Wrangling Report on the WeRateDogs Twitter Archive Using Twitter's API.

Introduction:

Data Wrangling is the most important part of the Data Analysis because of what it entails. The Data Wrangling process is iterative and non-linear. The Data Wrangling process is as follows:

- **Data Gathering**: In this step, the data is gathered through different methods. For my analysis, I got one file from manual download, another was downloaded programmatically and the final one, which was the crux of the whole project, was gotten via Twitter's API. There are other ways to get data but I'll limit it to these for the sake of the project.

```
In [9]: %%time  # To calculate the time it will take to run the code
        failed = {}  # To store the failed retrievals
        step = 0

        # Opening the file in write mode
        with open(API_FILE, "w") as file:
            # Looping through each id
            for ids in tweet_ids:
                step += 1
                if step % 5 == 0:
                    print(f"{step}: {ids}")

                # Fetching the data
                try:
                    tweet = api.get_status(ids, tweet_mode='extended')
                    json.dump(tweet._json, file)
                    file.write('\n')

                # Storing the errors
                except tweepy.errors.TweepyException as tte:
                    failed[ids] = tte
                    pass

        print(failed)
        len(failed)
```

Fig 1: API Data Gathering

- **Data Assessing**: This has to do with finding issues with the data you have gathered. The issues may be quality issues or tidiness issues. For this project I identified eight quality issues and two tidiness issues.

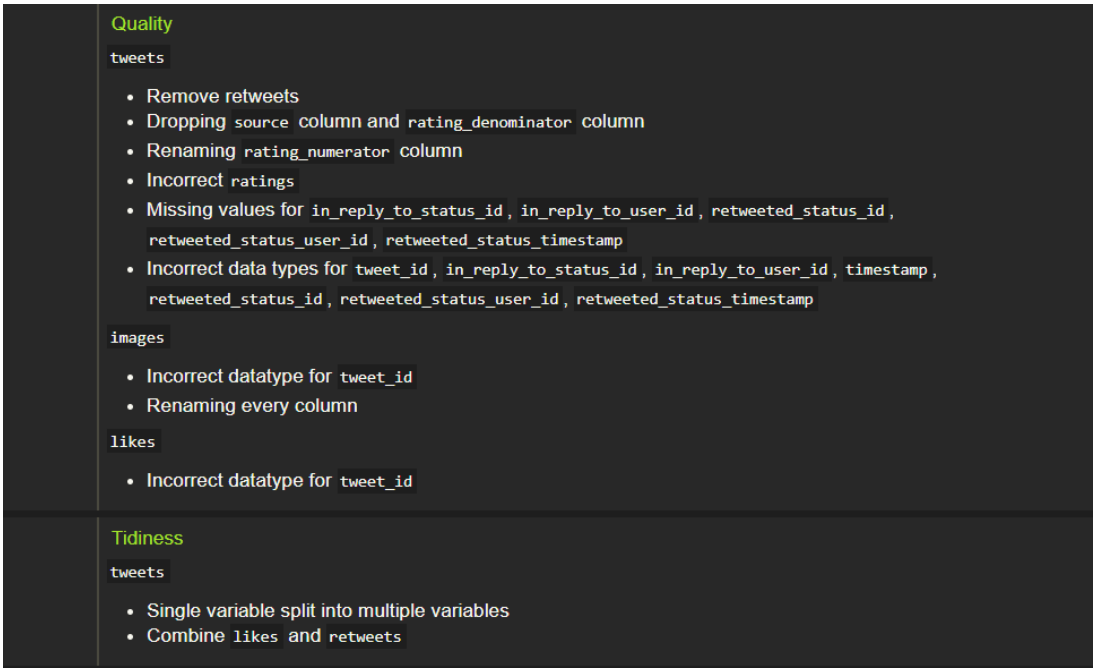

Fig 2: Quality and Tidiness Issues

- **Data Cleaning**: In this aspect, every issue identified above will have to be resolved. For this project, the *define-code-test* framework was implemented.

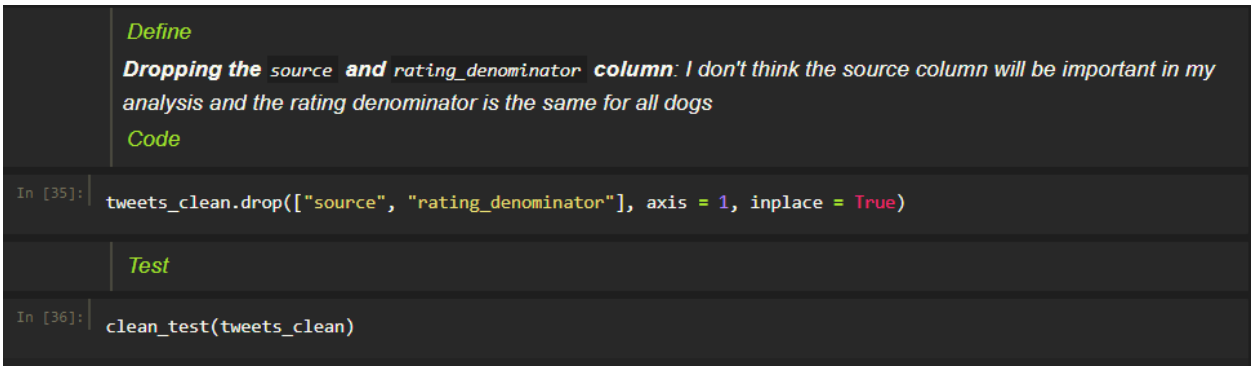

Fig 3: Define-Code-Test

Case Study & Data Set:

The Case Study for this project was @dog_rates twitter account. They rate dogs based on humour level, with a common denominator of 10 and varying levels of numerator. One of the intrigue of this account is that the ratings can go way higher than the denominator.

Analysis:

- The first thing I noticed was that the amount of retweets increased over time.

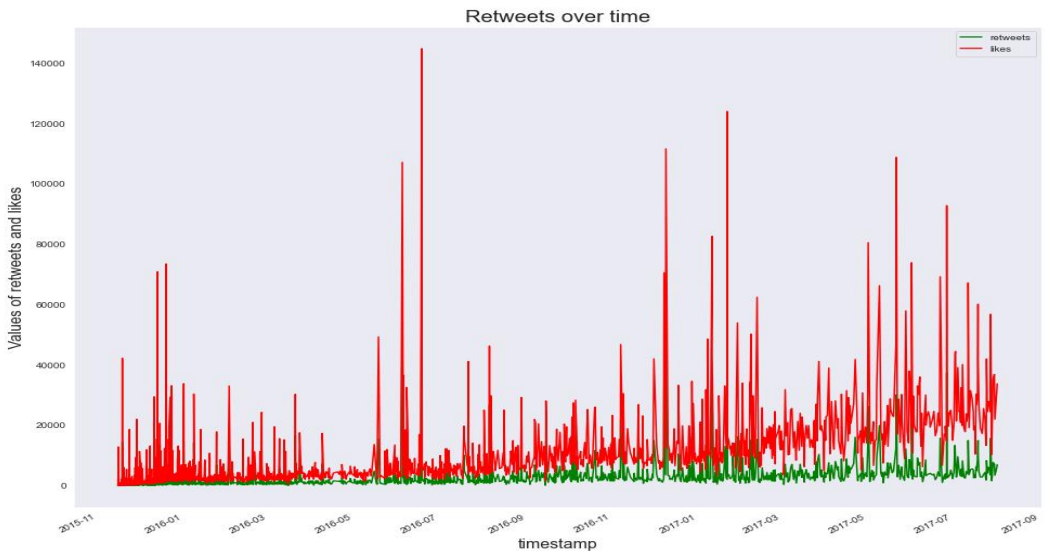

Fig 4: Retweets over Time

- As this project is about dogs, I had to find out which dog had the most likes and found it to be: https://twitter.com/dog_rates/status/744234799360020481 which, unsurprisingly, also had the highest retweets.
- Also the lowest liked, and retweeted, dog was: https://twitter.com/dog_rates/status/666102155909144576

- I also found out that there was a high correlation between the likes and retweets with a value of *0.92840227679343 78*