Project #2

TikTok User Engagement Exploratory Data Analysis

TikTok is a leading platform for short-form mobile videos. Given the high volume of user reports on videos, TikTok faces the challenge of efficiently reviewing them. To address this, TikTok aims to identify videos that make claims (as opposed to expressing opinions) as they are more likely to violate the platform's terms of service. The goal is to prioritize the review of such videos for potential policy violations.

In [1]:

```
# Import packages for data manipulation
import pandas as pd
import numpy as np

# Import packages for data visualization
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:

```
#Importing the CSV file from Kaggle
data = pd.read_csv("tiktok_dataset.csv")
```

In [3]:

```
# Print the head of the dataframe to sanity check
print(data.head())
```

```
   # claim_status    video_id  video_duration_sec  \
0  1        claim  7017666017                  59   
1  2        claim  4014381136                  32   
2  3        claim  9859838091                  31   
3  4        claim  1866847991                  25   
4  5        claim  7105231098                  19   

                            video_transcription_text verified_status  \
0  someone shared with me that drone deliveries a...    not verified   
1  someone shared with me that there are more mic...    not verified   
2  someone shared with me that american industria...    not verified   
3  someone shared with me that the metro of st. p...    not verified   
4  someone shared with me that the number of busi...    not verified   

  author_ban_status  video_view_count  video_like_count  video_share_coun
t  \
0      under review          343296.0           19425.0              241.
0   
1            active          140877.0           77355.0            19034.
0   
2            active          902185.0           97690.0             2858.
0   
3            active          437506.0          239954.0            34812.
0   
4            active           56167.0           34987.0             4110.
0   

   video_download_count  video_comment_count  
0                   1.0                  0.0  
1                1161.0                684.0  
2                 833.0                329.0  
3                1234.0                584.0  
4                 547.0                152.0  
```

In [4]:

```
# Print the tail of the dataframe to sanity check
print(data.tail())
```

```
           # claim_status    video_id  video_duration_sec  \
19377  19378          NaN  7578226840                  21   
19378  19379          NaN  6079236179                  53   
19379  19380          NaN  2565539685                  10   
19380  19381          NaN  2969178540                  24   
19381  19382          NaN  8132759688                  13   

      video_transcription_text verified_status author_ban_status  \
19377                      NaN    not verified            active   
19378                      NaN    not verified            active   
19379                      NaN        verified      under review   
19380                      NaN    not verified            active   
19381                      NaN    not verified            active   

       video_view_count  video_like_count  video_share_count  \
19377               NaN               NaN                NaN   
19378               NaN               NaN                NaN   
19379               NaN               NaN                NaN   
19380               NaN               NaN                NaN   
19381               NaN               NaN                NaN   

       video_download_count  video_comment_count  
19377                   NaN                  NaN  
19378                   NaN                  NaN  
19379                   NaN                  NaN  
19380                   NaN                  NaN  
19381                   NaN                  NaN  
```

In [5]:

```
# Output number of rows and columns
data.shape
```

Out[5]:

```
(19382, 12)
```

In [6]:

```
# Output basic information
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 19382 entries, 0 to 19381
Data columns (total 12 columns):
 #   Column                    Non-Null Count  Dtype  
---  ------                    --------------  -----  
 0   #                         19382 non-null  int64  
 1   claim_status              19084 non-null  object 
 2   video_id                  19382 non-null  int64  
 3   video_duration_sec        19382 non-null  int64  
 4   video_transcription_text  19084 non-null  object 
 5   verified_status           19382 non-null  object 
 6   author_ban_status         19382 non-null  object 
 7   video_view_count          19084 non-null  float64
 8   video_like_count          19084 non-null  float64
 9   video_share_count         19084 non-null  float64
 10  video_download_count      19084 non-null  float64
 11  video_comment_count       19084 non-null  float64
dtypes: float64(5), int64(3), object(4)
memory usage: 1.8+ MB
```

In [7]:

```
# Check for missing values
data.isna().sum()
```

Out[7]:

```
#                             0
claim_status                298
video_id                      0
video_duration_sec            0
video_transcription_text    298
verified_status               0
author_ban_status             0
video_view_count            298
video_like_count            298
video_share_count           298
video_download_count        298
video_comment_count         298
dtype: int64
```

In [8]:

```
# Generate a table of descriptive statistics about the data
data.describe()
```

Out[8]:

	#	video_id	video_duration_sec	video_view_count	video_like_count
count	19382.000000	1.938200e+04	19382.000000	19084.000000	19084.000000
mean	9691.500000	5.627454e+09	32.421732	254708.558688	84304.636030
std	5595.245794	2.536440e+09	16.229967	322893.280814	133420.546814
min	1.000000	1.234959e+09	5.000000	20.000000	0.000000
25%	4846.250000	3.430417e+09	18.000000	4942.500000	810.750000
50%	9691.500000	5.618664e+09	32.000000	9954.500000	3403.500000
75%	14536.750000	7.843960e+09	47.000000	504327.000000	125020.000000
max	19382.000000	9.999873e+09	60.000000	999817.000000	657830.000000

In [9]:

```
# Drop rows with missing values
data = data.dropna(axis=0)
```

In [10]:

```
# Print the tail of the dataframe now to check the data after dropping the
print(data.tail())
```

```
           # claim_status    video_id  video_duration_sec  \
19079  19080      opinion  1492320297                  49
19080  19081      opinion  9841347807                  23
19081  19082      opinion  8024379946                  50
19082  19083      opinion  7425795014                   8
19083  19084      opinion  4094655375                  58

                                video_transcription_text verified_status
\
19079  in our opinion the earth holds about 11 quinti...    not verified
19080  in our opinion the queens in ant colonies live...    not verified
19081  in our opinion the moon is moving away from th...    not verified
19082  in our opinion lightning strikes somewhere on ...    not verified
19083  in our opinion a pineapple plant can only prod...    not verified

      author_ban_status  video_view_count  video_like_count  \
19079            active            6067.0             423.0
19080            active            2973.0             820.0
19081            active             734.0             102.0
19082            active            3394.0             655.0
19083            active            5034.0             815.0

       video_share_count  video_download_count  video_comment_count
19079               81.0                   8.0                  2.0
19080               70.0                   3.0                  0.0
19081                7.0                   2.0                  1.0
19082              123.0                  11.0                  4.0
19083              281.0                  11.0                  1.0
```

In [11]:

```
# Now let us create a text_length column
data['text_length'] = data['video_transcription_text'].str.len()
data.head()
```

Out[11]:

	#	claim_status	video_id	video_duration_sec	video_transcription_text	verified_status
0	1	claim	7017666017	59	someone shared with me that drone deliveries a...	not verified
1	2	claim	4014381136	32	someone shared with me that there are more mic...	not verified
2	3	claim	9859838091	31	someone shared with me that american industria...	not verified
3	4	claim	1866847991	25	someone shared with me that the metro of st. p...	not verified
4	5	claim	7105231098	19	someone shared with me that the number of busi...	not verified

In [12]:

```
# Compute the mean `video_view_count` for each group in `verified_status`
data.groupby("verified_status")["video_duration_sec"].mean()
```

Out[12]:

```
verified_status
not verified    32.467345
verified        31.775000
Name: video_duration_sec, dtype: float64
```

In [13]:

```
# Compute the mean count of characters in text_length for each claim_status
data[['claim_status', 'text_length']].groupby('claim_status').mean()
```

Out[13]:

	text_length
claim_status	
claim	95.376978
opinion	82.722562

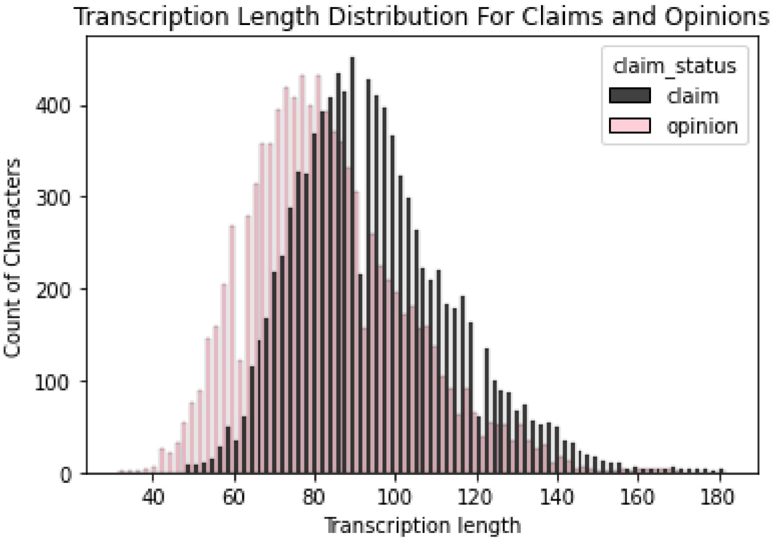
In [29]:

```
mean_duration = np.mean(video_duration_sec)
print(f"Mean Video Duration: {mean_duration} seconds")
mean_likes = np.mean(video_like_count)
print(f"Mean Like Count: {video_like_count}")
claims = data[data['claim_status'] == 'claim']
print('Mean view count claims:', claims['video_view_count'].mean())
claims = data[data['claim_status'] == 'opinion']
print('Mean view count claims:', claims['video_view_count'].mean())
```

```
Mean Video Duration: 36.25 seconds
Mean Like Count: [80000, 120000, 50000, 90000]
Mean view count claims: 501029.4527477102
Mean view count claims: 4956.43224989447
```

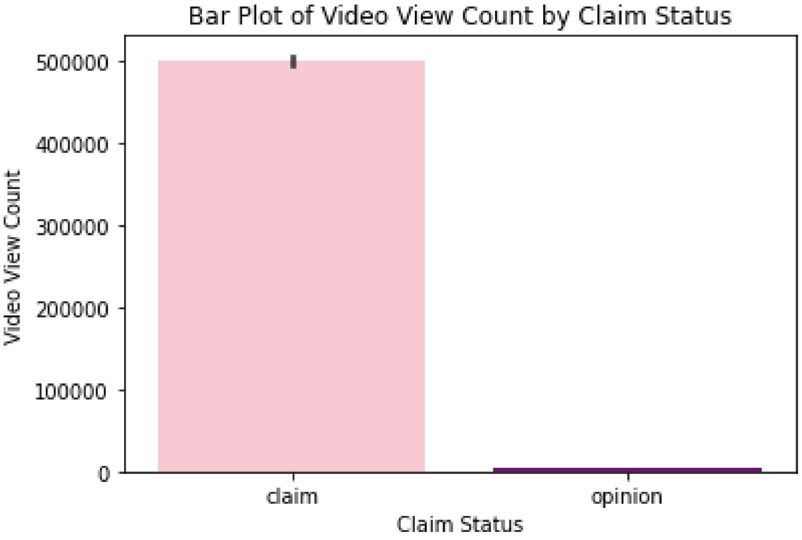
In [14]:

```
# Histogram plot distribution of Transcription length for claims and opinio
sns.histplot(data=data, stat="count", multiple="dodge", x= "text_length",
             palette=["black", "pink"], element="bars", legend=True)
plt.xlabel("Transcription length")
plt.ylabel("Count of Characters")
plt.title("Transcription Length Distribution For Claims and Opinions")
plt.show()
```



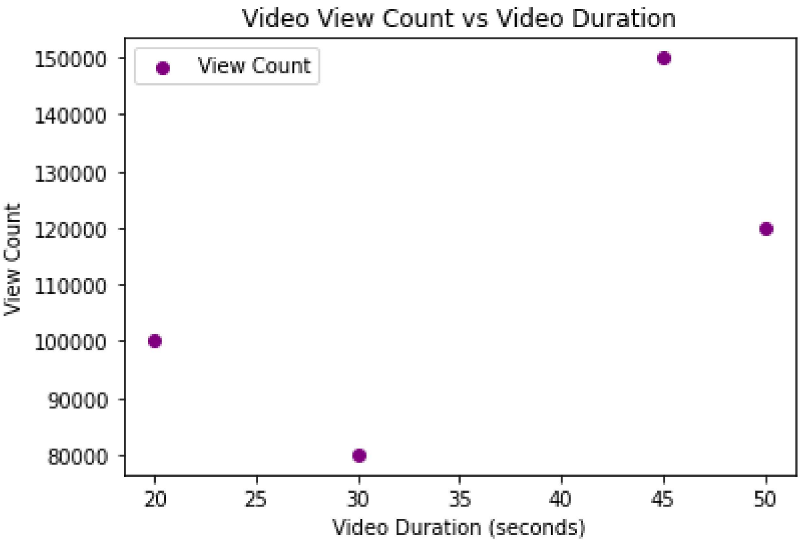
In [15]:

```
# Bar plot of video view count for each claim status
sns.barplot(x="claim_status", y="video_view_count", data=data, palette=["p
plt.title('Bar Plot of Video View Count by Claim Status')
plt.xlabel('Claim Status')
plt.ylabel('Video View Count')
plt.show()
```



In [18]:

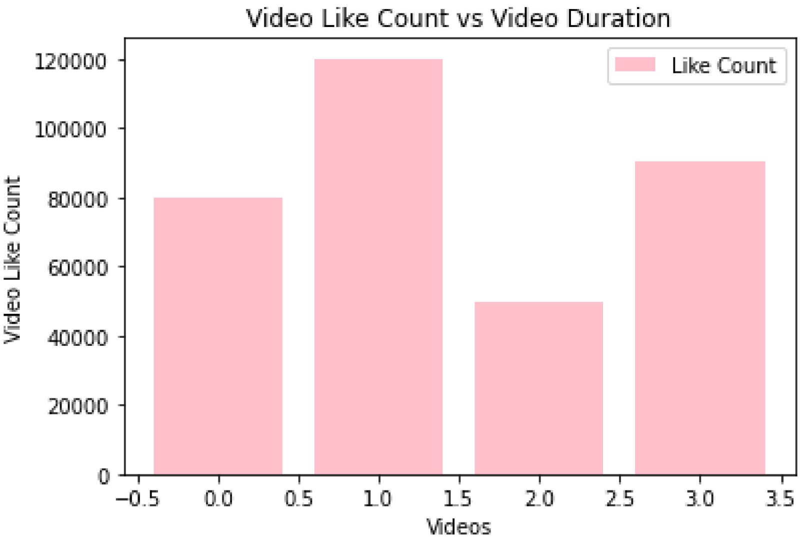
```
video_view_count = [100000, 150000, 80000, 120000]
video_duration_sec = [20, 45, 30, 50]
plt.scatter(video_duration_sec, video_view_count, color='purple', label='Vi
plt.xlabel('Video Duration (seconds)')
plt.ylabel('View Count')
plt.title('Video View Count vs Video Duration')
plt.legend()
plt.show()
```



In [21]:

```
video_like_count = [80000, 120000, 50000, 90000]
video_duration_sec = [20, 45, 30, 50]
plt.bar(range(len(video_duration_sec)), video_like_count, label='Like Count
plt.xlabel('Videos')
plt.ylabel('Video Like Count')
plt.title('Video Like Count vs Video Duration')
plt.legend()

plt.show()
```

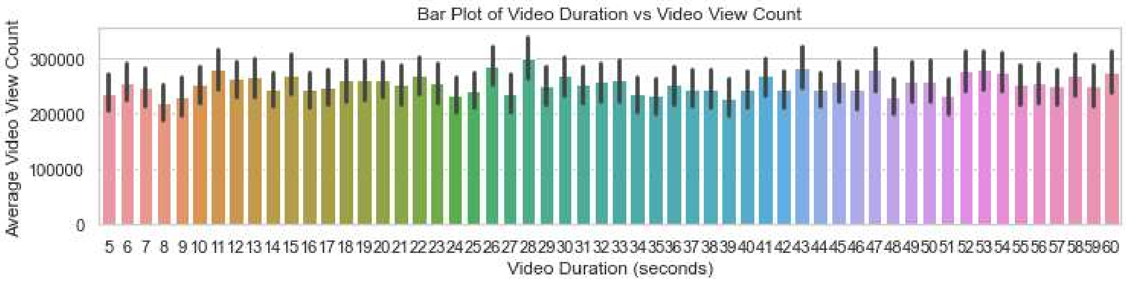


In [15]:

```
# Bar plot of average video_duration_sec for each variable
plt.figure(figsize=(12, 8))
plt.subplot(3, 1, 1)
sns.barplot(x="video_duration_sec", y="video_view_count", data=data)
plt.title('Bar Plot of Video Duration vs Video View Count')
plt.xlabel('Video Duration (seconds)')
plt.ylabel('Average Video View Count')
```

Out[15]:

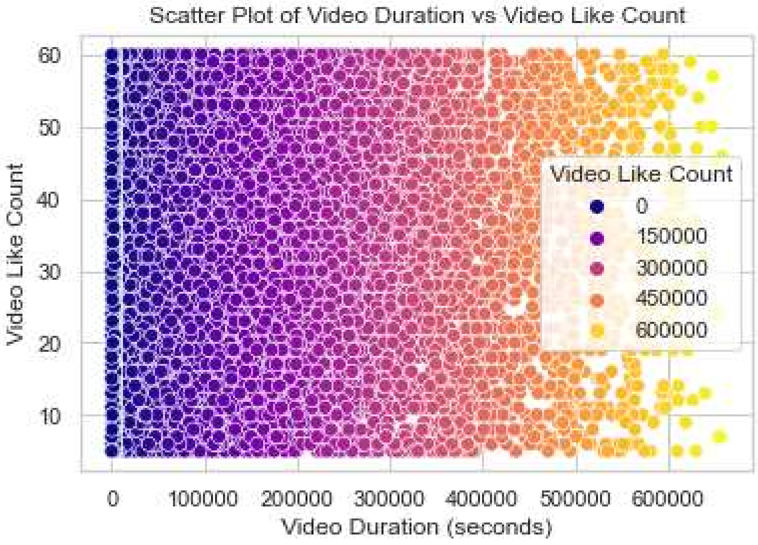
```
Text(0, 0.5, 'Average Video View Count')
```



In [48]:

```
# Scatter plot for video_duration_sec against video_like_count
sns.scatterplot(data=data, x="video_like_count", y="video_duration_sec", hu
                palette="plasma", s=50)
plt.xlabel("Video Duration (seconds)")
plt.ylabel("Video Like Count")

plt.title("Scatter Plot of Video Duration vs Video Like Count")
plt.legend(title="Video Like Count")
plt.show()
```


Scatter Plot of Video Duration vs Video Like Count
Video Like Count
Video Duration (seconds)
Video Like Count
● 0
● 150000
● 300000
● 450000
● 600000