Paper 258-25

A SAS Macro for Balancing a Weighted Sample

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

Abstract

It is often desirable to adjust the weights of a sample so that its marginal totals on specified characteristics agree with external control totals. Agreement is usually achieved iteratively, one variable at a time, by applying a proportional adjustment to the weights of the cases that belong to the same category of the variable. We developed a SAS macro for this operation (known as sample-balancing or raking), so that we would no longer have to export data from SAS and use a FORTRAN program that was originally designed to accept card input! The macro imposes no limit on the number of marginal variables or on the numbers of categories of those variables, and it allows the user to specify the tolerance for convergence. The paper reviews the algorithm and discusses experience with its use.

1. 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 weights of the cases in the sample so that the marginal totals of the adjusted weights on specified characteristics agree 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*.

The adjustment to control totals is sometimes achieved by creating a cross-classification of the control variables (e.g., age categories x gender x race x family income categories) and then matching the total of the weights in each cell to the control totals. This approach, however, can spread the sample thinly over a large number of cells. It also requires that the control totals exist for all cells of the cross-classification. Often this is not feasible (e.g., control totals may be available by age x gender x race but not when those cells are subdivided by family income). The use of marginal control totals for single variables (i.e., each margin involves only one control variable) often avoids many of these difficulties. In return, of course, the two-variable (and higher-order) distributions of the sample are not required to mimic those of the population.

A somewhat different problem motivated the original development of sample-balancing (Deming 1943). The Census Bureau needed to produce tabulations for the joint distribution of two (or more) variables in the U.S. population, in situations where information on the joint distribution was available only from a sample. The marginal totals, however, were available for the full population, and so the sample counts in the cells of the cross-classification were adjusted to provide an estimated tabulation that had the correct marginal totals.

Sample-balancing (or raking) usually proceeds one variable at a time, applying a proportional adjustment to the weights of the cases that belong to the same category of the control variable. Software for sample-balancing has been available for many years, but not as part of SAS or most other major systems. We are most familiar with a FORTRAN program developed in the 1960s by MarketMath, Inc. Although that program executes rapidly, it has a variety of disadvantages. One must create an ASCII input data set, painstakingly prepare control statements (the original program was designed to read input from cards), and then process its ASCII output data set. It can rake on at most 12 variables. Also it handles rounding in a way that can lose precision. For repeated use as part of a substantial analysis, a SAS macro has clear advantages.

In the remaining sections of this paper we briefly review the basic raking algorithm, describe the SAS macro, illustrate its use on a set of actual data, and discuss our experience with the macro.

2. Basic Algorithm

The procedure known as raking adjusts a set of data so that its marginal totals match specified control totals on a specified set of variables. (The term "raking" suggests an analogy with the process of smoothing the soil in a garden plot by alternately working it back and forth with a rake in two perpendicular directions.) It was originally devised in 1940 by W. Edwards Deming and Frederick F. Stephan for adjusting data from the U. S. Census.

In a simple 2-variable example the marginal totals in various categories for the two variables are known from the entire population, but the joint distribution of the two variables is known only from a sample (such as a 5% sample). In the cross-classification of the sample, arranged in rows and columns, one might begin with the rows, taking each row in turn and multiplying each entry in the row by the ratio of the population total to the sample total for that category, so that the row totals of the adjusted data agree with the population totals for that variable. The column totals of the adjusted data, however, may not yet agree with the population totals for the column variable. Thus the next step, taking each column in turn, multiplies each entry in the column by the ratio of the population total to the current total for that category. Now the column totals of the adjusted data agree with the population totals for that variable, but the new row totals may no longer match the corresponding population totals. The process continues, alternating between the rows and the columns, and agreement on both rows and columns is usually achieved after a few iterations. The result is a tabulation for the population that reflects the relation of the two variables in the sample.

The above sketch of the raking procedure focuses on the counts in the cells and on the margins of a two-variable cross-classification of the sample. In the applications that we encounter, involving data from complex surveys, it is more common to work with the survey weights of the n individual respondents. Thus, we describe the basic raking algorithm in terms of those individual weights, w_i, $i = 1, 2, \ldots, n$. For an unweighted (i.e., equally weighted) sample, one can simply take the initial weights to be $w_i = 1$ for each i.

In a cross-classification that has J rows and K columns, we denote the sum of the w_i in cell (j,k) by w_{jk}. To indicate further summation, we replace a subscript by a + sign. Thus, the initial row totals and column totals of the sample weights are w_{j+} and w_{+k} respectively. Analogously, we denote the corresponding population control totals by

T_{j+} and T_{+k} .

The iterative raking algorithm produces modified weights, whose sums we denote by a suitably subscripted m with a parenthesized superscript for the number of the step. Thus, in the two-variable cross-classification we use $m_{jk}^{(1)}$ for the sum of the modified weights in cell (j,k) at the end of step 1. If we begin by matching the control totals for the rows, T_{j+} , the initial steps of the algorithm are

$$m_{jk}^{(0)} = w_{jk} \ (j = 1,...,J;\ k=1,...,K)$$

$$m_{jk}^{(1)} = m_{jk}^{(0)} (T_{j+} / m_{j+}^{(0)})$$

$$m_{jk}^{(2)} = m_{jk}^{(1)} (T_{+k} / m_{+k}^{(1)})$$

The adjustment factors, $T_{j+} / m_{j+}^{(0)}$ and $T_{+k} / m_{+k}^{(1)}$, are actually applied to the individual weights, which we could denote by $m_i^{(2)}$, for example. In the iterative process an *iteration* rakes both rows and columns. Thus, for iteration s (s = 0, 1, ...) we may write

$$m_{jk}^{(2s+1)} = m_{jk}^{(2s)} (T_{j+} / m_{j+}^{(2s)})$$

$$m_{jk}^{(2s+2)} = m_{jk}^{(2s+1)} (T_{+k} / m_{+k}^{(2s+1)})$$

Raking can also adjust a set of data to control totals on three or more variables. In such situations the control totals often involve single variables, but they may involve two or more variables. In one example, in raking on three variables we might have control totals T_{a++}, T_{+b+}, and T_{++c}. In another example, the control totals might be T_{ab+} and T_{++c} --- a two-variable margin and a one-variable margin. In actually carrying out the raking for this second example (e.g., in using the SAS macro), it suffices to treat the two-variable margin as the one-variable margin for a composite variable, whose values simply index the cells of the underlying two-variable margin. Thus the SAS macro works only with one-variable margins.

Convergence

Convergence of the raking algorithm has received considerable attention in the statistical literature, especially in the context of iterative proportional fitting for log-linear models, where the number of variables is at least three and the process begins with a different set of initial values in the fitted table (often 1 in each cell). Bishop et al. (1975) discuss both iterative proportional fitting and raking. For our purposes it is enough that the iterative raking algorithm (ordinarily) converges, as one would expect from the fact that (in a suitable scale) the fitted cell counts produced by the raking are the weighted-least-squares fit to the observed cell counts in the full cross-classification of the sample by all the raking variables (Deming 1943, Chapter VII).

One simple definition of convergence 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 (e.g., because of sparseness or some other feature in the full cross-classification of the sample). One can guard against this possibility by setting an upper limit on the number of iterations. As elsewhere in data analysis, it is sensible to examine the sample (including its joint distribution with respect to all the raking variables) before doing any raking. For example, if the sample contains no cases in a category of one of the raking variables, it will be necessary to revise the set of categories and their control totals (say, by combining categories).

3. SAS Macro

Our SAS macro, RAKING, implements the basic algorithm in a way that gives the user considerable flexibility. For example, it imposes no limitation on the number of control variables or on the number of categories that a control variable may have. (Of course, an application with large numbers of control variables and categories will generally require a relatively long time to run.) Also, it gives the user control over the tolerance required for convergence and over the maximum number of iterations.

One must pass to the macro the following parameters:

- names of input and output data sets. They can be either permanent or work data sets. If the raking uses a BY- variable (described below), the user should set up a looping process outside the macro RAKING to retrieve the data for each level of the BY-variable and pass those data to the macro. The example in Section 4 demonstrates one way of doing this.

- names of the input weight to be raked and the resulting weight.

- names of data sets that contain the marginal control totals or marginal percentages (whichever the user has chosen) for each category of each variable (one data set for each raking variable). The macro checks that either the marginal percentages or marginal control totals are present and stops if both are missing. If both are present, the macro uses the marginal control totals. By default, the names of the marginal data sets are assumed to be the same as the names of the raking variables.

- name of BY-variable, if it is desired to rake separately for each level of a certain variable. In this case the data sets that contain the marginal percentages or control totals should have them for each level of the BY-variable. The example in Section 4 shows one possible way to prepare those data sets when a BY-variable is used. By default, raking is done over the whole input data set.

- list of names of raking variables and the number of them. There is no limit on the number of raking variables or on the number of levels of the raking variables; the macro checks that the number in the list agrees with the number of variables indicated by the user.

- general control total; it is required when marginal percentages rather than marginal control totals are used. If the raking uses a BY-variable, one must pass a general control total for each level of the BY-variable. The example in Section 4 demonstrates one possible way of creating a general control total for each level of the BY- variable outside the macro RAKING.

- convergence criterion (tolerance), which is given by the user as the desired maximum difference between each control total and the corresponding total of adjusted weights. The default is 1, which is often conservative (because it is small, relative to the control totals for typical populations). Increasing the tolerance to 100, for instance, may cut the number of iterations substantially.

- number of iterations; the default is 50.

In our experience the number of iterations needed for convergence ranged from 2 to 30 and depended on how heavily unbalanced the sample was initially and on what tolerance we offered. The default number of iterations is set to 50, but the macro terminates as soon as convergence is achieved. The most probable reason for non convergence may be that one or more levels of one or several raking variables are represented in the control totals but not in the sample. In this situation, macro terminates and writes the names of those variables into the LOG. The user may want to combine categories of those

raking variables --- in both the sample and the respective marginal data sets.

The working algorithm for the macro can be expressed in the following pseudo-code:

For iteration I from 1 to 50 do;
For variable VAR from TheList do;
For category C from 1st to Nth(VAR) do;

weighted_sample_total = sum (weight);
weight = weight (marginal_control_ total / weighted_sample_total);*
diff(VAR,C) = marginal_control_total - weighted_sample_ total;

end;
end;
if ALL abs(diff(VAR,C))<TOLERANCE then TERMINATE;
end;

Programmers should appreciate that raking may be time-consuming. To speed up the process, the input data set should contain only raking variables, input weight and, if needed, key variables for further merging with the rest of the raked data set. Loosening the tolerance for convergence could help to expedite the process, though at the expense of less thorough balancing.

The macro produces diagnostic output that contains the following information: number of iterations, name of raking variable currently being raked on, name of BY-variable if there is one, marginal control totals and calculated total weights for each level of the current raking variable, along with their difference. At termination, the macro gives the iteration number at which termination occurred and the reason, which is either that the termination tolerance has been met or that the process did not converge. The macro also writes diagnostics into the LOG, from several of the checks that it makes.

The following is the text of the raking macro with extensive explanatory comments.

```
%MACRO RAKING

(inds=,   /* input data set                 */
outds=,   /*output data set                 */
inwt=,    /*weight to be raked              */
freqlist=,/*list of data sets with marginal */
          /*freqs or marginal control totals*/
          /*they must contain name of raking*/
          /*   variable and either variable */
          /* PERCENT or MRGTOTAL, e.g. direct*/
          /* counts of marginal control totals;*/
          /* by default it is assigned names of*/
          /* raking variables                 */
outwt=,   /* resulting raked weight           */
byvar=,   /* variable BY which user might want */
          /* to set up raking.                */
          /* By default, raking is done       */
          /* over the whole input data set    */
varlist=, /* list of variables to be raked on */
numvar=,  /* number of raking variables       */
cntotal=, /* general control total; it is     */
          /* required if freqlist data sets   */
          /* contains PERCENT;                */
          /* if byvar not empty, cntotal      */
          /* must be present for each byvar   */
trmprec=1,  /* tolerance,default is 1         */
numiter=50);/*number of iterations,default is 50*/

%macro reqpar (param); /** checking on required **/
%if (&&&param eq ) %then %do; /** parameters    **/
     %put **** Program terminated: Macro parameter
%upcase(&PARAM) missing ****;
     endsas;
                          %end;
%mend;
%reqpar(inds)  %reqpar(outds) %reqpar (inwt)
%reqpar(outwt) %reqpar(varlist)
%reqpar(numvar) %reqpar(trmprec) %reqpar(numiter)
%let varlist=%upcase(&varlist);
%let byvar=%upcase(&byvar);
%let ls=%sysfunc(getoption(ls,keyword));
%let ps=%sysfunc(getoption(ps,keyword));

/*  checking on number of raking variables */
%if  (%scan(&varlist,&numvar) eq ) or
     (%scan(&varlist,%eval(&numvar+1)) ne )
%then %do;
 %put **** Program terminated: Number of variables in the
VARLIST ****;
 %put **** does not match NUMVAR ****;
     endsas;
                          %end;
data __i0;
set &inds;
weight=&inwt;
run;
%do i=1 %to &numiter; /* loop on iteration */
   %let sumterm = 0; /* cumulative termin flags*/
 %do j=1 %to &numvar;/* loop on raking variable */
    %let varrake=  %scan(&varlist,&j);
 %if (&freqlist ne ) %then
    %let dsfreq=%scan(&freqlist,&j);
  %else
     %let dsfreq=&varrake;

proc sort data=__i0;
by &varrake;
run;

proc summary nway data=__i0 ; *** calculate;
class &varrake;               *** adjusted;
var weight;                   ** weighted total;
output out=__i1(drop=_type_ _freq_) sum=sum&j;
run;

data __i0;   ** merge with marginal data set;
merge __i0(in=_1) __i1 &dsfreq(in=_2);
by &varrake;
%if &i=1 %then %do;
if (_1 and ^_2) or (_2 and ^_1) then do;
call symput('match','1');stop;end; else
call symput('match','2');
if mrgtotal ne . then call symput('mrg','1'); else call
symput('mrg','2');
if percent  ne . then call symput ('pct','1'); else call
symput('pct','2');
%end;
run;
%if &i=1 %then %do;
%if &match=1 %then %do;
%put
**** Program terminated: levels of variable &varrake do
not match ****;
%put
****in sample and marginal totals data sets ****;
endsas;
%end;
%if &pct = 1 and (&cntotal eq ) %then %do;
%put ** Program terminated: PERCENT is not missing and
CNTOTAL is missing **;
%put ** for raking variable &varrake **;
endsas;
%end;
%else
%if &pct=2 and &mrg=2 %then %do;
  %put **** Program terminated: Both PERCENT and MRGTOTAL
are missing ****;
endsas;
%end;
%end;

data __i0;
set __i0;
%if (&cntotal ne ) %then %do;
if mrgtotal ne . then /*case of marginal totals*/
cntmarg=mrgtotal;
else
if percent ne . then  /*case of marginal freqs*/
cntmarg=&cntotal.*percent/100;
%end;
%else %do;
```

```
if mrgtotal ne . then /*case of marginal totals*/
cntmarg=mrgtotal;
%end;
weight=weight*cntmarg/sum&j;***actual raking;
drop percent mrgtotal;
run;

data __i2(keep=&varrake sum&j cntmarg differ);
set __i0;
by &varrake;
if first.&varrake;
differ=cntmarg-sum&j;
run;

proc print label data=__i2; **print diagnostics;
%if  (&byvar ne) %then %do;
title3 "Raking &byvar - &s by &varrake, iteration - &i";
%end; %else %do;
title3 "Raking  by &varrake, iteration - &i ";
%end;
sum sum&j cntmarg;
label sum&j ='Calculated margin'
      differ='Difference'
      cntmarg='Margin Control Total';
run;

data __i2;
set __i2 end=eof;
retain comm 0;      *** termination test;
if  abs( differ)>&trmprec then comm=1;
if eof and comm=1 then
  call symput("term&j",'2'); ** continue with;
else                          ** iterations;
if eof then
call symput("term&j",'1'); ** convergence ;
run;                         ** achieved;
%let sumterm=%eval(&sumterm+&&term&j);
%end;

data __i0;
set __i0;
drop sum1-sum&numvar;
%if (&byvar ne ) %then
%put &byvar=&s iteration=&i numvar=&numvar
sumterm=&sumterm;
%else
%put  numvar=&numvar sumterm=&sumterm;
%if &sumterm=&numvar or &i=&numiter %then %do;
    /** termination criterion met **/
%if (&byvar ne ) %then
   %put **** Terminated &byvar &s at &i-th iteration;
                    %else
%put *** Task terminated at &i-th iteration ****;
title3 ' ';
data _null_; *** write diagostics to listing;
  set __i0;
if _n_=1;
    file print &ps &ls;
put ' ';
%if &sumterm=&numvar %then %do; ** convergence;
%if (&byvar ne ) %then %do;     ** achieved;
put "**** Program for &byvar &s terminated at iteration &i
because all calculated margins";
%end; %else %do;
put "**** Program terminated at iteration &i because all
calculated margins";
            %end;
put "differ from Marginal Control Totals by less than
&trmprec ";
run;
%end;
%else %do;          ** no convergence ;
%if (&byvar ne ) %then %do;
put "**** Program for &byvar &s terminated at iteration
&i";
                        %end; %else %do;
put "**** Program terminated at iteration &i ";
                                  %end;
put "**** No convergence achieved";
%end;

data &outds(drop=cntmarg); *** write output;
set __i0;                  *** data set;
rename weight=&outwt;
%let i=&numiter;
%end;
%end;
     proc datasets library=work nolist mt=data;
     delete __i0 __i1 __i2; ** purge work data;
   run;
     quit;
%mend;
```

4. Example

We illustrate the use of the macro with an example involving two raking variables, VAR1 and VAR2, and a BY-variable, AREA, which has two levels. The marginal percentage and general control total for each level of the BY-variable are obtained outside the example. They are in fact output data sets from PROC FREQ. Preliminary analyses of the data set showed that all categories of the raking variables represented in the marginal control data sets exist in the sample as well. The two-variable unweighted distribution is shown below. The actual raking uses the weights of the individual cases.

```
AREA=1
---------------------------------------------
              TABLE OF VAR1 BY VAR2

       VAR1      VAR2

       Frequency|       1|       2|  Total
       ---------+--------+--------+
              1 |      4 |     58 |     62
       ---------+--------+--------+
              2 |      1 |     38 |     39
       ---------+--------+--------+
              3 |      1 |     30 |     31
       ---------+--------+--------+
              4 |      1 |     32 |     33
       ---------+--------+--------+
              5 |      2 |    105 |    107
       ---------+--------+--------+
              6 |      3 |    140 |    143
       ---------+--------+--------+
       Total          12      403      415

 AREA=2
---------------------------------------------

             TABLE OF VAR1 BY VAR2

       VAR1      VAR2

       Frequency|       1|       2|  Total
       ---------+--------+--------+
              1 |      5 |     50 |     55
       ---------+--------+--------+
              2 |      4 |     49 |     53
       ---------+--------+--------+
              3 |      3 |     45 |     48
       ---------+--------+--------+
              4 |      3 |    115 |    118
       ---------+--------+--------+
              5 |      5 |    150 |    155
       ---------+--------+--------+
       Total          20      409      429
```

The following macro, %AREARAKE, processes marginal data sets and invokes the raking macro.

```
options nodate;
libname this '.';
filename automac '.';
options sasautos=automac;

%macro arearake;
proc sql noprint; *create macro variables with two;
                  * control totals;
select  total into: totals separated by '/' from
this.totals where area in (1,2);
%let nsqob =&sqlobs; quit;
%put Total AREA: &nsqob AREA TOTALS: &totals;
%do s=1 %to 2;
```

```
   data anal;
   set this.anal(where=(area=&s));* retrieve data set;
   run;                           * for AREA s       ;
%let areatota=%scan(&totals,&s,/);

data var1(drop=area); ** data set with margin for VAR1;
set this.freqvar1(where=(area=&s) keep=var1 area percent);

run;

data var2(drop=area); ** data set with margin - VAR2;
set this.freqvar2(where=(area=&s) keep=var2 area percent);

    /* invocation RAKING macro for s-th AREA */

%raking      (inds=anal,
              outds=out&s,
              inwt=inp_wgt,
              freqlist=,
              outwt=out_wgt,
              byvar=area,
              varlist=var1 var2,
              numvar=2,
              cntotal=&areatota,
              trmprec=1,
              numiter=50);
%end;
%mend;

%arearake;
```

The diagnostic output produced by the example is as follows:

Raking AREA - 1 by VAR1, iteration - 1

OBS	VAR1	Margin Control Total	Calculated Margin	Difference
1	1	8436.82	9780.03	-1343.21
2	2	7843.66	5900.72	1942.95
3	3	6279.48	4222.89	2056.59
4	4	9681.54	4658.27	5023.27
5	5	12613.97	15594.03	-2980.06
6	6	18846.60	21471.14	-2624.53
		==========	========	
		63702.07	61627.07	

Raking AREA - 1 by VAR2, iteration - 1

OBS	VAR2	Margin Control Total	Calculated margin	Difference
1	1	3985.13	4010.80	-25.6751
2	2	59716.94	59691.27	25.6751
		========	========	
		63702.07	63702.07	

Raking AREA - 1 by VAR1, iteration - 2

OBS	VAR1	Margin Control Total	Calculated margin	Difference
1	1	8436.82	8432.72	4.09860
2	2	7843.66	7843.33	0.33767
3	3	6279.48	6282.18	-2.70100
4	4	9681.54	9680.39	1.14329
5	5	12613.97	12615.18	-1.21712
6	6	18846.60	18848.27	-1.66143
		========	=========	
		63702.07	63702.07	

Raking AREA - 1 by VAR2, iteration - 2

OBS	VAR2	Margin Control Total	Calculated margin	Difference
1	1	3985.13	3985.65	-0.51898
2	2	59716.94	59716.42	0.51898
		========	=========	
		63702.07	63702.07	

Raking AREA - 1 by VAR1, iteration - 3

OBS	VAR1	Margin Control Total	Calculated margin	Difference
1	1	8436.82	8436.74	0.082867
2	2	7843.66	7843.66	0.006816
3	3	6279.48	6279.53	-0.054574
4	4	9681.54	9681.51	0.023099
5	5	12613.97	12613.99	-0.024611
6	6	18846.60	18846.64	-0.033598
		========	=========	
		63702.07	63702.07	

Raking AREA - 1 by VAR2, iteration - 3

OBS	VAR2	Margin Control Total	Calculated margin	Difference
1	1	3985.13	3985.14	-0.010489
2	2	59716.94	59716.93	0.010489
		========	=========	
		63702.07	63702.07	

Program for AREA 1 terminated at iteration 3 because all calculated margins differ from Marginal Control Totalby less than 1

Raking AREA - 2 by VAR1, iteration - 1

OBS	VAR1	Margin Control Total	Calculated Margin	Difference
1	1	14412.90	13656.37	756.53
2	2	11761.66	12797.00	-1035.33
3	3	17332.51	11652.68	5679.83
4	4	22902.75	25671.13	-2768.38
5	5	33801.18	34288.82	-487.64
		=========	=========	
		100211.00	98065.99	

Raking AREA - 2 by VAR2, iteration - 1

OBS	VAR2	Margin Control Total	Calculated margin	Difference
1	1	11677.85	12216.00	-538.151
2	2	88533.15	87995.00	538.151
		=========	==========	
		100211.00	100211.00	

Raking AREA - 2 by VAR1, iteration - 2

OBS	VAR1	Margin Control Total	Calculated margin	Difference
1	1	14412.90	14341.46	71.4371
2	2	11761.66	11722.09	39.5755
3	3	17332.51	17305.68	26.8278
4	4	22902.75	22976.50	-73.7537
5	5	33801.18	33865.26	-64.0866
		=========	==========	
		100211.00	100211.00	

Raking AREA - 2 by VAR2, iteration - 2

OBS	VAR2	Margin Control Total	Calculated margin	Difference
1	1	11677.85	11694.90	-17.0444
2	2	88533.15	88516.10	17.0444
		=========	==========	
		100211.00	100211.00	

Raking AREA - 2 by VAR1, iteration - 3

```
                 Margin
                Control     Calculated
OBS    VAR1      Total        margin      Difference

 1       1      14412.90      14410.63      2.26692
 2       2      11761.66      11760.41      1.25278
 3       3      17332.51      17331.67      0.84507
 4       4      22902.75      22905.08     -2.33193
 5       5      33801.18      33803.21     -2.03285
               =========    ==========
               100211.00     100211.00

            Raking AREA - 2 by VAR2, iteration - 3

                 Margin
                Control     Calculated
 OBS    VAR2      Total        margin      Difference

 1       1      11677.85      11678.39     -0.53928
 2       2      88533.15      88532.61      0.53928
               =========    ==========
               100211.00     100211.00

            Raking AREA - 2 by VAR1, iteration - 4

                 Margin
                Control     Calculated
 OBS    VAR1      Total        margin      Difference

 1       1      14412.90      14412.83      0.071728
 2       2      11761.66      11761.62      0.039637
 3       3      17332.51      17332.49      0.026733
 4       4      22902.75      22902.82     -0.073777
 5       5      33801.18      33801.24     -0.064321
               =========    ==========
               100211.00     100211.00

          Raking AREA - 2 by VAR2, iteration - 4

                 Margin
                Control     Calculated
OBS    VAR2      Total        margin      Difference

 1       1      11677.85      11677.87     -0.017062
 2       2      88533.15      88533.13      0.017062
               =========    ==========
               100211.00     100211.00

Program for AREA 2 terminated at iteration 4 because all
calculated margins differ from Marginal Control Totals by
less than 1
```

With the tolerance set to 1, the raking converged after 3 iterations for Area 1 and after 4 iterations for Area 2.

5. Conclusion

We have used the raking macro successfully on a variety of data sets, involving up to eight raking variables. For our work it offers much greater flexibility and convenience than the available stand-alone FORTRAN program. We find the diagnostic output very helpful in monitoring the progress of the iterations.

Acknowledgments

We thank Philip Primak, Senior Programmer at Genzyme Corporation, for reviewing the macro and making valuable suggestions.

References

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

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

Contact Information

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