# Project2: Wrangling and Analyzing Data

## Table of Contents


- Introduction
- Step 1: Gathering data
- Step 2: Assessing data
- Step 3: Cleaning data
- Step 4: Storing data
- Step 5: Analyzing and Visualizing data
- Step 6: Reporting
- Step 7: References


## Introduction

I am glad and privileged to be part of the second cohort of Udacity's Data Analyst Nanodegree program sponsored by ALX-T. This program prepares one for a career as a data analyst by helping to learn how to organize data, uncover patterns and insights, draw meaningful conclusions, and clearly communicate critical findings from various projects.

The second project is on WeRateDogs twitter account. WeRateDogs (dog_rates) is a Twitter account that rates people's dogs with a funny comment about the dog, our target is to wrangle its archive data from different sources to create a clean tidy dataframe, and then perform data analysis and visualization for the resulted dataframe.

This second project has to do with data wrangling. Wrangling Data includes; Data gathering, Assessing data and cleaning data. In this project the data were gathered using three methods; loading using pandas, using the request library and programmatically.

Data are assessed for quality and tidiness. Quality has to do with dirtiness of data while tidiness has to do with structure. Cleaning data involves, defining what to clean, coding and testing the code if it works the way the data analyst expected it to work.

Having laid the foundation, let's get ready to wrangle! 😊

### Step1: Gathering Data

The first thing is to import all the necessary python libraries to be used for the project. The code for importing relevant libraries is shown below. The libraries imported are pandas for dataframe, request to load our second dataset, tweepy and json to load the third dataset programmatically and others like seaborn, matplotlib for our data visualization.

```
# Import the libraries
import pandas as pd
import requests
import tweepy
from tweepy import OAuthHandler
import json
from timeit import default_timer as timer
import os
import numpy as np
import re
%matplotlib inline
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
```

Next is to load all the dataset that will be used from various sources.

1. Loading WeRateDogs Twitter archive data (twitter_archive_enhanced.csv).

WeRateDogs downloaded their Twitter archive and shared it exclusively for use in this project. This dataset contains 2356 rows and 16 columns.The data loading was achieved with just a line of code as shown below.

```
# Loading the first dataset directly
df_1= pd.read_csv('twitter-archive-enhanced.csv')
```

2. Using the Request library to download the tweet image prediction (image_predictions.tsv). It was hosted on Udacity's servers in tsv format and had to be downloaded using the requests library.

*Requests* is a versatile HTTP library in python with various applications. One of its applications is to download or open a file from web using the file URL.

The tweet image predictions file, i.e., what breed of dog is present in each tweet according to a neural network is stored in this file. The classification data has 3 breeds predictions p1, p2 and p3 with different probability. One good feature about this data, that these predictions are linked with tweet_id this will make joining these data with other tables easy. The images below showed how I used the requested library to download the file image-prediction.tsv

```
# Loading the second dataset using the request library
# df_2 for second DataFrame

df_2 = 'WeRateDogs_reviews'
if not os.path.exists(df_2):  # to check a folder exists or not
    os.makedirs(df_2)         # to create a directory/folder

url = 'https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv'

response = requests.get(url)
```

```
# Response [200] shows there is no error
response
```

```
<Response [200]>
```

```
with open(os.path.join(df_2, url.split('/')[-1]), mode = 'wb') as file:  #  extracting (image-predictions.tsv)
    file.write(response.content)
```

```
os.listdir(df_2)
```

```
['image-predictions.tsv']
```

```
# assigning df2 to the second dataset
df2 = pd.read_csv('image-predictions.tsv', sep='\t')
```

3. Loading the json file (tweet-json.txt) into a DataFrame.

The twitter API was not readily available during this project. Two files “twitter-api.py” and “tweet_json.txt” was provided by Udacity. Tweet data is stored in JSON format by Twitter.

I queried the Twitter API for each tweet’s JSON data using Python’s tweepy library. Further, each tweet’s entire set of JSON data was stored in a file called tweet_json.txt. Each tweet’s JSON data is written to its own line and then the .txt file is read line by line into a pandas DataFrame.

```
# converting the text file to a data list and DataFrame
df3_list = []  # creat an empty list for the dataframe

with open ('tweet-json.txt', encoding = 'utf-8') as file:
    for line in file:
        parse_json_to_dic = json.loads(line)  # parse jason string to python dictionary
        df3_list.append(parse_json_to_dic)
```

```
# passing the tweet_id,favorites_count and retweet_count into Dataframe

# creating the three list that will serve as column header
idlist = []
retweetlist = []
favoritelist = []

for info in df3_list:
    i_d = info['id']
    retweets = info['retweet_count']
    favorites = info ['favorite_count']


    idlist.append(i_d)
    retweetlist.append(retweets)
    favoritelist.append(favorites)
# create twwet_df dataframe
    df_3 = pd.DataFrame({'tweet_id': idlist, 'retweet_count': retweetlist, 'favorite_count':favoritelist})
```

## Step 2: Assessing Data

The requirements of this project are only to assess and clean at least 8 quality issues and at least 2 tidiness issues in this dataset. ***For this project, it is enough to find 8 quality issues, and 2 tidy***

*issue. And as the data might contain more than this, I am going to the follow the project requirements. Three dataframe were created , df_1, df_2 and df_3.*

Assessment Summary

Quality

(First DataFrame = df_1)

(1) Validity:The datatype in the timestamp column should be Datetime.

(2) Accuracy:Text cloumn contains tiny url and urls.

(3) Accuracy: Names like 'None', 'a' and 'an' are not real names.

(4) Accuracy: The statistics shows that rating denominator has low value(s) equal to 0 and as high as 170.

(5) Consistency: Retweets (reply to tweets-RT) are not needed.

-In the project's requirements, only original ratings that have images are required, not retweets nor replies.

(6) Consistency:The columns 'in_reply_to_status_id', 'in_reply_to_user_id', 'retweeted_status_id', 'retweeted_status_user_id', and 'retweeted_status_timestamp' are irrelevant and somehow repititive, and should be removed.

(7) Accuracy: The numerator has values as low as 0 and as high as 1776.

(Third DataFrame = df_3)

(8) Validity: tweet_id is integer instead of string datatype

Tidiness

Variable should form a column

(1) The “doggo”, “floorfer”, “pupper”, and “puppo” columns are phases in a dog's development, they can combine into one column (dog_phase).

An observational unit should form a table

(2) All the three tables/dataframe should be combined not a dataframe since they represent the same observation.

## Storing Data

The master dataframe (the cleaned data from the three dataset) was stored as twitter_archived_csv

## Analyzing and Visualizing Data

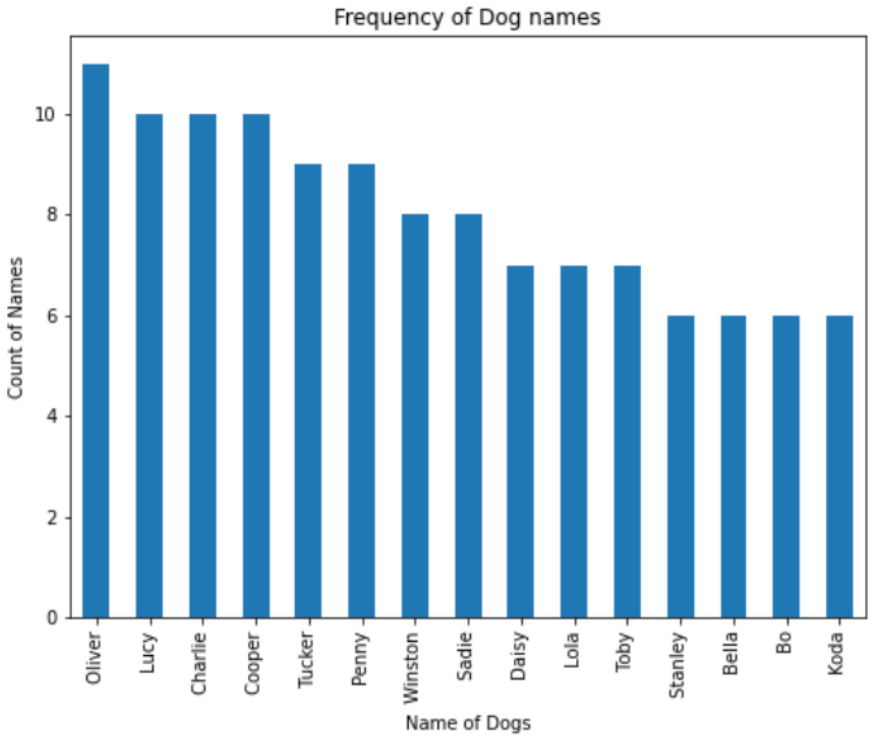
In analyzing and visualizing data a minimum of 3 insights and 1 visualization is required for this project. I tried  to answer these questions;

1 what us the most common dog name?

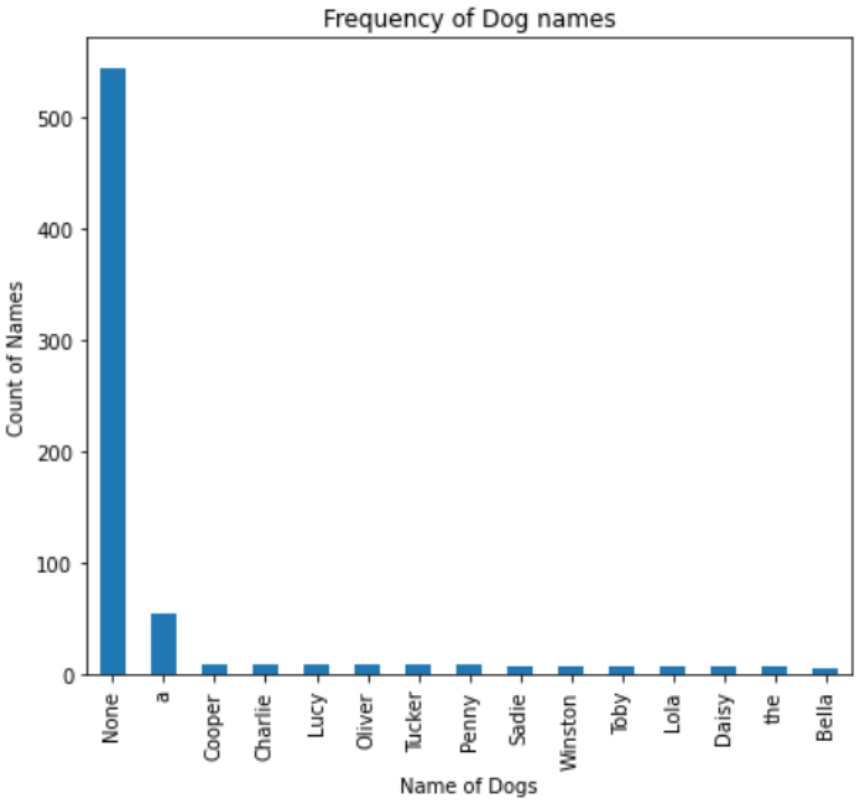
2 what would have happened if the data were not cleaned?

3 Which is the least favorite dog stage?

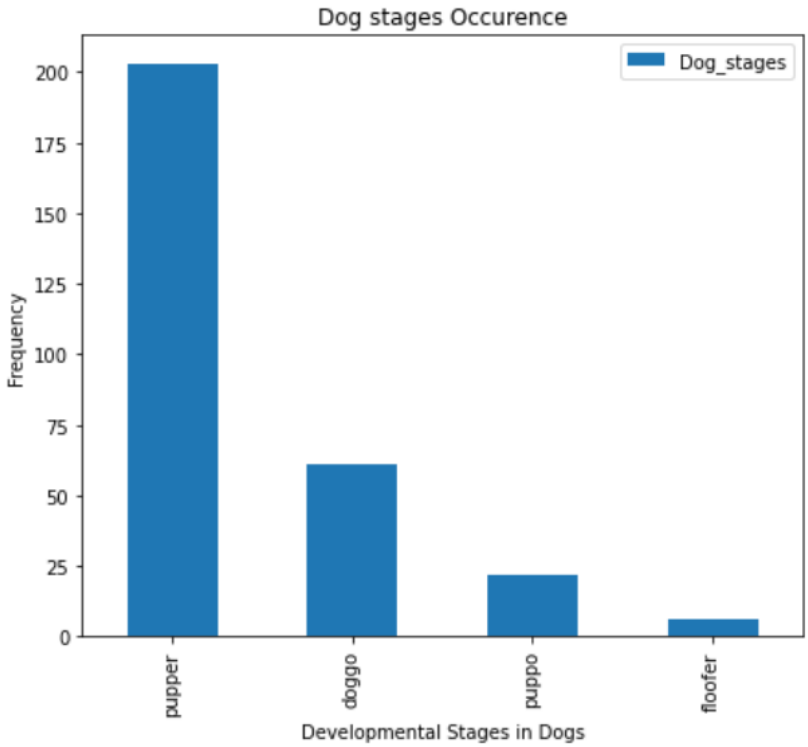
What is the most common dog name? The bar chart below clearly answers this. Oliver is the most popular dog name.



The result would have been very different if the data were not cleaned. Names like “None”, ”a” etc would have appeared on the plot, see below.




From the dataset there should be four dog developmental stages, the plot below shows the least preferred.




## Conclusion

Gathering, accessing and cleaning data could be very interesting when one knows the right tool to use. All these processes constitute the data wrangling aspect of data analysis. It is important to state here that data cleaning is a very important to data modelling.

Many other parameters in this dataset can be analyzed for different insights and visualizations, but this project requires minimum of (3) insight and (1) visualization.

What a laundry man does to dirty clothes is what a data wrangler does to data. In the introduction I said let's get ready to wrangle, I hope you have been entertained with the data wragling.