# act_report

January 27, 2018

### 0.1 Storing, Analyzing, and Visualizing Data for this Project

Once we have cleaned the three data sets from WeRateDogs data, we stored the clean DataFrame in a CSV file with the main one named **twitter_archive_master.csv**.

Next we will analyze and visualize our wrangled data.

```
In [1]: #connect to the internet
        import requests

        #deal with data
        import numpy as np
        import pandas as pd

        #deal with datetime
        import datetime as dt
        import pytz

        #deal with visualization
        import seaborn as sns
        %matplotlib inline
        import matplotlib.pyplot as plt

        #use padasql for SQL-query on dataframe
        #http://blog.yhat.com/posts/pandasql-intro.html
        from pandasql import sqldf
```

```
In [2]: df_master = pd.read_csv('twitter_archive_master.csv')
```

Finally we endet up with **1664 clean data sets** in the pandas dataframe df_master

```
In [3]: df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1664 entries, 0 to 1663
Data columns (total 15 columns):
tweet_id              1664 non-null int64
timestamp             1664 non-null object
source                1664 non-null object
```

```
text                   1664 non-null object
expanded_urls          1664 non-null object
rating_numerator       1664 non-null int64
rating_denominator     1664 non-null int64
name                   1203 non-null object
dog_stage              249 non-null object
retweet_count          1664 non-null int64
favorite_count         1664 non-null int64
jpg_url                1664 non-null object
dog                    1664 non-null object
conf                   1664 non-null float64
create_HH24            1664 non-null int64
dtypes: float64(1), int64(6), object(8)
memory usage: 195.1+ KB
```

In order to perform statistical we define the correct data types on the existing columns

```
In [4]: df_master.source = df_master.source.astype('category')
        df_master.dog_stage = df_master.dog_stage.astype('category')
        df_master.dog = df_master.dog.astype('category')
        df_master.create_HH24 = df_master.create_HH24.astype('category')
        df_master.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1664 entries, 0 to 1663
Data columns (total 15 columns):
tweet_id               1664 non-null int64
timestamp              1664 non-null object
source                 1664 non-null category
text                   1664 non-null object
expanded_urls          1664 non-null object
rating_numerator       1664 non-null int64
rating_denominator     1664 non-null int64
name                   1203 non-null object
dog_stage              249 non-null category
retweet_count          1664 non-null int64
favorite_count         1664 non-null int64
jpg_url                1664 non-null object
dog                    1664 non-null category
conf                   1664 non-null float64
create_HH24            1664 non-null category
dtypes: category(4), float64(1), int64(5), object(5)
memory usage: 156.5+ KB
```

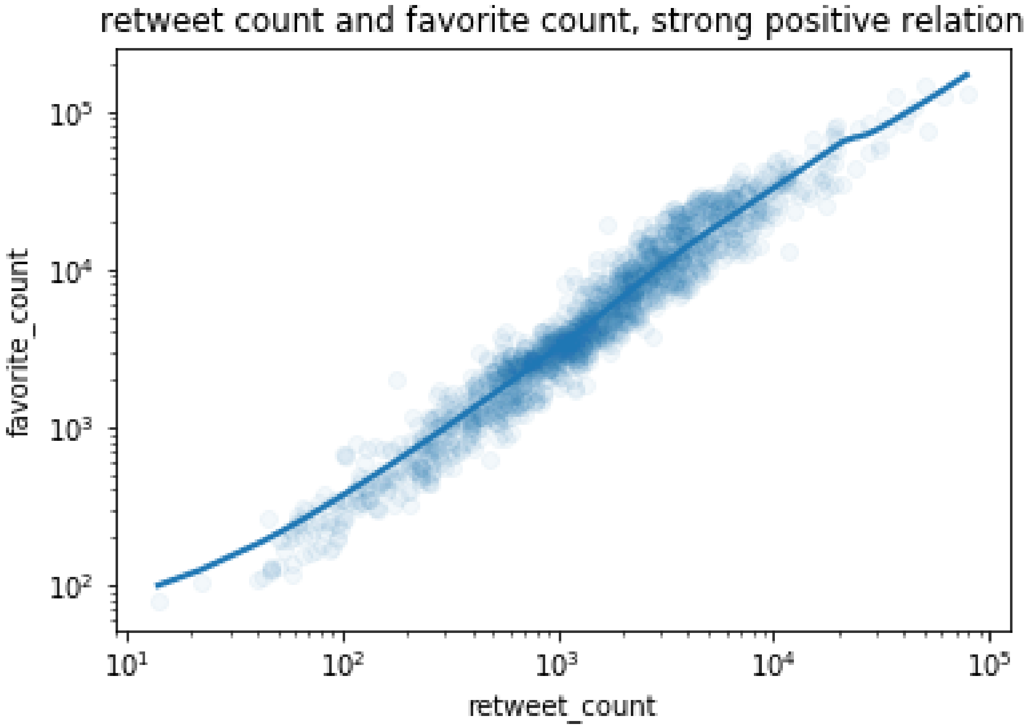
### 0.1.1 Correlation

Looking at persons correlation we ca just fin favorite_count and retweet_count significantly positive correlated. Interesting enough there is also a correlation between tweet_id and favorite_count. One possibe reson for this could be the increasing popularity of this site over time.

```
In [5]: df_master.corr(method='pearson')
```

```
Out[5]:                      tweet_id  rating_numerator  rating_denominator  \
        tweet_id             1.000000          0.550155                 NaN
        rating_numerator     0.550155          1.000000                 NaN
        rating_denominator        NaN               NaN                 NaN
        retweet_count        0.392921          0.317499                 NaN
        favorite_count       0.630534          0.420476                 NaN
        conf                 0.103490          0.142090                 NaN

                             retweet_count  favorite_count      conf
        tweet_id                  0.392921        0.630534  0.103490
        rating_numerator          0.317499        0.420476  0.142090
        rating_denominator             NaN             NaN       NaN
        retweet_count             1.000000        0.917411  0.027693
        favorite_count            0.917411        1.000000  0.059848
        conf                      0.027693        0.059848  1.000000
```

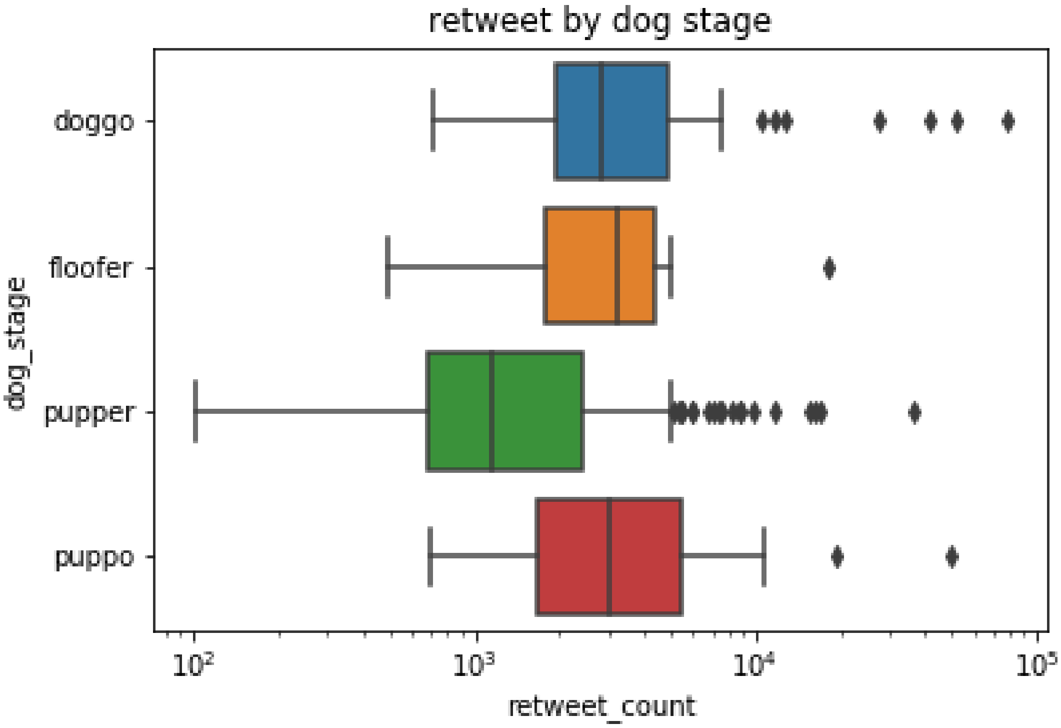
```
In [6]: # Initialize figure and ax
        fig, ax = plt.subplots()

        # Set the scale of the x-and y-axes
        ax.set(xscale="log", yscale="log")
        sns.regplot(x='retweet_count', y='favorite_count', data=df_master, ax=ax, scatter_kws={'
        plt.title('retweet count and favorite count, strong positive relation')
        plt.show()
```



There have been even books written about the differend dog stages.

According to WeRateDogs increased retweet count are possible for doggo, floofer and puppos while pupper have a clear disadvanted in retwwet counts

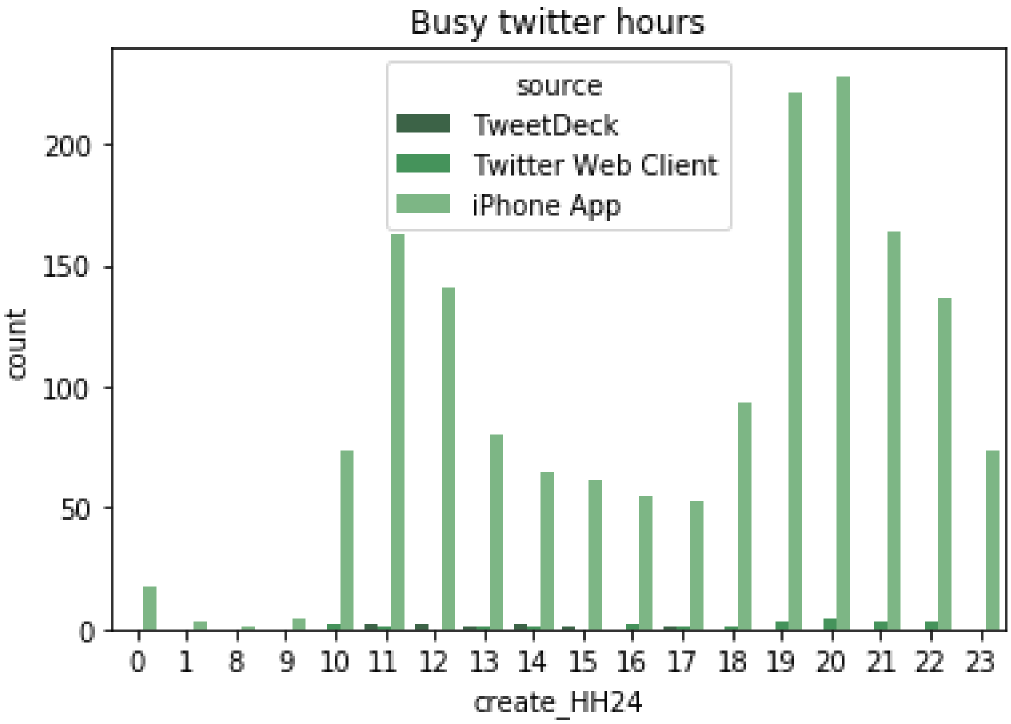
```
In [7]: # Create the boxplot
        ax = sns.boxplot(x="retweet_count", y="dog_stage", data=df_master)

        # Set the `xlim`
        ax.set(xscale="log")

        # Set title
        ax.set_title("retweet by dog stage")

        # Show the plot
        plt.show()
```



### 0.1.2 students work hard, . . . after 10 in the morning.

On the other hand we have to keep in mind that this site is maintained by a single student. Most of the time by using his iPhone! So let's proove this be looking at the creation time of the posts.

```
In [8]: ax = sns.countplot(x="create_HH24", hue="source", data=df_master, palette="Greens_d")
        ax.set_title('Busy twitter hours')
        plt.show()
```



Looking at the local posting hourse in EST we could confirm that all posts have been issued during daylinght time.

Our student like to get up not earlier thant 10 am in the morning. So that story could be confirmed by this simple countplot.

After all we have to commit that there are certain dog breeds which are simply cute.

If we just look at doog breeds which have been rated more than 10 times we have to admit . . .

```
In [9]: #pysqldf = lambda q: sqldf(q, globals())

        q = """
        SELECT
         dog
        ,count(*) as cnt
        ,avg(rating_numerator) as avg_rating_numerator
        ,avg(retweet_count) as avg_retweet_count
        ,avg(favorite_count) as avg_favorite_count
        FROM df_master
        GROUP BY 1
        having cnt>10
        ORDER BY avg_favorite_count desc
        ;
        """

In [10]: df_dog = pysqldf(q)
         df_dog.head(10)
```