To Scrape YouTube using YouTube Data API & Analyze and Visualize YouTube data

Name : Tarun Lagudu
Github : https://github.com/Lagudu-Tarun
Linkdein : https://www.linkedin.com/in/tarun-lagudu
Twitter : https://twitter.com/Lagudu_Tarun

In [ ]:

```
```

YouTube has revolutionized the way we consume content and has become a prominent platform for education and learning. With the vast amount of data available on YouTube, it provides an excellent opportunity for researchers and analysts to extract valuable insights.
This project aims to analyze the performance of three prominent educational channels on YouTube : Great Learning,Simplilearn, and Intellipaat.
These are e-learning platforms that offer courses and tutorials related to various topics in technology and business.

The project uses the YouTube API to extract data such as the number of subscribers, views, likes, and comments for each channel. The data collected is then analyzed to gain insights into the popularity and engagement of each channel.

The project intends to provide a comprehensive analysis of the three educational channels,
enabling us to understand their performance and audience engagement.
The analysis can provide valuable insights for educators and marketers who use YouTube as a platform to deliver their content.

In [ ]:

```
```

Tasks Performed-

1) Extracted channel id's of the desired youtube channels.

2) Retrieved channel statistics such as subscriber count,views, and video count, then create a plot to visualize the data.

3) Acquired video details of the channels using Video Id's.

4) Calculated the total & average number of views, likes, and comments for each channel and create a plot to visualize the data.

5) Calculate viewer the engagement rate for each channel.

6) Obtained the top 10 performing videos for each channel based on views and likes.

In [ ]:

```
```

Importing necessary libraries

In [266…

```
from googleapiclient.discovery import build
```

```
# This function is used to create a service object, which is used to make requests to a
# The "build" function takes several parameters, including the name of the API, the vers
# any additional options required for authentication and authorization.

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from googleapiclient.errors import HttpError

!pip install textblob
from textblob import TextBlob


# The "HttpError" class is used to handle errors that occur during API requests.
# When we request to a Google API fails, the API may return an HTTP error code
# along with an error message that provides more information about the problem.
# The "HttpError" class allows you to handle these errors in a structured way.
```

```
Requirement already satisfied: textblob in c:\users\ltaru\anaconda3\lib\site-packages
(0.17.1)
Requirement already satisfied: nltk>=3.1 in c:\users\ltaru\anaconda3\lib\site-packages
(from textblob) (3.7)
Requirement already satisfied: regex>=2021.8.3 in c:\users\ltaru\anaconda3\lib\site-pack
ages (from nltk>=3.1->textblob) (2022.7.9)
Requirement already satisfied: tqdm in c:\users\ltaru\anaconda3\lib\site-packages (from
nltk>=3.1->textblob) (4.64.1)
Requirement already satisfied: click in c:\users\ltaru\anaconda3\lib\site-packages (from
nltk>=3.1->textblob) (8.0.4)
Requirement already satisfied: joblib in c:\users\ltaru\anaconda3\lib\site-packages (fro
m nltk>=3.1->textblob) (1.1.0)
Requirement already satisfied: colorama in c:\users\ltaru\anaconda3\lib\site-packages (f
rom click->nltk>=3.1->textblob) (0.4.5)
```

In [ ]:

Extracting channel id's

Go to the desired youtube channel -> Right click -> view page source -> ctrl + F -> search channelid

For Generating API key go to https://console.cloud.google.com/projectselector2/apis/dashboard?pli=1&supportedpurview=project -> Create a new project -> Go to library -> Youtube Data API and click enable -> Go to credentials -> Create credentials -> API key

In [267…

```
api_key= 'AIzaSyAiE1O64AjGVh9TUnk5ZVbuWVj1mZ-lGJs'
channel_ids = ["UCsvqVGtbbyHaMoevxPAq9Fg",#Simplilearn
               "UCCktnahuRFYIBtNnKT5IYyg",#Intellipaat
               "UCObs0kLIrDjX2LLSybqNaEA",#Great Learning
              ]
```

In [268…

```
# Got to https://developers.google.com/youtube/v3/docs/channels/list?apix=true
#Show code - Python
youtube = build('youtube','v3', developerKey=api_key)

# This creates a service object for the YouTube Data API version 3 using the Google API
# We can use this to interact with the API, such as retrieving information about YouTube
# searching for videos, or uploading videos.
```

In [ ]:

To retrieve channel statistics such as subscriber count,views, and video count, then create a plot to visualize the data.

https://developers.google.com/youtube/v3/docs/channels/list

Returns a collection of zero or more channel resources that match the request criteria.

In [269…

```
#The part parameter specifies comma-separated list of channel resource properties that t

def get_channel_details(youtube,channel_ids):
    request = youtube.channels().list(part='snippet,contentDetails,statistics',id=','.jo
    response = request.execute()
    return response


# the "channels().list()" method of the "youtube" object is called to retrieve the detai
# The "part" parameter is set to 'snippet,contentDetails,statistics' to specify which pa
# include in the response.
# The "id" parameter is set to ','.join(channel_ids), which converts the list of channel
# This string is used to specify the channels for which details need to be retrieved.

# The "execute()" method is called on the "request" object, which sends the API request
# and returns the response as a JSON object.
```

In [270…

```
get_channel_details(youtube,channel_ids)
```

Out[270]:

```
{'kind': 'youtube#channelListResponse',
 'etag': '7XojdRWbADWygKuHCDXizxmuC5M',
 'pageInfo': {'totalResults': 3, 'resultsPerPage': 5},
 'items': [{'kind': 'youtube#channel',
   'etag': 'bMdTpYR0R7zT7_uSMGLwIemB7jU',
   'id': 'UCCktnahuRFYIBtNnKT5IYyg',
   'snippet': {'title': 'Intellipaat',
    'description': 'Intellipaat is a global online professional training provider. We ar
e offering some of the most updated, industry-designed certification training programs w
hich includes courses in Big Data, Data Science, Artificial Intelligence and 150 other t
op trending technologies.\nWe help professionals make the right career decisions, choose
the trainers with over a decade of industry experience, provide extensive hands-on proje
cts, rigorously evaluate learner progress and offer industry-recognized certifications.
We also assist corporate clients to upskill their workforce and keep them in sync with t
he changing technology and digital landscape.\n\nFor more information visit us at:- http
s://intellipaat.com\n\nContact us at:- sales@intellipaat.com | +91-7022374614 , US: 1-80
0-216-8930',
    'customUrl': '@intellipaat',
    'publishedAt': '2013-01-24T06:37:31Z',
    'thumbnails': {'default': {'url': 'https://yt3.ggpht.com/sm9KjocCAAXBPB5gyBMk24j6ZWk
2t8nOw2ID_00u5fiS2qBH3MIiphaklt5wq91KeSCHwlCz=s88-c-k-c0x00ffffff-no-rj',
      'width': 88,
      'height': 88},
     'medium': {'url': 'https://yt3.ggpht.com/sm9KjocCAAXBPB5gyBMk24j6ZWk2t8nOw2ID_00u5f
iS2qBH3MIiphaklt5wq91KeSCHwlCz=s240-c-k-c0x00ffffff-no-rj',
      'width': 240,
      'height': 240},
     'high': {'url': 'https://yt3.ggpht.com/sm9KjocCAAXBPB5gyBMk24j6ZWk2t8nOw2ID_00u5fiS
2qBH3MIiphaklt5wq91KeSCHwlCz=s800-c-k-c0x00ffffff-no-rj',
      'width': 800,
      'height': 800}},
    'localized': {'title': 'Intellipaat',
     'description': 'Intellipaat is a global online professional training provider. We a
re offering some of the most updated, industry-designed certification training programs
which includes courses in Big Data, Data Science, Artificial Intelligence and 150 other
top trending technologies.\nWe help professionals make the right career decisions, choos
e the trainers with over a decade of industry experience, provide extensive hands-on pro
jects, rigorously evaluate learner progress and offer industry-recognized certification
s. We also assist corporate clients to upskill their workforce and keep them in sync wit
h the changing technology and digital landscape.\n\nFor more information visit us at:- h
ttps://intellipaat.com\n\nContact us at:- sales@intellipaat.com | +91-7022374614 , US: 1
-800-216-8930'},
    'country': 'IN'},
   'contentDetails': {'relatedPlaylists': {'likes': '',
     'uploads': 'UUCktnahuRFYIBtNnKT5IYyg'}},
   'statistics': {'viewCount': '120061467',
    'subscriberCount': '1680000',
    'hiddenSubscriberCount': False,
    'videoCount': '3547'}},
  {'kind': 'youtube#channel',
   'etag': '5SNiSsAhKS91tYZW5hcN-mycyqY',
   'id': 'UCsvqVGtbbyHaMoevxPAq9Fg',
   'snippet': {'title': 'Simplilearn',
    'description': 'Simplilearn is the world’s #1 online Bootcamp and one of the world’s
leading certification training providers. Based in San Francisco, California, and Bangal
ore, India, we provide training in areas where technologies and best practices are chang
ing rapidly, and the demand for qualified candidates significantly exceeds the supply. W
e have trained over 3,000,000 professionals, have over 2000 qualified trainers on board,
and offer over 400 courses with 40 plus global accreditations. With live instructions fr
om leading experts, interactive labs & projects, peer-to-peer collaboration, on-demand l
essons, and 24/7 learning support, we provide learners with a comprehensive curriculum a
t a fraction of the cost of an on-campus program. \n\nFor more information, visit http
s://www.simplilearn.com/\nOur FREE courses via SkillUp by Simplilearn now come with Comp
letion Certificates! :https://www.simplilearn.com/skillup-free-online-courses\n\n',
    'customUrl': '@simplilearnofficial',
    'publishedAt': '2009-10-29T07:21:23Z',
    'thumbnails': {'default': {'url': 'https://yt3.ggpht.com/7q9t5rjeujEZYqY1xMLn0mvT4Zc
6MaZBYgtseDL2_Zh42AOhMze8ep7BUKdR5FnxytMy3csj=s88-c-k-c0x00ffffff-no-rj',
      'width': 88,
      'height': 88},
     'medium': {'url': 'https://yt3.ggpht.com/7q9t5rjeujEZYqY1xMLn0mvT4Zc6MaZBYgtseDL2_Z
h42AOhMze8ep7BUKdR5FnxytMy3csj=s240-c-k-c0x00ffffff-no-rj',
      'width': 240,
      'height': 240},
     'high': {'url': 'https://yt3.ggpht.com/7q9t5rjeujEZYqY1xMLn0mvT4Zc6MaZBYgtseDL2_Zh4
2AOhMze8ep7BUKdR5FnxytMy3csj=s800-c-k-c0x00ffffff-no-rj',
      'width': 800,
      'height': 800}},
    'localized': {'title': 'Simplilearn',
     'description': 'Simplilearn is the world’s #1 online Bootcamp and one of the worl
d’s leading certification training providers. Based in San Francisco, California, and Ba
ngalore, India, we provide training in areas where technologies and best practices are c
hanging rapidly, and the demand for qualified candidates significantly exceeds the suppl
y. We have trained over 3,000,000 professionals, have over 2000 qualified trainers on bo
ard, and offer over 400 courses with 40 plus global accreditations. With live instructio
ns from leading experts, interactive labs & projects, peer-to-peer collaboration, on-dem
and lessons, and 24/7 learning support, we provide learners with a comprehensive curricu
lum at a fraction of the cost of an on-campus program. \n\nFor more information, visit h
ttps://www.simplilearn.com/\nOur FREE courses via SkillUp by Simplilearn now come with C
ompletion Certificates! :https://www.simplilearn.com/skillup-free-online-courses\n\n'},
    'country': 'US'},
   'contentDetails': {'relatedPlaylists': {'likes': '',
     'uploads': 'UUsvqVGtbbyHaMoevxPAq9Fg'}},
   'statistics': {'viewCount': '299801087',
    'subscriberCount': '2970000',
    'hiddenSubscriberCount': False,
    'videoCount': '5448'}},
  {'kind': 'youtube#channel',
   'etag': 'o6XCuQYYaRvwDCxISmLKyTgrbYc',
   'id': 'UCObs0kLIrDjX2LLSybqNaEA',
   'snippet': {'title': 'Great Learning',
    'description': "Great Learning is Asia’s leading professional learning platform,  of
fering top-ranked programs in AI, Data Science, Analytics, Cloud Computing, Full-stack D
```

```
evelopment, Cybersecurity, Digital Marketing, Digital Business, Design Thinking and mor
e. Great Learning's programs are developed in collaboration with the world's foremost ac
ademic institutions like Stanford GSB, MIT, IITB, IITM, Great Lakes Inst of Management,
and The University of Texas at Austin. We have delivered over 101 million+ learning hour
s and impacted 5.9 million+ learners in 170+ Countries.\n\nGreat Learning aims to make q
uality education accessible to anyone who wants to learn. Great Learning Academy (http
s://www.mygreatlearning.com/academy), our commitment to aid free learning, offers 1200+
completely FREE courses in the most in-demand job domains. In addition to videos, you ca
n earn certificates of completion, do assignments and projects, attend live sessions and
interact with top faculty & industry experts.\n",
    'customUrl': '@greatlearning',
    'publishedAt': '2013-04-03T12:17:17Z',
    'thumbnails': {'default': {'url': 'https://yt3.ggpht.com/ytc/AGIKgqNGoOy-6MgjagP1Ytu
v7bhUPqFQR08Qfaz0EQXpbA=s88-c-k-c0x00ffffff-no-rj',
      'width': 88,
      'height': 88},
     'medium': {'url': 'https://yt3.ggpht.com/ytc/AGIKgqNGoOy-6MgjagP1Ytuv7bhUPqFQR08Qfa
z0EQXpbA=s240-c-k-c0x00ffffff-no-rj',
      'width': 240,
      'height': 240},
     'high': {'url': 'https://yt3.ggpht.com/ytc/AGIKgqNGoOy-6MgjagP1Ytuv7bhUPqFQR08Qfaz0
EQXpbA=s800-c-k-c0x00ffffff-no-rj',
      'width': 800,
      'height': 800}},
    'localized': {'title': 'Great Learning',
     'description': "Great Learning is Asia's leading professional learning platform,  o
ffering top-ranked programs in AI, Data Science, Analytics, Cloud Computing, Full-stack
Development, Cybersecurity, Digital Marketing, Digital Business, Design Thinking and mor
e. Great Learning's programs are developed in collaboration with the world's foremost ac
ademic institutions like Stanford GSB, MIT, IITB, IITM, Great Lakes Inst of Management,
and The University of Texas at Austin. We have delivered over 101 million+ learning hour
s and impacted 5.9 million+ learners in 170+ Countries.\n\nGreat Learning aims to make q
uality education accessible to anyone who wants to learn. Great Learning Academy (http
s://www.mygreatlearning.com/academy), our commitment to aid free learning, offers 1200+
completely FREE courses in the most in-demand job domains. In addition to videos, you ca
n earn certificates of completion, do assignments and projects, attend live sessions and
interact with top faculty & industry experts.\n"},
    'country': 'IN'},
   'contentDetails': {'relatedPlaylists': {'likes': '',
     'uploads': 'UUObs0kLIrDjX2LLSybqNaEA'}},
   'statistics': {'viewCount': '93179185',
    'subscriberCount': '763000',
    'hiddenSubscriberCount': False,
    'videoCount': '1821'}}]}
```

In [271…

```
#open https://jsonformatter.curiousconcept.com/# and paste the above output
#For each channel id deails are stored in each item
#Under content details we can find "uploads":"UUsvqVGtbbyHaMoevxPAq9Fg",to get playlistI

# The playlist ID is useful because it allows us to retrieve all the videos uploaded to
# which can be useful for various purposes.
```

In [272…

```
def get_channel_stats(youtube,channel_ids):
    request = youtube.channels().list(part='snippet,contentDetails,statistics',id=','.jo
    response = request.execute()
    all_data = []
    for i in range(len(response['items'])):
        data = dict(Channel_name = response['items'][i]['snippet']['title'],
                    Subscribers = response['items'][i]['statistics']['subscriberCount'],
                    Views = response['items'][i]['statistics']['viewCount'],
                    Total_videos = response['items'][i]['statistics']['videoCount'],
                    playlist_id = response['items'][i]['contentDetails']['relatedPlaylis
        all_data.append(data)

    return all_data
```

```
# The "data" dictionary is then appended to the "all_data" list, which contains the stat

# Finally, the function returns the "all_data" list as the output.
# This list contains a dictionary for each channel, with the channel name, subscriber co
# and playlist ID for the channel's "uploads" playlist.
# This data can be used to analyze and compare the statistics of multiple YouTube channe
```

In [273…

```
get_channel_stats(youtube,channel_ids)
```

Out[273]:

```
[{'Channel_name': 'Great Learning',
  'Subscribers': '763000',
  'Views': '93179185',
  'Total_videos': '1821',
  'playlist_id': 'UUObs0kLIrDjX2LLSybqNaEA'},
 {'Channel_name': 'Simplilearn',
  'Subscribers': '2970000',
  'Views': '299801087',
  'Total_videos': '5448',
  'playlist_id': 'UUsvqVGtbbyHaMoevxPAq9Fg'},
 {'Channel_name': 'Intellipaat',
  'Subscribers': '1680000',
  'Views': '120061467',
  'Total_videos': '3547',
  'playlist_id': 'UUCktnahuRFYIBtNnKT5IYyg'}]
```

In [274…

```
channel_stats = get_channel_stats(youtube,channel_ids)
data = pd.DataFrame(channel_stats)
data
```

Out[274]:

	Channel_name	Subscribers	Views	Total_videos	playlist_id
0	Great Learning	763000	93179185	1821	UUObs0kLIrDjX2LLSybqNaEA
1	Simplilearn	2970000	299801087	5448	UUsvqVGtbbyHaMoevxPAq9Fg
2	Intellipaat	1680000	120061467	3547	UUCktnahuRFYIBtNnKT5IYyg

In [ ]:

checking & updating the data type of the columns

In [275…

```
data.dtypes
```

Out[275]:

```
Channel_name    object
Subscribers     object
Views           object
Total_videos    object
playlist_id     object
dtype: object
```

In [276…

```
#Converting data type from object to a int

data['Subscribers'] = pd.to_numeric(data['Subscribers'])
data['Views'] = pd.to_numeric(data['Views'])
data['Total_videos'] = pd.to_numeric(data['Total_videos'])

data.dtypes
```

Out[276]:

```
Channel_name    object
Subscribers      int64
Views            int64
Total_videos     int64
```

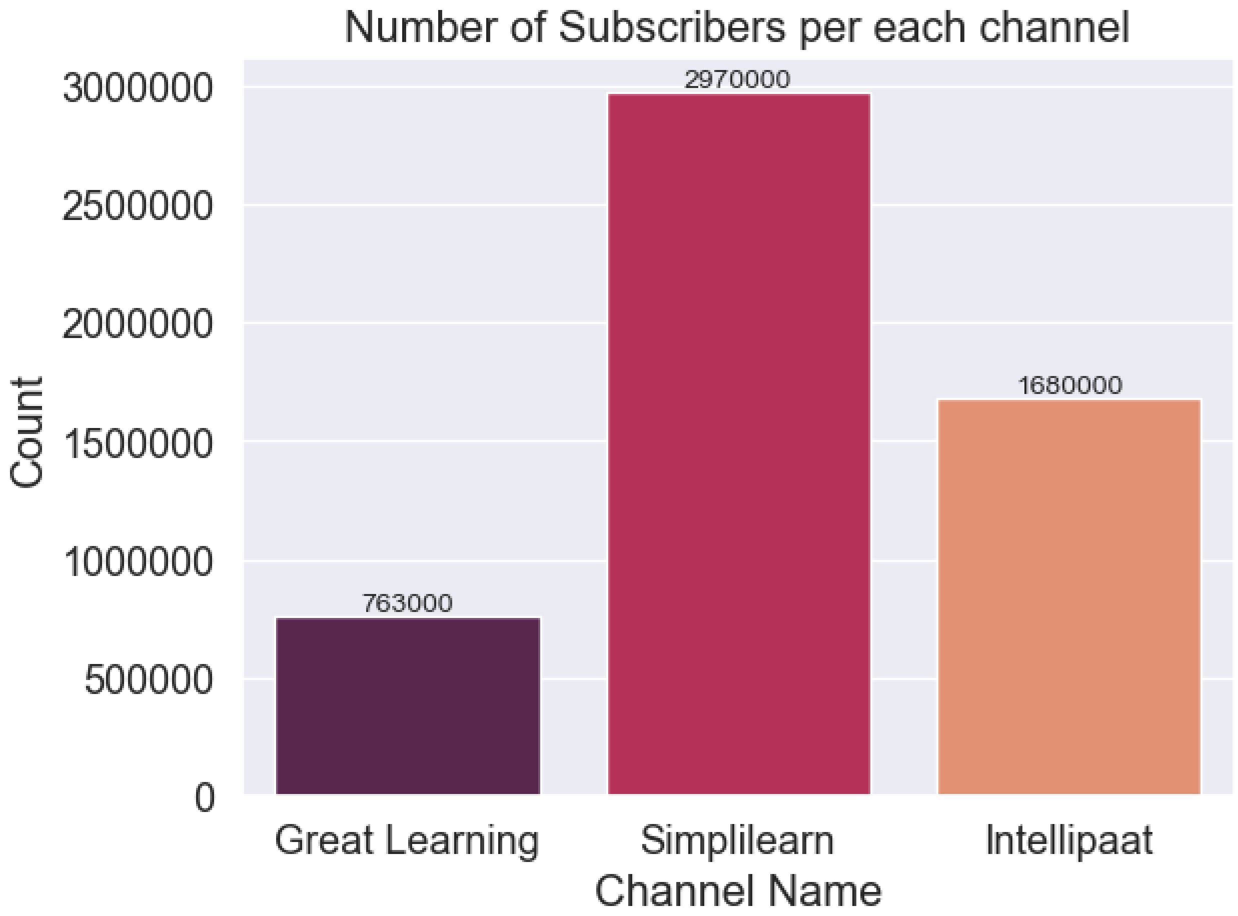
```
playlist_id     object
dtype: object
```

In [ ]:

Creating plots to visualize the data

In [277…

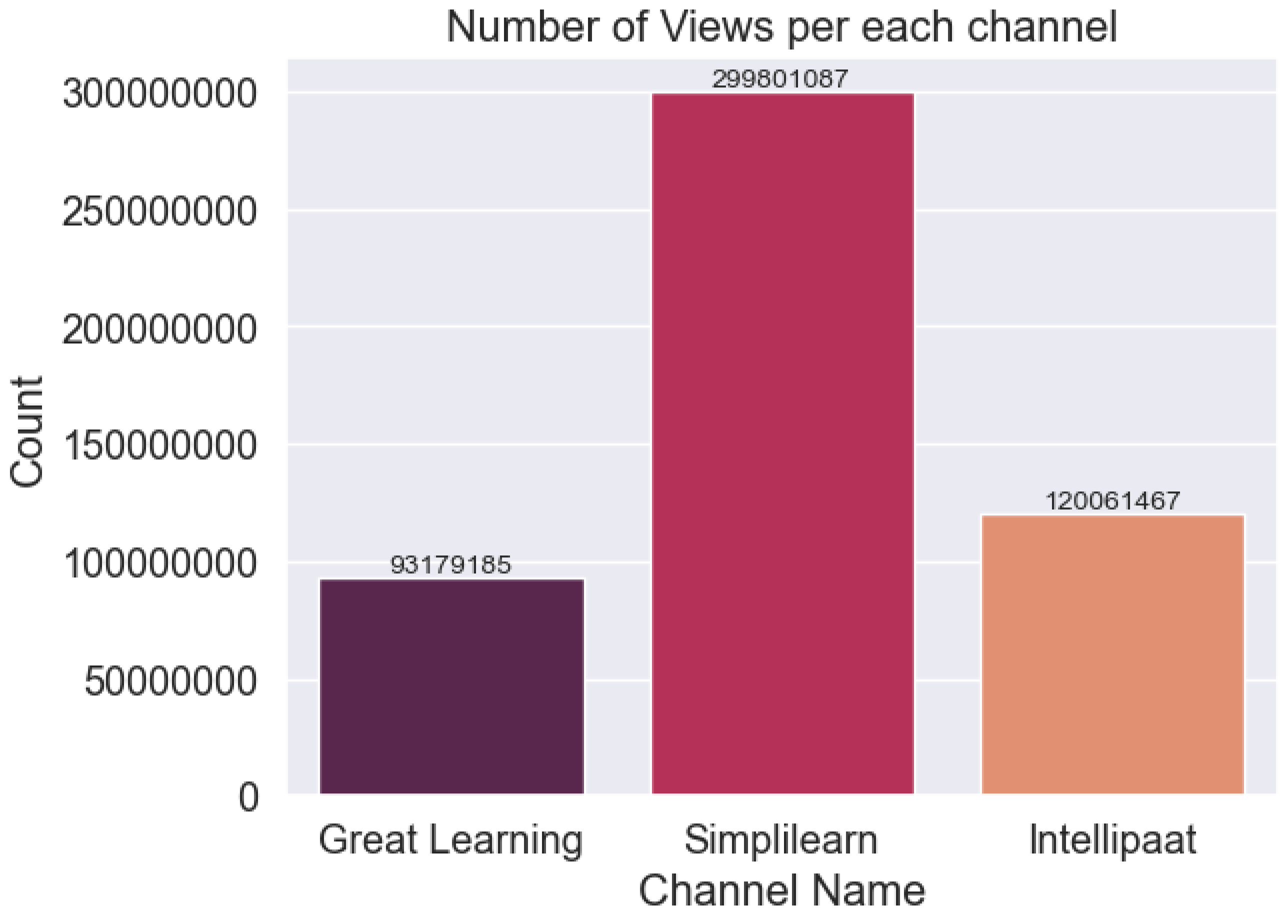
```python
sns.set(font_scale=1.3)
sns.set_style("darkgrid")
ax = sns.barplot(x='Channel_name', y='Subscribers', data=data,palette="rocket")
plt.title('Number of Subscribers per each channel')
plt.xlabel('Channel Name')
plt.ylabel('Count')

plt.ticklabel_format(style='plain', axis='y')  # turn off scientific notation on y-axis


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')

# fmt='%.0f' specifies that the labels should be formatted as floating point numbers wit
```



In [278…

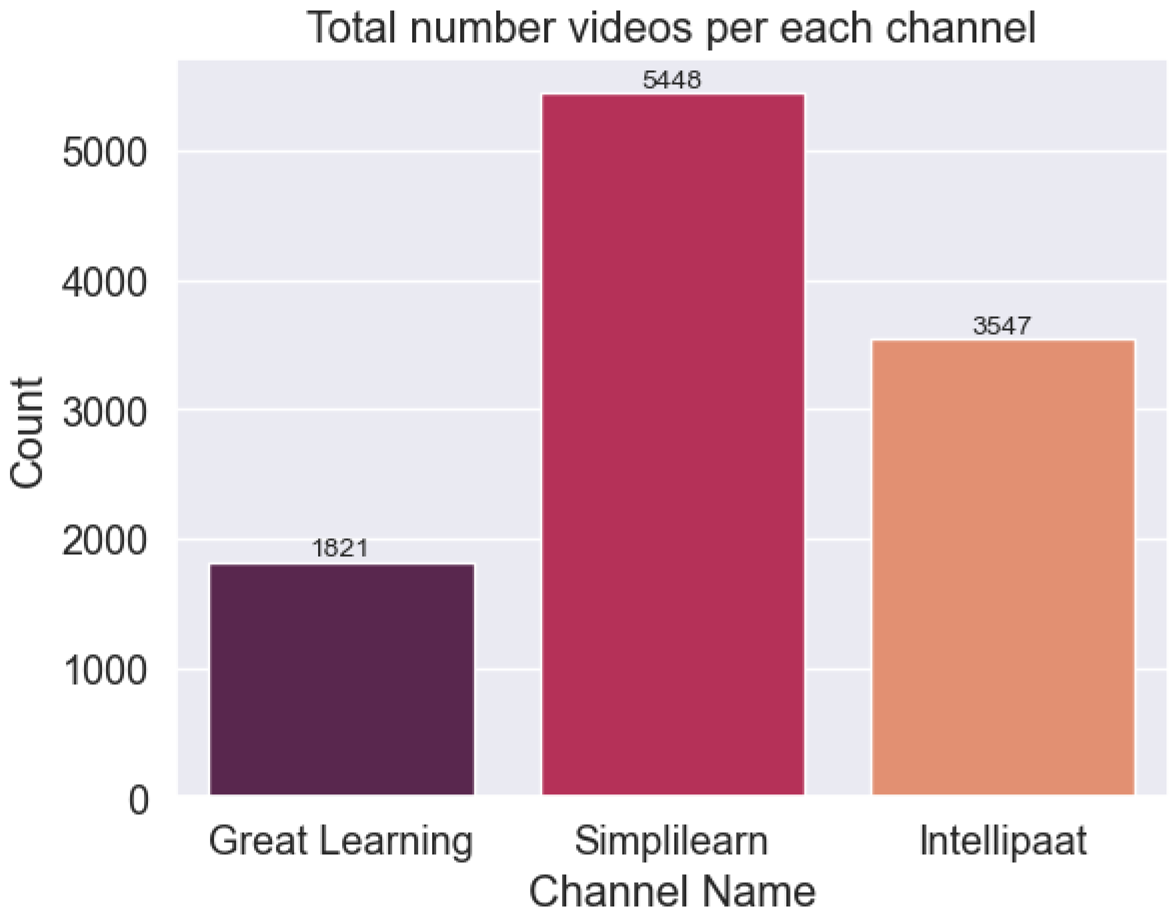
```python
sns.set(font_scale=1.3)
sns.set_style("darkgrid")
ax = sns.barplot(x='Channel_name', y='Views', data=data,palette="rocket")
plt.title('Number of Views per each channel')
plt.xlabel('Channel Name')
plt.ylabel('Count')

plt.ticklabel_format(style='plain', axis='y')  # turn off scientific notation on y-axis


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
```

```
# fmt='%.0f' specifies that the labels should be formatted as floating point numbers wit
```



In [279…

```
sns.set(font_scale=1.3)
sns.set_style("darkgrid")
ax = sns.barplot(x='Channel_name', y='Total_videos', data=data,palette="rocket")
plt.title('Total number videos per each channel')
plt.xlabel('Channel Name')
plt.ylabel('Count')


plt.ticklabel_format(style='plain', axis='y')   # turn off scientific notation on y-axis


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')

# fmt='%.0f' specifies that the labels should be formatted as floating point numbers wit
```



In [ ]:

To get video details of the channels using Video Id's

In [ ]:

To get all the Video Id's of the channel

https://developers.google.com/youtube/v3/docs/pla

PlaylistItems: list Returns a collection of playlist items that match the API request parameters. You can retrieve all of the playlist items in a specified playlist or retrieve one or more playlist items by their unique IDs.

In [280…

```
data
```

Out[280]:

	Channel_name	Subscribers	Views	Total_videos	playlist_id
0	Great Learning	763000	93179185	1821	UUObs0kLIrDjX2LLSybqNaEA
1	Simplilearn	2970000	299801087	5448	UUsvqVGtbbyHaMoevxPAq9Fg
2	Intellipaat	1680000	120061467	3547	UUCktnahuRFYIBtNnKT5IYyg

In [281…

```
#To Get Simplilearn playlist id from data
playlist_id_Simplilearn = data.loc[data['Channel_name']=='Simplilearn', 'playlist_id'].i
playlist_id_Simplilearn
```

```
#To Get Intellipaat playlist id from data
playlist_id_Intellipaat = data.loc[data['Channel_name']=='Intellipaat', 'playlist_id'].i
playlist_id_Intellipaat

#To Get Great Learning playlist id from data
playlist_id_GreatLearning = data.loc[data['Channel_name']=='Great Learning', 'playlist_i
playlist_id_GreatLearning

# "iloc[0]" method is used to retrieve the first row of the filtered DataFrame.
```

Out[281]:

```
'UUObs0kLIrDjX2LLSybqNaEA'
```

In [ ]:

In [282…

```
#open https://jsonformatter.curiousconcept.com/# and paste the above output
# We can observe that "totalResults":4698, "resultsPerPage":5
# updating the above function
```

In [283…

```
#The maxResults parameter specifies the maximum number of items that should be returned
#Acceptable values are 0 to 50, inclusive. The default value is 5.

#nextPageToken is present in json , which can be used to extract al the data
```

In [284…

```
def get_video_ids(youtube,playlist_id):

    request = youtube.playlistItems().list(part = "contentDetails",playlistId = playlist
    response = request.execute()

    video_ids = []


    for i in range(len(response['items'])):
        video_ids.append(response['items'][i]['contentDetails']['videoId'])

    next_page_token = response.get('nextPageToken')
    more_pages = True

    while more_pages:
        if next_page_token is None:
            more_pages = False
        else:
            request = youtube.playlistItems().list(
                        part='contentDetails',
                        playlistId = playlist_id,
                        maxResults = 50,
                        pageToken = next_page_token)
            response = request.execute()

            for i in range(len(response['items'])):
                video_ids.append(response['items'][i]['contentDetails']['videoId'])

            next_page_token = response.get('nextPageToken')

    return video_ids


# The "get()" method is called on the "response" object with the "nextPageToken" paramet
# if there are more pages of results. If there are more pages, the "more_pages" variable
# and a while loop is used to retrieve the video IDs from the next page of results.
# Inside the while loop, a new "playlistItems().list()" request is made with the same pa
# but with an additional "pageToken" parameter set to the "nextPageToken" value from the
# This retrieves the next page of results, and the video IDs are extracted and added to
# This loop continues until there are no more pages of results.
```

In [285…

```
video_ids_Simplilearn = get_video_ids(youtube,playlist_id_Simplilearn)
# video_ids_Simplilearn
```

In [ ]:

```
```

In [286…

```
video_ids_Intellipaat = get_video_ids(youtube,playlist_id_Intellipaat)
# video_ids_Intellipaat
```

In [ ]:

```
```

In [287…

```
video_ids_GreatLearning = get_video_ids(youtube,playlist_id_GreatLearning)
# video_ids_GreatLearning
```

In [288…

```
# We have extracted the video id's for all the channels
```

In [ ]:

```
```

To get video details

https://developers.google.com/youtube/v3/docs/videos/list

Videos: list

Returns a list of videos that match the API request parameters.

In [289…

```
# def get_video_detail(youtube, video_ids):


#     for i in range(0, len(video_ids), 50):
#         request = youtube.videos().list(
#                     part='snippet,statistics',
#                     id=','.join(video_ids[:50]))
#         response = request.execute()
#         return response
```

In [290…

```
# get_video_detail(youtube, video_ids_Simplilearn)
```

In [291…

```
#open https://jsonformatter.curiousconcept.com/# and paste the above output
#Under statistics we can find details like "viewCount","likeCount","favoriteCount","comm
# updating the above function
```

In [292…

```
def get_video_details(youtube, video_ids):
    all_video_stats = []

    for i in range(0, len(video_ids), 50):
        request = youtube.videos().list(
                    part='snippet,statistics',
                    id=','.join(video_ids[i:i+50]))
        response = request.execute()


        for video in response['items']:
            video_stats = dict(Title=video['snippet']['title'],
                       Published_date=video['snippet']['publishedAt'],
                       Views=video['statistics'].get('viewCount', 0),
                       Likes=video['statistics'].get('likeCount', 0),
                       Dislikes=video['statistics'].get('dislikeCount', 0),
                       Comments=video['statistics'].get('commentCount', 0)
                      )
            all_video_stats.append(video_stats)

    return all_video_stats
```

In [ ]:

In [293…

```
video_details_Simplilearn = get_video_details(youtube, video_ids_Simplilearn)
# video_details_Simplilearn
```

In [ ]:

In [294…

```
video_details_Intellipaat = get_video_details(youtube, video_ids_Intellipaat)
# video_details_Intellipaat
```

In [ ]:

In [295…

```
video_details_GreatLearning = get_video_details(youtube, video_ids_GreatLearning)
# video_details_GreatLearning
```

In [ ]:

To get video Data

In [ ]:

In [296…

```
video_data_Simplilearn = pd.DataFrame(video_details_Simplilearn)
```

In [297…

```
video_data_Simplilearn['Published_date'] = pd.to_datetime(video_data_Simplilearn['Publis
video_data_Simplilearn['Views'] = pd.to_numeric(video_data_Simplilearn['Views'])
video_data_Simplilearn['Likes'] = pd.to_numeric(video_data_Simplilearn['Likes'])
video_data_Simplilearn['Dislikes'] = pd.to_numeric(video_data_Simplilearn['Dislikes'])
video_data_Simplilearn['Views'] = pd.to_numeric(video_data_Simplilearn['Views'])
video_data_Simplilearn['Comments'] = pd.to_numeric(video_data_Simplilearn['Comments'])
```

In [298…

```
video_data_Simplilearn
```

Out[298]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	🔥 Become an Ethical Hacking Expert in 10 Hours ...	2023-04-21	200	6	0	0
1	🔥 DevOps Week Day - 2 | Docker And Kubernetes F...	2023-04-21	84	3	0	0
2	🔥 🔥 Gain The Skills And Experience With Caltech...	2023-04-21	324	9	0	0
3	🚀 Here's How Rose Switched From Teaching To Di...	2023-04-21	524	24	0	0
4	🔥 Top 5 Reasons to Learn Machine Learning | Why...	2023-04-21	834	20	0	0
...	...	...	...	...	...	...
5449	ISTQB Foundation Exam | Software Testing Train...	2012-05-23	6951	3	0	0
5450	What is Software Testing? | Agenda of CTFL Tra...	2012-05-23	5863	4	0	0
5451	What is Software Testing? | CTFL Software Test...	2012-05-23	12506	10	0	0
5452	Minitab Screen Layout | Minitab Tutorial Onlin...	2012-01-04	9720	11	0	0
5453	Process Groups And Knowledge Areas | CAPM Cert...	2012-01-02	2669	19	0	4

5454 rows × 6 columns

In [ ]:

```

```

In [299…

```
video_data_Intellipaat = pd.DataFrame(video_details_Intellipaat)
```

In [300…

```
video_data_Intellipaat['Published_date'] = pd.to_datetime(video_data_Intellipaat['Publis
video_data_Intellipaat['Views'] = pd.to_numeric(video_data_Intellipaat['Views'])
video_data_Intellipaat['Likes'] = pd.to_numeric(video_data_Intellipaat['Likes'])
video_data_Intellipaat['Dislikes'] = pd.to_numeric(video_data_Intellipaat['Dislikes'])
video_data_Intellipaat['Views'] = pd.to_numeric(video_data_Intellipaat['Views'])
video_data_Intellipaat['Comments'] = pd.to_numeric(video_data_Intellipaat['Comments'])
```

In [301…

```
video_data_Intellipaat
```

Out[301]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	Complete Data Analyst Roadmap | How to become ...	2023-04-21	188	22	0	0
1	Tag all the drama loves you know down in the c...	2023-04-21	1433	23	0	0
2	What Is Docker | Docker Tutorial For Beginners...	2023-04-20	759	33	0	4
3	Data Science RoadMap - How to become a Data Sc...	2023-04-20	723	40	0	4
4	Landed a Job Abroad with more than 100% of Hik...	2023-04-20	911	33	0	3
...	...	...	...	...	...	...
3542	Big Data Hadoop Tutorial for Beginners | What ...	2013-04-08	69637	378	0	196
3543	Hadoop Administration Tutorial | Hadoop Tutori...	2013-04-06	17941	106	0	93
3544	Hbase Training | Hbase Tutorial | Online Hbase...	2013-03-26	51098	228	0	109
3545	Pentaho Training | Pentaho Tutorial | Pentaho ...	2013-03-25	61864	286	0	135
3546	Hadoop training feedback- VP, Sony Corporation	2013-02-25	4827	68	0	0

3547 rows × 6 columns

In [ ]:

```

```

In [302…

```
video_data_GreatLearning = pd.DataFrame(video_details_GreatLearning)
```

In [303…

```
video_data_GreatLearning['Published_date'] = pd.to_datetime(video_data_GreatLearning['Pu
video_data_GreatLearning['Views'] = pd.to_numeric(video_data_GreatLearning['Views'])
video_data_GreatLearning['Likes'] = pd.to_numeric(video_data_GreatLearning['Likes'])
video_data_GreatLearning['Dislikes'] = pd.to_numeric(video_data_GreatLearning['Dislikes'
video_data_GreatLearning['Views'] = pd.to_numeric(video_data_GreatLearning['Views'])
video_data_GreatLearning['Comments'] = pd.to_numeric(video_data_GreatLearning['Comments'
```

In [304…

```
video_data_GreatLearning
```

Out[304]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	ChatGPT tutorials for Beginners - How to use C...	2023-04-20	1518	86	0	3
1	How Generative AI will transform our lives in ...	2023-04-18	901	32	0	5
2	All about admissions for MS in Machine Learnin...	2023-04-14	836	20	0	2
3	Boost your Career Prospects in the US after Ma...	2023-04-14	891	27	0	3
4	Is AI coming for human jobs? |Which jobs are m...	2023-04-11	1256	48	0	3

...	...	...	...	...	...	...
1817	Regression Analysis using Excel Hands-On | Pre...	2014-02-18	6319	8	0	0
1818	Demand Forecasting Methods | Importance of Dem...	2014-02-18	9769	21	0	0
1819	Regression Analysis using Excel Hands-On | Pre...	2013-07-18	3651	24	0	0
1820	Regression Analysis using Excel Hands-On | Pre...	2013-07-12	916	12	0	0
1821	Regression Analysis using Excel Hands-On | Pre...	2013-07-12	1816	24	0	1

1822 rows × 6 columns

In [ ]:

To compare the popularity of the channels in the year 2023

In [ ]:

Filter rows where Published_date is in 2023

In [305…

```
# GreatLearning
# Convert Published_date column to datetime type
video_data_GreatLearning['Published_date'] = pd.to_datetime(video_data_GreatLearning['Pu
GreatLearning_2023 = video_data_GreatLearning.loc[video_data_GreatLearning['Published_da

# Print the filtered rows
GreatLearning_2023
```

Out[305]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	ChatGPT tutorials for Beginners - How to use C...	2023-04-20	1518	86	0	3
1	How Generative AI will transform our lives in ...	2023-04-18	901	32	0	5
2	All about admissions for MS in Machine Learnin...	2023-04-14	836	20	0	2
3	Boost your Career Prospects in the US after Ma...	2023-04-14	891	27	0	3
4	Is AI coming for human jobs? |Which jobs are m...	2023-04-11	1256	48	0	3
5	Challenges faced by Study Abroad aspirants | M...	2023-04-06	954	19	0	1
6	How to fund your Masters in the U.S. | Study A...	2023-04-05	862	27	0	1
7	Top 6 Reasons Why Indian students should study...	2023-03-29	1422	32	0	1
8	How I cracked an Analyst job in 4months | 2600...	2023-03-25	2406	59	0	0
9	#AddWeightToYourCV with Top Management Program...	2023-02-27	7763	132	0	10
10	Great Learning Career Academy | #NoMoreExcuses	2023-01-25	10038	184	0	7

In [ ]:

In [306…

```
# Intellipaat
# Convert Published_date column to datetime type
video_data_Intellipaat['Published_date'] = pd.to_datetime(video_data_Intellipaat['Publis
Intellipaat_2023 = video_data_Intellipaat.loc[video_data_Intellipaat['Published_date'].d

# Print the filtered rows
Intellipaat_2023
```

Out[306]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	Complete Data Analyst Roadmap | How to become ...	2023-04-21	188	22	0	0
1	Tag all the drama loves you know down in the c...	2023-04-21	1433	23	0	0
2	What Is Docker | Docker Tutorial For Beginners...	2023-04-20	759	33	0	4
3	Data Science RoadMap - How to become a Data Sc...	2023-04-20	723	40	0	4
4	Landed a Job Abroad with more than 100% of Hik...	2023-04-20	911	33	0	3
...	...	...	...	...	...	...
230	Beginner's Guide to Electric Vehicles | Types ...	2023-01-02	2018	34	0	2
231	Career Opportunities in Electric Vehicle Indus...	2023-01-02	1286	22	0	3
232	Data Science Interview Questions | Top 40 Data...	2023-01-01	11925	285	0	11
233	Top 3 Most Important UI/UX Concepts | UI/UX | ...	2023-01-01	2500	110	0	0
234	Intellipaat Salesforce Training Review | Best ...	2023-01-01	1799	15	0	0

235 rows × 6 columns

In [ ]:

In [307…

```
# Intellipaat
# Convert Published_date column to datetime type
video_data_Simplilearn['Published_date'] = pd.to_datetime(video_data_Simplilearn['Publis
Simplilearn_2023 = video_data_Simplilearn.loc[video_data_Simplilearn['Published_date'].d

# Print the filtered rows
Simplilearn_2023
```

Out[307]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	🔥 Become an Ethical Hacking Expert in 10 Hours ...	2023-04-21	200	6	0	0
1	🔥 DevOps Week Day - 2 | Docker And Kubernetes F...	2023-04-21	84	3	0	0
2	🔥 🔥 Gain The Skills And Experience With Caltech...	2023-04-21	324	9	0	0
3	🚀 Here's How Rose Switched From Teaching To Di...	2023-04-21	524	24	0	0
4	🔥 Top 5 Reasons to Learn Machine Learning | Why...	2023-04-21	834	20	0	0
...	...	...	...	...	...	...
539	Neural Network In Artificial Intelligence | Ne...	2023-01-03	2450	76	0	1
540	How To Blur WhatsApp Messages? 🤩 😄 #Shorts | Si...	2023-01-03	4759	183	0	1
541	Top 50 Accenture Interview Questions and Answe...	2023-01-02	2389	48	0	1
542	AI Agents And Environments | Types Of AI Agent...	2023-01-02	5065	79	0	1
543	Top 5 Soft Skills To Master In 2022 🤩 😄 #Shorts...	2023-01-01	3241	151	0	1

544 rows × 6 columns

In [ ]:

To calculate the engagement rate & average number of views, likes, and comments per video for each channel

In [ ]:

```
```

Great Learning

In [308…

```
# To Calculate the sum of each column
GreatLearning_2023_total_views = GreatLearning_2023['Views'].sum()
GreatLearning_2023_total_likes = GreatLearning_2023['Likes'].sum()
GreatLearning_2023_total_comments = GreatLearning_2023['Comments'].sum()
GreatLearning_2023_total_videos = GreatLearning_2023.shape[0]
# Print the total views, likes, and comments for GreatLearning in 2023
print("In 2023, GreatLearning uploaded {} videos and had a total of {} views, {} likes,
```

```
In 2023, GreatLearning uploaded 11 videos and had a total of 28847 views, 666 likes, and
36 comments.
```

In [ ]:

```
```

In [309…

```
# To Calculate the average number of views, likes, and comments per video

GreatLearning_2023_avg_views = GreatLearning_2023_total_views/GreatLearning_2023_total_v
GreatLearning_2023_avg_likes = GreatLearning_2023_total_likes/GreatLearning_2023_total_v
GreatLearning_2023_avg_comments = GreatLearning_2023_total_comments/GreatLearning_2023_t

print("In 2023,on an average GreatLearning had a total of {} views, {} likes, and {} com
```

```
In 2023,on an average GreatLearning had a total of 2622.4545454545455 views, 60.54545454
545455 likes, and 3.272727272727273 comments.
```

In [ ]:

```
```

In [310…

```
# To calculate engagement rate

GreatLearning_engagement_rate_2023 = (GreatLearning_2023_total_likes + GreatLearning_202
                                       /GreatLearning_2023_total_views
print("The total engagement rate for GreatLearning is {:.2%}.".format(GreatLearning_enga


#  The :.2% format specifier is used to format the engagement rate as a percentage with
# The % character is used to indicate that the value should be displayed as a percentage
```

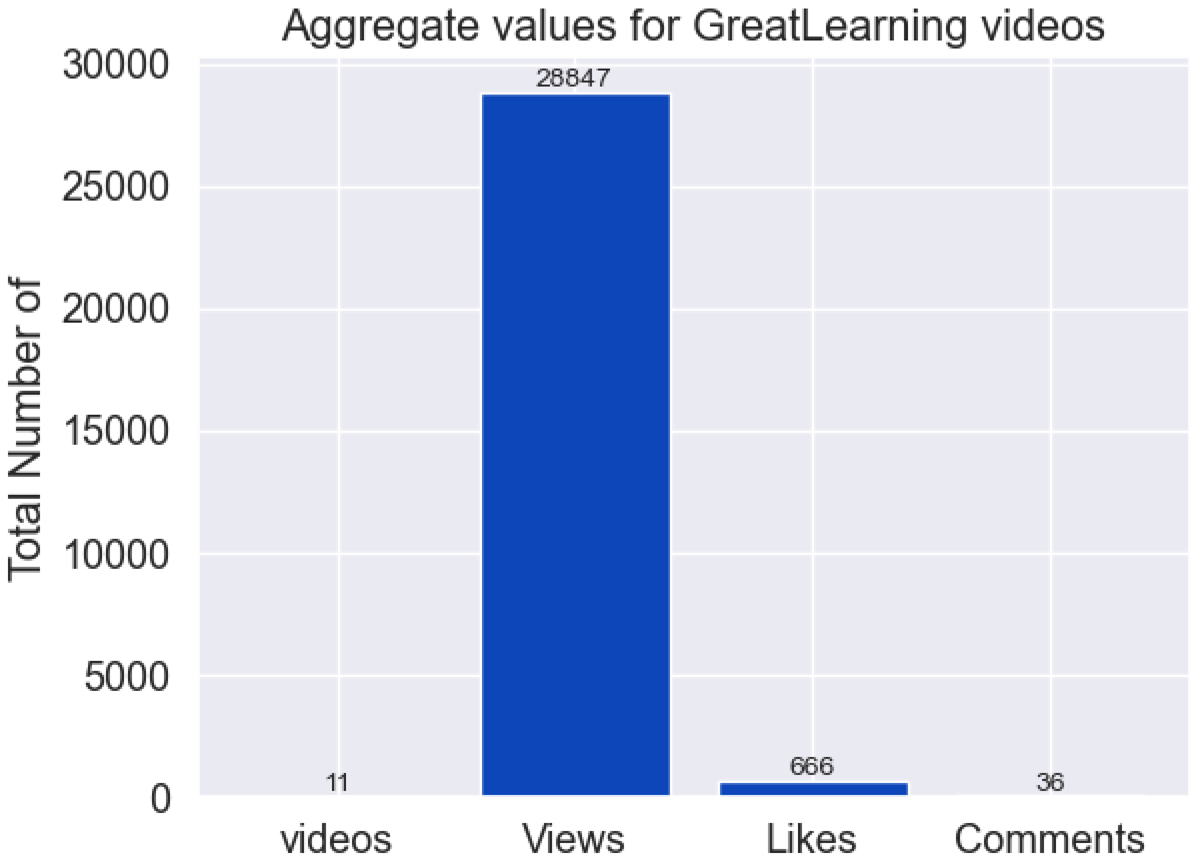
```
The total engagement rate for GreatLearning is 2.43%.
```

In [ ]:

```
```

In [311…

```
# Create a bar plot of the total values
fig, ax = plt.subplots()
ax.bar(['videos','Views', 'Likes', 'Comments'],
       [GreatLearning_2023_total_videos,GreatLearning_2023_total_views, GreatLearning_20
        GreatLearning_2023_total_comments],color='#0C46B9')
ax.set_ylabel('Total Number of')
ax.set_title('Aggregate values for GreatLearning videos')


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
plt.show()
```

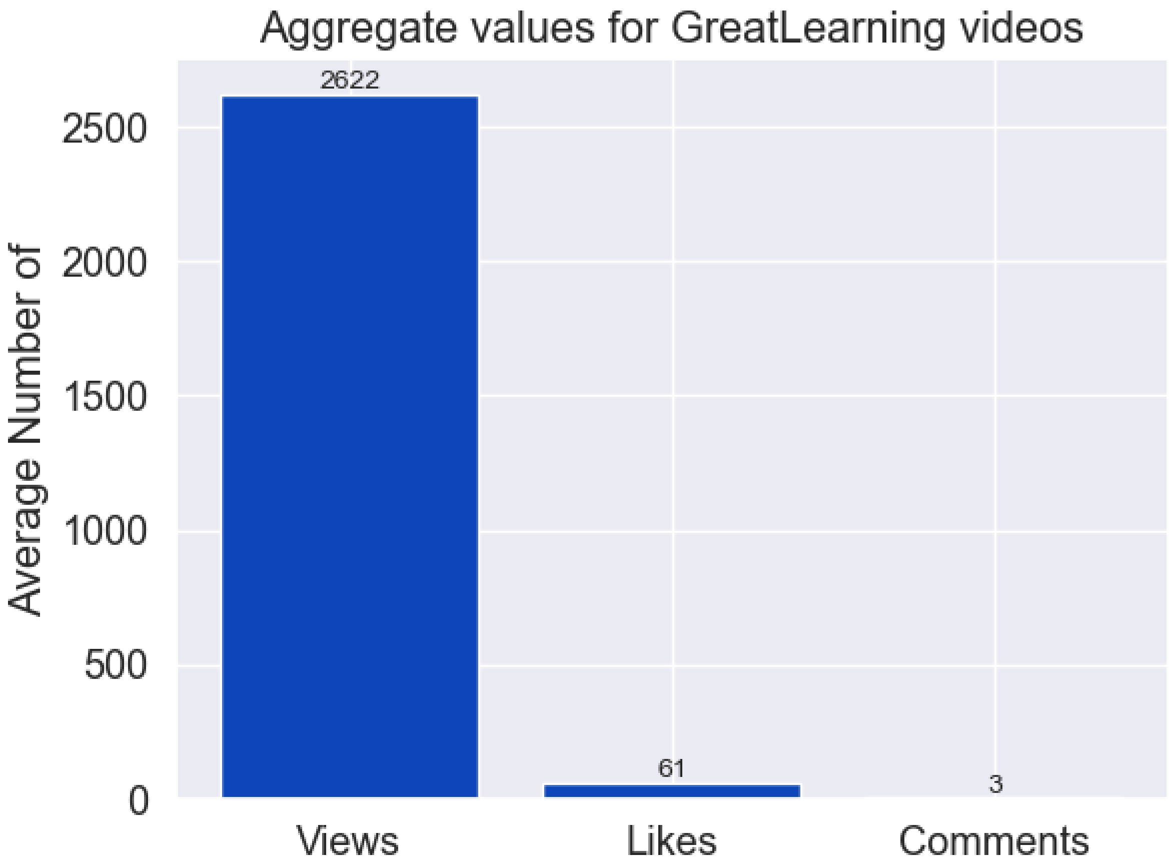


In [ ]:

```
```

In [312…

```
# Create a bar plot of the average values
fig, ax = plt.subplots()
ax.bar(['Views', 'Likes', 'Comments'],
       [GreatLearning_2023_avg_views, GreatLearning_2023_avg_likes, GreatLearning_2023_a
ax.set_ylabel('Average Number of')
ax.set_title('Aggregate values for GreatLearning videos')


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
plt.show()
```



In 2023, GreatLearning uploaded 10 videos and had a total of 27032 views, 575 likes, and 33 comments.

In [ ]:

In [ ]:

Intellipaat

In [313…

```
# Calculate the sum of each column
Intellipaat_2023_total_views = Intellipaat_2023['Views'].sum()
Intellipaat_2023_total_likes = Intellipaat_2023['Likes'].sum()
Intellipaat_2023_total_comments = Intellipaat_2023['Comments'].sum()
Intellipaat_2023_total_videos = Intellipaat_2023.shape[0]
# Print the total views, likes, and comments for GreatLearning in 2023
print("In 2023, Intellipaat uploaded {} videos and had a total of {} views, {} likes, an
```

```
In 2023, Intellipaat uploaded 235 videos and had a total of 947865 views, 21532 likes, a
nd 1074 comments.
```

In [ ]:

In [314…

```
# To Calculate the average number of views, likes, and comments per video

Intellipaat_2023_avg_views = Intellipaat_2023_total_views/Intellipaat_2023_total_videos
Intellipaat_2023_avg_likes = Intellipaat_2023_total_likes/Intellipaat_2023_total_videos
Intellipaat_2023_avg_comments = Intellipaat_2023_total_comments/Intellipaat_2023_total_v

print("In 2023,Intellipaat had a average of {} views, {} likes, and {} comments.".format
```

```
In 2023,Intellipaat had a average of 4033.468085106383 views, 91.62553191489361 likes, a
nd 4.5702127659574465 comments.
```

In [ ]:

In [315…

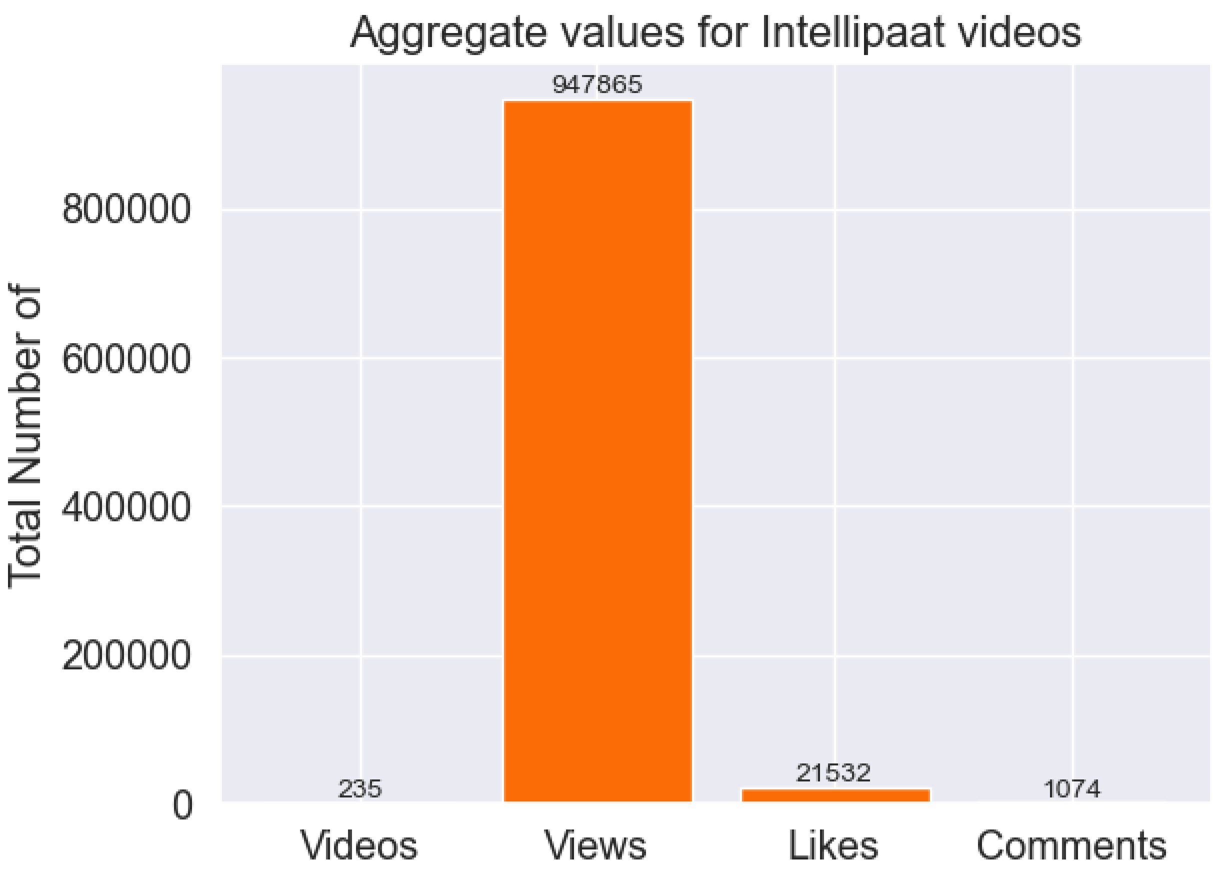
```
# To calculate engagement rate

Intellipaat_engagement_rate_2023 = (Intellipaat_2023_total_likes + Intellipaat_2023_tota
                                                  /Intellipaat_2023_total_views
print("The total engagement rate for Intellipaat is {:.2%}.".format(Intellipaat_engageme


#  The :.2% format specifier is used to format the engagement rate as a percentage with
# The % character is used to indicate that the value should be displayed as a percentage
```

```
The total engagement rate for Intellipaat is 2.38%.
```

In [ ]:

In [316…

```
# Create a bar plot of the total values
fig, ax = plt.subplots()
ax.bar(['Videos','Views', 'Likes', 'Comments'],
       [Intellipaat_2023_total_videos,Intellipaat_2023_total_views, Intellipaat_2023_tot
        Intellipaat_2023_total_comments],color="#FB6B06")
ax.set_ylabel('Total Number of')
ax.set_title('Aggregate values for Intellipaat videos')


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
plt.show()
```

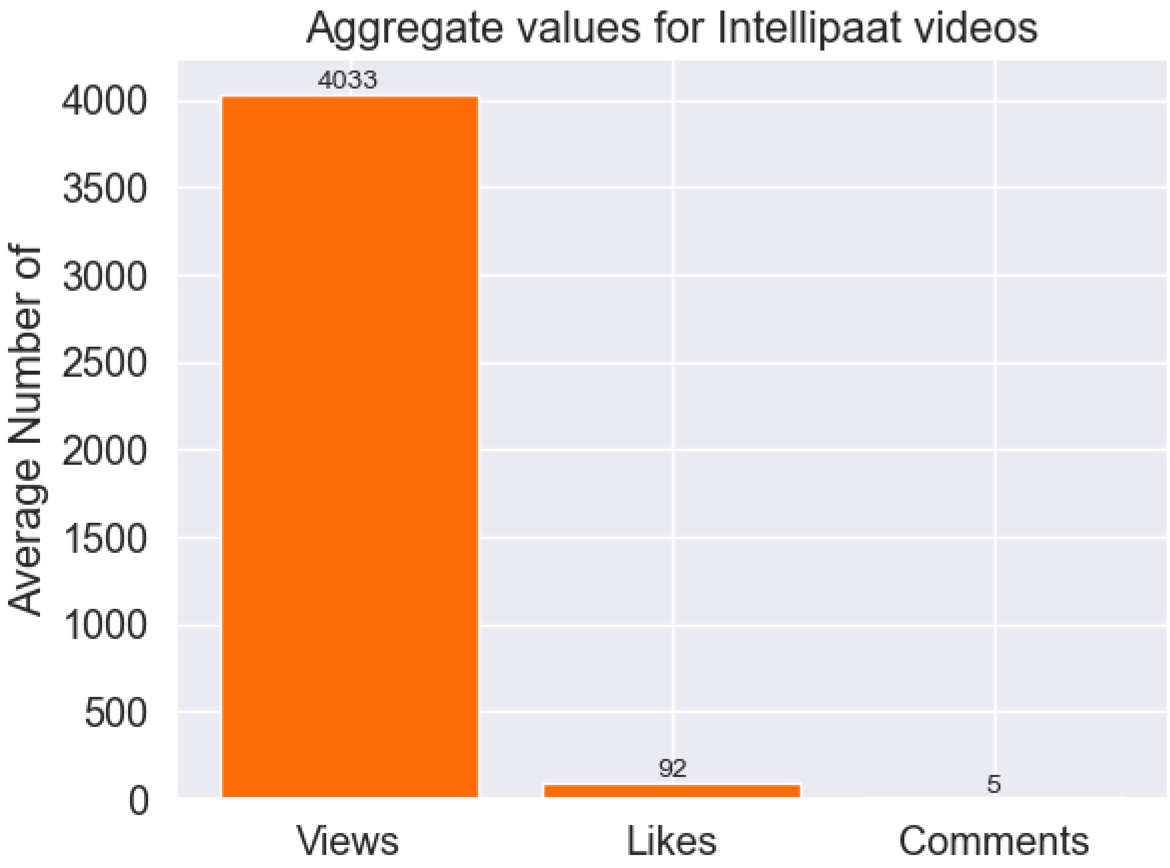


In [ ]:

In [317…

```
# Create a bar plot of the average values
fig, ax = plt.subplots()
ax.bar(['Views', 'Likes', 'Comments'],
       [Intellipaat_2023_avg_views, Intellipaat_2023_avg_likes, Intellipaat_2023_avg_com
ax.set_ylabel('Average Number of')
ax.set_title('Aggregate values for Intellipaat videos')
```

```
for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
plt.show()
```



In 2023, Intellipaat uploaded 230 videos and had a total of 932071 views, 21195 likes, and 1060 comments.

In [ ]:

In [ ]:

Simplilearn

In [318…

```
# Calculate the sum of each column
Simplilearn_2023_total_views = Simplilearn_2023['Views'].sum()
Simplilearn_2023_total_likes = Simplilearn_2023['Likes'].sum()
Simplilearn_2023_total_comments = Simplilearn_2023['Comments'].sum()
Simplilearn_2023_total_videos = Simplilearn_2023.shape[0]
# Print the total views, likes, and comments for GreatLearning in 2023
print("In 2023, Simplilearn uploaded {} videos and had a total of {} views, {} likes, an
```

```
In 2023, Simplilearn uploaded 544 videos and had a total of 4792203 views, 106609 likes,
and 3038 comments.
```

In [ ]:

In [319…

```
# To Calculate the average number of views, likes, and comments per video

Simplilearn_2023_avg_views = Simplilearn_2023_total_views/Simplilearn_2023_total_videos
Simplilearn_2023_avg_likes = Simplilearn_2023_total_likes/Simplilearn_2023_total_videos
Simplilearn_2023_avg_comments = Simplilearn_2023_total_comments/Simplilearn_2023_total_v

print("In 2023,Simplilearn had a average of {} views, {} likes, and {} comments.".format
In 2023,Simplilearn had a average of 8809.19669117647 views, 195.97242647058823 likes, a
nd 5.584558823529412 comments.
```

In [ ]:

```

```

In [320…

```
# To calculate engagement rate

Simplilearn_engagement_rate_2023 = (Simplilearn_2023_total_likes + Simplilearn_2023_tota
                                                     /Simplilearn_2023_total_views
print("The total engagement rate for Simplilearn is {:.2%}.".format(Simplilearn_engageme


#  The :.2% format specifier is used to format the engagement rate as a percentage with
# The % character is used to indicate that the value should be displayed as a percentage
```

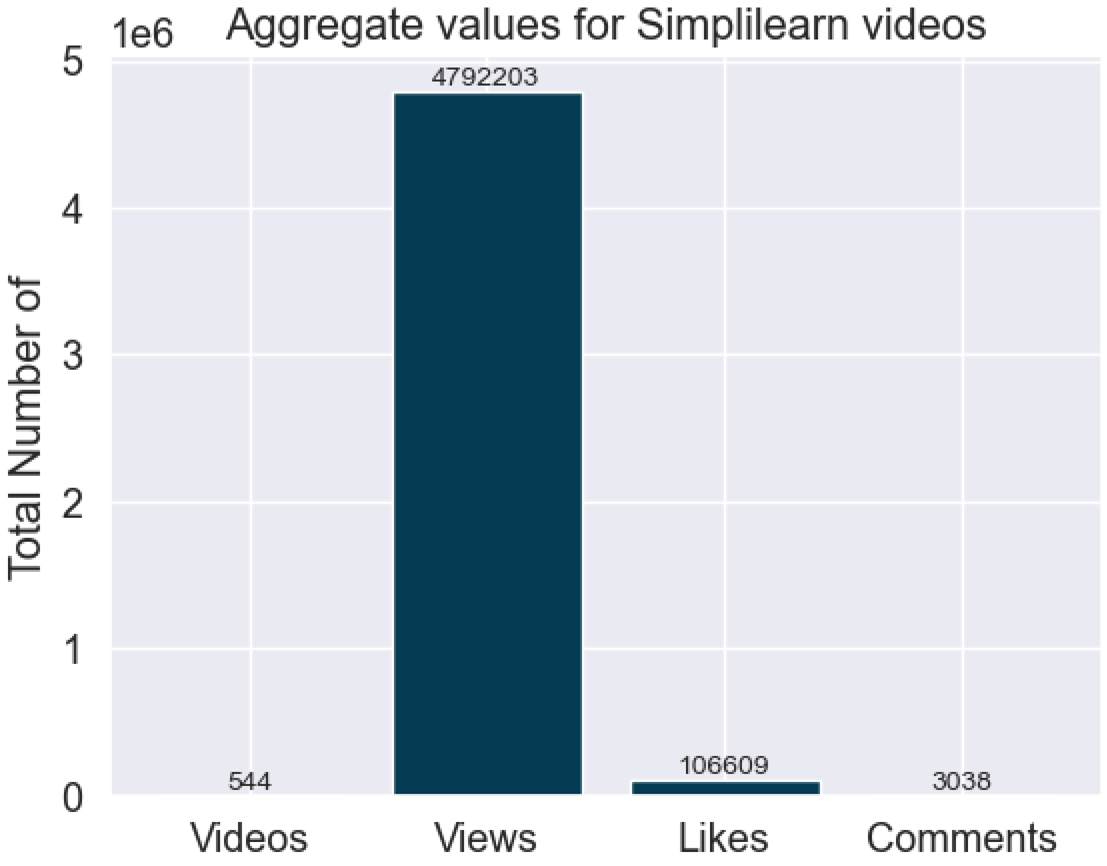
```
The total engagement rate for Simplilearn is 2.29%.
```

In [ ]:

```

```

In [321…

```
# Create a bar plot of the total values
fig, ax = plt.subplots()
ax.bar(['Videos','Views', 'Likes', 'Comments'],
       [Simplilearn_2023_total_videos,Simplilearn_2023_total_views, Simplilearn_2023_tot
        Simplilearn_2023_total_comments],color="#043B53")
ax.set_ylabel('Total Number of')
ax.set_title('Aggregate values for Simplilearn videos')


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
plt.show()
```



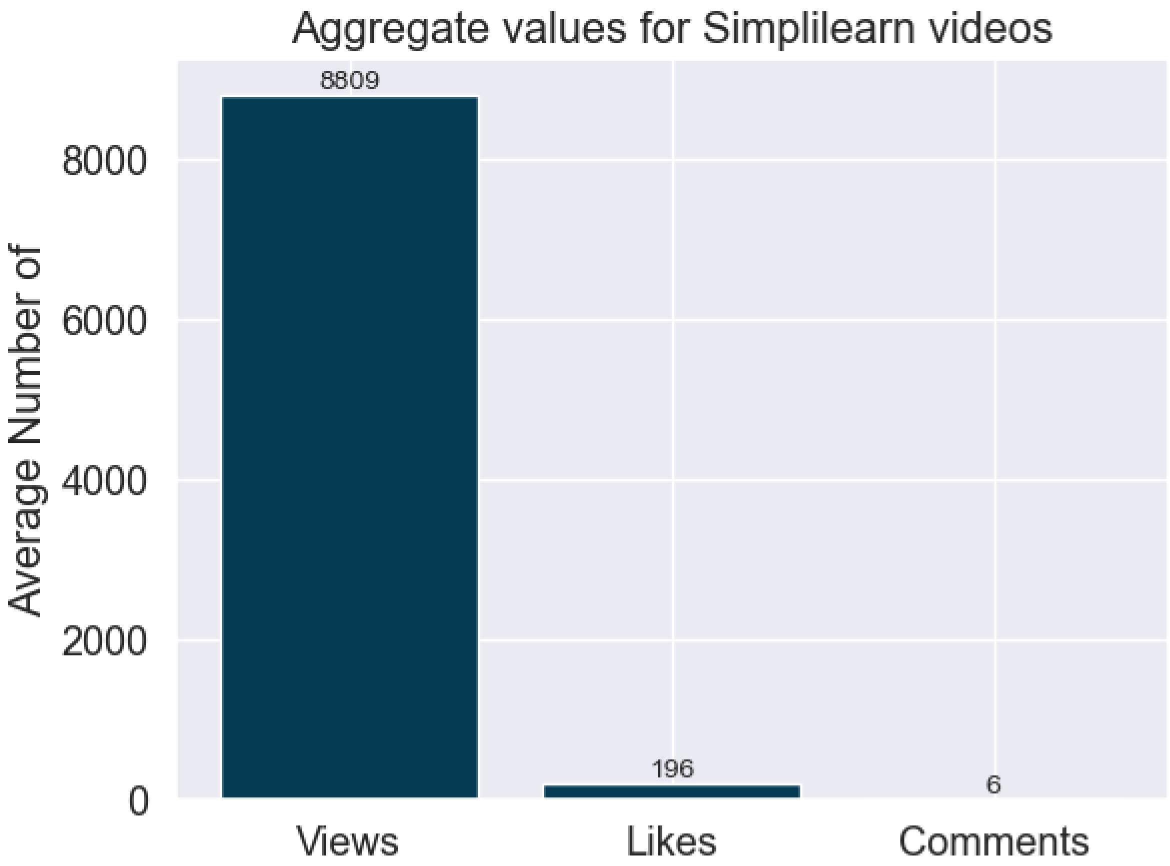
In [ ]:

```

```

In [322…

```
# Create a bar plot of the average values
fig, ax = plt.subplots()
```

```
ax.bar(['Views', 'Likes', 'Comments'],
       [Simplilearn_2023_avg_views, Simplilearn_2023_avg_likes, Simplilearn_2023_avg_com
ax.set_ylabel('Average Number of')
ax.set_title('Aggregate values for Simplilearn videos')


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.0f')
plt.show()
```



In [ ]:

```

```

In 2023, Simplilearn uploaded 529 videos and had a total of 4733201 views, 105351 likes, and 3014 comments.

In [ ]:

```

```

Engagement Rate

In [323…

```
# Create a pandas DataFrame with the  Engagement Rate values
Engagement_rate = pd.DataFrame({'Channel_name': ['GreatLearning', 'Intellipaat','Simplil
                     'Value': [GreatLearning_engagement_rate_2023, Intellipaat_engagement_

Engagement_rate
```

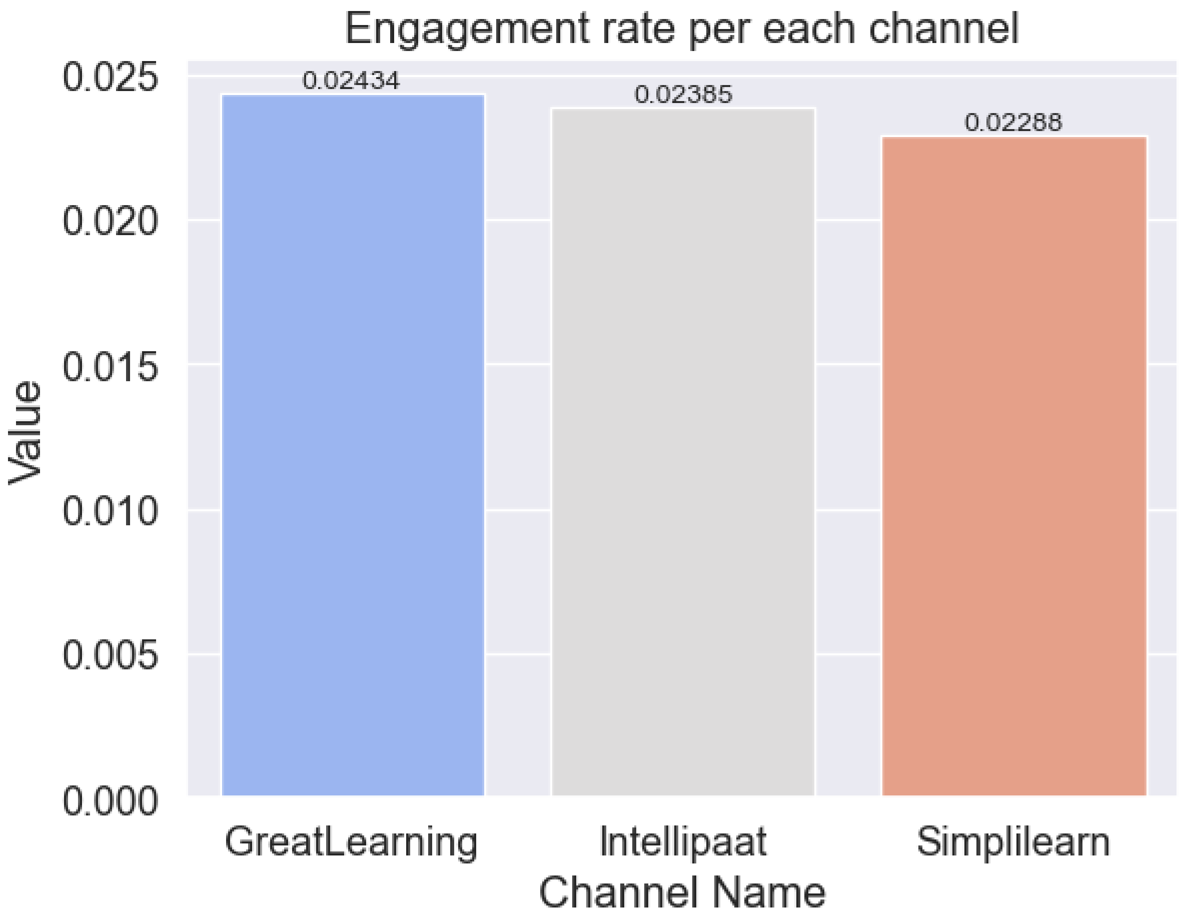
Out[323]:

	Channel_name	Value
0	GreatLearning	0.024335
1	Intellipaat	0.023849
2	Simplilearn	0.022880

In [324…

```
sns.set(font_scale=1.3)
```

```python
sns.set_style("darkgrid")
ax = sns.barplot(x='Channel_name', y='Value', data=Engagement_rate,palette="coolwarm")
plt.title('Engagement rate per each channel')
plt.xlabel('Channel Name')
plt.ylabel('Value')

plt.ticklabel_format(style='plain', axis='y')  # turn off scientific notation on y-axis


for i in ax.containers:
    ax.bar_label(i,label_type='edge', fontsize=10, fmt='%.5f')

# fmt='%.0f' specifies that the labels should be formatted as floating point numbers wit
```



In [ ]:

In [ ]:

The total engagement rate for GreatLearning is 2.25%.

The total engagement rate for Intellipaat is 2.39%.

The total engagement rate for Simplilearn is 2.29%.

In 2023, GreatLearning uploaded 10 videos and had a total of 27032 views, 575 likes, and 33 comments.

Intellipaat uploaded 231 videos and had a total of 935280 views, 21256 likes, and 1062 comments.

Simplilearn uploaded 529 videos and had a total of 4733201 views, 105351 likes, and 3014 comments.

In 2023,on an average GreatLearning had 2703.2 views, 57.5 likes, and 3.3 comments.

In 2023,Intellipaat had a average of 4050.982832618026 views, 91.96995708154506 likes, and 4.592274678111588 comments.

Simplilearn had a average of 8947.44990548204 views, 199.15122873345936 likes, and 5.697542533081285 comments.

Based on Views Simplilearn channel is the most popular among viewers

In [ ]:

```
```

In [ ]:

```
```

To get Top 10 videos of all time

In [ ]:

```
```

GreatLearning

In [325…

```
#Based on Views
```

In [326…

```
GreatLearning_t10 =video_data_GreatLearning.sort_values(by="Views",ascending = False).he
```

In [327…

```
GreatLearning_t10
```

Out[327]:

	Title	Published_date	Views	Likes	Dislikes	Comments
384	Bad appraisal | Great Learning for Great Caree...	2021-09-08	11748844	660	0	45
588	Virat Kohli's Learning Mantra | Great Learning	2021-04-12	11129188	1322	0	52
1091	Great Learning #PowerAhead 30s ad	2020-08-27	7614278	140	0	4
944	Lage Raho #PowerAhead 30 Sec	2020-10-26	4387954	227	0	9
1311	SQL Tutorial for Beginners | SQL Full Course i...	2020-05-08	3057478	54243	0	1876
383	Missed opportunities | Great Learning for Grea...	2021-09-08	2803715	738	0	26
917	Virat's Success Mantra - Seekhte Raho	2020-11-10	2731845	106	0	6
1348	Python Tutorial in Hindi | Learn Python in Hin...	2020-03-14	2135408	59977	0	3382
1105	Great Learning #PowerAhead 50s ad	2020-08-23	2076941	250	0	33
1465	Great Learning Presents The Office Dinosaur Ft...	2019-04-29	1720506	226	0	4

In [328…

```
#Based on likes
```

In [329…

```
GreatLearning_t10L =video_data_GreatLearning.sort_values(by="Likes",ascending = False).h
```

In [330…

```
GreatLearning_t10L
```

Out[330]:

	Title	Published_date	Views	Likes	Dislikes	Comments
1348	Python Tutorial in Hindi | Learn Python in Hin...	2020-03-14	2135408	59977	0	3382
1311	SQL Tutorial for Beginners | SQL Full Course i...	2020-05-08	3057478	54243	0	1876
1300	C++ Tutorial For Beginners in Hindi | C++ Prog...	2020-05-29	1616182	41169	0	1484

1371	Statistics for Data Science | Probability and ...	2019-12-20	1525658	35042	0	786
1354	Machine Learning with Python | Machine Learnin...	2020-02-13	1244384	31825	0	908
1357	Data Science Tutorial | Data Science for Begin...	2020-02-03	930308	27942	0	504
1325	Java Tutorial for Beginners in Hindi | Java in...	2020-04-23	1261860	26934	0	1132
664	Python for Data Science Full Course | Data Sci...	2021-03-07	809477	20810	0	718
1263	AWS Tutorial for beginners in Hindi | AWS Full...	2020-06-20	991769	16497	0	714
620	Build your first mobile app in Python | App De...	2021-03-25	541842	14066	0	401

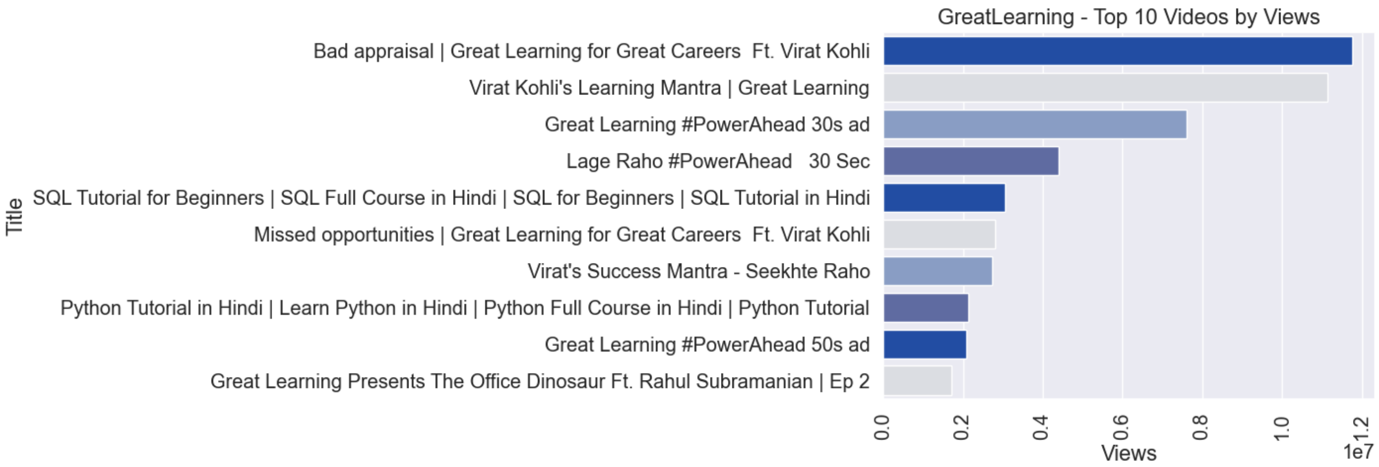
In [ ]:

In [331…

```
# To create a custom palette
GreatLearning_palette = sns.color_palette(["#0C46B9","#dadce3","#7F9ACE","#5464AC"])
```

In [332…

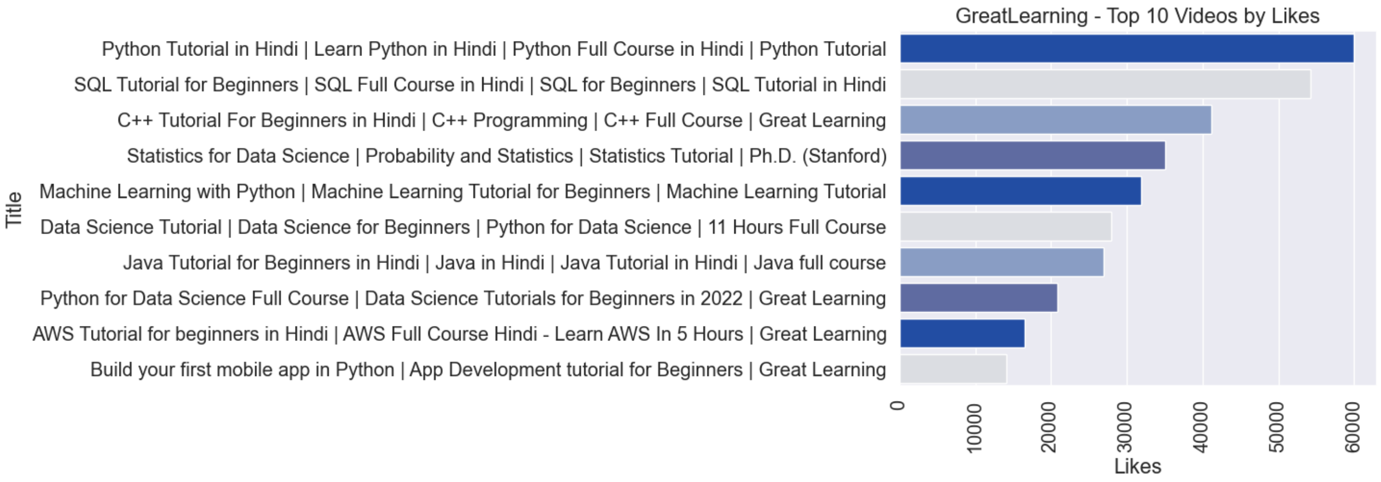
```
sns.barplot(x="Views", y="Title", data=GreatLearning_t10,palette=GreatLearning_palette)
plt.xticks(rotation=90) # adjust rotation angle as needed
plt.title("GreatLearning - Top 10 Videos by Views") # add the Title
plt.show()
```



In [ ]:

In [333…

```
sns.barplot(x="Likes", y="Title", data=GreatLearning_t10L,palette=GreatLearning_palette)
plt.xticks(rotation=90) # adjust rotation angle as needed
plt.title("GreatLearning - Top 10 Videos by Likes") # add the Title
plt.show()
```



In [ ]:

In [ ]:

```
```

Intellipaat

In [334…:

```
#Based on Views
```

In [335…:

```
Intellipaat_t10 =video_data_Intellipaat.sort_values(by="Views",ascending = False).head(1
```

In [336…:

```
Intellipaat_t10
```

Out[336]:

	Title	Published_date	Views	Likes	Dislikes	Comments
2603	Excel Tutorial | Microsoft Excel Tutorial | Ex...	2019-06-22	10211935	208518	0	3073
2291	How to Start Coding | Programming for Beginner...	2020-02-25	6831856	174437	0	8898
2129	Python Tutorial | Python Course | Intellipaat	2020-06-15	4854712	92388	0	1825
2551	Python Course | Python Tutorial for Beginners ...	2019-08-17	3877245	105417	0	4511
2554	Ethical Hacking Training | Ethical Hacking Tut...	2019-08-14	2802100	69094	0	1201
2415	Top 10 Highest Paying Jobs | Highest Paying IT...	2019-12-01	2386980	40449	0	2854
390	Intellipaat Best Data Science Certification Co...	2022-10-25	2006014	38	0	3
1400	IIT Roorkee EICT - Advanced Certification in C...	2021-08-11	1896532	120	0	10
1797	SQL Course | SQL Training | SQL Tutorial For B...	2021-01-12	1713801	31204	0	853
2385	Artificial Intelligence Tutorial | Artificial ...	2019-12-21	1656010	31531	0	768

In [337…:

```
#Based on Likes
```

In [338…:

```
Intellipaat_t10L =video_data_Intellipaat.sort_values(by="Likes",ascending = False).head(
```

In [339…:

```
Intellipaat_t10L
```

Out[339]:

	Title	Published_date	Views	Likes	Dislikes	Comments
2603	Excel Tutorial | Microsoft Excel Tutorial | Ex...	2019-06-22	10211935	208518	0	3073
2291	How to Start Coding | Programming for Beginner...	2020-02-25	6831856	174437	0	8898
2551	Python Course | Python Tutorial for Beginners ...	2019-08-17	3877245	105417	0	4511
2129	Python Tutorial | Python Course | Intellipaat	2020-06-15	4854712	92388	0	1825
2554	Ethical Hacking Training | Ethical Hacking Tut...	2019-08-14	2802100	69094	0	1201
2415	Top 10 Highest Paying Jobs | Highest Paying IT...	2019-12-01	2386980	40449	0	2854
1951	Ethical Hacking Course Online | Ethical Hackin...	2020-10-17	1342674	33450	0	505
2385	Artificial Intelligence Tutorial | Artificial ...	2019-12-21	1656010	31531	0	768
1797	SQL Course | SQL Training | SQL Tutorial For B...	2021-01-12	1713801	31204	0	853
1759	Web Development Tutorial For Beginners | Web D...	2021-02-08	1505518	29954	0	598

In [ ]:

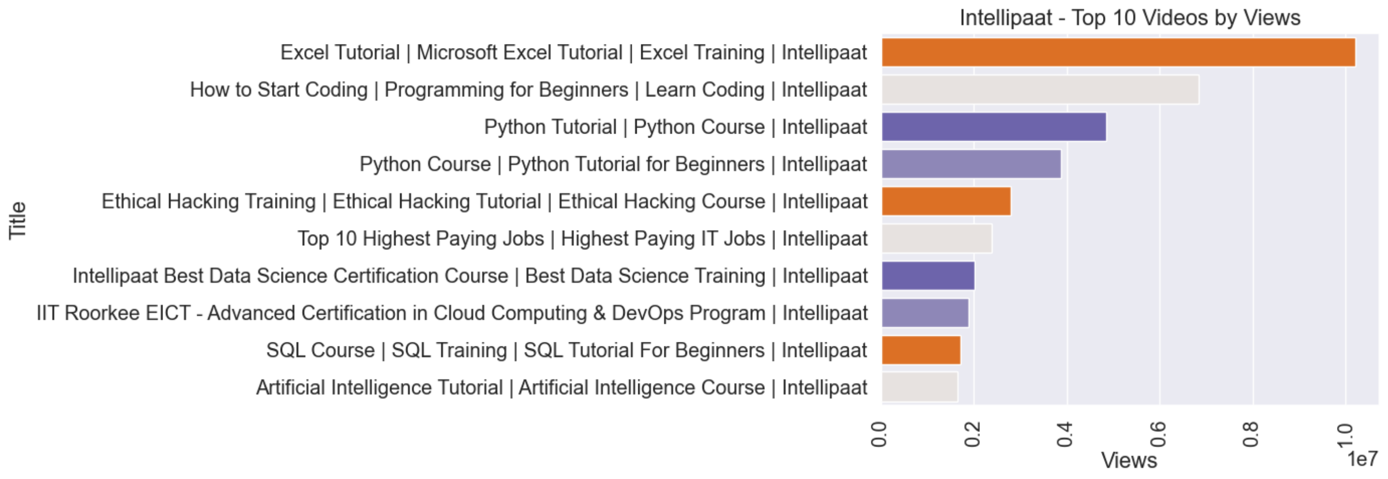
```
```

In [340…:

```
# To create a custom palette
Intellipaat_palette = sns.color_palette(["#FB6B06","#e8e2df","#6458B7","#887FBF"])
```

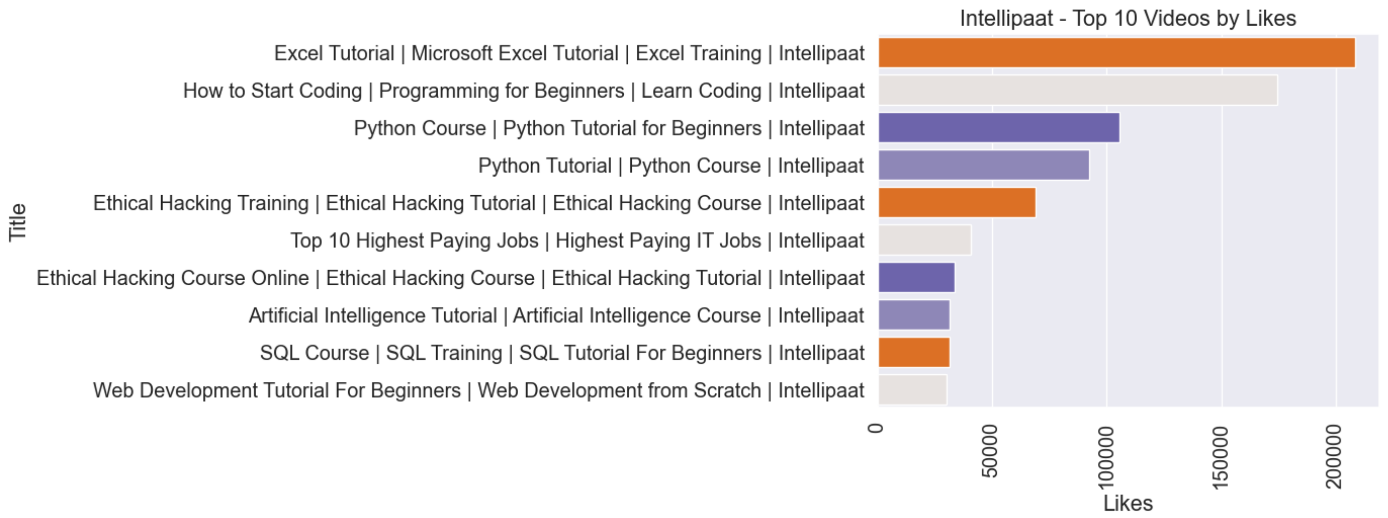
In [341…

```
sns.barplot(x="Views", y="Title", data=Intellipaat_t10,palette = Intellipaat_palette )
plt.xticks(rotation=90) # adjust rotation angle as needed
plt.title("Intellipaat - Top 10 Videos by Views") # add the Title
plt.show()
```



In [342…

```
sns.barplot(x="Likes", y="Title", data=Intellipaat_t10L,palette = Intellipaat_palette )
plt.xticks(rotation=90) # adjust rotation angle as needed
plt.title("Intellipaat - Top 10 Videos by Likes") # add the Title
plt.show()
```



In [ ]:

Simplilearn

In [343…

```
#Based on Views
```

In [344…

```
Simplilearn_t10 =video_data_Simplilearn.sort_values(by="Views",ascending = False).head(1
```

In [345…

```
Simplilearn_t10
```

Out[345]:

	Title	Published_date	Views	Likes	Dislikes	Comments
2459	Ambition #CannotBeLockedDown | Nikhil Got His...	2021-09-14	18214740	502	0	71
1911	#JobGuaranteed | Simplilearn Job Guarantee Pro...	2022-01-23	12498152	8091	0	47
1955	#JobGuaranteed | Simplilearn Job Guarantee Pro...	2022-01-11	11839624	8197	0	63
1609	Simplilearn Job Guarantee Programs - Your Job ...	2022-04-21	7619413	130	0	12
1551	Simplilearn Job Guarantee Programs - Your Job ...	2022-05-06	6563024	32	0	2

3025	How To Introduce Yourself In Interview | Self ...	2021-03-19	6436396	125870	0	593
4134	Machine Learning Basics | What Is Machine Lear...	2018-09-19	3746397	46790	0	2068
4086	Data Science In 5 Minutes | Data Science For B...	2018-12-04	3352999	49057	0	1085
3751	Digital Marketing Course Part - 1 🔥 | Digital M...	2020-03-16	3139973	75310	0	1578
3457	Digital Marketing In 5 Minutes | What Is Digit...	2020-09-23	2814308	66703	0	3370

In [346…

```
#Based on likes
```

In [347…

```
Simplilearn_t10L =video_data_Simplilearn.sort_values(by="Likes",ascending = False).head(
```

In [348…

```
Simplilearn_t10L
```

Out[348]:

	Title	Published_date	Views	Likes	Dislikes	Comments
3025	How To Introduce Yourself In Interview | Self ...	2021-03-19	6436396	125870	0	593
3751	Digital Marketing Course Part - 1 🔥 | Digital M...	2020-03-16	3139973	75310	0	1578
3457	Digital Marketing In 5 Minutes | What Is Digit...	2020-09-23	2814308	66703	0	3370
3860	SEO Tutorial For Beginners | SEO Full Course ...	2019-11-19	2143575	56724	0	1486
4086	Data Science In 5 Minutes | Data Science For B...	2018-12-04	3352999	49057	0	1085
4134	Machine Learning Basics | What Is Machine Lear...	2018-09-19	3746397	46790	0	2068
3775	Six Sigma In 9 Minutes | What Is Six Sigma? | ...	2020-02-25	2098256	42809	0	3640
3704	Lean Six Sigma In 8 Minutes | What Is Lean Six...	2020-05-06	1478983	38691	0	2405
3653	Cyber Security In 7 Minutes | What Is Cyber Se...	2020-06-10	1761180	36554	0	2468
1873	Advanced Excel Full Course 2022 | 🔥 Advanced Ex...	2022-02-04	2476119	35595	0	151

In [ ]:

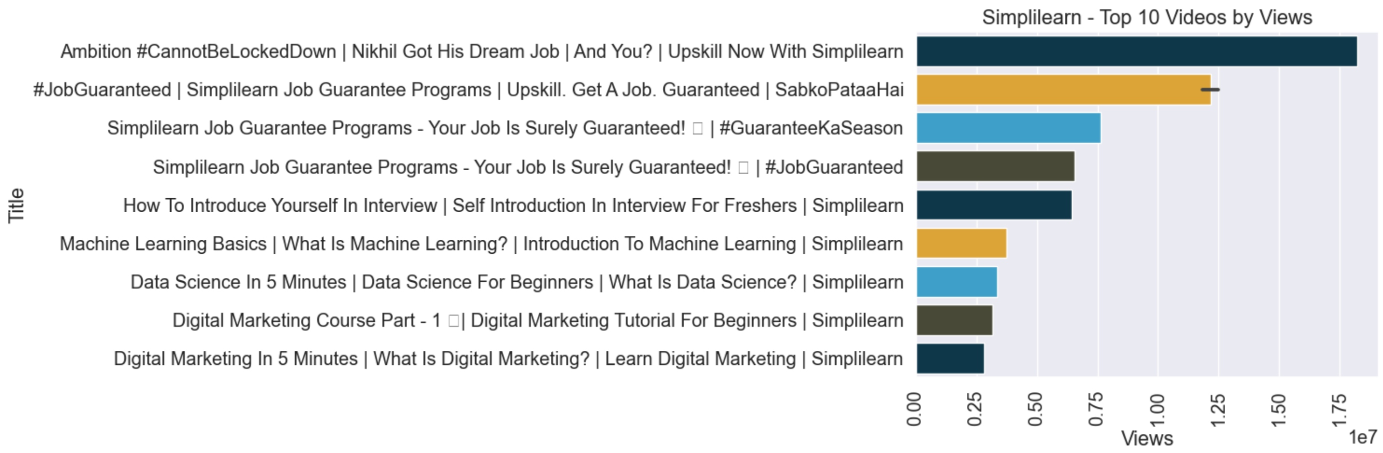
In [349…

```
# create a custom palette
Simplilearn_palette = sns.color_palette(["#043B53","#F8AD1B","#27A9E0","#4A4C34"])
```

In [350…

```
sns.barplot(x="Views", y="Title", data=Simplilearn_t10,palette=Simplilearn_palette)
plt.xticks(rotation=90) # adjust rotation angle as needed
plt.title("Simplilearn - Top 10 Videos by Views") # add the Title
plt.show()
```

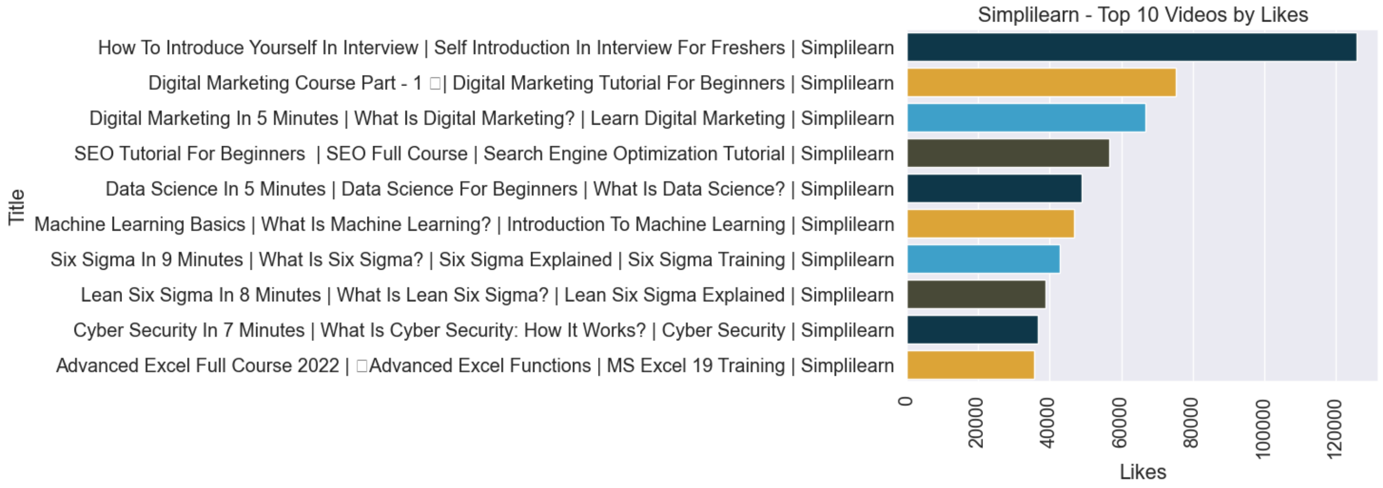
```
C:\Users\ltaru\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning:
Glyph 128526 (\N{SMILING FACE WITH SUNGLASSES}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
C:\Users\ltaru\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning:
Glyph 128293 (\N{FIRE}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



In [351…

```
sns.barplot(x="Likes", y="Title", data=Simplilearn_t10L,palette=Simplilearn_palette)
plt.xticks(rotation=90) # adjust rotation angle as needed
plt.title("Simplilearn - Top 10 Videos by Likes") # add the Title
plt.show()
```

```
C:\Users\ltaru\anaconda3\lib\site-packages\IPython\core\pylabtools.py:151: UserWarning:
Glyph 128293 (\N{FIRE}) missing from current font.
  fig.canvas.print_figure(bytes_io, **kw)
```



In [ ]:

Conclusions

In [ ]:

When we observe the graphs of GreatLearning & Simplilearn, it appears that there is a difference between the videos with the most views and those with the most likes. Specifically, videos with the most views tend to be advertisements with hashtags, while videos with the most likes tend to be tutorials related to various topics in technology.

From this observation, we can draw following conclusions:

The audience's viewing behavior differs from their engagement behavior: The difference in the videos with the most views versus the most likes suggests that the audience's viewing behavior may differ from their engagement behavior. In other words, viewers may be more likely to watch advertisements, but they engage more with tutorial videos.

Hashtags may be effective in increasing views: The fact that videos with hashtags tend to have more views suggests that the use of hashtags may be effective in increasing the visibility of videos and attracting viewers.

Tutorials may be more valuable to the audience: The fact that tutorial videos tend to have more likes suggests that the audience may value educational content related to various topics in technology. This insight may be useful for the edtech company in deciding which types of videos to produce in the future.

In 2023, GreatLearning uploaded 10 videos and had a average of 2703.2 views.

Intellipaat uploaded 231 videos and had a average of 4050.982832618026 views.

Simplilearn uploaded 529 videos and had a average of 8947.44990548204 views.

The total engagement rate for Intellipaat is 2.39%,
whereas 2.25% for GreatLearning and 2.29% for Simplilearn.
Even though there isn't musch difference, Intellipaat has the highest engagement rate.

In [ ]: