Final Midpoint check in

Nanzhu Liu

655394878

**Read Me**

1. Keep the “data” folder and .py in a folder. Data should be all in the “data folder”.
2. I created an empty folder named “result” to save the edited file. Now there are 5 csv files exported in it.
3. When you run the code, the results will show you the first 5 records of each data file I worked on.

**Narratives**

When I started my projects, I realized it is more difficult than I thought. I decided to decrease several deliverables.

Here are my deliverables (black is previous deliverables and green is what I am going to change to):

1. Collect and prepare population data by counties from NVSS. Create codes to read sas data file(I found it also provided in .txt file. So I decided to work on .txt file instead of sas file. In this way, I can use the knowledge which I learned from the class), keep the needed columns and rename their headers– Done by midpoint check in

2. Collect and prepare population, population by race, 100% federal poverty level, 200% federal poverty level, 400% federal poverty level and language data (I decided to do population and language instead of doing all of them. Those two are the most difficult.) by counties, townships, elementary school districts, unit school districts and municipalities from US Census ACS.(I will do by the counties) Create codes to read csv data files, keep or delete columns, rename headers, aggregate several columns –Done by midpoint check in

3. Merge all the data files based on their common attribute. Write codes to merge data. (unchanged)

4. Get the rate of each column from ACS data file. Create a function to calculate the rate at one time. (unchanged)

5. Adjusted the data based on the data from NVSS and the rate. Create a function to balance the data. (unchanged)

6. Output the data into a new file. (unchanged)

There are mainly two parts I have done by midpoint check in:

Part I:

I collected and prepared population data by counties from NVSS. The file is .txt file. The whole txt file contains population data across country from age 0 to 85. The txt file contains two columns. The first column tells the data information including series vintage, years, month, states FIPS code, counties FIPS code, ages, race, and Hispanic origin. The second column tells numbers. I needed to select data that satisfy the following requirements: data is Illinois and age is from 0 to 5.

First, I used readlines() to read .txt. Second, I selected IL data using split() and if condition. The delimiter of split() is tab. I used slicing to keep states FIPS code, counties FIPS code, ages from the first columns and deleted other information. I selected age from 0 to 9. There is a space before age if age is from 0 to 9. I used split() to separate the first column into two, and used if condition to select data with 2 lengths. The delimiter of split() is a space. If the age equals to 10 or above, there is no space, so the length is 1. Now, the first column represents “geoid”. The second column represents age. I used join function to joined geoid, age with numbers.

Current Problem:

I haven’t figured out how to select age from 0 to 5. I am thinking to convert the list into a table and use sql to select them.

Part II:

I collected and prepared population data by counties from US Census ACS. It is csv file. I imported csv file in Python by importing pandas. I selected the 4 columns: geo.id, name, male under 5, and female under 5. I aggregated male under 5 and female under 5 to get the total number of children under 5. I combined geo.id, name and the total number of children under 5 together by using concat() function. I renamed the header and exported into a new csv file.

The race data are also from US Census ACS. There are 4 csv files: race by white, race by black, race by American Indian and Alaska native, and race by Asian. The procedure is the same as the previous one. I renamed the header and exported them into new csv files based on the type of race.

Thus, there are total 5 files in the result folder.

Work will be done later on:

The procedure is almost the same. I am wondering if I could create a for loop to optimized my code.

https://drive.google.com/a/illinois.edu/file/d/1VIgsf4ucGtLTjF3100PTSrEzx-xzp9Qv/view?usp=sharing