1. **Folders / Files Explanation**

1. compactCode.do

run compactCode.do, need to install python3,

* In this file, first it calls all Python script to generate files from 1990 to 1998

1\_downloadfile.py (commented out)

2\_gen98\_96.py

3\_gen95.py (commented out)

4\_gen94.py (commented out)

5\_gen93.py

6\_gen92\_91.py

7\_gen90.py

Notice that each Python code requires 3 inputs:

path: the file path points to stored RawDataFiles

savePath: the file path points to saved CleanFiles

csvPath: the files path points to saved CSV files

Notice that I comment details for Python code in gen92\_91,

and others are similar thoughts.

Need to notice that I comment out Python script which call "gen95" and "gen94"

Because these two python codes required third-party package "**antiword**" to extract

DOC files, and this is not originally support in Python3.7/3.8 framework.

For getting more install information could follow website below:

https://docs.bitnami.com/installer/apps/resourcespace/configuration/install-antiword/

* Then it calls a do files which generate all files from 1999 to 2017

8\_cleanData1999\_2017.do

* Calls a do file to update FIPS code, FIPCTY code and County for 1995,1995 and 1990

9\_updateFip95\_94\_90.do

* Calls a do files to expend old\_1959\_1989 and append all data files

10\_appendData

(I did not change much for the code after append all data, because after I go through it carefully, I did not find any mistakes. )

2. code folder stores all Python codes and Stata codes required in compactCode.do

Other files in this folder:

checkMerge.do

(used to comparing original final output and reviewed output)

reviewcode.do

(used to implementing original final output)

removeblank.py

(used to remove blank lines, just applied for running reviewcode.do)

removelatin.py

(used to remove all Latin characters in 1995 and 1994 files)

3. rawFiles folder stores all raw data downloaded online

4. output folder contain two sub-folders:

cleanFile:

all data cleaned through Python

csv:

all data saved as csv form, from 1959\_1989, 1990 to 2017,

and final\_append dataset.

5. log file stored in log folder.

1. **Merge Result**
2. Merge year by year to check each comparison (from 2017 - 1990 - old years 1959)

I did some modifications to the original code (see in reviewcode.do) when reading files from 1998 to 1996. The reason is that I remove all special characters while clean data

with Python, because not all l50\_\*, l80\_\*, fmer, fmr\* have same length.

1998 file is right justified, so I think original code run appropriately.

But 1997, 1996 may not be all right justified, then I changed the way how original code

handle reading in data.

I read a line as one string, and split by space, then generate each variable name.

Lastly, I changed them to numerical type.

A screenshot of a cell phone

Description automatically generated A screenshot of a cell phone

Description automatically generated

Following are comparison results:

* 1997 and 1996 files one missing match: fipst = 49, st = UT, fipcty= 25, Kane County, msa=3739, misname code unknown

But both outputs have same value don't know what cause the issue.

* 1995

Original data:

Fips: 49 fipcty: 25 (empty values)

Riveiw Data:

Fips: 49 fipcty: 25 (has values)

* 1993

Problem with state names,

Original data:

- empty

P

V

Riveiw Data:

- empty = PAC

P = PR

V = VI

1. Create an unique ID for both reviewed output and original output, named *finalid .*

Consisted by:

st + countyID + year + fips + msa + median

Because I used county\_id while crating countyID, I keep county\_id in 1995, 1994 and 1990 for original do codes.

1. Compare two final output files by finalid:

Output saved in dataInOrigin.csv & dataInReview.csv in finalOutput folder.

1. Compare teo final outputs by median & finalid:

Output saved in dataInOriginMedian.csv & dataInReviewMedina.csv

in finalOutput folder.