Merge the Dateset

In [1]:

```
import pandas as pd
import numpy as np
import zipfile

# Function to load CSV from a ZIP file with multiple files
def load_csv_from_zip(zip_path, csv_filename):
    with zipfile.ZipFile(zip_path, 'r') as z:
        # Extract and read the specific CSV file
        with z.open(csv_filename) as f:
            return pd.read_csv(f)

# Load datasets from zipped CSV files specifying the correct CSV filen
df_ca = load_csv_from_zip('Datasets/CAvideos.csv.zip', 'CAvideos.csv')
df_de = load_csv_from_zip('Datasets/DEvideos.csv.zip', 'DEvideos.csv')
df_fr = load_csv_from_zip('Datasets/FRvideos.csv.zip', 'FRvideos.csv')
df_gb = load_csv_from_zip('Datasets/GBvideos.csv.zip', 'GBvideos.csv')
df_us = load_csv_from_zip('Datasets/USvideos.csv.zip', 'USvideos.csv')

# Add a new column 'location' to each DataFrame
df_ca['location'] = 'China'
df_de['location'] = 'Germany'
df_fr['location'] = 'France'
df_gb['location'] = 'Great Britain'
df_us['location'] = 'USA'

# Concatenate all DataFrames
merged_df = pd.concat([df_ca, df_de, df_fr, df_gb, df_us], ignore_inde

# Check the first few rows of the merged DataFrame
print(merged_df.head())
```

```
      video_id trending_date  \
0  n1WpP7iowLc      17.14.11   
1  0dBIkQ4Mz1M      17.14.11   
2  5qpjK5DgCt4      17.14.11   
3  d380meD0W0M      17.14.11   
4  2Vv-BfVoq4g      17.14.11   

                                            title channel_title  \
0         Eminem - Walk On Water (Audio) ft. Beyoncé    EminemVEVO   
1                      PLUSH - Bad Unboxing Fan Mail     iDubbbzTV   
2  Racist Superman | Rudy Mancuso, King Bach & Le...  Rudy Mancuso   
3                           I Dare You: GOING BALD!?      nigahiga   
4        Ed Sheeran - Perfect (Official Music Video)    Ed Sheeran   
```

```
   category_id              publish_time  \
0           10  2017-11-10T17:00:03.000Z   
1           23  2017-11-13T17:00:00.000Z   
2           23  2017-11-12T19:05:24.000Z   
3           24  2017-11-12T18:01:41.000Z   
4           10  2017-11-09T11:04:14.000Z   

                                                tags     views    lik
es  \
0  Eminem|"Walk"|"On"|"Water"|"Aftermath/Shady/In...  17158579   7874
25   
1  plush|"bad unboxing"|"unboxing"|"fan mail"|"id...   1014651   1277
94   
2  racist superman|"rudy"|"mancuso"|"king"|"bach"...   3191434   1460
35   
3  ryan|"higa"|"higatv"|"nigahiga"|"i dare you"|"...   2095828   1322
39   
4  edsheeran|"ed sheeran"|"acoustic"|"live"|"cove...  33523622  16341
30   

   dislikes  comment_count                                  thumbnail
_link  \
0     43420         125882  https://i.ytimg.com/vi/n1WpP7iowLc/defaul
t.jpg (https://i.ytimg.com/vi/n1WpP7iowLc/default.jpg)   
1      1688          13030  https://i.ytimg.com/vi/0dBIkQ4Mz1M/defaul
t.jpg (https://i.ytimg.com/vi/0dBIkQ4Mz1M/default.jpg)   
2      5339           8181  https://i.ytimg.com/vi/5qpjK5DgCt4/defaul
t.jpg (https://i.ytimg.com/vi/5qpjK5DgCt4/default.jpg)   
3      1989          17518  https://i.ytimg.com/vi/d380meD0W0M/defaul
t.jpg (https://i.ytimg.com/vi/d380meD0W0M/default.jpg)   
4     21082          85067  https://i.ytimg.com/vi/2Vv-BfVoq4g/defaul
t.jpg (https://i.ytimg.com/vi/2Vv-BfVoq4g/default.jpg)   

   comments_disabled  ratings_disabled  video_error_or_removed  \
0              False             False                   False   
1              False             False                   False   
2              False             False                   False   
3              False             False                   False   
4              False             False                   False   

                                         description location  
0  Eminem's new track Walk on Water ft. Beyoncé i...    China  
1  STill got a lot of packages. Probably will las...    China  
2  WATCH MY PREVIOUS VIDEO ▶ \n\nSUBSCRIBE ► http...    China  
3  I know it's been a while since we did this sho...    China  
4  🎧: https://ad.gt/yt-perfect\n💰: (https://ad.gt/yt-perfect\n💰:)
https://atlant... (https://atlant...)    China  
```

In [2]:

```
# drop missing values
merged_df1 = merged_df.dropna()
```

In [3]:

```
!pip install nltk
```

```
Requirement already satisfied: nltk in /Users/yujiacao/anaconda3/lib/
python3.11/site-packages (3.8.1)
Requirement already satisfied: click in /Users/yujiacao/anaconda3/li
b/python3.11/site-packages (from nltk) (8.0.4)
Requirement already satisfied: joblib in /Users/yujiacao/anaconda3/li
b/python3.11/site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in /Users/yujiacao/ana
conda3/lib/python3.11/site-packages (from nltk) (2022.7.9)
Requirement already satisfied: tqdm in /Users/yujiacao/anaconda3/lib/
python3.11/site-packages (from nltk) (4.65.0)
```

In [4]:

```
import nltk
nltk.download('stopwords')
from nltk.corpus import stopwords
import re

# Get the list of default English stopwords
stop_words = set(stopwords.words('english'))
stop_words = set(stopwords.words('chinese'))
stop_words = set(stopwords.words('french'))
stop_words = set(stopwords.words('german'))

# Function to remove stopwords and clean text
def clean_text(text):
    # Lowercase the text
    text = text.lower()

    # Remove non-alphabetical characters (retain only letters and spac
    text = re.sub(r'[^a-z\s]', '', text)

    # Split text into words
    words = text.split()

    # Remove stopwords
    remove_stopwords = [word for word in words if word not in stop_wor

    # Join the cleaned words back into a string
    new_text = ' '.join(remove_stopwords)

    return new_text
    data = {'title','description','text'}

# Apply the clean_text function to the 'title' column in merged_df1
```

```
merged_df1['new_text'] = merged_df1['title'].apply(clean_text)

# Display the cleaned DataFrame
print(merged_df1)
```

```
192957
202309      call of duty|"cod"|"activision"|"Black Ops 4"  10306119
357079

        dislikes  comment_count  \
0          43420         125882
1           1688          13030
2           5339           8181
3           1989          17518
4          21082          85067
...          ...            ...
202304      4052          62610
202305      1385           2657
202307      1032           3992
202308      2846          13088
202309    212976         144795

                                        thumbnail_link  comments_disa
bled  \
0       https://i.ytimg.com/vi/n1WpP7iowLc/default.jpg (https://i.yti
```

In [5]:

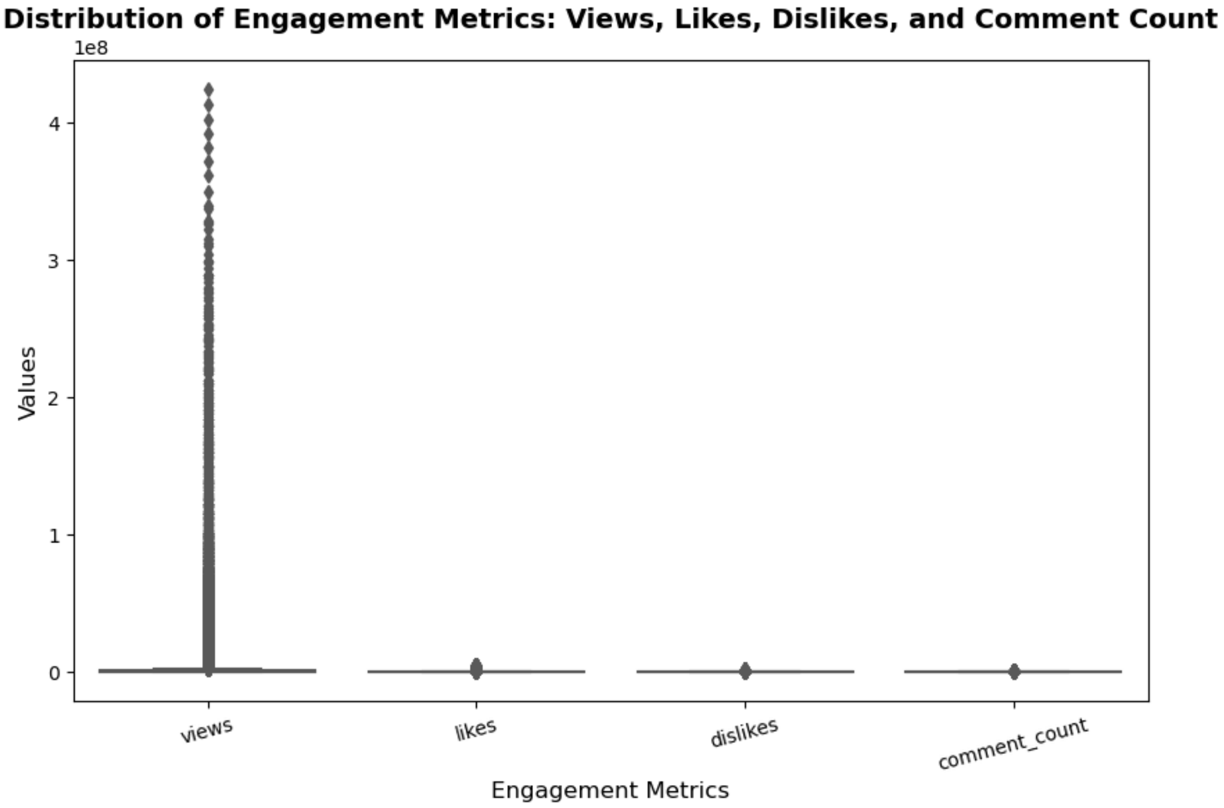
```
#drop columns needed
merged_df1.drop(columns=['thumbnail_link', 'video_id','comments_disabl

print(merged_df1.head())
```

```
  trending_date                                              title  \
0      17.14.11         Eminem - Walk On Water (Audio) ft. Beyoncé
1      17.14.11                      PLUSH - Bad Unboxing Fan Mail
2      17.14.11  Racist Superman | Rudy Mancuso, King Bach & Le...
3      17.14.11                           I Dare You: GOING BALD!?
4      17.14.11        Ed Sheeran - Perfect (Official Music Video)

  channel_title  category_id              publish_time  \
0    EminemVEVO           10  2017-11-10T17:00:03.000Z
1     iDubbbzTV           23  2017-11-13T17:00:00.000Z
2  Rudy Mancuso           23  2017-11-12T19:05:24.000Z
3      nigahiga           24  2017-11-12T18:01:41.000Z
4    Ed Sheeran           10  2017-11-09T11:04:14.000Z

                                               tags     views    lik
es  \
0  Eminem|"Walk"|"On"|"Water"|"Aftermath/Shady/In...  17158579   7874
25
1  plush|"bad unboxing"|"unboxing"|"fan mail"|"id...   1014651   1277
94
```

```
2  racist superman|"rudy"|"mancuso"|"king"|"bach"...   3191434   1460
35
3  ryan|"higa"|"higatv"|"nigahiga"|"i dare you"|"...   2095828   1322
39
4  edsheeran|"ed sheeran"|"acoustic"|"live"|"cove...  33523622  16341
30

   dislikes  comment_count                                           des
cription  \
0     43420         125882  Eminem's new track Walk on Water ft. Beyo
ncé i...
1      1688          13030  STill got a lot of packages. Probably wil
l las...
2      5339           8181  WATCH MY PREVIOUS VIDEO ▶ \n\nSUBSCRIBE ►
http...
3      1989          17518  I know it's been a while since we did thi
s sho...
4     21082          85067  🎧: https://ad.gt/yt-perfect\n💰: (http
s://ad.gt/yt-perfect\n💰:) https://atlant... (https://atlant...)

  location                                         new_text
0    China             eminem walk on water audio ft beyonc
1    China                        plush bad unboxing fan mail
2    China  racist superman rudy mancuso king bach lele pons
3    China                              i dare you going bald
4    China           ed sheeran perfect official music video

/var/folders/6z/mn847gls7x5fvn9pl3c9lfmw0000gn/T/ipykernel_9244/19438
13935.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: https://pandas.pydata.org/panda
s-docs/stable/user_guide/indexing.html#returning-a-view-versus-a-copy
(https://pandas.pydata.org/pandas-docs/stable/user_guide/indexing.htm
l#returning-a-view-versus-a-copy)
  merged_df1.drop(columns=['thumbnail_link', 'video_id','comments_dis
abled','ratings_disabled','video_error_or_removed'], inplace=True)
```

In [10]:

```
# outlier treatment part 1
import seaborn as sns
import matplotlib.pyplot as plt

# Create the boxplot with enhanced aesthetics
plt.figure(figsize=(10, 6))  # Adjust figure size for better clarity
sns.boxplot(data=merged_df[['views', 'likes', 'dislikes', 'comment_cou

# Add a title and labels to make the plot more informative
plt.title('Distribution of Engagement Metrics: Views, Likes, Dislikes,
plt.xlabel('Engagement Metrics', fontsize=12)
plt.ylabel('Values', fontsize=12)

# Rotate x-axis labels for better readability
plt.xticks(rotation=15)

# Display the plot
plt.show()
```



In [11]:

```
from sklearn.model_selection import train_test_split

X = merged_df.drop(columns=['views'])  # Drop 'views' from features to
y = merged_df['views']
# Assuming you have a dataset with features X and target y
X_train, X_test, y_train, y_test = train_test_split(X,y, test_size=0.2

train = pd.DataFrame(X_train)
train['views'] = y_train.values

test = pd.DataFrame(X_test)
test['views'] = y_test.values
```

In [12]:

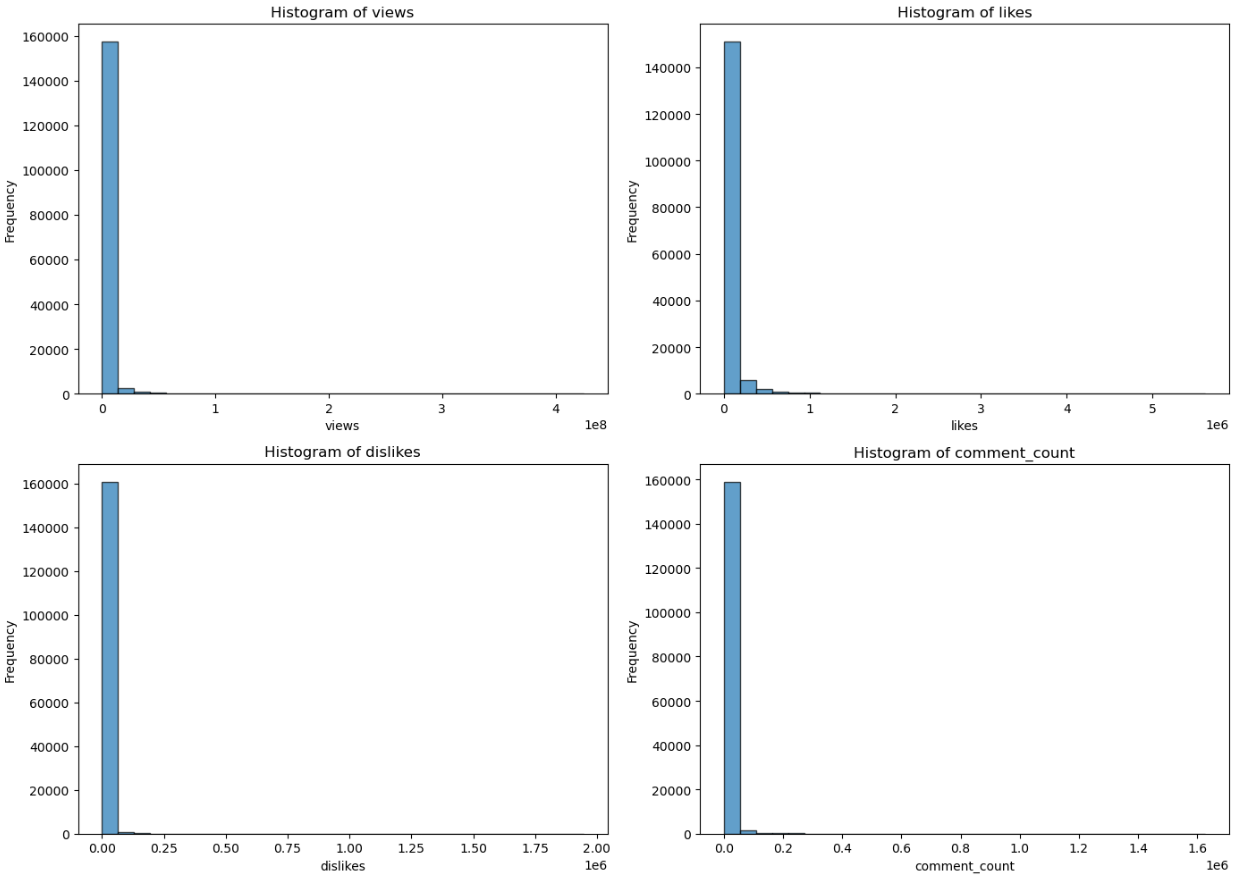
```
# Check the data types of each column
print(train.dtypes)
```

```
video_id                   object
trending_date              object
title                      object
channel_title              object
category_id                 int64
publish_time               object
tags                       object
likes                       int64
dislikes                    int64
comment_count               int64
thumbnail_link             object
comments_disabled            bool
ratings_disabled             bool
video_error_or_removed       bool
description                object
location                   object
views                       int64
dtype: object
```

Exploration of Data Analysis(EDA) for Numerical Variables

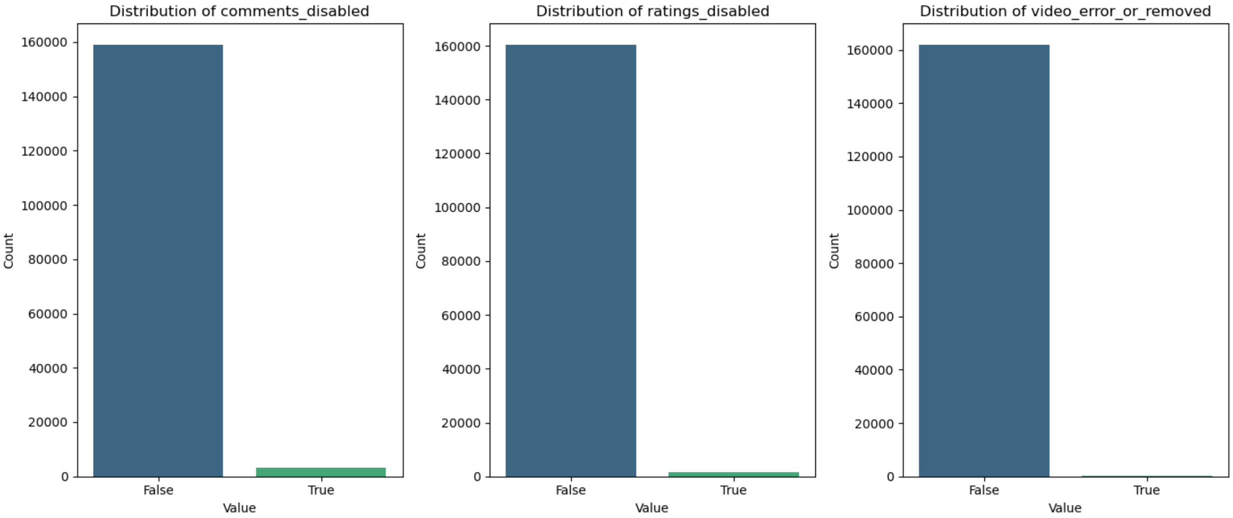
In [13]:

```
#data exploration for numerical columns
import matplotlib.pyplot as plt

# Define numerical columns
numerical_columns = ['views', 'likes', 'dislikes', 'comment_count']

# Create histograms for each numerical column
plt.figure(figsize=(14, 10))
for i, column in enumerate(numerical_columns, 1):
    plt.subplot(2, 2, i)
    plt.hist(train[column], bins=30, alpha=0.7, edgecolor='black')
    plt.title(f'Histogram of {column}')
    plt.xlabel(column)
    plt.ylabel('Frequency')

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



Exploration of Data Analysis(EDA) for Boolean Variables

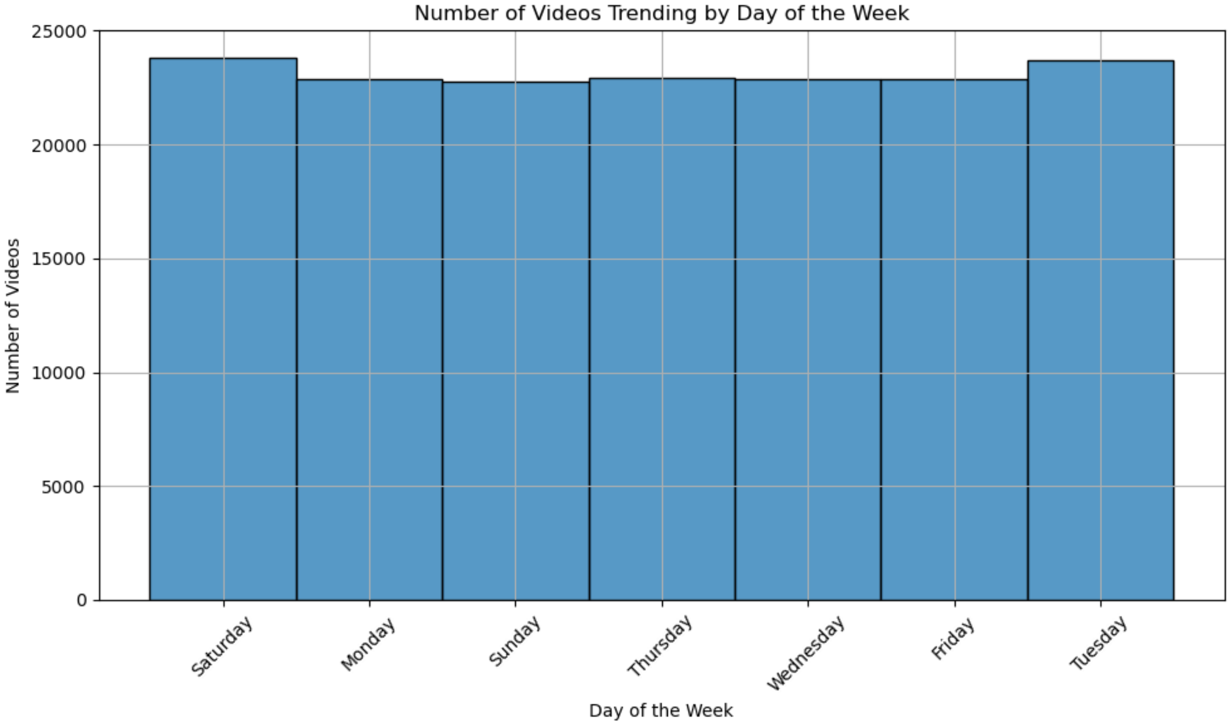
In [14]:
```
import seaborn as sns

# Define boolean columns
boolean_columns = ['comments_disabled', 'ratings_disabled', 'video_err

# Plot bar plots for each boolean column
plt.figure(figsize=(14, 6))
for i, column in enumerate(boolean_columns, 1):
    plt.subplot(1, 3, i)
    # Count the occurrences of each boolean value
    counts = train[column].value_counts()
    # Plot bar plot
    sns.barplot(x=counts.index, y=counts.values, palette='viridis')
    plt.title(f'Distribution of {column}')
    plt.xlabel('Value')
    plt.ylabel('Count')

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



Exploration of Data Analysis(EDA) for Date-Time Variables

In [15]:
```
# convert the trending_date to datetime type
train['trending_date'] = pd.to_datetime(train['trending_date'], format
# Extract day of the week from 'trending_date'
train['trending_day_of_week'] = train['trending_date'].dt.day_name()

# Plot histogram of trending day of the week
plt.figure(figsize=(10, 6))
sns.histplot(train['trending_day_of_week'], discrete=True, palette='vi
plt.title('Number of Videos Trending by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Videos')
plt.xticks(rotation=45)  # Rotate x-axis labels for better readability
plt.grid(True)
plt.tight_layout()
plt.show()
```

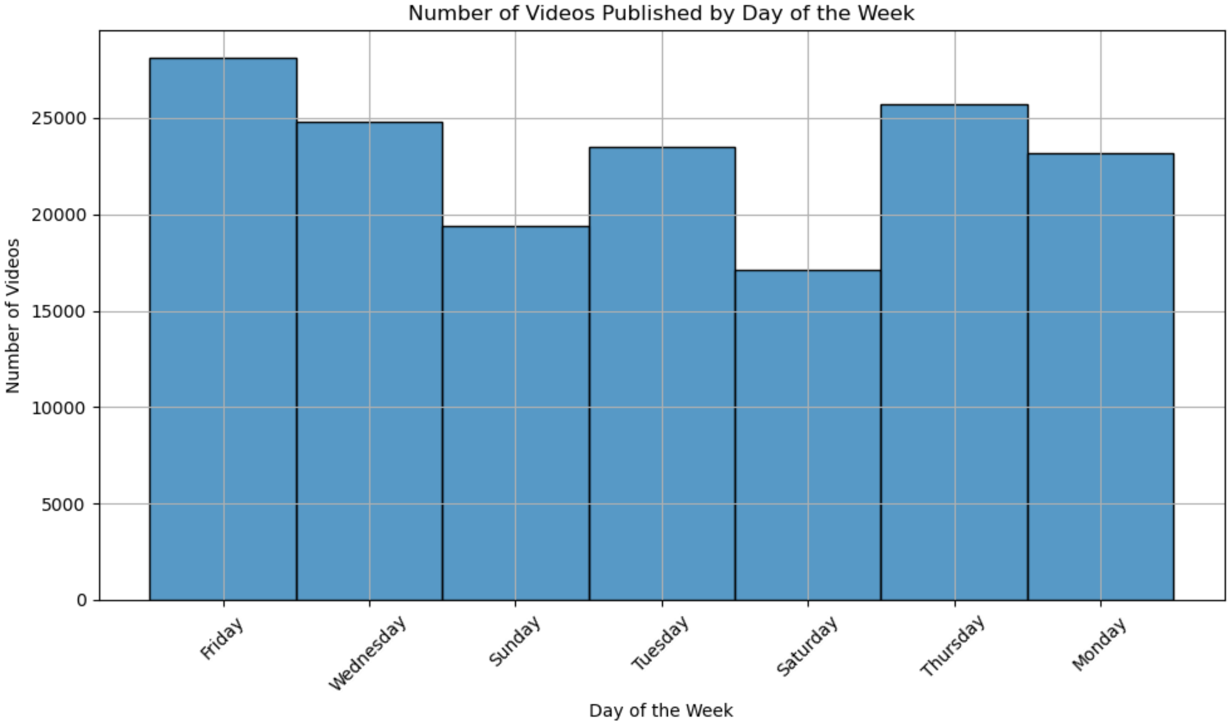
```
/var/folders/6z/mn847gls7x5fvn9pl3c9lfmw0000gn/T/ipykernel_9244/36707
67.py:8: UserWarning: Ignoring `palette` because no `hue` variable ha
s been assigned.
  sns.histplot(train['trending_day_of_week'], discrete=True, palette
='viridis')
```



In [16]:

```
#convert the publish_date to datetime type
train['publish_time'] = pd.to_datetime(train['publish_time'], format='
# Extract day of the week from 'publish_time'
train['day_of_week'] = train['publish_time'].dt.day_name()

# Plot histogram of day of the week
plt.figure(figsize=(10, 6))
sns.histplot(train['day_of_week'], discrete=True, palette='viridis')
plt.title('Number of Videos Published by Day of the Week')
plt.xlabel('Day of the Week')
plt.ylabel('Number of Videos')
plt.xticks(rotation=45)
plt.grid(True)
plt.tight_layout()
plt.show()
```

```
/var/folders/6z/mn847gls7x5fvn9pl3c9lfmw0000gn/T/ipykernel_9244/22089
40814.py:8: UserWarning: Ignoring `palette` because no `hue` variable
has been assigned.
  sns.histplot(train['day_of_week'], discrete=True, palette='viridi
s')
```



Statistical Description

In [17]:
```
numerical_description = train.describe()
print(numerical_description)
```

```
         category_id         likes      dislikes  comment_count  
views
count  161848.000000  1.618480e+05  1.618480e+05   1.618480e+05  1.61
8480e+05
mean       19.710395  5.702207e+04  3.038615e+03   6.177708e+03  2.05
0362e+06
std         7.365759  2.090197e+05  2.780134e+04   3.111470e+04  9.35
9045e+06
min         1.000000  0.000000e+00  0.000000e+00   0.000000e+00  2.23
0000e+02
25%        17.000000  1.445000e+03  6.700000e+01   2.090000e+02  7.51
2375e+04
50%        23.000000  7.591000e+03  2.890000e+02   9.190000e+02  3.08
3285e+05
75%        24.000000  3.221750e+04  1.150000e+03   3.522000e+03  1.10
2676e+06
max        44.000000  5.613827e+06  1.944971e+06   1.626501e+06  4.24
5389e+08
```

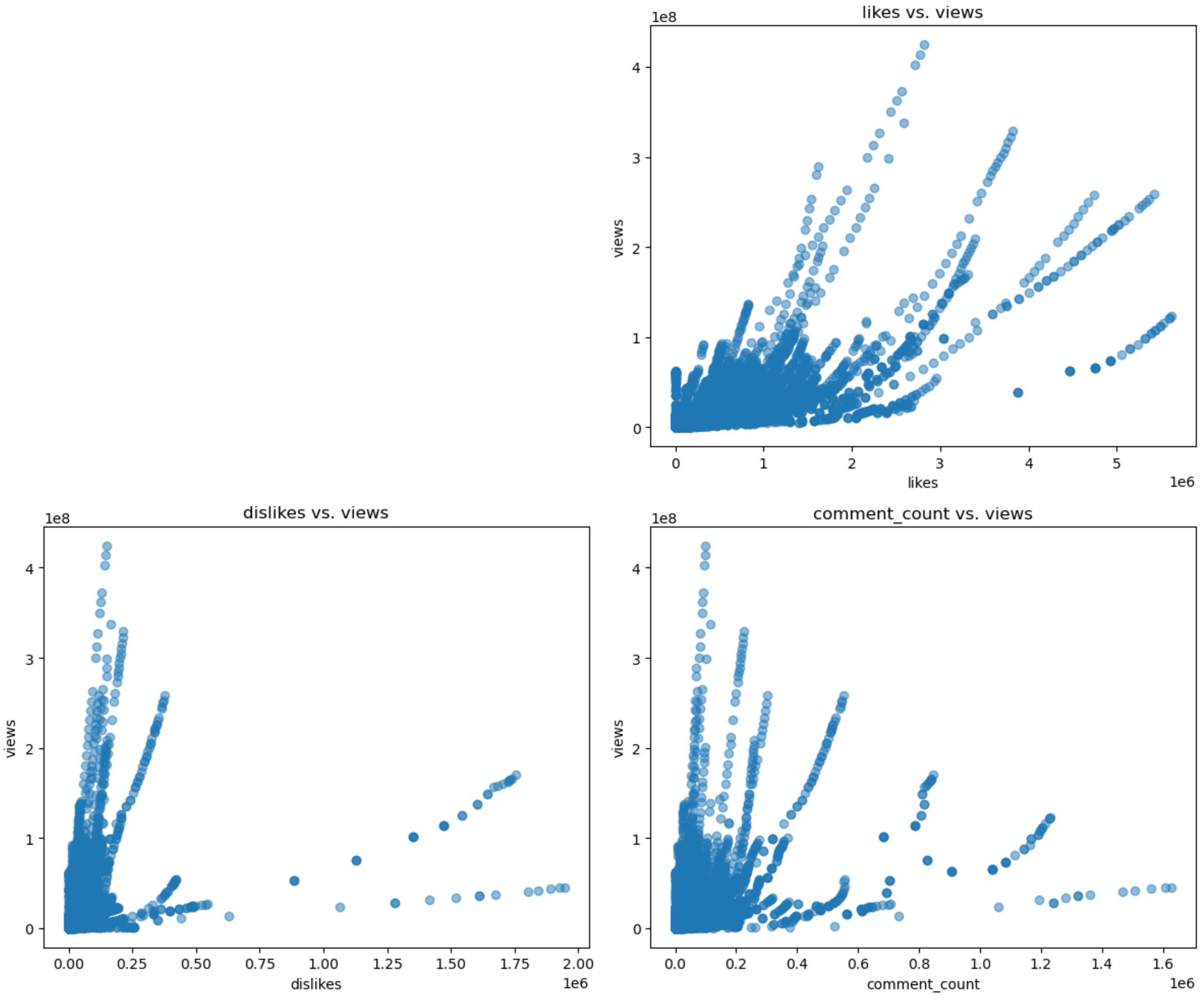
In [18]:
```
# Statistical description of categorical columns
categorical_description = train[['category_id', 'location']].describe(
print(categorical_description)
```

```
         category_id
count  161848.000000
mean       19.710395
std         7.365759
min         1.000000
25%        17.000000
50%        23.000000
75%        24.000000
max        44.000000
```

Exploration of Data Analysis for Numerical Values

In [19]:

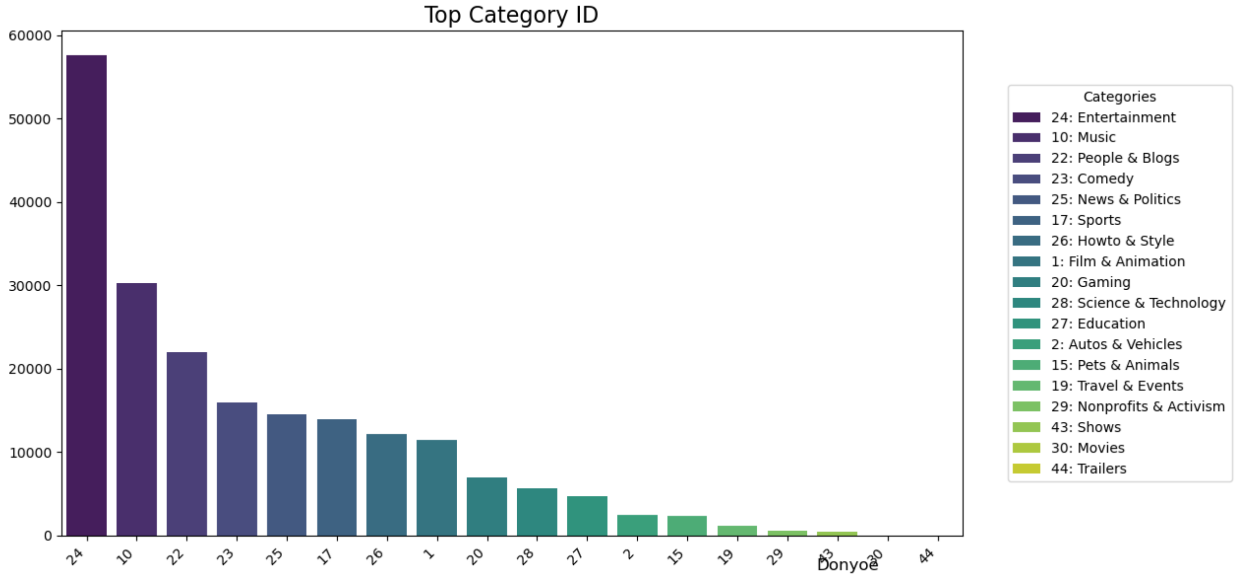
```
# Scatter plots for each numerical column vs. 'views'
plt.figure(figsize=(12, 10))
for i, column in enumerate(numerical_columns, 1):
    if column != 'views':
        plt.subplot(2, 2, i)
        plt.scatter(train[column], train['views'], alpha=0.5)
        plt.title(f'{column} vs. views')
        plt.xlabel(column)
        plt.ylabel('views')

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



In [20]:

```
tplotlib.pyplot as plt
aborn as sns
ndas as pd

/ 'category_id' and count occurrences
counts = merged_df.groupby('category_id').size().reset_index(name='N')
```

```
 'N' in descending order
counts = category_counts.sort_values(by='N', ascending=False)
counts['category_id'] = pd.Categorical(category_counts['category_id'],

a dictionary to map 'category_id' to descriptive names
names = {
: Film & Animation",
: Autos & Vehicles",
10: Music",
15: Pets & Animals",
17: Sports",
18: Short Movies",
19: Travel & Events",
20: Gaming",
21: Videoblogging",
22: People & Blogs",
23: Comedy",
24: Entertainment",
25: News & Politics",
26: Howto & Style",
27: Education",
28: Science & Technology",
29: Nonprofits & Activism",
30: Movies",
31: Anime/Animation",
32: Action/Adventure",
33: Classics",
34: Comedy",
35: Documentary",
36: Drama",
37: Family",
38: Foreign",
39: Horror",
40: Sci-Fi/Fantasy",
41: Thriller",
42: Shorts",
43: Shows",
44: Trailers"


tegory_id' to names in the 'category_counts' DataFrame
counts['category_name'] = category_counts['category_id'].map(category_r

ing seaborn
e(figsize=(10, 6))
 sns.barplot(data=category_counts, x='category_id', y='N', palette='vi

ze the plot to match your ggplot2 example
("Top Category ID", fontsize=16)
l(None)
l(None)
s(rotation=45, ha='right')
_layout()
xt(0.9, 0.02, "Donyoe", horizontalalignment='right', fontsize=12)

ustom legend for category names on the side
 barplot.patches
bels = [category_names[int(c)] for c in category_counts['category_id']]

n the legend on the right of the plot using 'bbox_to_anchor'
d(handles=handles[:len(legend_labels)], labels=legend_labels, title='Ca
  bbox_to_anchor=(1.05, 0.5), loc='center left', borderaxespad=0)

)
```



Creating Engagement Metrics

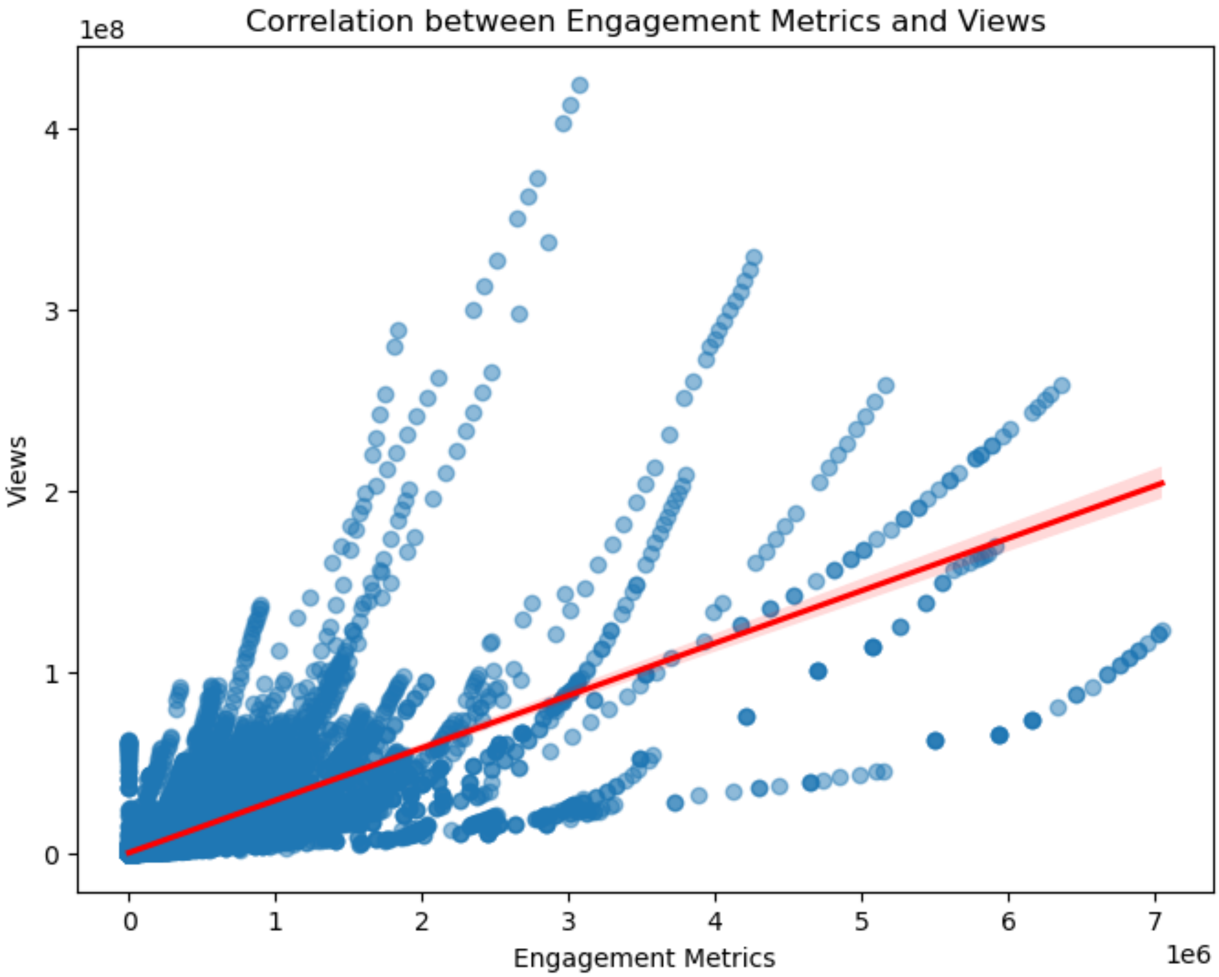
In [21]:

```
# Create a new column
train['Engagement Metrics'] = train['likes'] + train['dislikes'] + tra
# Display the DataFrame to check the new column
print(train[['likes', 'dislikes', 'comment_count', 'Engagement Metrics
```

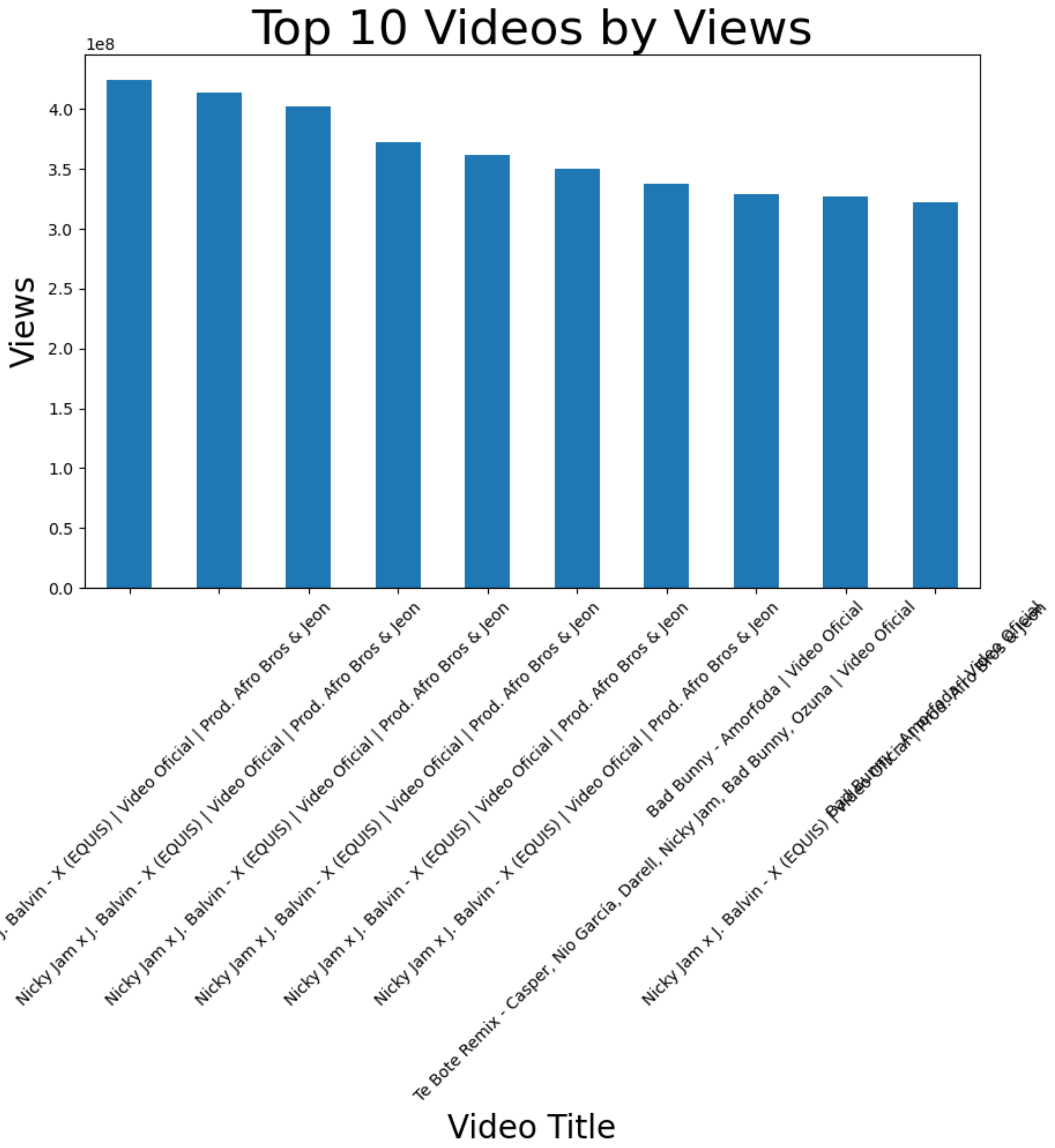
```
         likes  dislikes  comment_count  Engagement Metrics
50252      319        15             63                 397
15943     3621      1735           1967                7323
162168    4168       141            266                4575
110741     334        77            138                 549
142650  136181      1980          10259              148420
```

In [22]:

```
# Create a scatter plot with a regression line
plt.figure(figsize=(8, 6))
sns.regplot(x='Engagement Metrics', y='views', data=train, scatter_kws
plt.title('Correlation between Engagement Metrics and Views')
plt.xlabel('Engagement Metrics')
plt.ylabel('Views')
plt.show()
```



In [23]:
```
top_videos = train.nlargest(10, 'views')[['title', 'views']]

# To plot the chart

top_videos.set_index('title')['views'].plot(kind='bar', figsize=(10, 6
plt.xlabel('Video Title',fontsize=20)
plt.ylabel('Views',fontsize=20)
plt.title('Top 10 Videos by Views',fontsize=30)
plt.xticks(rotation=45)
plt.show()
```



In [24]:
```
# what about top 50?
## Display engagement metrics for top 50 videos
```

```
top_50_videos = train.nlargest(50, 'views')
print(top_50_videos[['title', 'Engagement Metrics','location']])
```

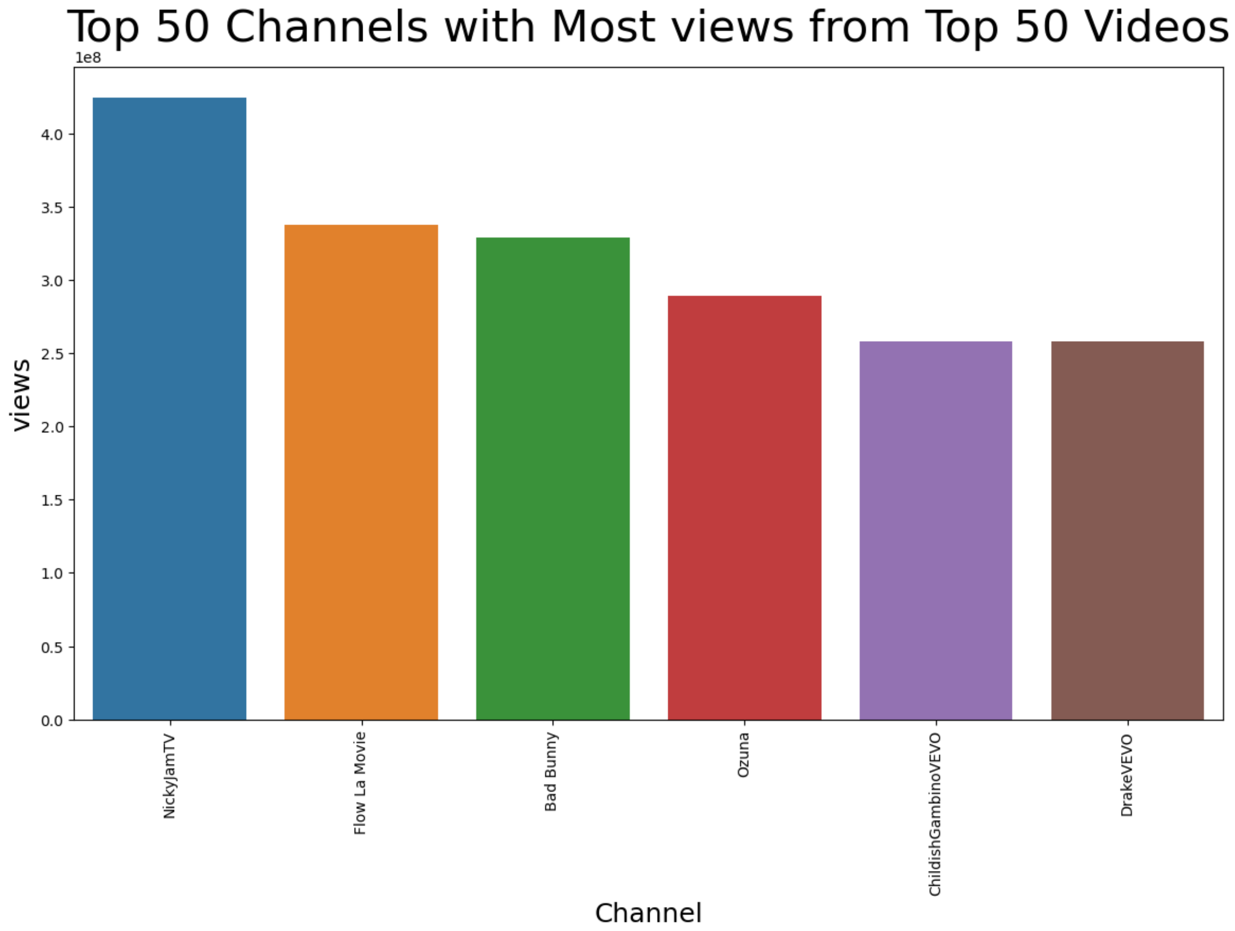
```
                                                    title  Engagement 
Metrics  \
150857  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
3067426   
150657  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
3011515   
150453  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2956724   
149869  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2786627   
149686  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2723032   
149497  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2650114   
156905  Te Bote Remix - Casper, Nio García, Darell, Ni...   
2862074   
147990               Bad Bunny - Amorfoda | Video Oficial   
4264625   
149116  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2505131   
147786               Bad Bunny - Amorfoda | Video Oficial   
4231351   
147582               Bad Bunny - Amorfoda | Video Oficial   
4198350   
148922  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2427694   
147380               Bad Bunny - Amorfoda | Video Oficial   
4167420   
147183               Bad Bunny - Amorfoda | Video Oficial   
4135956   
146985               Bad Bunny - Amorfoda | Video Oficial   
4103146   
148725  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...   
2350490   
156174  Te Bote Remix - Casper, Nio García, Darell, Ni...   
2661680   
146784               Bad Bunny - Amorfoda | Video Oficial   
4062651   
146582               Bad Bunny - Amorfoda | Video Oficial   
4026487   
143607           Ozuna x Romeo Santos - El Farsante Remix   
1836833   
146383               Bad Bunny - Amorfoda | Video Oficial   
3996243   
143402           Ozuna x Romeo Santos - El Farsante Remix   
1815236   
146173               Bad Bunny - Amorfoda | Video Oficial   
3966424
```

```
3966424
145973                Bad Bunny - Amorfoda | Video Oficial
3930504
155715  Te Bote Remix - Casper, Nio García, Darell, Ni...
2474011
148133  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...
2107200
145567                Bad Bunny - Amorfoda | Video Oficial
3849549
160680  Childish Gambino - This Is America (Official V...
6356524
148381                                  Drake - God’s Plan
5156827
155551  Te Bote Remix - Casper, Nio García, Darell, Ni...
2412367
160501  Childish Gambino - This Is America (Official V...
6286180
142798            Ozuna x Romeo Santos - El Farsante Remix
1741314
147927  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...
2031387
145371                Bad Bunny - Amorfoda | Video Oficial
3791325
160324  Childish Gambino - This Is America (Official V...
6243463
148186                                  Drake - God’s Plan
5089683
160150  Childish Gambino - This Is America (Official V...
6193738
155384  Te Bote Remix - Casper, Nio García, Darell, Ni...
2352426
159974  Childish Gambino - This Is America (Official V...
6156360
142596            Ozuna x Romeo Santos - El Farsante Remix
1711546
147979                                  Drake - God’s Plan
5024782
147722  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...
1960805
147775                                  Drake - God’s Plan
4962917
159634  Childish Gambino - This Is America (Official V...
6015384
155218  Te Bote Remix - Casper, Nio García, Darell, Ni...
2292496
144956                Bad Bunny - Amorfoda | Video Oficial
3682433
147520  Nicky Jam x J. Balvin - X (EQUIS) | Video Ofic...
1891126
159470  Childish Gambino - This Is America (Official V...
5955876
```

```
5955876
142381           Ozuna x Romeo Santos – El Farsante Remix
1682801
147571                                 Drake – God’s Plan
4901873


             location
150857  Great Britain
150657  Great Britain
150453  Great Britain
149869  Great Britain
149686  Great Britain
149497  Great Britain
156905  Great Britain
147990  Great Britain
149116  Great Britain
147786  Great Britain
147582  Great Britain
148922  Great Britain
147380  Great Britain
147183  Great Britain
146985  Great Britain
148725  Great Britain
156174  Great Britain
146784  Great Britain
146582  Great Britain
143607  Great Britain
146383  Great Britain
143402  Great Britain
146173  Great Britain
145973  Great Britain
155715  Great Britain
148133  Great Britain
145567  Great Britain
160680  Great Britain
148381  Great Britain
155551  Great Britain
160501  Great Britain
142798  Great Britain
147927  Great Britain
145371  Great Britain
160324  Great Britain
148186  Great Britain
160150  Great Britain
155384  Great Britain
159974  Great Britain
142596  Great Britain
147979  Great Britain
147722  Great Britain
147775  Great Britain
150634  Great Britain
```

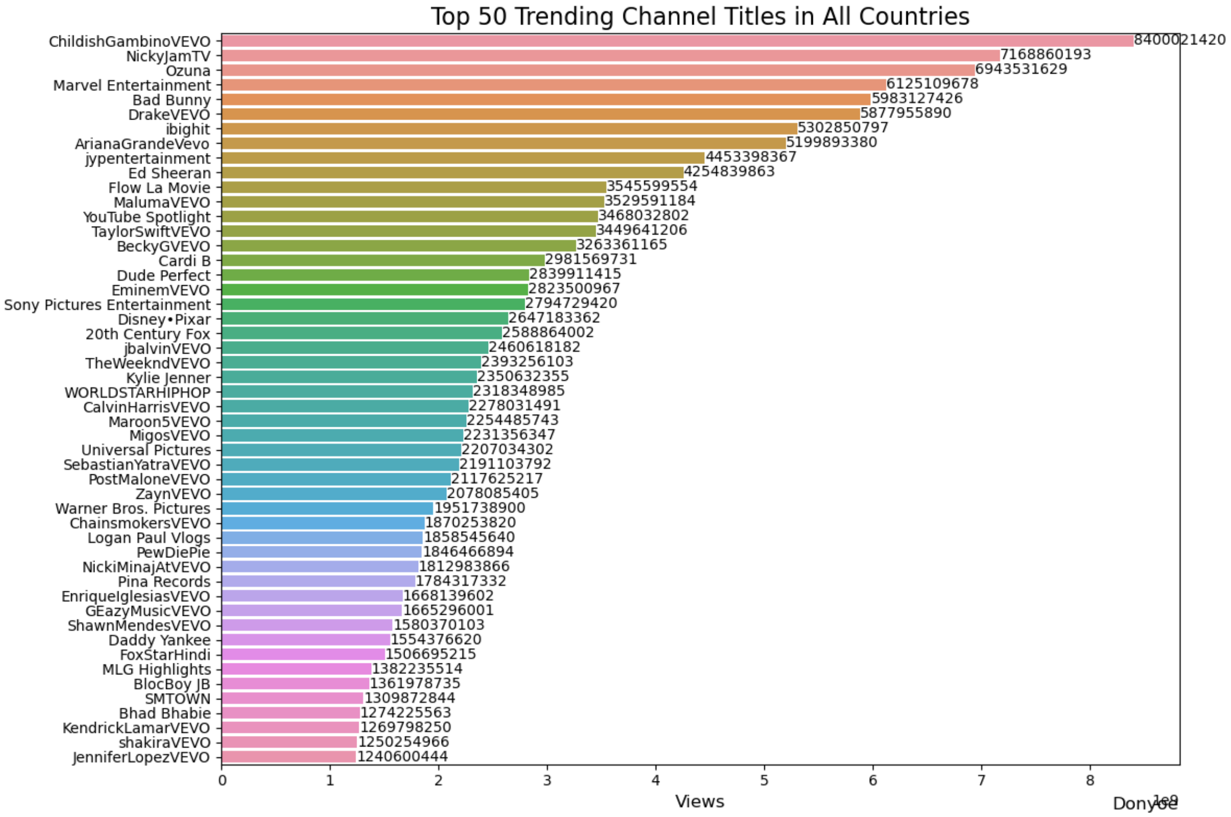
```
159634  Great Britain
155218  Great Britain
144956  Great Britain
147520  Great Britain
159470  Great Britain
142381  Great Britain

147571  Great Britain
```

In [25]:
```
import seaborn as snb
content = top_50_videos.groupby('channel_title')['views'].max()

# Sort values to get the top 50 channels with the most views
content = content.sort_values(ascending=False).head(50)
content = content.reset_index()  # Convert index to column

# Plotting the results
plt.figure(figsize=(14, 8))
snb.barplot(x='channel_title', y='views', data=content)
plt.title('Top 50 Channels with Most views from Top 50 Videos', fontsi
plt.ylabel('views', fontsize=18)
plt.xlabel('Channel', fontsize=18)
plt.xticks(rotation=90)
plt.show()
```



In [26]:
```
channel_counts = train.groupby('channel_title')['views'].sum().reset_i

# Sort values and select top 10 channels
top_10_channels = channel_counts.sort_values(by='views', ascending=Fal

# Plot using seaborn
```

```
plt.figure(figsize=(12, 8))
ax = sns.barplot(x='views', y='channel_title', data=top_10_channels,or

# Add labels
for index, value in enumerate(top_10_channels['views']):
    ax.text(value, index, str(value), va='center', ha='left', color='b

# Customize the plot
plt.title('Top 50 Trending Channel Titles in All Countries', fontsize=
plt.xlabel('Views', fontsize=12)
plt.ylabel(None)
plt.xticks(rotation=0)  # x-axis ticks don't need rotation in horizont
plt.tight_layout()

# Add caption
plt.figtext(0.95, 0.02, "Donyoe", horizontalalignment='right', fontsiz

# Show the plot
plt.show()
```

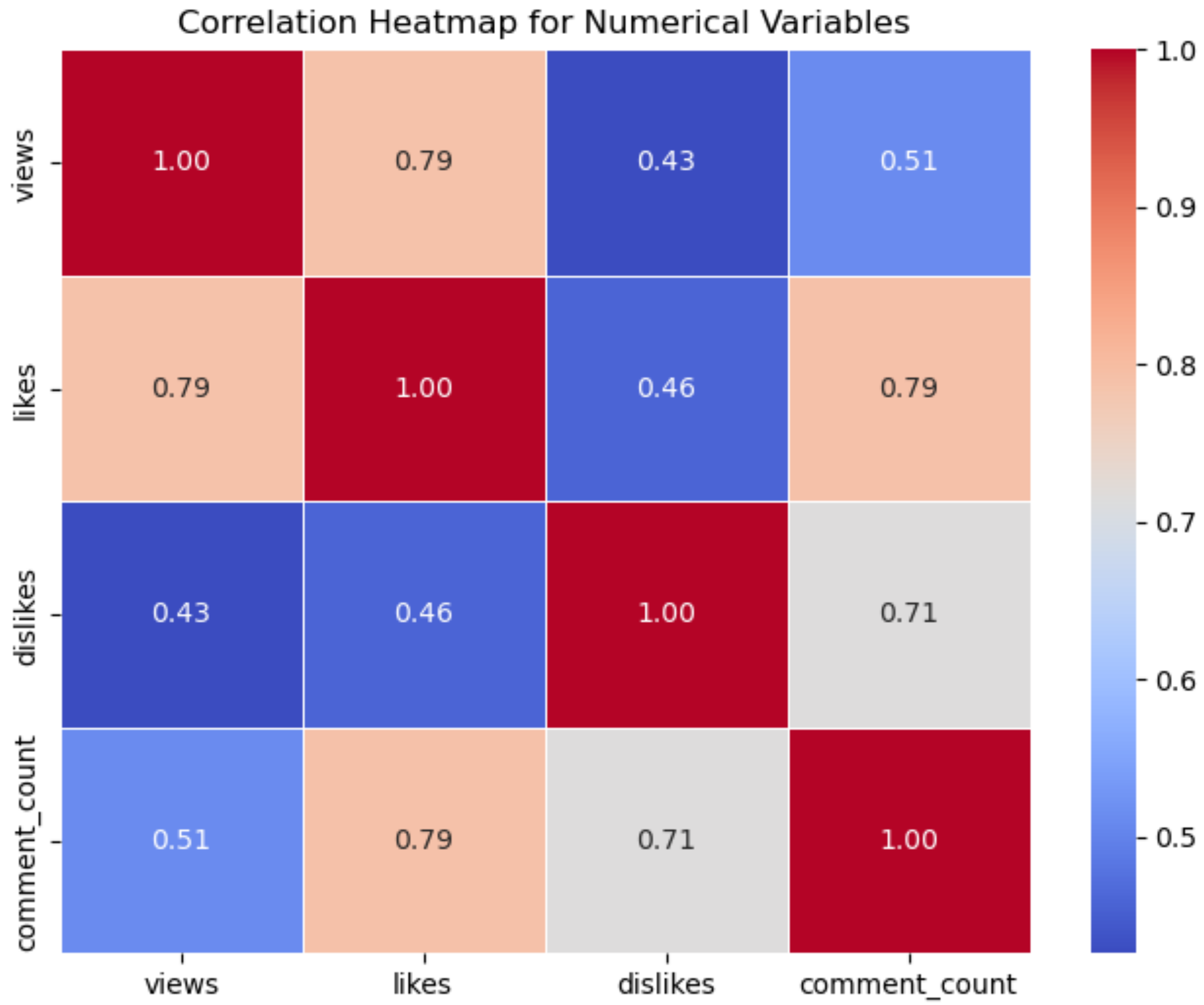


Correlation Metrics for Variables

In [27]:
```
# add category_id to numerical columns
numerical_columns = ['views', 'likes', 'dislikes', 'comment_count']
```

```
# Compute the correlation matrix
correlation_matrix = train[numerical_columns].corr()
# Display the correlation matrix
print(correlation_matrix)

# Plot the correlation matrix as a heatmap
plt.figure(figsize=(8, 6))
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', fmt='.2f'
plt.title('Correlation Heatmap for Numerical Variables')
plt.show()
```

```
                  views     likes  dislikes  comment_count
views          1.000000  0.787787  0.425866       0.511448
likes          0.787787  1.000000  0.458151       0.789545
dislikes       0.425866  0.458151  1.000000       0.713717
comment_count  0.511448  0.789545  0.713717       1.000000
```



Assign Score for Numerical Values

In [28]:

```
import pandas as pd

# Assuming the correlation values are manually entered from the heatma
correlation_values = {
    'likes': 0.784,           # Correlation of likes with views
    'dislikes': 0.416,        # Correlation of dislikes with views
    'comment_count': 0.502 # Correlation of comment_count with views
}

# Convert the correlation values to absolute values
abs_correlations = {key: abs(value) for key, value in correlation_valu

# Calculate the total sum of absolute correlations
total_correlation = sum(abs_correlations.values())

# Calculate weights by normalizing the absolute correlation values
weights = {key: value / total_correlation for key, value in abs_correl

# Convert the weights to a DataFrame for better visualization
weights_df = pd.DataFrame(list(weights.items()), columns=['Variable',

# Display the weights
print("Calculated Weights of Independent Variables Relative to 'Views'
print(weights_df)
```

```
Calculated Weights of Independent Variables Relative to 'Views':
        Variable    Weight
0          likes  0.460635
1       dislikes  0.244418
2  comment_count  0.294947
```

Creating Ranks Based on Score

In [29]:

```
import pandas as pd

weights = {
    'likes': 0.460435,
    'dislikes': 0.244418,
    'comment_count': 0.294947
}

train['score'] = (
    weights['likes'] * train['likes'] -
    weights['dislikes'] * train['dislikes'] +
    weights['comment_count'] * train['comment_count']
)

train['rank'] = train['score'].rank(ascending=False, method='min')

df_sorted = train.sort_values(by='rank')

print(df_sorted)

#output_filename = 'ranked_videos_combined.csv'
#df_sorted.to_csv(output_filename, index=False)

#print("Listing of Every Video with Individual Scores and Ranks Across
#print(df_sorted[['video_id', 'views', 'likes', 'dislikes', 'comment_c
#print(f"\nThe ranking of all videos from all locations has been savec
```

```
           video_id trending_date                                  ti
tle  \
199634  7C2z4GqqS5E    2018-06-01    BTS (방탄소년단) 'FAKE LOVE' Offic
ial MV
199433  7C2z4GqqS5E    2018-05-31    BTS (방탄소년단) 'FAKE LOVE' Offic
ial MV
158913  7C2z4GqqS5E    2018-05-31    BTS (방탄소년단) 'FAKE LOVE' Offic
ial MV
199222  7C2z4GqqS5E    2018-05-30    BTS (방탄소년단) 'FAKE LOVE' Offic
ial MV
199016  7C2z4GqqS5E    2018-05-29    BTS (방탄소년단) 'FAKE LOVE' Offic
ial MV
...             ...           ...
...
131591  LFhT6H6pRWg    2017-12-29  PSA from Chairman of the FCC Ajit
Pai
131799  LFhT6H6pRWg    2017-12-30  PSA from Chairman of the FCC Ajit
Pai
132020  LFhT6H6pRWg    2017-12-31  PSA from Chairman of the FCC Ajit
Pai
132222  LFhT6H6pRWg    2018-01-01  PSA from Chairman of the FCC Ajit
Pai
132430  LFhT6H6pRWg    2018-01-02  PSA from Chairman of the FCC Ajit
Pai

       channel_title  category_id        publish_time  \
199634       ibighit           10 2018-05-18 09:00:02
199433       ibighit           10 2018-05-18 09:00:02
158913       ibighit           10 2018-05-18 09:00:02
199222       ibighit           10 2018-05-18 09:00:02
199016       ibighit           10 2018-05-18 09:00:02
...              ...          ...                 ...
```

```
131591  Daily Caller           22 2017-12-13 22:52:57
131799  Daily Caller           22 2017-12-13 22:52:57
132020  Daily Caller           22 2017-12-13 22:52:57
132222  Daily Caller           22 2017-12-13 22:52:57
132430  Daily Caller           22 2017-12-13 22:52:57

                                                     tags    likes  d
islikes  \
199634  BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FAK...  56138
27    206892
199433  BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FAK...  55952
03    205565
158913  BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FAK...  55952
03    205565
199222  BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FAK...  55305
68    200995
199016  BIGHIT|"빅히트"|"방탄소년단"|"BTS"|"BANGTAN"|"방탄"|"FAK...  54863
49    197638
...                                                      ...      ...
...
131591  thedc|"dc"|"washington dc"|"washington"|"the d...    10426
253677
131799  thedc|"dc"|"washington dc"|"washington"|"the d...    10463
254899
132020  thedc|"dc"|"washington dc"|"washington"|"the d...    10501
255956
132222  thedc|"dc"|"washington dc"|"washington"|"the d...    10538
256816
132430  thedc|"dc"|"washington dc"|"washington"|"the d...    10576
258504

        comment_count  ... ratings_disabled  video_error_or_removed
\
199634        1228655  ...            False                   False
199433        1225326  ...            False                   False
158913        1225326  ...            False                   False
199222        1213172  ...            False                   False
199016        1204867  ...            False                   False
...               ...  ...              ...                     ...
131591          33486  ...            False                   False
131799          33651  ...            False                   False
132020          33816  ...            False                   False
132222          33681  ...            False                   False
132430          33809  ...            False                   False

                                              description       locat
ion  \
199634  BTS (방탄소년단) 'FAKE LOVE' Official MVDirector : ...
USA
199433  BTS (방탄소년단) 'FAKE LOVE' Official MVDirector : ...
```

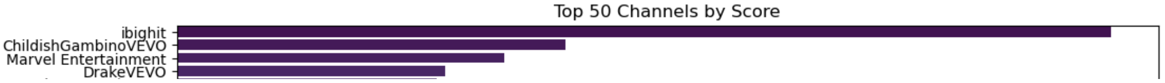
```
USA
158913  BTS (방탄소년단) 'FAKE LOVE' Official MVDirector : ...  Great B
ritain
199222  BTS (방탄소년단) 'FAKE LOVE' Official MVDirector : ...
USA
199016  BTS (방탄소년단) 'FAKE LOVE' Official MVDirector : ...
USA
...                                                   ...
...
131591  Ajit Pai has been at the heart of the net neut...  Great Brit
ain
131799  Ajit Pai has been at the heart of the net neut...  Great Brit
ain
132020  Ajit Pai has been at the heart of the net neut...  Great Brit
ain
132222  Ajit Pai has been at the heart of the net neut...  Great Brit
ain
132430  Ajit Pai has been at the heart of the net neut...  Great Brit
ain

            views trending_day_of_week  day_of_week Engagement Metric
s  \
199634  123010920               Friday       Friday            704937
4
199433  121219886             Thursday       Friday            702609
4
158913  121219886             Thursday       Friday            702609
4
199222  115664850            Wednesday       Friday            694473
5
199016  111882133              Tuesday       Friday            688885
4
...           ...                  ...          ...
...
131591    1324657               Friday    Wednesday             29758
9
131799    1331204             Saturday    Wednesday             29901
3
132020    1336646               Sunday    Wednesday             30027
3
132222    1342131               Monday    Wednesday             30103
5
132430    1348067              Tuesday    Wednesday             30288
9

               score      rank
199634  2.896622e+06       1.0
199433  2.887390e+06       2.0
158913  2.887390e+06       2.0
199222  2.855162e+06       4.0
```

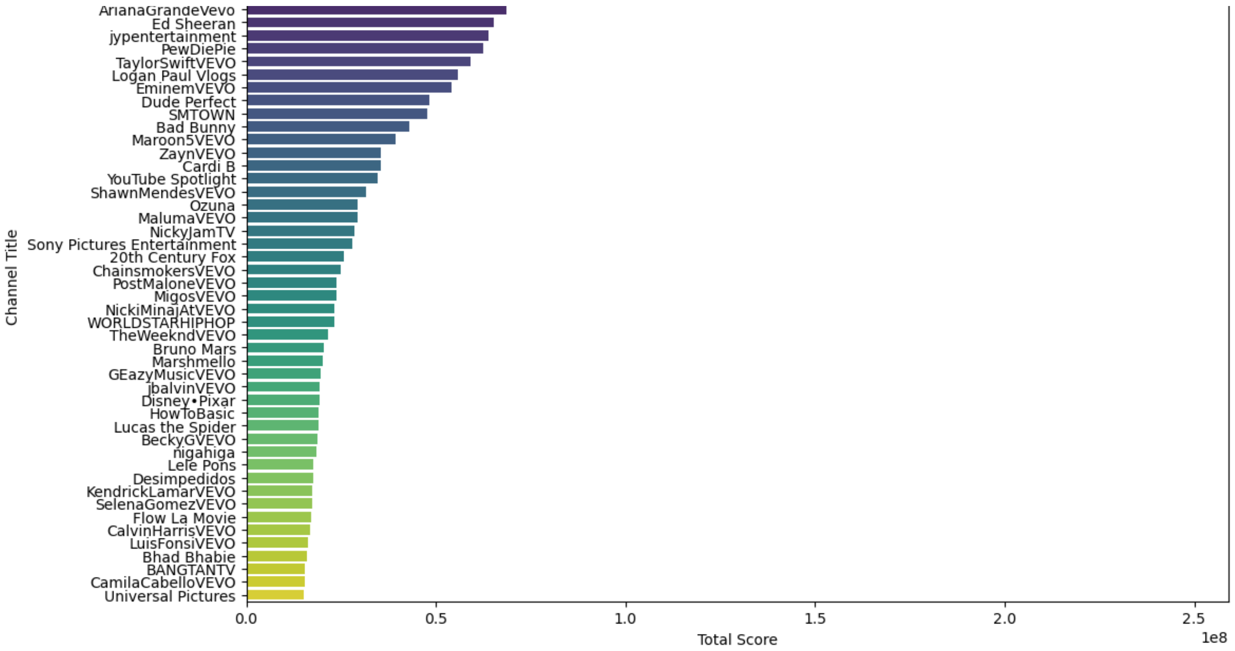
```
199016  2.833173e+06       5.0
...              ...       ...
131591 -4.732613e+04  161844.0
131799 -4.755911e+04  161845.0
132020 -4.775130e+04  161846.0
132222 -4.798428e+04  161847.0
132430 -4.834161e+04  161848.0

[161848 rows x 22 columns]
```

EDA for Score for Top 50 Channels

In [30]:

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Assuming your DataFrame is named 'train'
weights = {
    'likes': 0.460435,
    'dislikes': 0.244418,
    'comment_count': 0.294947
}

# Calculate score and rank
train['score'] = (
    weights['likes'] * train['likes'] -
    weights['dislikes'] * train['dislikes'] +
    weights['comment_count'] * train['comment_count']
)

train['rank'] = train['score'].rank(ascending=False, method='min')

# Group by channel_title and sum the scores
channel_scores = train.groupby('channel_title')['score'].sum().reset_i

# Sort by total score and get top 50 channels
top_channels = channel_scores.sort_values(by='score', ascending=False)

# Create a bar plot for the top 50 channels
plt.figure(figsize=(12, 8))
sns.barplot(x='score', y='channel_title', data=top_channels, palette='
plt.title('Top 50 Channels by Score')
plt.xlabel('Total Score')
plt.ylabel('Channel Title')
plt.show()
```





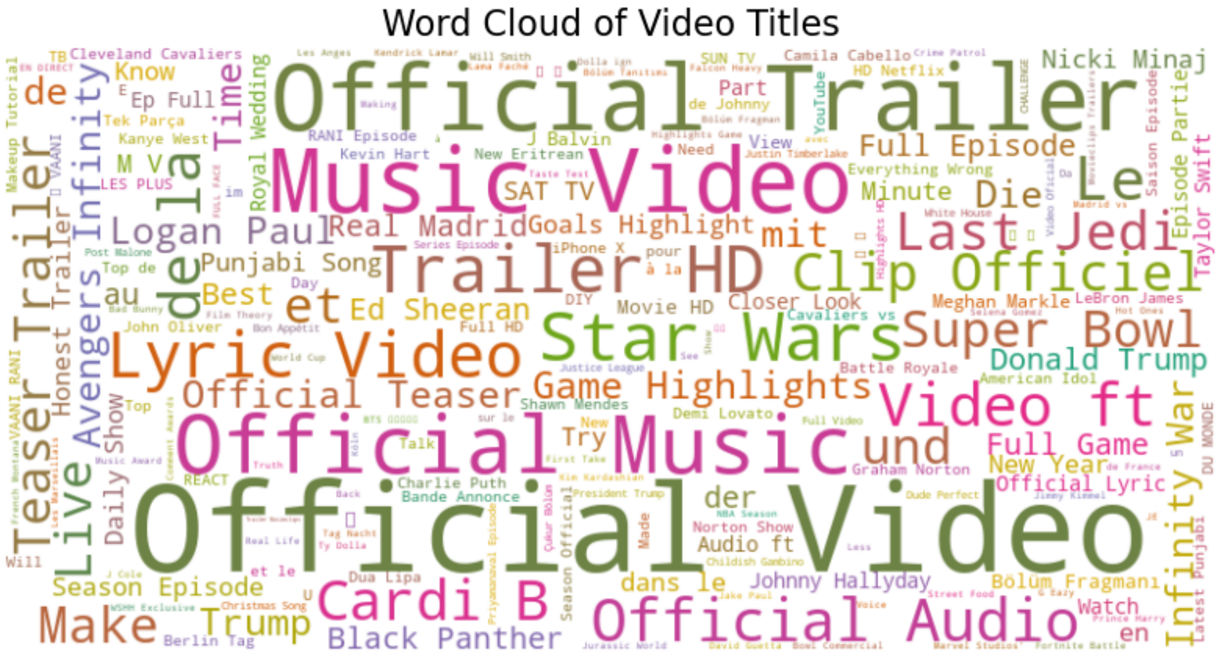
Creating Word Cloud

In [34]:

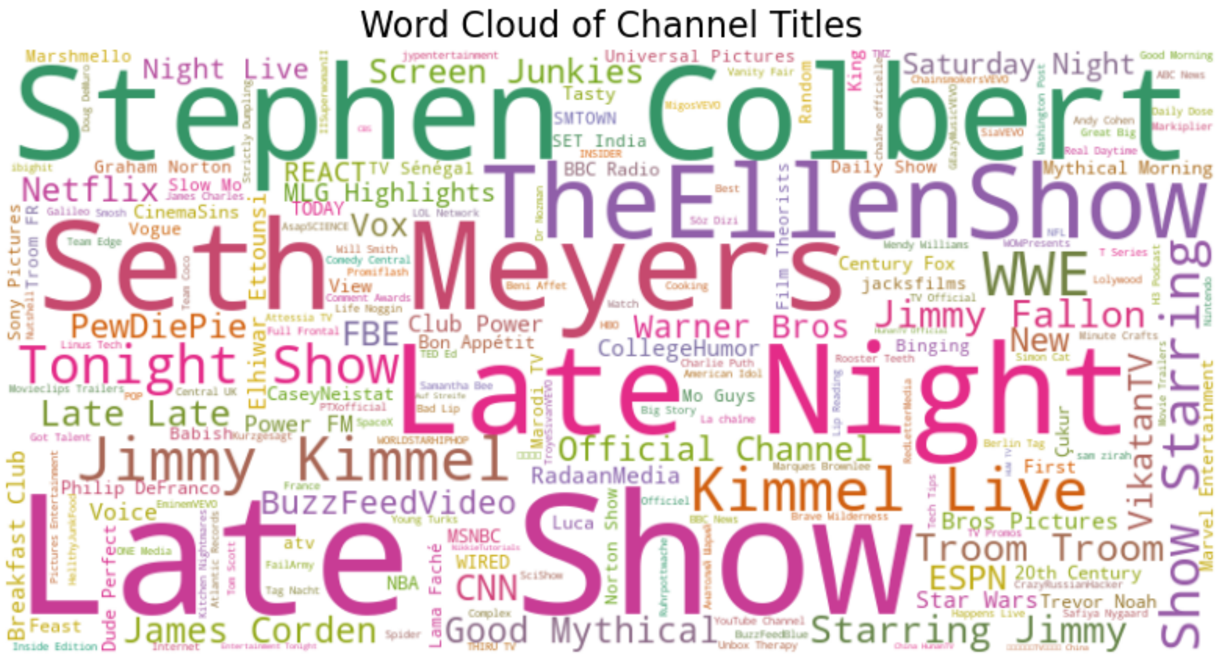
```
!pip install palettable

# Creating Word Cloud—Video Titles
from wordcloud import WordCloud
from palettable.colorbrewer.qualitative import Dark2_6

# Assuming your DataFrame is named 'mergeda_df'
# Concatenate all titles into a single string
all_titles = " ".join(train['title'].astype(str))

# Set up the color palette (equivalent to R's "Dark2")
cmap = Dark2_6.mpl_colormap

# Create a WordCloud object
wordcloud = WordCloud(
    background_color="white",
    max_words=200,
    colormap=cmap,
    width=800,
    height=400,
    random_state=42
)

# Generate the word cloud from the titles
wordcloud.generate(all_titles)

# Plot the word cloud
```

```
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")  # Turn off the axis
plt.title('Word Cloud of Video Titles', fontsize=16)
plt.show()
```

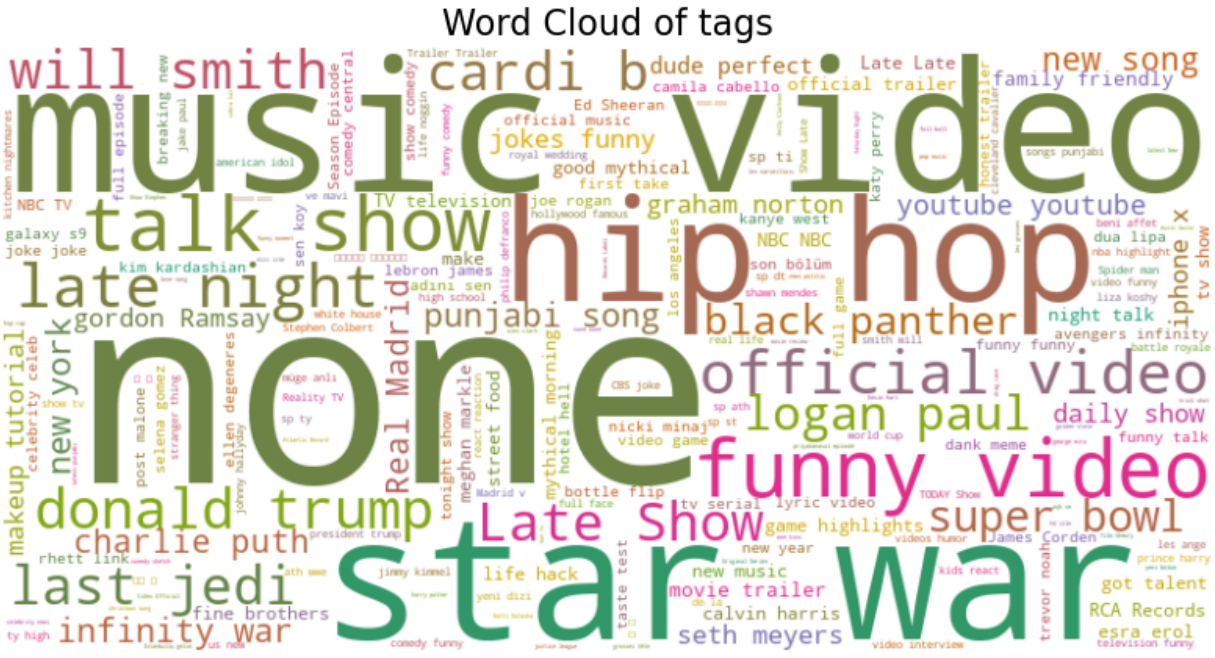
```
Collecting palettable
  Obtaining dependency information for palettable from https://files.
pythonhosted.org/packages/cf/f7/3367feadd4ab56783b0971c9b7edfbdd68e0c
70ce877949a5dd2117ed4a0/palettable-3.3.3-py2.py3-none-any.whl.metadat
a (https://files.pythonhosted.org/packages/cf/f7/3367feadd4ab56783b09
71c9b7edfbdd68e0c70ce877949a5dd2117ed4a0/palettable-3.3.3-py2.py3-non
e-any.whl.metadata)
  Downloading palettable-3.3.3-py2.py3-none-any.whl.metadata (3.3 kB)
Downloading palettable-3.3.3-py2.py3-none-any.whl (332 kB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 332.3/332.3 kB 5.0 MB/s e
ta 0:00:00a 0:00:01
Installing collected packages: palettable
Successfully installed palettable-3.3.3
```



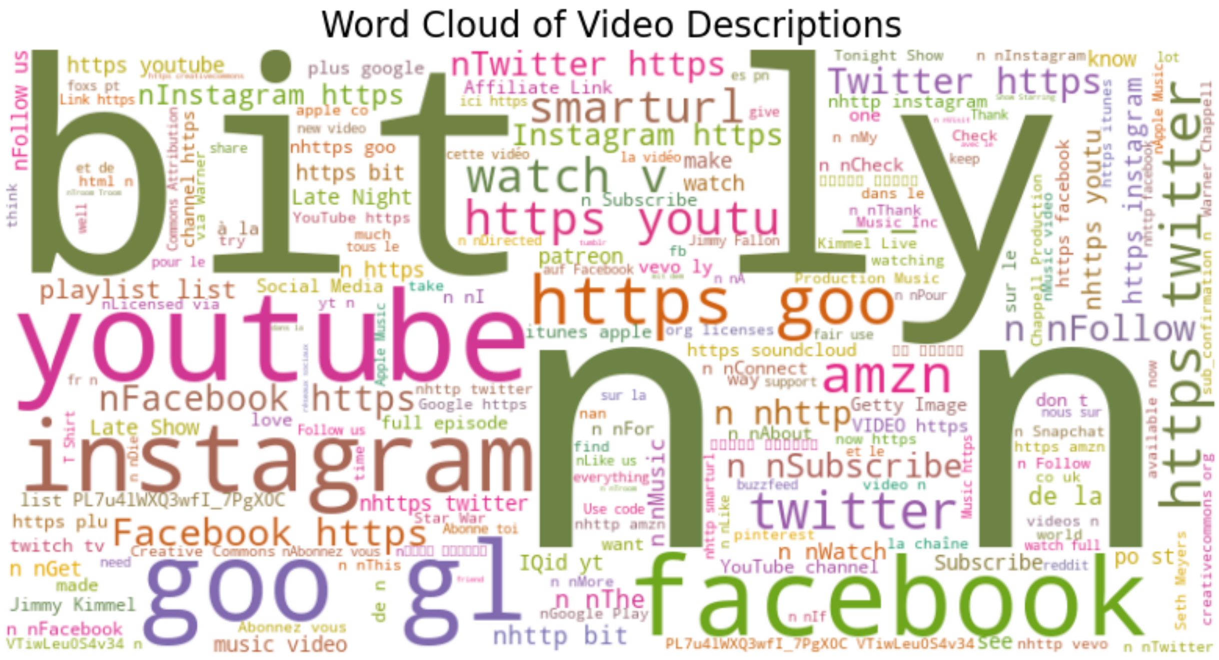
```
In [35]: # Creating Word Cloud-Channel Title
         all_channel_titles = " ".join(train['channel_title'].astype(str))

         # Set up the color palette (equivalent to R's "Dark2")
         cmap = Dark2_6.mpl_colormap

         # Create a WordCloud object
         wordcloud = WordCloud(
             background_color="white",
             max_words=200,
             colormap=cmap,
             width=800,
             height=400,
             random_state=42
         )

         # Generate the word cloud from the titles
         wordcloud.generate(all_channel_titles)

         # Plot the word cloud
         plt.figure(figsize=(10, 6))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis("off")  # Turn off the axis
         plt.title('Word Cloud of Channel Titles', fontsize=16)
         plt.show()
```



In [36]:
```
# Creating Word Cloud-tags
all_tags = " ".join(train['tags'].astype(str))

# Set up the color palette (equivalent to R's "Dark2")
cmap = Dark2_6.mpl_colormap

# Create a WordCloud object
wordcloud = WordCloud(
    background_color="white",
    max_words=200,
    colormap=cmap,
    width=800,
    height=400,
    random_state=42
)

# Generate the word cloud from the titles
wordcloud.generate(all_tags)

# Plot the word cloud
plt.figure(figsize=(10, 6))
plt.imshow(wordcloud, interpolation="bilinear")
plt.axis("off")  # Turn off the axis
plt.title('Word Cloud of tags', fontsize=16)
plt.show()
```



```
In [38]: # Creating Word Cloud-description
         all_description = " ".join(train['description'].astype(str))

         # Set up the color palette (equivalent to R's "Dark2")
         cmap = Dark2_6.mpl_colormap

         # Create a WordCloud object
         wordcloud = WordCloud(
             background_color="white",
             max_words=200,
             colormap=cmap,
             width=800,
             height=400,
             random_state=42
         )

         # Generate the word cloud from the titles
         wordcloud.generate(all_description)

         # Plot the word cloud
         plt.figure(figsize=(10, 6))
         plt.imshow(wordcloud, interpolation="bilinear")
         plt.axis("off")  # Turn off the axis
         plt.title('Word Cloud of Video Descriptions', fontsize=16)
         plt.show()
```



```
In [ ]:
```