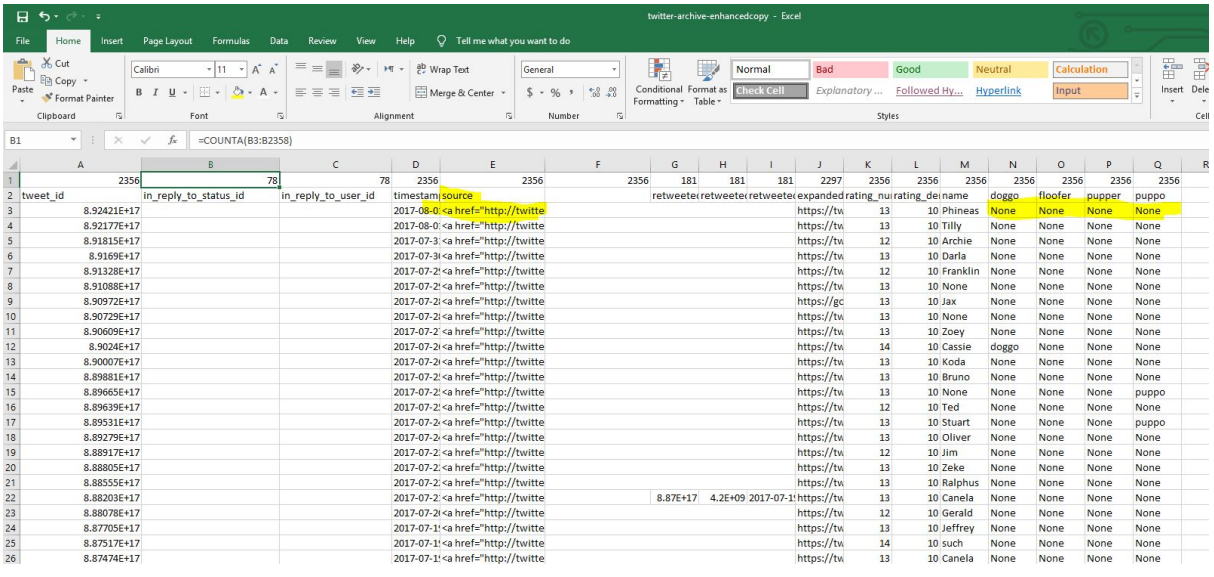
This file is a documentation of my wrangling effort on this project:

First of all Gathering Data:

- I found this part was the easiest and fastest part. I tried to import the files as required.
- The instructions to import these files was very straight to the point and all in the course content

Second part Assessing Data:

- This part was very challenging for me as it was very large data and I took too much time to digest the data and understand what really needed to be done
- I've Assessed the data programmatically using jupyter notebook and Visually using excel
- I tried to address each Issue I found to the method detected with in the ipynb files
- I've used many pandas methods like .info() / .head() / .describe()/ .value_counts()/ .nunique(), and also tried to be very explanatory in the jupyter notebook files to see how is my workflow is going
- For the visual part I also transformed TSV file to a CSV file so that I can read it easily as shown, Atom was very hard to see the patterns in it



My wrangling effort ended with these findings:

Tidiness Issues:

`archive_df`

- related rows to retweets and replies (259 row)
- 4 columns of dogs classification
- 5 columns of un needed data related to retweets and replies

`api_df`

- favorites and retweet count columns needed to be merged with the first table to create a unit of tweets' obeservation

Quality Issues:

`archive_df`

- tweet_id: is int #programatically
- timestamp: is object #programatically
- None are represented as string not NaN object #programatically
- tweets with empty images (extended_url) #programatically
- tweets has two classifications together #visually
- some dog names are not extracted correctly #visually
- there are (rating_denominator) > 10 which manipulate the scale of its numerator likr rows 903 and 1121 #programatically
- source column is not represented in clear naming #visually
- tweets with no dog classification #visually

`image_predictions_df`

- tweet_id: is int #programatically
- there images in this table is not in archive table -visually- #visually

`api_df`

- tweet_id: is int -programatically-

To be honest I didn't manage to solve 2 points of them thus I was short in time.
And yes I'm really looking forward for your help and criticize

---

Second part Assessing Data:

- Cleaning part was the most time consuming part I took alot of time to search and learn how to fix the issues I found
- I ensured that my cleaning was under the specifications ***(Define,Code,Test)***
- I've enjoyed so much about Pattern Matching with Regular Expressions and really was thrilled when I generated the dog names it was very exciting
- Also the part which I found it not done in a proper way was the melting the variables into one column, I tried my way but I believe there is better way to achieve that

All my cleaning process is detailed in the jupyter files I hope you can understand it and help me improving my skills