wrangle_act

January 1, 2019

0.1 Wrangle WeRateDogs Twitter data to create interesting and trustworthy analyses and visualizations

0.1.1 Step1: Gather Data

Import library

```
In [170]: import pandas as pd
          import numpy as np
          import requests
          import tweepy
          import os
          import matplotlib.pyplot as plt

          %matplotlib inline
          folder_name = 'Twitter'
          if not os.path.exists(folder_name):
              os.makedirs(folder_name)
```

Extract WeRateDogs Twitter archive file

```
In [171]: #Activity1 - The WeRateDogs Twitter archive. I am giving this file to you, so imagine
          twitter_archive= pd.read_csv('twitter-archive-enhanced.csv')
          twitter_ids=twitter_archive['tweet_id']
          twitter_ids.drop_duplicates(inplace=True)
```

Extract tweet image predictions File

```
In [172]: url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predict

          response = requests.get(url)
          with open(os.path.join(folder_name,url.split('/')[-1]),mode='wb') as file:
              file.write(response.content)

          image_predictions= pd.read_csv(folder_name+'/'+'image-predictions.tsv','\t')

          image_predictions.head(1)

          #image_predictions.to_csv("twitter_data.csv")
```

Out[172]:

	tweet_id	jpg_url	img_num	
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh

Extract Twitter feeds via API

In [ ]:
```
from timeit import default_timer as timer
import json
consumer_key = 'gLmSt5mvejJvElV9GMxM9aibM'
consumer_secret = '1dFI0dzqIBn7KrPd4amWkENxhb8ePGuxs3eHq0ykBNe9xcpOKN'
access_token = '2928378330-FL6ID7AiJar6vMhFF7iP2XoJekegx8oh1wPAD33'
access_secret = 'Qk1FSIv3LN3aRTxZ0oqQC09jGWVg3ZevsSPpe2PpDoGAv'

auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_secret)

api = tweepy.API(auth)


# Query Twitter's API for JSON data for each tweet ID in the Twitter archive file
count = 0
fails_dict = []
start = timer()
# Save each tweet's returned JSON as a new line in a .txt file
with open('tweet_json.txt', 'w') as outfile:
    # This loop will likely take 20-30 minutes to run because of Twitter's rate limit
    for tweet_id in twitter_ids:
        count += 1
        try:
            tweet = api.get_status(tweet_id, tweet_mode='extended',wait_on_rate_limit =
            #print("Success")
            json.dump(tweet._json, outfile)
            outfile.write('\n')
        except tweepy.TweepError as e:
            #print("Fail")
            fails_dict[count] = e
            pass
end = timer()
print(end - start)
print(fails_dict)
```

In [173]:
```
twitter_data = pd.read_json('tweet_json.txt',lines=True)
twitter_data.rename(index=str,columns={'id':'tweet_id'},inplace=True)
twitter_data=twitter_data[['tweet_id','retweet_count','favorite_count']]
twitter_data.head()
```

Out[173]:

	tweet_id	retweet_count	favorite_count
0	892420643555336193	8331	38089
1	892177421306343426	6154	32683

```
2  891815181378084864  4073           24599
3  891689557279858688  8473           41450
4  891327558926688256  9167           39625
```

0.1.2 Step2: Analyse

```
In [174]: # Check attribute types and missing values
          twitter_archive.info()
```

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

```
In [175]: # Check for duplicates
          twitter_archive[twitter_archive.duplicated()]
```

```
Out[175]: Empty DataFrame
          Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, tex
          Index: []
```

```
In [176]: twitter_archive['tweet_id'].value_counts().sort_values(ascending =False)
```

```
Out[176]: 700151421916807169    1
          676948236477857792    1
          677228873407442944    1
          708349470027751425    1
          861383897657036800    1
          668979806671884288    1
          684147889187209216    1
```

682242692827447297	1
723179728551723008	1
673320132811366400	1
673956914389192708	1
849051919805034497	1
747844099428986880	1
855860136149123072	1
850019790995546112	1
801115127852503040	1
883360690899218434	1
732732193018155009	1
668291999406125056	1
666835007768551424	1
690348396616552449	1
701981390485725185	1
685321586178670592	1
832040443403784192	1
874296783580663808	1
673270968295534593	1
802239329049477120	1
788552643979468800	1
728387165835677696	1
684969860808454144	1
692041934689402880	1
751538714308972544	1
857214891891077121	1
847978865427394560	1
686377065986265092	1
678798276842360832	1
712097430750289920	1
733460102733135873	1
801958328846974976	1
706538006853918722	1
727314416056803329	1
678021115718029313	1
849668094696017920	1
705975130514706432	1
789960241177853952	1
681891461017812993	1
834574053763584002	1
828381636999917570	1
771136648247640064	1
780601303617732608	1
670474236058800128	1
746507379341139972	1
777189768882946048	1
710269109699739648	1
696488710901260288	1

686007916130873345	1
773247561583001600	1
755206590534418437	1
666691418707132416	1
790277117346975746	1
676191832485810177	1
695794761660297217	1
668204964695683073	1
677565715327688705	1
679475951516934144	1
670755717859713024	1
670826280409919488	1
667165590075940865	1
674063288070742018	1
723673163800948736	1
887473957103951883	1
670403879788544000	1
790698755171364864	1
773704687002451968	1
808134635716833280	1
717428917016076293	1
741793263812808706	1
670086499208155136	1
773670353721753600	1
753298634498793472	1
780496263422808064	1
704480331685040129	1
840696689258311684	1
703407252292673536	1
750132105863102464	1
750026558547456000	1
692535307825213440	1
838831947270979586	1
771770456517009408	1
752173152931807232	1
744995568523612160	1
708130923141795840	1
748568946752774144	1
671882082306625538	1
772826264096874500	1
677687604918272002	1
886366144734445568	1
748337862848962560	1
681302363064414209	1
790946055508652032	1
813217897535406080	1
707629649552134146	1
835172783151792128	1

```
691444869282295808    1
678708137298427904    1
762464539388485633    1
890729181411237888    1
690248561355657216    1
667801013445750784    1
680889648562991104    1
872820683541237760    1
666428276349472768    1
706644897839910912    1
780074436359819264    1
666044226329800704    1
787322443945877504    1
754747087846248448    1
761004547850530816    1
804738756058218496    1
834786237630337024    1
677328882937298944    1
810896069567610880    1
891327558926688256    1
698710712454139905    1
773336787167145985    1
786051337297522688    1
871166179821445120    1
706169069255446529    1
761672994376806400    1
772615324260794368    1
676237365392908289    1
766423258543644672    1
889638837579907072    1
674767892831932416    1
680221482581123072    1
703382836347330562    1
684959798585110529    1
848213670039564288    1
770293558247038976    1
681523177663676416    1
691096613310316544    1
772193107915964416    1
769695466921623552    1
815990720817401858    1
693644216740769793    1
688894073864884227    1
778286810187399168    1
835685285446955009    1
782747134529531904    1
855862651834028034    1
817827839487737858    1
```

746790600704425984 1
687399393394311168 1
869988702071779329 1
845306882940190720 1
787111942498508800 1
672481316919734272 1
765719909049503744 1
885311592912609280 1
798933969379225600 1
669597912108789760 1
813112105746448384 1
677547928504967168 1
794332329137291264 1
835297930240217089 1
789599242079838210 1
682259524040966145 1
681981167097122816 1
788765914992902144 1
666337882303524864 1
827199976799354881 1
827324948884643840 1
671735591348891648 1
783839966405230592 1
683111407806746624 1
668960084974809088 1
727685679342333952 1
752701944171524096 1
693486665285931008 1
773308824254029826 1
711008018775851008 1
796031486298386433 1
796125600683540480 1
683828599284170753 1
743835915802583040 1
825147591692263424 1
801854953262350336 1
676440007570247681 1
678023323247357953 1
802265048156610565 1
673906403526995968 1
706593038911545345 1
684225744407494656 1
817423860136083457 1
709566166965075968 1
706153300320784384 1
694925794720792577 1
851464819735769094 1
676897532954456065 1

838921590096166913 1
666781792255496192 1
760290219849637889 1
669367896104181761 1
829878982036299777 1
863427515083354112 1
697463031882764288 1
671743150407421952 1
737678689543020544 1
680913438424612864 1
703769065844768768 1
867421006826221569 1
818145370475810820 1
747600769478692864 1
867051520902168576 1
729463711119904772 1
710833117892898816 1
883482846933004288 1
742534281772302336 1
847606175596138505 1
670782429121134593 1
798673117451325440 1
666373753744588802 1
672095186491711488 1
702899151802126337 1
819711362133872643 1
833732339549220864 1
672997845381865473 1
725786712245440512 1
793150605191548928 1
671891728106971137 1
698549713696649216 1
688916208532455424 1
778774459159379968 1
887705289381826560 1
744223424764059648 1
819588359383371776 1
685325112850124800 1
746369468511756288 1
667530908589760512 1
784826020293709826 1
667369227918143488 1
875144289856114688 1
682088079302213632 1
758041019896193024 1
750011400160841729 1
689977555533848577 1
705239209544720384 1

```
739544079319588864    1
770787852854652928    1
820749716845686786    1
                     ..
672264251789176834    1
734787690684657664    1
675853064436391936    1
785170936622350336    1
880872448815771648    1
747594051852075008    1
672968025906282496    1
801285448605831168    1
748692773788876800    1
674410619106390016    1
683742671509258241    1
667886921285246976    1
687312378585812992    1
800388270626521089    1
673707060090052608    1
760190180481531904    1
747963614829678593    1
702276748847800320    1
668992363537309700    1
759099523532779520    1
747242308580548608    1
749036806121881602    1
874680097055178752    1
757393109802180609    1
676146341966438401    1
772152991789019136    1
739932936087216128    1
715928423106027520    1
722613351520608256    1
816062466425819140    1
724983749226668032    1
801127390143516673    1
789903600034189313    1
708119489313951744    1
793210959003287553    1
670408998013820928    1
671561002136281088    1
891815181378084864    1
831650051525054464    1
739238157791694849    1
676613908052996102    1
732726085725589504    1
772114945936949249    1
884162670584377345    1
```

691756958957883396 1
670727704916926465 1
777641927919427584 1
787810552592695296 1
852936405516943360 1
809448704142938112 1
833479644947025920 1
666983947667116034 1
809220051211603969 1
820690176645140481 1
710844581445812225 1
669942763794931712 1
752309394570878976 1
879008229531029506 1
674255168825880576 1
800459316964663297 1
798576900688019456 1
668815180734689280 1
684594889858887680 1
816450570814898180 1
668852170888998912 1
765222098633691136 1
818614493328580609 1
798682547630837760 1
699779630832685056 1
773191612633579521 1
675888385639251968 1
716439118184652801 1
744234799360020481 1
689877686181715968 1
708469915515297792 1
758355060040593408 1
670822709593571328 1
767122157629476866 1
667902449697558528 1
790227638568808452 1
784057939640352768 1
874434818259525634 1
671768281401958400 1
684567543613382656 1
860184849394610176 1
675349384339542016 1
682662431982772225 1
666437273139982337 1
823699002998870016 1
788039637453406209 1
667885044254572545 1
798665375516884993 1

669661792646373376 1
758467244762497024 1
829011960981237760 1
668975677807423489 1
672160042234327040 1
677895101218201600 1
887101392804085760 1
855857698524602368 1
676821958043033607 1
860924035999428608 1
719551379208073216 1
714258258790387713 1
673662677122719744 1
671879137494245376 1
676593408224403456 1
693622659251335168 1
684122891630342144 1
673919437611909120 1
741438259667034112 1
874012996292530176 1
689154315265683456 1
791312159183634433 1
890971913173991426 1
774757898236878852 1
687096057537363968 1
869227993411051520 1
748932637671223296 1
744334592493166593 1
783695101801398276 1
832636094638288896 1
667544320556335104 1
692919143163629568 1
666073100786774016 1
674447403907457024 1
713900603437621249 1
813812741911748608 1
673708611235921920 1
856526610513747968 1
687807801670897665 1
688116655151435777 1
758854675097526272 1
740699697422163968 1
840728873075638272 1
859851578198683649 1
669926384437997569 1
793601777308463104 1
738166403467907072 1
672640509974827008 1

822872901745569793 1
674038233588723717 1
698355670425473025 1
667773195014021121 1
671497587707535361 1
835574547218894849 1
800513324630806528 1
693095443459342336 1
700167517596164096 1
670823764196741120 1
685547936038666240 1
839990271299457024 1
715220193576927233 1
723688335806480385 1
667550882905632768 1
672828477930868736 1
771380798096281600 1
844704788403113984 1
680191257256136705 1
691090071332753408 1
698953797952008193 1
854732716440526848 1
680055455951884288 1
668484198282485761 1
671347597085433856 1
738184450748633089 1
696754882863349760 1
689999384604450816 1
761371037149827077 1
669328503091937280 1
669363888236994561 1
799422933579902976 1
666804364988780544 1
691459709405118465 1
864197398364647424 1
670832455012716544 1
772877495989305348 1
699323444782047232 1
688908934925697024 1
865006731092295680 1
886680336477933568 1
672256522047614977 1
771908950375665664 1
668466899341221888 1
676101918813499392 1
798701998996647937 1
737322739594330112 1
690735892932222976 1

750868782890057730 1
666020888022790149 1
817120970343411712 1
673576835670777856 1
861769973181624320 1
674036086168010753 1
735635087207878657 1
769335591808995329 1
678389028614488064 1
667728196545200128 1
751132876104687617 1
805826884734976000 1
781655249211752448 1
678424312106393600 1
680070545539371008 1
700062718104104960 1
885167619883638784 1
677716515794329600 1
697881462549430272 1
725458796924002305 1
680130881361686529 1
670290420111441920 1
704847917308362754 1
750429297815552001 1
807621403335917568 1
742385895052087300 1
775364825476165632 1
673580926094458881 1
763183847194451968 1
851953902622658560 1
778383385161035776 1
668256321989451776 1
799757965289017345 1
679405845277462528 1
847116187444137987 1
668221241640230912 1
667924896115245057 1
805932879469572096 1
855138241867124737 1
681231109724700672 1
836648853927522308 1
800855607700029440 1
668932921458302977 1
792773781206999040 1
821765923262631936 1
700505138482569216 1
781251288990355457 1
669006782128353280 1

```
770414278348247044    1
666063827256086533    1
675135153782571009    1
865359393868664832    1
766069199026450432    1
773985732834758656    1
667517642048163840    1
808501579447930884    1
820446719150292993    1
832215909146226688    1
856282028240666624    1
681679526984871937    1
822610361945911296    1
749075273010798592    1
Name: tweet_id, Length: 2356, dtype: int64
```

```
In [177]: twitter_archive['name'].value_counts().sort_values(ascending=False)
```

```
Out[177]: None           745
          a               55
          Charlie         12
          Lucy            11
          Oliver          11
          Cooper          11
          Tucker          10
          Penny           10
          Lola            10
          Winston          9
          Bo               9
          the              8
          Sadie            8
          an               7
          Bailey           7
          Buddy            7
          Daisy            7
          Toby             7
          Rusty            6
          Scout            6
          Milo             6
          Jack             6
          Bella            6
          Koda             6
          Dave             6
          Leo              6
          Jax              6
          Stanley          6
          Oscar            6
          Oakley           5
```

Alfie	5
Chester	5
Sammy	5
George	5
Finn	5
Sunny	5
Gus	5
Larry	5
Phil	5
Louis	5
Bentley	5
very	5
one	4
Derek	4
Gerald	4
Duke	4
Gary	4
Bruce	4
Scooter	4
Jeffrey	4
just	4
Sophie	4
Clark	4
Hank	4
Maddie	4
Archie	4
quite	4
Walter	4
Boomer	4
Winnie	4
Clarence	4
Riley	4
Maggie	4
Jerry	4
Sampson	4
Chip	4
Cassie	4
Dexter	4
Beau	4
Moose	4
Reginald	4
Luna	4
Shadow	4
Maximus	4
Ruby	4
Reggie	4
Bear	4
Loki	4

Brody	4
Carl	4
Olive	3
Ellie	3
Sebastian	3
Peaches	3
Reese	3
Mia	3
Coco	3
Doug	3
Louie	3
Earl	3
Arnie	3
Max	3
Otis	3
Paisley	3
Nala	3
Ted	3
Wallace	3
Gizmo	3
Malcolm	3
Samson	3
Calvin	3
Steven	3
Rosie	3
Wyatt	3
Waffles	3
Vincent	3
Rory	3
Kyle	3
Lily	3
Lorenzo	3
Wilson	3
Frankie	3
Zoey	3
Colby	3
Jimothy	3
Zeke	3
Klevin	3
Cupcake	2
Griffin	2
Carly	2
Bungalo	2
Aspen	2
Levi	2
Percy	2
Lou	2
Blitz	2

Albus	2
Indie	2
Brad	2
Balto	2
Bubbles	2
Harold	2
Tyrone	2
Ken	2
getting	2
Shaggy	2
Penelope	2
Kyro	2
Titan	2
Olivia	2
Chompsky	2
Sugar	2
Cody	2
Hammond	2
Sam	2
Calbert	2
Cash	2
Meyer	2
Kilo	2
Phred	2
Jackson	2
Pickles	2
Remington	2
Rocco	2
Quinn	2
Ash	2
Raymond	2
Yogi	2
Django	2
Lilly	2
Trooper	2
Fiona	2
Atticus	2
Fizz	2
Logan	2
Pablo	2
Canela	2
Kenneth	2
Jiminy	2
Harper	2
Thumas	2
Watson	2
Sarge	2
Philbert	2

Chipson	2
Dakota	2
Hunter	2
Rizzy	2
Dash	2
Oshie	2
Davey	2
Abby	2
actually	2
Crystal	2
Jamesy	2
Franklin	2
Gabe	2
Bisquick	2
Doc	2
Kevin	2
Nollie	2
Mister	2
Misty	2
Betty	2
Opal	2
Rocky	2
Solomon	2
Coops	2
Juno	2
Belle	2
Baloo	2
Pippa	2
Ollie	2
Sansa	2
Elliot	2
Benedict	2
Neptune	2
Tyr	2
Eve	2
Gromit	2
Phineas	2
Leela	2
Linda	2
Maxaroni	2
Kenny	2
Bob	2
Moreton	2
Patrick	2
Fred	2
Curtis	2
Rufus	2
Theodore	2

Mattie	2
Paull	2
Kreggory	2
Eli	2
Oliviér	2
Layla	2
Luca	2
Panda	2
Nelly	2
Seamus	2
Churlie	2
Odie	2
Astrid	2
Ava	2
Butter	2
Smokey	2
Lennon	2
Happy	2
Finley	2
Chet	2
Dawn	2
Ozzy	2
Bernie	2
Chuckles	2
Marley	2
Baxter	2
Pipsy	2
Terry	2
	..
Mookie	1
Eriq	1
Barry	1
Gustaf	1
Maude	1
Duddles	1
Billy	1
Millie	1
Edd	1
Mauve	1
Rodney	1
Amélie	1
Aqua	1
Dug	1
Mason	1
Doobert	1
Tessa	1
Flurpson	1
Chuck	1

Glacier	1
Nimbus	1
Sprout	1
Miley	1
Geoff	1
Smiley	1
Covach	1
Andy	1
Strider	1
Tuco	1
Gunner	1
Mabel	1
Lorelei	1
Chef	1
Hanz	1
Jeremy	1
Arya	1
Bookstore	1
Travis	1
Kuyu	1
Blakely	1
Iggy	1
Genevieve	1
Brat	1
Jo	1
Timber	1
Harry	1
Tycho	1
Jarod	1
Ralph	1
Marlee	1
Dudley	1
Ricky	1
Pepper	1
Henry	1
Tripp	1
Kulet	1
Jameson	1
Sparky	1
Brudge	1
Rudy	1
Liam	1
Kramer	1
Donny	1
Ulysses	1
Godzilla	1
Boots	1
Holly	1

Gin	1
Jett	1
Colin	1
Keet	1
Ike	1
Tilly	1
Lipton	1
Kingsley	1
Beya	1
Sailor	1
Zeek	1
Deacon	1
Ralphson	1
Huxley	1
Maxwell	1
Mairi	1
Binky	1
Superpup	1
Skye	1
Spanky	1
Glenn	1
William	1
Bones	1
Toffee	1
Trigger	1
Jimbo	1
Tyrus	1
Sweets	1
Florence	1
Brady	1
Chubbs	1
Sid	1
Pluto	1
Beckham	1
Dot	1
Nugget	1
Odin	1
Crawford	1
Chuq	1
Taz	1
Barclay	1
Rooney	1
Swagger	1
Dwight	1
Snickers	1
Marq	1
Tassy	1
Coopson	1

```
Pancake        1
all            1
Tiger          1
Tobi           1
Bronte         1
Dex            1
Jaycob         1
old            1
Durg           1
Antony         1
Dobby          1
Mollie         1
Kayla          1
Zooey          1
Olaf           1
Remy           1
Rizzo          1
Monkey         1
Ember          1
Snicku         1
Nigel          1
Koko           1
Lucia          1
Gilbert        1
Jazzy          1
Andru          1
Humphrey       1
Burt           1
Maks           1
Ralphy         1
Todo           1
Bloop          1
Joey           1
Mingus         1
Edgar          1
Leonidas       1
Tess           1
Stella         1
Rilo           1
Berb           1
Kirk           1
Vince          1
Hermione       1
Baron          1
Ralphie        1
Shakespeare    1
Mona           1
Tove           1
```

Jed	1
Eevee	1
Dixie	1
Naphaniel	1
Eugene	1
Cilantro	1
Kollin	1
Pete	1
Sojourner	1
Mac	1
Hamrick	1
Perry	1
Tuck	1
Brandonald	1
Reptar	1
Longfellow	1
Gordon	1
Rumpole	1
Carll	1
Harnold	1
Josep	1
Corey	1
Link	1
Lulu	1
Sweet	1
Miguel	1
Obi	1
Rascal	1
Jareld	1
Kanu	1
Mutt	1
General	1
Brownie	1
Angel	1
Monster	1
Fillup	1
Edmund	1
Obie	1
Ester	1
Tonks	1
Brandi	1
Pupcasso	1
Sandra	1
by	1
Timofy	1
Coleman	1
Alfy	1
Nida	1

```
this           1
Dido           1
Major          1
Benny          1
Kenzie         1
Marvin         1
Lacy           1
Pilot          1
Freddery       1
Autumn         1
Pubert         1
Bowie          1
Lupe           1
Vinscent       1
Mya            1
Molly          1
Wishes         1
Newt           1
Frönq          1
Bert           1
Charl          1
Halo           1
Devón          1
Ivar           1
Chaz           1
Robin          1
Ed             1
Norman         1
Rupert         1
Arlen          1
Storkson       1
Zara           1
Milky          1
Simba          1
Harlso         1
Yoda           1
Rose           1
Combo          1
Lambeau        1
Name: name, Length: 957, dtype: int64
```

```
In [178]: twitter_archive['source'].value_counts()
```

```
Out[178]: <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
```

```
In [179]: twitter_archive['rating_numerator'].value_counts()

Out[179]: 12      558
          11      464
          10      461
          13      351
          9       158
          8       102
          7       55
          14      54
          5       37
          6       32
          3       19
          4       17
          1       9
          2       9
          420     2
          0       2
          15      2
          75      2
          80      1
          20      1
          24      1
          26      1
          44      1
          50      1
          60      1
          165     1
          84      1
          88      1
          144     1
          182     1
          143     1
          666     1
          960     1
          1776    1
          17      1
          27      1
          45      1
          99      1
          121     1
          204     1
          Name: rating_numerator, dtype: int64

In [180]: twitter_archive['rating_denominator'].value_counts()

Out[180]: 10     2333
          11     3
```

```
50     3
80     2
20     2
2      1
16     1
40     1
70     1
15     1
90     1
110    1
120    1
130    1
150    1
170    1
7      1
0      1
Name: rating_denominator, dtype: int64
```

```
In [181]: twitter_archive['doggo'].value_counts().sort_values(ascending=False)
```

```
Out[181]: None     2259
          doggo    97
          Name: doggo, dtype: int64
```

```
In [182]: twitter_archive['floofer'].value_counts().sort_values(ascending=False)
```

```
Out[182]: None       2346
          floofer    10
          Name: floofer, dtype: int64
```

```
In [183]: twitter_archive['pupper'].value_counts().sort_values(ascending=False)
```

```
Out[183]: None      2099
          pupper    257
          Name: pupper, dtype: int64
```

```
In [184]: twitter_archive['puppo'].value_counts().sort_values(ascending=False)
```

```
Out[184]: None     2326
          puppo    30
          Name: puppo, dtype: int64
```

```
In [185]: twitter_data.info()
          image_predictions.head(5) # check this through excel for easy visual analysis */
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 2340 entries, 0 to 2339
Data columns (total 3 columns):
tweet_id          2340 non-null int64
```

```
retweet_count     2340 non-null int64
favorite_count    2340 non-null int64
dtypes: int64(3)
memory usage: 73.1+ KB
```

Out[185]:

	tweet_id	jpg_url	img_num	
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	1	redbo
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	1	Germa
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	1	Rhode
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	1	minia

Quality issues -

1) Variable 'name' within twitter_archive file is noted to have None values which needs to be replaced by Nan
2) Variable "doggo","puppuer","puppo","floofer" have "None" seems incorrect based on the text columns
3) Source variable needs clean up, html tags needs to be removed.
4) Name variable is noticed to have articles like ("The","A","AN") which does not reflect the right names.
5) Few tweets captured from API have missing tweets which are captured in the exception
6) There are certain tweets which are not related to dogs, noted to have tweets of cats etc.
7) Timestamp is in string format, needs to be converted to timestamp
8) We need to remove all the retweets that are within the data
9) drop all tweets prior to August 1st, 2017
10) Timestamp needs to be converted from String to Timeformat

Tidiness issues -

1) Additional data via API and Predictions to be merged to the archived data and have single master data
2) in_reply_to_status_id and in_reply_to_user_id variable in Archive file are mostly NaN values, which is not much required for analysis
3) Single varible to depict the dog stage is needed and remove the individual variables "doggo","puppuer","puppo","floofer"

0.1.3 Cleaning

```
In [186]: # Take a copy of the data that needs to be cleaned
          twitter_data_clean = twitter_data.copy()
          twitter_archive_clean = twitter_archive.copy()
          image_predictions_clean = image_predictions.copy()
```

```
In [187]: # Define
          # Fix Tidiness issues, merge data as a single table

          # Code
          twitter_archive_clean = pd.merge(twitter_archive_clean, twitter_data_clean,how = 'inne
          twitter_archive_clean = pd.merge(twitter_archive_clean, image_predictions_clean,how =

          # Test
          twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2067 entries, 0 to 2066
Data columns (total 30 columns):
tweet_id                      2067 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2067 non-null object
source                        2067 non-null object
text                          2067 non-null object
retweeted_status_id           75 non-null float64
retweeted_status_user_id      75 non-null float64
retweeted_status_timestamp    75 non-null object
expanded_urls                 2067 non-null object
rating_numerator              2067 non-null int64
rating_denominator            2067 non-null int64
name                          2067 non-null object
doggo                         2067 non-null object
floofer                       2067 non-null object
pupper                        2067 non-null object
puppo                         2067 non-null object
retweet_count                 2067 non-null int64
favorite_count                2067 non-null int64
jpg_url                       2067 non-null object
img_num                       2067 non-null int64
p1                            2067 non-null object
p1_conf                       2067 non-null float64
p1_dog                        2067 non-null bool
p2                            2067 non-null object
p2_conf                       2067 non-null float64
p2_dog                        2067 non-null bool
p3                            2067 non-null object
p3_conf                       2067 non-null float64
```

```
p3_dog                        2067 non-null bool
dtypes: bool(3), float64(7), int64(6), object(14)
memory usage: 458.2+ KB
```

In [188]:
```
#Define
# Filter for tweets after August 2017, first convert to timestamp format to do this

# Code
twitter_archive_clean['timestamp']=pd.to_datetime(twitter_archive_clean['timestamp'])
twitter_archive_clean=twitter_archive_clean[twitter_archive_clean['timestamp']<='2017-

# Test
twitter_archive_clean['timestamp'].max()
```

Out[188]:
```
Timestamp('2017-08-01 16:23:56')
```

In [189]:
```
# Define
# Name variable is noticed to have articles like ("The","A","AN") which does not refle
pd.set_option('max_colwidth', -1)
#Code
temp=twitter_archive_clean[twitter_archive_clean['name'].isin(['O','an','the'])]
temp[['tweet_id','name','text']]
```

Out[189]:

	tweet_id	name	
605	778396591732486144	an	RT @dog_rates: This is an East African Chalupa Seal. We
619	776201521193218049	O	This is O'Malley. That is how he sleeps. Doesn't care w
833	746369468511756288	an	This is an Iraqi Speed Kangaroo. It is not a dog. Pleas
1137	703041949650034688	an	This is an East African Chalupa Seal. We only rate dogs
1281	690360449368465409	the	Stop sending in lobsters. This is the final warning. We
1347	685943807276412928	the	This is the newly formed pupper a capella group. They'r
1523	677269281705472000	the	This is the happiest pupper I've ever seen. 10/10 would
1540	676613908052996102	the	This is the saddest/sweetest/best picture I've been sen
1753	671561002136281088	the	This is the best thing I've ever seen so spread it like
1917	668636665813057536	an	This is an Irish Rigatoni terrier named Berta. Complete
2044	666337882303524864	an	This is an extremely rare horned Parthenon. Not amused.
2046	666287406224695296	an	This is an Albanian 3 1/2 legged Episcopalian. Loves w
2056	666063827256086533	the	This is the happiest dog you will ever see. Very commit
2057	666058600524156928	the	Here is the Rand Paul of retrievers folks! He's probabl
2060	666051853826850816	an	This is an odd dog. Hard on the outside but loving on t

In [190]:
```
# Define
# Extract the dog stage from the text column, as the variables in the indivial variabl
# places, also remove the individual variable

# Code
twitter_archive_clean['dog_Stage']=twitter_archive_clean[['doggo', 'floofer', 'pupper'
twitter_archive_clean['dog_Stage']=twitter_archive_clean['dog_Stage'].replace(regex=r'
twitter_archive_clean['dog_Stage']=twitter_archive_clean['dog_Stage'].replace(regex=r'
```

```
          twitter_archive_clean['dog_Stage']=twitter_archive_clean['dog_Stage'].replace(regex=r'

          # Test
          twitter_archive_clean['dog_Stage'].unique()
```

```
Out[190]: array(['', 'doggo', 'puppo', 'pupper', 'floofer', 'doggo,puppo',
                 'doggo,floofer', 'doggo,pupper'], dtype=object)
```

```
In [191]: #Define
          # drop unwanted columns like doggo, floofer,pupper,puppo

          #Code
          twitter_archive_clean.drop(['doggo','floofer','pupper','puppo'],axis=1,inplace=True)

          #Test -- Visually confirm if the columns have been dropped
          twitter_archive_clean.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 2067 entries, 0 to 2066
Data columns (total 27 columns):
tweet_id                      2067 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     2067 non-null datetime64[ns]
source                        2067 non-null object
text                          2067 non-null object
retweeted_status_id           75 non-null float64
retweeted_status_user_id      75 non-null float64
retweeted_status_timestamp    75 non-null object
expanded_urls                 2067 non-null object
rating_numerator              2067 non-null int64
rating_denominator            2067 non-null int64
name                          2067 non-null object
retweet_count                 2067 non-null int64
favorite_count                2067 non-null int64
jpg_url                       2067 non-null object
img_num                       2067 non-null int64
p1                            2067 non-null object
p1_conf                       2067 non-null float64
p1_dog                        2067 non-null bool
p2                            2067 non-null object
p2_conf                       2067 non-null float64
p2_dog                        2067 non-null bool
p3                            2067 non-null object
p3_conf                       2067 non-null float64
p3_dog                        2067 non-null bool
dog_Stage                     2067 non-null object
dtypes: bool(3), datetime64[ns](1), float64(7), int64(6), object(10)
```

```
memory usage: 409.8+ KB
```

```
In [192]: #Define
          # Certain data are not related to dogs, which are visually identified and are deleted

          #code
          (twitter_archive_clean[twitter_archive_clean['tweet_id']==746369468511756288])
          twitter_archive_clean=twitter_archive_clean[~twitter_archive_clean['tweet_id'].isin([7
          # Test
          twitter_archive_clean[twitter_archive_clean['tweet_id']==690360449368465409]
```

```
Out[192]: Empty DataFrame
          Columns: [tweet_id, in_reply_to_status_id, in_reply_to_user_id, timestamp, source, tex
          Index: []
```

```
In [193]: #Define
          # Clean the source attribute, remove HTML tags

          # Code
          twitter_archive_clean1=twitter_archive_clean.copy()
          twitter_archive_clean1['source']=twitter_archive_clean['source'].str.extract(r'[?:<\w*
          twitter_archive_clean1.head(1)

          # Test
          twitter_archive_clean1['source'].unique()
```

```
/opt/conda/lib/python3.6/site-packages/ipykernel_launcher.py:6: FutureWarning: currently extract
```

```
Out[193]: array(['Twitter for iPhone', 'Twitter Web Client', 'TweetDeck'], dtype=object)
```

```
In [194]: #Define
          # Drop all retweets from the analysis data

          # Code
          temp=twitter_archive_clean1[twitter_archive_clean1['retweeted_status_id'].notnull()]
          twitter_archive_clean1=twitter_archive_clean1.drop(temp.index[0:])

          # Test
          twitter_archive_clean1['retweeted_status_id'].notnull().any()
```

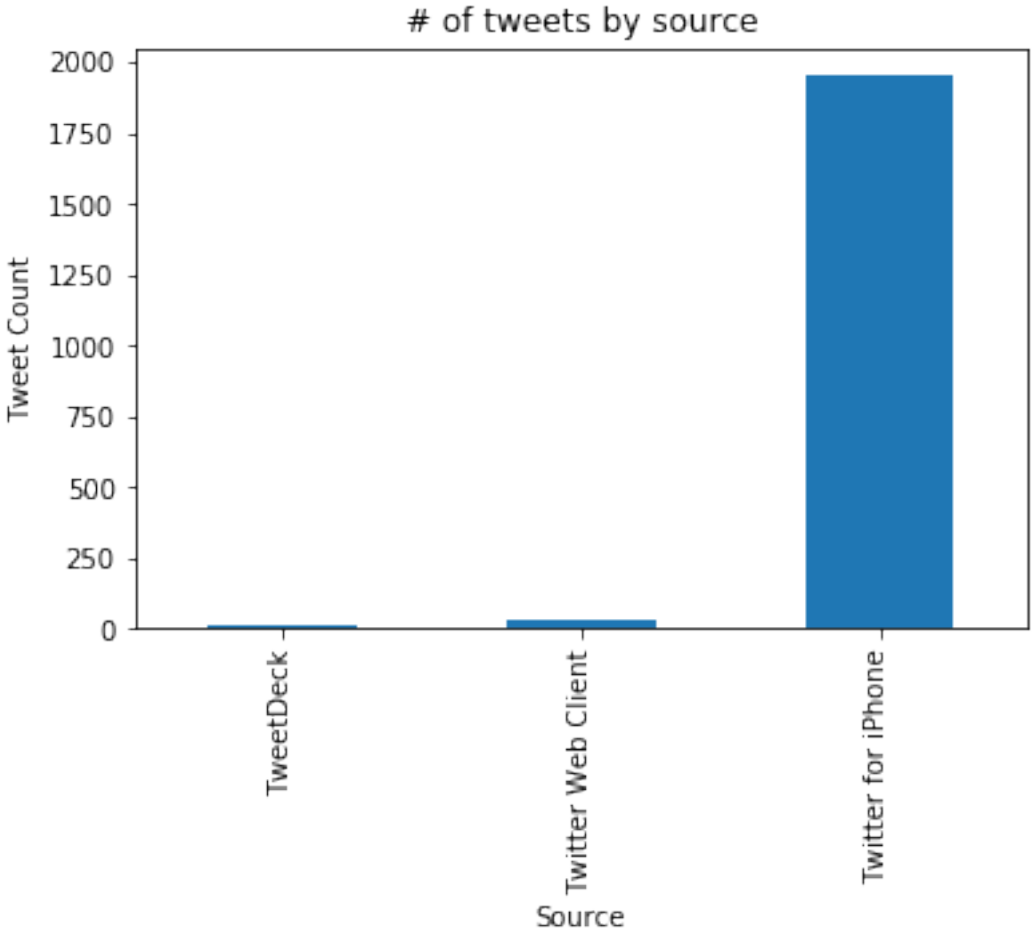
```
Out[194]: False
```

```
In [195]: # Store clean data into the master file
          twitter_archive_clean1.to_csv("twitter_archive_master.csv")
```

0.1.4 Analyse and Visualize

In [196]:
```
# Analyse on what sources contributed to the number of tweets
# Visualize
df=twitter_archive_clean1.groupby('source')['tweet_id'].count()
df
df.plot.bar()
plt.xlabel('Source')
plt.ylabel('Tweet Count')
plt.title('# of tweets by source')
```

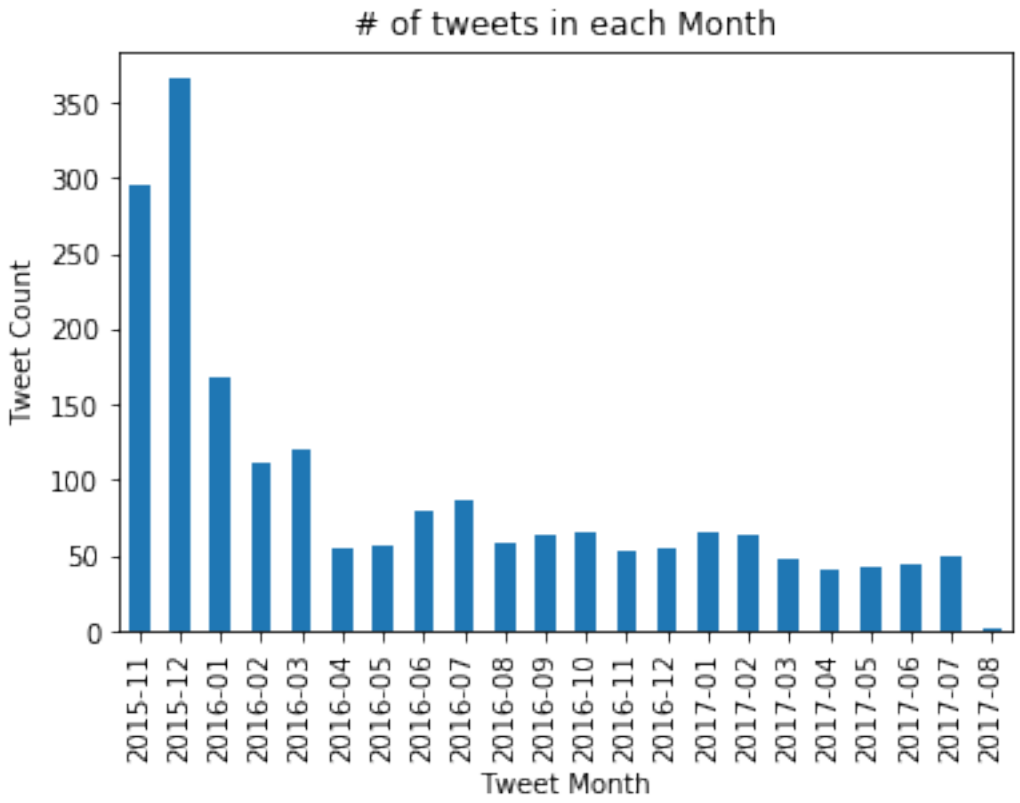
Out[196]:
```
Text(0.5,1,'# of tweets by source')
```



In [197]:
```
# Analyse the number of tweets in the time period captured
df = pd.DataFrame(columns=['date','Tweet_id'])
df['date']=twitter_archive_clean1['timestamp'].dt.to_period('M')
df['Tweet_id']=twitter_archive_clean1['tweet_id']
temp=df.groupby('date')['Tweet_id'].count()
```

```
temp.plot.bar()
plt.xlabel('Tweet Month')
plt.ylabel('Tweet Count')
plt.title('# of tweets in each Month')
```

Out[197]: Text(0.5,1,'# of tweets in each Month')

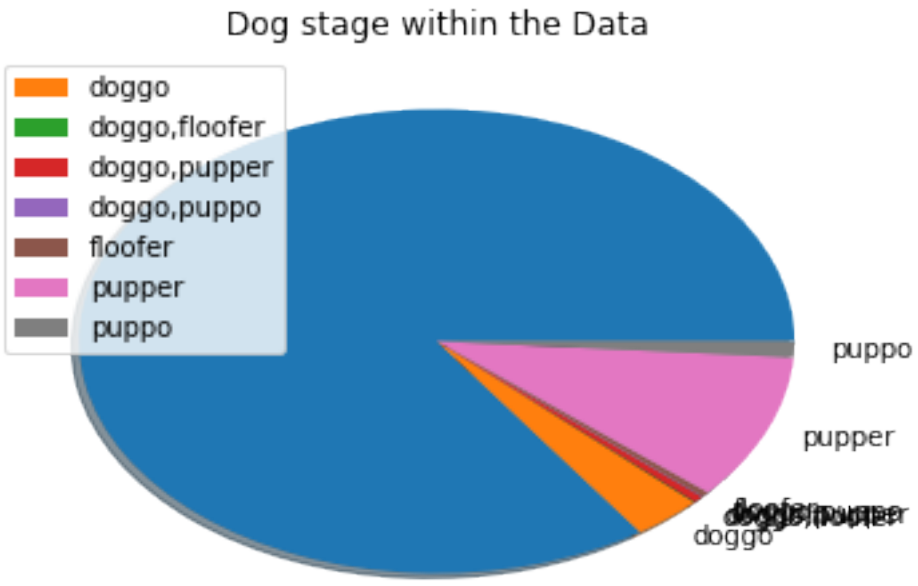


In [198]: twitter_archive_clean1.info()

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1990 entries, 0 to 2066
Data columns (total 27 columns):
tweet_id                      1990 non-null int64
in_reply_to_status_id         23 non-null float64
in_reply_to_user_id           23 non-null float64
timestamp                     1990 non-null datetime64[ns]
source                        1990 non-null object
text                          1990 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                 1990 non-null object
```

```
rating_numerator              1990 non-null int64
rating_denominator            1990 non-null int64
name                          1990 non-null object
retweet_count                 1990 non-null int64
favorite_count                1990 non-null int64
jpg_url                       1990 non-null object
img_num                       1990 non-null int64
p1                            1990 non-null object
p1_conf                       1990 non-null float64
p1_dog                        1990 non-null bool
p2                            1990 non-null object
p2_conf                       1990 non-null float64
p2_dog                        1990 non-null bool
p3                            1990 non-null object
p3_conf                       1990 non-null float64
p3_dog                        1990 non-null bool
dog_Stage                     1990 non-null object
dtypes: bool(3), datetime64[ns](1), float64(7), int64(6), object(10)
memory usage: 394.5+ KB
```

```
In [294]: df = twitter_archive_clean1[['dog_Stage']]
          temp=df.groupby(['dog_Stage']).size().reset_index(name='count')
          temp.iloc[:,0]
          plt.pie(temp.iloc[:,1],labels=temp.iloc[:,0],shadow=True);
          plt.title('Dog stage within the Data')
          plt.legend();
          #type(temp)
```



```
In [ ]:
```