Luke’s code for Tribal so far. (1/18/19)

**Summary**

For retailers, we are required to do the following tasks:

1. Confirm whether the retail establishment is in Indian Country;
2. Identify or update the associated federally-recognized tribe;
3. Confirm the name and address of each establishment and specify any corrections that are necessary for the name or address of the identified establishment;
4. Identify the legal owner for each establishment; and
5. Identify retailers that are no longer in operation.
6. Identify any tobacco retail establishments located in Indian Country that are not on the list provided by FDA
   1. Provide the name, address, associated federally-recognized tribe, and legal owner for all newly identified retail establishments

Currently, the code completes everything but item 4 on this list.

The code is broken out into 4 broad steps, as indicated by the prefixes of each .py file:

* Step 0: Do some cleaning with FDA data, and run those retailers without coordinates through Google/Census geocoders.
  + To do item: Indicate in output which coordinates are added as a result of this. This will need to be shown as a tracked change in what we send FDA
* Step 1: Use google to confirm the name, location, and operation status of each retailer (task #3, 5 above)
* Step 2: Confirm tribal status and associated tribe of retailers (tasks #1,2)
* Step 3: Geocode Nielsen POS data, find tribal status/association, compare with FDA list to figure out which ones are not already on it (task #6). There is some ownership information Nielsen included with the POS data, but we’ll need to run these retailers through the task #4 code we develop.

Another item that will come up eventually is to build the code that set up the documentation files for all of these determinations. Let’s get feedback from the sample file we sent in to FDA first before we work on this though.

I haven’t touched manufacturers, but I think that will be a quick adaptation once we finish everything on the retailer side.

**File Descriptions**

**Folders**

*Functions*

* Folder to house functions used multiple times across files.

*functions\cleaning\_functions.py*

* + Functions to standardize data for matching functions

*functions\manual\_functions.py*

* + Functions to separate out the manual parts of the code for easy reconfiguration in the future

*functions\matching\_functions.py*

* + Functions to match retailers by names and/or addresses

*Input*

* Folder to house input files

*input\aiannah shapefiles*

* + Shapefiles of Indian Country

*input\raw\_Nielsen\_POS\_data\_full.csv*

*input\Nielsen\_POS\_data\_full.xlsx*

* + POS data from Nielsen

*input\Public retail data.csv*

*input\Public retail data.xlsx*

* + Selected data elements from original FDA list

*step\_0\_work*

*step\_1\_work*

*step\_2\_work*

*step\_3\_work*

*step\_4\_work*

* Working folders for the corresponding python files.

*step\_5\_work*

* Empty folder

*logs*

*output*

* Some misc folders I used in development, don’t think anything relevant uses these now.

**Python files**

*0a\_fda\_list\_cleaning.py*

* Handling some encoding issues with foreign characters

*0b\_create\_retailer\_batches.py*

* Break up the retailers in batches so it can be run through the Census geocoder

*0c\_census\_geocode\_retailers.py*

* Run retailers through the Census geocoder
* Requires internet access to run

*0d\_google\_geocode\_retailers.py*

* For those retailers not found in the Census geocoder, run them through the Google geocoder
* Requires internet access to run

*0e\_clean\_merge\_matches.py*

* Clean and merge the Google and Census geocoding matches

*1a\_google\_api\_query.py*

* Run each FDA retailer through the Google Places API and attempt to find a matching record.
* Requires internet access

*1b\_process\_establish.py*

* For those retailers with multiple possible candidates, select which candidate is best
* Take the results from Google Places, and verify which results are true matches.

*1c\_process\_nonest.py*

* For those results that we determine are not true matches, we run the FDA retailers through Google Streetview’s API, and attempt to find the address on Google Maps. From there, we capture a panoramic picture of the address.
* I then had Karen Baldwin go through the pictures that were taken, and manually determine whether a commercial establishment was present at this location. We use this later on to determine operation status for these retailers. We also assume those with commercial establishment’s present are correctly identified in FDA’s list.
* Requires internet access

*1c\_FIX\_rerun\_pic\_seg.py*

* Had to rerun some of the pictures due to an error I made. This file does not need to be rerun in future replications.
* Requires internet access

*1d\_post\_manual\_process.py*

* Short file that cleans up some of the manual coding that Karen did.

*1e\_update\_address\_name.py*

* Merge together previous work, and finish making determinations regarding which candidates are true matches.
* Currently we have ~150 retailers with no match from Google. We may need to attempt a manual investigation of Google/other sources to verify their name/address/operation status
* For those with matches, figure out whether to recommend an update to their name/address in the FDA list and record updated values.

*2a\_tribal\_boundary\_compare.py*

* Compare geocoordinates of matched retailers with Tribal Country shapefile, and note which ones are or are not within tribal country.

*3a\_create\_retailer\_batches.py*

*3b\_census\_geocode\_retailers.py*

*3c\_google\_geocode\_retailers.py*

*3d\_clean\_merge\_matches.py*

* Same as 0a-0d, but for Nielsen POS retail list.

*3e\_tribal\_boundary\_compare.py*

* Same as 2a, but for Nielsen POS list

*3f\_FDA\_list\_compare.py*

* Compare Nielsen POS tribal retailers with FDA list to determine which POS retailers are and are not currently on the FDA list

*3g\_merge\_with\_FDA.p*y

* Append the new POS retailers not currently on the list with the existing FDA list.
* The “step\_3\_work\output\full\_retailer\_list.csv” produced by this file is the overall output of all the code written so far.