

## **04 Wrangle And Analyze Data Part 1**

DAVID LASSIG

2019-02-23

## Contents

# 1 Project 4: Wrangle and Analyze Data

## 1.1 Wrangle Act Part 1: Gathering, Assessing and Cleaning Data

### 1.1.1 Introduction

**Your tasks in this project are as follows:**

Data wrangling, which consists of: \* Gathering data (downloadable file in the Resources tab in the left most panel of your classroom and linked in step 1 below). \* Assessing data \* Cleaning data \* Storing, analyzing, and visualizing your wrangled data \* Reporting on 1) your data wrangling efforts and 2) your data analyses and visualizations

**Key points to keep in mind when data wrangling for this project:**

- You only want original ratings (no retweets) that have images. Though there are 5000+ tweets in the dataset, not all are dog ratings and some are retweets.
- Assessing and cleaning the entire dataset completely would require a lot of time, and is not necessary to practice and demonstrate your skills in data wrangling. Therefore, 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.
- Cleaning includes merging individual pieces of data according to the rules of tidy data.
- The fact that the rating numerators are greater than the denominators does not need to be cleaned. This unique rating system is a big part of the popularity of WeRateDogs.
- You do not need to gather the tweets beyond August 1st, 2017. You can, but note that you won't be able to gather the image predictions for these tweets since you don't have access to the algorithm used.

## 1.2 Environment Preparation

```
import pandas as pd
import numpy as np
import getapi
import json
import tweepy
import requests
import sys

if sys.version_info[0] < 3:
    from StringIO import StringIO
else:
    from io import StringIO
```

## 1.3 Data Gathering

### 1.3.1 Open We Rate Dogs Archive

```
wrd_df = pd.read_csv("twitter-archive-enhanced.csv")
```

### 1.3.2 Download and process associated twitter stats

For having the **Retweet Counts** and the **Favourite Counts** for each entry in the **twitter-archive-enhanced.csv**. I will download the whole Twitter API stats by using the Tweet ID. As the Twitter API allows only a certain amount of requests per time it will take a while. Moreover I will collect in the variable **missing** the Tweet ID's for which it wasn't possible to retrieve any additional information.

```
file_name="tweet_json.txt"
missing = []

api = getapi.get_twitter_api()

with open(file_name,mode="w") as file:
    for tid in wrd_df['tweet_id']:
        try:
            output = api.get_status(tid)
        except tweepy.TweepError as e:
            print(str(tid)+" "+str(e))
            missing.append(tid)
        file.write(json.dumps(output._json)+"\n")
```

Output:

```
888202515573088257:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```
Rate limit reached. Sleeping for: 34
```

```
873697596434513921:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```
872668790621863937:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```
869988702071779329:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```
866816280283807744:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```
861769973181624320:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```
845459076796616705:[{'code': 144, 'message': 'No status found with that
↪ ID.'}]
```

```

842892208864923648:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
837012587749474308:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
827228250799742977:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
812747805718642688:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
802247111496568832:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
775096608509886464:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
770743923962707968:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
754011816964026368:[{'code': 144, 'message': 'No status found with that
↳ ID.'}]
Rate limit reached. Sleeping for: 679

```

```

tweet_df = pd.DataFrame()

with open("tweet_json.txt", "r") as file:
    for index, line in enumerate(file):
        output = json.loads(line)
        tweet_df = tweet_df.append(pd.DataFrame.from_dict(output).head(1))
file.close()

```

```

tweet_df = tweet_df.reset_index(drop=True)
tweet_df = tweet_df[['id', 'retweet_count', 'favorite_count']]

```

### 1.3.3 Download and process image predictions

```

pred_url = "https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv"
response = requests.get(pred_url)

image_df = pd.read_csv(StringIO(response.text), sep="\t")
image_df.to_csv('image_predictions.tsv', sep='\t')

```

## 1.4 Data Assessing

### 1.4.1 First View

**wrd\_df**

```
wrd_df.head(1)
```

	tweet_id	reply_to_status_id	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
0	8924206435533015	NaN	NaN	2017-08-01T16:23:56+0000	ca href="http://twitter.com/download/iphone" data-kind="parent" data-rs="2">This is Phineas. He's a mystical pet. Only see...	NaN	NaN	NaN	NaN	https://twitter.com/dog_rates/status/892420643...13	13	10	Phineas	None	None	None	None

```
wrd_df.info()
```

Output:

```
<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
```

```
wrd_df[wrd_df.doggo != 'None'].iloc[:5]
```

	tweet_id	reply_to_status_id	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
9	890240255349188049	NaN	NaN	2017-07-26T15:59:31+0000	ca href="http://twitter.com/download/iphone" data-kind="parent" data-rs="2">This is Castle. She is a college prep. Studious...	NaN	NaN	NaN	NaN	https://twitter.com/dog_rates/status/890240255...14	14	10	Castle	doggo	None	None	None
43	884162670584377343	NaN	NaN	2017-07-09T21:20:42+0000	ca href="http://twitter.com/download/iphone" data-kind="ghost">	Meet Yogi. He doesn't have any important dog...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/884162670...12	12	10	Yogi	doggo	None	None	None
99	872967104147763200	NaN	NaN	2017-06-09T00:02:31+0000	ca href="http://twitter.com/download/iphone" data-kind="parent" data-rs="2">Here's a very large dog. He has a date later...	NaN	NaN	NaN	NaN	https://twitter.com/dog_rates/status/872967104...12	12	10	None	doggo	None	None	None
108	871515927908634025	NaN	NaN	2017-06-03T23:56:03+0000	ca href="http://twitter.com/download/iphone" data-kind="ghost">	This is Napoleon. He's a Raggedy East Neorg...	NaN	NaN	NaN	https://twitter.com/dog_rates/status/871515927...12	12	10	Napoleon	doggo	None	None	None
110	871102520638267392	NaN	NaN	2017-06-03T03:30:19+0000	ca href="http://twitter.com/download/iphone" data-kind="parent" data-rs="2">Never doubt a doggo's intelligence...	NaN	NaN	NaN	NaN	https://twitter.com/animalcog/status/871102520...1410	1410	10	None	doggo	None	None	None

```
wrd_df['doggo'].value_counts()
```

Output:

```
None      2259
doggo      97
Name: doggo, dtype: int64
```

```
wrd_df['pupper'].value_counts()
```

Output:

```
None      2099
pupper     257
Name: pupper, dtype: int64
```

```
wrd_df['floofer'].value_counts()
```

Output:

```
None      2346
floofer     10
Name: floofer, dtype: int64
```

```
wrd_df['puppo'].value_counts()
```

Output:

```
None      2326
puppo      30
Name: puppo, dtype: int64
```

```
wrd_df.groupby(['pupper', 'floofer', 'puppo']).doggo.value_counts()
```

Output:

pupper	floofer	puppo	doggo	
None	None	None	None	1976
			doggo	83
		puppo	None	29
			doggo	1
	floofer	None	None	9
			doggo	1
pupper	None	None	None	245
			doggo	12

Name: doggo, dtype: int64

```
type(wrd_df['timestamp'][0])
```

Output:

str

```
wrd_df['tweet_id'].nunique()
```

Output:

2356

```
wrd_df['name'].value_counts()[:5]
```

Output:

None	745
a	55

```

Charlie    12
Oliver     11
Lucy       11
Name: name, dtype: int64

```

## tweet\_df

```
tweet_df.head(1)
```

	id	retweet_count	favorite_count
0	892420643555336193	8281	37925

```
tweet_df.info()
```

### Output:

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 3 columns):
id                2356 non-null int64
retweet_count     2356 non-null int64
favorite_count    2356 non-null int64
dtypes: int64(3)
memory usage: 55.3 KB

```

## image\_df

```
image_df.head(1)
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh springer spaniel	0.465074	True	collie	0.156665	True	Shetland sheepdog	0.061428	True

```
image_df.info()
```

### Output:

```
<class 'pandas.core.frame.DataFrame'>
```



```
RangeIndex: 2075 entries, 0 to 2074
Data columns (total 12 columns):
tweet_id    2075 non-null int64
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(2), object(4)
memory usage: 152.1+ KB
```

```
type(image_df.p1_conf[0])
```

Output:

```
numpy.float64
```

```
wrd_df_clean = wrd_df.copy()
tweet_df_clean = tweet_df.copy()
image_df_clean = image_df.copy()
```

### 1.4.2 Quality Issues

With the overall impression of the assessed data I can identify several quality issues I need to clean for drawing any further conclusions.

#### WeRateDogs\_df (wrd\_df)

1. Remove columns that are unnecessary for further analysis from **wrd\_df**.
2. Remove columns that have almost only null values from **wrd\_df**.
3. Remove rows for which we didn't obtain a twitter status.
4. Replace four dog type columns into one categorical column in **wrd\_df**.

5. Convert timestamp in **wrd\_df** from string to datetime.
6. Remove names from **name** in **wrd\_df** that seems to be unvalid.

## image\_df

7. Remove columns that are unnecessary for further analysis from **image\_df**.
8. Remove second **p2** and third **p3** estimation from dataframe.

## tweet\_df

8. Rename Column **id** to **tweet\_id** for more easier merging.

## 1 Define

Remove **source** and **expanded\_urls** from **wrd\_df**

### 1 Code

```
wrd_df_clean.drop(columns=['source', 'expanded_urls'], inplace=True)
```

### 1 Test

```
wrd_df_clean.head(1)
```

	tweet_id	in_reply_to_status_id	in_reply_to_user_id	timestamp	text	retweeted_status_id	retweeted_status_user_id	retweeted_status_timestamp	rating_numerator	rating_denominator	name	doggo	floofer	pupper	puppo
0	892420643555336193	NaN	NaN	2017-08-01 16:23:56 +0000	This is Phineas. He's a mystical boy. Only eve...	NaN	NaN	NaN	13	10	Phineas	None	None	None	None

```
image_df_clean.head(1)
```

	tweet_id	jpg_url	img_num	p1	p1_conf	p1_dog	p2	p2_conf	p2_dog	p3	p3_conf	p3_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	1	Welsh springer spaniel	0.465074	True	collie	0.156665	True	Shetland sheepdog	0.061428	True

## 2 Define

Remove **\*in\_reply\_to\_status\_id \* in\_reply\_to\_user\_id \* retweeted\_status\_id \* retweeted\_status\_user\_id \* retweeted\_status\_time\_stamp**

from **wrd\_df** as it has almost only null values.

## 2 Code

```
wrd_df_clean.drop(columns=['in_reply_to_status_id',
                          'in_reply_to_user_id',
                          'retweeted_status_id',
                          'retweeted_status_user_id',
                          'retweeted_status_timestamp'], inplace=True)
```

## 2 Test

```
wrd_df_clean.info()
```

Output:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2356 entries, 0 to 2355
Data columns (total 10 columns):
tweet_id          2356 non-null int64
timestamp         2356 non-null object
text              2356 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: int64(3), object(7)
memory usage: 184.1+ KB
```

## 3 Define

Remove the rows for the **tweet\_id** we collected in the list **missing**.

## 3 Code

```
for tweet_id in missing:
    wrd_df_clean.drop(wrd_df_clean[wrd_df_clean['tweet_id'] == tweet_id].index[0], inplace=True)
```

### 3 Test

```
# if there is removed the right amount of rows, the calculation should result in zero
wrd_df.shape[0] - wrd_df_clean.shape[0] - len(missing)
```

Output:

0

### 4 Define

Take string values from **doggo**, **floofer**, **pupper** and **puppo** and put the not **None** values into one primary categorical column **dogtype** and taking the risk to remove a secondary label.

### 4 Code

```
categories = wrd_df_clean.keys()[-4:].tolist()
categories.append("none")
categories
```

Output:

['doggo', 'floofer', 'pupper', 'puppo', 'none']

```
wrd_df_clean['dogtype'] = pd.Series(pd.Categorical(values=["none"]*len(wrd_df_clean),categories=categories))
```

```
def check_dogtype(df, dogtype, dogtype_string):
```

```
    mask = dogtype != "None"
```

```
    for index,entry in df[mask].iterrows():
```

```
        df.loc[index,'dogtype'] = dogtype_string
```

```
check_dogtype(wrd_df_clean,wrd_df_clean.doggo,'doggo')
```

```
check_dogtype(wrd_df_clean,wrd_df_clean.pupper,'pupper')
```

```
check_dogtype(wrd_df_clean,wrd_df_clean.floofer,'floofer')
```

```
check_dogtype(wrd_df_clean,wrd_df_clean.puppo,'puppo')
```

```
wrd_df_clean.drop(columns=['doggo','floofer','pupper','puppo'],inplace=True)
```

## 4 Test

```
wrd_df_clean['dogtype'].value_counts()
```

Output:

```
none      1946
pupper    256
doggo      82
puppo      30
floofer    10
Name: dogtype, dtype: int64
```

## 5 Define

Convert the **timestamp** column from **wrd\_df** to datetime.

## 5 Code

```
wrd_df_clean['timestamp'] = pd.to_datetime(wrd_df_clean['timestamp'])
```

## 5 Test

```
wrd_df_clean['timestamp'][1] - wrd_df_clean['timestamp'][0]
```

Output:

```
Timedelta('-1 days +07:53:31')
```

## 6 Define

Remove names from **name** in **wrd\_df** that seems to be invalid like “a”.

## 6 Code

```
wrd_df_clean.query('name == "a"').apply(lambda x: "None" if x.name == "a" else False)
```

Output:

```
tweet_id      False
timestamp      False
text           False
rating_numerator  False
rating_denominator False
name           False
dogtype        False
dtype: bool
```

## 6 Test

```
wrd_df_clean.head(1)
```

	tweet_id	timestamp	text	rating_numerator	rating_denominator	name	dogtype
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	13	10	Phineas	none

## 7 Define

Remove **img\_num** from **image\_df**.

## 7 Code

```
image_df_clean.drop(columns=['img_num'], inplace=True)
```

## 7 Test

```
image_df_clean.head(5)
```

	tweet_id	jpg_url	p1	p1_conf	p1_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh_springer_spaniel	0.465074	True
1	666029285002620928	https://pbs.twimg.com/media/CT42GRgUYAA5iDo.jpg	redbone	0.506826	True
2	666033412701032449	https://pbs.twimg.com/media/CT4521TWwAEvMyu.jpg	German_shepherd	0.596461	True
3	666044226329800704	https://pbs.twimg.com/media/CT5Dr8HUEAA-lEu.jpg	Rhodesian_ridgeback	0.408143	True
4	666049248165822465	https://pbs.twimg.com/media/CT5IQmsXIAAKY4A.jpg	miniature_pinscher	0.560311	True

## 8 Define

Remove **p2**, **p2\_dog**, **p2\_conf**, **p3**, **p3\_dog** and **p3\_conf** from **image\_df** as it is enough for our purpose to remain the estimation with the highest confidence.

## 8 Code

```
image_df_clean.drop(columns=['p2', 'p2_dog', 'p2_conf', 'p3', 'p3_dog', 'p3_conf'], inplace=True)
```

## 8 Test

```
image_df_clean.head(1)
```

	tweet_id	jpg_url	p1	p1_conf	p1_dog
0	666020888022790149	https://pbs.twimg.com/media/CT4udn0WwAA0aMy.jpg	Welsh_springer_spaniel	0.465074	True

## 9 Define

Rename Column in **tweet\_df** from **id** to **tweet\_id**.

## 9 Code

```
tweet_df_clean.rename(columns={'id': 'tweet_id'}, inplace=True)
```

## 9 Test

```
tweet_df_clean.head(1)
```

	tweet_id	retweet_count	favorite_count
0	892420643555336193	8281	37925

### 1.4.3 Tidiness Issues

1. Convert **rating\_nominator** and **rating\_denominator** in **wrd\_df** to a single fraction.
2. **retweetCount** and **favouriteCount** should be merged by **tweet\_id** from **tweet\_df** to **wrd\_df**.

#### 1 Define

Convert **rating\_nominator** and **rating\_denominator** in **wrd\_df** to a single fraction **rating** and remove them.

#### 1 Code

```
wrd_df_clean['rating'] = wrd_df_clean['rating_nominator'] / wrd_df_clean['rating_denominator']
wrd_df_clean.drop(columns=['rating_nominator', 'rating_denominator'], inplace=True)
```

#### 1 Test

```
wrd_df_clean.head(1)
```

	tweet_id	timestamp	text	name	dogtype	rating
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	Phineas	none	1.3

#### 2 Define

Merge wrd\_df with tweet\_df by using the **tweet\_id** as the key.

#### 2 Code

```
twitter_archive_master = wrd_df_clean.merge(tweet_df_clean, how='outer', left_on='tweet_id', right_on='tweet_id')
```

#### 2 Test

```
twitter_archive_master.head(5)
```



	tweet_id	timestamp	text	name	dogtype	rating	retweet_count	favorite_count
0	892420643555336193	2017-08-01 16:23:56	This is Phineas. He's a mystical boy. Only eve...	Phineas	none	1.3	8281	37925
1	892177421306343426	2017-08-01 00:17:27	This is Tilly. She's just checking pup on you....	Tilly	none	1.3	6117	32566
2	891815181378084864	2017-07-31 00:18:03	This is Archie. He is a rare Norwegian Pouncin...	Archie	none	1.2	4051	24522
3	891689557279858688	2017-07-30 15:58:51	This is Darla. She commenced a snooze mid meal...	Darla	none	1.3	8422	41271
4	891327558926688256	2017-07-29 16:00:24	This is Franklin. He would like you to stop ca...	Franklin	none	1.2	9122	39452

#### 1.4.4 Write finished dataframes to csv

```
twitter_archive_master.to_csv('twitter_archive_master.csv', sep=',', index=False)
image_df_clean.to_csv('twitter_image_prediction.csv', sep=',', index=False)
```