In [1]:

```
from googleapiclient.discovery import build
import numpy as np
import pandas as pd
import pprint
from dateutil import parser
import isodate
import statistics

import warnings
warnings.filterwarnings("ignore")

import matplotlib.pyplot as plt
import matplotlib.ticker as ticker
import seaborn as sns
sns.set(style="darkgrid", color_codes=True)
```

In [2]:

```
pd.options.display.float_format = '{:.4f}'.format  #to not get scientific formatting in pan
```

In [3]:

```
api_key = 'AIzaSyCvPxELANzObsFIo0Wm_8cVip3MjqBd2rI'   #to be hidden while sending to someon
```

In [4]:

```
channel_ids = ['UCVOTBwF0vnSxMRIbfSE_K_g'] #can include more channels here
```

In [5]:

```
api_service_name = "youtube"
api_version = "v3"
#     client_secrets_file = "YOUR_CLIENT_SECRET_FILE.json"

    # Get credentials and create an API client
#     flow = google_auth_oauthlib.flow.InstalledAppFlow.from_client_secrets_file(
#         client_secrets_file, scopes)
#     credentials = flow.run_console()
youtube = build(
#         api_service_name, api_version, credentials=credentials)
        api_service_name, api_version, developerKey=api_key)

request = youtube.channels().list(
        part="snippet,contentDetails,statistics",
        id=','.join(channel_ids)
    )
response = request.execute()

pprint.pprint(response)
```

```
{'etag': 'AwgKvB8CdzqN5abtnwAu-zc7nzc',
 'items': [{'contentDetails': {'relatedPlaylists': {'likes': '',
                                                    'uploads': 'UUVOTBwF0vnS
xMRIbfSE_K_g'}},
            'etag': 'ZT8D6F4zon48ZC08cq4mIGz_Tew',
            'id': 'UCVOTBwF0vnSxMRIbfSE_K_g',
            'kind': 'youtube#channel',
            'snippet': {'country': 'IN',
                        'customUrl': '@labourlawadvisor',
                        'description': 'Labour Laws | Personal Finance | '
                                       'Taxation | Investment for employees
\n'
                                       '\n'
                                       "Have you checked out LLA's Jagruk "
                                       'Journal 2023? \n'
                                       'Visit: jagrukjournal.lla.in \n'
                                       '\n'
                                       'Say hello - rishabh@witbox.in\n'
                                       'Support - support@lla.in\n'
                                       'Be part of Team LLA - hr@lla.in\n',
                        'localized': {'description': 'Labour Laws | Personal
'
                                                     'Finance | Taxation | '
                                                     'Investment for '
                                                     'employees \n'
                                                     '\n'
                                                     'Have you checked out '
                                                     "LLA's Jagruk Journal "
                                                     '2023? \n'
                                                     'Visit: '
                                                     'jagrukjournal.lla.in
\n'
                                                     '\n'
                                                     'Say hello - '
                                                     'rishabh@witbox.in\n'
                                                     'Support - '
                                                     'support@lla.in\n'
                                                     'Be part of Team LLA -
'
```

```
                                                'hr@lla.in\n',
                                    'title': 'Labour Law Advisor'},
                        'publishedAt': '2017-08-12T09:53:22Z',
                        'thumbnails': {'default': {'height': 88,
                                                   'url': 'https://yt3.ggph
t.com/CVvE7vApeq2jgHhty_LsDBVJPnp-msvs7r3spAZo_14T_nBqd1CWTjhUdjg1TTAztO7MOx
u2=s88-c-k-c0x00ffffff-no-rj',
                                                   'width': 88},
                                       'high': {'height': 800,
                                                'url': 'https://yt3.ggpht.co
m/CVvE7vApeq2jgHhty_LsDBVJPnp-msvs7r3spAZo_14T_nBqd1CWTjhUdjg1TTAztO7MOxu2=s
800-c-k-c0x00ffffff-no-rj',
                                                'width': 800},
                                       'medium': {'height': 240,
                                                  'url': 'https://yt3.ggpht.
com/CVvE7vApeq2jgHhty_LsDBVJPnp-msvs7r3spAZo_14T_nBqd1CWTjhUdjg1TTAztO7MOxu2
=s240-c-k-c0x00ffffff-no-rj',
                                                  'width': 240}},
                        'title': 'Labour Law Advisor'},
            'statistics': {'hiddenSubscriberCount': False,
                           'subscriberCount': '3930000',
                           'videoCount': '914',
                           'viewCount': '845002134'}}],
 'kind': 'youtube#channelListResponse',
 'pageInfo': {'resultsPerPage': 5, 'totalResults': 1}}
```

In [6]:

```
def get_channel_stats(youtube, channel_ids):

    all_data = []

    request = youtube.channels().list(
        part="snippet,contentDetails,statistics",
        id=','.join(channel_ids)
    )
    response = request.execute()

    # loop through items
    for item in response['items']:
        data = {'channelName': item['snippet']['title'],
                'subscribers': item['statistics']['subscriberCount'],
                'views': item['statistics']['viewCount'],
                'totalVideos': item['statistics']['videoCount'],
                'playlistId': item['contentDetails']['relatedPlaylists']['uploads']
        }

        all_data.append(data)

    return(pd.DataFrame(all_data))
```

In [7]:

```
channel_stats = get_channel_stats(youtube, channel_ids)
print(channel_stats)
```

```
          channelName subscribers      views totalVideos  \
0  Labour Law Advisor     3930000  845002134         914

                 playlistId
0  UUVOTBwF0vnSxMRIbfSE_K_g
```

In [8]:

```
views_per_video = int(channel_stats['views'][0])/int(channel_stats['totalVideos'][0])
print(round(views_per_video,2))
subs_per_video = int(channel_stats['subscribers'][0])/int(channel_stats['totalVideos'][0])
print(round(subs_per_video,2))
views_per_sub = int(channel_stats['views'][0])/int(channel_stats['subscribers'][0])
print(round(views_per_sub,2))
```

```
924509.99
4299.78
215.01
```

In [9]:

```
deep_dive = pd.DataFrame({'Views/Sub' : round(views_per_sub,2) , 'Subs/Video' : round(subs_
deep_dive
```

Out[9]:

	Views/Sub	Subs/Video	Views/Video
0	215.0100	4299.7800	924509.9900

In [10]:

```
playlist_id = list(channel_stats['playlistId'])[0]

def get_video_ids(youtube, playlist_id):

    video_ids = []

    request = youtube.playlistItems().list(
        part="snippet,contentDetails",
        playlistId=playlist_id,
        maxResults = int(list(channel_stats['totalVideos'])[0])
    )
    response = request.execute()

    for item in response['items']:
        video_ids.append(item['contentDetails']['videoId'])

    next_page_token = response.get('nextPageToken')
    while next_page_token is not None:
        request = youtube.playlistItems().list(
                    part='contentDetails',
                    playlistId = playlist_id,
                    maxResults = int(list(channel_stats['totalVideos'])[0]),
                    pageToken = next_page_token)
        response = request.execute()

        for item in response['items']:
            video_ids.append(item['contentDetails']['videoId'])

        next_page_token = response.get('nextPageToken')

    return video_ids
```

In [11]:

```
video_ids = get_video_ids(youtube, list(channel_stats['playlistId'])[0])
```

In [12]:

```
len(video_ids)
```

Out[12]:

```
913
```

In [13]:

```
def get_video_details(youtube, video_ids):

    all_video_info = []

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


        for video in response['items']:
                    stats_to_keep = {'snippet': ['channelTitle', 'title', 'description', 't
                                     'statistics': ['viewCount', 'likeCount', 'commentCount
                                     'contentDetails': ['duration', 'definition', 'caption'
                                    }
                    video_info = {}
                    video_info['video_id'] = video['id']

                    for k in stats_to_keep.keys():
                        for v in stats_to_keep[k]:
                            try:
                                video_info[v] = video[k][v]
                            except:
                                video_info[v] = None

                    all_video_info.append(video_info)
    return pd.DataFrame(all_video_info)
```

In [14]:

```
video_df = get_video_details(youtube, video_ids)
video_df.head()
```

Out[14]:

	video_id	channelTitle	title	description	tags	publishedAt	viewCount	likeCou
0	70JHBff89Ko	Labour Law Advisor	Insurance ka Loop Hole! #LLAShorts 501	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-15T13:30:03Z	77553	132
1	6gi3BdX56ZY	Labour Law Advisor	EMI maaf #LLAShorts 500	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-14T13:30:02Z	114578	136
2	Db5wC-xNH80	Labour Law	Art of Emotional Marketing	Open Zerodha A/c, start	[lla, lla shorts, youtube	2023-01-	686935	640

In [15]:

```
video_df.shape
```

Out[15]:

```
(913, 13)
```

In [16]:

```
if video_df.shape[0] == len(video_ids):
    print("Data properly fetched")
else:
    print("Data not properly fetched.")
```

```
Data properly fetched
```

Data pre-processing:

In [17]:

```
video_df.isnull().any()
```

Out[17]:

```
video_id        False
channelTitle    False
title           False
description     False
tags             True
publishedAt     False
viewCount       False
likeCount       False
commentCount    False
dislikeCount     True
duration        False
definition      False
caption         False
dtype: bool
```

In [18]:

```
video_df.dtypes
```

Out[18]:

```
video_id        object
channelTitle    object
title           object
description     object
tags            object
publishedAt     object
viewCount       object
likeCount       object
commentCount    object
dislikeCount    object
duration        object
definition      object
caption         object
dtype: object
```

In [19]:

```
# Convert count columns to numeric
numeric_cols = ['viewCount','likeCount','commentCount','dislikeCount']
video_df[numeric_cols] = video_df[numeric_cols].apply(pd.to_numeric,errors = 'coerce', axis
```

In [20]:

```
# Publish day in the week
video_df['publishedAt'] = video_df['publishedAt'].apply(lambda x: parser.parse(x))
video_df['pushblishDayName'] = video_df['publishedAt'].apply(lambda x: x.strftime("%A"))
```

In [21]:

```
# convert duration to seconds
video_df['durationSecs'] = video_df['duration'].apply(lambda x: isodate.parse_duration(x))
video_df['durationSecs'] = video_df['durationSecs'].astype('timedelta64[s]')
video_df[['durationSecs', 'duration']]
```

Out[21]:

	durationSecs	duration
0	60.0000	PT1M
1	60.0000	PT1M
2	60.0000	PT1M
3	60.0000	PT1M
4	60.0000	PT1M
...	...	...
908	360.0000	PT6M
909	232.0000	PT3M52S
910	603.0000	PT10M3S
911	332.0000	PT5M32S
912	354.0000	PT5M54S

913 rows × 2 columns

In [22]:

```
# Add tag count
video_df['tagCount'] = video_df['tags'].apply(lambda x: 0 if x is None else len(x))
video_df.head()
```

Out[22]:

	video_id	channelTitle	title	description	tags	publishedAt	v
0	70JHBff89Ko	Labour Law Advisor	Insurance ka Loop Hole! #LLAShorts 501	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-15 13:30:03+00:00	7
1	6gi3BdX56ZY	Labour Law Advisor	EMI maaf #LLAShorts 500	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-14 13:30:02+00:00	11
2	Db5wC-xNH80	Labour Law Advisor	Art of Emotional Marketing #LLAShorts 499	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-13 15:37:36+00:00	68
3	MCajCb755PU	Labour Law Advisor	Why is Bluetooth called “Bluetooth”? #LLAShort...	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-12 15:51:57+00:00	31
4	dgmoIqupxGA	Labour Law Advisor	BP Ki Pill hai, Paneer tikka nahi! #LLAShorts 497	Insurance Samadhan : http://bit.ly/3IB5wpP\n.....	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-11 13:41:35+00:00	143

In [23]:

```
# Comments and likes per 1000 view ratio
video_df['likeToViewRatio'] = video_df['likeCount']/ video_df['viewCount'] * 1000
video_df['commentToViewRatio'] = video_df['commentCount']/ video_df['viewCount'] * 1000
# video_df['dislikeToViewRatio'] = video_df['likeCount']/ video_df['viewCount'] * 1000
# video_df['likeToDislikeRatio'] = video_df['likeCount']/ video_df['dislikeCount'] * 1000
video_df.head()
```

Out[23]:

	video_id	channelTitle	title	description	tags	publishedAt	viewCount	
0	70JHBff89Ko	Labour Law Advisor	Insurance ka Loop Hole! #LLAShorts 501	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-15 13:30:03+00:00	77553.0000	1
1	6gi3BdX56ZY	Labour Law Advisor	EMI maaf #LLAShorts 500	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-14 13:30:02+00:00	114578.0000	1
2	Db5wC-xNH80	Labour Law	Art of Emotional Marketing	Open Zerodha A/c, start	[lla, lla shorts, youtube	2023-01-13	686935.0000	6

In [24]:

```
#Marking the shorts and long format videos
video_df['type'] = np.where(video_df['durationSecs']<=60, 'shorts', 'long_video')
video_df.head()
```

Out[24]:

	video_id	channelTitle	title	description	tags	publishedAt	viewCount	
0	70JHBff89Ko	Labour Law Advisor	Insurance ka Loop Hole! #LLAShorts 501	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-15 13:30:03+00:00	77553.0000	1
1	6gi3BdX56ZY	Labour Law Advisor	EMI maaf #LLAShorts 500	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-14 13:30:02+00:00	114578.0000	1
2	Db5wC-xNH80	Labour Law	Art of Emotional Marketing	Open Zerodha A/c, start	[lla, lla shorts, youtube	2023-01-13	686935.0000	6

In [25]:

```
#calculating the engagement each video gets
video_df['engagementScore'] = ((video_df['likeCount']+video_df['commentCount'])/int(channel
video_df.head()
```

Out[25]:

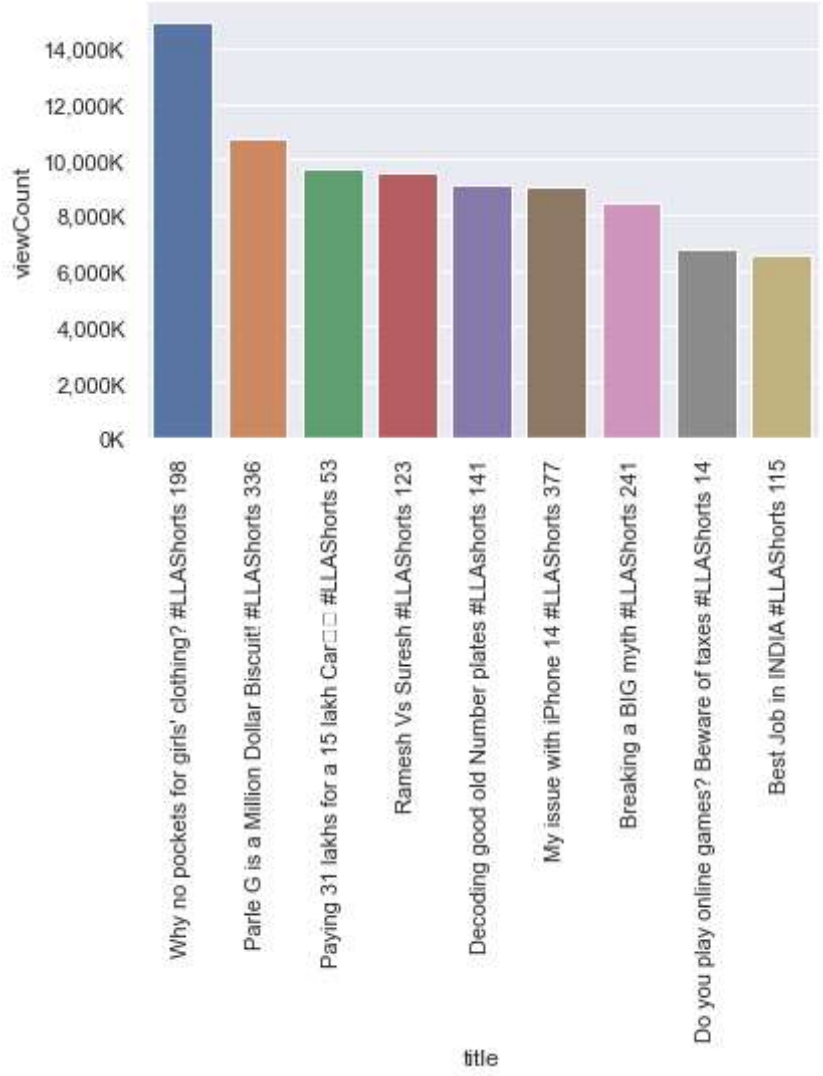
	video_id	channelTitle	title	description	tags	publishedAt	v
0	70JHBff89Ko	Labour Law Advisor	Insurance ka Loop Hole! #LLAShorts 501	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-15 13:30:03+00:00	7
1	6gi3BdX56ZY	Labour Law Advisor	EMI maaf #LLAShorts 500	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-14 13:30:02+00:00	11
2	Db5wC-xNH80	Labour Law Advisor	Art of Emotional Marketing #LLAShorts 499	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-13 15:37:36+00:00	68
3	MCajCb755PU	Labour Law Advisor	Why is Bluetooth called “Bluetooth”? #LLAShort...	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-12 15:51:57+00:00	31
4	dgmoIqupxGA	Labour Law Advisor	BP Ki Pill hai, Paneer tikka nahi! #LLAShorts 497	Insurance Samadhan : http://bit.ly/3IB5wpP\n.....	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-11 13:41:35+00:00	143

EDA

Best performing videos

In [26]:

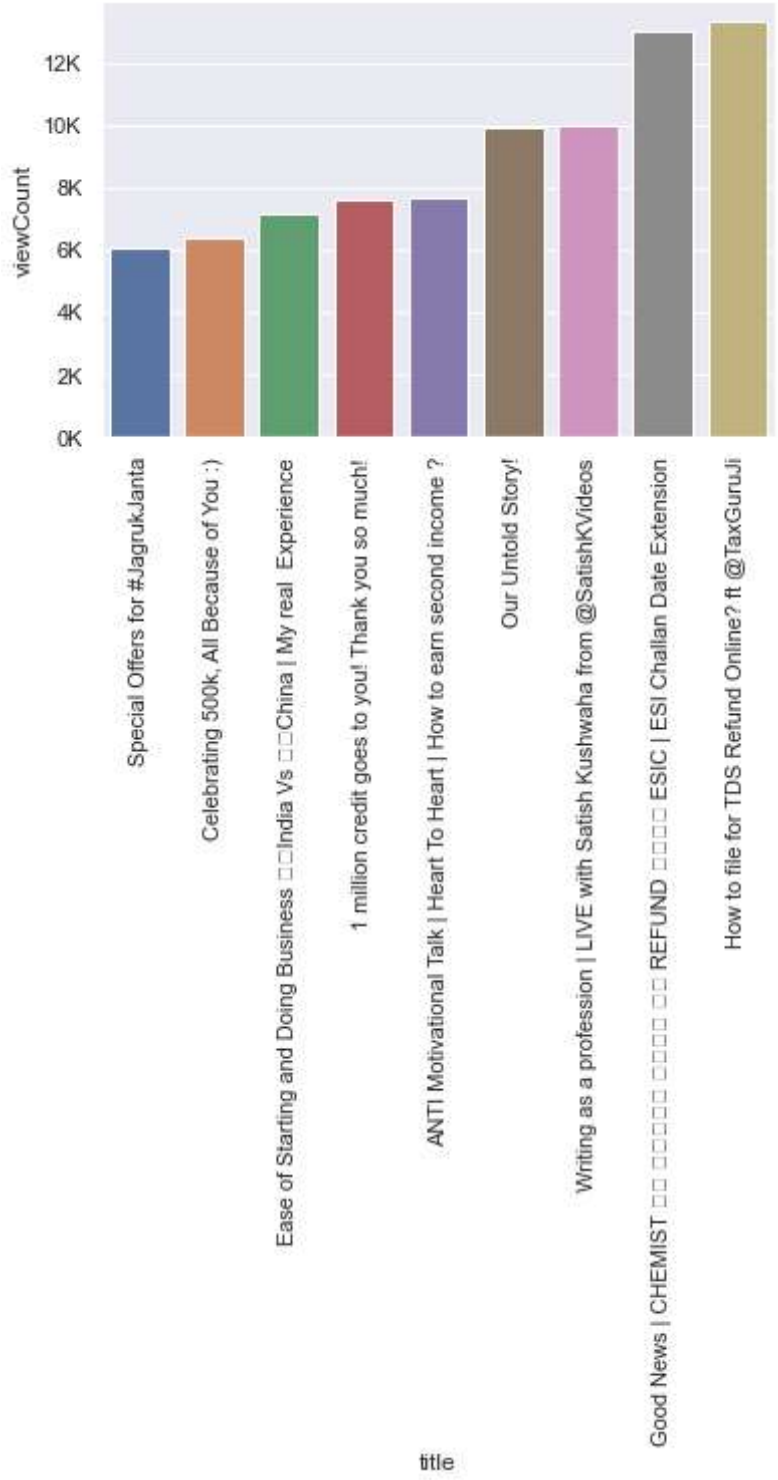
```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(x = 'title', y = 'viewCount', data = video_df.sort_values('viewCount',
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
    sns.set_theme()
```



Worst performing videos

In [27]:

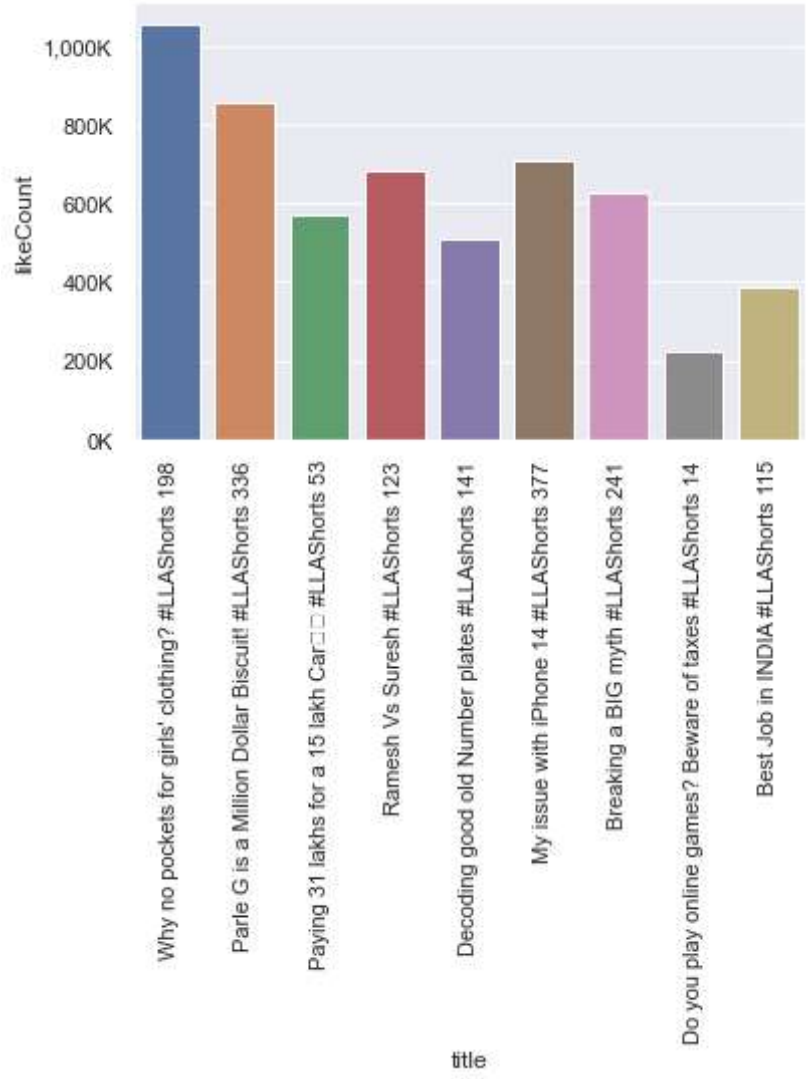
```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(x = 'title', y = 'viewCount', data = video_df.sort_values('viewCount',
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
```



Most Liked Videos

In [28]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(x = 'title', y = 'likeCount', data = video_df.sort_values('viewCount',
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
    sns.set_theme()
```



Most Disliked Videos

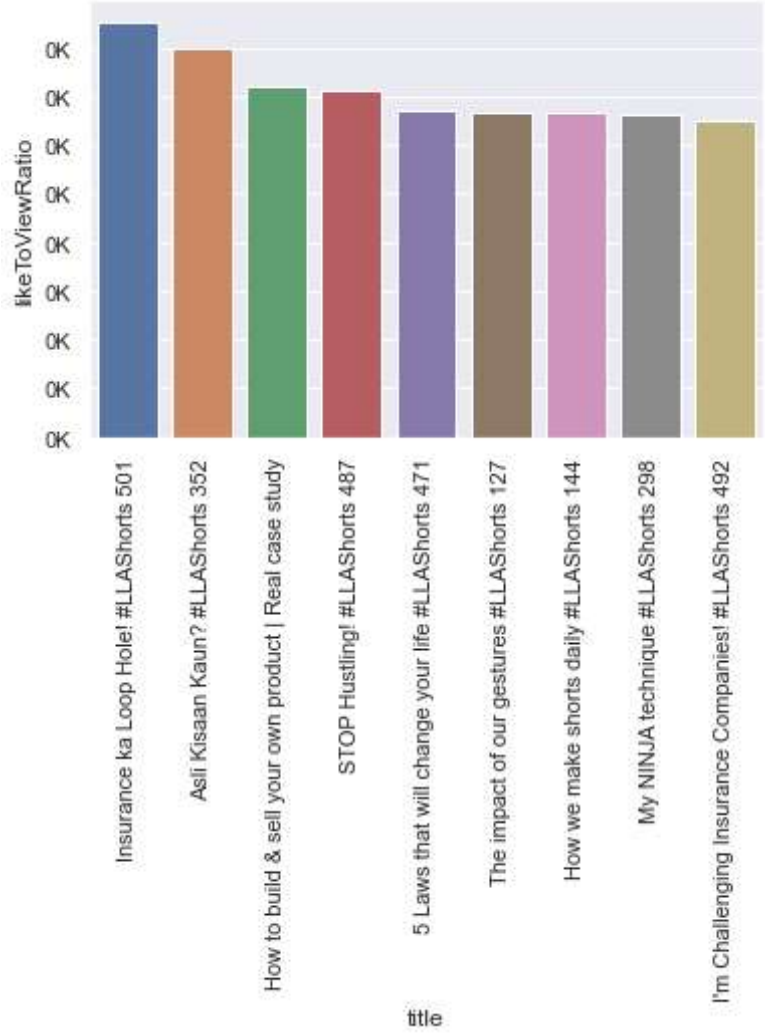
In [29]:

```
# with sns.axes_style("darkgrid"):
#     ax = sns.barplot(x = 'title', y = 'dislikeCount', data = video_df.sort_values('viewCo
#     plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
#     ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/10
#     sns.set_theme()
```

Video with highest likes per 1000 views

In [30]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(x = 'title', y = 'likeToViewRatio', data = video_df.sort_values('likeT
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
    sns.set_theme()
```



Video with highest dislikes per 1000 views

In [31]:

```
# with sns.axes_style("darkgrid"):
#     ax = sns.barplot(x = 'title', y = 'dislikeToViewRatio', data = video_df.sort_values('
#     plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
#     ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/10
#     sns.set_theme()
```

Video with highest likes per 1000 dislikes

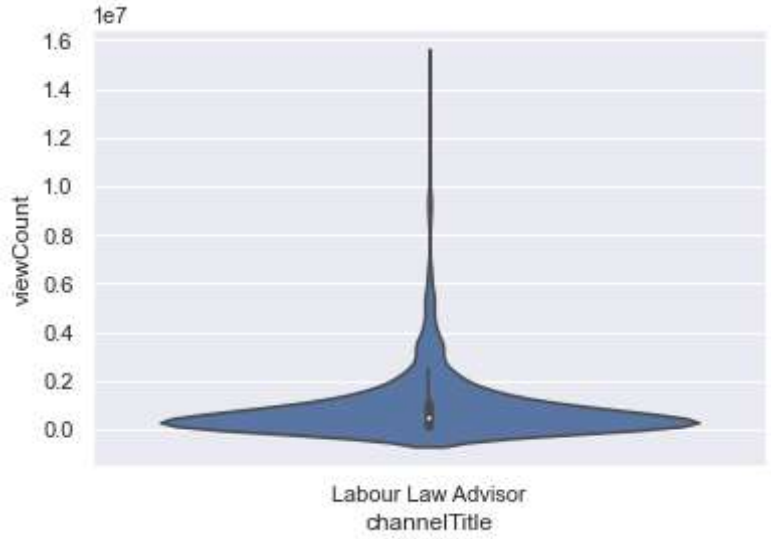
In [32]:

```
# with sns.axes_style("darkgrid"):
#     ax = sns.barplot(x = 'title', y = 'likeToDislikeRatio', data = video_df.sort_values('
#     plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
#     ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/10
#     sns.set_theme()
```

View distribution per video

In [33]:

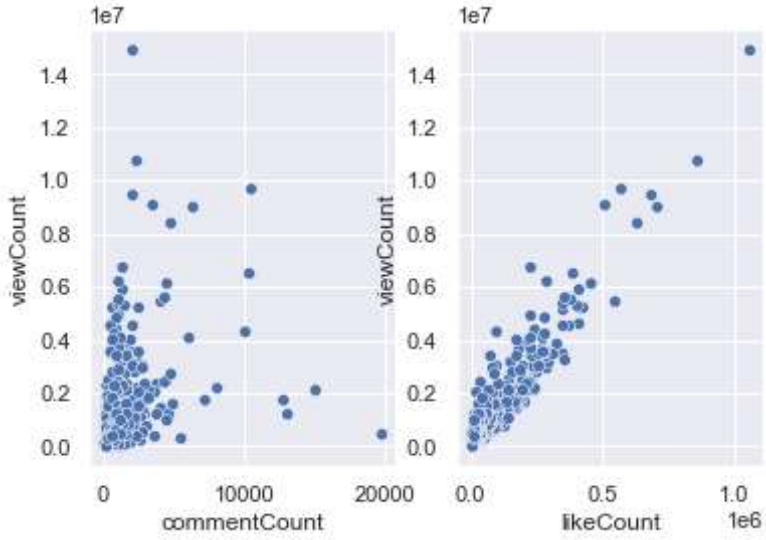
```
with sns.axes_style("darkgrid"):
    sns.violinplot(video_df['channelTitle'], video_df['viewCount'])
```



Views vs. likes and comments

In [34]:

```
with sns.axes_style("darkgrid"):
    fig, ax = plt.subplots(1,2)
    sns.scatterplot(data = video_df, x = 'commentCount', y = 'viewCount', ax = ax[0])
    sns.scatterplot(data = video_df, x = 'likeCount', y = 'viewCount', ax = ax[1])
    # ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/10
```



Distribution of duration of videos

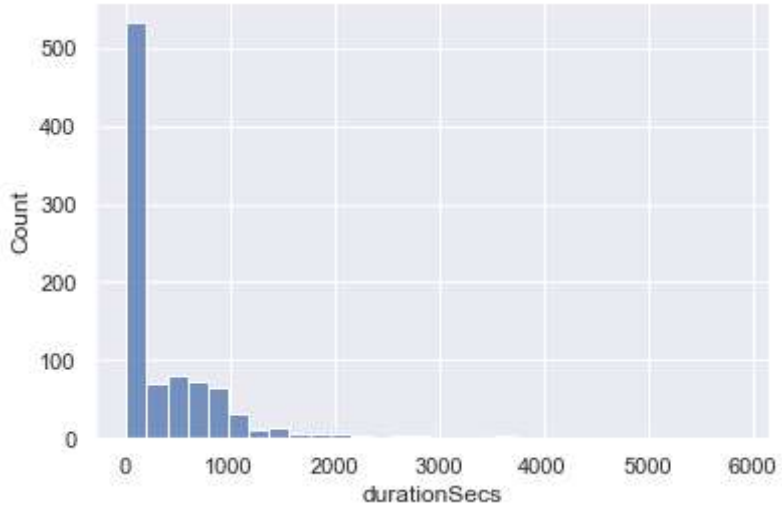
In [35]:

```
#Average duration of videos
print(str(round(statistics.mean(video_df['durationSecs']),2)) + " secs")
```

```
411.83 secs
```

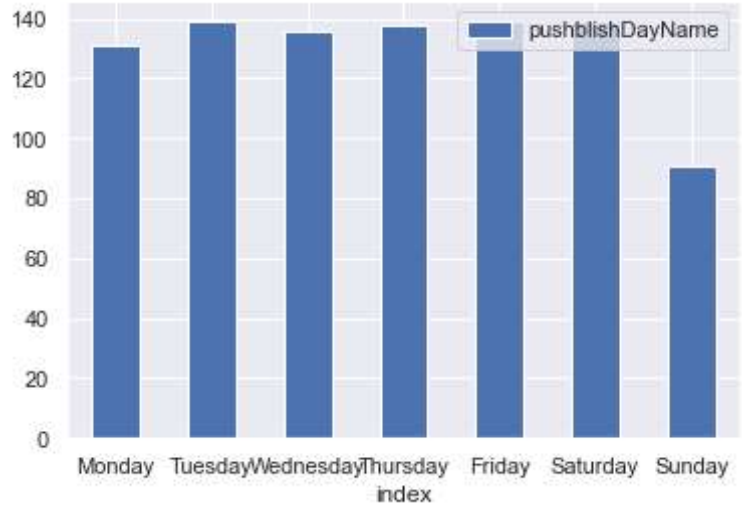
In [36]:

```
with sns.axes_style("darkgrid"):
    sns.histplot(data = video_df, x = 'durationSecs', bins=30)
```



In [37]:

```
day_df = pd.DataFrame(video_df['pushblishDayName'].value_counts())
weekdays = [ 'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday']
day_df = day_df.reindex(weekdays)
with sns.axes_style("darkgrid"):
    ax = day_df.reset_index().plot.bar(x='index', y='pushblishDayName', rot=0)
```

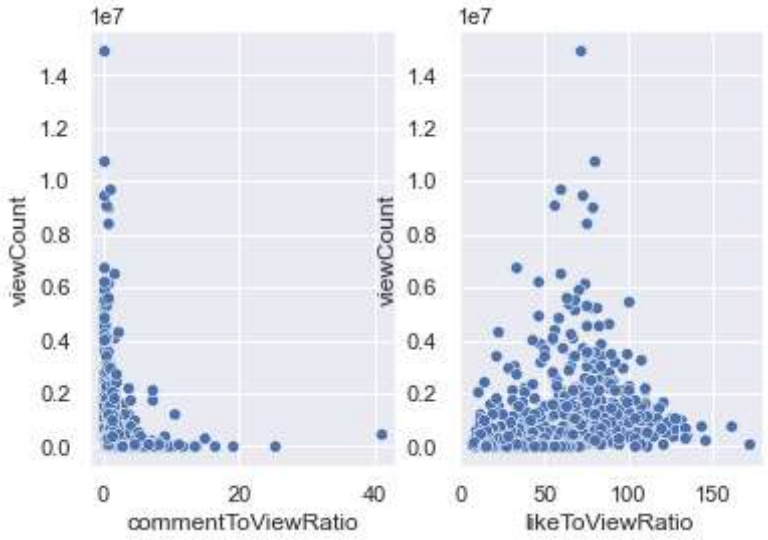


In [38]:

```
# plt.rcParams['figure.figsize'] = (18, 6)
# sns.violinplot(video_df['channelTitle'], video_df['viewCount'], palette = 'pastel')
# plt.title('Views per channel', fontsize = 14)
# plt.show()
```

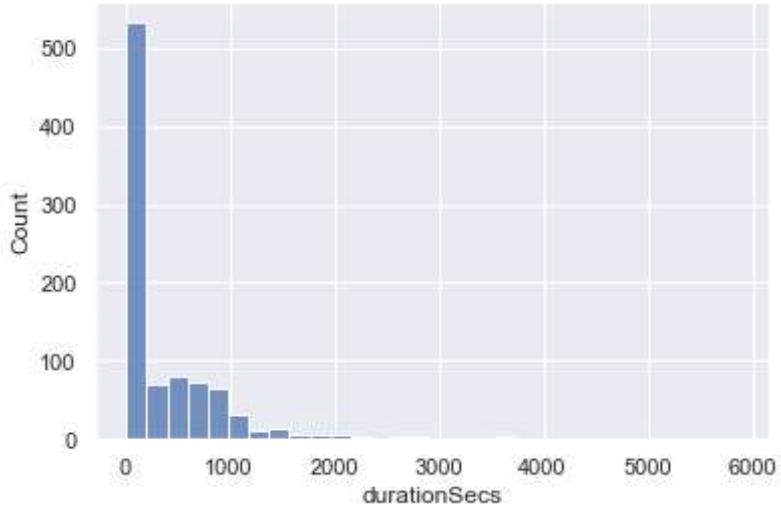
In [39]:

```
with sns.axes_style("darkgrid"):
    fig, ax =plt.subplots(1,2)
    sns.scatterplot(data = video_df, x = "commentToViewRatio", y = "viewCount", ax=ax[0])
    sns.scatterplot(data = video_df, x = "likeToViewRatio", y = "viewCount", ax=ax[1])
```



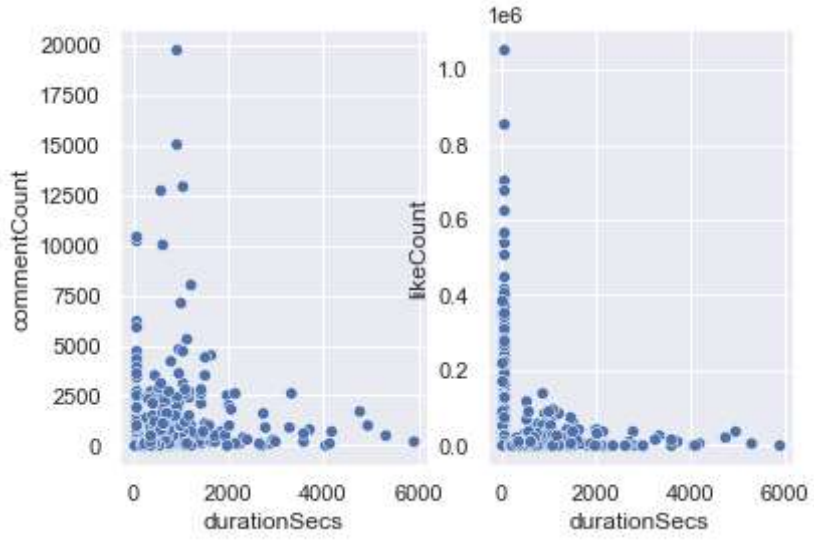
In [40]:

```
with sns.axes_style("darkgrid"):
    sns.histplot(data=video_df[video_df['durationSecs'] < 10000], x="durationSecs", bins=30
```



In [41]:

```
with sns.axes_style("darkgrid"):
    fig, ax =plt.subplots(1,2)
    sns.scatterplot(data = video_df, x = "durationSecs", y = "commentCount", ax=ax[0])
    sns.scatterplot(data = video_df, x = "durationSecs", y = "likeCount", ax=ax[1])
```



In [42]:

```
for i in video_df['title']:
    video_df['titleLength'] = video_df['title'].apply(lambda x: len(x.split()))
video_df.head()
```

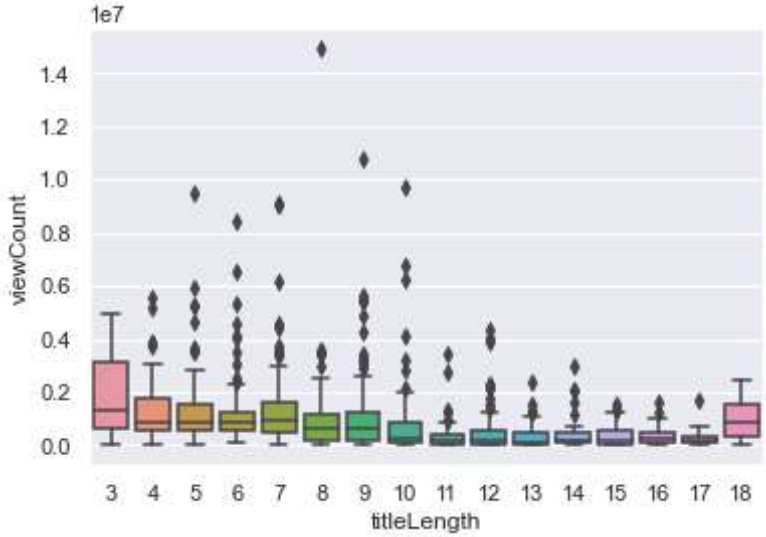
Out[42]:

	video_id	channelTitle	title	description	tags	publishedAt	v
0	70JHBff89Ko	Labour Law Advisor	Insurance ka Loop Hole! #LLAShorts 501	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-15 13:30:03+00:00	7
1	6gi3BdX56ZY	Labour Law Advisor	EMI maaf #LLAShorts 500	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-14 13:30:02+00:00	11
2	Db5wC-xNH80	Labour Law Advisor	Art of Emotional Marketing #LLAShorts 499	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-13 15:37:36+00:00	68
3	MCajCb755PU	Labour Law Advisor	Why is Bluetooth called “Bluetooth”? #LLAShort...	Open Zerodha A/c, start investing:\nhttps://li...	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-12 15:51:57+00:00	31
4	dgmoIqupxGA	Labour Law Advisor	BP Ki Pill hai, Paneer tikka nahi! #LLAShorts 497	Insurance Samadhan : http://bit.ly/3IB5wpP\n.....	[lla, lla shorts, youtube shorts, shorts, yt s...	2023-01-11 13:41:35+00:00	143

5 rows × 21 columns

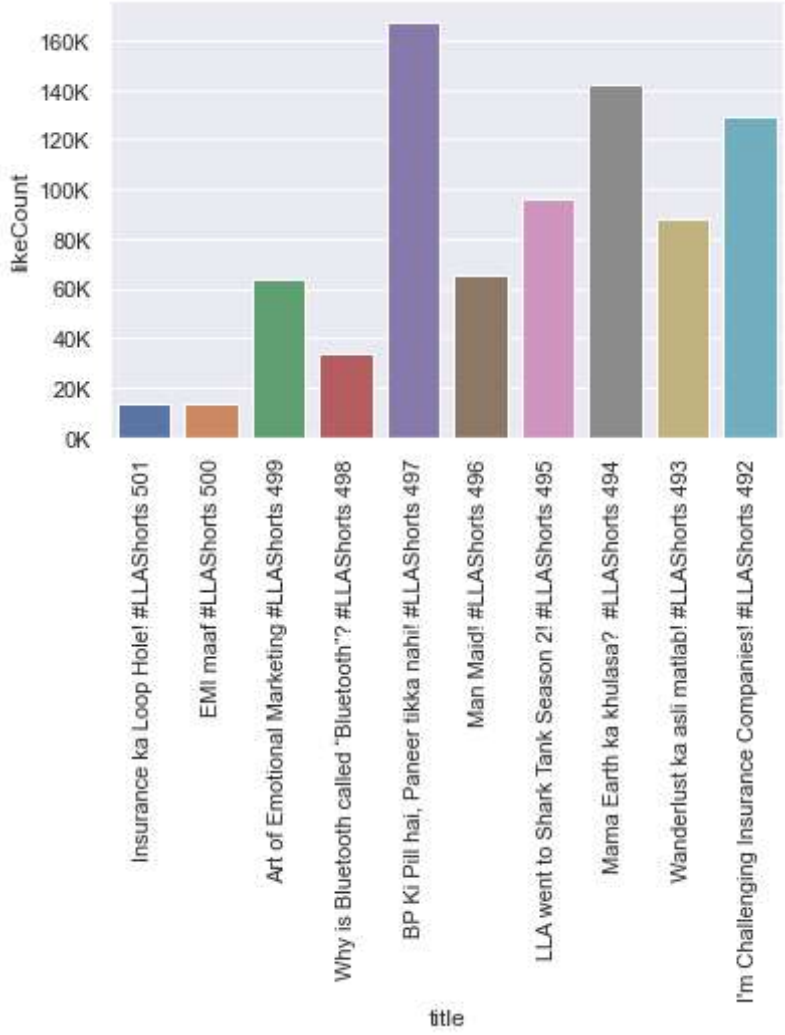
In [43]:

```
with sns.axes_style("darkgrid"):
    sns.boxplot(data = video_df, x = "titleLength", y = "viewCount")
```



In [44]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(x = 'title', y = 'likeCount', data = video_df.head(10))
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
    ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
    sns.set_theme()
```



Now, let's dig a bit deeper by looking at shorts and long videos separately

In [45]:

```
lv = video_df[video_df['type']=='long_video']
sv = video_df[video_df['type']=='shorts']
```

In [46]:

```
type(lv.shape[0])
```

Out[46]:

```
int
```

In [47]:

```
sv.shape
```

Out[47]:

```
(525, 21)
```

In [48]:

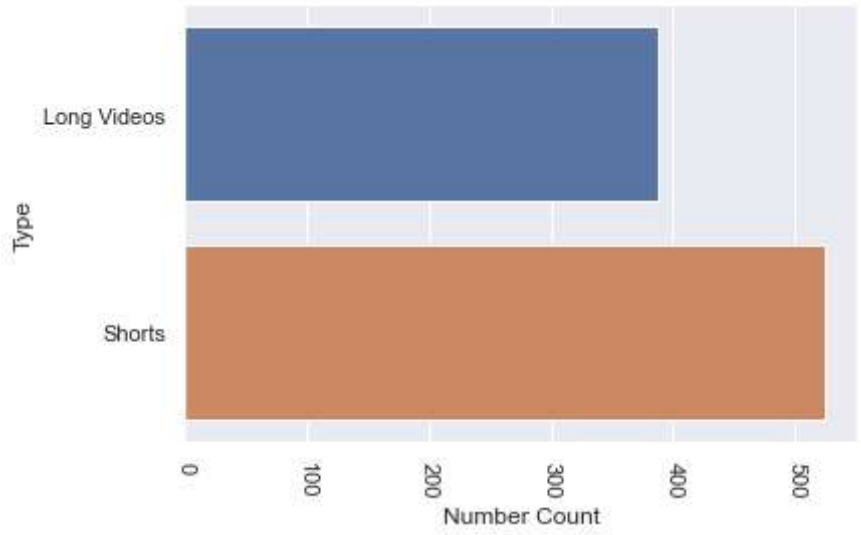
```
pd.DataFrame({'Type':['Long Videos', 'Shorts'], 'Number Count':[lv.shape[0],sv.shape[0]]})
```

Out[48]:

	Type	Number Count
0	Long Videos	388
1	Shorts	525

In [49]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(y = 'Type', x = 'Number Count', data = pd.DataFrame({'Type':['Long Vid
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=-90)
    ax.xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x)))
    sns.set_theme()
```



In [50]:

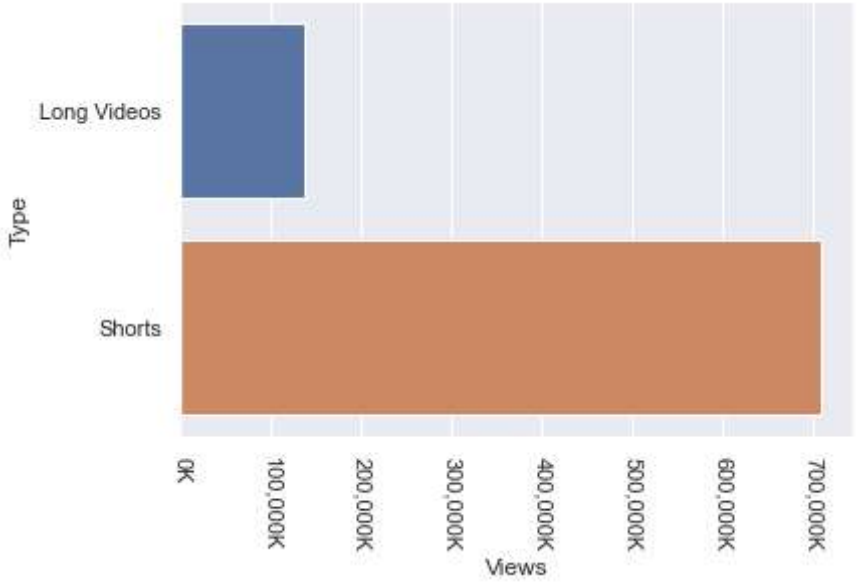
```
pd.DataFrame({'Type':['Long Videos', 'Shorts'], 'Views':[lv['viewCount'].sum(),sv['viewCoun
```

Out[50]:

	Type	Views
0	Long Videos	137552173.0000
1	Shorts	708678617.0000

In [51]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(y = 'Type', x = 'Views', data = pd.DataFrame({'Type':['Long Videos', '
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=-90)
    ax.xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
    sns.set_theme()
```



In [52]:

```
sv['viewCount'].sum()/lv['viewCount'].sum()
```

Out[52]:

```
5.152071403481209
```

In [53]:

```
print(f"Shorts are performing {round(sv['viewCount'].sum()/lv['viewCount'].sum(),2)} times
```

```
Shorts are performing 5.15 times better than long videos in terms of views.
```

In [54]:

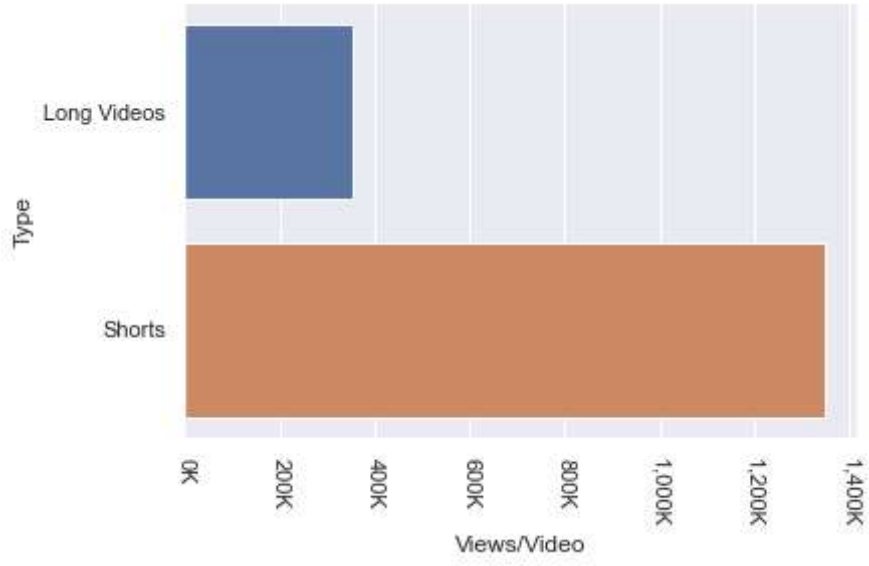
```
# On the flip side, let's check if that's the case when we look at them separately

pd.DataFrame({'Type':['Long Videos', 'Shorts'], 'Views/Video':[lv['viewCount'].sum()/lv.sha
```

Out[54]:

	Type	Views/Video
0	Long Videos	354515.9098
1	Shorts	1349864.0324

In [55]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(y = 'Type', x = 'Views/Video', data = pd.DataFrame({'Type':['Long Vide
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=-90)
    ax.xaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/1000
    sns.set_theme()
```



In [56]:

```
video_df.columns
```

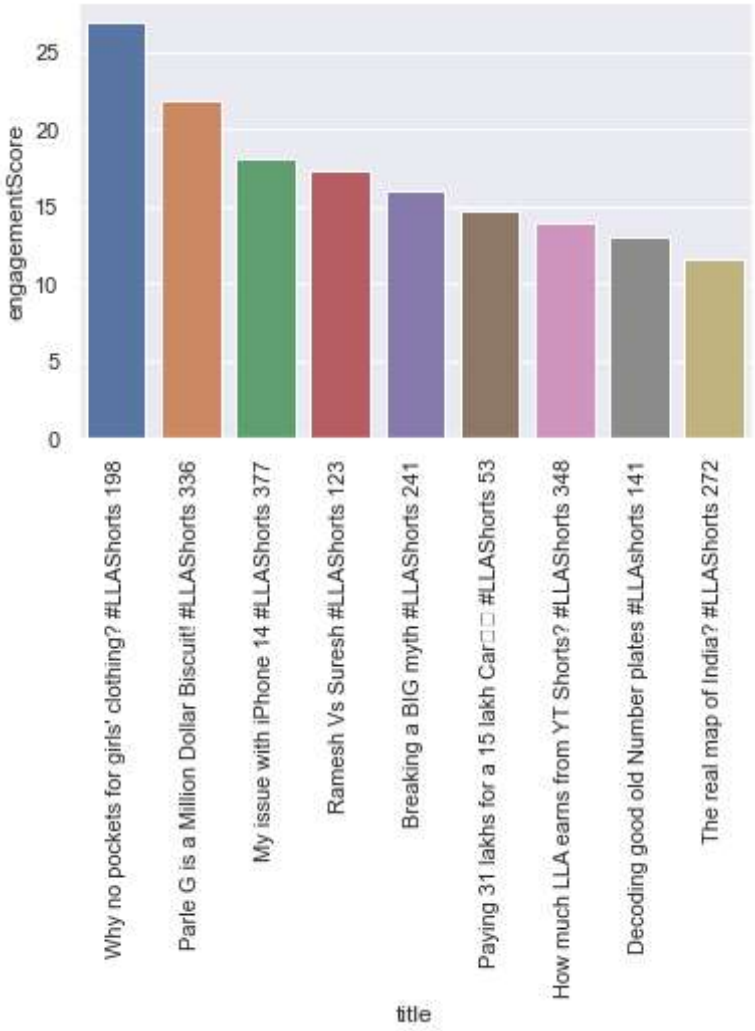
Out[56]:

```
Index(['video_id', 'channelTitle', 'title', 'description', 'tags',
       'publishedAt', 'viewCount', 'likeCount', 'commentCount', 'dislikeCoun
t',
       'duration', 'definition', 'caption', 'pushblishDayName', 'durationSec
s',
       'tagCount', 'likeToViewRatio', 'commentToViewRatio', 'type',
       'engagementScore', 'titleLength'],
      dtype='object')
```

Top 10 videos by engagement score

In [57]:

```
with sns.axes_style("darkgrid"):
    ax = sns.barplot(x = 'title', y = 'engagementScore', data = video_df.sort_values('engag
    plot = ax.set_xticklabels(ax.get_xticklabels(), rotation=90)
#     ax.yaxis.set_major_formatter(ticker.FuncFormatter(lambda x, pos:'{:,.0f}'.format(x/10
    sns.set_theme()
```



In [58]:

```
video_df['publishedAt'] = video_df['publishedAt'].dt.tz_localize(None)
lv['publishedAt'] = video_df['publishedAt'].dt.tz_localize(None)
sv['publishedAt'] = video_df['publishedAt'].dt.tz_localize(None)
```

In [59]:

```
writer = pd.ExcelWriter('viz_data.xlsx', engine='xlsxwriter')
```

In [60]:

```
video_df.to_excel(writer, sheet_name='all videos')
lv.to_excel(writer, sheet_name='long videos')
sv.to_excel(writer, sheet_name='shorts')
```

In [61]:

```
writer.close()
```