Wrangle Report

I started my wrangling efforts by gathering the data from the three sources. WeRateDogs provided Udacity with access to their Twitter archive which I then downloaded in csv format to my computer. The neural networks dataset was downloaded programmatically using the Requests library as a tsv file. Finally, I queried the Twitter API for each tweet's JSON data using the tweet IDs found in the archive file as a reference. Once all the datasets were collected they were stored in dataframes ready for the assessing stage.

My first step in assessing the data was to create copies of the three existing dataframes to be used for cleaning. I applied the describe, info, head and tail functions to each dataframe to get an overall look at the formatting and any obvious errors or faults. The tweet info dataset obtained through the API was relatively clean and only needed the tweet ID's changed from a number to a string. The neural network dataframe had a few quality issues including capitalisation of names, the use of underscores where spaces could be and data/time data being in string format and in the same column. The bulk of the cleaning required was performed on the archive table. Several quality and tidiness issues were observed, and these included retweets being included, messy entries and dog stages listed across 4 columns. After exploring the variables a little more, I found other quality issues such as the inclusion of character references for ampersand as well as inconsistencies for numerators that were listed as floats in the text variables.

The time/date column was split into datetime objects, one variable for time and the other for date. The dog stages were condensed to a single column and 'none' entries were replaced with NaN's. Using the str.replace and str.title functions the neural network dataframe was made more presentable by replacing underscores with spaces and capitalising the words. The messy source variable entries in the archive table were cleaned up and the HTML anchor tags removed.

As a final step, inner joins were performed using pd.merge function to merge all three tables into a master dataframe and irrelevant variables were then dropped from this.