

# How to Collect, Clean, and Use Twitter Data with Python

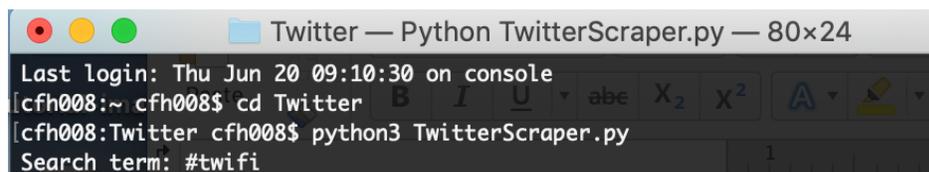
By Christian Howard-Sukhil  
June 2019

## Download the Python scripts

1. Go to the GitHub page for the Twitter project, “TwitLit” (<https://github.com/TwitLit/TwitLitSource>).
2. Download both the Python files (TwitterScrapper.py and count\_jsonl\_rows.py).
3. Put both of these files in the folder/directory in which you want to collect all of your Twitter data.

## Scrape Twitter using the Python Twitter Scraper

4. Open Terminal and navigate to the folder/directory that contains the python Twitter Scraper (TwitterScrapper.py).
  - a. To change directory in Terminal, enter the following into the command line: `cd [name of file]`
5. In this directory, run the python Twitter Scraping script by entering the following into the command line: `python3 TwitterScrapper.py`
6. This will then prompt you for the search term
  - a. The script can search for # specific terms; simply enter the hashtag followed by the term when prompted (e.g., #twitfiction)
  - b. To search for multiple terms simultaneously, put OR between terms



```
Twitter — Python TwitterScrapper.py — 80x24
Last login: Thu Jun 20 09:10:30 on console
[cfh008:~ cfh008$ cd Twitter
[cfh008:Twitter cfh008$ python3 TwitterScrapper.py
Search term: #twifi
```

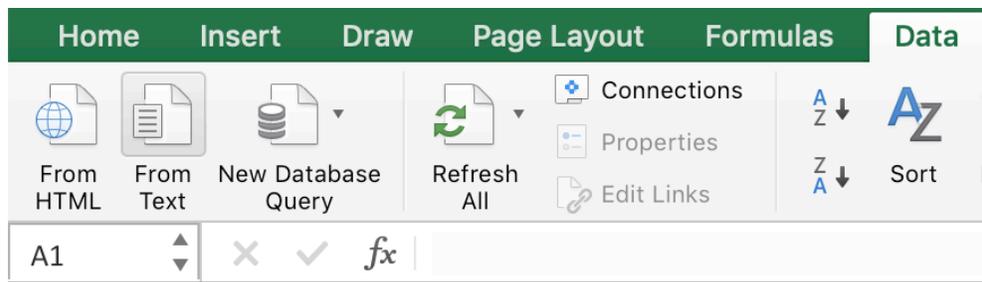
7. Once you enter the search term, you will be prompted to enter a date range. You'll be asked for the start date (YYYY-MM-DD); after inputting this and pressing enter, you'll be asked for the end date (YYYY-MM-DD).

```
Twitter — Python TwitterScrapper.py — 80x24
Last login: Thu Jun 20 09:10:30 on console
[cfh008:~ cfh008$ cd Twitter
[cfh008:Twitter cfh008$ python3 TwitterScrapper.py
Search term: #twifi
Find tweets since [yyyy-mm-dd]: 2019-01-01
Find tweets until [yyyy-mm-dd]: 2019-02-01
```

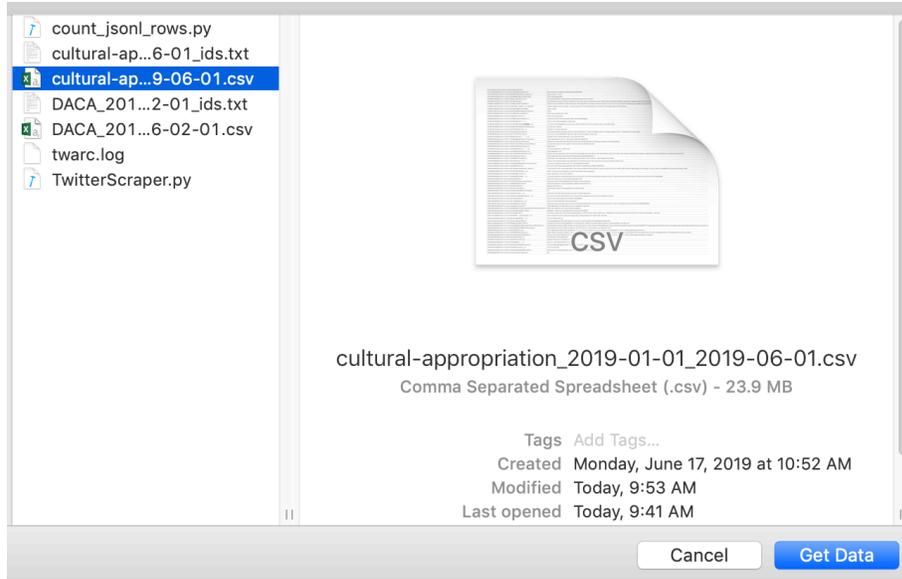
8. After you input the date range, the script will run. This could take a few seconds or several hours, depending on how many search results there are.
9. Once the script finishes running, you'll see both a txt file and a csv file in the folder containing your python Twitter Scraper. These will automatically be named in the following format: search-term\_start-date\_end-date. (NB: If you used a hashtag in your search term, the hashtag will not appear in the document name since doing so would cause problems with the file naming system.)
  - a. The txt file will include a full list of Tweet IDs (unhydrated).
  - b. The csv file will contain the Tweet IDs plus selected information about each tweet, including the full text of the tweet, the timestamp, and the user screen name.

## Cleaning up your csv files

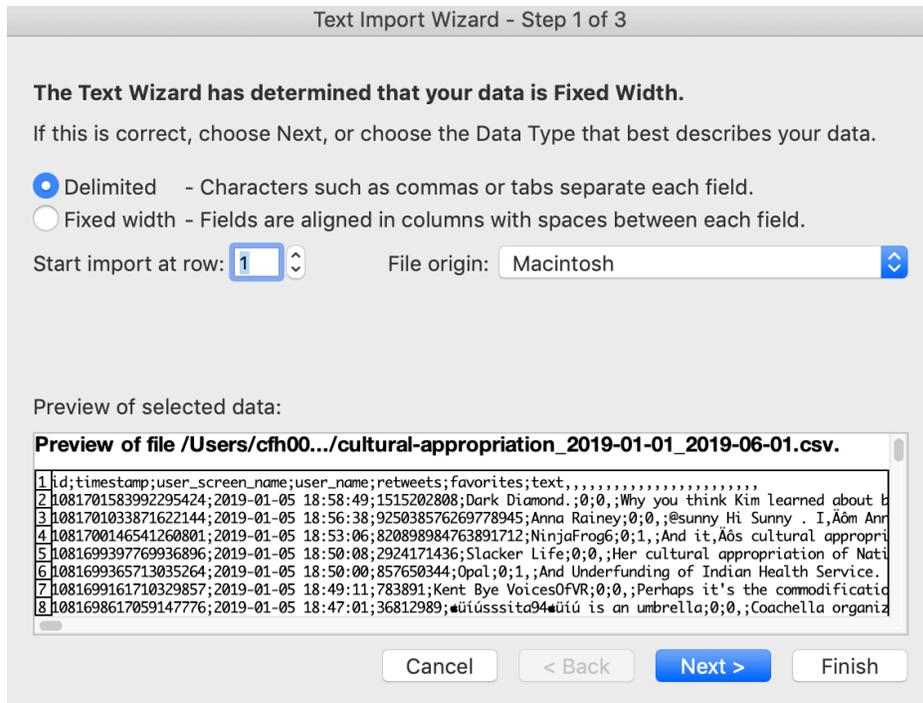
10. The csv file may use commas to separate columns instead of semicolons, which can make your data messy. To fix this, do the following:
  - a. Open a NEW Excel spreadsheet.
  - b. Under the data tab, click the “From text” button:



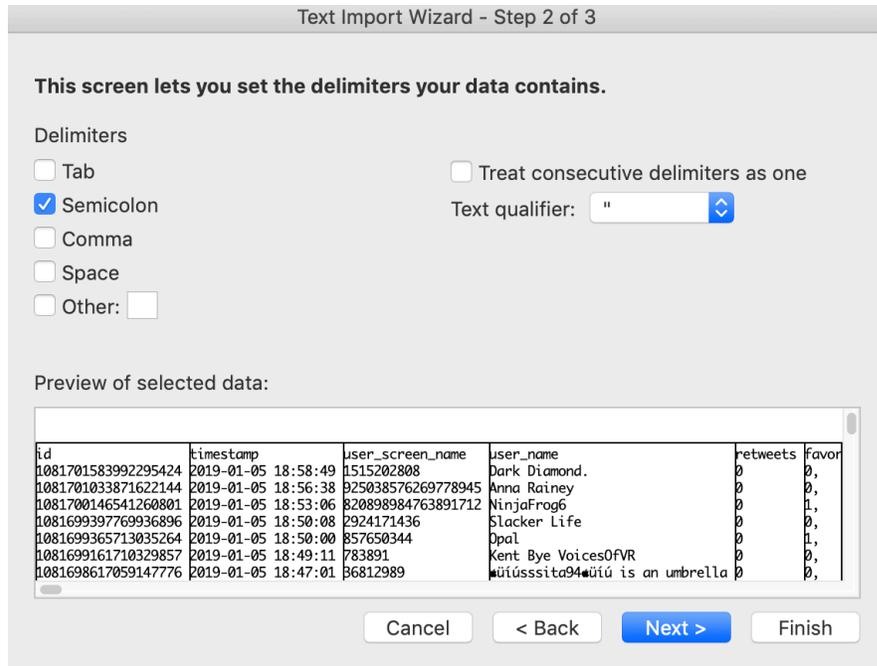
- c. You should receive a pop-up from which to select the file in question. (NB: If you don't, check to see if you already have the file open in a different excel sheet. If so, close this file and repeat this step.)
  - d. Select the file and click “Get Data.”



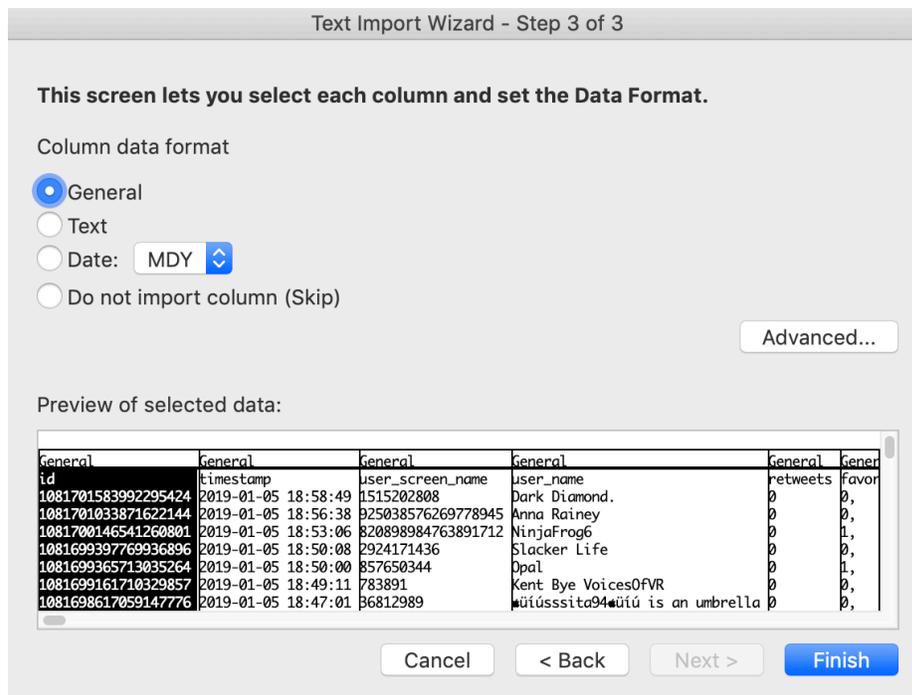
- e. You'll get another pop-up menu called "Text Import Wizard." Select the "Delimited" button and then click "Next."



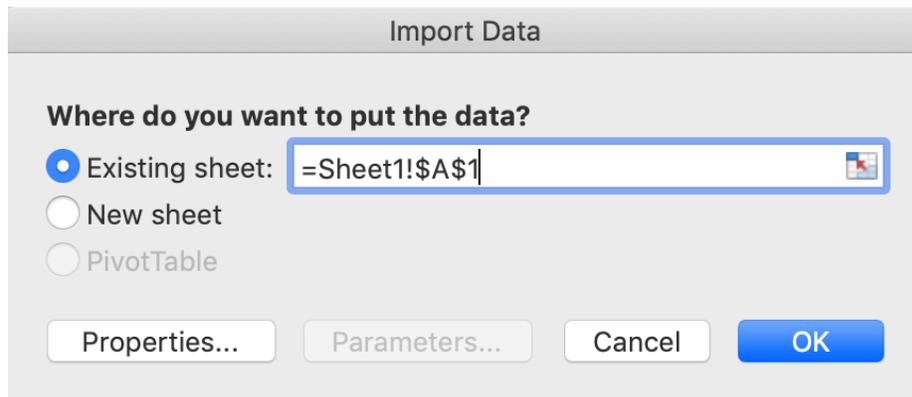
- f. Select the value you'd like to use as the column delimiter in the csv file. For Twitter data, you will most likely want to select "Semicolon." Make sure no other boxes are selected and press "Next."



- g. On the final page of the Text Import Wizard, make sure your “column data format” is checked appropriately. Usually, the “General” button should be fine. Click “Finish.”



- h. You’ll be asked where you want to import your data; the first box of the first column of the existing spreadsheet should be fine. Select this box, then press “OK.”



- i. Your data should import into the new Excel spreadsheet. Save the file as a csv using an appropriate file name.

## Analyzing your data using Twarc

(For more detailed instructions about using DocNow/Twarc, see Christian Howard's blog post, "Twitterature: Mining Twitter Data" at <https://scholarslab.lib.virginia.edu/blog/twitterature-mining-twitter-data/>)

11. If you want more information than that provided by the automatically-downloaded csv file, you'll need to download Twarc by Documenting the Now (DocNow - <https://www.docnow.io/>).
12. To hydrate the txt files (gathered when you scraped Twitter; see Step 9 above) using Twarc, you'll enter the following into your command line: `Twarc hydrate [tweet]_ids.txt > [tweet].jsonl`
  - a. Replace [tweet] with the name of your file
13. Once tweets are hydrated, you should use the "Dedupe" tool in order to ensure that you don't have any duplicates in your data. To do this, enter the following in your command line: `utils/deduplicate.py [tweet].jsonl > [tweet]_deduped.jsonl`
  - a. Replace [tweet] with the name of your file
14. After hydrating and deduping your dataset, you can play with different Twarc possibilities, such as creating word clouds from your tweets and identifying user locations using geojson. See DocNow/Twarc on GitHub for detailed instructions: <https://github.com/DocNow/twarc>.

## Counting your Tweets

15. If you want to find out how many tweets you have in a given search query, you'll need your data in a jsonl file. You'll also want to ensure that you aren't counting

any replicated tweets, so you should deduplicate your file first. To do this, follow instructions 11-13 above.

16. Once you have your deduped jsonl file, navigate to the folder/directory that contains your Python rows counting script (count\_jsonl\_rows.py). It is important that the jsonl file you want to count is in the same folder/directory as this counting script.
17. Enter the following into your command line: python count\_jsonl\_rows.py

```
[Macintosh-6:~ christianhoward$ cd Twitter-Search-API-Python/  
[Macintosh-6:Twitter-Search-API-Python christianhoward$ python count_jsonl_rows.py
```

18. This will prompt you to enter the name of the file.

```
[Macintosh-6:~ christianhoward$ cd Twitter-Search-API-Python/  
[Macintosh-6:Twitter-Search-API-Python christianhoward$ python count_jsonl_rows.py  
enter name of file: deduped_community_combined_2011.jsonl
```

19. After you enter the name of the file, the script will run and then output the “number of rows in the file.” Since each row corresponds to a unique tweet, you now know how many tweets are in the file in question.

```
[Macintosh-6:~ christianhoward$ cd Twitter-Search-API-Python/  
[Macintosh-6:Twitter-Search-API-Python christianhoward$ python count_jsonl_rows.py  
enter name of file: deduped_community_combined_2011.jsonl  
number of rows in file: 1472
```