Project 9: Twitter

Clarifications/Corrections

- Apr 8: fixed wording on q18 and q19 to match test.py (replaced "ascending" with "descending")
- Apr 9: reworded directions to download test.py (test.py isn't in a zip, so you don't need to "extract" it)
- Apr 11: if `num_liked` is a string containing a number followed by a suffix (e.g., `"869M"` or `"915k"`), convert it to an integer, multiplying by 1000 or 1000000 as appropriate. If it contains some other kind of string (e.g., `"unkown"`), use 0 for `num_liked` (do not discard the tweet in this case).
- Apr 11: fixed test.py for q32 (please re-download)

Introduction

In this project, you'll be analyzing a collection of actual tweets.
This data is messy! You'll face the following challenges:

- data is spread across multiple files
- some files will be CSVs, others JSONs
- the files may be missing values or be too corrupt to parse
- some integer values may be represented as strings with a suffix of "M", "K", or similar

In stage 1, you'll write code to cleanup the data, representing everything as Tweet objects (you'll create a new type for these). In stage 2, you'll analyze your clean data.

Setup

Step 1: download `tweets.zip` and extract it to a directory on your computer (using Mac directions or Windows directions).

Step 2: download `test.py` to the directory from step 1 (`test.py` be next to the `sample_data` directory, for example)

Step 3: create a `main.ipynb` in the same location. Do all work for both stages there, and turn it in when complete.

Note: Make sure `full_data`, `sample_data`, `main.ipynb` and `test.py` are in same directory.

The Stages

- Stage 1: parse a mix of CSV and JSON files to get Tweet objects
- Stage 2: learn about the tweeters and recurse