Mo Nayal, Gary Chen

IST-652

Homework 2: Semi-Structured Data

Date: 8/29/2021

**Homework 2 Write up**

Data and its source:

For this homework assignment, we pulled in Semi-structured data in the form of YouTube comments. We used Google’s own Api called YouTube Data API v3 to pull in data about a top trending YouTube video. At the time of project start, the top 3rd trending YouTube video was a “Submarine Minefield Battle” by a channel called Dude Perfect. We chose the 3rd highest trending because the first trending video was a Marvel Movie trailer and the second was a music video. We wanted to analyze comments on content by independent creators. The video source is:

<https://www.youtube.com/watch?v=BTVMLRSsb3o>.

Setting up the YouTube Api and writing the code to pull in a semi structured dataset was done with the help of the outlined code from the following source:

<https://www.thepythoncode.com/article/using-youtube-api-in-python#Extracting_YouTube_Comments>

**Data Cleaning, Exploration, and Transformation:**

The YouTube Api was enabled, and the following libraries were imported:

from googleapiclient.discovery import build

from google\_auth\_oauthlib.flow import InstalledAppFlow

from google.auth.transport.requests import Request

import urllib.parse as p

import re

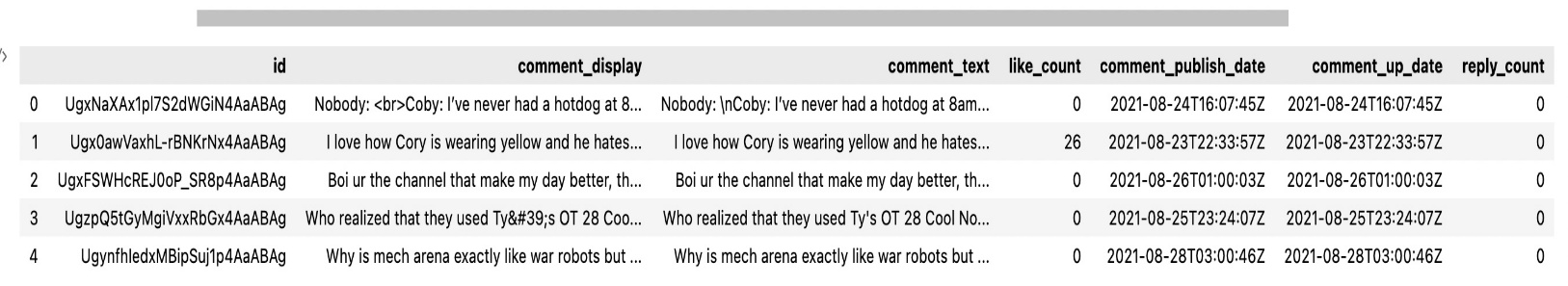
import os

import pickle

Then we brough in the ‘youtube\_authenticate’ function which authenticates with the YouTube API. It checks for your “token.pickle” and looks for the credentials.json to ensure you have all permissions to run the YouTube Api. Next, we gave it the URL of the video we wanted and had a function to check that the URL for a YouTube video. We then made it called on the YouTube Api module get\_comments with our given parameters to pull a comment id, comment text, posted date, updated date, the like count for a comment, and the reply count. The Google YouTube Data Api v3 has a limit of pulling 100 comments so we were forced to rerun it multiple times and concatenate the outputs together to get a data set large enough for the scope of this project.

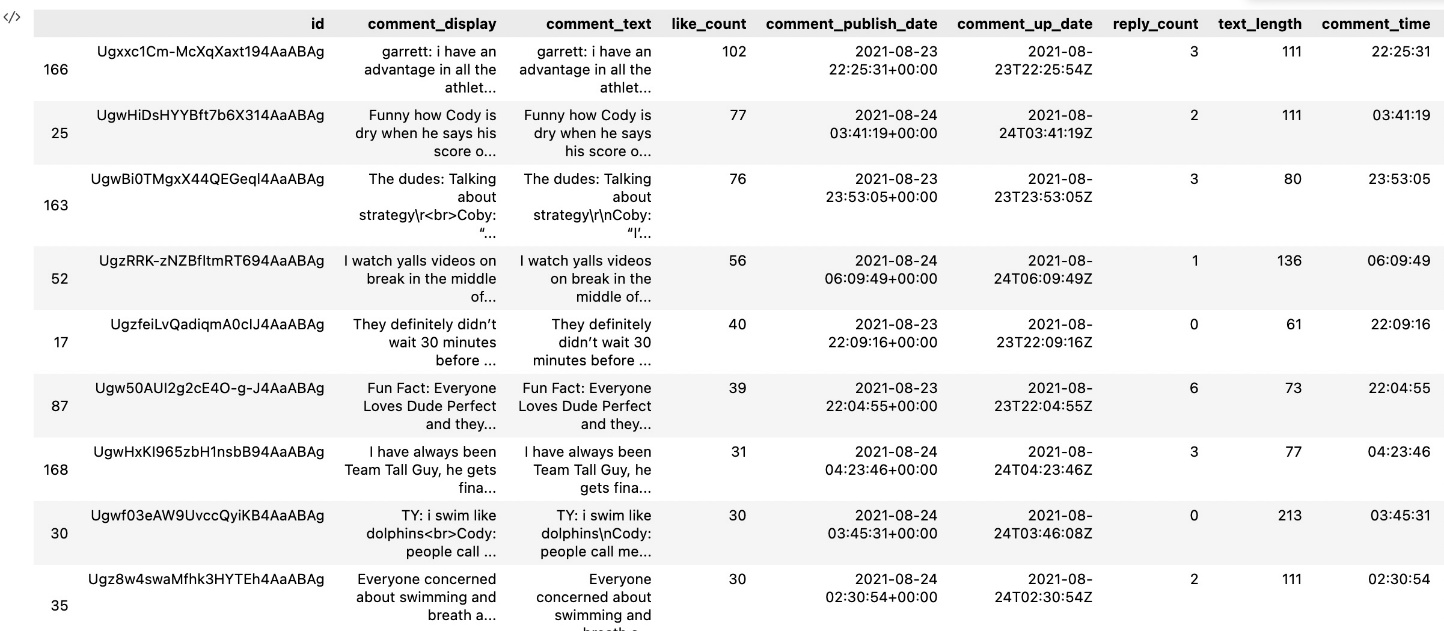
From the API source object, we were able to create a JSON object and then used Pandas to turn the JSON object into a dataframe. Our next step was to export the dataframe that still needed to be cleaned as a csv. We decided exporting the dataframe would allow us to work with the data without having to rely on the Api connection and get fresh data which might change with every iteration and make the analysis more difficult.

All further cleaning was done on the dataframe from the reimported csv file. To clean the dataframe column names were renamed to be more understandable. Columns that are not useful for analysis were dropped. The like counts and reply counts were turned into numeric datatypes and the posted and updated date columns were formatted as datetimes. A screenshot of the first few rows of our dataframe is below.

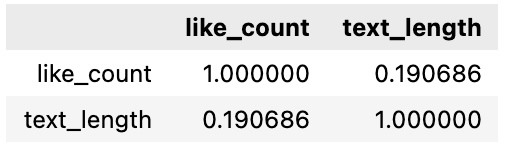


**Questions:**

The first thing we were interested to look at was briefly what were the top liked comments. During this step we noticed that either by virtue of having to rerun the Api multiple times to get more than 100 comments or because some comments were posted multiple times there were duplicates in our dataframe. During this step we removed any duplicates in the dataframe and unfortunately our number of entries dropped from over 400 to 252. But we were then able to display the top 10 comments from our dataframe with only unique entries. Output below.

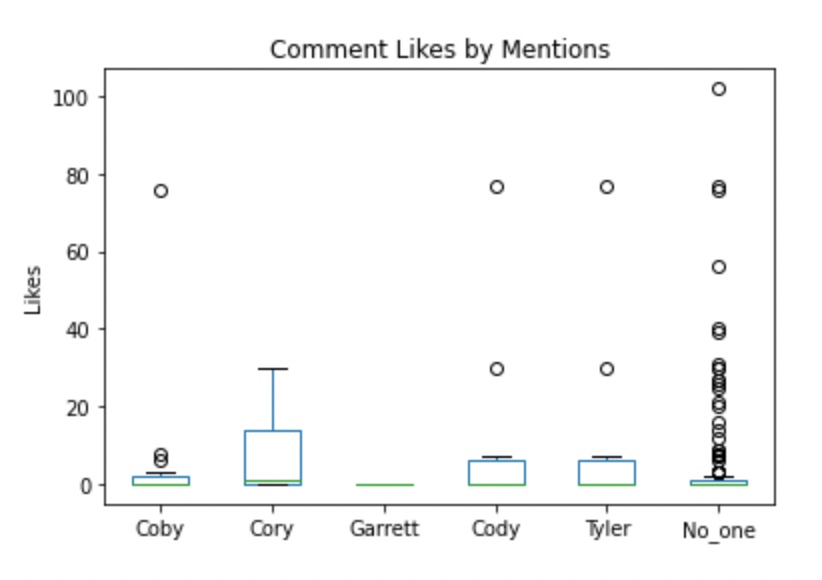


Our first question is whether the length of the comments affects how many likes it receives. We were wondering if exceptionally long comments were skipped over and received very few likes compared to short succinct comments. To tackle this question a new column in our dataframe was created called text\_length which counted the number of characters in the string of each comment. A correlation analysis using the ‘Pearson’ method was run on the two variables like\_count and text\_length. The output is below.

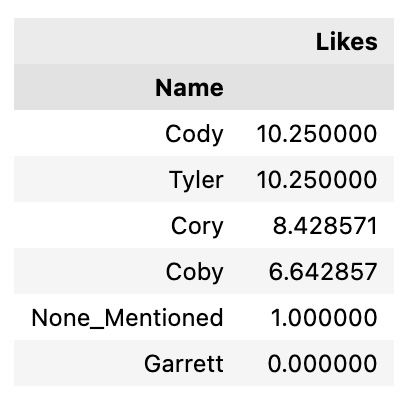


Based on this output like\_count and text\_length is not correlated because the output, 0.19 is much closer to 0 than either 1 or –1. From this we would not be able to say that the length of comment has an impact on the number of likes it receives.

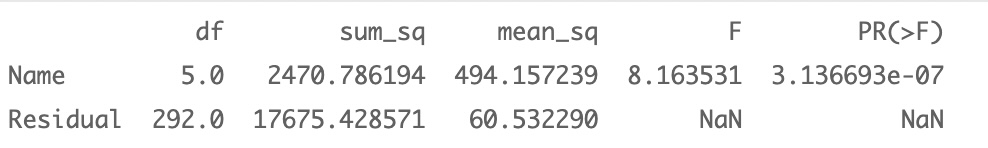
Our second question was do comments where one of the YouTube creators names are mentioned receive more likes. Looking at the top comments printed earlier it seemed that the YouTube creators’ names were present in quite a few. The YouTube creators’ names from the channel ‘Dude Perfect’ are Cory, Coby, Garrett, Cody, and Tyler. The first step to address this question was to create 5 new columns in the dataframe for each of the 5 members that contained a 1 if the string from comment display contained the string of their name or a 0 if it did not. Next another column was created that contained a 1 if none of the names were in the comment or 0 if at least one of their names was in the comment. A new dataframe was created from these 6 new columns and a boxplot was created to show how many likes mentioning each name resulted in and how many likes comments that did not mention any of the names resulted in. The output of the boxplot for this dataframe is below:



We can see from this boxplot that the median likes and the interquartile range for likes where comments mentioned Coby, Cory Cody, or Tyler was much higher than where no one was mentioned. However, it is also apparent that there are much more outliers in comments where No one was mentioned that had many likes. It looks like most comments where no one was mentioned resulted in a small number of likes, but on some rare occasions a comment that did not mention anyone's name received many likes. To look at this numerically the means were taken of each of these groups and the output is below:

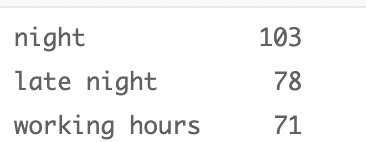


Finally, an Anova was conducted using the ols library from the statsmodels.api package to see if these differences in the means were statistically significant. The output is below:

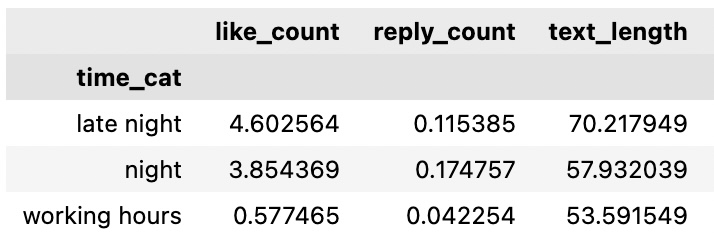


Because the p value on the F statistic is exceptionally low, and much lower than commonly used alpha levels of 0.05, we would reject a null hypothesis that comments with any of these named mentioned or none of these names mentioned are equally likely to receive the same number of likes. From this we can say commenting on a YouTube video and saying the creator’s name in the comment is more likely to get a high number of likes than not including their name in the comment.

Our third question is does the time of day that a comment is posted affect the number of likes it receives? For this question, the hour of day when each comment was posted was extracted from the datetime and added to a new column in the dataframe. Then the hours of the day were divided into three groups to get a comparison. The bins chosen spanned from 8AM to 4PM which we considered working hours, 4PM to 12AM which we called nighttime, and finally 12AM to 8AM which we called late night. Our first task to tackle this question was to check whether the number of comments for each category differs. The counts of comments posted during each time bin are below:



Next, we looked at some comment statistics based on comments in these time periods which included likes but also included other factors such as reply count and text length. These variables would help us determine if time of day impacts how likely it is that someone would interact with a comment in general by either liking or replying and text length addresses the question of does time of day change how much a person writes. The output is below:



Based on these means we can say on average the longest comments are written late at night and comments posted late at night are the most likely to receive higher likes. On average comments posted during working hours get the fewest replies and the fewest likes.

**Conclusion and Results:**

We were successfully able to pull semi structured data from YouTube and analyze data on YouTube comments. Our results show that the length of a comment is not correlated with how many likes it will get, time of day of posting a comment does impact likes and number of replies, mentioning a YouTube creator’s name in a comment results in a statistically significant increase in likes on a comment. The time of day is subjective to where you live, so the analysis of categorical time bins is imperfect. We are assuming the timestamp of comments posted is based on the Eastern time zone because that is where the API was ran. From looking at the time categorical data, we can conclude that Youtube users were more active during nighttime (4PM to 12 AM). That was when most comments were posted and when people were most likely to reply to other comments. This gave comments more visibility for other users to hit the like button. This had a spillover effect where most of the likes came late night. Working hours on average had the least amount of likes and replies.

**Team member contributions:**

Mo found the YouTube API source and walked the team through how to enable the YouTube Data API v3 and helped grant Credential access to our Unstructured data. Mo was successful in exporting the data into a JSON format. Mo also worked on the second question looking for the YouTubers’ names in comments and comparing the number of likes based on if any of them were mentioned as well as plotting the number of likes, getting the means, and running a one-way Anova on likes vs name mentioned.

Gary was able to change the datatypes of the original dataframe and get all the DataTimes formatted properly. Gary also tackled our first question on whether shorter comments result in higher likes by getting all the text lengths and running a correlation analysis between like counts and text length. Gary also found insights concerning our third question involving comment posted times and the number of likes. He created the bins of time of day, grouped the comments in bins of the day and summarized the output.