Title

ipfraking — Weight raking by iterative proportional fitting

Syntax

ipfraking [*if*] [*in*] **[pw=***weight***]** [, *options*]

options	Description
Control figures	
ctotal(*matname* [*matname* ...])	matrices of control totals
The new weight variable	
generate(*newvarname*)	new variable to write the raked weights to
replace	overwrite the existing weight variable
double	generate the new variable of double type
Convergence diagnostic and reporting	
iterate(#)	maximum number of iterations
tolerance(#)	convergence tolerance
ctrltolerance(#)	required accuracy of the controls
Trimming	
trimhirel(#)	the upper bound on the greatest factor by which the weights can increase
trimhiabs(#)	the upper bound on the greatest value the weights can achieve
trimlorel(#)	the lower bound on the smallest factor by which the weights can increase
trimloabs(#)	the lower bound on the smallest value the weights can achieve
trimfreqency(*keyword*)	stages of raking when weight trimming should be applied
Miscellaneous	
loglevel(#)	level of detail in the output
meta	store some meta-info concerning the raking procedure

Description

ipfraking performs iterative proportional fitting, or raking, to produce a set of calibrated survey weights such that the sample weighted totals of control variables match the known population totals. Typically, these control totals represent the number of population units in categories of a discrete variable, such as age groups in human surveys or industry in establishment surveys. These control totals must come from a census, a survey of much greater accuracy, or administrative data.

The adjustment of the weights is performed by adjusting each of the given control margins sequentially, until convergence is achieved. In other words, for a given control variable (e.g., gender), the total sizes of subpopulations are estimated, and the weights in separate categories (males, females) are multiplied by a group-specific factor (ratio of the known population total to the estimated total) so that the new set of weights produces total estimates conforming to the known totals.

Please cite this package as Kolenikov (2012), ipfraking: iterative proportional fitting weight calibration.

Options

Section

option(*whatever*)

Control figures

ctotal(*matrix_name* [*matrix_name* ...]) specifies the control totals.

Each matrix is expected to be a result of Stata total estimation command. If the latter was issued with **over(varname)** option, the matrix has to be additionally augmented with the name of that variable as a rowname. See Remarks and Examples.

See Remark 1 below.

Weight variable

generate(*newvarname*) specifies the name of the new variable to contain the raked weights.

replace overwrite the existing weight variable

double generate the new variable as double type

Convergence diagnostic and reporting

iterate(*#*) maximum number of iterations (default = 2000)

tolerance(*#*) convergence tolerance. Convergence will be declared if the largest relative difference of the weights in two successive iterations (a full cycle over all raking variables) does not exceed this value.

ctrltolerance(*#*) required accuracy of the controls. If, upon convergence of the algorithm (see the previous option), the relative difference of the weighted totals/means and the control totals/means is greater than this value, an error message will be issued.

See Remark 2 below.

Trimming

trimhirel(*#*) specifies the upper bound on the adjustment factor over the baseline weight. The weights that exceeds the baseline times this value will be trimmed down.

trimhiabs(*#*) specifies the upper bound on the greatest value of the raked weights. The weights that exceed this value will be trimmed down.

trimlorel(*#*) specifies the lower bound on the adjustment factor over the baseline weight. The weights that are smaller than the baseline times this value will be increased.

trimloabs(*#*) specifies the lower bound on the smallest value of the raked weights. The weights that are smaller than this value will be increased.

trimfreqency(*keyword*) specifies when the trimming operations are to be performed.

often means that the trimming operations will be performed after each marginal adjustment.

sometimes means that the trimming operations will be performed in the end of each iteration (cycle over all margins).

once means that the trimming operation will be performed once, after convergence has been achieved.

See Remark 3 below.

Miscellaneous

loglevel(*#*) level of detail in the output.

0 is the default value; only the iteration log will be produced.

1 provides additional output on the intermediate trimming steps.

2 is a lot of detailed (and not always useful) output.

meta puts the name(s) of the control vectors as a note stored with the variable specified in **generate()** option.

Returned values

Scalars:

r(converged)	1, if convergence of the algorithm was achieved, and 0 otherwise.
r(badcontrols)	1, if any of the control totals or means were not approximated accurately, and 0 otherwise.
r(maxreldif)	the largest relative difference of the weights at the last iteration
r(raked_mean)	the mean of the raked weights
r(raked_min)	the smallest raked weight; >= **trimloabs()** value if specified
r(raked_max)	the largest raked weight; <=

	trimhiabs() value if specified
r(raked_sd)	the standard deviation of the raked weights
r(raked_cv)	coefficient of variation of the raked weights; useful in eyeballing the design effect as

```
. display 1 + r(raked_cv)^2
```

r(factor_mean)	the average adjustment factor
r(factor_min)	the smallest adjustment factor; >= **trimlorel()** value if specified
r(factor_max)	the greatest adjustment factor; <= **trimhirel()** value if specified
r(factor_sd)	the standard deviation of the adjustment factor
r(factor_cv)	coefficient of variation of the adjustment factor

Macros:

r(ctotal)	the list of the control vectors (copy of the **ctotal()** option)

Remark 1 -- control vectors

Matrices that **ipfraking** expects to receive as inputs via **ctotal(...)** option must conform to the following specifications:

1. They need to be *1 x c* matrices (row-vectors)

2. They must have column names in estimation results format, i.e., *variable*:*#*.

3. They must have rowname that contains the categorical variable over the categories of which the totals were computed.

These requirements are easily satisfied by getting the matrices as result of

total *varname* [*weight*], **over(***varname*, **nolab)**

The **nolab** option is important, otherwise, the column names may contain the labels of the categorical variable that may be defined differently in the sample, or not defined at all. Also, only one variable should be specified in the **over()** option, as otherwise Stata provides generic column names **_subpop_***#* that are dependent on the data.

Remark 2 -- convergence

For algorithmic purposes, convergence is defined as achieving a stable state where the raked weights do not change (much) from iteration to iteration. In some sources, convergence of the raking algorithm is defined as whether the control totals are accurately approximated. These are two separate outcomes.

If the algorithm converges with inadequate accuracy of the totals (of which an error message will be issued), it means that the calibration constraints have been difficult to satisfy. The most common solutions to this problem is to omit some of the variables. In the (most common) case of the control totals being the sizes of subpopulation groups, one can collapse/combine some cells, thus specifying fewer control totals.

Remark 3 -- trimming

Weight trimming is often used in practice to reduce the spread of weights, and thus decrease the design effect. It may not be entirely clear what the effect of trimming might be on estimates that are but weakly related to the control variables, so this operation should be applied with caution.

The setting **trimfreq(sometimes)** appears to make the greatest sense. The weakness of the setting **trimfreq(once)** is that it does not guarantee that the resulting weights ensure the calibration constraints. The weakness of the setting **trimfreq(often)** is that the resulting weights may depend on the order in which the calibration variables are entered, especially when convergence is difficult to achieve.

Examples

Calibration over a single margin (post-stratification)

```
. webuse nhanes2, clear
. * setting up the totals
. generate byte _one = 1
. svy: total _one, over(sex, nolab)
. matrix total_sex = e(b)
. matrix rownames total_sex = sex
. * obtaining the sample
. sample 500, count by(region)
. * calibrating the weights
. ipfraking [pw=finalwgt], ctotal(total_sex) generate(rakedwgt1)
. * quality control
. total _one [pw=rakedwgt1], over(sex)
. matrix list e(b), format(%12.0g)
. matrix list total_sex, format(%12.0g)
```

Note that zero standard errors in the last estimation command are appropriate: there is no sampling variability in these totals since they are known. Generally, however, the variances will be overestimated, unlike with the proper {man poststratification}. Also, **ipfraking** performs the quality control internally and reports problems, if any.

Calibration over two margins

```
. webuse nhanes2, clear
. * setting up the totals
. generate byte _one = 1
. svy: total _one, over(sex, nolab)
. matrix total_sex = e(b)
. matrix rownames total_sex = sex
. svy: total _one, over(race, nolab)
. matrix total_race = e(b)
. matrix rownames total_race = race
. * obtaining the sample
. sample 500, count by(region)
. * calibrating the weights
```

```
. ipfraking [pw=finalwgt], ctotal(total_sex total_race)
    generate(rakedwgt2)
```

Calibration over two margins with weight trimming

```
. webuse nhanes2, clear
. * setting up the totals
. generate byte _one = 1
. svy: total _one, over(sex, nolab)
. matrix total_sex = e(b)
. matrix rownames total_sex = sex
. svy: total _one, over(race, nolab)
. matrix total_race = e(b)
. matrix rownames total_race = race
. * obtaining the sample
. sample 500, count by(region)
. * calibrating the weights
. ipfraking [pw=finalwgt], ctotal(total_sex total_race)
    trimhiabs(200000) generate(rakedwgt3)
```

Calibration over two margins with weight trimming, failure to achieve the control totals:

```
. webuse nhanes2, clear
. * setting up the totals
. generate byte _one = 1
. svy: total _one, over(sex, nolab)
. matrix total_sex = e(b)
. matrix rownames total_sex = sex
. svy: total _one, over(race, nolab)
. matrix total_race = e(b)
. matrix rownames total_race = race
. * obtaining the sample
. sample 500, count by(region)
. * calibrating the weights
. ipfraking [pw=finalwgt], ctotal(total_sex total_race)
    trimhiabs(200000) generate(rakedwgt4) trimhirel(5.4)
```

References

Deming, W. E., and Stephan, F. F. (1940). On a Least Squares Adjustment of a Sampled Frequency Table When the Expected Marginal Totals are Known. *Annals of Mathematical Statistics* 11 (4), 427–444. doi: 10.1214/aoms/1177731829.

Ruschendorf, L. (1995). Convergence of the Iterative Proportional Fitting Procedure. *The Annals of Statistics*, 23 (4), pp. 1160-1174. JSTOR link.

Deville, J.-C., Sarndal, C.-E., and Sautory, O. (1993). Generalized Raking Procedures in Survey Sampling *Journal of the American Statistical Association*, 88 (423) pp. 1013-1020. JSTOR link.

Kott, P. (2006) Using Calibration Weighting to Adjust for Nonresponse and Coverage Errors. *Survey Methodology*, 32 (2), pp. 133-142. Statistics Canada website access.

Author

Stanislav Kolenikov
Senior Survey Statistician
Abt SRBI
skolenik at gmail dot com

Also see

maxentropy package by M. Wittenberg (The Stata Journal article)