Wrangling report

1. Gathering the data

First of all, I gathered all the data from the data sources. It was quite easy, but I had some troubles with storing the data coming from Twitter API (because json output had different json top-level keys for different requests).

I though though that it is a bit weird that we have to store the entire set of JSON data for each tweet since we do not use it later. I would rather only store the elements we use (tweet ID, retweet count, and favorite count). If we really want the whole json per tweet (I am not sure why though), I would store each tweet information in a separate file in a json format (since json output per tweet varies quite a bit). Anyway, I was following the instructions (as I understood them) and find the solution quite dirty: I used all possible json top-level keys as columns and the corresponding values as strings (which were sometimes dictionaries, which I forced into a string format).

2. Assessing the data

To asses the data quality, I implemented the following checks:

- Is tweet_id unique in each of the dataframes?
- Check whether twitter_archive dataset contains retweets (in_reply_to_status_id is not null)
- Check whether all dataframes contain the same number of tweet_id's
- Check for which tweets the image is not available
- Check whether name of the dog is extrated correctly in twitter_archive dataframe
- Check whether rating_numerator & rating_denominator are extrated correctly in twitter_archive dataframe
- Check how doggo/floofer/pupper/puppo columns are formed in twitter_archive dataframe; can a dog be at the same time both doggo and floofer (or another combination)?

Found issues:

Data quality issues:

1. twitter_archive dataframe contains 259 retweets
2. Dogs names in twitter_archive dataframe contain stopwords like 'an'/'a'/'such'/'quite'/'the')
3. Dogs names in twitter_archive dataframe are missing when could be extracted
4. Dogs names in twitter_archive dataframe are stored as string 'None' if missing instead of Python None
5. Rating numerator and denominator in twitter_archive dataframe are not always extracted correctly
6. The same dog can be assigned to two different stages (puppo/doggo); it looks also more like a dummy variable with 0/1 as possible values
7. If value in columns doggo/floofer/pupper/puppo is not present, it is stored as string 'None' instead of Python None
8. In image_predictions dataframe, prediction can be written in different ways (with capital letter/ lower case)

9. Timestamp in twitter_archive is not stored in datetime format; create separate columns for date & time & week nr for further visualizations
10. Source in twitter_archive is not parsed correctly (still contains html)
11. In image_predictions dataframe, sometimes there is no single prediction per tweet that is dog

Data tidiness issues:

1. Stages of dog´s growth should be in one column instead of 4 columns (because 4 columns actually represent one variable).
2. image_predictions dataframe should have one prediction per tweet_id -> need to find the way to merge them
3. The same observation is in multiple tables (all collected dataframes should be merged; we will loose some observations in this way)
4. Column expanded_urls in twitter_archive contains multiple records (comma separated) per observation (see examples under)

I solved all the issues mentioned above and created one final dataframe containing all the information.