# DATA WRANGLING REPORT: *WERATEDOGS* TWITTER ACCOUNT

Data wrangling is a step in the data analysis process that involves acquiring a set of data from a single or multiple sources and then refining it to meet a set standard of quality that ensures reliable analysis and valid conclusions. For my project, I employed the steps in their order to effectively wrangle the *weratedogs* Twitter account data.

## Data Gathering:

The data wrangling of the *weratedogs* Twitter account data, began by collecting three (3) sets of data:

i. Twitter Archive Enhanced (.csv): A data set downloaded from Udacity ALX-T Data Analysis class room. Which contained 17 variables (columns) about basic information of tweets from the *weratedogs* twitter account from August 1, 2017.

ii. Image Predictions (.tsv): This was downloaded programmatically using the python *requests* module from Udacity's server. It contained 12 variables (columns), which provided information about predicted identity qualities for each dog rated in the data set.

iii. Tweet Json (.txt): This was supposed to be downloaded using a python Twitter Application Programming Interface (API) module – Tweepy. However, due to delay in Twitter responding to my developer account request, I had to use the alternative data set provided. This data set contained 3 variables (columns), which provided additional information for each tweet such as retweet count, and favorite count.

All together there were 32 variables contained in the 3 data sets. After successfully collecting all 3 data sets, they were read into a Jupyter notebook as a dataframe using the Python Pandas module and its methods.

## Data Assessment:

Before assessing the 3 data sets now dataframes, duplicate copies were made for each using the pandas *.copy()* method so as to preserve and easily retrieve the original dataframes. Afterwards, each dataframes were assessed visually and programmatically.

I mostly carried out visual assessment by calling the *df* (dataframe) directly on a cell to preview its content on the Jupyter notebook. Afterwards, I get an overview of it's other intricate properties using methods such as *df.info()* – check data types, missing values and variables, *df.shape* – get dataframe dimension, *df.unique()* – see number of unique values, *df.duplicated().any()* – check if there are any duplicates, *df.describe()* – summary statistics, among others. While noting observed quality and tidiness issues below each cell in my Jupyter notebook.

**Data Cleaning:**

Using the *define-code-test* technique, taking each data set at a time, I outlined how I intended fixing each particular I identified for each dataframe in the assessment stage (define), then I write the code needed to fix the issue (code). For example, I used the following code; *twt_arc.timestamp=pd.to_datetime(twt_arc.timestamp)* – to convert a column containing date series values from object to datetime data type. Finally, I write a code to confirm that the fix had been applied to the respective dataframe.

After cleaning the three (3) dataframes, I merge them together on the *tweet_id* column, being the common variable among the 3 data sets and then exported it as a .csv flat file using the pandas *.to_csv()* method.