislikes	views	likes-dislikes
ibighit	199247121	3467306	2235906679	195779815
ChildishGambinoVEVO	96700818	6054434	3758488765	90646384
Dude Perfect	60275557	1501477	1870085178	58774080
Marvel Entertainment	55873344	1031250	1808998971	54842094
ArianaGrandeVevo	52170970	1931230	1576959172	50239740



Outliers: https://www.youtube.com/results?search_query=ibighit

Initial Conclusion: There is a strong positive correlation between likes and views by Channels.

Most Popular Categories: GB



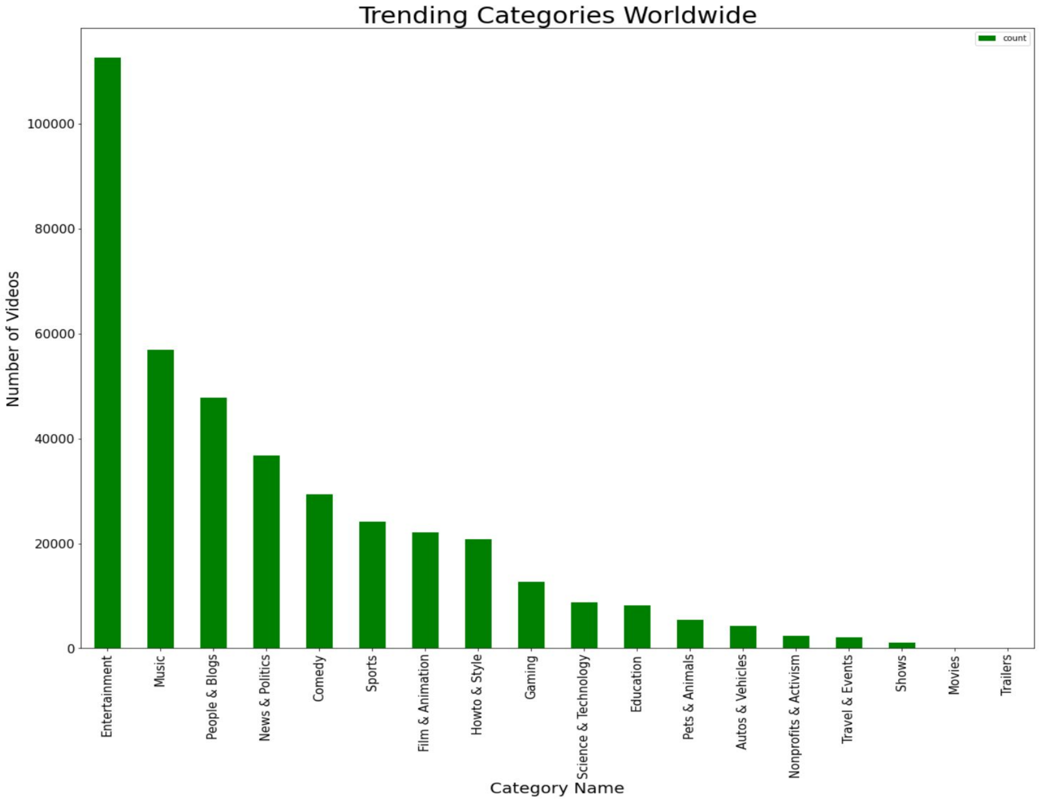
#Plotting chart with a created dataframe
GB_cat_df_plot = GB_cat_count[['category', 'count']].plot(kind='bar',ylabel='Number of Trending Videos', figsize=(20,10), title=('Most Trending GB Videos by Category'))

#labeling x axis
GB_cat_df_plot.set_xticklabels(GB_cat_count['category'],rotation=45)

# showing the chart + adjusting layout to tight layout
plt.show()
plt.tight_layout()

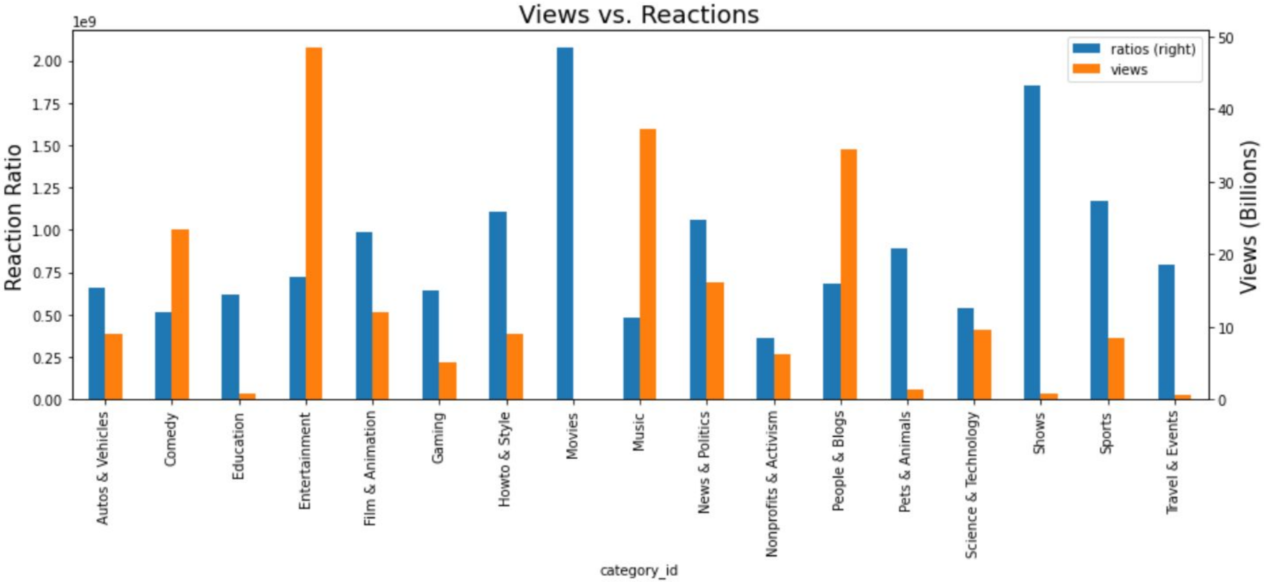
World's Most Trending Categories

```
ww_cat = gb_count_df.add(kr_count_df,
fill_value=0).add(jp_count_df,
fill_value=0).add…

ww_cat_sorted =
ww_cat.sort_values(by=["count"],
ascending=False)

ww_cat_sorted.plot(kind='bar',
figsize=(20,15), color='green')
```



Views vs. Reactions per Category: Russia

Different communities engage in videos differently



reactions_df_final.plot.bar(figsize =(15,5), secondary_y= 'ratios')

#Get and set new axes
ax1, ax2 = plt.gcf().get_axes()

#Title both axes
ax1.set_ylabel('Reaction Ratio')

ax2.set_ylabel('Views (Billions)')

Trends with likes and Dislikes: France

#Grouping Trendings, Likes and Dislikes with Bar-graph By Categories(FR)

Hypothesis

- ❑ Category with higher trending has higher likes and dislikes.
- ❑ Likes and Dislikes has parallel correlation.

Null Hypothesis

- ❑ There is no correlation between category with higher trending and amount of likes and dislikes.
- ❑ Likes and Dislikes has No/ inverse correlation.

Trends with likes and Dislikes: France

FR				
category_id	Trending	Likes	Dislikes	Comments
Entertainment	9819	1.18E+08	10351578	15729924
People & Blogs	5719	28927705	1760711	4104818
Comedy	4343	1.31E+08	3391288	9136814
Sports	4342	43964560	2145956	4575418
Music	3946	2.77E+08	9772318	25446289
News & Politics	3752	9301486	775868	1896101
Howto & Style	2361	15519633	643543	1759358
Film & Animation	2157	24631422	1092744	2841655
Gaming	1459	22502704	1097458	3047593
Science & Technology	802	18513625	511858	2926363
Education	769	8302644	201359	768074
Autos & Vehicles	673	1606767	52260	207973
Pets & Animals	237	1335449	44591	187590
Travel & Events	119	871774	10980	101892
Nonprofits & Activism	114	5987384	1231113	1848593
Shows	99	291212	103846	44882
Movies	11	24295	1048	1467
Trailers	2	192	9	0

Code

```
views_likes_dislikes = pd.DataFrame({'Trending': trending_count,
    'Likes': likes_count,
    'Dislikes': dislikes_count,
    'Comments': comment_count})
views_likes_dislikes
vlk_sorted =
views_likes_dislikes.sort_values(by=["Trending"], ascending=False)
vlk_sorted
```

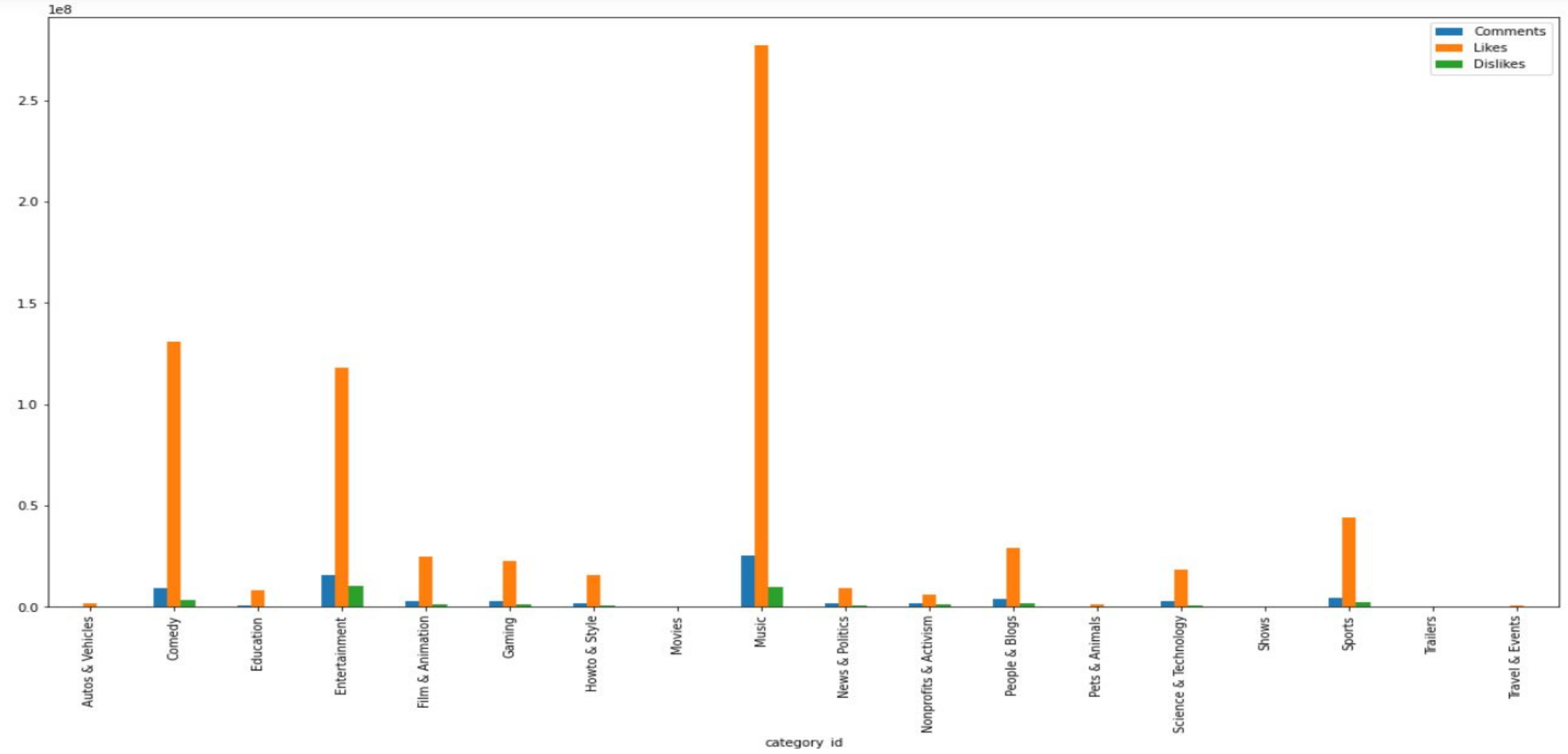
Trends with likes and Dislikes: France

#Grouping Trending, Like and Dislike with Bar-graph By Category(FR)

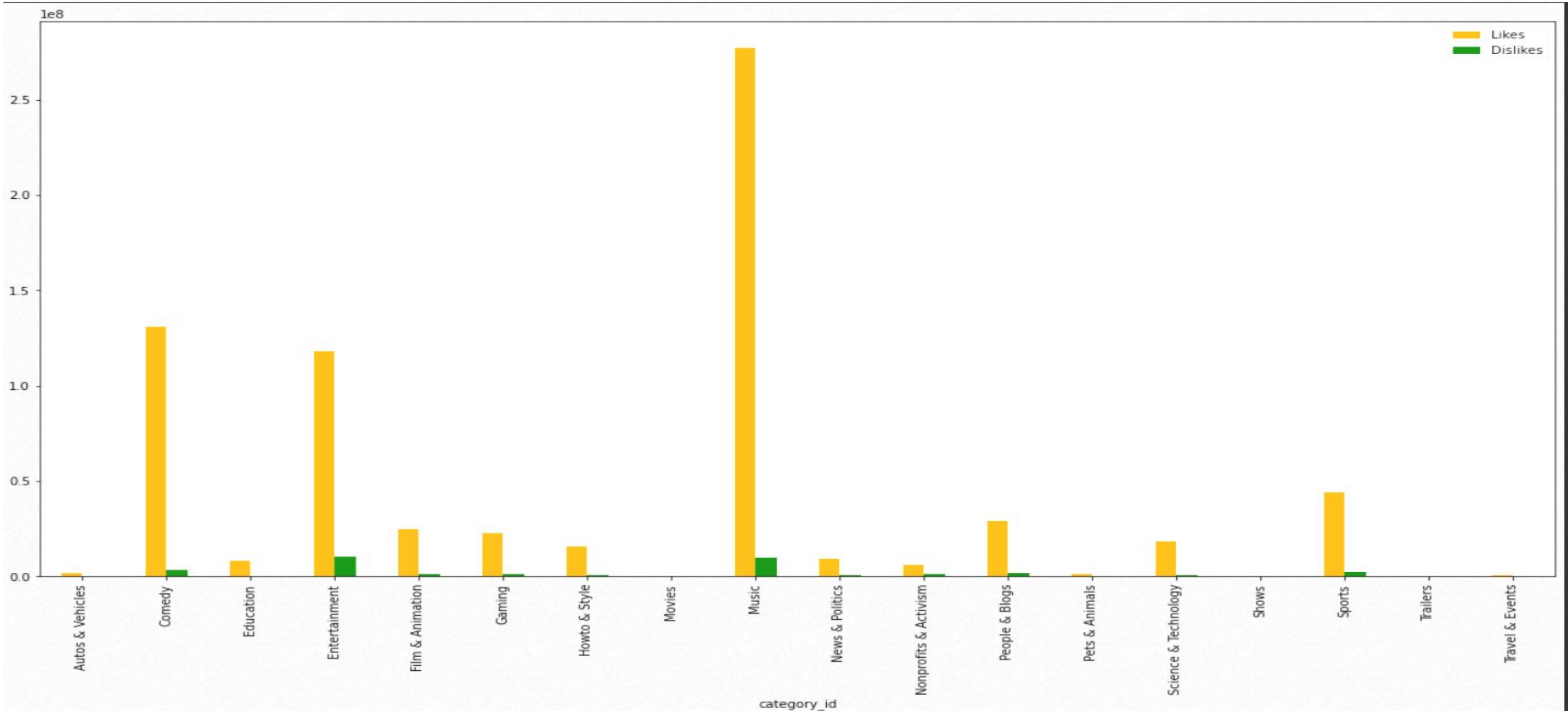
Code

```
#grouping
groupAll_df= views_likes_dislikes[['Comments', 'Likes', 'Dislikes']].groupby('category_id').sum()
groupAll_df.plot(subplots=False,kind= "bar", figsize =(20,10))
plt.show()
group_df= views_likes_dislikes[['Likes', 'Dislikes']].groupby('category_id').sum()
group_df.plot(subplots=False,kind= "bar", figsize =(20,10), color =['orange', 'green'])
plt.show()
```

Trends vs. Likes and Dislikes: France

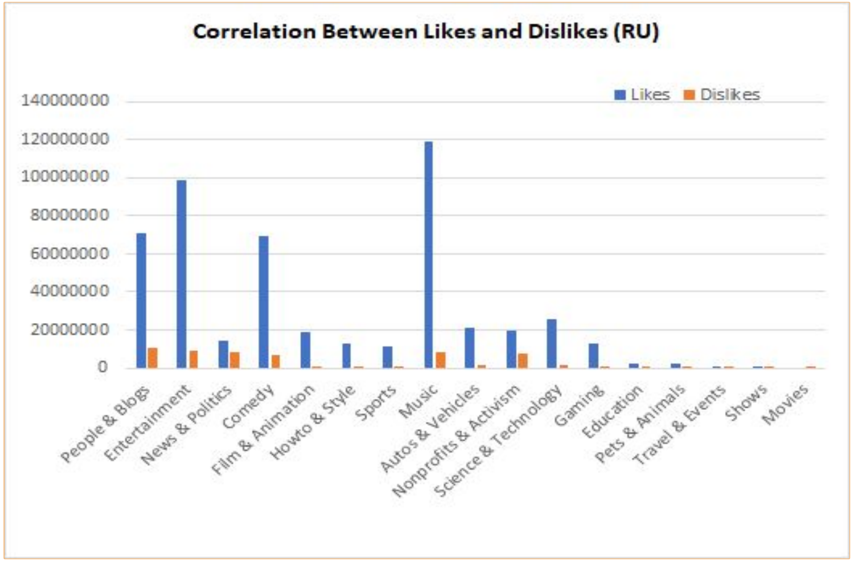


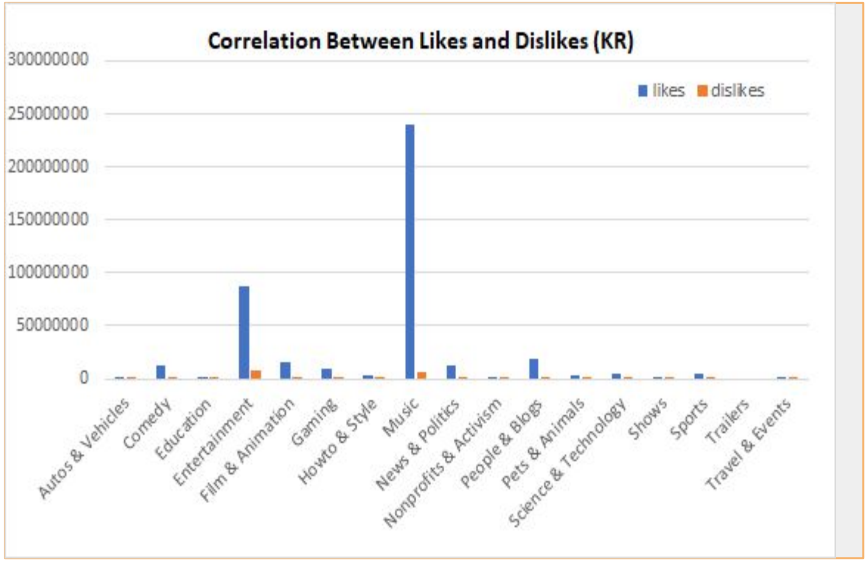
Correlations Between Likes And Dislikes: France



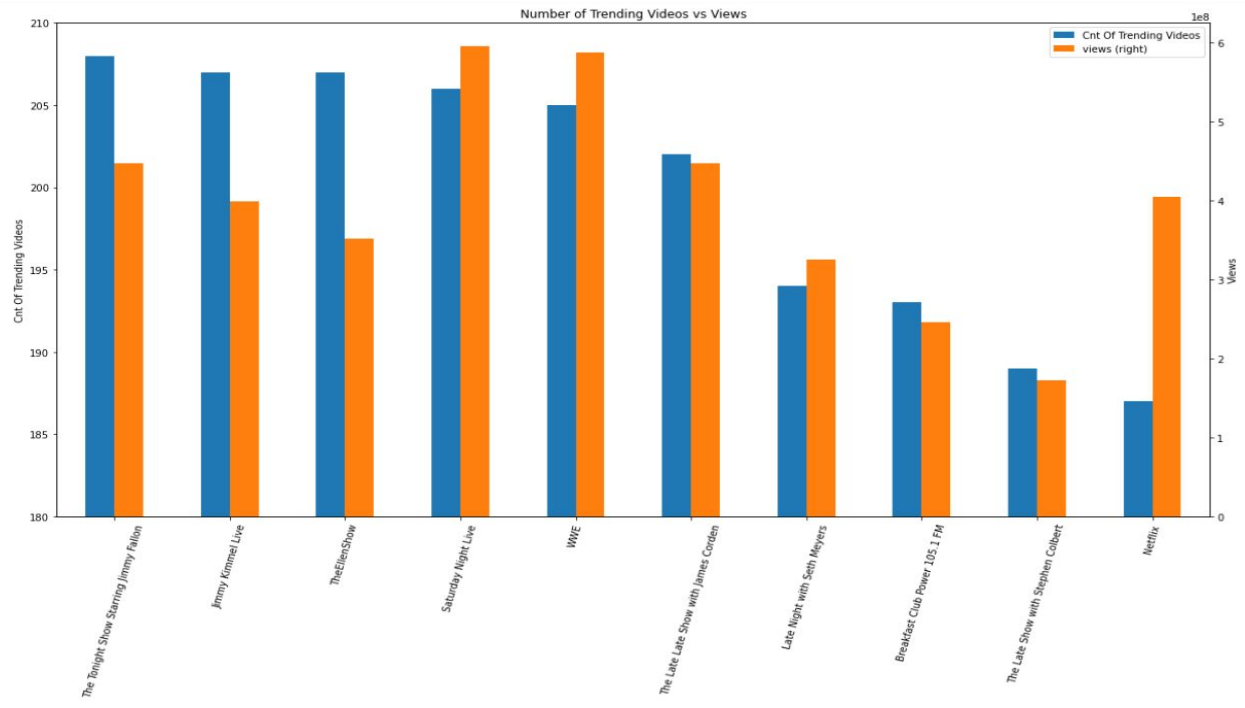
Trends with likes and Dislikes

#Grouping Trendings, Likes and Dislikes with Bar-graph By Categories(For Other Countries)





Top 10 Youtube Channels: GB



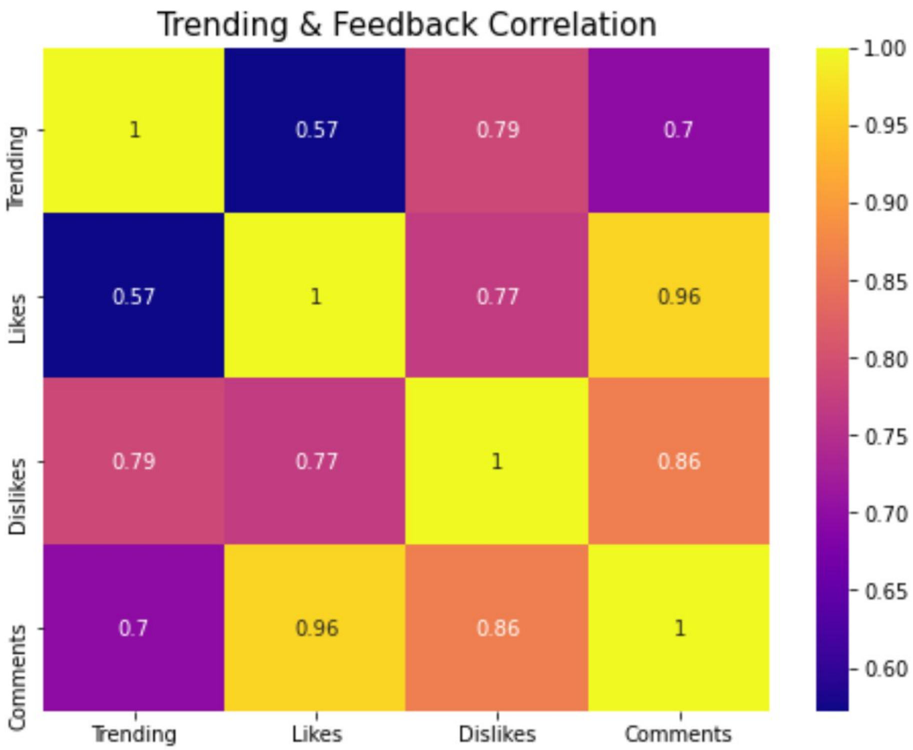
top10_chann_plot =

top10_chann.plot(kind ='bar', secondary_y='views',

ylabel='Cnt Of Trending Videos', figsize=(20,10),

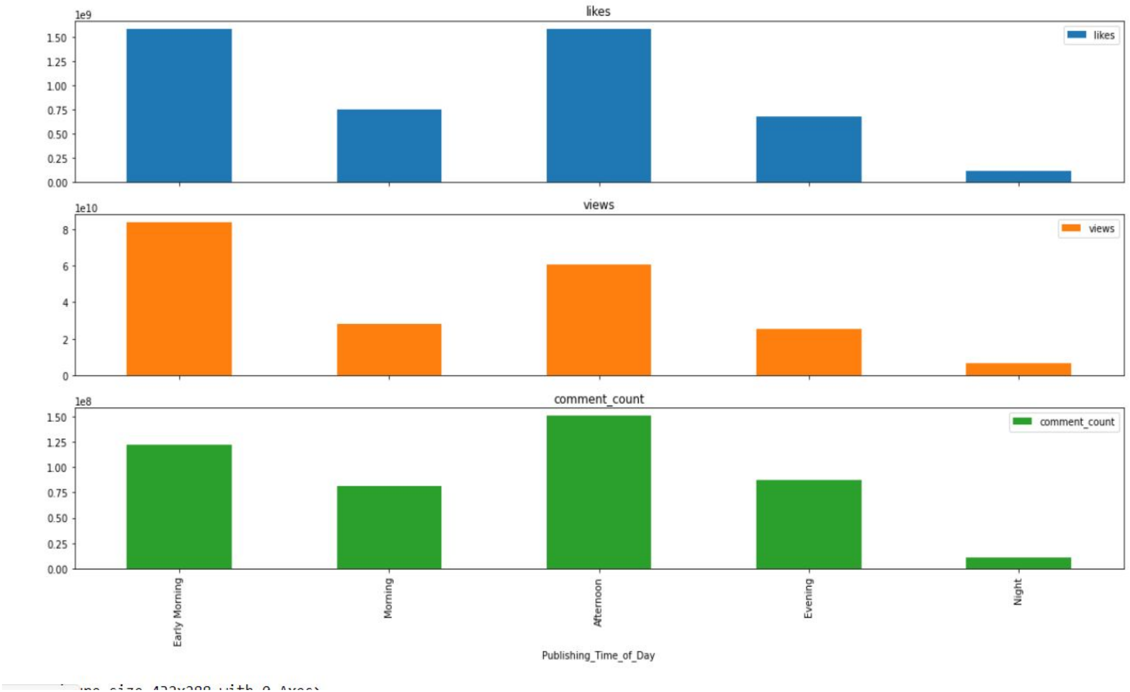
title =('Number of Trending Videos vs Views'))

What Makes a Video Trend?: Russia



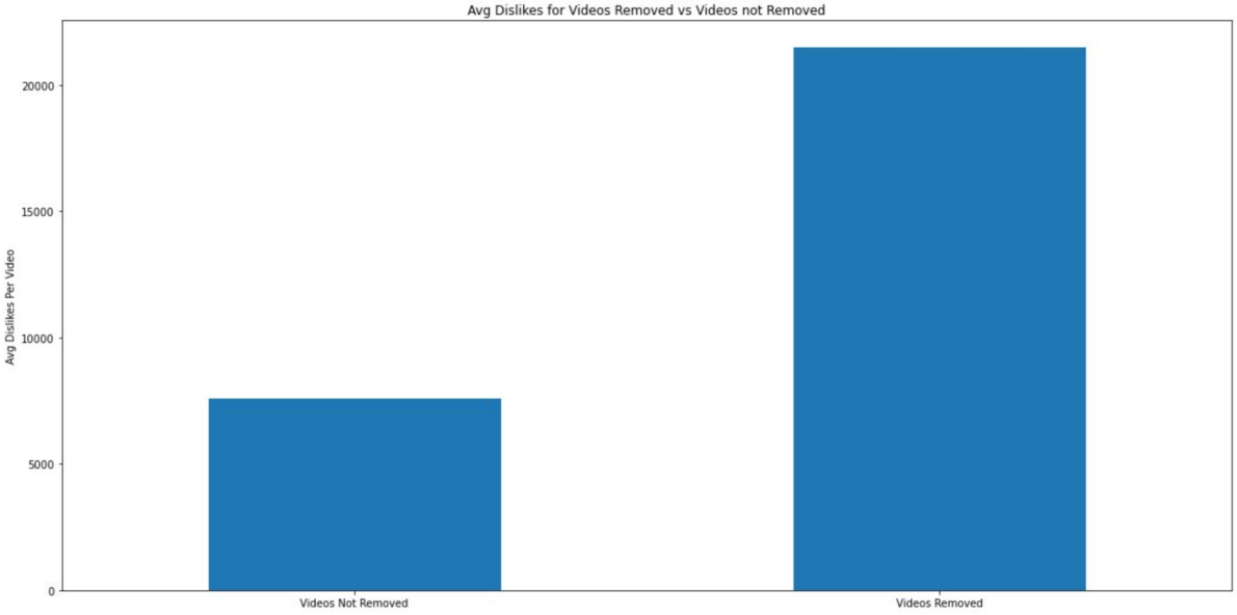

```
import seaborn as sns

corrmat = vlk_sorted.corr()

plt.figure(figsize=(8,6))

plt.title('Trending & Feedback Correlation', size=15)

sns.heatmap(vlk_sorted.corr(),annot=True,cmap='plasma')

corrmat
```

Best Time of Day to Post Videos: GB




```
bins = [0, 6, 11, 16, 19, 20]
group_names = ['Early Morning',
'Morning', 'Afternoon', 'Evening',
'Night']

# using cut function to bin pub hr
into groups
GB_data['Publishing_Time_of_Day']
=
pd.cut(GB_data['publish_hr(24hrs)'],
bins,

labels=group_names,
include_lowest=True)
```

Avg Dislikes for Videos Removed: GB

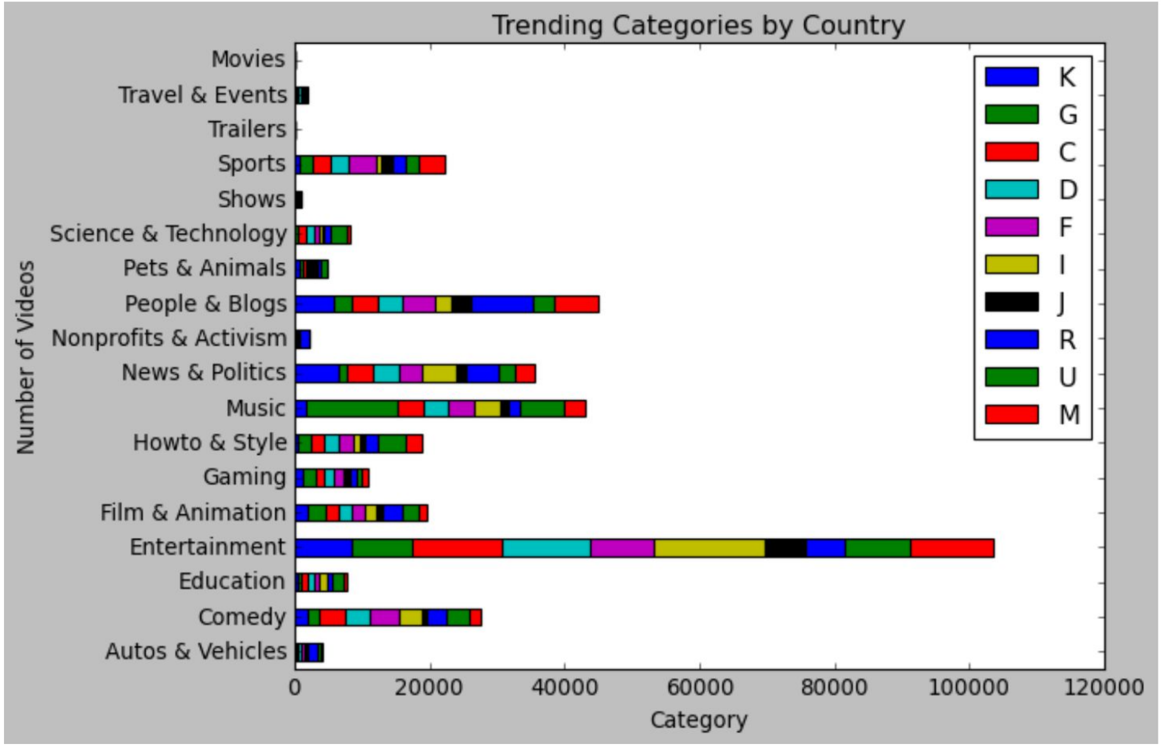


rm_df =
GB_data[['video_error_or_re
moved'

,'dislikes']].groupby('video_er
ror_or_removed').agg(['sum','
count'])

#adding column showing the
avg number of dislikes per
video
rm_df['Avg Dislikes Per
Video'] = round(rm_df['Total
Dislikes']/rm_df['Cnt of
Videos Removed'],2)

Source of Trending Categories

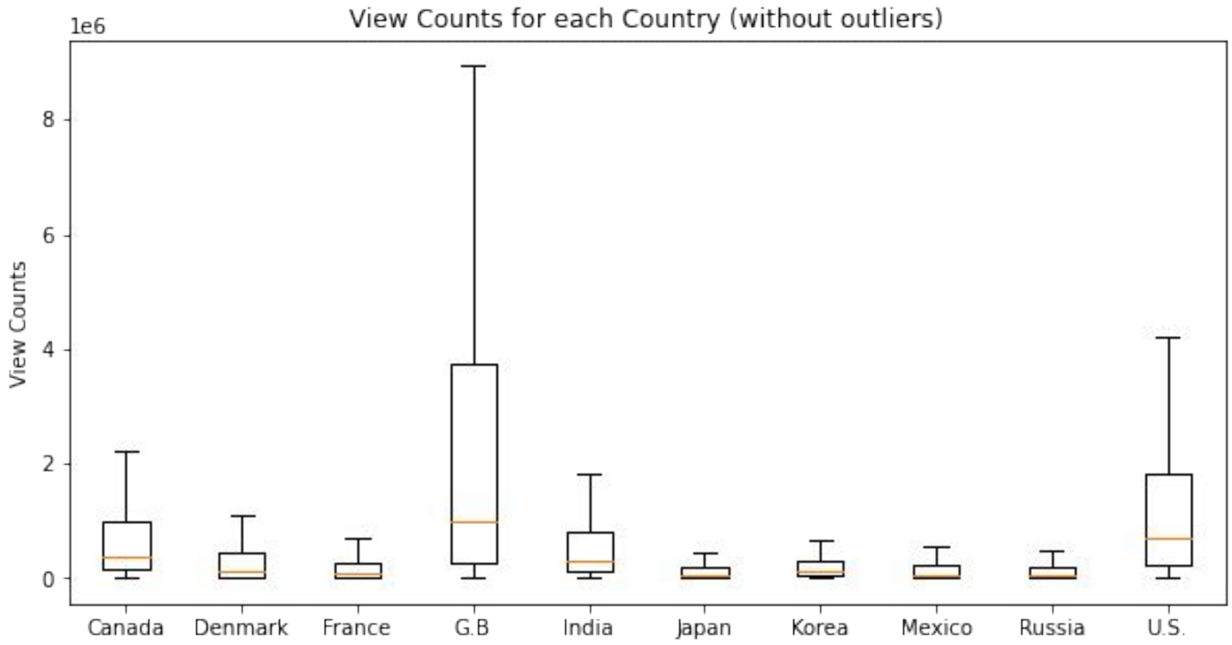



```
xxx_df = pd.concat({
    'KR': kr_count_df,
    'GB': gb_count_df...

xxx_df.fillna(0)

big_plot = xxx_df.plot(kind='barh',
stacked = True)

plt.style.use('classic')
```

Comparing View Counts Per Video of Each Country

What country has a video with the highest viewer?

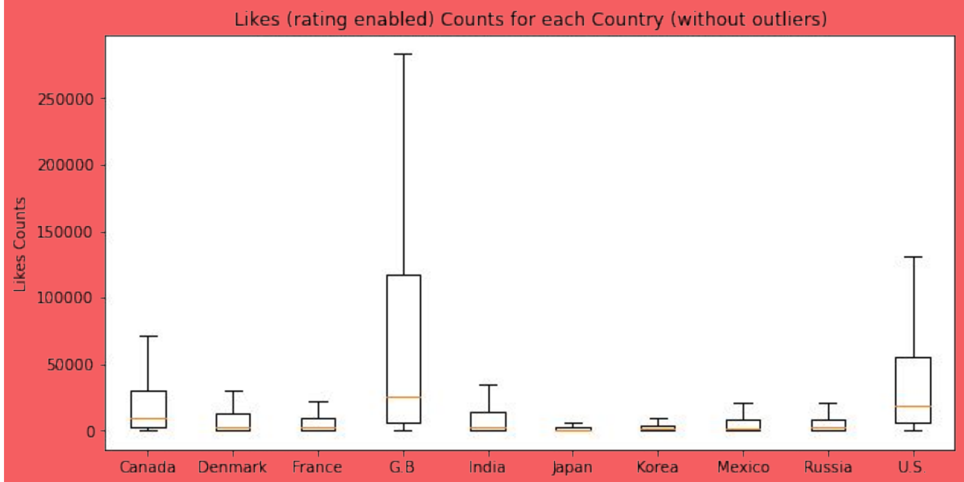
What does this tell us about the country?

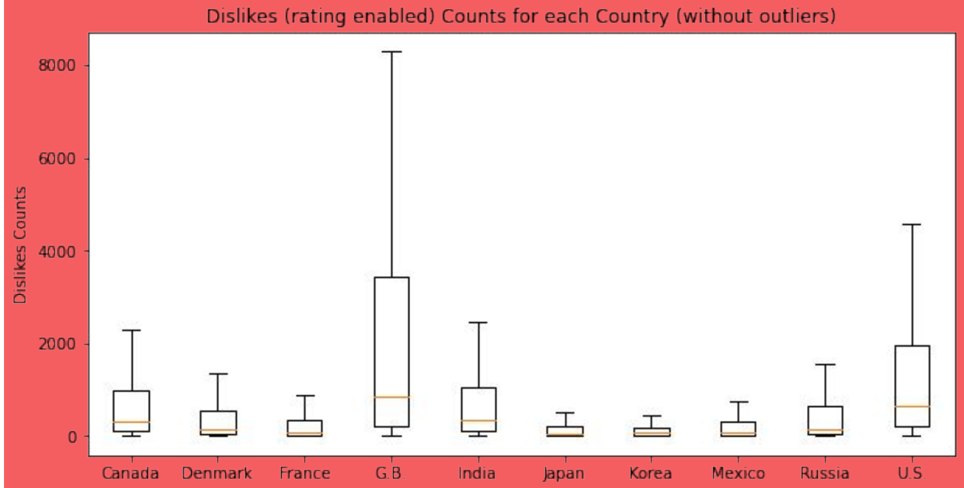


Looking at Likes and Dislikes

Notice the similarity!

```
plt.figure(figsize=(10,5))
plt.boxplot(views_ca, positions=[1],widths=0.4,showfliers=False)
.
.
.
plt.xticks([1,2,3,4,5,6,7,8,9,10], ["Canada", "Denmark", "France",
          "G.B", "India", "Japan", "Korea", "Mexico",
          "Russia", "U.S."])
plt.ylabel("View Counts")
plt.title("View Counts for each Country (without outliers)")
plt.show()
```

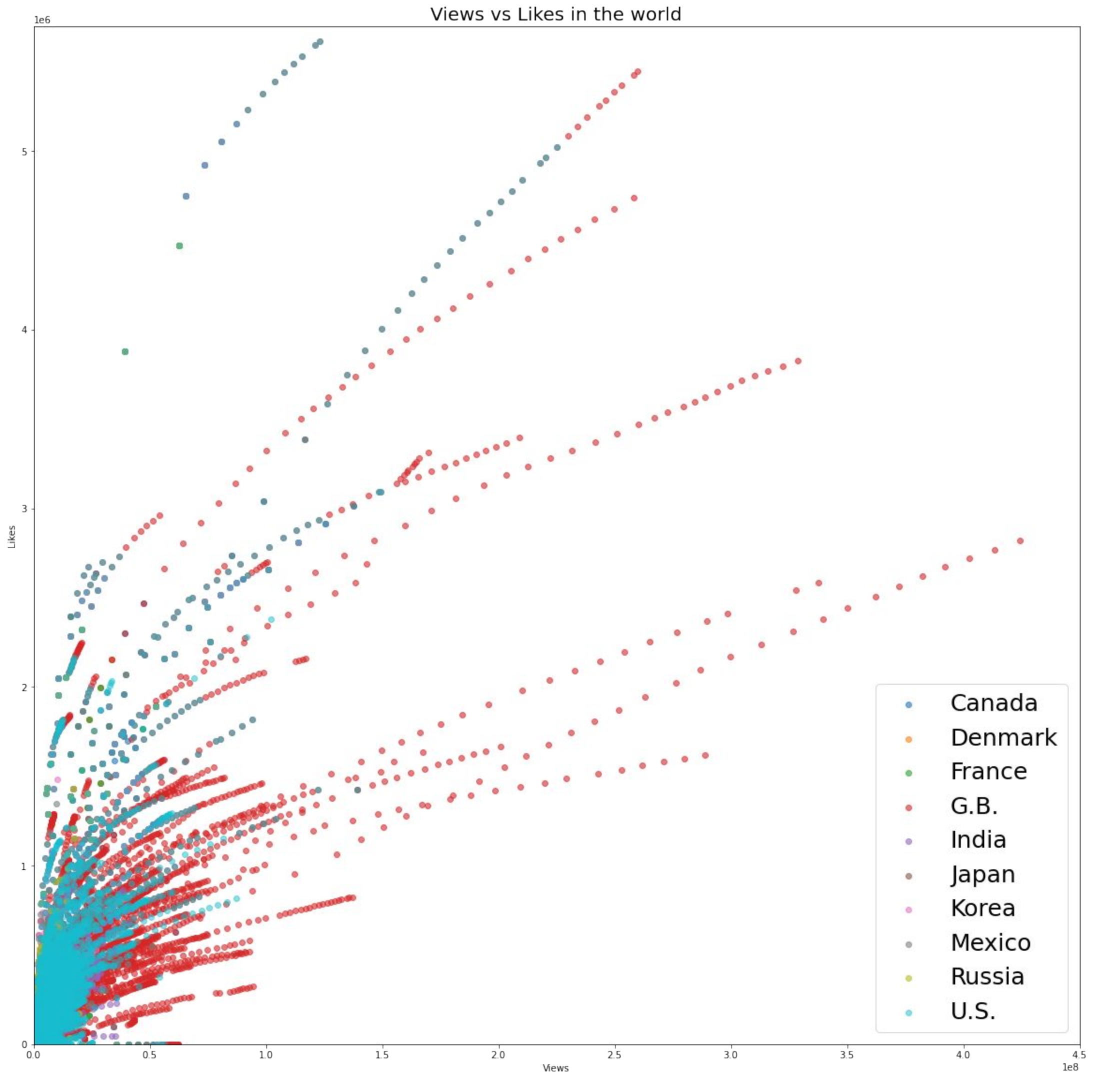


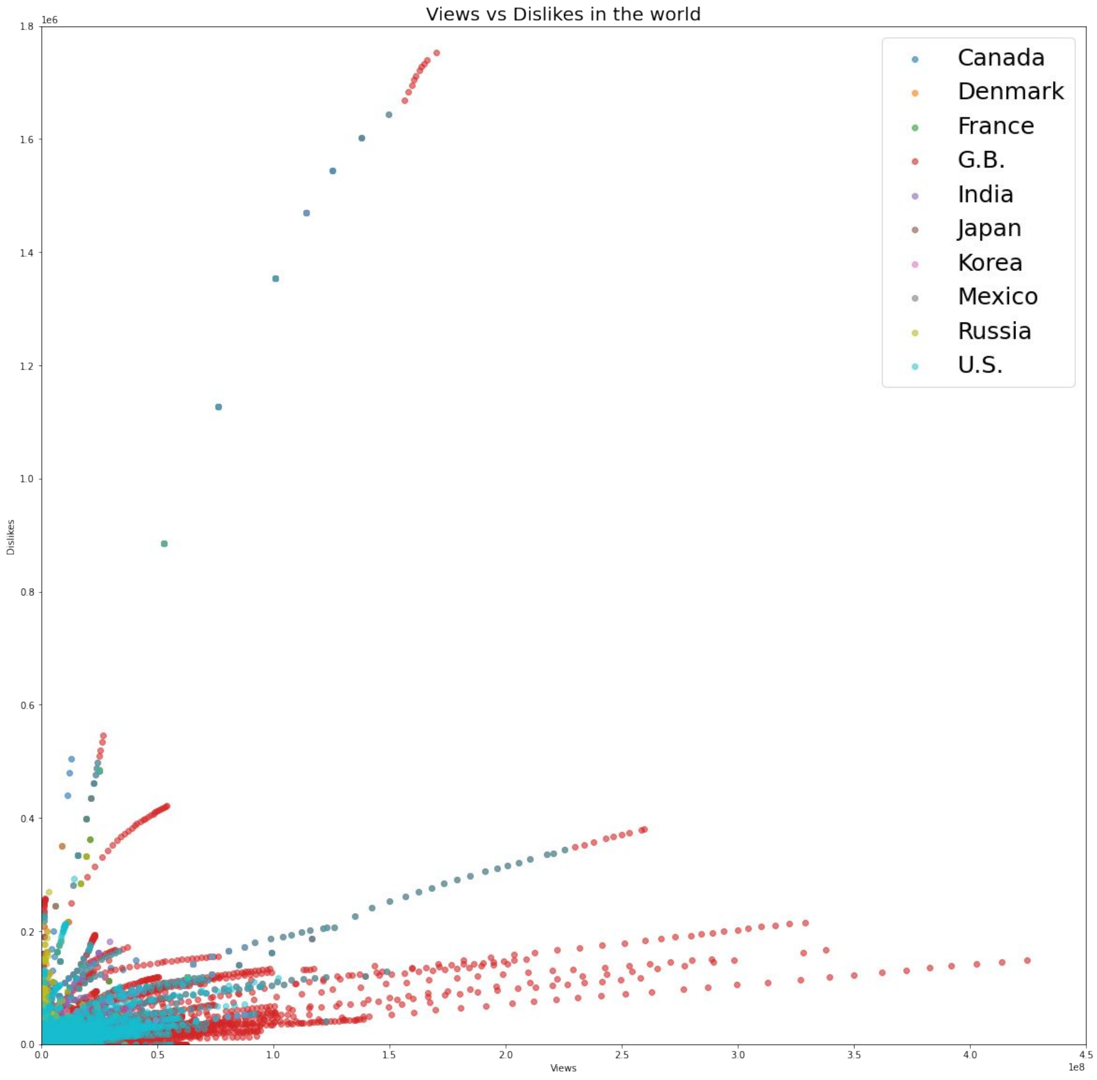


Views vs Likes of Each Country

What does this graph tell us about outliers of different country data?

```
# colors:
https://matplotlib.org/gallery/color/named_colors.html
plt.figure(figsize=(20,20))
plt.scatter(views_ca, likes_ca, alpha=0.6, color="tab:blue",
            label="Canada")
.
.
.
plt.legend(loc="lower right", prop={'size': 25})
plt.xlim(0, 4.5e8)
plt.ylim(0, 5.7e6)
plt.xlabel("Views")
plt.ylabel("Likes")
plt.title("Views vs Likes in the world", fontsize=20)
plt.show()
```

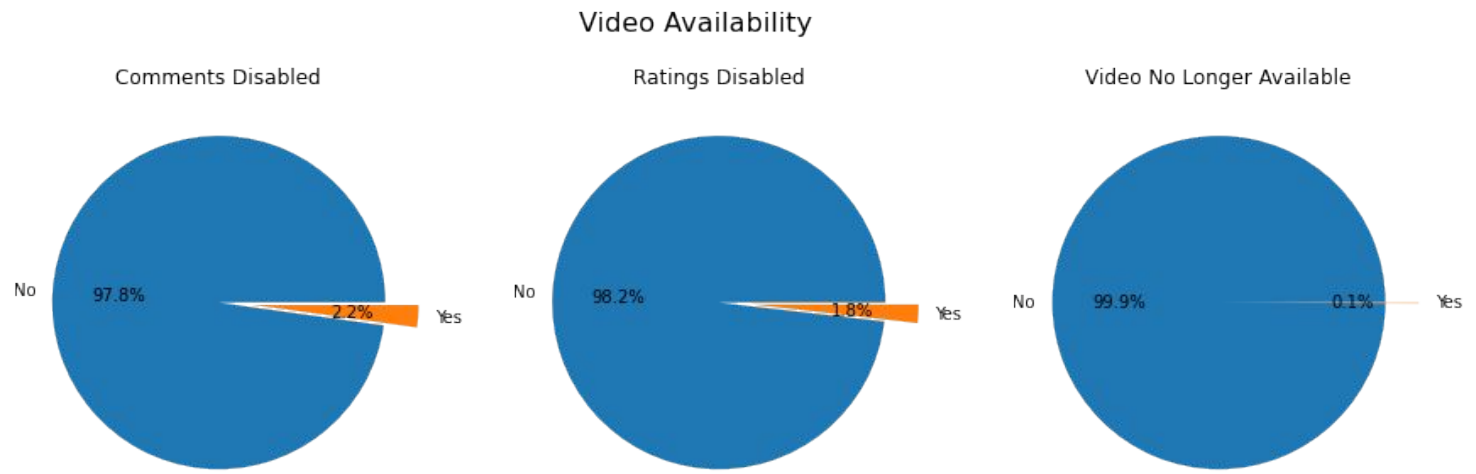




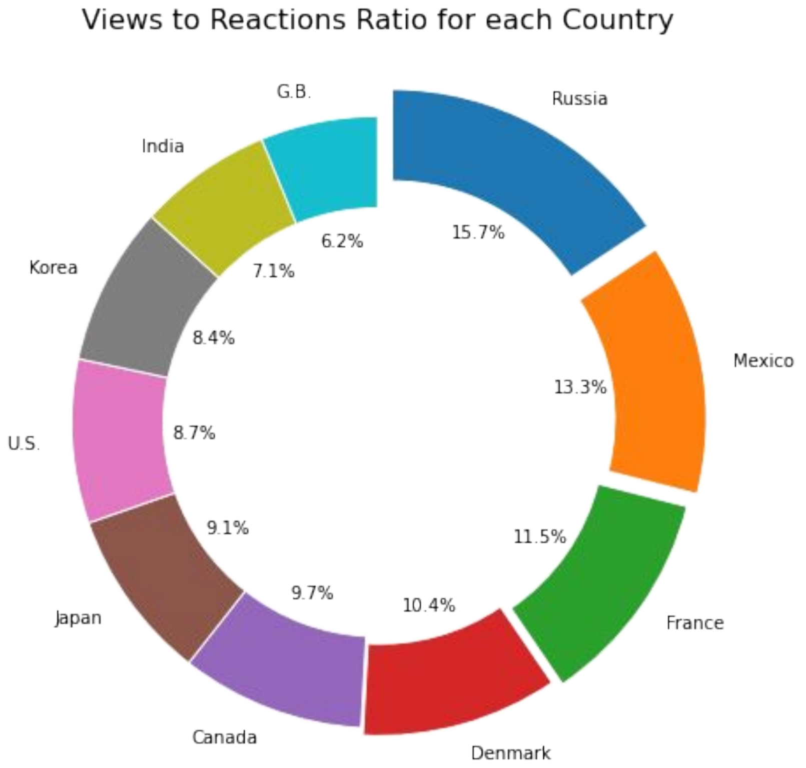
Views vs Dislikes of Each Country

What does this graph tell us about outliers of different country data?

Comments, Ratings, Video Availability!




```
fig, (ax1, ax2, ax3) = plt.subplots(1, 3, figsize=(15,5))
ax1.pie([sum(comments_t), sum(comments_f)], labels=["No", "Yes"], autopct="%.1f%%", explode=[0,0.2])
ax2.pie([sum(ratings_t), sum(ratings_f)], labels=["No", "Yes"], autopct="%.1f%%", explode=[0,0.2])
ax3.pie([sum(no_video_t), sum(no_video_f)], labels=["No", "Yes"], autopct="%.1f%%", explode=[0,0.2])
ax1.set_title("Comments Disabled")
ax2.set_title("Ratings Disabled")
ax3.set_title("Video No Longer Available")
fig.suptitle("Video Availability", fontsize=16)
plt.show()
```




```
viewreact = viewreact.sort_values(by=["V-R-Ratio"], ascending=False)
explode = [0.2, 0.15, 0.1, 0.05, 0, 0, 0, 0, 0, 0]
plt.pie(viewreact["V-R-Ratio"], labels=viewreact["Country"],
        autopct="%.1f%%", startangle=90, counterclock=False,
        explode=explode, radius=2, wedgeprops=dict(width=0.6,
        edgecolor="w"))
plt.title("Views to Reactions Ratio for each Country", y=1.5, fontsize=16)
plt.show()
```

Reaction: Sum of likes, dislikes, and comments.

Do all the country have similar amount of reaction rates compared to the views?

Q & A

Thank You!

Sources

DF's, JSON's: https://www.kaggle.com/datasnaek/youtube-new

GitHub: https://github.com/samuelokunola326/Group6_Project