1. Data Wrangling

In the wrangling process three main operations were performed. First; downloading the tweeter data and predictions of dog breeds. Second, assessing the data for quality and tidiness issues. Finally; creating a master dataset which was used in the analysis phase.

1.1. Gathering Data

In this project, gathering data involved:

- Manually downloading WeRateDogs' Twitter archive,
- Programmatically downloading missing properties using Twitter's API for Python, and
- Programmatically downloading image predictions using Requests.

1.2. Assessing Data

The process of finding problems from the collected data involved the use of visual and programmatic techniques. The goal was to detect and document at least 8 quality and 2 tidiness issues. Quality issues found at this stage were grouped in to completeness, validity, accuracy, and consistency categories.

1.2.1. Quality Issues in the Twitter Archive

- Invalid column datatypes.
    - Id columns (such as tweet_id, and in_reply_to_status_id) were int and float instead of string.
    - Timestamp columns (such as timestamp and retweeted_status_timestamp) were string instead of datetime.
- Inconsistent and inaccurate dog names.
    - Some dog names (such as 'a', and 'such') were in lowercase, while others were in uppercase.
    - All lowercase names were inappropriate.
    - Few names (such as O'Malley and Al Cabone) were represented by their beginning characters.
- Inaccurate dog stages.
    - Null dog stages were represented using the string None instead of Python's None.

    - In some tweets, multiple stages were used to represent a single dog.
- Inaccurate ratings.
    - Notations (such as 24/7, 9/11, 7/11, 4/20 (April 20), and 11/15/15) in some tweets were wrongly considered as ratings,
    - Decimal rating (such as 9.75, 11.27, and 11.26) in some tweets were not correctly retrieved,
    - Some rating-like notations (such as 3 1/2, and 50/50) were wrongly considered as ratings.

1.2.2. Quality Issues in the Json Data

- 179 retweets, which are not useful, were extracted through the API.
- Inconsistent tweet id datatypes.
    - In tweet_json, tweet ids were represented as strings, but twitter_archive used int.
- Incomplete records.
    - Twitter archive had 2356 records, but tweet_json had 2354.

1.2.3. Quality Issues in the Image Predictions

- Inconsistent tweet id datatypes.
    - In tweet_json, tweet ids were represented as strings, but image_predictions used int.
- Incomplete records.
    - Twitter archive had 2356 records, but image_predictions had 2075.

1.2.4. Tidiness Issues

- Doggo, floofer, pupper, and puppo Variables could be converted to dog_stage variable.
- Tweet json and image_predictions could be merged with the twitter_archive.

1.3. Cleaning Data

Fixing problems, which had been seen during the assessment stage, was started by making copies of the data. Then, each of the listed subsequent steps followed.

- Defining the plan for addressing the issues,
- Coding to fix the issues,

- Testing if the issues ware addressed properly, and
- Documenting the effort.

1.3.1. Fixing Quality Issues

To fix the quality issues mentioned above, the following built-in methods were applied.

- `DataFrame.copy()` was used to make copies of dataframes.
- `DataFrame.astype()` and `Series.astype()` were used to convert variables from one datatype to another.
- `pandas.to_datetime()` was used to convert variables to datetime.
- `Series.str.islower()` was used to get lowercase names.
- `Series.str.extractall()` was used in combination with regular expressions to extract ratings from tweet texts.
- `Series.notna()` and `Series.isna()` were used to detect non-missing and missing values respectively.
- `numpy.nan` and python `None` were used to represent empty records.

Other issues were addressed using techniques such as indexing and slicing.

1.3.2. Fixing Tidiness Issues

- `DataFrame.melt()` was used to convert multiple variables into rows, with their corresponding values.
- `pandas.merge()` was used to combine multiple dataframes into one.

1.4. Storing Data

The cleaned dataset was intended to be used for performing analysis and visualizations. This requires storing it permanently either in a file or database. In this project, the clean master dataset was written to a CSV file.