YouTube Analysis Project

The YouTube Channel Analysis project aims to gather data from YouTube using web scraping techniques, analyze the data using Pandas, and visualize insights with Seaborn. The focus is on extracting key metrics such as video views, likes, comments, and subscriber growth to understand the channel's performance and trends.

Key Components :

1. Data Collection using YouTube API(Web Scraping) :
   Objective: Retrieve data about videos, comments, and channel statistics directly from the YouTube API.
   Tools:'google-api-python-client' for interacting with the YouTube API.API key for authentication and data access.

2. Data Processing with Pandas :
   Objective: Clean, organize, and manipulate the scraped data for analysis.
   Tools:Pandas for data manipulation and analysis.

3. Data Analysis :
   Objective: Perform exploratory data analysis (EDA) to derive meaningful insights.
   Methods:Correlation analysis to identify relationships between different metrics.

4. Data Visualization with Seaborn:
   Objective: Visualize the data to uncover patterns and trends.
   Tools:Seaborn for creating visualizations such as line plots, bar charts, histograms, and scatter plots.

Steps in the Project

1. Set Up Environment :
    (a)Install necessary libraries (google-api-python-client, pandas, seaborn, matplotlib, etc.).
    (b)Obtain an API key from the Google Developer Console.

2. Data Collection using YouTube API :
    (a)Authenticate and connect to the YouTube API.
    (b)Write a script to fetch video data, including video IDs, titles, publication dates, view counts, likes,
       dislikes and comments.
    (c)Store the fetched data in a structured format, such as a CSV file or a Pandas DataFrame.

3. Data Cleaning :
    (a)Handle missing values, duplicates, and incorrect data types.
    (b)Normalize data (e.g., convert view counts to integers, dates to datetime objects)

Importing python libraries

In [9]:
```
from googleapiclient.discovery import build
import pandas as pd
import seaborn as sns
```

Including the youtube channels

In [12]:
```
api_key='AIzaSyBKE27ntrsmtQZ9hKJcRd6vSeXIF1ElDrs'
channel_ids=['UCXFxwj7DwumpUu5RNWlznTw','UCQYMhOMi_Cdj1CEAU-fv80A','UC7Q7pl0z0
youtube = build('youtube', 'v3',developerKey=api_key)
```

Extracting the details of the channel

In [13]:
```
def get_channel_stats(youtube, channel_ids):
    all_data = []
    request = youtube.channels().list(
                  part = 'snippet,contentDetails,statistics',
                  id = ','.join(channel_ids))
    response = request.execute()
    for i in range(len(response['items'])):
        data = dict(channel_name = response['items'][i]['snippet']['title'],
                    Subscribers = response['items'][i]['statistics']['subscrib
                    Views = response['items'][i]['statistics']['viewCount'],
                    Total_videos = response['items'][i]['statistics']['videoCo
                    Playlist_id = response['items'][i]['contentDetails']['rela
        all_data.append(data)

    return all_data
```

In [14]:
```
get_channel_stats(youtube, channel_ids)
```

Out[14]:
```
[{'channel_name': 'Aashish Solanki',
  'Subscribers': '900000',
  'Views': '253697233',
  'Total_videos': '162',
  'Playlist_id': 'UUaUr7y4F9lWGnZ0cbUZyzYA'},
 {'channel_name': 'SHEF',
  'Subscribers': '1380000',
  'Views': '634426483',
  'Total_videos': '516',
  'Playlist_id': 'UUu7Ld2o0QnLPm4j0gCn7Agw'},
 {'channel_name': 'Kreatryx GATE - EE, ECE & IN by Unacademy',
  'Subscribers': '331000',
  'Views': '88510665',
  'Total_videos': '12092',
  'Playlist_id': 'UUXFxwj7DwumpUu5RNWlznTw'},
 {'channel_name': 'MortaL',
  'Subscribers': '7010000',
  'Views': '1343919807',
  'Total_videos': '2140',
  'Playlist_id': 'UU7Q7pl0z0MrdayvmAnchlJQ'},
 {'channel_name': 'Neso Academy',
  'Subscribers': '2580000',
  'Views': '584388423',
  'Total_videos': '2688',
  'Playlist_id': 'UUQYMhOMi_Cdj1CEAU-fv80A'},
 {'channel_name': 'CodeWithHarry',
  'Subscribers': '6340000',
  'Views': '796776169',
  'Total_videos': '2376',
  'Playlist_id': 'UUeVMnSShP_Iviwkknt83cww'}]
```

In [15]:
```
channel_statistics = get_channel_stats(youtube, channel_ids)
channel_data = pd.DataFrame(channel_statistics)
channel_data
```

Out[15]:

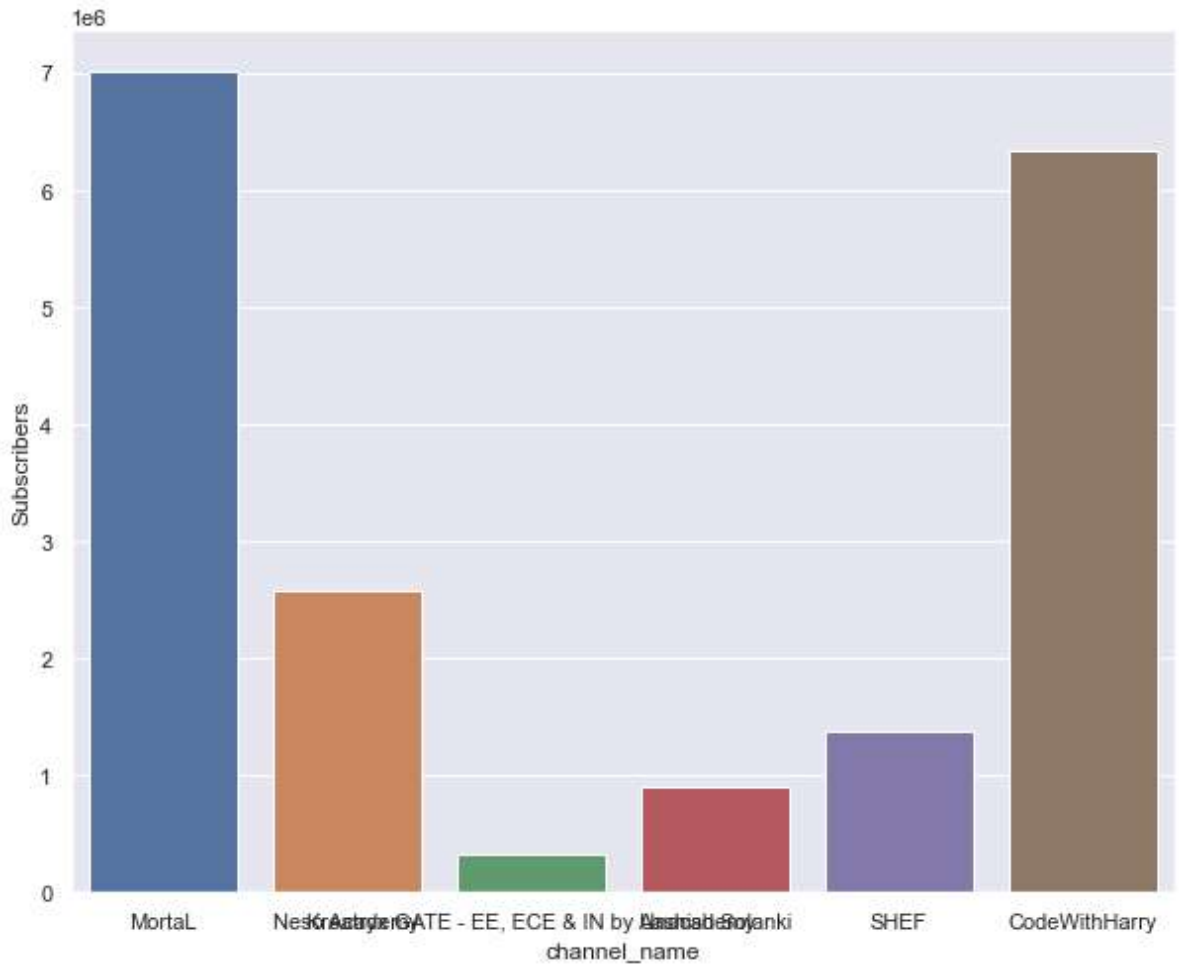
	channel_name	Subscribers	Views	Total_videos	Playlist_id
0	MortaL	7010000	1343919807	2140	UU7Q7pI0z0MrdayvmAnchIJQ
1	Neso Academy	2580000	584388423	2688	UUQYMhOMi_Cdj1CEAU-fv80A
2	Kreatryx GATE - EE, ECE & IN by Unacademy	331000	88510665	12092	UUXFxwj7DwumpUu5RNWIznTw
3	Aashish Solanki	900000	253697233	162	UUaUr7y4F9lWGnZ0cbUZyzYA
4	SHEF	1380000	634426483	516	UUu7Ld2o0QnLPm4j0gCn7Agw
5	CodeWithHarry	6340000	796776169	2376	UUeVMnSShP_Iviwkknt83cww

In [16]:
```
channel_data['Subscribers'] = pd.to_numeric(channel_data['Subscribers'])
channel_data['Views'] = pd.to_numeric(channel_data['Views'])
channel_data['Total_videos'] = pd.to_numeric(channel_data['Total_videos'])
channel_data.dtypes
```

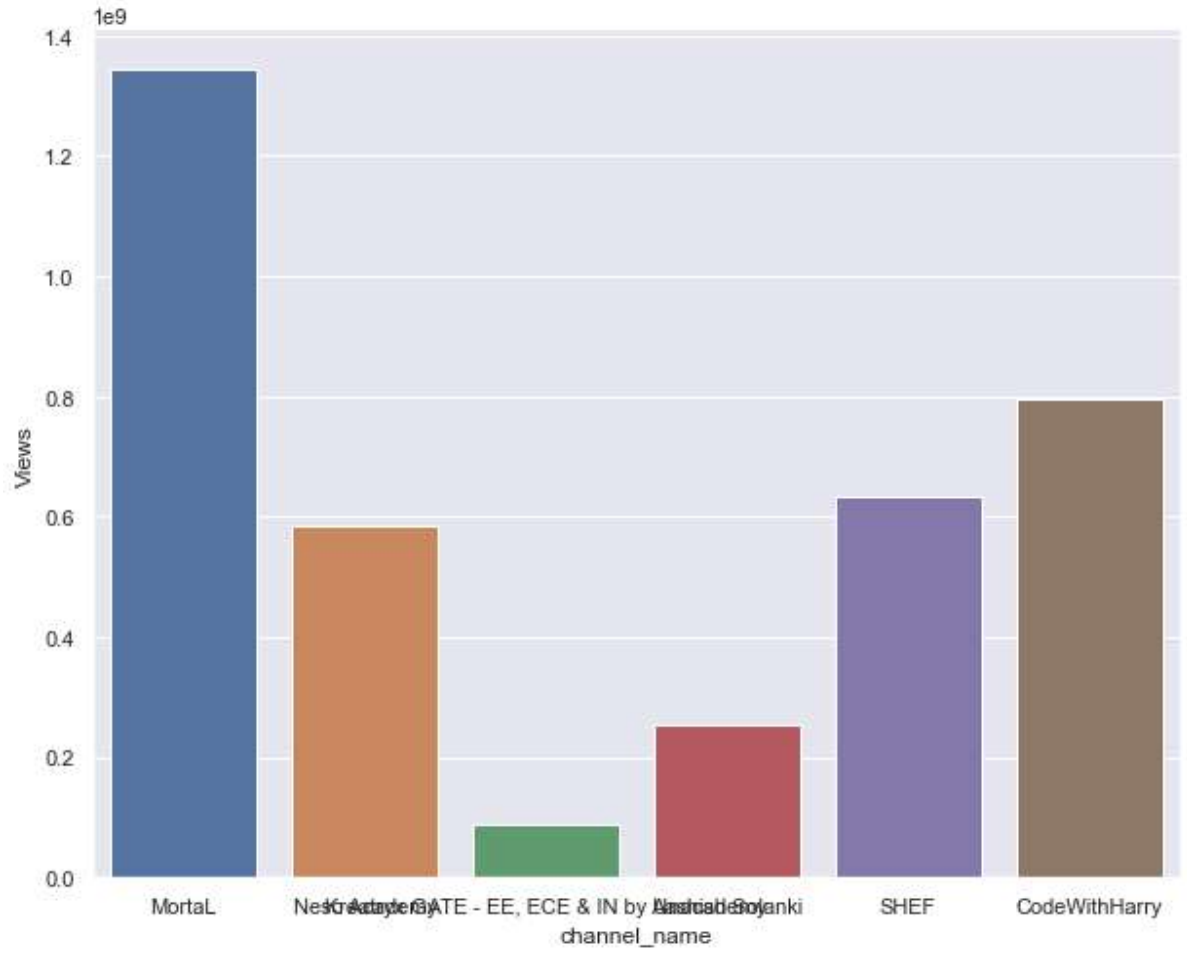
Out[16]:
```
channel_name    object
Subscribers      int64
Views            int64
Total_videos     int64
Playlist_id     object
dtype: object
```

Visualizing the DataFrame

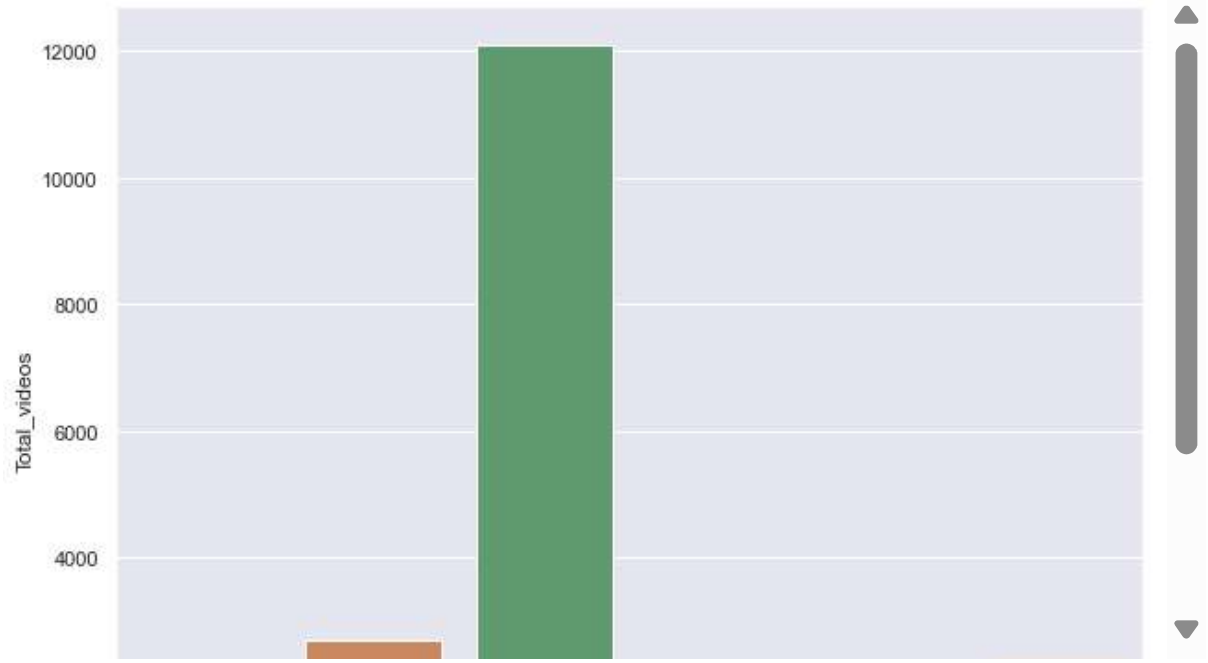
In [17]:
```
sns.set(rc = {'figure.figsize':(10,8)})
ax = sns.barplot(x='channel_name', y= 'Subscribers', data=channel_data)
```



In [18]:
```
ax = sns.barplot(x='channel_name', y= 'Views', data=channel_data)
```



In [19]:
```
ax = sns.barplot(x='channel_name', y= 'Total_videos', data=channel_data)
```



In [21]:
```
playlist_id = channel_data.loc[channel_data['channel_name']=='MortaL', 'Playli
playlist_id
```

Out[21]:
```
'UU7Q7pl0z0MrdayvmAnchlJQ'
```

Extracting video ids and details

In [22]:
```
def get_videos_ids(youtube, playlist_id):

    request= youtube.playlistItems().list(
                part='contentDetails',
                playlistId=playlist_id,
                maxResults = 50)
    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']['video

            next_page_token = response.get('nextPageToken')

    return video_ids
```

In [23]:
```
videos_ids = get_videos_ids(youtube, playlist_id)
videos_ids
```

Out[23]:
```
['BTCBkPQH2Js',
 'JIEHGuFLqIg',
 'n-wqHjLCRpI',
 'h0g9a2_aP68',
 '1drzswHsEhw',
 'wbS0iLVHLm8',
 'f_l7jjuxJTI',
 'TT_4eNUeW_I',
 'usofb18JpK8',
 'LjMOwkGcQyo',
 'JbNToHAwvGE',
 'EbKMf-yabwY',
 's7e0T4pLjz4',
 't_2rVyiUc9g',
 'GlQDypx-KJI',
 'dBm_EAASy2Y',
 'tdl14a8SGJc',
 'VTYM_IKIHuk',
 'xf3rEByotII',
```

Function to get the video details

In [45]:
```
def get_video_details(youtube, videos_ids):
    all_videos_stats = []

    for i in range(0, len(videos_ids), 50):
        request = youtube.videos().list(
                  part='snippet,statistics',
                  id=','.join(videos_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
                               Comments = video['statistics'].get('commentCount
            all_videos_stats.append(video_stats)

    return all_videos_stats
```

In [47]:
```
video_details = get_video_details(youtube, videos_ids)
video_details
```

Out[47]:
```
[{'Title': '7 WWCD | SUNA HAI NAYA UPDATE AAYA HAI !!',
  'Published_date': '2024-07-18T20:40:25Z',
  'Views': '370561',
  'Likes': '22360',
  'Dislikes': 0,
  'Comments': '41'},
 {'Title': 'Meet the Sanskari PUBG Squad 😂',
  'Published_date': '2024-07-18T07:19:21Z',
  'Views': '104886',
  'Likes': '10328',
  'Dislikes': 0,
  'Comments': '125'},
 {'Title': 'BHUL TOH NI GAYE HUME !!',
  'Published_date': '2024-07-17T21:16:20Z',
  'Views': '269308',
  'Likes': '15939',
  'Dislikes': 0,
  'Comments': '22'},
 {'Title': 'GUSSA THUK DO ( MUJPE NAI ) !!',
```

In [49]:
```
video_data = pd.DataFrame(video_details)
video_data
```

Out[49]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	7 WWCD | SUNA HAI NAYA UPDATE AAYA HAI !!	2024-07-18T20:40:25Z	370561	22360	0	41
1	Meet the Sanskari PUBG Squad 😂	2024-07-18T07:19:21Z	104886	10328	0	125
2	BHUL TOH NI GAYE HUME !!	2024-07-17T21:16:20Z	269308	15939	0	22
3	GUSSA THUK DO (MUJPE NAI) !!	2024-07-16T20:54:34Z	210537	12972	0	66
4	TUNE GYM KA FEES BHARA THANA ?? 😳😳	2024-07-15T19:59:11Z	148849	9841	0	17
...	...	...	...	...	...	...
2085	MorTal vs MiLiTian, 1 vs 1 best of three serie...	2016-10-11T13:09:43Z	1097873	42119	0	1245
2086	One of the best comeback victory ever? DOODLE ...	2016-10-03T16:22:08Z	14231	501	0	40
2087	~.MorTaL vs NA'VI vs DQ~Doflamingo. Mini milit...	2016-09-22T05:46:04Z	17983	656	0	60
2088	~.MorTaL vs ZÛKØ^{RPG} (Bilal khan)	2016-09-20T11:50:41Z	61524	2334	0	152
2089	~.MorTaL vs Noobie vs Your Daddy (Not Noobies...	2016-09-15T15:35:38Z	228380	12894	0	1664

2090 rows × 6 columns

In [50]:
```
video_data['Published_date'] = pd.to_datetime(video_data['Published_date']).dt
video_data['Views'] = pd.to_numeric(video_data['Views'])
video_data['Likes'] = pd.to_numeric(video_data['Likes'])
video_data['Dislikes'] = pd.to_numeric(video_data['Dislikes'])
video_data['Comments'] = pd.to_numeric(video_data['Comments'])
video_data
```

Out[50]:

	Title	Published_date	Views	Likes	Dislikes	Comments
0	7 WWCD | SUNA HAI NAYA UPDATE AAYA HAI !!	2024-07-18	370561	22360	0	41
1	Meet the Sanskari PUBG Squad 😂	2024-07-18	104886	10328	0	125
2	BHUL TOH NI GAYE HUME !!	2024-07-17	269308	15939	0	22
3	GUSSA THUK DO (MUJPE NAI) !!	2024-07-16	210537	12972	0	66
4	TUNE GYM KA FEES BHARA THANA ?? 😳😳	2024-07-15	148849	9841	0	17
...	...	...	...	...	...	...
2085	MorTal vs MiLiTian, 1 vs 1 best of three serie...	2016-10-11	1097873	42119	0	1245
2086	One of the best comeback victory ever? DOODLE ...	2016-10-03	14231	501	0	40
2087	~.MorTaL vs NA'VI vs DQ~Doflamingo. Mini milit...	2016-09-22	17983	656	0	60
2088	~.MorTaL vs ZÛKØ^{RPG} (Bilal khan)	2016-09-20	61524	2334	0	152
2089	~.MorTaL vs Noobie vs Your Daddy (Not Noobies...	2016-09-15	228380	12894	0	1664

2090 rows × 6 columns

Performing some sorting in the DataFrame

In [52]:
```
top10_videos = video_data.sort_values(by = 'Views', ascending = False).head(10
top10_videos
```

Out[52]:

	Title	Published_date	Views	Likes	Dislikes	Comments
1995	Every PUBG player will watch this Ending.	2018-10-21	20538236	1198649	0	28066
1726	This is why I left M24 behind for DP-28 | Pubg...	2019-07-08	14885605	442530	0	8550
1982	RECORD 51 Kills | BEAST SQUAD | PUBG MOBILE	2018-11-15	13292220	174015	0	6633
1488	Pubg Mobile a Roller Coaster Journey | Mortala...	2020-05-17	8801417	542560	0	8025
1369	Lets Calm down and talk| S8uL is in this toget...	2020-09-02	6648241	478525	0	3756
1144	Chota 8bit Thug says hello to Mortalarmy #shorts	2021-03-08	5975504	445837	0	2980
2005	CONQUEROR SOLO VS SQUAD | How difficult is Con...	2018-09-28	5896518	255512	0	16891
2020	6 ADVANCE TIPS TO BECOME PRO |PUBG MOBILE TIPS...	2018-08-16	5751352	131469	0	3983
1994	This match was played after I reached RANK #5 ...	2018-10-23	5709243	233767	0	8012
1555	CARRY - I ACCEPT YOUR CHALLENGE	2020-02-24	5338286	463848	0	12420

If head function is not used then it will show all the videos

In [54]:
```
video_data['Month'] = pd.to_datetime(video_data['Published_date']).dt.strftime
video_data
```

Out[54]:

	Title	Published_date	Views	Likes	Dislikes	Comments	Month
0	7 WWCD | SUNA HAI NAYA UPDATE AAYA HAI !!	2024-07-18	370561	22360	0	41	Jul
1	Meet the Sanskari PUBG Squad 😂	2024-07-18	104886	10328	0	125	Jul
2	BHUL TOH NI GAYE HUME !!	2024-07-17	269308	15939	0	22	Jul
3	GUSSA THUK DO (MUJPE NAI) !!	2024-07-16	210537	12972	0	66	Jul
4	TUNE GYM KA FEES BHARA THANA ?? 😳😳	2024-07-15	148849	9841	0	17	Jul
...	...	...	...	...	...	...	...
2085	MorTal vs MiLiTian, 1 vs 1 best of three serie...	2016-10-11	1097873	42119	0	1245	Oct
2086	One of the best comeback victory ever? DOODLE ...	2016-10-03	14231	501	0	40	Oct
2087	~.MorTaL vs NA'VI vs DQ~Doflamingo. Mini milit...	2016-09-22	17983	656	0	60	Sep
2088	~.MorTaL vs ZÛKØ^{RPG} (Bilal khan)	2016-09-20	61524	2334	0	152	Sep
2089	~.MorTaL vs Noobie vs Your Daddy (Not Noobies...	2016-09-15	228380	12894	0	1664	Sep

2090 rows × 7 columns

Include a separate column 'Month' in the DataFrame

In [55]:
```
videos_per_month = video_data.groupby('Month', as_index=False).size()
videos_per_month
```

Out[55]:

	Month	size
0	Apr	158
1	Aug	159
2	Dec	206
3	Feb	172
4	Jan	206
5	Jul	181
6	Jun	196
7	Mar	161
8	May	155
9	Nov	136
10	Oct	201
11	Sep	159

In [56]:
```
sort_order = ['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov
```

In [57]:
```
videos_per_month.index = pd.CategoricalIndex(videos_per_month['Month'], catego
```

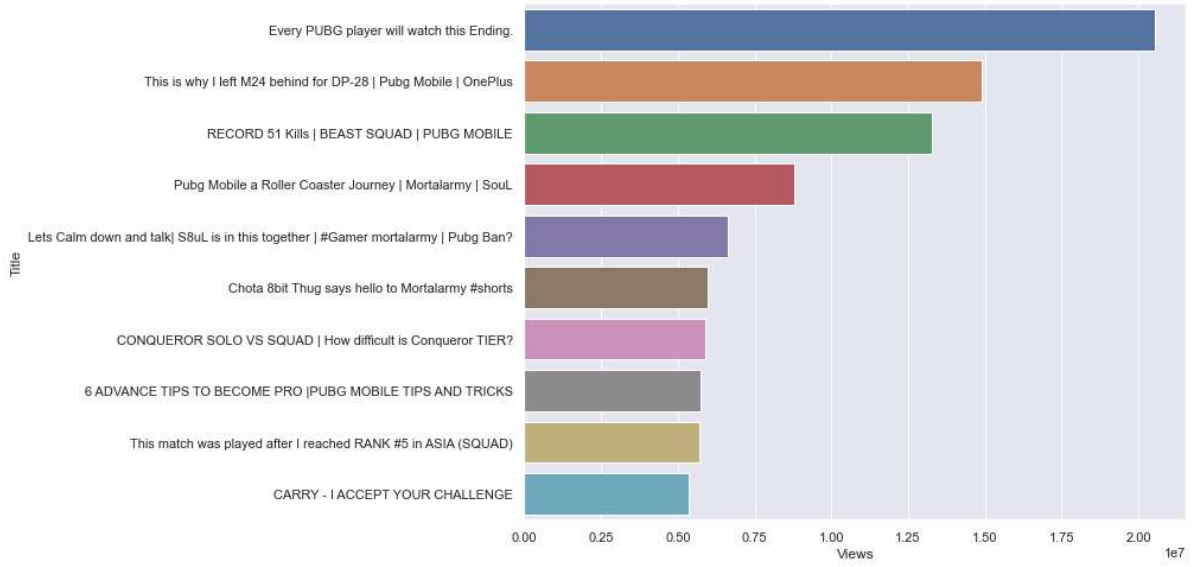
In [58]:
```
videos_per_month = videos_per_month.sort_index()
videos_per_month
```

Out[58]:

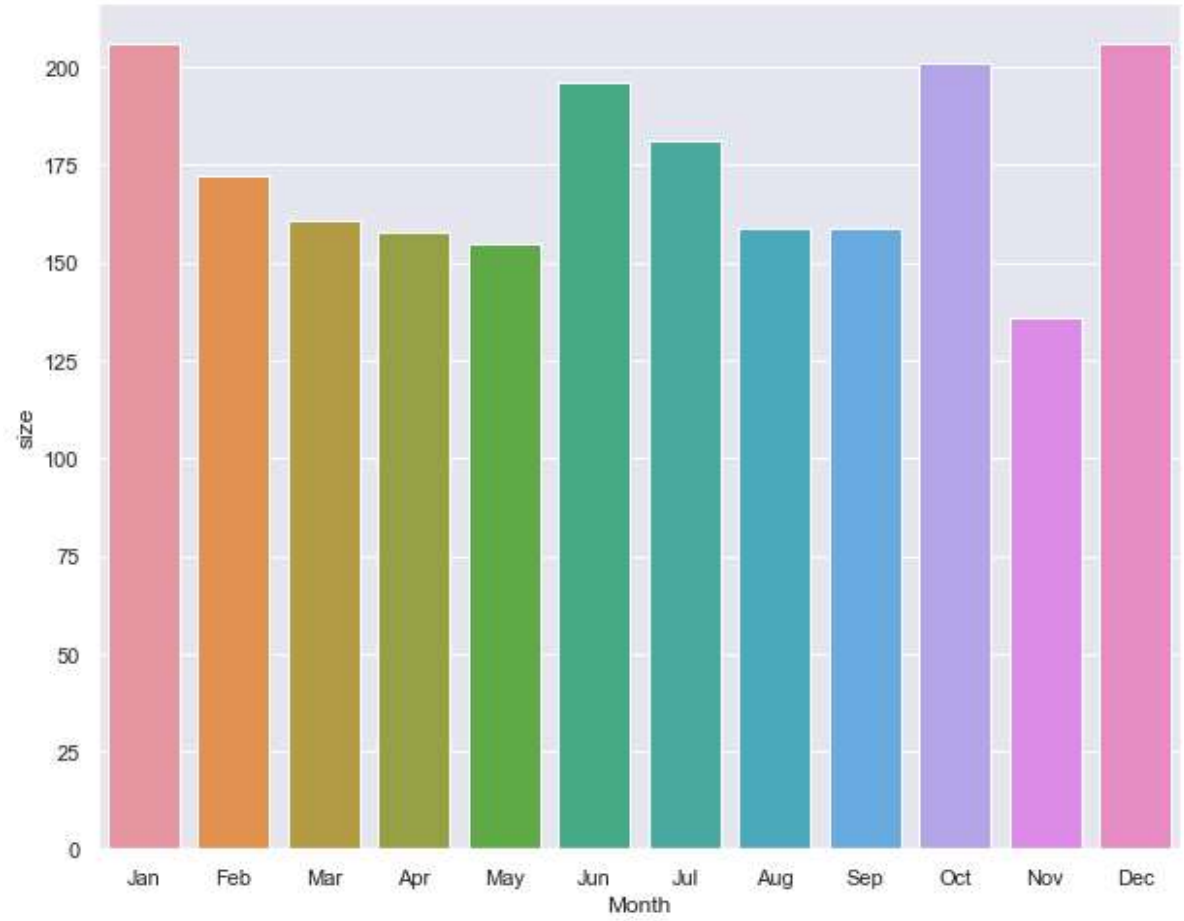
	Month	size
Month		
Jan	Jan	206
Feb	Feb	172
Mar	Mar	161
Apr	Apr	158
May	May	155
Jun	Jun	196
Jul	Jul	181
Aug	Aug	159
Sep	Sep	159
Oct	Oct	201
Nov	Nov	136
Dec	Dec	206

Visualization of video datas of a channel

In [59]:
```
ax1 = sns.barplot(x='Views', y='Title', data=top10_videos)
```



In [60]:
```
ax2 = sns.barplot(x='Month', y='size', data=videos_per_month)
```



Storing in CSV files

In [61]:
```
channel_data.to_csv('Channel_Details.csv')
```

In [62]:
```
video_data.to_csv('Video_Details(MortaL).csv')
```

The YouTube Channel Analysis project demonstrates a comprehensive approach to understanding and evaluating the performance of a YouTube channel using Python, Pandas, Seaborn, and the YouTube API. By leveraging the YouTube API for data collection, the project ensures accurate and up-to-date information about video statistics and channel metrics. Later the obtained datas are stored in two CSV files.

Key Achievements

```
1. Effective Data Collection:
        Utilized the YouTube API to gather extensive data on video pe
rformance, including views, likes, comments, and
        publication dates, ensuring reliability and completeness.

2. Robust Data Processing:
        Employed Pandas for efficient data cleaning, organization, an
d manipulation, transforming raw data into a
        structured format suitable for analysis.

3. In-Depth Data Analysis:
        Conducted exploratory data analysis (EDA) to uncover insights
into video performance and channel growth. Used
        descriptive statistics and time-series analysis to identify t
rends and patterns.

4. Insightful Data Visualization:
        Applied Seaborn to create clear and informative visualization
s, making it easier to interpret complex data and
        identify key trends. Visualized metrics such as view counts,
likes, and subscriber growth over time to provide a
        comprehensive view of the channel's performance.

5. Comprehensive Reporting:
        Summarized findings in a structured report, highlighting sign
ificant insights and supporting them with visual
        evidence. Provided actionable insights that can inform conten
t strategy and optimization efforts.
```

Project Impact

This project equips content creators, marketers, and analysts with valuable insights into their YouTube channel's performance. By understanding viewer engagement, identifying popular content, and tracking growth trends, stakeholders can make data-driven decisions to enhance their content strategy, improve audience retention, and ultimately grow their channel.