In [1]:

```
#importing libraries that is required for the project
import requests
import json
import pandas as pd
import re
import numpy as np
import seaborn as sb
import matplotlib.pyplot as plt
%matplotlib inline
```

Gathering:

For this project there are three data sets to be gathered:

1-Twitter-archive-enchanced.csv

2-image-predictions.tsv

3-twitter_api.py

4-tweet-json.text and tweet-json.csv

1-Twitter_archive_enchanced.csv

This file is downloaded manually by clicking on the link provided by the udacity.

In [2]:

```
#gathering twitter-archive-enhanced.csv file to twitter_archive
twitter_archive=pd.read_csv('twitter-archive-enhanced.csv')
```

2-image_predictions.tsv

This file is downloaded programmatically using the requests libarary. The url was provided in the udacity portal.

In [3]:

```
ring image predictions data from the udacity portal using request library and storing the da
request=requests.get(" https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_ima
pen('image-predictions.tsv', mode ='wb') as file:
le.write(image_request.content)
```

In [4]:

```
#gathered data is then stored into image_predict
image_predict=pd.read_csv('image-predictions.tsv',sep='\t')
```

3-twitter_api.py

this method is used to scrape the data from the twitter website using the twitter api. here is the code for it: ** import tweepy

from tweepy import OAuthHandler

import json

from timeit import default_timer as timer

consumer_key = 'HIDDEN'

consumer_secret = 'HIDDEN'

access_token = 'HIDDEN'

access_secret = 'HIDDEN'

auth = OAuthHandler(consumer_key, consumer_secret)

auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth, wait_on_rate_limit=True)

tweet_ids = df_1.tweet_id.values

len(tweet_ids)

count = 0

fails_dict = {}

start = timer()

with open('tweet_json.txt', 'w') as outfile: for tweet_id in tweet_ids:

```
        count += 1

        print(str(count) + ": " + str(tweet_id))

        try:

            tweet = api.get_status(tweet_id, tweet_mode='extended')

            print("Success")

            json.dump(tweet._json, outfile)

            outfile.write('\n')

        except tweepy.TweepError as e:

            print("Fail")

            fails_dict[tweet_id] = e

            pass
```

end = timer()

print(end - start)

print(fails_dict)

3-tweet_json.text and tweet_json.csv

For this part,Instead of using thw twitter Api i am going to use the tweet_json file from the udacity.using requests library we are going to download data from the udacity portal

In [5]:

```
#gathering tweet-json.txt data from the udacity portal using request library and storing th
request=requests.get('https://s3.amazonaws.com/video.udacity-data.com/topher/2018/November/
```

In [6]:

```
with open('tweet-json.txt','wb') as file:
    file.write(request.content)
```

In [7]:

```
#extracting the data from the tweet-json.txt file and appending all the extracted data into
tweets_list=[]
with open('tweet-json.txt') as f:
    for data in f:
        status=json.loads(data)
        created_at=status['created_at']
        id_str=status['id_str']
        full_text=status['full_text']
        retweet_count=status['retweet_count']
        favorite_count=status['favorite_count']
        source=status['source']
        tweets_list.append({'id_str':id_str,
                            'created_at':created_at,
                            'source':source,
                            'full_text':full_text,
                            'retweet_count':retweet_count,
                            'favorite_count':favorite_count,
                            })
        tweets=pd.DataFrame(tweets_list,columns=['id_str','created_at','source','full_text'
        tweets.to_csv('tweet-json.csv',index=False)
```

In [8]:

```
#tweet_json.csv file to tweet_json
tweet_json=pd.read_csv('tweet-json.csv')
```

Accessing:

Here we are going to access all the data sets that has been gathered above.

In [9]:

```
twitter_archive.head()
```

Out[9]:

	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://twitter.c
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.c
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://twitter.c
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	href="http://twitter.c
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	href="http://twitter.c

In [10]:

```
image_predict.head()
```

Out[10]:

	tweet_id	jpg_url	img_num	
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_spring
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German_
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian_
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature

In [11]:

```
tweet_json.head()
```

Out[11]:

	id_str	created_at	source	full_text	retweet_c
0	892420643555336193	Tue Aug 01 16:23:56 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	8
1	892177421306343426	Tue Aug 01 00:17:27 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	6
2	891815181378084864	Mon Jul 31 00:18:03 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	4
3	891689557279858688	Sun Jul 30 15:58:51 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	8
4	891327558926688256	Sat Jul 29 16:00:24 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	9

Now we gonna create a copy file for all the files:

In [36]:

```
#getting a copy of all the datas
image_predict_clean=image_predict.copy()
tweet_json_clean=tweet_json.copy()
twitter_archive_clean=twitter_archive.copy()
```

In [37]:

```
image_predict_clean.head()
```

Out[37]:

	tweet_id	jpg_url	img_num	
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh_spring
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	German
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhodesian
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	miniature

In [38]:

```
tweet_json_clean.head()
```

Out[38]:

	id_str	created_at	source	full_text	retweet_c
0	892420643555336193	Tue Aug 01 16:23:56 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	8
1	892177421306343426	Tue Aug 01 00:17:27 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	6
2	891815181378084864	Mon Jul 31 00:18:03 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	4
3	891689557279858688	Sun Jul 30 15:58:51 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	8
4	891327558926688256	Sat Jul 29 16:00:24 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	9

In [39]:

```
twitter_archive_clean.head()
```

Out[39]:

	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://twitter.c
1	892177421306343426	NaN	NaN	2017-08-01 00:17:27 +0000	href="http://twitter.c
2	891815181378084864	NaN	NaN	2017-07-31 00:18:03 +0000	href="http://twitter.c
3	891689557279858688	NaN	NaN	2017-07-30 15:58:51 +0000	href="http://twitter.c
4	891327558926688256	NaN	NaN	2017-07-29 16:00:24 +0000	href="http://twitter.c

In [40]:

```
tweet_json_clean.shape
```

Out[40]:

```
(2354, 6)
```

In [41]:

```
image_predict_clean.shape
```

Out[41]:

```
(2075, 12)
```

In [42]:

```
twitter_archive_clean.shape
```

Out[42]:

```
(2356, 17)
```

In [43]:

```
image_predict_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
jpg_url     2075 non-null object
img_num     2075 non-null int64
p1          2075 non-null object
p1_conf     2075 non-null float64
p1_dog      2075 non-null bool
p2          2075 non-null object
p2_conf     2075 non-null float64
p2_dog      2075 non-null bool
p3          2075 non-null object
p3_conf     2075 non-null float64
p3_dog      2075 non-null bool
dtypes: bool(3), float64(3), int64(2), object(4)
memory usage: 152.1+ KB
```

In [44]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 6 columns):
id_str            2354 non-null int64
created_at        2354 non-null object
source            2354 non-null object
full_text         2354 non-null object
retweet_count     2354 non-null int64
favorite_count    2354 non-null int64
dtypes: int64(3), object(3)
memory usage: 110.4+ KB
```

In [45]:

```
twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 17 columns):
tweet_id                      2356 non-null int64
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2356 non-null object
source                        2356 non-null object
text                          2356 non-null object
retweeted_status_id           181 non-null float64
retweeted_status_user_id      181 non-null float64
retweeted_status_timestamp    181 non-null object
expanded_urls                 2297 non-null object
rating_numerator              2356 non-null int64
rating_denominator            2356 non-null int64
name                          2356 non-null object
doggo                         2356 non-null object
floofer                       2356 non-null object
pupper                        2356 non-null object
puppo                         2356 non-null object
dtypes: float64(4), int64(3), object(10)
memory usage: 313.0+ KB
```

In [46]:

```
image_predict_clean.jpg_url.duplicated().value_counts()
```

Out[46]:

```
False    2009
True       66
Name: jpg_url, dtype: int64
```

In [47]:

```
tweet_json_clean.isnull().sum()
```

Out[47]:

```
id_str            0
created_at        0
source            0
full_text         0
retweet_count     0
favorite_count    0
dtype: int64
```

In [48]:

```
image_predict_clean.isnull().sum()
```

Out[48]:

```
tweet_id    0
jpg_url     0
img_num     0
p1          0
p1_conf     0
p1_dog      0
p2          0
p2_conf     0
p2_dog      0
p3          0
p3_conf     0
p3_dog      0
dtype: int64
```

In [52]:

```
#Extracting ratings(numerator) from the full_text column,filling the empty rows to '0' valu
tweet_json_clean['rating_numerator']=tweet_json_clean.full_text.str.extract('(\d+)/(10)',ex
```

In [53]:

```
#Extracting ratings(denominator) from the full_text column,filling the empty rows to '0' va
tweet_json_clean['rating_denominator']=tweet_json_clean.full_text.str.extract('(10)',expand
```

In [54]:

```
#Extracting dog stage(doggo) from full_text column from each users
tweet_json_clean['doggo']=tweet_json_clean.full_text.str.extract('(doggo)',expand=True)
```

In [55]:

```
#Extracting dog stage(floofer) from full_text column from each users
tweet_json_clean['floofer']=tweet_json_clean.full_text.str.extract('(floofer)',expand=True)
```

In [56]:

```
#Extracting dog stage(pupper) from full_text column from each users
tweet_json_clean['pupper']=tweet_json_clean.full_text.str.extract('(pupper)',expand=True)
```

In [57]:

```
#Extracting dog stage(puppo) from full_text column from each users
tweet_json_clean['puppo']=tweet_json_clean.full_text.str.extract('(puppo)',expand=True)
```

In [59]:

```
#Extracting link/webpage from full_text column from each users
tweet_json_clean['user_link']=tweet_json_clean.full_text.str.extract('(https://t.co/+(\w+\S
```

In [60]:

```
#checking the extracted data from tweet_json_clean
tweet_json_clean.head(10)
```

Out[60]:

	id_str	created_at	source	full_text	retweet_c
0	892420643555336193	Tue Aug 01 16:23:56 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	8
1	892177421306343426	Tue Aug 01 00:17:27 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	6
2	891815181378084864	Mon Jul 31 00:18:03 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	4
3	891689557279858688	Sun Jul 30 15:58:51 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	8
4	891327558926688256	Sat Jul 29 16:00:24 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	9
5	891087950875897856	Sat Jul 29 00:08:17 +0000 2017	<a href="http://twitter.com/download/iphone" r...	Here we have a majestic great white breaching ...	3
6	890971913173991426	Fri Jul 28 16:27:12 +0000 2017	<a href="http://twitter.com/download/iphone" r...	Meet Jax. He enjoys ice cream so much he gets ...	2
7	890729181411237888	Fri Jul 28 00:22:40 +0000 2017	<a href="http://twitter.com/download/iphone" r...	When you watch your owner call another dog a g...	16
8	890609185150312448	Thu Jul 27 16:25:51 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Zoey. She doesn't want to be one of th...	4
9	890240255349198849	Wed Jul 26 15:59:51 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Cassie. She is a college pup. Studying...	7

In [61]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 13 columns):
id_str                2354 non-null int64
created_at            2354 non-null object
source                2354 non-null object
full_text             2354 non-null object
retweet_count         2354 non-null int64
favorite_count        2354 non-null int64
rating_denominator    2354 non-null object
rating_numerator      2354 non-null object
doggo                 98 non-null object
floofer               4 non-null object
pupper                271 non-null object
puppo                 37 non-null object
user_link             2225 non-null object
dtypes: int64(3), object(10)
memory usage: 239.2+ KB
```

Cleaning:

Quality:

1-we have found that there are 66 duplicate values in th jpg_url column drop all those rows.

2-there are some missing decimal number that are not correctly extracted from the full_text column.

3-rating_numerator and rating_denominator are currently in str type to be converted to float.

4-convert all the Nan value to string None for the columns(doggo,floofer,puppo,pupper,user_links)in order to drop all the Nan values

5-created_at column is currently an object need to be converted to timestamp and change the column name to timestamp

6-store id_str values to tweet_id and drop id_str column.

7-drop all retweets in the full_text column.

8-after mergeing the data sets the tweet_id need to be converted to str.

9-Drop all Nan values

10-after merging img_num column is in the float to be converted to int

11-extracting the dog stages from the full_text column and droping the doggo,floofer,pupper and puppo columns.

12-replacment of None to np.Nan for column dog and conf and changing the type of conf column to float64

13-after creating the master data the tweet_id is tyep int to be converted to str.

14-drop all retweets in the full_text column

Tidiness:

1-merge the datasets tweet_json_clean and image_predict_clean.

2-create the merged dataset into one master dataset.

Define:

1-we have found that there are 66 duplicate values in th jpg_url column drop all those rows.

Code:

In [62]:

```
#checking the number of duplicate values in the jpg_url column from the image_predict_clean
image_predict_clean[image_predict_clean.duplicated('jpg_url',False)].count()
```

Out[62]:

```
tweet_id    132
jpg_url     132
img_num     132
p1          132
p1_conf     132
p1_dog      132
p2          132
p2_conf     132
p2_dog      132
p3          132
p3_conf     132
p3_dog      132
dtype: int64
```

In [63]:

```
image_predict_clean[image_predict_clean.duplicated('jpg_url',False)]
```

Out[63]:

	tweet_id	jpg_url	img_num	
85	667509364010450944	https://pbs.twimg.com/media/CUN4Or5UAAAa5K4.jpg	1	
224	670319130621435904	https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg	1	
241	670444955656130560	https://pbs.twimg.com/media/CU3mITUWIAAfyQS.jpg	1	Er
327	671896809300709376	https://pbs.twimg.com/media/CVMOlMiWwAA4YxI.jpg	1	
382	673320132811366400	https://pbs.twimg.com/media/CVgdFjNWEAAxmbq.jpg	3	
432	674291837063053312	https://pbs.twimg.com/media/CVuQ2LeUsAAIe3s.jpg	1	
480	675354435921575936	https://pbs.twimg.com/ext_tw_video_thumb/67535...	1	
487	675501075957489664	https://pbs.twimg.com/media/CV_cnjHWUAADc-c.jpg	1	
587	679062614270468097	https://pbs.twimg.com/media/CWyD2HGUYAQ1Xa7.jpg	2	
591	679158373988876288	https://pbs.twimg.com/media/CWza7kpWcAAdYLc.jpg	1	
602	679828447187857408	https://pbs.twimg.com/media/CW88XN4WsAAlo8r.jpg	3	
713	685325112850124800	https://pbs.twimg.com/media/CYLDikFWEAAIy1y.jpg	1	g
800	691416866452082688	https://pbs.twimg.com/media/CZhn-QAWwAASQan.jpg	1	L
915	701214700881756160	https://pbs.twimg.com/media/Cbs3DOAXIAAp3Bd.jpg	1	
930	703041949650034688	https://pbs.twimg.com/media/CcG07BYW0AErrC9.jpg	1	
985	707610948723478529	https://pbs.twimg.com/media/CdHwZd0VIAA4792.jpg	1	g
1033	711694788429553666	https://pbs.twimg.com/tweet_video_thumb/CeBym7...	1	
1045	712809025985978368	https://pbs.twimg.com/media/CeRoBaxWEAABi0X.jpg	1	Lab
1118	725842289046749185	https://pbs.twimg.com/media/ChK1tdBWwAQ1flD.jpg	1	
1150	732005617171337216	https://pbs.twimg.com/media/CiibOMzUYAA9Mxz.jpg	1	
1155	733109485275860992	https://pbs.twimg.com/media/CiyHLocU4AI2pJu.jpg	1	g
1188	739544079319588864	https://pbs.twimg.com/media/CkNjahBXAAQ2kWo.jpg	1	Lab
1201	741067306818797568	https://pbs.twimg.com/media/CkjMx99UoAM2B1a.jpg	1	g
1209	742423170473463808	https://pbs.twimg.com/media/Ck2d7tJWUAEPTL3.jpg	1	
1283	750429297815552001	https://pbs.twimg.com/media/CmoPdmHW8AAi8BI.jpg	1	g
1297	752309394570878976	https://pbs.twimg.com/ext_tw_video_thumb/67535...	1	
1315	754874841593970688	https://pbs.twimg.com/media/CWza7kpWcAAdYLc.jpg	1	
1333	757729163776290825	https://pbs.twimg.com/media/CWyD2HGUYAQ1Xa7.jpg	2	
1345	759159934323924993	https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg	1	
1349	759566828574212096	https://pbs.twimg.com/media/CkNjahBXAAQ2kWo.jpg	1	Lab
...	...	...	...	
1699	816829038950027264	https://pbs.twimg.com/media/CvoBPWRWgAA4het.jpg	1	
1703	817181837579653120	https://pbs.twimg.com/ext_tw_video_thumb/81596...	1	

	tweet_id	jpg_url	img_num	
1705	817423860136083457	https://pbs.twimg.com/ext_tw_video_thumb/81742...	1	
1712	818588835076603904	https://pbs.twimg.com/media/Crwxb5yWgAAX5P_.jpg	1	Norwe
1715	819004803107983360	https://pbs.twimg.com/media/C12whDoVEAALRxa.jpg	1	st
1716	819006400881917954	https://pbs.twimg.com/media/C12x-JTVIAAzdfl.jpg	4	
1717	819015331746349057	https://pbs.twimg.com/media/C12x-JTVIAAzdfl.jpg	4	
1718	819015337530290176	https://pbs.twimg.com/media/C12whDoVEAALRxa.jpg	1	st
1727	820446719150292993	https://pbs.twimg.com/media/CxqsX-8XUAAEvjD.jpg	3	g
1736	821813639212650496	https://pbs.twimg.com/media/CtVAvX-WIAAcGTf.jpg	1	
1738	822244816520155136	https://pbs.twimg.com/media/C2kzTGxWEAEOpPL.jpg	1	
1740	822489057087389700	https://pbs.twimg.com/media/C2oRbOuWEAAbVSl.jpg	1	
1742	822647212903690241	https://pbs.twimg.com/media/C2oRbOuWEAAbVSl.jpg	1	
1746	823269594223824897	https://pbs.twimg.com/media/C2kzTGxWEAEOpPL.jpg	1	
1755	824796380199809024	https://pbs.twimg.com/media/CwiuEJmW8AAZnit.jpg	2	
1767	826958653328592898	https://pbs.twimg.com/media/C3nygbBWQAAjwcW.jpg	1	g
1785	829374341691346946	https://pbs.twimg.com/media/C4KHj-nWQAA3poV.jpg	1	Staffords
1789	829878982036299777	https://pbs.twimg.com/media/C3nygbBWQAAjwcW.jpg	1	g
1791	830583320585068544	https://pbs.twimg.com/media/C4bTH6nWMAAX_bJ.jpg	1	Lab
1803	832040443403784192	https://pbs.twimg.com/media/Cq9guJ5WgAADfpF.jpg	1	mini
1804	832215726631055365	https://pbs.twimg.com/media/CwJR1okWIAA6XMp.jpg	1	
1858	841833993020538882	https://pbs.twimg.com/ext_tw_video_thumb/81742...	1	
1864	842892208864923648	https://pbs.twimg.com/ext_tw_video_thumb/80710...	1	
1903	851953902622658560	https://pbs.twimg.com/media/C4KHj-nWQAA3poV.jpg	1	Staffords
1944	861769973181624320	https://pbs.twimg.com/media/CzG425nWgAAnP7P.jpg	2	
1970	868880397819494401	https://pbs.twimg.com/media/DA7iHL5U0AA1OQo.jpg	1	
1992	873697596434513921	https://pbs.twimg.com/media/DA7iHL5U0AA1OQo.jpg	1	
2041	885311592912609280	https://pbs.twimg.com/media/C4bTH6nWMAAX_bJ.jpg	1	Lab
2051	887473957103951883	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg	2	
2055	888202515573088257	https://pbs.twimg.com/media/DFDw2tyUQAAAFke.jpg	2	

132 rows × 12 columns

In [64]:

```
#dropping all the duplicate values
image_predict_clean.drop_duplicates('jpg_url',keep='first',inplace=True)
```

Test:

In [65]:

```
#checking whether the duplicate values are dropped or not
image_predict_clean[image_predict_clean.duplicated('jpg_url',False)].count()
```

Out[65]:

```
tweet_id    0
jpg_url     0
img_num     0
p1          0
p1_conf     0
p1_dog      0
p2          0
p2_conf     0
p2_dog      0
p3          0
p3_conf     0
p3_dog      0
dtype: int64
```

In [66]:

```
image_predict_clean[image_predict_clean.duplicated('jpg_url',False)]
```

Out[66]:

tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dc

create two columns from the image_predict_clean where pn_dog is true then append the true dog to the columns:

In [67]:

```
#creating two columns dog and conf these two columns extract the data from the columns p1_d
#which has true value in the dog and its confidence level in conf
dog=[]
conf=[]


def p(image_predict_clean):
    if image_predict_clean.p1_dog==True:
        dog.append(image_predict_clean.p1)
        conf.append(image_predict_clean.p1_conf)
    elif image_predict_clean.p2_dog==True:
        dog.append(image_predict_clean.p2)
        conf.append(image_predict_clean.p2_conf)
    elif image_predict_clean.p3_dog==True:
        dog.append(image_predict_clean.p3)
        conf.append(image_predict_clean.p3_conf)
    else:
        dog.append('None')
        conf.append('None')
image_predict_clean.apply(p,axis=1)

image_predict_clean['dog']=dog
image_predict_clean['conf']=conf
```

In [68]:

```
image_predict_clean.dog.value_counts()
```

Out[68]:

```
None                              318
golden_retriever                  158
Labrador_retriever                108
Pembroke                           95
Chihuahua                          91
pug                                63
toy_poodle                         51
chow                               48
Samoyed                            42
Pomeranian                         42
malamute                           33
French_bulldog                     31
Chesapeake_Bay_retriever           31
cocker_spaniel                     30
miniature_pinscher                 25
Eskimo_dog                         22
Staffordshire_bullterrier          21
German_shepherd                    21
Cardigan                           21
Shih-Tzu                           20
beagle                             20
Siberian_husky                     20
Shetland_sheepdog                  19
Maltese_dog                        19
Rottweiler                         19
kuvasz                             19
Lakeland_terrier                   18
Italian_greyhound                  17
basset                             17
West_Highland_white_terrier        16
                                 ...
Tibetan_terrier                     4
giant_schnauzer                     4
keeshond                            4
bluetick                            4
Gordon_setter                       4
Welsh_springer_spaniel              4
komondor                            3
curly-coated_retriever              3
Greater_Swiss_Mountain_dog          3
Afghan_hound                        3
Leonberg                            3
toy_terrier                         3
briard                              3
Brabancon_griffon                   3
Irish_water_spaniel                 3
cairn                               3
Australian_terrier                  2
wire-haired_fox_terrier             2
groenendael                         2
Appenzeller                         2
black-and-tan_coonhound             2
Sussex_spaniel                      2
silky_terrier                       1
EntleBucher                         1
```

```
Japanese_spaniel                1
standard_schnauzer              1
Scotch_terrier                  1
Irish_wolfhound                 1
Bouvier_des_Flandres            1
clumber                         1
Name: dog, Length: 114, dtype: int64
```

In [69]:

```
image_predict_clean.conf
```

Out[69]:

```
0         0.465074
1         0.506826
2         0.596461
3         0.408143
4         0.560311
5         0.651137
6             None
7         0.692517
8       0.00795896
9         0.201493
10         0.77593
11        0.503672
12        0.260857
13        0.489814
14        0.195217
15         0.58233
16        0.298617
17            None
18            None
19        0.176053
20        0.857531
21            None
22        0.278407
23        0.858744
24        0.336874
25            None
26        0.326467
27        0.978108
28        0.529139
29            None
           ...
2044      0.943575
2045      0.999201
2046          None
2047      0.309706
2048      0.793469
2049      0.733942
2050      0.330741
2051      0.809197
2052          None
2053      0.821664
2054      0.995026
2056      0.700377
2057       0.46976
2058      0.714719
2059      0.626152
2060      0.953442
2061       0.99165
2062      0.966327
2063      0.377417
2064      0.957979
2065      0.511319
2066      0.487574
2067      0.566142
2068      0.341703
```

```
2069      0.425595
2070      0.555712
2071      0.168086
2072      0.716012
2073      0.323581
2074          None
Name: conf, Length: 2009, dtype: object
```

In [70]:

```
image_predict_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2009 entries, 0 to 2074
Data columns (total 14 columns):
tweet_id    2009 non-null int64
jpg_url     2009 non-null object
img_num     2009 non-null int64
p1          2009 non-null object
p1_conf     2009 non-null float64
p1_dog      2009 non-null bool
p2          2009 non-null object
p2_conf     2009 non-null float64
p2_dog      2009 non-null bool
p3          2009 non-null object
p3_conf     2009 non-null float64
p3_dog      2009 non-null bool
dog         2009 non-null object
conf        2009 non-null object
dtypes: bool(3), float64(3), int64(2), object(6)
memory usage: 194.2+ KB
```

In [71]:

```
image_predict_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2009 entries, 0 to 2074
Data columns (total 14 columns):
tweet_id    2009 non-null int64
jpg_url     2009 non-null object
img_num     2009 non-null int64
p1          2009 non-null object
p1_conf     2009 non-null float64
p1_dog      2009 non-null bool
p2          2009 non-null object
p2_conf     2009 non-null float64
p2_dog      2009 non-null bool
p3          2009 non-null object
p3_conf     2009 non-null float64
p3_dog      2009 non-null bool
dog         2009 non-null object
conf        2009 non-null object
dtypes: bool(3), float64(3), int64(2), object(6)
memory usage: 194.2+ KB
```

Quality:

Define:

2-there are some missing decimal number that are not correctly extracted from the full_text column.

Code:

In [72]:

```
#extracts the decimals value from the numerator
ratings_numerator_index=[]
ratings_numerator_value=[]

for index,numerator in tweet_json_clean['full_text'].iteritems():
    if bool(re.search('\d+\.\d+\/\d+', numerator)):
        ratings_numerator_index.append(index)
        ratings_numerator_value.append(re.search('\d+\.\d+', numerator).group())
ratings_numerator_index
```

Out[72]:

```
[44, 339, 694, 762, 1687, 1710]
```

In [73]:

```
ratings_numerator_value
```

Out[73]:

```
['13.5', '9.75', '9.75', '11.27', '9.5', '11.26']
```

In [75]:

```
#storing the decimal values
tweet_json_clean.loc[ratings_numerator_index[0],'rating_numerator'] = ratings_numerator_val
tweet_json_clean.loc[ratings_numerator_index[1],'rating_numerator'] = ratings_numerator_val
tweet_json_clean.loc[ratings_numerator_index[2],'rating_numerator'] = ratings_numerator_val
tweet_json_clean.loc[ratings_numerator_index[3],'rating_numerator'] = ratings_numerator_val
```

Test:

In [76]:

```
tweet_json_clean.loc[44]
```

Out[76]:

```
id_str                                              883482846933004288
created_at                                 Sat Jul 08 00:28:19 +0000 2017
source                 <a href="http://twitter.com/download/iphone" r...
full_text              This is Bella. She hopes her smile made you sm...
retweet_count                                                      10407
favorite_count                                                     46860
rating_denominator                                                    10
rating_numerator                                                    13.5
doggo                                                                NaN
floofer                                                              NaN
pupper                                                               NaN
puppo                                                                NaN
user_link                                      https://t.co/qjrljjt948 (htt
ps://t.co/qjrljjt948)
Name: 44, dtype: object
```

In [77]:

```
tweet_json_clean.loc[339]
```

Out[77]:

```
id_str                                              832215909146226688
created_at                                 Thu Feb 16 13:11:49 +0000 2017
source                 <a href="http://twitter.com/download/iphone" r...
full_text              RT @dog_rates: This is Logan, the Chow who liv...
retweet_count                                                       7069
favorite_count                                                         0
rating_denominator                                                    10
rating_numerator                                                    9.75
doggo                                                                NaN
floofer                                                              NaN
pupper                                                               NaN
puppo                                                                NaN
user_link                                    https://t.co/yBO5wu… (htt
ps://t.co/yBO5wu…)
Name: 339, dtype: object
```

In [78]:

```
tweet_json_clean.loc[694]
```

Out[78]:

```
id_str                                              786709082849828864
created_at                                 Thu Oct 13 23:23:56 +0000 2016
source                 <a href="http://twitter.com/download/iphone" r...
full_text              This is Logan, the Chow who lived. He solemnly...
retweet_count                                                       7069
favorite_count                                                     20296
rating_denominator                                                    10
rating_numerator                                                    9.75
doggo                                                                NaN
floofer                                                              NaN
pupper                                                               NaN
puppo                                                                NaN
user_link                                      https://t.co/yBO5wuqaPS (htt
ps://t.co/yBO5wuqaPS)
Name: 694, dtype: object
```

In [79]:

```
tweet_json_clean.loc[762]
```

Out[79]:

```
id_str                                              778027034220126208
created_at                                 Tue Sep 20 00:24:34 +0000 2016
source                 <a href="http://twitter.com/download/iphone" r...
full_text              This is Sophie. She's a Jubilant Bush Pupper. ...
retweet_count                                                       1885
favorite_count                                                      7320
rating_denominator                                                    10
rating_numerator                                                   11.27
doggo                                                                NaN
floofer                                                              NaN
pupper                                                               NaN
puppo                                                                NaN
user_link                                      https://t.co/QFaUiIHxHq (htt
ps://t.co/QFaUiIHxHq)
Name: 762, dtype: object
```

In [80]:

```
tweet_json_clean.loc[1687]
```

Out[80]:

```
id_str                                              681340665377193984
created_at                                 Mon Dec 28 05:07:27 +0000 2015
source                 <a href="http://twitter.com/download/iphone" r...
full_text              I've been told there's a slight possibility he...
retweet_count                                                        313
favorite_count                                                      1803
rating_denominator                                                    10
rating_numerator                                                       5
doggo                                                                NaN
floofer                                                              NaN
pupper                                                               NaN
puppo                                                                NaN
user_link                                                            NaN
Name: 1687, dtype: object
```

In [81]:

```
tweet_json_clean.loc[1710]
```

Out[81]:

```
id_str                                              680494726643068929
created_at                                 Fri Dec 25 21:06:00 +0000 2015
source                 <a href="http://twitter.com/download/iphone" r...
full_text              Here we have uncovered an entire battalion of ...
retweet_count                                                        542
favorite_count                                                      1879
rating_denominator                                                    10
rating_numerator                                                      26
doggo                                                                NaN
floofer                                                              NaN
pupper                                                            pupper
puppo                                                                NaN
user_link                                  https://t.co/eNm2S6p9BD (htt
ps://t.co/eNm2S6p9BD)
Name: 1710, dtype: object
```

In [82]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 13 columns):
id_str                2354 non-null int64
created_at            2354 non-null object
source                2354 non-null object
full_text             2354 non-null object
retweet_count         2354 non-null int64
favorite_count        2354 non-null int64
rating_denominator    2354 non-null object
rating_numerator      2354 non-null object
doggo                 98 non-null object
floofer               4 non-null object
pupper                271 non-null object
puppo                 37 non-null object
user_link             2225 non-null object
dtypes: int64(3), object(10)
memory usage: 239.2+ KB
```

In [83]:

```
tweet_json_clean[tweet_json_clean['id_str']==883482846933004288]
```

Out[83]:

	id_str	created_at	source	full_text	retweet_cou
44	883482846933004288	Sat Jul 08 00:28:19 +0000 2017	<a href="http://twitter.com/download/iphone" r...	This is Bella. She hopes her smile made you sm...	104(

In [84]:

```
tweet_json_clean[tweet_json_clean['id_str']==832215909146226688]
```

Out[84]:

	id_str	created_at	source	full_text	retweet
339	832215909146226688	Thu Feb 16 13:11:49 +0000 2017	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: This is Logan, the Chow who liv...	

Define:

3-rating_numerator and rating_denominator are currently in str type to be converted to float.:

Code:

In [85]:

```
#converting the data type to float for rating_numerator and rating_denominator columns
tweet_json_clean['rating_numerator']=tweet_json_clean['rating_numerator'].astype('float64')
```

In [86]:

```
tweet_json_clean['rating_denominator']=tweet_json_clean['rating_denominator'].astype('float
```

Test:

In [87]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 13 columns):
id_str                2354 non-null int64
created_at            2354 non-null object
source                2354 non-null object
full_text             2354 non-null object
retweet_count         2354 non-null int64
favorite_count        2354 non-null int64
rating_denominator    2354 non-null float64
rating_numerator      2354 non-null float64
doggo                 98 non-null object
floofer               4 non-null object
pupper                271 non-null object
puppo                 37 non-null object
user_link             2225 non-null object
dtypes: float64(2), int64(3), object(8)
memory usage: 239.2+ KB
```

Define:

4-convert all the Nan value to string None for the columns(doggo,floofer,puppo,pupper,user_links)in order to drop all the Nan values:

Code:

In [88]:

```
#converting the np.NaN to string None in order to drop the NaN values of other columns
tweet_json_clean.doggo=tweet_json_clean.doggo.replace(np.NaN, 'None')
tweet_json_clean.floofer=tweet_json_clean.floofer.replace(np.NaN, 'None')
tweet_json_clean.pupper=tweet_json_clean.pupper.replace(np.NaN, 'None')
tweet_json_clean.puppo=tweet_json_clean.puppo.replace(np.NaN, 'None')
tweet_json_clean.user_link=tweet_json_clean.user_link.replace(np.NaN, 'None')
```

Test:

In [89]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 13 columns):
id_str                2354 non-null int64
created_at            2354 non-null object
source                2354 non-null object
full_text             2354 non-null object
retweet_count         2354 non-null int64
favorite_count        2354 non-null int64
rating_denominator    2354 non-null float64
rating_numerator      2354 non-null float64
doggo                 2354 non-null object
floofer               2354 non-null object
pupper                2354 non-null object
puppo                 2354 non-null object
user_link             2354 non-null object
dtypes: float64(2), int64(3), object(8)
memory usage: 239.2+ KB
```

Define:

5-created_at column is currently an object need to be converted to timestamp and change the column name to timestamp:

Code:

In [90]:

```
#renaming the created_at column to timestamp and converting the column to datetime using pa
tweet_json_clean['timestamp']=pd.to_datetime(tweet_json_clean['created_at'])
tweet_json_clean.drop(['created_at'],axis=1,inplace=True)
```

Test:

In [91]:

```
tweet_json_clean.head(10)
```

Out[91]:

	id_str	source	full_text	retweet_count	favorit
0	892420643555336193	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	8853	
1	892177421306343426	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	6514	
2	891815181378084864	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	4328	
3	891689557279858688	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	8964	
4	891327558926688256	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	9774	
5	891087950875897856	<a href="http://twitter.com/download/iphone" r...	Here we have a majestic great white breaching ...	3261	
6	890971913173991426	<a href="http://twitter.com/download/iphone" r...	Meet Jax. He enjoys ice cream so much he gets ...	2158	
7	890729181411237888	<a href="http://twitter.com/download/iphone" r...	When you watch your owner call another dog a g...	16716	
8	890609185150312448	<a href="http://twitter.com/download/iphone" r...	This is Zoey. She doesn't want to be one of th...	4429	
9	890240255349198849	<a href="http://twitter.com/download/iphone" r...	This is Cassie. She is a college pup. Studying...	7711	

Define:

6-store id_str values to tweet_id and drop id_str column:

Code:

In [93]:

```
#storing id_str to tweet_id and drops the id_str column
```

In [92]:

```
tweet_json_clean['tweet_id']=tweet_json_clean['id_str']
tweet_json_clean.drop('id_str',axis=1,inplace=True)
```

Test:

In [94]:

```
tweet_json_clean.head()
```

Out[94]:

	source	full_text	retweet_count	favorite_count	rating_denom
0	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	8853	39467	
1	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	6514	33819	
2	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	4328	25461	
3	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	8964	42908	
4	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	9774	41048	

In [95]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 13 columns):
source                2354 non-null object
full_text             2354 non-null object
retweet_count         2354 non-null int64
favorite_count        2354 non-null int64
rating_denominator    2354 non-null float64
rating_numerator      2354 non-null float64
doggo                 2354 non-null object
floofer               2354 non-null object
pupper                2354 non-null object
puppo                 2354 non-null object
user_link             2354 non-null object
timestamp             2354 non-null datetime64[ns]
tweet_id              2354 non-null int64
dtypes: datetime64[ns](1), float64(2), int64(3), object(7)
memory usage: 239.2+ KB
```

Define:

7-drop all retweets in the full_text column.

Code:

In [96]:

```
#inorder to drop all the retweet values we need to extract the letter RT which indicates Re
#store it into a column called retweetand then dropping the retweets column
tweet_json_clean['retweet']=tweet_json_clean.full_text.str.extract('(RT)',expand=True)
```

In [97]:

```
tweet_json_clean.retweet.value_counts()
```

Out[97]:

```
RT    190
Name: retweet, dtype: int64
```

In [98]:

```
tweet_json_clean[tweet_json_clean['retweet']=='RT']
```

Out[98]:

	source	full_text	retweet_count	favorite_count
31	<a href="http://twitter.com/download/iphone" r...	RT @Athletics: 12/10 #BATP https://t.co/WxwJmv...	108	0
35	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: This is Lilly. She just paralle...	19297	0
67	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: This is Emmy. She was adopted t...	7181	0
72	<a href="http://twitter.com/download/iphone" r...	RT @dog_rates: Meet Shadow. In an attempt to r...	1349	0
	<a	RT @dog_rates: Meet Terrance. He's		

In [99]:

```
tweet_json_clean.drop(tweet_json_clean.loc[tweet_json_clean['retweet']=='RT'].index, inplac
```

In [100]:

```
tweet_json_clean[tweet_json_clean['retweet']=='RT']
```

Out[100]:

	source	full_text	retweet_count	favorite_count	rating_denominator	rating_numerator	doggo

In [101]:

```
tweet_json_clean.drop('retweet',axis=1,inplace=True)
```

In [102]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2164 entries, 0 to 2353
Data columns (total 13 columns):
source                2164 non-null object
full_text             2164 non-null object
retweet_count         2164 non-null int64
favorite_count        2164 non-null int64
rating_denominator    2164 non-null float64
rating_numerator      2164 non-null float64
doggo                 2164 non-null object
floofer               2164 non-null object
pupper                2164 non-null object
puppo                 2164 non-null object
user_link             2164 non-null object
timestamp             2164 non-null datetime64[ns]
tweet_id              2164 non-null int64
dtypes: datetime64[ns](1), float64(2), int64(3), object(7)
memory usage: 236.7+ KB
```

Tidiness:

Define:

1-merge the datasets tweet_json_clean and image_predict_clean:

Code:

In [103]:

```
#merge the two datasets tweet_json_clean image_predict_clean with the reference of tweet_id
tweet_json_clean=pd.merge(tweet_json_clean,image_predict_clean,on='tweet_id',how='left')
```

Test:

In [104]:

```
tweet_json_clean.head()
```

Out[104]:

	source	full_text	retweet_count	favorite_count	rating_denom
0	<a href="http://twitter.com/download/iphone" r...	This is Phineas. He's a mystical boy. Only eve...	8853	39467	
1	<a href="http://twitter.com/download/iphone" r...	This is Tilly. She's just checking pup on you....	6514	33819	
2	<a href="http://twitter.com/download/iphone" r...	This is Archie. He is a rare Norwegian Pouncin...	4328	25461	
3	<a href="http://twitter.com/download/iphone" r...	This is Darla. She commenced a snooze mid meal...	8964	42908	
4	<a href="http://twitter.com/download/iphone" r...	This is Franklin. He would like you to stop ca...	9774	41048	

5 rows × 26 columns

In [105]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2164 entries, 0 to 2163
Data columns (total 26 columns):
source                2164 non-null object
full_text             2164 non-null object
retweet_count         2164 non-null int64
favorite_count        2164 non-null int64
rating_denominator    2164 non-null float64
rating_numerator      2164 non-null float64
doggo                 2164 non-null object
floofer               2164 non-null object
pupper                2164 non-null object
puppo                 2164 non-null object
user_link             2164 non-null object
timestamp             2164 non-null datetime64[ns]
tweet_id              2164 non-null int64
jpg_url               1986 non-null object
img_num               1986 non-null float64
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null object
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null object
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null object
dog                   1986 non-null object
conf                  1986 non-null object
dtypes: datetime64[ns](1), float64(6), int64(3), object(16)
memory usage: 456.5+ KB
```

Quality:

Define:

7-after mergeing the data sets the tweet_id need to be converted to str.

Code:

In [106]:

```
#converting the tweet_id to str
tweet_json_clean['tweet_id']=tweet_json_clean['tweet_id'].astype(str)
```

Test:

In [107]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2164 entries, 0 to 2163
Data columns (total 26 columns):
source                2164 non-null object
full_text             2164 non-null object
retweet_count         2164 non-null int64
favorite_count        2164 non-null int64
rating_denominator    2164 non-null float64
rating_numerator      2164 non-null float64
doggo                 2164 non-null object
floofer               2164 non-null object
pupper                2164 non-null object
puppo                 2164 non-null object
user_link             2164 non-null object
timestamp             2164 non-null datetime64[ns]
tweet_id              2164 non-null object
jpg_url               1986 non-null object
img_num               1986 non-null float64
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null object
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null object
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null object
dog                   1986 non-null object
conf                  1986 non-null object
dtypes: datetime64[ns](1), float64(6), int64(2), object(17)
memory usage: 456.5+ KB
```

Quality:

Define:

8-Drop all Nan values

Code:

In [108]:

```
#drop all the Nan values in the dataset
tweet_json_clean.isnull().sum()
```

Out[108]:

```
source                0
full_text             0
retweet_count         0
favorite_count        0
rating_denominator    0
rating_numerator      0
doggo                 0
floofer               0
pupper                0
puppo                 0
user_link             0
timestamp             0
tweet_id              0
jpg_url             178
img_num             178
p1                  178
p1_conf             178
p1_dog              178
p2                  178
p2_conf             178
p2_dog              178
p3                  178
p3_conf             178
p3_dog              178
dog                 178
conf                178
dtype: int64
```

In [109]:

```
tweet_json_clean.dropna(inplace=True)
```

Test:

In [110]:

```
tweet_json_clean.isnull().sum()
```

Out[110]:

```
source                0
full_text             0
retweet_count         0
favorite_count        0
rating_denominator    0
rating_numerator      0
doggo                 0
floofer               0
pupper                0
puppo                 0
user_link             0
timestamp             0
tweet_id              0
jpg_url               0
img_num               0
p1                    0
p1_conf               0
p1_dog                0
p2                    0
p2_conf               0
p2_dog                0
p3                    0
p3_conf               0
p3_dog                0
dog                   0
conf                  0
dtype: int64
```

Quality:

Define:

9-after merging img_num column is in the float to be converted to int:

Code:

In [111]:

```
#converting the im_num column to int.
tweet_json_clean.isnull().img_num.value_counts()
```

Out[111]:

```
False    1986
Name: img_num, dtype: int64
```

In [112]:

```
tweet_json_clean['img_num']=tweet_json_clean['img_num'].astype('int64')
```

Test:

In [113]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1986 entries, 0 to 2163
Data columns (total 26 columns):
source                1986 non-null object
full_text             1986 non-null object
retweet_count         1986 non-null int64
favorite_count        1986 non-null int64
rating_denominator    1986 non-null float64
rating_numerator      1986 non-null float64
doggo                 1986 non-null object
floofer               1986 non-null object
pupper                1986 non-null object
puppo                 1986 non-null object
user_link             1986 non-null object
timestamp             1986 non-null datetime64[ns]
tweet_id              1986 non-null object
jpg_url               1986 non-null object
img_num               1986 non-null int64
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null object
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null object
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null object
dog                   1986 non-null object
conf                  1986 non-null object
dtypes: datetime64[ns](1), float64(5), int64(3), object(17)
memory usage: 418.9+ KB
```

In [114]:

```
tweet_json_clean.shape
```

Out[114]:

```
(1986, 26)
```

In [115]:

```
tweet_json_clean.isnull().sum()
```

Out[115]:

```
source                0
full_text             0
retweet_count         0
favorite_count        0
rating_denominator    0
rating_numerator      0
doggo                 0
floofer               0
pupper                0
puppo                 0
user_link             0
timestamp             0
tweet_id              0
jpg_url               0
img_num               0
p1                    0
p1_conf               0
p1_dog                0
p2                    0
p2_conf               0
p2_dog                0
p3                    0
p3_conf               0
p3_dog                0
dog                   0
conf                  0
dtype: int64
```

Quality:

Define:

10-extracting the dog stages from the full_text column and droping the doggo,floofer,pupper and puppo columns.

Code:

In [116]:

```
#extracting the dog stages doggo,floofer,puppo and pupper into one common column called dog
#and drop columns doggo,floofer,puppo and pupper
tweet_json_clean['dog_stage']=tweet_json_clean.full_text.str.extract('(doggo|floofer|pupper
tweet_json_clean['dog_stage']=tweet_json_clean['dog_stage'].replace(np.NaN,'None')
```

In [117]:

```
tweet_json_clean.drop(['doggo','floofer','pupper','puppo'],axis=1,inplace=True)
```

Test:

In [118]:

```
tweet_json_clean.isnull().sum()
```

Out[118]:

```
source                0
full_text             0
retweet_count         0
favorite_count        0
rating_denominator    0
rating_numerator      0
user_link             0
timestamp             0
tweet_id              0
jpg_url               0
img_num               0
p1                    0
p1_conf               0
p1_dog                0
p2                    0
p2_conf               0
p2_dog                0
p3                    0
p3_conf               0
p3_dog                0
dog                   0
conf                  0
dog_stage             0
dtype: int64
```

Tidiness:

Define:

2-Rearrangement of columns:

Code:

In [119]:

```
#rearrangement of columns for great data reading
columns_set=['tweet_id','timestamp','full_text','dog_stage','source','user_link','jpg_url',
tweet_json_clean=tweet_json_clean.reindex(columns=columns_set)
```

Test:

In [120]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1986 entries, 0 to 2163
Data columns (total 23 columns):
tweet_id              1986 non-null object
timestamp             1986 non-null datetime64[ns]
full_text             1986 non-null object
dog_stage             1986 non-null object
source                1986 non-null object
user_link             1986 non-null object
jpg_url               1986 non-null object
rating_numerator      1986 non-null float64
rating_denominator    1986 non-null float64
retweet_count         1986 non-null int64
favorite_count        1986 non-null int64
img_num               1986 non-null int64
dog                   1986 non-null object
conf                  1986 non-null object
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null object
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null object
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null object
dtypes: datetime64[ns](1), float64(5), int64(3), object(14)
memory usage: 372.4+ KB
```

Define:

11-replacment of None to np.Nan for column dog and conf and changing the type of conf column to float64

Code:

In [121]:

```
#replacment of None to np.Nan for column dog and conf and changing the type of conf column
tweet_json_clean['dog']=tweet_json_clean['dog'].replace('None',np.NaN)
```

In [122]:

```
tweet_json_clean['conf']=tweet_json_clean['conf'].replace('None',np.NaN)
```

In [123]:

```
tweet_json_clean['conf']=tweet_json_clean['conf'].astype('float64')
```

Test:

In [124]:

```
tweet_json_clean.isnull().info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1986 entries, 0 to 2163
Data columns (total 23 columns):
tweet_id              1986 non-null bool
timestamp             1986 non-null bool
full_text             1986 non-null bool
dog_stage             1986 non-null bool
source                1986 non-null bool
user_link             1986 non-null bool
jpg_url               1986 non-null bool
rating_numerator      1986 non-null bool
rating_denominator    1986 non-null bool
retweet_count         1986 non-null bool
favorite_count        1986 non-null bool
img_num               1986 non-null bool
dog                   1986 non-null bool
conf                  1986 non-null bool
p1                    1986 non-null bool
p1_conf               1986 non-null bool
p1_dog                1986 non-null bool
p2                    1986 non-null bool
p2_conf               1986 non-null bool
p2_dog                1986 non-null bool
p3                    1986 non-null bool
p3_conf               1986 non-null bool
p3_dog                1986 non-null bool
dtypes: bool(23)
memory usage: 60.1 KB
```

In [125]:

```
tweet_json_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1986 entries, 0 to 2163
Data columns (total 23 columns):
tweet_id              1986 non-null object
timestamp             1986 non-null datetime64[ns]
full_text             1986 non-null object
dog_stage             1986 non-null object
source                1986 non-null object
user_link             1986 non-null object
jpg_url               1986 non-null object
rating_numerator      1986 non-null float64
rating_denominator    1986 non-null float64
retweet_count         1986 non-null int64
favorite_count        1986 non-null int64
img_num               1986 non-null int64
dog                   1678 non-null object
conf                  1678 non-null float64
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null object
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null object
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null object
dtypes: datetime64[ns](1), float64(6), int64(3), object(13)
memory usage: 372.4+ KB
```

In [126]:

```
tweet_json_clean.isnull().sum()
```

Out[126]:

```
tweet_id                0
timestamp               0
full_text               0
dog_stage               0
source                  0
user_link               0
jpg_url                 0
rating_numerator        0
rating_denominator      0
retweet_count           0
favorite_count          0
img_num                 0
dog                   308
conf                  308
p1                      0
p1_conf                 0
p1_dog                  0
p2                      0
p2_conf                 0
p2_dog                  0
p3                      0
p3_conf                 0
p3_dog                  0
dtype: int64
```

Tidiness:

Define:

2-merge the datasets to one master data set.

Code:

In [127]:

```
#the cleaned dataset to twitter-archive-master.csv file
tweet_json_clean.to_csv('twitter_archive_master.csv',index=False)
```

Test:

In [128]:

```
tweet_master=pd.read_csv('twitter_archive_master.csv')
tweet_master.head(10)
```

Out[128]:

	tweet_id	timestamp	full_text	dog_stage	source
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	None	<a href="http://twitter.com/download/iphone" r..
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	None	<a href="http://twitter.com/download/iphone" r..
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	None	<a href="http://twitter.com/download/iphone" r..
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	None	<a href="http://twitter.com/download/iphone" r..
4	891327558926688256	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	None	<a href="http://twitter.com/download/iphone" r..
5	891087950875897856	2017-07-29 00:08:17	Here we have a majestic great white breaching ...	None	<a href="http://twitter.com/download/iphone" r..
6	890971913173991426	2017-07-28 16:27:12	Meet Jax. He enjoys ice cream so much he gets ...	None	<a href="http://twitter.com/download/iphone" r..
7	890729181411237888	2017-07-28 00:22:40	When you watch your owner call another dog a g...	None	<a href="http://twitter.com/download/iphone" r..
8	890609185150312448	2017-07-27 16:25:51	This is Zoey. She doesn't want to be one of th...	None	<a href="http://twitter.com/download/iphone" r..
9	890240255349198849	2017-07-26 15:59:51	This is Cassie. She is a college pup. Studying...	doggo	<a href="http://twitter.com/download/iphone" r..

10 rows × 23 columns

In [129]:

```
tweet_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1986 entries, 0 to 1985
Data columns (total 23 columns):
tweet_id              1986 non-null int64
timestamp             1986 non-null object
full_text             1986 non-null object
dog_stage             1986 non-null object
source                1986 non-null object
user_link             1986 non-null object
jpg_url               1986 non-null object
rating_numerator      1986 non-null float64
rating_denominator    1986 non-null float64
retweet_count         1986 non-null int64
favorite_count        1986 non-null int64
img_num               1986 non-null int64
dog                   1678 non-null object
conf                  1678 non-null float64
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null bool
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null bool
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null bool
dtypes: bool(3), float64(6), int64(4), object(10)
memory usage: 316.2+ KB
```

Quality:

Define:

12-after creating the master data the tweet_id is tyep int to be converted to str.

Code:

In [130]:

```
#after creating the master dataset the tweet_id column is coverted to int that need to be c
tweet_master['tweet_id']=tweet_master['tweet_id'].astype(str)
```

Test:

In [131]:

```
tweet_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1986 entries, 0 to 1985
Data columns (total 23 columns):
tweet_id              1986 non-null object
timestamp             1986 non-null object
full_text             1986 non-null object
dog_stage             1986 non-null object
source                1986 non-null object
user_link             1986 non-null object
jpg_url               1986 non-null object
rating_numerator      1986 non-null float64
rating_denominator    1986 non-null float64
retweet_count         1986 non-null int64
favorite_count        1986 non-null int64
img_num               1986 non-null int64
dog                   1678 non-null object
conf                  1678 non-null float64
p1                    1986 non-null object
p1_conf               1986 non-null float64
p1_dog                1986 non-null bool
p2                    1986 non-null object
p2_conf               1986 non-null float64
p2_dog                1986 non-null bool
p3                    1986 non-null object
p3_conf               1986 non-null float64
p3_dog                1986 non-null bool
dtypes: bool(3), float64(6), int64(3), object(11)
memory usage: 316.2+ KB
```

Analyze and Visualize

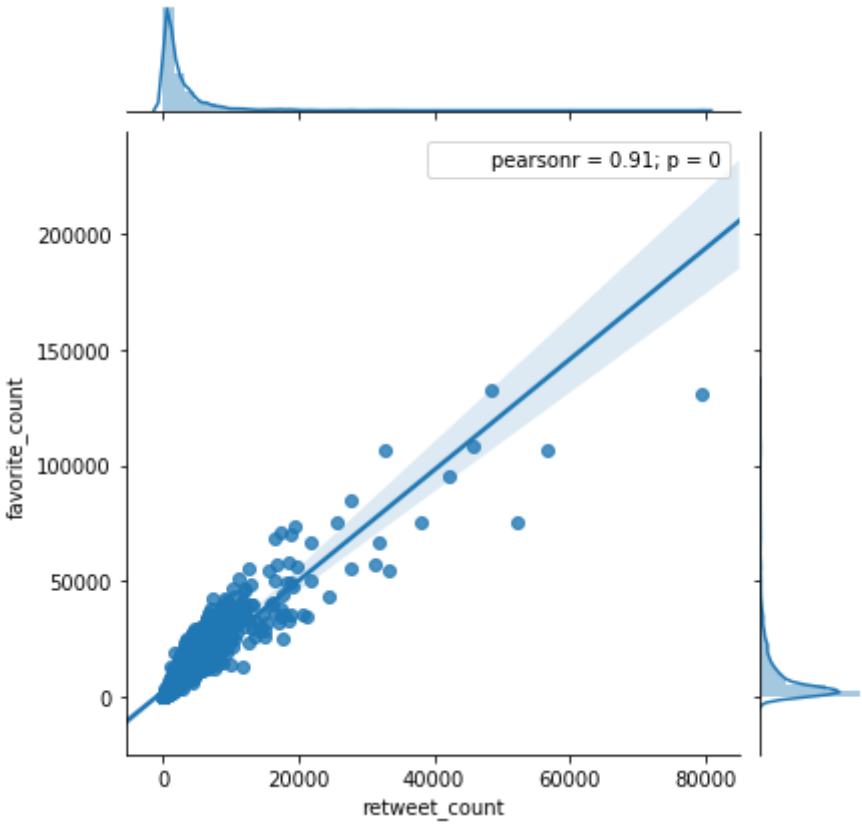
First Insight:

Retweet count v/s Favourite count:

In [133]:

```
g=sb.jointplot('retweet_count','favorite_count',data=tweet_master,kind='reg');
```

```
C:\Users\Prem Kumar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:646
2: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by t
he 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
C:\Users\Prem Kumar\Anaconda3\lib\site-packages\matplotlib\axes\_axes.py:646
2: UserWarning: The 'normed' kwarg is deprecated, and has been replaced by t
he 'density' kwarg.
  warnings.warn("The 'normed' kwarg is deprecated, and has been "
```



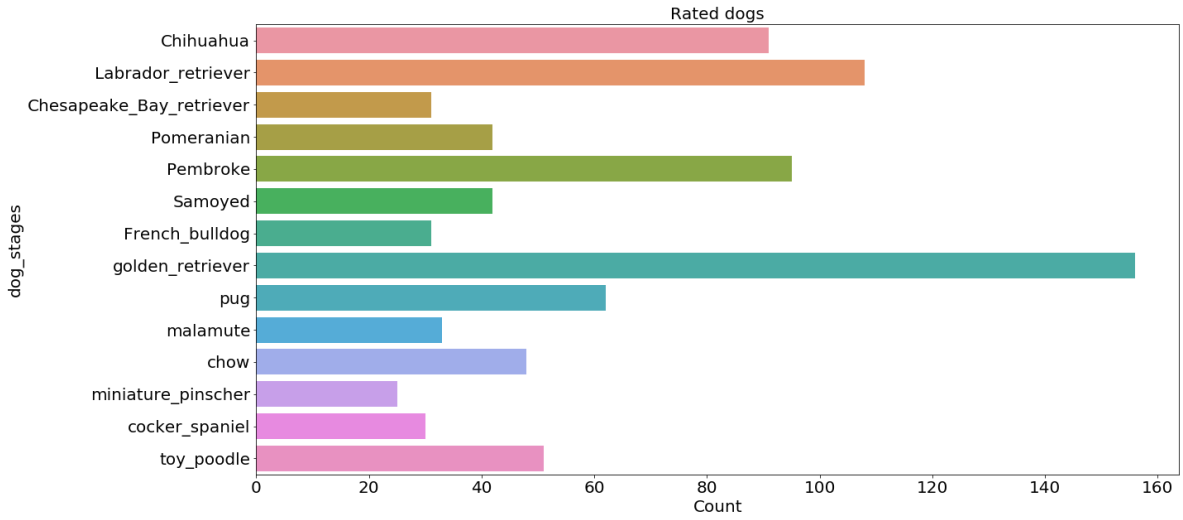
According to the above visualization the favorite count and retweet count are rapidly increasing at the start of the point.But majorly the data losses at an assumption of 30000 favorite count and 15000 retweet count.

Second Insight:

Rated Dogs:

In [134]:

```
dog=tweet_master.groupby('dog').filter(lambda dog:len(dog)>=25)
plt.figure(figsize=(20,10))
sb.countplot(data = dog, y = 'dog');
plt.xticks(fontsize=20);
plt.yticks(fontsize=20)
plt.xlabel('Count',fontsize=20);
plt.ylabel('dog_stages',fontsize=20);
plt.title('Rated dogs',fontsize=20);
```



As per the visualization above the highest dog stage is gloden retriever and lowest dog stage is the Miniature pinscher.

The Golden Retriever is the most commonly used by the peoples.where the count of it is 158.

In [135]:

```
tweet_master.dog.value_counts()
```

Out[135]:

```
golden_retriever                  156
Labrador_retriever                108
Pembroke                           95
Chihuahua                          91
pug                                62
toy_poodle                         51
chow                               48
Pomeranian                         42
Samoyed                            42
malamute                           33
Chesapeake_Bay_retriever           31
French_bulldog                     31
cocker_spaniel                     30
miniature_pinscher                 25
Eskimo_dog                         22
German_shepherd                    21
Cardigan                           21
Staffordshire_bullterrier          20
Shih-Tzu                           20
beagle                             20
Siberian_husky                     20
Maltese_dog                        19
Rottweiler                         19
Lakeland_terrier                   18
kuvasz                             18
Shetland_sheepdog                  18
basset                             17
Italian_greyhound                  17
American_Staffordshire_terrier     16
soft-coated_wheaten_terrier        15
                                  ...
Rhodesian_ridgeback                 4
Scottish_deerhound                  4
Gordon_setter                       4
giant_schnauzer                     4
Tibetan_mastiff                     4
curly-coated_retriever              3
komondor                            3
briard                              3
Saluki                              3
Greater_Swiss_Mountain_dog          3
cairn                               3
Afghan_hound                        3
Leonberg                            3
toy_terrier                         3
Irish_water_spaniel                 3
Brabancon_griffon                   3
wire-haired_fox_terrier             2
black-and-tan_coonhound             2
groenendael                         2
Sussex_spaniel                      2
Australian_terrier                  2
Appenzeller                         2
clumber                             1
standard_schnauzer                  1
```

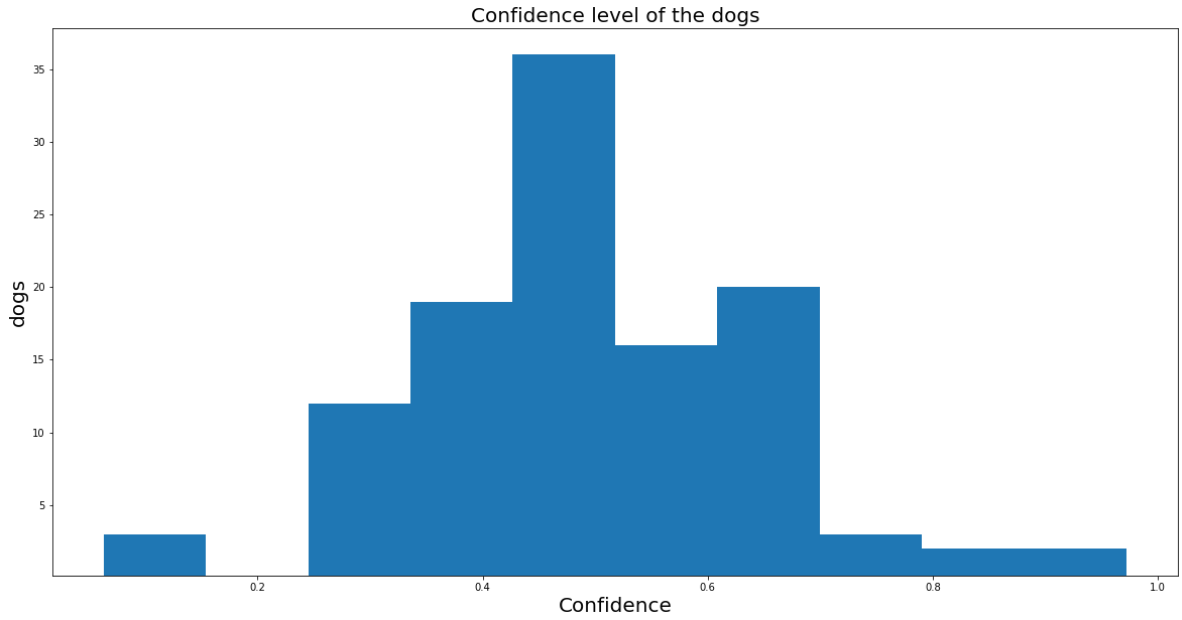
```
Scotch_terrier                  1
Bouvier_des_Flandres            1
Irish_wolfhound                 1
Japanese_spaniel                1
silky_terrier                   1
EntleBucher                     1
Name: dog, Length: 113, dtype: int64
```

Third Insight:

Confidence level of the dogs

In [136]:

```
dog_stage=tweet_master.groupby('dog')['conf'].mean()
plt.figure(figsize=(20,10))
plt.hist(dog_stage);
plt.ylim(0.150);
plt.xlabel('Confidence',fontsize=20);
plt.ylabel('dogs',fontsize=20);
plt.title('Confidence level of the dogs',fontsize=20);
```



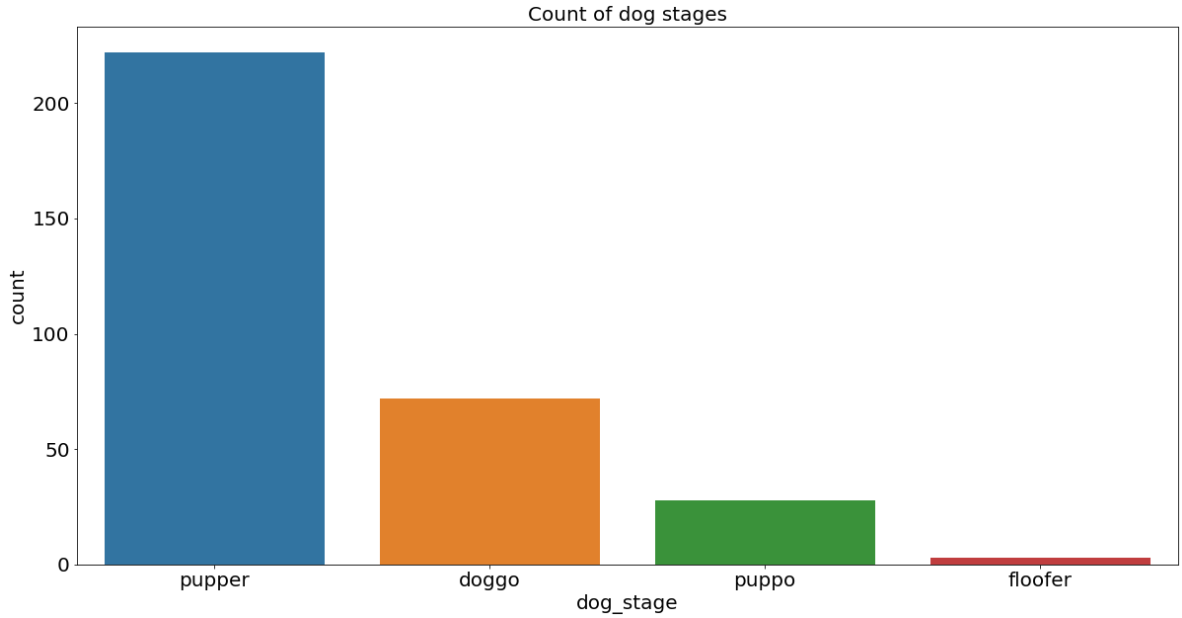
35 number of dogs have the medium confidence level of 0.5, one or two dogs have very low confidence level between 0-0.2 and high confidence level between 0.8-1.0

Fourth Insight:

Count of dog stages

In [137]:

```
tweet_master['dog_stage']=tweet_master.dog_stage.replace('None',np.NaN)
dog_stage=tweet_master.dog_stage.value_counts().index
plt.figure(figsize=(20,10));
sb.countplot(data=tweet_master,x='dog_stage',order=dog_stage);
plt.xlabel('dog_stage',fontsize=20);
plt.ylabel('count',fontsize=20);
plt.xticks(fontsize=20);
plt.yticks(fontsize=20);
plt.title('Count of dog stages',fontsize=20);
```



Note:

Here i have used the tweet-json file i order to practice the wranglind of a data.here i have not used the twitter-archive-enchanced.csv file