TikTok_project_EDA

April 18, 2025

1 TikTok project: Exploratory data analysis

The purpose of this project is to conduct exploratory data analysis on a provided data set. Your mission is to perform further EDA on this data with the aim of learning more about the variables. Of particular interest is information related to what distinguishes claim videos from opinion videos.

The goal is to explore the dataset and create visualizations. *This activity has 4 parts:*

Part 1: Imports, links, and loading

Part 2: Data Exploration * Data cleaning

Part 3: Build visualizations

Part 4: Evaluate and share results

2 Visualize a story in Python

Consider the questions in for planning and those below where applicable to craft your response: 1. Identify any outliers:

- What methods are best for identifying outliers?
- How do you make the decision to keep or exclude outliers from any future models?

Response:

1) What methods are best for identifying outliers?

- Use numpy functions to investigate the `mean()` and `median()` of the data and understand range of data values
- Use a boxplot to visualize the distribution of the data

2) How do you make the decision to keep or exclude outliers from any future models?

- There are three main options for dealing with outliers: keeping them as they are, deleting them, or reassigning them. Whether you keep outliers as they are, delete them, or reassign values is a decision that you make on a dataset-by-dataset basis, according to what your goals are for the model you are planning to construct.
- To help you make the decision, you can start with these general guidelines:
  - Delete them: If you are sure the outliers are mistakes, typos, or errors and the dataset will be used for modeling or machine learning, then you are more likely to decide to delete outliers. Of the three choices, you'll use this one the least.

- Reassign them: If the dataset is small and/or the data will be used for modeling or machine learning, you are more likely to choose a path of deriving new values to replace the outlier values.
- Leave them: For a dataset that you plan to do EDA/analysis on and nothing else, or for a dataset you are preparing for a model that is resistant to outliers, it is most likely that you are going to leave them in.

2.0.1 Task 1. Imports, links, and loading

For EDA of the data, import the packages that would be most helpful, such as `pandas`, `numpy`, `matplotlib.pyplot`, and `seaborn`.

[1]:
```
# Import packages for data manipulation
import numpy as np
import pandas as pd


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

```
Matplotlib is building the font cache; this may take a moment.
```

Then, load the dataset into a dataframe. Read in the data and store it as a dataframe object.

[3]:
```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

2.0.2 Task 2a: Data exploration and cleaning

The first step is to assess your data. Check the Data Source page on Tableau Public to get a sense of the size, shape and makeup of the data set.

Consider functions that help you understand and structure the data.

- `.head()`
- `.info()`
- `.describe()`
- `.groupby()`
- `.sort_values()`

Consider the following questions as you work:

What do you do about missing data (if any)?

Are there data outliers?

Start by discovering, using `.head()`, `.size`, and `.shape`.

[5]:
```
# Display and examine the first few rows of the dataframe
data.head()
```

[5]:
```
   # 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_count  \
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
```

[9]:
```
# Get the size of the data
data.size
```

[9]:
```
232584
```

[11]:
```
# Get the shape of the data
data.shape
```

[11]:
```
(19382, 12)
```

Get basic information about the data, using `.info()`.

[13]:
```
# Get basic information about the data
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
```

Generate a table of descriptive statistics, using `.describe()`.

[15]:
```
# Generate a table of descriptive statistics
data.describe()
```

[15]:

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

	video_like_count	video_share_count	video_download_count	\
count	19084.000000	19084.000000	19084.000000	
mean	84304.636030	16735.248323	1049.429627	
std	133420.546814	32036.174350	2004.299894	
min	0.000000	0.000000	0.000000	
25%	810.750000	115.000000	7.000000	
50%	3403.500000	717.000000	46.000000	
75%	125020.000000	18222.000000	1156.250000	
max	657830.000000	256130.000000	14994.000000	

	video_comment_count
count	19084.000000
mean	349.312146
std	799.638865
min	0.000000
25%	1.000000
50%	9.000000
75%	292.000000
max	9599.000000

2.0.3 Task 2b. Select visualization type(s)

Select data visualization types that will help you understand and explain the data.

Now that you know which data columns you'll use, it is time to decide which data visualization makes the most sense for EDA of the TikTok dataset. What type of data visualization(s) would be most helpful? Consider the distribution of the data.

- Line graph
- Bar chart
- Box plot
- Histogram
- Heat map
- Scatter plot
- A geographic map

Response:

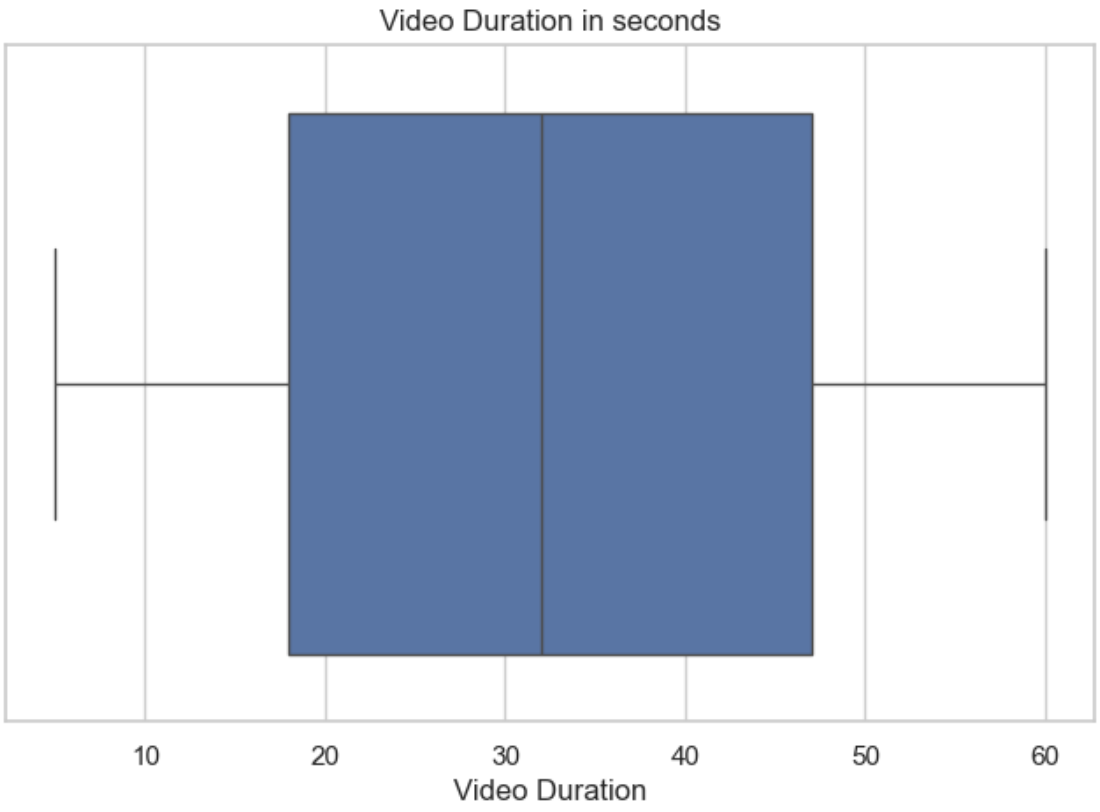
The visualizations most helpful for considering the distribution of the data include box plots and histograms. Visualizing the distribution of the data can inform the next steps and considerations in data analysis. For example, data distribution will inform which types of modeling is needed.

2.0.4 Task 3. Build visualizations

Now that you have assessed your data, it's time to plot your visualization(s).

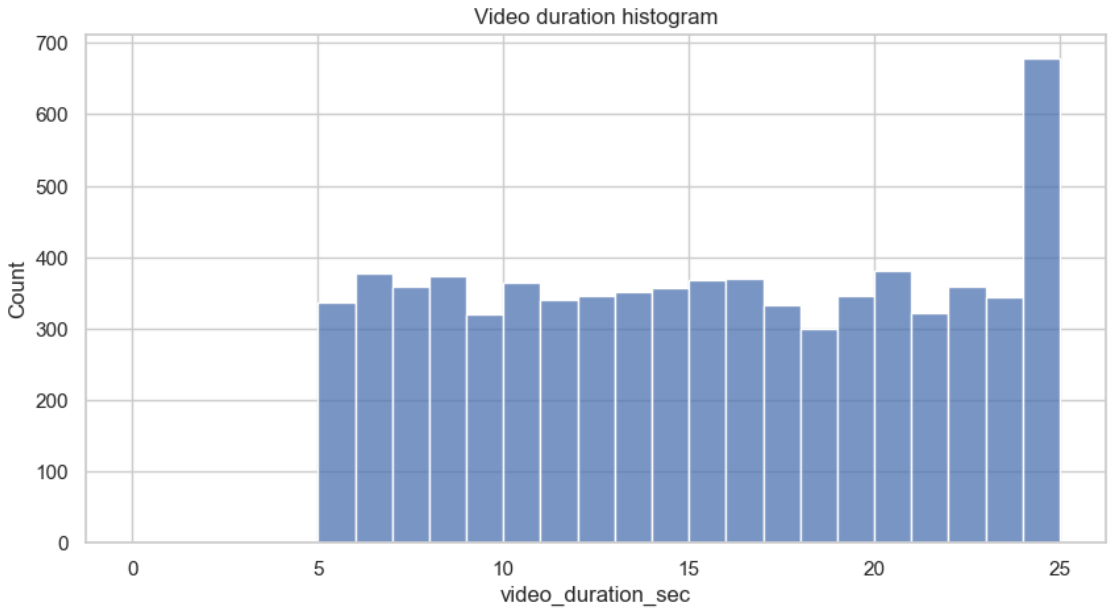
video_duration_sec Create a box plot to examine the spread of values in the `video_duration_sec` column.

[17]:
```
# Create a boxplot to visualize distribution of `video_duration_sec`
### YOUR CODE HERE ###

sns.set(style="whitegrid")  # Set seaborn style

plt.figure(figsize=(8, 5))  # Set figure size
box = sns.boxplot(x=data['video_duration_sec'])  # Create boxplot

g = plt.gca()  # Get current axes
plt.xlabel('Video Duration')
plt.title('Video Duration in seconds')

plt.show()  # Display plot
```



Create a histogram of the values in the `video_duration_sec` column to further explore the distribution of this variable.

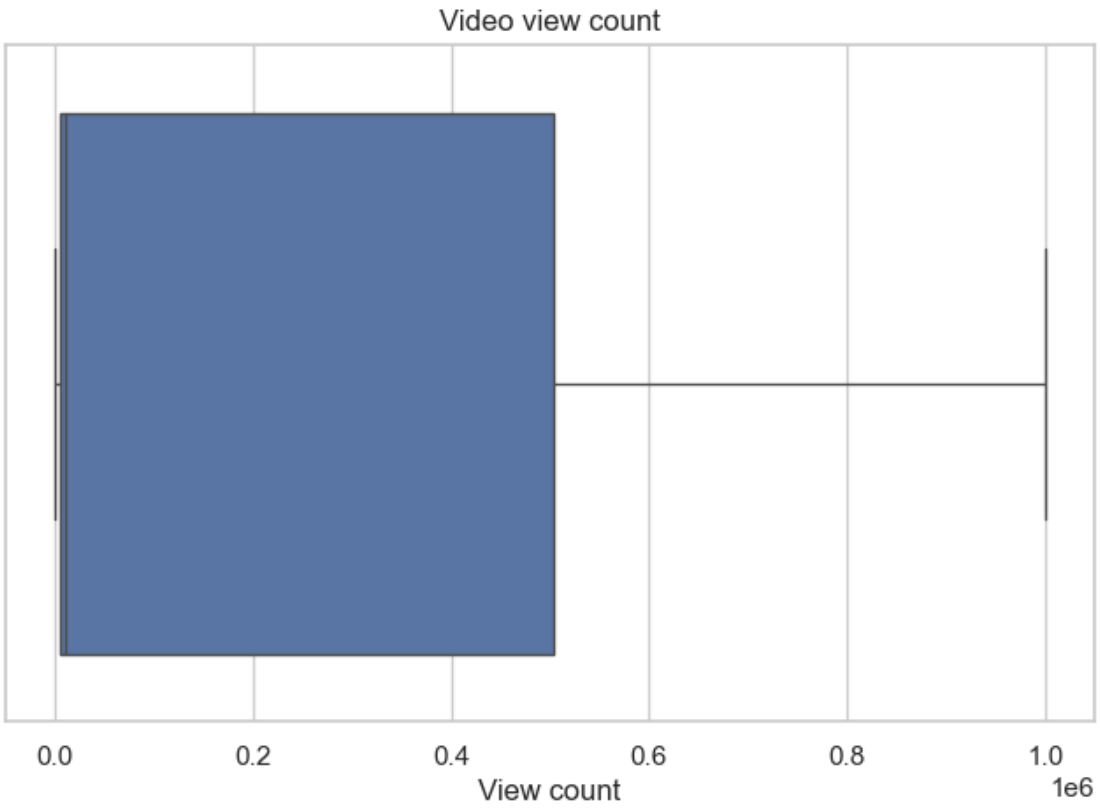
[19]:
```
# Create a histogram
plt.figure(figsize=(10,5))
sns.histplot(data['video_duration_sec'], bins=range(0,26,1))
plt.title('Video duration histogram');
```



Question: What do you notice about the duration and distribution of the videos? **Response:** All videos are 5-60 seconds in length, and the distribution is uniform.

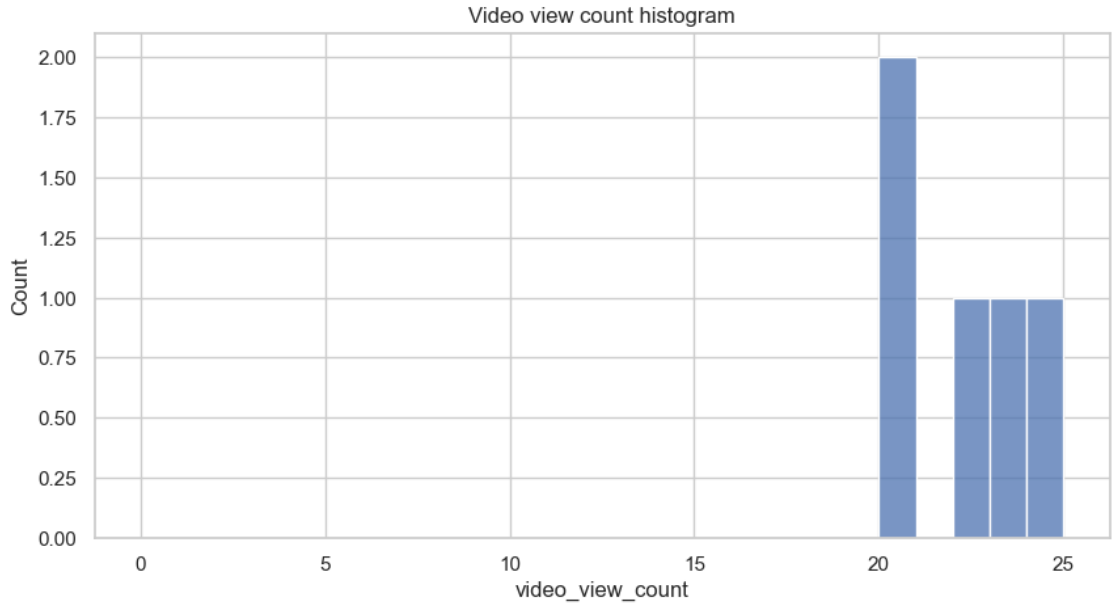
video_view_count Create a box plot to examine the spread of values in the `video_view_count` column.

[21]:
```
# Create a boxplot to visualize distribution of `video_view_count`

sns.set(style="whitegrid")  # Set seaborn style

plt.figure(figsize=(8, 5))  # Set figure size
box = sns.boxplot(x=data['video_view_count'])  # Create boxplot

g = plt.gca()  # Get current axes
plt.xlabel('View count')
plt.title('Video view count')

plt.show()  # Display plot
```



Create a histogram of the values in the `video_view_count` column to further explore the distribution of this variable.

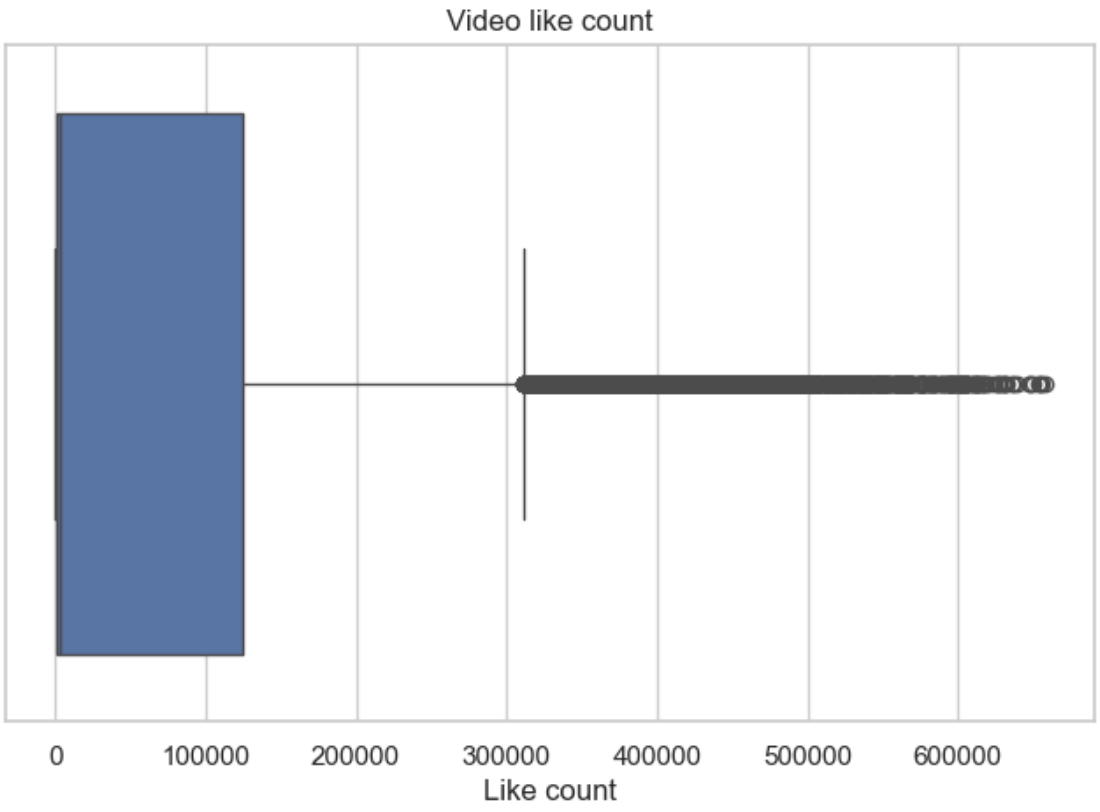
[23]:
```
# Create a histogram
plt.figure(figsize=(10,5))
sns.histplot(data['video_view_count'], bins=range(0,26,1))
plt.title('Video view count histogram');
```



Question: What do you notice about the distribution of this variable? **Response:** This variable has a very uneven distribution, with more than half the videos receiving fewer than 100,000 views. Distribution of view counts > 100,000 views is uniform.

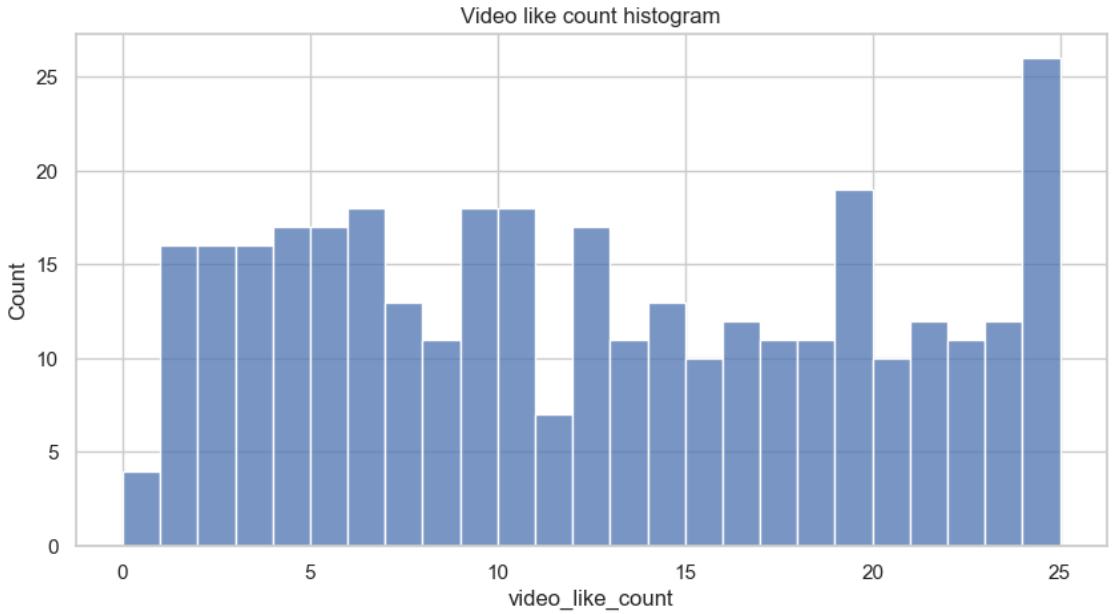
video_like_count Create a box plot to examine the spread of values in the `video_like_count` column.

[25]:
```
# Create a boxplot to visualize distribution of `video_like_count`

sns.set(style="whitegrid")  # Set seaborn style

plt.figure(figsize=(8, 5))  # Set figure size
box = sns.boxplot(x=data['video_like_count'])  # Create boxplot

g = plt.gca()  # Get current axes
plt.xlabel('Like count')
plt.title('Video like count')

plt.show()  # Display plot
```



Create a histogram of the values in the `video_like_count` column to further explore the distribution of this variable.

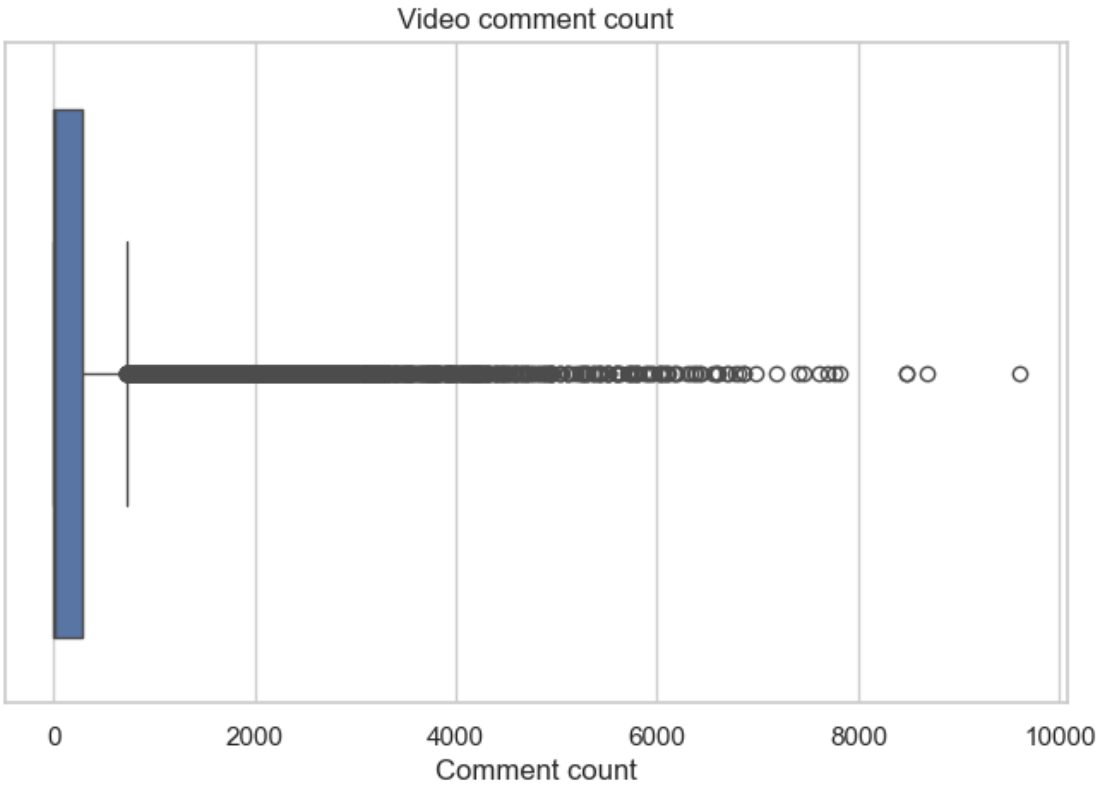
[27]:
```
# Create a histogram
plt.figure(figsize=(10,5))
sns.histplot(data['video_like_count'], bins=range(0,26,1))
plt.title('Video like count histogram');
```



Question: What do you notice about the distribution of this variable? **Response:** Similar to view count, there are far more videos with $< 100{,}000$ likes than there are videos with more. However, in this case, there is more of a taper, as the data skews right, with many videos at the upper extremity of like count.

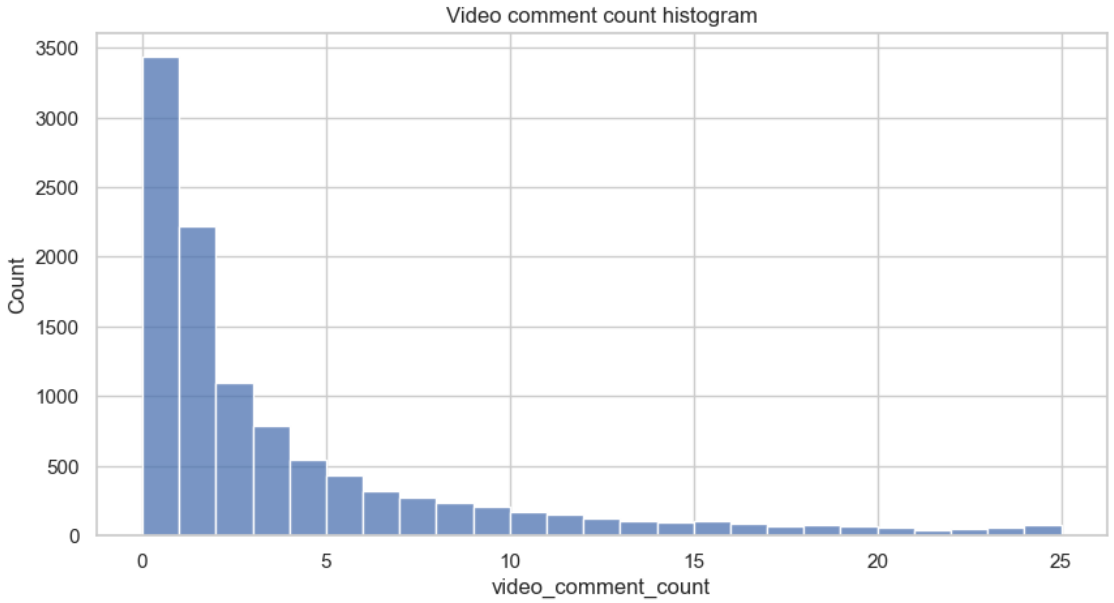
video_comment_count Create a box plot to examine the spread of values in the `video_comment_count` column.

[29]:
```
# Create a boxplot to visualize distribution of `video_comment_count`

sns.set(style="whitegrid")  # Set seaborn style

plt.figure(figsize=(8, 5))  # Set figure size
box = sns.boxplot(x=data['video_comment_count'])  # Create boxplot

g = plt.gca()  # Get current axes
plt.xlabel('Comment count')
plt.title('Video comment count')

plt.show()  # Display plot
```



Create a histogram of the values in the `video_comment_count` column to further explore the distribution of this variable.

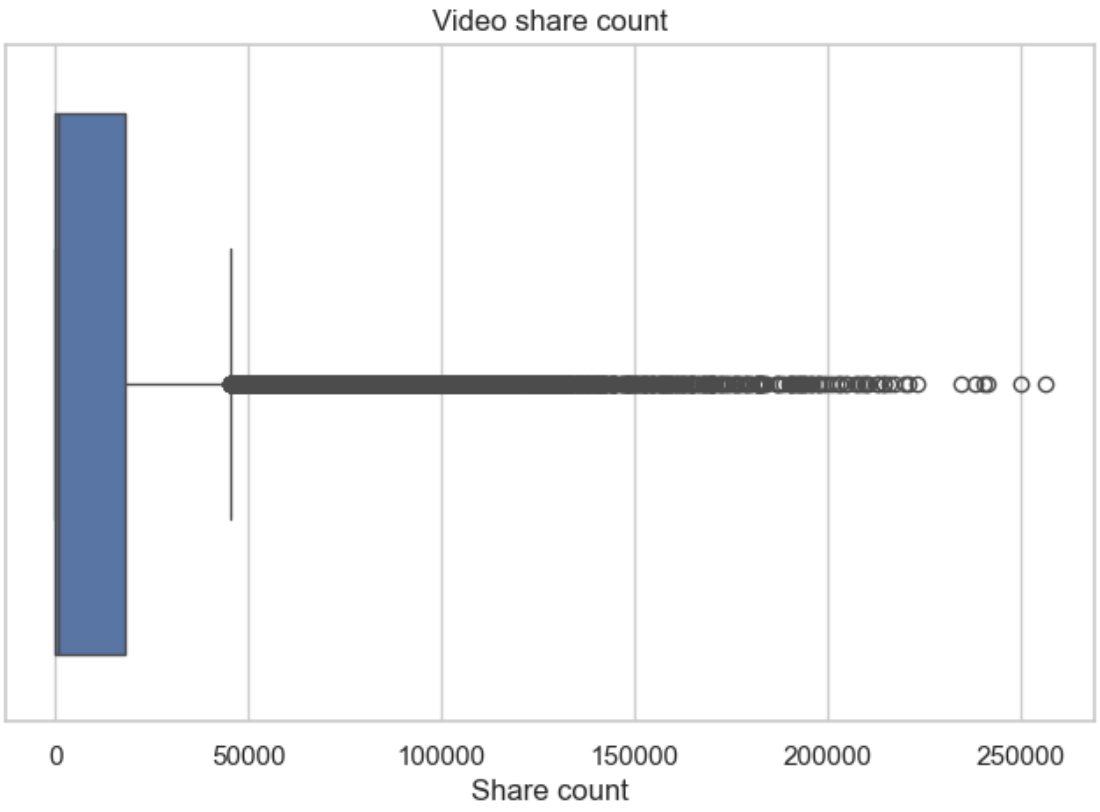
[31]:
```
# Create a histogram
plt.figure(figsize=(10,5))
sns.histplot(data['video_comment_count'], bins=range(0,26,1))
plt.title('Video comment count histogram');
```



Question: What do you notice about the distribution of this variable? **Response:** Again, the vast majority of videos are grouped at the bottom of the range of values for video comment count. Most videos have fewer than 100 comments. The distribution is very right-skewed.

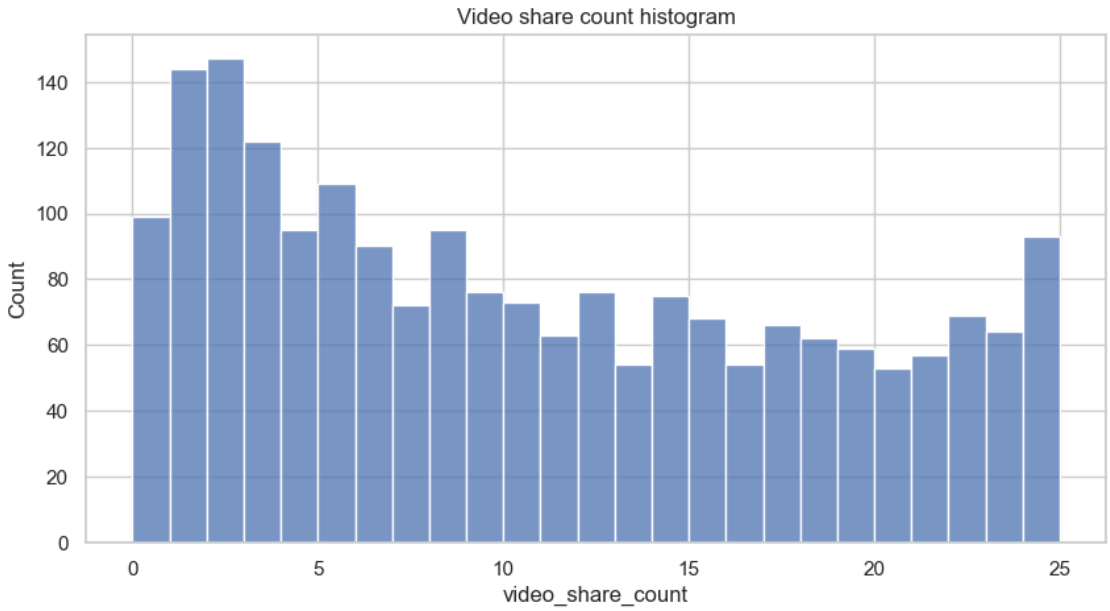
video_share_count Create a box plot to examine the spread of values in the `video_share_count` column.

[33]:
```
# Create a boxplot to visualize distribution of `video_share_count`

sns.set(style="whitegrid")  # Set seaborn style

plt.figure(figsize=(8, 5))  # Set figure size
box = sns.boxplot(x=data['video_share_count'])  # Create boxplot

g = plt.gca()  # Get current axes
plt.xlabel('Share count')
plt.title('Video share count')

plt.show()  # Display plot
```



Create a histogram of the values in the `video_share_count` column to further explore the distribution of this variable.

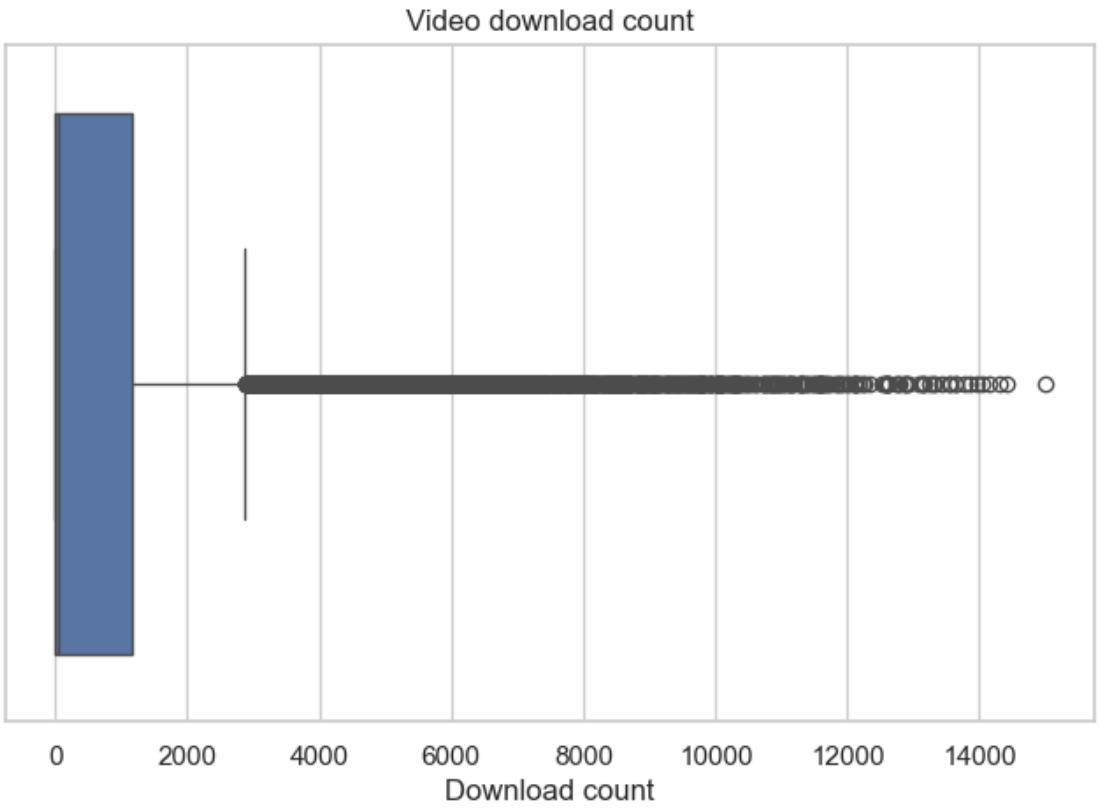
[35]:
```
# Create a histogram
plt.figure(figsize=(10,5))
sns.histplot(data['video_share_count'], bins=range(0,26,1))
plt.title('Video share count histogram');
```



Question: What do you notice about the distribution of this variable? **Response:** The overwhelming majority of videos had fewer than 10,000 shares. The distribution is very skewed to the right.

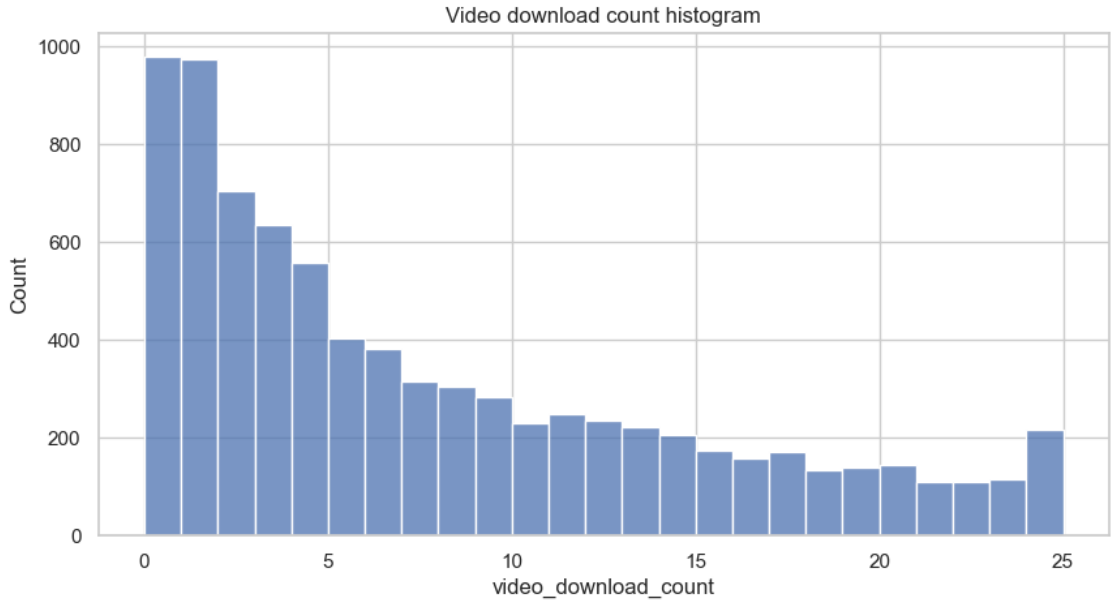
video_download_count Create a box plot to examine the spread of values in the `video_download_count` column.

[37]:
```
# Create a boxplot to visualize distribution of `video_download_count`

sns.set(style="whitegrid")  # Set seaborn style

plt.figure(figsize=(8, 5))  # Set figure size
box = sns.boxplot(x=data['video_download_count'])  # Create boxplot

g = plt.gca()  # Get current axes
plt.xlabel('Download count')
plt.title('Video download count')

plt.show()  # Display plot
```



Create a histogram of the values in the `video_download_count` column to further explore the distribution of this variable.

[39]:
```
# Create a histogram
plt.figure(figsize=(10,5))
sns.histplot(data['video_download_count'], bins=range(0,26,1))
plt.title('Video download count histogram');
```

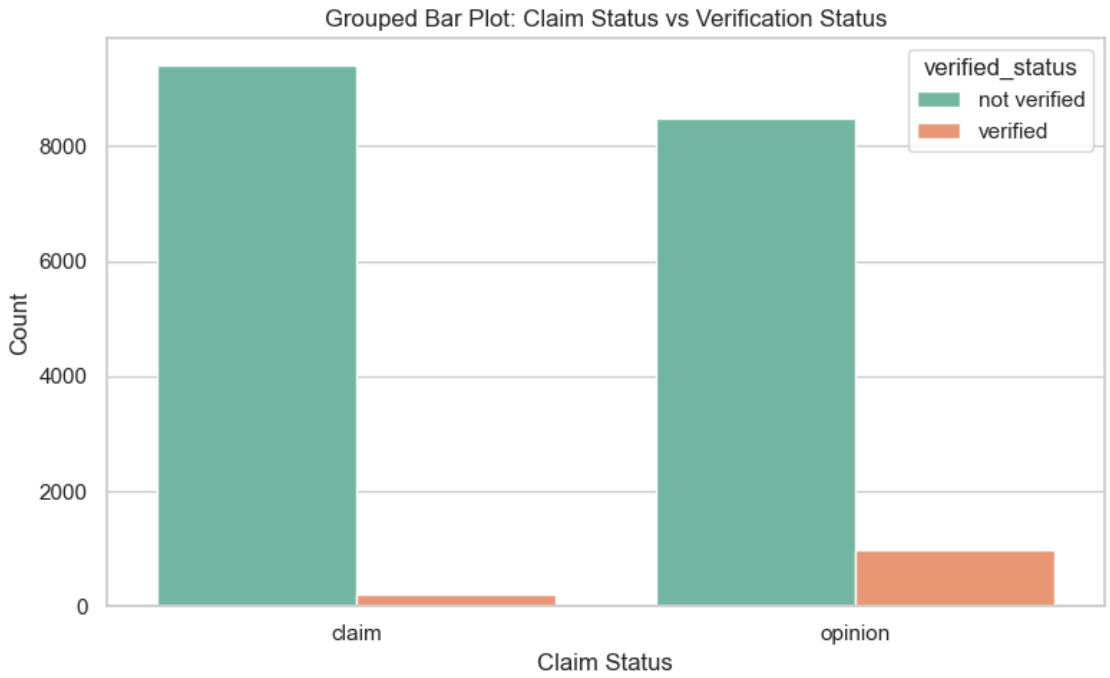


Question: What do you notice about the distribution of this variable? **Response:** The majority of videos were downloaded fewer than 500 times, but some were downloaded over 12,000 times. Again, the data is very skewed to the right.

Claim status by verification status Now, create a histogram with four bars: one for each combination of claim status and verification status.

[41]:
```
# Create a histogram

# Count combinations
counts = data.groupby(['claim_status', 'verified_status']).size().
 ↪reset_index(name='count')

# Plot using seaborn
plt.figure(figsize=(8, 5))
sns.barplot(
    data=counts,
    x='claim_status',
    y='count',
    hue='verified_status',
    palette='Set2'
)

plt.xlabel('Claim Status')
plt.ylabel('Count')
plt.title('Grouped Bar Plot: Claim Status vs Verification Status')
plt.tight_layout()
```

```
plt.show()
```

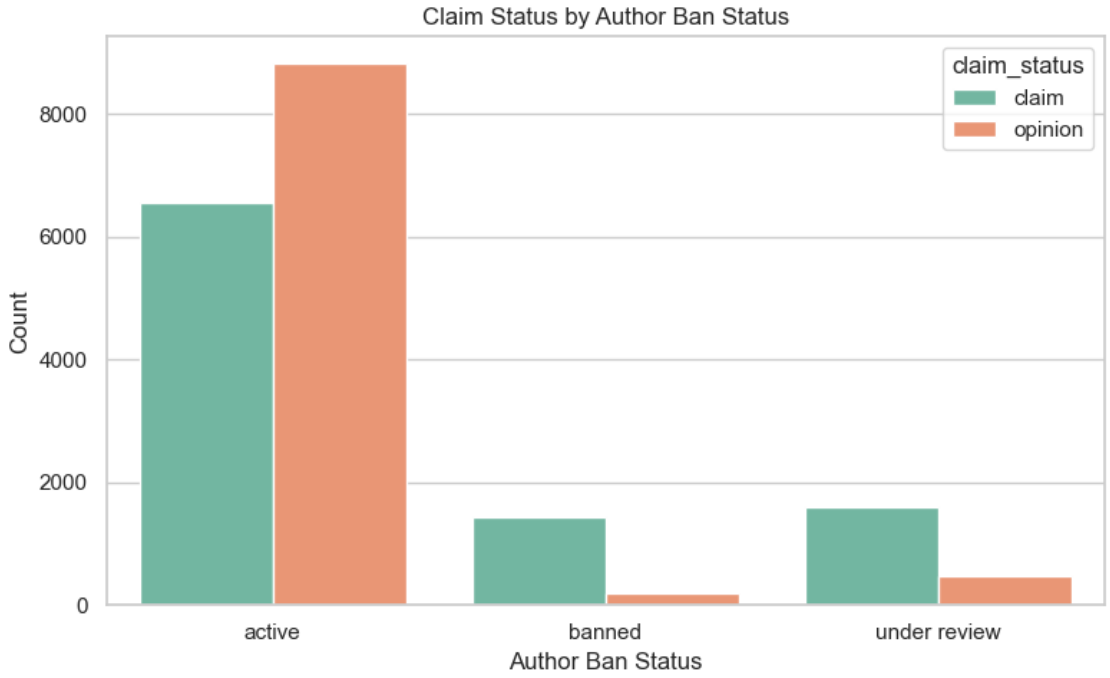


Question: What do you notice about the number of verified users compared to unverified? And how does that affect their likelihood to post opinions? **Response:** There are far fewer verified users than unverified users, but if a user *is* verified, they are much more likely to post opinions.

Claim status by author ban status The previous course used a `groupby()` statement to examine the count of each claim status for each author ban status. Now, use a histogram to communicate the same information.

[43]:
```
# Create a histogram

# Count combinations
counts = data.groupby(['author_ban_status', 'claim_status']).size().
 ↪reset_index(name='count')

# Plot using seaborn
plt.figure(figsize=(8, 5))
sns.barplot(
    data=counts,
    x='author_ban_status',
    y='count',
    hue='claim_status',
    palette='Set2'
)
```

```
plt.xlabel('Author Ban Status')
plt.ylabel('Count')
plt.title('Claim Status by Author Ban Status')
plt.tight_layout()
plt.show()
```

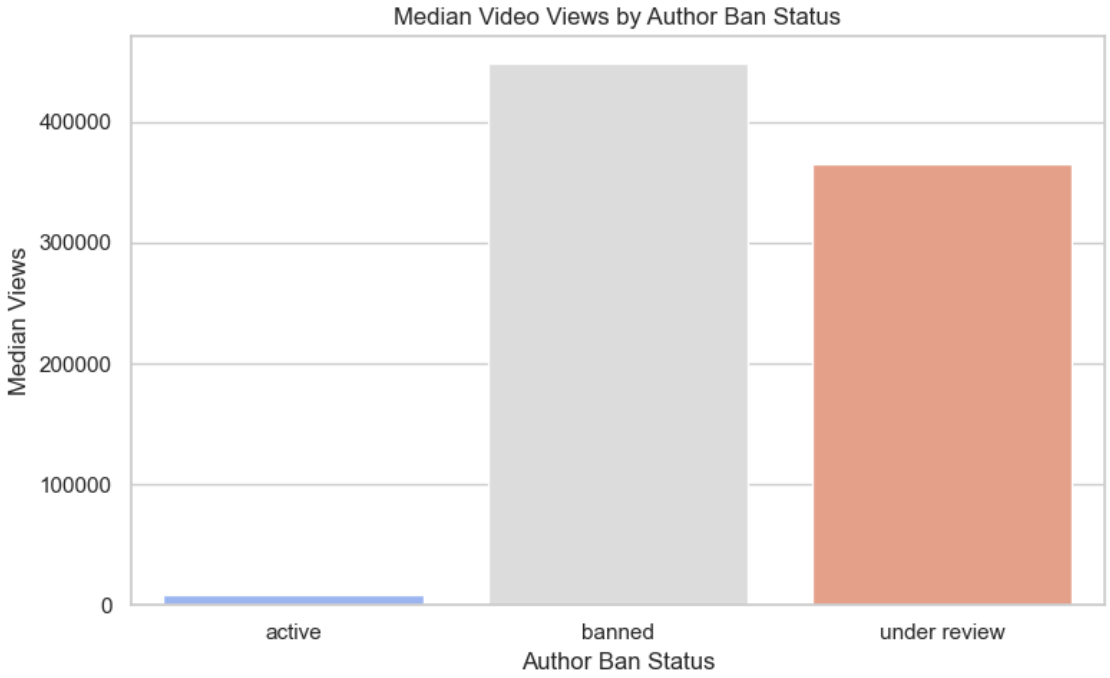


Question: What do you notice about the number of active authors compared to banned authors for both claims and opinions? **Response:** For both claims and opinions, there are many more active authors than banned authors or authors under review; however, the proportion of active authors is far greater for opinion videos than for claim videos. Again, it seems that authors who post claim videos are more likely to come under review and/or get banned.

Median view counts by ban status Create a bar plot with three bars: one for each author ban status. The height of each bar should correspond with the median number of views for all videos with that author ban status.

[53]:
```
# Create a bar plot

# Calculate median views for each author ban status
median_views = data.groupby('author_ban_status')['video_view_count'].median().
  ↪reset_index()

# Plot
plt.figure(figsize=(8, 5))
```

```
sns.barplot(
    data=median_views,
    x='author_ban_status',
    y='video_view_count',
    hue='author_ban_status',
    legend=False,
    palette='coolwarm'
)

plt.xlabel('Author Ban Status')
plt.ylabel('Median Views')
plt.title('Median Video Views by Author Ban Status')
plt.tight_layout()
plt.show()
```



Question: What do you notice about the median view counts for non-active authors compared to that of active authors? Based on that insight, what variable might be a good indicator of claim status? **Response:** The median view counts for non-active authors are many times greater than the median view count for active authors. Since you know that non-active authors are more likely to post claims, and that videos by non-active authors get far more views on aggregate than videos by active authors, then `video_view_count` might be a good indicator of claim status. Indeed, a quick check of the median view count by claim status bears out this assessment:

[55]:
```
# Calculate the median view count for claim status.
```

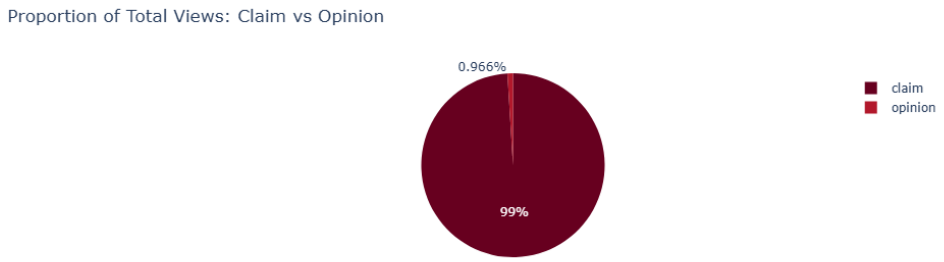
```
median_views_by_claim = data.groupby('claim_status')['video_view_count'].
↪median().reset_index()
```

[57]:
```
print(median_views_by_claim)
```

```
  claim_status  video_view_count
0        claim          501555.0
1      opinion            4953.0
```

Total views by claim status Create a pie graph that depicts the proportions of total views for claim videos and total views for opinion videos.

[61]:
```
# Create a pie graph

import plotly.express as px

# Group and sum views by claim status
views_sum = data.groupby('claim_status')['video_view_count'].sum().reset_index()

# Create interactive pie chart
fig = px.pie(
    views_sum,
    names='claim_status',
    values='video_view_count',
    title='Proportion of Total Views: Claim vs Opinion',
    color_discrete_sequence=px.colors.sequential.RdBu
)

# Show the chart
fig.show()
```



Question: What do you notice about the overall view count for claim status? **Response:** The overall view count is dominated by claim videos even though there are roughly the same number of each video in the dataset.

2.0.5 Task 4. Determine outliers

When building predictive models, the presence of outliers can be problematic. For example, if you were trying to predict the view count of a particular video, videos with extremely high view counts might introduce bias to a model. Also, some outliers might indicate problems with how data was captured or recorded.

The ultimate objective of the TikTok project is to build a model that predicts whether a video is a claim or opinion. The analysis you've performed indicates that a video's engagement level is strongly correlated with its claim status. There's no reason to believe that any of the values in the TikTok data are erroneously captured, and they align with expectation of how social media works: a very small proportion of videos get super high engagement levels. That's the nature of viral content.

Nonetheless, it's good practice to get a sense of just how many of your data points could be considered outliers. The definition of an outlier can change based on the details of your project, and it helps to have domain expertise to decide a threshold. You've learned that a common way to determine outliers in a normal distribution is to calculate the interquartile range (IQR) and set a threshold that is 1.5 * IQR above the 3rd quartile.

In this TikTok dataset, the values for the count variables are not normally distributed. They are heavily skewed to the right. One way of modifying the outlier threshold is by calculating the **median** value for each variable and then adding 1.5 * IQR. This results in a threshold that is, in this case, much lower than it would be if you used the 3rd quartile.

Write a for loop that iterates over the column names of each count variable. For each iteration: 1. Calculate the IQR of the column 2. Calculate the median of the column 3. Calculate the outlier threshold (median + 1.5 * IQR) 4. Calculate the numer of videos with a count in that column that exceeds the outlier threshold 5. Print "Number of outliers, {column name}: {outlier count}"

```
Example:
Number of outliers, video_view_count: ___
Number of outliers, video_like_count: ___
Number of outliers, video_share_count: ___
Number of outliers, video_download_count: ___
Number of outliers, video_comment_count: ___
```

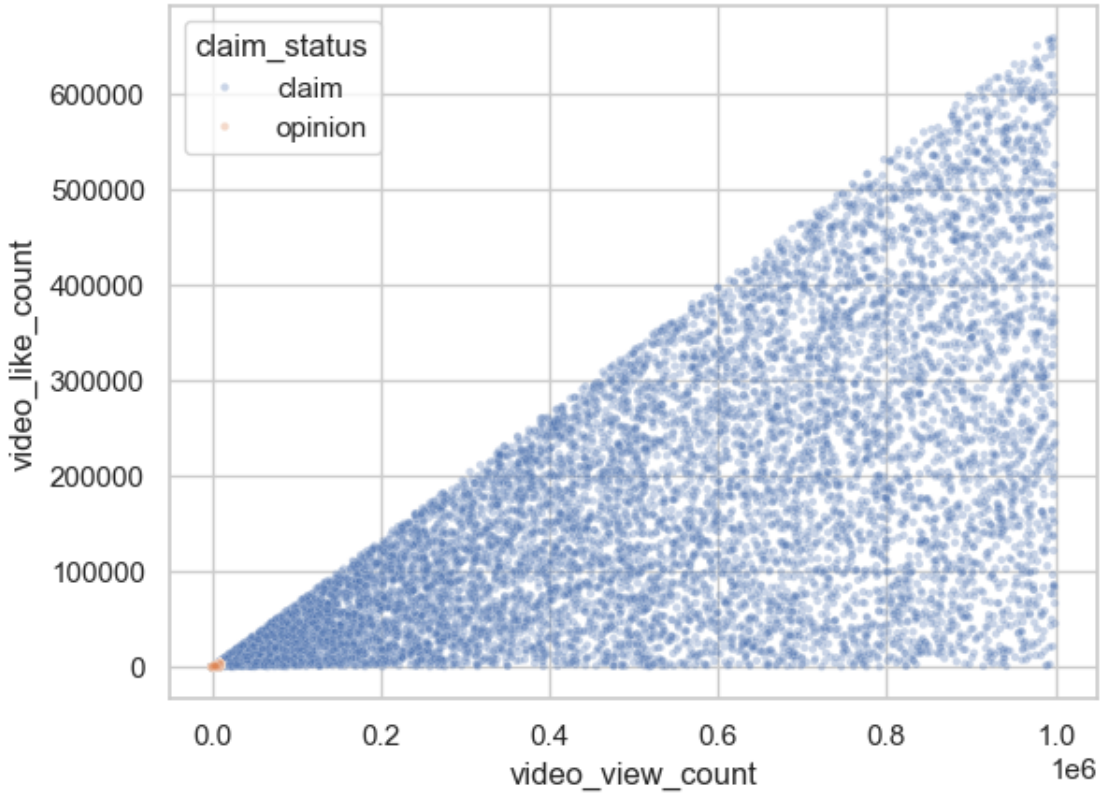
[64]:
```
count_cols = ['video_view_count',
              'video_like_count',
              'video_share_count',
              'video_download_count',
              'video_comment_count',
              ]

for column in count_cols:
    q1 = data[column].quantile(0.25)
    q3 = data[column].quantile(0.75)
    iqr = q3 - q1
    median = data[column].median()
    outlier_threshold = median + 1.5*iqr
```

```
    # Count the number of values that exceed the outlier threshold
    outlier_count = (data[column] > outlier_threshold).sum()
    print(f'Number of outliers, {column}:', outlier_count)
```

```
Number of outliers, video_view_count: 2343
Number of outliers, video_like_count: 3468
Number of outliers, video_share_count: 3732
Number of outliers, video_download_count: 3733
Number of outliers, video_comment_count: 3882
```

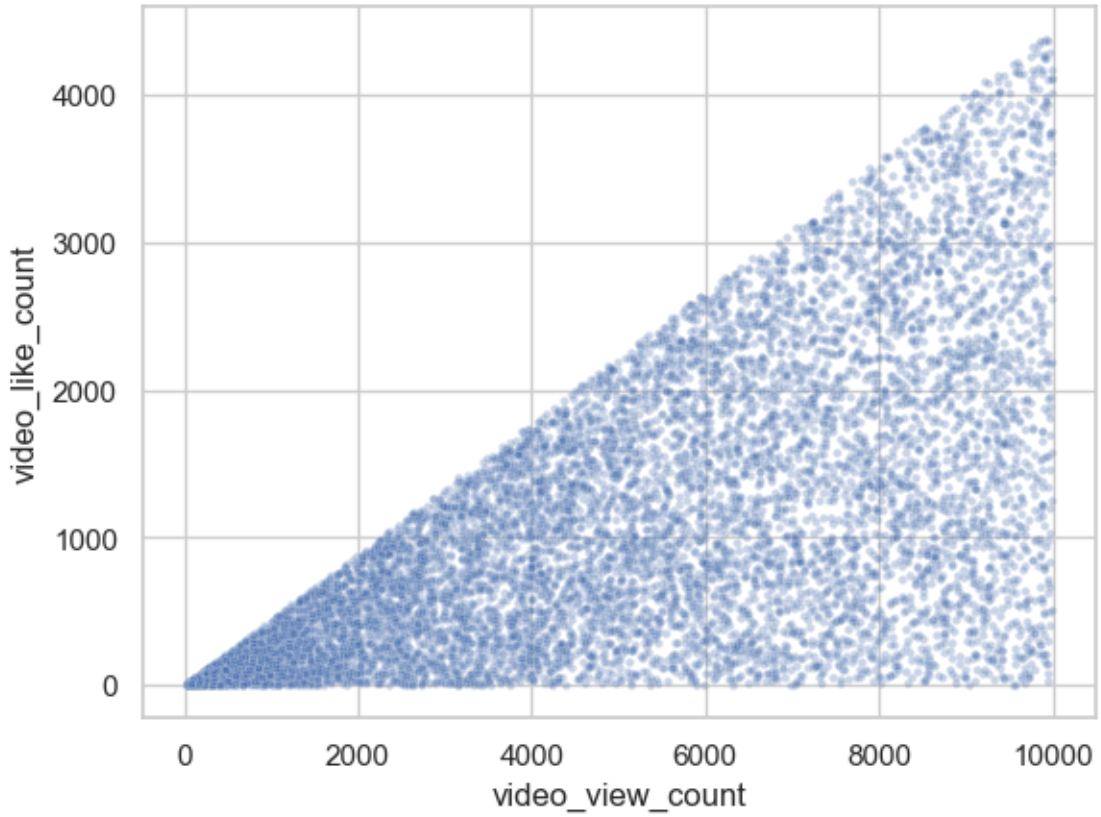
Scatterplot

[66]:
```
# Create a scatterplot of `video_view_count` versus `video_like_count`␣
 ↪according to 'claim_status'
sns.scatterplot(x=data["video_view_count"], y=data["video_like_count"],
                hue=data["claim_status"], s=10, alpha=.3)
plt.show()
```



[68]:
```
# Create a scatterplot of ``video_view_count` versus `video_like_count` for␣
 ↪opinions only
opinion = data[data['claim_status']=='opinion']
```

```
sns.scatterplot(x=opinion["video_view_count"], y=opinion["video_like_count"],
                s=10, alpha=.3)
plt.show()
```



2.0.6 Task 5a. Results and evaluation

Having built visualizations in Python, what have you learned about the dataset? What other questions have your visualizations uncovered that you should pursue?

Pro tip: Put yourself in your client's perspective, what would they want to know?

Use the following code cells to pursue any additional EDA. Also use the space to make sure your visualizations are clean, easily understandable, and accessible.

Ask yourself: Did you consider color, contrast, emphasis, and labeling?

Response:

What have I learned? *I examined the data distribution/spread, count frequencies, mean and median values, extreme values/outliers, missing data, and more. I analyzed correlations between variables, particularly between the claim_status variable and others.*

My questions: *I want to further investigate distinctive characteristics that apply only to claims or only to opinions. Also, I want to consider other variables that might be helpful in understanding*

the data.

My client would likely want to know: *My client would want to know the assumptions regarding what data might be predictive of claim_status.*

2.0.7 Task 5b. Conclusion

Make it professional and presentable

You have visualized the data you need to share with the director now. Remember, the goal of a data visualization is for an audience member to glean the information on the chart in mere seconds.

Questions to ask yourself for reflection: Why is it important to conduct Exploratory Data Analysis? What other visuals could you create?

EDA is important because: *EDA helps a data professional to get to know the data, understand its outliers, clean its missing values, and prepare it for future modeling.*

Visualizations helped me understand: *That we will need to make decisions on certain considerations prior to designing a model. (for example, what to do with outliers, duplicate values, or missing data)*