Project Proposal

ggteam - Will Chen, Katelyn Cai, Hannah Choi, Weston Slayton

```r
library(tidyverse)
library(tidymodels)
# add other packages as needed

# add code to load data
```

Introduction

TikTok now has over 1 billion users globally, and over 150 million Americans, making it one of the fastest growing social platforms in the world. As it has risen to prominence, so has its ubiquitous algorithm, which is said to generally account for account factors (likes and comments) and video information (captions, sounds, hashtags). An internal TikTok document contained by the New York Times explained the algorithm in simplistic terms: "Plike X Vlike + Pcomment X Vcomment + Eplaytime X Vplaytime + Pplay X Vplay." Essentially, likes, comments and playtime, as well as an indication that the video has been played. Given, that TikTok has been heavily criticized alongside other platforms for declining youth mental health outcomes and rising hate due to the addictive nature of its explore page, we decided to look at TikTok's data ourselves and look at what drives video views (video length, likes, shares, comments, number of hashtags, and followers). Our hypothesis is that while likes, shares, comments, number of hashtags, and followers drive up video view count, video length count will drive that down.

Data description

The dataset comes from the 'top_users_vids.csv' file (under folder 'Trending Videos Data Collection') of the Github repository found at: https://github.com/ivantran96/TikTok_famous/tree/main. The data was originally collected as part of the DataResolutions's Data Blog project exploring Tiktok's demographics and trending video analytics.

The original data curators collected the data using David Teather's open-source Unofficial Tiktok API (found at https://github.com/davidteather/TikTok-Api), which uses Python to scrape Tiktok data and fetch the most trending videos, specific user information, and much more. Using the list of top Tiktokers, the curators compiled a list of users with the getSuggestedUsersbyIDCrawler api method, which used the top TikTokers and collected the suggested users. Using the byUsername method, they collected video data of the 25 most recent posts of each user from the top TikTokers and the suggested list. The curators also used the API's bySound method to collect videos using some of the most famous songs on TikTok to get an idea of how the choice of music can impact the potential of a video to become a trending video.

The dataset has 13 columns and 12,559 rows. The columns cover important metrics for trending videos such as video length, hashtags used, and number of likes, shares, comments, plays, and followers (and their total number of likes and videos). There are also less relevant variables such as create time, video ID which we will not use in our analysis. Variables id, create_time, video_length, n_likes, n_shares, n_comments, n_plays, n_followers, n_total_likes, and n_total_vids are numerical while the others are categorical.

Initial exploratory data analysis

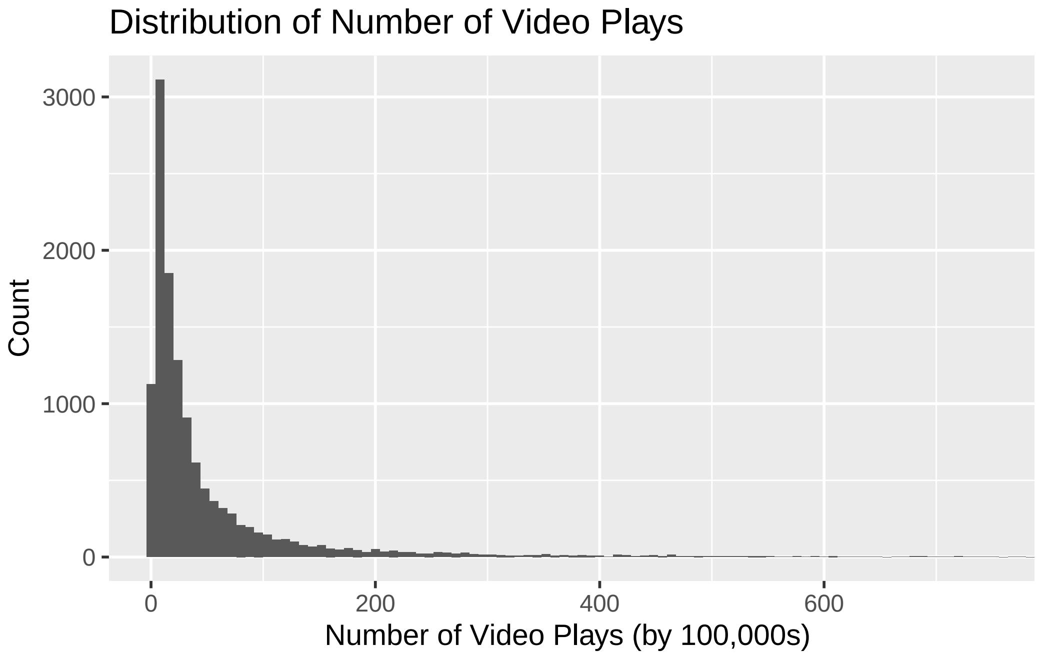
```
tiktok <- read.csv("data/top_users_vids.csv")
```

```
glimpse(tiktok)
```

```
Rows: 12,559
Columns: 13
$ id            <dbl> 6.892505e+18, 6.892162e+18, 6.892157e+18, 6.891688e+18, ~
$ create_time   <int> 1604786417, 1604706644, 1604705486, 1604596107, 16044396~
$ user_name     <chr> "charlidamelio", "charlidamelio", "charlidamelio", "char~
$ hashtags      <chr> "[]", "[]", "[]", "[]", "[]", "[]", "[]", "[]", "[]", "[~
$ song          <chr> "Adderall (Corvette Corvette)", "original sound", "origi~
$ video_length  <int> 15, 9, 4, 15, 13, 7, 13, 38, 13, 7, 12, 9, 6, 14, 11, 7,~
$ n_likes       <int> 480800, 3100000, 2400000, 3200000, 7500000, 7100000, 320~
$ n_shares      <int> 9256, 17200, 17800, 12700, 31100, 43000, 8610, 25000, 39~
$ n_comments    <int> 51300, 105700, 69200, 64100, 290300, 82000, 43600, 55500~
$ n_plays       <int> 1900000, 13300000, 10100000, 14600000, 34700000, 3330000~
$ n_followers   <int> 97400000, 97400000, 97400000, 97400000, 97400000, 974000~
$ n_total_likes <dbl> 7.6e+09, 7.6e+09, 7.6e+09, 7.6e+09, 7.6e+09, 7.6e+09, 7.~
$ n_total_vids  <int> 1642, 1642, 1642, 1642, 1642, 1642, 1642, 1642, 1642, 16~
```

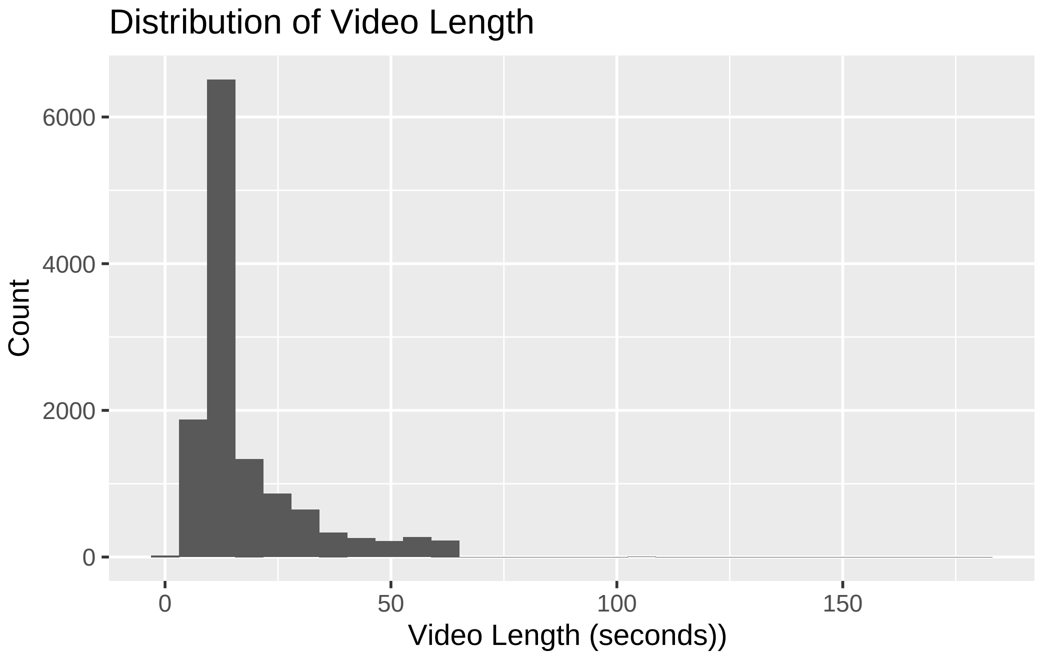
We are interested in predicting the number of times a video will be played. Therefore, we set n_plays as our response variable.

```
tiktok |>
  ggplot(aes(x = n_plays / 100000)
         ) +
  labs(x = "Number of Video Plays (by 100,000s) ", y = "Count", title = "Distribution of N
  geom_histogram(binwidth = 8) +
  coord_cartesian(xlim = c(0, 750))
```

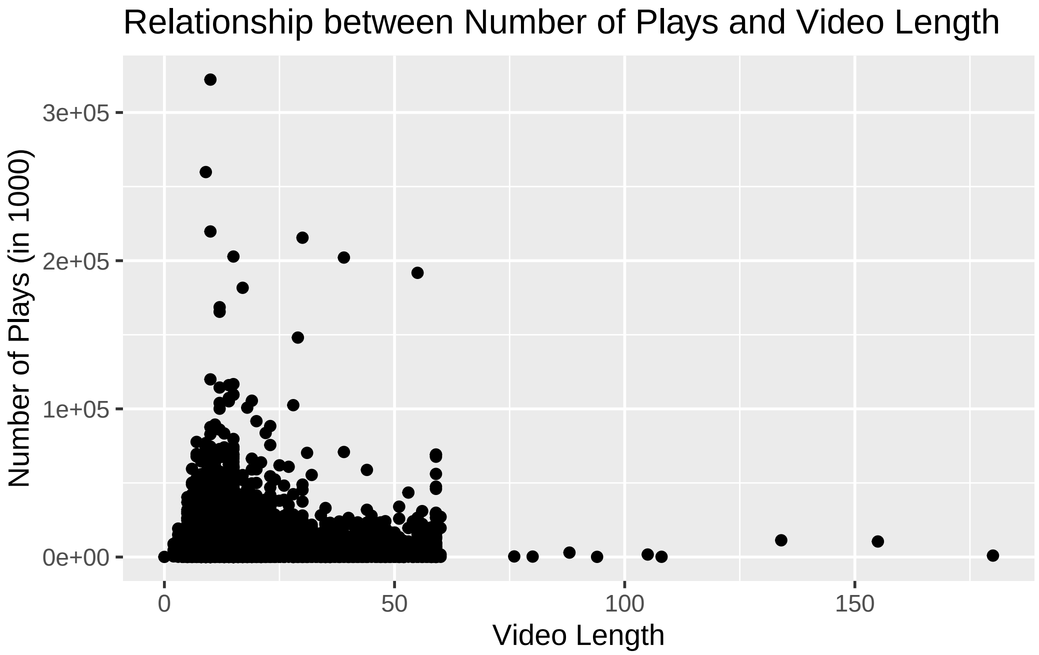


We put video_length as our quantitative predictor variable to explore. We are interested in knowing if Tiktok's algorithm prefers recommending shorter videos or if users are more likely to view short videos as opposed to longer ones.

```
ggplot(data = tiktok, aes(x = video_length)) +
  geom_histogram() +
  labs(x = "Video Length (seconds)) ", y = "Count", title = "Distribution of Video Length"
```



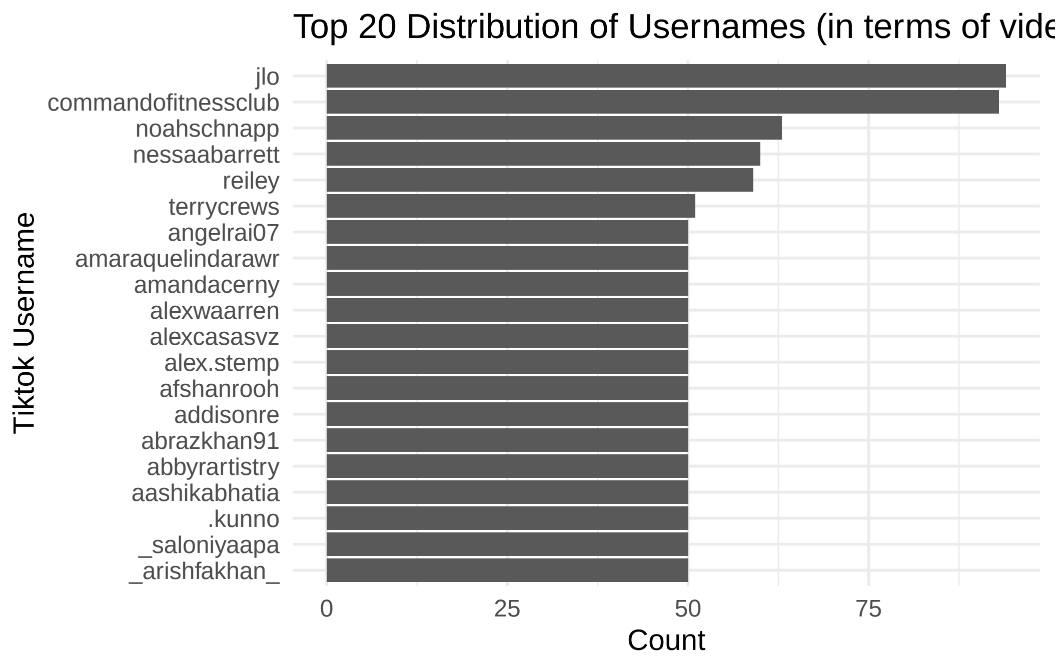
```
ggplot(tiktok, aes(x = video_length, y = n_plays / 1000)) +
  geom_point() +
  labs(title = "Relationship between Number of Plays and Video Length", x = "Video Length"
```



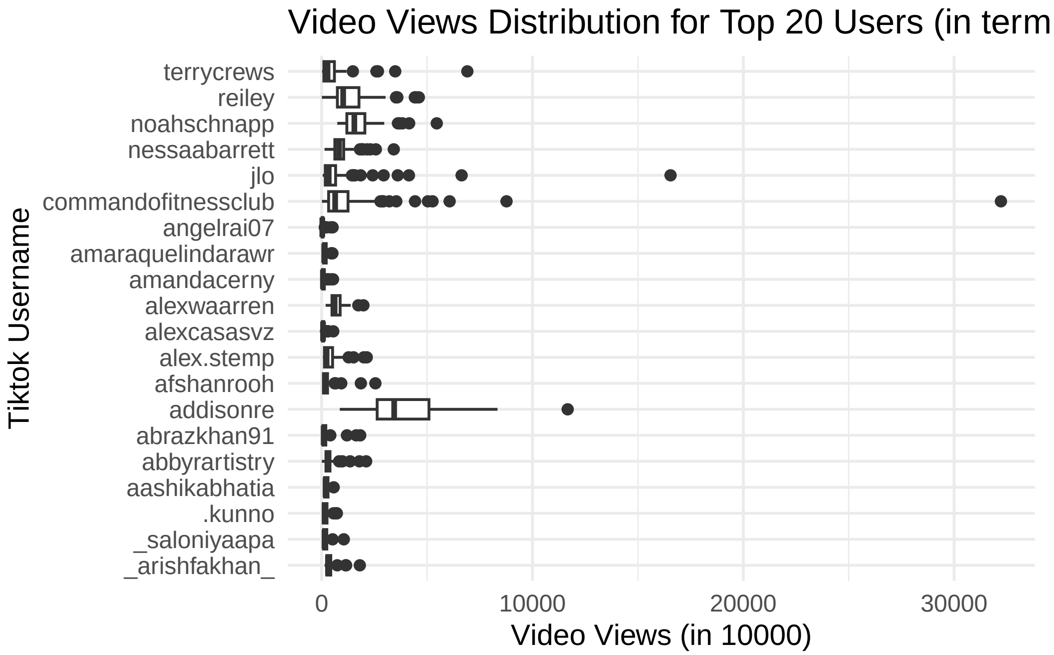
The relationship seems a bit weak, with a few outliers that includes video_length greater than 75. However there seems to be a greater overall average view with short video_length.

Now we are interesting in knowing if a few users are drawing in most of the views. Since currently we don't have access to user popularity, it suffices to just find the users who uploaded the most and find the respectively views for their videos.

```
# since there are too many user names, filter top 20
user_counts <- tiktok |>
  group_by(user_name) |>
  summarize(count = n()) |>
  arrange(-count)

N <- 20
top_users <- head(user_counts, N)

ggplot(data = top_users, aes(x = reorder(user_name, count), y = count)) +
  geom_bar(stat = "identity") +
  coord_flip() +
  labs(x = "Tiktok Username", y = "Count", title = "Top 20 Distribution of Usernames (in t
  theme_minimal()
```



```r
filtered_tiktok <- tiktok |>
  filter(user_name %in% top_users$user_name)

ggplot(data = filtered_tiktok, aes(x = user_name, y = n_plays / 10000)) +
  geom_boxplot() +
  coord_flip() +
  labs(x = "Tiktok Username", y = "Video Views (in 10000)", title = "Video Views Distribut
  theme_minimal()
```



From the plot, we can see a few users getting more views on average compared to other users. This might indicate a stronger relationship between the user and the number of plays they receive per video.

A potential interaction effect could be between number of likes and number of comments, since likes and comments can both increase a video's visibility, so their interaction might have a multiplicative effect on views:

$$\text{video\_views} = \beta_0 + \beta_1 \cdot \text{num\_comments} + \beta_2 \cdot \text{num\_likes} + \beta_3 \cdot (\text{num\_comments} \times \text{num\_likes}) + other\_variables + \epsilon$$

For the data cleaning and model preparation process, we'll delete variables that aren't really useful such as create time, sounds/songs used (since there are so many unique songs - as well as different spellings to the same song – and customs sounds that could use a particular song, it becomes useless in model) and hashtags (since many videos don't include one). We'll also add a column that includes the following of the user, which will be useful in determining if a video gets a certain amount of views.

```
tiktok |>
  summarize_all(~sum(is.na(.)))
```

```
id create_time user_name hashtags song video_length n_likes n_shares
```

```
1  0            0         0        0    0            0       0        0
  n_comments n_plays n_followers n_total_likes n_total_vids
1          0       0           0             0            0
```

There seems to be no missing NULL value in our dataset with the exception of hash tags (tidyverse counted the empty list as a value).

Analysis approach

The response variable that we will be using is n_plays. This represents the amount of views that a video obtained. Possible predictors include n_likes, n_shares, n_comments, n_followers, # of hashtags (which will have to be mutated from the hashtags variable), and video_length. We assume that each of these predictors will have a positive effect on n_plays except for video_length.

To perform this analysis, we will use multiple linear regression. This is because we are using a quantitative response variable. We will be able to predict the amount of views a video has based on varoius predictors. Logistic regression isn't useful here, as our response variable isn't categorical, and we aren't looking for a probability.

Data dictionary

The data dictionary can be found here. README link