Synthesize Data

David J Dorer

Synthesize Data

Preliminary Steps

Before we can synthesize data we need the following:

1. A model. See CreatingModels vignette for details - vignette("CreatingModels")

   1a. A unique model name.

   1b. PUMS PUMA variables. A named list of functions that are used to compute derived PUMS variables. See `vignette("CreatingModels")` for details.

   1c. Tract marginal tables. A named list of functions that are used to define the Tract marginal tables.

   1d. Model parameters. A list of model parameters. At a minimum, `"model"`, `"model.type"`, `"nages"`, `"nraces"`, `"geotype"`, `"comment"`. See `vignette("CreatingModels")` for details.

The variables functions and the marginal tables functions need to be defined prior to creating the model. If you put your R code in a ".R" file and the use `source("<your_file_name>")`, you need to put the variable functions and marginal table functions before the code used to define the model. Remember to run `PAT.test.model()` so that you can correct errors before you synthesize data.

2. A geography relationship or cross-walk file. This step is handled automatically by `PAT.synth.data()` and `PAT.synth.blockgroup()` so you don't need to do anything unless you use geographies prior to 2012. See `help(PUMA.Tract.Rel)` for more details.

   2a. `PUMA.2020.Tract.2020.RData` - `data(PUMA.2020.Tract.2020)`.

   2b. `PUMA.2012.Tract.2020.RData` - `data(PUMA.2012.Tract.2020)`.

   2c. These files are loaded automatically when you use `PAT.synth.data()`. The different vintages for the crosswalk files are determined by `PAT.pums.vintage()` and `PAT.vintage()`. We need these files because the PUMA geography needs to be "coordinated" with the Tracts that fall within the PUMA. For example for 2021 vintages, the PUMA geographies are based on the 2010 census and were put in to effect in 2012. But for the 2021 tract marginal tables, the 2020 census geography is used. Crosswalk files for these "mixed" geography situations can be downloaded from:

GEOCORR Website https://mcdc.missouri.edu/applications/geocorr.html

For GEOCORR the "source" geography should be the PUMA and the target geography should be the "Census Tract" for the `PAT.synth.data()` function or "Census Block Block Group" for `PAT.synth.blockgroup()`.

PAT.synth.data arguments

3. Running `PAT.synth.data()`. The following arguments are used:

   3a. `state`: You need to specify the FIPS code for your state. If you have Supplemental Poverty Measure (SPM) variables in your model two things will happen. First, the period for the PUMS/PUMA microdata will be changes to "1" as the SPM microdata files are 1 year files. Second you will need to create a state SPM file. See help(make.spm.state)

3b. `puma`: PUMA FIPS for the "large" area PUMA. Be sure to use quotes. If puma has length >1 then all the PUMAs in puma will be run.

3c. `iter`: Maximum number of iterations for the IPF (Iterative Proportional Fit) algorithm. Default 20.

3d. `vers`: version of IPF to use. 1: internal C routine. 2: Multidimensional Iterative Proportional Fitting (mipfp) package Ipfp() function. Used for cross check with internal C routine.

3e. `maxdev`: Maximum change between iterations of the relative difference between the for the iteration and the target marginals. Default 0.001

3f. `update`: Name of update file (located in `logdir()`). Geographies in the update file will be skipped. Usefull if the geography loop is interrupted the function can be restarted where it left off. `PAT.synth.data()` appends to the checkfile check point file. See checkfile argument below. Usually before a restart you should copy the current checkpoint file to the update file.

3g. `outtag`: a character string to append output file names. By default the log file is:

`synth_<state>_checkpoint<outtag>.txt`

3h. `vintage`: Marginal tables vintage. Default `PAT.vintage()`.

3i. `period`: Marginal tables period. Default `PAT.period()`.

3j. `pums.vintage`: PUMS/PUMA vintage. Default `PAT.pums.vintage()`.

3k. `pums.period`: PUMS/PUMA period (1 or 5). Default `PAT.pums.period()`. Note is SPM variables are used this value will be set to "1".

3l. `county`: Character vector of county FIPS codes. Default `character(0)`, run all counties within puma. If county has length 1, then the tracts within the county will be run. See tract argument next.

3m. `tract`: Character vector of tracts. Default character(). If tract has length 0, then all the tracts in the county or puma will be run depending on whether county is specified.

3n. `logfile`: Name of log file (including folder). Default logfile is `synth_<state>_log<outtag>.txt` and is placed in `logdir()`. Note logfile is overwritten with each run. If you want to save logfiles between runs, rename the logfile before starting a new run.

3o. `checkfile`: Name of checkpoint file. Default `synth_<state>_checkpoint<outtag>.txt`. The checkpoint file is appended to. Thus the checkpoint file grows with each run. If you don't want this, rename the checkpoint file between runs.

3p. `debug`: Level of messages. Default PAT.verbose().

3q. `key`: Census key. Default `PAT.census.key()`.

3r. `model`: Model to run. See vingnette("CreatingModels") for details.

3s. `odir`: Folder/directory for synthetic data files. Default `outdir()`.

Output files.

4. Output files:

4a. Synthesized data. By default the data goes in `outdir()`. File name format:

`synth_data_<state>_<puma>_<county>_<tract>.csv`

4b. Checkpoint file: Default name format: `synth_<state>_checkpoint<outtag>.txt` placed in `logdir()`. Grows with each run.

4c. Log file: Default name format: `synth_<state>_log<outtag>.txt` placed in `logdir()`. Truncated with each run.

8 Dec 2023 12:36