Reporting: wragle_report

- Create a 300-600 word written report called "wrangle_report.pdf" or "wrangle_report.html" that briefly describes your wrangling efforts. This is to be framed as an internal document.

- This report contains a description of the data wrangling efforts employed in this project

- The data used in this process is the Twitter Archive, containing the the tweet archive of twitter user @ dog rates, popularly known as WeRateDogs.
- This twitter account rates people's dogs with use of humorous comments about the dog.
- The ratings almost always have a denominator of 10 i.e 11/10, 12/10, 13/10, etc. WeRateDogs has over 4 million followers.
- This project was completed outside the udacity worksapce, but was conducted suing the support materials provided for by Udacity. The reports were also generated outside of Udacity workspcae using the Jupyter notebook and converted to PDF format.
- The wrangling process is divided into 3 main stages -:
  ```
  (i) Gathering data
  (ii) Assessing data
  (iii) Cleaning data
  ```

and finally creating data analysis and visualizations.

(a) Gathering Data

The data was gathered from three different sources namely:

i. Enhanced Twitter Archive. This contains data that was gathered programmatically and uploaded into the Udacity's dashboard.It was downloaded manually by clicking the following link, twitter_archive_enhanced.csv(provided link), once downloaded, it was uploaded and read into a pandas DataFrame.

- The data extracted contains ratings, dog name, dog stage(doggo, floofer, pupperm and puppo amd filtered to contain 2356 tweets out out of 5000+ tweets. Only the tweets with ratings were filtered.

ii. Image Predictions File

- This data was produced by running every image in the WeRateDogs twitter archive through a neural netwrok that classifies breed of Dogs. The results were a table full of image predictions(top three only) alomgside each tweet ID, image URL, and the image that corresponded to the most confident prediction(extracted from Udacity description of the dataset)

- This file is hosted in the udacity servers. It was downloaded programmatically using requests library and the following URL: https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv

iii). Additional Data via twitter API

- This data was obtained through querying twitter's API then stored in a text file; tweet-json.

- This analysis however used the already gathered data provided in the udacity workspace. The ready data was read into a pandas DataFrame with tweet ID, retweet count and favorite count, and saved into a 'image_pred_df.csv'

(b) Assessing Data.

- Data assessment for the three datasets was conducted visually and programmatically using methosds such as (.info. .head, .sample, etc).
- After a thorough assesment, the following findings were made: (a) Tidiness issues

  (i) Dog stages were separated into four different columns(doggo, floofer, pupper and puppo) (ii) All the data required for the analysis was related but split into three different dataframes.

  (b) Quality Issues (i) Enhanced Twitter Archive

  - Some dog names have invalid names(None, a, an , the instead of names or NaN)
  - 181 retweets available
  - Invalid data type for timestamp(string instead of date time)
  - Invalid data type for tweet ID(integer instead of string)
  - columns(doggo, floofer, pupper, puppo) have 'None' instead of NaN for missing values (ii) Image predictions
  - Underscores used to separate names, in columns p1, p2, and p3. Spaces should have been used.
  - Some P names start with uppercase letters while others start with lowercase letters.

    (iii) Data from twitter API
  - id column different from the other two datasets -missing entries

(c) Cleaning data

- The issues identified were addresed using appropriate methods resulting to a high quality and tidy Pandas DataFrame.
- The clean data is further used for data analysis and visualizations.

In [1]:
```
from subprocess import call
call(['python', '-m', 'nbconvert', 'wrangle_act-report.ipynb'])
```

Out[1]:
```
1
```