Paper 4284-2020

A SAS® Macro for Calibration of Survey Weights

Tony An, SAS Institute Inc.

ABSTRACT

In survey sampling, calibration is commonly used for adjusting weights to ensure that estimates for covariates in the sample match known auxiliary information, such as marginal totals from census data. Calibration can also be used to adjust for unit nonresponse. This paper discusses a macro for calibration that was developed using SAS/STAT® 15.1 and SAS/IML® software. The macro enables you to input the design information, the controls for the auxiliary variables, and your preferred calibration method, including either a linear or exponential method. Because an unbounded calibration method can result in extreme calibration weights, this macro also supports bounded versions of both linear and exponential calibration methods. The macro creates calibration replication weights according to the sample design and the specified calibration method. Examples are given to illustrate how to use the macro.

INTRODUCTION

In survey sampling, when a sample is drawn according to a complex sample design, data from both response variables and auxiliary variables are collected. Based on the sample design, sampling weights are created to ensure unbiased estimates for response variables in the population. Appropriate weighting is crucial, because it is well known that unweighted analysis or analysis that uses incorrect weights can yield biased estimates. For example, the outcome of the 2016 US presidential election differed greatly from the results predicted by many of the polls leading up to the election. In analyzing this discrepancy, Kennedy et al. (2018) pointed out that inadequate weighting for education was one of the culprits.

The primary goal of a survey is to estimate the population characteristics of certain response variables. To this end, it is both reasonable and statistically efficient that the weighted estimates for auxiliary variables (such as demographic variables) also match known information, such as marginal totals in census data.

For example, suppose that a business survey is to estimate utility usage among restaurants in a county. The total numbers of franchised and independent restaurants are readily available in the county's records. It is reasonable to expect that if you use the sampling weights to sum up the number of restaurants in the sample for each category of these establishments, the total number of the restaurants should match the records on file.

In order to achieve this match between estimates and known quantities, Deville and Särndal (1992) introduced calibration methods that involve modifying the original sampling weights. This weight adjustment method minimizes a distance measure between initial sampling weights and final modified weights subject to calibration equations defined by the known quantities to be matched. The final weights are also called calibration weights. When the calibration process is done properly, not only do the weighted survey estimates satisfy the desired auxiliary information constraints, but nonsampling errors such as nonresponse are also reduced, improving the precision of the estimates. See Haziza and Beaumont (2017) for an overview of calibration methods for survey weights.

This paper introduces a SAS® macro called %SurveyCalibrate that performs weight calibration in sample surveys. The macro uses SAS/STAT 15.1 and SAS/IML software to generate a set of optimal calibration weights that are based on the calibration controls for the auxiliary variables and by using your preferred calibration method.

After the calibration weights are created, both point estimates and associated inference should be based on specifying the calibration weights in the WEIGHT statement in the appropriate SAS/STAT survey data analysis procedure, such as the SURVEYMEANS, SURVEYFREQ, SURVEYREG, SURVEYLOGISTIC, or SURVEYPHREG procedure.

Calibration methods add variability in creating the calibration-adjusted weights because they use information from the observed sample. You must account for this extra variability in the analysis stage. If you use replication methods for variance estimation, then you must create calibration-adjusted replicate weights by applying the same calibration method in each set of replicate weights. The %SurveyCalibrate macro enables this by allowing you to input the design information in order to generate replicate weights derived from calibration weights for future analysis. By default, this macro automatically creates calibrated replicate weights by using the same calibration method that is used for the full sample. Optionally, you can request that the macro skip creating replicate weights. If you provide your own replicate weights, the macro calibrates these replicate weights.

The following sections discuss various calibration methods, explain the specification of macro parameters, and use an example to illustrate how the macro works.

CALIBRATION METHODS

The key to constructing calibration weights is to solve linear or nonlinear equations that are defined by the population quantities and the survey estimates that should match them. The calibration weights minimize a given distance function subject to a set of constraints.

Denote the sampling weights and the values of auxiliary variables in the sample as

$$(\mathbf{w}, \mathbf{X}) = (w_i, \mathbf{x}_i)$$

where

- $\mathbf{w}$ denotes the vector of sampling weights.
- $\mathbf{X}$ denotes the $n \times p$ matrix that corresponds to p auxiliary variables that you want to use to perform calibration. Each row $\mathbf{x}_i$ of $\mathbf{X}$ is the observed p-dimensional vector of auxiliary variables for the ith observation in the sample.
- $i = 1, 2, \ldots, n$ is the index of the observations in the sample.

Denote the population totals for the p auxiliary variables as $\mathbf{T} = (T_1, T_2, \ldots, T_p)$.

Let

$$\tilde{\mathbf{w}} = (\tilde{w}_1, \tilde{w}_2, \ldots, \tilde{w}_n)'$$

be the vector of desired calibration weights that satisfy the following constraints:

$$\tilde{\mathbf{w}}'\mathbf{X} = \sum_{i=1}^{n} \tilde{w}_i \mathbf{x}_i = \mathbf{T}$$

How these calibration weights are constructed depends on how you measure the distance between the original weights and the calibration weights. In general, a function $G(v, w)$ that measures the distance between two values v and w satisfies the following conditions:

- $G(v, w) \geq 0$ and $G(v, w | v = w) = 0$.
- $G(v, w)$ is differentiable with respect to v.
- $g(v, w) = \partial G(v, w) / \partial v$ is continuous.
- $G(v, w)$ is strictly convex.

Given such a distance function $G(v, w)$, the calibration weights $\tilde{\mathbf{w}}$ minimize the weighted distance between v and w. That is,

$$\sum_{i=1}^{n} w_i G(\tilde{w}_i, w_i) = \min_{\{\mathbf{v}: \sum_{i=1}^{n} v_i \mathbf{x}_i = \mathbf{T}\}} \sum_{i=1}^{n} w_i G(v_i, w_i)$$

Often the function $G(v, w)$ can be written as the single variable function $G(u)$, where the variable is the ratio $u = v/w$ such that $G(1) = 0$. If $G(v, w)$ is written as a single variable function $G(u)$, its derivative can be written as $g(u)$, and the inverse function of $g(u)$ is denoted as $\psi(u)$.

Using the p-dimensional Lagrange multiplier vector $\boldsymbol{\lambda}$, let

$$\phi(v_1, v_2, \ldots, v_n, \boldsymbol{\lambda}) = \sum_{i=1}^{n} w_i G(v_i, w_i) - \boldsymbol{\lambda}' \left(\sum_{i=1}^{n} v_i \mathbf{x}_i - \mathbf{T} \right)$$

Differentiating $\phi(v_1, v_2, \ldots, v_n, \boldsymbol{\lambda})$ with respect to v_i and setting the derivative to 0, you can write the calibration weight for the ith observation as

$$\tilde{w}_i = w_i \psi(\hat{\boldsymbol{\lambda}}' \mathbf{x}_i)$$

where $\hat{\boldsymbol{\lambda}}$ is the solution of the p equations:

$$\sum_{i=1}^{n} w_i \psi(\hat{\boldsymbol{\lambda}}' \mathbf{x}_i) \mathbf{x}_i = \mathbf{T}$$

The most commonly used calibration methods are linear, exponential, truncated linear, and logit (also known as truncated exponential), which correspond to various distance functions. These four calibration methods are available in the %SurveyCalibrate macro and described in the following sections.

Linear Method

The linear method, also often referred to as linear weighting, uses the generalized chi-square distance function

$$G(v, w) = \frac{1}{2}(v/w - 1)^2$$

The Lagrange multiplier vector $\boldsymbol{\lambda}$ can be solved as

$$\hat{\boldsymbol{\lambda}} = \left(\sum_{i=1}^{n} w_i \mathbf{x}_i \mathbf{x}_i' \right)^{-1} \left(\mathbf{T} - \sum_{i=1}^{n} w_i \mathbf{x}_i \right)$$

which gives the calibration weights as

$$\begin{aligned} \tilde{w}_i &= w_i (1 + \hat{\boldsymbol{\lambda}}' \mathbf{x}_i) \\ &= w_i \left(1 + \mathbf{x}_i' \left(\sum_{i=1}^{n} w_i \mathbf{x}_i \mathbf{x}_i' \right)^{-1} \left(\mathbf{T} - \sum_{i=1}^{n} w_i \mathbf{x}_i \right) \right) \end{aligned}$$

In rare cases, linear weighting can yield negative calibration weights. When this happens, the macro issues a warning, suggesting that you use an alternative calibration method.

Exponential Method

The exponential method uses the Kullback-Leibler information distance function:

$$G(v, w) = 1 + \frac{v}{w} \left(\log\left(\frac{v}{w} \right) - 1 \right)$$

Then if you use an iterative process such as Newton-Raphson optimization, the Lagrange multiplier vector $\boldsymbol{\lambda}$ can be determined from the following equation:

$$\sum_{i=1}^{n} w_i \exp(\hat{\boldsymbol{\lambda}}' \mathbf{x}_i) \mathbf{x}_i = \mathbf{T}$$

The calibration weights, often referred to as exponential weighting, are

$$\tilde{w}_i = w_i \exp(\hat{\boldsymbol{\lambda}}' \mathbf{x}_i)$$

Although exponential weighting can ensure positive final calibration weights, some weights can be relatively large compared to the rest of the weights. Worse, the solution $\hat{\boldsymbol{\lambda}}$ might not even exist.

When there are two categorical auxiliary variables, Deville, Särndal, and Sautory (1993) show that the exponential calibration method is equivalent to the traditional raking procedure proposed by Deming and Stephan (1940).

Truncated Linear Method

The truncated linear method sets bounds on the ratio of the final calibration weights to the original weights by L and U to ensure that all weights are positive compared to those produced by the linear method. It uses the modified generalized chi-square distance function with a lower and upper bound L and U $(0 < L < 1 < U)$:

$$G(v,w) = \begin{cases} \frac{1}{2}(\frac{v}{w} - 1)^2 & \text{if } L < \frac{v}{w} < U \\ \infty & \text{otherwise} \end{cases}$$

Then the calibration weights based on this distance function are

$$\tilde{w}_i = \begin{cases} w_i L & \text{if } \hat{\boldsymbol{\lambda}}'\mathbf{x}_i < L-1 \\ w_i(1+\hat{\boldsymbol{\lambda}}'\mathbf{x}_i) & \text{if } L-1 \le \hat{\boldsymbol{\lambda}}'\mathbf{x}_i \le U-1 \\ w_i U & \text{if } \hat{\boldsymbol{\lambda}}'\mathbf{x}_i > U-1 \end{cases}$$

where $\hat{\boldsymbol{\lambda}}$ is a solution to the estimation equation

$$\sum_{i=1}^{n} w_i \psi(\hat{\boldsymbol{\lambda}}'\mathbf{x}_i)\mathbf{x}_i = \mathbf{T}$$

where

$$\psi(u) = \begin{cases} L & \text{if } u < L-1 \\ 1+u & \text{if } L-1 \le u \le U-1 \\ U & \text{if } u > U-1 \end{cases}$$

However, if the bounds L and U are ill chosen—for example, if the length of (L, U) is too short—then there might be no $\hat{\boldsymbol{\lambda}}$ that solves the estimating equations.

You can specify either a lower bound or an upper bound, or both, when you call the %SurveyCalibrate macro. The macro issues an error message if calibration weights cannot be constructed by using the specified bounds. If you do not specify either or both bounds, the macro finds optimal bounds for you by minimizing the upper bound and maximizing the lower bound to avoid extreme calibration weights.

Logit Method

The logit method, also known as the truncated exponential method, modifies the exponential method in the same way that the truncated linear method modifies the linear method—by setting bounds on the ratio of the final calibration weights with respect to the original weights. Given a lower bound L and an upper bound U $(0 < L < 1 < U)$, the Kullback-Leibler information distance function used by the exponential method is modified as follows:

$$G(v,w) = \begin{cases} \left(\left(\frac{v}{w} - L\right)\log\left(\frac{v/w-L}{1-L}\right) + \left(U - \frac{v}{w}\right)\log\left(\frac{U-v/w}{U-1}\right)\right)\frac{(U-L)w}{(1-L)(U-1)} & \text{if } L < v/w < U \\ \infty & \text{otherwise} \end{cases}$$

Define

$$\phi(t) = \frac{L(U-1) + U(1-L)\exp\left(\frac{(U-L)t}{(1-L)(U-1)}\right)}{U-1+(1-L)\exp\left(\frac{(U-L)t}{(1-L)(U-1)}\right)}$$

and

$$\psi(t) = \begin{cases} L & \text{if } \phi(t) < L \\ \phi(t) & \text{if } L \le \phi(t) \le U \\ U & \text{if } \phi(t) > U \end{cases}$$

Let $\hat{\boldsymbol{\lambda}}$ be a solution to the estimation equation

$$\sum_{i=1}^{n} w_i \psi(\hat{\boldsymbol{\lambda}}'\mathbf{x}_i) = \mathbf{T}$$

Then the calibration weights based on this distance function are

$$\tilde{w}_i = w_i \psi(\hat{\boldsymbol{\lambda}}' \mathrm{x}_i)$$

The logit method sets bounds on the ratio of the final calibration weights with respect to the original weights by L and U to ensure that no calibration weights are too extreme. However, as in the case of the truncated linear method, if the bounds L and U are ill chosen, it is possible that there will be no solution to the estimation equations for $\hat{\boldsymbol{\lambda}}$.

You can specify either a lower bound or an upper bound, or both, for this method when you call the %SurveyCalibrate macro. The macro issues an error message if calibration weights cannot be constructed by using the specified bounds. If you do not specify either or both bounds, the macro finds optimal bounds for you by minimizing the upper bound and maximizing the lower bound.

MACRO PARAMETERS

The %SurveyCalibrate macro takes the following parameters, which are listed in the order of their importance and how frequently you might specify them. The parameters are classified into two groups: calibration parameters and replication parameters.

```
%macro SurveyCalibrate(
DATA=,       /* Input data set name                                    */
OUT=,        /* Output data set name                                   */
/*  Calibration parameters                                             */
METHOD=,     /* LINEAR | EXPONENTIAL | TRUNLINEAR | LOGIT              */
WEIGHT=,     /* Original weight variable                               */
CALWT=,      /* Calibration weight variable, default CalWt             */
CONTROLVAR=,/* Auxiliary control variables for calibration            */
CTRLTOTAL=, /* Marginal totals for CONTROLVAR                          */
EPS=,        /* Convergence criterion for stopping iteration, default=0.01 */
MAXITER=,    /* Maximum number of iteration, default=25                */
LOWER=,      /* Lower bound, must be in (0,1)                          */
UPPER=,      /* Upper bound, must be bigger than 1 or .                */
NOINT=,      /* Do not keep sum of sampling weights unchanged          */
/* Replication parameters                                              */
NOREPWT=,    /* Request no replicate weights                           */
VARMETHOD=, /* BRR | JK | BOOTSTRAP, default is JK                     */
REPS=,       /* Number of replicates for bootstrap or brr              */
CLUSTER=,    /* Cluster variables                                      */
STRATA=,     /* Strata variables                                       */
SEED=,       /* Random seed                                            */
FAY=,        /* Fay coefficient for BRR varmethod                      */
RATE=,       /* FPC for bootstrap replicate weights                    */
OUTJKCOEFS=,/* OUTJKCOEFS data set                                    */
REPWEIGHTS= /* Replicate weight variables                              */
);
```

Parameters in the first group control how the calibration weights are constructed. Parameters in the second group control how the replicate weights are constructed for future analysis. The second group takes effect only if you do not specify NOREPWT=TRUE.

If you want to create only the calibration weights and do not need the macro to generate the replicate weights, you can specify NOREPWT=TRUE.

The parameters that control the calibration method are as follows:

CALWT=*variable*

names a variable in the output data set to contain the calibration weights. The name must be different from any variable names in the input data set. If you do not specify this name, then by default the macro adds the prefix Cal_ to the name of the WEIGHT= *variable*. For example, if you specify WEIGHT=SampWt, the default calibration weight variable is named **Cal_SampWt** in the OUT= SAS data set.

CONTROLVAR=*variables*

specifies one or more variables in the DATA= input data set to be used as control variables for the calibration process. For multiple control variables, use spaces to separate control variable names. Both continuous and categorical variables can be used as control variables for the calibration process. Although you can specify a continuous variable directly, you must create indicator variables for each level of a categorical control variable. Each of these indicator variables has a value of 1 or 0. The value of 1 indicates that the observation belongs to the corresponding specific level that it presents for the category variable, and the value of 0 indicates otherwise.

For example, if you have a variable named **Gender** with the values 'M' and 'F' that you will use as the calibration control variables, you must create indicator variables for each level of **Gender** as follows:

```
data myDataSet; set myDataSet;
   Male = 0; Female = 0;
   if (Gender='M') then Male   = 1;
   if (Gender='F') then Female = 1;
run;
...
%SurveyCalibrate(
   DATA=myDataSet,
   ...
   CONTROLVAR=HouseholdIncome Male Female,
   ...
   );
```

CTRLTOTAL=*value-list*

specifies marginal totals, separated by spaces, as a list of positive numbers that correspond to the variables specified in the CONTROLVAR= parameter. This is a required parameter.

For example, if you have two control variables, **Male** and **Female**, indicating an observation's gender, with total population numbers of 300 and 400, respectively, then you specify these parameters as follows:

```
%SurveyCalibrate(
   ...
   CONTROLVAR=Male Female,
   CTRLTOTAL =300  400,
   ...
   );
```

DATA=*SAS-data-set*

names the input data set. This is a required parameter.

EPS=*value*

specifies the convergence criterion when the macro is searching for lower or upper bounds in the truncated linear or logit method. When the difference between bounds in two consecutive iterations is smaller than *value*, the desired bound is considered to be satisfied and the process stops. By default, EPS=0.01.

LOWER=*value*

specifies the lower bound that is used in the truncated linear or logit method. If you do not specify this option, the macro searches for an optimal lower bound that is based on the data. The *value* must be positive and less than 1.

MAXITER=*n*

specifies the maximum number of iterations when the macro searches for lower or upper bounds in the truncated linear or logit method. The value of *n* must be a positive integer. By default, MAXITER=25.

METHOD=LINEAR | EXPONENTIAL | TRUNLINEAR | LOGIT

specifies the calibration method. You can use the linear, exponential, truncated linear, or logit calibration method. If you do not specify this parameter, the procedure uses the linear method first by default. If the default linear method results in negative calibration weights, the macro switches to the exponential method.

NOINT=TRUE | FALSE

requests that the macro not use the constraint that keeps the sum of the calibration weights unchanged from the sum of the original weights. By default, NOINT=FALSE; in that case, the macro adds an intercept as a control variable, which ensures that the sum of the weights remains unchanged. But if you do not want this intercept to be a control variable in addition to what you specified in the CONTROLVAR= parameter (for example, because you already have one such intercept variable on the control variable list), then you must specifically request that the macro not create an intercept by specifying NOINT=TRUE.

OUT=*SAS-data-set*

specifies the output data set. If the calibration process is successful, the output data set will contain all the input data, as well as the calibration weights, and optionally contain the calibration-adjusted replicate weights. This is a required parameter.

WEIGHT=*variable*

names the variable that contains the sampling weights. This variable must be numeric, and the sampling weights must be positive numbers. If an observation has a weight that is nonpositive or missing, then the macro omits that observation from the calibration process. This is a required parameter.

UPPER=*value*

specifies the upper bound that is used in the truncated linear or logit method. If you do not specify this option, the macro searches for an optimal upper bound that is based on the data. The *value* must be greater than 1.

Among the calibration parameters, only the CONTROLVAR=, CTRLTOTAL=, DATA=, OUT=, and WEIGHT= are required parameters.

If you are not sure about the other specifications for your calibration process, you can simply leave them blank and let the macro select the optimal specifications. The macro will display relevant calibration information in the log that you can use as a reference for fine-tuning your future calibration processes.

The parameters in the second group control how the replicate weights are to be constructed for future analysis. You specify them exactly as you would in any SAS/STAT survey data analysis procedure. For more information about these specifications, see the chapters about PROC SURVEYMEANS, PROC SURVEYFREQ, PROC SURVEYREG, PROC SURVEYLOGISTIC, and PROC SURVEYPHREG in SAS Institute Inc. (2018).

If you provide your own replicate weights, the macro does not create new replicate weights. Instead it calibrates each replicate weight that you provide by using the same calibration method that it uses for the full sample.

Brief descriptions of the replication parameters follow:

CLUSTER=*variables*

names the variables that identify the clusters in a clustered sample design.

FAY=*value*

specifies the Fay coefficient for Fay's method, which is a modification of the balanced repeated replication (BRR) method. The *value* must be a nonnegative number less than 1. By default, the Fay coefficient is 0.5.

NOREPWT=TRUE | FALSE

requests that the macro not create replicate weights in the OUT= data set. If you do not specify this parameter, then by default NOREPWT=FALSE; in that case, the macro automatically creates calibrated replicate weights by using the same calibration method that it uses for the full sample. If you do not want to have these replicate weights created, you must specify NOREPWT=TRUE.

OUTJKCOEFS=*SAS-data-set*

names a SAS data set in which to store the jackknife coefficients for the jackknife variance estimation method.

RATE=*value* | *SAS-data-set*
specifies the sampling rate, which the macro uses to create calibrated bootstrap replicate weights. This parameter is ignored for the jackknife or the BRR variance estimation method.

REPS=*n*
specifies the number of replicates.

REPWEIGHTS=*variables*
names the prefix of the replicate weight variables that you provide in the DATA= parameter. If you specify these replicate weight variables, the macro does not create new replicate weights. Instead, the macro calibrates these replicate weights by using the same calibration method that it uses for the full sample. For example, if your replicate weight variables in the data set are **RepWt1**–**RepWt100**, then you specify

```
%SurveyCalibrate(
   ...
   REPWEIGHTS=RepWt,
   ...
   );
```

SEED=*n*
specifies the initial seed for random number generation that is used to produce replicate weights.

STRATA=*variables*
names the variables that identify the strata in a stratified sample design.

VARMETHOD=BOOTSTRAP | BRR | JK
specifies one of the following replication variance estimation methods: bootstrap, BRR, or jackknife.

EXAMPLE

This example uses a data set from the National Health and Nutrition Examination Survey I (NHANES I) Epidemiologic Followup Study (NHEFS). The NHEFS is a national longitudinal survey that is conducted by the National Center for Health Statistics, the National Institute on Aging, and other agencies of the Public Health Service in the United States. For more information about the survey and the data sets, see the Centers for Disease Control and Prevention website (https://wwwn.cdc.gov/nchs/nhanes/Default.aspx).

For purposes of illustration, 174 observations from the 1992 NHEFS vital and tracing status data set are used. The observations are obtained from five strata; each stratum contains either two or three primary sampling units.

The following variables, which are used in this example, are saved in the data set **Mortality**:

- **ID**, unit identification
- **VarStrata**, stratum identification
- **VarPSU**, identification for primary sampling units
- **SWeight**, sampling weight associated with each unit
- **Age**, the subject's reported age at the 1992 interview if the subject was alive at that time; otherwise, the subject's age at death
- **VitalStatus**, vital status of subject in 1992 (1 = alive, 3 = dead, 4 = unknown, 5 = traced alive with direct subject contact, 6 = traced alive without direct subject contact)
- **PovArInd**, indicator for poverty area where subject's household was located (1 = poverty area, 2 = nonpoverty area)
- **Gender**, gender of subject (1 = male, 2 = female)

The following statements create the **Mortality** data set:

```
data Mortality;
   input ID VarStrata VarPSU SWeight Age VitalStatus PovArInd Gender;

   datalines;
      1  03  1  13312    66  1   1   1
      2  03  1   7941    71  3   1   2
      3  03  1  16048     .  4   1   1
      4  03  3   9298    58  3   1   1
      5  03  2  15336    56  3   1   2
      6  03  1  14744    63  1   1   1
      7  03  2  83729    70  1   2   2
      8  03  3 106492    57  1   2   1
      9  03  3  78083    81  3   2   2

   ... more lines ...

   1272  10  2  34808    62  1   2   1
   1273  10  3   5005    81  3   1   2
;
```

For purposes of illustration, suppose that through other data sources, such as the census, the known marginal totals in the surveyed population by their residency and gender are as shown in Table 1.

Table 1 Marginal Totals by Residency and Gender

PovArInd	Number of Residents	Gender	Number of People
Poverty	536,207	Male	3,503,378
Nonpoverty	6,554,845	Female	3,587,674
Total	7,091,052	Total	7,091,052

However, the weighted sums in the sample over the demographic variables **PovArInd** and **Gender**, shown in Figure 1 on the next page, differ from these known marginal totals in the surveyed population.

To ensure that the weighted sums in the sample match the known population marginal totals in Table 1, you can call the %SurveyCalibrate macro to calibrate the weights that are saved in the variable **SWeight**.

Before calling the macro, because both **PovArInd** and **Gender** are categorical variables, the indicator variables **Poverty**, **NonPoverty**, **Male**, and **Female** are created as follows to present each level of these variables, so that the marginal totals in Table 1 can correspond to the weighted sum of these indicator variables:

```
data Mortality; set Mortality;
   Poverty=0; NonPoverty=0; Male=0; Female=0;
   if (Gender=1)   then Male      =1;
   if (Gender=2)   then Female    =1;
   if (PovArInd=1) then Poverty   =1;
   if (PovArInd=2) then NonPoverty=1;
run;
```

The following code uses the truncated linear method to calibrate the weights to the totals shown in Table 1. The macro call specifies only that the upper bound is 2.0 and lets the %SurveyCalibrate macro search for the lower bound.

```
%SurveyCalibrate(
   DATA       = Mortality,
   OUT        = Final,
   METHOD     = TRUNLINEAR,
   WEIGHT     = SWeight,
   CONTROLVAR = Poverty  NonPoverty Male     Female,
   CTRLTOTAL  = 536207   6554845    3503378  3587674,
   UPPER      = 2.0,
   VARMETHOD  = bootstrap,
   SEED       = 100,
   CLUSTER    = VarPSU,
   STRATA     = VarStrata
);
```

After the macro is finished, the bound that the truncated linear method uses is reported in the log:

```
NOTE: After 7 iterations, the lower bound is set to LOWER=0.3522109375 for
      the TRUNLINEAR method.
NOTE: The calibration weights Cal_SWeight are created by using the TRUNLINEAR
      method with LOWER=0.3522109375 and UPPER=2 bounds.
```

When you use the calibration weights **Cal_SWeight** in the SAS data set **Final** as the weight variable, the weighted sums in the sample over the demographic variables **PovArInd** and **Gender** now match the marginal totals in Table 1 for residency and gender, as shown in Figure 2.

Figure 1 Weighted Sums Using Original Sampling Weights **SWeight**

The SURVEYMEANS Procedure

Statistics

Variable	Level	Sum	Std Error of Sum
PovArInd	1	1507352	163940
	2	5583700	566791
Gender	1	3018151	390173
	2	4072901	490615

Figure 2 Weighted Sums Using Calibrated Weights **Cal_SWeight**

The SURVEYMEANS Procedure

Statistics

Variable	Level	Sum	Std Error of Sum
PovArInd	1	536207	58324
	2	6554845	657766
Gender	1	3503378	526071
	2	3587674	513790

To check the calibration adjustments made to the original weights, the following code creates the variable **wt_change** as the ratio of the calibrated weights to original sampling weights. Figure 3 gives a snapshot of the calibration adjustment to the sampling weights. The ratio ranges from 0.35 to 1.35.

```
data Final; set Final;
   wt_change=Cal_SWeight/SWeight;
proc surveymeans data = Final min max quartiles;
   var wt_change;
run;
```

Figure 3 Calibration Adjustment to **SWeight**

The SURVEYMEANS Procedure

Quantiles					
Variable	**Percentile**	**Estimate**	**Std Error**	**95% Confidence Limits**	
wt_change	0 Min	0.352211	.	.	.
	25 Q1	0.352211	0.002337	0.34759793	0.35682395
	50 Median	0.362100	0.170907	0.02476826	0.69943158
	75 Q3	1.036215	0.079975	0.87836266	1.19406728
	100 Max	1.351918	.	.	.

Figure 4 and Figure 5 show the analysis of the variables **VitalStatus** and **Age** by using the original sampling weights **SWeight** and the calibrated weights **Cal_SWeight**. The point estimates changed slightly for this example because of the calibration adjustment.

Figure 4 Analysis Using Original Sampling Weights **SWeight**

The SURVEYMEANS Procedure

Statistics			
Variable	**Level**	**Mean**	**Std Error of Mean**
Age		65.073909	0.949498
VitalStatus	**1**	0.644459	0.034795
	3	0.267700	0.028865
	4	0.034766	0.011432
	5	0.016649	0.012291
	6	0.036426	0.028146

Figure 5 Analysis Using Calibrated Weights **Cal_SWeight**

The SURVEYMEANS Procedure

Statistics			
Variable	**Level**	**Mean**	**Std Error of Mean**
Age		65.126584	1.155297
VitalStatus	**1**	0.659089	0.036309
	3	0.270262	0.029592
	4	0.026019	0.016890
	5	0.012743	0.013272
	6	0.031887	0.028144

SUMMARY

The %SurveyCalibrate macro provides a convenient and flexible tool for constructing calibration-adjusted survey weights by using the most common calibration methods. It also produces calibration-adjusted replicate weights for future analysis.

There is no magic rule for determining which calibration method should be used, but the following points might be helpful:

- The linear method obviously uses the fewest computing resources and always yields a solution. Therefore, it is the first choice for the macro to use if you do not specify a calibration method. However, it can sometimes yield negative calibration weights for some data.
- Thus, to always get positive weights, the second choice is the exponential method. This is the method that the macro switches to if you do not specify a particular method and if the linear method produces negative weights. However, the exponential method can sometimes result in extremely large weights.
- The truncated linear method and truncated exponential (logit) method serve as compromises for the linear and exponential methods. Properly set lower and upper bounds can ensure positive calibration weights that avoid extreme large weights. However, these methods are computationally intensive and sometimes find no solution. If your data are large, you might want to experiment and set your own lower and upper bounds first to reduce the computation time.

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* 11:427–444.

Deville, J.-C., and Särndal, C.-E. (1992). "Calibration Estimators in Survey Sampling." *Journal of the American Statistical Association* 87:376–382.

Deville, J.-C., Särndal, C.-E., and Sautory, O. (1993). "Generalized Raking Procedures in Survey Sampling." *Journal of the American Statistical Association* 88:1013–1020.

Haziza, D., and Beaumont, J.-F. (2017). "Construction of Weights in Surveys: A Review." *Statistical Science* 32:206–226.

Kennedy, C., Blumenthal, M., Clement, S., Clinton, J. D., Durand, C., Franklin, C., McGeeney, K., et al. (2018). "An Evaluation of the 2016 Election Polls in the United States." *Public Opinion Quarterly* 82:1–33.

SAS Institute Inc. (2018). *SAS/STAT 15.1 User's Guide*. Cary, NC: SAS Institute Inc. http://go.documentation.sas.com/?docsetId=statug&docsetTarget=titlepage.htm&docsetVersion=15.1&locale=en.

ACKNOWLEDGMENTS

The author would like to express his gratitude to Dr. David Haziza, Dr. Randy Tobias, Dr. Pushpal Mukhopadhyay, Dr. Donna Watts, and Ed Huddleston for their valuable assistance in the preparation of this manuscript.

CONTACT INFORMATION

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

Tony An
tony.an@sas.com