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.

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.

Visualize a story in Tableau and Python

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/

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

In [16]:
```
# Import packages for data manipulation
import pandas as pd
import numpy as np
```

```python
# Import packages for data visualization
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.

In [17]:
```python
# Load dataset into dataframe
data = pd.read_csv("tiktok_dataset.csv")
```

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?

Find these answers later in the notebook.

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

In [18]:
```python
# Display and examine the first few rows of the dataframe
data.head()
```

Out[18]:

	#	claim_status	video_id	video_duration_sec	video_transcription_text	verifiec
0	1	claim	7017666017	59	someone shared with me that drone deliveries a...	no
1	2	claim	4014381136	32	someone shared with me that there are more mic...	no
2	3	claim	9859838091	31	someone shared with me that american industria...	no
3	4	claim	1866847991	25	someone shared with me that the metro of st. p...	no
4	5	claim	7105231098	19	someone shared with me that the number of busi...	no

In [19]:

```
# Get the size of the data
data.size
```

Out[19]: 232584

In [20]:

```
# Get the shape of the data
data.shape
```

Out[20]: (19382, 12)

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

In [21]:

```
# 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()`.

In [22]:
```
# Generate a table of descriptive statistics
data.describe()
```

Out[22]:

	#	video_id	video_duration_sec	video_view_count	video_like
count	19382.000000	1.938200e+04	19382.000000	19084.000000	19084.(
mean	9691.500000	5.627454e+09	32.421732	254708.558688	84304.(
std	5595.245794	2.536440e+09	16.229967	322893.280814	133420.
min	1.000000	1.234959e+09	5.000000	20.000000	0.(
25%	4846.250000	3.430417e+09	18.000000	4942.500000	810.:
50%	9691.500000	5.618664e+09	32.000000	9954.500000	3403.!
75%	14536.750000	7.843960e+09	47.000000	504327.000000	125020.(
max	19382.000000	9.999873e+09	60.000000	999817.000000	657830.(

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.

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

Exemplar 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.

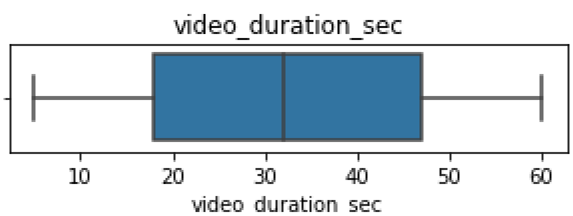
Task 3. Build visualizations

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

video_duration_sec

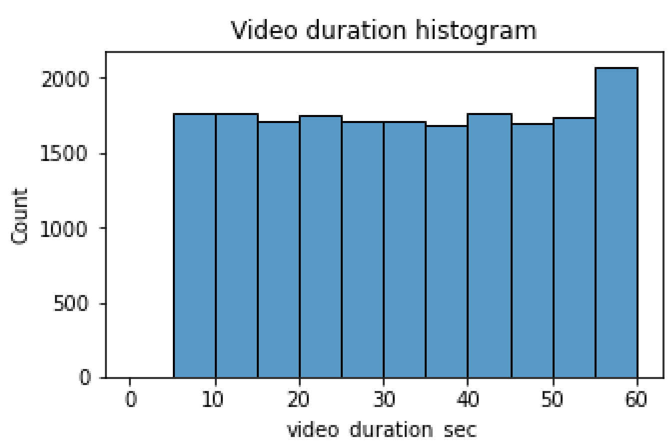
Create a box plot to examine the spread of values in the `video_duration_sec` column.

In [23]:
```
# Create a boxplot to visualize distribution of `video_duration_sec`
plt.figure(figsize=(5,1))
plt.title('video_duration_sec')
sns.boxplot(x=data['video_duration_sec']);
```



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

In [24]:
```
plt.figure(figsize=(5,3))
sns.histplot(data['video_duration_sec'], bins=range(0,61,5))
plt.title('Video duration histogram');
```

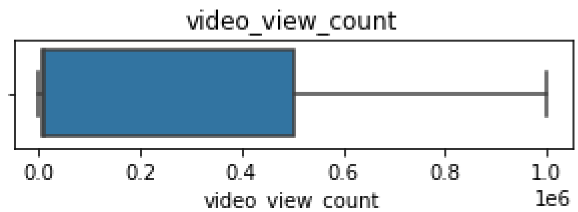


Response: All videos are 5-60 seconds in length, and the distribution is uniform.

video_view_count

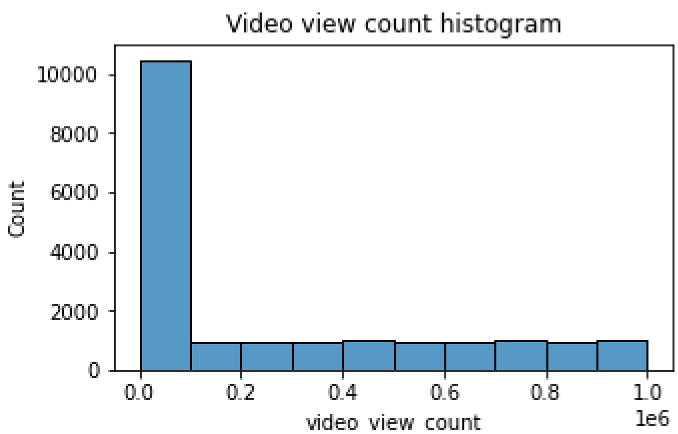
Create a box plot to examine the spread of values in the `video_view_count` column.

In [25]:
```
# Create a boxplot to visualize distribution of `video_view_count`
plt.figure(figsize=(5, 1))
plt.title('video_view_count')
sns.boxplot(x=data['video_view_count']);
```



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

In [26]:
```
plt.figure(figsize=(5,3))
sns.histplot(data['video_view_count'], bins=range(0,(10**6+1),10**5))
plt.title('Video view count histogram');
```

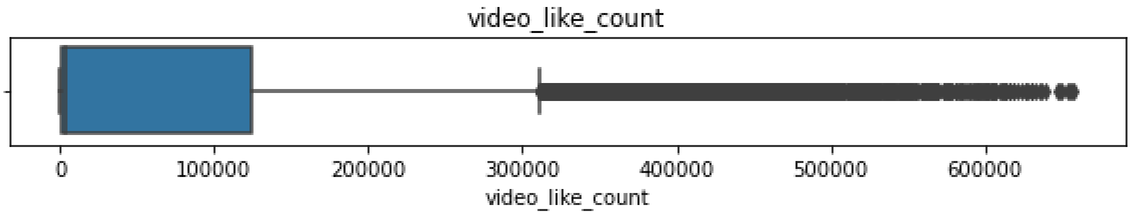


Response: This variable has a very uneven distribution, with more than half the videos receiving fewer than 100,000 views. Distribution of view counts > 100,000 views is uniform.

video_like_count

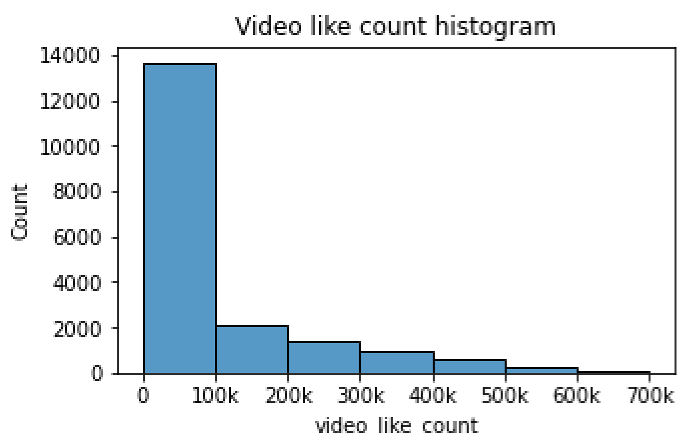
Create a box plot to examine the spread of values in the `video_like_count` column.

In [27]:
```
# Create a boxplot to visualize distribution of `video_like_count`
plt.figure(figsize=(10,1))
plt.title('video_like_count')
sns.boxplot(x=data['video_like_count']);
```



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

In [29]:
```
# Create the histogram
plt.figure(figsize=(5,3))
ax = sns.histplot(data['video_like_count'], bins=range(0,(7*10**5+1),10**5))

# Set the x-axis labels
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)

# Set the title
plt.title('Video like count histogram')

# Show the plot
plt.show()
```

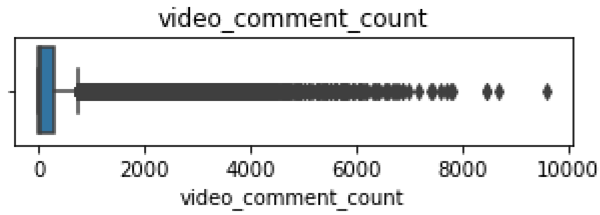


Response: Similar to view count, there are far more videos with < 100,000 likes than there are videos with more. However, in this case, there is more of a taper, as the data skews right, with many videos at the upper extremity of like count.

video_comment_count

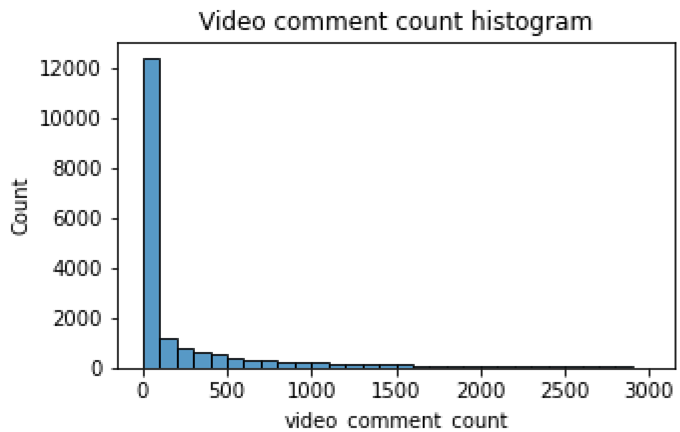
Create a box plot to examine the spread of values in the `video_comment_count` column.

In [30]:
```
# Create a boxplot to visualize distribution of `video_comment_count`
plt.figure(figsize=(5,1))
plt.title('video_comment_count')
sns.boxplot(x=data['video_comment_count']);
```



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

In [31]:
```
plt.figure(figsize=(5,3))
sns.histplot(data['video_comment_count'], bins=range(0,(3001),100))
plt.title('Video comment count histogram');
```

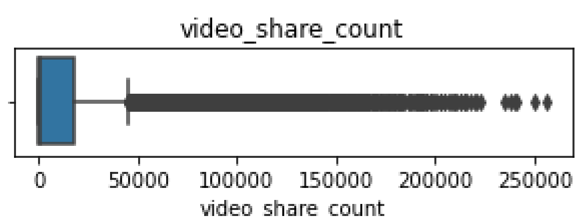


Response: Again, the vast majority of videos are grouped at the bottom of the range of values for video comment count. Most videos have fewer than 100 comments. The distribution is very right-skewed.

video_share_count

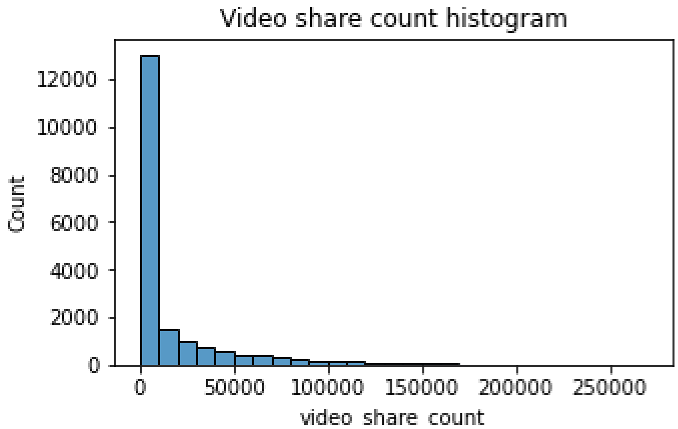
Create a box plot to examine the spread of values in the `video_share_count` column.

In [32]:
```
# Create a boxplot to visualize distribution of `video_share_count`
plt.figure(figsize=(5,1))
plt.title('video_share_count')
sns.boxplot(x=data['video_share_count']);
```



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

In [33]:
```python
plt.figure(figsize=(5,3))
sns.histplot(data['video_share_count'], bins=range(0,(270001),10000))
plt.title('Video share count histogram');
```

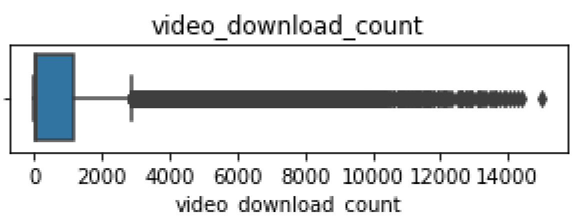


Response: The overwhelming majority of videos had fewer than 10,000 shares. The distribution is very skewed to the right.

video_download_count

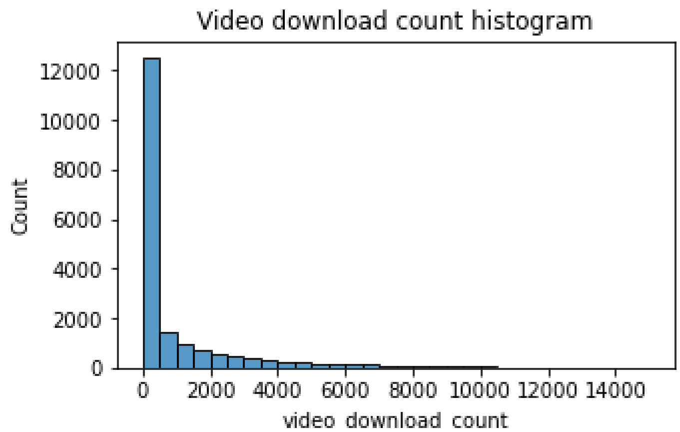
Create a box plot to examine the spread of values in the `video_download_count` column.

In [34]:
```python
# Create a boxplot to visualize distribution of `video_download_count`
plt.figure(figsize=(5,1))
plt.title('video_download_count')
sns.boxplot(x=data['video_download_count']);
```



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

In [35]:
```python
plt.figure(figsize=(5,3))
sns.histplot(data['video_download_count'], bins=range(0,(15001),500))
plt.title('Video download count histogram');
```

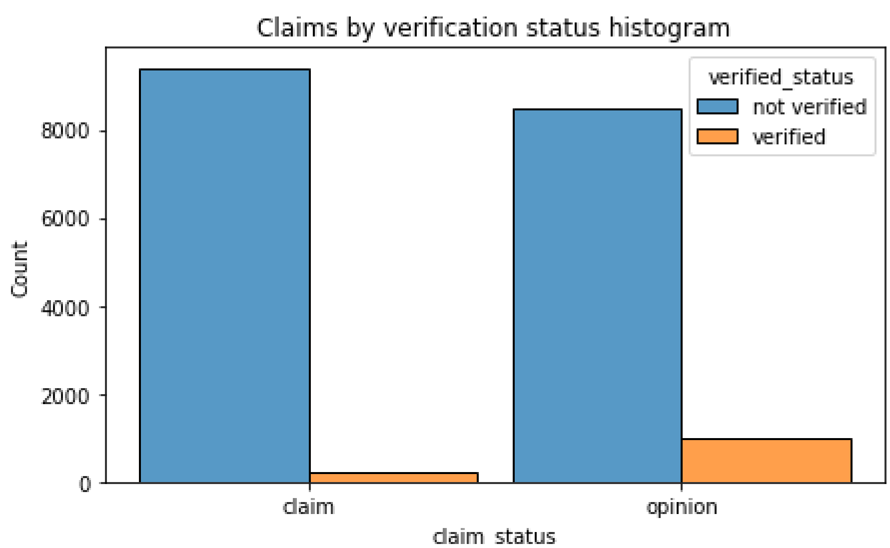


Response: The majority of videos were downloaded fewer than 500 times, but some were downloaded over 12,000 times. Again, the data is very skewed to the right.

Claim status by verification status

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

In [36]:
```
plt.figure(figsize=(7,4))
sns.histplot(data=data,
             x='claim_status',
             hue='verified_status',
             multiple='dodge',
             shrink=0.9)
plt.title('Claims by verification status histogram');
```

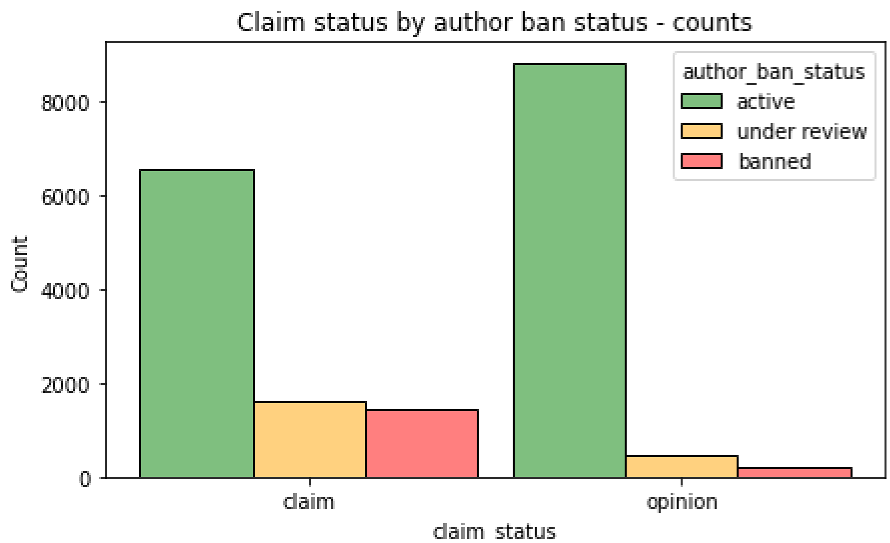


Response: There are far fewer verified users than unverified users, but if a user *is* verified, they are much more likely to post opinions.

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.

In [37]:
```
fig = plt.figure(figsize=(7,4))
sns.histplot(data, x='claim_status', hue='author_ban_status',
             multiple='dodge',
             hue_order=['active', 'under review', 'banned'],
             shrink=0.9,
             palette={'active':'green', 'under review':'orange', 'banned':'r
             alpha=0.5)
plt.title('Claim status by author ban status - counts');
```

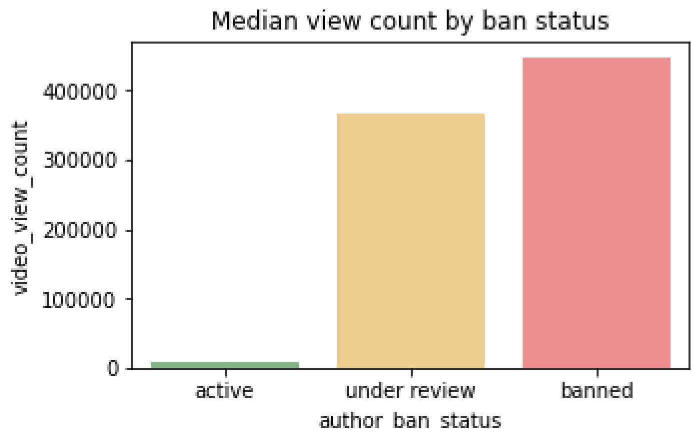


Response: For both claims and opinions, there are many more active authors than banned authors or authors under review; however, the proportion of active authors is far greater for opinion videos than for claim videos. Again, it seems that authors who post claim videos are more likely to come under review 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.

In [38]:
```
ban_status_counts = data.groupby(['author_ban_status']).median(
    numeric_only=True).reset_index()

fig = plt.figure(figsize=(5,3))
sns.barplot(data=ban_status_counts,
            x='author_ban_status',
            y='video_view_count',
            order=['active', 'under review', 'banned'],
            palette={'active':'green', 'under review':'orange', 'banned':'re
            alpha=0.5)
plt.title('Median view count by ban 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:

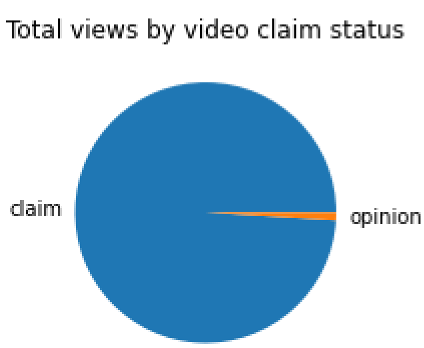
In [39]:
```
data.groupby('claim_status')['video_view_count'].median()
```

Out[39]:
```
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.

In [40]:
```
fig = plt.figure(figsize=(3,3))
plt.pie(data.groupby('claim_status')['video_view_count'].sum(), labels=['cla
plt.title('Total views by video 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.

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}"

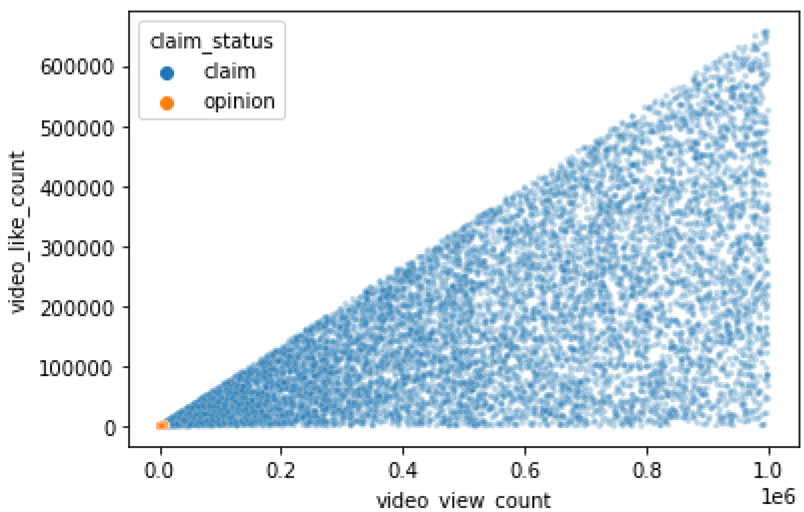
```
Example:
Number of outliers, video_view_count: ___
Number of outliers, video_like_count: ___
Number of outliers, video_share_count: ___
Number of outliers, video_download_count: ___
Number of outliers, video_comment_count: ___
```

```
In [41]: 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)
```

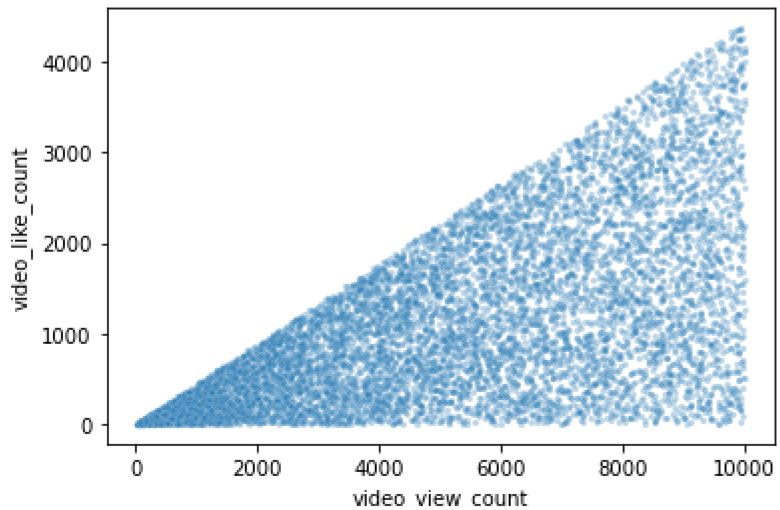
```
Number of outliers, video_view_count: 2343
Number of outliers, video_like_count: 3468
Number of outliers, video_share_count: 3732
Number of outliers, video_download_count: 3733
Number of outliers, video_comment_count: 3882
```

Scatterplot

```
In [42]: # Create a scatterplot of `video_view_count` versus `video_like_count` accor
         sns.scatterplot(x=data["video_view_count"], y=data["video_like_count"],
                         hue=data["claim_status"], s=10, alpha=.3)
         plt.show()
```




```
In [43]: # Create a scatterplot of `video_view_count` versus `video_like_count` for c
         opinion = data[data['claim_status']=='opinion']
         sns.scatterplot(x=opinion["video_view_count"], y=opinion["video_like_count"]
                          s=10, alpha=.3)
         plt.show()
```



No description has been provided for this image

PACE: Execute

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

Task 5a. Results and evaluation

Having built visualizations in Tableau and 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:

I have 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 other questions are ....

- *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.*

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.

Ask yourself: 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)*