Ex.No.8 **Mini Project**

15.10.202 EDA on YouTube Trending Videos Dataset

AIM:

To analyze and visualize YouTube trending video data to uncover insights about viewer engagement, content performance, and publishing patterns.

ALGORITHM:

1. Import required Python libraries.
    - Load the YouTube dataset using pandas.
    - Convert publish_date and trending_date to datetime format.
    - Create new feature days_since_publish = difference between trending and publish dates.
    - Convert published_day_of_week to categorical type.
    - Apply log transformation to views, likes, dislikes, and comment_count.
    - Compute correlation between numeric variables.
2. Plot visualizations using Seaborn and Plotly:
3. Correlation heatmap
4. Scatter plot (Likes vs Views)
5. Time series (Views over time)
6. Bar chart (Published day vs Count)
7. Box plot (Views by Category)
8. Interactive bubble chart (Likes vs Comments)
    - Analyze the visual outputs to identify patterns and insights.

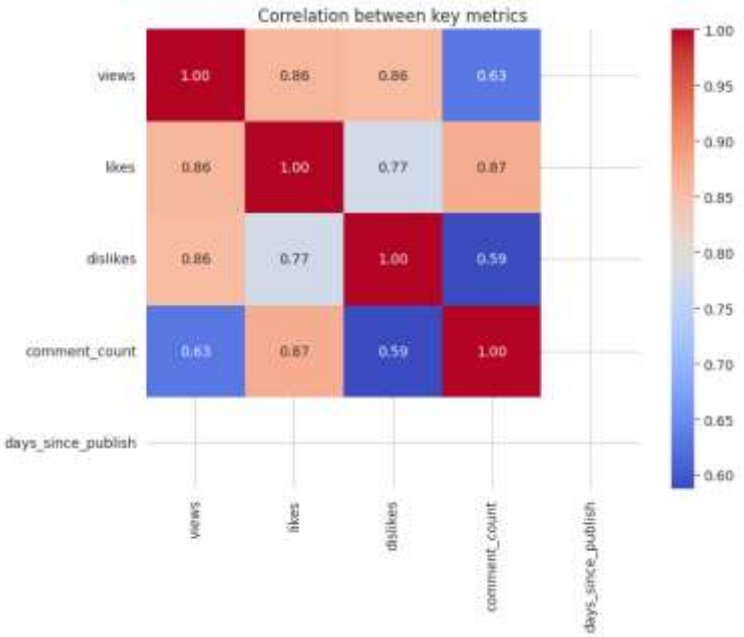
PROGRAM:

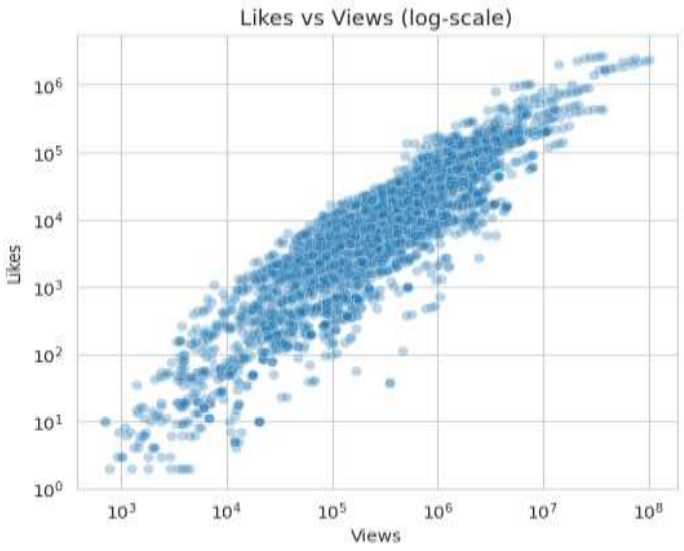
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px

sns.set_style("whitegrid")

df = pd.read_csv("youtube data.csv")
```

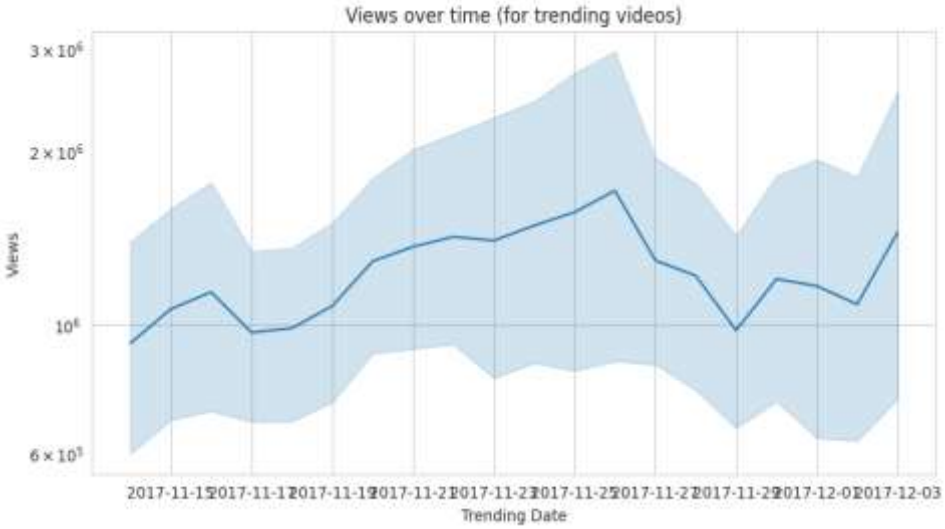
```
df['trending_date'] = pd.to_datetime(df['trending_date'], format='%y.%d.%m', errors='coerce')
df['publish_date'] = pd.to_datetime(df['publish_date'], format='%y.%d.%m', errors='coerce')

df['published_day_of_week'] = df['published_day_of_week'].astype('category')
df['days_since_publish'] = (df['trending_date'] - df['publish_date']).dt.days.clip(lower=0)

for col in ['views', 'likes', 'dislikes', 'comment_count']:
    df[f'{col}_log'] = np.log1p(df[col])

numeric_cols = ['views', 'likes', 'dislikes', 'comment_count', 'days_since_publish']
corr = df[numeric_cols].corr()
plt.figure(figsize=(8, 6))
sns.heatmap(corr, annot=True, fmt=".2f", cmap='coolwarm')
plt.title("Correlation between key metrics")
plt.show()

plt.figure(figsize=(6, 5))
sns.scatterplot(data=df, x='views', y='likes', alpha=0.3)
plt.xscale('log')
plt.yscale('log')
plt.xlabel("Views")
plt.ylabel("Likes")
plt.title("Likes vs Views (log-scale)")
plt.show()

plt.figure(figsize=(10, 5))
sns.lineplot(data=df.sort_values('trending_date'), x='trending_date', y='views')
plt.yscale('log')
plt.xlabel("Trending Date")
plt.ylabel("Views")
plt.title("Views over time (for trending videos)")
plt.show()

plt.figure(figsize=(8, 4))
sns.countplot(data=df, x='published_day_of_week',
order=df['published_day_of_week'].cat.categories)
plt.xlabel("Published Day of Week")
plt.ylabel("Count of Trending Videos")
plt.title("When (day of week) videos get published that go trending")
plt.show()
```

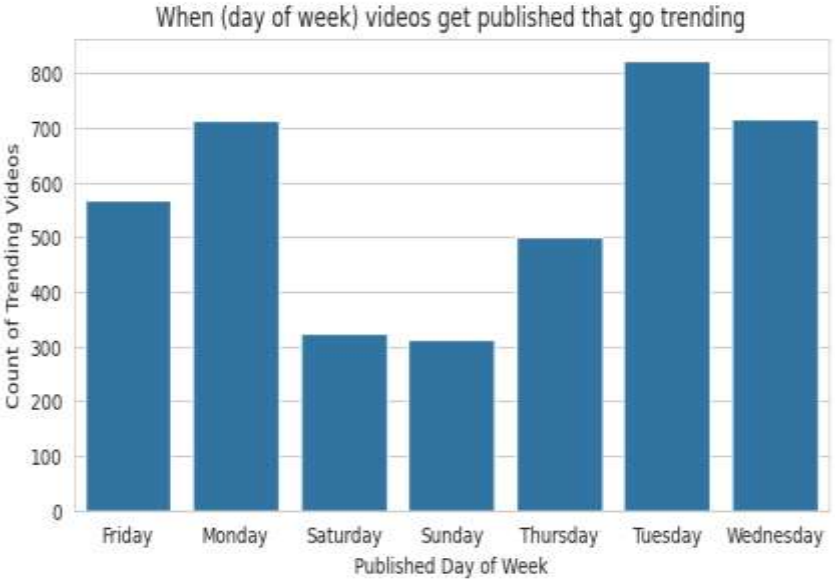
```
plt.figure(figsize=(12, 6))
sns.boxplot(data=df, x='category_id', y='views_log')
plt.xticks(rotation=90)
plt.xlabel("Category ID")
plt.ylabel("Log(Views)")
plt.title("Distribution of views by Category")
plt.show()
```

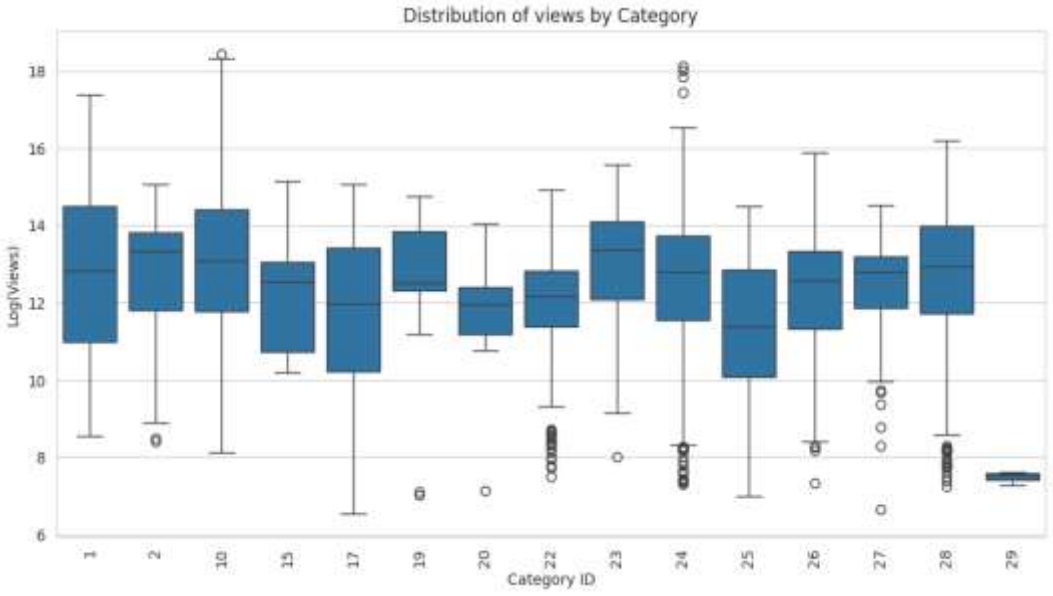
OUTPUT:



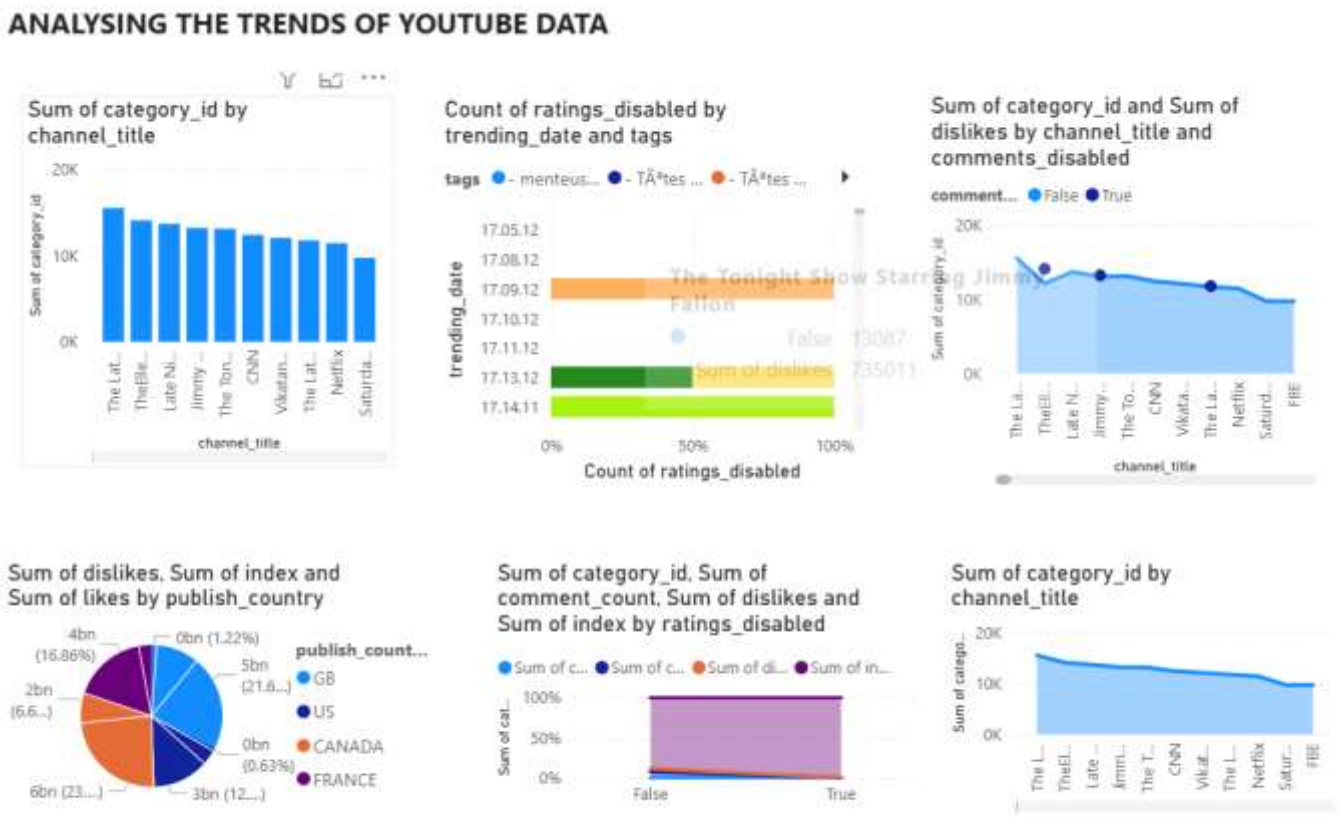



Views over time (for trending videos)
Views
Trending Date


When (day of week) videos get published that go trending
Count of Trending Videos
Friday
Monday
Saturday
Sunday
Thursday
Tuesday
Wednesday
Published Day of Week


Distribution of views by Category
Log(Views)
Category ID

POWER BI DASHBOARD:



RESULT:

Hence, The Mini project has been implemented successfully.