**Key Points**

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](https://cran.r-project.org/web/packages/tidyr/vignettes/tidy-data.html).
* The fact that the rating numerators are greater than the denominators does not need to be cleaned. This [unique rating system](http://knowyourmeme.com/memes/theyre-good-dogs-brent) 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.

## Project Details

Your tasks in this project are as follows:

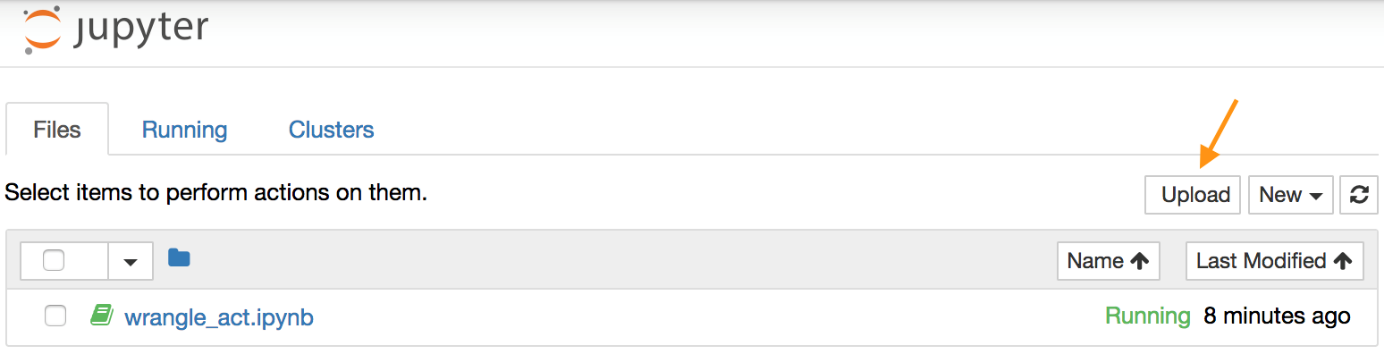
* Data wrangling, which consists of:
  + Gathering data (downloadable file in the Resources tab in the left most panel of your classroom and linked in step 1 below).
  + Assessing data
  + Cleaning data
* Storing, analyzing, and visualizing your wrangled data
* Reporting on 1) your data wrangling efforts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

### Gathering Data for this Project

Gather each of the three pieces of data as described below in a Jupyter Notebook titled wrangle\_act.ipynb:

1. The WeRateDogs Twitter archive. I am giving this file to you, so imagine it as a file on hand. Download this file manually by clicking the following link: [twitter\_archive\_enhanced.csv](https://d17h27t6h515a5.cloudfront.net/topher/2017/August/59a4e958_twitter-archive-enhanced/twitter-archive-enhanced.csv)
2. The tweet image predictions, i.e., what breed of dog (or other object, animal, etc.) is present in each tweet according to a neural network. This file (image\_predictions.tsv) is hosted on Udacity's servers and should be downloaded programmatically using the [Requests](http://docs.python-requests.org/en/master/) library and the following URL: <https://d17h27t6h515a5.cloudfront.net/topher/2017/August/599fd2ad_image-predictions/image-predictions.tsv>
3. Each tweet's retweet count and favorite ("like") count at minimum, and any additional data you find interesting. Using the tweet IDs in the WeRateDogs Twitter archive, query the Twitter API for each tweet's JSON data using Python's [Tweepy](http://www.tweepy.org/) library and store each tweet's entire set of JSON data in a file called tweet\_json.txt file. Each tweet's JSON data should be written to its own line. Then read this .txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count. Note: do not include your Twitter API keys, secrets, and tokens in your project submission.

If you decide to complete your project in the Project Workspace, note that you can upload files to the Jupyter Notebook Workspace by clicking the "Upload" button in the top righthand corner of the dashboard.

[[](https://classroom.udacity.com/nanodegrees/nd002/parts/af503f34-9646-4795-a916-190ebc82cb4a/modules/14d9f5f1-9e7b-4bfb-97f3-bcdbf4a3699c/lessons/a8085857-3e28-4fc7-aeb8-da64ccbc2e20/concepts/5919f3b1-899f-4295-80f1-17f091eb4df6)](https://classroom.udacity.com/nanodegrees/nd002/parts/af503f34-9646-4795-a916-190ebc82cb4a/modules/14d9f5f1-9e7b-4bfb-97f3-bcdbf4a3699c/lessons/a8085857-3e28-4fc7-aeb8-da64ccbc2e20/concepts/5919f3b1-899f-4295-80f1-17f091eb4df6)

[Jupyter Notebook dashboard with arrow pointing to the "Upload" button](https://classroom.udacity.com/nanodegrees/nd002/parts/af503f34-9646-4795-a916-190ebc82cb4a/modules/14d9f5f1-9e7b-4bfb-97f3-bcdbf4a3699c/lessons/a8085857-3e28-4fc7-aeb8-da64ccbc2e20/concepts/5919f3b1-899f-4295-80f1-17f091eb4df6)

### Assessing Data for this Project

After gathering each of the above pieces of data, assess them visually and programmatically for quality and tidiness issues. Detect and document at least **eight (8) quality issues** and **two (2) tidiness issues** in your wrangle\_act.ipynb Jupyter Notebook. To meet specifications, the issues that satisfy the Project Motivation (see the Key Points header on the previous page) must be assessed.

### Cleaning Data for this Project

Clean each of the issues you documented while assessing. Perform this cleaning in wrangle\_act.ipynb as well. The result should be a high quality and tidy master pandas DataFrame (or DataFrames, if appropriate). Again, the issues that satisfy the Project Motivation must be cleaned.

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

Store the clean DataFrame(s) in a CSV file with the main one named twitter\_archive\_master.csv. If additional files exist because multiple tables are required for tidiness, name these files appropriately. Additionally, you may store the cleaned data in a SQLite database (which is to be submitted as well if you do).

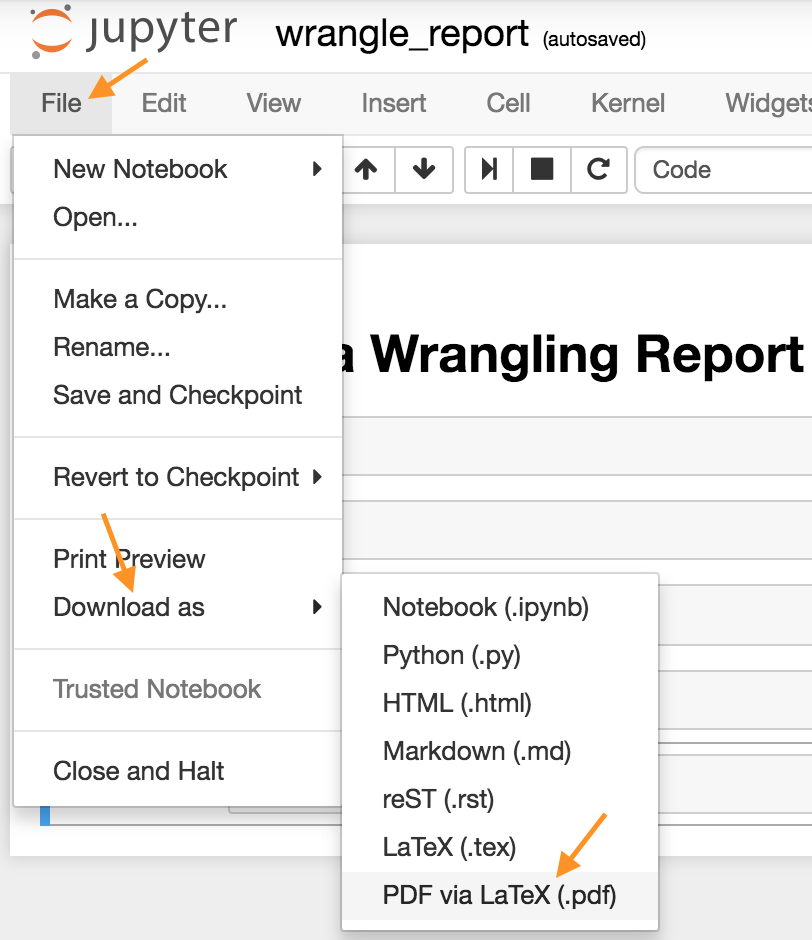
Analyze and visualize your wrangled data in your wrangle\_act.ipynb Jupyter Notebook. At least **three (3) insights and one (1) visualization** must be produced.

### Reporting for this Project

Create a **300-600 word written report** called wrangle\_report.pdf or wrangle\_report.html that briefly describes your wrangling efforts. This is to be framed as an internal document.

Create a **250-word-minimum written report** called act\_report.pdf or act\_report.html that communicates the insights and displays the visualization(s) produced from your wrangled data. This is to be framed as an external document, like a blog post or magazine article, for example.

Both of these documents can be created in separate Jupyter Notebooks using the [Markdown functionality](http://jupyter-notebook.readthedocs.io/en/stable/examples/Notebook/Working%20With%20Markdown%20Cells.html) of Jupyter Notebooks, then downloading those notebooks as PDF files or HTML files (see image below). You might prefer to use a word processor like Google Docs or Microsoft Word, however.

[[](https://classroom.udacity.com/nanodegrees/nd002/parts/af503f34-9646-4795-a916-190ebc82cb4a/modules/14d9f5f1-9e7b-4bfb-97f3-bcdbf4a3699c/lessons/a8085857-3e28-4fc7-aeb8-da64ccbc2e20/concepts/5919f3b1-899f-4295-80f1-17f091eb4df6)](https://classroom.udacity.com/nanodegrees/nd002/parts/af503f34-9646-4795-a916-190ebc82cb4a/modules/14d9f5f1-9e7b-4bfb-97f3-bcdbf4a3699c/lessons/a8085857-3e28-4fc7-aeb8-da64ccbc2e20/concepts/5919f3b1-899f-4295-80f1-17f091eb4df6)

[Download Jupyter Notebook in PDF form](https://classroom.udacity.com/nanodegrees/nd002/parts/af503f34-9646-4795-a916-190ebc82cb4a/modules/14d9f5f1-9e7b-4bfb-97f3-bcdbf4a3699c/lessons/a8085857-3e28-4fc7-aeb8-da64ccbc2e20/concepts/5919f3b1-899f-4295-80f1-17f091eb4df6)

## How to Query Twitter Data

In this project, you'll be using [Tweepy](http://www.tweepy.org/) to query Twitter's API for additional data beyond the data included in the WeRateDogs Twitter archive. This additional data will include retweet count and favorite count.

Some APIs are completely open, like MediaWiki (accessed via the [wptools](https://github.com/siznax/wptools/wiki) library) in Lesson 2. Others require authentication. The Twitter API is one that requires users to be authorized to use it. This means that before you can run your API querying code, you need to set up your own Twitter application. Here are the steps to do that on the Twitter site:

* First, if you do not already have one, you need to [sign up for a Twitter account](https://help.twitter.com/en/create-twitter-account).
* Next, to set up a developer account, follow the directions on [Twitter’s Developer Portal, in the “How to Apply” section](https://developer.twitter.com/en/docs/basics/developer-portal/overview).
* You will be guided through the steps, and asked to describe in your own words what you are building. Here is some suggested language you can use: "As a Udacity student, I need to access the Twitter API in order to complete a Data Wrangling student project. In this project, I'll be using Tweepy to query Twitter's API for data included in the WeRateDogs Twitter archive. This data will include retweet count and favorite count. Before I can run my API querying code, I need to set up my own Twitter application. Once I have this set up, I will develop some code to create an API object that I'll use to gather Twitter data. After querying each tweet ID, I will write its JSON data to a tweet\_json.txt file with each tweet's JSON data on its own line. I will then read this file, line by line, to create a pandas DataFrame that I will assess and clean. I may post this completed project on my GitHub account, where it will get viewers. Otherwise there will be no other readers or users of my Twitter data or project analysis beyond the Udacity instructors and reviewers."
* Once you submit your application, you should soon receive an email from Twitter letting you know they have approved your new Twitter developer account. Follow the link in the email from Twitter to a page of directions to get started creating your app.
* If you are asked for an app name, it can be anything appropriate, and if you’re asked for a Website URL, it can be anything in a standard URL format. You can do the same with other requested URLs, or perhaps leave them blank.
* If you’re asked to explain how your app will be used, you could say something like "I'm creating this for a student Data Wrangling project with Udacity, where we need to query and analyze Twitter data from WeRateDogs."
* You should then be given a Success message, and a new developer page displayed to you where you can manage your app.
* You can then go to the Keys and Tokens tab on this page to find or generate the Consumer API keys, and the Access Token and Access Token Secret that you will need.

**Note**: If you have any trouble creating this Twitter account or accessing the data, please see the section at the bottom of this page "Accessing Project Data Without a Twitter Account."

Once you have your Twitter account and Twitter app set up, the following code, which is provided in the [Getting started](https://media.readthedocs.org/pdf/tweepy/latest/tweepy.pdf) portion of the Tweepy documentation, will create an API object that you can use to gather Twitter data.

import tweepy

consumer\_key = 'YOUR CONSUMER KEY'

consumer\_secret = 'YOUR CONSUMER SECRET'

access\_token = 'YOUR ACCESS TOKEN'

access\_secret = 'YOUR ACCESS SECRET'

auth = tweepy.OAuthHandler(consumer\_key, consumer\_secret)

auth.set\_access\_token(access\_token, access\_secret)

api = tweepy.API(auth)

Tweet data is stored in JSON format by Twitter. Getting tweet JSON data via tweet ID using Tweepy is described well in this [StackOverflow answer](https://stackoverflow.com/questions/28384588/twitter-api-get-tweets-with-specific-id). Note that setting the tweet\_mode parameter to 'extended' in the get\_status call, i.e., api.get\_status(tweet\_id, tweet\_mode='extended'), can be useful.

Also, note that the tweets corresponding to a few tweet IDs in the archive may have been deleted. [Try-except blocks](https://wiki.python.org/moin/HandlingExceptions) may come in handy here.

## Do Not Include Your API Keys, Secrets, and Tokens in Your Submission

Do **not** include your API keys, secrets, and tokens in your project submission. This is standard practice for APIs and public code.

## Twitter's Rate Limit

Twitter's API has a rate limit. Rate limiting is used to control the rate of traffic sent or received by a server. As per [Twitter's rate limiting info page](https://developer.twitter.com/en/docs/basics/rate-limiting):

Rate limits are divided into 15 minute intervals

To query all of the tweet IDs in the WeRateDogs Twitter archive, 20-30 minutes of running time can be expected. Printing out each tweet ID after it was queried and [using a code timer](https://stackoverflow.com/questions/7370801/measure-time-elapsed-in-python) were both helpful for sanity reasons. Setting the wait\_on\_rate\_limit and wait\_on\_rate\_limit\_notify parameters to True in the [tweepy.api class](http://docs.tweepy.org/en/v3.2.0/api.html#API) is useful as well.

## Writing and Reading Twitter JSON

After querying each tweet ID, you will write its JSON data to the required tweet\_json.txt file with each tweet's JSON data on its own line. You will then read this file, line by line, to create a pandas DataFrame that you will soon assess and clean. This [Reading and Writing JSON to a File in Python](http://stackabuse.com/reading-and-writing-json-to-a-file-in-python/) article from Stack Abuse, will be useful.

## Accessing Project Data Without a Twitter Account

If you can't set up a Twitter developer account using the steps above, or you prefer not to create a Twitter account for some reason, you may instead follow the directions below to access the data necessary for the project. **Note**: We recommend that you follow the steps above to access the data using a Twitter developer account, because with the shortcut detailed below, you will miss practicing the valuable skill of gathering this data on your own. However, Twitter's updated process may not work for everyone, and we realize there are legitimate reasons that some students may prefer this approach, so we provide it to you here. **You choose the approach to access the data that works best for you. This shortcut approach will certainly work for you to pass the project equally well.**

#### Directions for accessing the Twitter data without actually creating a Twitter account:

At the bottom of this page you can find two files you can download:

* twitter\_api.py: This is the Twitter API code to gather some of the required data for the project. Read the code and comments, understand how the code works, then copy and paste it into your notebook.
* tweet\_json.txt: This is the resulting data from twitter\_api.py. You can proceed with the following part of "Gathering Data for this Project" on the Project Details page: "Then read this tweet\_json.txt file line by line into a pandas DataFrame with (at minimum) tweet ID, retweet count, and favorite count."