

Part 1

a) I made a function that creates all of the SQLite tables necessary, including the new Geo table. I call this function at the beginning of the function that populates the database (the function that creates first drops any existing tables). In the Tweets table, I added an attribute `geo_id_str` that will be the foreign key to the Geo table ID attribute (named `tweet_id_str`). I decided to use the `tweet_id_str` as the `geo_id_str` value in tweets that contain Geo information because it is unique per tweet. I do realize this creates some redundancy because tweets with Geo will contain the `id_str` value in both the `id_str` attribute and `geo_id_str`, but I wanted to keep the `geo_id_str` as a separate attribute. Having the `geo_id_str` as a separate attribute in the Tweets table allows for the value to be NULL when Geo is not present in a tweet. This makes finding the number of tweets without Geo easy (just need to look for where `geo_id_str` is NULL). Had I used the actual `id_str` attribute as the foreign key to the Geo table, I would have to have had an entry in the Geo table for every Tweet `id_str`, even if no Geo data was present (would have had to populate the Latitude, Longitude, and Type fields of the Geo table with None). This would cause the Geo table to be much larger than necessary and contain a lot of useless information. I also could have generated a different unique ID to be used as the `geo_id_str` in the Tweets table and the ID in the Geo table (as was done in part 3 for the users table insert statements), but I figured since I knew the Tweets `id_str` was unique already, I would just use it.

b) Below is a screenshot of the text file with the information downloaded from the web. Note that I wrote to the text file in binary format in order to maintain the UTF8 encoding of the web data. Since this text file was just being used to read from and no formatting was given, I assumed that the UTF8 encoding would be acceptable. This allowed me to use the same function to read the data from the text file that I used from the web since they are both in the same format. The final text file size was 3,078,309 KB.

The time to download tweets from web and save to text file: 970.64 secs (or 16 mins and 10.64 secs)

Reading from the text file and saving to SQLite took significantly less time than downloading from the web, which is to be expected (since the text file is local). I used the `c.execute` command during this part also to enter one row at a time to the database.

The number of rows loaded to the Tweets table is 999,553.

The number of rows loaded to the Users table is 855,488.

The number of rows loaded to the Geo table is 24,315.

e) The time to read the tweets from the text file and save to SQLite using 1,000 tweet batching: 101.07 secs (1 min 41 secs).

I used the `c.executemany` in this part to enter 1,000 rows at a time. This was faster than entering each row separately by 20 secs. This verifies that using the `executemany` method is faster.

The number of rows loaded to the Tweets table is 999,553.

The number of rows loaded to the Users table is 855,488.

The number of rows loaded to the Geo table is 24,315.

Part 1 Python Output Screen Captures:

```
IPython console
Console 1/A
Python 3.6.1 |Anaconda 4.4.0 (64-bit)| (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 5.3.0 -- An enhanced Interactive Python.
?                -> Introduction and overview of IPython's features.
%quickref        -> Quick reference.
help            -> Python's own help system.
object?         -> Details about 'object', use 'object??' for extra details.

In [1]: runfile('C:/Users/karip/Desktop/CSC 455 Final/KariPalmier_Final_Part1_InOrIg.py', wdir='C:/Users/karip/Desktop/CSC
455 Final')
Downloading tweets from web and storing to a text file...
Time to download tweets from webpage and save to text file: 970.635909318924 seconds.

Reading tweets from web and storing to SQLite...
Drop Tweets table command skipped
Drop Users table command skipped
Drop Geo table command skipped

Number of rows in Tweets table returned by select all query: 999553

The first 5 rows of the Tweets table are:
('Thu May 29 00:00:43 +0000 2014', '471803285746495489', 'There is no wealth but life. ~John Ruskin #wisdomink', '<a
href="http://www.hootsuite.com" rel="nofollow">HootSuite</a>', None, None, None, 0, None, 213646047, None)
('Thu May 29 00:00:43 +0000 2014', '471803285738106880', 'Mucho la Plop esto, la Plop aquello, pero de los viernes es la
fiesta con la gente más linda. \nEn las otras vienen directo de la frontera.', 'web', None, None, None, 0, None, 38950479,
None)
('Thu May 29 00:00:43 +0000 2014', '471803285767462913', 'motive. When a political idea finds its way into such heads', '<a
href="http://eto-secret4.ru" rel="nofollow">eto prosto NEW secret</a>', None, None, None, 0, None, 2443526930, None)
('Thu May 29 00:00:43 +0000 2014', '471803285750681600', '@im_2realbih bol!', '<a href="http://twitter.com/download/android"
rel="nofollow">Twitter for Android</a>', 290322075, 'im_2realbih', 471800733541486600, 0, None, 284813188, None)
('Thu May 29 00:00:43 +0000 2014', '471803285759078401', 'A veces no entendemos por que, cuando, donde, como y ahora que
hago, por que . Dios es perfecto y a veces... http://t.co/iCyBxph8s9', '<a href="http://www.facebook.com/twitter"
rel="nofollow">Facebook</a>', None, None, None, 0, None, 146270795, None)

Number of rows in Users table returned by select all query: 855488

The first 5 rows of the Users table are:
(850, 'Eugene Ventimiglia', 'eventi', 'Cool Dad, Ingestor of Social Data', 927)
(1503, 'Brij Singh', 'brij', 'Runner, Tinkerer and @techmeme fan. Future is going to be awesome!', 807)
(1541, 'Adam Hertz', 'AdamHertz', 'VP of Engineering, Comcast Silicon Valley', 671)
(2565, 'Nabeel Hyatt', 'nabeel', 'Entrepreneur, Investor, Hardware & Software Geek @sparkcapital. Make a dent.', 454)
(3065, 'Flávia Del Rio', 'flaviadurante', 'Jornalista, DJ e music freak. Paulistana criada em #SANTOSmelhoremtudo', 3988)

Number of rows in Geo table returned by select all query: 24315

The first 5 rows of the Geo table are:
('471803285741916160', 'Point', 14.670275, 121.043955)
('471803289961365504', 'Point', -7.351872, 110.213471)
('471803289961771009', 'Point', 47.8487, -122.222)
('471803294135119872', 'Point', 38.767654, -77.159617)
('471803294130511872', 'Point', -6.149429, 106.728999)

Number of Tweets read: 1000000
Number of Tweets entered into SQLite: 1000000
```

IPython console

Console 1/A

```
Number of Tweets read: 1000000
Number of Tweets entered into SQLite: 1000000
Number of Tweets with JSON errors: 0
Number of Tweets with SQLite errors: 0
Time to download tweets from webpage and save to SQLite: 1092.6567504405975 seconds.
```

Reading tweets from text file and storing to SQLite...

Number of rows in Tweets table returned by select all query: 999553

The first 5 rows of the Tweets table are:

```
('Thu May 29 00:00:43 +0000 2014', '471803285746495489', 'There is no wealth but life. ~John Ruskin #wisdomink', '<a href="http://www.hootsuite.com" rel="nofollow">HootSuite</a>', None, None, None, 0, None, 213646047, None)
('Thu May 29 00:00:43 +0000 2014', '471803285738106880', 'Mucho la Plop esto, la Plop aquello, pero de los viernes es la fiesta con la gente más linda. \nEn las otras vienen directo de la frontera.', 'web', None, None, None, 0, None, 38950479, None)
('Thu May 29 00:00:43 +0000 2014', '471803285767462913', 'motive. When a political idea finds its way into such heads', '<a href="http://eto-secret4.ru" rel="nofollow">eto prosto NEW secret</a>', None, None, None, 0, None, 2443526930, None)
('Thu May 29 00:00:43 +0000 2014', '471803285750681600', '@im_2realbih bol!', '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>', 290322075, 'im_2realbih', 471800733541486600, 0, None, 284813188, None)
('Thu May 29 00:00:43 +0000 2014', '471803285759078401', 'A veces no entendemos por que, cuando, donde , como y ahora que hago, por que . Dios es perfecto y a veces... http://t.co/iCyBxph8s9', '<a href="http://www.facebook.com/twitter" rel="nofollow">Facebook</a>', None, None, None, 0, None, 146270795, None)
```

Number of rows in Users table returned by select all query: 855488

The first 5 rows of the Users table are:

```
(850, 'Eugene Ventimiglia', 'eventi', 'Cool Dad, Ingestor of Social Data', 927)
(1503, 'Brij Singh', 'brij', 'Runner, Tinkerer and @techmeme fan. Future is going to be awesome!', 807)
(1541, 'Adam Hertz', 'AdamHertz', 'VP of Engineering, Comcast Silicon Valley', 671)
(2565, 'Nabeel Hyatt', 'nabeel', 'Entrepreneur, Investor, Hardware & Software Geek @sparkcapital. Make a dent.', 454)
(3065, 'Flávia Del Rio', 'flaviadurante', 'Jornalista, DJ e music freak. Paulistana criada em #SANTOSmelhoremtudo', 3988)
```

Number of rows in Geo table returned by select all query: 24315

The first 5 rows of the Geo table are:

```
('471803285741916160', 'Point', 14.670275, 121.043955)
('471803289961365504', 'Point', -7.351872, 110.213471)
('471803289961771009', 'Point', 47.8487, -122.222)
('471803294135119872', 'Point', 38.767654, -77.159617)
('471803294130511872', 'Point', -6.149429, 106.728999)
```

```
Number of Tweets read: 1000000
Number of Tweets entered into SQLite: 1000000
Number of Tweets with JSON errors: 0
Number of Tweets with SQLite errors: 0
Time to upload tweets from text file and save to SQLite: 121.26027369499207 seconds.
```

Reading tweets from text file and storing to SQLite in batches of 1000...

Number of rows in Tweets table returned by select all query: 999553

Number of rows in Tweets table returned by select all query: 999553

The first 5 rows of the Tweets table are:

```
('Thu May 29 00:00:43 +0000 2014', '471803285746495489', 'There is no wealth but life. ~John Ruskin #wisdomink', '<a href="http://www.hootsuite.com" rel="nofollow">HootSuite</a>', None, None, None, 0, None, 213646047, None)
('Thu May 29 00:00:43 +0000 2014', '471803285738106880', 'Mucho la Plop esto, la Plop aquello, pero de los viernes es la fiesta con la gente más linda. \nEn las otras vienen directo de la frontera.', 'web', None, None, None, 0, None, 38950479, None)
('Thu May 29 00:00:43 +0000 2014', '471803285767462913', 'motive. When a political idea finds its way into such heads,', '<a href="http://eto-secret4.ru" rel="nofollow">eto prosto NEW secret</a>', None, None, None, 0, None, 2443526930, None)
('Thu May 29 00:00:43 +0000 2014', '471803285750681600', '@im_2realbih bol!', '<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>', 290322075, 'im_2realbih', 471800733541486600, 0, None, 284813188, None)
('Thu May 29 00:00:43 +0000 2014', '471803285759078401', 'A veces no entendemos por que, cuando, donde , como y ahora que hago, por que . Dios es perfecto y a veces... http://t.co/iCy8xph8s9', '<a href="http://www.facebook.com/twitter" rel="nofollow">Facebook</a>', None, None, None, 0, None, 146270795, None)
```

Number of rows in Users table returned by select all query: 855488

The first 5 rows of the Users table are:

```
(850, 'Eugene Ventimiglia', 'eventi', 'Cool Dad, Ingestor of Social Data', 927)
(1503, 'Brij Singh', 'brij', 'Runner, Tinkerer and @techmeme fan. Future is going to be awesome!', 807)
(1541, 'Adam Hertz', 'AdamHertz', 'VP of Engineering, Comcast Silicon Valley', 671)
(2565, 'Nabeel Hyatt', 'nabeel', 'Entrepreneur, Investor, Hardware & Software Geek @sparkcapital. Make a dent.', 454)
(3065, 'Flávia Del Rio', 'flaviadurante', 'Jornalista, DJ e music freak. Paulistana criada em #SANTOSmelhoremtudo', 3988)
```

Number of rows in Geo table returned by select all query: 24315

The first 5 rows of the Geo table are:

```
('471803285741916160', 'Point', 14.670275, 121.043955)
('471803289961365504', 'Point', -7.351872, 110.213471)
('471803289961771009', 'Point', 47.8487, -122.222)
('471803294135119872', 'Point', 38.767654, -77.159617)
('471803294130511872', 'Point', -6.149429, 106.728999)
```

Number of Tweets read: 1000000

Number of Tweets entered into SQLite: 1000000

Number of Tweets with JSON errors: 0

Number of Tweets with SQLite errors: 0

Time to upload tweets from text file and save to SQLite using 1000 tweet batches: 101.06697130203247 seconds.

Summary:

Time to download tweets from webpage and save to text file: 970.635909318924 seconds.

Time to download tweets from webpage and save to SQLite: 1092.6567504405975 seconds.

Time to upload tweets from text file and save to SQLite: 121.26027369499207 seconds.

Time to upload tweets from text file and save to SQLite using 1000 tweet batches: 101.06697130203247 seconds.

In [2]: |

Part 2

a)

i) Instead of returning the entire tweet, I returned the tweet id_str of all tweets where the tweet id_str contained 44 or 77. I displayed the first 5 tweet IDs to show that they did in fact contain 44 or 77 in the Python output at the end of this section. The total number of tweet id_str attributes containing 44 or 77 was 222,673.

The time to run the query for tweet id_str containing 44 or 77 from SQLite: 0.41 secs

ii) The number of unique values of the in_reply_to_user_id entries was 180,405.

The time to run the query for unique in_reply_to_user_id from SQLite: 0.76 secs

iii) I displayed the first 5 tweets that had the shortest, longest, and average text message length in the Python output at the end of this section. The shortest text message length was 1 character. There were 947 tweets with this length text message. The longest text message length was 434 characters. There was 1 tweet with this length text message. The average text message length was 68. There were 8,404 tweets with this length text message. Note that the average returned by the AVG function in the inner query was a floating-point number. Since a message cannot contain a fraction of a character, I casted this value into an integer in the inner query using SELECT CAST(AVG(LENGTH(text)) AS INT). All of the tweets with text message lengths equal to this integer casted average value were then returned.

The time to run the query for all messages with the shortest message length: 0.830 secs

The time to run the query for all messages with the longest message length: 0.833 secs

The time to run the query for all messages with the average message length: 0.844 secs

iv) I displayed the first 5 values of the average latitude and longitude per user in the Python output at the end of this section. Since the Geo table I created only contains entries with latitude and longitude data present, this query only returned average latitude and longitude values that existed (in other words, it did not include users with no geo information). The number of user name, user ID, average latitude, and average longitude entries returned from the query was 22,648. This was not equal to the number of Geo table rows because some users had more than one entry in the Geo table (some users had more than one tweet, each of which had a different location value stored to the Geo table).

The time to run the query for the average latitude and longitude per user: 0.36 secs

v) The time it took to run the query for the average latitude and longitude per user 10 times: 3.38 secs. The time it took to run the same query 100 times: 32.31 secs. $32.31/3.38 = 9.56$, so the time increased 9.56 times from 10 times to 100 times. 9.56 is approximately 10 when rounded, so the time increase between running 10 times and 100 times can be considered to be 10 times longer.

b)

i) I also returned just the tweet id_str value in this section (as I did in section a i). Because duplicate tweets were present in the data saved to the text file (data was saved directly from the web to the text file without filtering), I had to keep track of the unique tweet id_str values in Python in this section to

ensure that I only returned information for unique tweets that contained 44 or 77 in their `id_str` attribute. I displayed the first 5 tweet IDs to show that they did in fact contain 44 or 77 in the Python output at the end of this section. The total number of tweet `id_str` attributes containing 44 or 77 was 222,673, which matched the number returned by SQLite in section a i.

The time to run the query for tweet `id_str` containing 44 or 77 from the text file: 738.84 secs (or 12 mins 18.84 secs). Note that this time is significantly longer than the time it took to get the same information by querying SQLite (approx. 13 min in Python versus less than 1 min in SQLite).

ii) The number of unique values of the `in_reply_to_user_id` entries was 180,405, which matched section a ii. Note that I had to keep track of unique `in_reply_to_user_id` values here similarly to as I did with the tweet `id_str` values in section b i to ensure the only unique values were counted in the query output.

The time to run the query for unique `in_reply_to_user_id` from the text file: 1591.29 secs (or 26 mins 31.29 secs). Note that this time is significantly longer than the time it took to get the same information by querying SQLite (approx. 27 min in Python versus less than 1 min in SQLite).

Part 2 Python Output Screen Captures:




```
Number of tweets with unique user ID and average latitude and longitude: 22648
The first 5 user names, ids, and average latitude and longitude values from SQL are:
User ID: 7819, User Name:Cody Landefeld, Average Latitude: 33.48155, Average Longitude: -112.073076
User ID: 231363, User Name:Molly Black, Average Latitude: 37.889099, Average Longitude: -122.316796
User ID: 355203, User Name:Jacqui Maher, Average Latitude: 40.6895, Average Longitude: -73.973056
User ID: 647853, User Name:ゼビウス, Average Latitude: 4.691722, Average Longitude: -74.034499
User ID: 970871, User Name:Sam Gerstenzang, Average Latitude: 37.369168, Average Longitude: -122.144176
Time to run query to average latitude and longitude per user: 0.3578021526 seconds.
```

```
Time to run query to average latitude and longitude per user 10x: 3.3794162273 seconds.
Time to run query to average latitude and longitude per user 100x: 32.3099372387 seconds.
```

```
Number of tweet ID strings with either 44 or 77 in them from file using Python: 222673
The first 5 tweet IDs with 44 or 77 in them read from file using Python are:
Matching entry 0 ID = 7819
Matching entry 1 ID = 231363
Matching entry 2 ID = 355203
Matching entry 3 ID = 647853
Matching entry 4 ID = 970871
Time to run query to find tweet id_str entriess with 44 or 77 from file using Python: 738.8439435959
seconds.
```

```
Number of unique in_reply_to_user_id entries from file using Python: 180405
Time to run query to find number of unique in_reply_to_user_id entries from file using Python:
1591.2853627205 seconds.
```

```
In [2]: |
```

Part 3

a) I created the unique user ID strings requested in this section by starting with a base string of AAAA and incrementing the letters each time through the loop. The letters are incremented from the right most first. When this character reaches the last possible letter (z), it is reset to A and the next character to the left is incremented by 1 (results in AAaz, then AABA). I used a 4-character ID because there are 52 total character combinations (A-Z and a-z). The number of possible combinations with a 3-character ID was $52^3 = 104,609$ combinations. This was not enough. The number of combinations with 4 characters was $52^4 = 7,311,616$, which is more than enough for what I need (only need 855,488 unique IDs).

I also had to convert the UTF8 binary data in the text messages, screen names, and names into printable text before writing the INSERT statements to the text file. This is because the formatting of the output file needs to have one INSERT per line (as if someone would copy and paste them into another database). Using a binary text file would not allow for each insert on a separate line (\n in a binary file is just another character and is not treated as a new line as it is in a text formatted file). Any UTF8 non-printable characters were converted to their \x representation by converting the string containing them to a byte array, then making that into a string, and lastly removing the b' and ' from the start and end of the string converted byte array.

The time to create the INSERT statements and save them to a text file using the SQLite database: 13.86 secs. The number of insert statements generated was 855,488. The output UserInserts_FromSQL.txt text file size is 210,712 KB.

```
UserInserts_FromSQL.txt - Notepad
File Edit Format View Help

INSERT INTO Users VALUES('AAAB', 850, 'Eugene Ventimiglia', 'eventi', 'Cool Dad, Ingestor of Social Data', 927);
INSERT INTO Users VALUES('AAAC', 1503, 'Brij Singh', 'brij', 'Runner, Tinkerer and @techmeme fan. Future is going to be awesome!', 807);
INSERT INTO Users VALUES('AAAD', 1541, 'Adam Hertz', 'AdamHertz', 'VP of Engineering, Comcast Silicon Valley', 671);
INSERT INTO Users VALUES('AAAE', 2565, 'Nabeel Hyatt', 'nabeel', 'Entrepreneur, Investor, Hardware & Software Geek @sparkcapital. Make a dent.', 454);
INSERT INTO Users VALUES('AAAF', 3065, 'Fl\xxc3\xa1via Del Rio', 'flaviadurante', 'Jornalista, DJ e music freak. Paulistana criada em #SANTOSmelhoremtudo', 3988);
INSERT INTO Users VALUES('AAG', 3271, 'deeje', 'deeje', 'Maker of elegant mobile-cloud user experiences. Recent works include tappr.tv, Blab, Hello Vino, Thirst, and Biophilia. #ios #boarder #wine #climber #cod', 1284);
INSERT INTO Users VALUES('AAAH', 3300, 'Jonathan Wight', 'schwa', 'No.', 532);
INSERT INTO Users VALUES('AAAI', 4233, 'Sean Oliver', 'Sean_Oliver', 'Sean Oliver is a consultant in Seattle, WA', 1231);
INSERT INTO Users VALUES('AAAJ', 7819, 'Cody Landefeld', 'codyl', '#Creative problem solver. Adopted by Christ, Husband to @raquelandefeld. Father to 3. Director at @modeeffect #WordPress #UX', 1063);
INSERT INTO Users VALUES('AAAK', 10399, 'John Infante', 'John Infante', 'Occasionally critical, often supportive, and never dumbed down', 524);
INSERT INTO Users VALUES('AAAL', 11036, 'New Zealand', 'newzealand', '"..we're always in other places, lost, like sheep." -- Janet Frame', 1630);
INSERT INTO Users VALUES('AAAM', 11957, 'earnest sewn', 'earnestsewn', 'Born in New York. We live for what we create. We desire to do great work and produce things that are true and will stand. We are earnest sewn.', 644);
INSERT INTO Users VALUES('AAAN', 12350, 'ian kennedy', 'iankennedy', 'product guy at http://gigaom.com', 420);
INSERT INTO Users VALUES('AAAO', 12514, 'Tom Coates', 'tomcoates', 'The personal Twitter account of Tom Coates, co-founder of Product Club: a new product development and invention company. Prev: Brickhouse, Fire Eagle, BBC', 816);
INSERT INTO Users VALUES('AAAP', 12574, 'DD', 'devildoll', 'Fueled by vegetables.', 206);
INSERT INTO Users VALUES('AAAQ', 12720, 'Wil Alambre', 'wilalambre', 'I'm a web developer, comicbook reader, movie lover, music listener, amateur fiction writer, casual videogamer, avid roleplayer, and all around okay kinda guy.', 313);
INSERT INTO Users VALUES('AAAR', 12727, 'AmandaHi', 'amandahi', 'runner, baker, lover of data, hater of deer, hopelessly Type A, program manager @washingtonpost', 620);
INSERT INTO Users VALUES('AAAS', 13374, 'Charles Edward Frith', 'charlesfrith', 'What cannot be said, above all must not be silenced, but written. \xe2\x80\x94 Jacques Derrida', 953);
INSERT INTO Users VALUES('AAAT', 13717, 'Natalie Luhrs', 'eilatan', 'Supervillain book reviewer, spreadsheet wrangler, knitter, spinner, geek. Acquisitions Editor for Masque Books.', 523);
INSERT INTO Users VALUES('AAAU', 16263, 'Matthew Oliphant', 'matto', 'Works at @ngenworks. Runs @RefreshPDX. Believes Android > iOS. Talked in 3rd-person before it was cool. Was just informed it isn't cool.', 286);
INSERT INTO Users VALUES('AAAV', 19783, 'Mark Allen', 'moustache', 'Hand-crafted tweets since 2006.', 294);
INSERT INTO Users VALUES('AAAW', 21633, 'Claire Armstrong', 'armst', 'UI \xe2\x80\xa2 UX \xe2\x80\xa2 HTML \xe2\x80\xa2 CSS \xe2\x80\xa2 LA \xe2\x80\xa2 RPGs \xe2\x80\xa2 \xe2\x80\xa2', 430);
INSERT INTO Users VALUES('AAAX', 22203, 'Josh Gee', 'jgee', 'earnestness, anxiety, something', 2531);
INSERT INTO Users VALUES('AAAY', 35383, 'David Chartier', 'chartier', 'Content Strategist, Writer, Tech Distiller. I run http://FinerTech.com, contribute to @Macworld @Mircutter @MacObserver. Herald for http://1Password.com.', 759);
INSERT INTO Users VALUES('AAAZ', 36823, 'Anil Dash', 'anildash', 'Cofounder @thinkup & @activateinc \xe2\x80\xa2 Blog at http://dashes.com \xe2\x80\xa2 anil@dashes.com \xe2\x80\xa2 646 833-8659 \xe2\x80\xa2 The intern who maintains this account has been fired.', 2313);
INSERT INTO Users VALUES('AAAB', 37613, 'Matthew Caldecutt', 'mcaldecutt', 'Live in Harlem. Specialize in B2B (ad/marketing) and tech PR. Work @blastpr. Opinions? My own.', 3634);
INSERT INTO Users VALUES('AAAb', 38323, 'FoxyStardust', 'FoxyStardust', 'I saw a rumor on the Internet that I'm dead.', 152);
INSERT INTO Users VALUES('AAAc', 39563, 'alovedlife', 'alovedlife', 'A free spirit, with a wild heart.', 531);
```

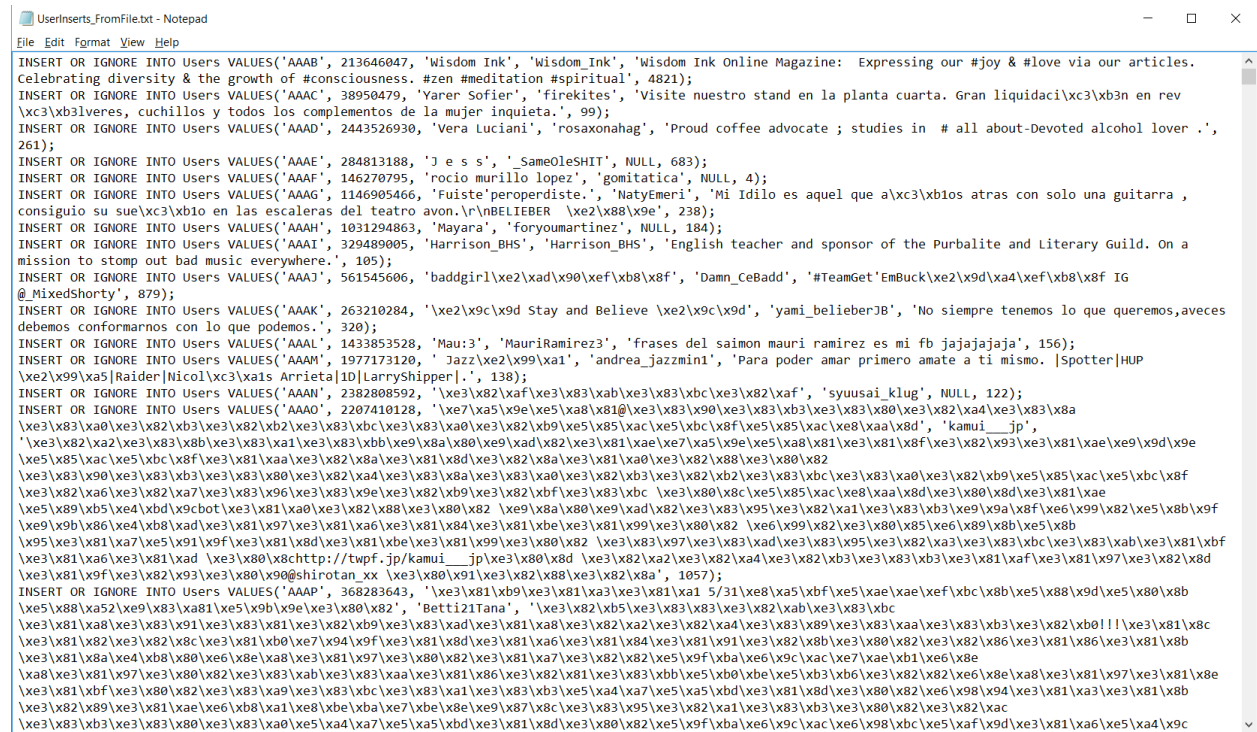
b) Since the text file containing the tweets contained duplicate entries, I performed this part 2 different ways. The first way was to generate INSERT OR IGNORE INTO statements for each tweet. Using the

INSERT OR IGNORE INTO would tell the new database these statements are being used to populate to ignore any duplicate user IDs being inserted instead of erroring. This would in effect product the same new table as in section a since duplicates would not be entered. The second way was to generate the exact same INSERT statements that were generated in section a (one for each unique user ID). This way took much longer because I had to keep track of all of the user IDs that already had INSERT statements generated for them, which meant maintain a large list of existing IDS. Note that the same conversion from binary to \x format was required during this section as well.

Time to create INSERT OR IGNORE statements and save them to a text file by reading tweets from text file and using Python: 74.67 secs (or 1 min 14.67 secs). Number of insert or ignore statements generated was 1,000,000. The output UserInserts_FromFile.txt text file size is 258,112 KB. Note that creating the insert statements by reading from the text file and creating INSERT OR IGNORE statements was considerably slower than by getting the information from SQLite (75 secs compared to 14 secs).

Time to create INSERT statements and save them to a text file by reading tweets from text file and using Python (only writing unique users): 9528.82 secs (or 2 hrs 38 mins 48.82 secs). Number of insert statements generated was 855,488. Note that this is much longer than creating the INSERT OR IGNORE statements, and is therefore even longer than creating INSERT statements from SQLite. The output UserInserts_FromFile_Uniq.txt text file size is 210,712 KB.

INSERT OR IGNORE file contents:



Unique INSERT file contents:


```
UserInserts_FromFile_Uniq.txt - Notepad
File Edit Format View Help

[INSERT INTO Users VALUES('AAAB', 213646047, 'Wisdom Ink', 'Wisdom Ink', 'Wisdom Ink Online Magazine: Expressing our #joy & #love via our articles. Celebrating diversity & the growth of #consciousness. #zen #meditation #spiritual', 4821);
INSERT INTO Users VALUES('AAAC', 38950479, 'Yarer Sofier', 'firekites', 'Visite nuestro stand en la planta cuarta. Gran liquidaci\xc3\xb3n en rev\xc3\xb3lveres, cuchillos y todos los complementos de la mujer inquieta.', 99);
INSERT INTO Users VALUES('AAD', 2443526930, 'Vera Luciani', 'rosaxonahag', 'Proud coffee advocate ; studies in # all about-Devoted alcohol lover .', 261);
INSERT INTO Users VALUES('AAAE', 284813188, 'J e s s', 'SameOLESHIT', NULL, 683);
INSERT INTO Users VALUES('AAAF', 146270795, 'roci murillo lopez', 'gomitatica', NULL, 4);
INSERT INTO Users VALUES('AAG', 1146905466, 'Fuiste'peroperdiste.', 'NatyEmeri', 'Mi Idilo es aquel que a\xc3\xb1os atras con solo una guitarra , consiguio su sue\xc3\xb1o en las escaleras del teatro avon.\r\nBELIEBER \xe2\x88\x9e', 238);
INSERT INTO Users VALUES('AAAH', 1031294863, 'Mayara', 'foryoumartinez', NULL, 184);
INSERT INTO Users VALUES('AAAI', 329489005, 'Harrison_BHS', 'Harrison_BHS', 'English teacher and sponsor of the Purbalite and Literary Guild. On a mission to stomp out bad music everywhere.', 105);
INSERT INTO Users VALUES('AAAJ', 561545606, 'baddgirl\xe2\xad\x90\xe2\x88\x9f', 'Damn ceBadd', '#TeamGetEmBuck\xe2\x9d\xad\xe2\x88\x9f IG @MixedShorty', 879);
INSERT INTO Users VALUES('AAAK', 263210284, '\xe2\x9c\x9d Stay and Believe \xe2\x9c\x9d', 'yami_belieberJB', 'No siempre tenemos lo que queremos,aveces debemos conformarnos con lo que podemos.', 320);
INSERT INTO Users VALUES('AAL', 1433853528, 'Mau:3', 'MauriRamirez3', 'frases del saimon mauri ramirez es mi fb jajajajaja', 156);
INSERT INTO Users VALUES('AAM', 1977173120, 'Jazz\xe2\x99\xa1', 'andrea_jazzmini1', 'Para poder amar primero amate a ti mismo. |Spotter|HUP\xe2\x99\xa5Raider|Nicol\xc3\xa1s Arrieta|D|LarryShipper|.', 138);
INSERT INTO Users VALUES('AAN', 2382808592, '\xe3\x82\xaf\xe3\x83\xab\xe3\x83\xbc\xe3\x82\xaf', 'syuusai_klug', NULL, 122);
INSERT INTO Users VALUES('AAAO', 2207410128, '\xe7\xa5\x9e\xe5\xa8\x81\xe3\x83\x90\xe3\x83\xb3\xe3\x83\x80\xe3\x82\xa4\xe3\x83\xa8\xe3\x83\x82\xb3\xe3\x82\xb2\xe3\x83\xbc\xe3\x82\xb9\xe5\x85\xac\xe5\xbc\x8f\xe5\x85\xac\xe8\xaa\x8d', 'kamui_jp', '\xe3\x82\xa2\xe3\x83\xb8\xe3\x83\xbb\xe9\x8a\x80\xe9\xad\x82\xe3\x81\xae\xe7\xa5\x9e\xe5\xa8\x81\xe3\x81\x8f\xe3\x82\x93\xe3\x81\xae\xe9\x9d\x9e\xe5\x85\xac\xe5\xbc\x8f\xe3\x81\xaa\xe3\x82\x8a\xe3\x81\x8d\xe3\x82\x8a\xe3\x81\xa0\xe3\x82\x88\xe3\x80\x82\xe3\x83\x90\xe3\x83\xb3\xe3\x83\x80\xe3\x82\xa4\xe3\x83\xa8\xe3\x83\xa0\xe3\x82\xb3\xe3\x82\xb2\xe3\x83\xbc\xe3\x82\xb9\xe5\x85\xac\xe5\xbc\x8f\xe3\x82\xa6\xe3\x82\xa7\xe3\x83\x96\xe3\x83\x9e\xe3\x82\xb9\xe3\x82\xbf\xe3\x83\xbc \xe3\x80\x8c\xe5\x85\xac\xe8\xaa\x8d\xe3\x80\x8d\xe3\x81\xae\xe5\x89\xb5\xe4\xbd\x9cbot\xe3\x81\xa0\xe3\x82\x88\xe3\x80\x82 \xe9\x8a\x80\xe9\xad\x82\xe3\x83\x95\xe3\x82\xa1\xe3\x83\xb3\xe9\x9a\x8f\xe6\x99\x82\xe5\x8b\x9f\xe9\x9b\x86\xe4\xb8\xad\xe3\x81\x97\xe3\x81\xa6\xe3\x81\x84\xe3\x81\xbe\xe3\x81\x99\xe3\x80\x82 \xe6\x99\x82\xe3\x80\x85\xe6\x89\x8b\xe5\x8b\xe9\x9b\x81\xa7\xe5\x91\x9f\xe3\x81\x8d\xe3\x81\xbe\xe3\x81\x99\xe3\x80\x82 \xe3\x83\x97\xe3\x83\xad\xe3\x83\x95\xe3\x82\xa3\xe3\x83\xbc\xe3\x83\xab\xe3\x81\xbf\xe3\x81\xa6\xe3\x81\xad \xe3\x80\x8chttp://twpf.jp/kamui_jp\xe3\x80\x8d \xe3\x82\xa2\xe3\x82\xa4\xe3\x82\xb3\xe3\x83\x93\xe3\x81\xaf\xe3\x81\x97\xe3\x82\x8d\xe3\x81\x9f\xe3\x82\x93\xe3\x80\x90shirotan_xx \xe3\x80\x91\xe3\x82\x88\xe3\x82\x8a', 1057);
INSERT INTO Users VALUES('AAP', 368283643, '\xe3\x81\xb9\xe3\x81\xa3\xe3\x81\xa1 5/31\xe8\xa5\xbf\xe5\xae\xae\xe2\x8b\xe5\x88\x9d\xe5\x80\x8b\xe5\x88\xa52\xe9\x83\xa81\xe5\x9b\x9e\xe3\x80\x82', 'Betti21Tana', '\xe3\x82\xb5\xe3\x83\x83\xe3\x82\xab\xe3\x83\xbc\xe3\x81\xa8\xe3\x83\x91\xe3\x83\x81\xe3\x82\xb9\xe3\x83\xad\xe3\x81\xa8\xe3\x82\xa2\xe3\x82\xa4\xe3\x83\x89\xe3\x83\xaa\xe3\x83\xb3\xe3\x82\xb0!!!\xe3\x81\x8c\xe3\x81\x82\xe3\x82\x8c\xe3\x81\xb0\xe7\x9d\xe3\x81\x8d\xe3\x81\xa6\xe3\x81\x84\xe3\x81\x91\xe3\x82\x8b\xe3\x80\x82\xe3\x82\x86\xe3\x81\x86\xe3\x81\x8b\xe3\x81\x8a\xe4\xb8\x80\xe6\x8e\xa8\xe3\x81\x97\xe3\x80\x82\xe3\x81\xa7\xe3\x82\x82\xe5\x9f\xba\xe6\x9c\xac\xe7\xae\xb1\xe6\x8e\xa8\xe3\x81\x97\xe3\x80\x82\xe3\x83\xab\xe3\x83\x8a\xe3\x81\x86\xe3\x82\x81\xe3\x83\xbb\xe5\xb0\xbe\xe5\xb3\xb6\xe3\x82\x82\xe6\x8e\xa8\xe3\x81\x97\xe3\x81\x8e\xe3\x81\xbf\xe3\x80\x82\xe3\x83\xa9\xe3\x83\xbc\xe3\x83\xa1\xe3\x83\xb3\xe5\xa4\xa7\xe5\xa5\xbd\xe3\x81\x8d\xe3\x80\x82\xe6\x98\x94\xe3\x81\xa3\xe3\x81\x8b\xe3\x82\x89\xe3\x81\xae\xe6\x8b\xa1\xe8\xbe\xba\xe7\xbe\x8e\xe9\x87\x8c\xe3\x83\x95\xe3\x82\xa1\xe3\x83\xb3\xe3\x80\x82\xe3\x82\xac\xe3\x83\xb3\xe3\x83\x80\xe3\x83\xa0\xe5\xa4\xa7\xe5\xa5\xbd\xe3\x81\x8d\xe3\x80\x82\xe5\x9f\xba\xe6\x9c\xac\xe6\x98\xbc\xe5\xaf\x9d\xe3\x81\xa6\xe5\xa4\x9c\xe4\xbd\x95\xe4\xba\x8b\xe4\xbd\xad\xe3\x80\x82', 438);
INSERT INTO Users VALUES('AAQ', 174088334, '\xe3\x81\xaa\xe3\x81\xaa\xe3\x81\x86\xe3\x81\xbf', 'ubutora', '\xe3\x81\x8a\xe5\x89\x8d\xe3\x82\x89\xe3\x81\x8c
```

Part 3 Python Output Screen Captures:

```
Python console
NOTE: The Python console is going to be REMOVED in Spyder 3.2. Please start to migrate your work to the IPython console instead.

Python 3.6.1 [Anaconda 4.4.0 (64-bit)] (default, May 11 2017, 13:25:24) [MSC v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> runfile('C:/Users/karip/Desktop/CSC 455 Final/KariPalmier_Final_Part3.py', wdir='C:/Users/karip/Desktop/CSC 455 Final')

The number of insert statements created from SQLite = 855488
The time to create a text file with insert statements for every SQLite Users table row: 13.971144199371338 seconds.

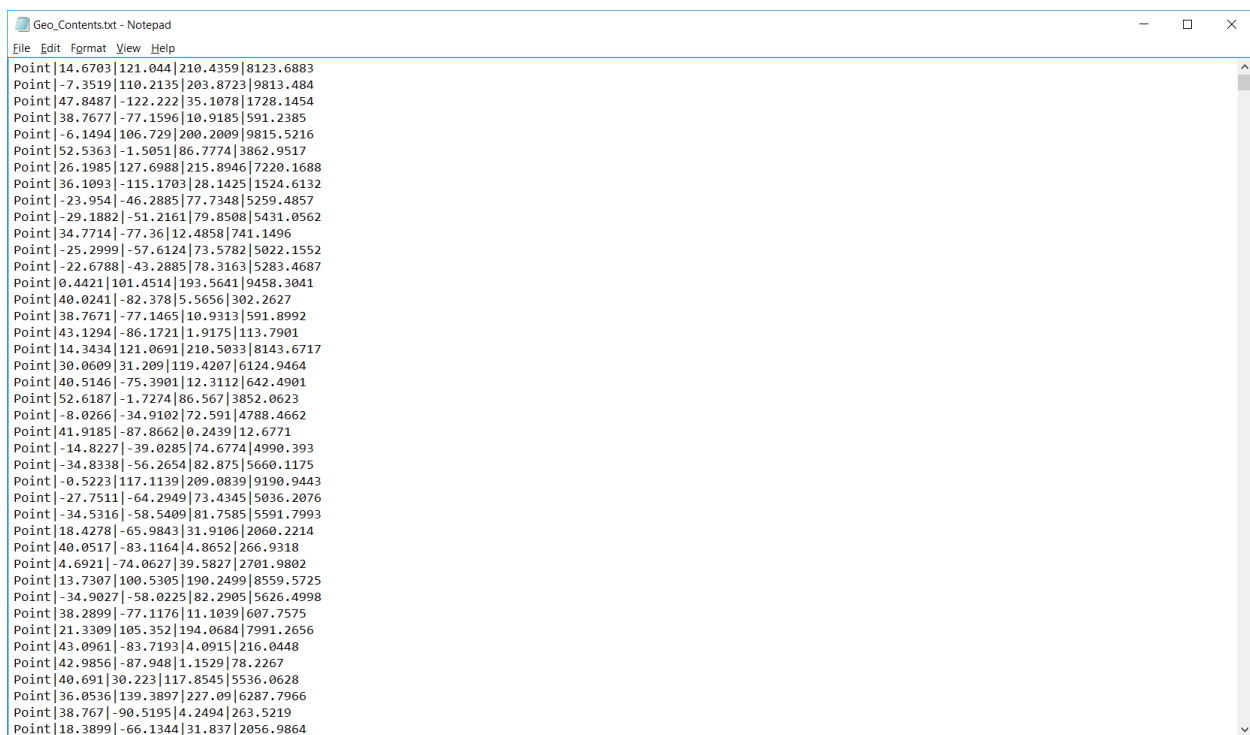
The number of insert statements with ignore option created from the tweet file = 1000000
The time to create a text file with insert statements with ignore option for every tweet file Users dict entry: 74.66609644889832 seconds.

The number of insert statements created from the tweet file = 855488
The time to create a text file with insert statements for every tweet file Users dict entry: 9528.819977998734 seconds.
>>>
```

Part 4

a) I included both Euclidean and actual miles between latitude and longitude points in my Geo contents table. The formula I implemented for the distance between CDM and the latitude and longitude data for each Geo table entry is below (taken from the <http://www.movable-type.co.uk/scripts/latlong.html> website). For the distance between CDM and 14.670275, 121.043955, my code calculated 8123.68 miles (13,073.80 km). The website I used for verification calculated 8,123.68 miles as well (screen capture is shown). I also verified this calculation using the distance calculation on the website I got the formulas from, which was within 3 kilometers. Since my calculations matched one website and were very close to the second (within 3 km or 1.84 miles), I believe my calculations to be valid.

The time it took to create the file with the Geo table contents: 0.54 secs. The number of Geo table rows written: 24,315. The Geo contents text file size is 983 KB.



```
Geo_Contents.txt - Notepad
File Edit Format View Help
Point|14.6703|121.044|210.4359|8123.6883
Point|-7.3519|110.2135|203.8723|9813.484
Point|47.8487|-122.222|35.1078|1728.1454
Point|38.7677|-77.1596|10.9185|591.2385
Point|-6.1494|106.729|200.2009|9815.5216
Point|52.5363|-1.5051|86.7774|3862.9517
Point|26.1985|127.6988|215.8946|7220.1688
Point|36.1093|-115.1703|28.1425|1524.6132
Point|-23.954|-46.2885|77.7348|5259.4857
Point|-29.1882|-51.2161|79.8508|5431.0562
Point|34.7714|-77.36|12.4858|741.1496
Point|-25.2999|-57.6124|73.5782|5022.1552
Point|-22.6788|-43.2885|78.3163|5283.4687
Point|0.4421|101.4514|193.5641|9458.3041
Point|40.0241|-82.378|5.5656|302.2627
Point|38.7671|-77.1465|10.9313|591.8992
Point|43.1294|-86.1721|1.9175|113.7901
Point|14.3434|121.0691|210.5033|8143.6717
Point|30.0609|31.209|119.4207|6124.9464
Point|40.5146|-75.3901|12.3112|642.4901
Point|52.6187|-1.7274|86.567|3852.0623
Point|-8.0266|-34.9102|72.591|4788.4662
Point|41.9185|-87.8662|0.2439|12.6771
Point|-14.8227|-39.0285|74.6774|4990.393
Point|-34.8338|-56.2654|82.875|5660.1175
Point|-0.5223|117.1139|209.0839|9190.9443
Point|-27.7511|-64.2949|73.4345|5036.2076
Point|-34.5316|-58.5409|81.7585|5591.7993
Point|18.4278|-65.9843|31.9106|2060.2214
Point|40.0517|-83.1164|4.8652|266.9318
Point|4.6921|-74.0627|39.5827|2701.9802
Point|13.7307|100.5305|190.2499|8550.5725
Point|-34.9027|-58.0225|82.2905|5626.4998
Point|38.2899|-77.1176|11.1039|607.7575
Point|21.3309|105.352|194.0684|7991.2656
Point|43.0961|-83.7193|4.0915|216.0448
Point|42.9856|-87.948|1.1529|78.2267
Point|40.691|30.223|117.8545|5536.0628
Point|36.0536|139.3897|227.09|6287.7966
Point|38.767|-90.5195|4.2494|263.5219
Point|18.3899|-66.1344|31.837|2056.9864
```

Greater circle distance calculations between two latitude and longitude points (phi = latitude, lambda = longitude):

$$a = \sin^2(\Delta\phi/2) + \cos \phi_1 \cdot \cos \phi_2 \cdot \sin^2(\Delta\lambda/2)$$

$$c = 2 \cdot \text{atan2}(\sqrt{a}, \sqrt{1-a})$$

$$d = R \cdot c$$

Webpage calculation between CDM and first latitude and longitude point of output file (latitude and longitude not rounded):

Thursday June 8, 2017 | Home | Search | FAQ | Message

Welcome to OnlineConversion.com

Great Circle distance and midpoint between Latitude/Longitude points

This will compute the [great-circle](#) distance between two latitude/longitude points, as well as the middle point. The script uses "Haversine" formula, which results in in approximations less than 1%.

Enter either:

- decimal latitudes/longitudes with minus sign for South and West
- degrees minutes seconds in a format like **E 32 14 9** (32°14'9" East longitude).

	Point 1	Point 2
Latitude	41.878668	14.670275
Longitude	-87.625555	121.043955
Distance	8123.688320233664 mile ▼	
Middle point	N 62 40 47.22 , W 190 17 1.57	

(Courtesy of Google Maps)

b) The number of tweets with known locations was 24,315. The number of tweets with unknown locations was 975,238. The percentage of tweets with location available was 2.43%.

The time it took to create the file with Tweet contents and user name and screen_name: 29.51 secs.
The number of Tweet table rows written: 999,553. The Tweet contents text file size is 327,989 KB.

Tweets_Contents.txt - Notepad

File Edit Format View Help

```
Thu May 29 00:00:43 +0000 2014|471803285746495489|There is no wealth but life. ~John Ruskin #wisdomink|<a href="http://www.hootsuite.com" rel="nofollow">HootSuite</a>|0|wisdom Ink|wisdom Ink
Thu May 29 00:00:43 +0000 2014|471803285738106880|MUCHO la Plop esto, la Plop aquello, pero de los viernes es la fiesta con la gente m\cx3\xais linda. \nEn las otras vienen directo de la frontera.|web|0|Varer Sofier|firekites
Thu May 29 00:00:43 +0000 2014|471803285767462913| motive. When a political idea finds its way into such heads,<a href="http://eto-secret4.ru" rel="nofollow">eto prosto NEW secret</a>|0|Vera Luciani|rosaxonahag
Thu May 29 00:00:43 +0000 2014|471803285750681600|@im_2realbih boll|<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>|290322075|im_2realbih|471800733541486600|0| e s |_SameOleSHIT
Thu May 29 00:00:43 +0000 2014|471803285759078401|A veces no entendemos por que, cuando, donde , como y ahora que hago, por que . Dios es perfecto y a veces... http://t.co/iCyBxph859|<a href="http://www.facebook.com/twitter" rel="nofollow">Facebook</a>|0|rocio murillo lopez|gomitatica
Thu May 29 00:00:43 +0000 2014|4718032857242317568|jajajajaja hay no el vr todo por queres ver si se podia grabar lpm :#|web|0|Fuiste|peroperdiste.|NatyEmeri
Thu May 29 00:00:43 +0000 2014|471803285763280897|@LariinAndofT9 kkkkkkkkkkkkkk ai gente to aqui um beb\cx3\xaa com uma rec\cx3\x9m-nascida e meu emocional?|web|1854700621|LariinAndofT9|471802996926713860|0|Mayara|foryoumartinez
Thu May 29 00:00:43 +0000 2014|471803285742288896|RT @beyles66: cant wait for next year with @Harrison_BHS and @ThePurbalite|<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>|0|Harrison_BHS|Harrison_BHS
Thu May 29 00:00:43 +0000 2014|471803285738094592|\xe2\x80\x9c@_Mf_Gandee26: This picture \xf0\x9f\x98\x8d\xf0\x9f\x92\x8b\xf0\x9f\x91\xaf\xf0\x9f\x8d\xbb\xf0\x9f\x91\x8c\xf0\x9f\x98\x9a @PleasurePink_ @Damn_CeBadd http://t.co/nmVo59x1di\xe2\x80\x9d\xf0\x9f\x98\x8d\xf0\x9f\x98\x8d\xf0\x9f\x98\x98|<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>|1083804685|_Mf_Gandee26|471803096180752400|0|baddgirl|x2\xad\x90\xef\x8b\x8f|Damn_CeBadd
Thu May 29 00:00:43 +0000 2014|471803285767462912|El s\cx3\x1bado es la reu y no voy..Segunda reuni\cx3\x3n que falto desde que entre a JBR0,mami forra.|<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>|0|\xe2\x9c\x9d Stay and Believe \xe2\x9c\x9d|yami_belieber|0
Thu May 29 00:00:43 +0000 2014|471803285771649024|@Emabeltran8 Uuuuuu gatas?|<a href="http://store.ov1.com/content/256340" rel="nofollow">Twitter for Nokia S40</a>|2241549092|Emabeltran8|471802279411351550|0|Mau:3|MauriRamirez3
Thu May 29 00:00:43 +0000 2014|471803285771657216|En tu cara.|<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>|0| Jazz \xe2\x99\x1a|andrea_jazzmini
Thu May 29 00:00:43 +0000 2014|471803285737717760|@null 79|<a href="http://twittbot.net/" rel="nofollow">twittbot.net</a>|3562471|0|\xe3\x82\xaf\xe3\x83\xab\xe3\x83\xbc\xe3\x83\xaf|syuusai_klug
Thu May 29 00:00:43 +0000 2014|471803285741895680|\xe3\x80\x8c\xe5\xae\x9a\xe6\x9c\x9f\xe3\x80\x8d\xe3\x83\x95\xe3\x82\x9a\xe3\x83\xad\xe3\x83\xbc\xe3\x83\xaf\xe3\x83\xbc\xe3\x81\x95\xe3\x82\x93\xe3\x82\x93\xe5\x8b\x9f\xe9\x9b\x86\xe4\xb8\xad\xe3\x80\x82\xe3\x83\x8d\xe3\x80\x80\xe7\xa5\x9e\xe5\xa8\xe3|<a href="http://twittbot.net/" rel="nofollow">twittbot.net</a>|0|\xe7\xa5\x9e\xe5\xa8\xe3\x81\xe3\x83\x90\xe3\x83\xb3\xe3\x83\x80\xe3\x82\xad\xe3\x83\x8a\xe3\x83\x80\xe3\x82\xb3\xe3\x82\xb2\xe3\x83\xbc\xe3\x83\x80\xe3\x82\xb9\xe5\x85\xac\xe5\xbc\xe3\x8f\xe5\x85\xac\xe8\xaa\xe3\x8d|kamui_jj
Thu May 29 00:00:43 +0000 2014|471803285746106368|\xe9\xa2\xa8\xe5\x91\xe2\xe3\x81\xab\xe3\x82\xad\xe3\x83\xa5\xe3\x83\x94\xe3\x83\x88\xe3\x82\xa5\xe3\x83\x94\xe5\x85\xa5\xe3\x82\xad\xe3\x83\xa5\xe3\x83\x94\xe3\x80\x82\xe3\x83\x8d\xe3\x83\xa5\xe3\x83\x94\xe3\x83\x88\xe3\x82\xa5\xe3\x83\x94\xe5\x85\xa5\xe3\x82\xad\xe3\x83\xa5\xe3\x83\x94\xe3\x80\x82|<a href="http://twitter.com/download/android" rel="nofollow">Twitter for Android</a>|0|\xe3\x81\xb9\xe3\x81\xa3\xe3\x81\xa3|5/31|x88\xa5\xbf\xe5\xa8\xe5\xae\xe3\x8b\xe5\x88\x9d\xe5\x80\x8b\xe5\x88\xa52\xe9\x83\x8a1\xe5\x9b\x9e\xe3\x80\x82|Bettit21Tana
Thu May 29 00:00:43 +0000 2014|471803285762883585|\xe3\x80\x90\xe5\xae\x9a\xe6\x99\x82\xe3\x80\x91\xe3\x83\x9e\xe3\x83\xab\xe3\x82\xad\xe3\x83\xa5\xe3\x82\xa6\xe3\x83\x9e\xe3\x83\xab\xe3\x83\x9e\xe3\x83\xab\n\n\xe5\xad\xa6\xe6\xa0\xa1\xe3\x81\xae\xe6\x97\xa5\xe3\x81\xaf\n\xe3\x81\x93\xe3\x82\x8c\xe3\x81\x8b\xe3\x82\x89\xe5\xa7\x8b\xe3\x81\x8b\xe3\x82\x8b\xe6\x8e\xe3\x85\xad\xe3\x81\xab\xe8\x90\xe3\x81\x88\xe3\x81\xbe\xe3\x81\x99\xe3\x81\xad\xe2\x80\xa6|<a href="http://twittbot.net/" rel="nofollow">twittbot.net</a>|0|\xe3\x81\xaa\xe3\x81\xaa\xe3\x81\xaa\xe3\x81\x86\xe3\x81\xbf|ubutora
Thu May 29 00:00:43 +0000 2014|47180328573519360|Happy world burger day everyone!|<a href="http://twitter.com/download/iphone" rel="nofollow">Twitter for iPhone</a>|0|Dyl|Big_Dyl97
```

c) The user name with the most tweets is Alma Arafat (had 61 tweets).

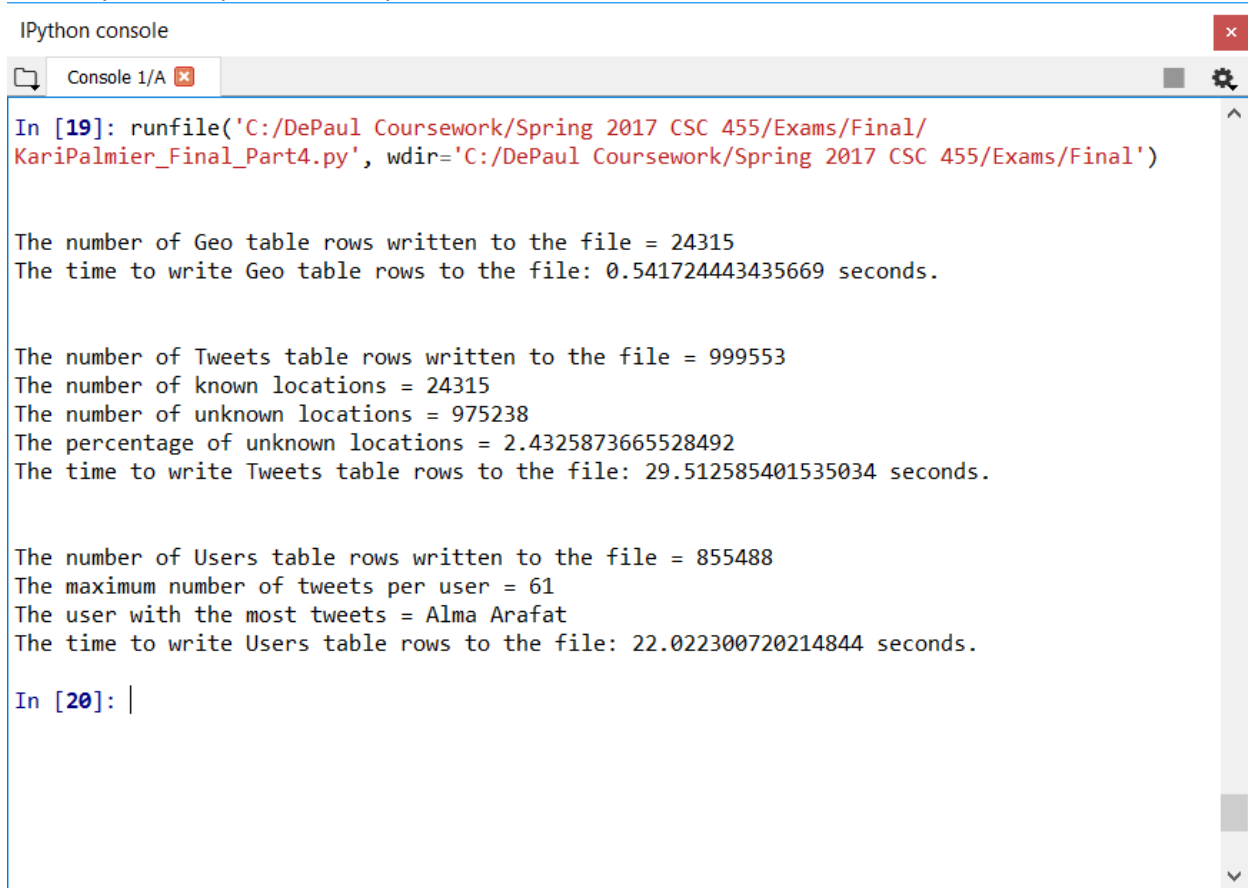
The time it took to create the file with User contents and number of tweets per user: 22.02 secs. The number of User table rows written: 855,488. The Tweet contents text file size is 174,422 KB.

Users_Contents.txt - Notepad

File Edit Format View Help

```
850|Eugene Ventimiglia|eventi|Cool Dad, Ingestor of Social Data|927|1
1503|Brij Singh|brij|Runner, Tinkerer and @techmeme fan. Future is going to be awesome!|807|1
1541|Adam Hertz|AdamHertz|VP of Engineering, Comcast Silicon Valley|671|1
2565|Nabeel Hyatt|nabeel|Entrepreneur, Investor, Hardware & Software Geek @sparkcapital. Make a dent.|454|1
3065|Flxc3\x1avia Del Rio|flaviadurante|Jornalista, DJ e music freak. Paulistana criada em #SANTOSmelhoremtudo|3988|1
3271|deeje|deeje|Maker of elegant mobile-cloud user experiences. Recent works include tappr.tv, Blab, Hello Vino, Thirst, and Biophilia. #ios #boarder #wine #climber #cod|1284|1
3300|Jonathan Might|schwa|No.|532|2
4233|Sean Oliver|Sean Oliver|Sean Oliver is a consultant in Seattle, WA|1231|1
7819|Cody Landefeld|codyl|creative problem solver. Adopted by Christ, Husband to @raquelandefeld. Father to 3. Director at @modeeffect #WordPress #UX|1063|1
10399|John Infante|john_infante|Occasionally critical, often supportive, and never dumbed down|524|1
11036|New Zealand|newzealand|" .we're always in other places, lost, like sheep." -- Janet Frame|1630|1
11957|earnest sewn|earnestsewn|Born in New York. We live for what we create. We desire to do great work and produce things that are true and will stand. We are earnest sewn.|644|1
12350|Ian Kennedy|iankennedy|product guy at http://gigaom.com|420|1
12514|Tom Coates|tomcoates|The personal Twitter account of Tom Coates, co-founder of Product Club: a new product development and invention company. Prev: Brickhouse, Fire Eagle, BBC|816|1
12574|DD|devildoll|Fueled by vegetables.|206|1
12720|Wil Alambre|wilalambre|I'm a web developer, comicbook reader, movie lover, music listener, amateur fiction writer, casual videogamer, avid roleplayer, and all around okay kinda guy.|313|1
12727|AmandaHi|amandahi|runner, baker, lover of data, hater of deer, hopelessly Type A, program manager @washingtonpost|620|1
13374|Charles Edward Frith|charlesfrith|What cannot be said, above all must not be silenced, but written. \xe2\x80\x94 Jacques Derrida\r\nBitcoin
107e2QXc1CrBSNDT17627EY3xwvM98d4Mk|9953|1
13717|Natalie Luhrs|eilatan|Supervillain book reviewer, spreadsheet wrangler, knitter, spinner, geek. Acquisitions Editor for Masque Books.|523|3
16263|Matthew Oliphant|matto|Works at @genworks. Runs @RefreshPDX. Believes Android &gt; iOS. Talked in 3rd-person before it was cool. Was just informed it isn't cool.|286|1
19783|Mark Allen|moustache|Hand-crafted tweets since 2006.|294|1
21633|Claire Armstrong|armst|UI \xe2\x80\x92 UX \xe2\x80\x92 HTML \xe2\x80\x92 CSS \xe2\x80\x92 LA \xe2\x80\x92 RPGs \xe2\x80\x92 \m/|430|1
22203|Josh Gee|jgee|earnestness, anxiety, something|2531|1
35383|David Chartier|chartier|Content Strategist, Writer, Tech Distiller. I run http://FinerTech.com, contribute to @Macworld @Wirecutter @MacObserver. Herald for http://1Password.com.|759|2
36823|Anil Dash|anildash|Cofounder @thinkup & @activateinc \xe2\x80\x92 Blog at http://dashes.com \xe2\x80\x92 anil@dashes.com \xe2\x80\x92 646 833-8659 \xe2\x80\x92 The intern who maintains this account has been fired.|2313|2
37613|Matthew Caldecutt|mcaldecutt|Live in Harlem. Specialize in B2B (ad/marketing) and tech PR. Work @blastpr. Opinions? My own.|3634|1
38323|FoxyStardust|FoxyStardust|I saw a rumor on the Internet that I'm dead.|152|1
39563|alovedlife|alovedlife|A free spirit, with a wild heart.|531|1
46963|Krista Summitt|Mista_Krista|Content marketing practitioner.Brain Currency creator. licensed preacher..DJ.Christ-follower, runner, Chicago Bears and Chicago Bulls fan. Tweets are mine.|1467|1
48023|Jorge Gobbi|morrissey|Travel blogger, docente Cs Soc UBA #fsoc y UNTREF; consultor en turismo y tecnolog\xc3\xada. En Twitter desde 7/12/2006, antes que los evangelizadores|525|1
50193|Hugh MacLeod|gapingvoid|Cartoonist. Corporate culture & #futureofwork. Clients: Rackspace, Roche, Zappos, Cisco, VM Ware, Microsoft etc.
```

Part 4 Python Output Screen Captures:



```
In [19]: runfile('C:/DePaul Coursework/Spring 2017 CSC 455/Exams/Final/
KariPalmier_Final_Part4.py', wdir='C:/DePaul Coursework/Spring 2017 CSC 455/Exams/Final')

The number of Geo table rows written to the file = 24315
The time to write Geo table rows to the file: 0.541724443435669 seconds.

The number of Tweets table rows written to the file = 999553
The number of known locations = 24315
The number of unknown locations = 975238
The percentage of unknown locations = 2.4325873665528492
The time to write Tweets table rows to the file: 29.512585401535034 seconds.

The number of Users table rows written to the file = 855488
The maximum number of tweets per user = 61
The user with the most tweets = Alma Arafat
The time to write Users table rows to the file: 22.022300720214844 seconds.

In [20]: |
```