Data Wrangling Project

June 19, 2019

1 Project: Data Wrangling WeRateDogs

1.1 Table of Contents

Introduction

This project Real-world data rarely comes clean. Using Python and its libraries, I will gather data from a variety of sources and in a variety of formats, assess its quality and tidiness, then clean it. This is called data wrangling. This task is intended for Udacity Nanodegree Data Wrangling Project.

The dataset that I will be wrangling (and analyzing and visualizing) is the tweet archive of Twitter user @dog_rates, also known as WeRateDogs. The data separated in 3 part, 1 from local file which provided by Udacity, second data from Udacity server, and the last one from twitter API.

The goal: wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations. The Twitter archive is great, but it only contains very basic tweet information. Additional gathering, then assessing and cleaning is required for "Wow!"-worthy analyses and visualizations.

Gathering The Data

I parse my library needed in each task so it will easier to know what kind library needed from that task.

1. Get Data Twitter Archive

Todo: 1. Import library needed 2. Read twitter_archive_enhanced.csv from the same folder 3. Make sure that data has been read correctly - print head

```
In [1]: import pandas as pd
```

In [2]:
```
twitter_archive_df = pd.read_csv('data_udacity/twitter-archive-enhanced.csv')
twitter_archive_df = twitter_archive_df.sort_values('timestamp')
twitter_archive_df.head(2)
```

Out[2]:
```
                 tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
2355  666020888022790149                    NaN                  NaN
2354  666029285002620928                    NaN                  NaN

                      timestamp  \
2355  2015-11-15 22:32:08 +0000
2354  2015-11-15 23:05:30 +0000

                                                 source  \
2355  <a href="http://twitter.com/download/iphone" r...
2354  <a href="http://twitter.com/download/iphone" r...

                                                   text  retweeted_status_id  \
2355  Here we have a Japanese Irish Setter. Lost eye...                  NaN
2354  This is a western brown Mitsubishi terrier. Up...                  NaN

      retweeted_status_user_id retweeted_status_timestamp  \
2355                       NaN                        NaN
2354                       NaN                        NaN

                                          expanded_urls  rating_numerator  \
2355  https://twitter.com/dog_rates/status/666020888...                 8
2354  https://twitter.com/dog_rates/status/666029285...                 7

      rating_denominator  name doggo floofer pupper puppo
2355                  10  None  None    None   None  None
2354                  10     a  None    None   None  None
```

2. Get Data Tweet Image Prediction

Todo: 1. Import library needed 2. Read image-predictions.tsv from Udacity's server that can be access from https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv 3. Make sure that data has been read correctly - print head - describe domain knowledge about the data

In [3]:
```
import requests
```

In [4]:
```
url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictio
response = requests.get(url)

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

In [5]:
```
#Read TSV file
image_prediction_df = pd.read_csv('data_udacity/image-predictions.tsv', sep='\t' )
image_prediction_df.head(2)
```

```
Out[5]:              tweet_id                                          jpg_url  \
        0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg
        1  666029285002620928  https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg

           img_num                      p1   p1_conf  p1_dog                  p2  \
        0        1  Welsh_springer_spaniel  0.465074    True              collie
        1        1                 redbone  0.506826    True  miniature_pinscher

            p2_conf  p2_dog                   p3   p3_conf  p3_dog
        0  0.156665    True    Shetland_sheepdog  0.061428    True
        1  0.074192    True  Rhodesian_ridgeback  0.072010    True
```

The description:

- tweet_id is the last part of the tweet URL after "status/" → https://twitter.com/dog_rates/status/889531135344209921
- p1 is the algorithm's #1 prediction for the image in the tweet → golden retriever
- p1_conf is how confident the algorithm is in its #1 prediction → 95%
- p1_dog is whether or not the #1 prediction is a breed of dog → TRUE
- p2 is the algorithm's second most likely prediction → Labrador retriever
- p2_conf is how confident the algorithm is in its #2 prediction → 1%
- p2_dog is whether or not the #2 prediction is a breed of dog → TRUE
- etc.

3. Configure Twitter Account
Todo: 1. Import library needed 2. Declare twitter configuration with consumer_key, consumer_secret, access_token, and access_secret 3. Make configuration

```
In [6]: import tweepy
```

```
In [7]: # for security reasons, I save my configuration in csv
        twitter_configuration = pd.read_csv("twitter_configuration.csv")
```

```
In [8]: try:
            auth = tweepy.OAuthHandler(twitter_configuration.consumer_key[0], twitter_configurat
            auth.set_access_token(twitter_configuration.access_token[0], twitter_configuration.a
        except tweepy.TweepError as t:
            print(t.message)

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

4. Get Data Twitter with API & JSON
Todo: 1. Import library needed (if not exist before) 2. Get twitter data in JSON by id from file point 1 - add data JSON from a list - add ids data that we can't find that with API - calculate the number id we wan to looking for - calculate number succes and fail data we looking for - save data tweets in txt file so we can accsess that many time 3. Read and save tweets data in dataframe so we can access in our notebook 4. Make sure that data has been read correctly - print head

```
In [9]: import json
        from timeit import default_timer as timer
```

```
In [10]: tweets = []
         ids_not_found_tweet = []
         ids_fail_get_tweet = []
         num_tweet_id = len(twitter_archive_df.tweet_id)
         num_succes_get_data = 0
         num_fail_get_data = 0

         start = timer()
         for tweet_id in twitter_archive_df.tweet_id:
             try:
                 temp = api.get_status(tweet_id)._json
                 tweets.append({'tweet_id':temp['id'],
                                'favorite_count':temp['favorite_count'],
                                'favorited':temp['favorited'],
                                'retweet_count':temp['retweet_count'],
                                'retweeted':temp['retweeted']})
                 num_succes_get_data += 1
                 print('{} : done, {}/{}'.format(tweet_id, num_succes_get_data, num_tweet_id))
             except tweepy.TweepError as t:
                 num_fail_get_data += 1
                 if (t.args[0][0]['message'] == 'No status found with that ID.'):
                     ids_not_found_tweet.append(tweet_id)
                 else:
                     ids_fail_get_tweet.append(tweet_id)
                 print('{} : {}, total fail= {}'.format(tweet_id, t, num_fail_get_data))

         end = timer()
         print("The time we need to get JSON file: {} second".format(end - start))
```

```
666020888022790149 : done, 1/2356
666029285002620928 : done, 2/2356
666033412701032449 : done, 3/2356
666044226329800704 : done, 4/2356
666049248165822465 : done, 5/2356
666050758794694657 : done, 6/2356
666051853826850816 : done, 7/2356
666055525042405380 : done, 8/2356
666057090499244032 : done, 9/2356
666058600524156928 : done, 10/2356
666063827256086533 : done, 11/2356
666071193221509120 : done, 12/2356
666073100786774016 : done, 13/2356
666082916733198337 : done, 14/2356
666094000022159362 : done, 15/2356
666099513787052032 : done, 16/2356
666102155909144576 : done, 17/2356
666104133288665088 : done, 18/2356
666268910803644416 : done, 19/2356
```

```
666273097616637952 : done, 20/2356
666287406224695296 : done, 21/2356
666293911632134144 : done, 22/2356
666337882303524864 : done, 23/2356
666345417576210432 : done, 24/2356
666353288456101888 : done, 25/2356
666362758909284353 : done, 26/2356
666373753744588802 : done, 27/2356
666396247373291520 : done, 28/2356
666407126856765440 : done, 29/2356
666411507551481857 : done, 30/2356
666418789513326592 : done, 31/2356
666421158376562688 : done, 32/2356
666428276349472768 : done, 33/2356
666430724426358785 : done, 34/2356
666435652385423360 : done, 35/2356
666437273139982337 : done, 36/2356
666447344410484738 : done, 37/2356
666454714377183233 : done, 38/2356
666644823164719104 : done, 39/2356
666649482315059201 : done, 40/2356
666691418707132416 : done, 41/2356
666701168228331520 : done, 42/2356
666739327293083650 : done, 43/2356
666776908487630848 : done, 44/2356
666781792255496192 : done, 45/2356
666786068205871104 : done, 46/2356
666804364988780544 : done, 47/2356
666817836334096384 : done, 48/2356
666826780179869698 : done, 49/2356
666835007768551424 : done, 50/2356
666837028449972224 : done, 51/2356
666983947667116034 : done, 52/2356
666996132027977728 : done, 53/2356
667012601033924608 : done, 54/2356
667044094246576128 : done, 55/2356
667062181243039745 : done, 56/2356
667065535570550784 : done, 57/2356
667070482143944705 : done, 58/2356
667073648344346624 : done, 59/2356
667090893657276420 : done, 60/2356
667119796878725120 : done, 61/2356
667138269671505920 : done, 62/2356
667152164079423490 : done, 63/2356
667160273090932737 : done, 64/2356
667165590075940865 : done, 65/2356
667171260800061440 : done, 66/2356
667174963120574464 : done, 67/2356
```

```
667176164155375616 : done, 68/2356
667177989038297088 : done, 69/2356
667182792070062081 : done, 70/2356
667188689915760640 : done, 71/2356
667192066997374976 : done, 72/2356
667200525029539841 : done, 73/2356
667211855547486208 : done, 74/2356
667369227918143488 : done, 75/2356
667393430834667520 : done, 76/2356
667405339315146752 : done, 77/2356
667435689202614272 : done, 78/2356
667437278097252352 : done, 79/2356
667443425659232256 : done, 80/2356
667453023279554560 : done, 81/2356
667455448082227200 : done, 82/2356
667470559035432960 : done, 83/2356
667491009379606528 : done, 84/2356
667495797102141441 : done, 85/2356
667502640335572993 : done, 86/2356
667509364010450944 : done, 87/2356
667517642048163840 : done, 88/2356
667524857454854144 : done, 89/2356
667530908589760512 : done, 90/2356
667534815156183040 : done, 91/2356
667538891197542400 : done, 92/2356
667544320556335104 : done, 93/2356
667546741521195010 : done, 94/2356
667549055577362432 : done, 95/2356
667550882905632768 : done, 96/2356
667550904950915073 : done, 97/2356
667724302356258817 : done, 98/2356
667728196545200128 : done, 99/2356
667766675769573376 : done, 100/2356
667773195014021121 : done, 101/2356
667782464991965184 : done, 102/2356
667793409583771648 : done, 103/2356
667801013445750784 : done, 104/2356
667806454573760512 : done, 105/2356
667832474953625600 : done, 106/2356
667861340749471744 : done, 107/2356
667866724293877760 : done, 108/2356
667873844930215936 : done, 109/2356
667878741721415682 : done, 110/2356
667885044254572545 : done, 111/2356
667886921285246976 : done, 112/2356
667902449697558528 : done, 113/2356
667911425562669056 : done, 114/2356
667915453470232577 : done, 115/2356
```

```
667924896115245057 : done, 116/2356
667937095915278337 : done, 117/2356
668113020489474048 : done, 118/2356
668142349051129856 : done, 119/2356
668154635664932864 : done, 120/2356
668171859951755264 : done, 121/2356
668190681446379520 : done, 122/2356
668204964695683073 : done, 123/2356
668221241640230912 : done, 124/2356
668226093875376128 : done, 125/2356
668237644992782336 : done, 126/2356
668248472370458624 : done, 127/2356
668256321989451776 : done, 128/2356
668268907921326080 : done, 129/2356
668274247790391296 : done, 130/2356
668286279830867968 : done, 131/2356
668291999406125056 : done, 132/2356
668297328638447616 : done, 133/2356
668466899341221888 : done, 134/2356
668480044826800133 : done, 135/2356
668484198282485761 : done, 136/2356
668496999348633600 : done, 137/2356
668507509523615744 : done, 138/2356
668528771708952576 : done, 139/2356
668537837512433665 : done, 140/2356
668542336805281792 : done, 141/2356
668544745690562560 : done, 142/2356
668567822092664832 : done, 143/2356
668587383441514497 : done, 144/2356
668614819948453888 : done, 145/2356
668620235289837568 : done, 146/2356
668623201287675904 : done, 147/2356
668625577880875008 : done, 148/2356
668627278264475648 : done, 149/2356
668631377374486528 : done, 150/2356
668633411083464705 : done, 151/2356
668636665813057536 : done, 152/2356
668641109086707712 : done, 153/2356
668643542311546881 : done, 154/2356
668645506898350081 : done, 155/2356
668655139528511488 : done, 156/2356
668779399630725120 : done, 157/2356
668815180734689280 : done, 158/2356
668826086256599040 : done, 159/2356
668852170888998912 : done, 160/2356
668872652652679168 : done, 161/2356
668892474547511297 : done, 162/2356
668902994700836864 : done, 163/2356
```

```
668932921458302977 : done, 164/2356
668955713004314625 : done, 165/2356
668960084974809088 : done, 166/2356
668967877119254528 : done, 167/2356
668975677807423489 : done, 168/2356
668979806671884288 : done, 169/2356
668981893510119424 : done, 170/2356
668986018524233728 : done, 171/2356
668988183816871936 : done, 172/2356
668989615043424256 : done, 173/2356
668992363537309700 : done, 174/2356
668994913074286592 : done, 175/2356
669000397445533696 : done, 176/2356
669006782128353280 : done, 177/2356
669015743032369152 : done, 178/2356
669037058363662336 : done, 179/2356
669203728096960512 : done, 180/2356
669214165781868544 : done, 181/2356
669216679721873412 : done, 182/2356
669324657376567296 : done, 183/2356
669327207240699904 : done, 184/2356
669328503091937280 : done, 185/2356
669351434509529089 : done, 186/2356
669353438988365824 : done, 187/2356
669354382627049472 : done, 188/2356
669359674819481600 : done, 189/2356
669363888236994561 : done, 190/2356
669367896104181761 : done, 191/2356
669371483794317312 : done, 192/2356
669375718304980992 : done, 193/2356
669393256313184256 : done, 194/2356
669564461267722241 : done, 195/2356
669567591774625800 : done, 196/2356
669571471778410496 : done, 197/2356
669573570759163904 : done, 198/2356
669583744538451968 : done, 199/2356
669597912108789760 : done, 200/2356
669603084620980224 : done, 201/2356
669625907762618368 : done, 202/2356
669661792646373376 : done, 203/2356
669680153564442624 : done, 204/2356
669682095984410625 : done, 205/2356
669683899023405056 : done, 206/2356
669684865554620416 : done, 207/2356
669749430875258880 : done, 208/2356
669753178989142016 : done, 209/2356
669923323644657664 : done, 210/2356
669926384437997569 : done, 211/2356
```

```
669942763794931712 : done, 212/2356
669970042633789440 : done, 213/2356
669972011175813120 : done, 214/2356
669993076832759809 : done, 215/2356
670003130994700288 : done, 216/2356
670037189829525505 : done, 217/2356
670040295598354432 : done, 218/2356
670046952931721218 : done, 219/2356
670055038660800512 : done, 220/2356
670061506722140161 : done, 221/2356
670069087419133954 : done, 222/2356
670073503555706880 : done, 223/2356
670079681849372674 : done, 224/2356
670086499208155136 : done, 225/2356
670093938074779648 : done, 226/2356
670290420111441920 : done, 227/2356
670303360680108032 : done, 228/2356
670319130621435904 : done, 229/2356
670338931251150849 : done, 230/2356
670361874861563904 : done, 231/2356
670374371102445568 : done, 232/2356
670385711116361728 : done, 233/2356
670403879788544000 : done, 234/2356
670408998013820928 : done, 235/2356
670411370698022913 : done, 236/2356
670417414769758208 : done, 237/2356
670420569653809152 : done, 238/2356
670421925039075328 : done, 239/2356
670427002554466305 : done, 240/2356
670428280563085312 : done, 241/2356
670433248821026816 : done, 242/2356
670434127938719744 : done, 243/2356
670435821946826752 : done, 244/2356
670442337873600512 : done, 245/2356
670444955656130560 : done, 246/2356
670449342516494336 : done, 247/2356
670452855871037440 : done, 248/2356
670465786746662913 : done, 249/2356
670468609693655041 : done, 250/2356
670474236058800128 : done, 251/2356
670668383499735048 : done, 252/2356
670676092097810432 : done, 253/2356
670679630144274432 : done, 254/2356
670691627984359425 : done, 255/2356
670704688707301377 : done, 256/2356
670717338665226240 : done, 257/2356
670727704916926465 : done, 258/2356
670733412878163972 : done, 259/2356
```

```
670755717859713024 : done, 260/2356
670764103623966721 : done, 261/2356
670778058496974848 : done, 262/2356
670780561024270336 : done, 263/2356
670782429121134593 : done, 264/2356
670783437142401025 : done, 265/2356
670786190031921152 : done, 266/2356
670789397210615808 : done, 267/2356
670792680469889025 : done, 268/2356
670797304698376195 : done, 269/2356
670803562457407488 : done, 270/2356
670804601705242624 : done, 271/2356
670807719151067136 : done, 272/2356
670811965569282048 : done, 273/2356
670815497391357952 : done, 274/2356
670822709593571328 : done, 275/2356
670823764196741120 : done, 276/2356
670826280409919488 : done, 277/2356
670832455012716544 : done, 278/2356
670833812859932673 : done, 279/2356
670838202509447168 : done, 280/2356
670840546554966016 : done, 281/2356
670842764863651840 : done, 282/2356
670995969505435648 : done, 283/2356
671109016219725825 : done, 284/2356
671115716440031232 : done, 285/2356
671122204919246848 : done, 286/2356
671134062904504320 : done, 287/2356
671138694582165504 : done, 288/2356
671141549288370177 : done, 289/2356
671147085991960577 : done, 290/2356
671151324042559489 : done, 291/2356
671154572044468225 : done, 292/2356
671159727754231808 : done, 293/2356
671163268581498880 : done, 294/2356
671166507850801152 : done, 295/2356
671182547775299584 : done, 296/2356
671186162933985280 : done, 297/2356
671347597085433856 : done, 298/2356
671355857343524864 : done, 299/2356
671357843010908160 : done, 300/2356
671362598324076544 : done, 301/2356
671390180817915904 : done, 302/2356
671485057807351808 : done, 303/2356
671486386088865792 : done, 304/2356
671488513339211776 : done, 305/2356
671497587707535361 : done, 306/2356
671504605491109889 : done, 307/2356
```

```
671511350426865664 : done, 308/2356
671518598289059840 : done, 309/2356
671520732782923777 : done, 310/2356
671528761649688577 : done, 311/2356
671533943490011136 : done, 312/2356
671536543010570240 : done, 313/2356
671538301157904385 : done, 314/2356
671542985629241344 : done, 315/2356
671544874165002241 : done, 316/2356
671547767500775424 : done, 317/2356
671550332464455680 : done, 318/2356
671561002136281088 : done, 319/2356
671729906628341761 : done, 320/2356
671735591348891648 : done, 321/2356
671743150407421952 : done, 322/2356
671744970634719232 : done, 323/2356
671763349865160704 : done, 324/2356
671768281401958400 : done, 325/2356
671789708968640512 : done, 326/2356
671855973984772097 : done, 327/2356
671866342182637568 : done, 328/2356
671874878652489728 : done, 329/2356
671879137494245376 : done, 330/2356
671882082306625538 : done, 331/2356
671891728106971137 : done, 332/2356
671896809300709376 : done, 333/2356
672068090318987265 : done, 334/2356
672082170312290304 : done, 335/2356
672095186491711488 : done, 336/2356
672125275208069120 : done, 337/2356
672139350159835138 : done, 338/2356
672160042234327040 : done, 339/2356
672169685991993344 : done, 340/2356
672205392827572224 : done, 341/2356
672222792075620352 : done, 342/2356
672231046314901505 : done, 343/2356
672239279297454080 : done, 344/2356
672245253877968896 : done, 345/2356
672248013293752320 : done, 346/2356
672254177670729728 : done, 347/2356
672256522047614977 : done, 348/2356
672264251789176834 : done, 349/2356
672267570918129665 : done, 350/2356
672272411274932228 : done, 351/2356
672466075045466113 : done, 352/2356
672475084225949696 : done, 353/2356
672481316919734272 : done, 354/2356
672482722825261057 : done, 355/2356
```

```
672488522314567680 : done, 356/2356
672523490734551040 : done, 357/2356
672538107540070400 : done, 358/2356
672591271085670400 : done, 359/2356
672591762242805761 : done, 360/2356
672594978741354496 : done, 361/2356
672604026190569472 : done, 362/2356
672609152938721280 : done, 363/2356
672614745925664768 : done, 364/2356
672622327801233409 : done, 365/2356
672640509974827008 : done, 366/2356
672828477930868736 : done, 367/2356
672834301050937345 : done, 368/2356
672877615439593473 : done, 369/2356
672884426393653248 : done, 370/2356
672898206762672129 : done, 371/2356
672902681409806336 : done, 372/2356
672964561327235073 : done, 373/2356
672968025906282496 : done, 374/2356
672970152493887488 : done, 375/2356
672975131468300288 : done, 376/2356
672980819271634944 : done, 377/2356
672984142909456390 : done, 378/2356
672988786805112832 : done, 379/2356
672995267319328768 : done, 380/2356
672997845381865473 : done, 381/2356
673148804208660480 : done, 382/2356
673213039743795200 : done, 383/2356
673240798075449344 : done, 384/2356
673270968295534593 : done, 385/2356
673295268553605120 : done, 386/2356
673317986296586240 : done, 387/2356
673320132811366400 : done, 388/2356
673342308415348736 : done, 389/2356
673343217010679808 : done, 390/2356
673345638550134785 : done, 391/2356
673350198937153538 : done, 392/2356
673352124999274496 : done, 393/2356
673355879178194945 : done, 394/2356
673359818736984064 : done, 395/2356
673363615379013632 : done, 396/2356
673576835670777856 : done, 397/2356
673580926094458881 : done, 398/2356
673583129559498752 : done, 399/2356
673612854080196609 : done, 400/2356
673636718965334016 : done, 401/2356
673656262056419329 : done, 402/2356
673662677122719744 : done, 403/2356
```

```
673680198160809984 : done, 404/2356
673686845050527744 : done, 405/2356
673688752737402881 : done, 406/2356
673689733134946305 : done, 407/2356
673697980713705472 : done, 408/2356
673700254269775872 : done, 409/2356
673705679337693185 : done, 410/2356
673707060090052608 : done, 411/2356
673708611235921920 : done, 412/2356
673709992831262724 : done, 413/2356
673711475735838725 : done, 414/2356
673715861853720576 : done, 415/2356
673716320723169284 : done, 416/2356
673887867907739649 : done, 417/2356
673906403526995968 : done, 418/2356
673919437611909120 : done, 419/2356
673956914389192708 : done, 420/2356
674008982932058114 : done, 421/2356
674014384960745472 : done, 422/2356
674019345211760640 : done, 423/2356
674024893172875264 : done, 424/2356
674036086168010753 : done, 425/2356
674038233588723717 : done, 426/2356
674042553264685056 : done, 427/2356
674045139690631169 : done, 428/2356
674051556661161984 : done, 429/2356
674053186244734976 : done, 430/2356
674063288070742018 : done, 431/2356
674075285688614912 : done, 432/2356
674082852460433408 : done, 433/2356
674255168825880576 : done, 434/2356
674262580978937856 : done, 435/2356
674265582246694913 : done, 436/2356
674269164442398721 : done, 437/2356
674271431610523648 : done, 438/2356
674291837063053312 : done, 439/2356
674307341513269249 : done, 440/2356
674318007229923329 : done, 441/2356
674330906434379776 : done, 442/2356
674372068062928900 : done, 443/2356
674394782723014656 : done, 444/2356
674410619106390016 : done, 445/2356
674416750885273600 : done, 446/2356
674422304705744896 : done, 447/2356
674436901579923456 : done, 448/2356
674447403907457024 : done, 449/2356
674468880899788800 : done, 450/2356
674606911342424069 : done, 451/2356
```

```
674632714662858753 : done, 452/2356
674638615994089473 : done, 453/2356
674644256330530816 : done, 454/2356
674646392044941312 : done, 455/2356
674664755118911488 : done, 456/2356
674670581682434048 : done, 457/2356
674690135443775488 : done, 458/2356
674737130913071104 : done, 459/2356
674739953134403584 : done, 460/2356
674742531037511680 : done, 461/2356
674743008475090944 : done, 462/2356
674752233200820224 : done, 463/2356
674754018082705410 : done, 464/2356
674764817387900928 : done, 465/2356
674767892831932416 : done, 466/2356
674774481756377088 : done, 467/2356
674781762103414784 : done, 468/2356
674788554665512960 : done, 469/2356
674790488185167872 : done, 470/2356
674793399141146624 : done, 471/2356
674800520222154752 : done, 472/2356
674805413498527744 : done, 473/2356
674999807681908736 : done, 474/2356
675003128568291329 : done, 475/2356
675006312288268288 : done, 476/2356
675015141583413248 : done, 477/2356
675047298674663426 : done, 478/2356
675109292475830276 : done, 479/2356
675111688094527488 : done, 480/2356
675113801096802304 : done, 481/2356
675135153782571009 : done, 482/2356
675145476954566656 : done, 483/2356
675146535592706048 : done, 484/2356
675147105808306176 : done, 485/2356
675149409102012420 : done, 486/2356
675153376133427200 : done, 487/2356
675166823650848770 : done, 488/2356
675334060156301312 : done, 489/2356
675349384339542016 : done, 490/2356
675354435921575936 : done, 491/2356
675362609739206656 : done, 492/2356
675372240448454658 : done, 493/2356
675432746517426176 : done, 494/2356
675483430902214656 : done, 495/2356
675489971617296384 : done, 496/2356
675497103322386432 : done, 497/2356
675501075957489664 : done, 498/2356
675517828909424640 : done, 499/2356
```

```
675522403582218240 : done, 500/2356
675531475945709568 : done, 501/2356
675534494439489536 : done, 502/2356
675706639471788032 : done, 503/2356
675707330206547968 : done, 504/2356
675710890956750848 : done, 505/2356
675740360753160193 : done, 506/2356
675781562965868544 : done, 507/2356
675798442703122432 : done, 508/2356
675820929667219457 : done, 509/2356
675822767435051008 : done, 510/2356
675845657354215424 : done, 511/2356
675849018447167488 : done, 512/2356
675853064436391936 : done, 513/2356
675870721063669760 : done, 514/2356
675878199931371520 : done, 515/2356
675888385639251968 : done, 516/2356
675891555769696257 : done, 517/2356
675898130735476737 : done, 518/2356
676089483918516224 : done, 519/2356
676098748976615425 : done, 520/2356
676101918813499392 : done, 521/2356
676121918416756736 : done, 522/2356
676146341966438401 : done, 523/2356
676191832485810177 : done, 524/2356
676215927814406144 : done, 525/2356
676219687039057920 : done, 526/2356
676237365392908289 : done, 527/2356
676263575653122048 : done, 528/2356
676430933382295552 : done, 529/2356
676440007570247681 : done, 530/2356
676470639084101634 : done, 531/2356
676496375194980353 : done, 532/2356
676533798876651520 : done, 533/2356
676575501977128964 : done, 534/2356
676582956622721024 : done, 535/2356
676588346097852417 : done, 536/2356
676590572941893632 : done, 537/2356
676593408224403456 : done, 538/2356
676603393314578432 : done, 539/2356
676606785097199616 : done, 540/2356
676613908052996102 : done, 541/2356
676617503762681856 : done, 542/2356
676776431406465024 : done, 543/2356
676811746707918848 : done, 544/2356
676819651066732545 : done, 545/2356
676821958043033607 : done, 546/2356
676864501615042560 : done, 547/2356
```

```
676897532954456065 : done, 548/2356
676916996760600576 : done, 549/2356
676936541936185344 : done, 550/2356
676942428000112642 : done, 551/2356
676946864479084545 : done, 552/2356
676948236477857792 : done, 553/2356
676949632774234114 : done, 554/2356
676957860086095872 : done, 555/2356
676975532580409345 : done, 556/2356
677187300187611136 : done, 557/2356
677228873407442944 : done, 558/2356
677269281705472000 : done, 559/2356
677301033169788928 : done, 560/2356
677314812125323265 : done, 561/2356
677328882937298944 : done, 562/2356
677331501395156992 : done, 563/2356
677334615166730240 : done, 564/2356
677335745548390400 : done, 565/2356
677530072887205888 : done, 566/2356
677547928504967168 : done, 567/2356
677557565589463040 : done, 568/2356
677565715327688705 : done, 569/2356
677573743309385728 : done, 570/2356
677644091929329666 : done, 571/2356
677662372920729601 : done, 572/2356
677673981332312066 : done, 573/2356
677687604918272002 : done, 574/2356
677698403548192770 : done, 575/2356
677700003327029250 : done, 576/2356
677716515794329600 : done, 577/2356
677895101218201600 : done, 578/2356
677918531514703872 : done, 579/2356
677961670166224897 : done, 580/2356
678021115718029313 : done, 581/2356
678023323247357953 : done, 582/2356
678255464182861824 : done, 583/2356
678278586130948096 : done, 584/2356
678334497360859136 : done, 585/2356
678341075375947776 : done, 586/2356
678380236862578688 : done, 587/2356
678389028614488064 : done, 588/2356
678396796259975168 : done, 589/2356
678399652199309312 : done, 590/2356
678410210315247616 : done, 591/2356
678424312106393600 : done, 592/2356
678446151570427904 : done, 593/2356
678643457146150913 : done, 594/2356
678675843183484930 : done, 595/2356
```

```
678708137298427904 : done, 596/2356
678740035362037760 : done, 597/2356
678755239630127104 : done, 598/2356
678764513869611008 : done, 599/2356
678767140346941444 : done, 600/2356
678774928607469569 : done, 601/2356
678798276842360832 : done, 602/2356
678800283649069056 : done, 603/2356
678969228704284672 : done, 604/2356
678991772295516161 : done, 605/2356
679001094530465792 : done, 606/2356
679047485189439488 : done, 607/2356
679062614270468097 : done, 608/2356
679111216690831360 : done, 609/2356
679132435750195208 : done, 610/2356
679148763231985668 : done, 611/2356
679158373988876288 : done, 612/2356
679405845277462528 : done, 613/2356
679462823135686656 : done, 614/2356
679475951516934144 : done, 615/2356
679503373272485890 : done, 616/2356
679511351870550016 : done, 617/2356
679527802031484928 : done, 618/2356
679530280114372609 : done, 619/2356
679722016581222400 : done, 620/2356
679729593985699840 : done, 621/2356
679736210798047232 : done, 622/2356
679777920601223168 : done, 623/2356
679828447187857408 : done, 624/2356
679844490799091713 : done, 625/2356
679854723806179328 : done, 626/2356
679862121895714818 : done, 627/2356
679872969355714560 : done, 628/2356
679877062409191424 : done, 629/2356
680055455951884288 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 1
680070545539371008 : done, 630/2356
680085611152338944 : done, 631/2356
680100725817409536 : done, 632/2356
680115823365742593 : done, 633/2356
680130881361686529 : done, 634/2356
680145970311643136 : done, 635/2356
680161097740095489 : done, 636/2356
680176173301628928 : done, 637/2356
680191257256136705 : done, 638/2356
680206703334408192 : done, 639/2356
680221482581123072 : done, 640/2356
680440374763077632 : done, 641/2356
680473011644985345 : done, 642/2356
```

```
680494726643068929 : done, 643/2356
680497766108381184 : done, 644/2356
680583894916304897 : done, 645/2356
680609293079592961 : done, 646/2356
680798457301471234 : done, 647/2356
680801747103793152 : done, 648/2356
680805554198020098 : done, 649/2356
680836378243002368 : done, 650/2356
680889648562991104 : done, 651/2356
680913438424612864 : done, 652/2356
680934982542561280 : done, 653/2356
680940246314430465 : done, 654/2356
680959110691590145 : done, 655/2356
680970795137544192 : done, 656/2356
681193455364796417 : done, 657/2356
681231109724700672 : done, 658/2356
681242418453299201 : done, 659/2356
681261549936340994 : done, 660/2356
681281657291280384 : done, 661/2356
681297372102656000 : done, 662/2356
681302363064414209 : done, 663/2356
681320187870711809 : done, 664/2356
681339448655802368 : done, 665/2356
681340665377193984 : done, 666/2356
681523177663676416 : done, 667/2356
681579835668455424 : done, 668/2356
681610798867845120 : done, 669/2356
681654059175129088 : done, 670/2356
681679526984871937 : done, 671/2356
681694085539872773 : done, 672/2356
681891461017812993 : done, 673/2356
681981167097122816 : done, 674/2356
682003177596559360 : done, 675/2356
682032003584274432 : done, 676/2356
682047327939461121 : done, 677/2356
682059653698686977 : done, 678/2356
682088079302213632 : done, 679/2356
682242692827447297 : done, 680/2356
682259524040966145 : done, 681/2356
682303737705140231 : done, 682/2356
682389078323662849 : done, 683/2356
682393905736888321 : done, 684/2356
682406705142087680 : done, 685/2356
682429480204398592 : done, 686/2356
682638830361513985 : done, 687/2356
682662431982772225 : done, 688/2356
682697186228989953 : done, 689/2356
682750546109968385 : done, 690/2356
```

```
682788441537560576 : done, 691/2356
682808988178739200 : done, 692/2356
682962037429899265 : done, 693/2356
683030066213818368 : done, 694/2356
683078886620553216 : done, 695/2356
683098815881154561 : done, 696/2356
683111407806746624 : done, 697/2356
683142553609318400 : done, 698/2356
683357973142474752 : done, 699/2356
683391852557561860 : done, 700/2356
683449695444799489 : done, 701/2356
683462770029932544 : done, 702/2356
683481228088049664 : done, 703/2356
683498322573824003 : done, 704/2356
683515932363329536 : done, 705/2356
683742671509258241 : done, 706/2356
683773439333797890 : done, 707/2356
683828599284170753 : done, 708/2356
683834909291606017 : done, 709/2356
683849932751646720 : done, 710/2356
683852578183077888 : done, 711/2356
683857920510050305 : done, 712/2356
684097758874210310 : done, 713/2356
684122891630342144 : done, 714/2356
684147889187209216 : done, 715/2356
684177701129875456 : done, 716/2356
684188786104872960 : done, 717/2356
684195085588783105 : done, 718/2356
684200372118904832 : done, 719/2356
684222868335505415 : done, 720/2356
684225744407494656 : done, 721/2356
684241637099323392 : done, 722/2356
684460069371654144 : done, 723/2356
684481074559381504 : done, 724/2356
684538444857667585 : done, 725/2356
684567543613382656 : done, 726/2356
684588130326986752 : done, 727/2356
684594889858887680 : done, 728/2356
684800227459624960 : done, 729/2356
684830982659280897 : done, 730/2356
684880619965411328 : done, 731/2356
684902183876321280 : done, 732/2356
684914660081053696 : done, 733/2356
684926975086034944 : done, 734/2356
684940049151070208 : done, 735/2356
684959798585110529 : done, 736/2356
684969860808454144 : done, 737/2356
685169283572338688 : done, 738/2356
```

```
685198997565345792 : done, 739/2356
685268753634967552 : done, 740/2356
685307451701334016 : done, 741/2356
685315239903100929 : done, 742/2356
685321586178670592 : done, 743/2356
685325112850124800 : done, 744/2356
685532292383666176 : done, 745/2356
685547936038666240 : done, 746/2356
685641971164143616 : done, 747/2356
685663452032069632 : done, 748/2356
685667379192414208 : done, 749/2356
685681090388975616 : done, 750/2356
685906723014619143 : done, 751/2356
685943807276412928 : done, 752/2356
685973236358713344 : done, 753/2356
686003207160610816 : done, 754/2356
686007916130873345 : done, 755/2356
686034024800862208 : done, 756/2356
686035780142297088 : done, 757/2356
686050296934563840 : done, 758/2356
686286779679375361 : done, 759/2356
686358356425093120 : done, 760/2356
686377065986265092 : done, 761/2356
686386521809772549 : done, 762/2356
686394059078897668 : done, 763/2356
686606069955735556 : done, 764/2356
686618349602762752 : done, 765/2356
686683045143953408 : done, 766/2356
686730991906516992 : done, 767/2356
686749460672679938 : done, 768/2356
686760001961103360 : done, 769/2356
686947101016735744 : done, 770/2356
687096057537363968 : done, 771/2356
687102708889812993 : done, 772/2356
687109925361856513 : done, 773/2356
687124485711986689 : done, 774/2356
687127927494963200 : done, 775/2356
687312378585812992 : done, 776/2356
687317306314240000 : done, 777/2356
687399393394311168 : done, 778/2356
687460506001633280 : done, 779/2356
687476254459715584 : done, 780/2356
687480748861947905 : done, 781/2356
687494652870668288 : done, 782/2356
687664829264453632 : done, 783/2356
687704180304273409 : done, 784/2356
687732144991551489 : done, 785/2356
687807801670897665 : done, 786/2356
```

```
687818504314159109 : done, 787/2356
687826841265172480 : done, 788/2356
687841446767013888 : done, 789/2356
688064179421470721 : done, 790/2356
688116655151435777 : done, 791/2356
688179443353796608 : done, 792/2356
688211956440801280 : done, 793/2356
688385280030670848 : done, 794/2356
688519176466644993 : done, 795/2356
688547210804498433 : done, 796/2356
688789766343622656 : done, 797/2356
688804835492233216 : done, 798/2356
688828561667567616 : done, 799/2356
688894073864884227 : done, 800/2356
688898160958271489 : done, 801/2356
688908934925697024 : done, 802/2356
688916208532455424 : done, 803/2356
689143371370250240 : done, 804/2356
689154315265683456 : done, 805/2356
689255633275777024 : done, 806/2356
689275259254616065 : done, 807/2356
689280876073582592 : done, 808/2356
689283819090870273 : done, 809/2356
689289219123089408 : done, 810/2356
689517482558820352 : done, 811/2356
689557536375177216 : done, 812/2356
689599056876867584 : done, 813/2356
689623661272240129 : done, 814/2356
689659372465688576 : done, 815/2356
689661964914655233 : done, 816/2356
689835978131935233 : done, 817/2356
689877686181715968 : done, 818/2356
689905486972461056 : done, 819/2356
689977555533848577 : done, 820/2356
689993469801164801 : done, 821/2356
689999384604450816 : done, 822/2356
690005060500217858 : done, 823/2356
690015576308211712 : done, 824/2356
690021994562220032 : done, 825/2356
690248561355657216 : done, 826/2356
690348396616552449 : done, 827/2356
690360449368465409 : done, 828/2356
690374419777196032 : done, 829/2356
690400367696297985 : done, 830/2356
690597161306841088 : done, 831/2356
690607260360429569 : done, 832/2356
690649993829576704 : done, 833/2356
690690673629138944 : done, 834/2356
```

```
690728923253055490 : done, 835/2356
690735892932222976 : done, 836/2356
690932576555528194 : done, 837/2356
690938899477221376 : done, 838/2356
690959652130045952 : done, 839/2356
690989312272396288 : done, 840/2356
691090071332753408 : done, 841/2356
691096613310316544 : done, 842/2356
691321916024623104 : done, 843/2356
691416866452082688 : done, 844/2356
691444869282295808 : done, 845/2356
691459709405118465 : done, 846/2356
691483041324204033 : done, 847/2356
691675652215414786 : done, 848/2356
691756958957883396 : done, 849/2356
691793053716221953 : done, 850/2356
691820333922455552 : done, 851/2356
692017291282812928 : done, 852/2356
692041934689402880 : done, 853/2356
692142790915014657 : done, 854/2356
692158366030913536 : done, 855/2356
692187005137076224 : done, 856/2356
692417313023332352 : done, 857/2356
692423280028966913 : done, 858/2356
692530551048294401 : done, 859/2356
692535307825213440 : done, 860/2356
692568918515392513 : done, 861/2356
692752401762250755 : done, 862/2356
692828166163931137 : done, 863/2356
692894228850999298 : done, 864/2356
692901601640583168 : done, 865/2356
692905862751522816 : done, 866/2356
692919143163629568 : done, 867/2356
693095443459342336 : done, 868/2356
693109034023534592 : done, 869/2356
693155686491000832 : done, 870/2356
693231807727280129 : done, 871/2356
693262851218264065 : done, 872/2356
693267061318012928 : done, 873/2356
693280720173801472 : done, 874/2356
693486665285931008 : done, 875/2356
693582294167244802 : done, 876/2356
693590843962331137 : done, 877/2356
693622659251335168 : done, 878/2356
693629975228977152 : done, 879/2356
693642232151285760 : done, 880/2356
693644216740769793 : done, 881/2356
693647888581312512 : done, 882/2356
```

```
693942351086120961 : done, 883/2356
693993230313091072 : done, 884/2356
694001791655137281 : done, 885/2356
694183373896572928 : done, 886/2356
694206574471057408 : done, 887/2356
694329668942569472 : done, 888/2356
694342028726001664 : done, 889/2356
694352839993344000 : done, 890/2356
694356675654983680 : done, 891/2356
694669722378485760 : done, 892/2356
694905863685980160 : done, 893/2356
694925794720792577 : done, 894/2356
695051054296211456 : done, 895/2356
695064344191721472 : done, 896/2356
695074328191332352 : done, 897/2356
695095422348574720 : done, 898/2356
695314793360662529 : done, 899/2356


Rate limit reached. Sleeping for: 604


695409464418041856 : done, 900/2356
695446424020918272 : done, 901/2356
695629776980148225 : done, 902/2356
695767669421768709 : done, 903/2356
695794761660297217 : done, 904/2356
695816827381944320 : done, 905/2356
696100768806522880 : done, 906/2356
696405997980676096 : done, 907/2356
696488710901260288 : done, 908/2356
696490539101908992 : done, 909/2356
696518437233913856 : done, 910/2356
696713835009417216 : done, 911/2356
696744641916489729 : done, 912/2356
696754882863349760 : done, 913/2356
696877980375769088 : done, 914/2356
696886256886657024 : done, 915/2356
696894894812565505 : done, 916/2356
696900204696625153 : done, 917/2356
697242256848379904 : done, 918/2356
697255105972801536 : done, 919/2356
697259378236399616 : done, 920/2356
697270446429966336 : done, 921/2356
697463031882764288 : done, 922/2356
697482927769255936 : done, 923/2356
697516214579523584 : done, 924/2356
697575480820686848 : done, 925/2356
```

```
697596423848730625 : done, 926/2356
697616773278015490 : done, 927/2356
697630435728322560 : done, 928/2356
697881462549430272 : done, 929/2356
697943111201378304 : done, 930/2356
697990423684476929 : done, 931/2356
697995514407682048 : done, 932/2356
698178924120031232 : done, 933/2356
698195409219559425 : done, 934/2356
698262614669991936 : done, 935/2356
698342080612007937 : done, 936/2356
698355670425473025 : done, 937/2356
698549713696649216 : done, 938/2356
698635131305795584 : done, 939/2356
698703483621523456 : done, 940/2356
698710712454139905 : done, 941/2356
698907974262222848 : done, 942/2356
698953797952008193 : done, 943/2356
698989035503689728 : done, 944/2356
699036661657767936 : done, 945/2356
699060279947165696 : done, 946/2356
699072405256409088 : done, 947/2356
699079609774645248 : done, 948/2356
699088579889332224 : done, 949/2356
699323444782047232 : done, 950/2356
699370870310113280 : done, 951/2356
699413908797464576 : done, 952/2356
699423671849451520 : done, 953/2356
699434518667751424 : done, 954/2356
699446877801091073 : done, 955/2356
699691744225525762 : done, 956/2356
699775878809702401 : done, 957/2356
699779630832685056 : done, 958/2356
699788877217865730 : done, 959/2356
699801817392291840 : done, 960/2356
700002074055016451 : done, 961/2356
700029284593901568 : done, 962/2356
700062718104104960 : done, 963/2356
700143752053182464 : done, 964/2356
700151421916807169 : done, 965/2356
700167517596164096 : done, 966/2356
700462010979500032 : done, 967/2356
700505138482569216 : done, 968/2356
700518061187723268 : done, 969/2356
700747788515020802 : done, 970/2356
700796979434098688 : done, 971/2356
700847567345688576 : done, 972/2356
700864154249383937 : done, 973/2356
```

```
700890391244103680 : done, 974/2356
701214700881756160 : done, 975/2356
701545186879471618 : done, 976/2356
701570477911896070 : done, 977/2356
701601587219795968 : done, 978/2356
701805642395348998 : done, 979/2356
701889187134500865 : done, 980/2356
701952816642965504 : done, 981/2356
701981390485725185 : done, 982/2356
702217446468493312 : done, 983/2356
702276748847800320 : done, 984/2356
702321140488925184 : done, 985/2356
702332542343577600 : done, 986/2356
702539513671897089 : done, 987/2356
702598099714314240 : done, 988/2356
702671118226825216 : done, 989/2356
702684942141153280 : done, 990/2356
702899151802126337 : done, 991/2356
702932127499816960 : done, 992/2356
703041949650034688 : done, 993/2356
703079050210877440 : done, 994/2356
703268521220972544 : done, 995/2356
703356393781329922 : done, 996/2356
703382836347330562 : done, 997/2356
703407252292673536 : done, 998/2356
703425003149250560 : done, 999/2356
703611486317502464 : done, 1000/2356
703631701117943808 : done, 1001/2356
703769065844768768 : done, 1002/2356
703774238772166656 : done, 1003/2356
704054845121142784 : done, 1004/2356
704113298707505153 : done, 1005/2356
704134088924532736 : done, 1006/2356
704347321748819968 : done, 1007/2356
704364645503647744 : done, 1008/2356
704480331685040129 : done, 1009/2356
704491224099647488 : done, 1010/2356
704499785726889984 : done, 1011/2356
704761120771465216 : done, 1012/2356
704819833553219584 : done, 1013/2356
704847917308362754 : done, 1014/2356
704859558691414016 : done, 1015/2356
704871453724954624 : done, 1016/2356
705066031337840642 : done, 1017/2356
705102439679201280 : done, 1018/2356
705223444686888960 : done, 1019/2356
705239209544720384 : done, 1020/2356
705428427625635840 : done, 1021/2356
```

```
705442520700944385 : done, 1022/2356
705475953783398401 : done, 1023/2356
705591895322394625 : done, 1024/2356
705786532653883392 : done, 1025/2356
705898680587526145 : done, 1026/2356
705970349788291072 : done, 1027/2356
705975130514706432 : done, 1028/2356
706153300320784384 : done, 1029/2356
706166467411222528 : done, 1030/2356
706169069255446529 : done, 1031/2356
706265994973601792 : done, 1032/2356
706291001778950144 : done, 1033/2356
706310011488698368 : done, 1034/2356
706346369204748288 : done, 1035/2356
706516534877929472 : done, 1036/2356
706538006853918722 : done, 1037/2356
706593038911545345 : done, 1038/2356
706644897839910912 : done, 1039/2356
706681918348251136 : done, 1040/2356
706901761596989440 : done, 1041/2356
706904523814649856 : done, 1042/2356
707014260413456384 : done, 1043/2356
707021089608753152 : done, 1044/2356
707038192327901184 : done, 1045/2356
707059547140169728 : done, 1046/2356
707297311098011648 : done, 1047/2356
707315916783140866 : done, 1048/2356
707377100785885184 : done, 1049/2356
707387676719185920 : done, 1050/2356
707411934438625280 : done, 1051/2356
707420581654872064 : done, 1052/2356
707610948723478529 : done, 1053/2356
707629649552134146 : done, 1054/2356
707693576495472641 : done, 1055/2356
707738799544082433 : done, 1056/2356
707741517457260545 : done, 1057/2356
707776935007539200 : done, 1058/2356
707969809498152960 : done, 1059/2356
707983188426153984 : done, 1060/2356
707995814724026368 : done, 1061/2356
708026248782585858 : done, 1062/2356
708109389455101952 : done, 1063/2356
708119489313951744 : done, 1064/2356
708130923141795840 : done, 1065/2356
708149363256774660 : done, 1066/2356
708349470027751425 : done, 1067/2356
708356463048204288 : done, 1068/2356
708400866336894977 : done, 1069/2356
```

```
708469915515297792 : done, 1070/2356
708479650088034305 : done, 1071/2356
708711088997666817 : done, 1072/2356
708738143638450176 : done, 1073/2356
708810915978854401 : done, 1074/2356
708834316713893888 : done, 1075/2356
708845821941387268 : done, 1076/2356
708853462201716736 : done, 1077/2356
709042156699303936 : done, 1078/2356
709158332880297985 : done, 1079/2356
709179584944730112 : done, 1080/2356
709198395643068416 : done, 1081/2356
709207347839836162 : done, 1082/2356
709225125749587968 : done, 1083/2356
709409458133323776 : done, 1084/2356
709449600415961088 : done, 1085/2356
709519240576036864 : done, 1086/2356
709556954897764353 : done, 1087/2356
709566166965075968 : done, 1088/2356
709852847387627521 : done, 1089/2356
709901256215666688 : done, 1090/2356
709918798883774466 : done, 1091/2356
710117014656950272 : done, 1092/2356
710140971284037632 : done, 1093/2356
710153181850935296 : done, 1094/2356
710269109699739648 : done, 1095/2356
710272297844797440 : done, 1096/2356
710283270106132480 : done, 1097/2356
710296729921429505 : done, 1098/2356
710588934686908417 : done, 1099/2356
710609963652087808 : done, 1100/2356
710658690886586372 : done, 1101/2356
710833117892898816 : done, 1102/2356
710844581445812225 : done, 1103/2356
710997087345876993 : done, 1104/2356
711008018775851008 : done, 1105/2356
711306686208872448 : done, 1106/2356
711363825979756544 : done, 1107/2356
711652651650457602 : done, 1108/2356
711694788429553666 : done, 1109/2356
711732680602345472 : done, 1110/2356
711743778164514816 : done, 1111/2356
711968124745228288 : done, 1112/2356
711998809858043904 : done, 1113/2356
712065007010385924 : done, 1114/2356
712085617388212225 : done, 1115/2356
712092745624633345 : done, 1116/2356
712097430750289920 : done, 1117/2356
```

```
712309440758808576 : done, 1118/2356
712438159032893441 : done, 1119/2356
712668654853337088 : done, 1120/2356
712717840512598017 : done, 1121/2356
712809025985978368 : done, 1122/2356
713175907180089344 : done, 1123/2356
713177543487135744 : done, 1124/2356
713411074226274305 : done, 1125/2356
713761197720473600 : done, 1126/2356
713900603437621249 : done, 1127/2356
713909862279876608 : done, 1128/2356
713919462244790272 : done, 1129/2356
714141408463036416 : done, 1130/2356
714214115368108032 : done, 1131/2356
714251586676113411 : done, 1132/2356
714258258790387713 : done, 1133/2356
714485234495041536 : done, 1134/2356
714606013974974464 : done, 1135/2356
714631576617938945 : done, 1136/2356
714957620017307648 : done, 1137/2356
714962719905021952 : done, 1138/2356
714982300363173890 : done, 1139/2356
715009755312439296 : done, 1140/2356
715200624753819648 : done, 1141/2356
715220193576927233 : done, 1142/2356
715342466308784130 : done, 1143/2356
715360349751484417 : done, 1144/2356
715680795826982913 : done, 1145/2356
715696743237730304 : done, 1146/2356
715704790270025728 : done, 1147/2356
715733265223708672 : done, 1148/2356
715758151270801409 : done, 1149/2356
715928423106027520 : done, 1150/2356
716080869887381504 : done, 1151/2356
716285507865542656 : done, 1152/2356
716439118184652801 : done, 1153/2356
716447146686459905 : done, 1154/2356
716730379797970944 : done, 1155/2356
716791146589110272 : done, 1156/2356
716802964044845056 : done, 1157/2356
717009362452090881 : done, 1158/2356
717047459982213120 : done, 1159/2356
717421804990701568 : done, 1160/2356
717428917016076293 : done, 1161/2356
717537687239008257 : done, 1162/2356
717790033953034240 : done, 1163/2356
717841801130979328 : done, 1164/2356
718234618122661888 : done, 1165/2356
```

```
718246886998687744 : done, 1166/2356
718454725339934721 : done, 1167/2356
718460005985447936 : done, 1168/2356
718540630683709445 : done, 1169/2356
718613305783398402 : done, 1170/2356
718631497683582976 : done, 1171/2356
718939241951195136 : done, 1172/2356
718971898235854848 : done, 1173/2356
719332531645071360 : done, 1174/2356
719339463458033665 : done, 1175/2356
719367763014393856 : done, 1176/2356
719551379208073216 : done, 1177/2356
719704490224398336 : done, 1178/2356
719991154352222208 : done, 1179/2356
720043174954147842 : done, 1180/2356
720059472081784833 : done, 1181/2356
720340705894408192 : done, 1182/2356
720389942216527872 : done, 1183/2356
720415127506415616 : done, 1184/2356
720775346191278080 : done, 1185/2356
720785406564900865 : done, 1186/2356
721001180231503872 : done, 1187/2356
721503162398597120 : done, 1188/2356
722613351520608256 : done, 1189/2356
722974582966214656 : done, 1190/2356
723179728551723008 : done, 1191/2356
723673163800948736 : done, 1192/2356
723688335806480385 : done, 1193/2356
723912936180330496 : done, 1194/2356
724004602748780546 : done, 1195/2356
724046343203856385 : done, 1196/2356
724049859469295616 : done, 1197/2356
724405726123311104 : done, 1198/2356
724771698126512129 : done, 1199/2356
724983749226668032 : done, 1200/2356
725458796924002305 : done, 1201/2356
725729321944506368 : done, 1202/2356
725786712245440512 : done, 1203/2356
725842289046749185 : done, 1204/2356
726224900189511680 : done, 1205/2356
726828223124897792 : done, 1206/2356
726887082820554753 : done, 1207/2356
726935089318363137 : done, 1208/2356
727155742655025152 : done, 1209/2356
727175381690781696 : done, 1210/2356
727286334147182592 : done, 1211/2356
727314416056803329 : done, 1212/2356
727524757080539137 : done, 1213/2356
```

```
727644517743104000 : done, 1214/2356
727685679342333952 : done, 1215/2356
728015554473250816 : done, 1216/2356
728035342121635841 : done, 1217/2356
728046963732717569 : done, 1218/2356
728387165835677696 : done, 1219/2356
728409960103686147 : done, 1220/2356
728653952833728512 : done, 1221/2356
728751179681943552 : done, 1222/2356
728760639972315136 : done, 1223/2356
728986383096946689 : done, 1224/2356
729113531270991872 : done, 1225/2356
729463711119904772 : done, 1226/2356
729823566028484608 : done, 1227/2356
729838605770891264 : done, 1228/2356
729854734790754305 : done, 1229/2356
730196704625098752 : done, 1230/2356
730211855403241472 : done, 1231/2356
730427201120833536 : done, 1232/2356
730573383004487680 : done, 1233/2356
730924654643314689 : done, 1234/2356
731156023742988288 : done, 1235/2356
731285275100512256 : done, 1236/2356
732005617171337216 : done, 1237/2356
732375214819057664 : done, 1238/2356
732585889486888962 : done, 1239/2356
732726085725589504 : done, 1240/2356
732732193018155009 : done, 1241/2356
733109485275860992 : done, 1242/2356
733460102733135873 : done, 1243/2356
733482008106668032 : done, 1244/2356
733822306246479872 : done, 1245/2356
733828123016450049 : done, 1246/2356
734559631394082816 : done, 1247/2356
734776360183431168 : done, 1248/2356
734787690684657664 : done, 1249/2356
734912297295085568 : done, 1250/2356
735137028879360001 : done, 1251/2356
735256018284875776 : done, 1252/2356
735274964362878976 : done, 1253/2356
735635087207878657 : done, 1254/2356
735648611367784448 : done, 1255/2356
735991953473572864 : done, 1256/2356
736010884653420544 : done, 1257/2356
736225175608430592 : done, 1258/2356
736365877722001409 : done, 1259/2356
736392552031657984 : done, 1260/2356
736736130620620800 : done, 1261/2356
```

```
737310737551491075 : done, 1262/2356
737322739594330112 : done, 1263/2356
737445876994609152 : done, 1264/2356
737678689543020544 : done, 1265/2356
737800304142471168 : done, 1266/2356
737826014890496000 : done, 1267/2356
738156290900254721 : done, 1268/2356
738166403467907072 : done, 1269/2356
738184450748633089 : done, 1270/2356
738402415918125056 : done, 1271/2356
738537504001953792 : done, 1272/2356
738883359779196928 : done, 1273/2356
738885046782832640 : done, 1274/2356
738891149612572673 : done, 1275/2356
739238157791694849 : done, 1276/2356
739485634323156992 : done, 1277/2356
739544079319588864 : done, 1278/2356
739606147276148736 : done, 1279/2356
739623569819336705 : done, 1280/2356
739844404073074688 : done, 1281/2356
739932936087216128 : done, 1282/2356
739979191639244800 : done, 1283/2356
740214038584557568 : done, 1284/2356
740359016048689152 : done, 1285/2356
740365076218183684 : done, 1286/2356
740373189193256964 : done, 1287/2356
740676976021798912 : done, 1288/2356
740699697422163968 : done, 1289/2356
740711788199743490 : done, 1290/2356
740995100998766593 : done, 1291/2356
741067306818797568 : done, 1292/2356
741099773336379392 : done, 1293/2356
741303864243200000 : done, 1294/2356
741438259667034112 : done, 1295/2356
741743634094141440 : done, 1296/2356
741793263812808706 : done, 1297/2356
742150209887731712 : done, 1298/2356
742161199639494656 : done, 1299/2356
742385895052087300 : done, 1300/2356
742423170473463808 : done, 1301/2356
742465774154047488 : done, 1302/2356
742528092657332225 : done, 1303/2356
742534281772302336 : done, 1304/2356
743210557239623680 : done, 1305/2356
743222593470234624 : done, 1306/2356
743253157753532416 : done, 1307/2356
743510151680958465 : done, 1308/2356
743545585370791937 : done, 1309/2356
```

```
743595368194129920 : done, 1310/2356
743609206067040256 : done, 1311/2356
743835915802583040 : done, 1312/2356
743895849529389061 : done, 1313/2356
743980027717509120 : done, 1314/2356
744223424764059648 : done, 1315/2356
744234799360020481 : done, 1316/2356
744334592493166593 : done, 1317/2356
744709971296780288 : done, 1318/2356
744971049620602880 : done, 1319/2356
744995568523612160 : done, 1320/2356
745057283344719872 : done, 1321/2356
745074613265149952 : done, 1322/2356
745314880350101504 : done, 1323/2356
745422732645535745 : done, 1324/2356
745433870967832576 : done, 1325/2356
745712589599014916 : done, 1326/2356
745789745784041472 : done, 1327/2356
746056683365994496 : done, 1328/2356
746131877086527488 : done, 1329/2356
746369468511756288 : done, 1330/2356
746507379341139972 : done, 1331/2356
746521445350707200 : done, 1332/2356
746542875601690625 : done, 1333/2356
746726898085036033 : done, 1334/2356
746757706116112384 : done, 1335/2356
746790600704425984 : done, 1336/2356
746818907684614144 : done, 1337/2356
746872823977771008 : done, 1338/2356
746906459439529985 : done, 1339/2356
747103485104099331 : done, 1340/2356
747204161125646336 : done, 1341/2356
747219827526344708 : done, 1342/2356
747242308580548608 : done, 1343/2356
747439450712596480 : done, 1344/2356
747461612269887489 : done, 1345/2356
747512671126323200 : done, 1346/2356
747594051852075008 : done, 1347/2356
747600769478692864 : done, 1348/2356
747648653817413632 : done, 1349/2356
747651430853525504 : done, 1350/2356
747816857231626240 : done, 1351/2356
747844099428986880 : done, 1352/2356
747885874273214464 : done, 1353/2356
747933425676525569 : done, 1354/2356
747963614829678593 : done, 1355/2356
748220828303695873 : done, 1356/2356
748307329658011649 : done, 1357/2356
```

```
748324050481647620 : done, 1358/2356
748337862848962560 : done, 1359/2356
748346686624440324 : done, 1360/2356
748568946752774144 : done, 1361/2356
748575535303884801 : done, 1362/2356
748692773788876800 : done, 1363/2356
748699167502000129 : done, 1364/2356
748705597323898880 : done, 1365/2356
748932637671223296 : done, 1366/2356
748977405889503236 : done, 1367/2356
749036806121881602 : done, 1368/2356
749064354620928000 : done, 1369/2356
749075273010798592 : done, 1370/2356
749317047558017024 : done, 1371/2356
749395845976588288 : done, 1372/2356
749403093750648834 : done, 1373/2356
749417653287129088 : done, 1374/2356
749774190421639168 : done, 1375/2356
749981277374128128 : done, 1376/2356
749996283729883136 : done, 1377/2356
750011400160841729 : done, 1378/2356
750026558547456000 : done, 1379/2356
750041628174217216 : done, 1380/2356
750056684286914561 : done, 1381/2356
750071704093859840 : done, 1382/2356
750086836815486976 : done, 1383/2356
750101899009982464 : done, 1384/2356
750117059602808832 : done, 1385/2356
750132105863102464 : done, 1386/2356
750147208377409536 : done, 1387/2356
750381685133418496 : done, 1388/2356
750383411068534784 : done, 1389/2356
750429297815552001 : done, 1390/2356
750506206503038976 : done, 1391/2356
750719632563142656 : done, 1392/2356
750868782890057730 : done, 1393/2356
751132876104687617 : done, 1394/2356
751205363882532864 : done, 1395/2356
751251247299190784 : done, 1396/2356
751456908746354688 : done, 1397/2356
751538714308972544 : done, 1398/2356
751583847268179968 : done, 1399/2356
751598357617971201 : done, 1400/2356
751793661361422336 : done, 1401/2356
751830394383790080 : done, 1402/2356
751937170840121344 : done, 1403/2356
751950017322246144 : done, 1404/2356
752173152931807232 : done, 1405/2356
```

```
752309394570878976 : done, 1406/2356
752334515931054080 : done, 1407/2356
752519690950500352 : done, 1408/2356
752568224206688256 : done, 1409/2356
752660715232722944 : done, 1410/2356
752682090207055872 : done, 1411/2356
752701944171524096 : done, 1412/2356
752917284578922496 : done, 1413/2356
752932432744185856 : done, 1414/2356
753026973505581056 : done, 1415/2356
753039830821511168 : done, 1416/2356
753294487569522689 : done, 1417/2356
753298634498793472 : done, 1418/2356
753375668877008896 : done, 1419/2356
753398408988139520 : done, 1420/2356
753420520834629632 : done, 1421/2356
753655901052166144 : done, 1422/2356
754011816964026368 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 2
754120377874386944 : done, 1423/2356
754449512966619136 : done, 1424/2356
754482103782404096 : done, 1425/2356
754747087846248448 : done, 1426/2356
754856583969079297 : done, 1427/2356
754874841593970688 : done, 1428/2356
755110668769038337 : done, 1429/2356
755206590534418437 : done, 1430/2356
755955933503782912 : done, 1431/2356
756275833623502848 : done, 1432/2356
756288534030475264 : done, 1433/2356
756303284449767430 : done, 1434/2356
756526248105566208 : done, 1435/2356
756651752796094464 : done, 1436/2356
756939218950160384 : done, 1437/2356
756998049151549440 : done, 1438/2356
757354760399941633 : done, 1439/2356
757393109802180609 : done, 1440/2356
757400162377592832 : done, 1441/2356
757596066325864448 : done, 1442/2356
757597904299253760 : done, 1443/2356
757611664640446465 : done, 1444/2356
757725642876129280 : done, 1445/2356
757729163776290825 : done, 1446/2356
757741869644341248 : done, 1447/2356
758041019896193024 : done, 1448/2356
758099635764359168 : done, 1449/2356
758355060040593408 : done, 1450/2356
758405701903519748 : done, 1451/2356
758467244762497024 : done, 1452/2356
```

```
758474966123810816 : done, 1453/2356
758740312047005698 : done, 1454/2356
758828659922702336 : done, 1455/2356
758854675097526272 : done, 1456/2356
759047813560868866 : done, 1457/2356
759099523532779520 : done, 1458/2356
759159934323924993 : done, 1459/2356
759197388317847553 : done, 1460/2356
759446261539934208 : done, 1461/2356
759447681597108224 : done, 1462/2356
759557299618865152 : done, 1463/2356
759566828574212096 : done, 1464/2356
759793422261743616 : done, 1465/2356
759846353224826880 : done, 1466/2356
759923798737051648 : done, 1467/2356
759943073749200896 : done, 1468/2356
760153949710192640 : done, 1469/2356
760190180481531904 : done, 1470/2356
760252756032651264 : done, 1471/2356
760290219849637889 : done, 1472/2356
760521673607086080 : done, 1473/2356
760539183865880579 : done, 1474/2356
760641137271070720 : done, 1475/2356
760656994973933572 : done, 1476/2356
760893934457552897 : done, 1477/2356
761004547850530816 : done, 1478/2356
761227390836215808 : done, 1479/2356
761292947749015552 : done, 1480/2356
761334018830917632 : done, 1481/2356
761371037149827077 : done, 1482/2356
761599872357261312 : done, 1483/2356
761672994376806400 : done, 1484/2356
761745352076779520 : done, 1485/2356
761750502866649088 : done, 1486/2356
761976711479193600 : done, 1487/2356
762035686371364864 : done, 1488/2356
762316489655476224 : done, 1489/2356
762464539388485633 : done, 1490/2356
762471784394268675 : done, 1491/2356
762699858130116608 : done, 1492/2356
763103485927849985 : done, 1493/2356
763167063695355904 : done, 1494/2356
763183847194451968 : done, 1495/2356
763837565564780549 : done, 1496/2356
763956972077010945 : done, 1497/2356
764259802650378240 : done, 1498/2356
764857477905154048 : done, 1499/2356
765222098633691136 : done, 1500/2356
```

```
765371061932261376 : done, 1501/2356
765395769549590528 : done, 1502/2356
765669560888528897 : done, 1503/2356
765719909049503744 : done, 1504/2356
766008592277377025 : done, 1505/2356
766069199026450432 : done, 1506/2356
766078092750233600 : done, 1507/2356
766313316352462849 : done, 1508/2356
766423258543644672 : done, 1509/2356
766693177336135680 : done, 1510/2356
766714921925144576 : done, 1511/2356
766793450729734144 : done, 1512/2356
766864461642756096 : done, 1513/2356
767122157629476866 : done, 1514/2356
767191397493538821 : done, 1515/2356
767500508068192258 : done, 1516/2356
767754930266464257 : done, 1517/2356
767884188863397888 : done, 1518/2356
768193404517830656 : done, 1519/2356
768473857036525572 : done, 1520/2356
768554158521745409 : done, 1521/2356
768596291618299904 : done, 1522/2356
768609597686943744 : done, 1523/2356
768855141948723200 : done, 1524/2356
768909767477751808 : done, 1525/2356
768970937022709760 : done, 1526/2356
769212283578875904 : done, 1527/2356
769335591808995329 : done, 1528/2356
769695466921623552 : done, 1529/2356
769940425801170949 : done, 1530/2356
770069151037685760 : done, 1531/2356
770093767776997377 : done, 1532/2356
770293558247038976 : done, 1533/2356
770414278348247044 : done, 1534/2356
770655142660169732 : done, 1535/2356
770743923962707968 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 3
770772759874076672 : done, 1536/2356
770787852854652928 : done, 1537/2356
771004394259247104 : done, 1538/2356
771014301343748096 : done, 1539/2356
771102124360998913 : done, 1540/2356
771136648247640064 : done, 1541/2356
771171053431250945 : done, 1542/2356
771380798096281600 : done, 1543/2356
771500966810099713 : done, 1544/2356
771770456517009408 : done, 1545/2356
771908950375665664 : done, 1546/2356
772102971039580160 : done, 1547/2356
```

```
772114945936949249 : done, 1548/2356
772117678702071809 : done, 1549/2356
772152991789019136 : done, 1550/2356
772193107915964416 : done, 1551/2356
772581559778025472 : done, 1552/2356
772615324260794368 : done, 1553/2356
772826264096874500 : done, 1554/2356
772877495989305348 : done, 1555/2356
773191612633579521 : done, 1556/2356
773247561583001600 : done, 1557/2356
773308824254029826 : done, 1558/2356
773336787167145985 : done, 1559/2356
773547596996571136 : done, 1560/2356
773670353721753600 : done, 1561/2356
773704687002451968 : done, 1562/2356
773922284943896577 : done, 1563/2356
773985732834758656 : done, 1564/2356
774314403806253056 : done, 1565/2356
774639387460112384 : done, 1566/2356
774757898236878852 : done, 1567/2356
775085132600442880 : done, 1568/2356
775096608509886464 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 4
775350846108426240 : done, 1569/2356
775364825476165632 : done, 1570/2356
775729183532220416 : done, 1571/2356
775733305207554048 : done, 1572/2356
775842724423557120 : done, 1573/2356
775898661951791106 : done, 1574/2356
776088319444877312 : done, 1575/2356
776113305656188928 : done, 1576/2356
776201521193218049 : done, 1577/2356
776218204058357768 : done, 1578/2356
776249906839351296 : done, 1579/2356
776477788987613185 : done, 1580/2356
776813020089548800 : done, 1581/2356
776819012571455488 : done, 1582/2356
777189768882946048 : done, 1583/2356
777621514455814149 : done, 1584/2356
777641927919427584 : done, 1585/2356
777684233540206592 : done, 1586/2356
777885040357281792 : done, 1587/2356
777953400541634568 : done, 1588/2356
778027034220126208 : done, 1589/2356
778039087836069888 : done, 1590/2356
778286810187399168 : done, 1591/2356
778383385161035776 : done, 1592/2356
778396591732486144 : done, 1593/2356
778408200802557953 : done, 1594/2356
```

```
778624900596654080 : done, 1595/2356
778650543019483137 : done, 1596/2356
778748913645780993 : done, 1597/2356
778764940568104960 : done, 1598/2356
778774459159379968 : done, 1599/2356
778990705243029504 : done, 1600/2356
779056095788752897 : done, 1601/2356
779123168116150273 : done, 1602/2356
779124354206535695 : done, 1603/2356
779377524342161408 : done, 1604/2356
779834332596887552 : done, 1605/2356
780074436359819264 : done, 1606/2356
780092040432480260 : done, 1607/2356
780192070812196864 : done, 1608/2356
780459368902959104 : done, 1609/2356
780476555013349377 : done, 1610/2356
780496263422808064 : done, 1611/2356
780543529827336192 : done, 1612/2356
780601303617732608 : done, 1613/2356
780800785462489090 : done, 1614/2356
780858289093574656 : done, 1615/2356
780931614150983680 : done, 1616/2356
781163403222056960 : done, 1617/2356
781251288990355457 : done, 1618/2356
781308096455073793 : done, 1619/2356
781524693396357120 : done, 1620/2356
781655249211752448 : done, 1621/2356
781661882474196992 : done, 1622/2356
781955203444699136 : done, 1623/2356
782021823840026624 : done, 1624/2356
782305867769217024 : done, 1625/2356
782598640137187329 : done, 1626/2356
782722598790725632 : done, 1627/2356
782747134529531904 : done, 1628/2356
782969140009107456 : done, 1629/2356
783085703974514689 : done, 1630/2356
783334639985389568 : done, 1631/2356
783347506784731136 : done, 1632/2356
783391753726550016 : done, 1633/2356
783466772167098368 : done, 1634/2356
783695101801398276 : done, 1635/2356
783821107061198850 : done, 1636/2356
783839966405230592 : done, 1637/2356
784057939640352768 : done, 1638/2356
784183165795655680 : done, 1639/2356
784431430411685888 : done, 1640/2356
784517518371221505 : done, 1641/2356
784826020293709826 : done, 1642/2356
```

```
785170936622350336 : done, 1643/2356
785264754247995392 : done, 1644/2356
785515384317313025 : done, 1645/2356
785533386513321988 : done, 1646/2356
785639753186217984 : done, 1647/2356
785872687017132033 : done, 1648/2356
785927819176054784 : done, 1649/2356
786036967502913536 : done, 1650/2356
786051337297522688 : done, 1651/2356
786233965241827333 : done, 1652/2356
786286427768250368 : done, 1653/2356
786363235746385920 : done, 1654/2356
786595970293370880 : done, 1655/2356
786664955043049472 : done, 1656/2356
786709082849828864 : done, 1657/2356
786729988674449408 : done, 1658/2356
786963064373534720 : done, 1659/2356
787111942498508800 : done, 1660/2356
787322443945877504 : done, 1661/2356
787397959788929025 : done, 1662/2356
787717603741622272 : done, 1663/2356
787810552592695296 : done, 1664/2356
788039637453406209 : done, 1665/2356
788070120937619456 : done, 1666/2356
788150585577050112 : done, 1667/2356
788178268662984705 : done, 1668/2356
788412144018661376 : done, 1669/2356
788552643979468800 : done, 1670/2356
788765914992902144 : done, 1671/2356
788908386943430656 : done, 1672/2356
789137962068021249 : done, 1673/2356
789268448748703744 : done, 1674/2356
789280767834746880 : done, 1675/2356
789314372632018944 : done, 1676/2356
789530877013393408 : done, 1677/2356
789599242079838210 : done, 1678/2356
789628658055020548 : done, 1679/2356
789903600034189313 : done, 1680/2356
789960241177853952 : done, 1681/2356
789986466051088384 : done, 1682/2356
790227638568808452 : done, 1683/2356
790277117346975746 : done, 1684/2356
790337589677002753 : done, 1685/2356
790581949425475584 : done, 1686/2356
790698755171364864 : done, 1687/2356
790723298204217344 : done, 1688/2356
790946055508652032 : done, 1689/2356
790987426131050500 : done, 1690/2356
```

```
791026214425268224 : done, 1691/2356
791312159183634433 : done, 1692/2356
791406955684368384 : done, 1693/2356
791672322847637504 : done, 1694/2356
791774931465953280 : done, 1695/2356
791780927877898241 : done, 1696/2356
791784077045166082 : done, 1697/2356
791821351946420224 : done, 1698/2356
792050063153438720 : done, 1699/2356
792394556390137856 : done, 1700/2356
792773781206999040 : done, 1701/2356
792883833364439040 : done, 1702/2356
792913359805018113 : done, 1703/2356
793120401413079041 : done, 1704/2356
793135492858580992 : done, 1705/2356
793150605191548928 : done, 1706/2356
793165685325201412 : done, 1707/2356
793180763617361921 : done, 1708/2356
793195938047070209 : done, 1709/2356
793210959003287553 : done, 1710/2356
793226087023144960 : done, 1711/2356
793241302385262592 : done, 1712/2356
793256262322548741 : done, 1713/2356
793271401113350145 : done, 1714/2356
793286476301799424 : done, 1715/2356
793500921481273345 : done, 1716/2356
793601777308463104 : done, 1717/2356
793614319594401792 : done, 1718/2356
793845145112371200 : done, 1719/2356
793962221541933056 : done, 1720/2356
794205286408003585 : done, 1721/2356
794332329137291264 : done, 1722/2356
794355576146903043 : done, 1723/2356
794926597468000259 : done, 1724/2356
794983741416415232 : done, 1725/2356
795076730285391872 : done, 1726/2356
795400264262053889 : done, 1727/2356
795464331001561088 : done, 1728/2356
796031486298386433 : done, 1729/2356
796080075804475393 : done, 1730/2356
796116448414461957 : done, 1731/2356
796125600683540480 : done, 1732/2356
796149749086875649 : done, 1733/2356
796177847564038144 : done, 1734/2356
796387464403357696 : done, 1735/2356
796484825502875648 : done, 1736/2356
796563435802726400 : done, 1737/2356
796759840936919040 : done, 1738/2356
```

```
796865951799083009 : done, 1739/2356
796904159865868288 : done, 1740/2356
797165961484890113 : done, 1741/2356
797236660651966464 : done, 1742/2356
797545162159308800 : done, 1743/2356
797971864723324932 : done, 1744/2356
798209839306514432 : done, 1745/2356
798340744599797760 : done, 1746/2356
798576900688019456 : done, 1747/2356
798585098161549313 : done, 1748/2356
798628517273620480 : done, 1749/2356
798644042770751489 : done, 1750/2356
798665375516884993 : done, 1751/2356
798673117451325440 : done, 1752/2356
798682547630837760 : done, 1753/2356
798686750113755136 : done, 1754/2356
798694562394996736 : done, 1755/2356
798697898615730177 : done, 1756/2356
798701998996647937 : done, 1757/2356
798705661114773508 : done, 1758/2356
798925684722855936 : done, 1759/2356
798933969379225600 : done, 1760/2356
799063482566066176 : done, 1761/2356
799297110730567681 : done, 1762/2356
799308762079035393 : done, 1763/2356
799422933579902976 : done, 1764/2356
799757965289017345 : done, 1765/2356
799774291445383169 : done, 1766/2356
800018252395122689 : done, 1767/2356
800141422401830912 : done, 1768/2356
800188575492947969 : done, 1769/2356
800388270626521089 : done, 1770/2356
800443802682937345 : done, 1771/2356
800459316964663297 : done, 1772/2356
800513324630806528 : done, 1773/2356
800751577355128832 : done, 1774/2356
800855607700029440 : done, 1775/2356
800859414831898624 : done, 1776/2356
801115127852503040 : done, 1777/2356
801127390143516673 : done, 1778/2356
801167903437357056 : done, 1779/2356
801285448605831168 : done, 1780/2356
801538201127157760 : done, 1781/2356
801854953262350336 : done, 1782/2356
801958328846974976 : done, 1783/2356
802185808107208704 : done, 1784/2356
802239329049477120 : done, 1785/2356
802247111496568832 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 5
```

```
802265048156610565 : done, 1786/2356
802323869084381190 : done, 1787/2356
802572683846291456 : done, 1788/2356
802600418706604034 : done, 1789/2356
802624713319034886 : done, 1790/2356
802952499103731712 : done, 1791/2356
803276597545603072 : done, 1792/2356
803321560782307329 : done, 1793/2356
803380650405482500 : done, 1794/2356
803638050916102144 : done, 1795/2356


Rate limit reached. Sleeping for: 613


803692223237865472 : done, 1796/2356
803773340896923648 : done, 1797/2356
804026241225523202 : done, 1798/2356
804413760345620481 : done, 1799/2356
804475857670639616 : done, 1800/2356
804738756058218496 : done, 1801/2356
805207613751304193 : done, 1802/2356
805487436403003392 : done, 1803/2356
805520635690676224 : done, 1804/2356
805823200554876929 : done, 1805/2356
805826884734976000 : done, 1806/2356
805932879469572096 : done, 1807/2356
805958939288408065 : done, 1808/2356
806219024703037440 : done, 1809/2356
806242860592926720 : done, 1810/2356
806542213899489280 : done, 1811/2356
806576416489959424 : done, 1812/2356
806620845233815552 : done, 1813/2356
806629075125202948 : done, 1814/2356
807010152071229440 : done, 1815/2356
807059379405148160 : done, 1816/2356
807106840509214720 : done, 1817/2356
807621403335917568 : done, 1818/2356
808001312164028416 : done, 1819/2356
808106460588765185 : done, 1820/2356
808134635716833280 : done, 1821/2356
808344865868283904 : done, 1822/2356
808501579447930884 : done, 1823/2356
808733504066486276 : done, 1824/2356
808838249661788160 : done, 1825/2356
809084759137812480 : done, 1826/2356
809220051211603969 : done, 1827/2356
809448704142938112 : done, 1828/2356
```

```
809808892968534016 : done, 1829/2356
809920764300447744 : done, 1830/2356
810254108431155201 : done, 1831/2356
810284430598270976 : done, 1832/2356
810657578271330305 : done, 1833/2356
810896069567610880 : done, 1834/2356
810984652412424192 : done, 1835/2356
811386762094317568 : done, 1836/2356
811627233043480576 : done, 1837/2356
811647686436880384 : done, 1838/2356
811744202451197953 : done, 1839/2356
811985624773361665 : done, 1840/2356
812372279581671427 : done, 1841/2356
812466873996607488 : done, 1842/2356
812503143955202048 : done, 1843/2356
812709060537683968 : done, 1844/2356
812747805718642688 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 6
812781120811126785 : done, 1845/2356
813051746834595840 : done, 1846/2356
813066809284972545 : done, 1847/2356
813081950185472002 : done, 1848/2356
813096984823349248 : done, 1849/2356
813112105746448384 : done, 1850/2356
813127251579564032 : done, 1851/2356
813130366689148928 : done, 1852/2356
813142292504645637 : done, 1853/2356
813157409116065792 : done, 1854/2356
813172488309972993 : done, 1855/2356
813187593374461952 : done, 1856/2356
813202720496779264 : done, 1857/2356
813217897535406080 : done, 1858/2356
813800681631023104 : done, 1859/2356
813812741911748608 : done, 1860/2356
813910438903693312 : done, 1861/2356
813944609378369540 : done, 1862/2356
814153002265309185 : done, 1863/2356
814530161257443328 : done, 1864/2356
814578408554463233 : done, 1865/2356
814638523311648768 : done, 1866/2356
814986499976527872 : done, 1867/2356
815390420867969024 : done, 1868/2356
815639385530101762 : done, 1869/2356
815736392542261248 : done, 1870/2356
815745968457060357 : done, 1871/2356
815966073409433600 : done, 1872/2356
815990720817401858 : done, 1873/2356
816014286006976512 : done, 1874/2356
816062466425819140 : done, 1875/2356
```

```
816091915477250048 : done, 1876/2356
816336735214911488 : done, 1877/2356
816450570814898180 : done, 1878/2356
816697700272001025 : done, 1879/2356
816816676327063552 : done, 1880/2356
816829038950027264 : done, 1881/2356
817056546584727552 : done, 1882/2356
817120970343411712 : done, 1883/2356
817171292965273600 : done, 1884/2356
817181837579653120 : done, 1885/2356
817415592588222464 : done, 1886/2356
817423860136083457 : done, 1887/2356
817502432452313088 : done, 1888/2356
817536400337801217 : done, 1889/2356
817777686764523521 : done, 1890/2356
817827839487737858 : done, 1891/2356
817908911860748288 : done, 1892/2356
818145370475810820 : done, 1893/2356
818259473185828864 : done, 1894/2356
818307523543449600 : done, 1895/2356
818536468981415936 : done, 1896/2356
818588835076603904 : done, 1897/2356
818614493328580609 : done, 1898/2356
818627210458333184 : done, 1899/2356
818646164899774465 : done, 1900/2356
819004803107983360 : done, 1901/2356
819006400881917954 : done, 1902/2356
819015331746349057 : done, 1903/2356
819015337530290176 : done, 1904/2356
819227688460238848 : done, 1905/2356
819238181065359361 : done, 1906/2356
819347104292290561 : done, 1907/2356
819588359383371776 : done, 1908/2356
819711362133872643 : done, 1909/2356
819924195358416896 : done, 1910/2356
819952236453363712 : done, 1911/2356
820013781606658049 : done, 1912/2356
820078625395449857 : done, 1913/2356
820314633777061888 : done, 1914/2356
820446719150292993 : done, 1915/2356
820494788566847489 : done, 1916/2356
820690176645140481 : done, 1917/2356
820749716845686786 : done, 1918/2356
820837357901512704 : done, 1919/2356
821044531881721856 : done, 1920/2356
821107785811234820 : done, 1921/2356
821149554670182400 : done, 1922/2356
821153421864615936 : done, 1923/2356
```

```
821407182352777218 : done, 1924/2356
821421320206483457 : done, 1925/2356
821522889702862852 : done, 1926/2356
821765923262631936 : done, 1927/2356
821813639212650496 : done, 1928/2356
821886076407029760 : done, 1929/2356
822163064745328640 : done, 1930/2356
822244816520155136 : done, 1931/2356
822462944365645825 : done, 1932/2356
822489057087389700 : done, 1933/2356
822610361945911296 : done, 1934/2356
822647212903690241 : done, 1935/2356
822859134160621569 : done, 1936/2356
822872901745569793 : done, 1937/2356
822975315408461824 : done, 1938/2356
823269594223824897 : done, 1939/2356
823322678127919110 : done, 1940/2356
823333489516937216 : done, 1941/2356
823581115634085888 : done, 1942/2356
823699002998870016 : done, 1943/2356
823719002937630720 : done, 1944/2356
823939628516474880 : done, 1945/2356
824025158776213504 : done, 1946/2356
824297048279236611 : done, 1947/2356
824325613288833024 : done, 1948/2356
824663926340194305 : done, 1949/2356
824775126675836928 : done, 1950/2356
824796380199809024 : done, 1951/2356
825026590719483904 : done, 1952/2356
825120256414846976 : done, 1953/2356
825147591692263424 : done, 1954/2356
825535076884762624 : done, 1955/2356
825829644528148480 : done, 1956/2356
825876512159186944 : done, 1957/2356
826115272272650244 : done, 1958/2356
826204788643753985 : done, 1959/2356
826240494070030336 : done, 1960/2356
826476773533745153 : done, 1961/2356
826598365270007810 : done, 1962/2356
826598799820865537 : done, 1963/2356
826615380357632002 : done, 1964/2356
826848821049180160 : done, 1965/2356
826958653328592898 : done, 1966/2356
827199976799354881 : done, 1967/2356
827228250799742977 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 7
827324948884643840 : done, 1968/2356
827600520311402496 : done, 1969/2356
827653905312006145 : done, 1970/2356
```

```
827933404142436356 : done, 1971/2356
828011680017821696 : done, 1972/2356
828046555563323392 : done, 1973/2356
828361771580813312 : done, 1974/2356
828372645993398273 : done, 1975/2356
828376505180889089 : done, 1976/2356
828381636999917570 : done, 1977/2356
828408677031882754 : done, 1978/2356
828409743546925057 : done, 1979/2356
828650029636317184 : done, 1980/2356
828708714936930305 : done, 1981/2356
828770345708580865 : done, 1982/2356
828801551087042563 : done, 1983/2356
829011960981237760 : done, 1984/2356
829141528400556032 : done, 1985/2356
829374341691346946 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 8
829449946868879360 : done, 1986/2356
829501995190984704 : done, 1987/2356
829861396166877184 : done, 1988/2356
829878982036299777 : done, 1989/2356
830097400375152640 : done, 1990/2356
830173239259324417 : done, 1991/2356
830583320585068544 : done, 1992/2356
830956169170665475 : done, 1993/2356
831262627380748289 : done, 1994/2356
831309418084069378 : done, 1995/2356
831315979191906304 : done, 1996/2356
831322785565769729 : done, 1997/2356
831552930092285952 : done, 1998/2356
831650051525054464 : done, 1999/2356
831670449226514432 : done, 2000/2356
831911600680497154 : done, 2001/2356
831926988323639298 : done, 2002/2356
831939777352105988 : done, 2003/2356
832032802820481025 : done, 2004/2356
832040443403784192 : done, 2005/2356
832088576586297345 : done, 2006/2356
832215726631055365 : done, 2007/2356
832215909146226688 : done, 2008/2356
832273440279240704 : done, 2009/2356
832369877331693569 : done, 2010/2356
832397543355072512 : done, 2011/2356
832636094638288896 : done, 2012/2356
832645525019123713 : done, 2013/2356
832682457690300417 : done, 2014/2356
832757312314028032 : done, 2015/2356
832769181346996225 : done, 2016/2356
832998151111966721 : done, 2017/2356
```

```
833124694597443584 : done, 2018/2356
833479644947025920 : done, 2019/2356
833722901757046785 : done, 2020/2356
833732339549220864 : done, 2021/2356
833826103416520705 : done, 2022/2356
833863086058651648 : done, 2023/2356
834086379323871233 : done, 2024/2356
834089966724603904 : done, 2025/2356
834167344700198914 : done, 2026/2356
834209720923721728 : done, 2027/2356
834458053273591808 : done, 2028/2356
834477809192075265 : done, 2029/2356
834574053763584002 : done, 2030/2356
834786237630337024 : done, 2031/2356
834931633769889797 : done, 2032/2356
835152434251116546 : done, 2033/2356
835172783151792128 : done, 2034/2356
835246439529840640 : done, 2035/2356
835264098648616962 : done, 2036/2356
835297930240217089 : done, 2037/2356
835309094223372289 : done, 2038/2356
835536468978302976 : done, 2039/2356
835574547218894849 : done, 2040/2356
835685285446955009 : done, 2041/2356
836001077879255040 : done, 2042/2356
836260088725786625 : done, 2043/2356
836380477523124226 : done, 2044/2356
836397794269200385 : done, 2045/2356
836648853927522308 : done, 2046/2356
836677758902222849 : done, 2047/2356
836753516572119041 : done, 2048/2356
836989968035819520 : done, 2049/2356
837012587749474308 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 9
837110210464448512 : done, 2050/2356
837366284874571778 : done, 2051/2356
837471256429613056 : done, 2052/2356
837482249356513284 : done, 2053/2356
837820167694528512 : done, 2054/2356
838083903487373313 : done, 2055/2356
838085839343206401 : done, 2056/2356
838150277551247360 : done, 2057/2356
838201503651401729 : done, 2058/2356
838476387338051585 : done, 2059/2356
838561493054533637 : done, 2060/2356
838831947270979586 : done, 2061/2356
838916489579200512 : done, 2062/2356
838921590096166913 : done, 2063/2356
838952994649550848 : done, 2064/2356
```

```
839239871831150596 : done, 2065/2356
839290600511926273 : done, 2066/2356
839549326359670784 : done, 2067/2356
839990271299457024 : done, 2068/2356
840268004936019968 : done, 2069/2356
840370681858686976 : done, 2070/2356
840632337062862849 : done, 2071/2356
840696689258311684 : done, 2072/2356
840698636975636481 : done, 2073/2356
840728873075638272 : done, 2074/2356
840761248237133825 : done, 2075/2356
841077006473256960 : done, 2076/2356
841314665196081154 : done, 2077/2356
841320156043304961 : done, 2078/2356
841439858740625411 : done, 2079/2356
841680585030541313 : done, 2080/2356
841833993020538882 : done, 2081/2356
842115215311396866 : done, 2082/2356
842163532590374912 : done, 2083/2356
842535590457499648 : done, 2084/2356
842765311967449089 : done, 2085/2356
842846295480000512 : done, 2086/2356
842892208864923648 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 10
843235543001513987 : done, 2087/2356
843604394117681152 : done, 2088/2356
843856843873095681 : done, 2089/2356
843981021012017153 : done, 2090/2356
844223788422217728 : done, 2091/2356
844580511645339650 : done, 2092/2356
844704788403113984 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 11
844973813909606400 : done, 2093/2356
844979544864018432 : done, 2094/2356
845098359547420673 : done, 2095/2356
845306882940190720 : done, 2096/2356
845397057150107648 : done, 2097/2356
845459076796616705 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 12
845677943972139009 : done, 2098/2356
845812042753855489 : done, 2099/2356
846042936437604353 : done, 2100/2356
846139713627017216 : done, 2101/2356
846153765933735936 : done, 2102/2356
846505985330044928 : done, 2103/2356
846514051647705089 : done, 2104/2356
846874817362120707 : done, 2105/2356
847116187444137987 : done, 2106/2356
847157206088847362 : done, 2107/2356
847251039262605312 : done, 2108/2356
847606175596138505 : done, 2109/2356
```

```
847617282490613760 : done, 2110/2356
847842811428974592 : done, 2111/2356
847962785489326080 : done, 2112/2356
847971574464610304 : done, 2113/2356
847978865427394560 : done, 2114/2356
848212111729840128 : done, 2115/2356
848213670039564288 : done, 2116/2356
848324959059550208 : done, 2117/2356
848690551926992896 : done, 2118/2356
849051919805034497 : done, 2119/2356
849336543269576704 : done, 2120/2356
849412302885593088 : done, 2121/2356
849668094696017920 : done, 2122/2356
849776966551130114 : done, 2123/2356
850019790995546112 : done, 2124/2356
850145622816686080 : done, 2125/2356
850333567704068097 : done, 2126/2356
850380195714523136 : done, 2127/2356
850753642995093505 : done, 2128/2356
851224888060895234 : done, 2129/2356
851464819735769094 : done, 2130/2356
851591660324737024 : done, 2131/2356
851861385021730816 : done, 2132/2356
851953902622658560 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 13
852189679701164033 : done, 2133/2356
852226086759018497 : done, 2134/2356
852311364735569921 : done, 2135/2356
852553447878664193 : done, 2136/2356
852672615818899456 : done, 2137/2356
852912242202992640 : done, 2138/2356
852936405516943360 : done, 2139/2356
853299958564483072 : done, 2140/2356
853639147608842240 : done, 2141/2356
853760880890318849 : done, 2142/2356
854010172552949760 : done, 2143/2356
854120357044912130 : done, 2144/2356
854365224396361728 : done, 2145/2356
854482394044301312 : done, 2146/2356
854732716440526848 : done, 2147/2356
855138241867124737 : done, 2148/2356
855245323840757760 : done, 2149/2356
855459453768019968 : done, 2150/2356
855818117272018944 : done, 2151/2356
855851453814013952 : done, 2152/2356
855857698524602368 : done, 2153/2356
855860136149123072 : done, 2154/2356
855862651834028034 : done, 2155/2356
856282028240666624 : done, 2156/2356
```

```
856288084350160898 : done, 2157/2356
856330835276025856 : done, 2158/2356
856526610513747968 : done, 2159/2356
856543823941562368 : done, 2160/2356
856602993587888130 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 14
857029823797047296 : done, 2161/2356
857062103051644929 : done, 2162/2356
857214891891077121 : done, 2163/2356
857263160327368704 : done, 2164/2356
857393404942143489 : done, 2165/2356
857746408056729600 : done, 2166/2356
857989990357356544 : done, 2167/2356
858107933456039936 : done, 2168/2356
858471635011153920 : done, 2169/2356
858843525470990336 : done, 2170/2356
858860390427611136 : done, 2171/2356
859074603037188101 : done, 2172/2356
859196978902773760 : done, 2173/2356
859607811541651456 : done, 2174/2356
859851578198683649 : done, 2175/2356
859924526012018688 : done, 2176/2356
860177593139703809 : done, 2177/2356
860184849394610176 : done, 2178/2356
860276583193509888 : done, 2179/2356
860524505164394496 : done, 2180/2356
860563773140209665 : done, 2181/2356
860924035999428608 : done, 2182/2356
860981674716409858 : done, 2183/2356
861005113778896900 : done, 2184/2356
861288531465048066 : done, 2185/2356
861383897657036800 : done, 2186/2356
861769973181624320 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 15
862096992088072192 : done, 2187/2356
862457590147678208 : done, 2188/2356
862722525377298433 : done, 2189/2356
862831371563274240 : done, 2190/2356
863062471531167744 : done, 2191/2356
863079547188785154 : done, 2192/2356
863427515083354112 : done, 2193/2356
863432100342583297 : done, 2194/2356
863471782782697472 : done, 2195/2356
863553081350529029 : done, 2196/2356
863907417377173506 : done, 2197/2356
864197398364647424 : done, 2198/2356
864279568663928832 : done, 2199/2356
864873206498414592 : done, 2200/2356
865006731092295680 : done, 2201/2356
865359393868664832 : done, 2202/2356
```

```
865718153858494464 : done, 2203/2356
866094527597207552 : done, 2204/2356
866334964761202691 : done, 2205/2356
866450705531457537 : done, 2206/2356
866686824827068416 : done, 2207/2356
866720684873056260 : done, 2208/2356
866816280283807744 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 16
867051520902168576 : done, 2209/2356
867072653475098625 : done, 2210/2356
867421006826221569 : done, 2211/2356
867774946302451713 : done, 2212/2356
867900495410671616 : done, 2213/2356
868552278524837888 : done, 2214/2356
868622495443632128 : done, 2215/2356
868639477480148993 : done, 2216/2356
868880397819494401 : done, 2217/2356
869227993411051520 : done, 2218/2356
869596645499047938 : done, 2219/2356
869702957897576449 : done, 2220/2356
869772420881756160 : done, 2221/2356
869988702071779329 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 17
870063196459192321 : done, 2222/2356
870308999962521604 : done, 2223/2356
870374049280663552 : done, 2224/2356
870656317836468226 : done, 2225/2356
870726314365509632 : done, 2226/2356
870804317367881728 : done, 2227/2356
871032628920680449 : done, 2228/2356
871102520638267392 : done, 2229/2356
871166179821445120 : done, 2230/2356
871515927908634625 : done, 2231/2356
871762521631449091 : done, 2232/2356
871879754684805121 : done, 2233/2356
872122724285648897 : done, 2234/2356
872261713294495745 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 18
872486979161796608 : done, 2235/2356
872620804844003328 : done, 2236/2356
872668790621863937 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 19
872820683541237760 : done, 2237/2356
872967104147763200 : done, 2238/2356
873213775632977920 : done, 2239/2356
873337748698140672 : done, 2240/2356
873580283840344065 : done, 2241/2356
873697596434513921 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 20
874012996292530176 : done, 2242/2356
874057562936811520 : done, 2243/2356
874296783580663808 : done, 2244/2356
874434818259525634 : done, 2245/2356
```

```
874680097055178752 : done, 2246/2356
875021211251597312 : done, 2247/2356
875097192612077568 : done, 2248/2356
875144289856114688 : done, 2249/2356
875747767867523072 : done, 2250/2356
876120275196170240 : done, 2251/2356
876484053909872640 : done, 2252/2356
876537666061221889 : done, 2253/2356
876838120628539392 : done, 2254/2356
877201837425926144 : done, 2255/2356
877316821321428993 : done, 2256/2356
877556246731214848 : done, 2257/2356
877611172832227328 : done, 2258/2356
877736472329191424 : done, 2259/2356
878057613040115712 : done, 2260/2356
878281511006478336 : done, 2261/2356
878316110768087041 : done, 2262/2356
878404777348136964 : done, 2263/2356
878604707211726852 : done, 2264/2356
878776093423087618 : done, 2265/2356
879008229531029506 : done, 2266/2356
879050749262655488 : done, 2267/2356
879130579576475649 : done, 2268/2356
879376492567855104 : done, 2269/2356
879415818425184262 : done, 2270/2356
879492040517615616 : done, 2271/2356
879674319642796034 : done, 2272/2356
879862464715927552 : done, 2273/2356
880095782870896641 : done, 2274/2356
880221127280381952 : done, 2275/2356
880465832366813184 : done, 2276/2356
880872448815771648 : done, 2277/2356
880935762899988482 : done, 2278/2356
881268444196462592 : done, 2279/2356
881536004380872706 : done, 2280/2356
881633300179243008 : done, 2281/2356
881666595344535552 : done, 2282/2356
881906580714921986 : done, 2283/2356
882045870035918850 : done, 2284/2356
882268110199369728 : done, 2285/2356
882627270321602560 : done, 2286/2356
882762694511734784 : done, 2287/2356
882992080364220416 : done, 2288/2356
883117836046086144 : done, 2289/2356
883360690899218434 : done, 2290/2356
883482846933004288 : done, 2291/2356
883838122936631299 : done, 2292/2356
884162670584377345 : done, 2293/2356
```

```
884247878851493888 : done, 2294/2356
884441805382717440 : done, 2295/2356
884562892145688576 : done, 2296/2356
884876753390489601 : done, 2297/2356
884925521741709313 : done, 2298/2356
885167619883638784 : done, 2299/2356
885311592912609280 : done, 2300/2356
885518971528720385 : done, 2301/2356
885528943205470208 : done, 2302/2356
885984800019947520 : done, 2303/2356
886054160059072513 : done, 2304/2356
886258384151887873 : done, 2305/2356
886267009285017600 : done, 2306/2356
886366144734445568 : done, 2307/2356
886680336477933568 : done, 2308/2356
886736880519319552 : done, 2309/2356
886983233522544640 : done, 2310/2356
887101392804085760 : done, 2311/2356
887343217045368832 : done, 2312/2356
887473957103951883 : done, 2313/2356
887517139158093824 : done, 2314/2356
887705289381826560 : done, 2315/2356
888078434458587136 : done, 2316/2356
888202515573088257 : [{'code': 144, 'message': 'No status found with that ID.'}], total fail= 21
888554962724278272 : done, 2317/2356
888804989199671297 : done, 2318/2356
888917238123831296 : done, 2319/2356
889278841981685760 : done, 2320/2356
889531135344209921 : done, 2321/2356
889638837579907072 : done, 2322/2356
889665388333682689 : done, 2323/2356
889880896479866881 : done, 2324/2356
890006608113172480 : done, 2325/2356
890240255349198849 : done, 2326/2356
890609185150312448 : done, 2327/2356
890729181411237888 : done, 2328/2356
890971913173991426 : done, 2329/2356
891087950875897856 : done, 2330/2356
891327558926688256 : done, 2331/2356
891689557279858688 : done, 2332/2356
891815181378084864 : done, 2333/2356
892177421306343426 : done, 2334/2356
892420643555336193 : done, 2335/2356
The time we need to get JSON file: 1989.8182586169999 second
```

```
In [11]: print("Success to get {} data, and fail to get {} data (no_tweet: {}, just fail: {}), f
               .format(num_succes_get_data, num_fail_get_data,\
```

```
                    len(ids_not_found_tweet), len(ids_fail_get_tweet),\
                    num_tweet_id))
```

```
Success to get 2335 data, and fail to get 21 data (no_tweet: 21, just fail: 0), from total 2356
```

```
In [12]: json.dump(tweets,open('data_generated/tweets.txt', 'w', encoding="utf8"), ensure_ascii=
         print('Success save the json file')
```

```
Success save the json file
```

```
In [13]: # read json file into dataframe
         with open('data_generated/tweets.txt','r') as f:
             data = json.load(f)

         scrapped_tweet_df = pd.DataFrame(data)
         scrapped_tweet_df.head(2)
```

Out[13]:

	favorite_count	favorited	retweet_count	retweeted	tweet_id
0	2498	False	493	False	666020888022790149
1	124	False	46	False	666029285002620928

Conclusion: - We get the third data - 21 data from tweet_id are failed to get from tweet API because the id is not found, the twitter must be deleted - We get first data from file that we save in same folder, second data from Udacity's server, and third data from Twitter API - After see tweet_json.txt from Udacity, I decide to get some column (not all column) because another column has been save in first data, and some cols not need yet (like column user) - Because twitter have range limit time, so we need extra time (because of sleep) to get all data. In this project we need 1989.8 second

Accessing The Data

For now, we have 3 data: twitter_archive_df, image_prediction_df, and scrapped_tweet_df . Todo in accessing data:

Check length of data
Check the type of data
Check the value of data
Check missing value of data
Check stat describe data
Founded Issues

```
In [14]: import matplotlib.pyplot as plt
         import seaborn as sns
         import numpy as np

         %matplotlib inline
```

1. Check length of data

```
In [15]: def print_length(name, data_frame):
             print("The length of {} is {}".format(name, len(data_frame)))
```

```
In [16]: print_length('twitter_archive_df', twitter_archive_df)
         print_length('image_prediction_df', image_prediction_df)
         print_length('scrapped_tweet_df', scrapped_tweet_df)
```

```
The length of twitter_archive_df is 2356
The length of image_prediction_df is 2075
The length of scrapped_tweet_df is 2335
```

From that data we get info that twitter_archive_df has different length with scrapped_tweet_df because we failed to get 22 data from twitter. We can delete some row in data so we will have the same length in each table.

2. Check the type of data

```
In [17]: twitter_archive_df.dtypes
```

```
Out[17]: tweet_id                        int64
         in_reply_to_status_id         float64
         in_reply_to_user_id           float64
         timestamp                      object
         source                         object
         text                           object
         retweeted_status_id           float64
         retweeted_status_user_id      float64
         retweeted_status_timestamp     object
         expanded_urls                  object
         rating_numerator                int64
         rating_denominator              int64
         name                           object
         doggo                          object
         floofer                        object
         pupper                         object
         puppo                          object
         dtype: object
```

```
In [18]: image_prediction_df.dtypes
```

```
Out[18]: tweet_id      int64
         jpg_url      object
         img_num       int64
         p1           object
         p1_conf     float64
         p1_dog         bool
         p2           object
         p2_conf     float64
         p2_dog         bool
         p3           object
         p3_conf     float64
         p3_dog         bool
         dtype: object
```

In [19]: scrapped_tweet_df.dtypes

```
Out[19]: favorite_count    int64
         favorited          bool
         retweet_count     int64
         retweeted          bool
         tweet_id          int64
         dtype: object
```

Object in the data type mean string, we not have some problem in there except timestamp. It must be date

3. Check the value of data

In [20]: twitter_archive_df.name.value_counts().head()

```
Out[20]: None       745
         a           55
         Charlie     12
         Oliver      11
         Cooper      11
         Name: name, dtype: int64
```

There is 5 sorted dog name with the biggest total value. We find that "None" is typically missing data, and I assumed that "a" also a missing data, so we must find and uniformly all missing data value in each label.

In [21]: twitter_archive_df.retweeted_status_id.value_counts().head()

```
Out[21]: 7.757333e+17    1
         8.001414e+17    1
         8.174239e+17    1
         6.742918e+17    1
         6.833919e+17    1
         Name: retweeted_status_id, dtype: int64
```

We only need original tweet (not retweeted by another tweet), so we must drop row that retweeted_status_id doesn't missing

In [22]: scrapped_tweet_df.retweeted.value_counts()

```
Out[22]: False    2335
         Name: retweeted, dtype: int64
```

Retweeted indicates whether this Tweet has been Retweeted by the authenticating user, because all value are false so this column be not informatif anymore.

In [23]: twitter_archive_df.duplicated(['tweet_id']).sum()

```
Out[23]: 0
```

```
In [24]: twitter_archive_df.duplicated(['expanded_urls']).sum()

Out[24]: 137

In [25]: twitter_archive_df[twitter_archive_df.duplicated(['expanded_urls'])].head(3)

Out[25]:                 tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         2189  668967877119254528           6.689207e+17         2.143566e+07
         2149  669684865554620416           6.693544e+17         4.196984e+09
         2038  671550332464455680           6.715449e+17         4.196984e+09

                               timestamp  \
         2189  2015-11-24 01:42:25 +0000
         2149  2015-11-26 01:11:28 +0000
         2038  2015-12-01 04:44:10 +0000

                                                           source  \
         2189  <a href="http://twitter.com/download/iphone" r...
         2149  <a href="http://twitter.com/download/iphone" r...
         2038  <a href="http://twitter.com/download/iphone" r...

                                                             text  retweeted_status_id  \
         2189                      12/10 good shit Bubka\n@wane15                  NaN
         2149  After countless hours of research and hundreds...                  NaN
         2038  After 22 minutes of careful deliberation this ...                  NaN

               retweeted_status_user_id retweeted_status_timestamp expanded_urls  \
         2189                      NaN                        NaN           NaN
         2149                      NaN                        NaN           NaN
         2038                      NaN                        NaN           NaN

               rating_numerator  rating_denominator  name doggo floofer pupper puppo
         2189                12                  10  None  None    None   None  None
         2149                11                  10  None  None    None   None  None
         2038                 1                  10  None  None    None   None  None

In [26]: twitter_archive_df[twitter_archive_df.duplicated(['expanded_urls'])].expanded_urls.valu

Out[26]: https://twitter.com/dog_rates/status/739238157791694849/video/1
         https://twitter.com/dog_rates/status/820749716845686786/photo/1,https://twitter.com/dog
         https://twitter.com/dog_rates/status/750719632563142656/photo/1
         Name: expanded_urls, dtype: int64

In [27]: twitter_archive_df.query("expanded_urls == 'https://twitter.com/dog_rates/status/767754

Out[27]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         837  767754930266464257                    NaN                  NaN
         558  803321560782307329                    NaN                  NaN
```

```
                     timestamp  \
837  2016-08-22 16:06:54 +0000
558  2016-11-28 19:35:59 +0000

                                                source  \
837  <a href="http://twitter.com/download/iphone" r...
558  <a href="http://twitter.com/download/iphone" r...

                                                  text  retweeted_status_id  \
837  This is Philbert. His toilet broke and he does...                  NaN
558  RT @dog_rates: This is Philbert. His toilet br...         7.677549e+17

     retweeted_status_user_id retweeted_status_timestamp  \
837                       NaN                        NaN
558              4.196984e+09  2016-08-22 16:06:54 +0000

                                         expanded_urls  rating_numerator  \
837  https://twitter.com/dog_rates/status/767754930...                11
558  https://twitter.com/dog_rates/status/767754930...                11

     rating_denominator      name doggo floofer pupper puppo
837                  10  Philbert  None    None   None  None
558                  10  Philbert  None    None   None  None
```

There are some images that duplicated, we must re-check are they are have same value in each cols (except the id, because we don't have any duplicate tweet id)

```
In [28]: scrapped_tweet_df.favorited.value_counts()
```

```
Out[28]: False    2335
         Name: favorited, dtype: int64
```

retweeted and favorited data only have 1 value, so it is not important anymore, we must to drop it.

```
In [29]: twitter_archive_df.source.value_counts()
```

```
Out[29]: <a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>
         <a href="http://vine.co" rel="nofollow">Vine - Make a Scene</a>
         <a href="http://twitter.com" rel="nofollow">Twitter Web Client</a>
         <a href="https://about.twitter.com/products/tweetdeck" rel="nofollow">TweetDeck</a>
         Name: source, dtype: int64
```

To make the data more clear, we need to change source cols value
4. Check missing value of data
list function name: get_missing_value_percentage

```
In [30]: def get_missing_value_percentage(data_frame):
             data_missing = data_frame.isna()
```

```
        num_data_missing = data_missing.sum()
        num_data = len(data_frame)
        return (num_data_missing * 100)/num_data
```

```
In [31]: get_missing_value_percentage(twitter_archive_df)
```

```
Out[31]: tweet_id                       0.000000
         in_reply_to_status_id         96.689304
         in_reply_to_user_id           96.689304
         timestamp                      0.000000
         source                         0.000000
         text                           0.000000
         retweeted_status_id           92.317487
         retweeted_status_user_id      92.317487
         retweeted_status_timestamp    92.317487
         expanded_urls                  2.504244
         rating_numerator               0.000000
         rating_denominator             0.000000
         name                           0.000000
         doggo                          0.000000
         floofer                        0.000000
         pupper                         0.000000
         puppo                          0.000000
         dtype: float64
```

```
In [32]: get_missing_value_percentage(image_prediction_df)
```

```
Out[32]: tweet_id    0.0
         jpg_url     0.0
         img_num     0.0
         p1          0.0
         p1_conf     0.0
         p1_dog      0.0
         p2          0.0
         p2_conf     0.0
         p2_dog      0.0
         p3          0.0
         p3_conf     0.0
         p3_dog      0.0
         dtype: float64
```

```
In [33]: get_missing_value_percentage(scrapped_tweet_df)
```

```
Out[33]: favorite_count    0.0
         favorited         0.0
         retweet_count     0.0
         retweeted         0.0
         tweet_id          0.0
         dtype: float64
```

Data twitter_archive_df have some missing value in variable in_reply_to_status_id (96.69%), in_reply_to_user_id (96.69%), retweeted_status_id (92.32%), retweeted_status_user_id (92.32%), retweeted_status_timestamp (92.32%), and expanded_urls (2.50%). Because of the large missing value (>90%), 5 cols in twitter_archive_df must be deleted. For expanded_urls, must be check after join with other table. Data image_prediction_df didn't have any missing value, the scrapped_tweet_df also didn't have missing value.

5. Check stat describe data

In [34]:
```
twitter_archive_df.describe()
```

Out[34]:
```
           tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
count  2.356000e+03           7.800000e+01         7.800000e+01
mean   7.427716e+17           7.455079e+17         2.014171e+16
std    6.856705e+16           7.582492e+16         1.252797e+17
min    6.660209e+17           6.658147e+17         1.185634e+07
25%    6.783989e+17           6.757419e+17         3.086374e+08
50%    7.196279e+17           7.038708e+17         4.196984e+09
75%    7.993373e+17           8.257804e+17         4.196984e+09
max    8.924206e+17           8.862664e+17         8.405479e+17

       retweeted_status_id  retweeted_status_user_id  rating_numerator  \
count         1.810000e+02              1.810000e+02       2356.000000
mean          7.720400e+17              1.241698e+16         13.126486
std           6.236928e+16              9.599254e+16         45.876648
min           6.661041e+17              7.832140e+05          0.000000
25%           7.186315e+17              4.196984e+09         10.000000
50%           7.804657e+17              4.196984e+09         11.000000
75%           8.203146e+17              4.196984e+09         12.000000
max           8.874740e+17              7.874618e+17       1776.000000

       rating_denominator
count         2356.000000
mean            10.455433
std              6.745237
min              0.000000
25%             10.000000
50%             10.000000
75%             10.000000
max            170.000000
```

In [35]:
```
image_prediction_df.describe()
```

Out[35]:
```
           tweet_id      img_num      p1_conf       p2_conf       p3_conf
count  2.075000e+03  2075.000000  2075.000000  2.075000e+03  2.075000e+03
mean   7.384514e+17     1.203855     0.594548  1.345886e-01  6.032417e-02
std    6.785203e+16     0.561875     0.271174  1.006657e-01  5.090593e-02
min    6.660209e+17     1.000000     0.044333  1.011300e-08  1.740170e-10
25%    6.764835e+17     1.000000     0.364412  5.388625e-02  1.622240e-02
50%    7.119988e+17     1.000000     0.588230  1.181810e-01  4.944380e-02
```

```
75%    7.932034e+17     1.000000     0.843855  1.955655e-01  9.180755e-02
max    8.924206e+17     4.000000     1.000000  4.880140e-01  2.734190e-01
```

In [36]:
```
scrapped_tweet_df.describe()
```

Out[36]:

	favorite_count	retweet_count	tweet_id
count	2335.000000	2335.000000	2.335000e+03
mean	7810.989293	2859.129336	7.419847e+17
std	12111.582433	4837.919767	6.820978e+16
min	0.000000	1.000000	6.660209e+17
25%	1359.500000	575.500000	6.783065e+17
50%	3402.000000	1338.000000	7.184600e+17
75%	9578.000000	3337.000000	7.986692e+17
max	161719.000000	82136.000000	8.924206e+17

tweed_id musn't describe as numeric variable that we can conclude the statistic description, it is more suitable as a string

Founded Issues:

quality issues: 1. Axist not original tweet 2. tweet_id format in third data doesn't like first data so maybe it can make some problem if we join the two table 3. tweet_id position in third table not same like the other table, so we can't easily see the id 4. timestamp in first table not in datetime format 5. Missing value was not uniformly, sometime NaN but some other None 6. There are exist columns that have >90% missing value, also exist dog name that just have 1 character ('a') 7. Cols retwitted and favorited have same value in all row 8. Cols source have html format 9. Cols expanded_urls and jpg_urls have duplicated value

tidiness issues: 1. Stage of dog must be 1 cols instead of 4 cols 2. Join all data is needed to make easier for analysis

Cleaning and Tidying The Data

In cleaning and tydinf data, we want to make sure that issues we founded before will not exist. Todo in cleaning data:

Cleaning: Delete not original tweet
Cleaning: Change Tweet id format in each table
Cleaning: Change tweet_id position into first col
Cleaning: Change timestamp format
Cleaning: Uniformly missing value
Tidying: Make dog stages into 1 column
Cleaning: Delete duplicated row from expanded and jpg urls
Cleaning: Delete col with missing value >90% from total rows
Cleaning: Delete cols with same value
Cleaning: Get source col without HTML format
Tidying: Join all table

1. Delete not original tweet

In [37]:
```
twitter_archive_df = twitter_archive_df[pd.isna(twitter_archive_df.retweeted_status_id)
pd.notna(twitter_archive_df['retweeted_status_id']).sum()
```

Out[37]: 0

In [38]:
```
pd.notna(twitter_archive_df.retweeted_status_timestamp).sum()
```

Out[38]: 0

In [39]:
```
pd.notna(twitter_archive_df.retweeted_status_user_id).sum()
```

Out[39]: 0

Now we only have the original tweet
2. Change Tweet id format in each table
list function name: convert_to_str

In [40]:
```
def convert_to_str(cols):
    return cols.astype(str).infer_objects()
```

In [41]:
```
twitter_archive_df.tweet_id = convert_to_str(twitter_archive_df.tweet_id)
twitter_archive_df.head(1)
```

Out[41]:
```
                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
2355  666020888022790149                    NaN                  NaN   

                      timestamp  \
2355  2015-11-15 22:32:08 +0000   

                                                 source  \
2355  <a href="http://twitter.com/download/iphone" r...   

                                                   text  retweeted_status_id  \
2355  Here we have a Japanese Irish Setter. Lost eye...                  NaN   

      retweeted_status_user_id retweeted_status_timestamp  \
2355                       NaN                        NaN   

                                          expanded_urls  rating_numerator  \
2355  https://twitter.com/dog_rates/status/666020888...                 8   

      rating_denominator  name doggo floofer pupper puppo  
2355                  10  None  None    None   None  None  
```

In [42]:
```
image_prediction_df.tweet_id = convert_to_str(image_prediction_df.tweet_id)
image_prediction_df.head(1)
```

Out[42]:
```
             tweet_id                                          jpg_url  \
0  666020888022790149  https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg   

   img_num                      p1   p1_conf  p1_dog      p2   p2_conf  \
0        1  Welsh_springer_spaniel  0.465074    True  collie  0.156665   

   p2_dog                 p3   p3_conf  p3_dog  
0    True  Shetland_sheepdog  0.061428    True  
```

In [43]:
```
scrapped_tweet_df.tweet_id = convert_to_str(scrapped_tweet_df.tweet_id)
scrapped_tweet_df.head(1)
```

Out[43]:

	favorite_count	favorited	retweet_count	retweeted	tweet_id
0	2498	False	493	False	666020888022790149

3. Change tweet_id position into first col

In [44]:
```
scrapped_tweet_df = scrapped_tweet_df.reindex(\
                    ['tweet_id','favorite_count','favorited','retweet_count','retwe
                    axis=1)
scrapped_tweet_df.head(1)
```

Out[44]:

	tweet_id	favorite_count	favorited	retweet_count	retweeted
0	666020888022790149	2498	False	493	False

4. Change timestamp format

In [45]:
```
twitter_archive_df.timestamp = pd.to_datetime(twitter_archive_df.timestamp)
twitter_archive_df.timestamp.head(1)
```

Out[45]:
```
2355   2015-11-15 22:32:08+00:00
Name: timestamp, dtype: datetime64[ns, UTC]
```

In [46]:
```
twitter_archive_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 2355 to 0
Data columns (total 17 columns):
tweet_id                      2175 non-null object
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2175 non-null datetime64[ns, UTC]
source                        2175 non-null object
text                          2175 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null object
expanded_urls                 2117 non-null object
rating_numerator              2175 non-null int64
rating_denominator            2175 non-null int64
name                          2175 non-null object
doggo                         2175 non-null object
floofer                       2175 non-null object
pupper                        2175 non-null object
puppo                         2175 non-null object
dtypes: datetime64[ns, UTC](1), float64(4), int64(2), object(10)
memory usage: 305.9+ KB
```

5. Uniformly missing value
list function name: uniformly_missing_value

```
In [47]: def uniformly_missing_value(data_frame):
             missing_value_names = ['NaN','None', 'N/A', 'NA', 'Unknown']
             for column in data_frame.columns:
                 for phrase in missing_value_names:
                     data_frame[column].replace(to_replace=missing_value_names, value=np.nan, in
             return data_frame
```

I assume that Dog Name 'a' is null value, so I will make it as NULL

```
In [48]: twitter_archive_df.loc[twitter_archive_df.query("name == 'a'").index,'name'] = np.nan
         twitter_archive_df.name.value_counts().head(5)
```

```
Out[48]: None       680
         Lucy        11
         Charlie     11
         Cooper      10
         Oliver      10
         Name: name, dtype: int64
```

To other variables, I uniformly that missing value with function uniformly_missing_value() that was declare before.

```
In [49]: twitter_archive_df = uniformly_missing_value(twitter_archive_df)
         twitter_archive_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2175 entries, 2355 to 0
Data columns (total 17 columns):
tweet_id                      2175 non-null object
in_reply_to_status_id         78 non-null float64
in_reply_to_user_id           78 non-null float64
timestamp                     2175 non-null datetime64[ns, UTC]
source                        2175 non-null object
text                          2175 non-null object
retweeted_status_id           0 non-null float64
retweeted_status_user_id      0 non-null float64
retweeted_status_timestamp    0 non-null float64
expanded_urls                 2117 non-null object
rating_numerator              2175 non-null int64
rating_denominator            2175 non-null int64
name                          1440 non-null object
doggo                         87 non-null object
floofer                       10 non-null object
pupper                        234 non-null object
puppo                         25 non-null object
dtypes: datetime64[ns, UTC](1), float64(5), int64(2), object(9)
memory usage: 385.9+ KB
```

from information above, we find that cols doggo, flooger, ... , puppo have a lot of missing value, but that data is untidy, it must be 1 column.

```
In [50]: image_prediction_df = uniformly_missing_value(image_prediction_df)
         image_prediction_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null object
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(1), object(5)
memory usage: 152.1+ KB
```

```
In [51]: scrapped_tweet_df = uniformly_missing_value(scrapped_tweet_df)
         scrapped_tweet_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2335 entries, 0 to 2334
Data columns (total 5 columns):
tweet_id          2335 non-null object
favorite_count    2335 non-null int64
favorited         2335 non-null bool
retweet_count     2335 non-null int64
retweeted         2335 non-null bool
dtypes: bool(2), int64(2), object(1)
memory usage: 59.4+ KB
```

6. Make dog stages into 1 column
Todo:
Validation check there is 1 single value for 1 row
Add new cols to save dog stages
Change value into dog stages
Remove cols not needed
1. Validation there is 1 single value for 1 row
list function name: is_not_nan

```
In [52]: def is_not_nan(data_frame, index:int, col:str):
             cell = data_frame.iloc[index,data_frame.columns.get_loc(col)]
             return pd.notna(cell)
```

```
In [53]: twitter_archive_df['validation'] = 0
         num_rows = len(twitter_archive_df)

         for i in range(num_rows):
             validation_value = twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('va

             twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('validation')] = \
                 validation_value + \
                 is_not_nan(twitter_archive_df, i, 'doggo') + \
                 is_not_nan(twitter_archive_df, i, 'floofer') + \
                 is_not_nan(twitter_archive_df, i, 'pupper') + \
                 is_not_nan(twitter_archive_df, i, 'puppo')

         twitter_archive_df['validation'].value_counts()
```

```
Out[53]: 0    1831
         1     332
         2      12
         Name: validation, dtype: int64
```

from the value_counts above we find they are 12 row not vallid because they have 2 type of dog. Let's see the data

```
In [54]: twitter_archive_df.query("validation > 1 ").head(2)
```

```
Out[54]:                tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         1113  733109485275860992                    NaN                  NaN
         1063  741067306818797568                    NaN                  NaN

                              timestamp  \
         1113 2016-05-19 01:38:16+00:00
         1063 2016-06-10 00:39:48+00:00


                                                          source  \
         1113  <a href="http://twitter.com/download/iphone" r...
         1063  <a href="http://twitter.com/download/iphone" r...


                                                            text  retweeted_status_id  \
         1113  Like father (doggo), like son (pupper). Both 1...                  NaN
         1063  This is just downright precious af. 12/10 for ...                  NaN

               retweeted_status_user_id  retweeted_status_timestamp  \
         1113                       NaN                         NaN
         1063                       NaN                         NaN
```

```
                                        expanded_urls  rating_numerator  \
1113  https://twitter.com/dog_rates/status/733109485...                12
1063  https://twitter.com/dog_rates/status/741067306...                12

      rating_denominator  name  doggo floofer  pupper puppo  validation
1113                  10   NaN  doggo     NaN  pupper   NaN           2
1063                  10  just  doggo     NaN  pupper   NaN           2
```

I don't know what the right stage,and the duplicate count just 14 row (0.5% from total row) so I decide to delete unvalid stage

```
In [55]: twitter_archive_df = twitter_archive_df[twitter_archive_df.validation <= 1]
         twitter_archive_df['validation'].value_counts()
```

```
Out[55]: 0    1831
         1     332
         Name: validation, dtype: int64
```

2. Add new colomn to save dog stage

```
In [56]: twitter_archive_df['dog_stage'] = np.nan
         twitter_archive_df.head(2)
```

```
Out[56]:                 tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         2355  666020888022790149                    NaN                  NaN
         2354  666029285002620928                    NaN                  NaN

                              timestamp  \
         2355 2015-11-15 22:32:08+00:00
         2354 2015-11-15 23:05:30+00:00

                                                          source  \
         2355  &lt;a href="http://twitter.com/download/iphone" r...
         2354  &lt;a href="http://twitter.com/download/iphone" r...

                                                            text  retweeted_status_id  \
         2355  Here we have a Japanese Irish Setter. Lost eye...                  NaN
         2354  This is a western brown Mitsubishi terrier. Up...                  NaN

               retweeted_status_user_id  retweeted_status_timestamp  \
         2355                       NaN                        NaN
         2354                       NaN                        NaN

                                               expanded_urls  rating_numerator  \
         2355  https://twitter.com/dog_rates/status/666020888...                 8
         2354  https://twitter.com/dog_rates/status/666029285...                 7

               rating_denominator name doggo floofer pupper puppo  validation  \
         2355                  10  NaN   NaN     NaN    NaN   NaN           0
```

```
2354                  10  NaN   NaN     NaN    NaN   NaN           0

      dog_stage
2355        NaN
2354        NaN
```

3. Change value column dog_stage
list used function: is_not_nan

In [57]:
```
num_rows = len(twitter_archive_df)

for i in range(num_rows):
    result = twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('dog_stage')]

    if(is_not_nan(twitter_archive_df, i, 'doggo')):
        result = twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('doggo')]
    elif(is_not_nan(twitter_archive_df, i, 'floofer')):
        result = twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('floofer'
    elif(is_not_nan(twitter_archive_df, i, 'pupper')):
        result = twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('pupper')
    elif(is_not_nan(twitter_archive_df, i, 'puppo')):
        result = twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('puppo')]

    twitter_archive_df.iloc[i,twitter_archive_df.columns.get_loc('dog_stage')] = result

twitter_archive_df.dog_stage.value_counts()
```

Out[57]:
```
pupper     224
doggo       75
puppo       24
floofer      9
Name: dog_stage, dtype: int64
```

4. Remove cols not needed

In [58]:
```
twitter_archive_df.columns
```

Out[58]:
```
Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
       'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
       'retweeted_status_timestamp', 'expanded_urls', 'rating_numerator',
       'rating_denominator', 'name', 'doggo', 'floofer', 'pupper', 'puppo',
       'validation', 'dog_stage'],
      dtype='object')
```

Because we add column dog_stage so we don't need columns: 'doggo', 'floofer', 'pupper', 'puppo', and we also don't need column validation.

In [59]:
```
twitter_archive_df.drop(['doggo', 'floofer', 'pupper', 'puppo','validation'], axis=1, i
twitter_archive_df.columns
```

```
Out[59]: Index(['tweet_id', 'in_reply_to_status_id', 'in_reply_to_user_id', 'timestamp',
               'source', 'text', 'retweeted_status_id', 'retweeted_status_user_id',
               'retweeted_status_timestamp', 'expanded_urls', 'rating_numerator',
               'rating_denominator', 'name', 'dog_stage'],
              dtype='object')
```

```
In [60]: get_missing_value_percentage(twitter_archive_df)
```

```
Out[60]: tweet_id                        0.000000
         in_reply_to_status_id          96.440129
         in_reply_to_user_id            96.440129
         timestamp                       0.000000
         source                          0.000000
         text                            0.000000
         retweeted_status_id           100.000000
         retweeted_status_user_id      100.000000
         retweeted_status_timestamp    100.000000
         expanded_urls                   2.681461
         rating_numerator                0.000000
         rating_denominator              0.000000
         name                           33.656958
         dog_stage                      84.650948
         dtype: float64
```

The missing value from dog_stage quietly high, but I think this variable save such as good information. So I decide to not delete it.
7. Ensure unique twitter by expanded_urls and jpg_urls

1. Expanded URLS

```
In [61]: twitter_archive_df.duplicated(['expanded_urls']).value_counts()
```

```
Out[61]: False    2105
         True       58
         dtype: int64
```

```
In [62]: twitter_archive_df[twitter_archive_df.duplicated(['expanded_urls'])].expanded_urls.valu
```

```
Out[62]: https://vine.co/v/ea0OwvPTx9l    1
         Name: expanded_urls, dtype: int64
```

```
In [63]: twitter_archive_df.query('expanded_urls == "https://vine.co/v/ea0OwvPTx9l"')
```

```
Out[63]:                 tweet_id  in_reply_to_status_id  in_reply_to_user_id  \
         2212  668587383441514497                    NaN                  NaN
         657   791774931465953280                    NaN                  NaN

                               timestamp  \
         2212 2015-11-23 00:30:28+00:00
```

```
657   2016-10-27 22:53:48+00:00

                                                    source  \
2212  <a href="http://vine.co" rel="nofollow">Vine -...
657   <a href="http://vine.co" rel="nofollow">Vine -...

                                                      text  retweeted_status_id  \
2212  Never forget this vine. You will not stop watc...                  NaN
657   Vine will be deeply missed. This was by far my...                  NaN

      retweeted_status_user_id  retweeted_status_timestamp  \
2212                       NaN                         NaN
657                        NaN                         NaN

                     expanded_urls  rating_numerator  rating_denominator  \
2212  https://vine.co/v/ea0OwvPTx9l                13                  10
657   https://vine.co/v/ea0OwvPTx9l                14                  10

     name dog_stage
2212  the       NaN
657   NaN       NaN
```

I don't know what the real value from that photo, and because of that value just appear in small row (2) so I decide to delete it. And for other duplicate row I also delete it because the expanded_urls value is missing value.

```
In [64]: twitter_archive_df.dropna(subset=['expanded_urls'], how='all', inplace = True)
         twitter_archive_df.duplicated(['expanded_urls']).value_counts()
```

```
Out[64]: False    2104
         True        1
         dtype: int64
```

```
In [65]: twitter_archive_df.drop_duplicates(subset=['expanded_urls'], keep=False, inplace = True
         twitter_archive_df.duplicated(['expanded_urls']).value_counts()
```

```
Out[65]: False    2103
         dtype: int64
```

2. JPG URLS

```
In [66]: image_prediction_df.duplicated(['jpg_url']).value_counts()
```

```
Out[66]: False    2009
         True       66
         dtype: int64
```

```
In [67]: image_prediction_df[image_prediction_df.duplicated(['jpg_url'])].jpg_url.head(5)
```

```
Out[67]: 1297    https://pbs.twimg.com/ext_tw_video_thumb/67535...
         1315      https://pbs.twimg.com/media/CWza7kpWcAAdYLc.jpg
         1333      https://pbs.twimg.com/media/CWyD2HGUYAQ1Xa7.jpg
         1345      https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg
         1349      https://pbs.twimg.com/media/CkNjahBXAAQ2kWo.jpg
         Name: jpg_url, dtype: object
```

```
In [68]: image_prediction_df.query("jpg_url == 'https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg'
```

```
Out[68]:                tweet_id                                          jpg_url  \
         224   670319130621435904  https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg
         1345  759159934323924993  https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg

               img_num             p1   p1_conf  p1_dog      p2   p2_conf  p2_dog  \
         224         1  Irish_terrier  0.254856    True  briard  0.227716    True
         1345        1  Irish_terrier  0.254856    True  briard  0.227716    True

                                        p3   p3_conf  p3_dog
         224   soft-coated_wheaten_terrier  0.223263    True
         1345  soft-coated_wheaten_terrier  0.223263    True
```

```
In [69]: twitter_archive_df.query("tweet_id == '670319130621435904'").expanded_urls
```

```
Out[69]: 2127    https://twitter.com/dog_rates/status/670319130...
         Name: expanded_urls, dtype: object
```

```
In [70]: twitter_archive_df.query("tweet_id == '759159934323924993'").expanded_urls
```

```
Out[70]: Series([], Name: expanded_urls, dtype: object)
```

From observasion above, I find that one of twitter id from duplicated jpg url, doesn't axist in first table. So I will elimited twitter_id that doesn't exist in first table.

```
In [71]: jpg_url_duplicated = image_prediction_df[image_prediction_df.duplicated(['jpg_url'])]['
```

```
In [72]: need_to_drop = image_prediction_df[image_prediction_df.jpg_url.isin(jpg_url_duplicated)
         need_to_drop.head(2)
```

```
Out[72]:               tweet_id                                          jpg_url  \
         85   667509364010450944  https://pbs.twimg.com/media/CUN4Or5UAAAa5K4.jpg
         224  670319130621435904  https://pbs.twimg.com/media/CU1zsMSUAAAS0qW.jpg

              img_num             p1   p1_conf  p1_dog                  p2   p2_conf  \
         85         1         beagle  0.636169    True  Labrador_retriever  0.119256
         224        1  Irish_terrier  0.254856    True              briard  0.227716

              p2_dog                           p3   p3_conf  p3_dog
         85     True             golden_retriever  0.082549    True
         224    True  soft-coated_wheaten_terrier  0.223263    True
```

In [73]:
```
# count row we must to keep
need_to_drop['tweet_id'].isin(twitter_archive_df.tweet_id).sum()
```

Out[73]: 64

In [74]:
```
# drop row need_to_drop if the id exist in first table
need_to_drop = need_to_drop[~need_to_drop['tweet_id'].isin(twitter_archive_df.tweet_id)
need_to_drop['tweet_id'].isin(twitter_archive_df.tweet_id).sum()
```

Out[74]: 0

In [75]:
```
# drop row not exist in first table
image_prediction_df = image_prediction_df[~image_prediction_df.tweet_id.isin(need_to_dr
image_prediction_df.duplicated(['jpg_url']).value_counts()
```

Out[75]:
```
False    2007
dtype: int64
```

8. Delete col with missing value >90% from total rows
list function name: drop_missing_value
list used function: drop_missing_value

In [76]:
```
def drop_missing_value(data_frame, treshold:int = 0.9):
    data = get_missing_value_percentage(data_frame)
    cols_will_drop = []

    for col,percentage_missing_value in data.items():
        if percentage_missing_value > 90:
            cols_will_drop.append(col)

    return data_frame.drop(cols_will_drop, axis = 1)
```

In [77]:
```
twitter_archive_df = drop_missing_value(twitter_archive_df)
get_missing_value_percentage(twitter_archive_df)
```

Out[77]:
```
tweet_id               0.000000
timestamp              0.000000
source                 0.000000
text                   0.000000
expanded_urls          0.000000
rating_numerator       0.000000
rating_denominator     0.000000
name                  31.811698
dog_stage             84.498336
dtype: float64
```

In [78]:
```
drop_missing_value(image_prediction_df)
get_missing_value_percentage(image_prediction_df)
```

Out[78]:
```
tweet_id    0.0
jpg_url     0.0
img_num     0.0
p1          0.0
p1_conf     0.0
p1_dog      0.0
p2          0.0
p2_conf     0.0
p2_dog      0.0
p3          0.0
p3_conf     0.0
p3_dog      0.0
dtype: float64
```

In [79]:
```
drop_missing_value(scrapped_tweet_df)
get_missing_value_percentage(scrapped_tweet_df)
```

Out[79]:
```
tweet_id          0.0
favorite_count    0.0
favorited         0.0
retweet_count     0.0
retweeted         0.0
dtype: float64
```

9. Delete cols with same value
list function name: drop_uniform_value

In [80]:
```
def drop_uniform_value(data_frame):
    cols = data_frame.columns
    cols_will_drop = []

    for col in cols:
        num_value = len(data_frame[col].unique().tolist())
        if(num_value <= 1):
            cols_will_drop.append(col)
    return data_frame.drop(cols_will_drop, axis = 1)
```

In [81]:
```
# first data before
twitter_archive_df.nunique()
```

Out[81]:
```
tweet_id              2103
timestamp             2103
source                   4
text                  2103
expanded_urls         2103
rating_numerator        34
rating_denominator      15
name                   951
dog_stage                4
dtype: int64
```

In [82]:
```
# second data before
image_prediction_df.nunique()
```

Out[82]:
```
tweet_id    2007
jpg_url     2007
img_num        4
p1           378
p1_conf     2004
p1_dog         2
p2           405
p2_conf     2002
p2_dog         2
p3           408
p3_conf     2004
p3_dog         2
dtype: int64
```

In [83]:
```
# third data before
scrapped_tweet_df.nunique()
```

Out[83]:
```
tweet_id          2335
favorite_count    1975
favorited            1
retweet_count     1698
retweeted            1
dtype: int64
```

Because the the only table exist 1 unique value is third data so we change only third data.

In [84]:
```
# third data after
scrapped_tweet_df = drop_uniform_value(scrapped_tweet_df)
scrapped_tweet_df.nunique()
```

Out[84]:
```
tweet_id          2335
favorite_count    1975
retweet_count     1698
dtype: int64
```

In [85]:
```
twitter_archive_df.rating_denominator.value_counts()
```

Out[85]:
```
10     2085
50        3
11        2
80        2
7         1
170       1
150       1
130       1
120       1
```

```
110        1
90         1
70         1
40         1
20         1
2          1
Name: rating_denominator, dtype: int64
```

For rating denominator I expect constanta value (10) but because I get information from Udacity that there is unique rating system is a big part of the popularity of WeRateDogs, so I decide to keep them.

10. Get source col without HTML format

```
In [86]: import re
```

list function name: get_name_in_source

```
In [87]: def get_name_in_source(col_source):
             return str(re.findall("<a.*?>(.+?)</a>", col_source)[0])
```

```
In [88]: for index in range(len(twitter_archive_df)):
             value = twitter_archive_df.iloc[index,twitter_archive_df.columns.get_loc('source')]
             twitter_archive_df.iloc[index,twitter_archive_df.columns.get_loc('source')] = get_n

         twitter_archive_df.source.value_counts()

Out[88]: Twitter for iPhone     1974
         Vine - Make a Scene      88
         Twitter Web Client       30
         TweetDeck                11
         Name: source, dtype: int64
```

11. Join all table

```
In [89]: # join first and second table
         twitter_df = pd.merge(twitter_archive_df, scrapped_tweet_df, how = 'inner', on = ['twee

         # join second and third table
         twitter_df = pd.merge(twitter_df, image_prediction_df, how = 'inner', on = ['tweet_id']

         # check the result
         twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1978 entries, 0 to 1977
Data columns (total 22 columns):
tweet_id              1978 non-null object
timestamp             1978 non-null datetime64[ns, UTC]
source                1978 non-null object
```

```
text                   1978 non-null object
expanded_urls          1978 non-null object
rating_numerator       1978 non-null int64
rating_denominator     1978 non-null int64
name                   1383 non-null object
dog_stage              294 non-null object
favorite_count         1978 non-null int64
retweet_count          1978 non-null int64
jpg_url                1978 non-null object
img_num                1978 non-null int64
p1                     1978 non-null object
p1_conf                1978 non-null float64
p1_dog                 1978 non-null bool
p2                     1978 non-null object
p2_conf                1978 non-null float64
p2_dog                 1978 non-null bool
p3                     1978 non-null object
p3_conf                1978 non-null float64
p3_dog                 1978 non-null bool
dtypes: bool(3), datetime64[ns, UTC](1), float64(3), int64(5), object(10)
memory usage: 314.9+ KB
```

list used function: drop_missing_value

In [90]:
```
# cek the percentage of missing value
get_missing_value_percentage(twitter_df)
```

Out[90]:
```
tweet_id                 0.000000
timestamp                0.000000
source                   0.000000
text                     0.000000
expanded_urls            0.000000
rating_numerator         0.000000
rating_denominator       0.000000
name                    30.080890
dog_stage               85.136502
favorite_count           0.000000
retweet_count            0.000000
jpg_url                  0.000000
img_num                  0.000000
p1                       0.000000
p1_conf                  0.000000
p1_dog                   0.000000
p2                       0.000000
p2_conf                  0.000000
p2_dog                   0.000000
p3                       0.000000
```

```
p3_conf                0.000000
p3_dog                 0.000000
dtype: float64
```

In [91]:
```python
twitter_df.head(2)
```

Out[91]:
```
             tweet_id                  timestamp              source  \
0  666020888022790149 2015-11-15 22:32:08+00:00  Twitter for iPhone
1  666029285002620928 2015-11-15 23:05:30+00:00  Twitter for iPhone

                                            text  \
0  Here we have a Japanese Irish Setter. Lost eye...
1  This is a western brown Mitsubishi terrier. Up...

                                   expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/666020888...                 8
1  https://twitter.com/dog_rates/status/666029285...                 7

   rating_denominator name dog_stage  favorite_count  ...  img_num  \
0                  10  NaN       NaN            2498  ...        1
1                  10  NaN       NaN             124  ...        1

                       p1   p1_conf p1_dog                  p2   p2_conf  \
0  Welsh_springer_spaniel  0.465074   True              collie  0.156665
1                 redbone  0.506826   True  miniature_pinscher  0.074192

  p2_dog                   p3   p3_conf p3_dog
0   True     Shetland_sheepdog  0.061428   True
1   True   Rhodesian_ridgeback  0.072010   True

[2 rows x 22 columns]
```

In [92]:
```python
# save csv
twitter_df.to_csv("data_generated/twitter_archive_master.csv", index=False)
```

In [93]:
```python
# save each of data
twitter_archive_df.to_csv("data_generated/first_data_twitter_archive.csv", index=False)
image_prediction_df.to_csv("data_generated/second_data_image_prediction.csv", index=Fal
scrapped_tweet_df.to_csv("data_generated/third_data_scrapped_tweet.csv", index=False)
```

Analyzing and Visualizing Data
Question:
Are there any outlier in the data?
How about correlation between variables?
Does the retweet count and favorite count increase with time?
Does the rating increase with time?
Are the rating affect with the number of favorite and retweet count?
How much each algorithm predict the picture is dog?
What are the most popular dog names?

What are the most popular dog predict?
What are the most popular dog predict when all algorithm predict the same dog?
This plot for answer number 1:

In [94]:
```
# to analyzing, I add new column "rating" that can be calculate by numerator/denominato
twitter_df['rating'] = pd.to_numeric((twitter_df.rating_numerator*1.0)/(twitter_df.rati
twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1978 entries, 0 to 1977
Data columns (total 23 columns):
tweet_id              1978 non-null object
timestamp             1978 non-null datetime64[ns, UTC]
source                1978 non-null object
text                  1978 non-null object
expanded_urls         1978 non-null object
rating_numerator      1978 non-null int64
rating_denominator    1978 non-null int64
name                  1383 non-null object
dog_stage             294 non-null object
favorite_count        1978 non-null int64
retweet_count         1978 non-null int64
jpg_url               1978 non-null object
img_num               1978 non-null int64
p1                    1978 non-null object
p1_conf               1978 non-null float64
p1_dog                1978 non-null bool
p2                    1978 non-null object
p2_conf               1978 non-null float64
p2_dog                1978 non-null bool
p3                    1978 non-null object
p3_conf               1978 non-null float64
p3_dog                1978 non-null bool
rating                1978 non-null float64
dtypes: bool(3), datetime64[ns, UTC](1), float64(4), int64(5), object(10)
memory usage: 330.3+ KB
```

In [95]:
```
# statistic description
twitter_df.describe()
```

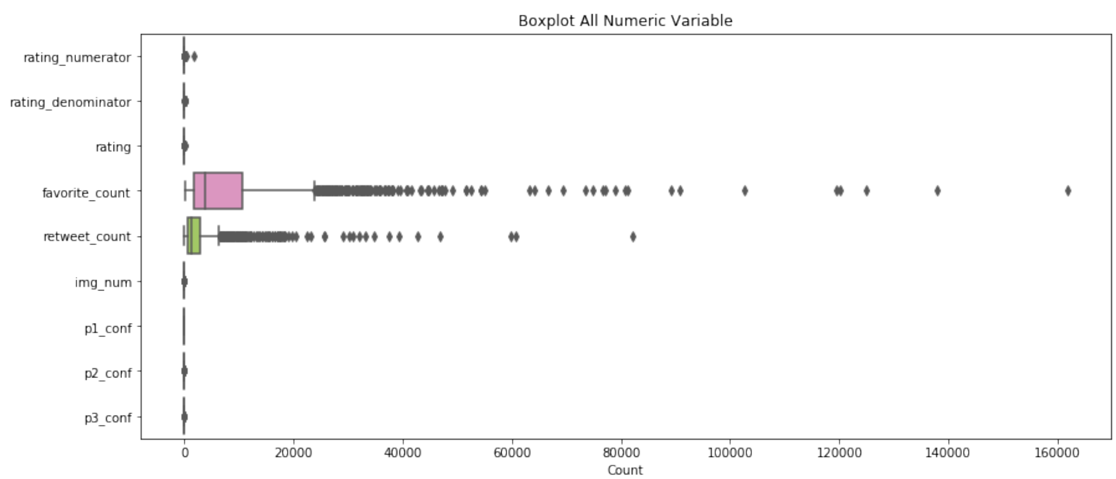
Out[95]:

	rating_numerator	rating_denominator	favorite_count	retweet_count	\
count	1978.000000	1978.000000	1978.000000	1978.000000	
mean	12.287159	10.536400	8511.986855	2576.833670	
std	41.664877	7.350117	12539.485491	4622.355184	
min	0.000000	2.000000	76.000000	11.000000	
25%	10.000000	10.000000	1832.750000	577.250000	
50%	11.000000	10.000000	3811.000000	1241.000000	
75%	12.000000	10.000000	10645.500000	2934.000000	

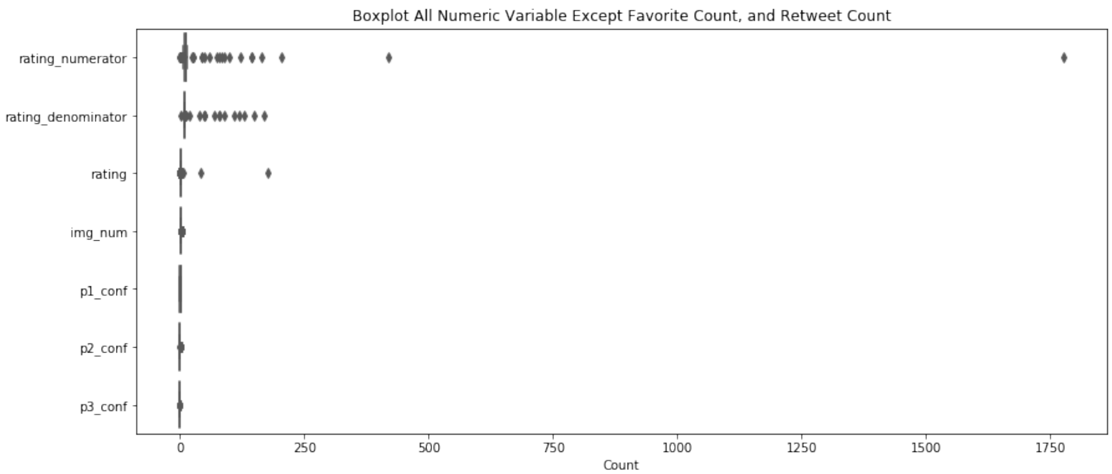
```
max        1776.000000          170.000000   161719.000000   82136.000000

           img_num      p1_conf       p2_conf       p3_conf        rating
count  1978.000000  1978.000000  1.978000e+03  1.978000e+03  1978.000000
mean      1.203741     0.592434  1.347591e-01  6.043538e-02     1.169405
std       0.562211     0.271780  1.006778e-01  5.090927e-02     4.083458
min       1.000000     0.044333  1.011300e-08  1.740170e-10     0.000000
25%       1.000000     0.360998  5.432547e-02  1.638385e-02     1.000000
50%       1.000000     0.586944  1.178485e-01  4.975535e-02     1.100000
75%       1.000000     0.841932  1.953582e-01  9.166433e-02     1.200000
max       4.000000     1.000000  4.880140e-01  2.734190e-01   177.600000
```

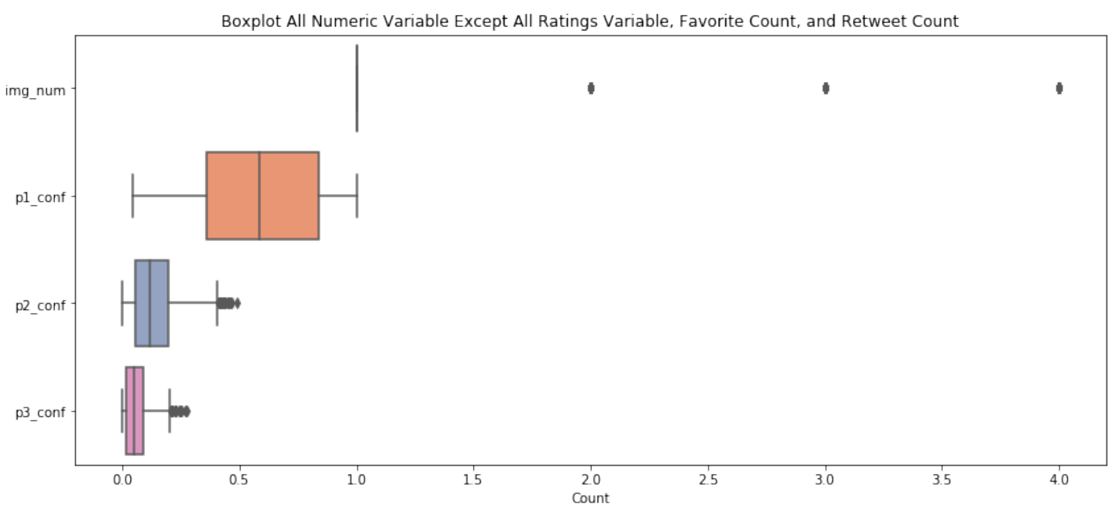
In [96]:
```
# visualitation
fig, ax = plt.subplots()
fig.set_size_inches(14, 6)
sns.boxplot(data=twitter_df[['rating_numerator','rating_denominator','rating','favorite
                                  'retweet_count','img_num','p1_conf','p2_conf','p3_con
                 orient="h", palette="Set2", ax=ax);
plt.title('Boxplot All Numeric Variable');
plt.xlabel('Count');
plt.savefig('plot/1_Boxplot All Numeric Variable.png')
```



In [97]:
```
fig, ax = plt.subplots()
fig.set_size_inches(14, 6)
sns.boxplot(data=twitter_df[['rating_numerator','rating_denominator','rating',\
                                  'img_num','p1_conf','p2_conf','p3_conf']],\
                 orient="h", palette="Set2", ax=ax);
plt.title('Boxplot All Numeric Variable Except Favorite Count, and Retweet Count');
plt.xlabel('Count');
plt.savefig('plot/2_Boxplot All Numeric Variable Except Favorite Count, and Retweet Cou
```

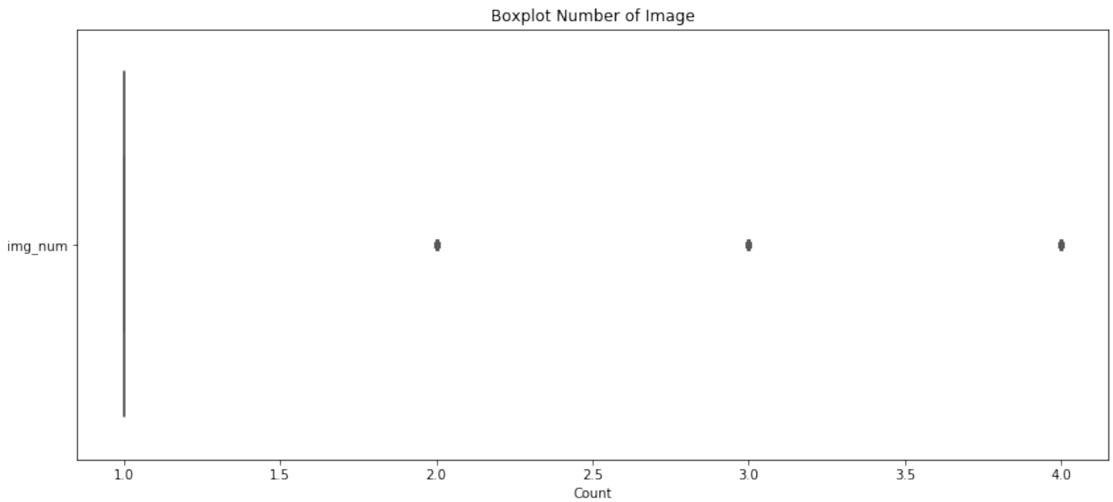



```
In [98]: fig, ax = plt.subplots()
         fig.set_size_inches(14, 6)
         sns.boxplot(data=twitter_df[['img_num','p1_conf','p2_conf','p3_conf']],\
                           orient="h", palette="Set2", ax=ax);
         plt.title('Boxplot All Numeric Variable Except All Ratings Variable, Favorite Count, an
         plt.xlabel('Count');
         plt.savefig('plot/3_Boxplot All Numeric Variable Except All Ratings Variable, Favorite
```




```
In [99]: fig, ax = plt.subplots()
         fig.set_size_inches(14, 6)
         sns.boxplot(data=twitter_df[['img_num']],\
                           orient="h", palette="Set2", ax=ax);
         plt.title('Boxplot Number of Image');
```

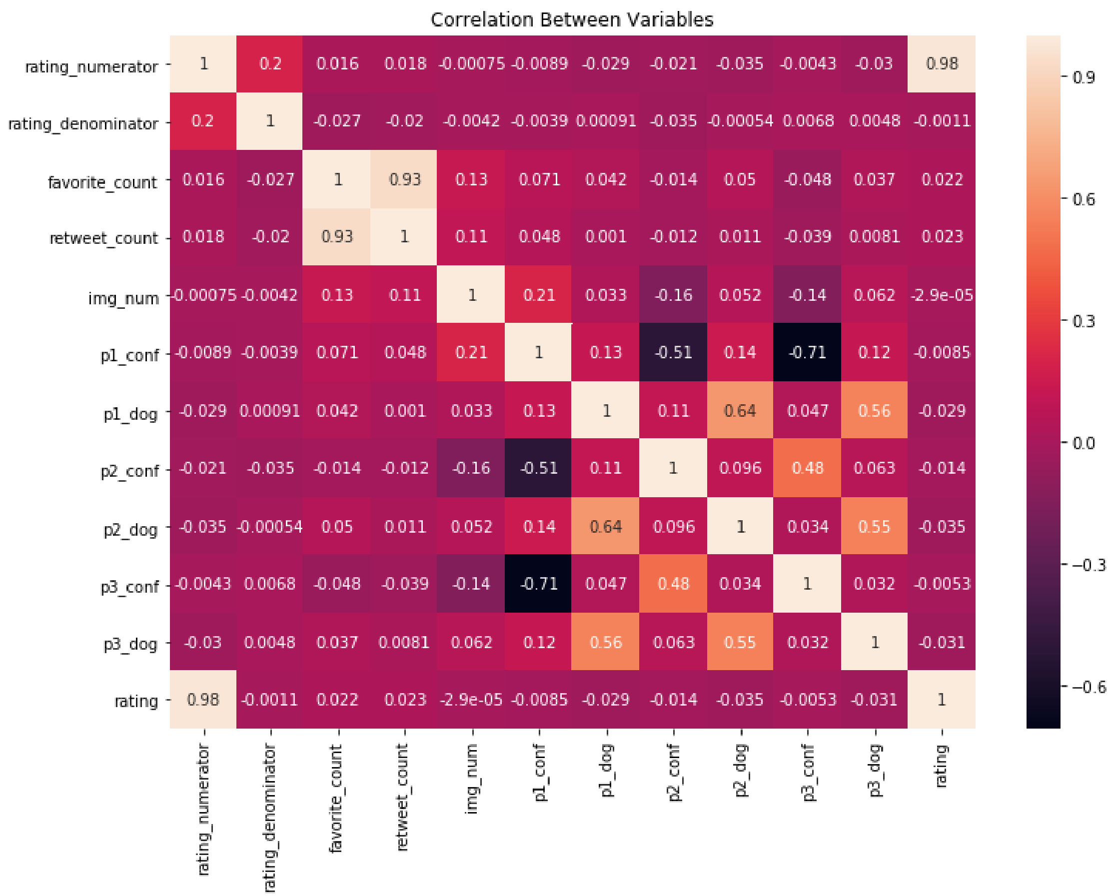
```
plt.xlabel('Count');
plt.savefig('plot/4_Boxplot Number of Image.png')
```



1. Are there any outlier in the data? Answer: - In numeric data, all cols have outlier except p1_conf and rating_denominator. Just like the information from udacity, some they use unique rating so the result maybe not between 0 until 1. - From statistic description, we found that distance min max from variables rating_numerator, rating_denominator, favorite_count, retweet_count, and ratings are high. But for all rating variable, we can find that Q3 is not too far from another Q, so the max value from that variables definitely outlier. - In img_num just like the stat desc, the min until Q3 the result are 1, so the other value except 1 was missing value

This plot for question number 2:

```
In [100]: # correlation
          fig, ax = plt.subplots()
          fig.set_size_inches(11.7, 8.27)
          sns.heatmap(twitter_df.corr(), annot=True, ax=ax);
          plt.title('Correlation Between Variables');
          plt.savefig('plot/5_Correlation Between Variables.png')
```

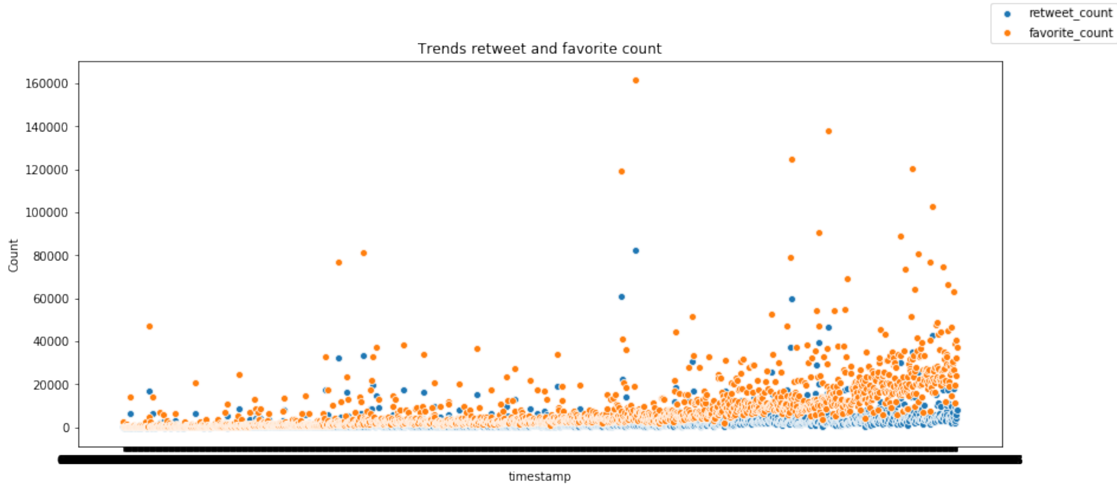


2. How about correlation between variables? Answer: Note: The correlation value between -1 until 1, negative just to make we know the correlation direction, the closer to the value 0, the smaller the correlation. It use pearson correlation so they just see the linear relationship between each variables. - To this plot please ignore correlation between rating and rating_numerator or rating_denominator because the result should be strong because rating is a calculation from both of them. But surprisingly the correlation between rating and rating_denominator is small. The answer can be found from stat desc that show if value rating_numerator is more varied than rating_denominator (std rating_denominator more hight than rating_numerator but their quantiles just similar each other) - We can see hight positive correlation between favorite_count and retweet_count. Its mean the more favorited the more retweeted - The correlation between all confidence variables also quite high. Somehow when p1_conf hight the confidence in p2 and p3 will decreese, but when confidence p3 increase the confidence in p2 will lightly increese.

This plot for question number 3:

```
In [101]: fig, ax = plt.subplots()
          fig.set_size_inches(14, 6)
          sns.scatterplot(x="timestamp", y="retweet_count", data=twitter_df,ax=ax);
          sns.scatterplot(x="timestamp", y="favorite_count", data=twitter_df,ax=ax);
          fig.legend(labels=['retweet_count','favorite_count']);
          plt.title('Trends retweet and favorite count');
```

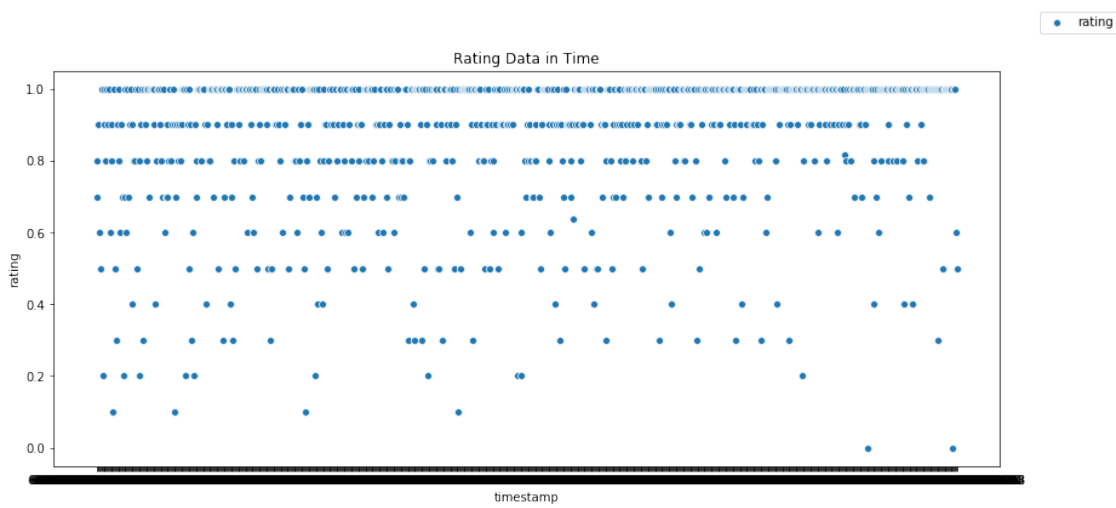
```python
plt.ylabel('Count');
plt.savefig('plot/6_Trends retweet and favorite count.png')
```



3. Does the retweet count and favorite count increase with time? Answer: From that visualization, favorite count and retweet count always increase with time. The trends are increasing for both variables. But favorite count growing larger than retweet count

This plot for question number 4:

```python
In [103]: fig, ax = plt.subplots()
          fig.set_size_inches(14, 6)
          sns.scatterplot(x="timestamp", y="rating", data=twitter_df.query('rating<=1'),ax=ax);
          fig.legend(labels=['rating']);
          plt.title('Rating Data in Time');
          plt.savefig('plot/7_Rating Data in Time.png');
```



4. Does the rating increase with time? Answer: Rating are sparated from min to max value in anytime, but from that plot from same date the rating are missing because the data was missing. Just like the correlation that rating didn't correlate with any variables.

This plot for question number 5:

In [105]:
```
# frist let make rating level
# this function will help to make level from quantile and return categories column
def get_class(df, column:str):
    # find quantile to decide that class
    min_value = df[column].min()
    quantile_1 = df[column].describe()[4]
    quantile_2 = df[column].describe()[5]
    quantile_3 = df[column].describe()[6]
    max_value = df[column].max()

    # bin edges that will be used to "cut" the data into groups
    bin_level = [ min_value, quantile_1, quantile_2, quantile_3, max_value]
    # labels for the four budget level groups
    bin_name = [ 'Low', 'Medium', 'High', 'Very High']
    # creates budget_levels column
    name = '{}_levels'.format(column)
    df[name] = pd.cut(df[column], bin_level, labels=bin_name, include_lowest = True)
    return df
```

In [106]:
```
rank_level_df = get_class(twitter_df, "rating")
rank_level_df = rank_level_df[['favorite_count','retweet_count','rating_levels']]
rank_level_df.head(3)
```

Out[106]:

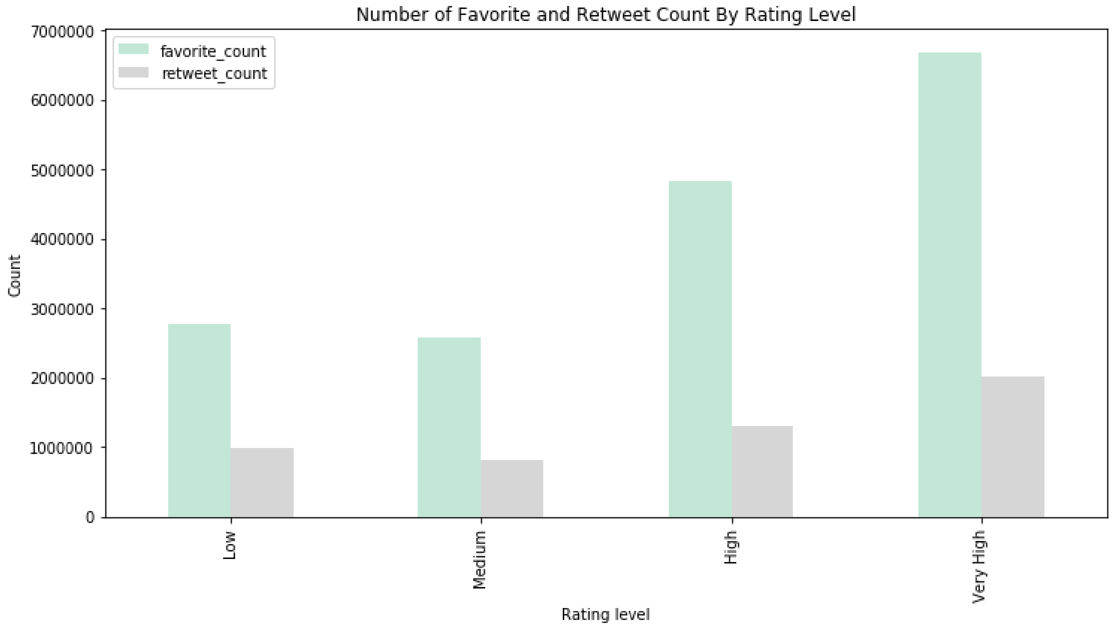
	favorite_count	retweet_count	rating_levels
0	2498	493	Low
1	124	46	Low
2	120	43	Low

In [107]:
```
rank_level_df = rank_level_df.groupby(['rating_levels']).sum()
rank_level_df
```

Out[107]:

rating_levels	favorite_count	retweet_count
Low	2757832	982252
Medium	2576693	810931
High	4822163	1296971
Very High	6680022	2006823

In [108]:
```
rank_level_df.plot(kind='bar',figsize=(12, 6),colormap='Pastel2',alpha=0.8).legend(loc
plt.xlabel('Rating level');
plt.ylabel('Count');
plt.title('Number of Favorite and Retweet Count By Rating Level');
plt.savefig('plot/8_Number of Favorite and Retweet Count By Rating Level.png')
```



5. Are the rating affect with the number of favorite and retweet count? Answer: From all data, we found that the higher the rating the higher the count (favorite and retweet). From that plot we also know that count in favorite alway higher than retweet.

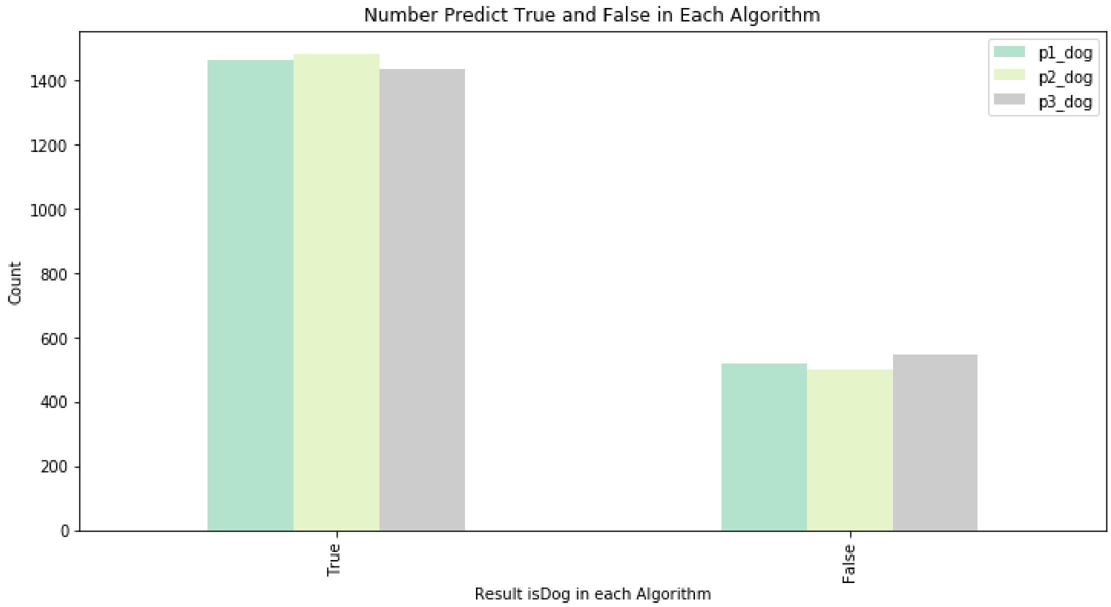
This plot for question number 6:

In [109]:
```
count = pd.DataFrame()
count['p1_dog'] = twitter_df.p1_dog.value_counts()
count['p2_dog'] = twitter_df.p2_dog.value_counts()
count['p3_dog'] = twitter_df.p3_dog.value_counts()
count
```

Out[109]:

	p1_dog	p2_dog	p3_dog
True	1462	1481	1433
False	516	497	545

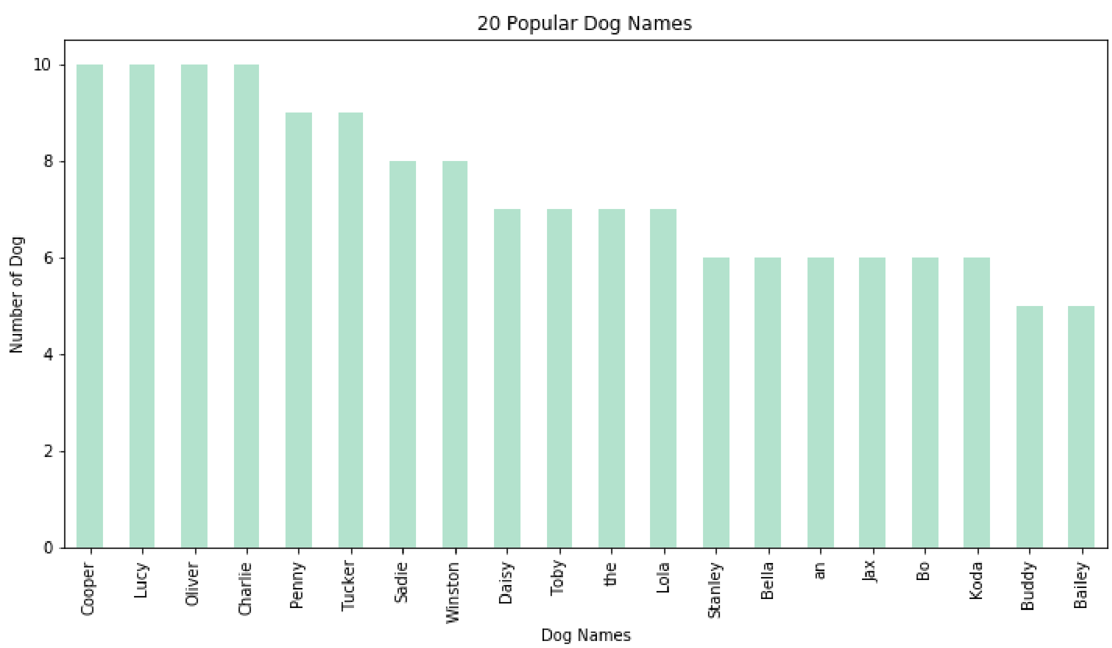
In [110]:
```
count.plot.bar(colormap='Pastel2',figsize=(12, 6));
plt.xlabel('Result isDog in each Algorithm');
plt.ylabel('Count');
plt.title('Number Predict True and False in Each Algorithm');
plt.savefig('plot/9_Number Predict True and False in Each Algorithm.png')
```



6. How much each algorithm predict the picture is dog? Answer: P2 predict picture dog large than p1 and p3. The smallest predicted is dog come from p3.

This plot for question number 7:

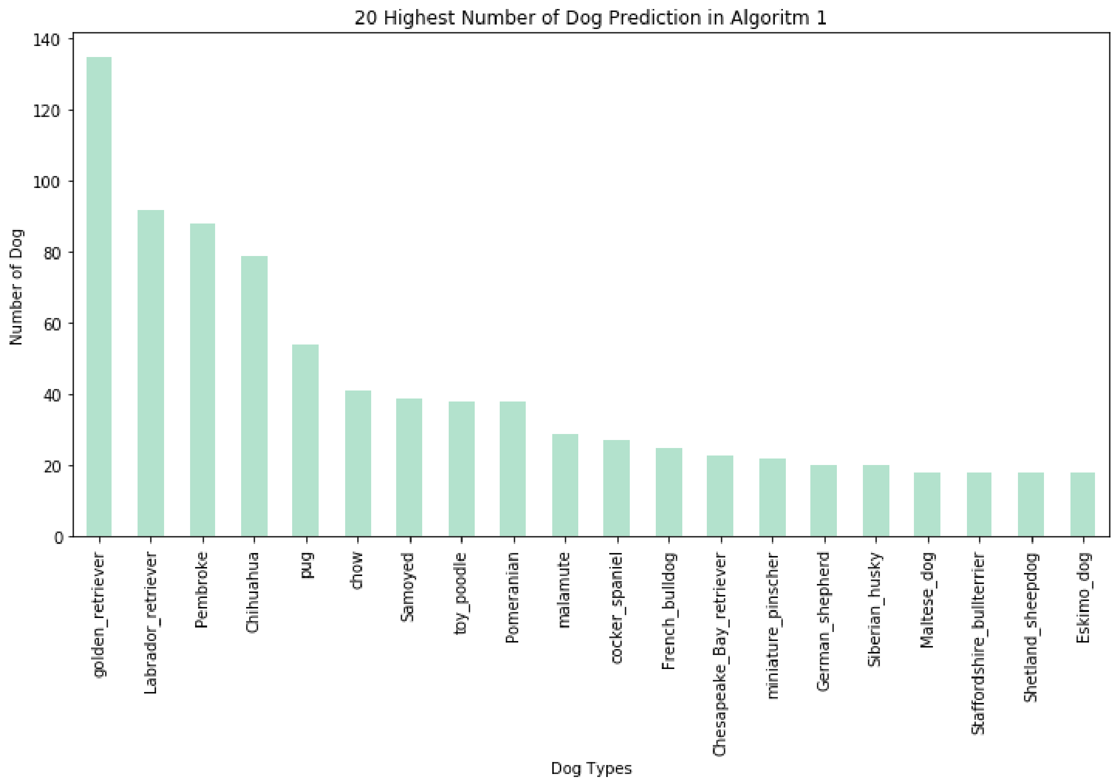
```
In [111]: twitter_df.name.value_counts()[0:20].plot.bar(colormap='Pastel2',figsize=(12, 6));
          plt.xlabel('Dog Names');
          plt.ylabel('Number of Dog');
          plt.title('20 Popular Dog Names');
          plt.savefig('plot/10_20 Popular Dog Names.png')
```



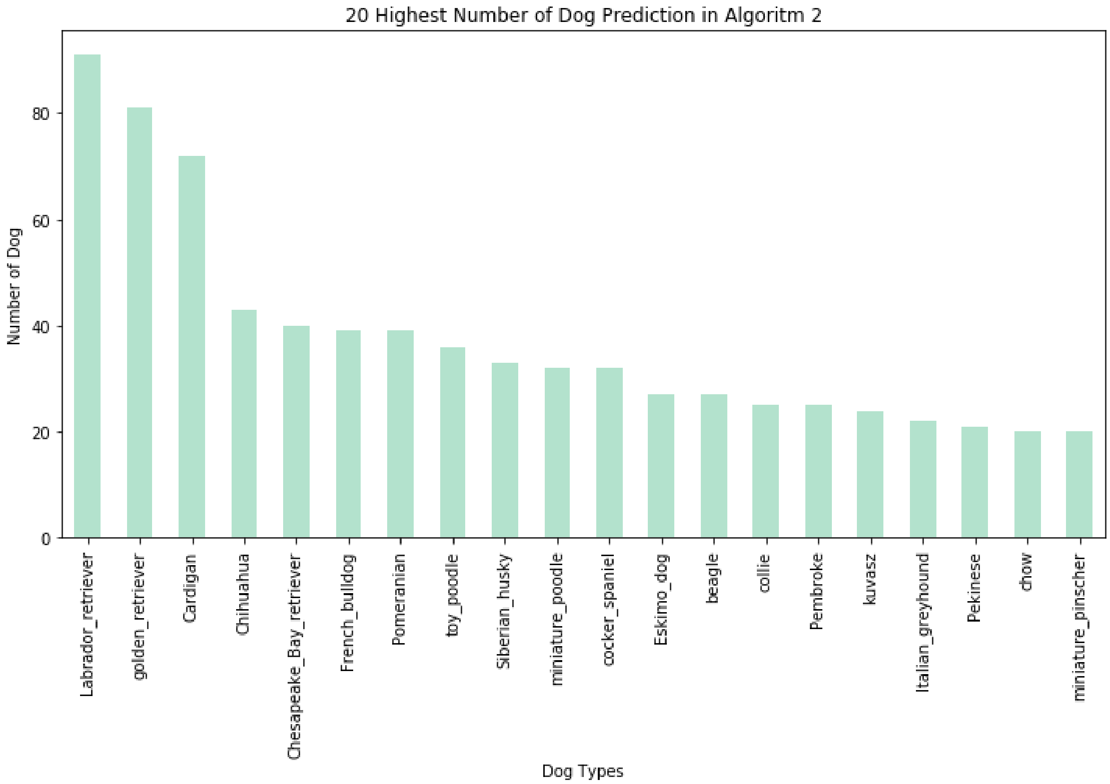
7. What are the most popular dog names? Answer: “Oliver”, “Lucy”, “Charlie”, and “Cooper” is the commond dog names in that table.

This plot for question number 8:

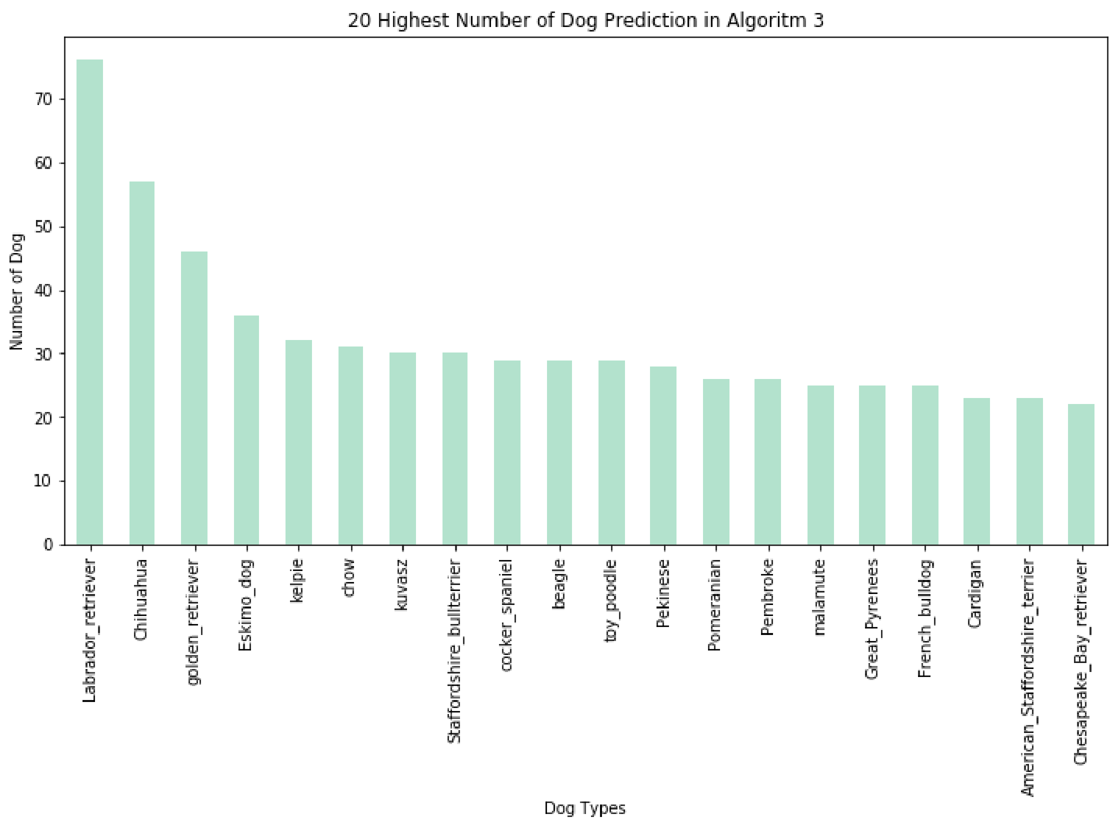
```
In [112]: twitter_df[twitter_df.p1_dog == True].p1.value_counts()[0:20].plot.bar(colormap='Paste
          plt.xlabel('Dog Types');
          plt.ylabel('Number of Dog');
          plt.title('20 Highest Number of Dog Prediction in Algoritm 1');
          plt.savefig('plot/11_20 Highest Number of Dog Prediction in Algoritm 1.png')
```




```
In [113]: twitter_df[twitter_df.p2_dog == True].p2.value_counts()[0:20].plot.bar(colormap='Paste
          plt.xlabel('Dog Types');
          plt.ylabel('Number of Dog');
          plt.title('20 Highest Number of Dog Prediction in Algoritm 2');
          plt.savefig('plot/12_20 Highest Number of Dog Prediction in Algoritm 2.png')
```



In [114]:
```
twitter_df[twitter_df.p3_dog == True].p3.value_counts()[0:20].plot.bar(colormap='Paste
plt.xlabel('Dog Types');
plt.ylabel('Number of Dog');
plt.title('20 Highest Number of Dog Prediction in Algoritm 3');
plt.savefig('plot/12_20 Highest Number of Dog Prediction in Algoritm 3.png')
```



8. What are the most popular dog predict? Answer: In Algorithm 1, golden retriever are the most popular dog, but in Algorithm 2 and 3, labrator retriever is the most popular dog.

This code for answer question number 9:

```
In [115]: top_dog = twitter_df.query('(p1 == p2) | (p2 == p3) | (p1 == p3)')
          top_dog
```

```
Out[115]:                tweet_id           timestamp              source  \
          1390  770414278348247044  770414278348247044  Twitter for iPhone

                                                           text  \
          1390  Meet Al Cabone. He's a gangsta puppa. Rather h...

                                                  expanded_urls  rating_numerator  \
          1390  https://twitter.com/dog_rates/status/770414278...                11

                rating_denominator name dog_stage  favorite_count  ...   p1_conf p1_dog  \
          1390                  10   Al       NaN            6587  ...  0.580528  False

                     p2   p2_conf  p2_dog                p3  p3_conf  p3_dog  rating  \
          1390  maillot  0.081449   False  golden_retriever  0.05357    True     1.1
```

```
      rating_levels
1390        Medium

[1 rows x 24 columns]
```

9. What are the most popular dog predict when all algorithm predict the same dog? Answer: When I select data with same answer in at least 2 Algorithm, there are just 1 data, that predict mailot and exist in algorithm 1 and 2.

Predict dog_stage

Because dog_stage have many missing value, so I try to predict them with simple decision tree model. Todo:

Make sure the type of our table

Define X and Y as predictor and label

Encode categorical predictor

Split train, and test data

Make prediction

Show Metrics from prediction

The predictions

1.Make sure the type of our table

In [116]:
```
# because rating_levels from analyzing data have categorical datatype,
# so we must parse it to object (just like another columns)
twitter_df.rating_levels = twitter_df.rating_levels.astype("object")
twitter_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1978 entries, 0 to 1977
Data columns (total 24 columns):
tweet_id              1978 non-null object
timestamp             1978 non-null object
source                1978 non-null object
text                  1978 non-null object
expanded_urls         1978 non-null object
rating_numerator      1978 non-null int64
rating_denominator    1978 non-null int64
name                  1383 non-null object
dog_stage             294 non-null object
favorite_count        1978 non-null int64
retweet_count         1978 non-null int64
jpg_url               1978 non-null object
img_num               1978 non-null int64
p1                    1978 non-null object
p1_conf               1978 non-null float64
p1_dog                1978 non-null bool
p2                    1978 non-null object
p2_conf               1978 non-null float64
p2_dog                1978 non-null bool
p3                    1978 non-null object
```

```
p3_conf               1978 non-null float64
p3_dog                1978 non-null bool
rating                1978 non-null float64
rating_levels         1978 non-null object
dtypes: bool(3), float64(4), int64(5), object(12)
memory usage: 345.8+ KB
```

2. Define X and Y as predictor and label

```
In [117]: # I use data with not na in dog stage tobe my dataset in my models
          dataset = twitter_df[pd.notna(twitter_df.dog_stage)]
          dataset.dog_stage.value_counts()

Out[117]: pupper     203
          doggo       62
          puppo       22
          floofer      7
          Name: dog_stage, dtype: int64
```

Because of imbalance data, I decide to upsampling the other label into 203 data just like 'puupper'

```
In [118]: def rand_sampling_dog_stage(query, n:int, dataFrame):
              col_df = dataFrame.query(query)
              return col_df.sample(n = n, replace=True, random_state=9835)

In [119]: new_dataset = pd.DataFrame()
          new_dataset = pd.concat([dataset.query('dog_stage == "pupper"'),\
                                   rand_sampling_dog_stage('dog_stage == "doggo"', 203,dataset),\
                                   rand_sampling_dog_stage('dog_stage == "puppo"', 203,dataset),\
                                   rand_sampling_dog_stage('dog_stage == "floofer"', 203,dataset)]
          new_dataset.dog_stage.value_counts()

Out[119]: puppo      203
          pupper     203
          doggo      203
          floofer    203
          Name: dog_stage, dtype: int64

In [120]: # define predictor and label

          Y = new_dataset.dog_stage
          X = new_dataset.drop(['dog_stage','timestamp','tweet_id','expanded_urls','jpg_url'], a
```

3. Encode categorical predictor

```
In [121]: # Make mask for categorical dtypes only
          categorical_feature_mask = X.dtypes==object
          # filter categorical columns using mask and turn it into a list
          categorical_cols = X.columns[categorical_feature_mask].tolist()
```

In [122]:
```python
from sklearn.preprocessing import LabelEncoder

# instantiate labelencoder object to help us encode each variable
le = LabelEncoder()
```

In [123]:
```python
# Our categorical variable
categorical_cols
```

Out[123]:
```
['source', 'text', 'name', 'p1', 'p2', 'p3', 'rating_levels']
```

In [124]:
```python
# apply our le to categorical feature columns

X[categorical_cols] = X[categorical_cols].apply(lambda col: le.fit_transform(col.astyp
X[categorical_cols].head(10)
```

Out[124]:

	source	text	name	p1	p2	p3	rating_levels
331	2	140	21	107	14	28	1
333	2	214	107	62	63	121	1
339	2	228	124	107	14	33	2
346	2	111	70	6	61	44	0
353	2	93	135	20	26	0	1
356	2	208	102	117	73	127	1
357	2	264	164	76	20	77	0
363	2	143	26	97	51	110	2
367	2	144	27	89	111	81	1
368	2	147	29	107	7	20	1

4. Split train, and test data

In [125]:
```python
from sklearn.model_selection import train_test_split

# split sratify so all lebel is balance, use 30% data to be our data test
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.3, stratify =
```

In [126]:
```python
# label in data train
y_train.value_counts()
```

Out[126]:
```
pupper     142
puppo      142
doggo      142
floofer    142
Name: dog_stage, dtype: int64
```

In [127]:
```python
# label in data test
y_test.value_counts()
```

Out[127]:
```
puppo      61
pupper     61
doggo      61
floofer    61
Name: dog_stage, dtype: int64
```

5. Make prediction

```
In [128]: from sklearn.tree import DecisionTreeClassifier

          # Make classifier object, in there I use gini
          clf_gini = DecisionTreeClassifier(criterion = "gini", random_state = 9835)

          # Train the model
          clf_gini.fit(X_train, y_train)

          # Predict with test sample
          y_pred = clf_gini.predict(X_test)
```

6. Show Metrics from prediction

```
In [130]: from sklearn.metrics import confusion_matrix, accuracy_score, classification_report, f

          print("Confusion Matrix:\n",
          confusion_matrix(y_test, y_pred))

          print ("Accuracy:\n",
          accuracy_score(y_test,y_pred)*100)

          print("Report:\n",
          classification_report(y_test, y_pred))
```

```
Confusion Matrix:
 [[59  0  1  1]
 [ 0 61  0  0]
 [14  0 42  5]
 [ 0  0  0 61]]
Accuracy:
 91.39344262295081
Report:
               precision    recall  f1-score   support

       doggo       0.81      0.97      0.88        61
     floofer       1.00      1.00      1.00        61
      pupper       0.98      0.69      0.81        61
       puppo       0.91      1.00      0.95        61

   micro avg       0.91      0.91      0.91       244
   macro avg       0.92      0.91      0.91       244
weighted avg       0.92      0.91      0.91       244
```

Because the metrics are good enough, so let's predict another row

In [131]: # get the data

```
data_predict = twitter_df.copy()
data_predict.head(2)
```

Out[131]:

```
              tweet_id           timestamp              source  \
0  666020888022790149  666020888022790149  Twitter for iPhone
1  666029285002620928  666029285002620928  Twitter for iPhone

                                                text  \
0  Here we have a Japanese Irish Setter. Lost eye...
1  This is a western brown Mitsubishi terrier. Up...

                                       expanded_urls  rating_numerator  \
0  https://twitter.com/dog_rates/status/666020888...                 8
1  https://twitter.com/dog_rates/status/666029285...                 7

   rating_denominator name dog_stage  favorite_count  ...   p1_conf p1_dog  \
0                  10  NaN       NaN            2498  ...  0.465074   True
1                  10  NaN       NaN             124  ...  0.506826   True

                  p2   p2_conf  p2_dog                  p3   p3_conf  \
0             collie  0.156665    True    Shetland_sheepdog  0.061428
1  miniature_pinscher  0.074192    True  Rhodesian_ridgeback  0.072010

   p3_dog  rating rating_levels
0    True     0.8           Low
1    True     0.7           Low

[2 rows x 24 columns]
```

In [132]: # encode the data

```
data_predict[categorical_cols] = data_predict[categorical_cols].apply(lambda col: le.f
data_predict[categorical_cols].head(10)
```

Out[132]:

```
   source  text  name   p1   p2   p3  rating_levels
0       2   174   919   77  149   73              1
1       2  1801   919  287  253   66              1
2       2   164   919   32  246  120              1
3       2  1792   919   60  304  257              1
4       2   171   919  243   63   25              1
5       2  1799   919   13   27   41              1
6       2  1810   910  119  263  365              1
7       2   158   919  137   75  194              1
8       2   530   919  304  319  200              1
9       2   166   926  244  230  345              1
```

In [133]: # make prediction

```
data_predict.loc[:,'dog_stage'] = clf_gini.predict(data_predict.drop(['dog_stage','tim
data_predict.head()
```

```
Out[133]:              tweet_id           timestamp  source  text  \
          0  666020888022790149  666020888022790149       2   174
          1  666029285002620928  666029285002620928       2  1801
          2  666033412701032449  666033412701032449       2   164
          3  666044226329800704  666044226329800704       2  1792
          4  666049248165822465  666049248165822465       2   171

                                                 expanded_urls  rating_numerator  \
          0  https://twitter.com/dog_rates/status/666020888...                 8
          1  https://twitter.com/dog_rates/status/666029285...                 7
          2  https://twitter.com/dog_rates/status/666033412...                 9
          3  https://twitter.com/dog_rates/status/666044226...                 6
          4  https://twitter.com/dog_rates/status/666049248...                 5

             rating_denominator  name dog_stage  favorite_count  ...   p1_conf p1_dog  \
          0                  10   919    pupper            2498  ...  0.465074   True
          1                  10   919    pupper             124  ...  0.506826   True
          2                  10   919    pupper             120  ...  0.596461   True
          3                  10   919    pupper             288  ...  0.408143   True
          4                  10   919    pupper             104  ...  0.560311   True

              p2   p2_conf  p2_dog   p3   p3_conf  p3_dog  rating  rating_levels
          0  149  0.156665    True   73  0.061428    True     0.8              1
          1  253  0.074192    True   66  0.072010    True     0.7              1
          2  246  0.138584    True  120  0.116197    True     0.9              1
          3  304  0.360687    True  257  0.222752    True     0.6              1
          4   63  0.243682    True   25  0.154629    True     0.5              1

          [5 rows x 24 columns]
```

```
In [134]: # let's make their into 1 table

          # change col name so it will not be duplicated
          data_predict = data_predict[['tweet_id','dog_stage']]
          data_predict.columns =["tweet_id", "predict_dog_stage"]
          data_predict.head()
```

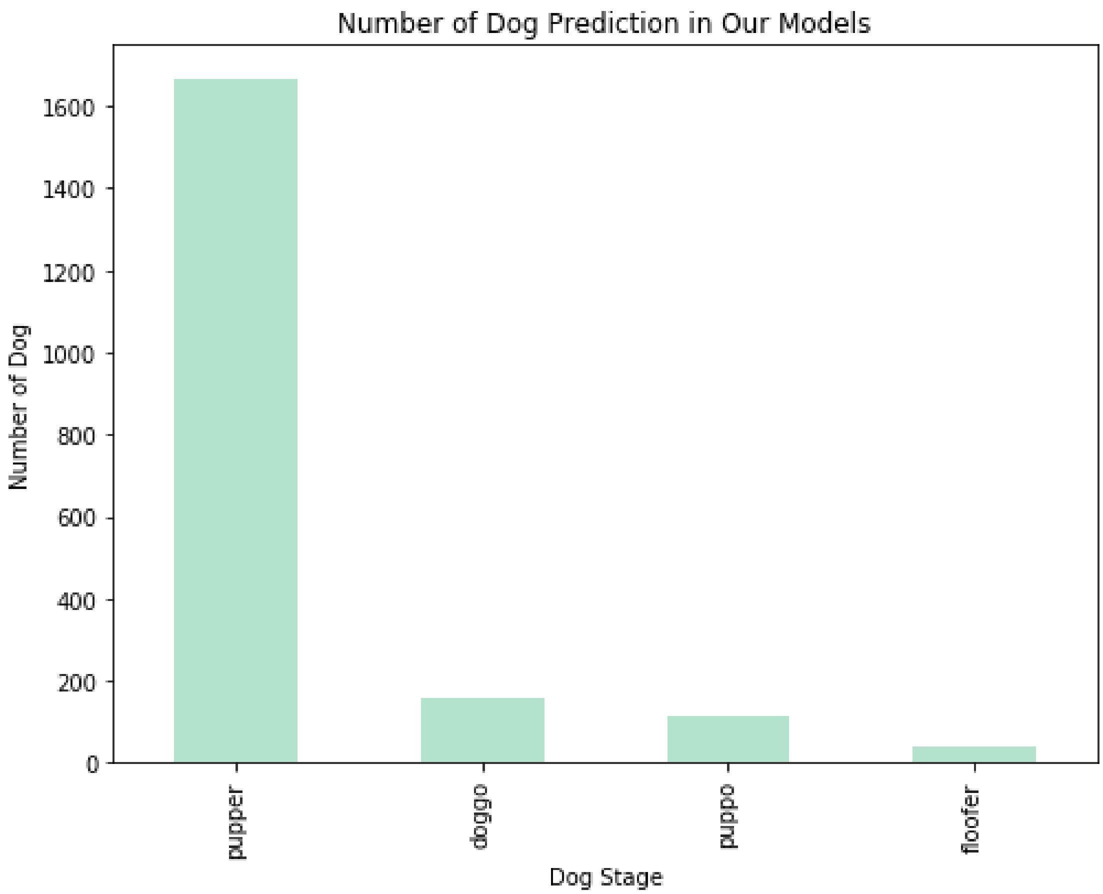
```
Out[134]:              tweet_id predict_dog_stage
          0  666020888022790149            pupper
          1  666029285002620928            pupper
          2  666033412701032449            pupper
          3  666044226329800704            pupper
          4  666049248165822465            pupper
```

```
In [135]: # join table
          twitter_predict_df = pd.merge(twitter_df, data_predict, how = 'inner', on = ['tweet_id
          twitter_predict_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1978 entries, 0 to 1977
```

```
Data columns (total 25 columns):
tweet_id              1978 non-null object
timestamp             1978 non-null object
source                1978 non-null object
text                  1978 non-null object
expanded_urls         1978 non-null object
rating_numerator      1978 non-null int64
rating_denominator    1978 non-null int64
name                  1383 non-null object
dog_stage             294 non-null object
favorite_count        1978 non-null int64
retweet_count         1978 non-null int64
jpg_url               1978 non-null object
img_num               1978 non-null int64
p1                    1978 non-null object
p1_conf               1978 non-null float64
p1_dog                1978 non-null bool
p2                    1978 non-null object
p2_conf               1978 non-null float64
p2_dog                1978 non-null bool
p3                    1978 non-null object
p3_conf               1978 non-null float64
p3_dog                1978 non-null bool
rating                1978 non-null float64
rating_levels         1978 non-null object
predict_dog_stage     1978 non-null object
dtypes: bool(3), float64(4), int64(5), object(13)
memory usage: 361.2+ KB
```

```
In [136]: twitter_predict_df.predict_dog_stage.value_counts()[0:20].plot.bar(colormap='Pastel2',
          plt.xlabel('Dog Stage');
          plt.ylabel('Number of Dog');
          plt.title('Number of Dog Prediction in Our Models');
          plt.savefig('plot/13_Number of Dog Prediction in Our Models');
```



Just like the value before uppersampling, the popular dog_stage is pupper.

In [137]:
```
# save the prediction dataframe
twitter_predict_df.to_csv('data_generated/twitter_predict_df.csv',index=False)
```