PROJECTA REPORT

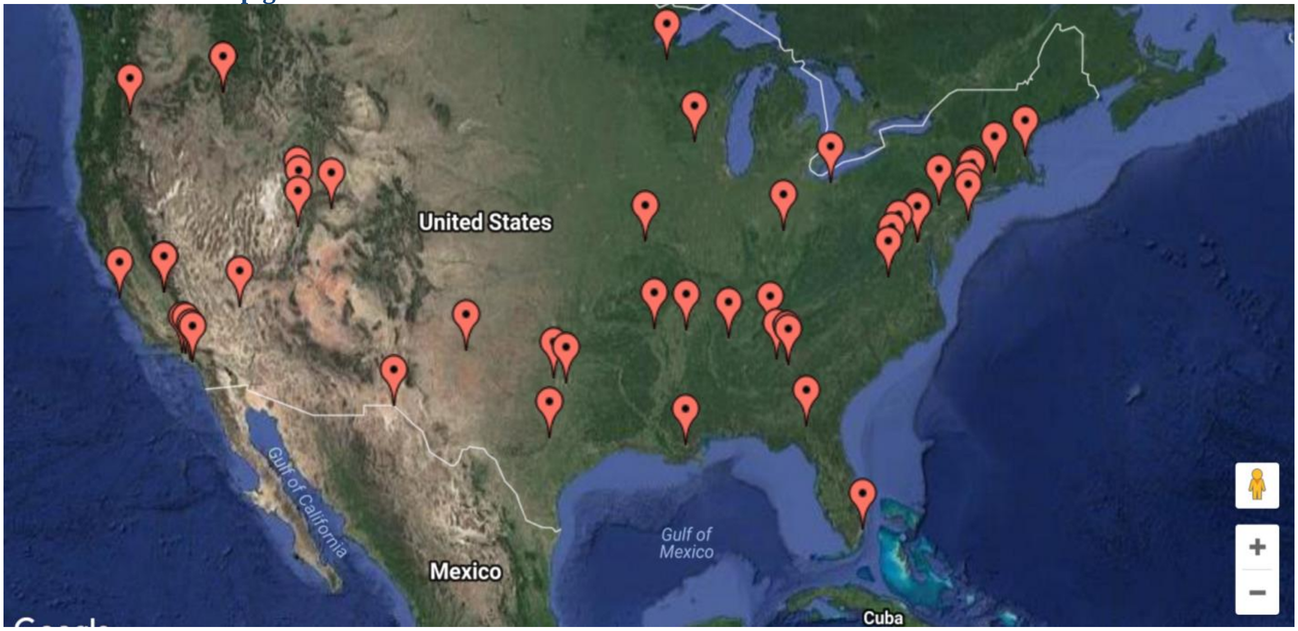
Management, Access, and Use of Big and Complex Data (I 535)

Submitted by: Diksha Yadav (yadavd@umail.iu.edu)

QUESTION 1 & 2:

- Exported portion of my MongoDB dataset in tab separated value form and,
- html file that underlies the Google map picture of my selected region is uploaded with names, “out.tsv” and “map2.html”

Screenshot of the map generated:



QUESTION 3:

a) The sources of help that I consulted while working on this project were:

1. The project material uploaded on canvas.
2. Websites:
   https://docs.mongodb.com
   https://www.tutorialspoint.com/mongodb/mongodb_create_collection.htm
   https://www.youtube.com/watch?v=5Fwd2ZB86gg
3. Help from AIs, Dimitar Nikolov and Yu Luo.

b)
i)

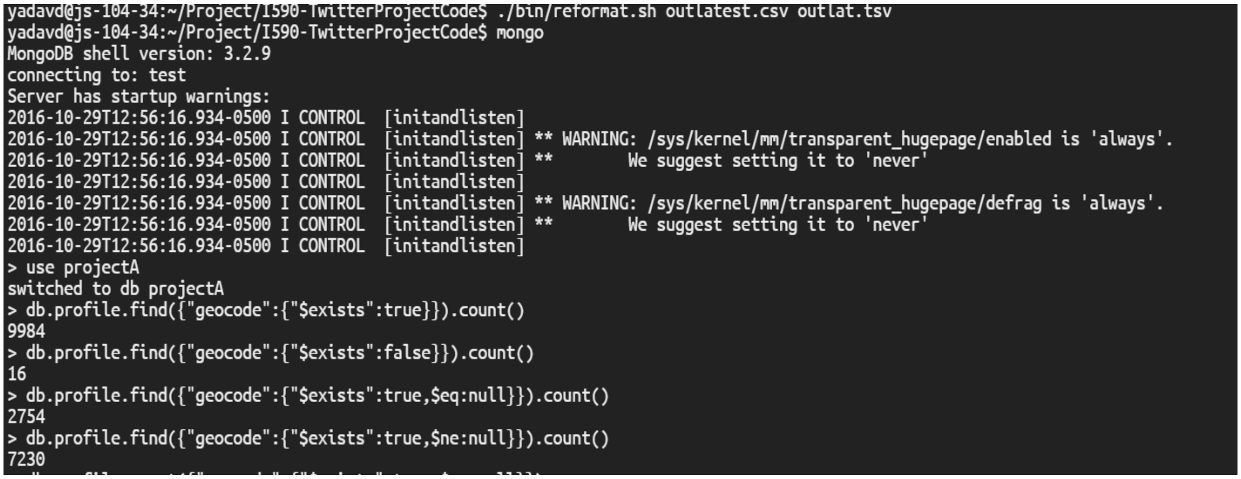
- Number of records for which geocode exists: 9984

- Number of records for which geocode exists but is null: 2754

- Number of records for which geocode exists and is not null: 7230

- Number of records for which geocode does not exist: 16

(Screenshot of queries to get the numbers mentioned above):



```
yadavd@js-104-34:~/Project/I590-TwitterProjectCode$ ./bin/reformat.sh outlatest.csv outlat.tsv
yadavd@js-104-34:~/Project/I590-TwitterProjectCode$ mongo
MongoDB shell version: 3.2.9
connecting to: test
Server has startup warnings:
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten]
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/enabled is 'always'.
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten]
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten] ** WARNING: /sys/kernel/mm/transparent_hugepage/defrag is 'always'.
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten] **        We suggest setting it to 'never'
2016-10-29T12:56:16.934-0500 I CONTROL  [initandlisten]
> use projectA
switched to db projectA
> db.profile.find({"geocode":{"$exists":true}}).count()
9984
> db.profile.find({"geocode":{"$exists":false}}).count()
16
> db.profile.find({"geocode":{"$exists":true,$eq:null}}).count()
2754
> db.profile.find({"geocode":{"$exists":true,$ne:null}}).count()
7230
```

UNRESOLVED RECORDS:

Records for which geocode exists but is null: 2754 and, records for which geocode does not exist:16 are the unresolved records.

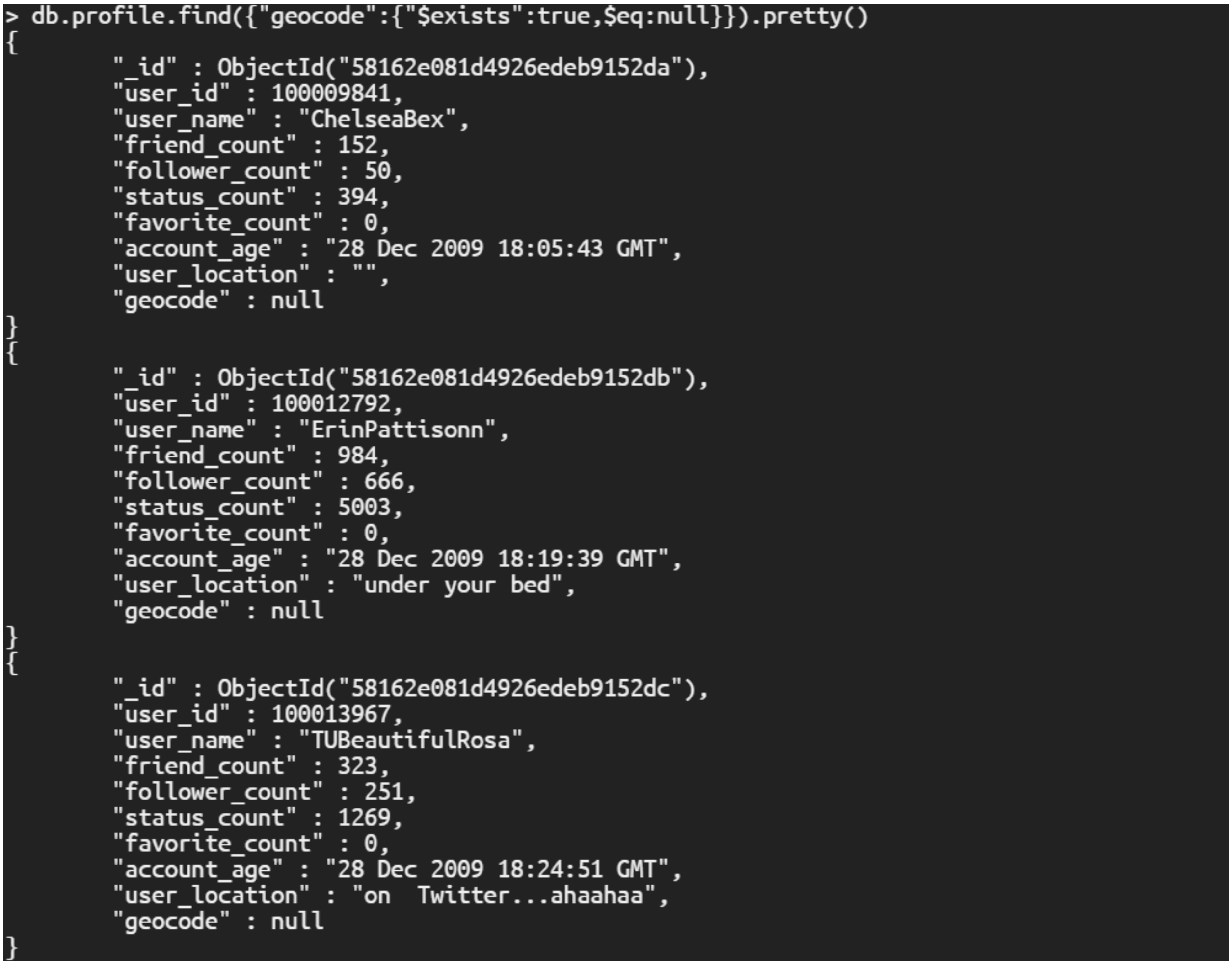
- According to me, the unresolved records could be resolved if the address field in the profile page of user is made of drop down lists instead of text input.

- For example, separate drop downs for address, city, state and country. If done this way, most of the unresolved records can be resolved where users have have updated random text in address instead of actual address like “Cheezy land.”, “my little bubble”, “Dickslanger” etc.

- Also, the address field must be made mandatory. As we have seen that some users have left the address field empty because of which geocode was not updated. If we make the address field mandatory, no such issue will arise.

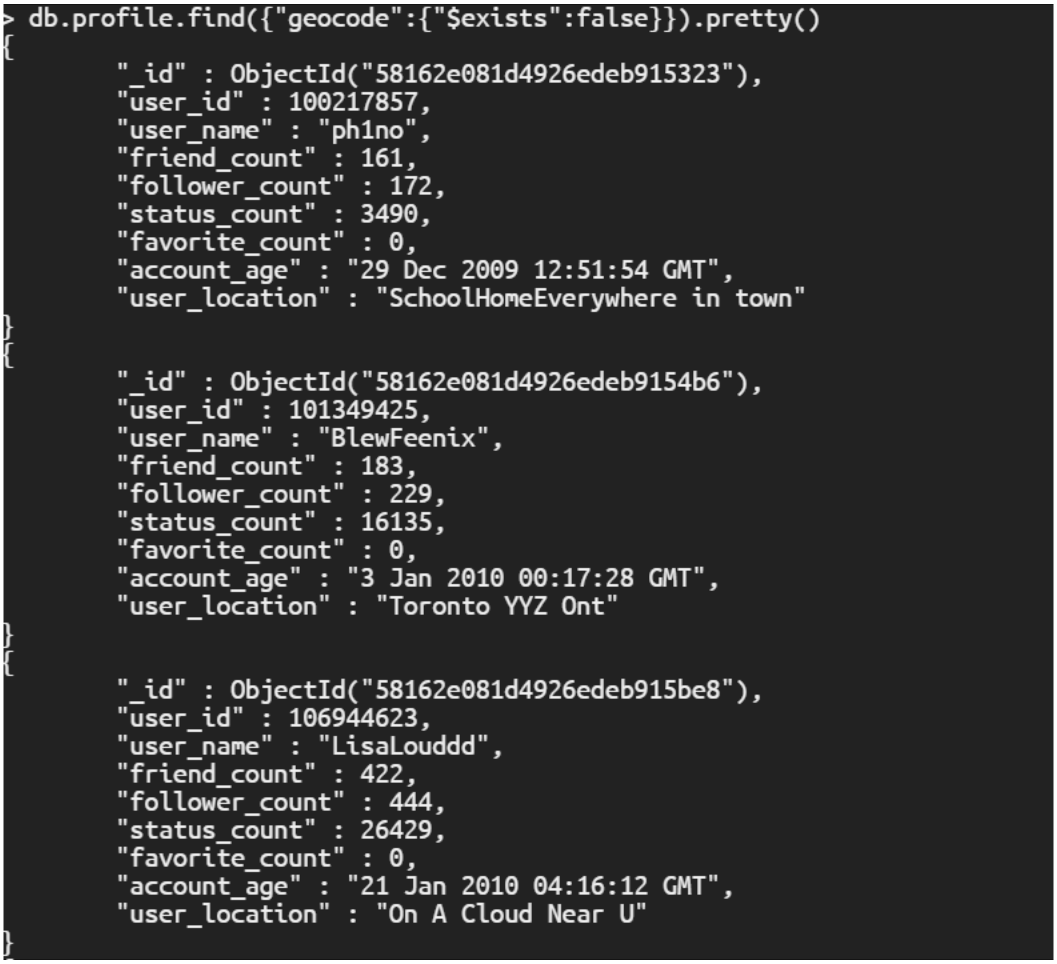
(Screenshot of some of the records that could not be resolved are attached below which shows the irrelevant address text entered by the users):

- Records for which geocode exists but is null.



```
> db.profile.find({"geocode":{"$exists":true,$eq:null}}).pretty()
{
        "_id" : ObjectId("58162e081d4926edeb9152da"),
        "user_id" : 100009841,
        "user_name" : "ChelseaBex",
        "friend_count" : 152,
        "follower_count" : 50,
        "status_count" : 394,
        "favorite_count" : 0,
        "account_age" : "28 Dec 2009 18:05:43 GMT",
        "user_location" : "",
        "geocode" : null
}
{
        "_id" : ObjectId("58162e081d4926edeb9152db"),
        "user_id" : 100012792,
        "user_name" : "ErinPattisonn",
        "friend_count" : 984,
        "follower_count" : 666,
        "status_count" : 5003,
        "favorite_count" : 0,
        "account_age" : "28 Dec 2009 18:19:39 GMT",
        "user_location" : "under your bed",
        "geocode" : null
}
{
        "_id" : ObjectId("58162e081d4926edeb9152dc"),
        "user_id" : 100013967,
        "user_name" : "TUBeautifulRosa",
        "friend_count" : 323,
        "follower_count" : 251,
        "status_count" : 1269,
        "favorite_count" : 0,
        "account_age" : "28 Dec 2009 18:24:51 GMT",
        "user_location" : "on  Twitter...ahaahaa",
        "geocode" : null
}
```

- Records for which geocode does not exist:



```
> db.profile.find({"geocode":{"$exists":false}}).pretty()
{
        "_id" : ObjectId("58162e081d4926edeb915323"),
        "user_id" : 100217857,
        "user_name" : "ph1no",
        "friend_count" : 161,
        "follower_count" : 172,
        "status_count" : 3490,
        "favorite_count" : 0,
        "account_age" : "29 Dec 2009 12:51:54 GMT",
        "user_location" : "SchoolHomeEverywhere in town"
}
{
        "_id" : ObjectId("58162e081d4926edeb9154b6"),
        "user_id" : 101349425,
        "user_name" : "BlewFeenix",
        "friend_count" : 183,
        "follower_count" : 229,
        "status_count" : 16135,
        "favorite_count" : 0,
        "account_age" : "3 Jan 2010 00:17:28 GMT",
        "user_location" : "Toronto YYZ Ont"
}
{
        "_id" : ObjectId("58162e081d4926edeb915be8"),
        "user_id" : 106944623,
        "user_name" : "LisaLouddd",
        "friend_count" : 422,
        "follower_count" : 444,
        "status_count" : 26429,
        "favorite_count" : 0,
        "account_age" : "21 Jan 2010 04:16:12 GMT",
        "user_location" : "On A Cloud Near U"
}
```

ii) This pipeline can be improved if:

- We automate the execution of the commands using shell scripting. Since, the commands were generic, we just need to insert the database and collection name in the script and all the commands can be executed by themselves.

- We check for the valid records before updating them. Since we have records where address field has useless input for mapping, we can remove such records beforehand. This will be convenient because we have maximum limit of records to be updated each day. We can save our time and effort by avoiding useless records and updating those which actually can be plotted.

- We can use job scheduling applications to carry out our QueryandUpdate command after every 24 hours till all the records are updated. The list of all such applications is present at https://en.wikipedia.org/wiki/List_of_job_scheduler_software