# "WeRateDogs" Wrangle Report

# Introduction

The purpose of this project is to wrangle the tweet archive of Twitter user @dog_rates, also known as WeRateDogs, to prepare it for further analysis and visulaization. WeRateDogs is a Twitter account that rates people's dogs along with posting comments about the dog.

In this report, I briefly state my wrangling efforts with the "WeRateDogs" data, discussing each of the following phases in more details, namely; gathering, assessing, and cleaning data.

# Gathering data

First, data was gathered from the following sources:

1. Twitter archive file: downloaded manually in csv format, as provided by Udacity.
2. image predictions: downloaded programmatically using the Requests library through the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv. It includes image predictions of dog breed produced by a neural network.
3. Tweet-JSON: This file contains each tweet's retweet count and favorite ("like") count, obtained by querying the twitter API for each tweet's JSON data using Python's Tweepy library. This .txt file was then read line by line into a pandas dataframe.

# Assessing Data

## First: Visual assessment:

First, data was assessed visually which is basically done by scrolling through samples of the data. Some quality issues were easily spotted through visual assessment such as missing values, or tidiness issues such as multiple columns for dog stage values.

## Second: Programmatic assessment:

Programmatic assessment was then done to get a more thorough understanding of the quality and tidiness issues in the data. Using useful pandas functions were used to do this including: info, describe, and value_counts methods.

After the assessment, issues with the data were identified and listed in the following:

# Quality Issues:

## a. twitter_archive:

1. Irrelevant Data (retweets data included)
2. Unneeded columns (retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, in_reply_to_status_id, in_reply_to_user_id, expanded_urls)
3. incorrect Dtype string for "timestamp" column
4. invalid entries in "name" column such as (None, a).

5. rating_numerator is of type int, does not allow for decimals.
6. rating_denominator with min value of 0, and values less than 10.

### image_prediction:

7. Missing image data (2075 entries instead of 2356)

### tweet_json_df:

8. Missing data (2354 entries instead of 2356)

## Tidiness Issues:

1. Dog types stored as multiple features (doggo, floofer, pupper, puppo)
2. Data divided across 3 dataframes

# Cleaning

In this phase, each of the previously detected issues was handled to get a clean version of the data. The cleaning steps are described below, each issue is listed along with the approach applied to solve it.

## Solving Tidiness Issues

### 1. Dog types stored as multiple features (doggo, floofer, pupper, puppo)

All 4 columns were merged into one column named "dog_stage"

### 2. Data divided across 3 dataframes

All dataframes were be merged into one dataframe.

## Solving Quality Issues

### 1. Irrelevant Data (retweets data included)

Remove rows about retweets.

### 2. Unneeded columns (retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp, in_reply_to_status_id, in_reply_to_user_id, expanded_urls)

Unneeded columns were removed using pandas drop function.

### 3. incorrect Dtype string for "timestamp" column

Convert timestamp column to datetime datatype using pandas to_datetime function.

### 4. invalid entries in "name" column such as (None, a).

Ideally, invalid entries should have been replaced, using pandas replace function, by NAs, or other valid entries extracted from the data. Yet, in our case, it was decided to drop the whole column as it does not add much value to the analysis later. Accordingly, 'name' column was dropped.

### 5. rating_numerator is of type int, does not allow for decimals.

Change type to float to allow for decimals.

### 6. rating_denominator with min value of 0, and values less than 10

Remove entries with denominator less than 10.

### 7. Missing image data (2354 entries instead of 2356)

Remove rows with no image, i.e keeping only rows were jpg_url is not null.

### 8. Missing data (2354 entries instead of 2356)

Remove missing entries (this was already achieved along with the previous step).

# Storing Data

Finally, the clean version of data was stored in a CSV format using to_csv function and saved to a file named twitter_archive_master.