Paper 207-29

To Rake or Not To Rake Is Not the Question Anymore with the Enhanced Raking Macro

David Izrael, David C. Hoaglin, and Michael P. Battaglia
Abt Associates Inc., Cambridge, Massachusetts

Abstract

The SAS® macro RAKING for balancing a weighted sample, presented at SUGI 25 by Izrael et al. [1], has attracted many users by its simplicity and versatility. Raking is a technique that almost all survey researchers use, but they may encounter slow convergence when raking by multiple variables and multiple categories. Therefore, we have enhanced the macro by adding several novel features that enable users to obtain *comprehensive information on convergence* of all categories of the raking variables and to build graphs for maximum visualization of the convergence process. Some raking algorithms (including our original raking macro) provide almost no navigational aid and require that the user work blindly with a raking that is converging slowly. Our new macro gives the user an ability to *diagnose and remedy convergence problems*. By using the graphs, we demonstrate how to collapse categories of slowly converging variables so as to accelerate convergence, and how to use a new *prediction* feature to better navigate a raking process. For matching sample marginal proportions to population proportions, we discuss a newly introduced *percentage tolerance* and present an example of *iterative proportional fitting*.

Introduction

A survey sample may cover segments of the target population in proportions that do not match the proportions of those segments in the population itself. The differences may arise, for example, from sampling fluctuations, from nonresponse, or because the sample design was not able to cover the entire population. In such situations one can often improve the relation between the sample and the population by adjusting the base weights of the cases in the sample so that the marginal totals of the adjusted weights agree on specified characteristics with the corresponding totals for the population. This operation is known as sample-balancing or raking, and the population totals are usually referred to as control totals. Izrael et al. [1] discussed a raking process, referring to Bishop et al. [2] and Deming [3], and introduced a SAS macro for raking (sometimes referred to as the IHB raking macro) that combines simplicity and versatility. In light of the authors' experience with the macro and users' comments, the IHB raking macro has been enhanced in two key ways.

First, one simple definition of convergence of the raking algorithm requires that each marginal total of the raked weights be within a specified tolerance of the corresponding control total. In practice, when a number of raking variables are involved, one must check for the possibility that the iterations do not converge. As an extreme example, for the 2 x 2 table shown in Table 1, convergence is impossible.

Table 1. A 2 x 2 Table for Which Raking Cannot Produce Agreement with the Control Totals

Variable 1	Variable 2		Marginal Control Total
	1	**2**	
1	20	0	70
2	0	10	30
Marginal Control Total	50	50	

It is also possible that convergence requires a large number of iterations. Oh and Scheuren [4] note that the available convergence proofs make strong assumptions about the cell counts in the cross-classification of the raking variables – that no cells are empty or that some particular combination of nonempty cells is present. They recommend setting up the raking problem in a "sensible" manner to avoid: 1) imposing too many marginal constraints on the sample, 2) defining marginal categories that contain a small percentage of the sample, and 3) imposing contradictory constraints on the sample. Our experience indicates that, in general, raking on a large number of variables slows the convergence process. However, other factors also affect convergence. One is the number of categories of the raking variables. Convergence will typically be slower for raking on 10 variables each with 5 categories than for 10 variables each with only 2 categories. A second factor is the number of sample cases in each category of the raking variables. Convergence may be slow if any categories contain fewer than 5% of the sample cases. A third factor is the size of the difference between each control total and the weighted sample margin prior to raking. If some differences are large, the number of iterations will typically be higher. One can guard against the possibility of nonconvergence or slow convergence by setting an upper limit on the number of iterations (e.g., 50). We enhanced the IHB raking macro to provide graphical displays of the progress of each raking variable to the convergence criterion and to also display similar results for the individual categories of each raking variable. Also, we have developed a method to predict the number of iterations that will be needed for convergence when the maximum number of iterations is reached and convergence did not occur.

Second, in some surveys the base weights of all cases are equal to one (i.e., the sample has not been weighted), and the marginal control are expressed as percentages instead of as totals. The IHB raking macro now accepts percentage marginal controls; this modification makes the macro a convenient tool for Iterative Proportional Fitting (IPF).

Enhanced IHB Raking Macro

The enhanced macro employs essentially the same algorithm as the original Izrael et al. [1], and a macro call now looks like:

```
%rakinge(inds=,           /*    input data set          */
         outds=,          /*   output data set          */
         inwt=,           /*  weight being adjusted      */
                          /*  if there is no weight, 1 is assigned */
         freqlist=,       /*  list of data sets with marginal control totals or
                                                    percents*/
         outwt=,          /*    resulting weight        */

         byvar=,          /*  BY variable    */

         varlist=,        /* list of raking variables    */
         numvar=,         /* number of raking variables  */
         cntotal=,        /*  general   control total       */
         trmprec=1,       /* termination criterion, 1 default*/
         trmpct=,         /* termination based on marginal percent */
         numiter=50,      /* number of iterations, default  50 */
         dircont=work,    /* directory to save convergence data sets */
         prdiag =Y);      /* print detailed diagnostics    */
```

All macro parameters except *trmpct*, *dircont*, and *prdiag* were described in Izrael et al. [1]. Here we discuss only the three new parameters:

- trmpct enables the user to specify a percentage tolerance if he/she is interested in matching marginal proportions to population proportions rather than weighted marginal totals to population totals. If this parameter is indicated, the parameter *trmprec* is ignored, and the macro terminates when *all* differences between adjusted weighted sample percents and corresponding population percents are less than *trmpct.* As an example*,* for samples we work with, the default *trmprec* = 1 usually corresponds to *trmpct* = .0001% to .001%.   When *trmpct* is specified and convergence takes place, the macro terminates with a message like this:

```
        *** Program terminated at iteration 5 because all Calculated Percents

                differ from Marginal Percents by less than 0.001
```

For those users interested in fitting marginal percentages and typically working with an *unweighted* sample, the enhanced macro accepts a blank input weight (*inwt*), setting it to 1. If *trmpct* is present, the data sets specified in *freqlist* must contain the variable PERCENT, and the parameter *cntotal* must be present – the user can specify any number here. We prefer to set it to 100 (see Table 2 for an example).

- *dircont* specifies a directory where a new module of the macro creates data sets (one for each raking variable) with comprehensive information on convergence of all categories. Those convergence data sets (CDS) are named &dircont.._table_&varrake._&byvar, where &varrake is a raking variable name from &varlist and &byvar is a BY variable (if there is one). The CDS contain a raking variable, an iteration number, and a difference between an adjusted weighted total and a control total for each category of each raking variable at this iteration. Note that even if the macro parameter *trmpct* is specified, the retained differences will still be weighted totals; convergence based on this difference and on the difference in percents are almost identical. The CDS have proved to be useful for analyzing the raking process in cases where convergence is slow. In the next section we present graphs, based on the data from the CDS, and show how to use those graphs to collapse levels of slowly converging variables to speed up the raking process.

From our own and users' experience, we learned that the most unpleasant situation arises when the raking process does not converge in a specified number of iterations. The user often starts either to feverishly investigate the sample data or to increase by guessing the number of iterations (*numiter*) or the tolerance (*trmprec*). All those actions are usually counterproductive and waste time. Thus, along with the module that creates the CDS, we incorporated a module that, in case of non-convergence, uses the data from the CDS to *predict* the number of iterations needed for convergence.

The prediction is based on an empirical assumption that the logarithm of the difference between an adjusted weighted total and its control total declines linearly with the number of iterations. From our experience, this assumption works reasonably well when a slowly converging raking process approaches the specified number of iterations (*numiter*=50 in most of our examples). The enhanced macro extrapolates the last iteration slope and calculates at which iteration a slowly converging variable will cross a given tolerance threshold (*trmprec*). In the next section we give examples of a good and not-so-good prediction and make recommendations on what to do in either situation.

If convergence has not occurred in a specified number of iterations, the output diagnostics now generate a message like the following:

```
        **** Program terminated at iteration 50

        **** No convergence achieved. Try NUMITER = 52
```

throwing the user Ariadne's Thread in the labyrinth of raking.

- *prdiag* makes printing of raking diagnostics optional. This option is useful when the number of raking variables is large, some of them have many levels, and convergence is sluggish. In such situations the diagnostics may consume hundreds of pages. Thus, users interested only in obtaining a final raked weight may now turn the printing off, leaving only the last message on macro termination. The enhanced macro calculates the adjusted weighted marginal percents, the population percents, and the difference between them. All percents are included in the raking diagnostics. Table 2 shows the output of one iteration.

3

Table 2. Output of one iteration of enhanced IHB Raking macro

```
                          Raking  by VAR1, iteration - 1
```

| VAR1	Calculated margin	Marginal Control Total	Difference	Calculated %	Marginal Control %	Difference in %
1	3	20	17	27.273	20.000	7.273
2	5	35	30	45.455	35.000	10.455
3	3	45	42	27.273	45.000	-17.727
	==========	========		==========	========	
	11	100		100.00	100.00	

Examples of Working with the Enhanced IHB Raking Macro

Collapse Categories To Accelerate Convergence

We consider a raking process to be "converging slowly" if either it does not converge in a specified number of iterations or convergence takes substantially more iterations than usual. In our work, convergence usually takes place in 5-20 iterations. However, when the number of raking variables is large (say, more than 8) and some of the raking variables have numerous levels (the variable *state*, for instance), the process may take much longer to converge or may even not converge in an initially set number of iterations. The user now has options to proceed with raking. The first one is by using the predicted number of iterations from the diagnostics to rerake the sample, trying to achieve complete convergence. We illustrate this option later. However, the predicted number of iterations may be impractically large. Then, as a second option, one may attempt to preprocess the sample data.

A common strategy *collapses categories* of slowly converging variables. If, for instance, *state* is such a variable (with a value for each U.S. state and D.C.), it could be collapsed into, say, Census Division (9 levels) or even Census Region (4 levels). Of course, the user may not always have flexibility in collapsing. He/she may be *required* to rake by the original variables, or the "slow" variables may already be *dichotomous*. But if there is some flexibility in the statistical weighting methods, we recommend trying collapsing to accelerate convergence.

How does one determine which raking variables are "slow"? The most effective way to examine a convergence process is to draw graphs based on the CDS. Figure 1 displays a plot of a slow raking process involving 12 variables; the x-axis is the iteration number, and the y-axis is log_{10} of the maximum (taken over all categories of a given raking variable) of the absolute value of the difference between the adjusted weighted total and the control total. The reference line indicates the tolerance level, $log_{10}(trmprec)$, which is 0 in this example. One can easily construct this kind of graph from the CDS using standard SAS/GRAPH facilities.

Figure 1. Convergence of a Raking Process Involving 12 Variables
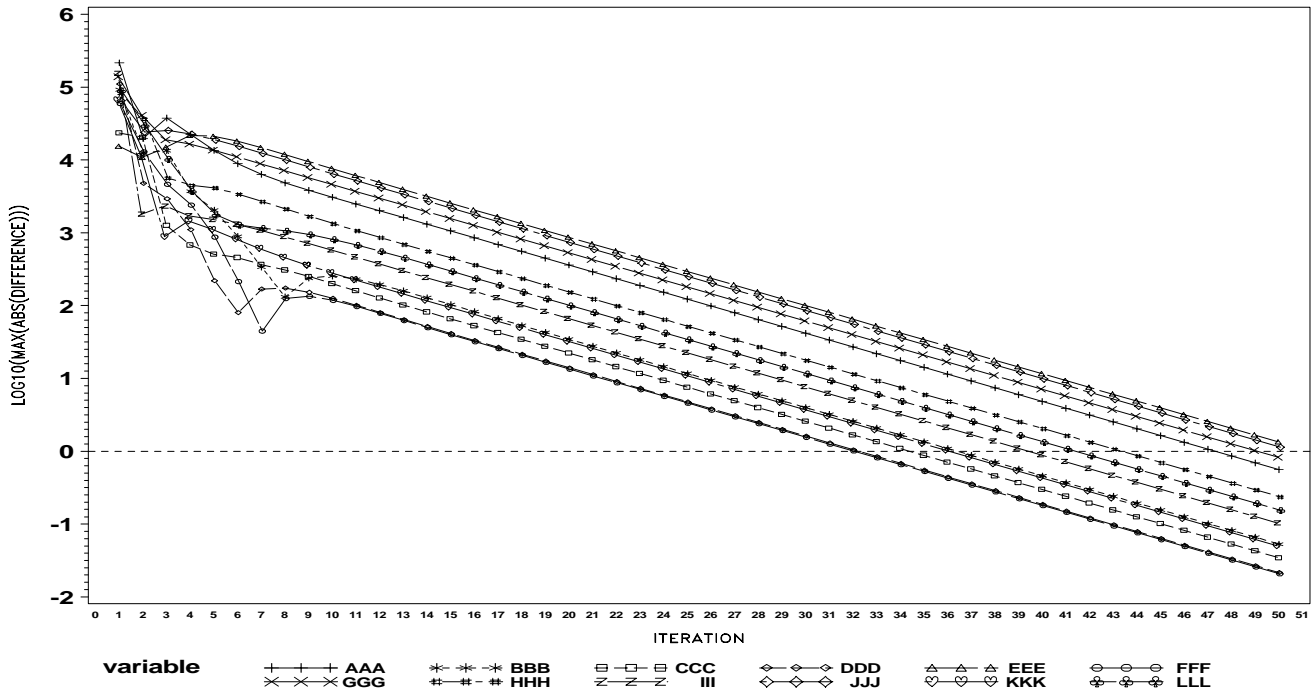
From the graph, one can easily single out the four slowest converging variables (their traces cluster distinctly higher): EEE, JJJ, GGG, and AAA. The variables GGG and AAA are dichotomous, so we are not able to collapse them. To explore how categories of the variables EEE and JJJ (which are ordinal) converge and which of them might be collapsed, we construct similar graphs *by categories* of those two variables (Figure 2).

Figure 2. Convergence of Variables EEE and JJJ before Collapsing
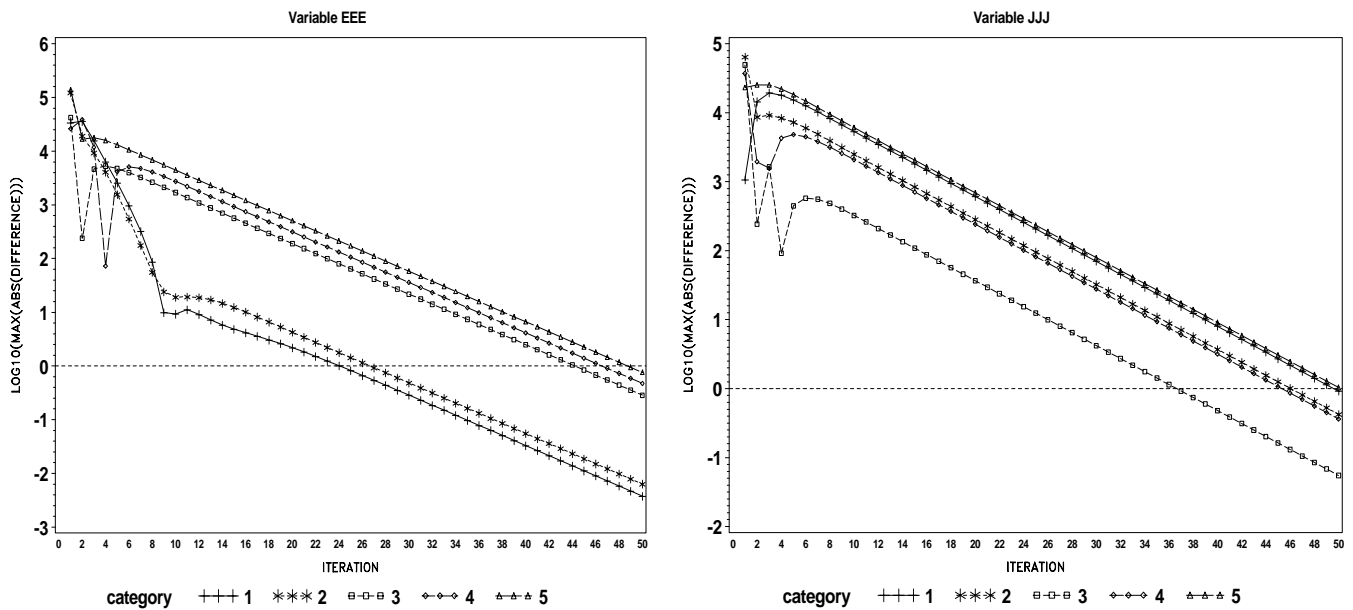
Besides visual exploration of convergence of slow categories, one should apply common sense when combining them. For *ordinal* variables, for instance, it would be logical to collapse *adjacent* categories. Taking the meaning of values of EEE and JJJ into account, in addition to the graphs in Figure 2, we collapsed categories 1 with 2, and 4 with 5 in both variables. Correspondingly, we combined the respective marginal totals, after which we reran the raking and constructed new convergence graphs for those two collapsed variables (Figure 3).

Figure 3. Convergence of Variables EEE and JJJ after Collapsing
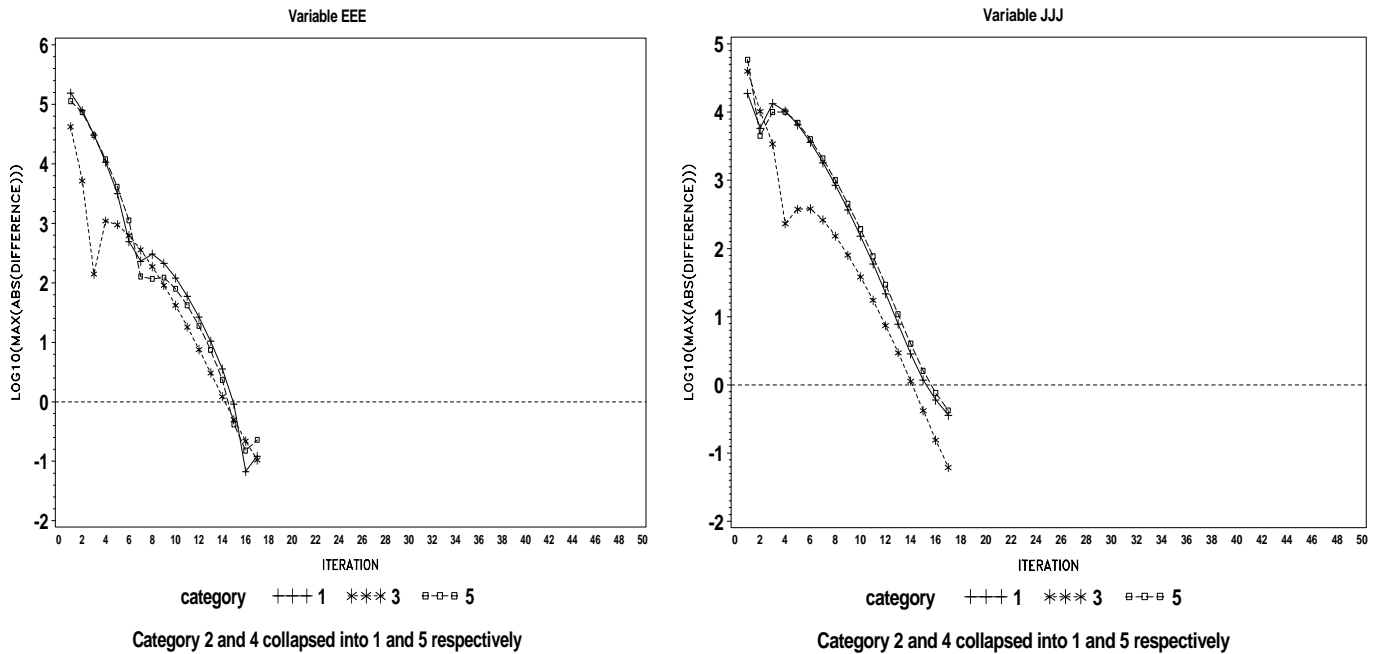
Since convergence of EEE and JJJ looked promising, we constructed a new overall convergence graph by raking variables (Figure 4).

Figure 4. Convergence of the Raking Process Involving 12 Variables. Variables EEE and JJJ Collapsed.
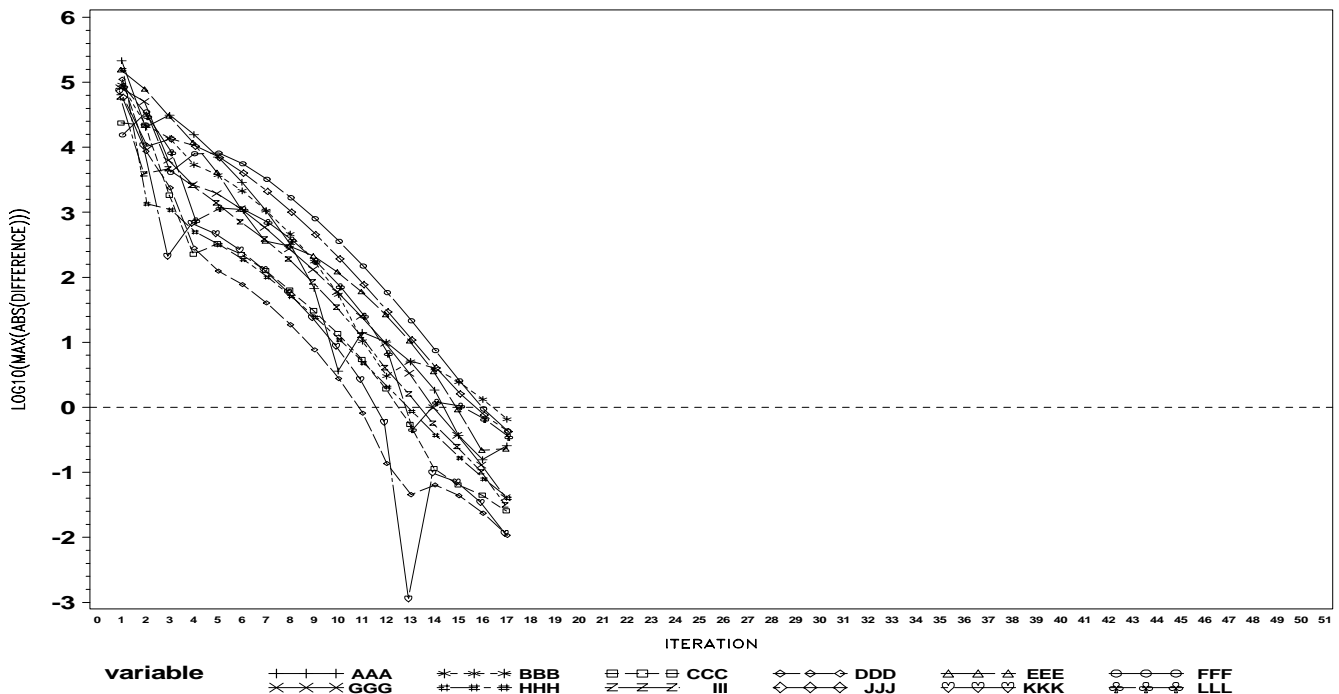
Comparing this graph with Figure 1, one can see that collapsing did play a dramatic role in speeding convergence. The raking process now converges in 17 iterations.

Use Predicted Number of Iterations

As we already noted, the user may not always have the flexibility to collapse categories, or he/she may still want to achieve convergence without altering the raking variables, i.e., to spend as many iterations as required. But how many are required? The enhanced macro calculates a predicted number of iterations needed for full convergence. The graph in Figure 5 demonstrates a two-variable raking process that initially did not converge in the default 50 iterations (vertical reference line) and predicted 65 as the needed number. When rerun, the raking did converge at exactly the 65-th iteration.

Figure 5. Good Prediction of the Number of Iterations Needed for Convergence

variable   +—+—+ AAA   *—*—* BBB

predicted number of iterations for convergence - 65
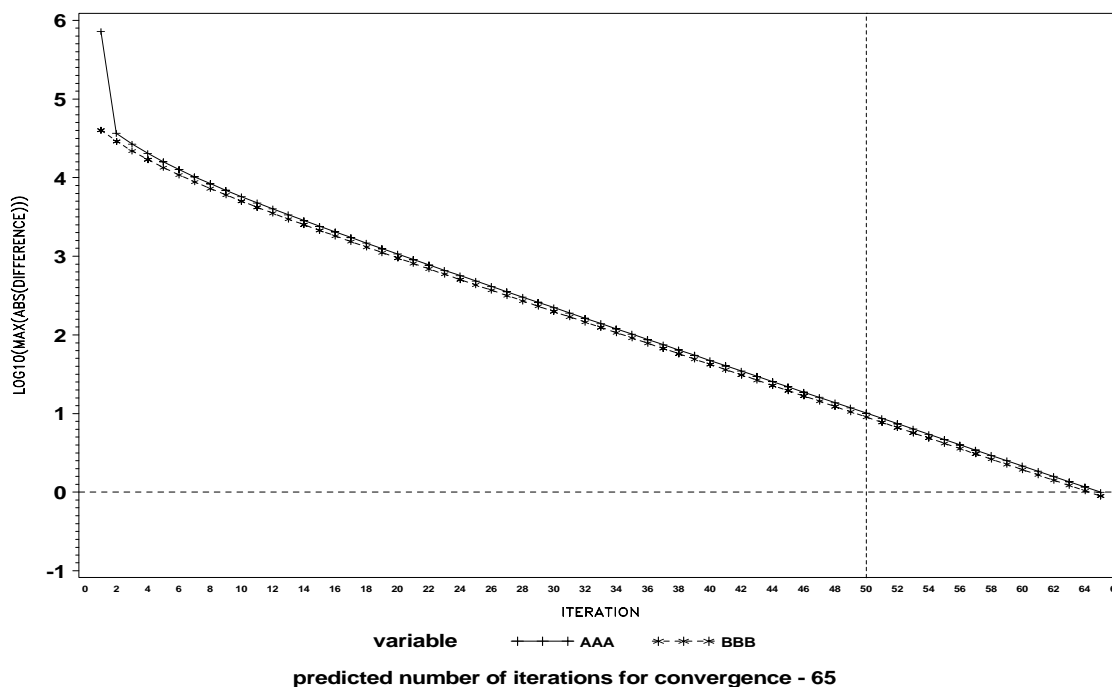
Note that we placed the predicted number of iterations in the footnote. This is an example of a "good" prediction.

In another example (Figure 6) --- a "not-so-good" prediction --- the raking process by 12 variables and tolerance 10 initially did not converge in 50 iterations (the vertical reference line), and the prediction was for 217 needed iterations (the dotted line extrapolates the slope from the 50-th iteration).

Figure 6. Not-so-good Prediction of the Number of Iterations Needed for Convergence

Predicted number of iterations for convergence - 217
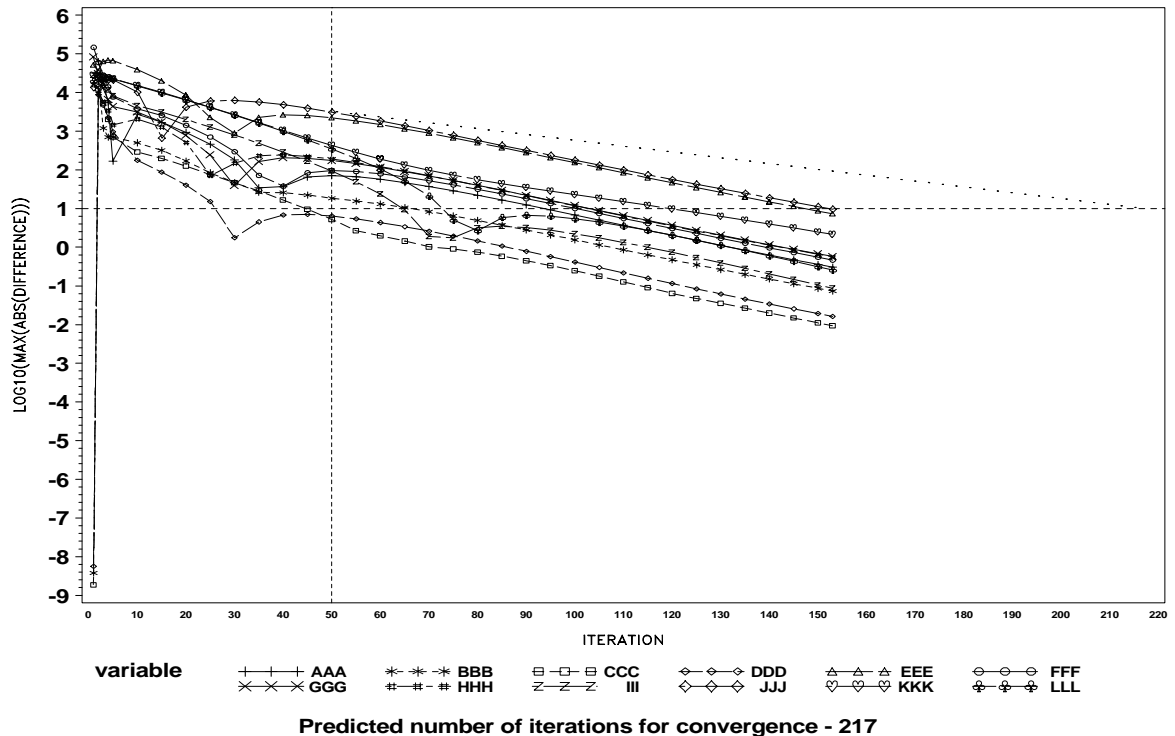
This graph shows that rerunning the raking with this predicted number resulted in convergence at the 153-rd iteration. The prediction overshot because the convergence curve of the slowest variable, JJJ, turned out to have a steeper slope beyond the 50-th iteration. Despite not hitting the target exactly, we nonetheless can be satisfied with this prediction as a guiding line in a fog of non-convergence.

In a fairly rare situation, rerunning the raking with the predicted number of iterations could give non-convergence again, with a new and much larger number of predicted iterations. In this case, it makes sense to thoroughly examine sample and population data and make appropriate changes.

If You Want To Fit Marginal Proportions and Have an Unweighted Sample

Frequently, the user working with a weighted or an unweighted sample needs to weight it to fit marginal population proportions. In the example shown below, we created an 11-case sample data set that contains two variables: VAR1, which takes values 1, 2, and 3 with frequencies 27.27%, 45.45% and 27.27%, respectively; and VAR2, which takes values 1 and 2 with frequencies 45.45% and 54.55%, respectively. We needed to weight this sample so that the distributions of VAR1 and VAR2 met the population distributions --- (20%, 35%, 45%) and (60%, 40%), respectively --- within a tolerance of 0.001%. The sample code looks like the following:

```
data sample;
input var1 var2;              /* input sample data */
cards;
1 2
2 1
3 1
2 1
3 2
```

8

```
2 2
1 1
2 1
3 2
2 2
1 2
;
run;

proc freq;          *** freqs of VAR1 and VAR2;
tables var1 var2;
run;

data var1;    *** create data set with marginal population percents for var1;
var1=1 ; percent=20; output;
var1=2 ; percent=35; output;
var1=3 ; percent=45; output;
run;

data var2;    *** create data set with marginal population percents for var2;
var2=1 ; percent=60;output;
var2=2 ; percent=40;output;
run;


%rakinge(    /* call enhanced raking */
inds=sample,
outds=outds,
inwt=,        /* if unweighted sample, weight =1 will be assigned by macro */
freqlist=,
outwt=outwgt,
byvar=,
varlist=var1 var2,
numvar=2,
cntotal=100,  /* any number here, 100 is most natural */
trmprec=1,
trmpct=0.001, /* macro will terminate based on this criterion */
numiter=50,
prdiag=Y
);

proc print data=outds;
run;
```

Original frequencies and output diagnostics for the first and last iterations are as follows:

The FREQ Procedure

var1	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	3	27.27	3	27.27
2	5	45.45	8	72.73
3	3	27.27	11	100.00

var2	Frequency	Percent	Cumulative Frequency	Cumulative Percent
1	5	45.45	5	45.45
2	6	54.55	11	100.00

Raking by VAR1, iteration - 1

VAR1	Calculated margin	Marginal Control Total	Difference	Calculated %	Marginal Control %	Difference in %
1	3	20	17	27.273	20.000	7.273
2	5	35	30	45.455	35.000	10.455
3	3	45	42	27.273	45.000	-17.727
	==========	========		==========	========	

9

```
        11              100                  100.00        100.00


                     Raking  by VAR2, iteration - 1

                     Marginal                          Marginal
           Calculated Control                Calculated Control  Difference
   VAR2      margin    Total    Difference        %        %       in %

     1       42.667      60      17.3333       42.667    60.000    -17.333
     2       57.333      40     -17.3333       57.333    40.000     17.333
            =========  ========              =========  ========
            100.000     100                   100.00     100.00


. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
.

. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
.

                     Raking  by VAR1, iteration - 5

                     Marginal                          Marginal
           Calculated Control                Calculated Control  Difference
   VAR1      margin    Total    Difference        %        %       in %

     1       20.000      20    0.000256716      20.000    20.000    -0.000
     2       35.001      35    -.000834329      35.001    35.000     0.001
     3       44.999      45    0.000577612      44.999    45.000    -0.001
            =========  ========              =========  ========
            100.000     100                   100.00     100.00


                     Raking  by VAR2, iteration - 5

                     Marginal                          Marginal
           Calculated Control                Calculated Control  Difference
   VAR2      margin    Total    Difference        %        %       in %

     1       60.000      60    0.000205597      60.000    60.000    -0.000
     2       40.000      40    -.000205597      40.000    40.000     0.000
            =========  ========              =========  ========
            100.000     100                   100.00     100.00


**** Program terminated at iteration 5 because all Calculated Percents differ from Marginal
Percents by less than 0.001
```

The output data set with a raked weight is as follows:

```
            Obs    var1    var2     outwgt

             1      1       1      10.2750
             2      2       1       8.8687
             3      2       1       8.8687
             4      2       1       8.8687
             5      3       1      23.1188
             6      1       2       4.8625
             7      1       2       4.8625
             8      2       2       4.1970
             9      2       2       4.1970
            10      3       2      10.9406
            11      3       2      10.9406
```

In fact, the operation performed in this example is identical to Iterative Proportional Fitting, which can also be
solved by IPF function of PROC IML, but in a more complicated way.

Conclusion

As the examples illustrate, the enhanced raking macro provides diagnostic facilities for monitoring the process of convergence. When a raking is converging slowly, graphs based on the convergence data sets can suggest a new target number of iterations or identify the slowest-converging variables and, within them, the slowest-converging categories, as candidates for collapsing. They may also help in understanding situations in which convergence is not possible. The ability of the enhanced macro to accept a percentage tolerance provides greater generality and permits convenient use for iterative proportional fitting.

References

1. Izrael, David, Hoaglin, David C., and Battaglia, Michael P. (2000), "A SAS Macro for Balancing a Weighted Sample." *Proceedings of the Twenty-Fifth Annual SAS Users Group International Conference*, Paper 275.

2. Bishop, Yvonne M. M., Fienberg, Stephen E., and Holland, Paul W. (1975), *Discrete Multivariate Analysis: Theory and Practice*. Cambridge, MA: MIT Press.

3. Deming, W. Edwards (1943), *Statistical Adjustment of Data*. New York: Wiley.

4. Oh, H. Lock and Scheuren, Fritz (1978), "Some Unresolved Application Issues in Raking Ratio Estimation." *1978 Proceedings of the Section on Survey Research Methods*, Washington, DC: American Statistical Association, pp. 723-728.

Contact Information

David Izrael
Abt Associates Inc.
55 Wheeler Street
Cambridge, MA 02138
617-349-2434
David_Izrael@abtassoc.com

11