# A SAS® Macro for Calibration of Survey Weights

Tony An, PhD

SAS Institute Inc.

# %SurveyCalibrate: A Supplement to the SAS Survey PROCs

- PROC SURVEYSELECT
- PROC SURVEYIMPUTE
- PROC SURVEYMEANS
- PROC SURVEYFREQ
- PROC SURVEYREG
- PROC SURVEYLOGISTIC
- PROC SURVEYPHREG

Requires SAS/STAT and SAS/IML to run the macro

# Outline

- Introduction
- Calibration methods
- %SurveyCalibrate macro
- Example
- Discussion

# Sampling Weights

- Reduce bias
- Reflect sample design
- Adjust for nonresponses
- Estimate the variance

# Weight Adjustments

- Nonresponse adjustment to reduce bias
- Weight trimming and smoothing
- Calibration to match the known population totals for some auxiliary variables (such as demographic variables)

# Example

- Restaurant utility usage
- Franchise or independent
- Known number of restaurants in each category

# Weighted Sum of Restaurants

| Restaurant Type | Sum in Sample | Known Totals |
|---|---|---|
| Franchise | 231 | 251 |
| Independent | 267 | 210 |

# Remedy - Calibration

- Adjust the weights -> calibration weights
- Calibration weights are as “close” as possible to the original weights
- Estimates over control variables with calibration weights match known quantities

# Calibration Method

- Define a distance function *G* to measure the "closeness" between two sets of weights
- Find a solution that minimizes *G* under the *constraints* – matching known population totals *T* for a set of controls variables *X*

# Calibration Method

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

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

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

# Common Calibration Methods

- Linear
- Exponential (raking)
- Truncated linear
- Logit (truncated exponential)

# Linear Method

Distance function:

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

Calibration weights:

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

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

# Linear Methods

- A solution always exists
- Might have negative calibration weights

# Exponential Method

Distance function:

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

Calibration weights:

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

# Exponential Method

- Calibration weights always positive
- Might produce extremely large weights
- Solution might not exist
- Equivalent to the raking method when there are two categorical control variables

# Truncated Linear Method

Distance function:

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

Calibration weights:

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

# Truncated Linear Methods

- Weights are bounded
- Solution might not exist
- Computation requires more resources
- The macro can choose the lower, upper, or both bounds for you

# LOGIT Method

Distance function:

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

Calibration weights:

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

# LOGIT Methods

- It’s also called the truncated exponential method
- Weights are bounded
- Solution might not exist
- Computation requires more resources

# %SurveyCalibrate Macro

```
%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                                   */
);
```

# Macro Parameters

- Calibration parameters
- Replication parameters
- Some parameters can be left blank

# Required Calibration Parameters

```
%SurveyCalibrate(
  DATA= /* Input data set name */,
  OUT= /* Output data set name */,
  WEIGHT=/* Original weight variable */,
  CONTROLVAR=/* Auxiliary control variables for calibration */,
  CTRLTOTAL= /* Marginal totals for CONTROLVAR */
)
```

# Specifying Controls

CONTROLVAR=/* Auxiliary control variables for calibration */

- Control variables can be either continuous or categorical variables
- For categorical variables, you need to create indicator variables with data step before calling the macro

# Optional Calibration Parameters

METHOD= /* LINEAR | EXPONENTIAL | TRUNLINEAR | LOGIT */

CALWT= /* Calibration weight variable, default CalWt */

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 */

# Calibration for Replicates

- Use the same calibration method for each replicate
- Ensure the correct variance estimation after the calibration
- Skip if no variance estimation needed

# 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 */

# Example

- National Health and Nutrition Examination Survey I (NHANES I) Epidemiologic Followup Study (NHEFS)
- 174 observations from the 1992 NHEFS vital and tracing status data set

# Variables in the Data Set

- 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
- VitalStatus, vital status of subject in 1992 contact
- PovArInd, indicator subject's household location in terms of poverty area (1 = poverty area, 2 = nonpoverty area)
- Gender, gender of subject (1 = male, 2 = female)

# Known Totals vs. Reality

| PovArInd | Known Population | Sample Estimate | Gender | Known Population | Sample Estimate |
|---|---|---|---|---|---|
| Poverty | 536207 | 1507352 | Male | 3503378 | 3018151 |
| Nonpoverty | 6554845 | 5583700 | Female | 3587674 | 4072901 |

# Create 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;
```

# Calibration

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

# Macro Log Messages

- 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.

# Results

| PovArInd | Known Population | Sample Estimate | After Calibration | Gender | Known Population | Sample Estimate | After Calibration |
|---|---|---|---|---|---|---|---|
| Poverty | 536207 | 1507352 | 536207 | Male | 3503378 | 3018151 | 3503378 |
| Non-poverty | 6554845 | 5583700 | 6554845 | Female | 3587674 | 4072901 | 3587674 |

# Analysis

| | Using Original Weights | | Using Calibration Weights | |
|---|---|---|---|---|
| Variable | Mean | Std Error | Mean | Std Error |
| Age | 65.073909 | 0.949498 | 65.126584 | 1.155297 |
| VitalStatus=1 | 0.644459 | 0.034795 | 0.659089 | 0.036309 |
| VitalStatus=3 | 0.267700 | 0.028865 | 0.270262 | 0.029592 |
| VitalStatus=4 | 0.034766 | 0.011432 | 0.026019 | 0.016890 |
| VitalStatus=5 | 0.016649 | 0.012291 | 0.012743 | 0.013272 |
| VitalStatus=6 | 0.036426 | 0.028146 | 0.031887 | 0.028144 |

# Discussion

- No magic rule for choosing a calibration method
- Use the linear method first
- Try the exponential method if the linear method fails
- Compromise with truncated linear or logit methods at the price of computation resources
- Experiment with different settings

# Summary

- Flexible and easy to use
- Accommodates most calibration needs
- Provides replicate weights for future analyses
- Leave optional parameters blank if not sure what to use

# How to Get the Macro

Contact: Tony.An@sas.com

https://github.com/tony-an-sas/SASWeightCalibration