

Project: Wrangling and Analyze Data

```
In [1]: import sqlite3
import urllib

from configparser import ConfigParser
from io import BytesIO
from itertools import islice
from json import dump, loads
from os import environ
from pathlib import Path

import requests
import tweepy

import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from IPython.display import clear_output, display, Image

sns.set_theme(style="darkgrid")
%matplotlib inline
```

We'll store all downloaded and generated files in this directory. We create it if it doesn't exist.

```
In [2]: data_dir = Path("data")
data_dir.mkdir(exist_ok=True)
```

Table of Contents

- [Data Gathering](#)
- [Assessing Data](#)
 - [twitter_archived_enhanced_df](#)
 - [image_predictions_df](#)
 - [tweet_json_txt_df](#)
- [Quality issues](#)
- [Tidiness issues](#)
- [Cleaning Data](#)
- [Storing Data](#)
- [Analyzing and Visualizing Data](#)

Data Gathering

In the cell below, gather **all** three pieces of data for this project and load them in the notebook. **Note:** the methods required to gather each data are different.

1. Directly download the WeRateDogs Twitter archive data
(twitter_archive_enhanced.csv)

We have to assume this file was downloaded manually, we'll first check if it exists, and if it doesn't, we'll download it and open it as if it was already on the filesystem.

```
In [3]: twitter_archived_enhanced_csv = data_dir / "twitter-archive-enhanced.csv"
```

```
In [4]: if not twitter_archived_enhanced_csv.exists():
        with requests.get(
            "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_t
            stream=True
        ) as response:
            response.raise_for_status()
            with open(twitter_archived_enhanced_csv, "wb") as f:
                for chunk in response.iter_content(chunk_size=32768):
                    f.write(chunk)
```

Before we open the file, let's check the first few lines to determine the proper way of open it with pandas:

```
In [5]: with open(twitter_archived_enhanced_csv, "r") as f:
        for line in islice(f, 5):
            print(line)
```

tweet_id,in_reply_to_status_id,in_reply_to_user_id,timestamp,source,text,retweeted_status_id,retweeted_status_user_id,retweeted_status_timestamp,expanded_urls,rating_numerator,rating_denominator,name,doggo,floofer,pupper,puppo

892420643555336193,,,2017-08-01 16:23:56 +0000,"Twitter for iPhone","This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU,,,https://twitter.com/dog_rates/status/892420643555336193/photo/1,13,10,Phineas,None,None,None,None

892177421306343426,,,2017-08-01 00:17:27 +0000,"Twitter for iPhone","This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13/10 https://t.co/0Xxu7lqeIV",,,,https://twitter.com/dog_rates/status/892177421306343426/photo/1,13,10,Tilly,None,None,None,None

891815181378084864,,,2017-07-31 00:18:03 +0000,"Twitter for iPhone","This is Archie. He is a rare Norwegian Pouncing Corgi. Lives in the tall grass. You never know when one may strike. 12/10 https://t.co/wUnZnhtVJB,,,https://twitter.com/dog_rates/status/891815181378084864/photo/1,12,10,Archie,None,None,None,None

891689557279858688,,,2017-07-30 15:58:51 +0000,"Twitter for iPhone","This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qLQ,,,https://twitter.com/dog_rates/status/891689557279858688/photo/1,13,10,Darla,None,None,None,None

There's nothing out of the ordinary, so we can just use `pandas.read_csv` as usual:

```
In [6]: twitter_archived_enhanced_df = pd.read_csv(twitter_archived_enhanced_csv)
```

```
In [7]: twitter_archived_enhanced_df.head(1)
```

```
Out[7]:
```

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href=""http://t/download/i

2. Use the Requests library to download the tweet image prediction (image_predictions.tsv)

Since we're hitting an external service, we first check if we haven't saved the loaded dataframe before. This isn't part of the wrangling process per se, but a convenience to avoid hitting external services multiple times while running the notebook.

```
In [8]: image_predictions_df = None
```

```
In [9]: image_predictions_df_pkl = data_dir / "image_predictions_df.pkl"
```

```
In [10]: if image_predictions_df_pkl.exists():
         image_predictions_df = pd.read_pickle(image_predictions_df_pkl)
```

If we haven't created the local pickled version, proceed with the expected download process:

```
In [11]: image_predictions_tsv_url = "https://d17h27t6h515a5.cloudfront.net/topher/20
```

```
In [12]: if image_predictions_df is None:
         # First we read the first few lines of the file to determine the right w
         with requests.get(image_predictions_tsv_url) as response:
             response.raise_for_status()

             for line in islice(
                 filter(lambda l: bool(l), response.iter_lines(decode_unicode=True
                     5
                 ):
                 print(line)
```

As we might have guessed by the file extension, this is a tab separated file with a header, which pandas can handle without any issues if the right `sep` value is used:

```
In [13]: if image_predictions_df is None:
         with requests.get(image_predictions_tsv_url) as response:
             response.raise_for_status()
             image_predictions_df = pd.read_csv(BytesIO(response.content), sep="\
             image_predictions_df.to_pickle(image_predictions_df_pkl)
```

Alternatively, we could have saved the file and then open it with pandas:

```
with requests.get(image_predictions_tsv_url) as response:
    response.raise_for_status()

    with open(data_dir / "image_predictions.tsv", "wb") as f:
        f.write(response.content)
```

```
image_predictions_df = pd.read_csv(data_dir /
"image_predictions.tsv", sep="\t")
```

Or even better, let pandas handle everything for us:

```
image_predictions_df = pd.read_csv(image_predictions_tsv_url,
sep="\t")
```

```
In [14]: image_predictions_df.head()
```

```
Out[14]:
```

	tweet_id	jpg_url	img_num	p1	p1_conf
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-IEu.jpg	1	Rhodesian_ridgeback	0.408143
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311

3. Use the Tweepy library to query additional data via the Twitter API (tweet_json.txt)

First we need to instantiate tweepy's API object. Notice that Twitter's credentials need to be set in config.ini :

```
In [15]: config = ConfigParser()
config.read("config.ini");
```

```
In [16]: assert "twitter" in config.sections()
assert {'consumer_key', 'consumer_secret', 'access_token', 'access_token_secret'} == set(config["twitter"].keys())
assert bool(config["twitter"]["consumer_key"])
assert bool(config["twitter"]["consumer_secret"])
assert bool(config["twitter"]["access_token"])
assert bool(config["twitter"]["access_token_secret"])
```

```
In [17]: auth = tweepy.OAuthHandler(
    config["twitter"]["consumer_key"],
    config["twitter"]["consumer_secret"],
    access_token=config["twitter"]["access_token"],
    access_token_secret=config["twitter"]["access_token_secret"]
)
api = tweepy.API(
    auth,
    wait_on_rate_limit=True
)
```

Given Twitter's API rate limits, we need to compute the rate at which we can safely fetch the tweet info based on the total number of tweets. As of this writing, we can only fetch 900 tweets every 15 minutes, so I need to wait at least 1 second $((15 * 60) / 900)$ between each request. Given that the total number of unique tweets is:

```
In [18]: total_no_tweets = twitter_archived_enhanced_df.tweet_id.nunique()
print(total_no_tweets)
```

2356

It'll take us about 40 minutes to fetch the tweet information.

```
In [19]: tweet_json_txt = data_dir / "tweet_json.txt"
```

```
In [20]: errors = [] # We'll keep track of errors

if not tweet_json_txt.exists():
    with open(tweet_json_txt, "w") as f:
        # We're iterating of unique tweet ids in case there are duplicates of
        for i, tweet_id in enumerate(twitter_archived_enhanced_df.tweet_id.unique()):
            tweet = None

            try:
                tweet = api.get_status(
                    tweet_id,
                    tweet_mode='extended' # Requesting extended tweets as per docs
                )

                #
                # Print progress
                #
                clear_output(wait=True)
                print(f"{i}/{total_no_tweets}: {tweet_id}")

                #
                # Notice that we're using a JSONL (JSON lines) format. This
                # the progress as we can tail tweet_json.txt while we're fetching
                #
                dump(tweet._json, f)
                f.write("\n")
            except Exception as e:
                print(f"Exception raised while fetching tweet status. tweet_id: {tweet_id}, error: {e}")
                errors.append((tweet_id, e, tweet))
```

Let's check how many errors were encountered during the fetch process:

```
In [21]: len(errors) # It might be 0 if we skipped fetching the tweet information
```

Out[21]: 0

Let's show the first few lines of `tweet_json.txt` to verify it was written correctly. We've used the JSONL (JSON lines) format they're easier to work with than streaming JSON data.

```
In [22]: with open(tweet_json_txt, "r") as f:
        for line in islice(f, 5): # Take only the first 5 lines
            print(line)
```

```

{"created_at": "Tue Aug 01 16:23:56 +0000 2017", "id": 892420643555336193,
"id_str": "892420643555336193", "full_text": "This is Phineas. He's a mysti
cal boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ
76dJU", "truncated": false, "display_text_range": [0, 85], "entities": {"ha
shtags": [], "symbols": [], "user_mentions": [], "urls": [], "media": [{"i
d": 892420639486877696, "id_str": "892420639486877696", "indices": [86, 10
9], "media_url": "http://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg", "media_u
rl_https": "https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg", "url": "http
s://t.co/MgUWQ76dJU", "display_url": "pic.twitter.com/MgUWQ76dJU", "expande
d_url": "https://twitter.com/dog_rates/status/892420643555336193/photo/1",
"type": "photo", "sizes": {"thumb": {"w": 150, "h": 150, "resize": "crop"},
"medium": {"w": 540, "h": 528, "resize": "fit"}, "small": {"w": 540, "h": 5
28, "resize": "fit"}, "large": {"w": 540, "h": 528, "resize": "fit"}}}}, "
extended_entities": {"media": [{"id": 892420639486877696, "id_str": "892420
639486877696", "indices": [86, 109], "media_url": "http://pbs.twimg.com/med
ia/DGKD1-bXoAAIAUK.jpg", "media_url_https": "https://pbs.twimg.com/media/DG
KD1-bXoAAIAUK.jpg", "url": "https://t.co/MgUWQ76dJU", "display_url": "pic.t
witter.com/MgUWQ76dJU", "expanded_url": "https://twitter.com/dog_rates/stat
us/892420643555336193/photo/1", "type": "photo", "sizes": {"thumb": {"w": 1
50, "h": 150, "resize": "crop"}, "medium": {"w": 540, "h": 528, "resize": "
fit"}, "small": {"w": 540, "h": 528, "resize": "fit"}, "large": {"w": 540,
"h": 528, "resize": "fit"}}}}], "source": "<a href=\"http://twitter.com/dow
nload/iphone\" rel=\"nofollow\">Twitter for iPhone</a>", "in_reply_to_statu
s_id": null, "in_reply_to_status_id_str": null, "in_reply_to_user_id": nul
l, "in_reply_to_user_id_str": null, "in_reply_to_screen_name": null, "use
r": {"id": 4196983835, "id_str": "4196983835", "name": "WeRateDogs\u00ae",
"screen_name": "dog_rates", "location": "all our links \u279c", "descriptio
n": "Your Only Source For Professional Dog Ratings \nInstagram and Facebook
\u279c WeRateDogs \npartnerships@weratedogs.com | nonprofit: @15outof10 \u2
800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800", "ur
l": "https://t.co/YPc2Xqmwyc", "entities": {"url": {"urls": [{"url": "http
s://t.co/YPc2Xqmwyc", "expanded_url": "http://links.weratedogs.com", "displ
ay_url": "links.weratedogs.com", "indices": [0, 23]}]}}, "description": {"ur
ls": []}}, "protected": false, "followers_count": 9339224, "friends_count":
21, "listed_count": 7566, "created_at": "Sun Nov 15 21:41:29 +0000 2015", "
favourites_count": 147205, "utc_offset": null, "time_zone": null, "geo_enab
led": true, "verified": true, "statuses_count": 16089, "lang": null, "contr
ibutors_enabled": false, "is_translator": false, "is_translation_enabled":
false, "profile_background_color": "000000", "profile_background_image_ur
l": "http://abs.twimg.com/images/themes/theme1/bg.png", "profile_background
_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "pr
ofile_background_tile": false, "profile_image_url": "http://pbs.twimg.com/p
rofile_images/1552995729014247425/TaJbIdmK_normal.jpg", "profile_image_url_
https": "https://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_
normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/4
196983835/1617810473", "profile_link_color": "F5ABB5", "profile_sidebar_bor
der_color": "000000", "profile_sidebar_fill_color": "000000", "profile_text
_color": "000000", "profile_use_background_image": false, "has_extended_pro
file": false, "default_profile": false, "default_profile_image": false, "fo
llowing": true, "follow_request_sent": false, "notifications": false, "tran
slator_type": "none", "withheld_in_countries": []}, "geo": null, "coordinat
es": null, "place": null, "contributors": null, "is_quote_status": false, "
retweet_count": 6978, "favorite_count": 33722, "favorited": false, "retweeted": false, "possibly_sensitive": false, "possibly_sensitive_appealable": false, "lang": "en"}

```

```
{"created_at": "Tue Aug 01 00:17:27 +0000 2017", "id": 892177421306343426,
"id_str": "892177421306343426", "full_text": "This is Tilly. She's just che
cking pup on you. Hopes you're doing ok. If not, she's available for pats,
snugs, boops, the whole bit. 13/10 https://t.co/0Xxu71qeIV", "truncated": f
alse, "display_text_range": [0, 138], "entities": {"hashtags": [], "symbol
s": [], "user_mentions": [], "urls": [], "media": [{"id": 89217741319462502
4, "id_str": "892177413194625024", "indices": [139, 162], "media_url": "htt
p://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg", "media_url_https": "https://p
bs.twimg.com/media/DGGmoV4XsAAUL6n.jpg", "url": "https://t.co/0Xxu71qeIV",
"display_url": "pic.twitter.com/0Xxu71qeIV", "expanded_url": "https://twitt
er.com/dog_rates/status/892177421306343426/photo/1", "type": "photo", "size
s": {"thumb": {"w": 150, "h": 150, "resize": "crop"}, "medium": {"w": 1055,
"h": 1200, "resize": "fit"}, "small": {"w": 598, "h": 680, "resize": "fi
t"}, "large": {"w": 1407, "h": 1600, "resize": "fit"}}}], "extended_entiti
es": {"media": [{"id": 892177413194625024, "id_str": "892177413194625024",
"indices": [139, 162], "media_url": "http://pbs.twimg.com/media/DGGmoV4XsAA
UL6n.jpg", "media_url_https": "https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.
jpg", "url": "https://t.co/0Xxu71qeIV", "display_url": "pic.twitter.com/0Xx
u71qeIV", "expanded_url": "https://twitter.com/dog_rates/status/89217742130
6343426/photo/1", "type": "photo", "sizes": {"thumb": {"w": 150, "h": 150,
"resize": "crop"}, "medium": {"w": 1055, "h": 1200, "resize": "fit"}, "smal
l": {"w": 598, "h": 680, "resize": "fit"}, "large": {"w": 1407, "h": 1600,
"resize": "fit"}}}], "source": "<a href=\"http://twitter.com/download/iphon
e\" rel=\"nofollow\">Twitter for iPhone</a>", "in_reply_to_status_id": nul
l, "in_reply_to_status_id_str": null, "in_reply_to_user_id": null, "in_repl
y_to_user_id_str": null, "in_reply_to_screen_name": null, "user": {"id": 41
96983835, "id_str": "4196983835", "name": "WeRateDogs\u00ae", "screen_nam
e": "dog_rates", "location": "all our links \u279c", "description": "Your O
nly Source For Professional Dog Ratings \nInstagram and Facebook \u279c WeR
ateDogs \npartnerships@weratedogs.com | nonprofit: @15outof10 \u2800\u2800\
\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800", "url": "http
s://t.co/YPc2XqmwYC", "entities": {"url": {"urls": [{"url": "https://t.co/Y
Pc2XqmwYC", "expanded_url": "http://links.weratedogs.com", "display_url": "
links.weratedogs.com", "indices": [0, 23]}]}, "description": {"urls": []}},
"protected": false, "followers_count": 9339224, "friends_count": 21, "liste
d_count": 7566, "created_at": "Sun Nov 15 21:41:29 +0000 2015", "favourites
_count": 147205, "utc_offset": null, "time_zone": null, "geo_enabled": tru
e, "verified": true, "statuses_count": 16089, "lang": null, "contributors_e
nabled": false, "is_translator": false, "is_translation_enabled": false, "p
rofile_background_color": "000000", "profile_background_image_url": "htt
p://abs.twimg.com/images/themes/theme1/bg.png", "profile_background_image_u
rl_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "profile_ba
ckground_tile": false, "profile_image_url": "http://pbs.twimg.com/profile_i
mages/1552995729014247425/TaJbIdmK_normal.jpg", "profile_image_url_https":
"https://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.j
pg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/419698383
5/1617810473", "profile_link_color": "F5ABB5", "profile_sidebar_border_colo
r": "000000", "profile_sidebar_fill_color": "000000", "profile_text_color":
"000000", "profile_use_background_image": false, "has_extended_profile": fa
lse, "default_profile": false, "default_profile_image": false, "following":
true, "follow_request_sent": false, "notifications": false, "translator_typ
e": "none", "withheld_in_countries": [], "geo": null, "coordinates": null,
"place": null, "contributors": null, "is_quote_status": false, "retweet_cou
nt": 5279, "favorite_count": 29251, "favorited": false, "retweeted": false,
"possibly_sensitive": false, "possibly_sensitive_appealable": false, "lan
g": "en"}
```


[illegible]

```
{"created_at": "Sun Jul 30 15:58:51 +0000 2017", "id": 891689557279858688,
"id_str": "891689557279858688", "full_text": "This is Darla. She commenced
a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qL
Q", "truncated": false, "display_text_range": [0, 79], "entities": {"hashta
gs": [], "symbols": [], "user_mentions": [], "urls": [], "media": [{"id": 8
91689552724799489, "id_str": "891689552724799489", "indices": [80, 103], "m
edia_url": "http://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg", "media_url_http
ps": "https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg", "url": "https://t.c
o/tD36da7qLQ", "display_url": "pic.twitter.com/tD36da7qLQ", "expanded_url":
"https://twitter.com/dog_rates/status/891689557279858688/photo/1", "type":
"photo", "sizes": {"thumb": {"w": 150, "h": 150, "resize": "crop"}, "smal
l": {"w": 510, "h": 680, "resize": "fit"}, "medium": {"w": 901, "h": 1200,
"resize": "fit"}, "large": {"w": 1201, "h": 1600, "resize": "fit"}}}], "ex
tended_entities": {"media": [{"id": 891689552724799489, "id_str": "89168955
2724799489", "indices": [80, 103], "media_url": "http://pbs.twimg.com/media
/DF_q7IAWsAEuuN8.jpg", "media_url_https": "https://pbs.twimg.com/media/DF_q
7IAWsAEuuN8.jpg", "url": "https://t.co/tD36da7qLQ", "display_url": "pic.twi
tter.com/tD36da7qLQ", "expanded_url": "https://twitter.com/dog_rates/status
/891689557279858688/photo/1", "type": "photo", "sizes": {"thumb": {"w": 15
0, "h": 150, "resize": "crop"}, "small": {"w": 510, "h": 680, "resize": "fi
t"}, "medium": {"w": 901, "h": 1200, "resize": "fit"}, "large": {"w": 1201,
"h": 1600, "resize": "fit"}}}], "source": "<a href=\\\"http://twitter.com/do
wnload/iphone\\\" rel=\\\"nofollow\\\">Twitter for iPhone</a>", "in_reply_to_stat
us_id": null, "in_reply_to_status_id_str": null, "in_reply_to_user_id": nul
l, "in_reply_to_user_id_str": null, "in_reply_to_screen_name": null, "use
r": {"id": 4196983835, "id_str": "4196983835", "name": "WeRateDogs\u00ae",
"screen_name": "dog_rates", "location": "all our links \u279c", "descriptio
n": "Your Only Source For Professional Dog Ratings \nInstagram and Facebook
\u279c WeRateDogs \npartnerships@weratedogs.com | nonprofit: @15outof10 \u2
800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800", "ur
l": "https://t.co/YPc2XqmwYC", "entities": {"url": {"urls": [{"url": "http
s://t.co/YPc2XqmwYC", "expanded_url": "http://links.weratedogs.com", "displ
ay_url": "links.weratedogs.com", "indices": [0, 23]}]}}, "description": {"ur
ls": []}}, "protected": false, "followers_count": 9339224, "friends_count":
21, "listed_count": 7566, "created_at": "Sun Nov 15 21:41:29 +0000 2015", "
favourites_count": 147205, "utc_offset": null, "time_zone": null, "geo_enab
led": true, "verified": true, "statuses_count": 16089, "lang": null, "contr
ibutors_enabled": false, "is_translator": false, "is_translation_enabled":
false, "profile_background_color": "000000", "profile_background_image_ur
l": "http://abs.twimg.com/images/themes/theme1/bg.png", "profile_background
_image_url_https": "https://abs.twimg.com/images/themes/theme1/bg.png", "pr
ofile_background_tile": false, "profile_image_url": "http://pbs.twimg.com/p
rofile_images/1552995729014247425/TaJbIdmK_normal.jpg", "profile_image_url_
https": "https://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_
normal.jpg", "profile_banner_url": "https://pbs.twimg.com/profile_banners/4
196983835/1617810473", "profile_link_color": "F5ABB5", "profile_sidebar_bor
der_color": "000000", "profile_sidebar_fill_color": "000000", "profile_text
_color": "000000", "profile_use_background_image": false, "has_extended_pro
file": false, "default_profile": false, "default_profile_image": false, "fo
llowing": true, "follow_request_sent": false, "notifications": false, "tran
slator_type": "none", "withheld_in_countries": [], "geo": null, "coordinat
es": null, "place": null, "contributors": null, "is_quote_status": false, "
retweet_count": 7198, "favorite_count": 36817, "favorited": false, "retweet
ed": false, "possibly_sensitive": false, "possibly_sensitive_appealable": f
alse, "lang": "en"}
```

```

{"created_at": "Sat Jul 29 16:00:24 +0000 2017", "id": 891327558926688256,
"id_str": "891327558926688256", "full_text": "This is Franklin. He would li
ke you to stop calling him \"cute.\" He is a very fierce shark and should b
e respected as such. 12/10 #BarkWeek https://t.co/AtUZn91f7f", "truncated":
false, "display_text_range": [0, 138], "entities": {"hashtags": [{"text": "
BarkWeek", "indices": [129, 138]}], "symbols": [], "user_mentions": [], "ur
ls": [], "media": [{"id": 891327551943041024, "id_str": "89132755194304102
4", "indices": [139, 162], "media_url": "http://pbs.twimg.com/media/DF6hr6A
VYAAZ8G8.jpg", "media_url_https": "https://pbs.twimg.com/media/DF6hr6AVYAAZ
8G8.jpg", "url": "https://t.co/AtUZn91f7f", "display_url": "pic.twitter.com
/AtUZn91f7f", "expanded_url": "https://twitter.com/dog_rates/status/8913275
58926688256/photo/1", "type": "photo", "sizes": {"medium": {"w": 720, "h":
540, "resize": "fit"}, "large": {"w": 720, "h": 540, "resize": "fit"}, "thu
mb": {"w": 150, "h": 150, "resize": "crop"}, "small": {"w": 680, "h": 510,
"resize": "fit"}}}], "extended_entities": {"media": [{"id": 89132755194304
1024, "id_str": "891327551943041024", "indices": [139, 162], "media_url": "
http://pbs.twimg.com/media/DF6hr6AVYAAZ8G8.jpg", "media_url_https": "http
s://pbs.twimg.com/media/DF6hr6AVYAAZ8G8.jpg", "url": "https://t.co/AtUZn91f
7f", "display_url": "pic.twitter.com/AtUZn91f7f", "expanded_url": "https://
twitter.com/dog_rates/status/891327558926688256/photo/1", "type": "photo",
"sizes": {"medium": {"w": 720, "h": 540, "resize": "fit"}, "large": {"w": 7
20, "h": 540, "resize": "fit"}, "thumb": {"w": 150, "h": 150, "resize": "cr
op"}, "small": {"w": 680, "h": 510, "resize": "fit"}}}], {"id": 891327551947
157504, "id_str": "891327551947157504", "indices": [139, 162], "media_url":
"http://pbs.twimg.com/media/DF6hr6BUMAAZgT.jpg", "media_url_https": "http
s://pbs.twimg.com/media/DF6hr6BUMAAZgT.jpg", "url": "https://t.co/AtUZn91f
7f", "display_url": "pic.twitter.com/AtUZn91f7f", "expanded_url": "https://
twitter.com/dog_rates/status/891327558926688256/photo/1", "type": "photo",
"sizes": {"medium": {"w": 720, "h": 540, "resize": "fit"}, "large": {"w": 7
20, "h": 540, "resize": "fit"}, "thumb": {"w": 150, "h": 150, "resize": "cr
op"}, "small": {"w": 680, "h": 510, "resize": "fit"}}}], "source": "<a href=\\\"http://twitter.com/download/iphone\\\" rel=\\\"nofollow\\\">Twitter for iPhon
e</a>", "in_reply_to_status_id": null, "in_reply_to_status_id_str": null, "
in_reply_to_user_id": null, "in_reply_to_user_id_str": null, "in_reply_to_s
creen_name": null, "user": {"id": 4196983835, "id_str": "4196983835", "nam
e": "WeRateDogs\u00ae", "screen_name": "dog_rates", "location": "all our li
nks \u279c", "description": "Your Only Source For Professional Dog Ratings
\nInstagram and Facebook \u279c WeRateDogs \npartnerships@weratedogs.com |
nonprofit: @15outof10 \u2800\u2800\u2800\u2800\u2800\u2800\u2800\u2800\u280
0\u2800\u2800\u2800", "url": "https://t.co/YPc2XqmwYC", "entities": {"url":
{"urls": [{"url": "https://t.co/YPc2XqmwYC", "expanded_url": "http://links.
weratedogs.com", "display_url": "links.weratedogs.com", "indices": [0, 2
3]}]}, "description": {"urls": []}}, "protected": false, "followers_count":
9339224, "friends_count": 21, "listed_count": 7566, "created_at": "Sun Nov
15 21:41:29 +0000 2015", "favourites_count": 147205, "utc_offset": null, "t
ime_zone": null, "geo_enabled": true, "verified": true, "statuses_count": 1
6089, "lang": null, "contributors_enabled": false, "is_translator": false,
"is_translation_enabled": false, "profile_background_color": "000000", "pro
file_background_image_url": "http://abs.twimg.com/images/themes/theme1/bg.p
ng", "profile_background_image_url_https": "https://abs.twimg.com/images/th
emes/theme1/bg.png", "profile_background_tile": false, "profile_image_url":
"http://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jp
g", "profile_image_url_https": "https://pbs.twimg.com/profile_images/155299
5729014247425/TaJbIdmK_normal.jpg", "profile_banner_url": "https://pbs.twim
g.com/profile_banners/4196983835/1617810473", "profile_link_color": "F5ABB

```

```
5", "profile_sidebar_border_color": "000000", "profile_sidebar_fill_color":  
"000000", "profile_text_color": "000000", "profile_use_background_image": f  
alse, "has_extended_profile": false, "default_profile": false, "default_pro  
file_image": false, "following": true, "follow_request_sent": false, "notif  
ications": false, "translator_type": "none", "withheld_in_countries": []},  
"geo": null, "coordinates": null, "place": null, "contributors": null, "is_  
quote_status": false, "retweet_count": 7722, "favorite_count": 35202, "favo  
ritied": false, "retweeted": false, "possibly_sensitive": false, "possibly_s  
ensitive_appealable": false, "lang": "en"}
```

The first few lines are ok. Let's load all of the lines onto a list.

```
In [23]: tweet_json_txt_json = []  
with open(tweet_json_txt, "r") as f:  
    tweet_json_txt_json = [  
        loads(line)  
        for line in f  
    ]  
tweet_json_txt_json[:5]
```

```

Out[23]: [{'created_at': 'Tue Aug 01 16:23:56 +0000 2017',
  'id': 892420643555336193,
  'id_str': '892420643555336193',
  'full_text': "This is Phineas. He's a mystical boy. Only ever appears in
the hole of a donut. 13/10 https://t.co/MgUWQ76dJU",
  'truncated': False,
  'display_text_range': [0, 85],
  'entities': {'hashtags': [],
  'symbols': [],
  'user_mentions': [],
  'urls': [],
  'media': [{'id': 892420639486877696,
    'id_str': '892420639486877696',
    'indices': [86, 109],
    'media_url': 'http://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg',
    'url': 'https://t.co/MgUWQ76dJU',
    'display_url': 'pic.twitter.com/MgUWQ76dJU',
    'expanded_url': 'https://twitter.com/dog_rates/status/8924206435553361
93/photo/1',
    'type': 'photo',
    'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
    'medium': {'w': 540, 'h': 528, 'resize': 'fit'},
    'small': {'w': 540, 'h': 528, 'resize': 'fit'},
    'large': {'w': 540, 'h': 528, 'resize': 'fit'}}}],
  'extended_entities': {'media': [{'id': 892420639486877696,
    'id_str': '892420639486877696',
    'indices': [86, 109],
    'media_url': 'http://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg',
    'url': 'https://t.co/MgUWQ76dJU',
    'display_url': 'pic.twitter.com/MgUWQ76dJU',
    'expanded_url': 'https://twitter.com/dog_rates/status/8924206435553361
93/photo/1',
    'type': 'photo',
    'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
    'medium': {'w': 540, 'h': 528, 'resize': 'fit'},
    'small': {'w': 540, 'h': 528, 'resize': 'fit'},
    'large': {'w': 540, 'h': 528, 'resize': 'fit'}}}],
  'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Tw
itter for iPhone</a>',
  'in_reply_to_status_id': None,
  'in_reply_to_status_id_str': None,
  'in_reply_to_user_id': None,
  'in_reply_to_user_id_str': None,
  'in_reply_to_screen_name': None,
  'user': {'id': 4196983835,
    'id_str': '4196983835',
    'name': 'WeRateDogs@',
    'screen_name': 'dog_rates',
    'location': 'all our links →',
    'description': 'Your Only Source For Professional Dog Ratings \nInstagra
m and Facebook → WeRateDogs \npartnerships@weratedogs.com | nonprofit: @15o
utof10',
    'url': 'https://t.co/YPc2XqmwYC',
    'entities': {'url': {'urls': [{'url': 'https://t.co/YPc2XqmwYC',

```

```
    'expanded_url': 'http://links.weratedogs.com',
    'display_url': 'links.weratedogs.com',
    'indices': [0, 23]}}},
    'description': {'urls': []}},
    'protected': False,
    'followers_count': 9339224,
    'friends_count': 21,
    'listed_count': 7566,
    'created_at': 'Sun Nov 15 21:41:29 +0000 2015',
    'favourites_count': 147205,
    'utc_offset': None,
    'time_zone': None,
    'geo_enabled': True,
    'verified': True,
    'statuses_count': 16089,
    'lang': None,
    'contributors_enabled': False,
    'is_translator': False,
    'is_translation_enabled': False,
    'profile_background_color': '000000',
    'profile_background_image_url': 'http://abs.twimg.com/images/themes/them
e1/bg.png',
    'profile_background_image_url_https': 'https://abs.twimg.com/images/them
es/theme1/bg.png',
    'profile_background_tile': False,
    'profile_image_url': 'http://pbs.twimg.com/profile_images/15529957290142
47425/TaJbIdmK_normal.jpg',
    'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1552995
729014247425/TaJbIdmK_normal.jpg',
    'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/
1617810473',
    'profile_link_color': 'F5ABB5',
    'profile_sidebar_border_color': '000000',
    'profile_sidebar_fill_color': '000000',
    'profile_text_color': '000000',
    'profile_use_background_image': False,
    'has_extended_profile': False,
    'default_profile': False,
    'default_profile_image': False,
    'following': True,
    'follow_request_sent': False,
    'notifications': False,
    'translator_type': 'none',
    'withheld_in_countries': [],
    'geo': None,
    'coordinates': None,
    'place': None,
    'contributors': None,
    'is_quote_status': False,
    'retweet_count': 6978,
    'favorite_count': 33722,
    'favorited': False,
    'retweeted': False,
    'possibly_sensitive': False,
    'possibly_sensitive_appealable': False,
    'lang': 'en'},
```

```
{'created_at': 'Tue Aug 01 00:17:27 +0000 2017',
  'id': 892177421306343426,
  'id_str': '892177421306343426',
  'full_text': "This is Tilly. She's just checking pup on you. Hopes you're
doing ok. If not, she's available for pats, snugs, boops, the whole bit. 13
/10 https://t.co/0Xxu71qeIV",
  'truncated': False,
  'display_text_range': [0, 138],
  'entities': {'hashtags': [],
  'symbols': [],
  'user_mentions': [],
  'urls': [],
  'media': [{'id': 892177413194625024,
    'id_str': '892177413194625024',
    'indices': [139, 162],
    'media_url': 'http://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg',
    'url': 'https://t.co/0Xxu71qeIV',
    'display_url': 'pic.twitter.com/0Xxu71qeIV',
    'expanded_url': 'https://twitter.com/dog_rates/status/8921774213063434
26/photo/1',
    'type': 'photo',
    'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
    'medium': {'w': 1055, 'h': 1200, 'resize': 'fit'},
    'small': {'w': 598, 'h': 680, 'resize': 'fit'},
    'large': {'w': 1407, 'h': 1600, 'resize': 'fit'}}}],
  'extended_entities': {'media': [{'id': 892177413194625024,
    'id_str': '892177413194625024',
    'indices': [139, 162],
    'media_url': 'http://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg',
    'url': 'https://t.co/0Xxu71qeIV',
    'display_url': 'pic.twitter.com/0Xxu71qeIV',
    'expanded_url': 'https://twitter.com/dog_rates/status/8921774213063434
26/photo/1',
    'type': 'photo',
    'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
    'medium': {'w': 1055, 'h': 1200, 'resize': 'fit'},
    'small': {'w': 598, 'h': 680, 'resize': 'fit'},
    'large': {'w': 1407, 'h': 1600, 'resize': 'fit'}}}],
  'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Tw
itter for iPhone</a>',
  'in_reply_to_status_id': None,
  'in_reply_to_status_id_str': None,
  'in_reply_to_user_id': None,
  'in_reply_to_user_id_str': None,
  'in_reply_to_screen_name': None,
  'user': {'id': 4196983835,
    'id_str': '4196983835',
    'name': 'WeRateDogs@',
    'screen_name': 'dog_rates',
    'location': 'all our links →',
    'description': 'Your Only Source For Professional Dog Ratings \nInstagra
m and Facebook → WeRateDogs \npartnerships@weratedogs.com | nonprofit: @15o
utof10',
    'url': 'https://t.co/YPc2Xqmwyc',
```

```
'entities': {'url': {'urls': [{'url': 'https://t.co/YPc2XqmwyC',
    'expanded_url': 'http://links.weratedogs.com',
    'display_url': 'links.weratedogs.com',
    'indices': [0, 23]}]}},
'description': {'urls': []}},
'protected': False,
'followers_count': 9339224,
'friends_count': 21,
'listed_count': 7566,
'created_at': 'Sun Nov 15 21:41:29 +0000 2015',
'favourites_count': 147205,
'utc_offset': None,
'time_zone': None,
'geo_enabled': True,
'verified': True,
'statuses_count': 16089,
'lang': None,
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': '000000',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/them
e1/bg.png',
'profile_background_image_url_https': 'https://abs.twimg.com/images/them
es/theme1/bg.png',
'profile_background_tile': False,
'profile_image_url': 'http://pbs.twimg.com/profile_images/15529957290142
47425/TaJbIdmK_normal.jpg',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1552995
729014247425/TaJbIdmK_normal.jpg',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/
1617810473',
'profile_link_color': 'F5ABB5',
'profile_sidebar_border_color': '000000',
'profile_sidebar_fill_color': '000000',
'profile_text_color': '000000',
'profile_use_background_image': False,
'has_extended_profile': False,
'default_profile': False,
'default_profile_image': False,
'following': True,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'none',
'withheld_in_countries': [],
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 5279,
'favorite_count': 29251,
'favorited': False,
'retweeted': False,
'possibly_sensitive': False,
'possibly_sensitive_appealable': False,
```



```
'lang': 'en'},
{'created_at': 'Mon Jul 31 00:18:03 +0000 2017',
 'id': 891815181378084864,
 'id_str': '891815181378084864',
 'full_text': 'This is Archie. He is a rare Norwegian Pouncing Corgo. Live
s in the tall grass. You never know when one may strike. 12/10 https://t.co
/wUnZnhtVJB',
 'truncated': False,
 'display_text_range': [0, 121],
 'entities': {'hashtags': [],
 'symbols': [],
 'user_mentions': [],
 'urls': [],
 'media': [{'id': 891815175371796480,
 'id_str': '891815175371796480',
 'indices': [122, 145],
 'media_url': 'http://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg',
 'media_url_https': 'https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg',
 'url': 'https://t.co/wUnZnhtVJB',
 'display_url': 'pic.twitter.com/wUnZnhtVJB',
 'expanded_url': 'https://twitter.com/dog_rates/status/8918151813780848
64/photo/1',
 'type': 'photo',
 'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
 'small': {'w': 510, 'h': 680, 'resize': 'fit'},
 'medium': {'w': 901, 'h': 1200, 'resize': 'fit'},
 'large': {'w': 1201, 'h': 1600, 'resize': 'fit'}}}],
 'extended_entities': {'media': [{'id': 891815175371796480,
 'id_str': '891815175371796480',
 'indices': [122, 145],
 'media_url': 'http://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg',
 'media_url_https': 'https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg',
 'url': 'https://t.co/wUnZnhtVJB',
 'display_url': 'pic.twitter.com/wUnZnhtVJB',
 'expanded_url': 'https://twitter.com/dog_rates/status/8918151813780848
64/photo/1',
 'type': 'photo',
 'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
 'small': {'w': 510, 'h': 680, 'resize': 'fit'},
 'medium': {'w': 901, 'h': 1200, 'resize': 'fit'},
 'large': {'w': 1201, 'h': 1600, 'resize': 'fit'}}}],
 'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Tw
itter for iPhone</a>',
 'in_reply_to_status_id': None,
 'in_reply_to_status_id_str': None,
 'in_reply_to_user_id': None,
 'in_reply_to_user_id_str': None,
 'in_reply_to_screen_name': None,
 'user': {'id': 4196983835,
 'id_str': '4196983835',
 'name': 'WeRateDogs®',
 'screen_name': 'dog_rates',
 'location': 'all our links →',
 'description': 'Your Only Source For Professional Dog Ratings \nInstagra
m and Facebook → WeRateDogs \npartnerships@weratedogs.com | nonprofit: @15o
utof10
',
```

```
'url': 'https://t.co/YPc2XqmwyC',
'entities': {'url': {'urls': [{'url': 'https://t.co/YPc2XqmwyC',
    'expanded_url': 'http://links.weratedogs.com',
    'display_url': 'links.weratedogs.com',
    'indices': [0, 23]}]}},
'description': {'urls': []},
'protected': False,
'followers_count': 9339224,
'friends_count': 21,
'listed_count': 7566,
'created_at': 'Sun Nov 15 21:41:29 +0000 2015',
'favourites_count': 147205,
'utc_offset': None,
'time_zone': None,
'geo_enabled': True,
'verified': True,
'statuses_count': 16089,
'lang': None,
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': '000000',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_tile': False,
'profile_image_url': 'http://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jpg',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jpg',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/1617810473',
'profile_link_color': 'F5ABB5',
'profile_sidebar_border_color': '000000',
'profile_sidebar_fill_color': '000000',
'profile_text_color': '000000',
'profile_use_background_image': False,
'has_extended_profile': False,
'default_profile': False,
'default_profile_image': False,
'following': True,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'none',
'withheld_in_countries': [],
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 3465,
'favorite_count': 21985,
'favorited': False,
'retweeted': False,
'possibly_sensitive': False,
```

```

    'possibly_sensitive_appealable': False,
    'lang': 'en'},
    {'created_at': 'Sun Jul 30 15:58:51 +0000 2017',
      'id': 891689557279858688,
      'id_str': '891689557279858688',
      'full_text': 'This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qLQ',
      'truncated': False,
      'display_text_range': [0, 79],
      'entities': {'hashtags': [],
                    'symbols': [],
                    'user_mentions': [],
                    'urls': []},
      'media': [{'id': 891689552724799489,
                    'id_str': '891689552724799489',
                    'indices': [80, 103],
                    'media_url': 'http://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
                    'media_url_https': 'https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
                    'url': 'https://t.co/tD36da7qLQ',
                    'display_url': 'pic.twitter.com/tD36da7qLQ',
                    'expanded_url': 'https://twitter.com/dog_rates/status/891689557279858688/photo/1',
                    'type': 'photo',
                    'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
                              'small': {'w': 510, 'h': 680, 'resize': 'fit'},
                              'medium': {'w': 901, 'h': 1200, 'resize': 'fit'},
                              'large': {'w': 1201, 'h': 1600, 'resize': 'fit'}}}],
      'extended_entities': {'media': [{'id': 891689552724799489,
                                          'id_str': '891689552724799489',
                                          'indices': [80, 103],
                                          'media_url': 'http://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
                                          'media_url_https': 'https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg',
                                          'url': 'https://t.co/tD36da7qLQ',
                                          'display_url': 'pic.twitter.com/tD36da7qLQ',
                                          'expanded_url': 'https://twitter.com/dog_rates/status/891689557279858688/photo/1',
                                          'type': 'photo',
                                          'sizes': {'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
                                                    'small': {'w': 510, 'h': 680, 'resize': 'fit'},
                                                    'medium': {'w': 901, 'h': 1200, 'resize': 'fit'},
                                                    'large': {'w': 1201, 'h': 1600, 'resize': 'fit'}}}]},
      'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>',
      'in_reply_to_status_id': None,
      'in_reply_to_status_id_str': None,
      'in_reply_to_user_id': None,
      'in_reply_to_user_id_str': None,
      'in_reply_to_screen_name': None,
      'user': {'id': 4196983835,
                 'id_str': '4196983835',
                 'name': 'WeRateDogs®',
                 'screen_name': 'dog_rates',
                 'location': 'all our links →',
                 'description': 'Your Only Source For Professional Dog Ratings \nInstagram and Facebook → WeRateDogs \npartnerships@weratedogs.com | nonprofit: @150utof10',
                 'url': 'https://weratedogs.com',
                 'protected': False,
                 'verified': False,
                 'statuses_count': 1000,
                 'followers_count': 1000,
                 'friends_count': 1000,
                 'listed_count': 1000,
                 'profile_image_url': 'https://pbs.twimg.com/profile_images/891689552724799489/891689552724799489.jpg',
                 'profile_image_url_https': 'https://pbs.twimg.com/profile_images/891689552724799489/891689552724799489.jpg',
                 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/1500000000',
                 'default_profile': False,
                 'default_profile_image': False}]}],
    'next': 'https://api.twitter.com/1.1/statuses/user_timeline.json?cursor=next%3A891689552724799489&max_results=20&tweet_mode=extended&user_id=4196983835',
    'previous': 'https://api.twitter.com/1.1/statuses/user_timeline.json?cursor=previous%3A891689552724799489&max_results=20&tweet_mode=extended&user_id=4196983835',
    'response_metadata': {'errors': []}}

```

```
'url': 'https://t.co/YPc2XqmwyC',
'entities': {'url': {'urls': [{'url': 'https://t.co/YPc2XqmwyC',
    'expanded_url': 'http://links.weratedogs.com',
    'display_url': 'links.weratedogs.com',
    'indices': [0, 23]}]}},
'description': {'urls': []}},
'protected': False,
'followers_count': 9339224,
'friends_count': 21,
'listed_count': 7566,
'created_at': 'Sun Nov 15 21:41:29 +0000 2015',
'favourites_count': 147205,
'utc_offset': None,
'time_zone': None,
'geo_enabled': True,
'verified': True,
'statuses_count': 16089,
'lang': None,
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': '000000',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_tile': False,
'profile_image_url': 'http://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jpg',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jpg',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/1617810473',
'profile_link_color': 'F5ABB5',
'profile_sidebar_border_color': '000000',
'profile_sidebar_fill_color': '000000',
'profile_text_color': '000000',
'profile_use_background_image': False,
'has_extended_profile': False,
'default_profile': False,
'default_profile_image': False,
'following': True,
'follow_request_sent': False,
'notifications': False,
'translator_type': 'none',
'withheld_in_countries': [],
'geo': None,
'coordinates': None,
'place': None,
'contributors': None,
'is_quote_status': False,
'retweet_count': 7198,
'favorite_count': 36817,
'favorited': False,
'retweeted': False,
'possibly_sensitive': False,
```

```
{
  'possibly_sensitive_appealable': False,
  'lang': 'en'},
{'created_at': 'Sat Jul 29 16:00:24 +0000 2017',
  'id': 891327558926688256,
  'id_str': '891327558926688256',
  'full_text': 'This is Franklin. He would like you to stop calling him "cute." He is a very fierce shark and should be respected as such. 12/10 #BarkWeek https://t.co/AtUZn91f7f',
  'truncated': False,
  'display_text_range': [0, 138],
  'entities': {'hashtags': [{'text': 'BarkWeek', 'indices': [129, 138]}],
    'symbols': [],
    'user_mentions': [],
    'urls': []},
  'media': [{'id': 891327551943041024,
    'id_str': '891327551943041024',
    'indices': [139, 162],
    'media_url': 'http://pbs.twimg.com/media/DF6hr6AVYAAZ8G8.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DF6hr6AVYAAZ8G8.jpg',
    'url': 'https://t.co/AtUZn91f7f',
    'display_url': 'pic.twitter.com/AtUZn91f7f',
    'expanded_url': 'https://twitter.com/dog_rates/status/891327558926688256/photo/1',
    'type': 'photo',
    'sizes': {'medium': {'w': 720, 'h': 540, 'resize': 'fit'},
      'large': {'w': 720, 'h': 540, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
      'small': {'w': 680, 'h': 510, 'resize': 'fit'}}}],
  'extended_entities': {'media': [{'id': 891327551943041024,
    'id_str': '891327551943041024',
    'indices': [139, 162],
    'media_url': 'http://pbs.twimg.com/media/DF6hr6AVYAAZ8G8.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DF6hr6AVYAAZ8G8.jpg',
    'url': 'https://t.co/AtUZn91f7f',
    'display_url': 'pic.twitter.com/AtUZn91f7f',
    'expanded_url': 'https://twitter.com/dog_rates/status/891327558926688256/photo/1',
    'type': 'photo',
    'sizes': {'medium': {'w': 720, 'h': 540, 'resize': 'fit'},
      'large': {'w': 720, 'h': 540, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
      'small': {'w': 680, 'h': 510, 'resize': 'fit'}}}],
    {'id': 891327551947157504,
    'id_str': '891327551947157504',
    'indices': [139, 162],
    'media_url': 'http://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg',
    'media_url_https': 'https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg',
    'url': 'https://t.co/AtUZn91f7f',
    'display_url': 'pic.twitter.com/AtUZn91f7f',
    'expanded_url': 'https://twitter.com/dog_rates/status/891327558926688256/photo/1',
    'type': 'photo',
    'sizes': {'medium': {'w': 720, 'h': 540, 'resize': 'fit'},
      'large': {'w': 720, 'h': 540, 'resize': 'fit'},
      'thumb': {'w': 150, 'h': 150, 'resize': 'crop'},
      'small': {'w': 680, 'h': 510, 'resize': 'fit'}}}]},
  'possibly_sensitive': False}
```

```
'source': '<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>',
'in_reply_to_status_id': None,
'in_reply_to_status_id_str': None,
'in_reply_to_user_id': None,
'in_reply_to_user_id_str': None,
'in_reply_to_screen_name': None,
'user': {'id': 4196983835,
'id_str': '4196983835',
'name': 'WeRateDogs@',
'screen_name': 'dog_rates',
'location': 'all our links →',
'description': 'Your Only Source For Professional Dog Ratings \nInstagram and Facebook → WeRateDogs \npartnerships@weratedogs.com | nonprofit: @150utof10',
'url': 'https://t.co/YPc2XqmwYC',
'entities': {'url': {'urls': [{'url': 'https://t.co/YPc2XqmwYC',
'expanded_url': 'http://links.weratedogs.com',
'display_url': 'links.weratedogs.com',
"indices': [0, 23]}]}},
'description': {'urls': []}},
'protected': False,
'followers_count': 9339224,
'friends_count': 21,
'listed_count': 7566,
'created_at': 'Sun Nov 15 21:41:29 +0000 2015',
'favourites_count': 147205,
'utc_offset': None,
'time_zone': None,
'geo_enabled': True,
'verified': True,
'statuses_count': 16089,
'lang': None,
'contributors_enabled': False,
'is_translator': False,
'is_translation_enabled': False,
'profile_background_color': '000000',
'profile_background_image_url': 'http://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_tile': False,
'profile_image_url': 'http://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jpg',
'profile_image_url_https': 'https://pbs.twimg.com/profile_images/1552995729014247425/TaJbIdmK_normal.jpg',
'profile_banner_url': 'https://pbs.twimg.com/profile_banners/4196983835/1617810473',
'profile_link_color': 'F5ABB5',
'profile_sidebar_border_color': '000000',
'profile_sidebar_fill_color': '000000',
'profile_text_color': '000000',
'profile_use_background_image': False,
'has_extended_profile': False,
'default_profile': False,
'default_profile_image': False,
```

```
'following': True,  
'follow_request_sent': False,  
'notifications': False,  
'translator_type': 'none',  
'withheld_in_countries': [],  
'geo': None,  
'coordinates': None,  
'place': None,  
'contributors': None,  
'is_quote_status': False,  
'retweet_count': 7722,  
'favorite_count': 35202,  
'favorited': False,  
'retweeted': False,  
'possibly_sensitive': False,  
'possibly_sensitive_appealable': False,  
'lang': 'en']}]
```

Now we can create a dataframe with `DataFrame.from_records` :

```
In [24]: tweet_json_txt_df = pd.DataFrame.from_records(tweet_json_txt_json)
```

```
In [25]: tweet_json_txt_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2327 entries, 0 to 2326
Data columns (total 32 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   created_at                           2327 non-null   object
1   id                                    2327 non-null   int64
2   id_str                               2327 non-null   object
3   full_text                            2327 non-null   object
4   truncated                            2327 non-null   bool
5   display_text_range                   2327 non-null   object
6   entities                             2327 non-null   object
7   extended_entities                    2057 non-null   object
8   source                               2327 non-null   object
9   in_reply_to_status_id                77 non-null     float64
10  in_reply_to_status_id_str             77 non-null     object
11  in_reply_to_user_id                  77 non-null     float64
12  in_reply_to_user_id_str               77 non-null     object
13  in_reply_to_screen_name               77 non-null     object
14  user                                  2327 non-null   object
15  geo                                   0 non-null      object
16  coordinates                           0 non-null      object
17  place                                 1 non-null      object
18  contributors                           0 non-null      object
19  is_quote_status                       2327 non-null   bool
20  retweet_count                         2327 non-null   int64
21  favorite_count                        2327 non-null   int64
22  favorited                             2327 non-null   bool
23  retweeted                             2327 non-null   bool
24  possibly_sensitive                    2195 non-null   object
25  possibly_sensitive_appealable         2195 non-null   object
26  lang                                   2327 non-null   object
27  retweeted_status                      160 non-null    object
28  quoted_status_id                      26 non-null     float64
29  quoted_status_id_str                  26 non-null     object
30  quoted_status_permalink                26 non-null     object
31  quoted_status                          24 non-null     object
dtypes: bool(4), float64(3), int64(3), object(22)
memory usage: 518.2+ KB

```

```
In [26]: tweet_json_txt_df.head()
```


Out [26]:	created_at	id	id_str	full_text	truncated	display_
0	Tue Aug 01 16:23:56 +0000 2017	892420643555336193	892420643555336193	This is Phineas. He's a mystical boy. Only eve...	False	
1	Tue Aug 01 00:17:27 +0000 2017	892177421306343426	892177421306343426	This is Tilly. She's just checking pup on you....	False	
2	Mon Jul 31 00:18:03 +0000 2017	891815181378084864	891815181378084864	This is Archie. He is a rare Norwegian Pouncin...	False	
3	Sun Jul 30 15:58:51 +0000 2017	891689557279858688	891689557279858688	This is Darla. She commenced a snooze mid meal...	False	
4	Sat Jul 29 16:00:24 +0000 2017	891327558926688256	891327558926688256	This is Franklin. He would like you to stop ca...	False	

5 rows × 32 columns

Since we only need id, retweet count, and favorite count, let's project the rest of the columns out:

```
In [27]: tweet_json_txt_df = tweet_json_txt_df[["id", "retweet_count", "favorite_count"]]
```

Assessing Data

In this section, detect and document at least **eight (8) quality issues and two (2) tidiness issue**. You must use **both** visual assessment programmatic assesement to assess the data.

Note: pay attention to the following key points when you access the data.

- You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills in data wrangling. Therefore, the requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This [unique rating system](#) is a big part of the popularity of WeRateDogs.
- You do not need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the algorithm used.

twitter_archived_enhanced_df

```
In [28]: twitter_archived_enhanced_df.head()
```

Out[28]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href="http://i /download/
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://i /download/
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://i /download/
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	href="http://i /download/
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	href="http://i /download/

In [29]: `twitter_archived_enhanced_df.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   tweet_id                             2356 non-null   int64
1   in_reply_to_status_id                 78 non-null     float64
2   in_reply_to_user_id                   78 non-null     float64
3   timestamp                             2356 non-null   object
4   source                                2356 non-null   object
5   text                                  2356 non-null   object
6   retweeted_status_id                   181 non-null    float64
7   retweeted_status_user_id              181 non-null    float64
8   retweeted_status_timestamp            181 non-null    object
9   expanded_urls                         2297 non-null   object
10  rating_numerator                       2356 non-null   int64
11  rating_denominator                     2356 non-null   int64
12  name                                   2356 non-null   object
13  doggo                                 2356 non-null   object
14  floofer                               2356 non-null   object
15  pupper                                2356 non-null   object
16  puppo                                 2356 non-null   object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

No missing text, ratings, names, or stages.

```
In [30]: twitter_archived_enhanced_df.tweet_id.duplicated().sum() # No duplicate twe
```

```
Out[30]: 0
```

```
In [31]: (~twitter_archived_enhanced_df.in_reply_to_status_id.isna()).sum() # There
```

```
Out[31]: 78
```

```
In [32]: (~twitter_archived_enhanced_df.retweeted_status_id.isna()).sum() # There ar
```

```
Out[32]: 181
```

Let's filter out replies and retweets (only for assessing purposes):

```
In [33]: twitter_archived_enhanced_valid_df = twitter_archived_enhanced_df[
    twitter_archived_enhanced_df.in_reply_to_status_id.isna()
    & twitter_archived_enhanced_df.retweeted_status_id.isna()
]
```

```
In [34]: twitter_archived_enhanced_valid_df.shape[0] # We've got 2097 potentially va
```

```
Out[34]: 2097
```

Let's look for rating numerators/denominators outside of the traditional WeRateDogs rating system. Historically, the ranking goes from 0/10 to 15/10. Let's look at numerators:

```
In [35]: twitter_archived_enhanced_valid_df[
    twitter_archived_enhanced_valid_df.rating_numerator > 15
].rating_numerator.value_counts()
```

```
Out[35]: 84      1
         24      1
         88      1
        144      1
         26      1
        121      1
         44      1
         60      1
         45      1
         80      1
         99      1
         50      1
        204      1
       1776      1
        165      1
         27      1
         75      1
        420      1
Name: rating_numerator, dtype: int64
```

```
In [36]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 84
         ]["tweet_id", "text"].values
```

```
Out[36]: array([[820690176645140481,
         'The floofs have been released I repeat the floofs have been releas
         ed. 84/70 https://t.co/NIYC820tmd']],
         dtype=object)
```

```
In [37]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 24
         ]["tweet_id", "text"].values
```

```
Out[37]: array([[810984652412424192,
         'Meet Sam. She smiles 24/7 & secretly aspires to be a reindeer.
         \nKeep Sam smiling by clicking and sharing this link:\nhttps://t.co/98tB8y7
         y7t https://t.co/LouL5vdvxx']],
         dtype=object)
```

```
In [38]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 88
         ]["tweet_id", "text"].values
```

```
Out[38]: array([[675853064436391936,
         'Here we have an entire platoon of puppies. Total score: 88/80 woul
         d pet all at once https://t.co/y93p6FLvVw']],
         dtype=object)
```

```
In [39]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 26
         ]["tweet_id", "text"].values
```

```
Out[39]: array([[680494726643068929,
         'Here we have uncovered an entire battalion of holiday puppies. Ave
         rage of 11.26/10 https://t.co/eNm2S6p9BD']],
         dtype=object)
```

```
In [40]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 121
         ]["tweet_id", "text"].values
```

```
Out[40]: array([[684222868335505415,
         'Someone help the girl is being mugged. Several are distracting her
         while two steal her shoes. Clever puppies 121/110 https://t.co/1zfnTJLt55
         ']],
         dtype=object)
```

```
In [41]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 44
         ]["tweet_id", "text"].values
```

```
Out[41]: array([[697463031882764288,
         'Happy Wednesday here's a bucket of pups. 44/40 would pet all at on
         ce https://t.co/HppvrYuamZ']],
         dtype=object)
```

```
In [42]: twitter_archived_enhanced_valid_df[
        twitter_archived_enhanced_valid_df.rating_numerator == 60
        ]["tweet_id", "text"].values
```

```
Out[42]: array([[704054845121142784,
        'Here is a whole flock of puppies. 60/50 I'll take the lot http
        s://t.co/9dpcw6MdWa']],
        dtype=object)
```

```
In [43]: twitter_archived_enhanced_valid_df[
        twitter_archived_enhanced_valid_df.rating_numerator == 45
        ]["tweet_id", "text"].values
```

```
Out[43]: array([[709198395643068416,
        'From left to right:\nCletus, Jerome, Alejandro, Burp, & Titso
        n\nNone know where camera is. 45/50 would hug all at once https://t.co/sedr
        elivTK']],
        dtype=object)
```

```
In [44]: twitter_archived_enhanced_valid_df[
        twitter_archived_enhanced_valid_df.rating_numerator == 80
        ]["tweet_id", "text"].values
```

```
Out[44]: array([[710658690886586372,
        'Here's a brigade of puppies. All look very prepared for whatever h
        appens next. 80/80 https://t.co/0eb7R10m12']],
        dtype=object)
```

```
In [45]: twitter_archived_enhanced_valid_df[
        twitter_archived_enhanced_valid_df.rating_numerator == 99
        ]["tweet_id", "text"].values
```

```
Out[45]: array([[713900603437621249,
        'Happy Saturday here's 9 puppies on a bench. 99/90 good work everyb
        ody https://t.co/mpvaVxKmc1']],
        dtype=object)
```

```
In [46]: twitter_archived_enhanced_valid_df[
        twitter_archived_enhanced_valid_df.rating_numerator == 50
        ]["tweet_id", "text"].values
```

```
Out[46]: array([[716439118184652801,
        'This is Bluebert. He just saw that both #FinalFur match ups are sp
        lit 50/50. Amazed af. 11/10 https://t.co/Kky1DPG4iq']],
        dtype=object)
```

```
In [47]: twitter_archived_enhanced_valid_df[
        twitter_archived_enhanced_valid_df.rating_numerator == 204
        ]["tweet_id", "text"].values
```

```
Out[47]: array([[731156023742988288,
        'Say hello to this unbelievably well behaved squad of doggos. 204/1
        70 would try to pet all at once https://t.co/yGQI3He3xv']],
        dtype=object)
```

A joke rating but technically still valid:

```
In [48]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 1776
         ]["tweet_id", "text"].values
```

```
Out[48]: array([[749981277374128128,
         "This is Atticus. He's quite simply America af. 1776/10 https://t.c
         o/GRXwMxLBkh"]],
         dtype=object)
```

```
In [49]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 165
         ]["tweet_id", "text"].values
```

```
Out[49]: array([[758467244762497024,
         'Why does this never happen at my front door... 165/150 https://t.c
         o/HmwrdfEfUE']],
         dtype=object)
```

```
In [50]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 27
         ]["tweet_id", "text"].values
```

```
Out[50]: array([[778027034220126208,
         "This is Sophie. She's a Jubilant Bush Pupper. Super h*ckin rare. A
         ppears at random just to smile at the locals. 11.27/10 would smile back htt
         ps://t.co/QFaUiIHxHq"]],
         dtype=object)
```

```
In [51]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 75
         ]["tweet_id", "text"].values
```

```
Out[51]: array([[786709082849828864,
         "This is Logan, the Chow who lived. He solemnly swears he's up to l
         ots of good. H*ckin magical af 9.75/10 https://t.co/yB05wuqaPS"]],
         dtype=object)
```

```
In [52]: twitter_archived_enhanced_valid_df[
         twitter_archived_enhanced_valid_df.rating_numerator == 420
         ]["tweet_id", "text"].values
```

```
Out[52]: array([[670842764863651840,
         'After so many requests... here you go.\n\nGood dogg. 420/10 http
         s://t.co/yfAAo1gdeY']],
         dtype=object)
```

Now let's check the denominators. Anything that is not a 10 and does not contain multiple doggos is to be considered suspicious.

```
In [53]: twitter_archived_enhanced_valid_df[
          (twitter_archived_enhanced_valid_df.rating_denominator != 10) &
          (~twitter_archived_enhanced_valid_df.tweet_id.isin([
              820690176645140481, 675853064436391936, 677716515794329600, 68422286
              704054845121142784, 709198395643068416, 710658690886586372, 71390060
              758467244762497024
          ]))]
          ].rating_denominator.value_counts()
```

```
Out[53]: 11    2
          7     1
          20    1
          50    1
           2     1
          Name: rating_denominator, dtype: int64
```

```
In [54]: twitter_archived_enhanced_valid_df[
          twitter_archived_enhanced_valid_df.rating_denominator == 11
          ][["tweet_id", "text"]].values
```

```
Out[54]: array([[740373189193256964,
                  'After so many requests, this is Bretagne. She was the last survivi
                  ng 9/11 search dog, and our second ever 14/10. RIP https://t.co/XAVDNDaVgQ
                  '],
                  [682962037429899265,
                  'This is Darrel. He just robbed a 7/11 and is in a high speed polic
                  e chase. Was just spotted by the helicopter 10/10 https://t.co/7EsP8LmSp5
                  ']],
          dtype=object)
```

```
In [55]: twitter_archived_enhanced_valid_df[
          twitter_archived_enhanced_valid_df.rating_denominator == 7
          ][["tweet_id", "text"]].values
```

```
Out[55]: array([[810984652412424192,
                  'Meet Sam. She smiles 24/7 & secretly aspires to be a reindeer.
                  \nKeep Sam smiling by clicking and sharing this link:\nhttps://t.co/98tB8y7
                  y7t https://t.co/LouL5vdvxx']],
          dtype=object)
```

```
In [56]: twitter_archived_enhanced_valid_df[
          twitter_archived_enhanced_valid_df.rating_denominator == 20
          ][["tweet_id", "text"]].values
```

```
Out[56]: array([[722974582966214656,
                  'Happy 4/20 from the squad! 13/10 for all https://t.co/eV1diwds8a
                  ']],
          dtype=object)
```

```
In [57]: twitter_archived_enhanced_valid_df[
          twitter_archived_enhanced_valid_df.rating_denominator == 2
          ][["tweet_id", "text"]].values
```



```
Out[57]: array([[666287406224695296,
                'This is an Albanian 3 1/2 legged  Episcopalian. Loves well-polishe
                d hardwood flooring. Penis on the collar. 9/10 https://t.co/d9NcXFKwLv']],
                dtype=object)
```

Now let's check the names:

```
In [58]: twitter_archived_enhanced_valid_df.name.value_counts()
```

```
Out[58]: None          603
a              55
Lucy           11
Charlie        11
Oliver         10
...
Lenox          1
Harvey         1
Blanket        1
Burt           1
Christoper     1
Name: name, Length: 955, dtype: int64
```

```
In [59]: twitter_archived_enhanced_valid_df[twitter_archived_enhanced_valid_df.name =
```

```

Out[59]: array([[881536004380872706,
                'Here is a pupper approaching maximum borkdrive. Zooming at never b
efore seen speeds. 14/10 paw-inspiring af \n(IG: puffie_the_chow) https://
t.co/ghXBIIeQZF'],
                [792913359805018113,
                'Here is a perfect example of someone who has their priorities in o
rder. 13/10 for both owner and Forrest https://t.co/LRyMrU7Wfq'],
                [772581559778025472,
                'Guys this is getting so out of hand. We only rate dogs. This is a
Galapagos Speed Panda. Pls only send dogs... 10/10 https://t.co/8lpAGaZRFn
'],
                [747885874273214464,
                'This is a mighty rare blue-tailed hammer sherk. Human almost lost
a limb trying to take these. Be careful guys. 8/10 https://t.co/TGenMeXreW
'],
                [747816857231626240,
                'Viewer discretion is advised. This is a terrible attack in progres
s. Not even in water (tragic af). 4/10 bad sherk https://t.co/L3U0j14N5R'],
                [746872823977771008,
                'This is a carrot. We only rate dogs. Please only send in dogs. You
all really should know this by now ...11/10 https://t.co/9e48aPrBm2'],
                [743222593470234624,
                'This is a very rare Great Alaskan Bush Pupper. Hard to stumble upo
n without spooking. 12/10 would pet passionately https://t.co/x0BKCdpzaa'],
                [717537687239008257,
                'People please. This is a Deadly Mediterranean Plop T-Rex. We only
rate dogs. Only send in dogs. Thanks you... 11/10 https://t.co/2ATDsgHD4n
'],
                [715733265223708672,
                'This is a taco. We only rate dogs. Please only send in dogs. Dogs
are what we rate. Not tacos. Thank you... 10/10 https://t.co/cxl6xGY8B9'],
                [704859558691414016,
                'Here is a heartbreaking scene of an incredible pupper being laid t
o rest. 10/10 RIP pupper https://t.co/81mvJ0rGRu'],
                [704054845121142784,
                'Here is a whole flock of puppies. 60/50 I'll take the lot http
s://t.co/9dpcw6MdWa'],
                [703079050210877440,
                'This is a Butternut Cumberfloof. It's not windy they just look lik
e that. 11/10 back at it again with the red socks https://t.co/hMjzhdUHa
W'],
                [702539513671897089,
                'This is a Wild Tuscan Poofwiggle. Careful not to startle. Rare ton
gue slip. One eye magical. 12/10 would def pet https://t.co/4EnShAQjv6'],
                [700864154249383937,
                '"Pupper is a present to world. Here is a bow for pupper." 12/10 pr
ecious as hell https://t.co/ItSsE92gCW'],
                [692187005137076224,
                'This is a rare Arctic Wubberfloof. Unamused by the happenings. No
longer has the appetites. 12/10 would totally hug https://t.co/krvbacIX0N
'],
                [679530280114372609,
                'Guys this really needs to stop. We've been over this way too many
times. This is a giraffe. We only rate dogs.. 7/10 https://t.co/yavgkHYPO
C'],
                [677644091929329666,

```

'This is a dog swinging. I really enjoyed it so I hope you all do a
s well. 11/10 <https://t.co/0zo9KHTRND>'],
[675706639471788032,
"This is a Sizzlin Menorah spaniel from Brooklyn named Wylie. Lovab
le eyes. Chiller as hell. 10/10 and I'm out.. poof <https://t.co/7E0AiJXPmI>'],
[675534494439489536,
'Seriously guys?! Only send in dogs. I only rate dogs. This is a ba
by black bear... 11/10 <https://t.co/H7kpabTfLj>'],
[675109292475830276,
"C'mon guys. We've been over this. We only rate dogs. This is a co
w. Please only submit dogs. Thank you..... 9/10 <https://t.co/WjcELNEqN2>'],
[675047298674663426,
'This is a fluffy albino Bacardi Columbia mix. Excellent at the twe
ets. 11/10 would hug gently <https://t.co/diboDRUuEI>'],
[674082852460433408,
'This is a Sagitariot Baklava mix. Loves her new hat. 11/10 radiant
pup <https://t.co/Bko5kFJYUU>'],
[673715861853720576,
'This is a heavily opinionated dog. Loves walls. Nobody knows how t
he hair works. Always ready for a kiss. 4/10 <https://t.co/dFiaKZ9cDl>'],
[673636718965334016,
'This is a Lofted Aphrodisiac Terrier named Kip. Big fan of bed n b
reakfasts. Fits perfectly. 10/10 would pet firmly <https://t.co/gKlLpNzIl3>
'],
[672604026190569472,
'This is a baby Rand Paul. Curls for days. 11/10 would cuddle the h
ell out of <https://t.co/xHXNaPAYRe>'],
[671743150407421952,
'This is a Tuscaloosa Alcatraz named Jacob (Yacōb). Loves to sit in
swing. Stellar tongue. 11/10 look at his feet <https://t.co/2IslQ8ZSc7>'],
[671147085991960577,
"This is a Helvetica Listerine named Rufus. This time Rufus will be
ready for the UPS guy. He'll never expect it 9/10 <https://t.co/340hVhMkVr>'],
[670427002554466305,
'This is a Deciduous Trimester mix named Spork. Only 1 ear works. N
o seat belt. Incredibly reckless. 9/10 still cute <https://t.co/CtuJoLHiDo>
'],
[670361874861563904,
"This is a Rich Mahogany Seltzer named Cherokee. Just got destroyed
by a snowball. Isn't very happy about it. 9/10 <https://t.co/98ZBi6o4dj>'],
[670303360680108032,
'This is a Speckled Cauliflower Yosemite named Hemry. He's terrifie
d of intruder dog. Not one bit comfortable. 9/10 <https://t.co/yV3Qgjh8iN>'],
[669923323644657664,
'This is a spotted Lipitor Rumpelstiltskin named Alphred. He can't
wait for the Turkey. 10/10 would pet really well <https://t.co/6GUG07azNX>'],
[669661792646373376,
"This is a brave dog. Excellent free climber. Trying to get closer
to God. Not very loyal though. Doesn't bark. 5/10 <https://t.co/0DnILTr4QM>'],
[669564461267722241,
'This is a Coriander Baton Rouge named Alfredo. Loves to cuddle wit
h smaller well-dressed dog. 10/10 would hug lots <https://t.co/eCRdwouKCl>'],
[668955713004314625,

'This is a Slovakian Helter Skelter Feta named Leroi. Likes to skip on roofs. Good traction. Much balance. 10/10 wow! <https://t.co/Dmy2mY2Qj5>'],

[668815180734689280,

'This is a wild Toblerone from Papua New Guinea. Mouth always open. Addicted to hay. Acts blind. 7/10 handsome dog <https://t.co/IGmVbz07tZ>'],

[668614819948453888,

'Here is a horned dog. Much grace. Can jump over moons (dam!). Paws not soft. Bad at barking. 7/10 can still pet tho <https://t.co/2Su7gmsnZm>'],

[668507509523615744,

'This is a Birmingham Quagmire named Chuk. Loves to relax and watch the game while sippin on that iced mocha. 10/10 <https://t.co/HvNg9JWxFt>'],

[668466899341221888,

"Here is a mother dog caring for her pups. Snazzy red mohawk. Doesn't wag tail. Pups look confused. Overall 4/10 <https://t.co/Y0He6lf09m>'],

[668171859951755264,

'This is a Trans Siberian Kellogg named Alfonso. Huge ass eyeballs. Actually Dobby from Harry Potter. 7/10 <https://t.co/XpseHBLAAb>'],

[667861340749471744,

'This is a Shotokon Macadamia mix named Cheryl. Sophisticated af. Looks like a disappointed librarian. Shh (lol) 9/10 <https://t.co/J4GnJ5Swba>'],

[667773195014021121,

'This is a rare Hungarian Pinot named Jessiga. She is either mid-stroke or got stuck in the washing machine. 8/10 <https://t.co/ZU0i0KJyqD>'],

[667538891197542400,

'This is a southwest Coriander named Klint. Hat looks expensive. Still on house arrest :(\n9/10 <https://t.co/IQTOMqDUIe>'],

[667470559035432960,

'This is a northern Wahoo named Kohl. He runs this town. Chases tumbleweeds. Draws gun wicked fast. 11/10 legendary <https://t.co/J4vn2r0YFK>'],

[667177989038297088,

"This is a Dasani Kingfisher from Maine. His name is Daryl. Daryl doesn't like being swallowed by a panda. 8/10 <https://t.co/jpaeu6LNmW>'],

[666983947667116034,

'This is a curly Ticonderoga named Pepe. No feet. Loves to jet ski. 11/10 would hug until forever <https://t.co/cyDfaK8NBc>'],

[666781792255496192,

'This is a purebred Bacardi named Octaviath. Can shoot spaghetti out of mouth. 10/10 <https://t.co/uEvsGL0FHh>'],

[666701168228331520,

'This is a golden Buckminsterfullerene named Johm. Drives trucks. Lumberjack (?). Enjoys wall. 8/10 would hug softly <https://t.co/uQbZJM2DQB>'],

[666407126856765440,

'This is a southern Vesuvius bumblegruff. Can drive a truck (wow). Made friends with 5 other nifty dogs (neat). 7/10 <https://t.co/LopTBkKa8h>'],

[666293911632134144,

"This is a funny dog. Weird toes. Won't come down. Loves branch. Refuses to eat his food. Hard to cuddle with. 3/10 <https://t.co/IIXis0zta0>'],

[666057090499244032,

'My oh my. This is a rare blond Canadian terrier on wheels. Only \$8.98. Rather docile. 9/10 very rare <https://t.co/yWBqbrzy80>'],

[666055525042405380,

'Here is a Siberian heavily armored polar bear mix. Strong owner. 1

```

0/10 I would do unspeakable things to pet this dog https://t.co/rdivxLiqEt
'],
    [666050758794694657,
      'This is a truly beautiful English Wilson Staff retriever. Has a nice phone. Privileged. 10/10 would trade lives with https://t.co/fvIbQfHjIe
'],
    [666044226329800704,
      'This is a purebred Piers Morgan. Loves to Netflix and chill. Always looks like he forgot to unplug the iron. 6/10 https://t.co/DWnyCjf2mx'],
    [666033412701032449,
      'Here is a very happy pup. Big fan of well-maintained decks. Just look at that tongue. 9/10 would cuddle af https://t.co/y671yMhoiR'],
    [666029285002620928,
      'This is a western brown Mitsubishi terrier. Upset about leaf. Actually 2 dogs here. 7/10 would walk the shit out of https://t.co/r7m0b2m0UI
']],
    dtype=object)

```

Now let's check the stages:

```
In [60]: twitter_archived_enhanced_valid_df.doggo.value_counts()
```

```

Out[60]: None      2014
         doggo      83
         Name: doggo, dtype: int64

```

```
In [61]: twitter_archived_enhanced_valid_df.floofer.value_counts()
```

```

Out[61]: None      2087
         floofer     10
         Name: floofer, dtype: int64

```

```
In [62]: twitter_archived_enhanced_valid_df.pupper.value_counts()
```

```

Out[62]: None      1867
         pupper     230
         Name: pupper, dtype: int64

```

```
In [63]: twitter_archived_enhanced_valid_df.puppo.value_counts()
```

```

Out[63]: None      2073
         puppo      24
         Name: puppo, dtype: int64

```

image_predictions_df

```
In [64]: image_predictions_df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  -
0   tweet_id    2075 non-null   int64
1   jpg_url     2075 non-null   object
2   img_num     2075 non-null   int64
3   p1          2075 non-null   object
4   p1_conf     2075 non-null   float64
5   p1_dog      2075 non-null   bool
6   p2          2075 non-null   object
7   p2_conf     2075 non-null   float64
8   p2_dog      2075 non-null   bool
9   p3          2075 non-null   object
10  p3_conf     2075 non-null   float64
11  p3_dog      2075 non-null   bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB

```

```
In [65]: image_predictions_df.head()
```

```

Out[65]:

```

	tweet_id	jpg_url	img_num	p1	p1_conf
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.465074
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.506826
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.596461
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-IEu.jpg	1	Rhodesian_ridgeback	0.408143
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.560311

```
In [66]: image_predictions_df.shape[0] # There's a different number of records from
```

```
Out[66]: 2075
```

How many dog predictions are there? Are there any rows without a dog prediction?

```
In [67]: (image_predictions_df.p1_dog | image_predictions_df.p2_dog | image_predictions_df.p3_dog)
```

```
Out[67]: 1751
```

```
In [68]: ((~image_predictions_df.p1_dog) & (~image_predictions_df.p2_dog) & (~image_predictions_df.p3_dog))
```

```
Out[68]: 324
```

```
In [69]: image_predictions_df[(~image_predictions_df.p1_dog) & (~image_predictions_df
```

```
Out[69]:
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p
6	666051853826850816	https://pbs.twimg.com/media/CT5KoJ1WoAAJash.jpg	1	box_turtle	0.933012	
17	666104133288665088	https://pbs.twimg.com/media/CT56LSZWoaAIJ2.jpg	1	hen	0.965932	
18	666268910803644416	https://pbs.twimg.com/media/CT8QCd1WEAADXws.jpg	1	desktop_computer	0.086502	
21	666293911632134144	https://pbs.twimg.com/media/CT8mx7KW4AEQu8N.jpg	1	three-toed_sloth	0.914671	
25	666362758909284353	https://pbs.twimg.com/media/CT9IXGsUcAAyUft.jpg	1	guinea_pig	0.996496	

Are the confidence values consistent?

```
In [70]: image_predictions_df.p1_conf.describe()
```

```
Out[70]: count    2075.000000
mean         0.594548
std          0.271174
min          0.044333
25%          0.364412
50%          0.588230
75%          0.843855
max          1.000000
Name: p1_conf, dtype: float64
```

```
In [71]: image_predictions_df.p2_conf.describe()
```

```
Out[71]: count    2.075000e+03
mean    1.345886e-01
std     1.006657e-01
min     1.011300e-08
25%     5.388625e-02
50%     1.181810e-01
75%     1.955655e-01
max     4.880140e-01
Name: p2_conf, dtype: float64
```

```
In [72]: image_predictions_df.p3_conf.describe()
```

```
Out[72]: count    2.075000e+03
         mean     6.032417e-02
         std      5.090593e-02
         min      1.740170e-10
         25%      1.622240e-02
         50%      4.944380e-02
         75%      9.180755e-02
         max      2.734190e-01
         Name: p3_conf, dtype: float64
```

Anything weird on the dog breeds?

```
In [73]: image_predictions_df[image_predictions_df.p1_dog].p1.value_counts()
```

```
Out[73]: golden_retriever    150
         Labrador_retriever  100
         Pembroke            89
         Chihuahua           83
         pug                 57
         ...
         Japanese_spaniel     1
         Scotch_terrier       1
         standard_schnauzer    1
         EntleBucher          1
         clumber              1
         Name: p1, Length: 111, dtype: int64
```

```
In [74]: image_predictions_df[image_predictions_df.p2_dog].p2.value_counts()
```

```
Out[74]: Labrador_retriever    104
         golden_retriever      92
         Cardigan              73
         Chihuahua            44
         Pomeranian           42
         ...
         affenpinscher         1
         Japanese_spaniel      1
         Kerry_blue_terrier     1
         komondor              1
         Bernese_mountain_dog   1
         Name: p2, Length: 113, dtype: int64
```



```
In [75]: image_predictions_df[image_predictions_df.p3_dog].p3.value_counts()
```

```
Out[75]: Labrador_retriever      79
          Chihuahua              58
          golden_retriever       48
          Eskimo_dog             38
          kelpie                 35
          ..
          Irish_wolfhound        2
          affenpinscher          1
          Kerry_blue_terrier     1
          standard_schnauzer     1
          Sussex_spaniel         1
          Name: p3, Length: 116, dtype: int64
```

tweet_json_txt_df

```
In [76]: set(twitter_archived_enhanced_valid_df.tweet_id.values) - set(tweet_json_txt
```

```
Out[76]: {680055455951884288,
          754011816964026368,
          759923798737051648,
          779123168116150273,
          829374341691346946,
          837366284874571778,
          844704788403113984,
          872261713294495745}
```

Quality issues

1. There are 78 replies and 181 retweets on `twitter_archived_enhanced_df`.
([Solution](#))
2. On tweets `820690176645140481`, `675853064436391936`,
`677716515794329600`, `684222868335505415`, `697463031882764288`,
`704054845121142784`, `709198395643068416`, `710658690886586372`,
`713900603437621249`, `731156023742988288`, `758467244762497024` there
are multiple doggos with unusual ratings. ([Solution](#))
3. On tweets `810984652412424192`, `716439118184652801`,
`716439118184652801`, `740373189193256964`, `682962037429899265`,
`722974582966214656`, `666287406224695296` there are multiple expressions
confused for ratings. ([Solution](#))
4. On tweets `680494726643068929`, `778027034220126208`,
`786709082849828864` there are fractional ratings. ([Solution](#))
5. On tweet `670842764863651840` the rating is invalid as the subject is not a
doggo. ([Solution](#))
6. Although the rating on tweet `749981277374128128` is technically valid (1776),
we consider it an outlier. ([Solution](#))
7. Invalid name "a" on 55 tweets. ([Solution](#))
8. There were missing tweets when fetching the retweet and favourite counts with
Twitter's API. ([Solution](#))
9. There are 324 records without a dog prediction on `image_predictions_df`.
([Solution](#))

Tidiness issues

1. Multiple tables to represent the same observational unit: the tweet. ([Solution](#))
2. Multiple columns for the doggo stage variable. ([Solution](#))
3. Multiple columns to represent the breed prediction and confidence. ([Solution](#))
4. The predictions are a string but should be a category. ([Solution](#))
5. `timestamp` is a string, could be a datetime. ([Solution](#))
6. Dog breed names could be made consistent and readable. ([Solution](#))
7. The rating is a single variable represented in two columns. ([Solution](#))

Cleaning Data

In this section, clean **all** of the issues you documented while assessing.

Note: Make a copy of the original data before cleaning. Cleaning includes merging individual pieces of data according to the rules of [tidy data](#). The result should be a high-quality and tidy master pandas DataFrame (or DataFrames, if appropriate).

```
In [77]: # Make copies of original pieces of data

twitter_archived_enhanced_clean_df = twitter_archived_enhanced_df.copy()
image_predictions_clean_df = image_predictions_df.copy()
tweet_json_txt_clean_df = tweet_json_txt_df.copy()
```

As suggested during the cleaning data lessons, we'll tackle the tidyness issues first as this makes the rest of the tasks simpler.

Issue #1: Multiple columns to represent the breed prediction and confidence

Define:

- Turn `p1`, `p2` and `p3` into a single `prediction` column which shows the first prediction of a dog breed or `NaN` if no dog breed prediction was made.
- Turn `p1_conf`, `p2_conf` and `p3_conf` into a single `prediction_confidence` column which shows the confidence of the first prediction of a dog breed or `NaN` if no dog breed prediction was made.

Code

```
In [78]: image_predictions_clean_df.sample(5)
```

```
Out[78]:
```

	tweet_id	jpg_url	img_num	p1	p1_conf
1181	738184450748633089	https://pbs.twimg.com/media/Cj6O1G9UYAAIU-1.jpg	1	Bedlington_terrier	0.289471
1902	851861385021730816	https://pbs.twimg.com/media/C8W6sY_W0AEmttW.jpg	1	pencil_box	0.662183
441	674468880899788800	https://pbs.twimg.com/media/CVwx3dQXAAA0ksL.jpg	2	chow	0.526230
1953	863907417377173506	https://pbs.twimg.com/media/C_03NPeUQAAgrMl.jpg	1	marmot	0.358828
1939	860563773140209665	https://pbs.twimg.com/media/C_FWL0vVwAA13N7.jpg	1	Cardigan	0.583936

```
In [79]: (image_predictions_clean_df.p1_dog | image_predictions_clean_df.p2_dog | ima
```

```
Out[79]: 1751
```

```
In [80]: image_predictions_clean_df = pd.concat([
    image_predictions_clean_df[image_predictions_clean_df.p1_dog][["tweet_id",
        "p1": "prediction", "p1_conf": "prediction_confidence"]], axis=1),
    image_predictions_clean_df[(~image_predictions_clean_df.p1_dog) & image_
        "p2": "prediction", "p2_conf": "prediction_confidence"]], axis=1),
    image_predictions_clean_df[(~image_predictions_clean_df.p1_dog) & (~imag
        "p3": "prediction", "p3_conf": "prediction_confidence"]], axis=1),
    image_predictions_clean_df[~(image_predictions_clean_df.p1_dog | image_p
)])
```

```
In [81]: image_predictions_clean_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2075 entries, 0 to 2074
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   tweet_id              2075 non-null   int64
1   jpg_url               2075 non-null   object
2   prediction             1751 non-null   object
3   prediction_confidence  1751 non-null   float64
dtypes: float64(1), int64(1), object(2)
memory usage: 81.1+ KB
```

```
In [82]: image_predictions_clean_df.head()
```

```
Out[82]:
```

	tweet_id	jpg_url	prediction	prediction_confidence
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh_springer_spaniel	0.4650
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	redbone	0.5068
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	German_shepherd	0.5964
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-IEu.jpg	Rhodesian_ridgeback	0.4081
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	miniature_pinscher	0.5603

Test

```
In [83]: for _, original_row in image_predictions_df.iterrows():
          row = image_predictions_clean_df[image_predictions_clean_df.tweet_id ==
          original_row.tweet_id]

          if original_row.p1_dog:
              assert row.prediction == original_row.p1
              assert row.prediction_confidence == original_row.p1_conf
          elif original_row.p2_dog:
              assert row.prediction == original_row.p2
              assert row.prediction_confidence == original_row.p2_conf
          elif original_row.p3_dog:
              assert row.prediction == original_row.p3
              assert row.prediction_confidence == original_row.p3_conf
          else:
              assert np.isnan(row.prediction)
              assert np.isnan(row.prediction_confidence)
```

Issue #2: Dog breed names could be made consistent and readable

Define

Replace the underscore (_) character on the prediction column and capitalize each word.

Code

```
In [84]: image_predictions_clean_df["prediction"] = image_predictions_clean_df.prediction
```

Test

```
In [85]: image_predictions_clean_df.head()
```

```
Out[85]:
```

	tweet_id	jpg_url	prediction	prediction_confidence
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh Springer Spaniel	0.465074
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	Redbone	0.506826
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	German Shepherd	0.596461
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	Rhodesian Ridgeback	0.408143
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	Miniature Pinscher	0.560311

```
In [86]: image_predictions_clean_df.prediction.str.contains("_").sum()
```

```
Out[86]: 0
```

```
In [87]: image_predictions_clean_df.prediction.value_counts()
```

```
Out[87]: Golden Retriever      173
Labrador Retriever      113
Pembroke                96
Chihuahua               95
Pug                     65
...
Entlebucher            1
Japanese Spaniel        1
Scotch Terrier          1
Standard Schnauzer      1
Bouvier Des Flandres    1
Name: prediction, Length: 113, dtype: int64
```

Issue #3: The predictions are a string but should be a category

Define

Cast prediction as category .

Code

```
In [88]: image_predictions_clean_df["prediction"] = image_predictions_clean_df.predic
```

Test

```
In [89]: image_predictions_clean_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2075 entries, 0 to 2074
Data columns (total 4 columns):
#   Column                Non-Null Count  Dtype
---  -
0   tweet_id              2075 non-null   int64
1   jpg_url               2075 non-null   object
2   prediction            1751 non-null   category
3   prediction_confidence 1751 non-null   float64
dtypes: category(1), float64(1), int64(1), object(1)
memory usage: 71.8+ KB
```

Issue #4: Multiple columns for the doggo stage variable

Define

Convert the doggo , floofer , pupper , puppo columns onto a stage column.

Code

```
In [90]: stages = ["doggo", "floofer", "pupper", "puppo"]
```

We use np.select to assign a stage value depending on each condition.

```
In [91]: twitter_archived_enhanced_clean_df["stage"] = np.select(
    [
        # A stage is defined (one condition per stage)
        twitter_archived_enhanced_clean_df[stage] == stage
        for stage in stages
    ], stages, default=None)
```

```
In [92]: twitter_archived_enhanced_clean_df.stage.value_counts()
```

```
Out[92]: pupper      245
doggo      97
puppo      29
floofer      9
Name: stage, dtype: int64
```

Test

```
In [93]: for stage in stages:
          assert twitter_archived_enhanced_clean_df[
              (twitter_archived_enhanced_clean_df.stage == stage) &
              (twitter_archived_enhanced_clean_df["stage"] != stage)
          ].empty

          assert twitter_archived_enhanced_clean_df[
              twitter_archived_enhanced_clean_df["doggo"].isna() &
              twitter_archived_enhanced_clean_df["floofer"].isna() &
              twitter_archived_enhanced_clean_df["pupper"].isna() &
              twitter_archived_enhanced_clean_df["puppo"].isna()
          ].stage.isna().all()
```

We drop the old columns stage columns

```
In [94]: twitter_archived_enhanced_clean_df = twitter_archived_enhanced_clean_df.drop
```

Issue #5: timestamp is a string, could be a datetime

Define

Convert the timestamp column to datetime .

Code

```
In [95]: twitter_archived_enhanced_clean_df["timestamp"] = pd.to_datetime(twitter_arc
```

Test

```
In [96]: twitter_archived_enhanced_clean_df.info()
```



```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   tweet_id                             2356 non-null   int64
1   in_reply_to_status_id                 78 non-null     float64
2   in_reply_to_user_id                   78 non-null     float64
3   timestamp                             2356 non-null   datetime64[ns, UTC]
4   source                                2356 non-null   object
5   text                                  2356 non-null   object
6   retweeted_status_id                   181 non-null     float64
7   retweeted_status_user_id              181 non-null     float64
8   retweeted_status_timestamp            181 non-null     object
9   expanded_urls                         2297 non-null   object
10  rating_numerator                       2356 non-null   int64
11  rating_denominator                     2356 non-null   int64
12  name                                   2356 non-null   object
13  stage                                  380 non-null    object
dtypes: datetime64[ns, UTC](1), float64(4), int64(3), object(6)
memory usage: 257.8+ KB

```

Issue #6: Multiple doggo ratings on a single tweet

Define

Drop the offending tweets.

Code

```

In [97]: twitter_archived_enhanced_clean_df[twitter_archived_enhanced_clean_df.tweet_
820690176645140481, 675853064436391936, 677716515794329600, 684222868335
704054845121142784, 709198395643068416, 710658690886586372, 713900603437
758467244762497024
]]][["tweet_id", "text", "rating_numerator", "rating_denominator"]]

```

Out[97]:	tweet_id	text	rating_numerator	rating_denominator
433	820690176645140481	The floofs have been released I repeat the flo...	84	70
902	758467244762497024	Why does this never happen at my front door.....	165	150
1120	731156023742988288	Say hello to this unbelievably well behaved sq...	204	170
1228	713900603437621249	Happy Saturday here's 9 puppers on a bench. 99...	99	90
1254	710658690886586372	Here's a brigade of puppers. All look very pre...	80	80
1274	709198395643068416	From left to right:\nCletus, Jerome, Alejandro...	45	50
1351	704054845121142784	Here is a whole flock of puppers. 60/50 I'll ...	60	50
1433	697463031882764288	Happy Wednesday here's a bucket of pups. 44/40...	44	40
1635	684222868335505415	Someone help the girl is being mugged. Several...	121	110
1779	677716515794329600	IT'S PUPPERGEDDON. Total of 144/120 ...I think...	144	120
1843	675853064436391936	Here we have an entire platoon of puppers. Tot...	88	80

```
In [98]: twitter_archived_enhanced_clean_df = twitter_archived_enhanced_clean_df[~twitter_archived_enhanced_clean_df['tweet_id'].isin([820690176645140481, 675853064436391936, 677716515794329600, 684222868335505415, 704054845121142784, 709198395643068416, 710658690886586372, 713900603437621249, 758467244762497024])]

```

Test

```
In [99]: assert twitter_archived_enhanced_clean_df[~twitter_archived_enhanced_clean_df['tweet_id'].isin([820690176645140481, 675853064436391936, 677716515794329600, 684222868335505415, 704054845121142784, 709198395643068416, 710658690886586372, 713900603437621249, 758467244762497024])].empty

```

Issue #7: multiple expressions confused for ratings

Define

When there are multiple matches on a tweet for segments that look like a rating, pick the one with a "10" as denominator.

Code

```
In [100]: twitter_archived_enhanced_clean_df[twitter_archived_enhanced_clean_df.tweet_id.isin([810984652412424192, 716439118184652801, 716439118184652801, 74037318919322974582966214656, 666287406224695296])][["tweet_id", "text", "rating_numerator", "rating_denominator"]]
```

```
Out[100]:
```

	tweet_id	text	rating_numerator	rating_denominator
516	810984652412424192	Meet Sam. She smiles 24/7 & secretly aspir...	24	7
1068	740373189193256964	After so many requests, this is Bretagne. She ...	9	11
1165	722974582966214656	Happy 4/20 from the squad! 13/10 for all https...	4	20
1202	716439118184652801	This is Bluebert. He just saw that both #Final...	50	50
1662	682962037429899265	This is Darrel. He just robbed a 7/11 and is i...	7	11
2335	666287406224695296	This is an Albanian 3 1/2 legged Episcopalian...	1	2

```
In [101]: new_ratings_df = twitter_archived_enhanced_clean_df.text.str.extract(r"(?P<rating_numerator>\d+)/(?P<rating_denominator>10)")
new_ratings_df
```

Out[101]:

	rating_numerator	rating_denominator
0	13	10
1	13	10
2	12	10
3	13	10
4	12	10
...
2351	5	10
2352	6	10
2353	9	10
2354	7	10
2355	8	10

	rating_numerator	rating_denominator
0	13	10
1	13	10
2	12	10
3	13	10
4	12	10
...
2351	5	10
2352	6	10
2353	9	10
2354	7	10
2355	8	10

2345 rows × 2 columns

```
In [102...] new_ratings_df.loc[
    ~new_ratings_df.isnull().any(axis=1),
    ["rating_numerator", "rating_denominator"]
] = new_ratings_df[~new_ratings_df.isnull().any(axis=1)][["rating_numerator"
```

```
In [103...] twitter_archived_enhanced_clean_df.loc[
    new_ratings_df.index,
    ["rating_numerator", "rating_denominator"]
] = new_ratings_df[["rating_numerator", "rating_denominator"]]
```

```
In [104...] twitter_archived_enhanced_clean_df = twitter_archived_enhanced_clean_df[~(
    twitter_archived_enhanced_clean_df.rating_numerator.isna() |
    twitter_archived_enhanced_clean_df.rating_denominator.isna()
)]
```

Test

```
In [105...] twitter_archived_enhanced_clean_df[twitter_archived_enhanced_clean_df.tweet_
    810984652412424192, 716439118184652801, 716439118184652801, 740373189193
    722974582966214656, 666287406224695296
]]][["tweet_id", "text", "rating_numerator", "rating_denominator"]]
```

Out[105]:	tweet_id	text	rating_numerator	rating_denominator
1068	740373189193256964	After so many requests, this is Bretagne. She ...	14	10
1165	722974582966214656	Happy 4/20 from the squad! 13/10 for all https...	13	10
1202	716439118184652801	This is Bluebert. He just saw that both #Final...	11	10
1662	682962037429899265	This is Darrel. He just robbed a 7/11 and is i...	10	10
2335	666287406224695296	This is an Albanian 3 1/2 legged Episcopalian...	9	10

Issue #8: There tweets with fractional ratings

Define

Convert the rating numerator and denominator to float. Parse the fractional ratings properly.

Code

```
In [106...] twitter_archived_enhanced_clean_df[["rating_numerator", "rating_denominator"]
```

```
In [107...] new_ratings_df = twitter_archived_enhanced_clean_df.text.str.extract(
    r"(?P<rating_numerator>\d+\.\d+)/(?P<rating_denominator>10)"
).dropna()
new_ratings_df
```

Out[107]:	rating_numerator	rating_denominator
45	13.5	10
340	9.75	10
695	9.75	10
763	11.27	10
1689	9.5	10
1712	11.26	10

```
In [108...] twitter_archived_enhanced_clean_df.loc[
    new_ratings_df.index,
    ["rating_numerator", "rating_denominator"]
] = new_ratings_df[["rating_numerator", "rating_denominator"]].astype(float)
```

Test

```
In [109]: twitter_archived_enhanced_clean_df[twitter_archived_enhanced_clean_df.tweet_id.isin([680494726643068929, 778027034220126208, 786709082849828864])] [['tweet_id', 'text', 'rating_numerator', 'rating_denominator']]
```

```
Out[109]:
```

	tweet_id	text	rating_numerator	rating_denominator
695	786709082849828864	This is Logan, the Chow who lived. He solemnly...	9.75	10.0
763	778027034220126208	This is Sophie. She's a Jubilant Bush Pupper. ...	11.27	10.0
1712	680494726643068929	Here we have uncovered an entire battalion of ...	11.26	10.0

Issue #9: The rating is a single variable represented in two columns

Define

Create a new column named `rating` which is `rating_numerator` divided by `rating_denominator`.

Code

```
In [110]: twitter_archived_enhanced_clean_df["rating"] = twitter_archived_enhanced_clean_df["rating_numerator"] / twitter_archived_enhanced_clean_df["rating_denominator"]
```

```
In [111]: twitter_archived_enhanced_clean_df.rating.sample(5)
```

```
Out[111]: 1408    1.2
          939    1.3
          1891   1.0
          2310   0.2
          280    1.2
          Name: rating, dtype: float64
```

Test

```
In [112]: for index, row in twitter_archived_enhanced_clean_df.iterrows():
          ~~~~~
          assert row.rating == row.rating_numerator / row.rating_denominator
```

Issue #10: Multiple tables to represent the same observational unit: the tweet

Define

Merge all dataframes on the `tweet_id` column to a dataframe called `twitter_archive_master_df`.

Code

```
In [113]: twitter_archive_master_df = pd.merge(
            twitter_archived_enhanced_clean_df,
            image_predictions_clean_df,
            on=["tweet_id"],
            how="left"
        )
twitter_archive_master_df = pd.merge(
    twitter_archive_master_df,
    tweet_json_txt_clean_df,
    on=["tweet_id"],
    how="left"
)
```

Test

```
In [114]: twitter_archive_master_df.head()
```

```
Out[114]:
```

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56+00:00	href="r/dowr
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27+00:00	href="r/dowr
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03+00:00	href="r/dowr
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51+00:00	href="r/dowr
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24+00:00	href="r/dowr

```
In [115]: twitter_archive_master_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2340 entries, 0 to 2339
Data columns (total 20 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   tweet_id                             2340 non-null   int64
1   in_reply_to_status_id                 74 non-null     float64
2   in_reply_to_user_id                  74 non-null     float64
3   timestamp                             2340 non-null   datetime64[ns, UTC]
4   source                                2340 non-null   object
5   text                                  2340 non-null   object
6   retweeted_status_id                  181 non-null     float64
7   retweeted_status_user_id             181 non-null     float64
8   retweeted_status_timestamp           181 non-null     object
9   expanded_urls                        2284 non-null   object
10  rating_numerator                      2340 non-null   float64
11  rating_denominator                   2340 non-null   float64
12  name                                  2340 non-null   object
13  stage                                380 non-null     object
14  rating                               2340 non-null   float64
15  jpg_url                              2062 non-null   object
16  prediction                           1739 non-null   category
17  prediction_confidence                 1739 non-null   float64
18  retweet_count                        2311 non-null   float64
19  favorite_count                       2311 non-null   float64
dtypes: category(1), datetime64[ns, UTC](1), float64(10), int64(1), object
(7)
memory usage: 372.9+ KB
```

Issue #11: There are 78 replies and 181 retweets on twitter_archived_enhanced_df

Define

Drop rows where in_reply_to_status_id or retweeted_status_id are not
NaN . Drop the unnecessary columns.

Code

```
In [116]: twitter_archive_master_df[~(twitter_archive_master_df.in_reply_to_status_id.
```

```
Out[116]: 255
```

```
In [117]: twitter_archive_master_df = twitter_archive_master_df[twitter_archive_master
```

Test

We drop the unnecessary columns AFTER the test.


```
In [118... twitter_archive_master_df[~(twitter_archive_master_df.in_reply_to_status_id.
```

```
Out[118]: 0
```

```
In [119... twitter_archive_master_df = twitter_archive_master_df.drop(  
    [  
        "in_reply_to_status_id", "in_reply_to_user_id", "retweeted_status_id"  
        "retweeted_status_timestamp"  
    ],  
    axis=1  
)
```

```
In [120... twitter_archive_master_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 2085 entries, 0 to 2339  
Data columns (total 15 columns):  
#   Column                                Non-Null Count  Dtype  
---  ---                                -  
0   tweet_id                             2085 non-null   int64  
1   timestamp                             2085 non-null   datetime64[ns, UTC]  
2   source                               2085 non-null   object  
3   text                                 2085 non-null   object  
4   expanded_urls                        2082 non-null   object  
5   rating_numerator                     2085 non-null   float64  
6   rating_denominator                   2085 non-null   float64  
7   name                                 2085 non-null   object  
8   stage                               336 non-null    object  
9   rating                               2085 non-null   float64  
10  jpg_url                              1959 non-null   object  
11  prediction                           1655 non-null   category  
12  prediction_confidence                 1655 non-null   float64  
13  retweet_count                         2077 non-null   float64  
14  favorite_count                       2077 non-null   float64  
dtypes: category(1), datetime64[ns, UTC](1), float64(6), int64(1), object  
(6)  
memory usage: 251.3+ KB
```

Issue #12: There were missing tweets when fetching the retweet and favourite counts with Twitter's API

Define

Drop columns where retweet_count or favorite_count as NaN .

Code

```
In [121... twitter_archive_master_df[twitter_archive_master_df.retweet_count.isna() | t
```

```
Out[121]: 8
```

```
In [122... twitter_archive_master_df = twitter_archive_master_df[~(twitter_archive_mast
```

Test

```
In [123]: twitter_archive_master_df[twitter_archive_master_df.retweet_count.isna() | t  
Out[123]: 0
```

Issue #13: There are 324 records without a dog prediction on image_predictions_df

Define

Drop rows where prediction is NaN .

Code

```
In [124]: twitter_archive_master_df.columns  
Out[124]: Index(['tweet_id', 'timestamp', 'source', 'text', 'expanded_urls',  
                'rating_numerator', 'rating_denominator', 'name', 'stage', 'rating',  
                'jpg_url', 'prediction', 'prediction_confidence', 'retweet_count',  
                'favorite_count'],  
               dtype='object')  
  
In [125]: twitter_archive_master_df.prediction.isna().sum()  
Out[125]: 430  
  
In [126]: twitter_archive_master_df = twitter_archive_master_df[~twitter_archive_maste
```

Test

```
In [127]: twitter_archive_master_df.prediction.isna().sum()  
Out[127]: 0
```

Issue #14: On tweet 670842764863651840 the rating is invalid as the subject is not a doggo

Define

Drop the offending tweet.

Code

```
In [128... twitter_archive_master_df[twitter_archive_master_df.tweet_id == 670842764863
```

```
Out[128]: tweet_id timestamp source text expanded_urls rating_numerator rating_denominator
```

The tweet has already been taken care of by another fix.

Test

Issue #15: Remove outlier on tweet 749981277374128128

Define

Drop the offending tweet.

Code

```
In [129... twitter_archive_master_df[twitter_archive_master_df.tweet_id == 749981277374
```

```
Out[129]: tweet_id timestamp source text expanded_urls rating_numerator rating_denominator
```

Test

The tweet has already been taken care of by another fix.

Issue #16: Invalid name "a" on 55 tweets

Define

Replace "a" with None on column name .

Code

```
In [130... twitter_archive_master_df.loc[twitter_archive_master_df.name == "a", "name"]
```

Test

```
In [131... (twitter_archive_master_df.name == "a").sum()
```

```
Out[131]: 0
```

Storing Data

Save gathered, assessed, and cleaned master dataset to a CSV file named "twitter_archive_master.csv".

Saving onto a CSV file is actually quite trivial:

```
In [132...] twitter_archive_master_df.to_csv(data_dir / "twitter_archive_master.csv", in
```

However, if we load this CSV again, we lose some of the column type information:

```
In [133...] pd.read_csv(data_dir / "twitter_archive_master.csv").info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1647 entries, 0 to 1646
Data columns (total 15 columns):
#   Column                Non-Null Count  Dtype
---  -
0   tweet_id              1647 non-null   int64
1   timestamp             1647 non-null   object
2   source                1647 non-null   object
3   text                  1647 non-null   object
4   expanded_urls         1647 non-null   object
5   rating_numerator      1647 non-null   float64
6   rating_denominator    1647 non-null   float64
7   name                  1602 non-null   object
8   stage                 256 non-null    object
9   rating                1647 non-null   float64
10  jpg_url               1647 non-null   object
11  prediction            1647 non-null   object
12  prediction_confidence 1647 non-null   float64
13  retweet_count         1647 non-null   float64
14  favorite_count        1647 non-null   float64
dtypes: float64(6), int64(1), object(8)
memory usage: 193.1+ KB
```

Due to this limitation, I think it's better to use a portable and type aware format like feather .

```
In [134...] twitter_archive_master_df.reset_index().to_feather(data_dir / "twitter_archi
```

```
In [135...] pd.read_feather(data_dir / "twitter_archive_master.feather").info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1647 entries, 0 to 1646
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   index                                1647 non-null   int64
1   tweet_id                             1647 non-null   int64
2   timestamp                             1647 non-null   datetime64[ns, UTC]
3   source                                1647 non-null   object
4   text                                  1647 non-null   object
5   expanded_urls                         1647 non-null   object
6   rating_numerator                      1647 non-null   float64
7   rating_denominator                   1647 non-null   float64
8   name                                  1602 non-null   object
9   stage                                256 non-null    object
10  rating                                1647 non-null   float64
11  jpg_url                              1647 non-null   object
12  prediction                           1647 non-null   category
13  prediction_confidence                1647 non-null   float64
14  retweet_count                        1647 non-null   float64
15  favorite_count                       1647 non-null   float64
dtypes: category(1), datetime64[ns, UTC](1), float64(6), int64(2), object
(6)
memory usage: 199.7+ KB

```

We could have also save to a sqlite table as well.

```

In [136... database = data_dir / "twitter_archive_master.sqlite"
database.unlink(missing_ok=True)

```

```

In [137... with sqlite3.connect(database) as connection:
    twitter_archive_master_df.to_sql(name='twitter_archive_master', con=conn

```

```

In [138... with sqlite3.connect(database) as connection:
    pd.read_sql(sql="SELECT * FROM twitter_archive_master", con=connection).

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1647 entries, 0 to 1646
Data columns (total 16 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   index                                1647 non-null   int64
1   tweet_id                            1647 non-null   int64
2   timestamp                           1647 non-null   object
3   source                              1647 non-null   object
4   text                                1647 non-null   object
5   expanded_urls                       1647 non-null   object
6   rating_numerator                    1647 non-null   float64
7   rating_denominator                 1647 non-null   float64
8   name                                1602 non-null   object
9   stage                              256 non-null    object
10  rating                              1647 non-null   float64
11  jpg_url                             1647 non-null   object
12  prediction                           1647 non-null   object
13  prediction_confidence               1647 non-null   float64
14  retweet_count                       1647 non-null   float64
15  favorite_count                      1647 non-null   float64
dtypes: float64(6), int64(2), object(8)
memory usage: 206.0+ KB

```

But once again we lose the type information.

Analyzing and Visualizing Data

In this section, analyze and visualize your wrangled data. You must produce at least **three (3) insights and one (1) visualization**.

The following cells contain the exploratory work that led to the insights.

```
In [139]: twitter_archive_master_df.name.value_counts(dropna=False)[:10]
```

```

Out[139]: None      390
          None      45
          Cooper    10
          Charlie     9
          Lucy       9
          Tucker     9
          Oliver     9
          Penny      8
          Sadie      7
          the        7
          Name: name, dtype: int64

```

```
In [140]: twitter_archive_master_df.rating.mean()
```

```
Out[140]: 1.0820752884031573
```

```
In [141]: twitter_archive_master_df.rating.std()
```

Out[141]: 0.17777543390270223

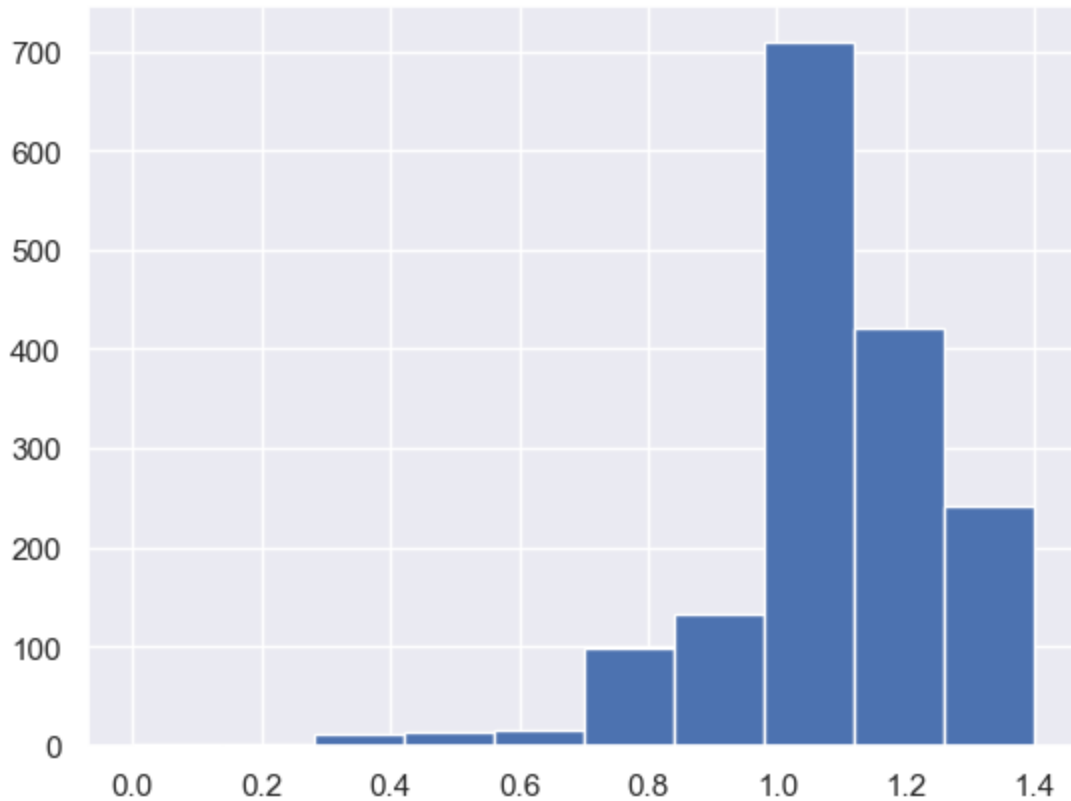
```
In [142]: np.percentile(twitter_archive_master_df.rating, [25, 50, 75])
```

Out[142]: array([1. , 1.1, 1.2])

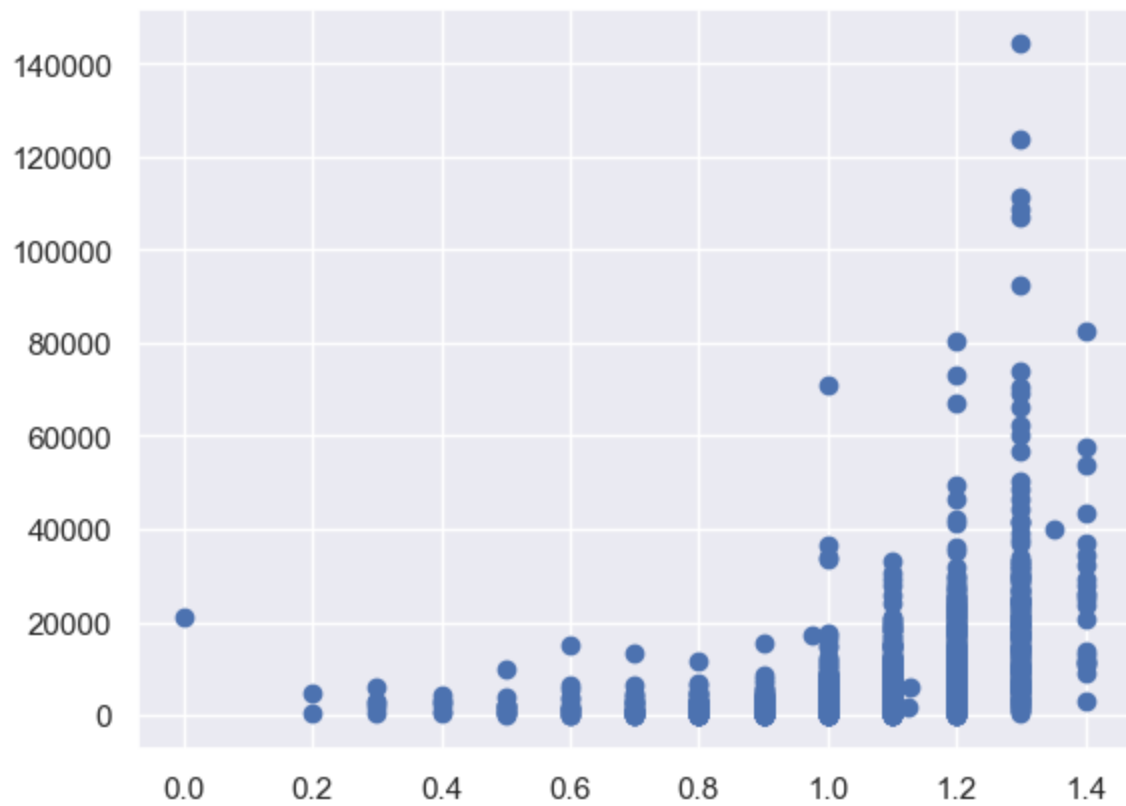
```
In [143]: twitter_archive_master_df.rating.min(), twitter_archive_master_df.rating.max
```

Out[143]: (0.0, 1.4)

```
In [144]: twitter_archive_master_df.rating.hist();
```



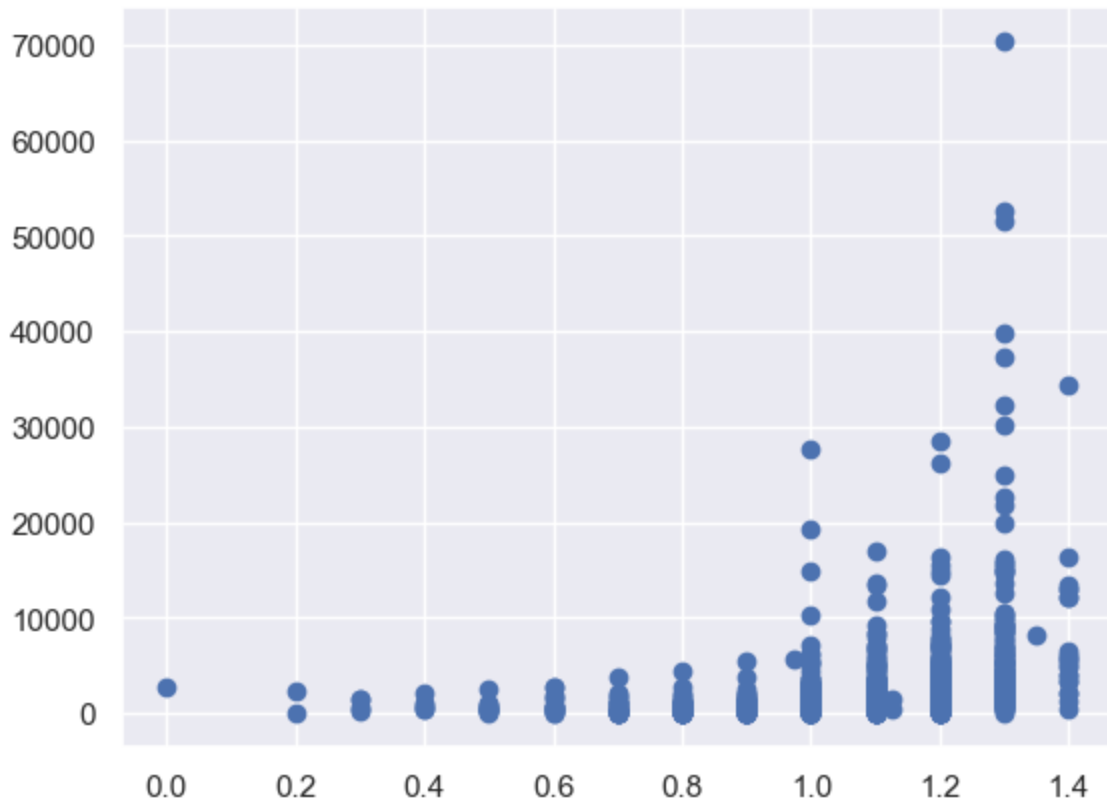
```
In [145]: plt.scatter(
    twitter_archive_master_df.rating,
    twitter_archive_master_df.favorite_count
);
```



```
In [146]: twitter_archive_master_df.rating.corr(twitter_archive_master_df.favorite_cou
```

```
Out[146]: 0.4112524328126853
```

```
In [147]: plt.scatter(  
    twitter_archive_master_df.rating,  
    twitter_archive_master_df.retweet_count  
);
```

```
In [148]: twitter_archive_master_df.rating.corr(twitter_archive_master_df.favorite_cou
```

```
Out[148]: 0.4112524328126853
```

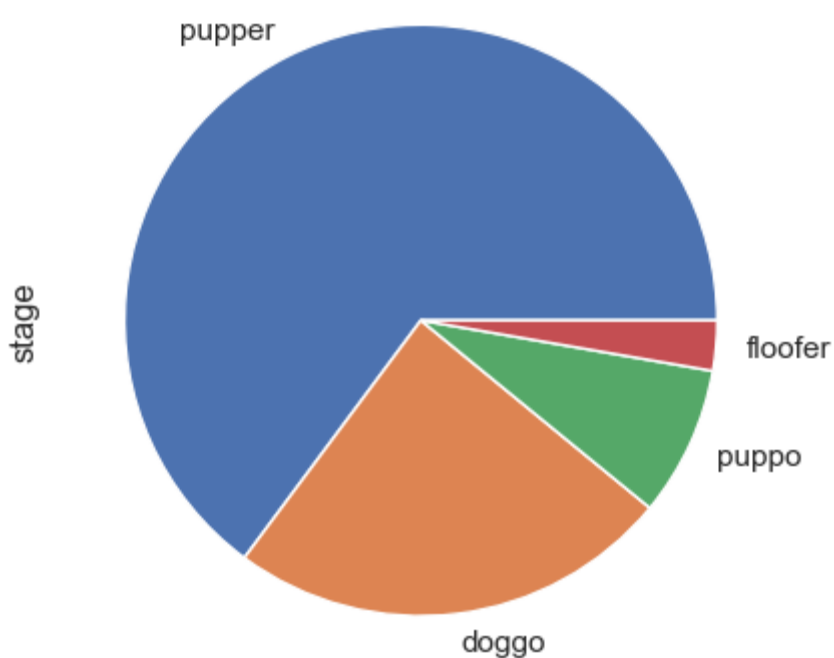
```
In [149]: twitter_archive_master_df.prediction.value_counts()[:10]
```

```
Out[149]: Golden Retriever      154
Labrador Retriever      100
Pembroke                 94
Chihuahua                90
Pug                      62
Toy Poodle               49
Chow                     47
Pomeranian               41
Samoyed                  41
Malamute                 33
Name: prediction, dtype: int64
```

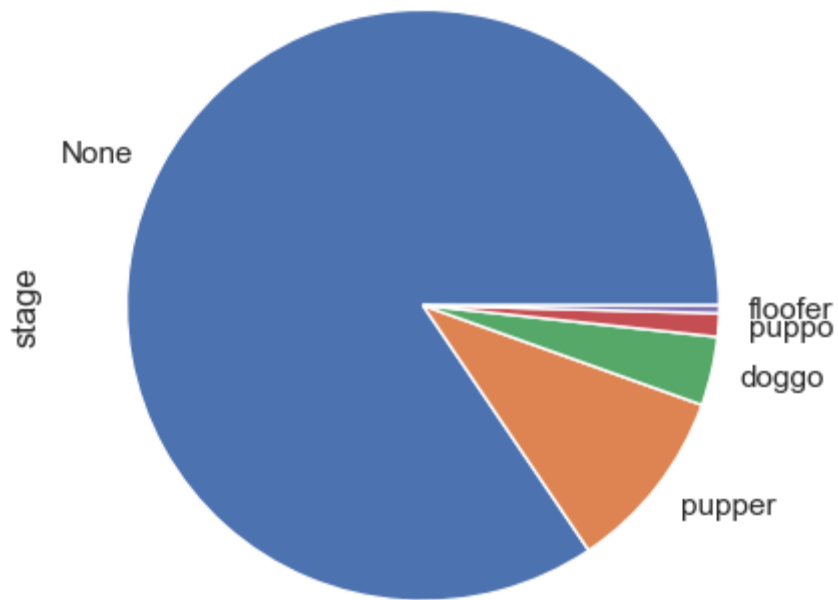
```
In [150]: twitter_archive_master_df.groupby("prediction").rating.mean().sort_values(as
```

```
Out[150]: prediction
Bouvier Des Flandres    1.300000
Saluki                  1.250000
Briard                  1.233333
Tibetan Mastiff         1.225000
Border Terrier          1.214286
Silky Terrier           1.200000
Standard Schnauzer      1.200000
Irish Setter            1.175000
Gordon Setter           1.175000
Samoyed                 1.173171
Name: rating, dtype: float64
```

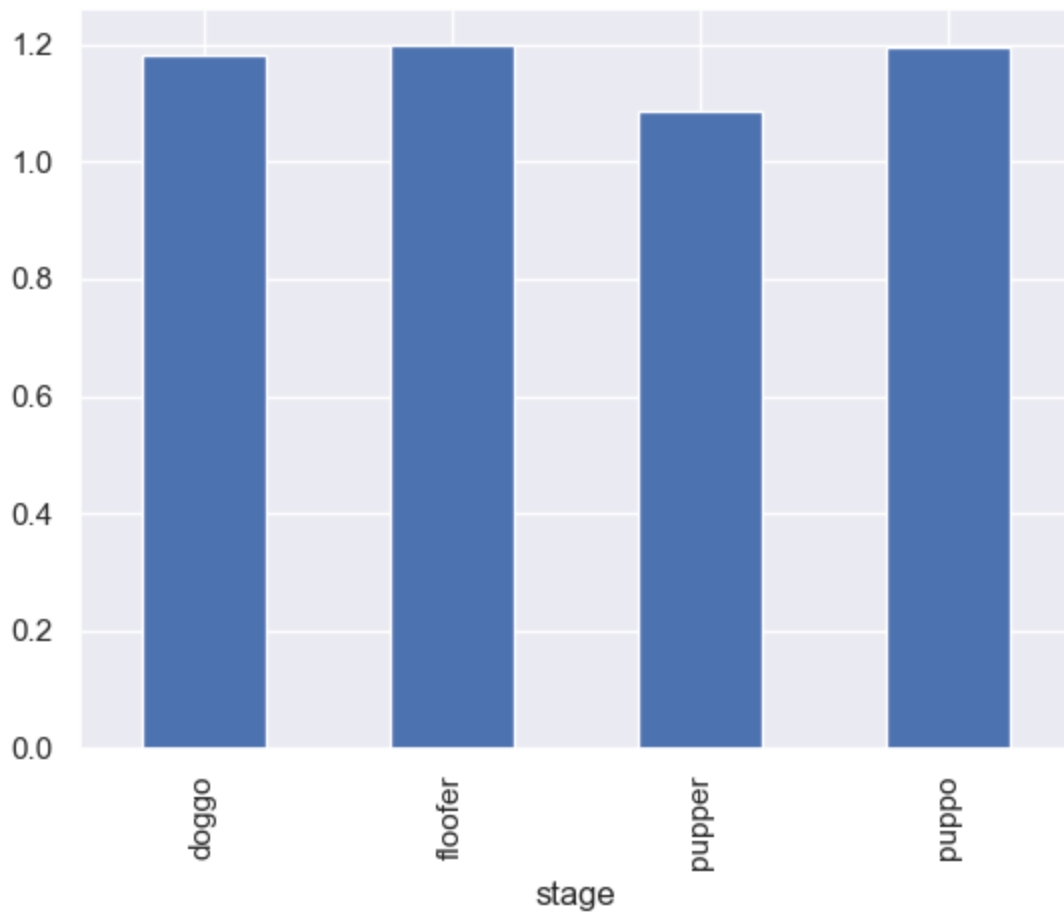
```
In [151]: twitter_archive_master_df.stage.value_counts().plot(kind="pie");
```



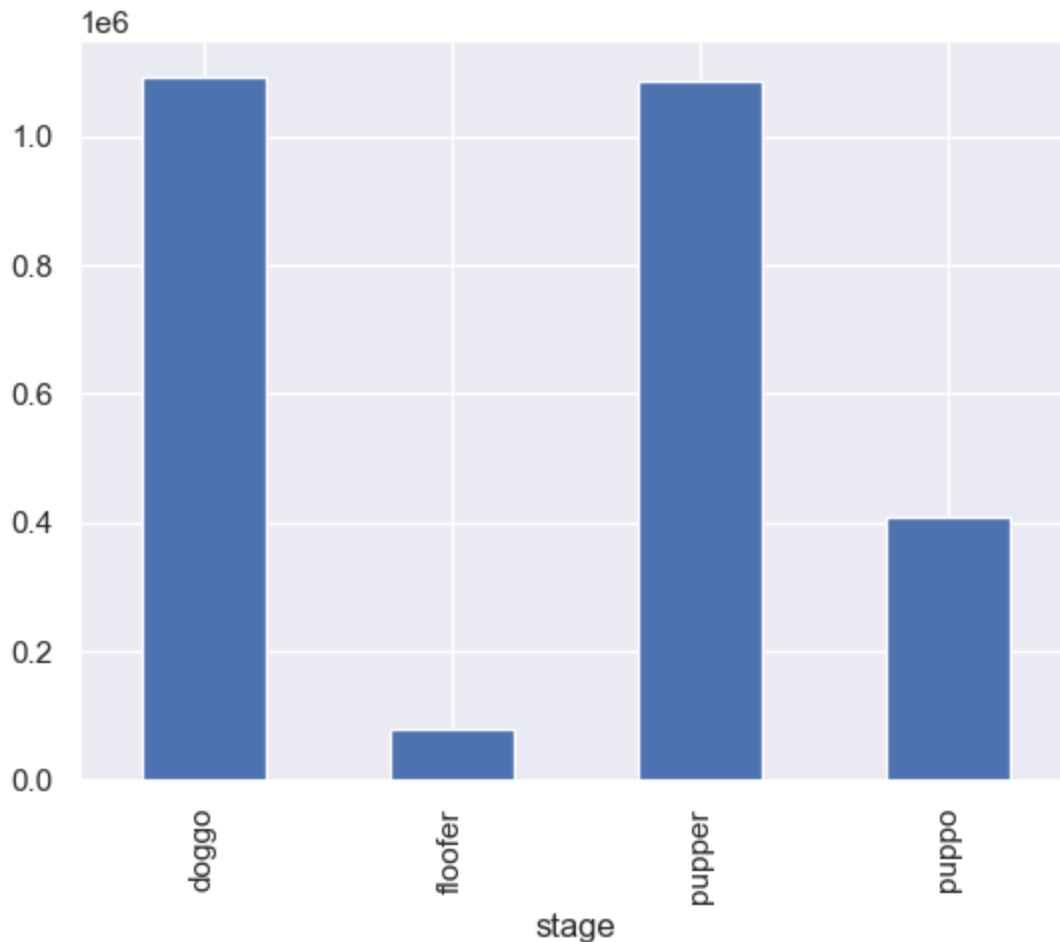
```
In [152]: twitter_archive_master_df.stage.value_counts(dropna=False).plot(kind="pie");
```



In [153]: `twitter_archive_master_df.groupby("stage").rating.mean().plot(kind="bar");`



In [154]: `twitter_archive_master_df.groupby("stage").favorite_count.sum().plot(kind="b`



Insights:

1. Most ratings are over 10/10 (1.0)
2. Golden retrievers are the most popular breed
3. There's a slight correlation between rating and favorite and retweet counts
4. All stages are evenly rated.
5. All doggos are good doggos.

Insight: Most ratings are over 10/10 (1.0)

The "Average Rating" histogram, as well as the 5 number summary clearly shows that most ratings are over 1.0, which means (assuming that all valid ratings have a 10 as denominator) that most ratings are over 10/10. We used

`twitter_archive_master_df.rating` for this insight, which was computed dividing the `rating_numerator` column over the `rating_denominator` column. In turn, this columns, which were originally provided, required some cleaning as they were invalid ratings (due to pattern matching) as well as some outliers.

```
In [155]: twitter_archive_master_df.rating.mean() # mean
```

```
Out[155]: 1.0820752884031573
```

```
In [156]: twitter_archive_master_df.rating.std() # standard deviation
```

```
Out[156]: 0.17777543390270223
```

```
In [157]: np.percentile(twitter_archive_master_df.rating, [25, 50, 75]) # quartiles
```

```
Out[157]: array([1. , 1.1, 1.2])
```

```
In [158]: twitter_archive_master_df.rating.min(), twitter_archive_master_df.rating.max
```

```
Out[158]: (0.0, 1.4)
```

Insight: Golden retrievers are the most popular breed

The following ranking (as well as the chart "Popular Breed Prediction" included in the visualizations section) shows the 10 most popular (predicted) breeds amongst the rated tweets. The chart ""

```
In [159]: twitter_archive_master_df.prediction.value_counts()[:10]
```

```
Out[159]: Golden Retriever      154
          Labrador Retriever    100
          Pembroke              94
          Chihuahua             90
          Pug                   62
          Toy Poodle            49
          Chow                  47
          Pomeranian            41
          Samoyed               41
          Malamute              33
          Name: prediction, dtype: int64
```

The prediction column came from a consolidation of the p1 , p2 , and p3 columns from the image_predictions.tsv file. I select the first column that predicted a dog breed for the prediction column. Afterwards we cleaned up the strings to make them readable, and turned them into a category. If you need a reason for why this breed is so popular, just check the visualization section.

Insight: There's a slight correlation between rating and favorite and retweet counts

The following correlation factors, shows there's a slight correlation between `rating` and `favorite_count` and `retweet_count` columns. These last two were missing from the original CSV file and had to be fetched from Twitter's API. Not much cleaning was needed, but there were a handful of deleted tweets for which we couldn't get these values and were dropped eventually.

```
In [160]: twitter_archive_master_df.rating.corr(twitter_archive_master_df.favorite_cou
```

```
Out[160]: 0.4112524328126853
```

```
In [161]: twitter_archive_master_df.rating.corr(twitter_archive_master_df.retweet_coun
```

```
Out[161]: 0.3117353235682869
```

The "Rating vs. Favorite Count" and "Rating vs. Retweet Count" scatter plots are consistent with these correlation factors.

Insight: All stages are evenly rated

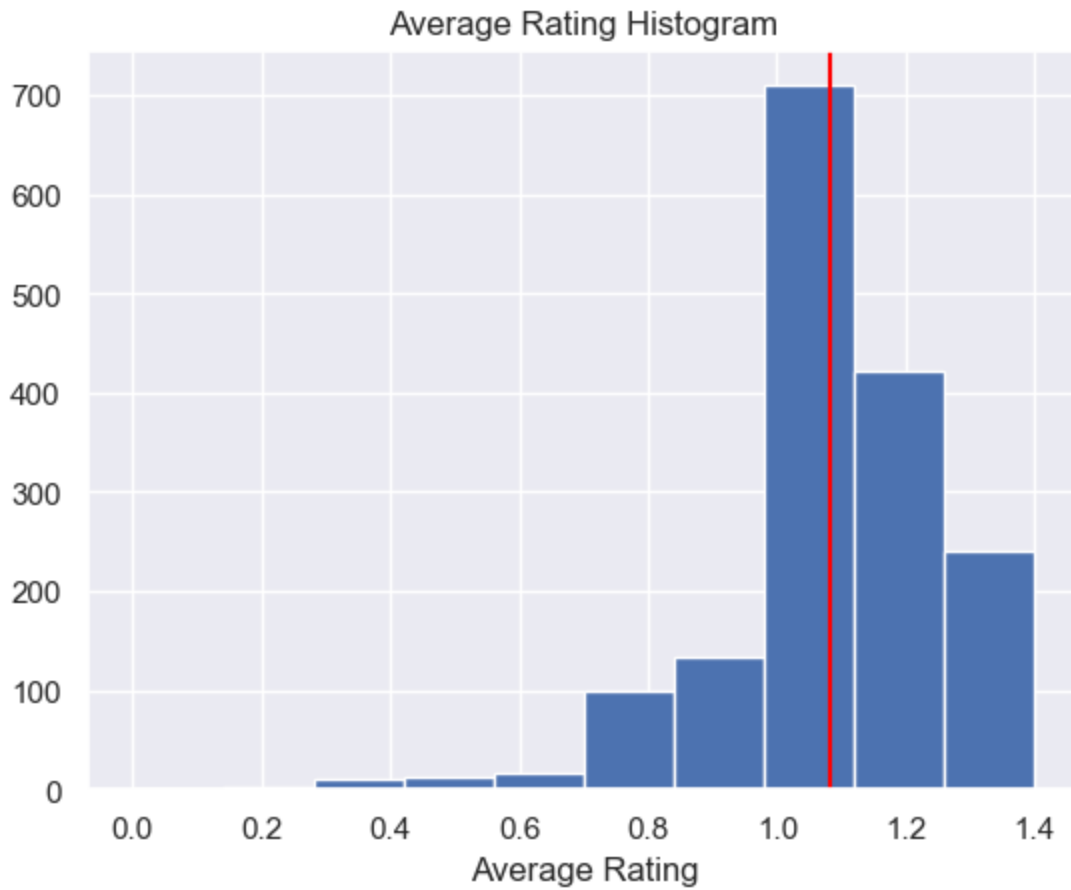
The "Stage Rating Average" clearly shows that all stages are similarly rated (on average). The `stage` column is a combination of the columns `doggo` , `pupper` , `puppo` and `fluffer` from `twitter-archive-enhanced.csv` . When each column had a value (with the same string as the name of the column) we took that as the stage of the individual.

Insight: All doggos are good doggos

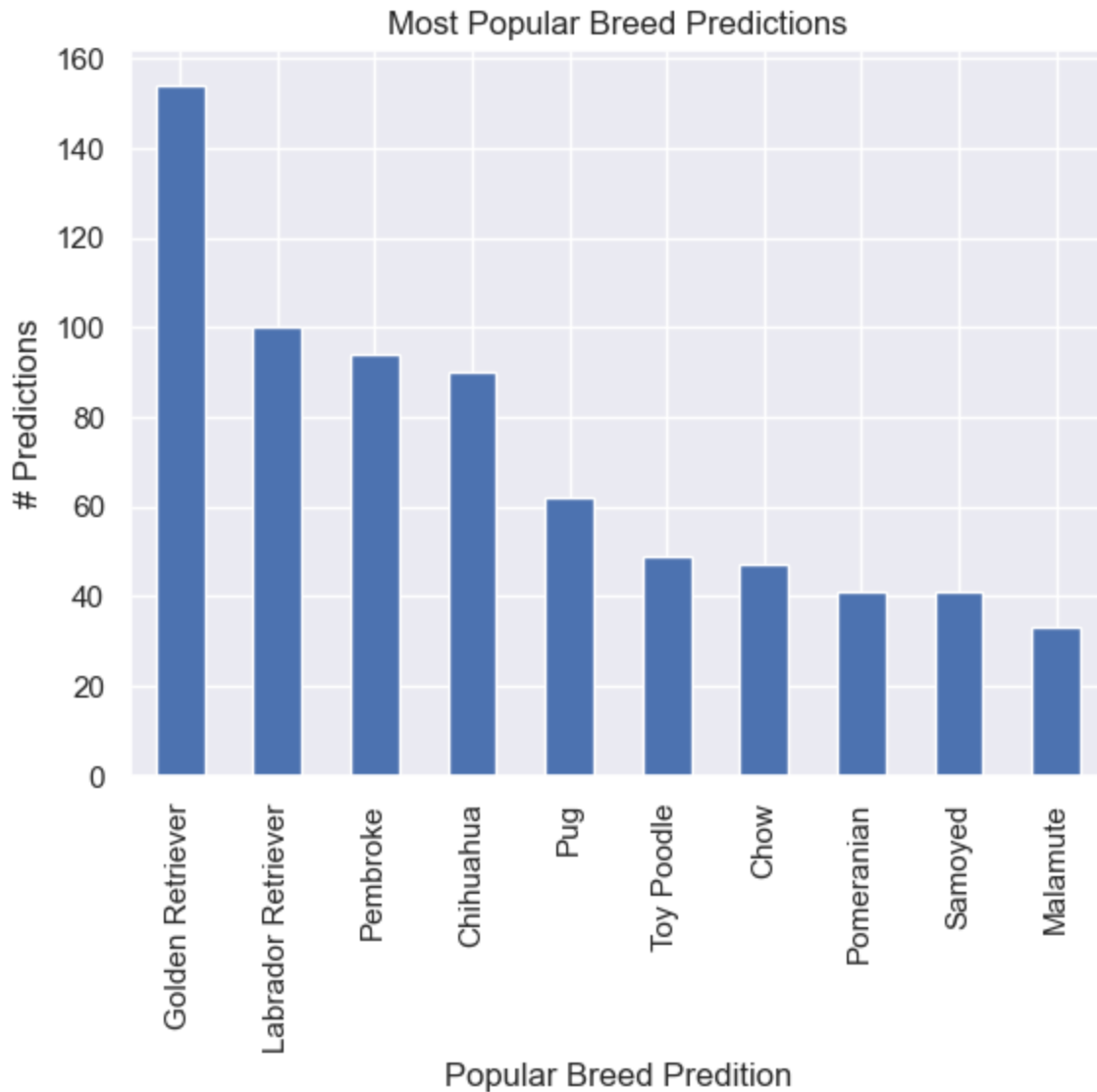
Juss a fact of life.

Visualization

```
In [162]: plt.hist(twitter_archive_master_df.rating);  
plt.xlabel('Average Rating')  
plt.title('Average Rating Histogram')  
plt.axvline(twitter_archive_master_df.rating.mean(), color="red");
```



```
In [163... twitter_archive_master_df.prediction.value_counts()[:10].plot(kind="bar");  
plt.xlabel('Popular Breed Prediction');  
plt.ylabel('# Predictions');  
plt.title('Most Popular Breed Predictions');
```



```
In [164...] twitter_archive_master_df[twitter_archive_master_df.prediction == "Golden Re
```

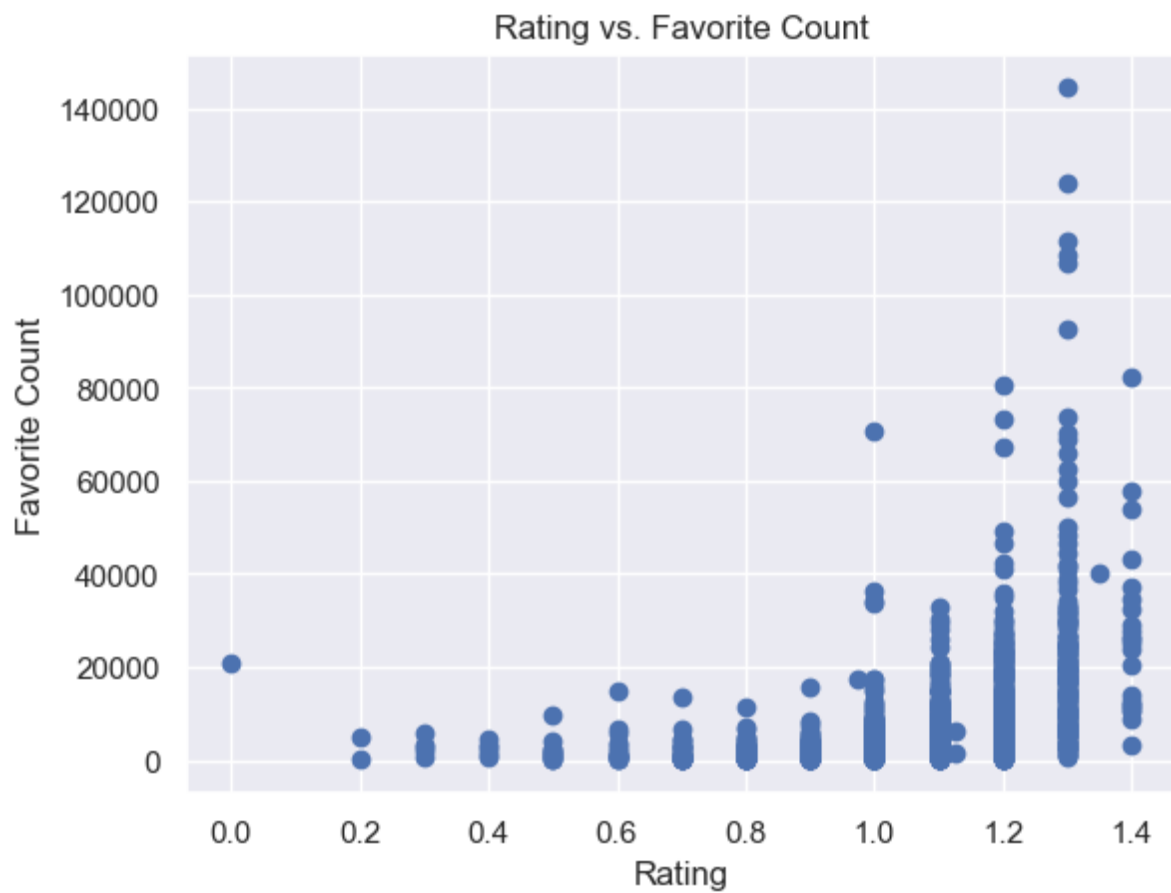
```
Out[164]: 482    https://pbs.twimg.com/media/C04taUjWIAA6Mo4.jpg  
84    https://pbs.twimg.com/media/DCnll_dUQAakBdG.jpg  
115    https://pbs.twimg.com/media/DBQwlFCXkAACSkI.jpg  
47    https://pbs.twimg.com/media/DEF2-_hXoAAs62q.jpg  
Name: jpg_url, dtype: object
```



```
In [165... urls = twitter_archive_master_df[twitter_archive_master_df.prediction == "Go
images = [
    plt.imread(urllib.request.urlopen(url), format="jpg")
    for url in urls
]
fig, ((ax1, ax2), (ax3, ax4)) = plt.subplots(2, 2, sharex=True, sharey=True)
ax1.imshow(images[0]);
ax1.axis('off');
ax2.imshow(images[1]);
ax2.axis('off');
ax3.imshow(images[2]);
ax3.axis('off');
ax4.imshow(images[3]);
ax4.axis('off');
```



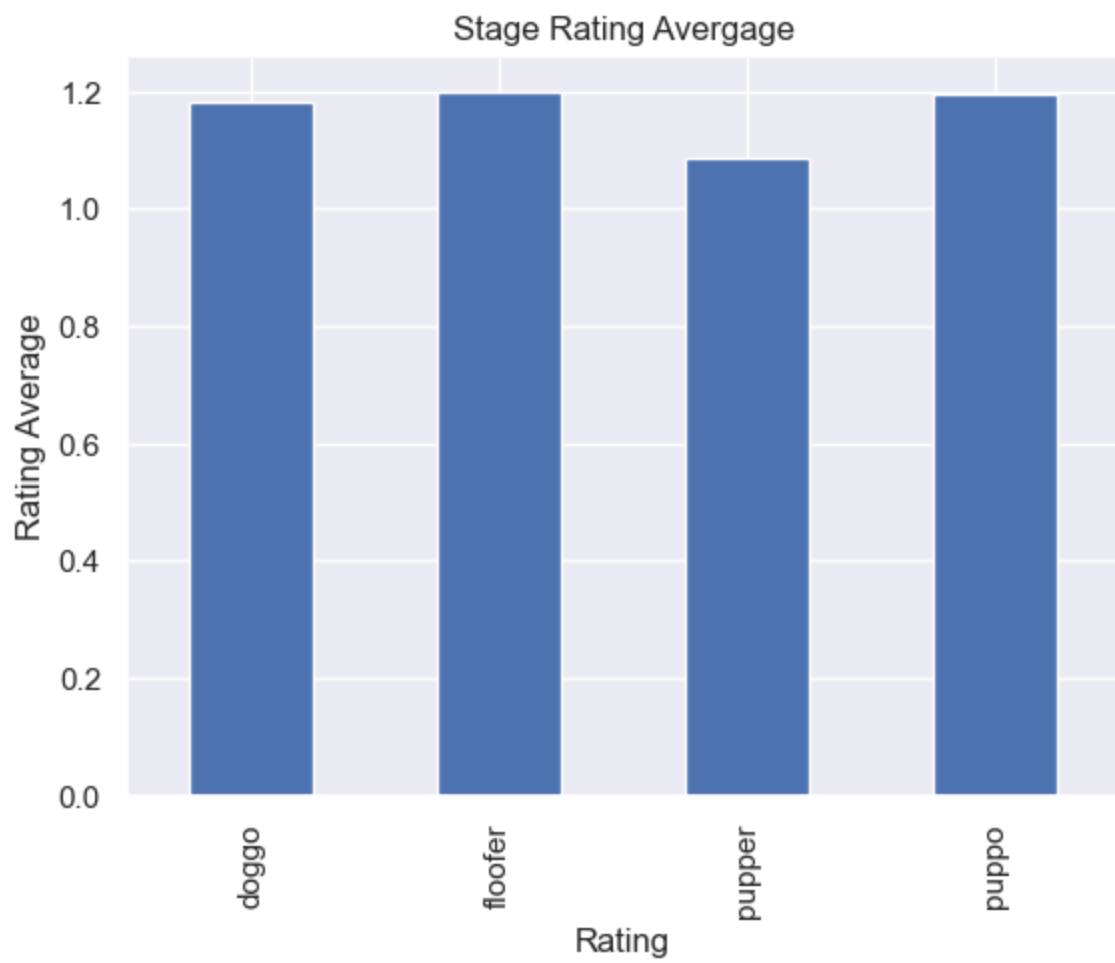
```
In [166... plt.scatter(
    twitter_archive_master_df.rating,
    twitter_archive_master_df.favorite_count
);
plt.xlabel('Rating');
plt.ylabel('Favorite Count');
plt.title('Rating vs. Favorite Count');
```



```
In [167... plt.scatter(  
    twitter_archive_master_df.rating,  
    twitter_archive_master_df.retweet_count  
);  
plt.xlabel('Rating');  
plt.ylabel('Retweet Count');  
plt.title('Rating vs. Retweet Count');
```



```
In [168... twitter_archive_master_df.groupby("stage").rating.mean().plot(kind="bar");  
plt.xlabel('Rating');  
plt.ylabel('Rating Average');  
plt.title('Stage Rating Average');
```



In []: