Using Data Analysis techniques to Analyse the given Twitter user @dog_rates data set.

1. Introduction

The dataset is the is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs, its a Twitter account that rates people's dogs with a humorous comment about the dog. These ratings almost always have a denominator of 10 and the numerators is always greater than 10. 11/10, 12/10, 13/10. It has over 4 million followers and has received international media coverage.

My goal is to wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations.

2. Methods

2.1 Data Collection

For this analysis I used tweet data for all 4000+ of @dog_rates tweets, Also, get the tweets data using the tweet IDs in the WeRateDogs Twitter archive and query the Twitter API for each tweet's JSON data using Python's Tweepy library.

2.1 Data Accessing

For this Analysis, I used Pandas dataframe to read the data from the CSV file provided and also query the Twitter API for each tweet's data using Python's Tweepy library. While Accessing the data I found out some missing values tidyness issues which I cleaned and also verifed the quality of the data. This data set consist of Dogs Tweet information that I used to analysis with variables like tweet_id source, text
expanded_urls rating_numerator
rating_denominator
name

From the tweet's text column we need to extract rating, dog name, and dog "stage" (i.e. doggo, floofer, pupper, and puppo).

There are some missing values in the data set for [in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp] so we don't need these in our anaysis.

```
In [4]: tweets_data1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

Cleaning Sequences

There are multiple ways of sequencing the data cleaning process. I followed the Define, Code, and Test Markdown headers were used once in this sequence, with multiple definitions, cleaning operations, and tests under each header, respectively. I found 8 Quality issues and 3 tidiness issues that I tried to clean and shown in the wrangle_act.ipynb file.

```
In [ ]:
```