Project: Wrangling and Analyze 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)

In [2]:

```
#import libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import os
import json
import requests
%matplotlib inline
```

In [3]:

```
#importing the Enhanced twitter archive dataset
twitter_archive = pd.read_csv('twitter-archive-enhanced.csv')
twitter_archive
```

Out[3]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	s
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	href="http://twitter.com/download/ip
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.com/download/ip
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://twitter.com/download/ip
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	href="http://twitter.com/download/ip
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	href="http://twitter.com/download/ip
...	...	...	...	...	
2351	666049248165822465	NaN	NaN	2015-11-16 00:24:50 +0000	href="http://twitter.com/download/ip

2352	666044226329800704	NaN	NaN	2015-11-16 00:04:52 +0000	href="http://twitter.com/download/ip
2353	666033412701032449	NaN	NaN	2015-11-15 23:21:54 +0000	href="http://twitter.com/download/ip
2354	666029285002620928	NaN	NaN	2015-11-15 23:05:30 +0000	href="http://twitter.com/download/ip
2355	666020888022790149	NaN	NaN	2015-11-15 22:32:08 +0000	href="http://twitter.com/download/ip

2356 rows × 17 columns

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

dowloading twitter image predictions

In [4]:
```
url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictio

response = requests.get(url)

with open('image-prediction.tsv', mode = 'wb') as file:
    file.write(response.content)
```

In [5]:
```
#loading image predictions data into pandas DataFrame
image_pred_df = pd.read_csv('image-prediction.tsv', sep = '\t')
image_pred_df
```

Out[5]:

	tweet_id	jpg_url	img_num	p1	p
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.4
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.5
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.5
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.4
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.5
...	...	...	...	...	
2070	891327558926688256	https://pbs.twimg.com/media/DF6hr6BUMAAzZgT.jpg	2	basset	0.5
2071	891689557279858688	https://pbs.twimg.com/media/DF_q7IAWsAEuuN8.jpg	1	paper_towel	0.1

2072	891815181378084864	https://pbs.twimg.com/media/DGBdLU1WsAANxJ9.jpg	1	Chihuahua	0.7
2073	892177421306343426	https://pbs.twimg.com/media/DGGmoV4XsAAUL6n.jpg	1	Chihuahua	0.3
2074	892420643555336193	https://pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg	1	orange	0.0

2075 rows × 12 columns

In [ ]:

```

```

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

loading tweets data into pandas DataFrame

In [6]:

```
with open('tweet-json.txt') as file: #loading tweets data in pandas DataFame
    twitter_api = pd.read_json(file, lines = True, encoding = 'utf-8')

twitter_api.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 31 columns):
 #   Column                         Non-Null Count  Dtype
---  ------                         --------------  -----
 0   created_at                     2354 non-null   datetime64[ns, UTC]
 1   id                             2354 non-null   int64
 2   id_str                         2354 non-null   int64
 3   full_text                      2354 non-null   object
 4   truncated                      2354 non-null   bool
 5   display_text_range             2354 non-null   object
 6   entities                       2354 non-null   object
 7   extended_entities              2073 non-null   object
 8   source                         2354 non-null   object
 9   in_reply_to_status_id          78 non-null     float64
 10  in_reply_to_status_id_str      78 non-null     float64
 11  in_reply_to_user_id            78 non-null     float64
 12  in_reply_to_user_id_str        78 non-null     float64
 13  in_reply_to_screen_name        78 non-null     object
 14  user                           2354 non-null   object
 15  geo                            0 non-null      float64
 16  coordinates                    0 non-null      float64
 17  place                          1 non-null      object
 18  contributors                   0 non-null      float64
 19  is_quote_status                2354 non-null   bool
 20  retweet_count                  2354 non-null   int64
 21  favorite_count                 2354 non-null   int64
 22  favorited                      2354 non-null   bool
 23  retweeted                      2354 non-null   bool
 24  possibly_sensitive             2211 non-null   float64
 25  possibly_sensitive_appealable  2211 non-null   float64
 26  lang                           2354 non-null   object
 27  retweeted_status               179 non-null    object
 28  quoted_status_id               29 non-null     float64
 29  quoted_status_id_str           29 non-null     float64
 30  quoted_status                  28 non-null     object
dtypes: bool(4), datetime64[ns, UTC](1), float64(11), int64(4), object(11)
memory usage: 505.9+ KB
```

In [7]:

```
twitter_api.head()
```

Out[7]:

	created_at	id	id_str	full_text	truncated	display_text_range	er

0	2017-08-01 16:23:56+00:00	892420643555336193	892420643555336192	This is Phineas. He's a mystical boy. Only eve...	False	[0, 85]	{'hashta 'symb 'user_ment
1	2017-08-01 00:17:27+00:00	892177421306343426	892177421306343424	This is Tilly. She's just checking pup on you....	False	[0, 138]	{'hashta 'symb 'user_ment
2	2017-07-31 00:18:03+00:00	891815181378084864	891815181378084864	This is Archie. He is a rare Norwegian Pouncin...	False	[0, 121]	{'hashta 'symb 'user_ment
3	2017-07-30 15:58:51+00:00	891689557279858688	891689557279858688	This is Darla. She commenced a snooze mid meal...	False	[0, 79]	{'hashta 'symb 'user_ment
4	2017-07-29 16:00:24+00:00	891327558926688256	891327558926688256	This is Franklin. He would like you to stop ca...	False	[0, 138]	{'hash ['Bark 'indi

5 rows × 31 columns

In [8]:

```
#only three columns are required from the tweet data
twitter_api_df = pd.DataFrame(twitter_api, columns=['id', 'retweet_count', 'favorite_cou
twitter_api_df
```

Out[8]:

	id	retweet_count	favorite_count
0	892420643555336193	8853	39467
1	892177421306343426	6514	33819
2	891815181378084864	4328	25461
3	891689557279858688	8964	42908
4	891327558926688256	9774	41048
...	...	...	...
2349	666049248165822465	41	111
2350	666044226329800704	147	311
2351	666033412701032449	47	128
2352	666029285002620928	48	132
2353	666020888022790149	532	2535

2354 rows × 3 columns

In [ ]:

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 assessement 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.

(i) Assessing Twitter archive enhanced dataset

In [9]:
```
twitter_archive.head(5)
```

Out[9]:

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

In [10]:
```
twitter_archive.shape # to get the dimension of the dataframe
```

Out[10]:
```
(2356, 17)
```

In [11]:

```
twitter_archive.info() #general information about the dataframe
```

```
<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
```

In [12]:

```
twitter_archive.sample(5)
```

Out[12]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	s
1916	674307341513269249	NaN	NaN	2015-12-08 19:19:32 +0000	<a href="http://vi rel="nofollow">V
1354	703631701117943808	NaN	NaN	2016-02-27 17:24:05 +0000	href="http://twitter.com/download/ip
2181	668994913074286592	NaN	NaN	2015-11-24 03:29:51 +0000	href="http://twitter.com/download/ip
2001	672482722825261057	NaN	NaN	2015-12-03 18:29:09 +0000	href="http://twitter.com/download/ip
2271	667495797102141441	NaN	NaN	2015-11-20 00:12:54 +0000	<a href="http://twitter rel="nofollow"

In [13]:

```
#getting the number of names
#shows that some names have invalid names(a, the, an, none)
twitter_archive.name.unique()
```

Out[13]:

```
array(['Phineas', 'Tilly', 'Archie', 'Darla', 'Franklin', 'None', 'Jax',
       'Zoey', 'Cassie', 'Koda', 'Bruno', 'Ted', 'Stuart', 'Oliver',
       'Jim', 'Zeke', 'Ralphus', 'Canela', 'Gerald', 'Jeffrey', 'such',
       'Maya', 'Mingus', 'Derek', 'Roscoe', 'Waffles', 'Jimbo', 'Maisey',
       'Lilly', 'Earl', 'Lola', 'Kevin', 'Yogi', 'Noah', 'Bella',
       'Grizzwald', 'Rusty', 'Gus', 'Stanley', 'Alfy', 'Koko', 'Rey',
       'Gary', 'a', 'Elliot', 'Louis', 'Jesse', 'Romeo', 'Bailey',
       'Duddles', 'Jack', 'Emmy', 'Steven', 'Beau', 'Snoopy', 'Shadow',
       'Terrance', 'Aja', 'Penny', 'Dante', 'Nelly', 'Ginger', 'Benedict',
       'Venti', 'Goose', 'Nugget', 'Cash', 'Coco', 'Jed', 'Sebastian',
       'Walter', 'Sierra', 'Monkey', 'Harry', 'Kody', 'Lassie', 'Rover',
       'Napolean', 'Dawn', 'Boomer', 'Cody', 'Rumble', 'Clifford',
       'quite', 'Dewey', 'Scout', 'Gizmo', 'Cooper', 'Harold', 'Shikha',
       'Jamesy', 'Lili', 'Sammy', 'Meatball', 'Paisley', 'Albus',
       'Neptune', 'Quinn', 'Belle', 'Zooey', 'Dave', 'Jersey', 'Hobbes',
       'Burt', 'Lorenzo', 'Carl', 'Jordy', 'Milky', 'Trooper', 'Winston',
       'Sophie', 'Wyatt', 'Rosie', 'Thor', 'Oscar', 'Luna', 'Callie',
       'Cermet', 'George', 'Marlee', 'Arya', 'Einstein', 'Alice',
       'Rumpole', 'Benny', 'Aspen', 'Jarod', 'Wiggles', 'General',
       'Sailor', 'Astrid', 'Iggy', 'Snoop', 'Kyle', 'Leo', 'Riley',
       'Gidget', 'Noosh', 'Odin', 'Jerry', 'Charlie', 'Georgie', 'Rontu',
       'Cannon', 'Furzey', 'Daisy', 'Tuck', 'Barney', 'Vixen', 'Jarvis',
       'Mimosa', 'Pickles', 'Bungalo', 'Brady', 'Margo', 'Sadie', 'Hank',
       'Tycho', 'Stephan', 'Indie', 'Winnie', 'Bentley', 'Ken', 'Max',
       'Maddie', 'Pipsy', 'Monty', 'Sojourner', 'Odie', 'Arlo', 'Sunny',
       'Vincent', 'Lucy', 'Clark', 'Mookie', 'Meera', 'Buddy', 'Ava',
       'Rory', 'Eli', 'Ash', 'Tucker', 'Tobi', 'Chester', 'Wilson',
       'Sunshine', 'Lipton', 'Gabby', 'Bronte', 'Poppy', 'Rhino',
       'Willow', 'not', 'Orion', 'Eevee', 'Smiley', 'Logan', 'Moreton',
       'Klein', 'Miguel', 'Emanuel', 'Kuyu', 'Dutch', 'Pete', 'Scooter',
       'Reggie', 'Kyro', 'Samson', 'Loki', 'Mia', 'Malcolm', 'Dexter',
       'Alfie', 'Fiona', 'one', 'Mutt', 'Bear', 'Doobert', 'Beebop',
       'Alexander', 'Sailer', 'Brutus', 'Kona', 'Boots', 'Ralphie',
       'Phil', 'Cupid', 'Pawnd', 'Pilot', 'Ike', 'Mo', 'Toby', 'Sweet',
       'Pablo', 'Nala', 'Balto', 'Crawford', 'Gabe', 'Mattie', 'Jimison',
       'Hercules', 'Duchess', 'Harlso', 'Sampson', 'Sundance', 'Luca',
       'Flash', 'Finn', 'Peaches', 'Howie', 'Jazzy', 'Anna', 'Bo',
       'Seamus', 'Wafer', 'Chelsea', 'Tom', 'Moose', 'Florence', 'Autumn',
       'Dido', 'Eugene', 'Herschel', 'Strudel', 'Tebow', 'Chloe', 'Betty',
       'Timber', 'Binky', 'Dudley', 'Comet', 'Larry', 'Levi', 'Akumi',
       'Titan', 'Olivia', 'Alf', 'Oshie', 'Bruce', 'Chubbs', 'Sky',
       'Atlas', 'Eleanor', 'Layla', 'Rocky', 'Baron', 'Tyr', 'Bauer',
       'Swagger', 'Brandi', 'Mary', 'Moe', 'Halo', 'Augie', 'Craig',
       'Sam', 'Hunter', 'Pavlov', 'Maximus', 'Wallace', 'Ito', 'Milo',
       'Ollie', 'Cali', 'Lennon', 'incredibly', 'Major', 'Duke',
       'Reginald', 'Sansa', 'Shooter', 'Django', 'Diogi', 'Sonny',
       'Philbert', 'Marley', 'Severus', 'Ronnie', 'Anakin', 'Bones',
       'Mauve', 'Chef', 'Doc', 'Sobe', 'Longfellow', 'Mister', 'Iroh',
       'Baloo', 'Stubert', 'Paull', 'Timison', 'Davey', 'Pancake',
       'Tyrone', 'Snicku', 'Ruby', 'Brody', 'Rizzy', 'Mack', 'Butter',
       'Nimbus', 'Laika', 'Dobby', 'Juno', 'Maude', 'Lily', 'Newt',
       'Benji', 'Nida', 'Robin', 'Monster', 'BeBe', 'Remus', 'Mabel',
       'Misty', 'Happy', 'Mosby', 'Maggie', 'Leela', 'Ralphy', 'Brownie',
       'Meyer', 'Stella', 'mad', 'Frank', 'Tonks', 'Lincoln', 'Oakley',
       'Dale', 'Rizzo', 'Arnie', 'Pinot', 'Dallas', 'Hero', 'Frankie',
       'Stormy', 'Mairi', 'Loomis', 'Godi', 'Kenny', 'Deacon', 'Timmy',
       'Harper', 'Chipson', 'Combo', 'Dash', 'Bell', 'Hurley', 'Jay',
       'Mya', 'Strider', 'an', 'Wesley', 'Solomon', 'Huck', 'very', 'O',
       'Blue', 'Finley', 'Sprinkles', 'Heinrich', 'Shakespeare', 'Fizz',
       'Chip', 'Grey', 'Roosevelt', 'Gromit', 'Willem', 'Dakota', 'Dixie',
       'Al', 'Jackson', 'just', 'Carbon', 'DonDon', 'Kirby', 'Lou',
       'Nollie', 'Chevy', 'Tito', 'Louie', 'Rupert', 'Rufus', 'Brudge',
       'Shadoe', 'Colby', 'Angel', 'Brat', 'Tove', 'my', 'Aubie', 'Kota',
       'Eve', 'Glenn', 'Shelby', 'Sephie', 'Bonaparte', 'Albert',
       'Wishes', 'Rose', 'Theo', 'Rocco', 'Fido', 'Emma', 'Spencer',
       'Lilli', 'Boston', 'Brandonald', 'Corey', 'Leonard', 'Chompsky',
       'Beckham', 'Devón', 'Gert', 'Watson', 'Rubio', 'Keith', 'Dex',
       'Carly', 'Ace', 'Tayzie', 'Grizzie', 'Fred', 'Gilbert', 'Zoe',
       'Stewie', 'Calvin', 'Lilah', 'Spanky', 'Jameson', 'Piper',
       'Atticus', 'Blu', 'Dietrich', 'Divine', 'Tripp', 'his', 'Cora',
       'Huxley', 'Keurig', 'Bookstore', 'Linus', 'Abby', 'Shaggy',
       'Shiloh', 'Gustav', 'Arlen', 'Percy', 'Lenox', 'Sugar', 'Harvey',
       'Blanket', 'actually', 'Geno', 'Stark', 'Beya', 'Kilo', 'Kayla',
       'Maxaroni', 'Doug', 'Edmund', 'Aqua', 'Theodore', 'Chase',
       'getting', 'Rorie', 'Simba', 'Charles', 'Bayley', 'Axel',
       'Storkson', 'Remy', 'Chadrick', 'Kellogg', 'Buckley', 'Livvie',
       'Terry', 'Hermione', 'Ralpher', 'Aldrick', 'this', 'unacceptable',
       'Rooney', 'Crystal', 'Ziva', 'Stefan', 'Pupcasso', 'Puff',
       'Flurpson', 'Coleman', 'Enchilada', 'Raymond', 'all', 'Rueben',
       'Cilantro', 'Karll', 'Sprout', 'Blitz', 'Bloop', 'Lillie',
       'Ashleigh', 'Kreggory', 'Sarge', 'Luther', 'Ivar', 'Jangle',
       'Schnitzel', 'Panda', 'Berkeley', 'Ralphé', 'Charleson', 'Clyde',
       'Harnold', 'Sid', 'Pippa', 'Otis', 'Carper', 'Bowie',
       'Alexanderson', 'Suki', 'Barclay', 'Skittle', 'Ebby', 'Flávio',
       'Smokey', 'Link', 'Jennifur', 'Ozzy', 'Bluebert', 'Stephanus',
       'Bubbles', 'old', 'Zeus', 'Bertson', 'Nico', 'Michelangelope',
       'Siba', 'Calbert', 'Curtis', 'Travis', 'Thumas', 'Kanu', 'Lance',
       'Opie', 'Kane', 'Olive', 'Chuckles', 'Staniel', 'Sora', 'Beemo',
       'Gunner', 'infuriating', 'Lacy', 'Tater', 'Olaf', 'Cecil', 'Vince',
       'Karma', 'Billy', 'Walker', 'Rodney', 'Klevin', 'Malikai',
       'Bobble', 'River', 'Jebberson', 'Remington', 'Farfle', 'Jiminus',
       'Clarkus', 'Finnegus', 'Cupcake', 'Kathmandu', 'Ellie', 'Katie',
       'Kara', 'Adele', 'Zara', 'Ambrose', 'Jimothy', 'Bode', 'Terrenth',
       'Reese', 'Chesterson', 'Lucia', 'Bisquick', 'Ralphson', 'Socks',
       'Rambo', 'Rudy', 'Fiji', 'Rilo', 'Bilbo', 'Coopson', 'Yoda',
       'Millie', 'Chet', 'Crouton', 'Daniel', 'Kaia', 'Murphy', 'Dotsy',
       'Eazy', 'Coops', 'Fillup', 'Miley', 'Charl', 'Reagan', 'Yukon',
       'CeCe', 'Cuddles', 'Claude', 'Jessiga', 'Carter', 'Ole', 'Pherb',
       'Blipson', 'Reptar', 'Trevith', 'Berb', 'Bob', 'Colin', 'Brian',
       'Oliviér', 'Grady', 'Kobe', 'Freddery', 'Bodie', 'Dunkin', 'Wally',
       'Tupawc', 'Amber', 'Edgar', 'Teddy', 'Kingsley', 'Brockly',
       'Richie', 'Molly', 'Vinscent', 'Cedrick', 'Hazel', 'Lolo', 'Eriq',
       'Phred', 'the', 'Oddie', 'Maxwell', 'Geoff', 'Covach', 'Durg',
       'Fynn', 'Ricky', 'Herald', 'Lucky', 'Ferg', 'Trip', 'Clarence',
       'Hamrick', 'Brad', 'Pubert', 'Frönq', 'Derby', 'Lizzie', 'Ember',
       'Blakely', 'Opal', 'Marq', 'Kramer', 'Barry', 'Gordon', 'Baxter',
       'Mona', 'Horace', 'Crimson', 'Birf', 'Hammond', 'Lorelei', 'Marty',
       'Brooks', 'Petrick', 'Hubertson', 'Gerbald', 'Oreo', 'Bruiser',
       'Perry', 'Bobby', 'Jeph', 'Obi', 'Tino', 'Kulet', 'Sweets', 'Lupe',
       'Tiger', 'Jiminy', 'Griffin', 'Banjo', 'Brandy', 'Lulu', 'Darrel',
       'Taco', 'Joey', 'Patrick', 'Kreg', 'Todo', 'Tess', 'Ulysses',
       'Toffee', 'Apollo', 'Asher', 'Glacier', 'Chuck', 'Champ', 'Ozzie',
       'Griswold', 'Cheesy', 'Moofasa', 'Hector', 'Goliath', 'Kawhi',
       'by', 'Emmie', 'Penelope', 'Willie', 'Rinna', 'Mike', 'William',
       'Dwight', 'Evy', 'officially', 'Rascal', 'Linda', 'Tug', 'Tango',
       'Grizz', 'Jerome', 'Crumpet', 'Jessifer', 'Izzy', 'Ralph', 'Sandy',
       'Humphrey', 'Tassy', 'Juckson', 'Chuq', 'Tyrus', 'Karl',
       'Godzilla', 'Vinnie', 'Kenneth', 'Herm', 'Bert', 'Striker',
       'Donny', 'Pepper', 'Bernie', 'Buddah', 'Lenny', 'Arnold', 'Zuzu',
       'Mollie', 'Laela', 'Tedders', 'Superpup', 'Rufio', 'Jeb', 'Rodman',
       'Jonah', 'Chesney', 'life', 'Henry', 'Bobbay', 'Mitch', 'Kaiya',
       'Acro', 'Aiden', 'Obie', 'Dot', 'Shnuggles', 'Kendall', 'Jeffri',
       'Steve', 'Mac', 'Fletcher', 'Kenzie', 'Pumpkin', 'Schnozz',
       'Gustaf', 'Cheryl', 'Ed', 'Leonidas', 'Norman', 'Caryl', 'Scott',
       'Taz', 'Darby', 'Jackie', 'light', 'Jazz', 'Franq', 'Pippin',
       'Rolf', 'Snickers', 'Ridley', 'Cal', 'Bradley', 'Bubba', 'Tuco',
       'Patch', 'Mojo', 'Batdog', 'Dylan', 'space', 'Mark', 'JD',
       'Alejandro', 'Scruffers', 'Pip', 'Julius', 'Tanner', 'Sparky',
       'Anthony', 'Holly', 'Jett', 'Amy', 'Sage', 'Andy', 'Mason',
       'Trigger', 'Antony', 'Creg', 'Traviss', 'Gin', 'Jeffrie', 'Danny',
       'Ester', 'Pluto', 'Bloo', 'Edd', 'Willy', 'Herb', 'Damon',
       'Peanut', 'Nigel', 'Butters', 'Sandra', 'Fabio', 'Randall', 'Liam',
       'Tommy', 'Ben', 'Raphael', 'Julio', 'Andru', 'Kloey', 'Shawwn',
       'Skye', 'Kollin', 'Ronduh', 'Billl', 'Saydee', 'Dug', 'Tessa',
       'Sully', 'Kirk', 'Ralf', 'Clarq', 'Jaspers', 'Samsom', 'Harrison',
       'Chaz', 'Jeremy', 'Jaycob', 'Lambeau', 'Ruffles', 'Amélie', 'Bobb',
       'Banditt', 'Kevon', 'Winifred', 'Hanz', 'Churlie', 'Zeek',
       'Timofy', 'Maks', 'Jomathan', 'Kallie', 'Marvin', 'Spark',
       'Gòrdón', 'Jo', 'DayZ', 'Jareld', 'Torque', 'Ron', 'Skittles',
       'Cleopatricia', 'Erik', 'Stu', 'Tedrick', 'Filup', 'Kial',
       'Naphaniel', 'Dook', 'Hall', 'Philippe', 'Biden', 'Fwed',
       'Genevieve', 'Joshwa', 'Bradlay', 'Clybe', 'Keet', 'Carll',
       'Jockson', 'Josep', 'Lugan', 'Christoper'], dtype=object)
```

In [14]:

```
twitter_archive['expanded_urls'].isnull().sum() # missing entries
```

Out[14]:

```
59
```

In [15]:

```
#summary statitics for rating_numerator and rating_denominator
twitter_archive['rating_numerator'].describe()
```

Out[15]:

```
count    2356.000000
mean       13.126486
std        45.876648
min         0.000000
25%        10.000000
50%        11.000000
75%        12.000000
max      1776.000000
Name: rating_numerator, dtype: float64
```

In [16]:

```
twitter_archive[['doggo', 'floofer', 'pupper', 'puppo']]
```

Out[16]:

	doggo	floofer	pupper	puppo
0	None	None	None	None
1	None	None	None	None
2	None	None	None	None
3	None	None	None	None
4	None	None	None	None
...	...	...	...	...
2351	None	None	None	None
2352	None	None	None	None
2353	None	None	None	None
2354	None	None	None	None
2355	None	None	None	None

2356 rows × 4 columns

In [17]:

```
twitter_archive['rating_denominator'].describe()  #summary statistics
```

Out[17]:

```
count    2356.000000
mean       10.455433
std         6.745237
min         0.000000
25%        10.000000
50%        10.000000
75%        10.000000
max       170.000000
Name: rating_denominator, dtype: float64
```

In [18]:

```
#getting the number of ratings below 10
mask = twitter_archive.query('rating_numerator < 10')
mask.count()[0]
```

Out[18]:

```
440
```

In [19]:

```python
pd.set_option('display.max_colwidth', None) # to display full length of texts
```

In [20]:

```python
#querying the min rating = 0 from the rating denominator
twitter_archive.query('rating_denominator == 0').text
#seems like the id from this tweet is one, and is not a rating. Should be dropped during
```

Out[20]:

```
313    @jonnysun @Lin_Manuel ok jomny I know you're excited but 960/00 isn't a valid rat
ing, 13/10 is tho
Name: text, dtype: object
```

In [21]:

```python
# the rating denominator should be strictly 10. Querying ratings that are not 10
lower_ratings = twitter_archive.query('rating_denominator != 10')
lower_ratings.count()[0]
```

Out[21]:

```
23
```

In [22]:

```python
twitter_archive.head(10)
```

Out[22]:

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	sour
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
5	891087950875897856	NaN	NaN	2017-07-29 00:08:17 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
6	890971913173991426	NaN	NaN	2017-07-28 16:27:12 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
7	890729181411237888	NaN	NaN	2017-07-28 00:22:40 +0000	< href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a

8	890609185150312448	NaN	NaN	2017-07-27 16:25:51 +0000	 href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a
9	890240255349198849	NaN	NaN	2017-07-26 15:59:51 +0000	 href="http://twitter.com/download/iphon rel="nofollow">Twitter for iPhone</a

In [23]:

```
twitter_archive.text
```

Out[23]:

```
0                                                              This is Phineas. He's a mys
tical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU
1        This is Tilly. She's just checking pup on you. Hopes you're doing ok. If not, sh
e's available for pats, snugs, boops, the whole bit. 13/10 https://t.co/0Xxu71qeIV
2                         This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in
the tall grass. You never know when one may strike. 12/10 https://t.co/wUnZnhtVJB
3                                                                    This is Darla. She co
mmenced a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qLQ
4        This is Franklin. He would like you to stop calling him "cute." He is a very fie
rce shark and should be respected as such. 12/10 #BarkWeek https://t.co/AtUZn91f7f

...
2351                                                 Here we have a 1949 1st generation vul
pix. Enjoys sweat tea and Fox News. Cannot be phased. 5/10 https://t.co/4B7cOc1EDq
2352                                  This is a purebred Piers Morgan. Loves to Netflix and c
hill. Always looks like he forgot to unplug the iron. 6/10 https://t.co/DWnyCjf2mx
2353                                            Here is a very happy pup. Big fan of well-mainta
ined decks. Just look at that tongue. 9/10 would cuddle af https://t.co/y671yMhoiR
2354                                    This is a western brown Mitsubishi terrier. Upset about l
eaf. Actually 2 dogs here. 7/10 would walk the shit out of https://t.co/r7mOb2m0UI
2355                                           Here we have a Japanese Irish Setter. Lost eye in
Vietnam (?). Big fan of relaxing on stair. 8/10 would pet https://t.co/BLDqew2Ijj
Name: text, Length: 2356, dtype: object
```

(ii) Assessing Image prediction dataset

In [24]:

```
image_pred_df.head(5)
```

Out[24]:

	tweet_id	jpg_url	img_num	p1	p1_co
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_springer_spaniel	0.4650
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbone	0.5068
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_shepherd	0.5964
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_ridgeback	0.4081
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature_pinscher	0.5603

In [25]:

```
image_pred_df.shape
```

Out[25]:

```
(2075, 12)
```

In [26]:

```
image_pred_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 [27]:
```
sum(image_pred_df.duplicated()) #getting duplicates of the df
# shows there are none
```

Out[27]:
```
0
```

In [28]:
```
image_pred_df.sample(5)
```

Out[28]:

	tweet_id	jpg_url	img_num	p1	p1
2001	876484053909872640	https://pbs.twimg.com/media/DCnll_dUQAAkBdG.jpg	1	golden_retriever	0.8
375	672997845381865473	https://pbs.twimg.com/media/CVb39_1XIAAMolv.jpg	1	chow	0.5
638	681281657291280384	https://pbs.twimg.com/media/CXRmDfWWMAADCdc.jpg	1	Saint_Bernard	0.9
1363	761334018830917632	https://pbs.twimg.com/media/CpDNQGkWEAENiYZ.jpg	1	Norwegian_elkhound	0.8
1579	796177847564038144	https://pbs.twimg.com/media/Cwx99rpW8AMk_Ie.jpg	1	golden_retriever	0.6

(iii) Assessing tweet data gotten from Twitter API

In [29]:
```
twitter_api_df.head()
```

Out[29]:

	id	retweet_count	favorite_count
0	892420643555336193	8853	39467
1	892177421306343426	6514	33819
2	891815181378084864	4328	25461
3	891689557279858688	8964	42908
4	891327558926688256	9774	41048

In [30]:
```
twitter_api_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   id              2354 non-null   int64
 1   retweet_count   2354 non-null   int64
 2   favorite_count  2354 non-null   int64
dtypes: int64(3)
memory usage: 55.3 KB
```

In [31]:
```
twitter_api_df.sample(5)
```

Out[31]:

	id	retweet_count	favorite_count
1908	674416750885273600	157	731
1456	695074328191332352	1239	3116
655	791780927877898241	4432	0
2101	670676092097810432	45	267
1854	675522403582218240	316	1122

In [32]:
```
twitter_api_df['retweet_count']
```

Out[32]:
```
0       8853
1       6514
2       4328
3       8964
4       9774
        ...
2349      41
2350     147
2351      47
2352      48
2353     532
Name: retweet_count, Length: 2354, dtype: int64
```

Quality issues

(a) Enhanced Twitter Archive

1.Some of the dogs have invalid names (None, a, an, by, quite and the)

NB:all the invalid dog names start with lower case letters

2.columns(doggo, floofer, pupper, puppo) have 'None' instead of NaN for missing values

3.Name column has 'None' instead of NaN for missing values, also has too many invalid entries.

4.Wrong timestamp data type, it has string instead of date time

5.181 retweets available. we are only interested in tweets only. tetweeted_status id should be removed from the table.

6.440 rating_numerator ratings than are less than 10

7.1, O rating_denominator rating

8.there are 23 rating denominators not equal to, that is greater or less than 10(the rating numerator must always be 10)

9.Missing values for expanded urls(59 missing entries)

(b) Image Predictions

1.incostistent name format, p columns having some names starting with ippercase letters while others start with lowercase.

2.P columns have Underscores instead of spaces between the names

3.The dataframe should contain 2356 entries nut it has 2075 entries

(c) Tweets from Twitter Api

1.2354 observations instead of 2356

2.id column is different from other two datasets

Tidiness issues

1.in the twitter archive dataset, dog stage has four different columns

2.Some columns are not useful and should be dropped(such as image_num from image predictions, and retweet columns from twitter archive

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 [33]:

```
# Make copies of original pieces of data
clean_twitter_archive = twitter_archive.copy()
cleaned_iPred = image_pred_df.copy()
clean_twitter_api = twitter_api_df.copy()
```

In [34]:

```
cleaned_iPred.p1
```

Out[34]:

```
0       Welsh_springer_spaniel
1                      redbone
2              German_shepherd
3          Rhodesian_ridgeback
4           miniature_pinscher
                 ...
2070                    basset
2071               paper_towel
2072                 Chihuahua
2073                 Chihuahua
2074                    orange
Name: p1, Length: 2075, dtype: object
```

Issue #1: invalid dog names

Define:

Convert the invalid dog names to NaN.

Extract the correct wrong names from the text column

Code

In [38]:

```
clean_twitter_archive['name'].replace(regex = ['^[a-z]+', 'None'], value = np.nan, inpla
# replacin the matched results with NaN
```

Test

In [39]:

```
clean_twitter_archive['name'].isnull().sum() # number of missing values for dog name aft
```

Out[39]:

```
854
```

Issue #2: Wrong timestamp data type, it has string instead of date time

Define

Correct inavlid data type by converting timestamp to date time

code

In [40]:

```
clean_twitter_archive.timestamp = pd.to_datetime(clean_twitter_archive.timestamp)
```

Test

In [41]:

```
clean_twitter_archive.timestamp
```

Out[41]:

```
0       2017-08-01 16:23:56+00:00
1       2017-08-01 00:17:27+00:00
2       2017-07-31 00:18:03+00:00
3       2017-07-30 15:58:51+00:00
4       2017-07-29 16:00:24+00:00
                   ...
2351    2015-11-16 00:24:50+00:00
2352    2015-11-16 00:04:52+00:00
2353    2015-11-15 23:21:54+00:00
2354    2015-11-15 23:05:30+00:00
2355    2015-11-15 22:32:08+00:00
Name: timestamp, Length: 2356, dtype: datetime64[ns, UTC]
```

Issue #3: .181 retweets available. we are only interested in tweets only. tetweeted_status id should be removed from the table

Define

Delete entries that have retweets and all related columns related to retweets

code

In [42]:

```
clean_twitter_archive = clean_twitter_archive[clean_twitter_archive.retweeted_status_id.
```

In [43]:

```
clean_twitter_archive = clean_twitter_archive.drop(columns = ['retweeted_status_user_id'
```

In [44]:

```
clean_twitter_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 0 to 2355
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype              
---  ------                 --------------  -----              
 0   tweet_id               2175 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              2175 non-null   datetime64[ns, UTC]
 4   source                 2175 non-null   object             
 5   text                   2175 non-null   object             
 6   retweeted_status_id    0 non-null      float64            
 7   expanded_urls          2117 non-null   object             
 8   rating_numerator       2175 non-null   int64              
 9   rating_denominator     2175 non-null   int64              
 10  name                   1391 non-null   object             
 11  doggo                  2175 non-null   object             
 12  floofer                2175 non-null   object             
 13  pupper                 2175 non-null   object             
 14  puppo                  2175 non-null   object             
dtypes: datetime64[ns, UTC](1), float64(3), int64(3), object(8)
memory usage: 271.9+ KB
```

In [45]:

```
clean_twitter_archive = clean_twitter_archive[clean_twitter_archive.retweeted_status_id.
```

In [46]:

```
clean_twitter_archive = clean_twitter_archive[clean_twitter_archive.in_reply_to_status_i
```

In [47]:

```
clean_twitter_archive.drop(['in_reply_to_status_id', 'in_reply_to_user_id'], axis = 1)
clean_twitter_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 15 columns):
 #   Column                 Non-Null Count  Dtype              
---  ------                 --------------  -----              
 0   tweet_id               2097 non-null   int64              
 1   in_reply_to_status_id  0 non-null      float64            
 2   in_reply_to_user_id    0 non-null      float64            
 3   timestamp              2097 non-null   datetime64[ns, UTC]
 4   source                 2097 non-null   object             
 5   text                   2097 non-null   object             
 6   retweeted_status_id    0 non-null      float64            
 7   expanded_urls          2094 non-null   object             
 8   rating_numerator       2097 non-null   int64              
 9   rating_denominator     2097 non-null   int64              
 10  name                   1390 non-null   object             
 11  doggo                  2097 non-null   object             
 12  floofer                2097 non-null   object             
 13  pupper                 2097 non-null   object             
 14  puppo                  2097 non-null   object             
dtypes: datetime64[ns, UTC](1), float64(3), int64(3), object(8)
memory usage: 262.1+ KB
```

In [48]:

```
clean_twitter_archive.drop(['in_reply_to_status_id'], axis = 1, inplace = True)
```

Test

In [49]:

```
clean_twitter_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 14 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   tweet_id               2097 non-null   int64
 1   in_reply_to_user_id    0 non-null      float64
 2   timestamp              2097 non-null   datetime64[ns, UTC]
 3   source                 2097 non-null   object
 4   text                   2097 non-null   object
 5   retweeted_status_id    0 non-null      float64
 6   expanded_urls          2094 non-null   object
 7   rating_numerator       2097 non-null   int64
 8   rating_denominator     2097 non-null   int64
 9   name                   1390 non-null   object
 10  doggo                  2097 non-null   object
 11  floofer                2097 non-null   object
 12  pupper                 2097 non-null   object
 13  puppo                  2097 non-null   object
dtypes: datetime64[ns, UTC](1), float64(2), int64(3), object(8)
memory usage: 245.7+ KB
```

In [50]:

```
clean_twitter_archive.drop("in_reply_to_user_id", axis = 1, inplace = True)
```

In [51]:

```
cleaned_iPred.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
```

Issue #4: Inconsistent name format for P columns. Some names start with upper case letters while others start with lower case

Define

Convert lower case starting letters to Uppercase using .title() method

strip the underscore character between the names.

code

In [52]:

```
cleaned_iPred['p1'] = cleaned_iPred.p1.str.title()
cleaned_iPred['p2'] = cleaned_iPred.p2.str.title()
```

```python
cleaned_iPred['p3'] = cleaned_iPred.p3.str.title()
```

In [53]:
```python
cleaned_iPred['p1'] = cleaned_iPred['p1'].str.replace('_', ' ')
cleaned_iPred['p2'] = cleaned_iPred['p2'].str.replace('_', ' ')
cleaned_iPred['p3'] = cleaned_iPred['p3'].str.replace('_', ' ')
```

Test

In [54]:
```python
cleaned_iPred['p1']
```

Out[54]:
```
0       Welsh Springer Spaniel
1                      Redbone
2              German Shepherd
3          Rhodesian Ridgeback
4           Miniature Pinscher
                 ...
2070                    Basset
2071               Paper Towel
2072                 Chihuahua
2073                 Chihuahua
2074                    Orange
Name: p1, Length: 2075, dtype: object
```

In [55]:
```python
cleaned_iPred['p2']
```

Out[55]:
```
0                   Collie
1       Miniature Pinscher
2                 Malinois
3                  Redbone
4               Rottweiler
               ...
2070      English Springer
2071    Labrador Retriever
2072              Malamute
2073              Pekinese
2074                 Bagel
Name: p2, Length: 2075, dtype: object
```

In [56]:
```python
cleaned_iPred['p3']
```

Out[56]:
```
0                    Shetland Sheepdog
1                  Rhodesian Ridgeback
2                           Bloodhound
3                   Miniature Pinscher
4                             Doberman
                   ...
2070    German Short-Haired Pointer
2071                        Spatula
2072                         Kelpie
2073                       Papillon
2074                         Banana
Name: p3, Length: 2075, dtype: object
```

Issue #5: The twitter Api table has a different id name colume from the other two datasets

Define

Change the name of 'id' to 'tweet_id'

Code

In [57]:

```
clean_twitter_api = clean_twitter_api.rename(columns = {'id':'tweet_id'})
```

Test

In [58]:

```
clean_twitter_api.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 3 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   tweet_id        2354 non-null   int64
 1   retweet_count   2354 non-null   int64
 2   favorite_count  2354 non-null   int64
dtypes: int64(3)
memory usage: 55.3 KB
```

Tidiness issues

Issue #6: The four dog stage columns are about the same thing. they should be joined to dorm one column.

Define

Create a new column: dog_stage.

Extract dog stage from the text column in the twitter Archive table

Code

In [59]:

```
clean_twitter_archive['dog_stage'] = clean_twitter_archive['text'].str.extract('(doggo|f
```

In [60]:

```
cols =['doggo', 'floofer', 'pupper', 'puppo']# deleting unrequired columns
clean_twitter_archive = clean_twitter_archive.drop(columns = cols)
```

Test

In [61]:

```
clean_twitter_archive.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2355
Data columns (total 10 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   tweet_id             2097 non-null   int64
 1   timestamp            2097 non-null   datetime64[ns, UTC]
 2   source               2097 non-null   object
 3   text                 2097 non-null   object
 4   retweeted_status_id  0 non-null      float64
 5   expanded_urls        2094 non-null   object
 6   rating_numerator     2097 non-null   int64
 7   rating_denominator   2097 non-null   int64
 8   name                 1390 non-null   object
 9   dog_stage            353 non-null    object
dtypes: datetime64[ns, UTC](1), float64(1), int64(3), object(5)
memory usage: 180.2+ KB
```

In [62]:
```
clean_twitter_archive.dog_stage.value_counts()
```

Out[62]:
```
pupper     240
doggo       80
puppo       29
floofer      4
Name: dog_stage, dtype: int64
```

Issue 7: Three different data table when they should be just one.

Define

Merge the three DataFrames to form one, based on the column 'tweet_id'

Code

In [63]:
```
clean_twitter_df = pd.merge(clean_twitter_archive, cleaned_iPred, on = 'tweet_id', how =
```

In [64]:
```
clean_twitter_df = pd.merge(clean_twitter_df, clean_twitter_api, on = 'tweet_id', how =
```

In [65]:
```
clean_twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2096
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   tweet_id             2097 non-null   int64
 1   timestamp            2097 non-null   datetime64[ns, UTC]
 2   source               2097 non-null   object
 3   text                 2097 non-null   object
 4   retweeted_status_id  0 non-null      float64
 5   expanded_urls        2094 non-null   object
 6   rating_numerator     2097 non-null   int64
 7   rating_denominator   2097 non-null   int64
 8   name                 1390 non-null   object
 9   dog_stage            353 non-null    object
 10  jpg_url              1971 non-null   object
 11  img_num              1971 non-null   float64
 12  p1                   1971 non-null   object
 13  p1_conf              1971 non-null   float64
 14  p1_dog               1971 non-null   object
 15  p2                   1971 non-null   object
 16  p2_conf              1971 non-null   float64
 17  p2_dog               1971 non-null   object
 18  p3                   1971 non-null   object
 19  p3_conf              1971 non-null   float64
 20  p3_dog               1971 non-null   object
 21  retweet_count        2097 non-null   int64
 22  favorite_count       2097 non-null   int64
dtypes: datetime64[ns, UTC](1), float64(5), int64(5), object(12)
memory usage: 393.2+ KB
```

```
In [66]: #Dropping unrequired columns from merged Dataframe
         cols = ['img_num', 'retweeted_status_id']

         clean_twitter_df.drop(columns = cols)
```

Out[66]:

	tweet_id	timestamp	source	text	
0	892420643555336193	2017-08-01 16:23:56+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Phineas. He's a mystical boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76dJU	
1	892177421306343426	2017-08-01 00:17:27+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	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	
2	891815181378084864	2017-07-31 00:18:03+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Archie. He is a rare Norwegian Pouncing Corgo. Lives in the tall grass. You never know when one may strike. 12/10 https://t.co/wUnZnhtVJB	
3	891689557279858688	2017-07-30 15:58:51+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	This is Darla. She commenced a snooze mid meal. 13/10 happens to the best of us https://t.co/tD36da7qLQ	
4	891327558926688256	2017-07-29 16:00:24+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	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	https
...	...	...	...	...	
2092	666049248165822465	2015-11-16 00:24:50+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	Here we have a 1949 1st generation vulpix. Enjoys sweat tea and Fox News. Cannot be phased. 5/10 https://t.co/4B7cOc1EDq	
2093	666044226329800704	2015-11-16 00:04:52+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	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	
2094	666033412701032449	2015-11-15 23:21:54+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	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	
2095	666029285002620928	2015-11-15 23:05:30+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	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/r7mOb2m0UI
2096	666020888022790149	2015-11-15 22:32:08+00:00	<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>	Here we have a Japanese Irish Setter. Lost eye in Vietnam (?). Big fan of relaxing on stair. 8/10 would pet https://t.co/BLDqew2Ijj

2097 rows × 21 columns

In [67]:

```
twitter_df = clean_twitter_df.copy()
```

In [68]:

```
cols = ['img_num', 'retweeted_status_id']

twitter_df.drop(cols, axis = 1, inplace = True)
```

In [69]:

```
twitter_df['tweet_id'] = twitter_df['tweet_id'].astype(str)
```

In [70]:

```
twitter_df.dropna(subset = ['jpg_url'], inplace = True)
```

In [71]:

```
twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1971 entries, 0 to 2096
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype              
---  ------              --------------  -----              
 0   tweet_id            1971 non-null   object             
 1   timestamp           1971 non-null   datetime64[ns, UTC]
 2   source              1971 non-null   object             
 3   text                1971 non-null   object             
 4   expanded_urls       1971 non-null   object             
 5   rating_numerator    1971 non-null   int64              
 6   rating_denominator  1971 non-null   int64              
 7   name                1349 non-null   object             
 8   dog_stage           322 non-null    object             
 9   jpg_url             1971 non-null   object             
 10  p1                  1971 non-null   object             
 11  p1_conf             1971 non-null   float64            
 12  p1_dog              1971 non-null   object             
 13  p2                  1971 non-null   object             
 14  p2_conf             1971 non-null   float64            
 15  p2_dog              1971 non-null   object             
 16  p3                  1971 non-null   object             
 17  p3_conf             1971 non-null   float64            
 18  p3_dog              1971 non-null   object             
 19  retweet_count       1971 non-null   int64              
 20  favorite_count      1971 non-null   int64              
dtypes: datetime64[ns, UTC](1), float64(3), int64(4), object(13)
memory usage: 338.8+ KB
```

In [72]:

```
# Testing
clean_twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2096
Data columns (total 23 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   tweet_id             2097 non-null   int64
 1   timestamp            2097 non-null   datetime64[ns, UTC]
 2   source               2097 non-null   object
 3   text                 2097 non-null   object
 4   retweeted_status_id  0 non-null      float64
 5   expanded_urls        2094 non-null   object
 6   rating_numerator     2097 non-null   int64
 7   rating_denominator   2097 non-null   int64
 8   name                 1390 non-null   object
 9   dog_stage            353 non-null    object
 10  jpg_url              1971 non-null   object
 11  img_num              1971 non-null   float64
 12  p1                   1971 non-null   object
 13  p1_conf              1971 non-null   float64
 14  p1_dog               1971 non-null   object
 15  p2                   1971 non-null   object
 16  p2_conf              1971 non-null   float64
 17  p2_dog               1971 non-null   object
 18  p3                   1971 non-null   object
 19  p3_conf              1971 non-null   float64
 20  p3_dog               1971 non-null   object
 21  retweet_count        2097 non-null   int64
 22  favorite_count       2097 non-null   int64
dtypes: datetime64[ns, UTC](1), float64(5), int64(5), object(12)
memory usage: 393.2+ KB
```

In [73]:

```
cols = ['img_num', 'retweeted_status_id']

clean_twitter_df.drop(cols, axis = 1, inplace = True)
```

In [74]:

```
clean_twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2096
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   tweet_id            2097 non-null   int64
 1   timestamp           2097 non-null   datetime64[ns, UTC]
 2   source              2097 non-null   object
 3   text                2097 non-null   object
 4   expanded_urls       2094 non-null   object
 5   rating_numerator    2097 non-null   int64
 6   rating_denominator  2097 non-null   int64
 7   name                1390 non-null   object
 8   dog_stage           353 non-null    object
 9   jpg_url             1971 non-null   object
 10  p1                  1971 non-null   object
 11  p1_conf             1971 non-null   float64
 12  p1_dog              1971 non-null   object
 13  p2                  1971 non-null   object
 14  p2_conf             1971 non-null   float64
 15  p2_dog              1971 non-null   object
 16  p3                  1971 non-null   object
 17  p3_conf             1971 non-null   float64
 18  p3_dog              1971 non-null   object
 19  retweet_count       2097 non-null   int64
 20  favorite_count      2097 non-null   int64
dtypes: datetime64[ns, UTC](1), float64(3), int64(5), object(12)
memory usage: 360.4+ KB
```

In [75]:

```
clean_twitter_df['tweet_id'] = clean_twitter_df['tweet_id'].astype(str)
clean_twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2097 entries, 0 to 2096
Data columns (total 21 columns):
 #   Column              Non-Null Count  Dtype              
---  ------              --------------  -----              
 0   tweet_id            2097 non-null   object             
 1   timestamp           2097 non-null   datetime64[ns, UTC]
 2   source              2097 non-null   object             
 3   text                2097 non-null   object             
 4   expanded_urls       2094 non-null   object             
 5   rating_numerator    2097 non-null   int64              
 6   rating_denominator  2097 non-null   int64              
 7   name                1390 non-null   object             
 8   dog_stage           353 non-null    object             
 9   jpg_url             1971 non-null   object             
 10  p1                  1971 non-null   object             
 11  p1_conf             1971 non-null   float64            
 12  p1_dog              1971 non-null   object             
 13  p2                  1971 non-null   object             
 14  p2_conf             1971 non-null   float64            
 15  p2_dog              1971 non-null   object             
 16  p3                  1971 non-null   object             
 17  p3_conf             1971 non-null   float64            
 18  p3_dog              1971 non-null   object             
 19  retweet_count       2097 non-null   int64              
 20  favorite_count      2097 non-null   int64              
dtypes: datetime64[ns, UTC](1), float64(3), int64(4), object(13)
memory usage: 360.4+ KB
```

Storing Data

In [76]:

```
clean_twitter_df.to_csv('twitter_archive_master.csv')
```

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.**

In [89]:

```
dog_stages = clean_twitter_df.dog_stage #creates a dataframe of just the dog stages
```

In [94]:

```
dog_stages.value_counts()  #provides counts of unique value in each category
```

Out[94]:

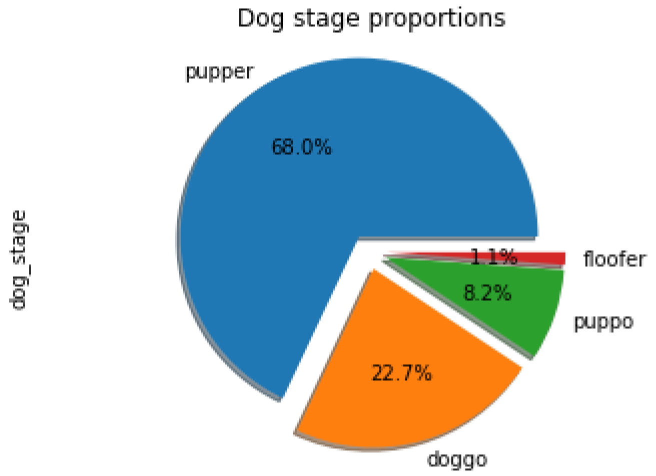
```
pupper     240
doggo       80
puppo       29
floofer      4
Name: dog_stage, dtype: int64
```

In [93]:

```
# creates a pie chart to display the proportions of dog stages
dog_stages = clean_twitter_df.dog_stage
label = ['pupper', 'doggo', 'puppo', 'floofer']
dog_stages.value_counts().plot(kind = 'pie', labels = label,shadow = True,explode = (0.1
plt.title('Dog stage proportions')
plt.axis('equal')
plt.show
```

Out[93]:

```
<function matplotlib.pyplot.show(close=None, block=None)>
```



Insights:

1.Pupper takes the highest proportion, with 68%

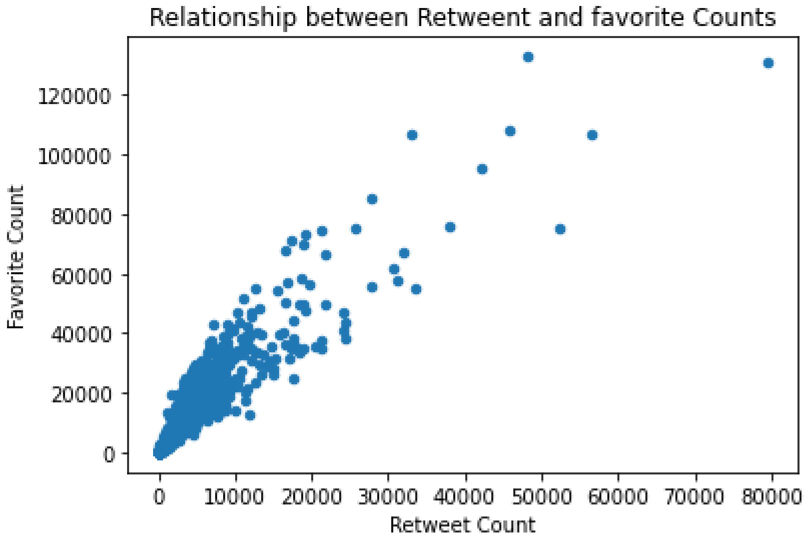
2.doggo comes second with 22.7%, followed by puppo

3.floofer has the lowest proportion

(ii) Relationship between Retweet count and Favorite count

```
In [92]: clean_twitter_df.plot.scatter(x = 'retweet_count', y = 'favorite_count') # creates a sca
         plt.title('Relationship between Retweent and favorite Counts')
         plt.xlabel('Retweet Count')
         plt.ylabel('Favorite Count')
         plt.show
```

```
Out[92]: <function matplotlib.pyplot.show(close=None, block=None)>
```



Insights

- The plot displays a positive linear relationship between the two variables

```
In [ ]:
```