## This task involves performing exploratory data analysis on a dataset.Create visualizations to understand the distribution of variables, identify outliers, and check for correlations between variables.

*Importing Libraries*

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


from datetime import datetime
import datetime
```

*Reading Data*

```
# Reading a CSV file
data = pd.read_csv("/content/USvideos.csv")
```

*Data Exploration*

```
data.shape
```

```
(40949, 16)
```

```
data.describe()
```

| | category_id | views | likes | dislikes | comment_count |
|---|---|---|---|---|---|
| **count** | 40949.000000 | 4.094900e+04 | 4.094900e+04 | 4.094900e+04 | 4.094900e+04 |
| **mean** | 19.972429 | 2.360785e+06 | 7.426670e+04 | 3.711401e+03 | 8.446804e+03 |
| **std** | 7.568327 | 7.394114e+06 | 2.288853e+05 | 2.902971e+04 | 3.743049e+04 |
| **min** | 1.000000 | 5.490000e+02 | 0.000000e+00 | 0.000000e+00 | 0.000000e+00 |
| **25%** | 17.000000 | 2.423290e+05 | 5.424000e+03 | 2.020000e+02 | 6.140000e+02 |
| **50%** | 24.000000 | 6.818610e+05 | 1.809100e+04 | 6.310000e+02 | 1.856000e+03 |
| **75%** | 25.000000 | 1.823157e+06 | 5.541700e+04 | 1.938000e+03 | 5.755000e+03 |
| **max** | 43.000000 | 2.252119e+08 | 5.613827e+06 | 1.674420e+06 | 1.361580e+06 |

```
data.head(3)
```

| | video_id | trending_date | title | channel_title | category_id | publish_ |
|---|---|---|---|---|---|---|
| **0** | 2kyS6SvSYSE | 17.14.11 | WE WANT TO TALK ABOUT OUR MARRIAGE | CaseyNeistat | 22 | 2017<br>13T17:13:01.( |
| **1** | 1ZAPwfrtAFY | 17.14.11 | The Trump Presidency: Last Week Tonight with J... | LastWeekTonight | 24 | 2017<br>13T07:30:00.( |
| **2** | 5qpjK5DgCt4 | 17.14.11 | Racist Superman \| Rudy Mancuso, King Bach & Le... | Rudy Mancuso | 23 | 2017<br>12T19:05:24.( |

Next steps: View recommended plots

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40949 entries, 0 to 40948
```

```
Data columns (total 16 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   video_id                40949 non-null  object
 1   trending_date           40949 non-null  object
 2   title                   40949 non-null  object
 3   channel_title           40949 non-null  object
 4   category_id             40949 non-null  int64 
 5   publish_time            40949 non-null  object
 6   tags                    40949 non-null  object
 7   views                   40949 non-null  int64 
 8   likes                   40949 non-null  int64 
 9   dislikes                40949 non-null  int64 
 10  comment_count           40949 non-null  int64 
 11  thumbnail_link          40949 non-null  object
 12  comments_disabled       40949 non-null  bool  
 13  ratings_disabled        40949 non-null  bool  
 14  video_error_or_removed  40949 non-null  bool  
 15  description             40379 non-null  object
dtypes: bool(3), int64(5), object(8)
memory usage: 4.2+ MB
```

*Data Cleaning*

```
# Removing the duplicate rows from the DataFrame
data = data.drop_duplicates()
```

```
# Droping specified columns from the DataFrame
columns_to_remove = ['thumbnail_link','description']
data = data.drop(columns = columns_to_remove)
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 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: 3.9+ MB
```

*Data Transformation*

```
data['trending_date'] = data['trending_date'].apply(lambda x : datetime.datetime.strptime(x,'%y.%d.%m'))
data.head(3)
```

| | video_id | trending_date | title | channel_title | category_id | publish_ |
|---|---|---|---|---|---|---|
| **0** | 2kyS6SvSYSE | 2017-11-14 | WE WANT TO TALK ABOUT OUR MARRIAGE | CaseyNeistat | 22 | 2017<br>13T17:13:01. |
| **1** | 1ZAPwfrtAFY | 2017-11-14 | The Trump Presidency: Last Week Tonight with J... | LastWeekTonight | 24 | 2017<br>13T07:30:00. |
| **2** | 5qpjK5DgCt4 | 2017-11-14 | Racist Superman \| Rudy Mancuso, King Bach & Le... | Rudy Mancuso | 23 | 2017<br>12T19:05:24. |

Next steps: View recommended plots

```
# Extracing components like month, day, and hour from the 'publish_time' column
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(3)
```

| | video_id | trending_date | title | channel_title | category_id | publish_tir |
|---|---|---|---|---|---|---|
| **0** | 2kyS6SvSYSE | 2017-11-14 | WE WANT TO TALK ABOUT OUR MARRIAGE | CaseyNeistat | 22 | 2017-11-1<br>17:13:01+00:0 |
| **1** | 1ZAPwfrtAFY | 2017-11-14 | The Trump Presidency: Last Week Tonight with J... | LastWeekTonight | 24 | 2017-11-1<br>07:30:00+00:0 |
| **2** | 5qpjK5DgCt4 | 2017-11-14 | Racist Superman \| Rudy Mancuso, King Bach & Le... | Rudy Mancuso | 23 | 2017-11-1<br>19:05:24+00:0 |

Next steps: View recommended plots

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

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

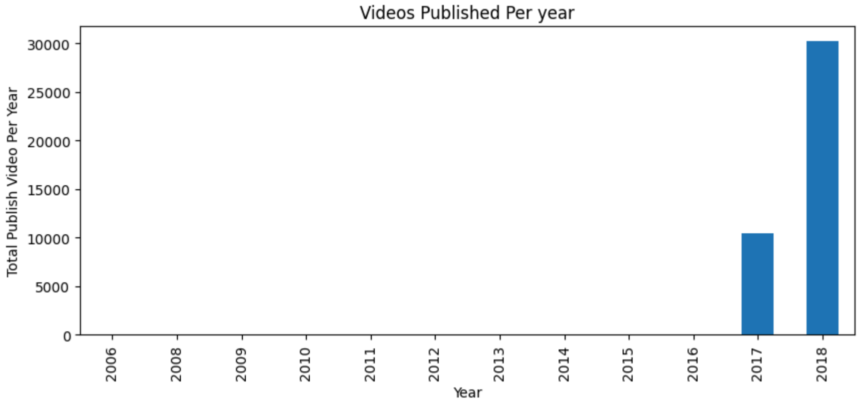
```
# Maping the category IDs to their corresponding names
data['category_name'] = np.nan
data.loc[(data['category_id'] == 1), 'category_name'] = 'Film and Animation'
data.loc[(data['category_id'] == 2), 'category_name'] = 'Autos and Vehicles'
data.loc[(data['category_id'] == 10), 'category_name'] = 'Music'
data.loc[(data["category_id"] == 15), "category_name"] = 'Pets and Animals'
data.loc[(data ["category_id"] == 17 ), "category_name"] = 'Sports'
data.loc[(data["category_id"] == 19), "category_name"] =  'Travel and Events'
data.loc[(data["category_id"] == 20 ), "category_name"] = 'Gaming'
data.loc[(data["category_id"] == 22 ), "category_name"] = 'People and Blogs'
data.loc[(data["category_id"]== 23), "category_name"] = 'Comedy'
data.loc[(data["category_id"]== 24), "category_name"] = 'Entertainment'
data.loc[(data["category_id"] == 25), "category_name"] = 'News and Politics'
data.loc[(data["category_id"] == 26), "category_name"] = 'How to and Style'
data.loc[(data["category_id"]== 27), "category_name"] =  'Education'
data.loc[(data["category_id"] == 28), "category_name"] = 'Science and Tech'
data.loc[(data["category_id"] == 29), "category_name"] = 'Non Profits'
data.loc[(data["category_id"] == 30), "category_name"] = 'Movies'
data.loc[(data["category_id"] == 43), "category_name"] = 'Shows'
data.head(3)
```

| | video_id | trending_date | title | channel_title | category_id | publish_tir |
|---|---|---|---|---|---|---|
| **0** | 2kyS6SvSYSE | 2017-11-14 | WE WANT TO TALK ABOUT OUR MARRIAGE | CaseyNeistat | 22 | 2017-11-1<br>17:13:01+00:0 |
| **1** | 1ZAPwfrtAFY | 2017-11-14 | The Trump Presidency: Last Week Tonight with J... | LastWeekTonight | 24 | 2017-11-1<br>07:30:00+00:0 |
| **2** | 5qpjK5DgCt4 | 2017-11-14 | Racist Superman \| Rudy Mancuso, King Bach & Le... | Rudy Mancuso | 23 | 2017-11-1<br>19:05:24+00:0 |

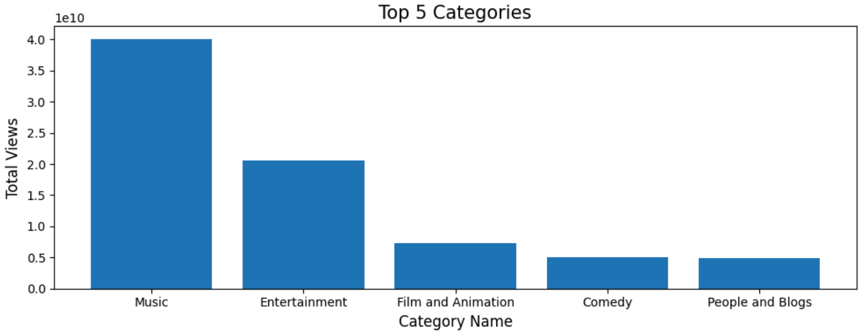
Next steps: View recommended plots

*Data Visualization*

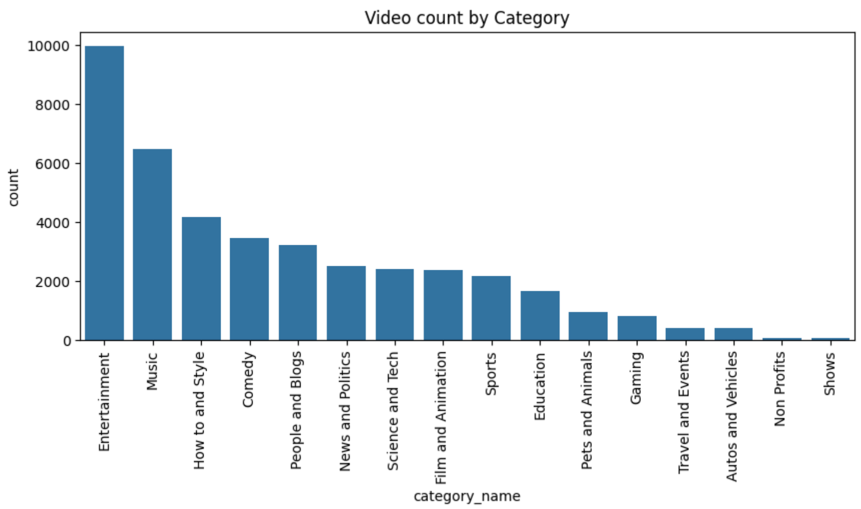
```
# Bar plot of the total number of videos published per year
plt.figure(figsize = (10,4))
data['year'] = data['publish_time'].dt.year
yearly_counts = data.groupby('year')['video_id'].count()
plt.title("Videos Published Per year")
yearly_counts.plot(kind = 'bar', xlabel = 'Year', ylabel = 'Total Publish Video Per Year')
plt.show()
```



```
# Bar plot of the total views for the top 5 video categories
plt.figure(figsize = (10,4))
category_views = data.groupby('category_name')['views'].sum().reset_index()
top_categories = category_views.sort_values(by='views', ascending = False).head(5)
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()
```

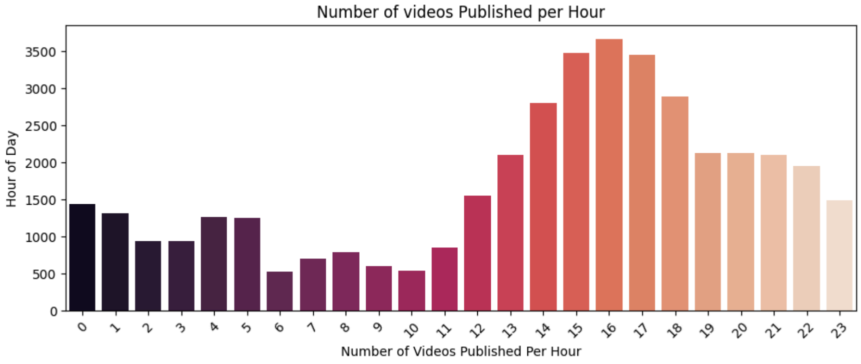


```
# Count plot of the distribution of videos across different categories
plt.figure(figsize = (10,4))
sns.countplot(x = 'category_name', data=data, order=data['category_name'].value_counts().index)
plt.xticks(rotation=90)
plt.title('Video count by Category')
plt.show()
```

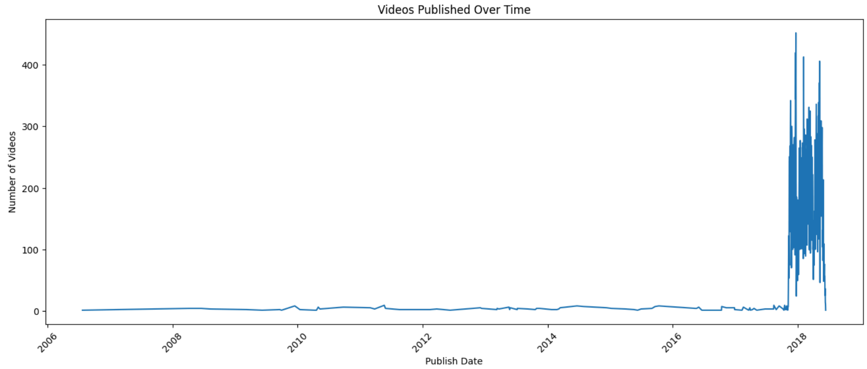


```
# Bar plot for the number of videos published per hour
videos_per_hour = data['publish_hour'].value_counts().sort_index()
plt.figure(figsize=(11,4))
sns.barplot(x= videos_per_hour.index, y = videos_per_hour.values, palette = 'rocket')
plt.title('Number of videos Published per Hour')
plt.xlabel('Number of Videos Published Per Hour')
plt.ylabel('Hour of Day')
plt.xticks(rotation = 45)
plt.show()
```

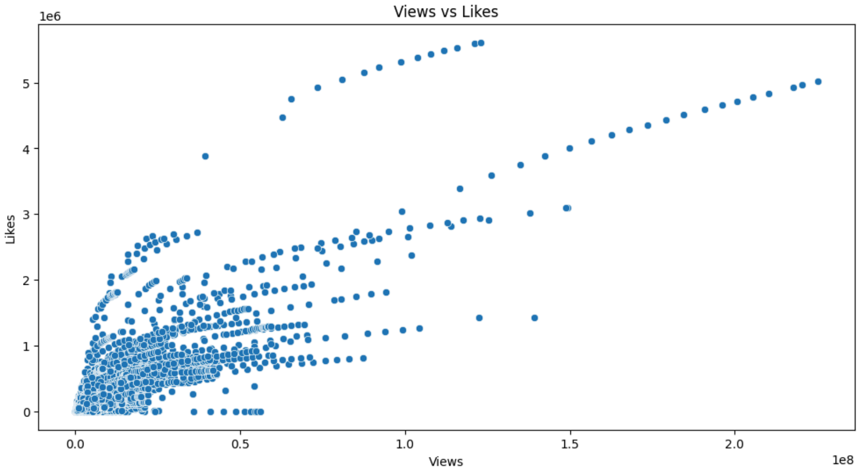
```
<ipython-input-244-f48f0c236caf>:4: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14

  sns.barplot(x= videos_per_hour.index, y = videos_per_hour.values, palette = 'rocke
```



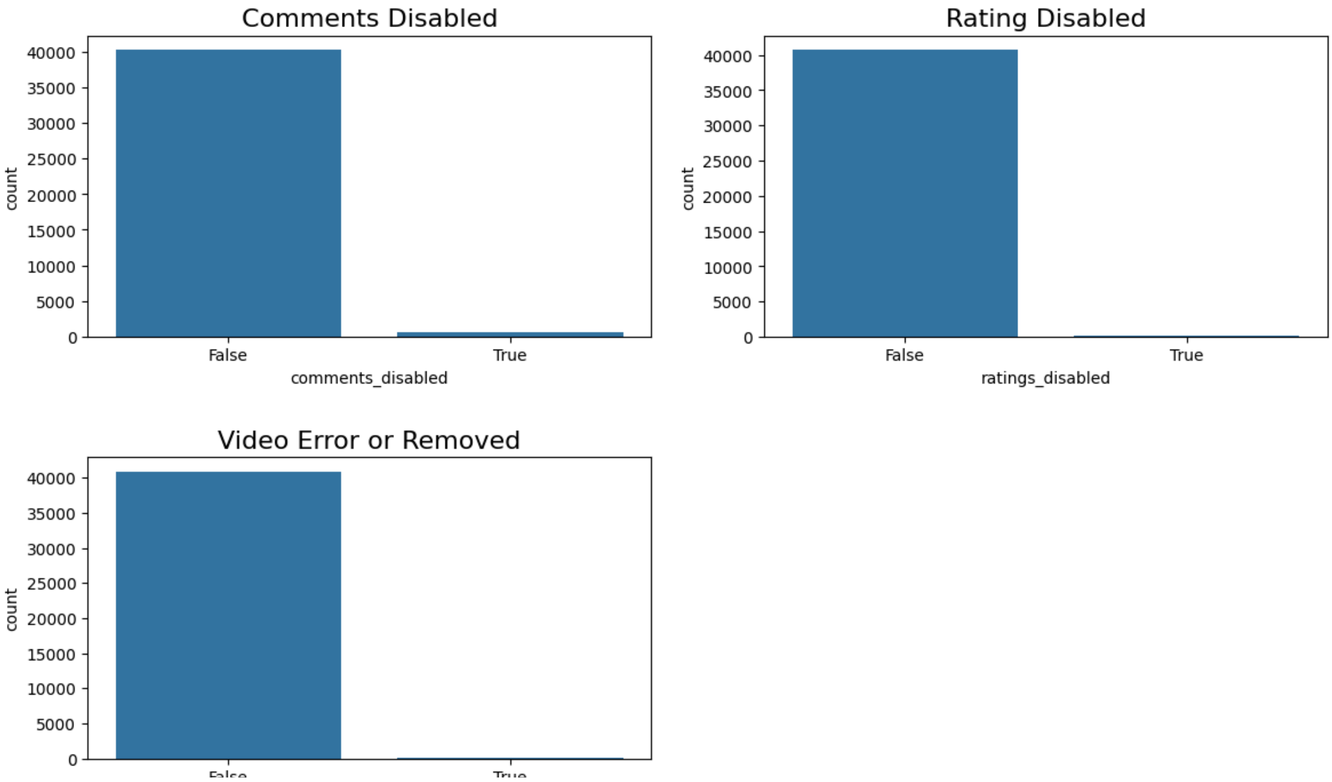
```
# Line plot for the trend of videos published over time
plt.figure(figsize = (16,6))
data['publish_time'] = pd.to_datetime(data['publish_time'])
data['publish_date'] = data['publish_time'].dt.date
video_count_by_date = data.groupby('publish_date').size()
sns.lineplot(data = video_count_by_date)
plt.title("Videos Published Over Time")
plt.xlabel('Publish Date')
plt.ylabel('Number of Videos')
plt.xticks(rotation = 45)
plt.show()
```



```
# Scatter plot for the relationship between views and likes
plt.figure(figsize = (12,6))
sns.scatterplot(data = data, x = 'views', y = 'likes')
plt.title('Views vs Likes')
plt.xlabel('Views')
plt.ylabel('Likes')
plt.show()
```



```
# Subplots for counts of videos with comments disabled, ratings disabled, and videos with errors or removed.
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("Rating 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()
```



Comments Disabled

Rating Disabled

Video Error or Removed

*Correlation Calculation*

```
# Calculates the correlation coefficient between 'views' and 'likes'
corr_matrix = data['views'].corr(data['likes'])
corr_matrix
```

```
0.8491785476230508
```