# We Rate Dogs Wrangle Report

**By Marie-Luise Klaus**
20/04/2019 Berlin, Germany
**for internal use**

## 1. Introduction

This report describes the wrangling efforts of the We Rate Dogs data analysis project within the Udacity Data Analysis Nanodegree program. The data used refers to the WeRateDogs Twitter account, which humorously introduces dogs.

https://twitter.com/dog_rates/ (https://twitter.com/dog_rates/)

Below, we will introduce the steps throughout the wrangling process - gathering, assessing and cleaning data. Further, we will go briefly into detail about particularities to this data set. The wrangling itself can be found in the wrangling section of *'wrangle_act.ipynb'*.

## 2. Gathering Data

The data was gathered from 3 different sources using different methodologies.

### 2.1 WeRateDogs Twitter Archive

Udacity provided a file *'twitter-archive-enhanced.csv'*, containing information about WeRateDogs tweets. We read in the csv file into a pandas dataframe: *'twitter_archive'*.

### 2.2 Image Predictions Data

We programatically downloaded a tsv using Pythons request library from

https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv (https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv).

The file was provided by Udacity and contains predictions about dog breeds occurring in the WeRateDog tweets, which was created using a neural network. We saved the file locally as *'image-predictions.tsv'* and loaded it into a pandas dataframe '*image_predictions'* thereafter.

### 2.3 Twitter API Data

The third source of data comes from Twitter's API. We created a Twitter developer's account and used the Python library Tweepy for the API requests. To locate relevant tweets, we used tweet_ids from the twitter archive. During the request, we missed 19 tweets due to deleted tweets. Subsequently, we saved the JSON-formatted tweet data to a file *'tweet_json.txt'*, line-by-line. We continued by reading in our file and parsing the data into a pandas dataframe *'tweets_df'*.

## 3. Assessing Data

This section involved assessing data for quality and tidiness issues. We started by visually assessing our data sheets in Jupyter Notebook and Excel. To get a first understanding we used methodologies like .info() and .describe().
Thereafter, we got into detail by programmatically assessing dataframe by dataframe.

**Issues**
The assessment section ends with a documentation of unclean data issues found during assessment. 12 quality issues and 2 tidiness issues aroused. The quality issues spread across all 3 dataframes. However, the Twitter archive appears to be the dirtiest.

We found irrelevant tweets that didn't serve the purpose of the account, which is to rate dogs using a specific pattern. Mostly, corrupt replies were retweets and replies to other tweets, or tweets not related to any dog content. We also found missing data as one of the main issues. Dogs were not named, wrongly or not classified into their 'stage' or incorrectly rated. There were incorrect datatypes and impractical formats. Regardless of us addressing 12 content issues throughout the cleaning process, there are more issues that could be cleaned, depending on the desired outcome of the analysis.

On the side of tidiness/structural issues, we found that dog stages were split in multiple stages across the Twitter archive instead of being in one column. To make a comprehensive analysis possible, we found it made sense to have all 3 datasets available in one.

## 4. Cleaning Data

In this section we took a programmatic approach to clean our data from quality and tidiness issues found during assessment. We began by copying each dataframe to work with in oder to preserve the original data. Thereafter, we began cleaning each issue taking three steps:

1. **Define** how to clean up the issue,
2. Write **Code** to fix the issue and then
3. **Test** whether the intended result was successful without any unwanted side effects. In the course of this, we took a test-driven development approach (TDD), meaning we took step 3 before step 2.

We started by fixing the tidiness issues first. As a result we generated a combined dataframe *master_df*, which was then used to fix quality issues.

The assessment and cleaning of the dataframe were not straightforward. While cleaning, we found further issues, which made us go back and forth between assessment to cleaning processes.

## 5. Conclusion

As a result of the wrangling, we removed corrupt, inaccurate and unnecessary record in preparation for the following analysis and visualization. The clean master dataframe was stored in a the file *'twitter_archive_master.csv'*. By the end of the cleaning process, we managed to keep

1670 out of the 2356 records, which was sufficient for the purpose of our analysis and visualization.

Despite our efforts and the iterative process of wrangling, we see potential for further cleaning of the data. One critical point that we did not manage to clean, was it's up-to-dateness. The youngest tweet is almost 2 years old and it would have been interesting to find out the WeRateDogs account performed over a longer time frame.

In general, the cleaner a dataframe, the more accurate are the final conclusions. Data wrangling processes aim to prepare it's data, depending on it's desired analysis. And for our analysis the wrangling resulted in sufficient data to find interesting conclusions.