Gathering data:

The wrangling exercise for this project began with gathering the various files for this project namely:

1. twitter-archive-enhanced.csv
2. image_predictions.tsv
3. tweet_json.txt

However, the said files were not provided as a direct download with the exception of the first and even with the first it was still necessary to unzip the contents using the zipfile module which of course has to be first imported into the Jupyter notebook environment.

For the second file, namely, image_predictions.tsv, this file has to be downloaded from the internet. To accomplish this, a folder was created name ‘folder’ and with the help of the requests module and the given URL this file was finally downloaded programmatically.

For the third file, namely, tweet_json.txt, this file was downloaded by querying Twitter’s API. To successfully query the api I had to ensure the following guidelines were followed:

1. Setup an application with Twitter, which I already had previously.
2. Obtain a consumer_key and consumer_secret as well as an access_token_secret pin.
3. It was necessary to follower Twitter’s set guidelines to avoid a revocation of my license in that case I had to specify the following limit:
   a. api = tweepy.API(auth,wait_on_rate_limit=True, wait_on_rate_limit_notify=True)

The tweets from Twitter were then downloaded and saved in the tweet_json.txt file.

Reading in files:

The files were then read into a pandas dataframe. For the first file the pd.read_csv function was used.

For the second file I had to use BeautifulSoup to parse the content of the response.content into a file handle and assigned it to a variable ‘soup’ of course this was after having extracted the last content using pandas split function.

The said last content (being the image_predictions.tsv) of the html file returned by BeautifulSoup was then written as a binary file into a file and the file was read into a pandas dataframe with the ‘sep’ parameter set to =’\ t’ because the file is a tap separated file as against the first which was a comma separated variable.

For the third file, namly, tweet_json.txt, a pandas dataframe was first created to hold the columns then the json files were appended .

Visual assessment was made by looking at the file through the pandas dataframe and sublime text editor.

The following quality issues were identified:

1. Null values for the following columns: in_reply_to_status_id, in_reply_to_user_id, retweeted_status_id, retweeted_status_user_id, retweeted_status_timestamp.
2. timestamp column contains +0000 which looks redundant
3. Incorrect representation for name in the name column such as a, an, None etc.

Then programmatic assessments were carried out with the following codes respectively:

```
df.info()

image_predictions.info()

tweet_json.info()

df.source.value_counts()

df.name.value_counts()

df[df.name.duplicated()]
```

```
df.rating_denominator.value_counts()

df[df['rating_numerator']<10]

df.rating_numerator.value_counts()

df.doggo.value_counts()

df.floofer.value_counts()

df.pupper.value_counts()

df.puppo.value_counts()

df.loc[(df[['doggo', 'floofer', 'pupper', 'puppo']] != 'None').sum(axis=1) > 1]
```

on the execution of the above codes some quality and tidiness issues were revealed.

cleaning exercise was then carried out on the inspected dataset and the dataset was tested at each successful cleaning exercise to verify the integrity of the cleaning exercise. To limit this report to the specified maximum word count of 600 I will not supply the codes used for both cleaning and testing but can be accessed from the main project document titled ‘wrangle_act.ipynb’