Wrangle Report

1. After gathering all the data I needed, I started to assess and clean the data. When open table `df1`, I found out these columns, `doggo`, `floofer`, `pupper`, and `puppo` are the same variable, they should go together. So, I create a new column called dog_stage to store them. Before this, I used .replace()function to replace the missing value “None” with blank.

2. When I checked up table `df3`, I realized that `df3`only has 3 columns and it should be a part of table`df1`. So, I decided to combine table `df3` into table `df1` by the common column `tweet_id`. But before the merge, I need to correct the columns’ datatype from string to int including `tweet_id`, `retweets` and `favorites` columns. I converted `tweet_id` datatype back to string after merge.

3. From table `df2`, I found out there are some image are not dogs. At the same time, if one of prediction is False in table `df2`, with the same tweeter id, the picture in `df1` also isn’t dog. So,I used .query() function to remove `p1_dog`, `p2_dog`, and `p3_dog` are all "False". And then I used pd.merge() function to merge `df1` into `df2`.

4. I used .query() to group rating_denominator != 10, rating_numerator > 15, and rating_numerator < 5. When I read the text, I found out there are so many data extracted wrong, so, I corrected them one by one, totally is 13. Maybe there is a better way to do this.

5. In `name` column, missing dog names recorded as 'None', I used .replace() function again to convert them to NaN. And then, I also converted the typing wrong in name being 'a', 'an', 'the' to NaN.

6. By using .info () function, I found out `timestamp` ’s datatype is string and it should be datetime form. So, I used the .datatime() to converted it.

7. I used .dorp() function to drop off the unnecessary columns including `retweeted_status_id`, `retweeted_status_user_id`, `retweeted_status_timestamp`, `expanded_urls`, and `source`.

8. In table`df2`, I created a new column for predicting god breeds and using conditional statement If, Else to let all the stages in the same column. After that, I used .drop() function to drop off the unnecessary columns:
`img_num`,`p1`,`p1_conf`,`p1_dog`,`p2`,`p2_conf`,`p2_dog`,`p3`,`p3_conf`,`p3_dog`.

9. In the table `df1`, the image of `in_reply_to_status_id` and `in_reply_to_user_id` is not WeRateDog orignal tweet, so I decided to clean out.

10. My analysis result with strongly related to the data of `retweets` and `favorites`, so I used .query() function again to clean out all NaN value on those two columns.