TASK-4

Description:

This task involves performing exploratory data analysis on a dataset.

Responsibility:

Create visualizations to understand the distribution of variables, identify outliers, and check for correlations between variables.

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

```
df = pd.read_csv("//content//USvideos.csv")
```

```
df.head()
```

	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
2	5qpjK5DgCt4	17.14.11	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z
3	puqaWrEC7tY	17.14.11	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13T11:00:04.000Z
4	d380meD0W0M	17.14.11	I Dare You: GOING BALD!?	nigahiga	24	2017-11-12T18:01:41.000Z

Next steps: Generate code with df View recommended plots

```
df.shape
```

```
(40949, 16)
```

```
df = df.drop_duplicates()
df.shape
```

```
(40901, 16)
```

```
df.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
```

```
df.describe()
```

	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

```
columns_to_remove = ['thumbnail_link', 'description']
df = df.drop(columns=columns_to_remove)
df.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
```

```
from datetime import datetime
```

```
import datetime
```

```
df["trending_date"] = pd.to_datetime(df["trending_date"], format="%y.%d.%m")
df.head(3)
```

	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
2	5qpjK5DgCt4	2017-11-14	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12T19:05:24.000Z

Next steps: Generate code with df View recommended plots

```
df["publish_time"] = pd.to_datetime(df["publish_time"])
df.head(2)
```

	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	2017-11-13 17:13:01+00:00	
1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13 07:30:00+00:00	pr

Next steps: Generate code with df View recommended plots

```
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)
```

	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	2017-11-13 17:13:01+00:00	
1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13 07:30:00+00:00	pr

Next steps: Generate code with df View recommended plots

```
print(sorted(df["category_id"].unique()))
```

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

Suggested code may be subject to a license | UtkarshK10/Youtube-Data-Analysis

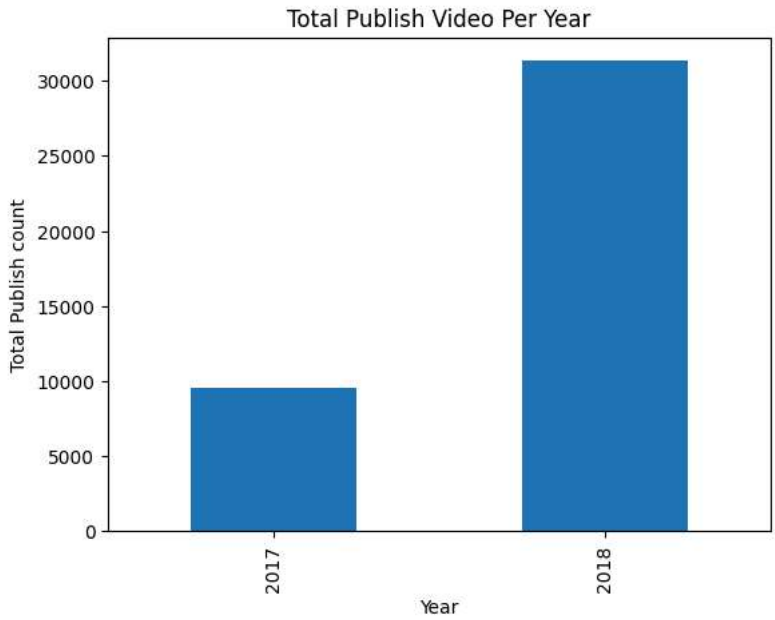
```
df["category_name"] = np.nan
df.loc[df["category_id"] == 1, "category_name"] = "Film & Animation"
df.loc[df["category_id"] == 2, "category_name"] = "Autos & Vehicles"
df.loc[df["category_id"] == 10, "category_name"] = "Music"
df.loc[df["category_id"] == 15, "category_name"] = "Pets & Animals"
```

```
df.loc[df["category_id"] == 17, "category_name"] = "Sports"
df.loc[df["category_id"] == 19, "category_name"] = "Travel & Events"
df.loc[df["category_id"] == 20, "category_name"] = "Gaming"
df.loc[df["category_id"] == 22, "category_name"] = "People & 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 & Politics"
df.loc[df["category_id"] == 26, "category_name"] = "Howto & Style"
df.loc[df["category_id"] == 27, "category_name"] = "Education"
df.loc[df["category_id"] == 28, "category_name"] = "Science & Technology"
df.loc[df["category_id"] == 29, "category_name"] = "Nonprofits & Activism"
df.loc[df["category_id"] == 30, "category_name"] = "Movies"
df.loc[df["category_id"] == 43, "category_name"] = "Shows"
df.head()
```

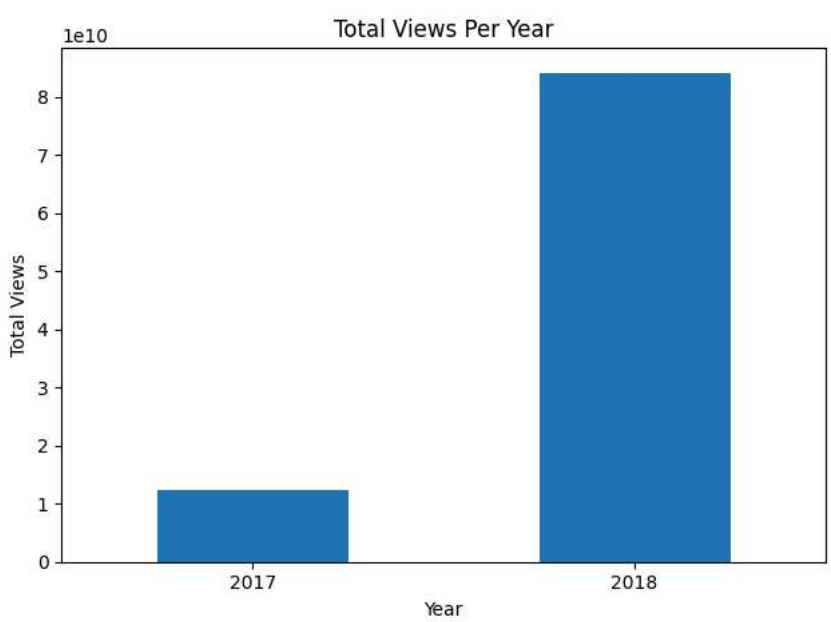
	video_id	trending_date	title	channel_title	category_id	publish_time
0	2kyS6SvSYSE	2017-11-14	WE WANT TO TALK ABOUT OUR MARRIAGE	CaseyNeistat	22	2017-11-13 17:13:01+00:00
1	1ZAPwfrtAFY	2017-11-14	The Trump Presidency: Last Week Tonight with J...	LastWeekTonight	24	2017-11-13 07:30:00+00:00
2	5qpjK5DgCt4	2017-11-14	Racist Superman | Rudy Mancuso, King Bach & Le...	Rudy Mancuso	23	2017-11-12 19:05:24+00:00
3	puqaWrEC7tY	2017-11-14	Nickelback Lyrics: Real or Fake?	Good Mythical Morning	24	2017-11-13 11:00:04+00:00
4	d380meD0W0M	2017-11-14	I Dare You: GOING BALD!?	nigahiga	24	2017-11-12 18:01:41+00:00

Next steps: Generate code with df View recommended plots

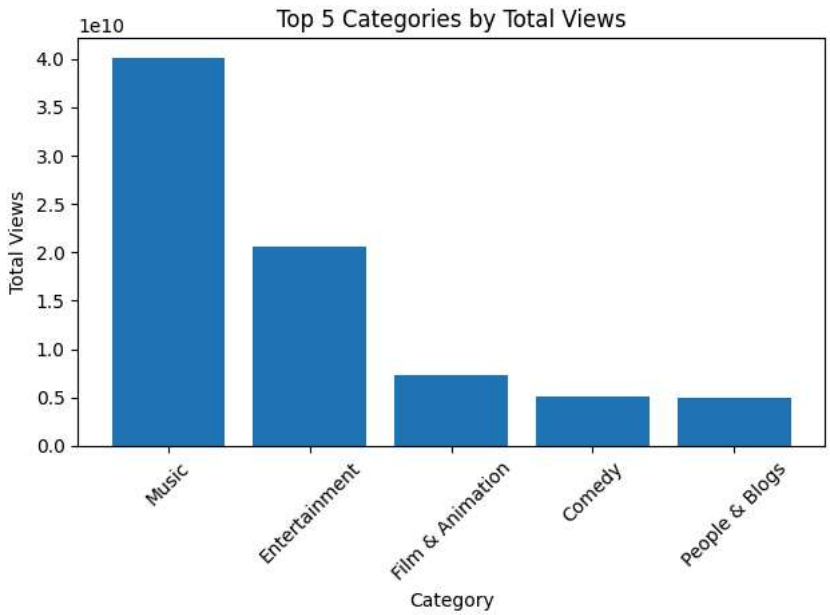
```
df["year"] = df["trending_date"].dt.year
yearly_counts = df.groupby("year")["video_id"].count()

#create a bar chart
yearly_counts.plot(kind="bar" , xlabel="Year", ylabel="Total Publish count", title="Total Publish Video Per Year")
plt.show()
```



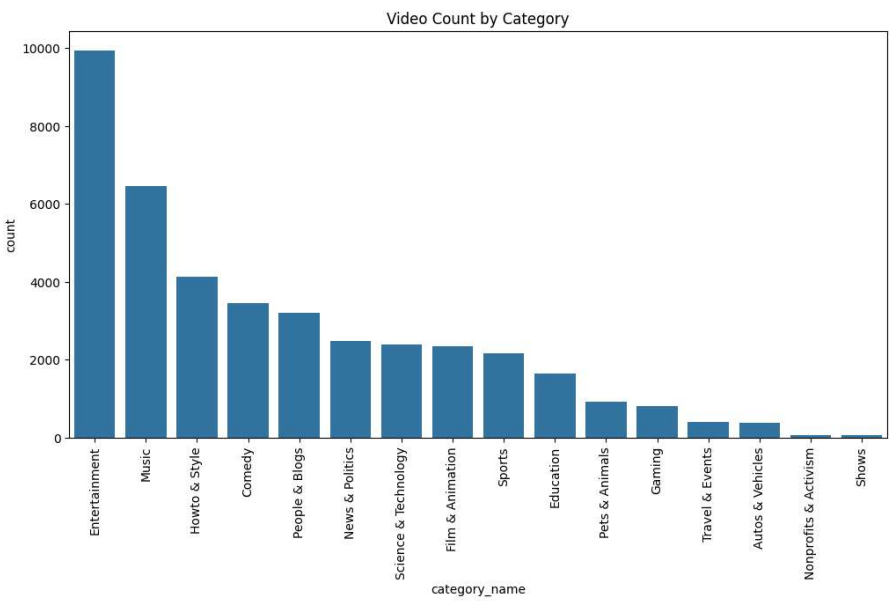

```
yearly_views = df.groupby("year")["views"].sum()

#create a bar chart
yearly_views.plot(kind="bar" , xlabel="Year", ylabel="Total Views", title="Total Views Per Year")
plt.xticks(rotation=0)
plt.tight_layout()
plt.show()
```




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

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

# Create a bar chart to visualize the top 5 categories
plt.bar(top_category["category_name"], top_category["views"])
plt.xlabel("Category")
plt.ylabel("Total Views")
plt.title("Top 5 Categories by Total Views")
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



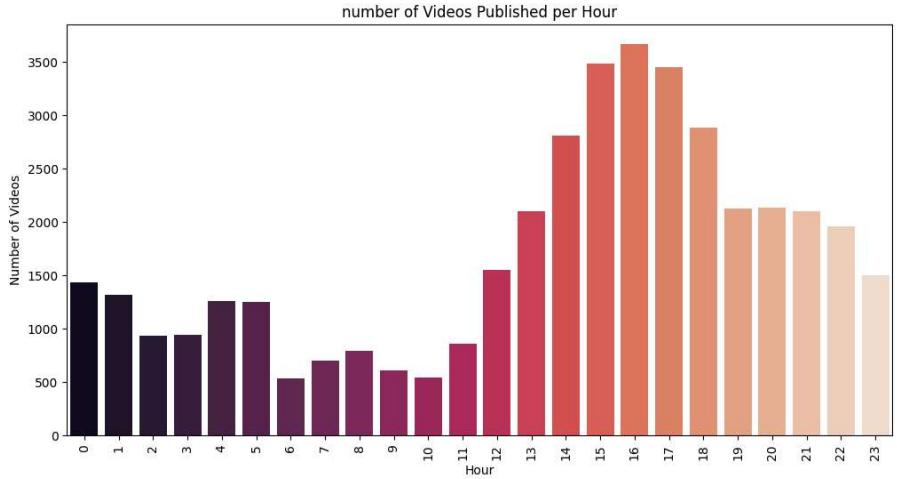

```
plt.figure(figsize=(12, 6))
sns.countplot(x="category_name", data=df, order=df["category_name"].value_counts().index)
plt.xticks(rotation=90)
plt.title("Video Count by Category")
plt.show()
```



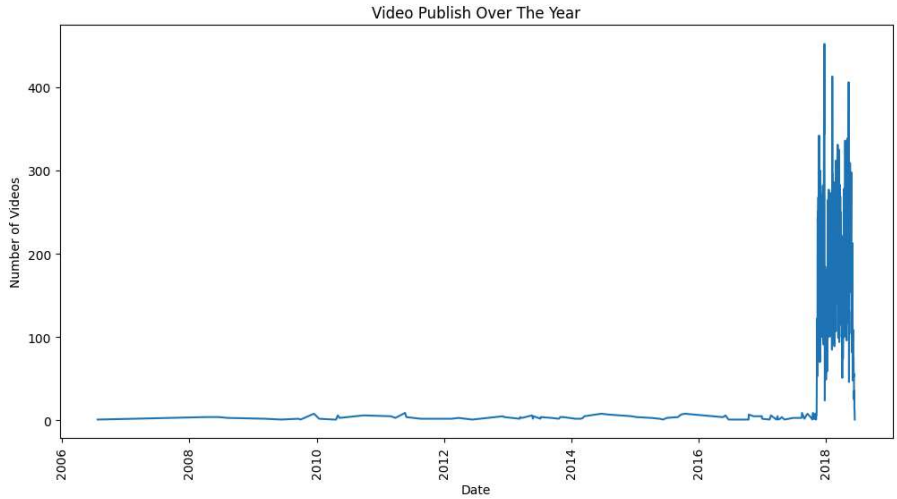

```
#Count the number of Videos Published per Hour
videos_per_hour = df["publish_hour"].value_counts().sort_index()

#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")
plt.ylabel("Number of Videos")
plt.xticks(rotation=90)
plt.show()
```

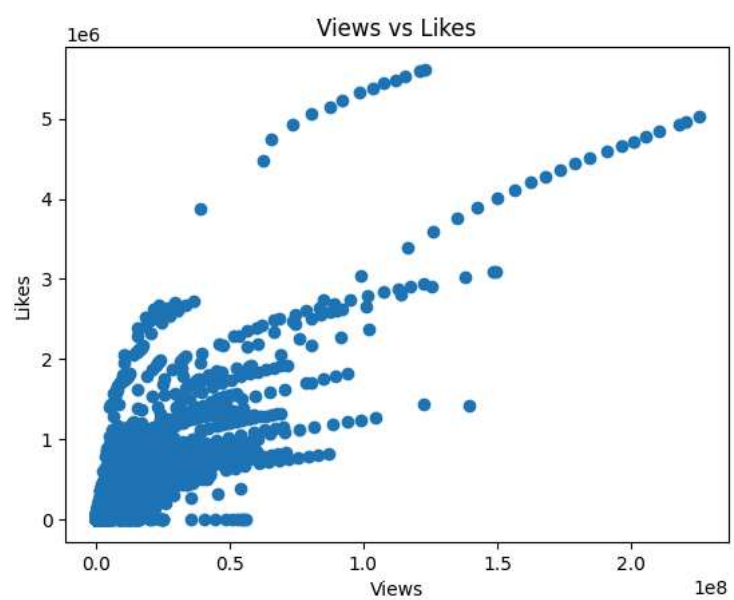
```
<ipython-input-39-b6b10a4d4c99>:6: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0.

  sns.barplot(x=videos_per_hour.index, y= videos_per_hour.values, palette="rocket")
```




```
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("Video Publish Over The Year")
plt.xlabel("Date")
plt.ylabel("Number of Videos")
plt.xticks(rotation=90)
plt.show()
```




```
#scatter plot between "views" and "likes"
plt.scatter(df["views"], df["likes"])
plt.xlabel("Views")
plt.ylabel("Likes")
plt.title("Views vs Likes")
plt.show()
```




```
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")
plt.subplot(2,2,2)
g=sns.countplot(x="ratings_disabled",data=df)
g.set_title("Ratings Disabled")
plt.subplot(2,2,3)
g=sns.countplot(x="video_error_or_removed",data=df)
g.set_title("Video Error Or Removed")
plt.show()
```

Comments Disabled Ratings Disabled