I just wanted to outline some difficulties that I came across while cleaning the data and why I did not go about cleaning the data in the most efficient way. When I assessed the numerators and denominators, it was possible to see that many denominators are not equal to 10 because the rating is given for multiple dogs or there is another (non-rating) fraction in the tweet text field that caused the data to be incorrectly extracted. The best approach would have been to resolve this programmatically, but I was not sure about how to handle multiple dog ratings (did the author intend to give them equal rating?); I chose to drop all the denominator values that are not equal to 10. Similarly, some of the numerators were not only of consolidated ratings but were also improperly extracted. There were instances with multiple fractions in the tweet, for example: “She was the last surviving 9/11 search dog, and our second ever 14/10” or “This is Darrel. He just robbed a 7/11 and is in a high speed police chase. Was just spotted by the helicopter 10/10,” these are only some of the examples where there was a both a contextual reference and a rating. Some of the numerators were also decimal values, I decided to keep only those values that are less than 14, but I manually changed the decimals. Similarly, a lot of the extracted names were erroneous, with words such as “a,” “an,” and “the” registering as names. Initially, I thought about ways to properly extract them, but some tweets did not have names, while others held various naming formats as “*This is* Calvin,” “*Meet* Olive,” “*Here is* George,” or even combined to something like “*This is* a southwest Coriander *named* Klint,” where the two nearby upper-case letters can easily come to qualify as names in any extraction process. I decided to just gather all words that are in lowercase and to replace them with “None.” I was uncertain of how to iterate and apply regex to so many variations (I found even the simpler cases of where I extracted difficult to work with). I understand, that while a lot has been done to wrangle the data, this is only the beginning of assessing the dataset for quality and tidiness and a lot more can be done still to make the data clean.

From the “Most Common Dog Type” visualization, it is possible to see that “Pupper,” “Doggo,” and “Puppo” have the greatest counts, this is verifiable by the value counts displayed below the visualisationd.

When comparing the “Dog Type and Retweets,” it is possible to observe that while “Doggo” has the greatest dispersion of retweet counts, with the greatest retweet count values, “Pupper” values have the greatest concentration toward the lower retweet counts, with many more overall. I then decided to observe the retweet count of the top 75% percentile and notice that majority of the top retweets have a rating of above 11.

Similarly, in comparing “Dog Type and Favorite,” “Doggo” values have the highest favorite counts, the greatest dispersion of values and a strong concentration, whereas “Pupper” has not as high favorite counts values, but is more concentrated to lower favorite counts. From the favorite count of the top 75% percentile, it can be ovserved that the top favorite counts mostly have a rating of 12 and above.

To tie everything together within the “Dog Type and Rating” boxplot, it can be observed that “Pupper” gets lower rating, with the lowest median and mean, whereas “Floofer” has overall more higher ratings than lower. “Doggo” and “Puppo” are very similar, but “Doggo” does have a tendency to get lower rathing.

The other two visualisations just represent the top 15 most common dog names, excluding “None” and the top 10 Dog Breeds, with Golden retvievers taking a considerable lead.