# WRANGLING AND ANALYZE DATA

Project 4 Report

Dalal Bin Gheshiyan

## INTRODUCTION

This wrangle report is part of the Wrangle and Analyze Data project to document the project's wrangling efforts. The dataset used in this project for wrangling, analyzing, and visualizing is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. WeRateDogs is a Twitter account that rates people's dogs with a humorous comment about the dog. The wrangle report documents the three steps of data wrangling: gathering, assessing, and cleaning data.

## STEP 1: GATHERING DATA

to complete this project, I need to gather data from multiple sources with different data formats, which required different methods.

1. The WeRateDogs Twitter archive. The file was provided by the project and can be downloaded directly from the Udacity website, This archive contains basic tweet data (tweet ID, timestamp, text, etc.)
2. The tweet image predictions. The file is hosted on Udacity's servers. I downloaded this file programmatically by using the Requests library in Python, which is a table full of image predictions alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction.
3. Get retweets count and favorite count information missing from the Twitter archive from another file. I chose to download the tweet JSON file programmatically by using the Requests library, as Twitter decline my request for using their Tweepy - Twitter API-.

## STEP 2: ASSESSING DATA

After gathering all three pieces of data, I assessed data visually and programmatically for quality and tidiness issues.

### QUALITY:

issues with content. Low-quality data is also known as dirty data.

1. Completeness
2. Validity
3. Accuracy
4. Consistency

### TIDINESS:

issues with the structure that prevent easy analysis. Untidy data is also known as messy data.

1. Each variable forms a column.
2. Each observation forms a row.
3. Each type of observational unit forms a table.

## QUALITY ISSUES

### DF_ARCHIVE_ENHANCED TABLE:

1. retweeted_status_id, retweeted_status_user_id and retweeted_status_timestamp are useless columns.
2. Erroneous datatypes ( timestamp ) should be of type date.
3. Erroneous datatypes(floofer, pupper and puppo) should be category data type.
4. Erroneous datatypes (tweet_id, in_reply_to_status_id, in_reply_to_user_id, and columns) should be of type object (string).
5. Unreal rating_denominator values (Equal 0).
6. Unreal rating_numerator values( Equal 1776).
7. Missing information for the dog stages.

### IMAGE_PREDICTION TABLE

1. Erroneous datatypes (tweet_id) should be of type object (string).
2. Some rows have the value false for p1_dog, p2_dog, and p3_dog so they are not dogs.

### TWEET_JSON TABLE

1. Erroneous datatypes (tweet_id) should be of type object (string).

## TIDINESS ISSUES

1. df_archive_enhanced, image_prediction, and tweet_json tables should be in one table.
2. doggo, floofer, pupper and puppo should be in one Column.

## STEP 3: CLEANING DATA

Cleaning data is the third step in data wrangling. where I fixed the quality and tidiness issues that were identified in the assessment step, I used programmatic data cleaning methods.

### THE PROGRAMMATIC DATA CLEANING PROCESS:

1. Define: convert our assessments into defined cleaning tasks.
2. Code: convert those definitions to code and run that code.
3. Test: test the dataset, visually or with code, to make sure the cleaning operations worked.

## THE FLOWING CLEANING ACTIONS I DID:

1. remove retweets Columns using isnull and drop method, only want original ratings (no retweets).

2. Convert the timestamp column's data type from an object to a DateTime using astype, and remove the '0000' using string slicing.

3. use drop method to drop columns that have the value false for p1_dog, p2_dog, and p3_dog.

4. Use the drop method to drop rows that have the value rating_denominator not equal to 10.
5. Use drop method to rows columns that have the value rating_numerator greater than 19.

6. n the df_archive_enhanced_clean table, image_prediction_clean table and tweet_json_clean table, change the dtype of column tweet_id, from int64 to object, using the astype() function.

7. Use the drop method to drop rows that have none value for doggo, floofer, puppera, and puppo columns.

8. Use pd.melt to melt the doggo, floofer, pupper and puppo columns to a type and dog_stage column. Drop the intermediate column.

9. In the df_archive_enhanced_clean table, change the type of column dog_stage, from object to categorical data type, using the astype() function.

10. Take both the df_archive_enhanced_clean and image_prediction_clean tables and merge them into one table using the join() method on the columns tweet_id, then merge it with the twitter_image_tweet_json table.