In [ ]:
```
#Task-4
'''Create visualizations to understand the
distribution of variables, identify outliers,
and check for correlations between variables.'''
```

In [ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
data = pd.read_csv('USvideos.csv')
```

In [4]:
```
data.shape
```

Out[4]:
```
(40949, 16)
```

In [5]:
```
data.head(5)
```

Out[5]:

	video_id	trending_date	title	channel_title	category_id	publish_time	
0	2kyS6SvSYSE	17.14.11	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z	
1	1ZAPwfrtAFY	17.14.11	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13T07:30:00.000Z	last week tonigh
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	superman|"rudy"|"m
3	puqaWrEC7tY	17.14.11	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13T11:00:04.000Z	rhett and linl
4	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga	24	2017-11-12T18:01:41.000Z	ryan|"higa"|"h

In [8]:
```
data = data.drop_duplicates()
```

In [9]:
```
data.describe()
```

Out[9]:

	category_id	views	likes	dislikes	comment_count
count	40901.000000	4.090100e+04	4.090100e+04	4.090100e+04	4.090100e+04
mean	19.970588	2.360678e+06	7.427173e+04	3.711722e+03	8.448567e+03
std	7.569362	7.397719e+06	2.289999e+05	2.904624e+04	3.745139e+04
min	1.000000	5.490000e+02	0.000000e+00	0.000000e+00	0.000000e+00
25%	17.000000	2.419720e+05	5.416000e+03	2.020000e+02	6.130000e+02
50%	24.000000	6.810640e+05	1.806900e+04	6.300000e+02	1.855000e+03
75%	25.000000	1.821926e+06	5.533800e+04	1.936000e+03	5.752000e+03
max	43.000000	2.252119e+08	5.613827e+06	1.674420e+06	1.361580e+06

In [10]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 40901 entries, 0 to 40948
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   video_id                40901 non-null  object
 1   trending_date           40901 non-null  object
 2   title                   40901 non-null  object
 3   channel_title           40901 non-null  object
 4   category_id             40901 non-null  int64 
 5   publish_time            40901 non-null  object
 6   tags                    40901 non-null  object
 7   views                   40901 non-null  int64 
 8   likes                   40901 non-null  int64 
 9   dislikes                40901 non-null  int64 
 10  comment_count           40901 non-null  int64 
 11  thumbnail_link          40901 non-null  object
 12  comments_disabled       40901 non-null  bool  
 13  ratings_disabled        40901 non-null  bool  
 14  video_error_or_removed  40901 non-null  bool  
 15  description             40332 non-null  object
dtypes: bool(3), int64(5), object(8)
memory usage: 3.2+ MB
```

In [11]:
```
columns_to_remove = ['thumbnail_link','description']
data = data.drop(columns = columns_to_remove)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 40901 entries, 0 to 40948
Data columns (total 14 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   video_id                40901 non-null  object
 1   trending_date           40901 non-null  object
 2   title                   40901 non-null  object
 3   channel_title           40901 non-null  object
 4   category_id             40901 non-null  int64 
 5   publish_time            40901 non-null  object
 6   tags                    40901 non-null  object
 7   views                   40901 non-null  int64 
 8   likes                   40901 non-null  int64 
 9   dislikes                40901 non-null  int64 
 10  comment_count           40901 non-null  int64 
 11  comments_disabled       40901 non-null  bool  
 12  ratings_disabled        40901 non-null  bool  
 13  video_error_or_removed  40901 non-null  bool  
dtypes: bool(3), int64(5), object(6)
memory usage: 2.9+ MB
```

In [12]:
```
from datetime import datetime
import datetime
```

In [14]:
```
data['trending_date'] = data['trending_date'].apply(lambda x : datetime.datetime.strptim
data.head(3)
```

Out[14]:

	video_id	trending_date	title	channel_title	category_id	publish_time
0	2kyS6SvSYSE	2017-11-14	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13T17:13:01.000Z

1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13T07:30:00.000Z	last week tonight t
2	5qpjK5DgCt4	2017-11-14	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z	superman|"rudy"|"ma

In [15]:
```
data['publish_time'] = pd.to_datetime(data['publish_time'])
data['publish_month'] = data['publish_time'].dt.month
data['publish_day'] = data['publish_time'].dt.day
data['publish_hour'] = data['publish_time'].dt.hour
data.head(2)
```

Out[15]:

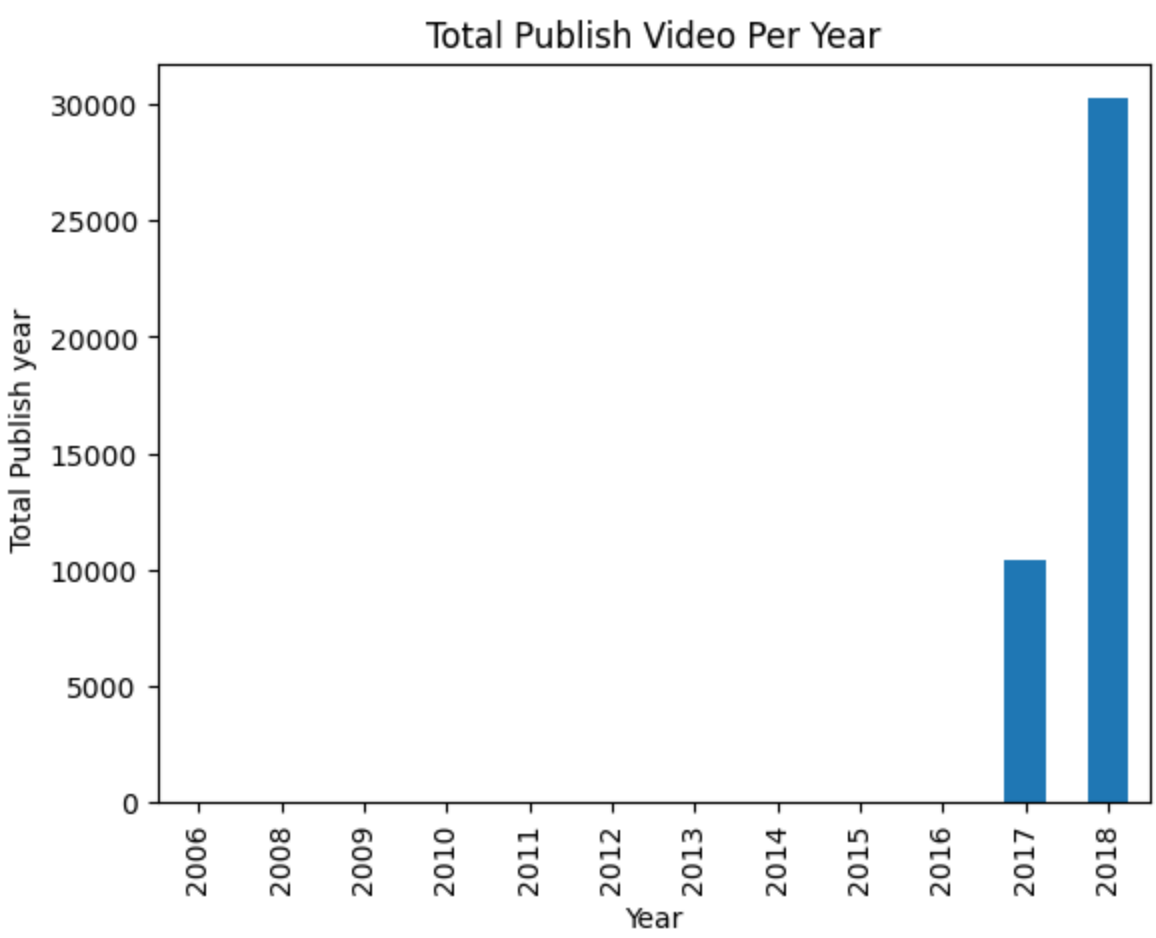
	video_id	trending_date	title	channel_title	category_id	publish_time	tags	vie
0	2kyS6SvSYSE	2017-11-14	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13 17:13:01+00:00	SHANtell martin	7483
1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13 07:30:00+00:00	last week tonight trump presidency|"last week ...	24187

In [16]:
```
print(sorted(data['category_id'].unique()))
```

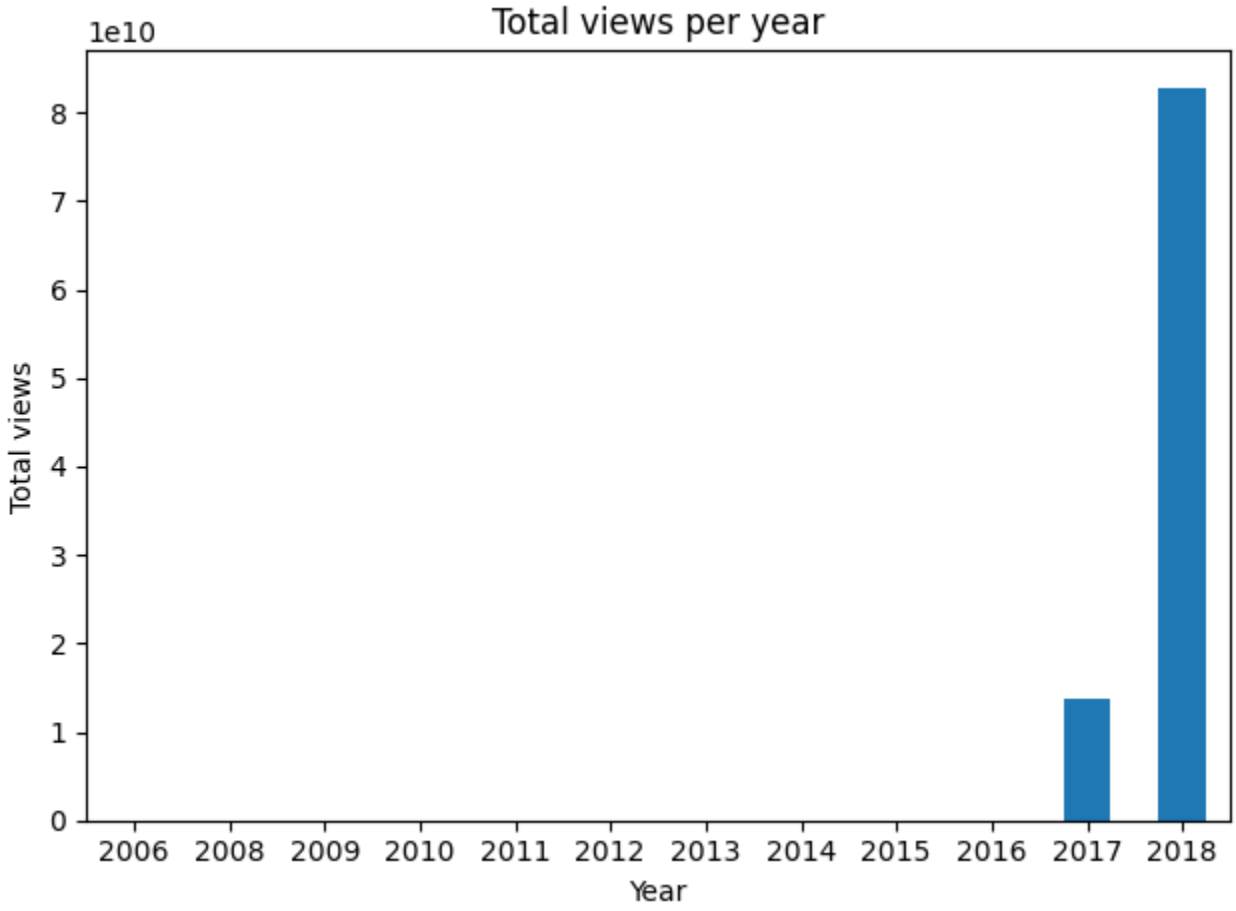
```
[1, 2, 10, 15, 17, 19, 20, 22, 23, 24, 25, 26, 27, 28, 29, 43]
```

In [17]:
```
category_names = {
1: 'Film and Animation',
2: 'Autos and Vehicles',
10: 'Music',
15: 'Pets and Animals',
17: 'Sports',
19: 'Travel and Events',
20: 'Gaming',
22: 'People and Blogs',
23: 'Comedy',
24: 'Entertainment',
25: 'News and Politics',
26: 'How to and Style',
27: 'Education',
28: 'Science and Technology',
29: 'Non Profits and Activism',
30: 'Movies',
43: 'Shows'
}
data['category_name'] = data['category_id'].map(category_names)
```

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

#Create a bar chart.
yearly_counts.plot(kind= 'bar', xlabel= 'Year', ylabel= ' Total Publish year', title= 'T

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

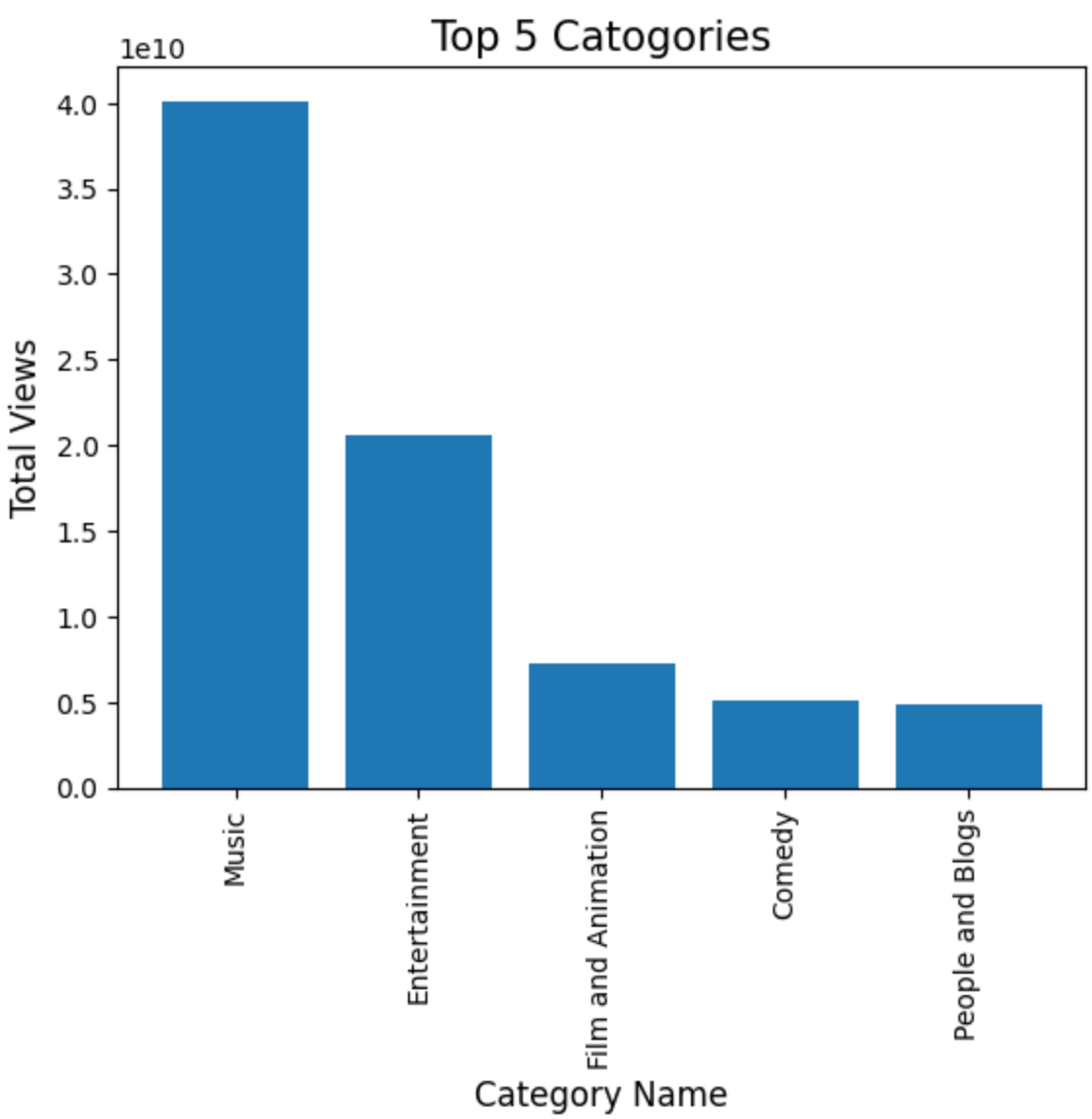


In [21]:
```
#Group by year and sum the views for each year.
yearly_views = data.groupby('year')['views'].sum()

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

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



In [22]:

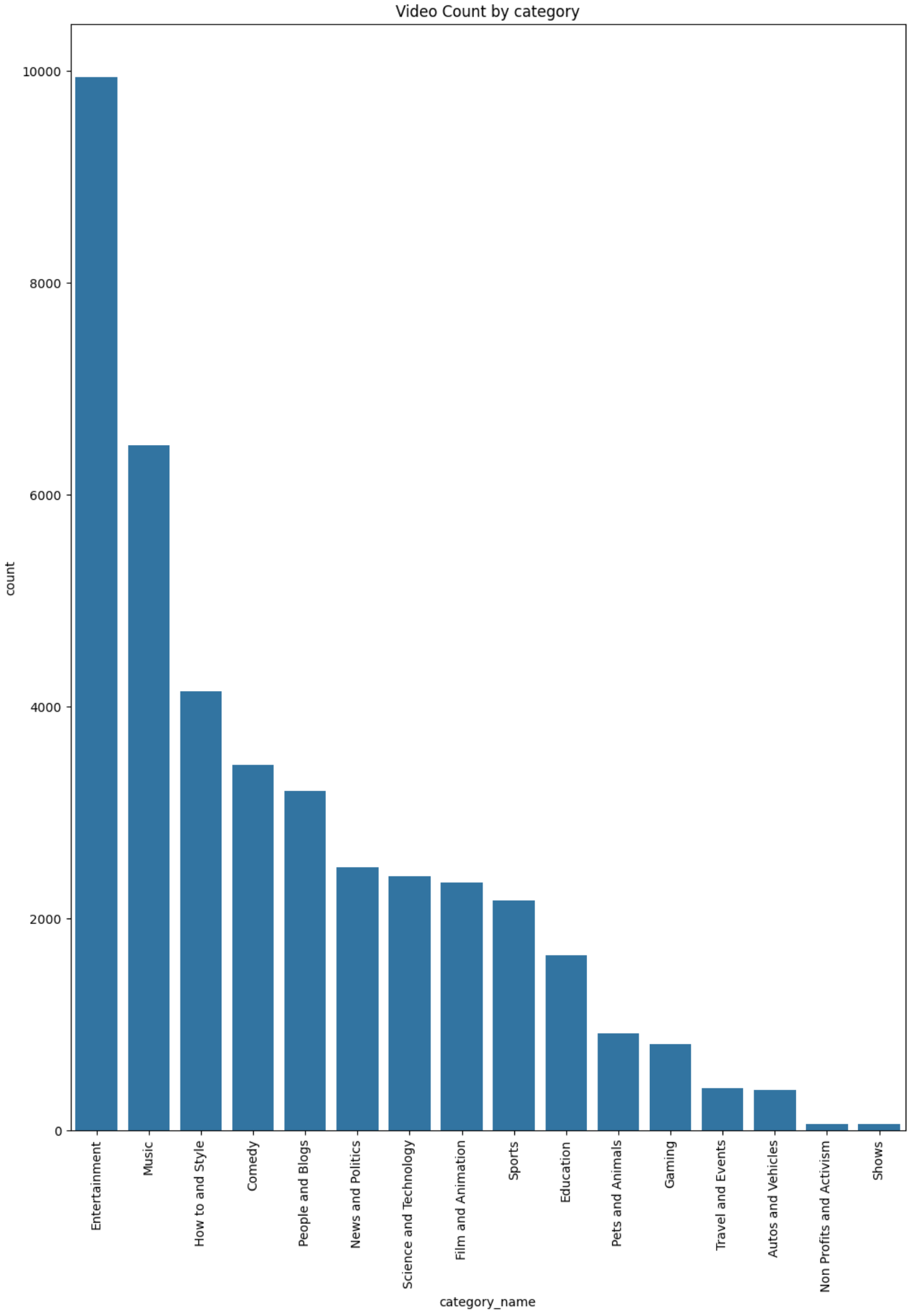
```
#Group the data by 'category_name' and calculate the sum of 'views' in each category.
category_views = data.groupby('category_name')['views'].sum().reset_index()

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

#Creating 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.xticks(rotation=90)
plt.ylabel('Total Views', fontsize = 12)
plt.title('Top 5 Catogories', fontsize = 15)
plt.show()
```



In [25]:
```
%pip install seaborn
```

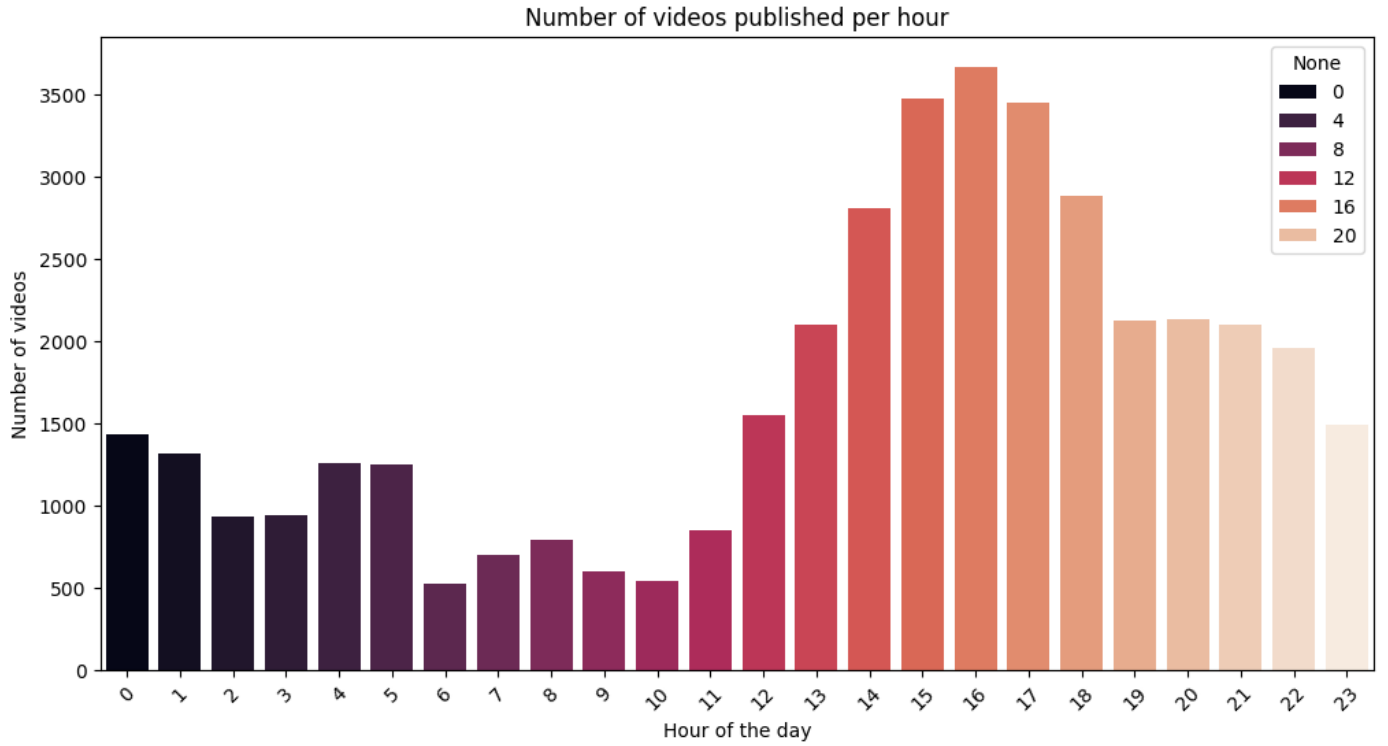
In [26]:
```
import seaborn as sns
```

In [27]:
```
plt.figure(figsize=(12,16))
sns.countplot(x='category_name', data=data, order=data['category_name'].value_counts().i
plt.xticks(rotation=90)
plt.title('Video Count by category')
plt.show()
```



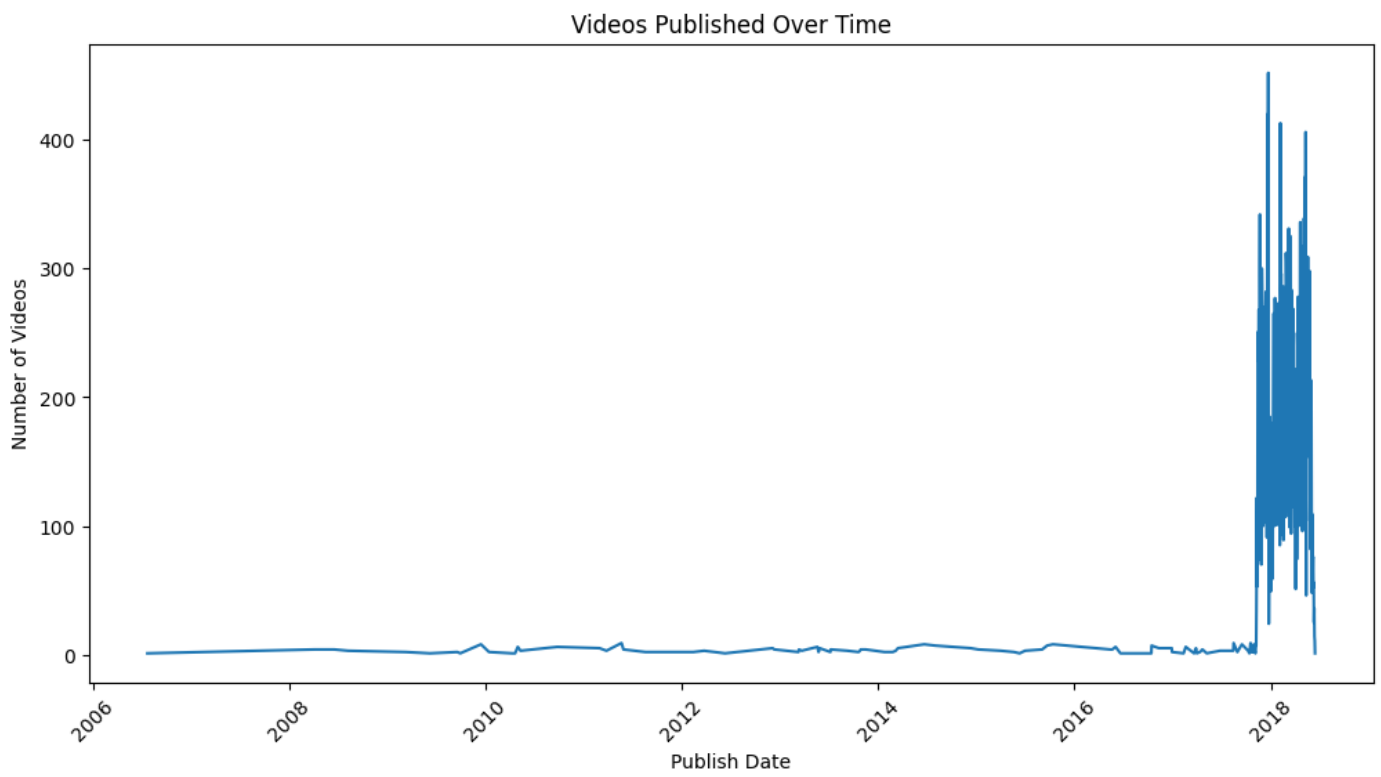
In [28]:
```
# Count the number of videos published per hour.
videos_per_hour = data['publish_hour'].value_counts().sort_index()
```

```
# Create a bar plot.
plt.figure(figsize=(12,6))
sns.barplot(x=videos_per_hour.index, y=videos_per_hour.values,hue=videos_per_hour.index,
plt.title('Number of videos published per hour')
plt.xlabel('Hour of the day')
plt.ylabel('Number of videos')
plt.xticks(rotation=45)
plt.show()
```

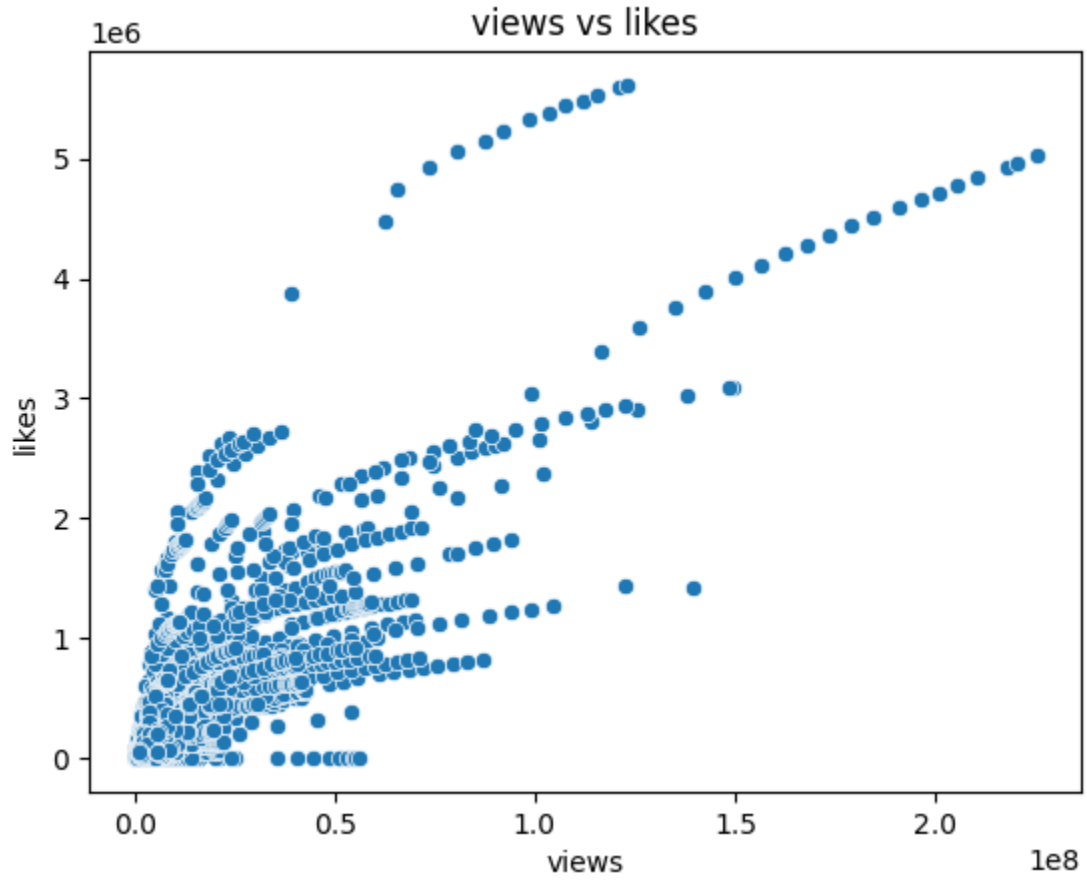


In [29]:

```
data['publish_time'] = pd.to_datetime(data['publish_time'])
data['publish_date'] = (data['publish_time']).dt.date
video_count_by_date = data.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=data,x='views', y='likes')
plt.title('views vs likes')
plt.xlabel('views')
plt.ylabel('likes')
plt.show()
```



In [ ]:
```
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 =data)
g.set_title("comments_disabled", fontsize=16)
plt.subplot(2,2,2)
g1=sns.countplot(x= 'ratings_disabled',data =data)
g1.set_title("ratings_disabled", fontsize=16)
plt.subplot(2,2,3)
g2=sns.countplot(x= 'video_error_or_removed',data =data)
g2.set_title("video_error_or_removed", fontsize=16)
plt.show()
```