# GATHERING DATA

In [1]:

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
%matplotlib inline
import seaborn as sns
sns.set_style('darkgrid')
import requests
import os
# from bs4 import BeautifulSoup
import json
from sqlalchemy import create_engine
import re
```

In [2]:

```
image_preidiction = pd.read_csv('image-predictions.tsv', sep='\t')
twitter_archive_enhanced = pd.read_csv('twitter-archive-enhanced.csv')
```

In [3]:

```
image_preidiction.head()
```

Out[3]:

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

In [4]:

```
twitter_archive_enhanced.head(1)
```

Out[4]:

| | tweet_id | in_reply_to_status_id | in_reply_to_user_id | timestamp | |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | NaN | NaN | 2017-08-01 16:23:56 +0000 | href="http://twitter.c |

In [5]:

```
url = 'https://video.udacity-data.com/topher/2018/November/5be5fb7d_tweet-json/tweet-json.t
respons = requests.get(url)

with open('tweet_json.txt', 'wb') as file:
    file.write(respons.content)
```

In [6]:

```
with open('tweet_json.txt', 'r') as f:
    print(json.loads(f.readline()))
```

{'created_at': 'Tue Aug 01 16:23:56 +0000 2017', 'id': 892420643555336193,
'id_str': '892420643555336193', 'full_text': "This is Phineas. He's a mystic
al boy. Only ever appears in the hole of a donut. 13/10 https://t.co/MgUWQ76
dJU", (https://t.co/MgUWQ76dJU",) 'truncated': False, 'display_text_range':
 [0, 85], 'entities': {'hashtags': [], 'symbols': [], 'user_mentions': [],
 'urls': [], 'media': [{'id': 892420639486877696, 'id_str': '892420639486877
696', 'indices': [86, 109], 'media_url': 'http://pbs.twimg.com/media/DGKD1-b
XoAAIAUK.jpg', 'media_url_https': 'https://pbs.twimg.com/media/DGKD1-bXoAAIA
UK.jpg', 'url': 'https://t.co/MgUWQ76dJU', 'display_url': 'pic.twitter.com/M
gUWQ76dJU', 'expanded_url': 'https://twitter.com/dog_rates/status/8924206435
55336193/photo/1', 'type': 'photo', 'sizes': {'large': {'w': 540, 'h': 528,
 'resize': 'fit'}, 'thumb': {'w': 150, 'h': 150, 'resize': 'crop'}, 'small':
{'w': 540, 'h': 528, 'resize': 'fit'}, 'medium': {'w': 540, 'h': 528, 'resiz
e': 'fit'}}}]}, 'extended_entities': {'media': [{'id': 892420639486877696,
 'id_str': '892420639486877696', 'indices': [86, 109], 'media_url': 'http://
pbs.twimg.com/media/DGKD1-bXoAAIAUK.jpg', 'media_url_https': 'https://pbs.tw
img.com/media/DGKD1-bXoAAIAUK.jpg', 'url': 'https://t.co/MgUWQ76dJU', 'displ
ay_url': 'pic.twitter.com/MgUWQ76dJU', 'expanded_url': 'https://twitter.com/
dog_rates/status/892420643555336193/photo/1', 'type': 'photo', 'sizes': {'la
rge': {'w': 540, 'h': 528, 'resize': 'fit'}, 'thumb': {'w': 150, 'h': 150,
 'resize': 'crop'}, 'small': {'w': 540, 'h': 528, 'resize': 'fit'}, 'mediu
m': {'w': 540, 'h': 528, 'resize': 'fit'}}}]}, 'source': '<a href="http://tw
itter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>', 'in_reply
_to_status_id': None, 'in_reply_to_status_id_str': None, 'in_reply_to_user_i
d': None, 'in_reply_to_user_id_str': None, 'in_reply_to_screen_name': None,
 'user': {'id': 4196983835, 'id_str': '4196983835', 'name': 'WeRateDogs™ (au
thor)', 'screen_name': 'dog_rates', 'location': 'DM YOUR DOGS, WE WILL RAT
E', 'description': '#1 Source for Professional Dog Ratings | STORE: @ShopWeR
ateDogs | IG, FB & SC: WeRateDogs MOBILE APP: @GoodDogsGame | Business: dogr
atingtwitter@gmail.com', 'url': 'https://t.co/N7sNNHAEXS', 'entities': {'ur
l': {'urls': [{'url': 'https://t.co/N7sNNHAEXS', 'expanded_url': 'http://wer
atedogs.com', 'display_url': 'weratedogs.com', 'indices': [0, 23]}]}, 'descr
iption': {'urls': []}}, 'protected': False, 'followers_count': 3200889, 'fri
ends_count': 104, 'listed_count': 2784, 'created_at': 'Sun Nov 15 21:41:29 +
0000 2015', 'favourites_count': 114031, 'utc_offset': None, 'time_zone': Non
e, 'geo_enabled': True, 'verified': True, 'statuses_count': 5288, 'lang': 'e
n', 'contributors_enabled': False, 'is_translator': False, 'is_translation_e
nabled': False, 'profile_background_color': '000000', 'profile_background_im
age_url': 'http://abs.twimg.com/images/themes/theme1/bg.png', 'profile_backg
round_image_url_https': 'https://abs.twimg.com/images/themes/theme1/bg.png',
'profile_background_tile': False, 'profile_image_url': 'http://pbs.twimg.co
m/profile_images/861415328504569856/R2xOOfwe_normal.jpg', 'profile_image_url
_https': 'https://pbs.twimg.com/profile_images/861415328504569856/R2xOOfwe_n
ormal.jpg', 'profile_banner_url': 'https://pbs.twimg.com/profile_banners/419
6983835/1501129017', 'profile_link_color': 'F5ABB5', 'profile_sidebar_border
_color': '000000', 'profile_sidebar_fill_color': '000000', 'profile_text_col
or': '000000', 'profile_use_background_image': False, 'has_extended_profil
e': True, 'default_profile': False, 'default_profile_image': False, 'followi
ng': True, 'follow_request_sent': False, 'notifications': False, 'translator
_type': 'none'}, 'geo': None, 'coordinates': None, 'place': None, 'contribut
ors': None, 'is_quote_status': False, 'retweet_count': 8853, 'favorite_coun
t': 39467, 'favorited': False, 'retweeted': False, 'possibly_sensitive': Fal
se, 'possibly_sensitive_appealable': False, 'lang': 'en'}

In [7]:

```
df_list = []
json_errors = []

with open('tweet_json.txt', 'r') as f:

    for x in f:
        try:
            timestampe = json.loads(x)['created_at']
            tweet_id = json.loads(x)['id']
            tweet_text = json.loads(x)['full_text']
            tweet_image_url = json.loads(x)['extended_entities']['media'][0]['expanded_url'
            source = json.loads(x)['source']
            retweet_count = json.loads(x)['retweet_count']
            favorite_count = json.loads(x)['favorite_count']
            follwers_count = json.loads(x)['user']['followers_count']

            dict = {
                'timestampe': timestampe,
                'tweet_id': tweet_id,
                'tweet_text': tweet_text,
                'tweet_image_url': tweet_image_url,
                'source': source,
                'retweet_count': retweet_count,
                'favorite_count': favorite_count,
                'follwers_count': follwers_count
            }

            df_list.append(dict)

        except:
            json_errors.append(json.loads(x))
```

In [8]:

```
for x in json_errors:

    timestampe = x['created_at']
    tweet_id = x['id']
    tweet_text = x['full_text']
    tweet_image_url = 'NAN'
    source = x['source']
    retweet_count = x['retweet_count']
    favorite_count = x['favorite_count']
    follwers_count = x['user']['followers_count']

    dict = {
                'timestampe': timestampe,
                'tweet_id': tweet_id,
                'tweet_text': tweet_text,
                'tweet_image_url': tweet_image_url,
                'source': source,
                'retweet_count': retweet_count,
                'favorite_count': favorite_count,
                'follwers_count': follwers_count
            }

    df_list.append(dict)
```

In [9]:

```
len(df_list)
```

Out[9]:

```
2354
```

In [10]:

```
weRateDog_df = pd.DataFrame(df_list, columns=['timestampe', 'tweet_id', 'tweet_text', 'twee
                                    'favorite_count', 'follwers_count'])
```

In [11]:

```
weRateDog_df.head()
```

Out[11]:

| | timestampe | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|---|
| **0** | Tue Aug 01 16:23:56 +0000 2017 | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | href="http://twitter. |
| **1** | Tue Aug 01 00:17:27 +0000 2017 | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | href="http://twitter. |
| **2** | Mon Jul 31 00:18:03 +0000 2017 | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin | https://twitter.com/dog_rates/status/891815181... | href="http://twitter. |

In [12]:

```
weRateDog_df.to_csv('WeRateDog.csv', index=False)
```

# Access

In [13]:

```
weRateDog_df
```

Out[13]:

| | timestampe | tweet_id | tweet_text | tweet_imag |
|---|---|---|---|---|
| **0** | Tue Aug 01 16:23:56 +0000 2017 | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420 |
| **1** | Tue Aug 01 00:17:27 +0000 2017 | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177 |
| **2** | Mon Jul 31 00:18:03 +0000 2017 | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815 |
| **3** | Sun Jul 30 15:58:51 +0000 2017 | 891689557279858688 | This is Darla. She commenced a snooze mid meal... | https://twitter.com/dog_rates/status/891689 |
| **4** | Sat Jul 29 16:00:24 +0000 2017 | 891327558926688256 | This is Franklin. He would like you to stop ca... | https://twitter.com/dog_rates/status/891327 |
| **...** | ... | ... | ... | |
| **2349** | Tue Dec 01 04:44:10 +0000 2015 | 671550332464455680 | After 22 minutes of careful deliberation this ... | |
| **2350** | Thu Nov 26 01:11:28 +0000 2015 | 669684865554620416 | After countless hours of research and hundreds... | |
| **2351** | Tue Nov 24 01:42:25 +0000 2015 | 668967877119254528 | 12/10 good shit Bubka\n@wane15 | |
| **2352** | Mon Nov 23 00:30:28 +0000 2015 | 668587383441514497 | Never forget this vine. You will not stop watc... | |
| **2353** | Wed Nov 18 20:02:51 +0000 2015 | 667070482143944705 | After much debate this dog is being upgraded t... | |

2354 rows × 8 columns

In [14]:

```
weRateDog_df.tweet_text
```

Out[14]:

```
0       This is Phineas. He's a mystical boy. Only eve...
1       This is Tilly. She's just checking pup on you....
2       This is Archie. He is a rare Norwegian Pouncin...
3       This is Darla. She commenced a snooze mid meal...
4       This is Franklin. He would like you to stop ca...
                              ...
2349    After 22 minutes of careful deliberation this ...
2350    After countless hours of research and hundreds...
2351                        12/10 good shit Bubka\n@wane15
2352    Never forget this vine. You will not stop watc...
2353    After much debate this dog is being upgraded t...
Name: tweet_text, Length: 2354, dtype: object
```

In [15]:

```
weRateDog_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 8 columns):
 #   Column           Non-Null Count  Dtype 
---  ------           --------------  ----- 
 0   timestampe       2354 non-null   object
 1   tweet_id         2354 non-null   int64 
 2   tweet_text       2354 non-null   object
 3   tweet_image_url  2354 non-null   object
 4   source           2354 non-null   object
 5   retweet_count    2354 non-null   int64 
 6   favorite_count   2354 non-null   int64 
 7   follwers_count   2354 non-null   int64 
dtypes: int64(4), object(4)
memory usage: 147.2+ KB
```

In [16]:

```
weRateDog_df.source.head()
```

Out[16]:

```
0    <a href="http://twitter.com/download/iphone" r...
1    <a href="http://twitter.com/download/iphone" r...
2    <a href="http://twitter.com/download/iphone" r...
3    <a href="http://twitter.com/download/iphone" r...
4    <a href="http://twitter.com/download/iphone" r...
Name: source, dtype: object
```

## Quality Issues

- invalid time format for timestamp column
- time tweeted in time stamp
- day of the week in timestampe
- ratting of dogs in tweet text
- extre url links for imagse in text
- dog stage in tweet text

- invalid format for source column
- inconsistent tweet text containing retweet instead of actual tweet

### Tidiness Issues

- inaccuare timestampe colunms data type
- inaccuare dog rating data type

# Clean

#### Define

- **Create a sepreate columns for days of the week using `appy()` and `.split()` mathods to target the first index**
- **append the new columns to the weRatedog data frame**
- **replace abrivation of day with full naem**

#### Code

In [17]:

```
weRateDog_df.timestampe.head()
```

Out[17]:

```
0    Tue Aug 01 16:23:56 +0000 2017
1    Tue Aug 01 00:17:27 +0000 2017
2    Mon Jul 31 00:18:03 +0000 2017
3    Sun Jul 30 15:58:51 +0000 2017
4    Sat Jul 29 16:00:24 +0000 2017
Name: timestampe, dtype: object
```

In [18]:

```
day_tweeted = weRateDog_df.timestampe.apply(lambda x: x.split()[0])
```

In [19]:

```
weRateDog_df['day_tweeted'] = day_tweeted.astype(str)
```

In [20]:

```
days_abv_correction = [('sunday', 'Sun'), ('saturday', 'Sat'), ('monday', 'Mon'),
                       ('tuesday', 'Tue'), ('Thursday', 'Thu'), ('wednessday', 'Wed'), ('fr

for days in days_abv_correction:
    weRateDog_df['day_tweeted'].replace(days[1], days[0], inplace=True)
```

#### Test

In [21]:

```
weRateDog_df.head(3)
```

Out[21]:

| | timestampe | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|---|
| **0** | Tue Aug 01 16:23:56 +0000 2017 | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | hre |
| **1** | Tue Aug 01 00:17:27 +0000 2017 | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | hre |
| **2** | Mon Jul 31 00:18:03 +0000 2017 | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | hre |

### Define

- **Create seperate columns for the time tweeted using `apply()` mathod and `split()` function to on timestamp columns**
- **append the new columns to the weRatedog data frame**

### Code

In [22]:

```
time_tweeted = weRateDog_df.timestampe.apply(lambda x: x.split()[3])
```

In [23]:

```
weRateDog_df['time_tweeted'] = time_tweeted
```

### Test

In [24]:

```
weRateDog_df.head()
```

Out[24]:

| | timestampe | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|---|
| **0** | Tue Aug 01 16:23:56 +0000 2017 | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | hr |
| **1** | Tue Aug 01 00:17:27 +0000 2017 | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | hr |
| **2** | Mon Jul 31 00:18:03 +0000 2017 | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | hr |
| **3** | Sun Jul 30 15:58:51 +0000 2017 | 891689557279858688 | This is Darla. She commenced a snooze mid meal... | https://twitter.com/dog_rates/status/891689557... | hr |
| **4** | Sat Jul 29 16:00:24 +0000 2017 | 891327558926688256 | This is Franklin. He would like you to stop ca... | https://twitter.com/dog_rates/status/891327558... | hr |

### Define

- **Change The invalide timestampe columns formate to [ `yyyy-mm-dd` ] using `apply()` mathod for iterating throug the timestampe columns**
- **using the `split()` string function to convet each valuse to list**
- **converting the the arrary to numpy array by `np.array` and target the index of the numpy array using `np.r_[]`**
- **join the array by `-` using `join()` string function**
- **add coulmn to weRatedog dataframe and delete `timestampe` column**

### Code

In [25]:

```
date_tweeted = weRateDog_df.timestampe.apply(lambda x: "-".join(np.array(x.split())[np.r_[5
```

In [26]:

```
weRateDog_df['date_tweeted'] = date_tweeted
weRateDog_df = weRateDog_df.drop(columns=['timestampe'])
```

### Test

In [27]:

```
weRateDog_df.head()
```

Out[27]:

| | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|
| 0 | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | href="http://twitte |
| 1 | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | href="http://twitte |
| 2 | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | href="http://twitte |
| 3 | 891689557279858688 | This is Darla. She commenced a snooze mid meal... | https://twitter.com/dog_rates/status/891689557... | href="http://twitte |
| 4 | 891327558926688256 | This is Franklin. He would like you to stop ca... | https://twitter.com/dog_rates/status/891327558... | href="http://twitte |

### Define

- **targetting ratting of dogs in tweet text using `apply()` function and regular expression `re` for extraction**
- **create seperate columns for numeratore and denominatore**

### Code

In [28]:

```
# rate_valuse = weRateDog_df.tweet_text.str.extract(r'([0-9]?[0-9]?[1234567890]?[.]?./[0-9]
# tweeter_ratings =

numeratore = weRateDog_df.tweet_text.apply(lambda x: re.findall(r'\b\d+\b', x)[0])
denominatore = weRateDog_df.tweet_text.apply(lambda x: re.findall(r'\b\d+\b', x)[-1])
```

In [29]:

```
weRateDog_df['dog_ratings_numeratore'] = numeratore
weRateDog_df['dog_ratings_denominatore'] = denominatore
```

### Test

In [30]:

```
weRateDog_df.head()
```

Out[30]:

| | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|
| 0 | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | href="http://twitter.com/download/ |
| 1 | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | href="http://twitter.com/download/ |
| 2 | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin | https://twitter.com/dog_rates/status/891815181... | href="http://twitter.com/download/ |

### Define

- **extre url links for imagse in text using `apply()` function and regulare expresion `re`**
- **create a seperate columns for extrar_tweet_url for the extration of url line**

### Code

In [31]:

```
tweet_text_url = weRateDog_df.tweet_text.apply(lambda x: ''.join(re.findall(r'(https?://\S+
```

In [32]:

```
weRateDog_df['tweet_text_url'] = tweet_text_url
```

### Test

In [33]:

```
weRateDog_df.head()
```

Out[33]:

| | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|
| **0** | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | href="http://twitte |
| **1** | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | href="http://twitte |
| **2** | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | href="http://twitte |
| **3** | 891689557279858688 | This is Darla. She commenced a snooze mid meal... | https://twitter.com/dog_rates/status/891689557... | href="http://twitte |
| **4** | 891327558926688256 | This is Franklin. He would like you to stop ca... | https://twitter.com/dog_rates/status/891327558... | href="http://twitte |

### Define

- **exstact dog stage in tweet text using `apply` function to target each valuse text**
- **use list comprehension, `.join()` function and `strip()`**
- **fill empty cell with a dog stage use for any dog called `floof`**

### Code

In [34]:

```
dog_stages = ['doggo', 'pupper', 'puppo', 'floof', 'snoot', 'blep']

dog_stage = weRateDog_df.tweet_text.apply(lambda x: ''.join([dog if dog in x else '' for do
```

In [35]:

```
weRateDog_df['dog_stage'] = dog_stage
```

In [36]:

```
weRateDog_df['dog_stage'] = weRateDog_df['dog_stage'].replace('', 'floof')
weRateDog_df['dog_stage'] = weRateDog_df['dog_stage'].fillna('floof')
```

### Test

In [37]:

```
weRateDog_df.head(50)
```

Out[37]:

| | tweet_id | tweet_text | tweet_image_url | |
|---|---|---|---|---|
| **0** | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | href="http://twitter.com/downloa |
| **1** | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | href="http://twitter.com/downloa |
| **2** | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | href="http://twitter.com/downloa |

In [38]:

```
weRateDog_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 14 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   tweet_id                  2354 non-null   int64 
 1   tweet_text                2354 non-null   object
 2   tweet_image_url           2354 non-null   object
 3   source                    2354 non-null   object
 4   retweet_count             2354 non-null   int64 
 5   favorite_count            2354 non-null   int64 
 6   follwers_count            2354 non-null   int64 
 7   day_tweeted               2354 non-null   object
 8   time_tweeted              2354 non-null   object
 9   date_tweeted              2354 non-null   object
 10  dog_ratings_numeratore    2354 non-null   object
 11  dog_ratings_denominatore  2354 non-null   object
 12  tweet_text_url            2354 non-null   object
 13  dog_stage                 2354 non-null   object
dtypes: int64(4), object(10)
memory usage: 257.6+ KB
```

### Define

- **exstarct the accual sourse of tweet using apply() method split() function**

- **index througe the split() list and use the replace() function to replace [</a] with empty space**
- **delete source function**

### Code

In [39]:

```
twitter_source = weRateDog_df.source.apply(lambda x: x.split('>')[1].replace('</a', ''))
```

In [40]:

```
weRateDog_df['twitter_source'] = twitter_source
```

In [41]:

```
weRateDog_df = weRateDog_df.drop(columns='source')
```

### Test

In [42]:

```
weRateDog_df.head()
```

Out[42]:

| | tweet_id | tweet_text | tweet_image_url | retweet_count |
|---|---|---|---|---|
| **0** | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | 8853 |
| **1** | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | 6514 |
| **2** | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | 4328 |
| **3** | 891689557279858688 | This is Darla. She commenced a snooze mid meal... | https://twitter.com/dog_rates/status/891689557... | 8964 |
| **4** | 891327558926688256 | This is Franklin. He would like you to stop ca... | https://twitter.com/dog_rates/status/891327558... | 9774 |

### Define

- **Create a columns for that identify retweet and tweet**

- **use `.apply()` to tagert each text**
- **list comprehention and indexing to target results**

### Code

In [43]:

```
isRetweet = ['RT']

tweet_type = weRateDog_df.tweet_text.apply(lambda x: ['Retweet' if t in x else 'Tweet' for
```

In [44]:

```
weRateDog_df['tweet_type'] = tweet_type
```

### Test

In [45]:

```
weRateDog_df.head(50)
```

Out[45]:

| | tweet_id | tweet_text | tweet_image_url | retweet_count | favorite_cou |
|---|---|---|---|---|---|
| **0** | 892420643555336193 | This is Phineas. He's a mystical boy. Only eve... | https://twitter.com/dog_rates/status/892420643... | 8853 | 394 |
| **1** | 892177421306343426 | This is Tilly. She's just checking pup on you.... | https://twitter.com/dog_rates/status/892177421... | 6514 | 338 |
| **2** | 891815181378084864 | This is Archie. He is a rare Norwegian Pouncin... | https://twitter.com/dog_rates/status/891815181... | 4328 | 254 |

In [46]:

```
weRateDog_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype 
---  ------                    --------------  ----- 
 0   tweet_id                  2354 non-null   int64 
 1   tweet_text                2354 non-null   object
 2   tweet_image_url           2354 non-null   object
 3   retweet_count             2354 non-null   int64 
 4   favorite_count            2354 non-null   int64 
 5   follwers_count            2354 non-null   int64 
 6   day_tweeted               2354 non-null   object
 7   time_tweeted              2354 non-null   object
 8   date_tweeted              2354 non-null   object
 9   dog_ratings_numeratore    2354 non-null   object
 10  dog_ratings_denominatore  2354 non-null   object
 11  tweet_text_url            2354 non-null   object
 12  dog_stage                 2354 non-null   object
 13  twitter_source            2354 non-null   object
 14  tweet_type                2354 non-null   object
dtypes: int64(4), object(11)
memory usage: 276.0+ KB
```

### Define

- Convert data_tweeted columns data type from string to time data type using `pd.to_datetime()` method

### Code

In [47]:

```
weRateDog_df['date_tweeted'] = pd.to_datetime(weRateDog_df.date_tweeted)
```

### Test

In [48]:

```
weRateDog_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype         
---  ------                    --------------  -----         
 0   tweet_id                  2354 non-null   int64         
 1   tweet_text                2354 non-null   object        
 2   tweet_image_url           2354 non-null   object        
 3   retweet_count             2354 non-null   int64         
 4   favorite_count            2354 non-null   int64         
 5   follwers_count            2354 non-null   int64         
 6   day_tweeted               2354 non-null   object        
 7   time_tweeted              2354 non-null   object        
 8   date_tweeted              2354 non-null   datetime64[ns]
 9   dog_ratings_numeratore    2354 non-null   object        
 10  dog_ratings_denominatore  2354 non-null   object        
 11  tweet_text_url            2354 non-null   object        
 12  dog_stage                 2354 non-null   object        
 13  twitter_source            2354 non-null   object        
```

### Define

- **Convert dog_ratings_numeratore and dog_ratings_denominatore columns from object to integer using `astype()`**

### Code

In [49]:

```
weRateDog_df[['dog_ratings_numeratore', 'dog_ratings_denominatore']] = weRateDog_df[['dog_r
```

### Test

In [50]:

```
weRateDog_df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2354 entries, 0 to 2353
Data columns (total 15 columns):
 #   Column                    Non-Null Count  Dtype         
---  ------                    --------------  -----         
 0   tweet_id                  2354 non-null   int64         
 1   tweet_text                2354 non-null   object        
 2   tweet_image_url           2354 non-null   object        
 3   retweet_count             2354 non-null   int64         
 4   favorite_count            2354 non-null   int64         
 5   follwers_count            2354 non-null   int64         
 6   day_tweeted               2354 non-null   object        
 7   time_tweeted              2354 non-null   object        
 8   date_tweeted              2354 non-null   datetime64[ns]
 9   dog_ratings_numeratore    2354 non-null   int32         
 10  dog_ratings_denominatore  2354 non-null   int32         
 11  tweet_text_url            2354 non-null   object        
 12  dog_stage                 2354 non-null   object        
 13  twitter_source            2354 non-null   object        
```

In [ ]:

```

```

In [51]:

```
weRateDog_df.to_csv('twitter_archive_master.csv', index=False)
```

In [52]:

```
if not os.path.exists('twitter_archive_master.db'):
    engine = create_engine('sqlite:///twitter_archive_master.db')
    weRateDog_df.to_sql('master', engine, index=False)
```

# Explore Data Analysis

## WHAT TYPE OF DOG STAGE HAS MORE LIKES ON AVERAGE

In [53]:

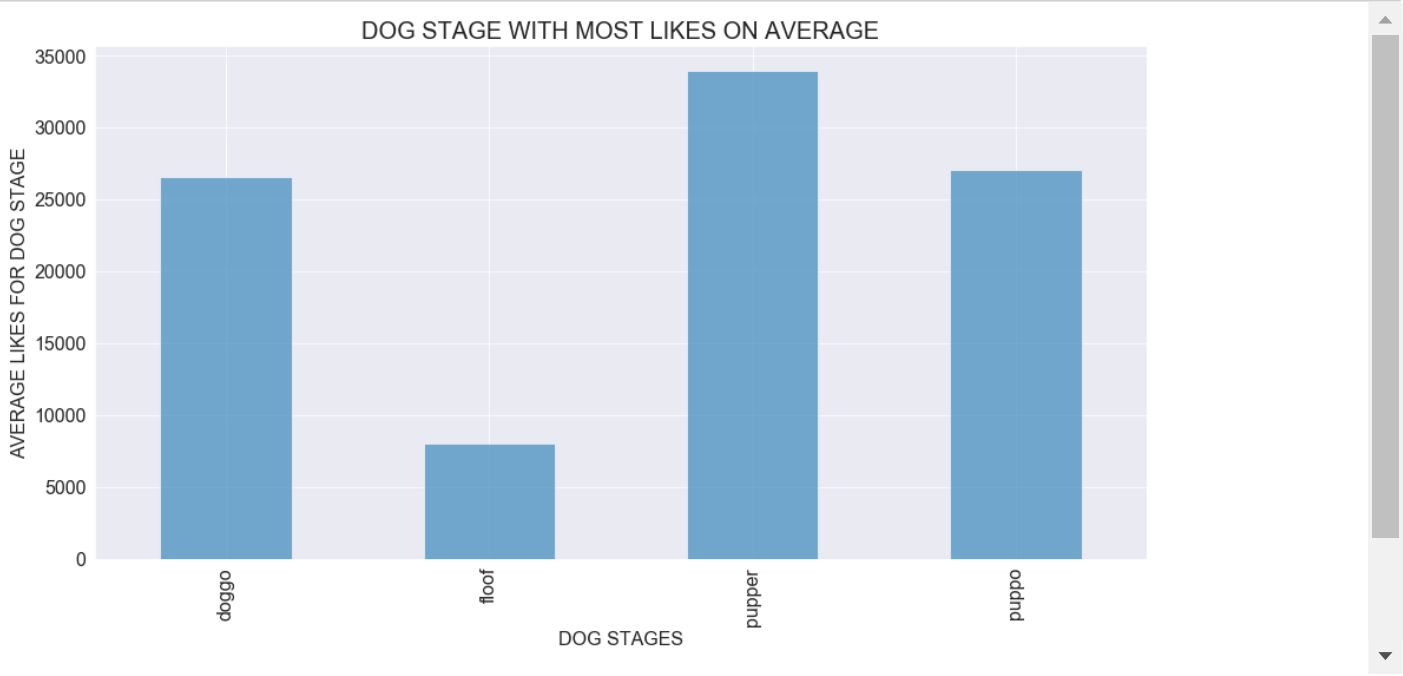
```
weRateDog_df.groupby('dog_stage')['favorite_count'].mean()
```

Out[53]:

```
dog_stage
doggo     26619.00000
floof      8015.86536
pupper    33983.00000
puppo     27088.50000
Name: favorite_count, dtype: float64
```

In [54]:

```
weRateDog_df.groupby('dog_stage')['favorite_count'].mean().plot(kind='bar', alpha=.6,figsiz
plt.title('DOG STAGE WITH MOST LIKES ON AVERAGE', fontsize=20)
plt.xlabel('DOG STAGES', fontsize=16)
plt.ylabel('AVERAGE LIKES FOR DOG STAGE', fontsize=16);
```




## WHAT TYPEP OF DOG STAGE HAS THE HEIGHEST NUMERATORE RATTINGS ON AVERAGE

In [55]:

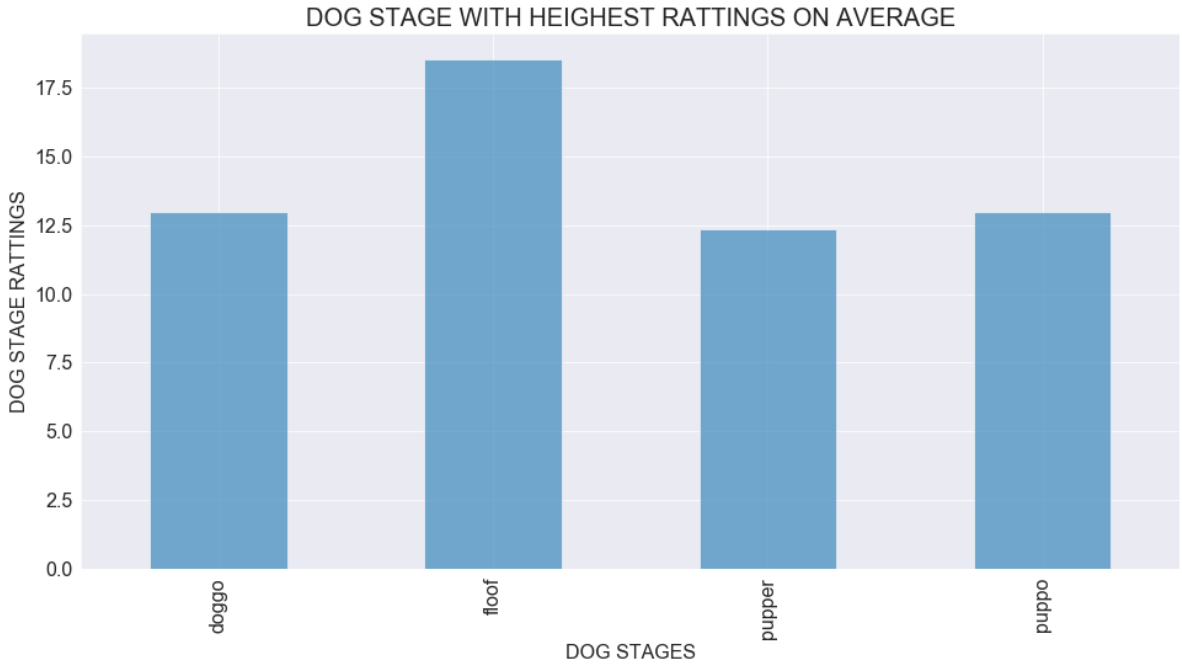
```
weRateDog_df.groupby('dog_stage')['dog_ratings_numeratore'].mean()
```

Out[55]:

```
dog_stage
doggo     13.000000
floof     18.531743
pupper    12.333333
puppo     13.000000
Name: dog_ratings_numeratore, dtype: float64
```

In [56]:

```
weRateDog_df.groupby('dog_stage')['dog_ratings_numeratore'].mean().plot(kind='bar',alpha=.6
plt.title('DOG STAGE WITH HEIGHEST RATTINGS ON AVERAGE', fontsize=20)
plt.xlabel('DOG STAGES', fontsize=16)
plt.ylabel('DOG STAGE RATTINGS', fontsize=16);
```




## WHAT DAY OF THE WEEK HAS MORE TWEETS OVER THE YEARS

In [57]:

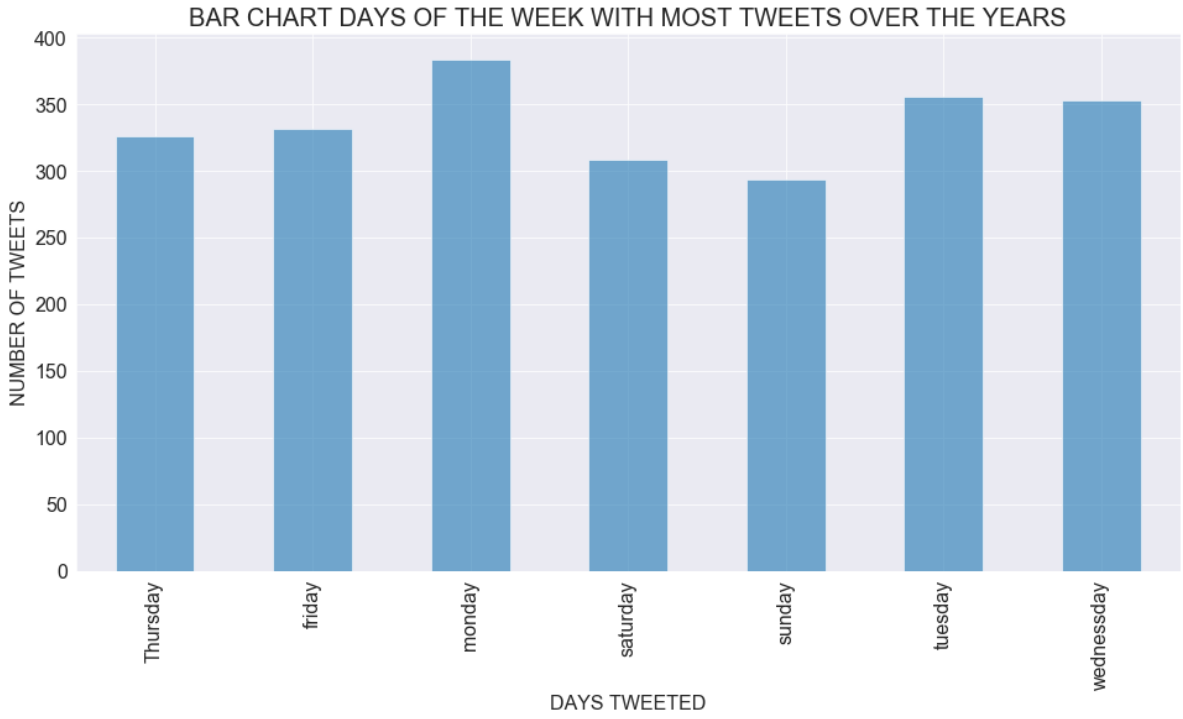
```
weRateDog_df.groupby('day_tweeted').count()['tweet_type']
```

Out[57]:

```
day_tweeted
Thursday      326
friday        332
monday        384
saturday      309
sunday        294
tuesday       356
wednessday    353
Name: tweet_type, dtype: int64
```

In [58]:

```
weRateDog_df.groupby('day_tweeted').count()['tweet_type'].plot(kind='bar', alpha=.6,figsize
plt.title('BAR CHART DAYS OF THE WEEK WITH MOST TWEETS OVER THE YEARS', fontsize=20)
plt.xlabel('DAYS TWEETED', fontsize=16)
plt.ylabel('NUMBER OF TWEETS', fontsize=16);
```




## WHICH DOG STAGE ON AVERAGE HAS THE MOST LIKES IN EACH YEAR

In [59]:

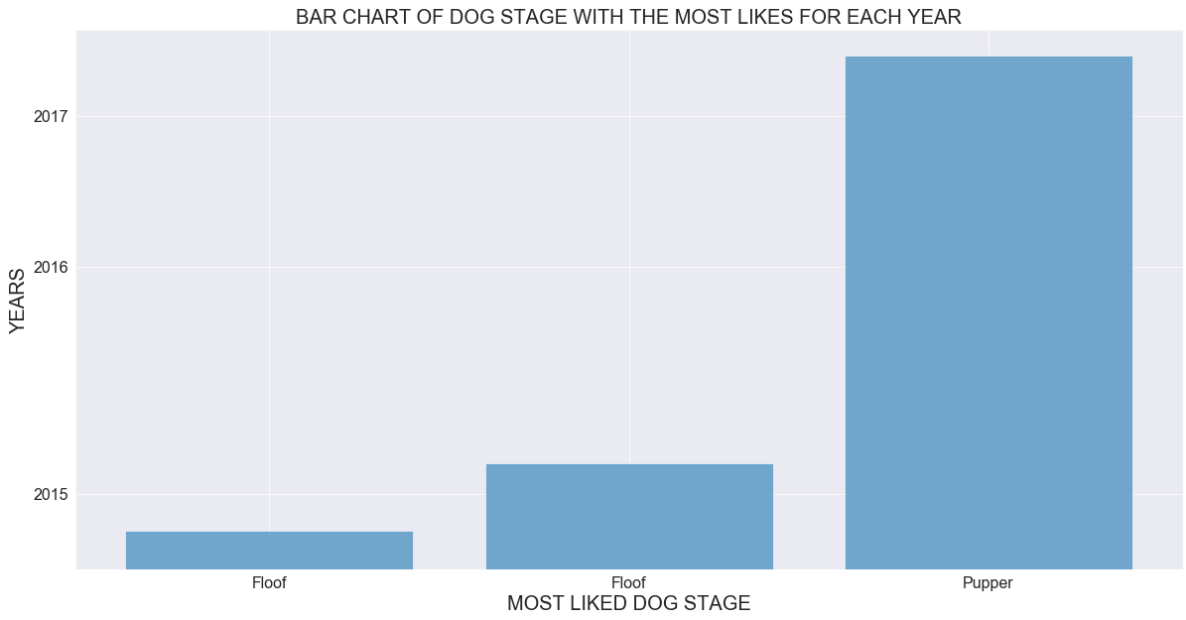
```
year_tweeted = weRateDog_df['date_tweeted'].apply(lambda x: str(x).split('-')[0])

weRateDog_df['year_tweeted'] = year_tweeted.astype(int)

dogStageYear = weRateDog_df.groupby(['year_tweeted', 'dog_stage'], as_index=False).mean().i

dogStageYear
```

Out[59]:

| | year_tweeted | dog_stage | favorite_count |
|---|---|---|---|
| **0** | 2015 | floof | 2519.078261 |
| **1** | 2016 | floof | 6997.131134 |
| **4** | 2017 | pupper | 33983.000000 |

In [60]:

```
y = dogStageYear['favorite_count']
x = range(3)
plt.figure(figsize=(20,10))
plt.rc('xtick', labelsize=16)
plt.rc('ytick', labelsize=16)
plt.bar(x, y, alpha=.6)
plt.title('BAR CHART OF DOG STAGE WITH THE MOST LIKES FOR EACH YEAR', fontsize=20)
plt.xlabel('MOST LIKED DOG STAGE', fontsize=20)
plt.ylabel('YEARS', fontsize=20)
X_location = [0.0, 1.0, 2.0]
X_label = ['Floof', 'Floof', 'Pupper']
plt.xticks(X_location, X_label)
y_loc = [30000, 20000, 5000]
y_label = [2017, 2016, 2015]
plt.yticks(y_loc, y_label);
```




## NUMBER OF RETWEET VAS TWEET OVER THE YEARS

In [61]:

```
Retweet = weRateDog_df.groupby(['tweet_type', 'year_tweeted'], as_index=False).count().quer
Retweet
```

Out[61]:

| | tweet_type | year_tweeted | tweet_text |
|---|---|---|---|
| **0** | Retweet | 2015 | 5 |
| **1** | Retweet | 2016 | 101 |
| **2** | Retweet | 2017 | 84 |

In [62]:

```
Tweet = weRateDog_df.groupby(['tweet_type', 'year_tweeted'], as_index=False).count().query(
Tweet
```

Out[62]:

| | tweet_type | year_tweeted | tweet_text |
|---|---|---|---|
| **3** | Tweet | 2015 | 685 |
| **4** | Tweet | 2016 | 1081 |
| **5** | Tweet | 2017 | 398 |