# SAS® Macros for Constraining Arrays of Numbers

Coleman, Charles

US Census Burea

September 2015

**Paper POS131**

# SAS® Macros for Constraining Arrays of Numbers

Charles D. Coleman, U.S. Census Bureau

## ABSTRACT

Many applications require constraining arrays of numbers to controls in one or two dimensions. Example applications include survey estimates, disclosure avoidance, input-output tables, and population and other estimates and projections. If the results are allowed to take on any nonnegative values, raking (a.k.a. scaling) solves the problem in one dimension and two-way iterative raking solves it in two dimensions. Each of these raking macros has an option for the user to output a dataset containing the rakes. The problem is more complicated in one dimension if the data can be of any sign, the so-called "plus-minus" problem, as simple raking may produce unacceptable results. This problem is addressed by generalized raking, which preserves the structure of the data at the cost of a nonunique solution. Often, results are required to be rounded so as to preserve the original totals. The Cox-Ernst algorithm accomplishes an optimal controlled rounding in two dimensions. In one dimension, the Greatest Mantissa algorithm is a simplified version of the Cox-Ernst algorithm.

Each macro contains error control code. The macro variable `&errorcode` is made available to the programmer to enable error trapping.

## DISCLAIMER

This report is released to inform interested parties of research and to encourage discussion. The views expressed on methodological and technical issues are those of the author and not necessarily those of the U.S. Census Bureau.

## INTRODUCTION

Survey estimation, demographic work and disclosure avoidance often require constraining arrays of numbers to controls in one or two dimensions. For example, subnational population projections may be constrained to a national projection. Another example is fine-grained data being constrained to coarser-grained data of greater precision. The final data in every case should preserve, in some way, the structure of the original data. In one dimension, this means that if one initial data element is greater than another, then the transformed elements should preserve this relationship. Optimally, the ratios between elements should be preserved, but this can only be done in the specific case of controlling a vector whose nonzero elements are of the same sign as the control value, with the result left unrounded. Controlled rounding then destroys the ratios, but preserves the order up to elements whose rounded values are identical. The effect on the ratios depends on the magnitudes of the initial data elements and unit of rounding. The effect of controlled rounding on these ratios in vectors with large elements relative to the unit of rounding is minimal. On the other hand, initial vectors with elements about the same magnitude as the unit of rounding will find these ratios greatly perturbed. "Generalized raking" (Coleman, 2006a) of a vector of mixed sign to zero or a control of opposite sign to the nonzero data preserves the order of the original elements, while destroying the ratios. The two-dimensional equivalent of raking minimizes a function that measures the distortion from the original matrix.

One-dimensional raking multiplies a vector of data by the ratio of the control to the sum of the initial data. Damage to the structure of the original data is avoided only when the control is of the same sign as the nonzero initial data. When the data are of mixed sign or the control is zero or has the opposite sign of the nonzero data, generalized raking takes a weighted average of the ordinarily raked data and the projection of the original data onto the hyperplane defined by the control. The result, except in the cases of a zero control or zero initial sum, is nonunique. Generalized raking has several advantages over the earlier Akers-Siegel (1965) procedure: a continuous transformation using arithmetic operations is used instead of separate rakes for positive and negative data. It easily handles zeroes in the data. There is never any need to arbitrarily shift and then rake the data, when it is impossible to apply the Akers-Siegel procedure to the original data.

Two-dimensional raking, a.k.a. "iterative proportional fitting," the "RAS algorithm," and other names, is a much-rediscovered method for constraining a nonnegative matrix to positive row and column controls. The sums of the row and column controls (or "marginals") must be equal for it to work. It proceeds by alternately raking row data to row controls and column data to column controls until convergence. It minimizes a function that measures the distortion of the data. The result is unique. A sufficient condition for feasibility is that the original matrix be positive. When this is not attained, an algorithm based on linear programming can be used to determine feasibility. Some practical guidance for handling zeroes and "low" (i.e., values too low to be reported) is given to speed convergence. This

procedure does not generalize to three or more dimensions: a positive array with positive marginals can still be infeasible (Cox, 2003).

When the row and column controls are given in ranges whose sums overlap, the requirement that the two sets of controls have the same sum can be relaxed. The Range-RAS algorithm can be used to constrain the data so that the sum constraints are satisfied. It is very similar to the RAS algorithm, with the difference lying in how the rakes are computed.

The Cox-Ernst (1982) controlled rounding algorithm for matrices assures that integers are unchanged and nonintegers are rounded to one of their closest integers. A cost to rounding up is defined and minimized. Because of its complexity, this algorithm is not described in detail. This cost is decreasing in the remainder, so that it is less costly to round up 0.9 than 0.1. For vectors, the Greatest Mantissa algorithm (Coleman, 2006b) performs controlled rounding by rounding up numbers in order of their mantissas (a.k.a. remainders). This simplification of the Cox-Ernst algorithm is simple to describe and program.

Constraining data creates both benefits and costs. Estimates, by definition, contain error. Estimates may be constrained to enhance data usability or provide disclosure avoidance at the cost of changing variances. This cost may be offset by correcting anomalies that may be present in the data. Rounding can be used to enhance presentation by reflecting expected precision, whether in monetary units such as dollars and cents (and not fractional cents) or in numerical units such as thousands and millions.

After each algorithm's description is given, the SAS macro call is displayed. The macros themselves are in the accompanying file POS131.Coleman.zip.

## SOME DEFINITIONS

| | |
|---|---|
| Raking: | Multiplicatively constraining data to a control. In two dimensions, this is an iterative process. Also known as "scaling" in the mathematical literature. |
| Rounding: | Changing a number to a multiple of a prespecified value. For example, changing a fractional number to an integer. |
| Conventional Rounding: | The ordinary definition of rounding: changing a number to its closest multiple of a prespecified value. |
| Unconventional Rounding: | Rounding a number to a multiple of a prespecified value other than the closest one. This is, in effect, an "incorrect" rounding. |
| Controlled Rounding: | Rounding data to add up to a control. |

## NOTATION

### ONE DIMENSION

All operations are performed on the $n$ element vector $\vec{\mathbf{x}} = (x_i, \ldots, x_n)$. The resulting, transformed vector is denoted as $\vec{\mathbf{x}}' = (x_1', \ldots, x_n')$. The control value is $c$. The sum of the elements of $\vec{\mathbf{x}}$ is denoted as $b$. Let $\mathcal{H}$ be the hyperplane defined by the constraint $c = \sum_{i=1}^{n} x_i'$. Let $\lfloor y \rfloor$ be the greatest integer less than or equal to y. $m = y - \lfloor y \rfloor$ is the mantissa or fractional part of y.

### TWO DIMENSIONS

The original matrix is denoted by $\mathbf{A} = [a_{ij}]$ with $m$ rows and $n$ columns. The vectors of row and column controls are denoted by $\vec{\mathbf{u}} = (u_1, \ldots, u_n)'$ and $\vec{\mathbf{v}} = (v_1, \ldots, v_n)$, respectively. The convergence or tolerance criterion is denoted by $tol$.

## SAS NOTES

All macro file names are the same as the macro names in lowercase. All macros (except helpers) require SAS Interactive Matrix Language ®. One helper macro is `%EXPANDNAMES`, which generates sequentially numbered variables. An example usage is `%EXPANDNAMES(DEMO,2011,2013)`, which returns the string "`DEMO2011 DEMO2012 DEMO2013`". The other macro is `%NUMOBS(dsn=,n=)`, a variant of a macro created by Tyndall (2007, 4). This macro returns the number of observations in data set `dsn` in the global macro variable specified by `n`.

Each one-dimensional macro has required arguments:

- `dsin`: Input data set.
- `dsout`: Output data set

- `ctrldsin`: Control data set

and either

- `var`: Variables to be controlled, separated by blanks
- `ctrlvar`: Control variables, separated by blanks

or

- `varstem`: Stem of sequentially numbered variables
- `ctrlvarstem`: Stem of controls, same numerical sequence
- `first`: First number
- `last`: Last number

The macros have optional arguments. The macros that do not do BY-group processing have an option `idvar` to list variables to copy from the input data set to the output data set. Its main use is to identify observations. Macros with BY-groups dispense with `idvar` and require the mandatory arguments `byvar` and `byvar2`:

- `byvar`: BY-variables common to both input and control data sets
- `byvar2`: BY-variables unique to input data set

The combination of `&byvar` and `&byvar2` uniquely identifies an observation.

All raking macros (except generalized raking) have an optional `rakes=` argument to specify a data set to hold the rakes. The rake variables are sequentially numbered `rake1-raken`, where `n` is the number of variables. Macro variables `&idvar` (if specified) or `&byvar` and `&byvar2` are placed on these data sets.

The two-way raking macros have the following optional arguments:

- `missingok`: When set to 'Y', missing values in the input matrix are accepted and set to 0.
- `maxit`: Maximum iterations. Default is 100.

## RAKING

Raking, or scaling, "multiplicatively" controls data to controls. In one dimension, when the control and nonzero data are of the same sign, this is the ordinary rake which is simple multiplication. When one or both of these assumptions do not hold, the generalized rake, which is a weighted sum of the ordinary rake and the orthogonal projection to the control hyperplane is used. In two dimensions, when the marginals are fixed and equal the same sum, the two-way rake is used. When the marginals are in ranges, the Range-RAS algorithm is used.

### ONE DIMENSION

One dimension is equivalent to individual SAS variables. The simplest problem occurs when the control and nonzero input data are of the same sign. In addition to being solved by a simple multiplication (the ordinary rake) with a unique solution, the results satisfy several desirable properties. When the restriction on signs is removed, the generalized rake is used at the costs of a nonunique solution and loss of the ratio property: the ratios of the input data are not preserved.

#### Raking when Control and Nonzero Input Data are of Same Sign (Ordinary Rake)

The ordinary rake simply multiplies every element $x_i$ by $c/b$:

$$x_i' = x_i \frac{c}{b}. \qquad (1)$$

This is simple to program. The sufficient condition is that the nonzero elements of $\vec{\mathbf{x}}$ be of the same sign as $c$.

Consider raking the vector (1, 2, 3, 4) to sum to 20. The original vector sums to 10, so the rake factor is 20/10 = 2. Multiplying each element of the original vector by 2 produces the raked vector (2, 4, 6, 8).

The ordinary rake is monotonic and preserves the structure of the data in both order and ratio. That is,

$$c \gtreqless b \Rightarrow x_i' \gtreqless x_i \quad \forall x_i \neq 0 \qquad \text{(monotonicity)}, \qquad (2)$$

$$x_i > x_j \Rightarrow x_i' > x_j' \quad \forall i \neq j \qquad \text{(order)}, \qquad (3)$$

and

$$\frac{x_i'}{x_j'} = \frac{x_i}{x_j} \quad \forall x_j \neq 0 \quad \text{(ratio)}. \tag{4}$$

Monotonicity means that all nonzero data are shifted in the same direction as the change in their sum. The order property means that the ranking of the data is preserved. When the sign restrictions are relaxed, the ratio property must be lost to preserve the other two.

The SAS macro call is

```
%RAKE(dsin=,dsout=,ctrldsin=,var=,ctrlvar=,idvar=,first=,last=,varstem=,
ctrlvarstem=,rakes=);
```

Note: `%RAKE` is written with the assumption that all data are nonnegative. If negative data are encountered, a warning will be written to the output, `&errorcode` will be set to a negative value and the macro will run normally.

**Raking Data of Mixed Sign (Generalized Rake)**

The idea behind the generalize rake is to perturb the ordinary rake so as to preserve the order property. The generalized rake is a weighted average of the ordinary rake and the orthogonal projection of $\vec{\mathbf{x}}$ onto $\mathcal{H}$:

$$x_i' = w\frac{c}{b}x_i + (1-w)(x_i) + \frac{c-b}{n} \quad \forall i \tag{5}$$

The value of $w$ is determined heuristically. $w = 1$ reproduces the ordinary rake. $w = 0$ is the pure orthogonal projection, the only feasible solution for $b$ or $c = 0$. The heuristic should aim to produce a high (but not the highest, as this leads to a corner solution) value of $w < 1$ to produce an acceptable trade-off between the two pure $w = \{0,1\}$ solutions. A practical way to do this is to start with $w = 1$ and decrement $w$ by 0.1 until the order conditions (3) are satisfied. By starting with $w = 1$, the ordinary rake solution is not ignored when it is feasible. Allowing w to equal 0 provides insurance against missing the solution when $c = 0$. It is possible for $w$ to be negative, when $b$ and $c$ are of opposite sign.

Consider again the problem of raking (1, 2, 3, 4). This time, let the control total be −10. The ordinary rake uses a rake factor of −1, which simply changes the signs of the original data and reverses their order. The generalized rake using the heuristic above selects $w = 0.4$ to produce the final vector (−2.8, −2.6, −2.4, −2.2).

The generalized rake avoids many of the weaknesses of the Akers-Siegel (1965) procedure, at the cost of always having a nonunique solution when both $b, c \neq 0$ (Coleman, 2006a). Further research may be able to refine the algorithm to provide a unique solution given tuning parameter inputs.

The generalized rake strengthens the monotonicity condition (2) to

$$c \gtreqless b \Rightarrow x_i' \gtreqless x_i \quad \forall x_i \qquad \text{(monotonicity)}. \tag{2a}$$

Condition (2a) allows the generalized rake to operate on zero data. Of course, if any zero data represent true zeroes, they should be removed before raking.

The SAS macro call is

```
%GENRAKE(dsin=,dsout=,ctrldsin=,var=,ctrlvar=,idvar=,first=,last=,varstem=,
ctrlvarstem=);
```

The `rakes=` option is not supported, as it is not clear what rakes mean in this context.

**TWO DIMENSIONS**

The problem of controlling data in two dimensions has been in the literature since at least 1937 (Schneider and Zenios, 1990, 444). The original method of controlling positive data to positive marginals is considered first. Bregman (1967) gave it an optimality interpretation. Censor and Zenios (1979) loosened the constraints so that data can be controlled to ranges instead of scalar vectors, albeit without an optimality interpretation. This is a special case of the Constrained RAS (KRAS) algorithm (Lenzen, Gallego and Wood, 2009 and Temurshoev, Miller and Bouwmeester, 2013). The KRAS algorithm is not covered because of its complexity and remaining defects (Temorshoev et al., 2013).

**Two-Way Raking**

The two-way rake, a.k.a. the RAS Algorithm and Iterative Proportional Fitting, inter alia, alternately rakes rows and columns to their respective controls (or "marginals"). All elements of the initial matrix $\mathbf{A}$ must be nonnegative. The sum of the row controls must equal the sum of the column controls. The algorithm can be described as follows:

*Step 0.* (Initialization) Set $k = 0$ and $\mathbf{A}^0 = \mathbf{A}$.

*Step 1.* (Row Raking) For $i = 1,2,\dots,m$ define $\rho_i^k = u_i/\sum_j a_{ij}^k$, where $u_i$ is the marginal for row *i*. Define matrix $\mathbf{B}$ by the elements $b_{ij} = \rho_i^k a_{ij}^k$ for $i = 1,2,\dots,m$ and $j = 1,2,\dots,n$.
*Step 2.* (Column Raking) For $j = 1,2,\dots,n$ define $\sigma_j^k = v_j/\sum_i a_{ij}^k$, where $v_j$ is the marginal for column *j*. Define matrix $\mathbf{A}^{k+1}$ by the elements $a_{ij}^{k+1} = \sigma_j^k b_{ij}$ for $i = 1,2,\dots,m$ and $j = 1,2,\dots,n$.
*Step 3.* (Convergence Test) Compute $maxdiff = \max_{i,j}\left|a_{ij}^{k+1} - a_{ij}^k\right|$. If $maxdiff$ <then output $\mathbf{A}^{k+1}$ and stop.
*Step 4.* Replace $k$ with $k+1$ and return to Step 1.

It does not matter whether rows or columns are raked first (Bregman, 1967).

The value of *tol* should be less than the roundoff criterion, if any. For data that are to be rounded to integers, I like to use 0.1.

This algorithm can be troubled by the presence of zeroes, which can produce infeasibility. Fagan and Greenberg (1984) describe a procedure to determine the feasibility of the problem. A simpler method is to replacezeroes with small values, such as 1e-7. The end result will "steal" an insignificant amount from the positive entries. The output is then compared to the input to see if recoded zero cells have changed significantly. This indicates that the initial problem is infeasible. If feasible, the "stolen" data become irrelevant if the data are to be rounded afterwards.

Some data sets, such as those in the Bureau of Economic Analysis's Regional Economic Information System (REIS), come with "low" values suppressed, but included in summations. Any value that is flagged as low should be replaced with a small, nontrivial value less than the threshold for suppressing low values. For example, REIS values below 5 are flagged and replaced by zeroes. Replacing these values with 1 creates a good starting point. Moreover, this replacement may be necessary for a feasible solution to exist.

This problem is a type of matrix balancing problem. See Schneider and Zenios (1990) for a discussion of these problems. Bacharach (1970,79–80) proved that the solution is the unique nonnegative matrix $\mathbf{Y}$ that minimizes the information gain $\sum_{i,j,a_{ij}>0} y_{ij}\left[\ln\left(y_{ij}/a_{ij}\right) - 1\right]$. (The constant 1 is arbitrary, but convenient.) Bregman (1967) proved, for feasible problems, that the sequence $\{\mathbf{A}^k\}$ always converges to $\mathbf{Y}$. Schneider and Zenios (1990) provide additional results and some examples.

The SAS macro call is

```
%RAKE2WAYS(dsin=,dsout=,rowctrlds=,var=,varstem=,rowctrlvar=,colctrlds=,
colctrlvar=,idvar=,tol=,first=,last=,colctrlvarstem=,rakes=);
```

The required arguments to specify the marginals are:

- `rowctrlds:` Row control data set
- `rowctrlvar:` Row control variable name
- `colctrlds:` Column control variable data set
- `colctrlvar:` Column control variables (if &VAR is specified)

or

- `colctrlvarstem:` Column control variable name stem (if &VARSTEM is specified).

Argument `tol` is an optional convergence tolerance. Its default value is 0.1.

**The Range-RAS Algorithm**

When the marginals are in ranges whose sums overlap, the Range-RAS algorithm (Zenios and Censor, 1991) can control a matrix so that the sum of each row and column lies within its permitted range. The Range-RAS algorithm reduces to the RAS algorithm when each marginal is single-valued. Similar to two-way raking, all elements of the initial matrix $\mathbf{A}$ must be nonnegative.

Before describing the algorithm, define the middle function as $\mathrm{mid}(a,b,c)$, which returns the element in the middle (i.e., second-largest, equivalently, second-smallest.)

The Range-RAS algorithm proceeds as below:

*Step 0.* (Initialization) Set $k = 0$ and $\mathbf{A}^0 = \mathbf{A}$. Set initial row rakes $\rho_i^0 = 1, i = 1,\dots,m$. Set initial column rakes $\sigma_j^0 = 1, j = 1,\dots,n$.
*Step 1.* (Row Raking) For $i = 1,2,\dots,m$ define $\bar{\rho}_i^k = \bar{u}_i/\sum_j a_{ij}^k$ and $\underline{\rho}_i^k = \underline{u}_i/\sum_j a_{ij}^k$. Compute $\Delta\rho_i^k = \mathrm{mid}(\rho_i^k, \underline{\rho}_i^k, \bar{\rho}_i^k)$. Define matrix $\mathbf{B}$ by the elements $b_{ij} = \Delta\rho_i^k a_{ij}^k$ for $i = 1,2,\dots,m$ and $j = 1,2,\dots,n$.
*Step 2.* (Update Row Rakes) For $i = 1,2,\dots,m$ set $\rho_i^{k+1} = \rho_i^k/\Delta\rho_i^k$.

*Step 3.* (Column Raking) For $j = 1, 2, \ldots, n$ define $\bar{\sigma}_j^k = \bar{v}_j / \sum_i a_{ij}^k$ and $\underline{\sigma}_j^k = \underline{v}_j / \sum_i a_{ij}^k$. Compute $\Delta\sigma_j^k = \mathrm{mid}(\sigma_j^k, \underline{\sigma}_j^k, \bar{\sigma}_j^k)$. Define matrix $\mathbf{A}^{k+1}$ by the elements $a_{ij}^{k+1} = \Delta\sigma_j^k b_{ij}$ for $i = 1,2, \ldots, m$ and $j = 1,2, \ldots, n$.
*Step 4.* (Update Column Rakes) For $j = 1, 2, \ldots, n$ set $\sigma_j^{k+1} = \sigma_j^k / \Delta\sigma_j^k$.
*Step 5.* (Convergence Test) Compute $maxdiff = \max_{i,j} \left| a_{ij}^{k+1} - a_{ij}^k \right|$. If $maxdiff < tol$ then output $\mathbf{A}^{k+1}$ and stop.
*Step 6.* Replace $k$ with $k + 1$ and return to Step 1..

Since the Range-RAS algorithm is a variant of the RAS algorithm, it shares its difficulties with zeroes. It has the same objective function.

The SAS macro call is

```
%RRAS(dsin=,dsout=,var=,minrowctrlds=,minrowctrlvar=,mincolctrlds=,mincolctrlvar=,
maxrowctrlds=,maxrowctrlvar=,maxcolctrlds=,maxcolctrlvar=,idvar=,tol=,missingok=,
maxit=,first=,last=,varstem=,mincolctrlvarstem=,maxcolctrlvarstem=,rakes=);
```

The control arguments are

- `minrowctrlds:` Minimum row control data set
- `mincolctrlds:` Minimum row control data set
- `maxrowctrlds:` Maximum row control data set
- `maxcolctrlds:` Maximum row control data set
- `minrowctrlvar:` Minimum row control variable
- `maxrowctrlvar:` Maximum row control variable

and either

- `mincolctrlvar`: Minimum column control variables
- `maxcolctrlvar`: Maximum column control variables

or

- `mincolctrlvarstem`: Minimum column control variable stem
- `maxcolctrlvarstem`: Maximum column control variable stem

The other arguments have the same definitions as for `%RAKE2WAYS`.

## CONTROLLED ROUNDING

All of the controlled rounding presented is based on the Cox-Ernst algorithm, which seeks to 1) preserve integers and 2) minimize unconventional roundings. It does this by assigning a cost to rounding up, which increases in the distance between the original, unrounded data and their upwardly rounded values. It then minimizes the total cost of rounding. Cox and Ernst developed their algorithm for matrices. The Greatest Mantissa algorithm is a specialization to vectors (i.e., individual variables).

### TWO DIMENSIONS: THE COX-ERNST ALGORITHM

The Cox-Ernst algorithm is a type of transportation model (Causey, Cox and Ernst, 1985). Macro `%CONTROLROUND` is an adaptation of the SAS Institute's implementation of a transportation model (SAS Institute, n.d.). The PROC OPTMODEL call has been modified by including constraints to preserve zeroes (i.e., zero-restriction) and to bound the output by 0 and 1 to assure that every input cell is rounded to a nearest integer. Sands (2003) explains the Cox-Ernst algorithm to SAS users.

The SAS macro call is

```
%CONTROLROUND(dsin=,dsout=,var=,idvar=,first=,last=,varstem=,roundoff=);
```

`roundoff` is an optional argument that tells `%CONTROLROUND` to round off to the nearest integral multiple of the value specified. The default is 1, to round off to integers.

`%CONTROLROUND` calls PROC OPTMODEL's linear program solver. In SAS version 9.2, the dual simplex algorithm is used, possibly resulting in roundoff errors that result in an incorrect solution. In later versions, the network simplex algorithm is called to avoid these errors at the cost of increased memory usage. Invoking SAS with the -MEMSIZE option may be required.

#### ONE DIMENSION: THE GREATEST MANTISSA ALGORITHM

The Greatest Mantissa Algorithm (Coleman, 2006b) consists of taking each element of $\vec{\mathbf{x}}$ and subtracting the largest integer to obtain the mantissa. That is, $m_i = x_i - \lfloor x_i \rfloor$, where $\lfloor \cdot \rfloor$ is the `floor()` function. Then, assume that the difference between the control and the sum of the largest integers is $k = c - \sum_{i=1}^{n} \lfloor x_i \rfloor$. The algorithm then rounds up the $x_i$ corresponding to the $k$ largest $m_i$.

Ties may occur during this procedure: that is, the $k$th largest mantissa may be shared by multiple $x_i$. Thus, the user has to implement a tie-breaking procedure.

The Greatest Mantissa Algorithm is a one-dimensional simplification of the Cox-Ernst (1982) controlled rounding algorithm. In one dimension, this simplifies to assigning a cost to rounding up mantissas, $c(m_i)$, where $c'(m_i) < 0$, and exists for all $0 < m_i < 1$. The exact form of $c$ is irrelevant. Zeroes may not be rounded up. Theorem 1 of Coleman (2006b) shows that the cost of rounding is minimized by rounding up the largest mantissas. The signs of the data do not matter, so it is applicable to vectors of negative or mixed sign.

Table 1 shows an example of an unrounded vector with its conventional and controlled roundings and steps used to construct the controlled rounding. The original vector sums to 26, as shown in the rightmost column. Conventional rounding produces the vector shown immediately below. This vector sums to 25, 1 less than the original total. Therefore, conventional rounding does not preserve this vector's original sum and cannot be used when the sum has to be preserved. The next rows show the operation of the greatest mantissa algorithm. First, the integral parts are extracted. Their sum is 24, 2 less than the original total. Thus, two mantissas will have to be rounded up. The mantissas are shown below the integral parts. Their order of rounding is shown in the next row. Element 1 is rounded up first, as its mantissa, 0.9, is the largest. Element 5 has the second-largest mantissa, 0.4, so it is rounded up, too. The final vector is shown as the "Controlled Rounding" in the bottom row. This vector differs from the "Conventional Rounding" vector in element 5, which has been unconventionally rounded up. This demonstrates the necessity of unconventionally rounding numbers in a controlled rounding.

| Element | 1 | 2 | 3 | 4 | 5 | 6 | Sum |
|---|---|---|---|---|---|---|---|
| Original Vector | 5.9 | 6.1 | 5.1 | 4.2 | 2.4 | 2.3 | 26 |
| Conventional Rounding | 6 | 6 | 5 | 4 | 2 | 2 | 25 |
| Integral Parts | 5 | 6 | 5 | 4 | 2 | 2 | 24 |
| Mantissas | 0.9 | 0.1 | 0.1 | 0.2 | 0.4 | 0.3 | 2 |
| Order of Rounding Up | 1 | | | | 2 | | |
| Final Vector | 6 | 6 | 5 | 4 | 3 | 2 | 26 |

**Table 1. Greatest Mantissa Algorithm Example**

The SAS macro call is

```
%GMROUND(dsin=,dsout=,var=,idvar=,first=,last=,varstem=,seed=);
```

Macro `%GMROUND` breaks ties using a two-stage procedure. The first tie-breaker is the absolute value of the observation. The second is the order of the observations. If desired, the user may specify `seed` for random tie-breaking. This will cause global macro variable `&iseed` to be returned. `&iseed` can then be used for later random number generation.

## RAKING AND ROUNDING

Users often want to rake and round data. For example, in demography, one is interested in whole people: fractional people do not exist. Surveys may similarly require integral results.

The macros for this are:

One dimension:

```
%RAKEANDGMROUND(dsin=,dsout=,ctrldsin=,var=,ctrlvar=,idvar=,first=,last=,varstem=,
ctrlvarstem=,seed=,roundoff=,rakes=);
```

Two dimensions:

```
%RAKEANDROUND2WAYS(dsin=,dsout=,rowctrlds=,var=,rowctrlvar=,colctrlds=,colctrlvar=,
idvar=,tol=,first=,last=,varstem=,colctrlvarstem=,rakes=,missingok,roundoff=,
maxit=);
```

These macros' arguments are the unions of the arguments of their respective raking and rounding macros. Both

macros require uncommenting and altering the path in the first statement to include the appropriate rounding macro.

## BY-GROUP PROCESSING

I have created four corresponding macros with BY-groups for raking with or without rounding. Each macro replaces the optional argument `&idvar` with the required arguments `&byvar` and `&byvar2`.

Ordinary rake:

```
%RAKEBY(dsin=,dsout=,ctrldsin=,var=,ctrlvar=,first=,last=,varstem=,ctrlvarstem=,
byvar=,byvar2=,rakes=);
```

Round vectors:

```
%GMROUNDBY(dsin=,dsout=,var=,first=,last=,varstem=,seed=,byvar=,byvar2=);
```

Rake and round vectors:

```
%RAKEANDGMROUNDBY(dsin=,dsout=,ctrldsin=,var=,ctrlvar=,first=,last=,varstem=,
ctrlvarstem=,seed=,byvar=,byvar2=,rakes=);
```

Rake and round matrices:

```
%RAKEANDROUND2WAYSBY(dsin=,dsout=,rowctrlds=,var=,rowctrlvar=,colctrlds=,
colctrlvar=,byvar=,byvar2=,tol=,first=,last=,varstem=,colctrlvarstem=,maxit=,
roundoff=,rakes=);
```

In addition to uncommenting the reference to `%CONTROLROUND`, `%RAKEANDROUND2WAYSBY` requires uncommenting the reference to `%NUMOBS`.

## CONCLUSION

This paper has presented a variety of macros to control one- and two-dimensional data. The macros themselves can be found at https://sourceforge.net/p/constrainingarrays/code/ci/master/tree/. These macros should be considered works in progress, as they are subject to enhancements and revisions based on new knowledge. Suggestions for improvements are welcome.

## REFERENCES

Akers, D.S. and J.S. Siegel. 1965. "National Census Survival Rates, by Color and Sex, for 1950–1960." *Current Population Reports*, series P-23, no. 15. Washington, DC: U.S. Bureau of the Census.

Bacharach, M. 1970. *Biproportional Matrices and Input-output Change*. Cambridge: Cambridge University Press.

Bregman, L.M. 1967. "Proof of the Convergence of Sheleikhovskii's Method for a Problem with Transportation Constraints." *USSR Computational Mathematics and Mathematical Physics* **1**(1), 191–204.

Causey, B.D., L. H. Cox and L. R. Ernst. 1985. "Applications of Transportation Theory to Statistical Problems." *Journal of the American Statistical Association*. **80**(392), 903–909.

Censor, Y. and S. A. Zenios. 1991. "Interval-Constrained Matrix Balancing." *Linear Algebra and Its Applications*. **150,** 393-421.

Coleman, C. D. 2006a. "Generalized Raking." Manuscript, Washington, DC: U.S. Census Bureau.

Coleman, C. D. 2006b. "The Greatest Mantissa Algorithm." Manuscript, Washington, DC: U.S. Census Bureau.

Cox, L. H. 2003. "On Properties of Multi-Dimensional Statistical Tables." *Journal of Statistical Planning and Inference*. **117**(2), 251–273.

Cox, L. H. and Lawrence R. Ernst, 1982, "Controlled Rounding," *INFOR*. **20**, 423–452.

Fagan, J. and B. Greenberg, "Making Tables Additive in the Presence of Zeros," Statistical Research Division Report No. CENSUS/SRD/RR-84/17. Washington, DC: U.S. Bureau of the Census. Available at http://www.census.gov/srd/papers/pdf/rr84-17.pdf.

Lenzen M., B. Gallego and R. Wood. 2009. "Matrix Balancing under Conflicting Information." *Economic Systems Research*. **21**(1), 23–440.

Sands, R. 2003. "A SAS® Macro for the Controlled Rounding of One- and Two-Dimensional Tables of Real Numbers," *NESUG 2003: Proceedings.* Available at http://www.nesug.org/html/Proceedings/nesug03/st/st001.pdf.

SAS Institute. No date. "A Transportation Problem." Available at http://support.sas.com/documentation/cdl/en/ormpug/59679/HTML/default/viewer.htm#optmodel_sect5.htm

Schneider, M. H. and S. A. Zenios. 1990. "A Comparative Study of Algorithms for Matrix Balancing." *Operations Research.* **38**(3), 439–455.

Temurshoev U., R. E. Miller, and M. C. Bouwmeester. 2013. "A Note on the GRAS Method." *Economic Systems Research*. **25(**3), 361–367

Tyndall, R. 2007. "Give Your Macro Code an Extreme Makeover: Tips for Even the Most Seasoned Macro Programmer." SAS Institute Technical Note TS-739. Available at http://support.sas.com/techsup/technote/ts739.pdf.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Name: Charles D. "Chuck" Coleman
Enterprise: Construction Survey Statistics Methods Branch, Economic Statistical Methods Division, U.S. Census Bureau
Address: CENHQ 7K070E
City, State ZIP: Washington, DC 20233
Work Phone: (301)763-6068
E-mail: charles.d.coleman@census.gov