# Wrangle and Analyze Data Project

## Introduction.

The dataset that i wrangled (and 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. These ratings almost always have a denominator of 10. The numerators.

## Project Details

- Gathering Data
- Assessing Data
- Cleaning Data
- Visualization and Analyze Data

## What Software Do I Used?

- Pandas
- Numpy
- Matplotlib
- OS
- Tweepy
- Requests
- Json

## Gathering Data.

In this phase i gather 3 different types of files

1. **Twitter archive** :- the twitter_enhanced_archive.csv was provided on UDACITY classroom
2. **Tweet image prediction** :- based on neural networks  a table full of image predictions (the top three only) alongside each tweet ID, image URL, and the image number that corresponded to the most confident prediction (numbered 1 to 4 since tweets can have up to four images).
3. **Twitter API & JSON :-** this step was a little bet challenging to me but after searching on web for solutions I did it by using tweets IDs in tweeter archive I queried The tweeter API for each tweet's JSON Data using python's library tweepy then save the tweet in file called tweet_json.txt, then I read the entire file line by line in a pandas DataFrame with (tweet_id, favorite_count, retweet_count, foloowers_count, source, retweet_status)

## Assessing Data.

I did this phase in to steps

1. Visual assessment

This done by printing the three data frame on jupyter notebook and looking for problems

2. Programming assessment

This done by using pandas methods (info() / value_counts() / sample() / duplicated() …ect ).

Then I separated the problems into Quality and Tidiness

## Cleaning Data

The magic happened here…

First of all I created a copy of three DataFrames

The challenge in this phase was in many steps one of the in

Image_prediction data frame when I created a function to catch the largest value of prediction confident to filter the prediction process into only two columns  then pass it to apply method to apply in the whole dataset.

And another one when I melt the dogs_stage into one column