Gather

The gatherings consisted of downloading and importing a csv file, a tsv file and a json-like txt file with retweets and favorite counts.

The twitter-archive-enhanced.csv I directly downloaded from a link in the course material. I downloaded the breed image prediction tsv-file programmatically accessing an url..

Then I was supposed to connect to the Twitter API to access the last data needed. But my request was never filed. The other code for connecting to the API did not work either. I tried several times. Eventually I got the tweet-json.txt file instead and first tried to load it as a normal json file. But it is a text file, so after some help from a mentor I managed to read in the file line by line getting the keys.

Assessing

I started with opening in Google Sheets and based on the notes that we are only looking for original tweets with images before August 2017 which has an image I actually did following pre-clean before assessing.

1. Merge df_enhanced with df_images and with df_retweetfav - removes tweets with no image and rows with no match
2. Remove all tweets not starting with "This is.."
3. Remove all tweets before 1st of August 2017

I went on assessing the full table by calling .head() and .info() . I also checked for duplicated. After this I did visual assessment and found following quality issues

1. There is 2 columns with only one value in each
2. There are 3 empty columns (on the border of perhaps being a tidiness issue, but I also regard it as a consistency, validation issue)
3. The dog names are sometimes just a random word
4. Not so descriptive column names
5. There are links in text cells
6. +0000 in timestamp can be cleaned out
7. The rating has odd numbers sometimes. They seem to capture the wrong value from the text.
8. Breed names are mixed lower and uppercase and inconsistent use of dashes

Note on more than we have e.g. the fact that not all images are dogs and some odd peaks in confidence intervals for the probability numbers. The list can go on but now focus on the above. I have ignored some columns like source and img_num which do not necessarily add value for the query to examine original tweets and images.

After this I did tidiness assessment:

1. Create one column for breed combining the p1-p3
2. 28 columns!, let's create one dataset with interesting things, 10 columns WeRateDogs_tweets = name + breed + image + WeRateDogs_rating + text + (number_of_retweets) + (number_of_likes) + link_to_tweet + timestamp + tweet_id

Cleaning

Missing data

→ Removed row where index 1891 with drop function

→ Removed columns in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id and retweeted_status_user_id with drop function

Tidiness things

→ Created one column for breed combining the p1-p3 where if true in p1_dog it add that value, if false in p1_dog but true in p2_dog add that value, if false look in p3_dog and if true add that value, else print - 'not a dog'. Using np.select which I found here

→ Created a new dataframe with columns from the big df: name + breed + image + WeRateDogs_rating + text + retweet_count + favorite_count + link_to_tweet + timestamp + tweet_id

Quality issues

→ If the word after "This is" is a word starting with a capital letter than fill the cell with that value, else print the string 'No Name'

→ Renamed jpg_url' to 'image', 'rating_numerator'to 'weratedogs_rating' and 'expanded_urls': to 'link_to_tweet'}

→ Remove all strings starting with 'https' in text string

→ Used datetime to make time a bit more readable

→ I will try with another method to catch the rating

→ Used title and replace method to fix lowercase and dashes