**Princeton Data Cleaning**

1. **Purpose:** Clean the records of which every attribute value is “null”.

**How:** Hand delete the records in a json file. (8079 records out of 21730 records should be deleted).

**Outfile:** princeton\_clean1\_deletenull.json”. Test the length of list in “princeton\_clean1\_deletenull.py”. The length is “13651”, which means 8079 records have been deleted.

1. **Observation:** Run the json file “princeton\_clean1\_deletenull.json” through json parser and find there are only 7 GeoCode values in this file. There are 8 GeoCode values in the raw file. It is because the records of which the GeoCode value “null” have been all deleted.
2. **Purpose:** Delete those records of which the geocode value is “Place depicted” (7462 records deleted) , “(not assigned)” (17 records deleted),“geography attribution” (9 records deleted).

**How:** princeton\_clean2\_removegeocode.py

**Outfile:** princeton\_clean2\_removegeocode.json. The length of the list is 6163 (tested in the code princeton\_clean2\_removegeocode.py), which means all the result have been written out.

1. **Purpose:** Delete the duplicate records.

**How:** princeton\_clean3\_removeduplicates.py

**Outfile:** princeton\_clean3\_removeduplicates.json. The length of the data is 5373.

1. **Observation:** Run the file “princeton\_clean2\_removegeocode.json” through the json parser and there are 5373 unique geoID (the identifier of the record in the dataset) in this file, which means all the duplicates have been removed.
2. **Purpose:** Fixing typo:
3. replace the continent value “Europe?” as “Europe”. (2 records have been replaced)
4. replace the continent value “Ellis Island” as “North America”. (1 record have been replaced)
5. replace the continent value “Central America” as “North America”. (6 records have been replaced)
6. replace the continent value “Providence” as “North America”. (1 record have been replaced)
7. replace the continent value “None” as “North America”. (40 records have been replaced)
8. Change the Continent values of the two records of which the city values are “Stoke-Upon-Trent” and “Burslem”. The reason is explained in the “princeton\_clean4\_fixtypo.py”

line 30-33.

**How:** princeton\_clean4\_fixtypo.py

**Outfile:** The result us written is a file “princeton\_clean4\_typofixed.json”.

##The Continent data is cleaned!

## Check the country value

7. **Observation:** Regarding the country values in the records, there are 24 records of which country value is “null”.

8. **Purpose:** Search the country name, city name, state name according to the given latitude, longitude number and geoname values.

**How:** Hand search the country name, city name, state name according to the given latitude, longitude number and geoname values on website: <https://www.latlong.net/Show-Latitude-Longitude.html>.

The following are the records I searched. The yellow highlighed lines are the results I entered to the file princeton\_clean4\_typofixed.json”.

（Geoid, Latitude, Longitude, GeoNames）

40839 63 12 [www.geonames.org/2614165/scandinavia.html](http://www.geonames.org/2614165/scandinavia.html)

Location: Norway (Country), Trøndelag (State), Tydal(City)

40836 48.69096 9.14062 [www.geonames.org/6255148/europe.html](http://www.geonames.org/6255148/europe.html)

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

40837 48.69096 9.14062 [www.geonames.org/6255148/europe.html](http://www.geonames.org/6255148/europe.html)

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

40838 48.69096 9.14062 [www.geonames.org/6255148/europe.html](http://www.geonames.org/6255148/europe.html)

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

41349 48.69096 9.14062 <http://www.geonames.org/6255148/europe.html>

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

41350 48.69096 9.14062 <http://www.geonames.org/6255148/europe.html>

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

41351 48.69096 9.14062 <http://www.geonames.org/6255148/europe.html>

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

17860 NULL None None

Location：England (Country), state and city information exist

17887 NULL None None

Location: England (Country), state and city information exist

775 None None None

Location: missing (Country), missing (state), missing (city)

709 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

710 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

8345 25.32417 -99.66797 <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

728 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

195 None None None

Location: missing (Country), missing (state), missing (city)

7 None None None

Location: missing (Country), missing (state), missing (city)

13 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

1024 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

16678 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

41355 48.69096 9.14062 <http://www.geonames.org/6255148/europe.html>

Location: German (Country), Baden-Württemberg (state), Leinfelden-Echterdingen (city)

21 25.32417 -99.66797 <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

946 None None <http://www.geonames.org/7729892/central-america.html>

Mexico (country), Nuevo Leon (state) General Terán (city).

35108 40.3476掳N -74.6576掳W None

Location: United States (country), New Jersey (States), Princeton (city)

39762 40.3496 -74.6513 None

Location: United States (country), New Jersey (States), Princeton (city)

9. Purpose: Replace the country value from “Mexico or Guatemala” to “Mexico”. The country name can be inferred from the given longitude and latitude number from the records. Add the state and city value also.

10. Purpose: Replace the country value from “United State” to “United States”.

11. Replace the country value from “North America” to “United States”. (2 records is changed)

12. Replace the country value from 'probably Guatemala' to “Guatemala” and add state, city information according to the given longitude and latitude number.

The given location is 'Latitude': '15.5', 'Longitude': '-90.25, it is located in San Pedro Carchá (city), Alta Verapaz (state), Guatemalan (country)

13. For the 6 records of which the country value is 'Guatemala or Mexico', modify the country to be either “Mexico” or “Guatemala” according to the given longitude and latitude information. If the state and city information can be inferred, modify them too.

1. ('Latitude': '19.16667', 'Longitude': '-99.66797') and ObjGeographyID': 675

The location is Mexico (country), Zacango(state), Calimaya (city)

1. ('Latitude': '25.32417', 'Longitude': '-99.66797') and 'ObjGeographyID': 699

The location is Mexico (country), Nuevo Leon (state) General Terán (city)

1. 'Latitude': '16.83333', 'Longitude': '-90' and ObjGeographyID': 10

The location is Guatemala (country), El Petén (state) San Benito (city)

14. Replace the country value from 'Belize, Guatemala or Mexico' to “Mexico” and add the city, state information according to the given longitude/latitude number. 4 records have been changed.

15. Replace the country value from 'Northern Belize or Quintana Roo, Mexico' to “Mexico” and add the city, state information according to the given longitude/latitude number. 1 record is changed.

16. Replace the country value from 'possibly Guatemala' to “Guatemala” and add the city, state information according to the given longitude/latitude number. 1 record is changed.

17. Replace the country value from 'Belize, Guatemala, or Mexico' to “Mexico” and add the city, state information according to the given longitude/latitude number.7 records have been changed.

18. Replace the country value from 'Mexico, Guatemala, or Belize' to “Mexico” and add the city, state information according to the given longitude/latitude number. 1 record is changed.

19. How: princeton\_clean5\_countryvaluesfix.py

20. Outfile: princeton\_clean5\_countryvaluesfix.json”

# The Country values are clean now!

# Now let’s clean the state data

21. **Observation:** The to-be-cleaned state values are as followed:

1. 'probably Chiapas': 1,
2. 'possibly Maine': 1,
3. 'probably Oaxaca': 1,
4. 'possibly Veracruz': 5,
5. 'Guanajuato or Michoacan': 8,
6. 'possibly Guerrero': 1,
7. 'Veracruz?': 1,
8. 'Oaxaca?': 1,
9. 'Colima or Jalisco': 1,
10. 'reportedly found in southern Oaxaca': 1,
11. 'Chiapas or Tabasco': 2,
12. 'Veracruz ?': 1,
13. 'said to be from Oaxaca': 1,
14. 'Pet茅n or Campeche': 1,
15. 'Veracruz or Tabasco': 1,
16. 'possibly Pet茅n': 1,
17. 'Esquintla?': 1,
18. 'Arkansas (?)': 2,
19. 'Possibly Connecticut': 1, '
20. Possibly Ohio': 1,
21. 'Jalisco/Nayarit border': 1,
22. 'possibly from Guerrero': 1,
23. 'Southern Guanajuato or Northern Michoac谩n': 1,
24. 'Masaya Department': 1,
25. 'South Carolina': 1}
26. 'null': 2323,

Since there is too many state values should be cleaned, thus, it would takes too long time to fix it one by one. I decide to use a reversegeocode package, in which I need to input longitude and latitude, and the output would be “state”, “city”, “country”.

22. **Purpose:** Split the princeton\_clean5\_countryvaluesfix.json.

There are some records with neither longitude nor latitude. I split the file “princeton\_clean5\_countryvaluesfix.json” into two subfiles.

**How:** Hand split the file to two subfiles, one includes those records with no either longitude or latitude. One includes those records with both longitude and latitude.

**Outfile:** “princeton\_clean7\_withlongitudelatitude.json” and “princeton\_clean7\_nonelist.json”.

23. **Purpose**: Get the country, state, city information values using reversegeocode package.

**How:** princeton\_clean8\_fromlongilatinumbertostatecity\_finalversion.py

**Outfile:** “princeton\_clean9\_transferlocation.csv”.

24. **Purpose:** Add state, city values to records in “princeton\_clean7\_nonelist.json”,

**How:** princeton\_clean11\_nonlist.py

I change the records of which state value is null to “missing”. I change the records of which city value is null to “missing”.

**Outfile**: princeton\_clean11\_nonlist.json.

25. **Purpose**: Combine two files: princeton\_clean9\_transferlocation.csv and princeton\_clean9\_withlongitudelatitude

**How:** Hand combine.

**Outfile**: princeton\_clean10\_withlongitudelatitude\_final.csv.

25. **Porpose**: Final Combination:

**How**: Hand combine

Continent: continent value from “princeton\_clean10\_withlongitudelatitude\_final.csv” + Continent value in “princeton\_clean11\_nonlist.json”.

Country:

Country value from “princeton\_clean10\_withlongitudelatitude\_final.json” + Country value from “princeton\_clean11\_nonlist.json”

State:

State value from “princeton\_clean11\_nonlist.json” + admin1 value from “princeton\_clean10\_withlongitudelatitude\_final.json”

City:

City value from “princeton\_clean11\_nonlist.json” + name value from “princeton\_clean10\_withlongitudelatitude\_final.json”

**Outfile:** The final file names “princeton\_clean 12\_final version.csv”.

**GM data cleaning process**

1. **Raw data**: The raw file names “gm\_geography\_raw.csv”.
2. **Purpose:** Delete columns: Delete the extra columns and keep “ObjectID”, “term”, “ThesXrefType”,”LatitudeNumber”,”LongitudeNumber” columns.

**How:** Hand delete columns.

**Outfile:** gm\_geography\_clean1\_columndelete.csv

1. **Purpose:** Delete the records of which the ThesXrefType value is either “place”, “0900”, or “culture area”.

**How:** GM\_GeographyTerm\_Clean.py

**Outfile:** GM\_Clean2\_CorrectGeographyTerm.csv

**Test**: In original csv file (gm\_geography\_clean1\_columndelete.csv), using search function, I searched among column “ThesXrefType”. The term “Place found” occurs 7085 times. The term “Place Made” occurs 1004 times. Thus, it is expected that there is 8089 (7085+1004) rows in the outcome csv file. In the outcome csv file (GM\_Clean2\_CorrectGeographyTerm.csv), there are 8089 rows.

1. **Observation:** Using the CSV parser script, I found among the 8080 rows (each row includes one object ID), there are only 3835 unique object ID (documented in file “GM\_Clean2\_CorrectGeographyTerm\_DataProfile.md). I am trying to read each row and seperate the geography terms into different columns “country”, “state”, “city”. However, it really takes too long time to do so. Finally, I give up.
2. **Purpose:** Remove the index line and remove duplicates in csv file (GM\_Clean2\_CorrectGeographyTerm.csv).

**How:** GM\_CSV\_RemoveDuplicates.py

**Outfile:** The results are written to a new csv file named “GM\_Clean3\_CorrectGeographyTerm\_withoutDuplicate.csv”. 8036 rows are left.

1. **Purpose:** Delete rows with repeated object ID. For rows with repeated object ID, only keep the first-occurred row.

**How**: GM\_Clean4\_RemoveDuplicates\_ObjectID.py

**Outfile**: GM\_clean4\_RemoveDuplicate\_ObjectID.csv

1. **Observation:** In file GM\_clean4\_RemoveDuplicate\_ObjectID.csv, there are 225 records of which latitude number and longitude number are both 0. There are 57 records of which latitude number and longitude number are empty (no data is given).
2. **Purpose:** Split the file GM\_clean4\_RemoveDuplicate\_ObjectID.csv into two sub-files. One in which each record has both longitude and latitude number. The other one in which each record has neither longitude nor latitude number.

**How**: Handly split the file

**Outfile**: The file with latitude and longitude numbers names “GM\_clean5\_RemoveDuplicate\_ObjectID\_RemoveEmptyLine.csv” (3553 records) and the other one names “GM\_clean6\_RemoveDuplicate\_NoLocationNumber.csv” (57 records).

1. **Purpose:** Transfer the longitude number and latitude number to country, state, city.

**How:** I tried many two geocoding packages, namely reverse-geocode and reverse-geocoder. I finally select reverse-geocoder package because it can infer country, state, city information while the other one can infer country, city information only. The script names “gm\_clean7\_fromlatitudelongitudetolocation.py”.

**Outfile:** gm\_clean7\_fromlatitudelongitudetolocation.csv

1. **Observation**: In the file “gm\_clean7\_fromlatitudelongitudetolocation.csv”, the country name is described as ISO 3166-1 alpha-2 country code. I need to transfer the code to country name. Also, continent name is missing still. It would be great if I can transfer the country code to continent name.
2. **Purpose:** Transfer the ISO 3166-1 alpha-2 country code to country name and continent name.

**How:** gm\_clean8\_codetocountryandcontinent.py

**Outfile:** (1) gm\_clean8\_continent.csv; (2) gm\_clean8\_country.csv

1. **Purpose:** Combine three files: (1) gm\_clean7\_fromlatitudelongitudetolocation.csv (2) gm\_clean8\_continent.csv; (3) gm\_clean8\_country.csv

**How:** Hand combine three files

**Outfile:** gm\_clean9\_withlongilanti\_final.csv

1. **Purpose**: process the records with neither longitude number nor latitude number

**How**: There are 57 records in file GM\_clean6\_RemoveDuplicate\_NoLocationNumber.csv. The records in this file have no latitude number and longitude number. I search enter the Term value, for example, central Mexico on <https://www.wolframalpha.com/> and google. I will enter the continent, country, state, city names if I can find from these two website.I will enter the names into the file GM\_clean6\_RemoveDuplicate\_NoLocationNumber.csv and save it as GM\_clean10\_RemoveDuplicate\_NoLocationNumber\_withlocationnumber.csv.

For example, if the term if “central Mexico”, the website will show the location is 23°N, 102°W. Then I will enter 23 as latitude number and -102 as longitude number. Some of the terms are too vague, for example, “country”. In this case, I will delete the record.

**Outfile:** GM\_clean10\_RemoveDuplicate\_NoLocationNumber\_withlocationnumber.csv.

1. **Purpose:** Combine file

“GM\_clean10\_RemoveDuplicate\_NoLocationNumber\_withlocationnumber.csv” with file “gm\_clean9\_withlongilanti\_final.csv”.

**How**: Hand combine

**Outfile:** gm\_clean11\_finalversion.csv

**SAAM data cleaning process:**

1. **Purpose**: Remove duplicate rows (16200 rows)

**How**: SAAM\_removeDuplicates.py

**Outfile**: saam\_WebConGeography\_CleanDataRemoveDuplicates.csv (16197 rows)

1. **Purpose**: remove those rows in which ConGeoCode value is “(not assigned)”, “Associated Place”, “Associated Place (at time of NEA award)”, ““Last Known Residence”. Keep the records of which ConGeoCode values are “Place of Birth” and “Place of Death”.

**How**: saam\_clean2\_removegeocode.py

**Outfile**: saam\_clean2\_correctgeographyterm.csv

1. **Observation**: 329 records do not have a city value. 78 records do not have a country value. 2462 records do not have a state value.
2. **Purpose:** Fix the country value and city value.

**How:** I added “missing” to the cells to represent the city name is missing in this record. I added the country name according to the given city name and state name. If no enough information is given to have a country name, then I enter “missing” in the cells. If the state name is missing, I enter “missing” in the cell.

**Outfile**: saam\_clean3\_correctgeographyterm\_addmissingvalue.csv

1. **Purpose:** Add continent value

**How:** Hand add continent values according to the country name.

**Outfile**: saam\_clean4\_addcontinent.csv