

```
!pip install isodate
import pandas as pd
from googleapiclient.discovery import build
from google.oauth2 import service_account
import matplotlib.pyplot as plt
import seaborn as sb
import numpy as np
import warnings
import isodate
warnings.filterwarnings('ignore')
```

Collecting isodate  
 Downloading isodate-0.6.1-py2.py3-none-any.whl.metadata (9.6 kB)  
 Requirement already satisfied: six in /usr/local/lib/python3.10/dist-packages (from isodate) (1.16.0)  
 Downloading isodate-0.6.1-py2.py3-none-any.whl (41 kB)  
 41.7/41.7 kB 1.4 MB/s eta 0:00:00  
 Installing collected packages: isodate  
 Successfully installed isodate-0.6.1

```
api_key = 'REDACTED'
def get_trending_videos(api_key, max_results=144):
    # build the youtube service
    youtube = build('youtube', 'v3', developerKey=api_key)

    # initialize the list to hold video details
    videos = []

    # fetch the most popular videos
    request = youtube.videos().list(
        part='snippet,contentDetails,statistics',
        chart='mostPopular',
        regionCode='IN',
        maxResults=50
    )

    # paginate through the results if max_results > 50
    while request and len(videos) < max_results:
        response = request.execute()
        for item in response['items']:
            video_details = {
                'video_id': item['id'],
                'title': item['snippet']['title'],
                'description': item['snippet']['description'],
                'published_at': item['snippet']['publishedAt'],
                'channel_id': item['snippet']['channelId'],
                'channel_title': item['snippet']['channelTitle'],
                'category_id': item['snippet']['categoryId'],
                'tags': item['snippet'].get('tags', []),
                'duration': item['contentDetails']['duration'],
                'definition': item['contentDetails']['definition'],
                'caption': item['contentDetails'].get('caption', 'false'),
                'view_count': item['statistics'].get('viewCount', 0),
                'like_count': item['statistics'].get('likeCount', 0),
                'dislike_count': item['statistics'].get('dislikeCount', 0),
                'favorite_count': item['statistics'].get('favoriteCount', 0),
                'comment_count': item['statistics'].get('commentCount', 0)
            }
            videos.append(video_details)

        # get the next page token
        request = youtube.videos().list_next(request, response)

    return videos[:max_results]

trending_videos = get_trending_videos(api_key)
df = pd.DataFrame(trending_videos)
df.to_csv('trending_videos.csv', index=False)
save_to_csv = pd.read_csv('trending_videos.csv')
trending_videos = pd.read_csv('trending_videos.csv')
trending_videos.head()
```

	video_id	title	description	published_at	channel_id	channel_title	categ
0	FEMR5alT7CY	जन्माष्टमी आरती: आरती कुंज बिहारी की, Aarti Ku...	🌺🌺https://youtu.be/FEMR5alT7CY🌺🌺 nSubscribe: h...	2024-08-26T12:30:12Z	UCaayLD9i5x4MmloVZxXSv_g	T-Series Bhakti Sagar	
1	Rv5mGPT5Fdc	Kabhi Main Kabhi Tum Episode 15   Fahad Mustaf...	Watch all the episode of Kabhi Main Kabhi Tum ...	2024-08-26T16:05:08Z	UC4JCksJF76g_MdzPVBJoC3Q	ARY Digital HD	
2	m3O1nPJszk	INDIA'S GOT LATENT   EP 05 ft. @KunalKamra @At...	Latent' is an ability that is hidden deep insi...	2024-08-26T15:44:07Z	UCAov2BBv1ZJav0c_yHEciAw	Samay Raina	
3	Kycaxw_f4RI	Maestro New Released Hindi Dubbed Movie 2024  ...	Presenting New south indian movies dubbed in h...	2024-08-24T13:30:08Z	UCX_uPA_dGf7wXjuMEaSKLJA	Aditya Movies	
4	7ugnQV4vIwA	We Make Boat Using Bottles 🚢 - Will It Survive?	SUBSCRIBE: https://youtube.com/MRINDIANHACKER?...	2024-08-26T12:08:24Z	UCSiDGB0MnHFGjs4E2WKvShw	MR. INDIAN HACKER	

Next steps: [Generate code with trending\\_videos](#) [View recommended plots](#) [New interactive sheet](#)

Start coding or [generate](#) with AI.

```
blank = trending_videos.isnull().sum()
blank
```

	0
video_id	0
title	0
description	4
published_at	0
channel_id	0
channel_title	0
category_id	0
tags	0
duration	0
definition	0
caption	0
view_count	0
like_count	0
dislike_count	0
favorite_count	0
comment_count	0

```
trending_videos['published_at'] = pd.to_datetime(trending_videos['published_at'])
trending_videos['published_at'] = trending_videos['published_at'].dt.date
trending_videos['duration_seconds'] = trending_videos['duration'].apply(lambda x: isodate.parse_duration(x).total_seconds())
trending_videos['duration_range'] = pd.cut(trending_videos['duration_seconds'], bins=[0, 300, 600, 1200, 3600, 7200], labels=['0-5 min', '5-10 min', '10-15 min', '15-20 min', '20-30 min', '30-45 min', '45-60 min', '60-90 min', '90-120 min', '120-180 min', '180-240 min', '240-360 min', '360-720 min', '720-1440 min', '1440-2880 min', '2880-5760 min', '5760-11520 min', '11520-23040 min', '23040-46080 min', '46080-92160 min', '92160-184320 min', '184320-368640 min', '368640-737280 min', '737280-1474560 min', '1474560-2949120 min', '2949120-5898240 min', '5898240-11796480 min', '11796480-23592960 min', '23592960-47185920 min', '47185920-94371840 min', '94371840-188743680 min', '188743680-377487360 min', '377487360-754974720 min', '754974720-1509949440 min', '1509949440-3019898880 min', '3019898880-6039797760 min', '6039797760-12079595520 min', '12079595520-24159191040 min', '24159191040-48318382080 min', '48318382080-96636764160 min', '96636764160-193273528320 min', '193273528320-386547056640 min', '386547056640-773094113280 min', '773094113280-1546188226560 min', '1546188226560-3092376453120 min', '3092376453120-6184752906240 min', '6184752906240-12369505812480 min', '12369505812480-24739011624960 min', '24739011624960-49478023249920 min', '49478023249920-98956046499840 min', '98956046499840-197912092999680 min', '197912092999680-395824185999360 min', '395824185999360-791648371998720 min', '791648371998720-1583296743997440 min', '1583296743997440-3166593487994880 min', '3166593487994880-6333186975989760 min', '6333186975989760-12666373951979520 min', '12666373951979520-25332747903959040 min', '25332747903959040-50665495807918080 min', '50665495807918080-101330991615836160 min', '101330991615836160-202661983231672320 min', '202661983231672320-405323966463344640 min', '405323966463344640-810647932926689280 min', '810647932926689280-1621295865853378560 min', '1621295865853378560-3242591731706757120 min', '3242591731706757120-6485183463413514240 min', '6485183463413514240-12970366926827028480 min', '12970366926827028480-25940733853654056960 min', '25940733853654056960-51881467707308113920 min', '51881467707308113920-103762935414616227840 min', '103762935414616227840-207525870829232455680 min', '207525870829232455680-415051741658464911360 min', '415051741658464911360-830103483316929822720 min', '830103483316929822720-1660206966633859645440 min', '1660206966633859645440-3320413933267719290880 min', '3320413933267719290880-6640827866535438581760 min', '6640827866535438581760-13281655733070877163520 min', '13281655733070877163520-26563311466141754327040 min', '26563311466141754327040-53126622932283508654080 min', '53126622932283508654080-106253245864567017308160 min', '106253245864567017308160-212506491729134034616320 min', '212506491729134034616320-425012983458268069232640 min', '425012983458268069232640-850025966916536138465280 min', '850025966916536138465280-1700051933833072276930560 min', '1700051933833072276930560-3400103867666144553861120 min', '3400103867666144553861120-6800207735332289107722240 min', '6800207735332289107722240-13600415470664578215444480 min', '13600415470664578215444480-27200830941329156430888960 min', '27200830941329156430888960-5440166188265831286177920 min', '5440166188265831286177920-10880332376531662572355840 min', '10880332376531662572355840-21760664753063325144711680 min', '21760664753063325144711680-43521329506126650289423360 min', '43521329506126650289423360-87042659012253300578846720 min', '87042659012253300578846720-174085318024506601157693440 min', '174085318024506601157693440-348170636049013202315386880 min', '348170636049013202315386880-696341272098026404630773760 min', '696341272098026404630773760-1392682544196052809261547520 min', '1392682544196052809261547520-2785365088392105618523095040 min', '2785365088392105618523095040-5570730176784211237046190080 min', '5570730176784211237046190080-11141460353568422474092380160 min', '11141460353568422474092380160-22282920707136844948184760320 min', '22282920707136844948184760320-44565841414273689896369520640 min', '44565841414273689896369520640-89131682828547379792739041280 min', '89131682828547379792739041280-178263365657094759585478082560 min', '178263365657094759585478082560-356526731314189519170956165120 min', '356526731314189519170956165120-713053462628379038341912330240 min', '713053462628379038341912330240-1426106925256758076683824660480 min', '1426106925256758076683824660480-2852213850513516153367649320960 min', '2852213850513516153367649320960-5704427701027032306735298641920 min', '5704427701027032306735298641920-11408855402054064613470597283840 min', '11408855402054064613470597283840-22817710804108129226941194567680 min', '22817710804108129226941194567680-45635421608216258453882389135360 min', '45635421608216258453882389135360-91270843216432516907764778270720 min', '91270843216432516907764778270720-182541686432865033815529556541440 min', '182541686432865033815529556541440-365083372865730067631059113082880 min', '365083372865730067631059113082880-730166745731460135262118226165760 min', '730166745731460135262118226165760-1460333491462920270524236452331520 min', '1460333491462920270524236452331520-2920666982925840541048472904663040 min', '2920666982925840541048472904663040-5841333965851681082096945809326080 min', '5841333965851681082096945809326080-11682667931703362164193891618652160 min', '11682667931703362164193891618652160-23365335863406724328387783237304320 min', '23365335863406724328387783237304320-46730671726813448656775566474608640 min', '46730671726813448656775566474608640-93461343453626897313551132949217280 min', '93461343453626897313551132949217280-186922686907253794627102265898434560 min', '186922686907253794627102265898434560-373845373814507589254204531796869120 min', '373845373814507589254204531796869120-747690747629015178508409063593738240 min', '747690747629015178508409063593738240-1495381495258030357016818127187476480 min', '1495381495258030357016818127187476480-2990762990516060714033636254374952960 min', '2990762990516060714033636254374952960-5981525981032121428067272508749905920 min', '5981525981032121428067272508749905920-11963051962064242856134545017499811840 min', '11963051962064242856134545017499811840-23926103924128485712269090034999623680 min', '23926103924128485712269090034999623680-47852207848256971424538180069999247360 min', '47852207848256971424538180069999247360-95704415696513942849076360139998494720 min', '95704415696513942849076360139998494720-191408831393027885698152720279996989440 min', '191408831393027885698152720279996989440-382817662786055771396305440559993978880 min', '382817662786055771396305440559993978880-765635325572111542792610881119979557760 min', '765635325572111542792610881119979557760-1531270651144223085585221762239959115520 min', '1531270651144223085585221762239959115520-306254130228844617117044352447991823040 min', '306254130228844617117044352447991823040-612508260457689234234088704895983646080 min', '612508260457689234234088704895983646080-1225016520915378468468177409791967292160 min', '1225016520915378468468177409791967292160-2450033041830756936936354819583934584320 min', '2450033041830756936936354819583934584320-4900066083661513873872709639167869168640 min', '4900066083661513873872709639167869168640-9800132167323027747745419278335738337280 min', '9800132167323027747745419278335738337280-19600264334646055495490838556671476674560 min', '19600264334646055495490838556671476674560-39200528669292110990981677113342953349120 min', '39200528669292110990981677113342953349120-78401057338584221981963354226685906698240 min', '78401057338584221981963354226685906698240-156802114677168443963926708453371813396480 min', '156802114677168443963926708453371813396480-313604229354336887927853416906743626792960 min', '313604229354336887927853416906743626792960-627208458708673775855706833813487253585920 min', '627208458708673775855706833813487253585920-1254416917417347551711413667626975111171840 min', '1254416917417347551711413667626975111171840-2508833834834695103422827335253950222343680 min', '2508833834834695103422827335253950222343680-5017667669669390206845654670507900444687360 min', '5017667669669390206845654670507900444687360-10035335339338780413691309341015800889374720 min', '10035335339338780413691309341015800889374720-20070670678677560827382618682031601778749440 min', '20070670678677560827382618682031601778749440-40141341357355121654765237364063203557498880 min', '40141341357355121654765237364063203557498880-80282682714710243309530474728126407114997760 min', '80282682714710243309530474728126407114997760-16056536542942048661906094945625282822999520 min', '16056536542942048661906094945625282822999520-32113073085884097323812189891250565645999040 min', '32113073085884097323812189891250565645999040-64226146171768194647624379782501131291998080 min', '64226146171768194647624379782501131291998080-128452292343536389295248759565002262583996160 min', '128452292343536389295248759565002262583996160-256904584687072778590497519130004525167992320 min', '256904584687072778590497519130004525167992320-513809169374145557180995038260009050335984640 min', '513809169374145557180995038260009050335984640-1027618338748291114361990076520018100671969280 min', '1027618338748291114361990076520018100671969280-2055236677496582228723980153040036201343938560 min', '2055236677496582228723980153040036201343938560-4110473354993164457447960306080072402687877120 min', '4110473354993164457447960306080072402687877120-8220946709986328914895920612160144805375754240 min', '8220946709986328914895920612160144805375754240-16441893419972657829791841224320289611151508480 min', '16441893419972657829791841224320289611151508480-32883786839945315659583682448640579222303016960 min', '32883786839945315659583682448640579222303016960-65767573679890631319167364897281144446060233920 min', '65767573679890631319167364897281144446060233920-131535147359781262638334729794562288892120467840 min', '131535147359781262638334729794562288892120467840-263070294719562525276669459589124577842240935680 min', '263070294719562525276669459589124577842240935680-52614058943912505055333891917824915568448191360 min', '52614058943912505055333891917824915568448191360-105228117887825010110667783835649231136896382720 min', '105228117887825010110667783835649231136896382720-210456235775650020221335567671298462273792765440 min', '210456235775650020221335567671298462273792765440-420912471551300040442671135342596924547585530880 min', '420912471551300040442671135342596924547585530880-841824943102600080885342270685193849095171061760 min', '841824943102600080885342270685193849095171061760-1683649886205200161770684541370387698190342123520 min', '1683649886205200161770684541370387698190342123520-3367299772410400323541369082740775396380684247040 min', '3367299772410400323541369082740775396380684247040-6734599544820800647082738165481550792761368494080 min', '6734599544820800647082738165481550792761368494080-13469199089641601294165476330963101585522736988160 min', '13469199089641601294165476330963101585522736988160-26938398179283202588330952661926203171045473976320 min', '26938398179283202588330952661926203171045473976320-53876796358566405176661905323852406342090947952640 min', '53876796358566405176661905323852406342090947952640-107753592717132810353323810647704812684181895905280 min', '107753592717132810353323810647704812684181895905280-215507185434265620706647621295409625368363791810560 min', '215507185434265620706647621295409625368363791810560-431014370868531241413295242590819250736727583621120 min', '431014370868531241413295242590819250736727583621120-862028741737062482826590485181638501473455172242240 min', '862028741737062482826590485181638501473455172242240-1724057483474124965653180970363277002946910344444480 min', '1724057483474124965653180970363277002946910344444480-344811496694824993130636194072655400589380688888960 min', '344811496694824993130636194072655400589380688888960-689622993389649986261272388145310811778771377777920 min', '689622993389649986261272388145310811778771377777920-1379245
```



channel\_title category

T-Series Bhakti  
Sagar

ARY Digital HD

Samay Raina

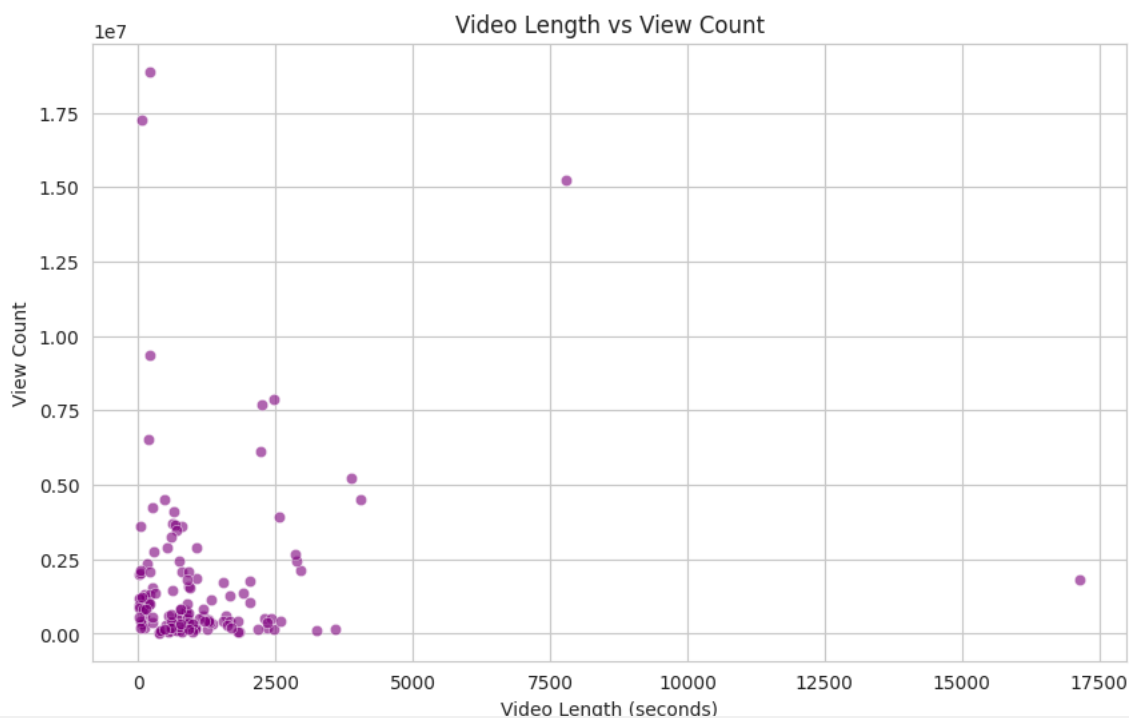
Aditya Movies

MR. INDIAN  
HACKER


Next steps:





[Generate code with trending\\_videoses](#)[View recommended plots](#)[New interactive sheet](#)

```
plt.figure(figsize=(10, 6))
sb.scatterplot(x='duration_seconds', y='view_count', data=trending_videoses, alpha=0.6, color='purple')
plt.title('Video Length vs View Count')
plt.xlabel('Video Length (seconds)')
plt.ylabel('View Count')
plt.show()
```



```
trending_videoses['description'].fillna('No description', inplace=True)
trending_videoses['tags'] = trending_videoses['tags'].apply(lambda x: ', '.join(x))
# trending_videoses['tags'] = trending_videoses['tags'].apply(lambda x:eval(x) if isinstance(x,str) else x)
trending_videoses.head()
```

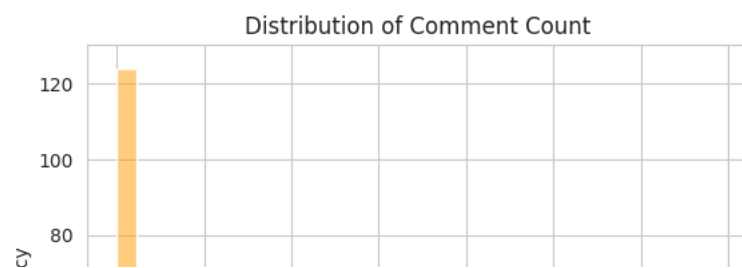
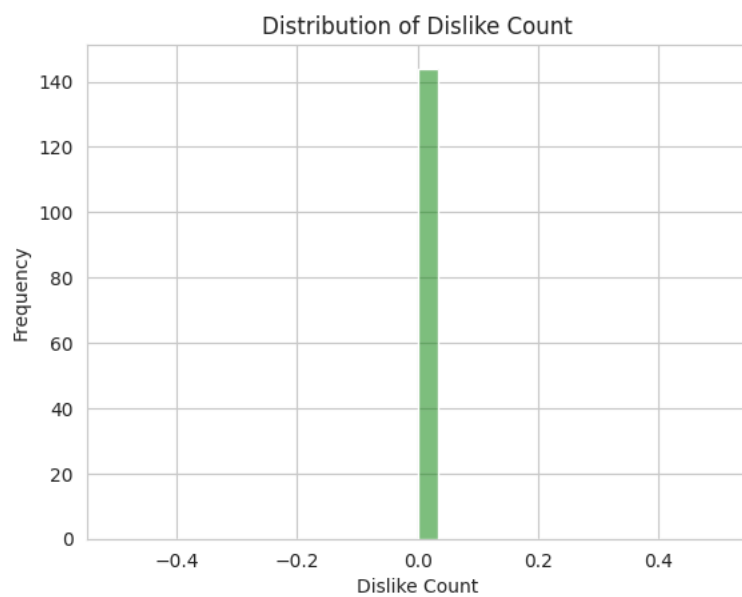
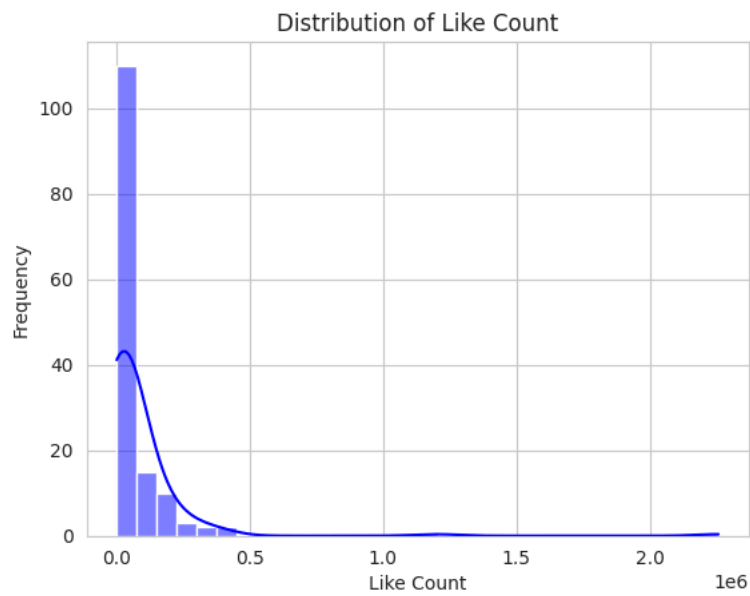
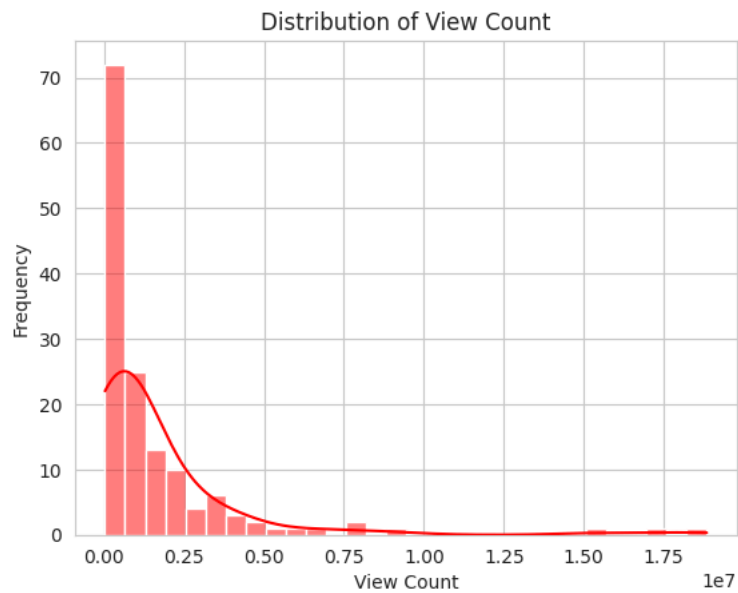


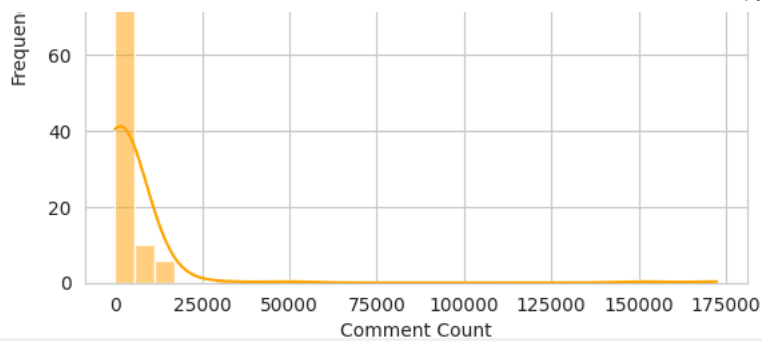
	video_id	title	description	published_at	channel_id	channel_title	categ
0	FEMR5alT7CY	जन्माष्टमी आरती: आरती कुंज बिहारी की, Aarti Ku...	  <a href="https://youtu.be/FEMR5alT7CY">https://youtu.be/FEMR5alT7CY</a>   \nSubscribe: h...	2024-08-26	UCaayLD9i5x4MmloVZxXSv_g	T-Series Bhakti Sagar	
1	Rv5mGpt5Fdc	Kabhi Main Kabhi Tum Episode 15   Fahad Mustaf...	Watch all the episode of Kabhi Main Kabhi Tum ...	2024-08-26	UC4JCksJF76g_MdzPVBJoC3Q	ARY Digital HD	
2	m3O1lnPJszk	INDIA'S GOT LATENT   EP 05 ft. @KunalKamra @At...	Latent' is an ability that is hidden deep insi...	2024-08-26	UCAov2BBv1ZJav0c_yHEciAw	Samay Raina	
3	Kycaxw_f4RI	Maestro New Released Hindi Dubbed Movie 2024  ...	Presenting New south indian movies dubbed in h...	2024-08-24	UCX_uPA_dGf7wXjuMEaSKLJA	Aditya Movies	
4	7ugnQV4vlwA	We Make Boat Using Bottles 🤖 - Will It Survive?	SUBSCRIBE: <a href="https://youtube.com/MRINDIANHACKER?...">https://youtube.com/MRINDIANHACKER?...</a>	2024-08-26	UCSiDGb0MnHFGjs4E2WKvShw	MR. INDIAN HACKER	

Next steps: [Generate code with trending\\_videoses](#) [View recommended plots](#) [New interactive sheet](#)

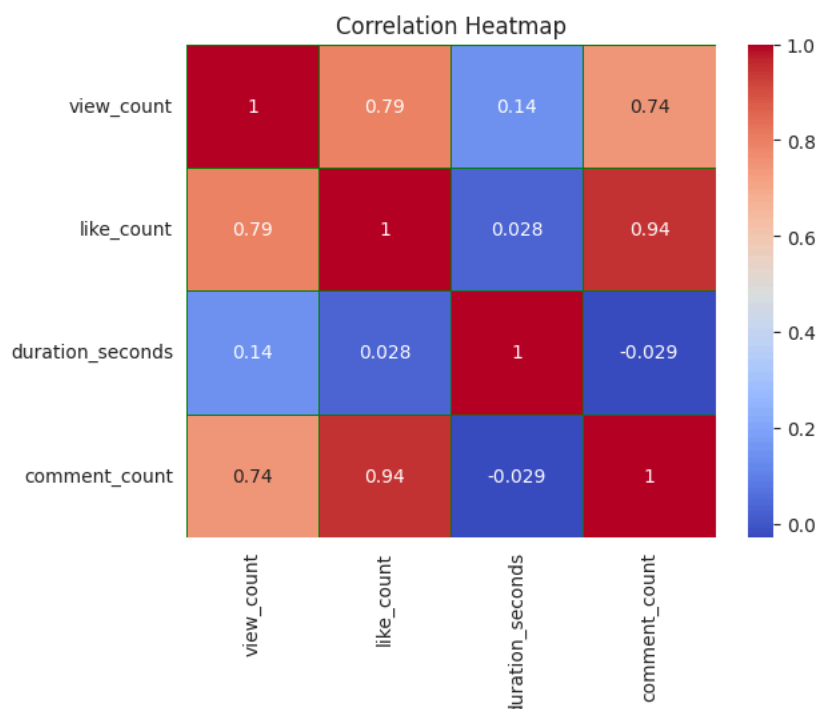
```
sb.set_style('whitegrid')

sb.histplot(trending_videoses['view_count'],bins=30 , kde = True ,color = 'red')
plt.xlabel('View Count')
plt.ylabel('Frequency')
plt.title('Distribution of View Count')
plt.show()
sb.histplot(trending_videoses['like_count'],bins=30 , kde = True ,color = 'blue')
plt.xlabel('Like Count')
plt.ylabel('Frequency')
plt.title('Distribution of Like Count')
plt.show()
sb.histplot(trending_videoses['dislike_count'],bins=30 , kde = True ,color = 'green')
plt.xlabel('Dislike Count')
plt.ylabel('Frequency')
plt.title('Distribution of Dislike Count')
plt.show()
sb.histplot(trending_videoses['comment_count'],bins=30 , kde = True ,color = 'orange')
plt.xlabel('Comment Count')
plt.ylabel('Frequency')
plt.title('Distribution of Comment Count')
plt.show()
```





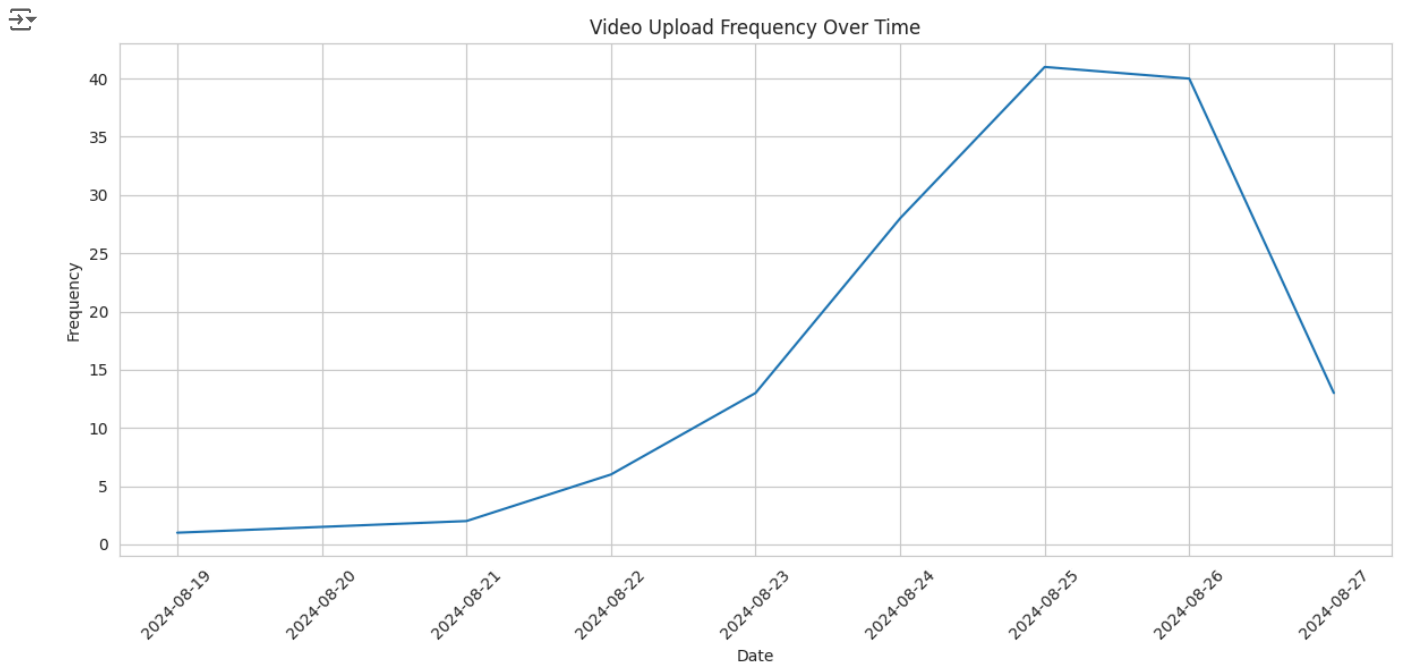
```
coorelation_matrix = trending_videos[['view_count', 'like_count', 'duration_seconds', 'comment_count']].corr()
coorelation_matrix
sb.heatmap(coorelation_matrix, annot=True, cmap='coolwarm', linewidths=0.5, linecolor='green')
plt.title('Correlation Heatmap')
plt.show()
```



```
trending_videos['published_at'] = pd.to_datetime(trending_videos['published_at']) # Ensure datetime format
```

```
# Group by 'published_at' and count occurrences
video_counts = trending_videos.groupby('published_at').size()
```

```
# Create the plot
plt.figure(figsize=(12, 6))
plt.plot(video_counts.index, video_counts.values)
plt.grid=False
plt.xlabel('Date')
plt.ylabel('Frequency')
plt.title('Video Upload Frequency Over Time')
plt.xticks(rotation=45)
plt.tight_layout()
plt.show()
```



```

youtube = build('youtube', 'v3', developerKey=api_key)
request = youtube.videoCategories().list(
    part='snippet',
    regionCode='IN'
)
response = request.execute()
response
category_dict = {}
for item in response['items']:
    category_id = item['id']
    category_name = item['snippet']['title']
    category_dict[category_id] = category_name
print(category_dict)

```

```

{'1': 'Film & Animation', '2': 'Autos & Vehicles', '10': 'Music', '15': 'Pets & Animals', '17': 'Sports', '18': 'Short Movies', '19'

```

```

def get_category_dict(api_key):
    youtube = build('youtube', 'v3', developerKey=api_key)
    request = youtube.videoCategories().list(
        part='snippet',
        regionCode='IN'
    )
    response = request.execute()
    return {item['id']: item['snippet']['title'] for item in response['items']}
trending_videos = get_trending_videos(api_key)
#category_dict = {}
df = pd.DataFrame(trending_videos)
df['category_name'] = df['category_id'].map(category_dict)

# Now you have the 'category_name' column in your DataFrame
print(df.head())
plt.figure(figsize=(10, 6))
sb.countplot(data=df, x='category_name', palette='viridis', order=df['category_name'].value_counts().index)
plt.xticks(rotation=45)
plt.xlabel('Number of Videos')
plt.ylabel('Category')
plt.title('Number of Trending Videos by Category')
plt.show()

```

```

video_id      title \
0 FEMR5a1T7CY जन्माष्टमी आरती: आरती कुंज बिहारी की, Aarti Ku...
1 Rv5mGpt5Fdc Kabhi Main Kabhi Tum Episode 15 | Fahad Mustaf...
2 m301lnPJszk INDIA'S GOT LATENT | EP 05 ft. @KunalKamra @At...
3 Kycaxw_f4RI Maestro New Released Hindi Dubbed Movie 2024 |...
4 7ugnQV4vIwA We Make Boat Using Bottles 🚤 - Will It Survive?

description    published_at \
0 🚩🚩🚩https://youtu.be/FEMR5a1T7CY🚩🚩🚩\nSubscribe: h... 2024-08-26T12:30:12Z
1 Watch all the episode of Kabhi Main Kabhi Tum ... 2024-08-26T16:05:08Z
2 Latent' is an ability that is hidden deep insi... 2024-08-26T15:44:07Z
3 Presenting New south indian movies dubbed in h... 2024-08-24T13:30:08Z
4 SUBSCRIBE: https://youtube.com/MRINDIANHACKER?... 2024-08-26T12:08:24Z

channel_id      channel_title category_id \
0 UCaayLD9i5x4MmIoVZxXSv_g T-Series Bhakti Sagar 10
1 UC4JCKsJF76g_MdzPVBJoC3Q ARY Digital HD 24
2 UCAov2BBv1ZJav0c_yHEciAw Samay Raina 20
3 UCX_uPA_dGf7wXjuMEaSKLJA Aditya Movies 1
4 UCSiDGB0MnHFGjs4E2WKvShw MR. INDIAN HACKER 28

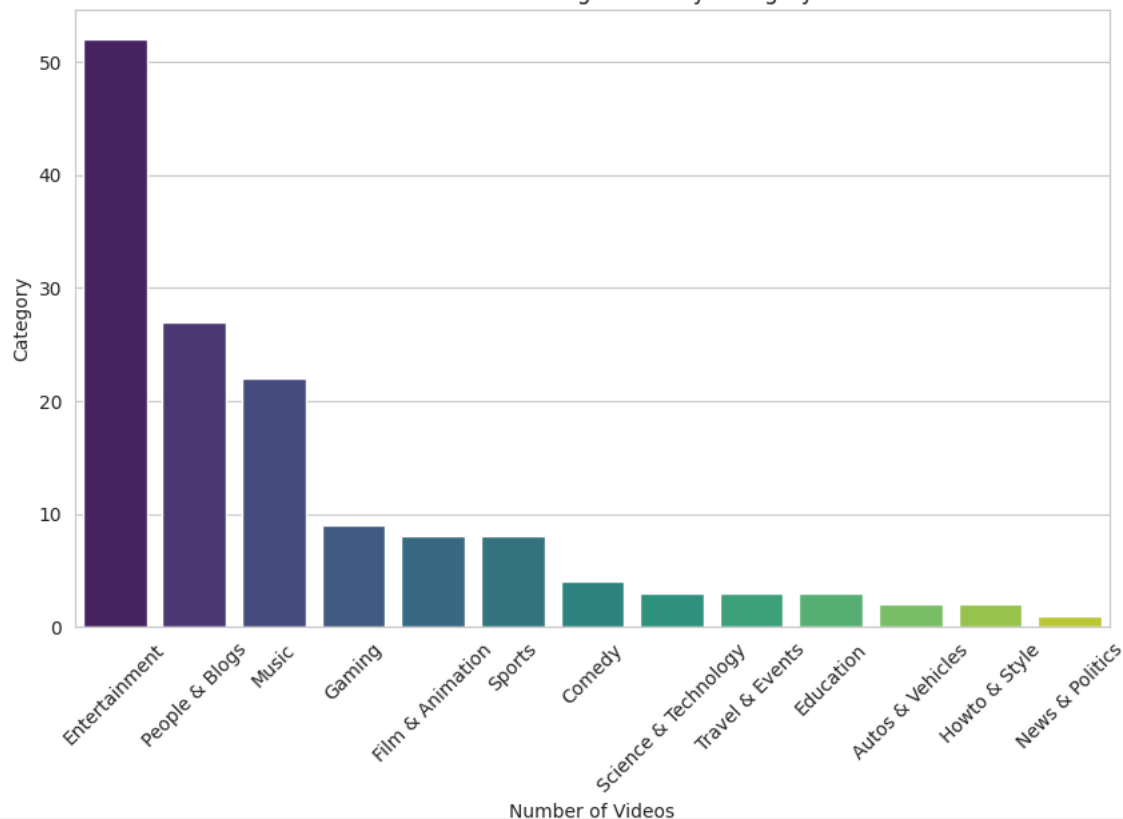
tags    duration definition \
0 [devotional songs, bhakti bhajans, bhakti song... PT4M28S hd
1 [ary, ary digital, new Pakistani dramas, Kabhi... PT41M25S hd
2 [samay raina, samay raina comedy, samay raina ... PT1H7M27S hd
3 [Maestro, maestro full movie in hindi dubbed, ... PT2H10M4S hd
4 [boat, bottles, boatfrombottles, waterbottles,... PT12M24S hd

caption view_count like_count dislike_count favorite_count comment_count \
0 true 1545945 14966 0 0 553
1 true 7884402 162648 0 0 5903
2 false 4521058 366628 0 0 17321
3 true 15234544 379236 0 0 10640
4 false 2439303 220796 0 0 11681

category_name
0 Music
1 Entertainment
2 Gaming
3 Film & Animation
4 Science & Technology

```

Number of Trending Videos by Category





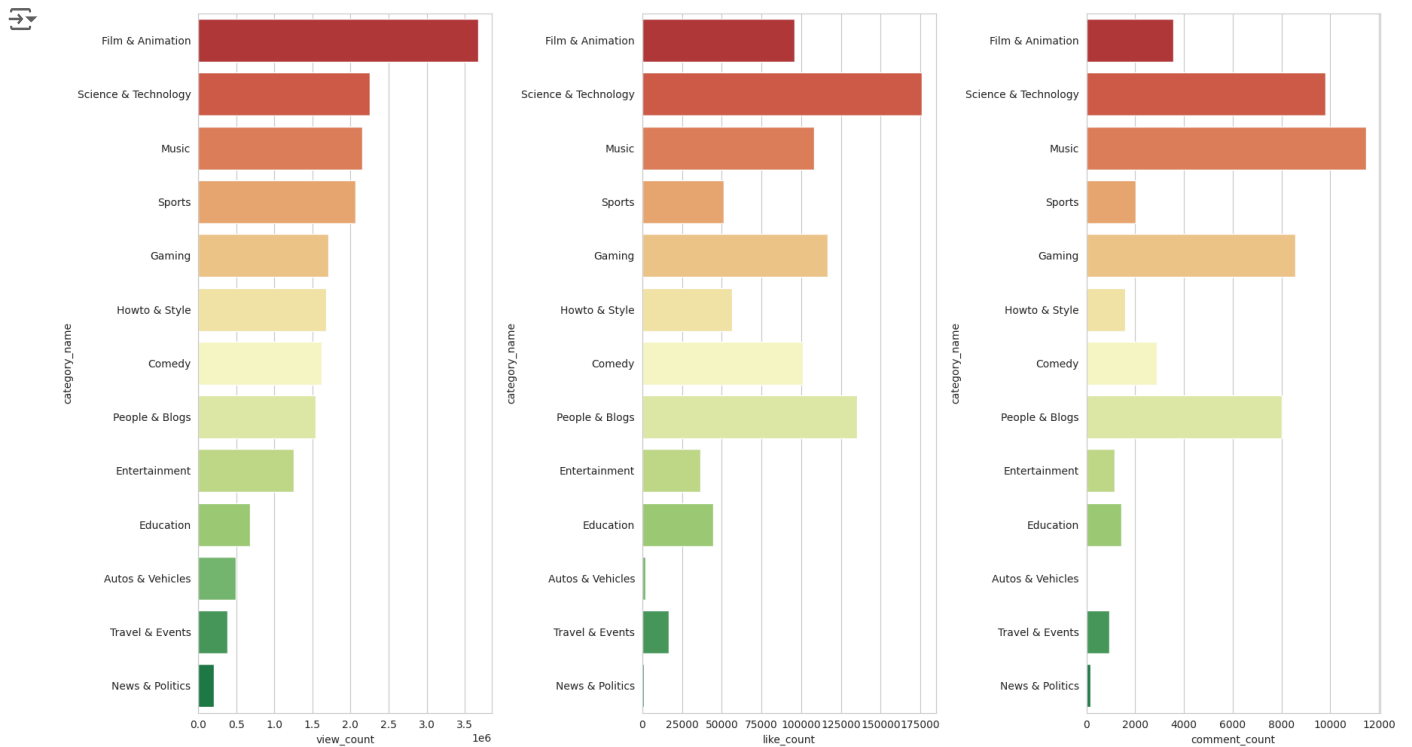
```

df['view_count'] = pd.to_numeric(df['view_count'])
df['like_count'] = pd.to_numeric(df['like_count'])
df['comment_count'] = pd.to_numeric(df['comment_count'])
category_stats = df.groupby('category_name')[['view_count', 'like_count', 'comment_count']].mean().sort_values(by='view_count', ascending=False)
# ({
#     'view_count': 'mean',
#     'like_count': 'mean',
#     'comment_count': 'mean'
# }).reset_index()
category_stats
sb.set_style('whitegrid')
fig, axes = plt.subplots(1, 3, figsize=(18, 10))
sb.barplot(data=category_stats.reset_index(), x='view_count', y='category_name', ax=axes[0], palette='RdYlGn')
# plt.xlabel('Category')
# plt.ylabel('Average View Count')
# plt.title('Average View Count by Category')

sb.barplot(data=category_stats.reset_index(), x='like_count', y='category_name', ax=axes[1], palette='RdYlGn')
# plt.xlabel('Category')
# plt.ylabel('Average Like Count')
# plt.title('Average Like Count by Category')

sb.barplot(data=category_stats.reset_index(), x='comment_count', y='category_name', ax=axes[2], palette='RdYlGn')
plt.tight_layout()

```



```

df['duration_seconds'] = df['duration'].apply(lambda x: isodate.parse_duration(x).total_seconds())
df['duration_range'] = pd.cut(df['duration_seconds'], bins=[0, 300, 600, 1200, 3600, 7200], labels=['0-5 min', '5-10 min', '10-20 min',

```

```
plt.figure(figsize=(10, 6))
sb.set_style('whitegrid')
sb.scatterplot(x='duration_seconds', y='view_count', data=df, hue='duration_range', palette='viridis', alpha=0.6)

plt.title('Video Length vs View Count')
plt.xlabel('Video Length (seconds)')
plt.ylabel('View Count(log)')
plt.yscale('log')
plt.legend(title='Duration Range')
plt.show()

length_engagement = df.groupby('duration_range')[['view_count', 'like_count', 'comment_count']].mean()
fig, axes = plt.subplots(1, 3, figsize=(18, 10))
sb.barplot(y='view_count', x=length_engagement.index, data=length_engagement.reset_index(), ax=axes[0], palette='viridis')
sb.barplot(y='like_count', x=length_engagement.index, data=length_engagement.reset_index(), ax=axes[1], palette='viridis')
sb.barplot(y='comment_count', x=length_engagement.index, data=length_engagement.reset_index(), ax=axes[2], palette='viridis')
plt.tight_layout()
plt.show()
```

