In [2]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from datetime import datetime
```

In [3]:
```
df=pd.read_csv('C:\\Users\\hp\\Downloads\\INDIAvi.csv')
```

In [4]:
```
df.head()
```

Out[4]:

	video_id	trending_date	title	channel_title	category_id	publish_time	
0	kzwfHumJyYc	17.14.11	Sharry Mann: Cute Munda (Song Teaser) | Parmi...	Lokdhun Punjabi	1	2017-11-12T12:20:39.000Z	sharry song"|
1	zUZ1z7FwLc8	17.14.11	पीरियड्स के समय, पेट पर पति करता ऐसा, देखकर दं...	HJ NEWS	25	2017-11-13T05:43:56.000Z	पीरियड् ऐ
2	10L1hZ9qa58	17.14.11	Stylish Star Allu Arjun @ ChaySam Wedding Rece...	TFPC	24	2017-11-12T15:48:08.000Z	Stylish @ Chay
3	N1vE8iiEg64	17.14.11	Eruma Saani | Tamil vs English	Eruma Saani	23	2017-11-12T07:08:48.000Z	Erum Videos"|"
4	kJzGH0PVQHQ	17.14.11	why Samantha became EMOTIONAL @ Samantha naga ...	Filmylooks	24	2017-11-13T01:14:16.000Z	Film n

In [5]:
```
df.shape
```

Out[5]:
```
(37352, 16)
```

In [6]:
```
df=df.drop_duplicates()
df.shape
```

Out[6]:
```
(33089, 16)
```

In [7]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 33089 entries, 0 to 37330
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                33089 non-null  object
 1   trending_date           33089 non-null  object
 2   title                   33089 non-null  object
 3   channel_title           33089 non-null  object
 4   category_id             33089 non-null  int64
 5   publish_time            33089 non-null  object
 6   tags                    33089 non-null  object
 7   views                   33089 non-null  int64
 8   likes                   33089 non-null  int64
 9   dislikes                33089 non-null  int64
 10  comment_count           33089 non-null  int64
 11  thumbnail_link          33089 non-null  object
 12  comments_disabled       33089 non-null  bool
 13  ratings_disabled        33089 non-null  bool
 14  video_error_or_removed  33089 non-null  bool
 15  description             32562 non-null  object
dtypes: bool(3), int64(5), object(8)
memory usage: 3.6+ MB
```

In [8]:
```
df.describe()
```

Out[8]:

	category_id	views	likes	dislikes	comment_count
count	33089.000000	3.308900e+04	3.308900e+04	3.308900e+04	33089.000000
mean	21.628154	9.963425e+05	2.558762e+04	1.576535e+03	2524.777660
std	6.493615	3.148111e+06	9.647320e+04	1.689573e+04	14769.825108
min	1.000000	4.024000e+03	0.000000e+00	0.000000e+00	0.000000
25%	23.000000	1.127190e+05	7.870000e+02	9.800000e+01	72.000000
50%	24.000000	2.750270e+05	2.757000e+03	2.890000e+02	298.000000
75%	24.000000	7.320220e+05	1.201100e+04	9.320000e+02	1169.000000
max	43.000000	1.254322e+08	2.912710e+06	1.545017e+06	827755.000000

In [9]:
```
columns_to_remove = ['thumbnail_link','description']
df = df.drop(columns=columns_to_remove)
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 33089 entries, 0 to 37330
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   video_id                33089 non-null  object
 1   trending_date           33089 non-null  object
 2   title                   33089 non-null  object
 3   channel_title           33089 non-null  object
 4   category_id             33089 non-null  int64
 5   publish_time            33089 non-null  object
 6   tags                    33089 non-null  object
 7   views                   33089 non-null  int64
 8   likes                   33089 non-null  int64
 9   dislikes                33089 non-null  int64
 10  comment_count           33089 non-null  int64
 11  comments_disabled       33089 non-null  bool
 12  ratings_disabled        33089 non-null  bool
 13  video_error_or_removed  33089 non-null  bool
dtypes: bool(3), int64(5), object(6)
memory usage: 3.1+ MB
```

In [11]:
```
df["trending_date"] = df["trending_date"].apply(lambda x : datetime.strptime(x, '%y
df.head(2)
```

Out[11]:

	video_id	trending_date	title	channel_title	category_id	publish_time	tags
0	kzwfHumJyYc	2017-11-14	Sharry Mann: Cute Munda (Song Teaser) | Parmi...	Lokdhun Punjabi	1	2017-11-12T12:20:39.000Z	sharry mann|"sharry mann new song"|"sharry man...
1	zUZ1z7FwLc8	2017-11-14	पीरियड्स के समय, पेट पर पति करता ऐसा, देखकर दं...	HJ NEWS	25	2017-11-13T05:43:56.000Z	पीरियड्स के समय|"पेट पर पति करता ऐसा"|"देखकर द...

In [12]:
```
df['publish_time'] = pd.to_datetime(df['publish_time'])
df.head(2)
```

Out[12]:

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	
0	kzwfHumJyYc	2017-11-14	Sharry Mann: Cute Munda (Song Teaser) | Parmi...	Lokdhun Punjabi	1	2017-11-12 12:20:39+00:00	sharry mann|"sharry mann new song"|"sharry man...	1
1	zUZ1z7FwLc8	2017-11-14	पीरियड्स के समय, पेट पर पति करता ऐसा, देखकर दं...	HJ NEWS	25	2017-11-13 05:43:56+00:00	पीरियड्स के समय|"पेट पर पति करता ऐसा"|"देखकर द...	

In [13]:
```
df['publish_month'] = df['publish_time'].dt.month
df['publish_day'] = df['publish_time'].dt.day
df['publish_hour'] = df['publish_time'].dt.hour
df.head(2)
```

Out[13]:

	video_id	trending_date	title	channel_title	category_id	publish_time	tags	
0	kzwfHumJyYc	2017-11-14	Sharry Mann: Cute Munda (Song Teaser) | Parmi...	Lokdhun Punjabi	1	2017-11-12 12:20:39+00:00	sharry mann|"sharry mann new song"|"sharry man...	1
1	zUZ1z7FwLc8	2017-11-14	पीरियड्स के समय, पेट पर पति करता ऐसा, देखकर दं...	HJ NEWS	25	2017-11-13 05:43:56+00:00	पीरियड्स के समय|"पेट पर पति करता ऐसा"|"देखकर द...	

In [14]:
```
print (sorted(df["category_id"].unique()))
[1, 2, 10, 15, 17, 19, 20, 222, 23, 24, 25, 26, 27, 28, 29, 30, 43]
```

Out[14]:
```
[1, 2, 10, 15, 17, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 30, 43]
[1, 2, 10, 15, 17, 19, 20, 222, 23, 24, 25, 26, 27, 28, 29, 30, 43]
```

In [15]:
```
df['category_name'] = np.nan
df.loc[(df["category_id"] == 1), "category_name"] = 'Film and Animation'
df.loc[(df["category_id"] == 2), "category_name"] = 'Autos and Vehicles'
df.loc[(df["category_id"] == 10), "category_name"] = 'Music'
df.loc[(df["category_id"] == 15), "category_name"] = 'Pets and Animals'
df.loc[(df["category_id"] == 17), "category_name"] = 'Sports'
df.loc[(df["category_id"] == 19), "category_name"] = 'Travel and Events'
df.loc[(df["category_id"] == 20), "category_name"] = 'Gaming'
df.loc[(df["category_id"] == 22), "category_name"] = 'People and Blogs'
df.loc[(df["category_id"] == 23), "category_name"] = 'Comedy'
```

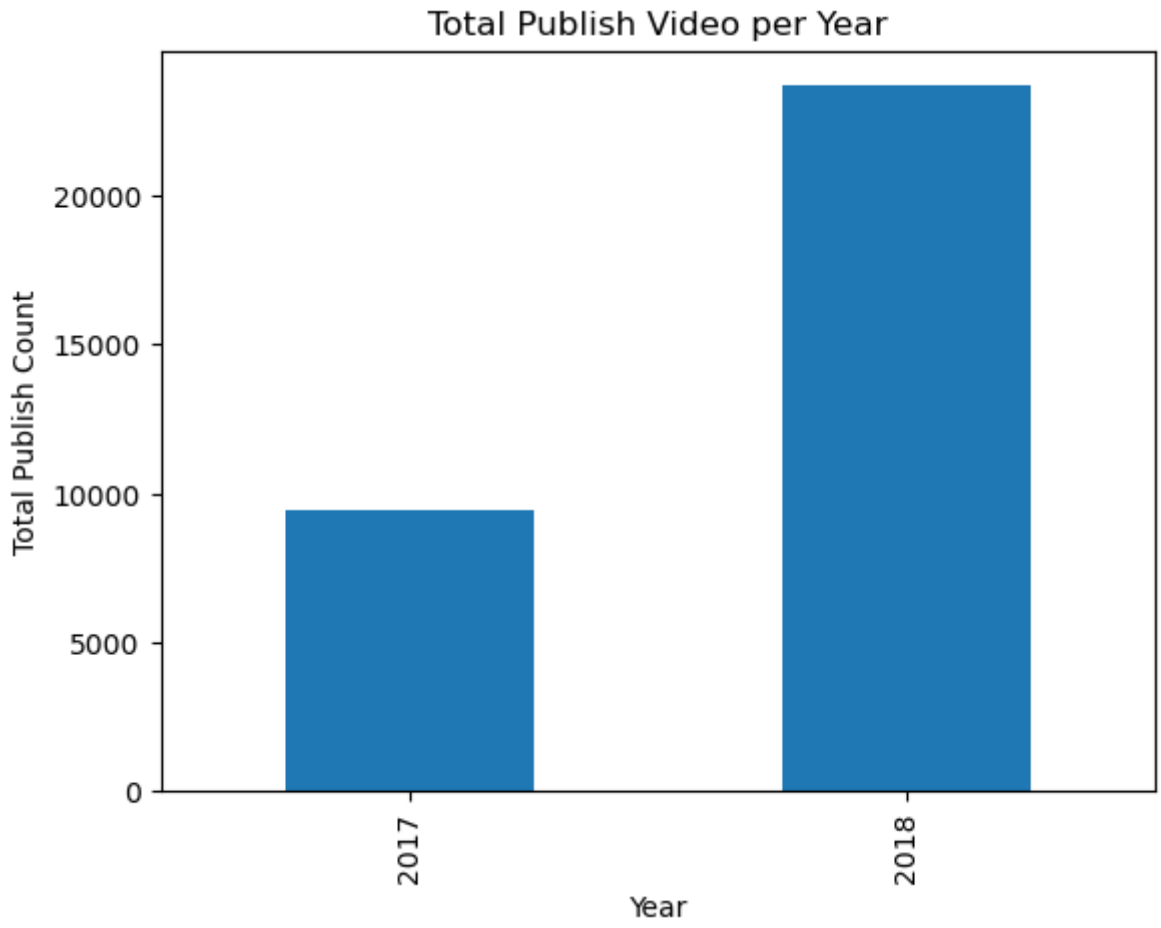
```
df.loc[(df["category_id"] == 24), "category_name"] = 'Entertainment'
df.loc[(df["category_id"] == 25), "category_name"] = 'News and Politics'
df.loc[(df["category_id"] == 26), "category_name"] = 'How to and Style'
df.loc[(df["category_id"] == 27), "category_name"] = 'Education'
df.loc[(df["category_id"] == 28), "category_name"] = 'Science and Technology'
df.loc[(df["category_id"] == 29), "category_name"] = 'Non Profits and Activism'
df.loc[(df["category_id"] == 30), "category_name"] = 'Movies'
df.loc[(df["category_id"] == 43), "category_name"] = 'Shows'
```

In [16]:
```
df.head()
```

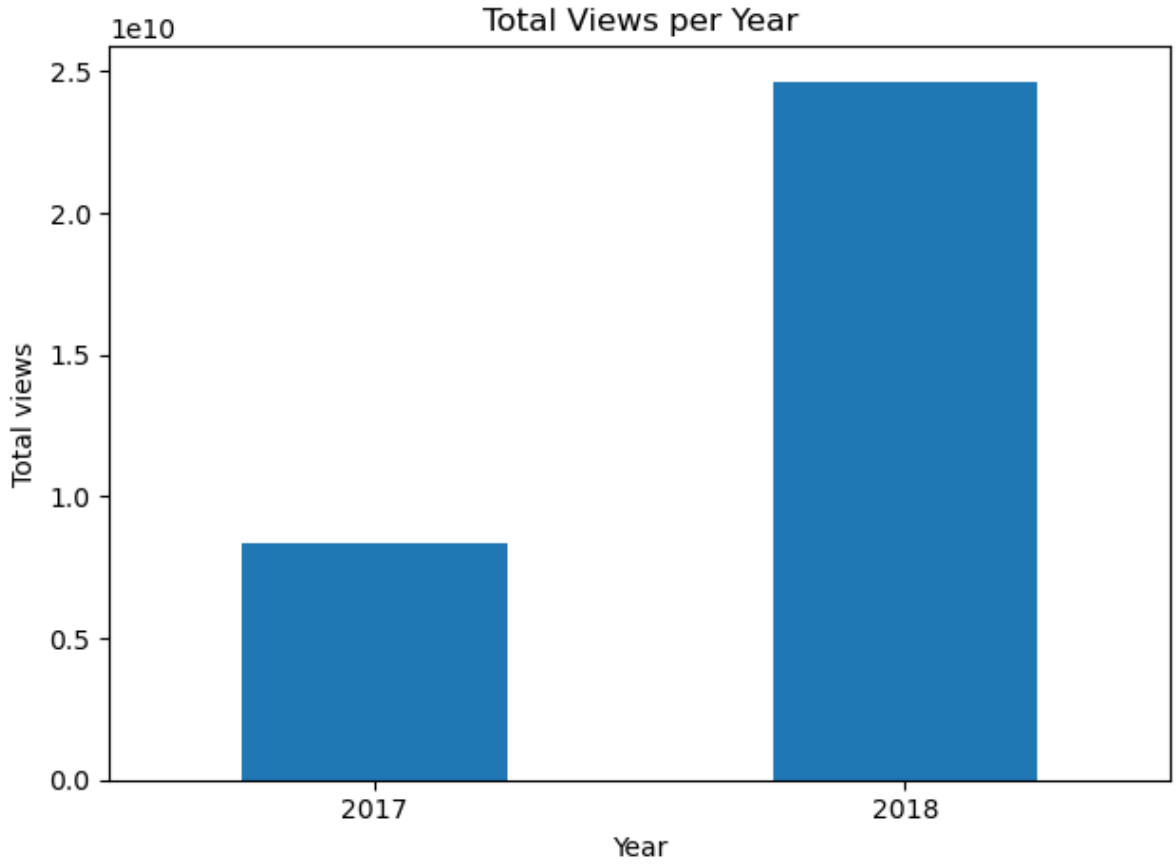
Out[16]:

	video_id	trending_date	title	channel_title	category_id	publish_time	
0	kzwfHumJyYc	2017-11-14	Sharry Mann: Cute Munda (Song Teaser) | Parmi...	Lokdhun Punjabi	1	2017-11-12 12:20:39+00:00	sharry m song"|"s
1	zUZ1z7FwLc8	2017-11-14	पीरियड्स के समय, पेट पर पति करता ऐसा, देखकर दं...	HJ NEWS	25	2017-11-13 05:43:56+00:00	पीरियड्स ऐसा"
2	10L1hZ9qa58	2017-11-14	Stylish Star Allu Arjun @ ChaySam Wedding Rece...	TFPC	24	2017-11-12 15:48:08+00:00	Stylish Sta @ ChaySa
3	N1vE8iiEg64	2017-11-14	Eruma Saani | Tamil vs English	Eruma Saani	23	2017-11-12 07:08:48+00:00	Eruma Videos"|"Fi
4	kJzGH0PVQHQ	2017-11-14	why Samantha became EMOTIONAL @ Samantha naga ...	Filmylooks	24	2017-11-13 01:14:16+00:00	Filmyl ne mo

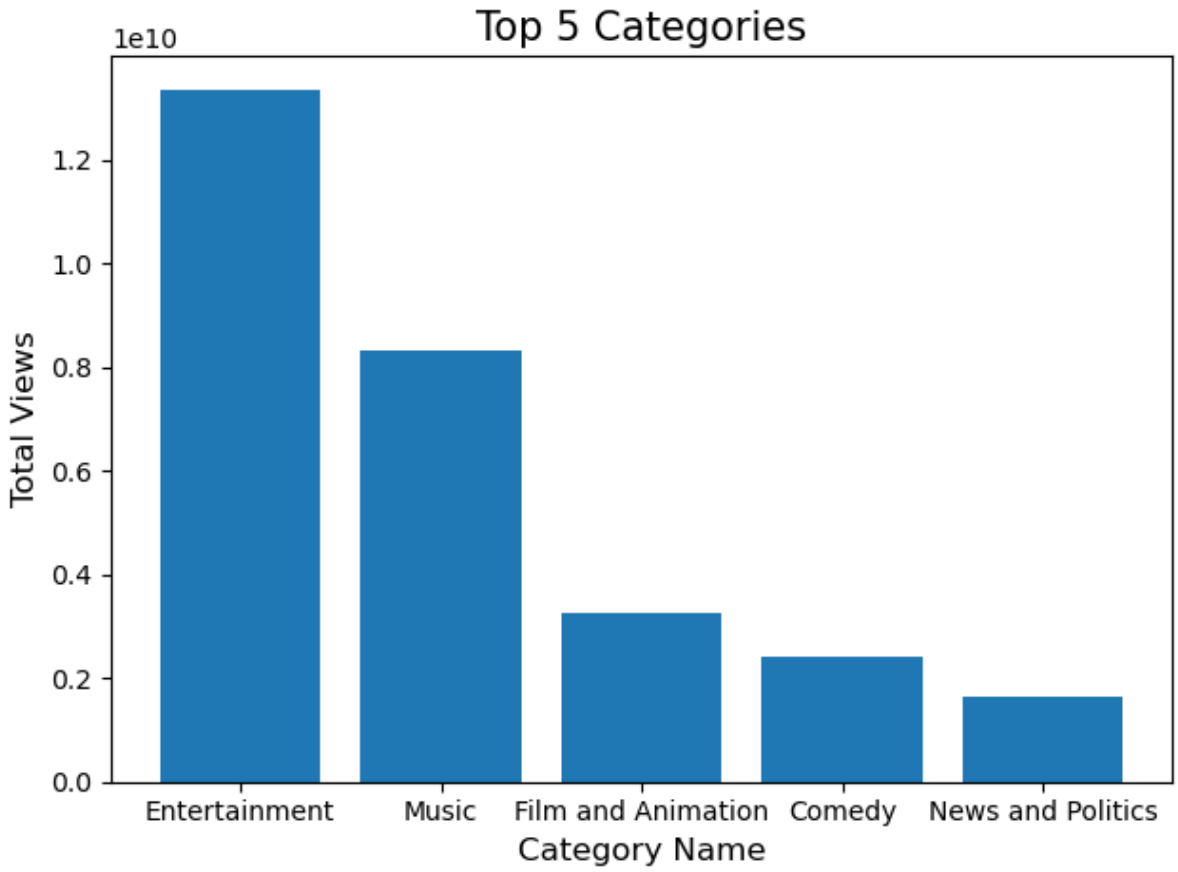
In [19]:
```
df['year'] = df['publish_time'].dt.year
yearly_counts = df.groupby('year')['video_id'].count()

#create a bar chart
yearly_counts.plot(kind='bar', xlabel='Year', ylabel='Total Publish Count', title='

#show the chart
plt.show()
```



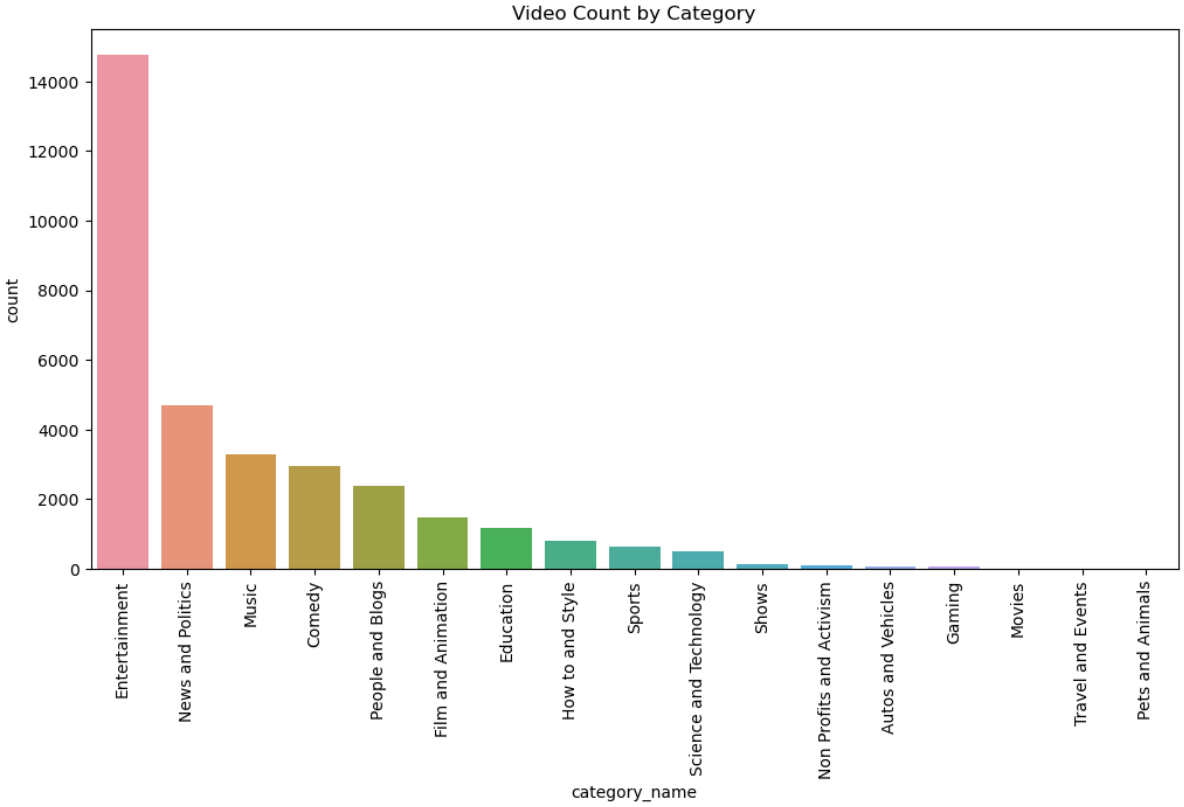
In [18]:
```
#Group by year and sum the views for each year
yearly_views = df.groupby('year')['views'].sum()

#Create a bar chart
yearly_views.plot(kind='bar', xlabel='Year', ylabel='Total views', title='Total Vie
plt.xticks(rotation=0)
plt.tight_layout()

#Show the bar chart
plt.show()
```



In [21]:
```
# Group the data by 'category_name' and calculate the sum of 'views' in each categ
category_views = df.groupby('category_name')['views'].sum().reset_index()

#Sort the categories by views in descending order
top_categories = category_views.sort_values(by='views', ascending=False).head(5)

#create a bar plot to visualize the top 5 categories

plt.bar(top_categories['category_name'], top_categories['views'])
plt.xlabel('Category Name',fontsize=12)
plt.ylabel('Total Views',fontsize=12)
plt.title('Top 5 Categories',fontsize=15)
plt.tight_layout()
plt.show()
```

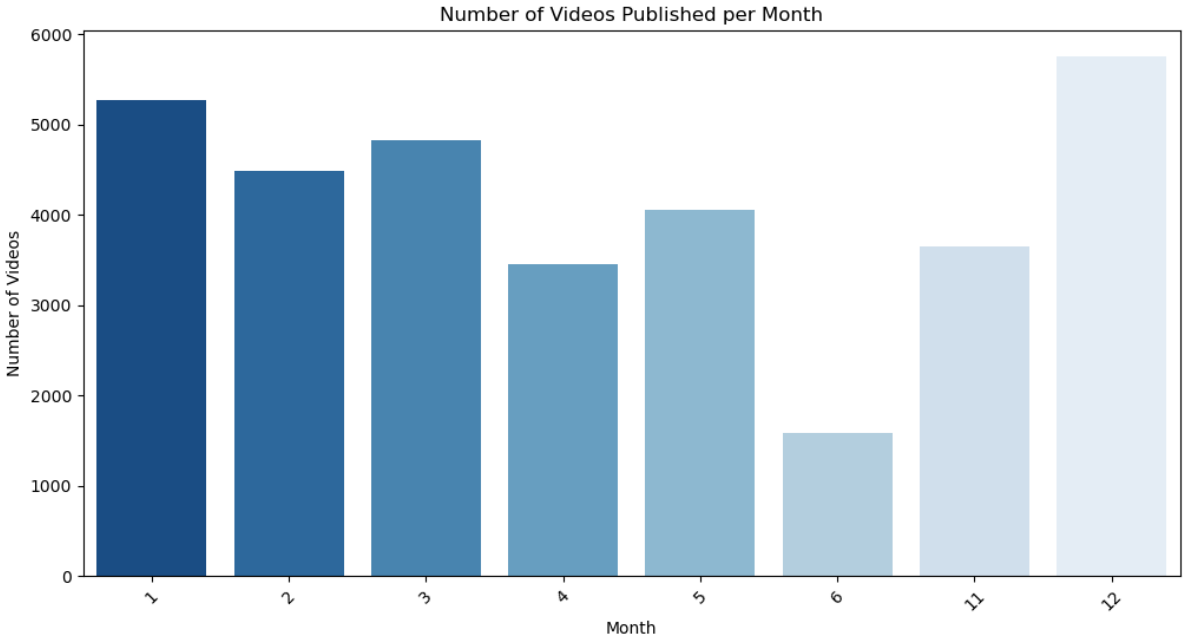


In [23]:
```
plt.figure(figsize=(12, 6))
sns.countplot(x='category_name', data=df, order=df['category_name'].value_counts().
plt.xticks(rotation=90)
plt.title('Video Count by Category')
plt.show()
```



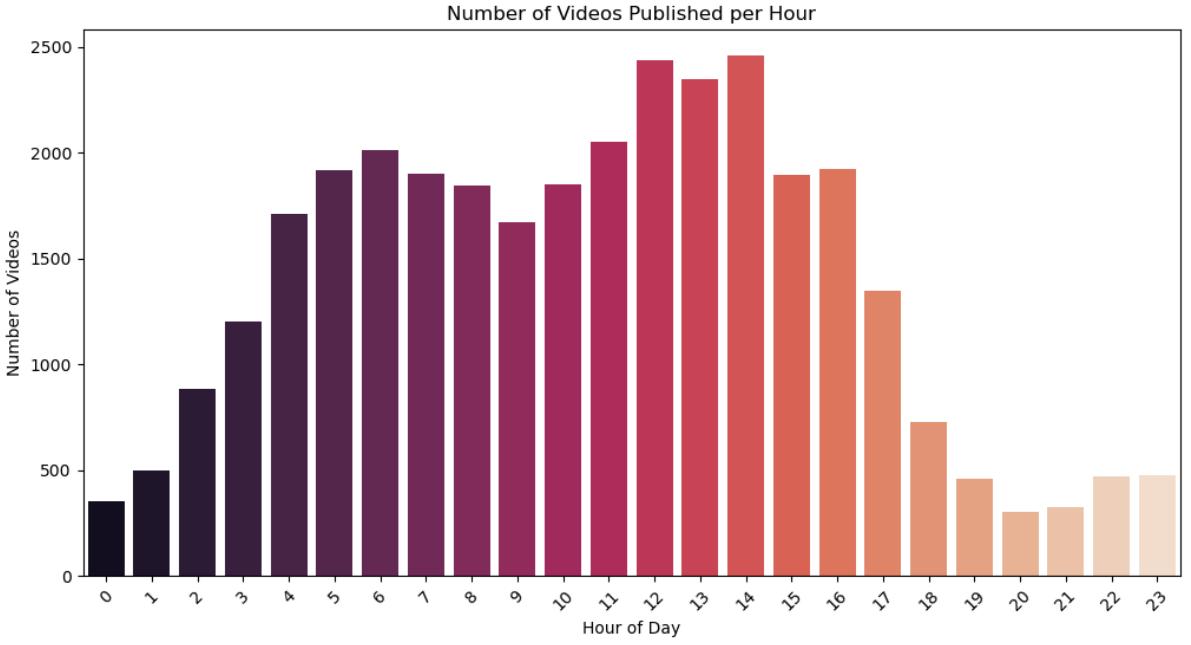
In [24]:
```
#Group the data by 'publish_month' and count the number of videos in each month
videos_per_month = df['publish_month'].value_counts().sort_index()
```

```
#Create a bar plot to visualize the number of videos per month
plt.figure(figsize=(12, 6))
sns.barplot(x=videos_per_month.index, y=videos_per_month.values, palette='Blues_r')
plt.title('Number of Videos Published per Month')
plt.xlabel('Month')
plt.ylabel('Number of Videos')
plt.xticks(rotation=45)
plt.show()
```

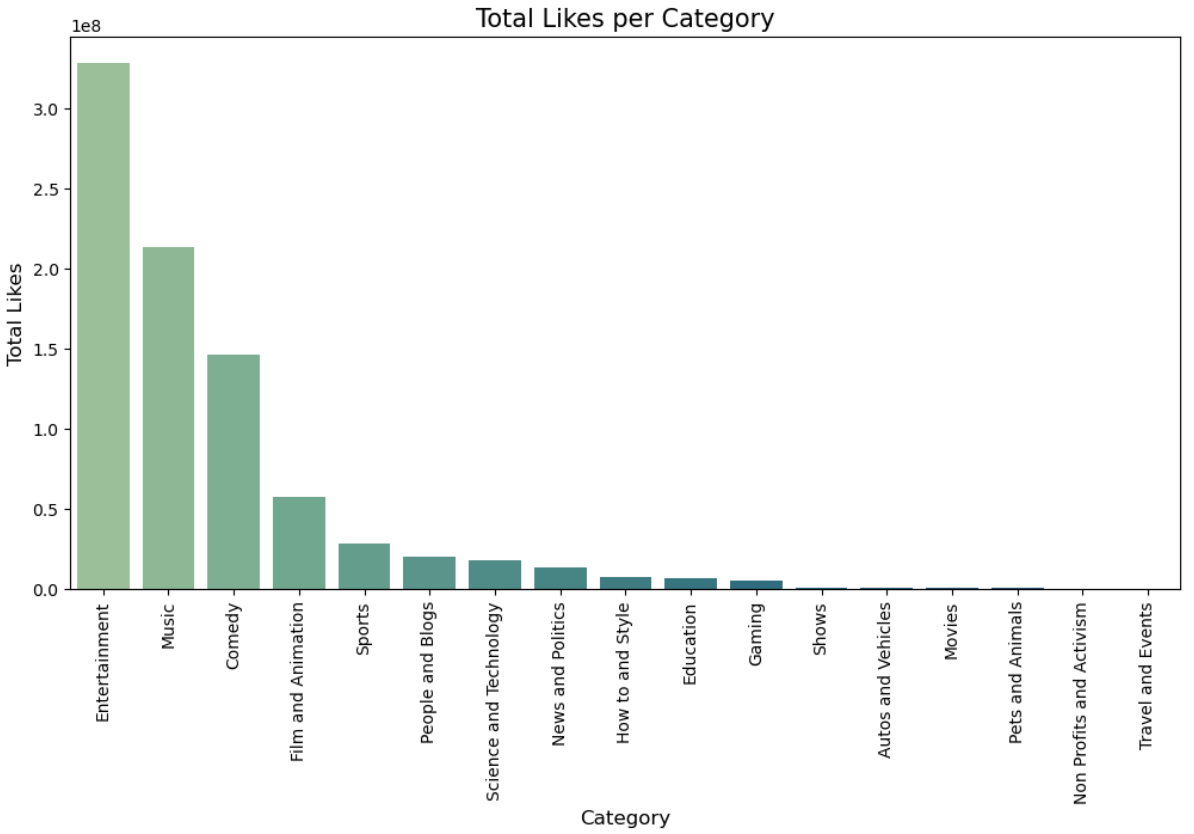


In [25]:

```
#Count the number of videos published per hour
videos_per_hour = df['publish_hour'].value_counts().sort_index()

#create a bar plot
plt.figure(figsize=(12, 6))
sns.barplot(x=videos_per_hour.index, y=videos_per_hour.values, palette='rocket')
plt.title('Number of Videos Published per Hour')
plt.xlabel('Hour of Day')
plt.ylabel('Number of Videos')
plt.xticks(rotation=45)
plt.show()
```

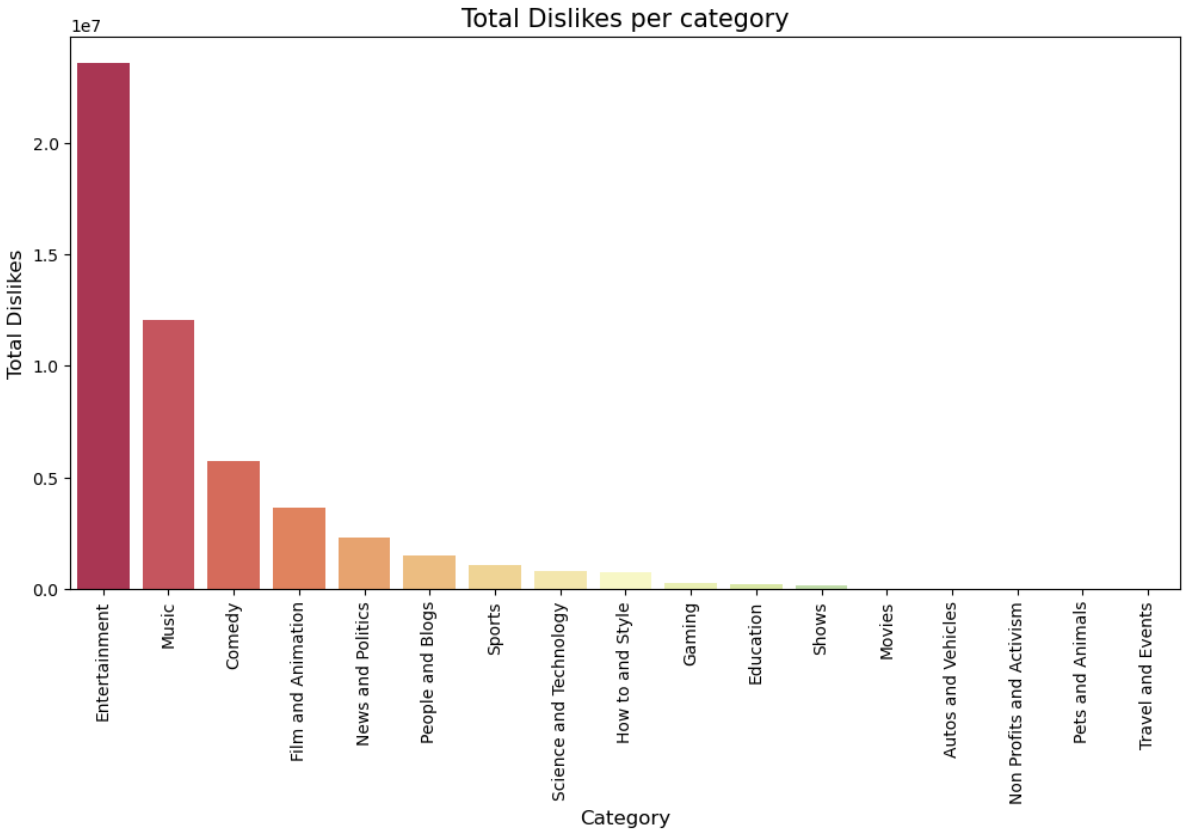


In [26]:
```
#Group the data by 'category_name' and sum the 'likes' for each category
category_likes = df.groupby('category_name')['likes'].sum().sort_values(ascending=F

#Create a bar plot to visualize the total likes per category
plt.figure(figsize=(12, 6))
sns.barplot(x=category_likes.index, y=category_likes.values, palette='crest')
plt.title('Total Likes per Category', fontsize=15)
plt.xlabel('Category', fontsize=12)
plt.ylabel('Total Likes', fontsize=12)
plt.xticks(rotation=90)
plt.show()

#print the category with the most likes
most_liked_category = category_likes.idxmax()
print(f"The category with the most likes is '{most_liked_category}' with {category_
```



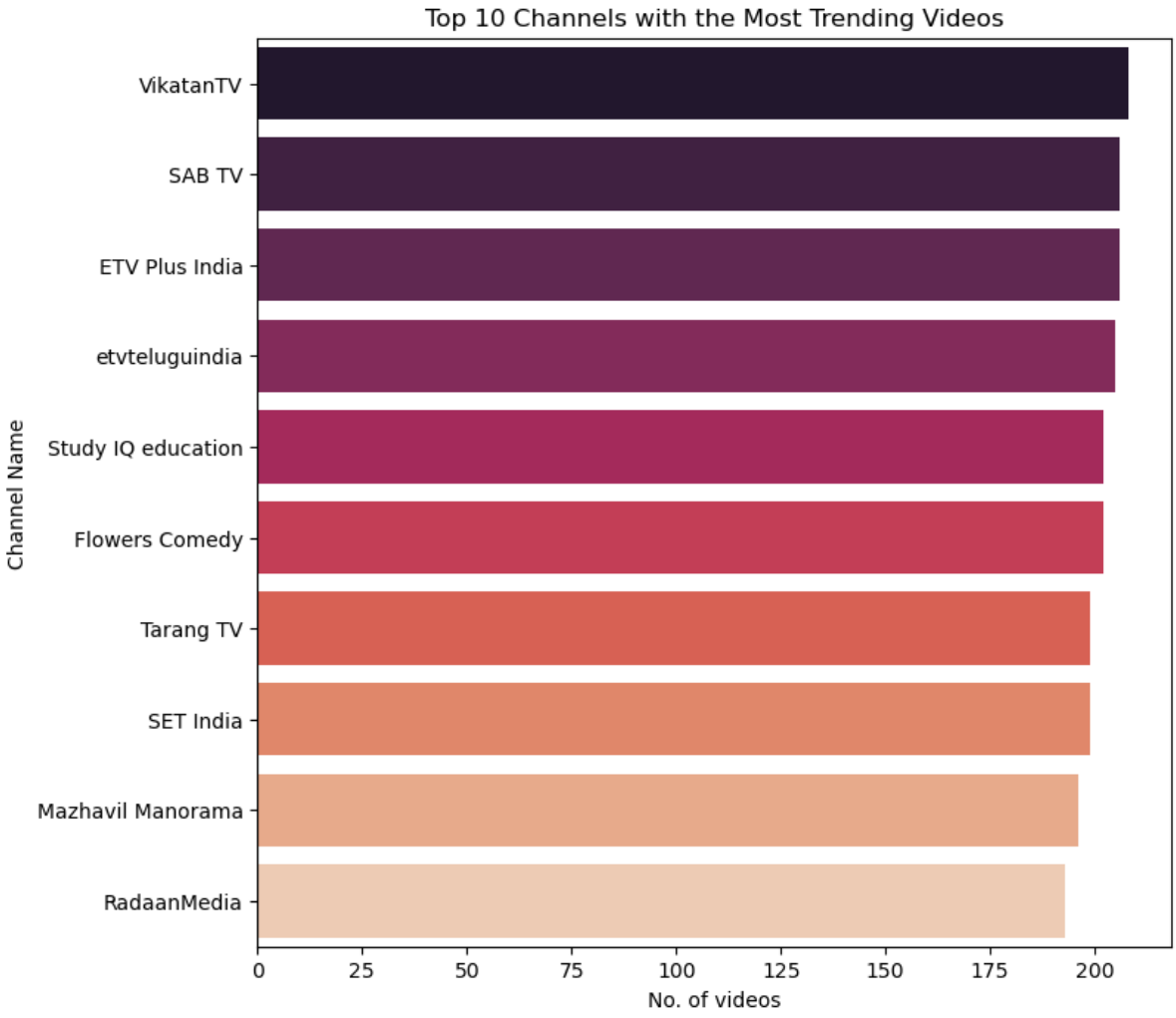

```
The category with the most likes is 'Entertainment' with 328,229,550 likes.
```

In [27]:
```
#Group the data by 'category_name' and sum the 'dislikes' for each category
category_dislikes = df.groupby('category_name')['dislikes'].sum().sort_values(ascer

#Create a bar plot to visualize the total dislikes per category
plt.figure(figsize=(12, 6))
sns.barplot(x=category_dislikes.index, y=category_dislikes.values,palette="Spectral
plt.title('Total Dislikes per category', fontsize=15)
plt.xlabel('Category', fontsize=12)
plt.ylabel('Total Dislikes', fontsize=12)
plt.xticks(rotation=90)
plt.show()

#print the category with the most dislikes
most_disliked_category = category_dislikes.idxmax()
print(f"The category with the most dislikes is '{most_disliked_category}' with {cat
```



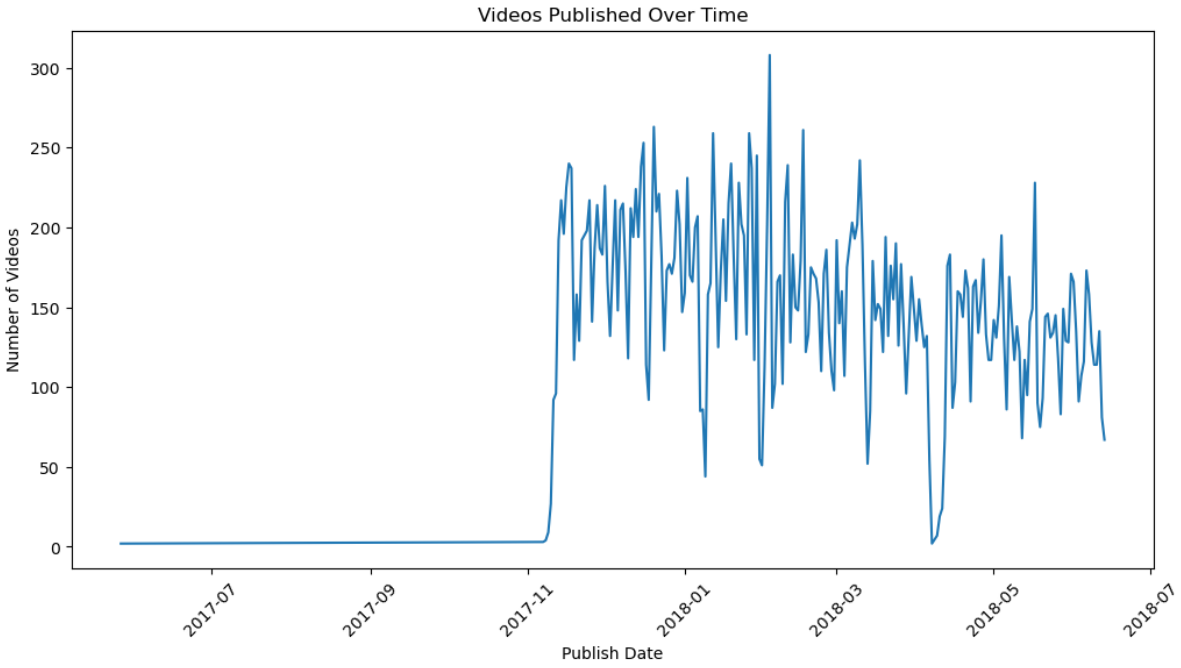
The category with the most dislikes is 'Entertainment' with 23,577,924 dislikes.

In [28]:
```
cdf = df.groupby("channel_title").size().reset_index(name="video_count") \
      .sort_values("video_count", ascending=False).head(10)

fig, ax = plt.subplots(figsize=(8,8))
_ = sns.barplot(x="video_count", y="channel_title", data=cdf, palette="rocket", ax=
_ = ax.set(xlabel="No. of videos", ylabel="Channel Name",)
plt.title('Top 10 Channels with the Most Trending Videos');
```

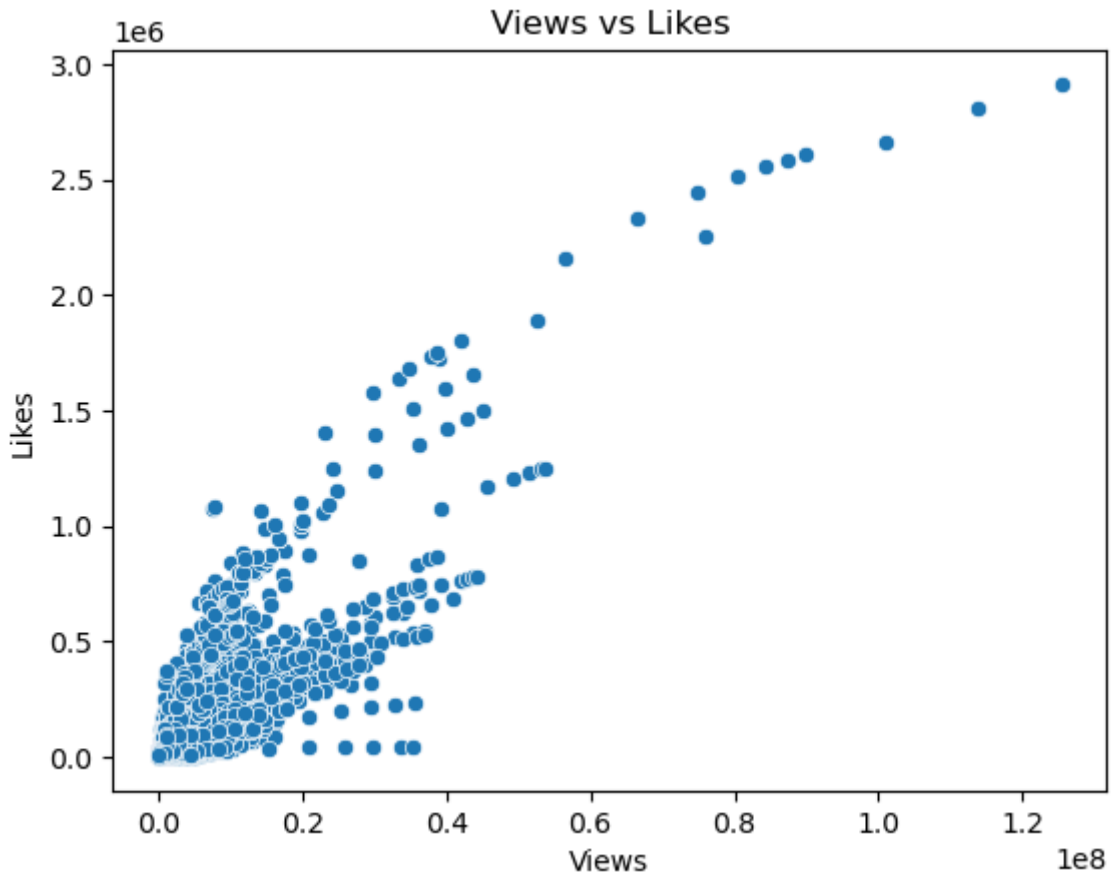


In [29]:

```
df['publish_time'] = pd.to_datetime(df['publish_time'])
df['publish_date'] = df['publish_time'].dt.date
video_count_by_date = df.groupby('publish_date').size()
plt.figure(figsize=(12, 6))
sns.lineplot(data=video_count_by_date)
plt.title("Videos Published Over Time")
plt.xlabel('Publish Date')
plt.ylabel('Number of Videos')
plt.xticks(rotation=45)
plt.show()
```



In [30]:
```
#Scatter plot between 'views' and 'likes'
sns.scatterplot(data=df, x= 'views', y='likes')
plt.title('Views vs Likes')
plt.xlabel('Views')
plt.ylabel('Likes')
plt.show()
```



In [31]:
```
plt.figure(figsize = (14,8))
plt.subplots_adjust(wspace = 0.2, hspace = 0.4, top = 0.9)

plt.subplot(2,2,1)
g = sns.countplot(x='comments_disabled', data=df)
g.set_title("Comments Disabled", fontsize=16)
```

```
plt.subplot(2,2,2)
g1 = sns.countplot(x='ratings_disabled', data=df)
g1.set_title("Rating Disabled", fontsize=16)

plt.subplot(2,2,3)
g2 = sns.countplot(x='video_error_or_removed', data=df)
g2.set_title("Video Error or Removed", fontsize=16)
plt.show()
```