**Wrangle report**

As an Udacity Scholar, the Data Analyst Nanodegree Program from Udacity provided me with the incredible chance to go through the data wrangling process. The project focuses on everything I've learned, including acquiring data from many sources in various formats, analyzing data quality and tidiness, cleaning the data, and demonstrating my wrangling efforts through analysis and visualizations.

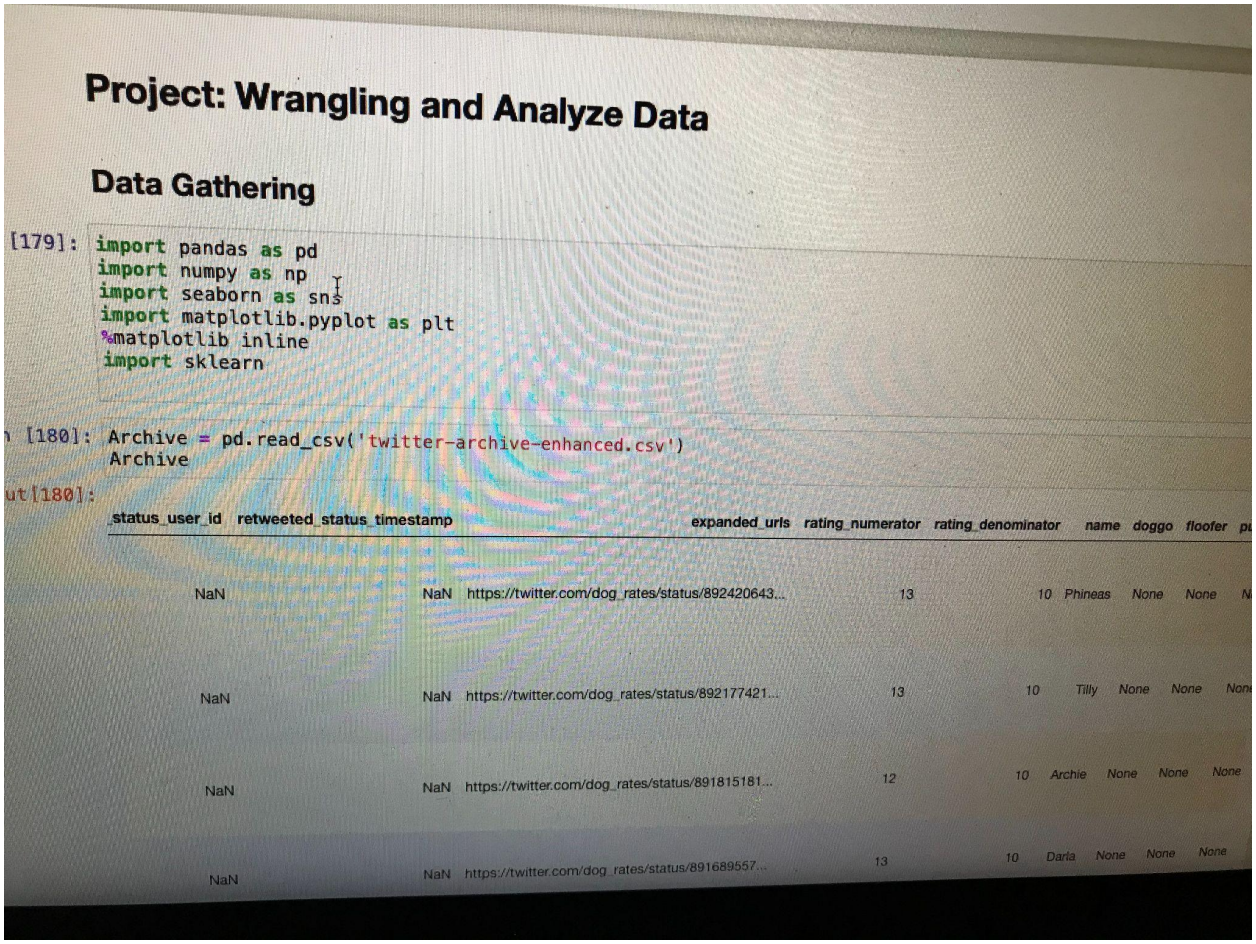
I wrangled and analyzed the tweet archive of WeRateDogs, better known as @dog rates on Twitter. WeRateDogs is a Twitter account that assesses people's dogs by making funny remarks about them, giving the dog a rating that is almost always more than 10, which is their rating denominator.

Brent is the name given to dogs who receive a rating of 10 or above.

What exactly is data wrangling, given that it is the core of this project?

Data wrangling is the process of cleaning, organizing, and translating raw data into the format requested by analysts for quick decision-making.

I began by importing the Python libraries and packages that would be needed for this research.



Project: Wrangling and Analyze Data

Data Gathering

```
[179]: import pandas as pd
       import numpy as np
       import seaborn as sns
       import matplotlib.pyplot as plt
       %matplotlib inline
       import sklearn
```

```
[180]: Archive = pd.read_csv('twitter-archive-enhanced.csv')
       Archive
```

Out[180]:

| status_user_id | retweeted_status_timestamp | | expanded_urls | rating_numerator | rating_denominator | name | doggo | floofer | pu |
|---|---|---|---|---|---|---|---|---|---|
| NaN | NaN | https://twitter.com/dog_rates/status/892420643... | | 13 | 10 | Phineas | None | None | N |
| NaN | NaN | https://twitter.com/dog_rates/status/892177421... | | 13 | 10 | Tilly | None | None | Non |
| NaN | NaN | https://twitter.com/dog_rates/status/891815181... | | 12 | 10 | Archie | None | None | None |
| NaN | NaN | https://twitter.com/dog_rates/status/891689557... | | 13 | 10 | Darla | None | None | None |

**Gathering the Data**

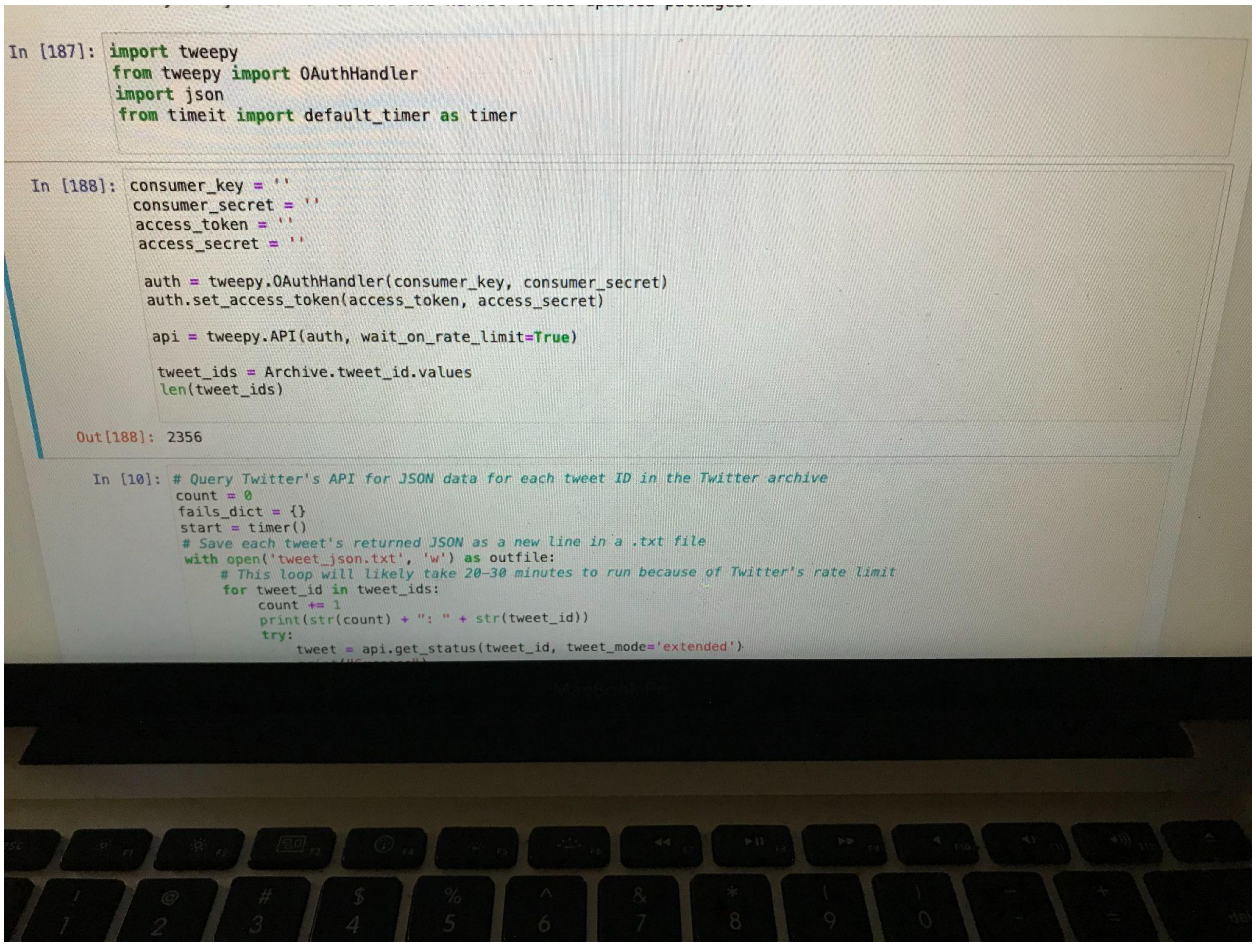
The data for this project came in three different formats:

1. Twitter Archive File from WeRateDogs: WeRateDogs downloaded their Twitter archive and sent it to Udacity via email exclusively for use in this project. This archive contains basic tweet data (tweet ID, timestamp, text, etc.) for all 5000+ of their tweets as they stood on August 1, 2017.

2. Image Prediction File: The photographs in the WeRateDogs Twitter archive (the first dataset above) were processed by a neural network capable of classifying dog breeds. This file contains the picture forecasts. This TSV file was hosted on Udacity's server and was downloaded programmatically using the Python Requests library's URL.

Requests is a Python HTTP library with a wide range of uses. One of its applications is to use the URL to download or open a file from the internet.

3. Tweeter API JSON File: Using the tweet IDs from the WeRateDogs Twitter archive, I queried the Twitter API for each tweet's JSON data with Python's Tweepy package and saved each tweet's whole set of JSON data in a file named tweet json.txt. The JSON data for each tweet was written to its own line. Then I read the.txt file into a pandas DataFrame line by line.



```
In [187]: import tweepy
          from tweepy import OAuthHandler
          import json
          from timeit import default_timer as timer

In [188]: consumer_key = ''
          consumer_secret = ''
          access_token = ''
          access_secret = ''

          auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
          auth.set_access_token(access_token, access_secret)

          api = tweepy.API(auth, wait_on_rate_limit=True)

          tweet_ids = Archive.tweet_id.values
          len(tweet_ids)

Out[188]: 2356

In [10]: # Query Twitter's API for JSON data for each tweet ID in the Twitter archive
         count = 0
         fails_dict = {}
         start = timer()
         # Save each tweet's returned JSON as a new line in a .txt file
         with open('tweet_json.txt', 'w') as outfile:
             # This loop will likely take 20-30 minutes to run because of Twitter's rate limit
             for tweet_id in tweet_ids:
                 count += 1
                 print(str(count) + ": " + str(tweet_id))
                 try:
                     tweet = api.get_status(tweet_id, tweet_mode='extended')
```

Tweepy is an open-source Python package that gives you a very convenient way to access the Twitter API with Python. You can find more details on setting up an app in Twitter and accessing Twitter API using Python.

And that enabled me to get the retweet count, favorite count, e.t.c of each tweet.

**Assessing the Data**

The three data points have been collected and correctly analyzed. I looked for quality and tidiness concerns during the assessment.

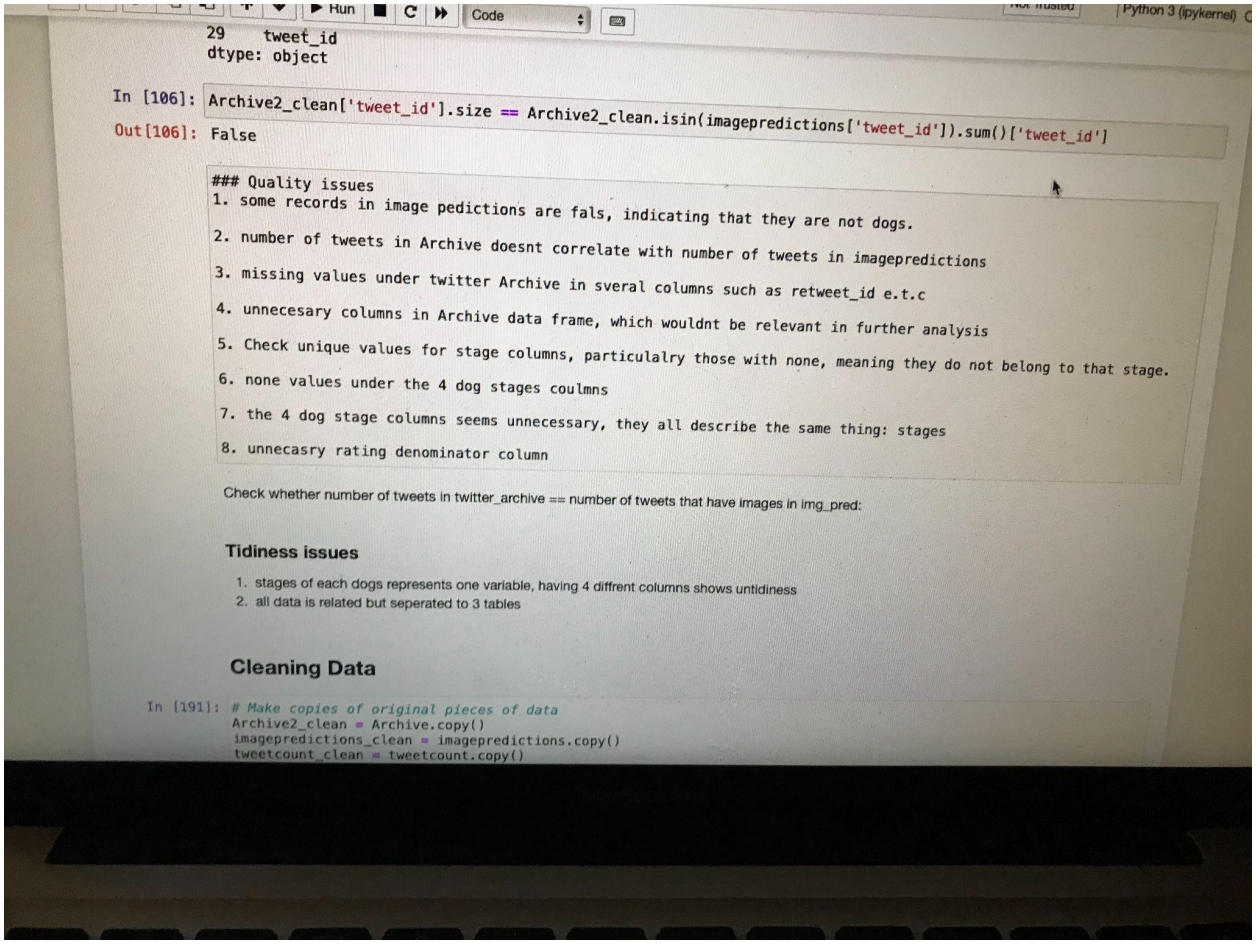
Quality: Low-quality data is also known as filthy data. The substance of dirty data is problematic. Completeness, Validity, Accuracy, and Consistency are the Data Quality Dimensions.

Untidiness: Untidiness is sometimes referred to as "messy" data. Messy data has structural difficulties.

1. Each variable becomes a column in tidy data.
2. Each observation is arranged in a row.
3. A table is formed by each sort of observational unit.

After physically inspecting the data in DataFrames and Excel spreadsheets, as well as programmatically inspecting the three DataFrames separately, I discovered eight quality concerns and two tidiness issues in total, which I recorded in my Python notebook.

.




**Cleaning the Data**

During the assessment step, I cleaned up all of the flaws I documented. It's important to understand that cleaning the data does not imply changing its format. It simply implies increasing the data's quality so that it may be used. To increase its quality and tidiness, the data is cleansed.
I cleaned the data using the Define, Code, and Test cleaning steps.

Define: the concerns discovered during the evaluation are transformed into cleaning chores.
Cleaning tasks are turned into code and then executed.
To ensure that my cleaning efforts were successful, I employed codes.

Keeping the Data
After cleaning the data, I aggregated the three cleaned datasets using a common attribute, the Twitter id, and saved the master dataset as 'twitter archive master.csv'.

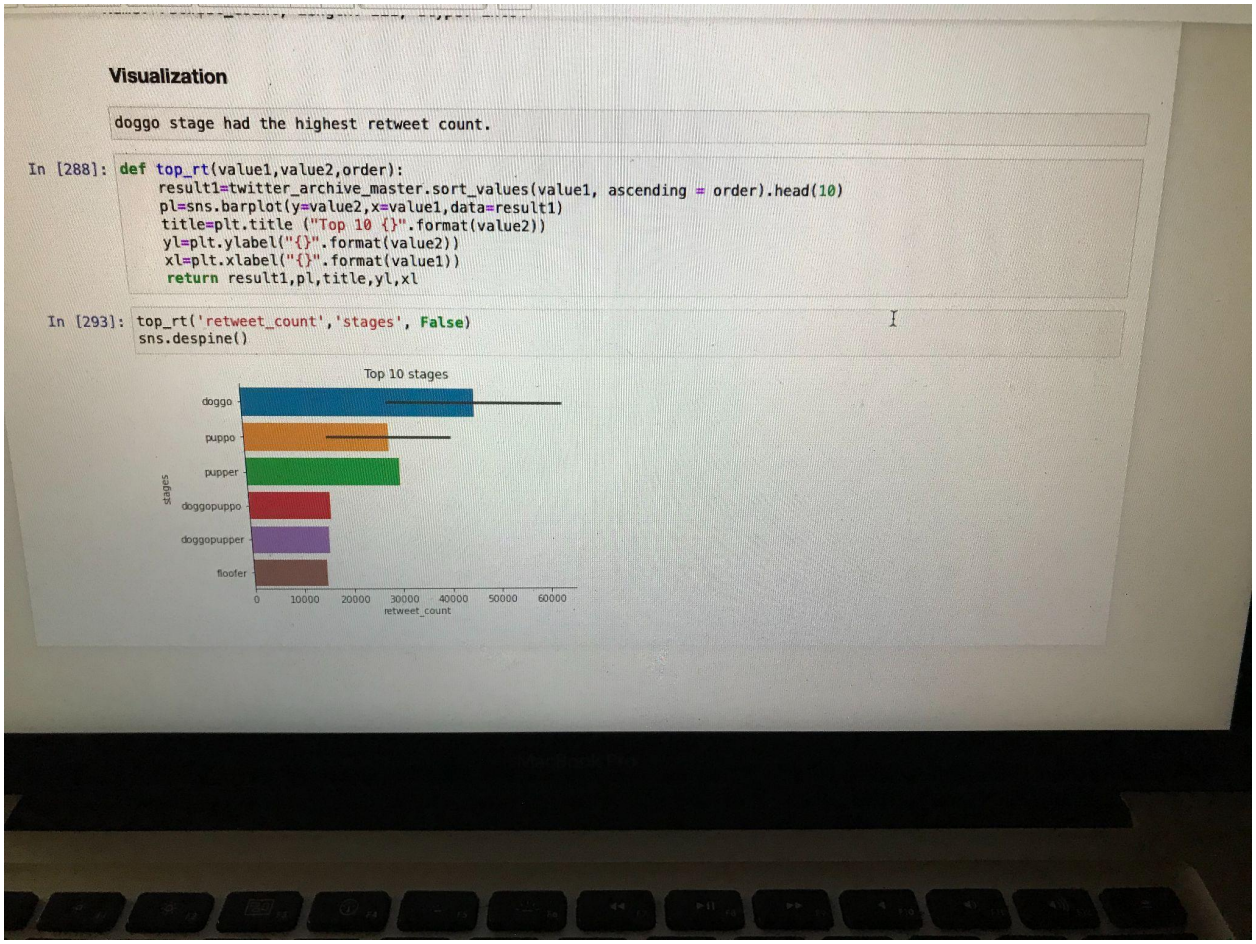
**Analysis and Visualizations**
The data has been collected, evaluated, and cleaned, and it is now available for analysis. I asked the data several questions in order to gain insights from it.

According to the distribution of dog photos, the most common dog stage is 'pupper' (a little doggo, generally younger), followed by 'doggo' and 'puppo.'

Let me explain what these dog phases imply.
A pupper is a tiny, generally younger dog. Can be as mature as, if not more mature than, certain doggos.
A doggo is a large, generally elderly pupper. It looks to be in control of its life. Probably knows about taxes and such.
Puppo: A puppo is a stage between a pupper and a doggo. Easily understood as the canine equivalent of an adolescent.

So, according to our study, the majority of the image forecasts were at the pupper stage. It might be because a young and immature dog is typically cuter than an adult dog, which is why most people buy, adopt, and own them. It should also be noted that there is a large amount of missing data in the master dataset's dog stage column, thus the distribution may not be accurate.

We can observe that the pupper dog stage has the highest amount of retweet count making it the people favorite doggo stage.




**Conclusion**

This dataset may be subjected to additional analysis and visualization, however because data wrangling is the primary emphasis of this project, more effort was spent on that area.

I used my jupyter notebook to submit two reports for my Udacity project. The first report is named 'wrangle report,' and it explains your wrangling efforts concisely. This report is also designed to be used internally. The second report, known as the 'act report,' presents all of the insights and displays the visuals generated by the wrangled data. This second report is formatted as an external document, such as a blog post or magazine article.

This was an intriguing and thrilling assignment for me. It has improved my abilities as a data wrangler. I've been able to utilize Python libraries to manipulate data, and I'm excited to see what projects I'll be working on in the future.