October 2018
Michael Ciaccio
Udacity Data Analysis Nanodegree candidate

The following Jeff Leek referenced and quoted in the Udacity Data Wrangling Course influenced my approach.
https://simplystatistics.org/2016/02/17/non-tidy-data/
In other words, the goal is to solve a particular problem and the format I chose is the one that makes it most direct/easy to solve that problem, rather than one that is theoretically optimal.

JAVA classes and programming languages or libraries are all examples of the encapsulation of logical units of work.

I am submitting multiple Jupyter notebooks. Each Jupyter notebook encapsulates a logical unit of work. These individual Jupyter notebooks are called or invoked from the 'parent' wrangle_act.ipynb. Requirements compliance is achieved by calling or invoking the functional Jupyter notebooks from the required wrangle_act.ipynb Jupyter notebook.

Segmenting lengthy and complex algorithms into discreet logical units of work is compliant with best practices.

- wrangle_act.ipynb.ipynb
    - twitter_archive_enhanced_csv_proj_submit.ipynb
    - maturity_stage_tidy_proj_submit.ipynb
    - image_predictions_tsv_proj_submit.ipynb
    - image-pred_twitter-arch-common_tweet_id_proj_submit.ipynb
    - tweepy_real_time_get_proj_submit.ipynb
    - t_archive_images_tweepy_merge.ipynb
    - insights_ visualizations.ipynb

twitter_archive_enhanced_csv_proj_submit.ipynb

Create DataFrame - twitter_archive_enhanced_df
from the provided WeRateDogs tweeter archive.

Clean - remove retweet columns
Clean - tweet_id from dtype('int64') to object
Clean - in_reply_to_status_id from dtype('float64') to dtype('O')
Clean - in_reply_to_user_id from dtype('float64') to dtype('O')
Clean - timestamp from non-null object to datetime64[ns]

maturity_stage_tidy_proj_submit.ipynb

Consolidate doggo, floofer, pupper, puppo columns into 1 Tidy compliant pandas Series, column - maturity

image_predictions_tsv_proj_submit.ipynb

instantiated pandas DataFrame based on image-predictions.tsv
image-predictions.tsv obtained using Python import requests library

Clean - tweet_id from non-null int64 to object

Requirements compliance -

- Data Analyst Nanodegree Program
- 8. Data Wrangling
- Project
- 2. Project Motivation
- Key Points
- You only want original ratings ... that have images.

Rather than naively assume jpg_url(s) were valid, I opted to verify the url. I interrogated HTTP status codes identifying unreachable url(s). In addition I identified and managed spurious and often non reproducible network related Python exceptions using the try clause paired with multiple except clauses.

image-pred_twitter-arch-common_tweet_id_proj_submit.ipynb

Preparation for pending Tidy required, tweet_id driven merge of 3 pandas DataFrames. Identify tweet_id(s) common to both the twitter_archive_enhanced_clean_df and image_predictions_df_clean DataFrames.

Clean - retain only rows with common tweet_id(s) in both DataFrames

tweepy_real_time_get_proj_submit.ipynb

Real time Twitter get.
Write api.get_status to tweet_json.txt.
Capture "No status found with that ID" tweet_id(s)
Managed spurious Exceptions with Python try - except clauses.
Create pandas DataFrame from saved tweet_json.txt.

Synchronize tweet_id(s) in all 3 DataFrames in preparation for DataFrame merge.

- twitter_archive_common_df
- image_common_df
- retrieved_tweet_data_df

t_archive_images_tweepy_merge.ipynb

Tidy driven merge 3 DataFrames

- image_common_df
- twitter_archive_common_df,
- retrieved_tweet_data_df

insights_visualizations.ipynb

extract tweet_hour from datetime pandas Series
populate tweet_hour pandas Series
generate 4 plots for act_report.pdf