

Creating Models

David J Dorer

Creating Models

Model Components

Models are lists with 4 components:

1. `name`: A name for the model without spaces.
2. `variables`: A named list of functions that can be used to compute derived PUMS variables. By convention the function name ends in “.v” and the list element name does not include the “.v” This convention is used to provide conceptual clarity and to avoid name collisions when defining the variable and marginal table functions.
3. `marginal.tables`: A named list of marginal tables. By convention the marginal table function names end in “.m” and the name of the list element does not have the “.m”
4. `structural.zeros`: An optional component of the model specified by a formula defining structural zeros. These combinations of variables will be zero in the synthetic data. These are combinations of variable that must be zero for logical reasons based on the categories of the value of the variables. The formula has the following format:

Example:

```
structural.zeros=  
paste(c("((Education==Under_18)&(Age==18_to_24_years))"  
, "((Education==Under_18)&(Age==25_to_44_years))"  
, "((Education==Under_18)&(Age==45_to_54_years))"  
, "((Education==Under_18)&(Age==55_to_64_years))"  
, "((Education==Under_18)&(Age==65_years_and_over)))"  
, collapse="|")
```

The formula is an “or” of an “and”. Remember to run `PAT.test.model` to check that the format of the formula is correct.

5. `parameters`: A named list with parameters. Required:
 - 5a. `model`: the model name from the “name” component.
 - 5b. `model.type`: Values “person” or “house.” A “person” `model.type` means that the PUMS variables and model table are based on the PUMS person level data. A “house” `model.type` uses the PUMS house level data.
 - 5c. `geotype`: A value of “tract” or “blockgroup” A “tract” `geotype` means that synthetic data will be produced for census tracts. A “blockgroup” `geotype` means that block group level synthetic data will be produced.
 - 5d. `nraces`: Number of race categories for variables and marginal tables that use race. See `help(PAT.race.code)`.

5e. nages: Number of age categories for variables and marginal tables that use age. See `help(PAT.age.code)`.

5f. comment: A comment to be used for outputs.

5g. ... Optional user defined parameters to be passed to the variable functions and marginal table functions.

Model Component variables

Before the model list can be created the variable functions and marginal table functions must be defined (created) so that they can be used. A simple example illustrates this process:

```
Sex.v<-function(SEX) {  
  sex<-as.numeric(as.character(SEX))  
  factor(sex,levels=c(1,2),labels=c("Male","Female"))  
};
```

A few things to notice.

1. the functions is named “Sex.v” To keep things straight use the convention that the function starts with a capital letter and ends with a “.v”
2. SEX is a PUMS person level variable. Review the PUMS codebook, e.g. PUMS documentation page
3. The return value of the Sex.v function is a factor variable. When PUMS data is downloaded via the API (Application Programming Interface) the resulting R data.frame has character variable. To be sure we use “as.character” in our code.
4. The derived variable in the PUMS data.frame will be “Sex” the name of the model list component “variables.”
5. To keep things organized the factor labels should the following convention:
 - 5a. The factor label starts with a capitalized letter (or a numeral) and then uses lower case.
 - 4b. The factor labels use “_” not “ ” so that there are no blanks in the factor levels. Also don’t use any punctuation other than “_” in the factor levels. This will save a lot of headaches later.
6. Optionally, the variable function can use arguments from the defining model list. For example when you are using AGE as one of your arguments, you might want to add an “nages” argument. You can use any additional parameters that you might like as long as your parameter appears in the model “parameters” list component.
7. When the PUMS data is downloaded the variables that you have defined will be added to the PUMS data.frame with the name of the model “variables” list component. This is why when you construct the model “list” you use the convention that the function used to define the PUMS “Sex” variable is named “Sex.v”.

Model Component marginal.tables

1. Like the variable name/function definition the convention is that the marginal table name correspond to the marginal table function except that the marginal table function ends in “.m”. For example you might like to have a marginal table that corresponds to the model PUMS variable function “Sex.” That marginal table function should be a function named “Sex.m”.
2. An important point using Iterative Proportional Fitting is to adjust a PUMS table/cross tabulation at the PUMA geography to an ACS “B”, “S”, and “DP” table at the tract or block group geography. To do the adjustment the marginal table dimnames variable names need to correspond to model PUMS variable names. Also the marginal table “dimnames” need to have PUMS variable as names and the

dimnames of the marginal table categories (labels component of the factor variable) that agree exactly with the PUMS variables factor labels.

Here is an example of a marginal table function:

```
Sex.m<-function (state = "25", puma = "", county = "021", tract = "400100",
  blockgroup = "", csd = "", place = "", vintage = PAT.vintage(),
  key = PAT.census.key(), period = PAT.period(), download = PAT.download(),
  debug = PAT.verbose())
{
  gat <- PAT.acs.table("B01001", vintage = vintage, period = period,
    state = state, county = county, tract = tract, blockgroup = blockgroup,
    puma = puma, csd = csd, place = place, download = download,
    debug = debug);
  dat <- gat$data;
  rownames(dat) <- toupper(rownames(dat));
  dat <- dat[c("MALE", "FEMALE"), c("Est", "MoE")];
  dimnames(dat) <- list(Sex = c("Male", "Female"), stat = c("Est", "MoE"));
  dat
};
```

Try the function on a PUMA (this is what PAT.test.model does.)

```
Sex.m(state=25,puma="03400");
      stat
Sex      Est  MoE
Male    71006 1164
Female  80261 1161
```

A few things to notice about the Sex.m return value.

1. The rownames are the categories are the same as the value of Sex.v()
2. The names of the dimnames are "Sex" and "stat." "Sex" is the same as the name in variables list component in the defining model (not shown). The "stat" part of the dimnames is c("Est", "MoE"). These correspond to the column names of an ACS ("B", "S" or "DP") table downloaded via the Application Programming Interface (API).

Testing Your Model

The function PAT.test.model will test your model by running your model on a PUMA, the state and puma FIPS codes are arguments.

PAT.test.model downloads the PUMS data for the selected geography. Next PAT.test.model downloads your defined marginal tables for the same PUMA geography. When you are synthesizing data at the tract level, the PAT.synth.data function will use tract level geography for marginal tables.

For the testing phase we want to check that PUMA marginal tables are close to the corresponding PUMS variable cross tabulations. The marginal tables will not match exactly with the PUMS cross tabulation as the PUMS data is a sub-sample of the ACS data records that go into the marginal tables. If there are inconsistencies in variable names or differences in factor labels between the PUMS variables and the marginal tables, these errors will show up at this testing step.

If PAT.test.model produces errors, go back and revise your PUMS variable definitions, marginal table computations, and the structural.zeros formula if you use one. If the PUMA marginal tables are too different from the PUMS data table cross tabulations, you should be suspicious and go back and check your model definition functions. This is a matter of judgment because of the sampling used in constructing the PUMS data sets and the PAT.test.model step will not produce an exact agreement. Also the vintages and periods for the marginal tables and the PUMS dataset might not be the same. For example you might want to use 5

year period marginals with 1 year PUMS data or even different vintages between the PUMS data and the marginal tables.

v3.0 4 May 2024 11:00