Youtube API EDA

In [304]:
```
from googleapiclient.discovery import build
import pandas as pd
import numpy as np
import json
import plotly.express as px
import plotly.graph_objs as go

import matplotlib.pyplot as plt
import seaborn as sns
from dateutil import parser
import isodate

import datetime
from datetime import timedelta

import nltk
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
from wordcloud import WordCloud
```

In [141]:
```
api_key = "AIzaSyBHU9cCo329HQLiQ32SnOUxD4RIFAoYlW8"

channel_ids = ["UCOmHUn--16B90oW2L6FRR3A", #BlackPink
               "UCLkAepWjdylmXSltofFvsYQ", #BTS
               "UCIwFjwMjI0y7PDBVEO9-bkQ", #Justin Beiber
               "UCEdvpU2pFRCVqU6yIPyTpMQ", #Marshmello
               "UCfM3zsQsOnfWNUppiycmBuw", #Eminem
               "UC0C-w0YjGpqDXGB8IHb662A", #Ed Sheeran
               "UC9CoOnJkIBMdeijd9qYoT_g", #Ariana Grande
               "UCqECaJ8Gagnn7YCbPEzWH6g", #Taylor Swift
               "UCiGm_E4ZwYSHV3bcW1pnSeQ", #Billie Eilish
               "UCmBA_wu8xGg1OfOkfW13Q0Q"  #Bad Bunny
              ]
```

In [142]:
```
api_service_name = "youtube"
api_version = "v3"

# Get credentials and create an API client
youtube = build(api_service_name, api_version, developerKey=api_key)

# To view Json Data more nicely
# print(json.dumps(response, indent=4, sort_keys=True))
```

In [143]:
```
playlist_id = []
def get_channel_stats(youtube, channel_ids):

#     Get channel statistics: title, subscriber count, view count, video count, upload playlist
#     youtube: the build object from googleapiclient.discovery
#     channels_ids: list of channel IDs

#     Returns:
#     Dataframe containing the channel statistics for all channels in the provided list: title, subscriber count, view count, video count,

    all_data = []
    request = youtube.channels().list(
                part='snippet,contentDetails,statistics',
                id=','.join(channel_ids))
    response = request.execute()

    for i in range(len(response['items'])):
        data = dict(channelName = response['items'][i]['snippet']['title'],
                    subscribers = response['items'][i]['statistics']['subscriberCount'],
                    views = response['items'][i]['statistics']['viewCount'],
                    totalVideos = response['items'][i]['statistics']['videoCount'],
                    playlistId = response['items'][i]['contentDetails']['relatedPlaylists']['uploads'])
        playlist_id.append(response['items'][i]['contentDetails']['relatedPlaylists']['uploads'])
        all_data.append(data)

    return pd.DataFrame(all_data)
```

In [144]:
```
channel_data = get_channel_stats(youtube, channel_ids)
channel_data
```

Out[144]:

	channelName	subscribers	views	totalVideos	playlistId
0	Marshmello	56100000	14082871636	407	UUEdvpU2pFRCVqU6yIPyTpMQ
1	Taylor Swift	50900000	27924804818	205	UUqECaJ8Gagnn7YCbPEzWH6g
2	Bad Bunny	45200000	28429952757	127	UUmBA_wu8xGg1OfOkfW13Q0Q
3	EminemMusic	55200000	25545052490	140	UUfM3zsQsOnfWNUppiycmBuw
4	BLACKPINK	84200000	28975517800	473	UUOmHUn--16B90oW2L6FRR3A
5	Ed Sheeran	52900000	29142568032	262	UU0C-w0YjGpqDXGB8IHb662A
6	Billie Eilish	47300000	12903949122	51	UUiGm_E4ZwYSHV3bcW1pnSeQ
7	Justin Bieber	70900000	29709487712	249	UUIwFjwMjI0y7PDBVEO9-bkQ
8	Ariana Grande	52300000	23161232211	146	UU9CoOnJkIBMdeijd9qYoT_g
9	BANGTANTV	73500000	19429326297	2104	UULkAepWjdylmXSltofFvsYQ

In [98]:
```
playlist_id
```

Out[98]:
```
['UUOmHUn--16B90oW2L6FRR3A',
 'UU9CoOnJkIBMdeijd9qYoT_g',
 'UULkAepWjdylmXSltofFvsYQ',
 'UUmBA_wu8xGg1OfOkfW13Q0Q',
 'UUqECaJ8Gagnn7YCbPEzWH6g',
 'UUIwFjwMjI0y7PDBVEO9-bkQ',
 'UUfM3zsQsOnfWNUppiycmBuw',
 'UU0C-w0YjGpqDXGB8IHb662A',
 'UUEdvpU2pFRCVqU6yIPyTpMQ',
 'UUiGm_E4ZwYSHV3bcW1pnSeQ']
```

In [99]:
```
def get_video_ids(youtube, playlist_id):
    video_ids = []

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

    for item in response['items']:
        video_ids.append(item['contentDetails']['videoId'])

    next_page_token = response.get('nextPageToken')
    while next_page_token is not None:
        request = youtube.playlistItems().list(
                    part='contentDetails',
                    playlistId = playlist_id,
                    maxResults = 50,
                    pageToken = next_page_token)
        response = request.execute()

        for item in response['items']:
            video_ids.append(item['contentDetails']['videoId'])

        next_page_token = response.get('nextPageToken')

    return video_ids
```

In [100]:
```
a = get_video_ids(youtube, playlist_id[0])
b = get_video_ids(youtube, playlist_id[1])
c = get_video_ids(youtube, playlist_id[2])
d = get_video_ids(youtube, playlist_id[3])
e = get_video_ids(youtube, playlist_id[4])
f = get_video_ids(youtube, playlist_id[5])
g = get_video_ids(youtube, playlist_id[6])
h = get_video_ids(youtube, playlist_id[7])
i = get_video_ids(youtube, playlist_id[8])
j = get_video_ids(youtube, playlist_id[9])
```

In [101]:
```
video_ids = a+b+c+d+e+f+g+h+i+j
```

In [102]:
```
def get_video_details(youtube, video_ids):

    all_video_info = []

    for i in range(0, len(video_ids), 50):
        request = youtube.videos().list(
            part="snippet,contentDetails,statistics",
            id=','.join(video_ids[i:i+50])
        )
        response = request.execute()

        for video in response['items']:
            stats_to_keep = {'snippet': ['channelTitle', 'title', 'description', 'tags', 'publishedAt'],
                             'statistics': ['viewCount', 'likeCount', 'favouriteCount', 'commentCount'],
                             'contentDetails': ['duration', 'definition', 'caption']
                            }
            video_info = {}
            video_info['video_id'] = video['id']

            for k in stats_to_keep.keys():
                for v in stats_to_keep[k]:
                    try:
                        video_info[v] = video[k][v]
                    except:
                        video_info[v] = None

            all_video_info.append(video_info)

    return pd.DataFrame(all_video_info)
```

In [103]:
```
video_df = get_video_details(youtube, video_ids)
```

In [104]:
```
video_df
```

Out[104]:

	video_id	channelTitle	title	description	tags	publishedAt	viewCount	likeCount	favouriteCount	commentCount
0	uWWN_kqSvYc	BLACKPINK	BLACKPINK WORLD TOUR [BORN PINK] ABU DHABI HIG...	#BLACKPINK #블랙핑크 #WORLDTOUR #BORNPINK #ABUDHAB...	None	2023-02-03T12:00:41Z	433351	83295	None	1799
1	vB3dItmMVXM	BLACKPINK	BLACKPINK - 'B.P.M.' Roll #10	#BLACKPINK #블랙핑크 #BPM #BORN_PINK_MEMORIES #Rol...	[YG Entertainment, YG, 와이지, K-pop, BLACKPINK, ...	2023-02-03T09:00:42Z	1592153	204667	None	8904
2	NNmtPJcV3AE	BLACKPINK	BLACKPINK - 'B.P.M.' Roll #9	#BLACKPINK #블랙핑크 #BPM #BORN_PINK_MEMORIES #Rol...	[YG Entertainment, YG, 와이지, K-pop, BLACKPINK, ...	2023-01-27T09:00:15Z	1473668	183802	None	6886
3	CoocTMrspKs	BLACKPINK	BLACKPINK WORLD TOUR [BORN PINK] RIYADH HIGHLI...	#BLACKPINK #블랙핑크 #WORLDTOUR #BORNPINK #RIYADH ...	None	2023-01-26T12:00:06Z	567704	103727	None	2414
4	zfCo-X7UmKI	BLACKPINK	BLACKPINK - 'B.P.M.' Roll #8	#BLACKPINK #블랙핑크 #BPM #BORN_PINK_MEMORIES #Rol...	[YG Entertainment, YG, 와이지, K-pop, BLACKPINK, ...	2023-01-20T09:00:40Z	2009764	239517	None	8256
...	...	...	...	...	...	...	...	...	...	...
5031	hG4lT4fxj8M	BillieEilishVEVO	Billie Eilish - Ocean Eyes (Dance Performance ...	Listen to "ocean eyes" from "dont smile at me"...	[Billie, Eilish, Ocean, Eyes, Darkroom/Intersc...	2016-11-22T15:00:04Z	95716087	1605728	None	27685
5032	FQ0iq10ULNA	Billie Eilish	Billie Eilish - Six Feet Under	follow me @ \nhttps://twitter.com/Billie_eilis...	[music video, billie eilish, six feet under, o...	2016-07-01T04:10:33Z	32500527	530121	None	10072
5033	viimfQi_pUw	Billie Eilish	Billie Eilish - Ocean Eyes (Official Music Video)	follow me @\n\nhttps://www.facebook.com/billie...	None	2016-03-24T15:46:50Z	422870974	5716419	None	164165
5034	d--DyK0wtYo	Billie Eilish	Billie Eilish - Ocean Eyes (Official Audio) - ...	-Ocean Eyes- \n\nI've been watching you\nFor s...	[Ocean Eyes (Musical Album), Lyrics (Website C...	2015-11-30T20:50:31Z	2217443	103749	None	4866
5035	es0Y8ikir1k	Billie Eilish	Billie Eilish and Simone Midby do Aerial Silks...	Billie Eilish and Simone Midby do Aerial Silks...	[iMovie]	2014-02-17T00:24:59Z	18549935	814226	None	32317

5036 rows × 13 columns

In [105]:
```
def get_comments_in_videos(youtube, video_ids):

#     Get top level comments as text from all videos with given IDs (only the first 10 comments due to quote limit of Youtube API):

#     youtube: the build object from googleapiclient.discovery
#     video_ids: list of video IDs

#     Returns:
#     Dataframe with video IDs and associated top level comment in text.

    all_comments = []

    for video_id in video_ids:
        try:
            request = youtube.commentThreads().list(
                part="snippet,replies",
                videoId=video_id
            )
            response = request.execute()

            comments_in_video = [comment['snippet']['topLevelComment']['snippet']['textOriginal'] for comment in response['items'][0:10]]
            comments_in_video_info = {'video_id': video_id, 'comments': comments_in_video}

            all_comments.append(comments_in_video_info)

        except:
            # When error occurs - most likely because comments are disabled on a video
            print('Could not get comments for video ' + video_id)

    return pd.DataFrame(all_comments)
```

In [145]:
```
comments_df = get_comments_in_videos(youtube, video_ids)
```

```
Could not get comments for video OqEq4tOSpZI
Could not get comments for video 8hOo7Ms5Syk
Could not get comments for video rKltTMnp4G4
Could not get comments for video Wxjo9OJskYM
Could not get comments for video StW5s8UA2YU
Could not get comments for video YgG9f4MJ1eU
Could not get comments for video PfLCyR6Efvw
Could not get comments for video jpStWV2GHF4
Could not get comments for video adMBDxnhJMw
Could not get comments for video SlMOw2vzDLU
Could not get comments for video Iiukq_ilT0Y
Could not get comments for video RmZ3DPJQo2k
Could not get comments for video L38H9yVb3d8
Could not get comments for video oWVYzCPs3nE
Could not get comments for video e7HLU5-0VU4
Could not get comments for video gOr6L844U9I
Could not get comments for video Yio7S8uMNeE
Could not get comments for video y1iJV2snVoA
Could not get comments for video jON5yjNDM_U
Could not get comments for video 7U9CKtcIwwk
Could not get comments for video BgcF5Pu8lRg
Could not get comments for video trG0aD4Buv4
Could not get comments for video ko8gepM8MBU
Could not get comments for video KJWZSEkCrAM
Could not get comments for video Zv4ftT043S0
Could not get comments for video 5yIGhsydtUQ
Could not get comments for video 7xUYUJ69BnE
Could not get comments for video dhcHYYyrjJc
Could not get comments for video pBmixgZ7MLA
Could not get comments for video A8cXaCtUrT8
Could not get comments for video Xv4oBVBHoQA
Could not get comments for video B-yotIpRLns
Could not get comments for video 7s4LphPaYvc
Could not get comments for video 0wpBUcYzFG0
Could not get comments for video 3YolugGV-Og
```

In [128]:
```
comments_df
```

Out[128]:

	video_id	comments
0	uWWN_kqSvYc	[I love you, HELP FOR TURKEY PLS, 브라질로 와주세요, B...
1	vB3dItmMVXM	[I was pleasantly surprised to see Selena! Gre...
2	NNmtPJcV3AE	[안녕하세요, 귀하의 요청이 있습니다. 터키에서 큰 지진이 일어났습니다. Hatay...
3	CoocTMrspKs	[👑 👑 👑 👑 🌠, Turkey please, انا مش اقدر اروح الحف...
4	zfCo-X7UmKI	[Love BLACK PINK Love Roze and Lisa and Jinni ...
...	...	...
1244	HoMAsXuPDog	[I'm Dynamite.🎶🎤, 😙😙😙😙🍮🍮🍮🍦🍦🍦🦊🦊🦊🦊🥺🥺🥺😭💜💜💜,...
1245	-vUKNPHT1EM	[Omg, Jimin you're so handsome,, 👹👹👹👹👹👹👹👹👹👹👹😍😍😍...
1246	5wOqLWi7Txw	[HES SO ANGELIC WTH 👹🥺, Dios t guíe siempre mi...
1247	xIzBCXEdmEo	[Love you jiminaaaaaaaaa💜💜💜💜💜💜💜💜💜💜💜💜💜💜💜💜💜💜💜💜...
1248	F43Rj9c1TQs	[BTS travel book 届きました🎁\nQRコードで飛んできました〜✈️, ı I...

1249 rows × 2 columns

In [ ]:

Preprocessing

In [108]:
```
video_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5036 entries, 0 to 5035
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   video_id        5036 non-null   object
 1   channelTitle    5036 non-null   object
 2   title           5036 non-null   object
 3   description     5036 non-null   object
 4   tags            4748 non-null   object
 5   publishedAt     5036 non-null   object
 6   viewCount       5015 non-null   object
 7   likeCount       5011 non-null   object
 8   favouriteCount  0 non-null      object
 9   commentCount    5020 non-null   object
 10  duration        5036 non-null   object
 11  definition      5036 non-null   object
 12  caption         5036 non-null   object
dtypes: object(13)
memory usage: 511.6+ KB
```

In [205]:
```
# Convert count columns to numeric columns
numeric_cols = ['subscribers', 'views', 'totalVideos']
channel_data[numeric_cols] = channel_data[numeric_cols].apply(pd.to_numeric, errors='coerce')
```

In [206]:
```
channel_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 5 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   channelName  10 non-null     object
 1   subscribers  10 non-null     int64 
 2   views        10 non-null     int64 
 3   totalVideos  10 non-null     int64 
 4   playlistId   10 non-null     object
dtypes: int64(3), object(2)
memory usage: 528.0+ bytes
```

In [111]:
```
video_df.isna().any()
```

Out[111]:
```
video_id          False
channelTitle      False
title             False
description       False
tags               True
publishedAt       False
viewCount          True
likeCount          True
favouriteCount     True
commentCount       True
duration          False
definition        False
caption           False
dtype: bool
```

In [112]:
```
video_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5036 entries, 0 to 5035
Data columns (total 13 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   video_id        5036 non-null   object
 1   channelTitle    5036 non-null   object
 2   title           5036 non-null   object
 3   description     5036 non-null   object
 4   tags            4748 non-null   object
 5   publishedAt     5036 non-null   object
 6   viewCount       5015 non-null   object
 7   likeCount       5011 non-null   object
 8   favouriteCount  0 non-null      object
 9   commentCount    5020 non-null   object
 10  duration        5036 non-null   object
 11  definition      5036 non-null   object
 12  caption         5036 non-null   object
dtypes: object(13)
memory usage: 511.6+ KB
```

In [113]:
```
# Dropping Irrelevant Data
video_df.drop("favouriteCount",axis=1,inplace=True)
```

In [237]:
```
# Coverting Video datatypes
cols = ['viewCount', 'likeCount', 'commentCount']
video_df[cols] = video_df[cols].apply(pd.to_numeric, errors='coerce', axis=1)
```

In [129]:
```
video_df.isna().sum()
```

Out[129]:
```
video_id          0
channelTitle      0
title             0
description       0
tags            288
publishedAt       0
viewCount        21
likeCount        25
commentCount     16
duration          0
definition        0
caption           0
day_of_week       0
durationSecs      0
tagsCount         0
titleLength       0
dtype: int64
```

In [130]:
```
# To find Day through Published Data
video_df['publishedAt'] = pd.to_datetime(video_df['publishedAt'], format="%Y-%m-%dT%H:%M:%SZ")
video_df['day_of_week'] = video_df['publishedAt'].dt.strftime("%A")
```

In [131]:
```
# Convert duration to seconds
video_df['durationSecs'] = video_df['duration'].apply(lambda x: isodate.parse_duration(x))
video_df['durationSecs'] = video_df['durationSecs'].astype('timedelta64[s]')
```

In [132]:
```
# Count number of tags
video_df['tagsCount'] = video_df['tags'].apply(lambda x: 0 if x is None else len(x))
```

In [133]:
```
# Length of the title
video_df['titleLength'] = video_df['title'].apply(lambda x: len(x))
```

In [134]:
```
percent = (video_df.isnull().sum()/video_df.isnull().count()*100)
```

In [135]:
```
percent
```

Out[135]:
```
video_id        0.000000
channelTitle    0.000000
title           0.000000
description     0.000000
tags            5.718824
publishedAt     0.000000
viewCount       0.416998
likeCount       0.496426
commentCount    0.317712
duration        0.000000
definition      0.000000
caption         0.000000
day_of_week     0.000000
durationSecs    0.000000
tagsCount       0.000000
titleLength     0.000000
dtype: float64
```

In [149]:
```
video_df.fillna(value=0,inplace=True)
```

In [162]:
```
video_df.to_excel("Videodata_Youtube.xlsx")
```

```
C:\Users\Harshavardhan\anaconda3\lib\site-packages\xlsxwriter\worksheet.py:941: UserWarning: Ignoring URL 'http://www.myspace.com/justinbi
eber

Justin%20Bieber%20singing%20an%20original%20song%20in%20*STEREO*%20written%20by%20Jake%20Leiske%20and%20produced%20by%20Jay%20Riehl%20call
ed%20%22Set%20a%20Place%20at%20Your%20Table%22.

The%20song%20was%20produced%20for%20a%20CD%20to%20raise%20money%20for%20the%20local%20food%20bank.%20

The%20Christmas%20CD%20is%20released%20now,%20but%20unfortunately,%20the%20CD%20will%20not%20be%20sold%20online.%20100%25%20of%20proceeds%
20are%20still%20going%20to%20the%20food%20bank%20here,%20but%20the%20CD%20will%20only%20be%20sold%20in%20local%20stores.

Justin%20sang%202%20songs%20and%20the%20rest%20are%20sung%20by%20other%20great%20talented%20local%20kids.

If%20you%20haven't%20subscribed%20already%20please%20do!%20
http://www.youtube.com/subscription_c...%20 (http://www.youtube.com/subscription_c...%20)
Thanks%20to%20all%20of%20you%20who%20have,%20and%20for%20all%20of%20your%20love%20and%20support!%20

This%20CD%20is%20a%20way%20that%20we%20can%20help%20our%20local%20food%20bank.%20Please%20consider%20supporting%20your%20local%20food%20ba
nk%20as%20well%20this%20Christmas.%20Happy%20Holidays!


%22Set%20a%20Place%20at%20Your%20Table%22
Performed%20by%20Justin%20Bieber
Written%20by%20Jake%20Leiske
Produced%20by%20Jay%20Riehl


%22Things%20seemed%20strange%20this%20morning
As%20I%20packed%20my%20bag%20for%20school
The%20smiles%20I'm%20used%20to%20seeing%20just%20weren't%20there
I%20heard%20my%20father%20talking%20
To%20my%20mother%20down%20the%20hallway
Above%20the%20whispers%20all%20that%20I%20could%20hear...%20was

Chorus:
Set%20a%20place%20at%20the%20table%20for%20someone%20who%20might%20need%20it
Set%20a%20place%20at%20the%20table%20for%20someone%20to%20sit%20down%20and%20eat
Kick%20their%20shoes%20off,%20put%20their%20feet%20up
Forget%20about%20their%20troubles%20for%20awhile
When%20there's%20a%20place%20at%20your%20table
There's%20a%20little%20hope%20for%20you%20and%20i

i%20was%20feeling%20kinda'%20helpless%20
About%20the%20words%20that%20I%20just%20heard
Could%20it%20be%20the%20family%202%20doors%20down?
They%20said%20a%20couple%20months%20of%20bad%20luck
Had%20them%20up%20against%20the%20wall
And%20though%20it%20may%20seem%20small%20
i%20know%20there's%20somethin'%20i%20can%20do

Chorus

Make%20room%20at%20your%20table%20for%20your%20friends%20and%20family
Take%20the%20time%20to%20share%20your%20day,%20just%20laugh%20talk%20and%20eat

Chorus

...there's%20a%20little%20hope%20for%20you%20and%20I' with link or location/anchor > 2079 characters since it exceeds Excel's limit for UR
LS
  warn("Ignoring URL '%s' with link or location/anchor > %d "
```

In [220]:
```
to_replace = {'BANGTANTV': 'BTS', 'MarshmelloVEVO': 'Marshmello', 'EdSheeranVEVO': 'EdSheeran',
              'TaylorSwiftVEVO': 'Taylor Swift', 'JustinBieberVEVO': 'Justin Bieber', 'ArianaGrande': 'Ariana Grande',
              'EminemVEVO': 'EminemMusic', 'BillieEilishVEVO': 'Billie Eilish', 'EdSheeran': 'Ed Sheeran',}
video_df.channelTitle.replace(to_replace,inplace=True)
```

In [161]:
```
video_df.channelTitle.unique()
```

Out[161]:
```
array(['BLACKPINK', 'Ariana Grande', 'BTS', 'Bad Bunny', 'Taylor Swift',
       'Justin Bieber', 'EminemMusic', 'Ed Sheeran', 'Marshmello',
       'Billie Eilish'], dtype=object)
```

Graphs

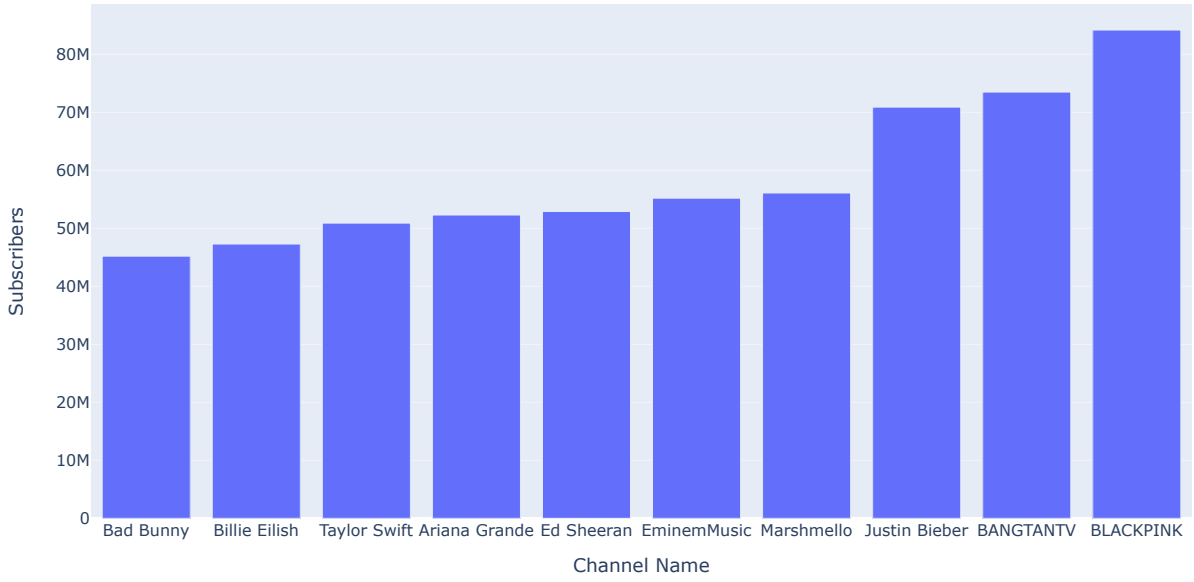
In [222]:
```
channel_data
```

Out[222]:

	channelName	subscribers	views	totalVideos	playlistId
0	Marshmello	56100000	14082871636	407	UUEdvpU2pFRCVqU6yIPyTpMQ
1	Taylor Swift	50900000	27924804818	205	UUqECaJ8Gagnn7YCbPEzWH6g
2	Bad Bunny	45200000	28429952757	127	UUmBA_wu8xGg1OfOkfW13Q0Q
3	EminemMusic	55200000	25545052490	140	UUfM3zsQsOnfWNUppiycmBuw
4	BLACKPINK	84200000	28975517800	473	UUOmHUn--16B90oW2L6FRR3A
5	Ed Sheeran	52900000	29142568032	262	UU0C-w0YjGpqDXGB8IHb662A
6	Billie Eilish	47300000	12903949122	51	UUiGm_E4ZwYSHV3bcW1pnSeQ
7	Justin Bieber	70900000	29709487712	249	UUIwFjwMjI0y7PDBVEO9-bkQ
8	Ariana Grande	52300000	23161232211	146	UU9CoOnJkIBMdeijd9qYoT_g
9	BANGTANTV	73500000	19429326297	2104	UULkAepWjdylmXSltofFvsYQ

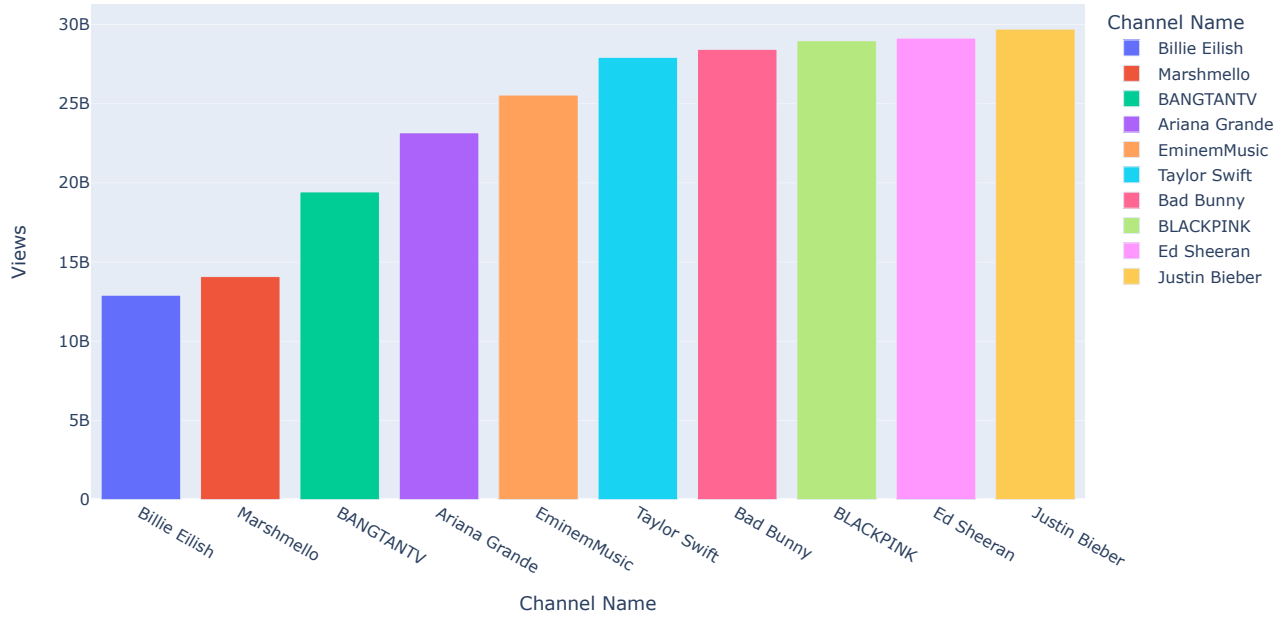
1) Channels VS Subscribers

In [250]:
```
sorted_channel_data = channel_data.sort_values(by="subscribers")
fig = px.bar(sorted_channel_data,
             x="channelName",
             y="subscribers",
             labels={"subscribers":"Subscribers","channelName":"Channel Name"})
fig.show()
```



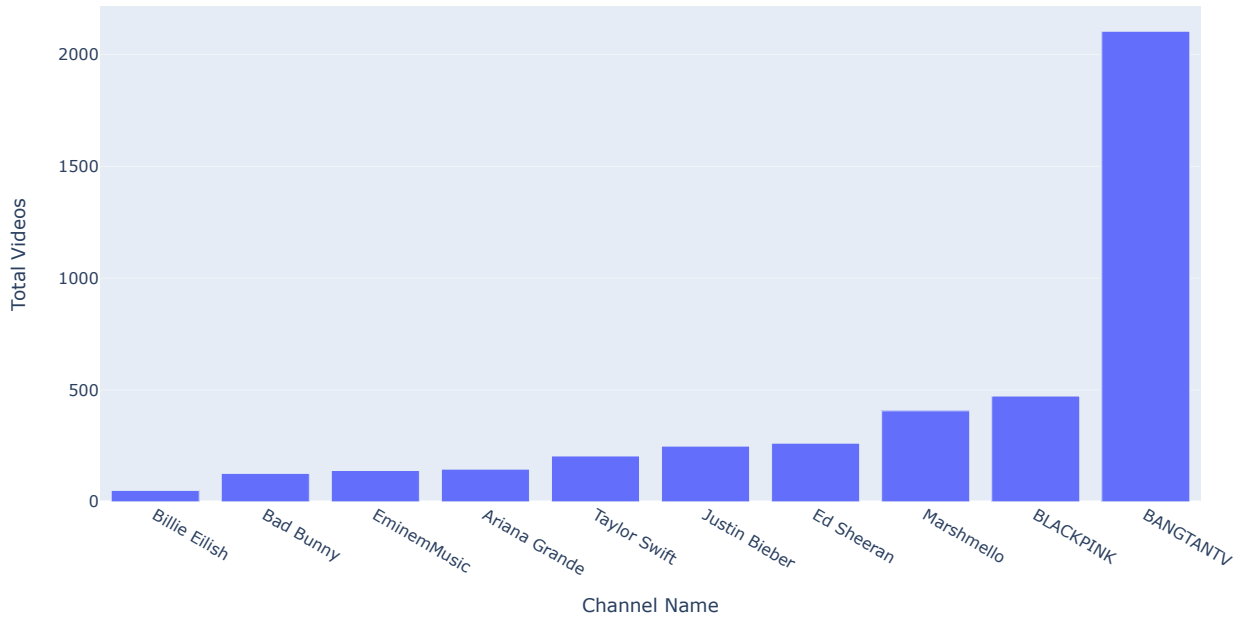
2) Channel vs Views

In [249]:
```
sorted_channel_data = channel_data.sort_values(by="views")
fig = px.bar(sorted_channel_data,
             x="channelName",
             y="views",color="channelName",
             labels={"views":"Views","channelName":"Channel Name"})
fig.show()
```

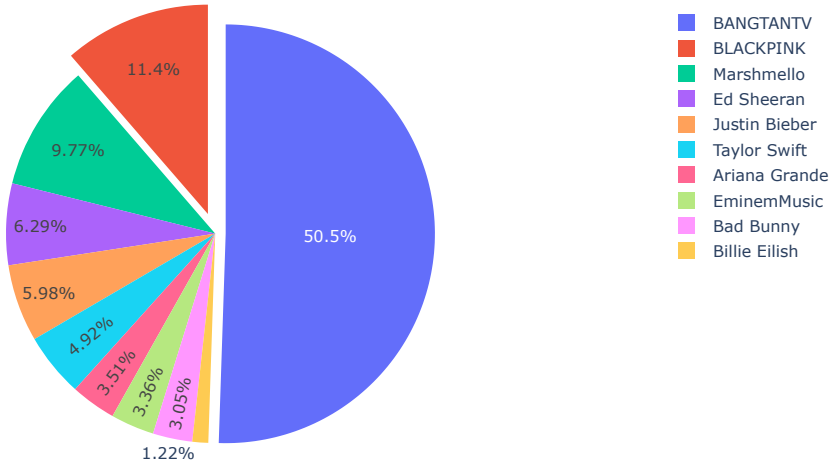


3) Channel vs Total Uploads

In [248]:
```
sorted_channel_data = channel_data.sort_values(by="totalVideos")
fig = px.bar(sorted_channel_data,
             x="channelName",
             y="totalVideos",
             labels={"totalVideos":"Total Videos","channelName":"Channel Name"})
fig.show()
```



In [359]:
```
fig = go.Figure(data=[go.Pie(labels=sorted_channel_data.channelName,
                             values=sorted_channel_data.totalVideos,
                             pull=[0,0,0,0,0,0,0,0,0.05,0.1])])
fig
```



In [225]:
```
video_df
```

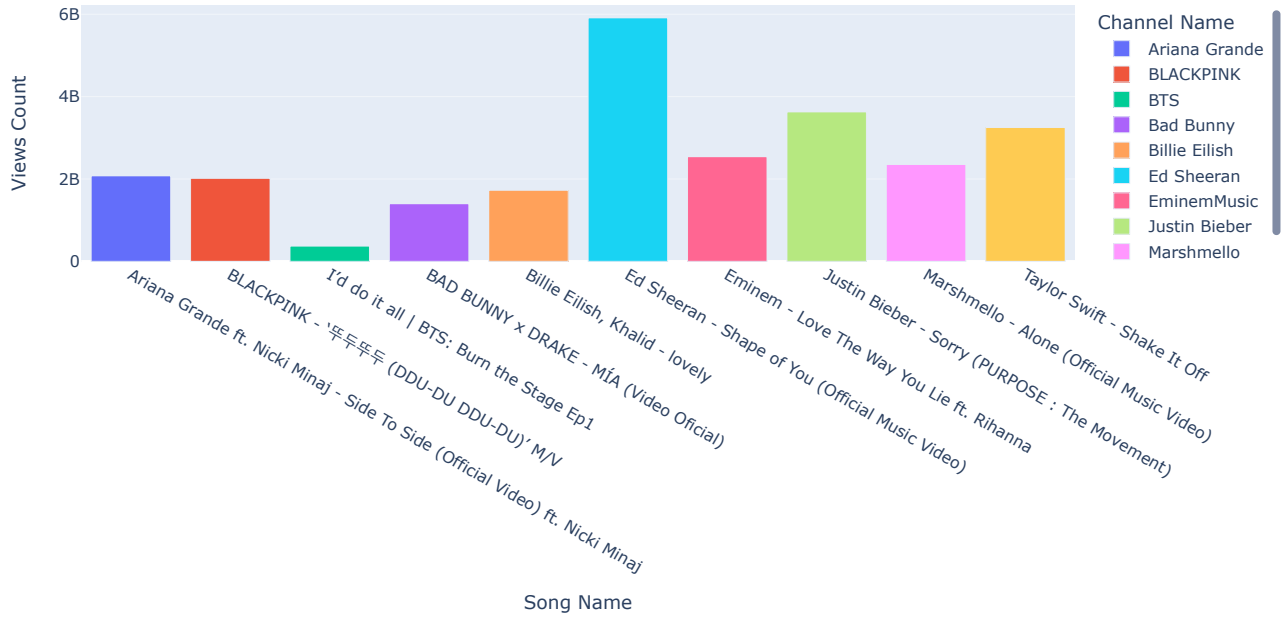
Out[225]:

	video_id	channelTitle	title	description	tags	publishedAt	viewCount	likeCount	commentCount	duration	definit
0	uWWN_kqSvYc	BLACKPINK	BLACKPINK WORLD TOUR [BORN PINK] ABU DHABI HIG...	#BLACKPINK #블랙핑크 #WORLDTOUR #BORNPINK #ABUDHAB...	0	2023-02-03 12:00:41	433351	83295	1799	PT45S	
1	vB3dItmMVXM	BLACKPINK	BLACKPINK - 'B.P.M.' Roll #10	#BLACKPINK #블랙핑크 #BPM #BORN_PINK_MEMORIES #Rol...	[YG Entertainment, YG, 와이지, K-pop, BLACKPINK, ...	2023-02-03 09:00:42	1592153	204667	8904	PT10M18S	
2	NNmtPJcV3AE	BLACKPINK	BLACKPINK - 'B.P.M.' Roll #9	#BLACKPINK #블랙핑크 #BPM #BORN_PINK_MEMORIES #Rol...	[YG Entertainment, YG, 와이지, K-pop, BLACKPINK, ...	2023-01-27 09:00:15	1473668	183802	6886	PT10M3S	
3	CoocTMrspKs	BLACKPINK	BLACKPINK WORLD TOUR [BORN PINK] RIYADH HIGHLI...	#BLACKPINK #블랙핑크 #WORLDTOUR #BORNPINK #RIYADH ...	0	2023-01-26 12:00:06	567704	103727	2414	PT41S	
4	zfCo-X7UmKI	BLACKPINK	BLACKPINK - 'B.P.M.' Roll #8	#BLACKPINK #블랙핑크 #BPM #BORN_PINK_MEMORIES #Rol...	[YG Entertainment, YG, 와이지, K-pop, BLACKPINK, ...	2023-01-20 09:00:40	2009764	239517	8256	PT10M15S	
...	...	...	...	...	...	...	...	...	...	...	
5031	hG4IT4fxj8M	Billie Eilish	Billie Eilish - Ocean Eyes (Dance Performance ...	Listen to "ocean eyes" from "dont smile at me"...	[Billie, Eilish, Ocean, Eyes, Darkroom/Intersc...	2016-11-22 15:00:04	95716087	1605728	27685	PT3M21S	
5032	FQ0iq10ULNA	Billie Eilish	Billie Eilish - Six Feet Under	follow me @ \nhttps://twitter.com/Billie_eilis...	[music video, billie eilish, six feet under, o...	2016-07-01 04:10:33	32500527	530121	10072	PT3M14S	
5033	viimfQi_pUw	Billie Eilish	Billie Eilish - Ocean Eyes (Official Music Video)	follow me @\n\nhttps://www.facebook.com/billie...	0	2016-03-24 15:46:50	422870974	5716419	164165	PT3M21S	
5034	d--DyK0wtYo	Billie Eilish	Billie Eilish - Ocean Eyes (Official Audio) - ...	-Ocean Eyes- \n\nI've been watching you\nFor s...	[Ocean Eyes (Musical Album), Lyrics (Website C...	2015-11-30 20:50:31	2217443	103749	4866	PT3M21S	
5035	es0Y8ikir1k	Billie Eilish	Billie Eilish and Simone Midby do Aerial Silks...	Billie Eilish and Simone Midby do Aerial Silks...	[iMovie]	2014-02-17 00:24:59	18549935	814226	32317	PT3M36S	

5036 rows × 16 columns

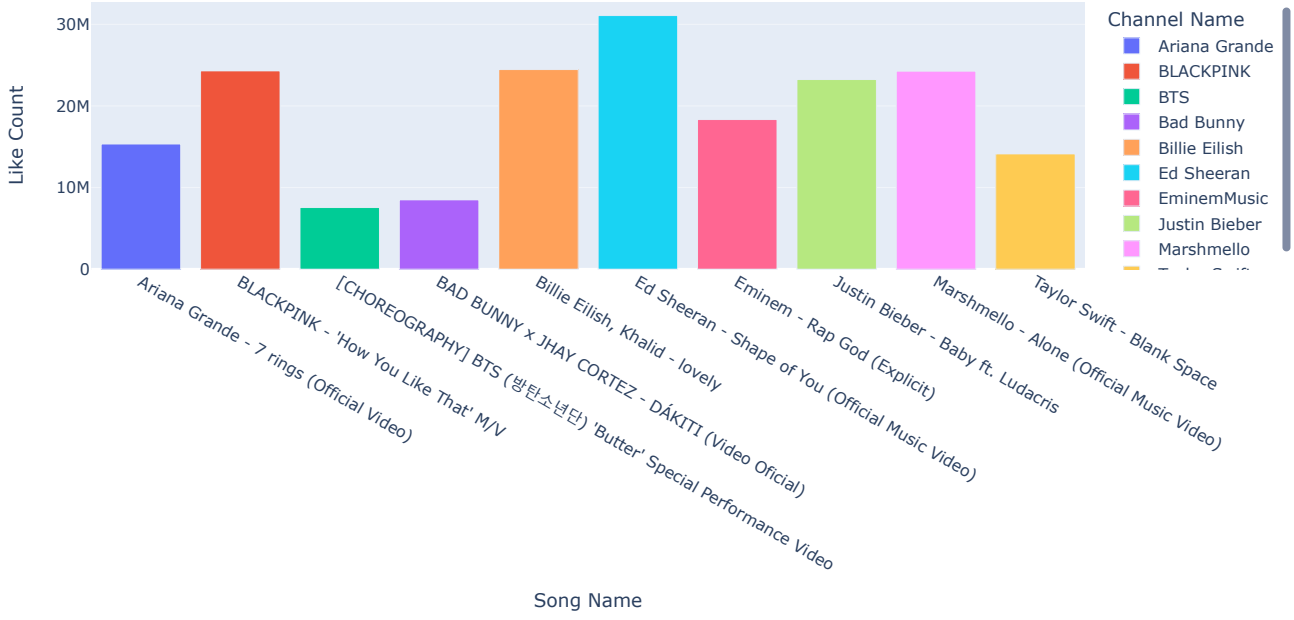
4) Most Viewed Videos of each Channel

In [322]:
```
grouped_df = video_df.groupby(['channelTitle']).apply(lambda x: x.nlargest(1, 'viewCount'))
px.bar(grouped_df,
       x="title",
       y="viewCount",
       color="channelTitle",
       labels={"viewCount":"Views Count","title":"Song Name","channelTitle" :"Channel Name"}
      )
```



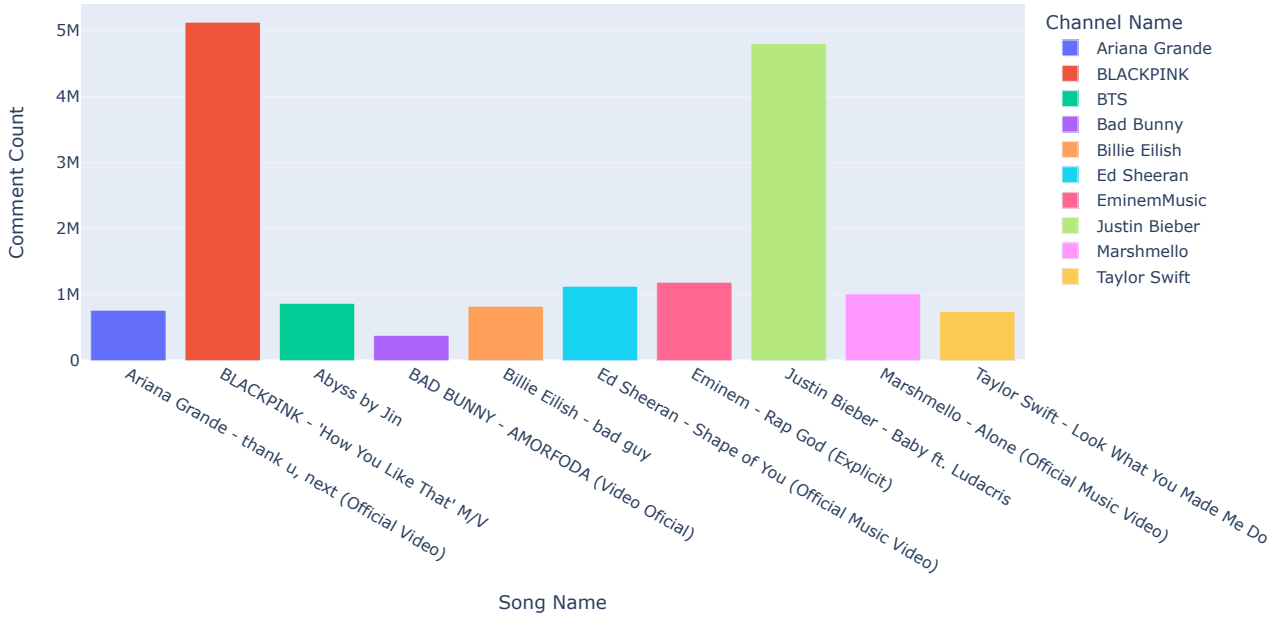
5) Most Liked Videos of each Channel

In [323]:
```
grouped_df = video_df.groupby(['channelTitle']).apply(lambda x: x.nlargest(1, 'likeCount'))
px.bar(grouped_df,
       x="title",
       y="likeCount",
       color="channelTitle",
       labels={"likeCount":"Like Count","title":"Song Name","channelTitle" :"Channel Name"})
```



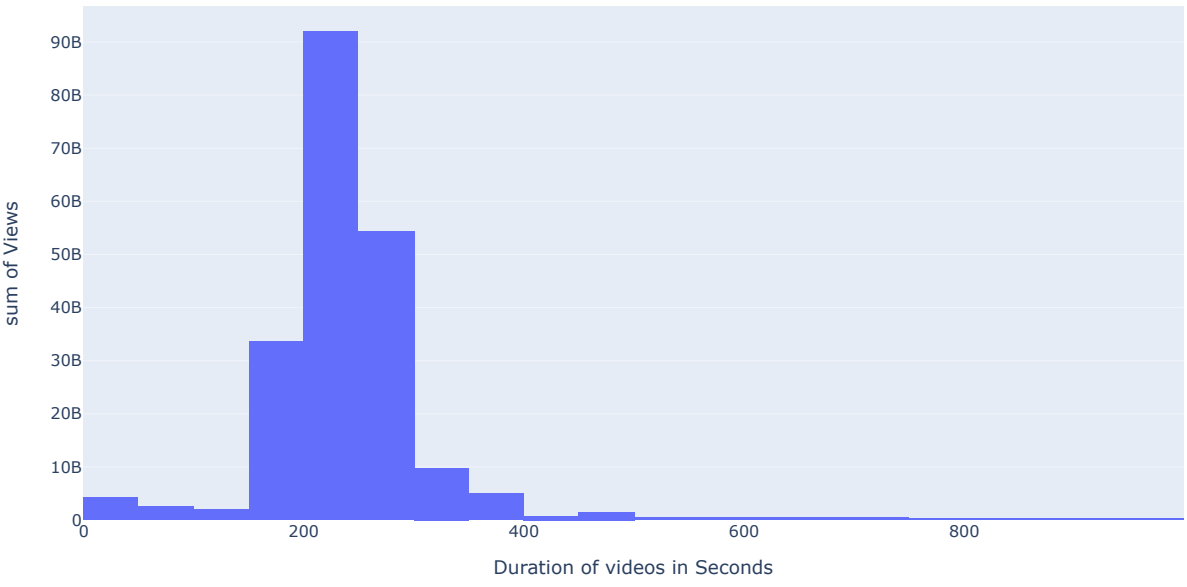
6) Most Commented Videos of each Channel

In [324]:
```
grouped_df = video_df.groupby(['channelTitle']).apply(lambda x: x.nlargest(1, 'commentCount'))
px.bar(grouped_df,
       x="title",
       y="commentCount",
       color="channelTitle",
       labels={"commentCount":"Comment Count","title":"Song Name","channelTitle" :"Channel Name"})
```



7) Does the video duration matter for views and interaction (likes/ comments)?

In [328]:
```
px.histogram(video_df[video_df['durationSecs'] < 1000],
             x="durationSecs",
             y="viewCount",nbins = 40,
             labels={"viewCount" : "Views","durationSecs":"Duration of videos in Seconds"})
```



8) Wordcloud for title

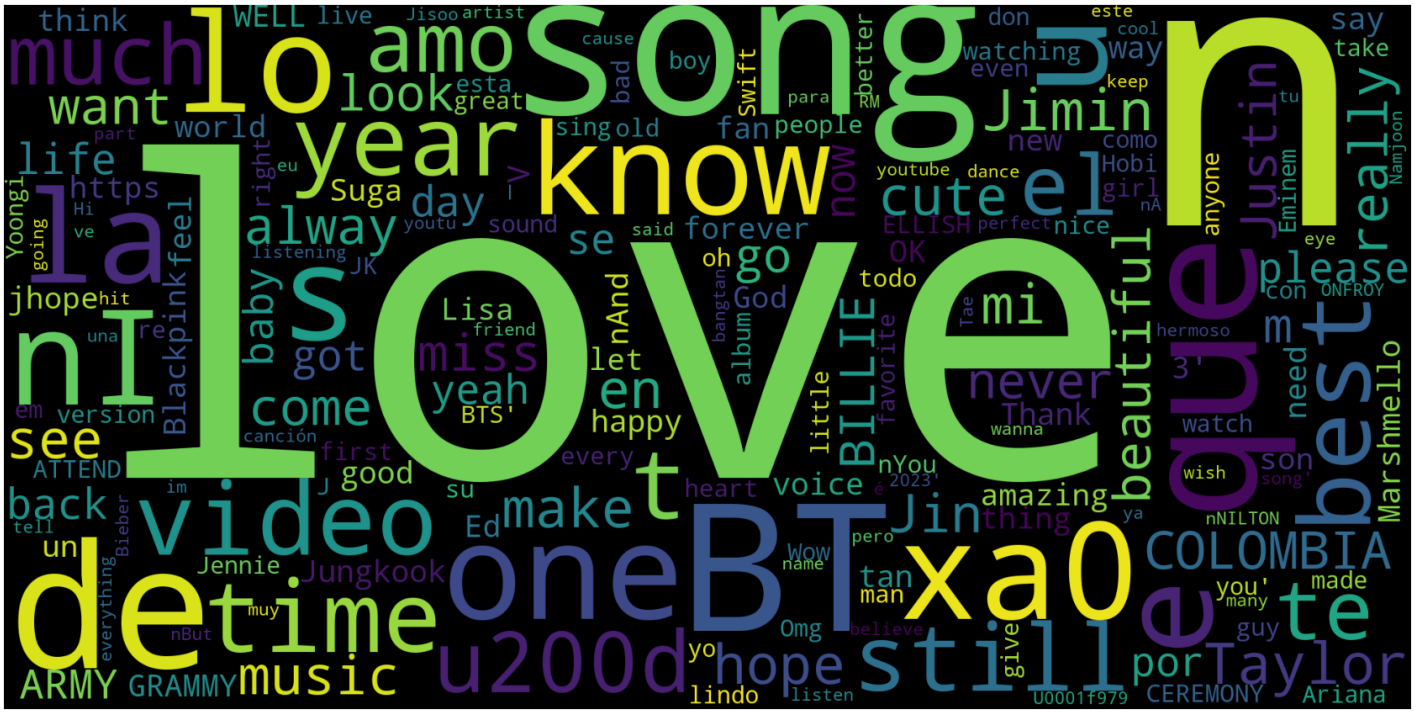
In [305]:
```
stop_words = set(stopwords.words('english'))
video_df['title_no_stopwords'] = video_df['title'].apply(lambda x: [item for item in str(x).split() if item not in stop_words])

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

In [319]:
```
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='white',
                      colormap='tab20b', collocations=False).generate(all_words_str)
plot_cloud(wordcloud)
```

9) Wordcloud for video comments

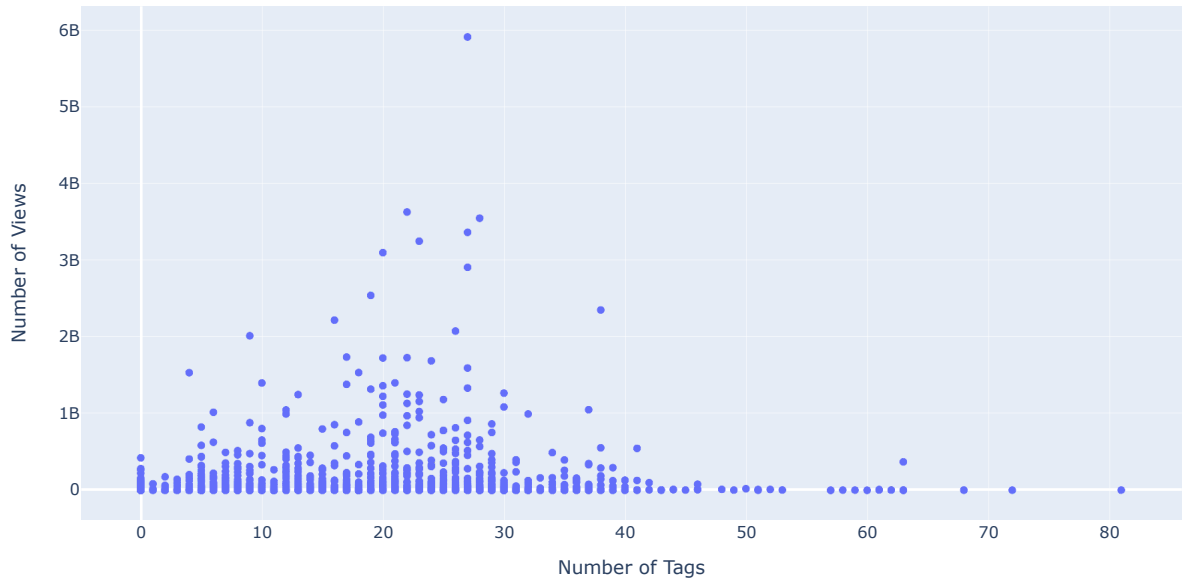
In [329]:
```
stop_words = set(stopwords.words('english'))
comments_df['comments_no_stopwords'] = comments_df['comments'].apply(lambda x: [item for item in str(x).split() if item not in stop_words])

all_words = list([a for b in comments_df['comments_no_stopwords'].tolist() for a in b])
all_words_str = ' '.join(all_words)
```

In [330]:
```
wordcloud = WordCloud(width = 2000, height = 1000, random_state=1, background_color='black',
                      colormap='viridis', collocations=False).generate(all_words_str)
plot_cloud(wordcloud)
```



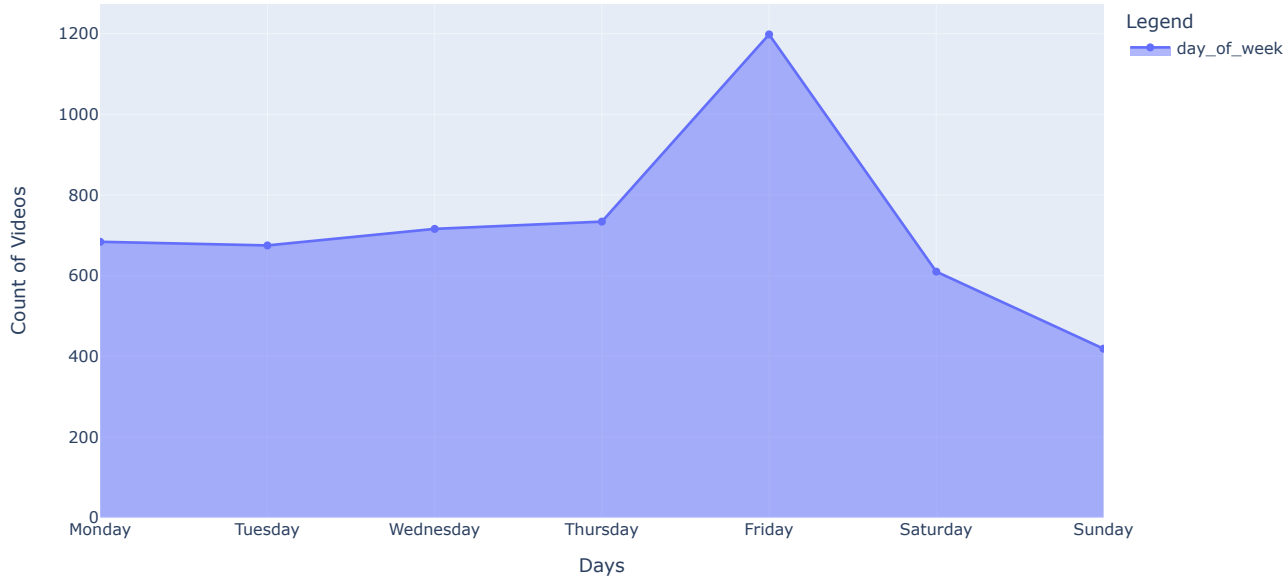
10) Does Number of Tags Affect Views?

In [335]:
```
px.scatter(video_df,
           x = "tagsCount",
           y = "viewCount",labels={"tagsCount":"Number of Tags","viewCount":"Number of Views"})
```



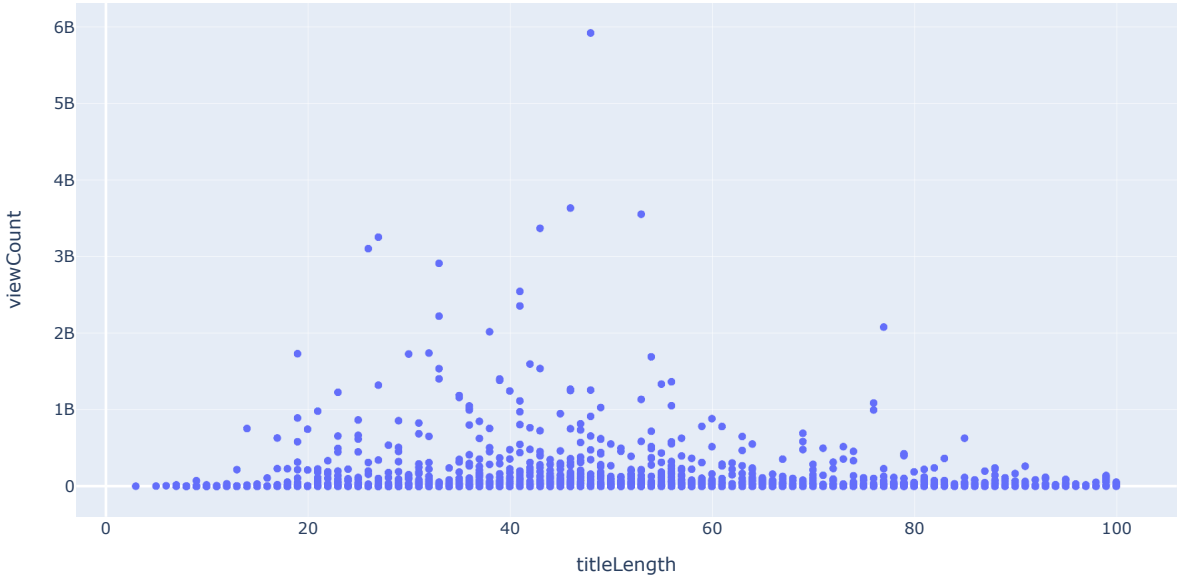
11) Which day in the week are most videos uploaded?

In [375]:
```
day_df = pd.DataFrame(video_df['day_of_week'].value_counts())
weekdays = [ 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
day_df = day_df.reindex(weekdays)
px.area(day_df,markers=True,
        labels={"value":"Count of Videos","index":"Days","variable":"Legend"})
```



12) Does title length matter for views?

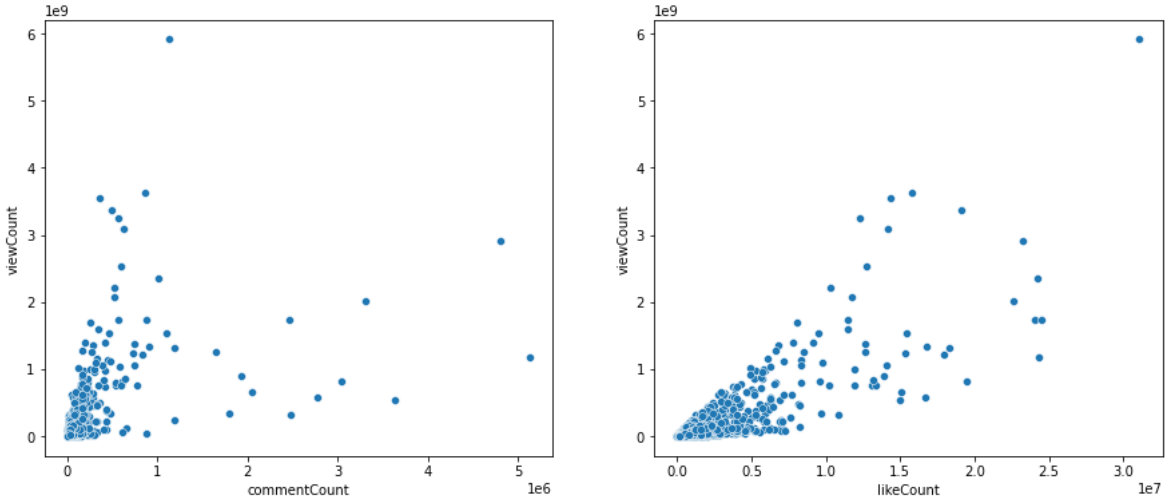
In [376]:
```
px.scatter(video_df, x = "titleLength", y = "viewCount")
```



13) Does the number of likes and comments matter for a video to get more views?

In [393]:
```
fig, ax =plt.subplots(1,2)
fig.set_size_inches(15,6)
sns.scatterplot(data = video_df, x = "commentCount", y = "viewCount", ax=ax[0])
sns.scatterplot(data = video_df, x = "likeCount", y = "viewCount", ax=ax[1])
```

Out[393]: <AxesSubplot:xlabel='likeCount', ylabel='viewCount'>



In [ ]:

In [ ]: