

# youtube

June 16, 2023

## 1 New's channel analysis

```
[ ]: # !pip install --upgrade google-api-python-client
from googleapiclient.discovery import build
from dateutil import parser
import pandas as pd
from IPython.display import JSON

# Data viz packages
import seaborn as sns
import matplotlib.pyplot as plt
import matplotlib.ticker as ticker

# NLP
import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
nltk.download('stopwords')
nltk.download('punkt')
from wordcloud import WordCloud

[ ]: api_key = 'AIzaSyDlkH3DJakiTiYYQVh2AiXcdREhgCC9cus'
channel_id = ['UCNye-wNBqNL5ZzHSJj3l8Bg', #al jajera
              'UC_gUM8rL-Lrg6O3adPW9K1g', #wion
              'UCIvaYmXn9l0QMdemBG3v1pQ', #zee news
              'UC16niRr50-MSBwi03YDb3RA', #bbc
              'UC7wXt18f2iA3EDXeqAVuKng', #republicBharat
              'UChqUTb7kYRX8-EiaN3XFrSQ', #Reuters
              'UCgrNz-aDmcr2uuto8_DL2jg' ] #China Global Television Network
youtube = build('youtube', 'v3', developerKey=api_key)
```

### 1.1 Function to get channel Stats:

```
[ ]: def get_channel_stats(youtube, channel_id):
    all_data=[]
    request = youtube.channels().
    ↪list(part="snippet,contentDetails,statistics",id=', '.join(channel_id))
```

```
# request = youtube.channels().
↳ list(part="snippet,contentDetails,statistics",)
response = request.execute()

for i in range(len(response['items'])):
    data = dict(Channel_name = response['items'][i]['snippet']['title'] ,
                PublishedAt = response['items'][i]['snippet']['publishedAt']_
↳ ,
                Country = response['items'][i]['snippet']['country'],
                Views = response['items'][i]['statistics']['viewCount'],
                Subscriber =_
↳ response['items'][i]['statistics']['subscriberCount'],
                Total_video =_
↳ response['items'][i]['statistics']['videoCount'],
                Playlist_id =_
↳ response['items'][i]['contentDetails']['relatedPlaylists']['uploads'],
                url =_
↳ response['items'][i]['snippet']['thumbnails']['default']['url'])
    all_data.append(data)

return all_data
```

```
[ ]: channel_stat =get_channel_stats(youtube,channel_id)
```

```
[ ]: df = pd.DataFrame(channel_stat)
```

```
[63]: df.head()
```

```
[63]:
```

	Channel_name	PublishedAt	Country	Views	\
0	CGTN	2013-01-24 13:31:24+00:00	CN	2023565678	
1	Al Jazeera English	2006-11-23 14:12:43+00:00	QA	3301156060	
2	WION	2016-05-19 12:55:08+00:00	IN	3953857079	
3	Zee News	2007-06-19 12:05:49+00:00	IN	16486366334	
4	BBC News	2006-04-08 05:51:05+00:00	GB	4506232124	

	Subscriber	Total_video	Playlist_id	\
0	3020000	117368	UUgrNz-aDmcr2uuto8_DL2jg	
1	10400000	104849	UUNye-wNBqNL5ZzHSJj3l8Bg	
2	7540000	104085	UU_gUM8rL-Lrg603adPW9K1g	
3	30300000	176803	UUIvaYmXn910QMdemBG3v1pQ	
4	14300000	18673	UU16niRr50-MSBwi03YDb3RA	

	url
0	https://yt3.ggpht.com/5a5D_Wc4hjQ_u5F3rOWWaaA9...
1	https://yt3.ggpht.com/YKrh5sIYpesEIHbjfgqWfjFx...

```

2 https://yt3.ggpht.com/yt3/AGIKgqNWtVPoF30RtRIf...
3 https://yt3.ggpht.com/vZFILuFswtjUIU5oRCc79rxR...
4 https://yt3.ggpht.com/y_esGAQ0hX4rTpWvrALerAJl...

```

```
[64]: df.dtypes
```

```

[64]: Channel_name      object
      PublishedAt      datetime64[ns, UTC]
      Country          object
      Views            int64
      Subscriber       int64
      Total_video      int64
      Playlist_id      object
      url              object
      dtype: object

```

## 1.2 Preprocessing

```

[ ]: df['Views'] = pd.to_numeric(df['Views'])
     df['Subscriber'] = pd.to_numeric(df['Subscriber'])
     df['Total_video'] = pd.to_numeric(df['Total_video'])

```

```
[ ]: df['PublishedAt'] = pd.to_datetime(df.PublishedAt)
```

```
[65]: df.dtypes
```

```

[65]: Channel_name      object
      PublishedAt      datetime64[ns, UTC]
      Country          object
      Views            int64
      Subscriber       int64
      Total_video      int64
      Playlist_id      object
      url              object
      dtype: object

```

```
[66]: df.isnull().count()
```

```

[66]: Channel_name      7
      PublishedAt      7
      Country          7
      Views            7
      Subscriber       7
      Total_video      7
      Playlist_id      7
      url              7
      dtype: int64

```

```
[ ]: df.describe
```

```
[68]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Channel_name    7 non-null     object
1   PublishedAt     7 non-null     datetime64[ns, UTC]
2   Country         7 non-null     object
3   Views          7 non-null     int64
4   Subscriber      7 non-null     int64
5   Total_video     7 non-null     int64
6   Playlist_id     7 non-null     object
7   url            7 non-null     object
dtypes: datetime64[ns, UTC](1), int64(3), object(4)
memory usage: 576.0+ bytes
```

```
[69]: df.shape
```

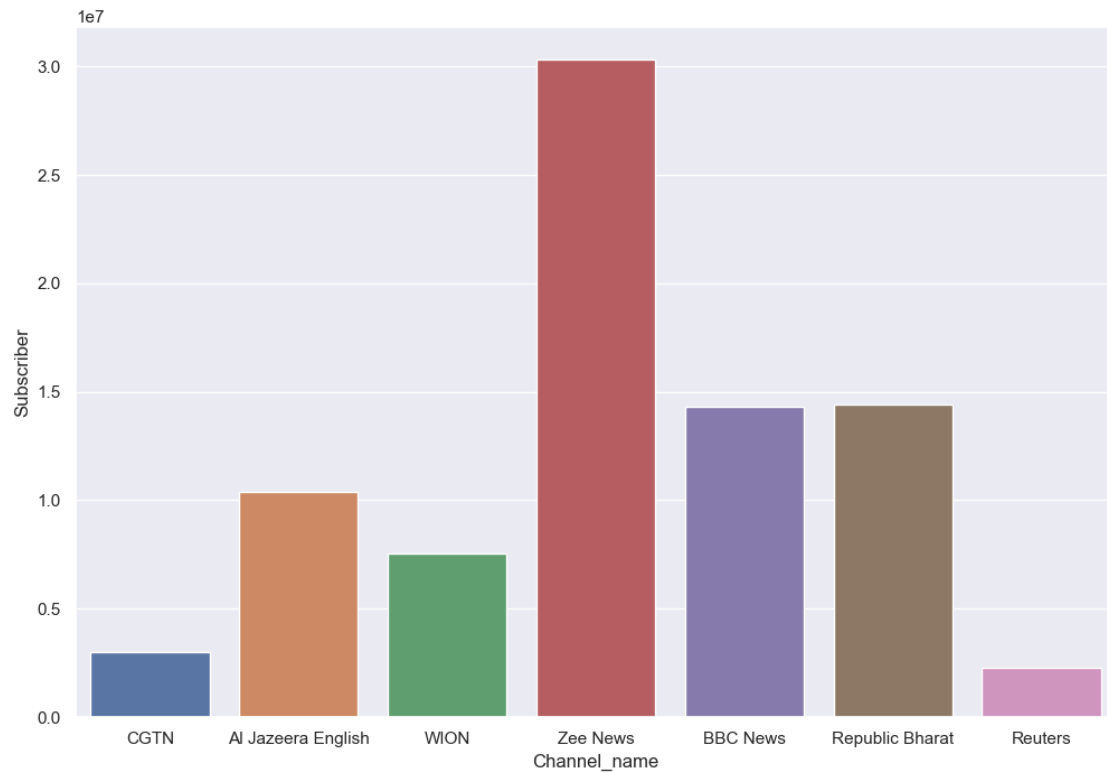
```
[69]: (7, 8)
```

## 1.3 EDA

### 1.3.1 Graphical approach

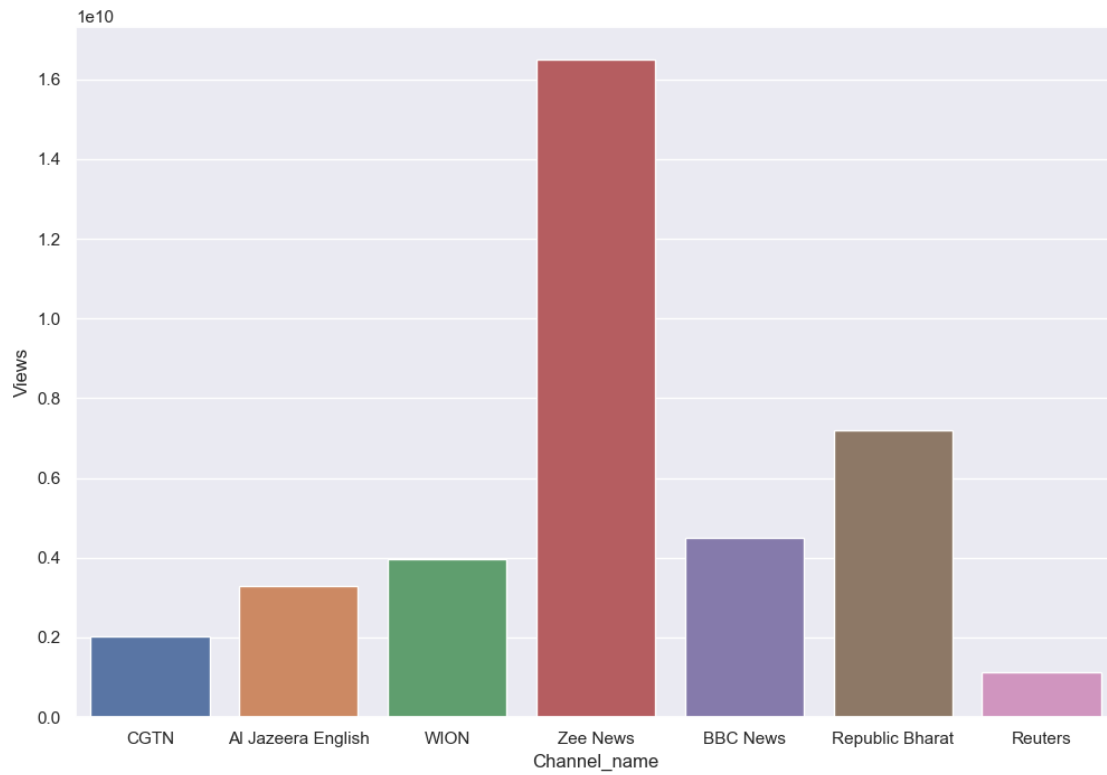
#### 1.3.2 Total number of Subscriber in million by Channel name

```
[70]: sns.set(rc={'figure.figsize':(12,8)})
ax = sns.barplot(x='Channel_name' , y = 'Subscriber' , data=df)
```



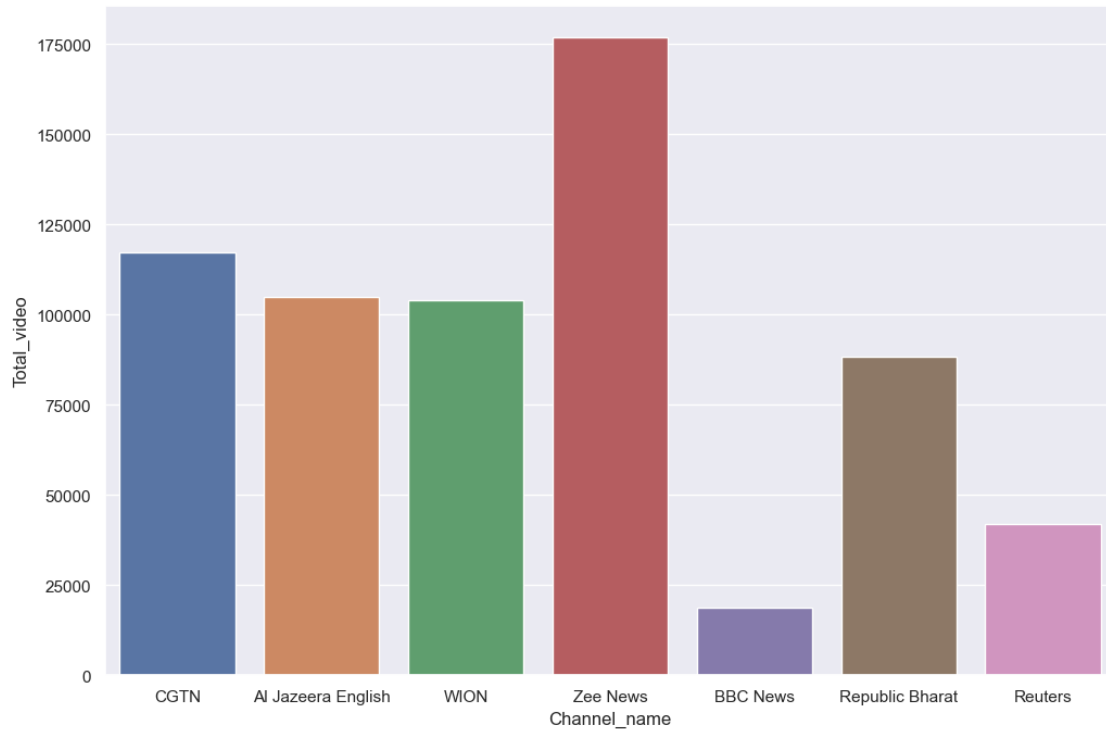
### 1.3.3 Number of views by channel

```
[71]: sns.set(rc={'figure.figsize':(12,8)})  
ax = sns.barplot(x='Channel_name' , y = 'Views' , data=df)
```



### 1.3.4 Total No of video by channel

```
[72]: sns.set(rc={'figure.figsize':(12,8)})  
ax = sns.barplot(x='Channel_name' , y = 'Total_video' , data=df)
```



#### 1.4 Funtion to get Video Id

```
[ ]: playlist_id = df.loc[df.Channel_name == 'Al Jazeera English' , 'Playlist_id'].
    ↪iloc[0]
```

```
[73]: playlist_id
```

```
[73]: 'UU16niRr50-MSBwi03YDb3RA'
```

```
[ ]: def get_video_id(youtube,playlist_id):

    request = youtube.playlistItems().list(part = '
    ↪'contentDetails',playlistId=playlist_id , maxResults=50)
    response = request.execute()

    video_id = []
    for i in range(len(response['items'])):
        video_id.append(response['items'][i]['contentDetails']['videoId'])

    next_page_token = response.get('nextPageToken')
    more_pages =True

    cnt = 0
```

```

while more_pages:
    if cnt==39:
        more_pages=False
    else:
        request = youtube.playlistItems().list(part =
↪'contentDetails',playlistId=playlist_id , maxResults=50,
                                                pageToken = next_page_token)

        response = request.execute()
        cnt+=1

        for i in range(len(response['items'])):
            video_id.
↪append(response['items'][i]['contentDetails']['videoId'])

            next_page_token = response.get('nextPageToken')

return video_id

```

```
[ ]: video_ids=get_video_id(youtube,playlist_id)
```

```
[ ]: video_ids
```

## 1.5 Function to get Video details

```

[ ]: def get_video_details(youtube,video_ids):
    all_video_stat = []
    for i in range(0,5000,50):

        request = youtube.videos().list( part = 'snippet,statistics',
                                            id = ','.join(video_ids[i:i+50]))

        response = request.execute()
        for video in response['items']:
            video_stats = dict(Title = video['snippet']['title'],
                                publish_date = video['snippet']['publishedAt'] ,
                                View = video['statistics']['viewCount'] ,
                                like_count = video['statistics']['likeCount'] ,
                                Comments = video['statistics']['commentCount'] )
            all_video_stat.append(video_stats)

    return all_video_stat

```

```
[ ]: video_details=get_video_details(youtube , video_ids)
```

```
[ ]: video_df = pd.DataFrame(video_details)
```



```
[75]: video_df
```

```
[75]:
```

	Title	publish_date	View	\
0	Will Germany's security strategy work?   Insid...	2023-06-16	10	
1	What makes a cult?	2023-06-16	2107	
2	Moment Armenians shoot at Azerbaijani guards o...	2023-06-16	3748	
3	Headlines - African peace mission   Mohammed B...	2023-06-16	6925	
4	Sudan: Thousands flee into Chad as West Darfur...	2023-06-16	1181	
...	...	...	...	
1995	World Water Day: Venezuelans without tap water...	2023-03-22	20842	
1996	'UK is a refugee-making country, not a refugee...	2023-03-22	12670	
1997	Lebanon economic crisis: Banking system replac...	2023-03-22	1037882	
1998	What life is like without a water supply   Al ...	2023-03-22	3463	
1999	Why Israel keeps the bodies of Palestinians   ...	2023-03-22	20711	

	like_count	Comments	title_no_stopwords
0	0	0	[Will, Germany's, security, strategy, work?,  ...
1	122	0	[What, makes, cult?]
2	57	34	[Moment, Armenians, shoot, Azerbaijani, guards...
3	214	0	[Headlines, -, African, peace, mission,  , Moh...
4	45	0	[Sudan:, Thousands, flee, Chad, West, Darfur, ...
...	...	...	...
1995	79	9	[World, Water, Day:, Venezuelans, without, tap...
1996	300	194	['UK, refugee-making, country,, refugee-taking...
1997	3588	1753	[Lebanon, economic, crisis:, Banking, system, ...
1998	40	2	[What, life, like, without, water, supply,  , ...
1999	477	473	[Why, Israel, keeps, bodies, Palestinians,  , ...

[2000 rows x 6 columns]

```
[76]: video_df.dtypes
```

```
[76]: Title           object
publish_date      object
View             int64
like_count        int64
Comments          int64
title_no_stopwords object
dtype: object
```

```
[ ]: video_df.View = pd.to_numeric(video_df.View)
video_df.publish_date = pd.to_datetime(video_df.publish_date).dt.date
```

```
[ ]: video_df.like_count = pd.to_numeric(video_df.like_count)
video_df.Comments = pd.to_numeric(video_df.Comments)
```

```
[77]: video_df
```

```
[77]:
```

	Title	publish_date	View	\
0	Will Germany's security strategy work?   Insid...	2023-06-16	10	
1	What makes a cult?	2023-06-16	2107	
2	Moment Armenians shoot at Azerbaijani guards o...	2023-06-16	3748	
3	Headlines - African peace mission   Mohammed B...	2023-06-16	6925	
4	Sudan: Thousands flee into Chad as West Darfur...	2023-06-16	1181	
...	...	...	...	
1995	World Water Day: Venezuelans without tap water...	2023-03-22	20842	
1996	'UK is a refugee-making country, not a refugee...	2023-03-22	12670	
1997	Lebanon economic crisis: Banking system replac...	2023-03-22	1037882	
1998	What life is like without a water supply   Al ...	2023-03-22	3463	
1999	Why Israel keeps the bodies of Palestinians   ...	2023-03-22	20711	

	like_count	Comments	title_no_stopwords
0	0	0	[Will, Germany's, security, strategy, work?,  ...
1	122	0	[What, makes, cult?]
2	57	34	[Moment, Armenians, shoot, Azerbaijani, guards...
3	214	0	[Headlines, -, African, peace, mission,  , Moh...
4	45	0	[Sudan:, Thousands, flee, Chad, West, Darfur, ...
...	...	...	...
1995	79	9	[World, Water, Day:, Venezuelans, without, tap...
1996	300	194	['UK, refugee-making, country,, refugee-taking...
1997	3588	1753	[Lebanon, economic, crisis:, Banking, system, ...
1998	40	2	[What, life, like, without, water, supply,  , ...
1999	477	473	[Why, Israel, keeps, bodies, Palestinians,  , ...

[2000 rows x 6 columns]

```
[78]: video_df.dtypes
```

```
[78]: Title          object
publish_date      object
View              int64
like_count        int64
Comments          int64
title_no_stopwords object
dtype: object
```

```
[79]: video_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  -
0   Title           2000 non-null  object
1   publish_date    2000 non-null  object
```

```

2   View                2000 non-null   int64
3   like_count          2000 non-null   int64
4   Comments            2000 non-null   int64
5   title_no_stopwords  2000 non-null   object
dtypes: int64(3), object(3)
memory usage: 93.9+ KB

```

```
[80]: video_df.describe()
```

```

[80]:
      View    like_count  Comments
count  2.000000e+03  2000.000000  2000.000000
mean   4.900238e+04   558.787000   244.124000
std    2.171418e+05  2031.671909   622.353296
min    1.000000e+01    0.000000    0.000000
25%    4.931500e+03   59.000000   18.000000
50%    1.169650e+04  136.000000   57.500000
75%    3.075950e+04  393.000000  213.000000
max    5.355938e+06 54682.000000 12606.000000

```

```
[ ]: import duckdb
```

## 1.6 Best Performing Video

```
[102]: duckdb.query('select Title , View from video_df order by View desc limit 10')
```

```
[102]:
```

View	Title
5355938	Gold Mafia - Episode 1 - The Laundry Service I Al Jazeera Investigations
5133372	Fighting erupts in Sudan's capital between army, paramilitary   Al Jazeera Newsfeed
2792625	Over 18 million people unable to afford basic needs in Argentina
2737627	Gold Mafia - Episode 2 - Smoke & Mirrors   Al Jazeera Investigations
1757756	Israeli police raid Al-Aqsa Mosque again   Al Jazeera Newsfeed
1747269	Gold Mafia - Episode 3 - El Dorado I Al Jazeera Investigations
1589378	Gold Mafia - Episode 4 - Have The King With You I Al Jazeera Investigations

What's happening in Sudan? | Start Here

1349649

Inside Miami's deadly plastic surgery industry | Fault Lines Documentary

1241032

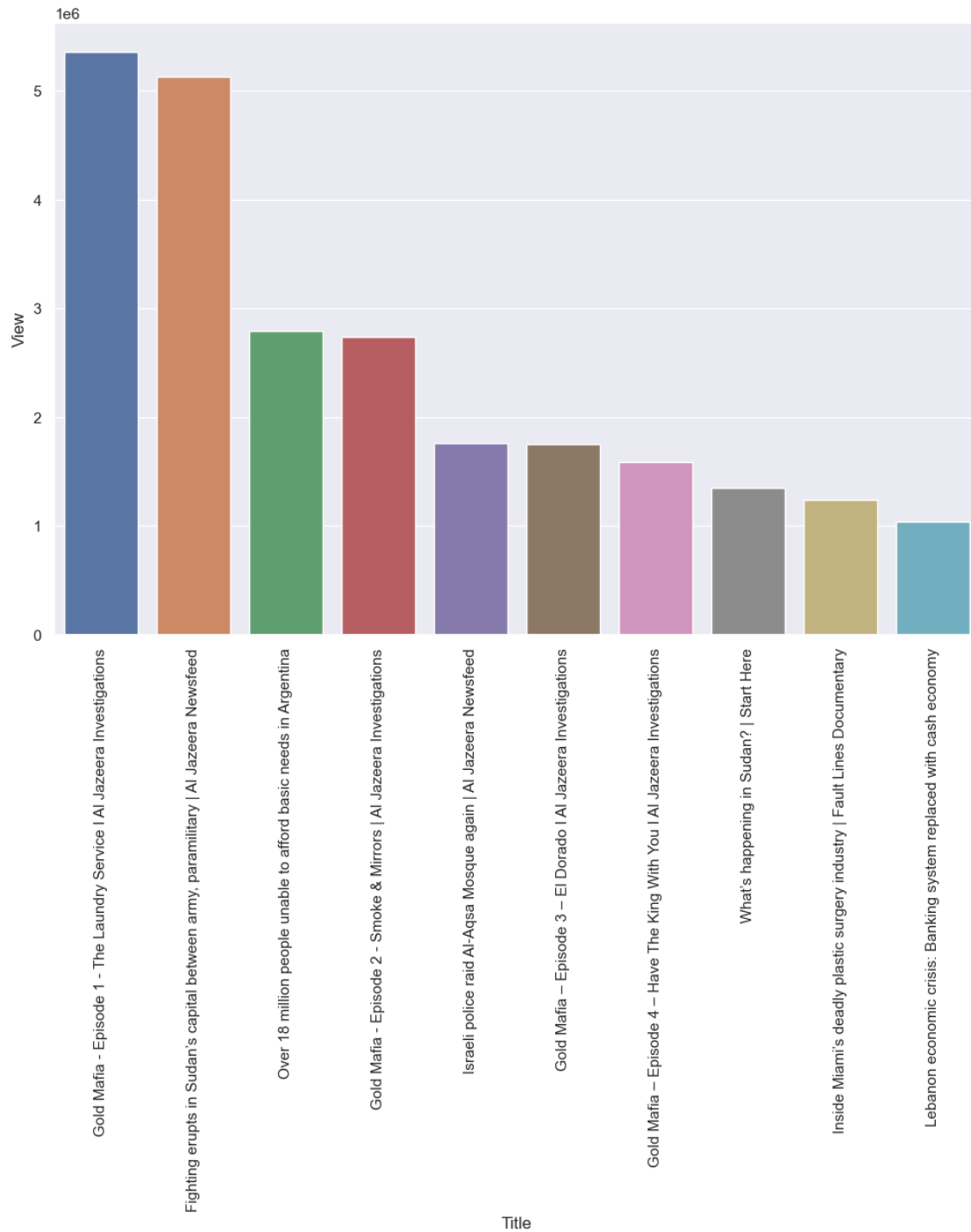
Lebanon economic crisis: Banking system replaced with cash economy

1037882

10 rows

2 columns

```
[103]: ax= sns.barplot(x='Title' , y = 'View' , data = video_df.sort_values('View' ,  
↪ascending=False)[0:10])  
plot = ax.set_xticklabels(ax.get_xticklabels() , rotation=90)
```



## 1.7 Worst Performing Video

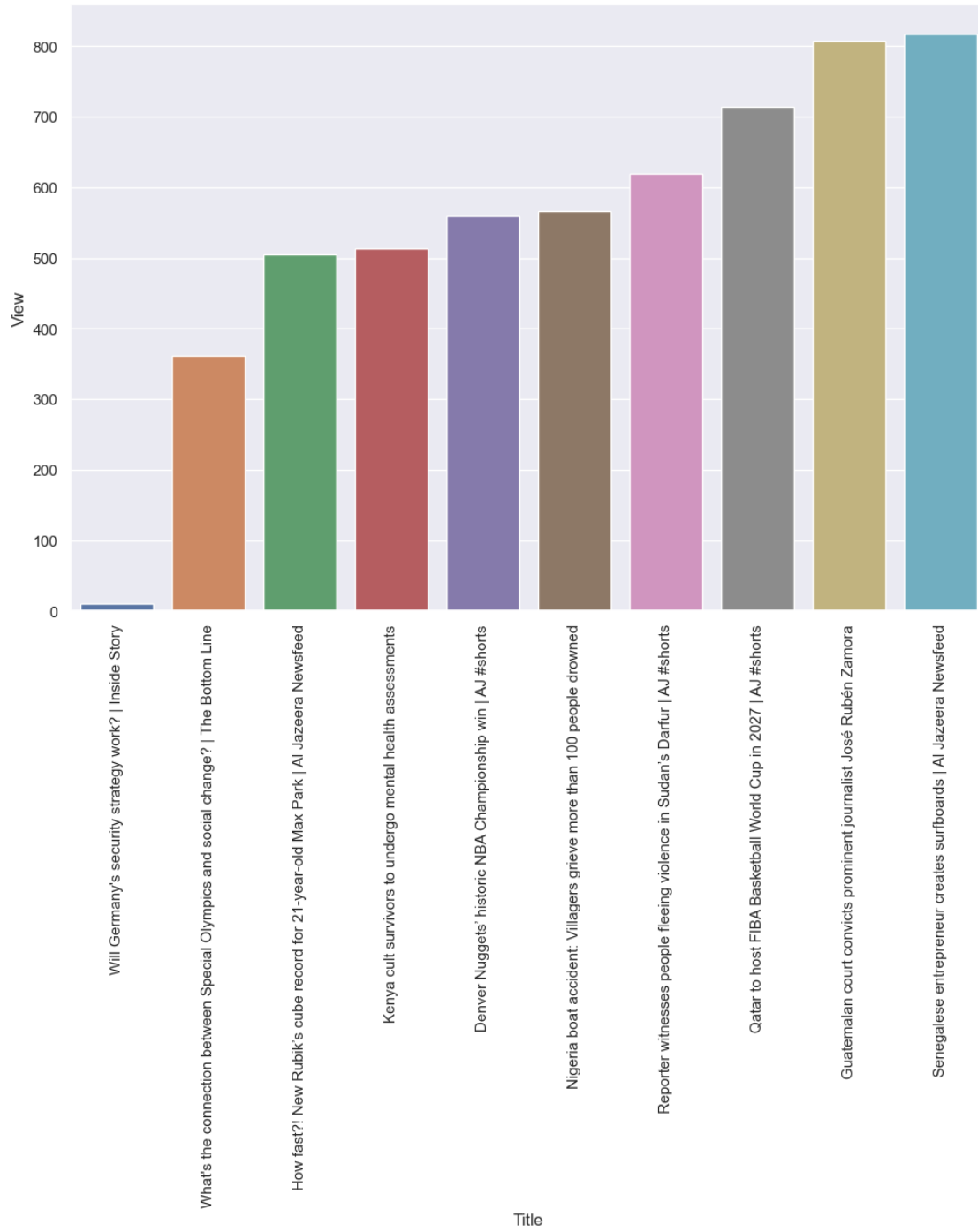
```
[81]: duckdb.query('select Title , View from video_df order by View limit 10')
```

[81]:

	Title
View	
	varchar
int64	
	Will Germany's security strategy work?   Inside Story
10	
	What's the connection between Special Olympics and social change?   The Bottom
Line	361
	How fast?! New Rubik's cube record for 21-year-old Max Park   Al Jazeera
Newsfeed	505
	Kenya cult survivors to undergo mental health assessments
514	
	Denver Nuggets' historic NBA Championship win   AJ #shorts
559	
	Nigeria boat accident: Villagers grieve more than 100 people drowned
567	
	Reporter witnesses people fleeing violence in Sudan's Darfur   AJ #shorts
619	
	Qatar to host FIBA Basketball World Cup in 2027   AJ #shorts
714	
	Guatemalan court convicts prominent journalist José Rubén Zamora
807	
	Senegalese entrepreneur creates surfboards   Al Jazeera Newsfeed
818	
10 rows	
2 columns	

```
[ ]: _ = pd.DataFrame({'title' : video_df.iloc[:, 0]})
_['view'] = video_df.View
```

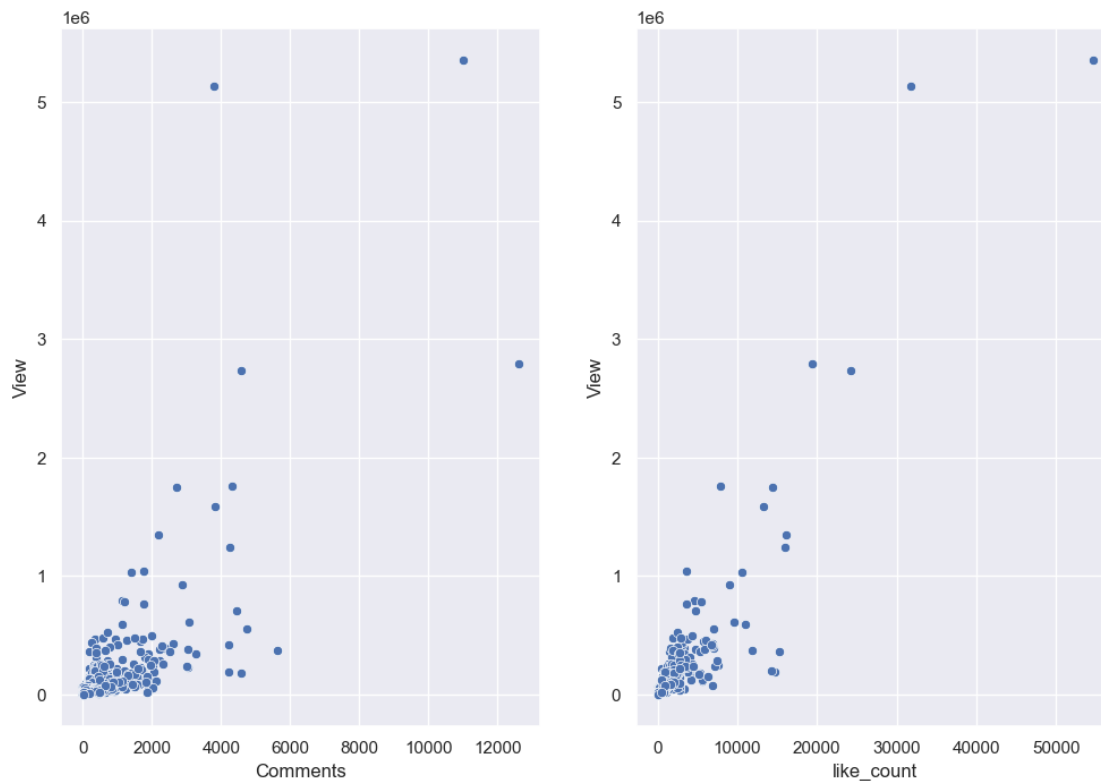
```
[82]: ax= sns.barplot(x='Title' , y = 'View' , data = video_df.sort_values('View' ,
↪ascending=True)[0:10])
plot = ax.set_xticklabels(ax.get_xticklabels() , rotation=90)
```



## 1.8 View vs likes and Comment

```
[83]: fig, ax = plt.subplots(1,2)
sns.scatterplot(data = video_df , x ='Comments' , y= 'View',ax=ax[0])
sns.scatterplot(data=video_df,x='like_count' , y='View',ax=ax[1])
```

```
[83]: <AxesSubplot:xlabel='like_count', ylabel='View'>
```



```
[ ]: from wordcloud import wordcloud
stop_words = set(stopwords.words('english'))
# stop_words.add('Al Jazeera')
video_df['title_no_stopwords'] = video_df['Title'].apply(lambda x: [item for
    ↪ item in str(x).split() if item not in stop_words])

all_words = list([a for b in video_df['title_no_stopwords'].tolist() for a in
    ↪ b])
# all_words.remove('Al Jazeera')
all_words.remove('Al')
all_words.remove('Jazeera')
all_words_str = ' '.join(all_words)
```







### 1.10 Al Jazeera reports on UK

```
[ ]: res=duckdb.query("select Title , View ,title_no_stopwords from video_df where
↳Title Like '%UK%' " ).df()
```

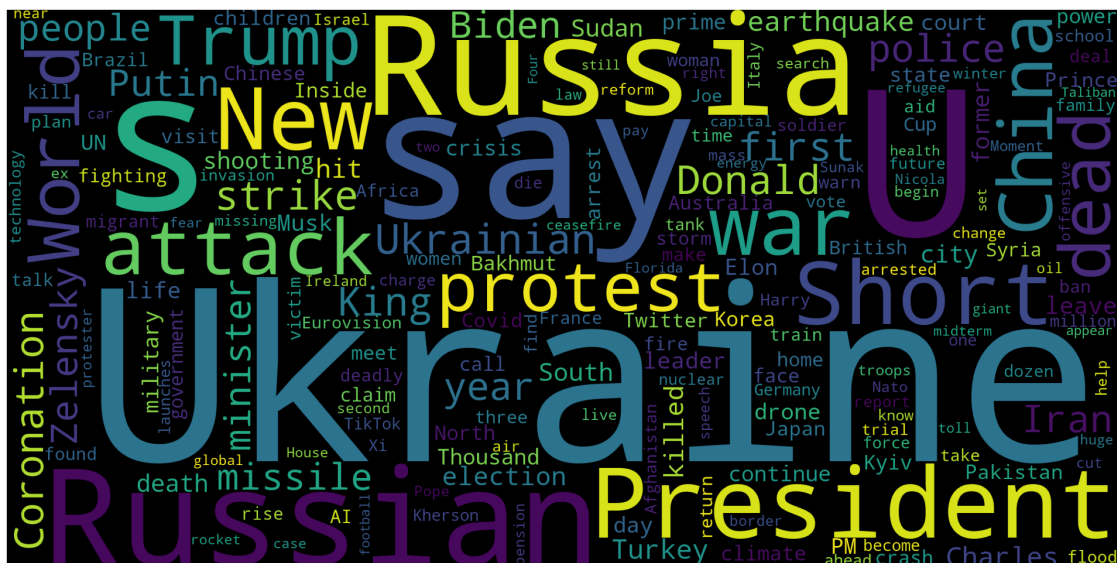
```
[ ]: all_words = list([a for b in res['title_no_stopwords'].tolist() for a in b])
all_words_str = ' '.join(all_words)
```

```
[87]: all_words_str=all_words_str.replace('Al Jazeera',"")
all_words_str=all_words_str.replace('Newsfeed',"")
all_words_str=all_words_str.replace('UK',"")
all_words_str=all_words_str.replace('Headline',"")
```

```
def plot_cloud(wordcloud):
    plt.figure(figsize=(30, 20))
    plt.imshow(wordcloud)
    plt.axis("off");
```

```
wordcloud = WordCloud(width = 2000, height = 1000, random_state=1,
    ↪background_color='black',
    colormap='viridis', collocations=False).
    ↪generate(all_words_str)
```

```
plot_cloud(wordcloud)
```



```
[ ]: video_df.to_excel(r'E:\desktop\Project\Material\video_df.xlsx',
    ↪,index=False,header=True)
```

### 1.11 Analysis for BBC

```
[ ]: playlist_id = df.loc[df.Channel_name == 'BBC News' , 'Playlist_id'].iloc[0]

def get_video_id(youtube,playlist_id):

    request = youtube.playlistItems().list(part =
    ↪'contentDetails',playlistId=playlist_id , maxResults=50)
    response = request.execute()

    video_id = []
    for i in range(len(response['items'])):
        video_id.append(response['items'][i]['contentDetails']['videoId'])

    next_page_token = response.get('nextPageToken')
    more_pages =True

    cnt = 0
    while more_pages:
        if cnt==39:
            more_pages=False
        else:
            request = youtube.playlistItems().list(part =
            ↪'contentDetails',playlistId=playlist_id , maxResults=50,
                                                    pageToken = next_page_token)

            response = request.execute()
            cnt+=1

            for i in range(len(response['items'])):
                video_id.
            ↪append(response['items'][i]['contentDetails']['videoId'])

            next_page_token = response.get('nextPageToken')

    return video_id

video_ids=get_video_id(youtube,playlist_id)
```

```

# Function to get Video details

def get_video_details(youtube,video_ids):
    all_video_stat = []
    for i in range(0,5000,50):

        request = youtube.videos().list( part ='snippet,statistics',
                                          id = ','.join(video_ids[i:i+50]))

        response = request.execute()
        for video in response['items']:
            video_stats = dict(Title = video['snippet']['title'],
                               publish_date = video['snippet']['publishedAt'] ,
                               View = video['statistics']['viewCount'] )
            # like_count = video['statistics']['likeCount'] ,
            # Comments = video['statistics']['commentCount'] )

            all_video_stat.append(video_stats)

    return all_video_stat

#calling above Function

video_details=get_video_details(youtube , video_ids)

video_df_re = pd.DataFrame(video_details)

video_df_re.View = pd.to_numeric(video_df_re.View)
video_df_re.publish_date = pd.to_datetime(video_df_re.publish_date).dt.date
# video_df.like_count = pd.to_numeric(video_df.like_count)
# video_df_re.Comments = pd.to_numeric(video_df_re.Comments)

```

```
[88]: video_df_re.dtypes
```

```

[88]: Title                object
      publish_date         object
      View                int64
      title_no_stopwords   object
      dtype: object

```

```
[89]: video_df_re
```

```

[89]:
      Title publish_date  View \
0  How many Russians have died in the Ukraine war...  2023-06-16  12641
1  Greece migrant boat disaster: What we know so ...  2023-06-16   6653
2  Prince Harry and Meghan: Spotify ends podcast ...  2023-06-16   2606

```

3	Ukraine war: Missile and drone strikes as Afri...	2023-06-16	32500
4	Is the UK's illegal migrant crackdown working?...	2023-06-15	29362
...	...	...	...
1995	Singapore city planners face hybrid working ch...	2022-09-28	42630
1996	Ukraine war round-up: Discredited referendums,...	2022-09-27	350616
1997	Brazil set to vote in tightly fought election ...	2022-09-27	46064
1998	Ukraine accuses Russia of gas pipeline terror ...	2022-09-27	328447
1999	Hurricane Ian makes landfall in Cuba - BBC News	2022-09-27	56153

	title_no_stopwords
0	[How, many, Russians, died, Ukraine, war?, -, ...
1	[Greece, migrant, boat, disaster:, What, know,...
2	[Prince, Harry, Meghan:, Spotify, ends, podcas...
3	[Ukraine, war:, Missile, drone, strikes, Afric...
4	[Is, UK's, illegal, migrant, crackdown, workin...
...	...
1995	[Singapore, city, planners, face, hybrid, work...
1996	[Ukraine, war, round-up:, Discredited, referen...
1997	[Brazil, set, vote, tightly, fought, election,...
1998	[Ukraine, accuses, Russia, gas, pipeline, terr...
1999	[Hurricane, Ian, makes, landfall, Cuba, -, BBC...

[2000 rows x 4 columns]

## 1.12 Best Performing Video

```
[97]: duckdb.query('select Title , View from video_df_re order by View desc limit 10')
```

[97]:

View	Title
int64	varchar
5232362	Family secretly film life in Russian-occupied Ukraine - BBC News
4709713	Scan of Titanic reveals wreck as never seen before - BBC News
3285596	Why China's ex-leader was escorted out of Communist Party congress - BBC News
3156827	Prince Harry arrives at King Charles's Coronation in Westminster Abbey - BBC News
2993749	British fighters in Ukraine share footage of the frontline with BBC Newsnight
	Andrew Tate BBC interview: Influencer challenged on misogyny and rape

```

allegations - BBC News 2892780
Elon Musk's SpaceX Starship rocket explodes after launch - BBC News
2832086
'Massive' drone attack on Black Sea Fleet, Russia says - BBC News
2632627
Russia to withdraw from Ukrainian city of Kherson, western officials say - BBC
News 2486474
Huge blow for Russia as it abandons key Ukrainian city of Kherson - BBC News
2461562

10 rows
2 columns

```

### 1.13 Worst Performing Video

```
[98]: duckdb.query('select Title , View from video_df_re order by View limit 10')
```

[98]:

View	Title
int64	varchar

```

Prince Harry and Meghan: Spotify ends podcast deal with couple #Royals
##PrinceHarry #BBCNews 2606
How Ukraine's cultural treasures are being saved from war - BBC News
4830
Greece migrant boat disaster: What we know so far - BBC News
6653
Is time running out for cabinet secretary Simon Case? - BBC Newsnight
9256
Fears for Sudan animal sanctuary - BBC News
10343
Life as an Iraq refugee 20 years on - BBC News
10638
TVORCHI - Heart Of Steel: Ukraine's 2023 Eurovision entry - BBC News
11448
Malawi schools reopen after deadly cholera outbreak - BBC News
11714
Woman jailed in Iran calls for environmentalists' release - BBC News
11749
Uvalde one year on: How many more mass shootings have there been? - BBC News

```

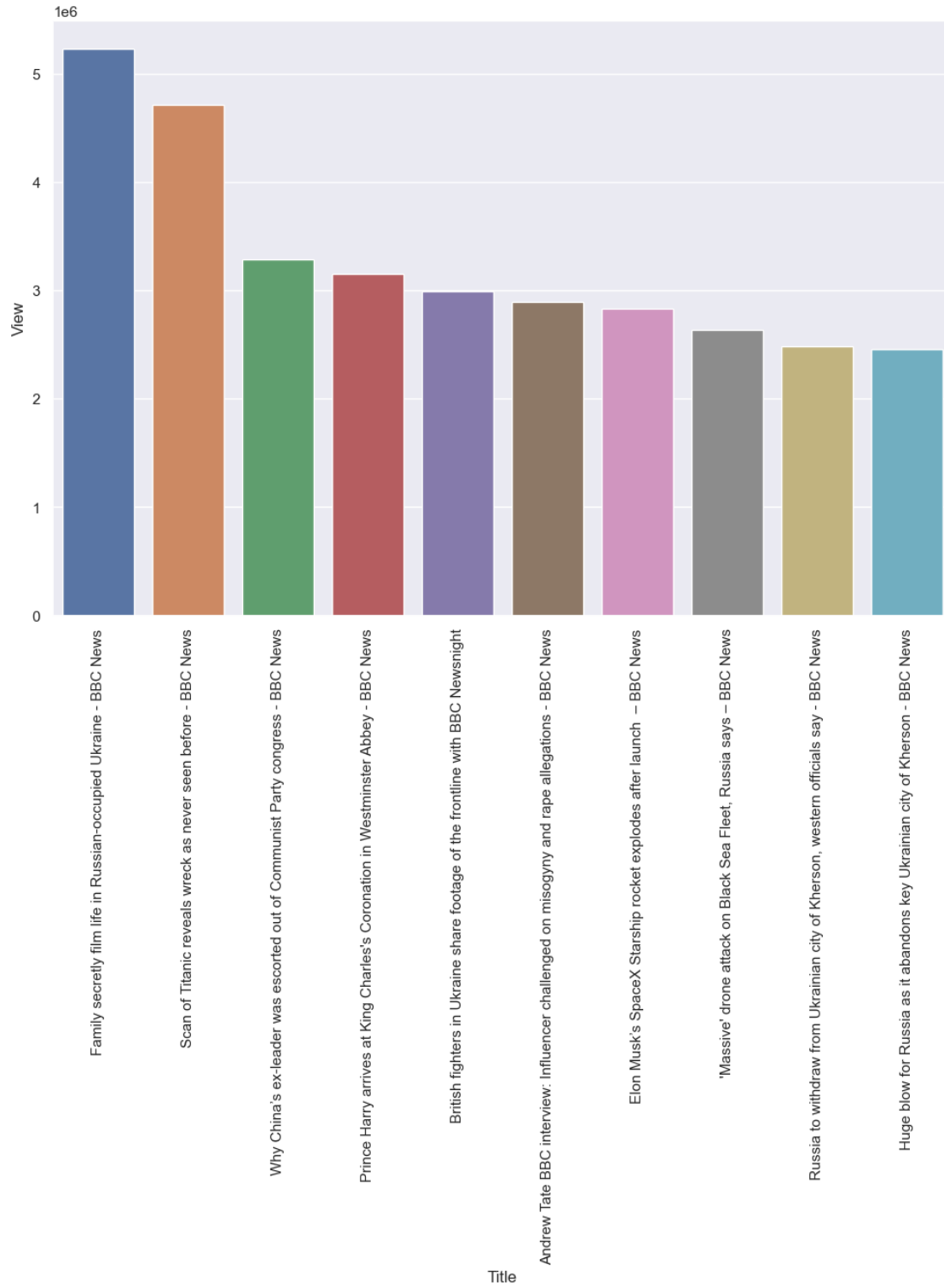
11881

10 rows  
2 columns

### 1.14 Title vs view for Top 10 videos

```
[99]: ax= sns.barplot(x='Title' , y = 'View' , data = video_df_re.sort_values('View'↵  
↵, ascending=False)[0:10])  
plot = ax.set_xticklabels(ax.get_xticklabels() , rotation=90)
```





## 1.15 Wordcloud Map

```
[100]: from wordcloud import wordcloud
stop_words = set(stopwords.words('english'))
# stop_words.add('Al Jazeera')
video_df_re['title_no_stopwords'] = video_df_re['Title'].apply(lambda x: [item_
    ↪for item in str(x).split() if item not in stop_words])

all_words = list([a for b in video_df_re['title_no_stopwords'].tolist() for a_
    ↪in b])
# all_words.remove('BBC NEWS')
# all_words.remove('NEW')
# all_words.remove('Jazeera')
all_words_str = ' '.join(all_words)

all_words_str=all_words_str.replace('BBC','')
all_words_str=all_words_str.replace('News','')
all_words_str=all_words_str.replace('night','')
def plot_cloud(wordcloud):
    plt.figure(figsize=(30, 20))
    plt.imshow(wordcloud)
    plt.axis("off");

wordcloud = WordCloud(width = 2000, height = 1000, random_state=1,
    ↪background_color='black',
    colormap='viridis', collocations=False).
    ↪generate(all_words_str)

plot_cloud(wordcloud)
```

