Activity_Course 3 TikTok project lab

December 16, 2023

1 TikTok Project

Course 3 - Go Beyond the Numbers: Translate Data into Insights

Your TikTok data team is still in the early stages of their latest project. So far, you've completed a project proposal and used Python to inspect and organize the TikTok dataset.

Orion Rainier, a Data Scientist at TikTok, is pleased with the work you have already completed and is requesting your assistance with some Exploratory Data Analysis (EDA) and data visualization. The management team asked to see a Python notebook showing data structuring and cleaning, as well as any matplotlib/seaborn visualizations plotted to help us understand the data. At the very least, include a graph comparing claim counts to opinion counts, as well as boxplots of the most important variables (like "video duration," "video like count," "video comment count," and "video view count") to check for outliers. Also, include a breakdown of "author ban status" counts.

Additionally, the management team has recently asked all EDA to include Tableau visualizations. Tableau visualizations are particularly helpful in status reports to the client and board members. For this data, create a Tableau dashboard showing a simple claims versus opinions count, as well as stacked bar charts of claims versus opinions for variables like video view counts, video like counts, video share counts, and video download counts. Make sure it is easy to understand to someone who isn't data savvy, and remember that the assistant director is a person with visual impairments.

You also notice a follow-up email from the Data Science Lead, Willow Jaffey. Willow suggests including an executive summary of your analysis to share with teammates.

A notebook was structured and prepared to help you in this project. Please complete the following questions.

2 Course 3 End-of-course project: Exploratory data analysis

In this activity, you will examine data provided and prepare it for analysis. You will also design a professional data visualization that tells a story, and will help data-driven decisions for business needs.

Please note that the Tableau visualization activity is optional, and will not affect your completion of the course. Completing the Tableau activity will help you practice planning out and plotting a data visualization based on a specific business need. The structure of this activity is designed to emulate the proposals you will likely be assigned in your career as a data professional. Completing this activity will help prepare you for those career moments.

The purpose of this project is to conduct exploratory data analysis on a provided data set. Your mission is to continue the investigation you began in C2 and 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

Follow the instructions and answer the question below to complete the activity. Then, you will complete an executive summary using the questions listed on the PACE Strategy Document.

Be sure to complete this activity before moving on. The next course item will provide you with a completed exemplar to compare to your own work.

3 Visualize a story in Tableau and Python

4 PACE stages

Throughout these project notebooks, you'll see references to the problem-solving framework PACE. The following notebook components are labeled with the respective PACE stage: Plan, Analyze, Construct, and Execute.

4.1 PACE: Plan

Consider the questions in your PACE Strategy Document 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?

The best way to identify outliers: 1. Understand all the variables within the dataset 2. Use python built in functions like info and describe to see the min,max,average and other mathematical details that could be of value to see if there are any outstanding values that should not be there. 3. Create visualizations such as box plots to see if there are any outliers within the dataset, and if so we can dive deeper into them to see if they need to be deleted or not.

4.1.1 Task 1. Imports, links, and loading

Go to Tableau Public The following link will help you complete this activity. Keep Tableau Public open as you proceed to the next steps.

Link to supporting materials: Public Tableau: https://public.tableau.com/s/. Note that the TikTok dataset can be downloaded directly from this notebook by going to "Lab Files" in the menu bar at the top of the page, clicking into the "/home/jovyan/work" folder, selecting `tiktok_dataset.csv`, and clicking "Download" above the list of files.

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
### YOUR CODE HERE ###
import pandas as pd
import numpy as np

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

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

Note: As shown in this cell, the dataset has been automatically loaded in for you. You do not need to download the .csv file, or provide more code, in order to access the dataset and proceed with this lab. Please continue with this activity by completing the following instructions.

[2]:
```
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

4.2 PACE: Analyze

Consider the questions in your PACE Strategy Document and those below where applicable to complete your code.

4.2.1 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`.

[3]:
```
# Display and examine the first few rows of the dataframe
### YOUR CODE HERE ###
data.head(10)
```

[3]:

	#	claim_status	video_id	video_duration_sec	video_transcription_text	verified_status	author_ban_status	video_view_count	video_like_count	video_share_count	video_download_count	video_comment_count
0	1	claim	7017666017	59	someone shared with me that drone deliveries a…	not verified	under review	343296.0	19425.0	241.0	1.0	0.0
1	2	claim	4014381136	32	someone shared with me that there are more mic…	not verified	active	140877.0	77355.0	19034.0	1161.0	684.0
2	3	claim	9859838091	31	someone shared with me that american industria…	not verified	active	902185.0	97690.0	2858.0	833.0	329.0
3	4	claim	1866847991	25	someone shared with me that the metro of st. p…	not verified	active	437506.0	239954.0	34812.0	1234.0	584.0
4	5	claim	7105231098	19	someone shared with me that the number of busi…	not verified	active	56167.0	34987.0	4110.0		
5	6	claim	8972200955	35	someone shared with me that gross domestic pro…	not verified	under review	336647.0	175546.0	62303.0		
6	7	claim	4958886992	16	someone shared with me that elvis presley has …	not verified	active	750345.0	486192.0	193911.0		
7	8	claim	2270982263	41	someone shared with me that the best selling s…	not verified	active	547532.0	1072.0	50.0		
8	9	claim	5235769692	50	someone shared with me that about half of the …	not verified	active	24819.0	10160.0	1050.0		
9	10	claim	4660861094	45	someone shared with me that it would take a 50…	verified	active	931587.0	171051.0	67739.0		

```
4                547.0                152.0
5               4293.0               1857.0
6               8616.0               5446.0
7                 22.0                 11.0
8                 53.0                 27.0
9               4104.0               2540.0
```

[4]:
```
# Get the size of the data
### YOUR CODE HERE ###
data.size
```

[4]:
```
232584
```

[5]:
```
# Get the shape of the data
### YOUR CODE HERE ###
data.shape
```

[5]:
```
(19382, 12)
```

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

[6]:
```
# Get basic information about the data
### YOUR CODE HERE ###
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()`.

[7]:
```
# Generate a table of descriptive statistics
### YOUR CODE HERE ###
data.describe()
```

[7]:

	#	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

4.2.2 Task 2b. Assess data types

In Tableau, staying on the data source page, double check the data types of the columns in the dataset. Refer to the dimensions and measures in Tableau.

Review the instructions linked in the previous Activity document to create the required Tableau visualization.

4.2.3 Task 2c. 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

The data visualizations that will be most helpful for this dataset within the EDA phase would be: **box plots** for looking at any outliers as well as **histograms** to look at the distribution of variables. To distinguish between the percentage of videos that are claims vs opinions we can show a **pie chart** that will distinguish the percent difference.

4.3 PACE: Construct

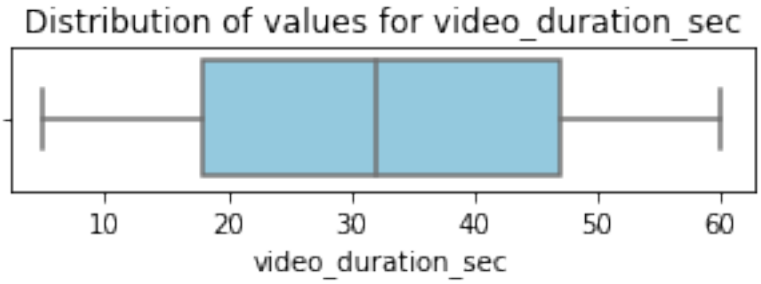
Consider the questions in your PACE Strategy Document to reflect on the Construct stage.

4.3.1 Task 3. Build visualizations

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

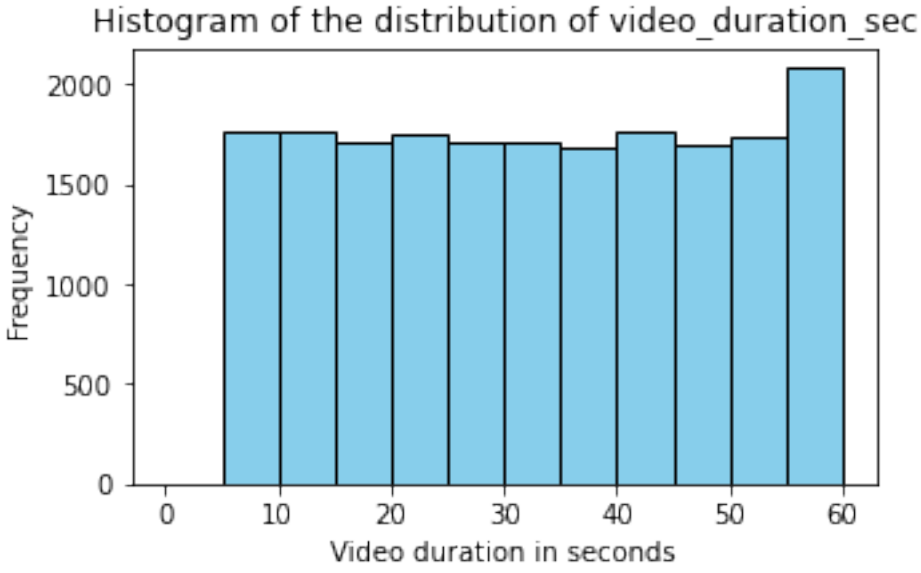
video_duration_sec Create a box plot to examine the spread of values in the `video_duration_sec` column.

[11]:
```
# Create a boxplot to visualize distribution of `video_duration_sec`
### YOUR CODE HERE ###
# lets make the size of the box plot
plt.figure(figsize=(5,1))
# the labels for the x axis as well as the title for the chart
plt.xlabel('Video duration in seconds')
plt.title('Distribution of values for video_duration_sec')
# Now we want to make a box plot of the values in the variable for video␣
 ↪duration
sns.boxplot(x =data['video_duration_sec'],color='skyblue')
plt.show()
```



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

[12]:
```
# Create a histogram
### YOUR CODE HERE ###
# determine the size of the graph
plt.figure(figsize= (5,3))
# once again we want the video duration and since tha values are between 5 and␣
 ↪60 we will use bins tell us that range of 0-60 with a bin width of 5.
plt.
 ↪hist(data['video_duration_sec'],bins=range(0,61,5),color='skyblue',edgecolor='black')
plt.xlabel('Video duration in seconds')
plt.ylabel('Frequency')
plt.title('Histogram of the distribution of video_duration_sec')
plt.show()
```

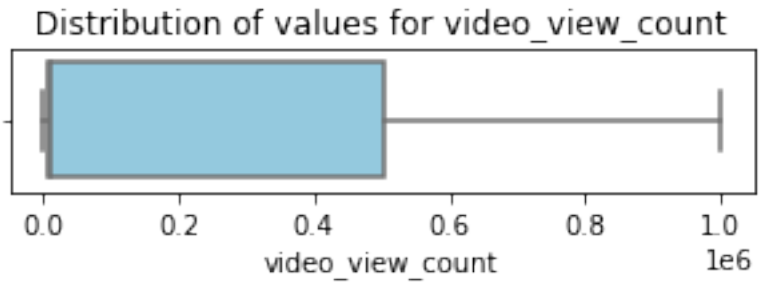


Question: What do you notice about the duration and distribution of the videos?

All the videos in the dataset are between the length of 5 and 60 seconds. Also the distribution of the values is **uniform** as there are no outstanding values and the count is similar across video durations.

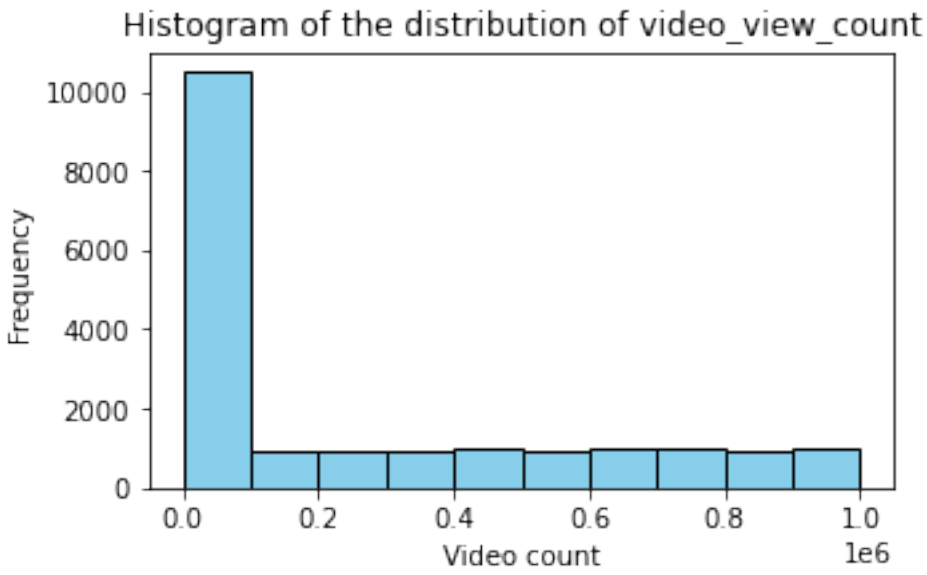
video_view_count Create a box plot to examine the spread of values in the `video_view_count` column.

[14]:
```
# Create a boxplot to visualize distribution of `video_view_count`
### YOUR CODE HERE ###
plt.figure(figsize=(5,1))
plt.xlabel('video view count')
plt.title('Distribution of values for video_view_count')
sns.boxplot(x=data['video_view_count'],color='skyblue')
plt.show()
```



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

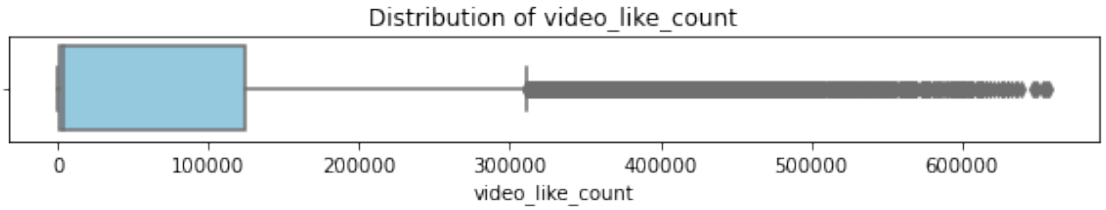
[15]:
```
# Create a histogram
### YOUR CODE HERE ###
plt.figure(figsize= (5,3))
plt.
 ↪hist(data['video_view_count'],bins=range(0,(10**6+1),10**5),color='skyblue',edgecolor='blac
plt.xlabel('Video count')
plt.ylabel('Frequency')
plt.title('Histogram of the distribution of video_view_count')
plt.show()
```



Question: What do you notice about the distribution of this variable? From looking at the plot I am noticing an extremely uneven distribution with more than half the videos having less than 100,000 views. Also the distribution of views greater than 100,000 is **uniform**.

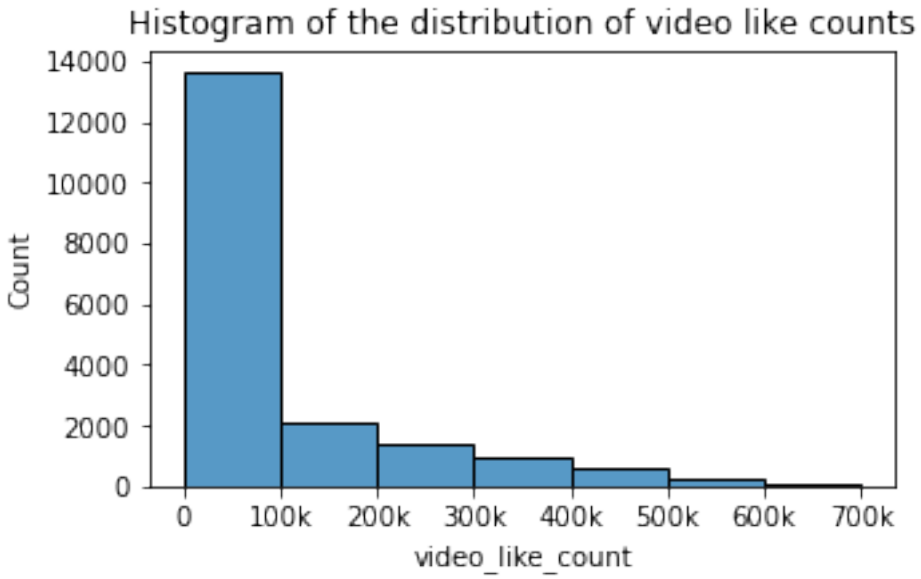
video_like_count Create a box plot to examine the spread of values in the `video_like_count` column.

[17]:
```
# Create a boxplot to visualize distribution of `video_like_count`
### YOUR CODE HERE ###
plt.figure(figsize=(10,1))
plt.title('Distribution of video_like_count')
plt.xlabel('Video likes')
sns.boxplot(x= data['video_like_count'],color='skyblue')
plt.show()
```



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

[18]:
```
# Create a histogram
### YOUR CODE HERE ###
''' below is the code for the histogram where we want to find the distribution␣
 ↪of video
likes and the bins we want to set are between 100k and 701k with it␣
 ↪incrementing
by 100k each bin
'''
plt.figure(figsize=(5,3))
ax= sns.histplot(data =␣
 ↪data['video_like_count'],bins=range(0,(7*10**5+1),10**5))
# Now we want the labels readable for each of the x values on the axis
labels= [0] + [str(i)+ 'k' for i in range(100, 701, 100)]
ax.set_xticks(range(0,7*10**5+1,10**5))
ax.set_xticklabels(labels)
plt.title('Histogram of the distribution of video like counts');
```



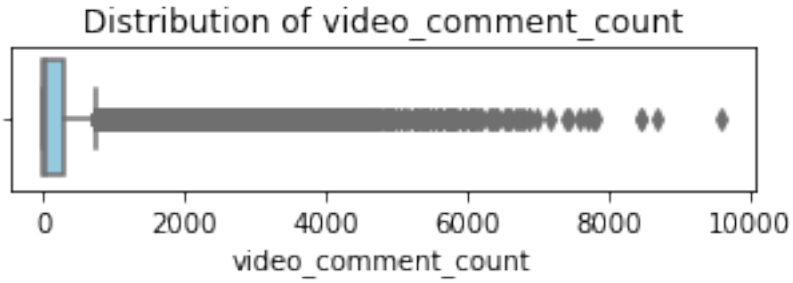
Question: What do you notice about the distribution of this variable? Like the previous video view count there are far more videos with a like count of fewer than 100k, but unlike the previous plot this plot is scewed to the right.

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

[19]:
```
# Create a boxplot to visualize distribution of `video_comment_count`
### YOUR CODE HERE ###
plt.figure(figsize=(5,1))
```

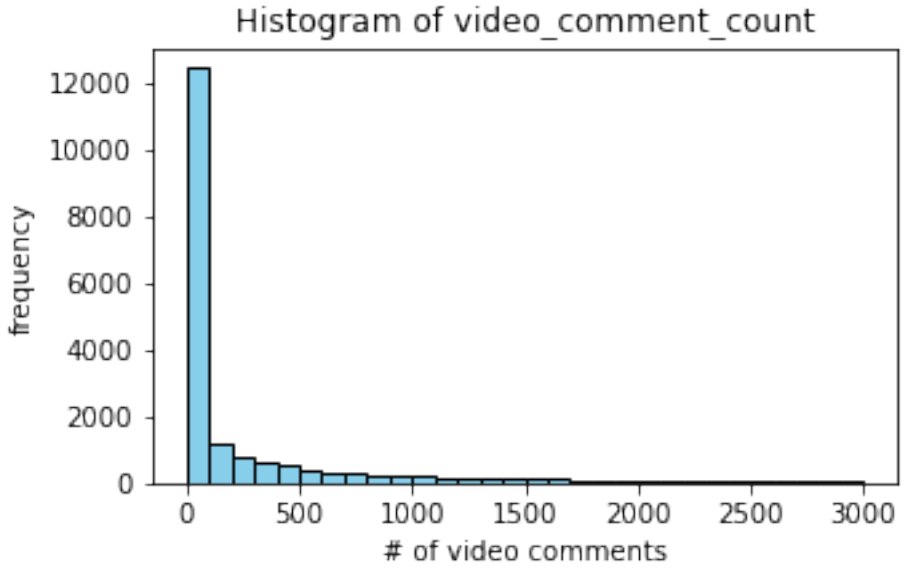
```
sns.boxplot(x=data['video_comment_count'],color = 'skyblue')
plt.title('Distribution of video_comment_count')
```

[19]:
```
Text(0.5, 1.0, 'Distribution of video_comment_count')
```



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

[20]:
```
# Create a histogram
### YOUR CODE HERE ###
plt.figure(figsize=(5,3))
plt.hist(data['video_comment_count'],bins=␣
 ↪range(0,(3001),100),color='skyblue',edgecolor='black')
plt.title('Histogram of video_comment_count')
plt.xlabel('# of video comments')
plt.ylabel('frequency')
plt.show()
```

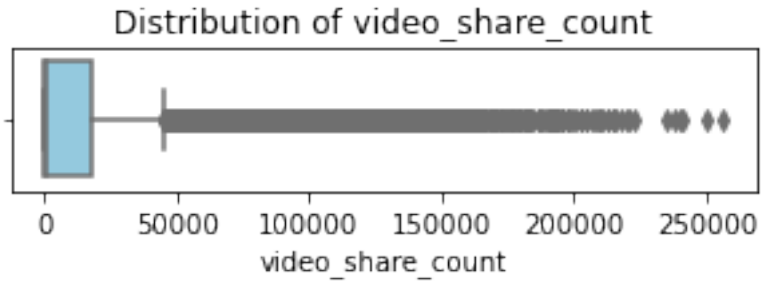


Question: What do you notice about the distribution of this variable? Once again we see most the data is grouped at the bottom of the range so this histogram of video comment count is very **right skewed**.

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

[21]:
```
# Create a boxplot to visualize distribution of `video_share_count`
### YOUR CODE HERE ###
plt.figure(figsize=(5,1))
plt.title('Distribution of video_share_count')
plt.xlabel(' Number of video shares')
sns.boxplot(data['video_share_count'],color='skyblue')
```

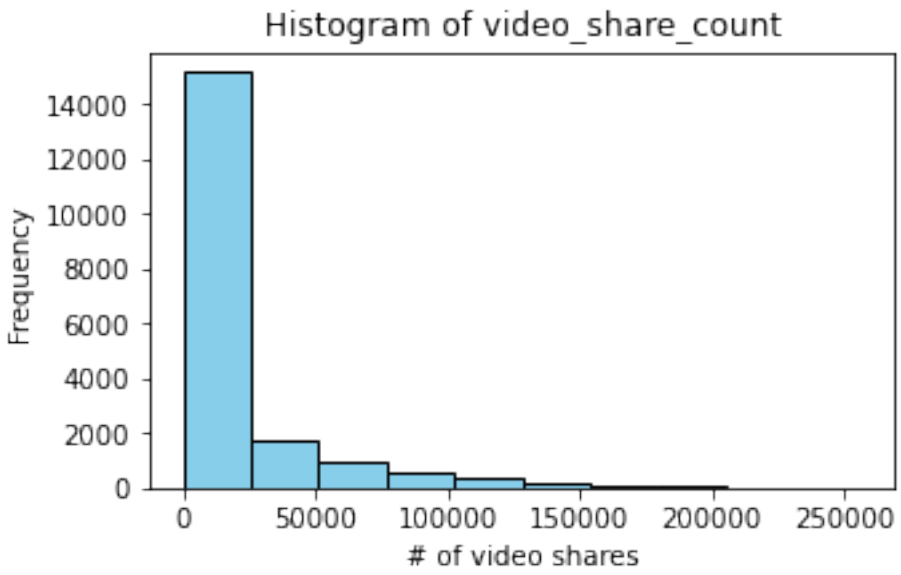
[21]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f74a1853f10>
```



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

[22]:
```
# Create a histogram
### YOUR CODE HERE ###
plt.figure(figsize=(5,3))
plt.hist(data['video_share_count'],color='skyblue',edgecolor='black')
plt.title('Histogram of video_share_count')
plt.xlabel('# of video shares')
plt.ylabel('Frequency')
```

[22]:
```
Text(0, 0.5, 'Frequency')
```

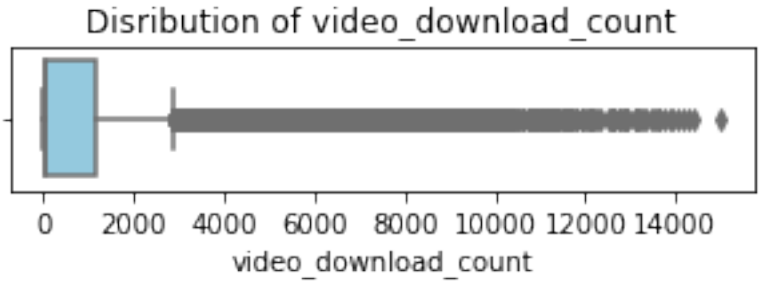


Question: What do you notice about the distribution of this variable? This is a very similar story to the other histograms as the data is very **skewed to the right**. Also, it seems that the majority of videos have **less than 10k shares.**

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

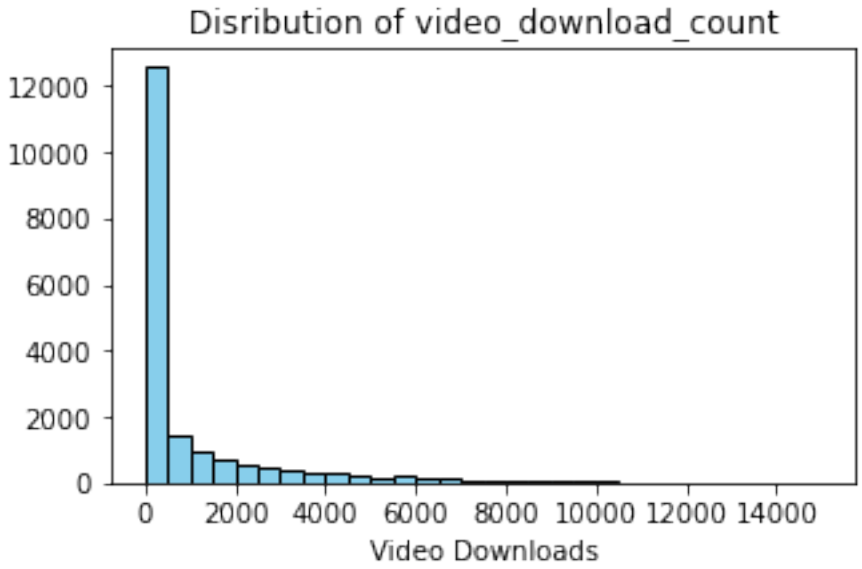
[23]:
```
# Create a boxplot to visualize distribution of `video_download_count`
### YOUR CODE HERE ###
plt.figure(figsize=(5,1))
plt.title('Disribution of video_download_count')
plt.xlabel('Video Downloads')
sns.boxplot(data['video_download_count'],color='skyblue')
```

[23]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f74a16b9e50>
```



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

[24]:
```
# Create a histogram
### YOUR CODE HERE ###
plt.figure(figsize=(5,3))
plt.title('Disribution of video_download_count')
plt.xlabel('Video Downloads')
plt.
 ↪hist(data['video_download_count'],bins=range(0,(15001),500),color='skyblue',edgecolor='blac
plt.show()
```

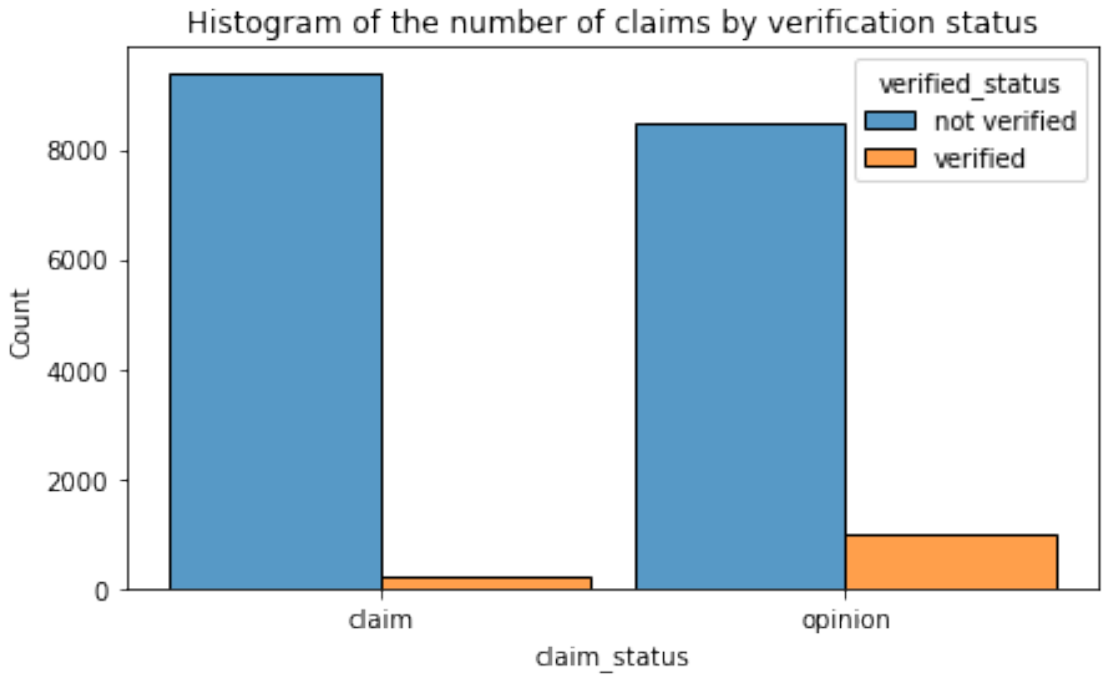


Question: What do you notice about the distribution of this variable? The majority of video downloads occured with videos that were downloaded **fewer than 500 times.** Also the data is very **right skewed**.

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

[25]:
```
# Create a histogram
### YOUR CODE HERE ###
# okay now we are going to create a histogram plot based on categorial variable
plt.figure(figsize=(7,4))
# we have the value set to count the claim status and the categorial variable␣
 ↪to verified_status for comparison
sns.histplot(data=data, x = 'claim_status', hue = 'verified_status',multiple =␣
 ↪'dodge',shrink=0.9)
plt.title('Histogram of the number of claims by verification status')
```

```
plt.show()
```

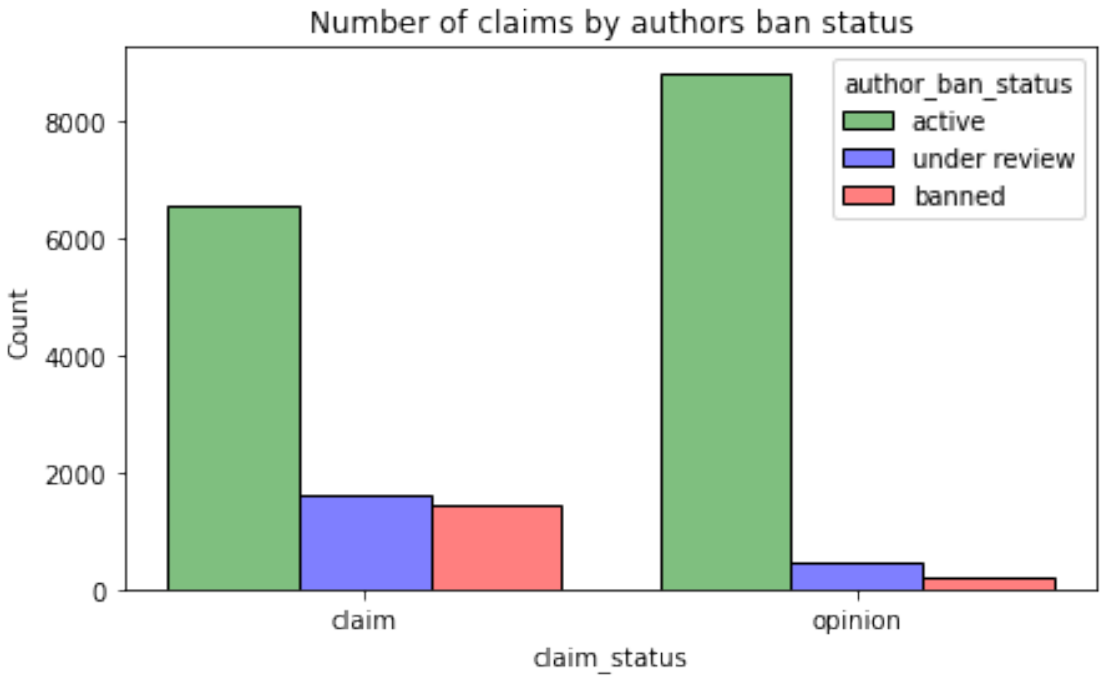


Question: What do you notice about the number of verified users compared to unverified? And how does that affect their likelihood to post opinions? Based on the graph plotted above we see that there are fewer verified users than not verified. But, if a user is verified we see that they are most likely to post an opinion.

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.

[26]:
```
# Create a histogram
### YOUR CODE HERE ###
# Now we will create a histogram for the claim status based on the categories␣
 ↪of the authors ban status
plt.figure(figsize=(7,4))
''' we have our histogram here with the x axis being claim status and the y␣
 ↪axis being the count of each.
Also I specified the order I want the ban status of the categories to be. As␣
 ↪well as the color I wanted them
to be. Lastly, I wanted the colors to not be too dark so changed the alpha and␣
 ↪left some space
between the bars so one can distinguish between claims and opinions.
'''
```

```
sns.histplot(data=data,x='claim_status',
         hue='author_ban_status',
         hue_order = ['active','under review','banned'],
         multiple = 'dodge',
         palette= {'active':'green','under review':'blue','banned':'red'},
         alpha = 0.5,
        shrink = 0.8
        )
plt.title('Number of claims by authors ban status')
plt.show()
```



[ ]:

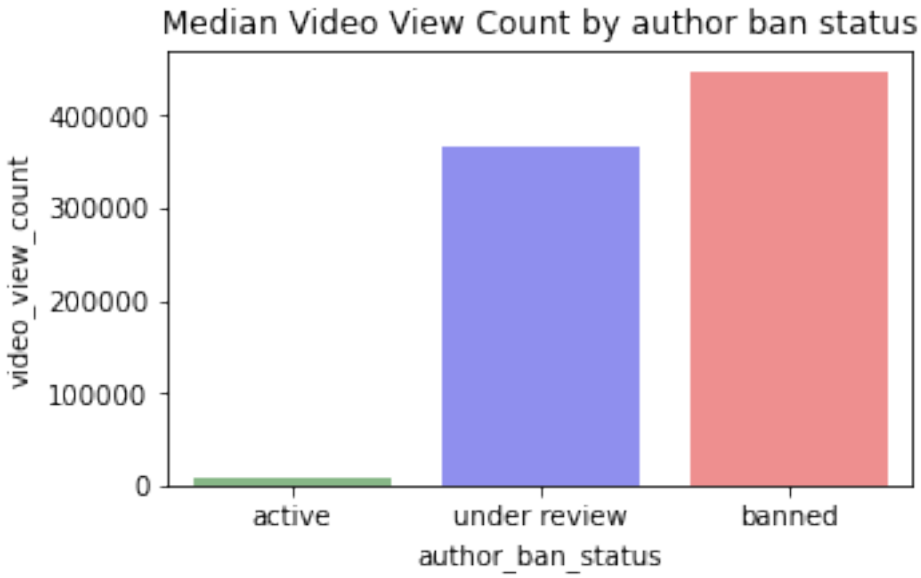
Question: What do you notice about the number of active authors compared to banned authors for both claims and opinions?

We notice that authors how have an active status are far greater count than those who have a under review or banned status. As well as we see that the porportion of active status authors is larger for opions based claim_status than those of opinions. Lastly, authors who are banned or under review are most likely to be claim videos over those of opinion based.

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.

[33]:
```
# Create a bar plot
### YOUR CODE HERE ###
# so first we want to do is group the data by the author ban status and the␣
 ↪median count values for each
author_ban_counts = data.groupby(data['author_ban_status']).median(numeric_only␣
 ↪= True).reset_index()
# Now that we have the data grouped lets create a bar chart using seaborn
fig = plt.figure(figsize=(5,3))
sns.barplot(data = author_ban_counts,
            x='author_ban_status',
            y= 'video_view_count',
            order = ['active','under review','banned'],
            palette= {'active':'green','under review':'blue','banned':'red'},
            alpha = 0.5,
           )
plt.title('Median Video View Count by author ban status')
```

[33]:
```
Text(0.5, 1.0, 'Median Video View Count by author ban status')
```



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? The median view counts are far greater for non active authors than for active authors. Since we know that non active audiences tend to post more claim videos and that they have a higher video view count , we can say that video view might be a good indicator for claim_status.

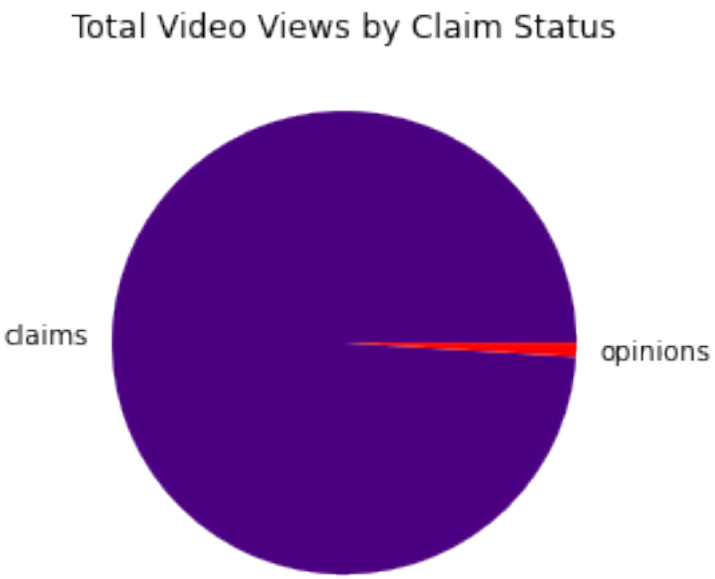
[37]:
```
# Calculate the median view count for claim status.
### YOUR CODE HERE ###
```

```
median_view_counts = data.groupby('claim_status')['video_view_count'].median()
median_view_counts
```

[37]:
```
claim_status
claim      501555.0
opinion      4953.0
Name: video_view_count, dtype: float64
```

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.

[53]:
```
# Create a pie graph
### YOUR CODE HERE ###
fig = plt.figure(figsize=(4,4))
plt.pie(data.groupby('claim_status')['video_view_count'].
 ↪sum(),labels=['claims','opinions'],colors=('indigo','red'))
plt.title('Total Video Views by Claim Status')
plt.show()
```



Question: What do you notice about the overall view count for claim status?

The overall video view count is dominated by claim videos.

4.3.2 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: ___
```

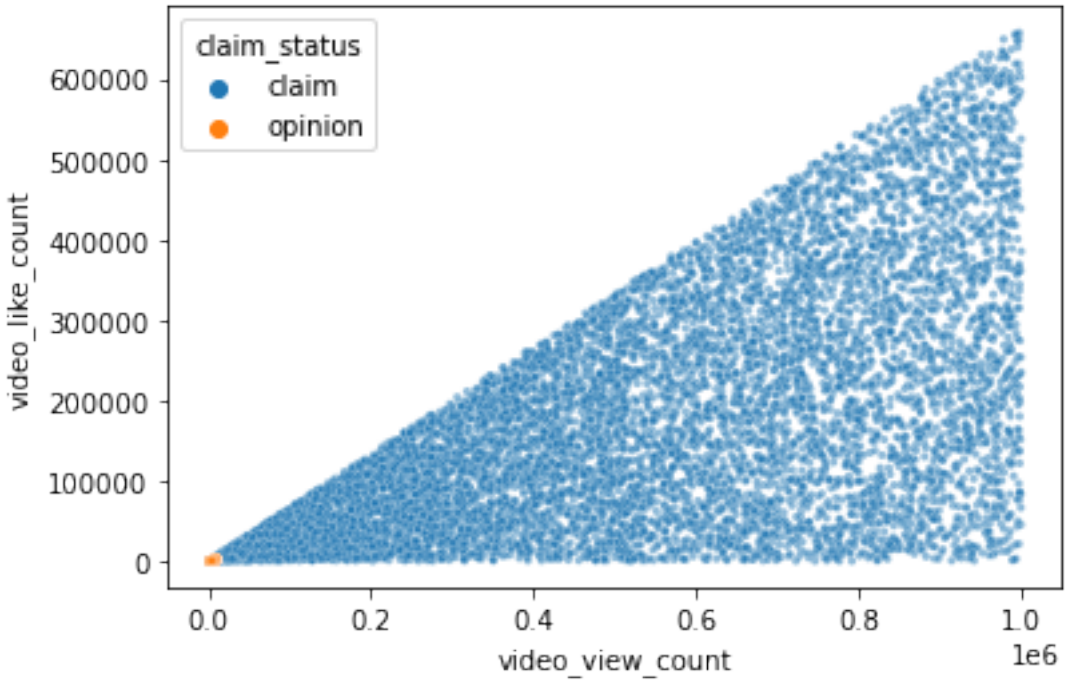
[56]:
```
### YOUR CODE HERE ###
# first we want to create a list of the column names within our dataset
column_names = ['video_view_count',
                'video_like_count',
                'video_share_count',
                'video_download_count',
                'video_comment_count']
# Now that we have a list with the column names we can begin writing our for␣
 ↪loop
for column in column_names:
    # here we want the lower 25 percentile
    q1 = data[column].quantile(0.25)
    # now here we want the upper 75th percentile
```

```
    q3 = data[column].quantile(0.75)
    # to find the inner quartile range we subtract q1 from q3
    iqr = q3 - q1
    # this calculates th median of each column
    median = data[column].median()
    # now we need to get the threshold count of each column
    threshold_outlier = median + 1.5*iqr
    # this will count the number of outliers within each column
    outlier_count = (data[column] > threshold_outlier).sum()
    # now here we just print each columns outliers
    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
```

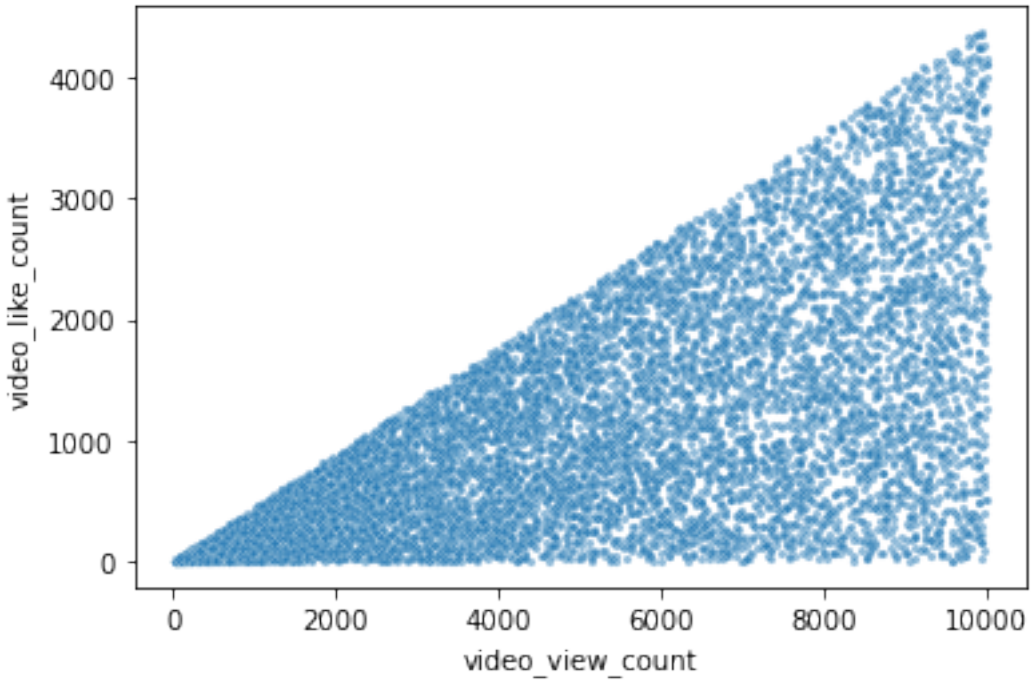
Scatterplot

[60]:
```
# Create a scatterplot of `video_view_count` versus `video_like_count`␣
 ↪according to 'claim_status'
### YOUR CODE HERE ###
sns.scatterplot(x=data['video_view_count'],
                y= data['video_like_count'],
                hue=data['claim_status'],
                s=10,
                alpha=0.5
               )
plt.show()
```



[ ]:
```
### YOUR CODE HERE ###
```

[61]:
```
# Create a scatterplot of ``video_view_count` versus `video_like_count` for␣
 ↪opinions only
### YOUR CODE HERE ###
# first we want to create a mask where we get the data for claim status for␣
 ↪opinions only
opinions= data[data['claim_status']=='opinion']
# Now we can plot the data with only the opinion based claims
sns.scatterplot(x=opinions['video_view_count'],
                y=opinions['video_like_count'],
                s=10,
                alpha=0.5)
plt.show()
```



You can do a scatterplot in Tableau Public as well, which can be easier to manipulate and present. If you'd like step by step instructions, you can review the instructions linked in the previous Activity page.

4.4 PACE: Execute

Consider the questions in your PACE Strategy Document to reflect on the Execute stage.

You've now completed a professional data visualization according to a business need. Well done! Be sure to save your work as a reference for later work in Tableau.

Congratulations! You've completed this lab. However, you may not notice a green check mark next to this item on Coursera's platform. Please continue your progress regardless of the check mark. Just click on the "save" icon at the top of this notebook to ensure your work has been logged.