# DATA WRANGLING REPORT

## -MEET PATEL

## 1. Gathering the Data :

- We have used the following three pieces of data in a Jupyter Notebook titled wrangle_act.ipynb:

  a. The “WeRateDogs” Twitter archive.
     The file has been downloaded manually.
     Link : data/twitter_archive_enhanced.csv

  b. The Tweet Image Predictions.
     This file (image_predictions.tsv) is hosted on Udacity's servers and was downloaded programmatically using the Requests library and the following URL:
     https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

  c. Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API for each tweet's JSON data using Python's Tweepy library and store each tweet's entire set of JSON data in a file called tweet_json.txt file.

## 2. Assessing the Data :

- ARCHIVE

  a. The values corresponding to "tweet_id" column, currently present as 'INT' values, have to be of 'string' data types.
  b. The rows need to be suppressed if the value is a non-null to conserve only original tweets for the column data corresponding to : "in_reply_to_status_id", "in_reply_to_user_id", "retweeted_status_id", "retweeted_status_user_id", "retweeted_statud_timestamp".

c. The source contains html residues.
d. Some entries are missing expanded URL's.
e. The variables doggo, floofer, pupper and puppo present in the data as depicted above, all represent one single variable and as per the tidyness rule, should be a part of a single column.
f. We can observe that there are some 'None' objects present in "doggo", "floofer", "pupper" and "puppo" which have to be convert into 'NaN'.
g. 14 rows are in two categories
h. 'None' object in "doggo", "floofer", "pupper" and "puppo" have to be converted into 'NaN'.
i. The "timestamp" is present as an object, which is supposed to be converted into datetime to be exploitable.
j. The "name" column has multiple stop words present.
k. As evident from above, the "name" column has a total of 745 'None' strings present.
l. We can clearly visualize from above that a total of 639 double links are present within "expanded_urls" column data.
m. We can see that a total of 23 ratings do not have the denominator value as 10, i.e. these many values are not rated out of 10.
n. Thus, we can see that a total of 440 entries have a Numerator value less than 10.
o. It is pretty evident from the above result that we are getting incorrect or false data corresponding to the Numerators, when it has a Decimal Value, for all numerators in "rating_numerator" column.

- IMAGE

a. The values corresponding to "tweet_id" column, currently present as 'INT' values, have to be of 'string' data types.
b. We can observe that the columns "p1", "p2" and "p3" possess inconsistent writings.
c. As seen above, for a total of 324 rows in the Data, no dogs are recognized.

- INFO

a. The 'id' has to be of 'string' data-type.

- GLOBAL

a. The len of each document should be same.

b. We need to have one dataframe.

## 3. Cleaning the Data

- <u>ARCHIVE</u>

a. Suppressed the rows with retweet and unnecessary columns

- This was done by suppressing the rows with values as non-null entities to conserve only the original tweets for the following columns: “in_reply_to_status_id”, “in_reply_to_user_id”, “retweeted_status_id”, “retweeted_status_user_id”, “retweeted_statud_timestamp”.

b. HTML residues were suppressed in the Source.
c. Changed the data-type of “timestamp” to datetime.
d. Changed the data-type of “tweet_id” to string.
e. Suppressed stop-words in “name”.
f. Converted the values in “name” column to lowercase.

g. Altered the 'None' valued strings to ‘NaN’.
- We observed that the “name” field had 745 ‘None’ string.
- The data corresponding to “doggo”, “floofer”, “pupper” and “puppo” had ‘None’ string.

h. Double-links and missings URLs were corrected by modifying or generating URLs
- We found a total of 639 double links in “expanded_urls”.
- We had various entries with missing expanded URLs.

i. Fixed the inappropriate Numerators and Denominators
- The entries were validated when no problem of scrapping existed(irrational).
- The entries were changed when scrapping error(including decimal) prevailed.

j. Dogs stage
- Created a single variable.
- Corrected double stage when is needed.

- <u>IMAGE</u>

a. Changed the data-type of “tweet_id” to ‘string’.

b. Fixed the p1, p2, p3 which had inconsistent writings.
c. No dogs were recognized in 324 entries.
   - to observe impact = tweet sympathetic, so just a column

- INFO

a. Changed the data-type of id to ‘string’.

- GLOBAL

a. Maintained the same length for each document.
b. Maintained a single data-frame at the end,
   - Merged and suppressed rows without image.
c. The data was stored in 'data/twitter_archive_master.csv'.

## 4. Storing Data and Reports

- The cleansed data is stored in 'data/twitter_archive_master.csv'.

## *5. References*

- Panda column with loop: https://chrisalbon.com/python/data_wrangling/pandas_create_column_with_loop/
- Suppress HTML: https://stackoverflow.com/questions/13682044/pandas-dataframe-remove-unwanted-parts-from-strings-in-a-column
- from try/except: https://wiki.python.org/moin/HandlingExceptions
- Replace column: https://github.com/pandas-dev/pandas/issues/9106
- Stop-words: https://martinapugliese.github.io/english-stopwords/