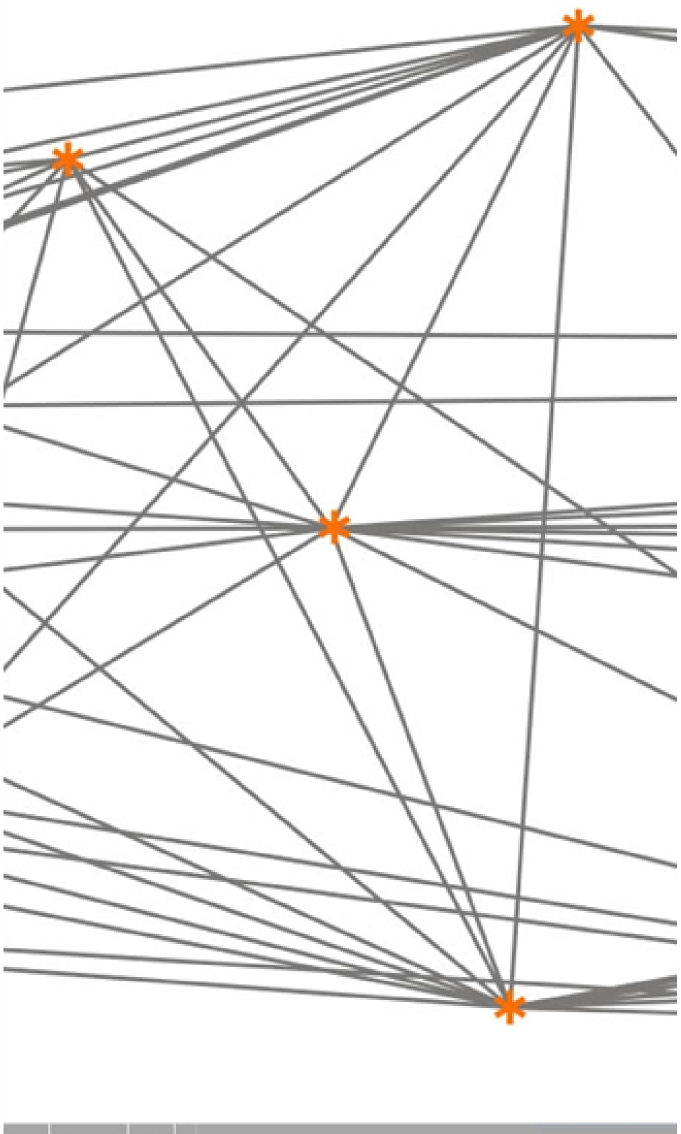
SHARE

Survey of Health, Ageing and Retirement in Europe

Computing Calibrated Weights



mea

March 29, 2018

Stata program to compute calibrated weights from scientific usefile and additional database

Giuseppe De Luca
University of Palermo, Italy

Claudio Rossetti
University of Naples Federico II, Italy

Abstract

This report provide a description of the Stata programs available to create calibrated weights from scientific usefile and additional database. After reviewing the key features of the calibration approach of Deville and Särndal (1992), we show how to compute calibrated cross-sectional weights in the first wave of the SHARE panel and calibrated longitudinal weights between the first two waves. Both types of weights are computed at the individual level for inference to the target population of individuals and at the household level for inference to the target population of households.

Contents

1 Introduction **2**

2 The calibration approach **2**

3 Computing calibrated weights in Stata **6**

3.1 My SHARE databases 6
3.2 Vectors of calibration margins 16
3.3 Calibrated cross-sectional individual weights 18
3.4 Calibrated cross-sectional household weights 28
3.5 Calibrated longitudinal individual weights 33
3.6 Calibrated longitudinal household weights 38

4 Conclusions **41**

References **42**

1 Introduction

In this user guide, we describe the Stata programs available to create calibrated weights from scientific usefile and additional database.

After reviewing the key features of the calibration approach introduced by Deville and Särndal (1992), we provide a variety of examples on the construction of calibrated weights based on data from the Survey on Health, Ageing and Retirement in Europe (SHARE). Here, calibration is used to compensate for both problems of unit nonresponse in the baseline and refreshment samples of each wave, and problems of attrition in the longitudinal samples of different waves. The cross-sectional and longitudinal dimensions of the SHARE data, as well as their multi-national nature, allow us to illustrate the great generality of the calibration approach. More precisely, we show how to compute calibrated cross-sectional weights in the first wave of the SHARE panel and calibrated longitudinal weights between the first two waves. Since the basic units of analysis can be either individuals or households, both types of weights are computed at the individual level for inference to the target population of individuals and at the household level for inference to the target population of households. In these examples, calibrated weights allow us to adjust the original design weights so that weighted survey estimates match the known population totals (the so-called calibration margins) for a given set of control variables. Calibration margins for the target populations investigated by SHARE are taken from the regional demographic statistics given by Eurostat. This external source of data contains population figures and number of deaths by year, NUTS1 regional area, gender and age. The rationale behind the calibration adjustment is that by ensuring consistency between the sample and the population distributions of these benchmark variables, the calibrated weights will also perform well when applied to other study variables of interest.

2 The calibration approach

The calibration approach of Deville and Särndal (1992) is a well-known and widely used procedure to adjust the survey weights so that the weighted sum of a vector of benchmark variables over the sample units equals the corresponding vector of known population totals.

Let $U = \{1, \dots, i, \dots, N\}$ be a finite population of N elements, from which a probability sample

$s = \{1, \ldots, i, \ldots, n\} \subseteq U$ of size $n \leq N$ is drawn according to the sampling design $p(\cdot)$. Unless otherwise specified, we shall assume that the inclusion probability $\pi_i = \Pr(i \in s)$ and the associated *design weights* $w_i = \pi_i^{-1}$ are known and strictly positive for all population units. The availability of the design weights w_i allows us to account for the randomness due to probability sampling. For example, if we wish to estimate the population total $t_y = \sum_{i \in U} y_i$ of a study variable y, then the Horvitz-Thompson estimator

$$\widehat{t}_y = \sum_{i \in s} w_i y_i \tag{1}$$

is known to be design unbiased, that is $\mathbb{E}_p(\widehat{t}_y) = t_y$, where $\mathbb{E}_p(\cdot)$ denotes the expectation with respect to the sampling design.

Next, we assume that additional information is available to construct a class of more efficient estimators. More precisely, let $x_i = (x_{i1}, \ldots, x_{iq})^\top$ be a q-vector of auxiliary (categorical) variables for which we known the corresponding vector of population totals $t_x = \sum_{i \in U} x_i$ from either the sampling frame or other external sources such as census data and administrative archives. We shall refer to the auxiliary variables x_i as calibration variables and to the population totals t_x as U-level calibration margins. The basic idea of the calibration approach is to determine a new set of *calibrated weights* w_i^* that are close as possible (in an average sense with respect to a given distance function) to the design weights w_i, while also satisfying the constraints

$$\sum_{i \in s} w_i^* x_i = t_x. \tag{2}$$

Thus, given a distance function $G(w_i^*, w_i)$, calibration consists of minimizing the aggregate distance $\sum_{i \in s} G(w_i^*, w_i)$ with respect to w_i^* subject to the q equality constraints in (2). Following Deville and Särndal (1992), we assume that the distance function $G(w_i^*, w_i)$ is nonnegative, differentiable with respect to w_i^*, strictly convex, and such that $G(w_i, w_i) = 0$. We also assume that its first partial derivative with respect to w_i^* can be written as

$$\frac{\partial G(w_i^*, w_i)}{\partial w_i^*} = g\left(\frac{w_i^*}{w_i}\right),$$

where $g(\cdot)$ is continuous and strictly increasing function of w_i^*/w_i, such that $g(1) = 0$ and $g'(1) = 1$.[1] Under these regularity conditions, the Lagrangian for this constrained optimization problem leads to the following set of first-order conditions

$$g\left(\frac{w_i^*}{w_i}\right) - \eta_i = 0, \qquad i = 1, \ldots, n,$$

[1] Exact regularity conditions for the specification of the distance function $G(w_i^*, w_i)$ and its first partial derivative can be found in Deville and Särndal (1992, p. 377)

where $\eta_i = x_i^\top \lambda$ is a linear combination of the calibration variables x_i and $\lambda = (\lambda_1, \ldots, \lambda_q)^\top$ is a q-vector of Lagrangian multipliers associated with the constraints (2). If a solution exists, then it is unique and is given by

$$w_i^* = w_i F(\eta_i), \qquad i = 1, \ldots, n, \tag{3}$$

where $F(u) = g^{-1}(u)$ denote the inverse function of $g(\cdot)$. Substituting (3) in (2) gives

$$t_x = \sum_{i \in s} w_i F(x_i^\top \lambda) x_i,$$

so that the vector of Lagrange multipliers can be obtained by minimizing the function

$$f(\lambda) = t_x - \hat{t}_x = \sum_{i \in s} w_i (F(x_i^\top \lambda) - 1) x_i, \tag{4}$$

where $\hat{t}_x = \sum_{i \in s} w_i x_i$ is the sample estimate of t_x based on the sampling design weights w_i. Give a solution for λ, we can find the calibrated weights w_i^* directly from (3).

A distinguishing feature of this approach is that many traditional re-weighting procedure such as post-stratification, raking, and generalized linear regression (GREG) can be viewed as special cases of the calibration estimator

$$\widehat{t}_y^* = \sum_{i \in s} w_i^* y_i, \tag{5}$$

for particular choices of the calibration function $F(\cdot)$ (or, equivalently, the distance function $G(\cdot, \cdot)$) and the vector of calibration variables x_i. Popular specifications of the distance function $G(w_i^*, w_i)$ and the associated functions $g(w_i^*/w_i)$ and $F(\eta_i)$ are listed in Table 1 of Deville and Särndal (1992). The chi-quare distance function $G(w_i^*, w_i) = (w_i^* - w_i)^2/2w_i$, which leads to the widely used GREG estimator, has the advantage of ensuring a closed form solution for the calibrated weights w_i^*. The main drawback is that, depending on the chosen set of calibration variables, the resulting weights can be negative or extremely large because this distance function is unbounded. Other specifications of the calibration function allow us to avoid these issues, but a solution for the calibration problem may not exist and the associated Lagrangian multipliers must be obtained by some iterative procedure. For example, the distance function $G(w_i^*, w_i) = w_i^*(\ln(w_i^*/w_i) - 1) - w_i$ and the associated calibration function $F(\eta_i) = \exp(\eta_i)$ still ensure that a solution for the calibrated weights w_i^* exists. However, this distance function is still subject to the issue of large variability of the calibrated weights which in turn may affect the precision of the calibration estimator $\widehat{t}_y^*$. The logit specification

$$G(w_i^*, w_i; u, l) \propto \left(\frac{w_i^*}{w_i} - l \right) \ln \left(\frac{w_i^*/w_i - l}{1 - l} \right) + \left(u - \frac{w_i^*}{w_i} \right) \ln \left(\frac{u - w_i^*/w_i}{u - 1} \right)$$

which leads to a calibrated function of the form

$$F(\eta_i; u, l) = \frac{l(u-1) + u(1-l)\exp(a\eta_i)}{u - 1 + (1-l)\exp(a\eta_i)},$$

with $a = ((1-l)(m-1))^{-1}(u-l)$, is usually preferred to the other specifications because it allows to restrict in advance the range of feasible values for the calibrated weights by suitable choices of the lower bound l and the upper bound u. More precisely, if a solution exists, then this distance function ensures that $lw_i \leq w_i^* \leq uw_i$.

As argued by Deville and Särndal (1992), effectiveness of the calibrated weights depends crucially on the correlation between the study variable y and the calibration variables x. In the extreme case when y can be expressed as a linear combination of x the calibrated estimator $\hat{t}_y^*$ gives an exact estimate of t_y for every realized sample s. They also show that, under suitable regularity conditions, the calibration estimator $\hat{t}_y^*$ has desirable asymptotic properties. Moreover, the calibration estimators resulting from alternative specifications of the distance function are asymptotically equivalent to the GREG estimator resulting from the chi-squared specification. Thus, in large samples, calibrated weights are robust to arbitrary choices of the functional form for $F(\cdot)$.

Unfortunately, this robustness property does not necessarily extend to the more realistic setting where survey data are affected by nonresponse errors. The calibration approach can be easily applied to the complete-case data $\{(y_i, x_i, w_i) : i \in s_r\}$, where $s_r \subseteq s$ is a subsample of $n_r \leq n$ units who agree to participate into the survey. However, the statistical properties of the nonresponse calibration estimator can be substantially different from those achieved in the complete response setting because of the additional randomness due to the nonresponse mechanism. Lundström and Särndal (1999) give expressions for the bias, the variance and the MSE of the GREG estimator which is a special case of the nonresponse calibration estimator when $F(\cdot)$ has a linear specification. A more general expression for the bias of the nonresponse calibrated estimator can be found in Haziza and Lesage (2016). These studies suggests that, unlike the complete response setting, alternative specifications of the calibration function $F(\cdot)$ correspond in practice to different parametric models for the relationship between the response propensity and the calibration variables. In other words, assumptions about the nonresponse mechanism are implicit in the specification of the calibration function $F(\cdot)$ and the misspecification of this functional form may now lead to biased nonresponse calibrated estimators. A more explicit approach to deal with nonresponse errors can be based on two-step procedure which involves a propensity-score adjustment of sampling design weights in the first step and a standard calibration adjustment of the propensity-score adjusted weights in the second step (see, e.g., Lundström and Särndal 1999; Särndal and Lundström 2005; Brick

2013; Haziza and Lesage 2016). Notice that, if $\widehat{w}_i$ and $\widehat{w}_i^*$ denote, respectively, the propensity-score adjusted weights and the nonresponse calibrated weights computed in this two-step procedure, then the calibration equations in the second step can be based on either a set of U-level calibration margins

$$\sum_{i \in s_r} \widehat{w}_i^* x_i = t_x \tag{6}$$

or a set of S-level calibration margins

$$\sum_{i \in s_r} \widehat{w}_i^* x_i = \sum_{i \in s} \widehat{w}_i x_i. \tag{7}$$

The latter set of constraints requires that the auxiliary variables x_i are observed for all sample units, but it allows us to replace the possibly lacking information on the population totals t_x with an unbiased sample estimate. A review of the available approaches for estimating the propensity-score adjusted weights $\widehat{w}_i$ can be found in Brick (2013).

3 Computing calibrated weights in Stata

This section focuses on the computation of calibrated weights in Stata by the `sreweight` command (Pacifico 2014). We provide a variety of examples based on data from release 6.0.0 of SHARE, a unique longitudinal survey with rich micro-level information on people aged 50 and older in 20 European countries and Israel. Although five waves of SHARE are currently available, we restrict attention to the calibrated cross-sectional weights in the first wave of the SHARE panel and to the calibrated longitudinal weights between the first two waves. Both types of weights are computed at the individual level for inference to the target population of individuals and at the household level for inference to the target population of households.

3.1 My SHARE databases

SHARE data can be used for a variety of cross-sectional and longitudinal studies. For the purposes of cross-sectional studies, the target population in each country consists of persons of 50 years or older at a particular point in time and their possibly younger spouses/partners, who speak (one of) the official language(s) of the country (regardless of nationality and citizenship) and who do not live either abroad or in institutions such as prisons and hospitals during the entire fieldwork period (Bergmann et al. 2017). For longitudinal studies, the target population is typically defined as the cross-sectional target population at the beginning of a time reference period that survives up to the end of the period considered (see, e.g., Lynn, 2009). These target populations could also be defined in terms of households as all households with at least one member belonging to the cross-sectional/longitudinal target population of individuals.

In this section we show how to extract cross-sectional and longitudinal samples of individuals and households for representing these alternative variants of the SHARE target population. Below, we consider the sample of individuals interviewed in the first wave.

```
. *---------------------------------------------------------------------------
. * Extract individual data from SHARE wave 1
. *---------------------------------------------------------------------------
. local SHARE_w1 "C:\DATA\SHARE\sharew1\Release 6.0.0"
.
. qui use "`SHARE_w1´\sharew1_rel6-0-0_cv_r", clear
. keep mergeid hhid1 country gender yrbirth interview
. qui gen age_w1=2004-yrbirth if yrbirth>0
. qui keep if interview==1
. drop interview
. qui merge mergeid using "`SHARE_w1´\sharew1_rel6-0-0_gv_weights",       ///
>         keep(dw_w1 cciw_w1) sort
. assert _merge==3
. drop _merge
. qui merge mergeid using "`SHARE_w1´\sharew1_rel6-0-0_gv_housing",       ///
>         keep(nuts1_2003) sort
. assert _merge==3
. drop _merge
. sort mergeid
. qui saveold mydata_w1_ind, replace
.
. tab country

    Country |
 identifier |      Freq.     Percent        Cum.
------------+-----------------------------------
    Austria |      1,569        5.16        5.16
    Germany |      2,997        9.85       15.00
     Sweden |      3,049       10.02       25.02
Netherlands |      2,968        9.75       34.77
      Spain |      2,316        7.61       42.38
      Italy |      2,553        8.39       50.77
     France |      3,122       10.26       61.03
    Denmark |      1,706        5.61       66.64
     Greece |      2,897        9.52       76.15
Switzerland |        997        3.28       79.43
    Belgium |      3,810       12.52       91.95
     Israel |      2,450        8.05      100.00
------------+-----------------------------------
      Total |     30,434      100.00

.
```

```
. describe

Contains data from mydata_w1_ind.dta
  obs:        30,434
 vars:            10                          27 Dec 2017 12:10
 size:     2,495,588                          (_dta has notes)
----------------------------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
----------------------------------------------------------------------------------------------------
mergeid         str12   %12s                  Person identifier (fix across modules and waves)
hhid1           str11   %11s                  Household identifier (wave 1)
country         byte    %14.0g     country    Country identifier
gender          byte    %10.0f     gender     Male or female
yrbirth         int     %10.0f     dkrf       Year of birth
age_w1          float   %9.0g
dw_w1           double  %10.0g                Design weight - wave 1
cciw_w1         double  %10.0g                Calibrated cross-sectional individual weight - wave 1
nuts1_2003      str27   %27s                  NUTS level 1: nomenclature of territorial units for
                                                statistics
----------------------------------------------------------------------------------------------------
Sorted by: mergeid

.
. list mergeid hhid1 country gender yrbirth in 1/5, sepby(hhid1) noobs

  +----------------------------------------------------------+
  |      mergeid         hhid1   country   gender   yrbirth  |
  |----------------------------------------------------------|
  | AT-000327-01   AT-000327-A   Austria     Male      1952  |
  | AT-000327-02   AT-000327-A   Austria   Female      1955  |
  |----------------------------------------------------------|
  | AT-001816-01   AT-001816-A   Austria   Female      1943  |
  | AT-001816-02   AT-001816-A   Austria     Male      1948  |
  |----------------------------------------------------------|
  | AT-002132-01   AT-002132-A   Austria   Female      1933  |
  +----------------------------------------------------------+

. *--------------------------------------------------------------------------------
```

In total, the release 6.0.0 of the wave 1 data includes 30,434 respondents, with national samples ranging from a minimum size of 997 observations in Switzerland and a maximum size of 3,810 observations in Belgium. The database `mydata_w1_ind` contains information on the individual and household identifiers, the country indicator, basic demographic characteristics of the respondents (gender, year of birth, age at the time of the wave 1 interview, and NUTS level 1), the design weights, and the calibrated cross-sectional individual weights. In Section 3.3, we shall illustrate how to reproduce the calibrated weights `cciw_w1`.

In addition to individual level information, SHARE collects a variety of household level data about consumption, income and wealth. For this type of studies, we are often interested in constructing a sample of households. Our Stata code to extract household level data is similar before, but some attention is needed when reshaping the individual level data to select one observation for each household.

```
. *---------------------------------------------------------------------------
. * Extract household data from SHARE wave 1
. *---------------------------------------------------------------------------
. local SHARE_w1 "C:\DATA\SHARE\sharew1\Release 6.0.0"
.
. qui use "`SHARE_w1´\sharew1_rel6-0-0_cv_r", clear
. keep mergeid hhid1 country gender yrbirth interview
. qui gen age_w1=2004-yrbirth if yrbirth>0
. sort mergeid
. bys hhid1: gen member=_n
. drop mergeid
. rename gender gender_
. rename yrbirth yrbirth_
. rename age_w1 age_w1_
. rename interview interview_
. reshape wide gender_ yrbirth_ age_ interview_, i(hhid1) j(member)
(note: j = 1 2 3 4 5 6 7 8 9 10)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                    43993   ->   20809
Number of variables                   7   ->      42
j variable (10 values)           member   ->   (dropped)
xij variables:
                                gender_   ->   gender_1 gender_2 ... gender_10
                               yrbirth_   ->   yrbirth_1 yrbirth_2 ... yrbirth_10
                                age_w1_   ->   age_w1_1 age_w1_2 ... age_w1_10
                             interview_   ->   interview_1 interview_2 ... interview_10
-----------------------------------------------------------------------------
. qui compress
.
. sort hhid1
. qui merge hhid1 using "`SHARE_w1´\sharew1_rel6-0-0_gv_weights",         ///
>        keep(dw_w1 cchw_w1)
. assert _merge==3
. qui bys hhid1: keep if _n==1
. drop _merge
.
. sort hhid1
. qui merge hhid1 using "`SHARE_w1´\sharew1_rel6-0-0_gv_housing",         ///
>        keep(nuts1_2003)
. assert _merge==3
. qui bys hhid1: keep if _n==1
. drop _merge
.
. sort hhid1
. qui saveold mydata_w1_hhs, replace
.
```

```
. tab country

     Country │
  identifier │      Freq.     Percent        Cum.
─────────────┼───────────────────────────────────
     Austria │      1,173        5.64        5.64
     Germany │      1,993        9.58       15.21
      Sweden │      2,137       10.27       25.48
 Netherlands │      1,946        9.35       34.84
       Spain │      1,686        8.10       42.94
       Italy │      1,772        8.52       51.45
      France │      2,053        9.87       61.32
     Denmark │      1,175        5.65       66.97
      Greece │      1,981        9.52       76.49
 Switzerland │        706        3.39       79.88
     Belgium │      2,519       12.11       91.98
      Israel │      1,668        8.02      100.00
─────────────┼───────────────────────────────────
       Total │     20,809      100.00
. describe

Contains data from mydata_w1_hhs.dta
  obs:        20,809
 vars:            45                          29 Dec 2017 00:49
 size:     2,226,563                          (_dta has notes)
────────────────────────────────────────────────────────────────────────────────────────────────────

              storage   display    value
variable name   type    format     label      variable label
────────────────────────────────────────────────────────────────────────────────────────────────────

hhid1           str11   %11s                  Household identifier (wave 1)
gender_1        byte    %10.0f     gender     1 gender_
yrbirth_1       int     %10.0f     dkrf       1 yrbirth_
interview_1     byte    %21.0g     interview
                                              1 interview_
age_w1_1        int     %9.0g                 1 age_w1_
  (output omitted)
gender_10       byte    %10.0f     gender     10 gender_
yrbirth_10      int     %10.0f     dkrf       10 yrbirth_
interview_10    byte    %21.0g     interview
                                              10 interview_
age_w1_10       byte    %9.0g                 10 age_w1_
country         byte    %14.0g     country    Country identifier
dw_w1           double  %10.0g                Design weight - wave 1
cchw_w1         double  %10.0g                Calibrated cross-sectional household weight - wave 1
nuts1_2003      str27   %27s                  NUTS level 1: nomenclature of territorial units for
                                                statistics
────────────────────────────────────────────────────────────────────────────────────────────────────
Sorted by: hhid1
. *--------------------------------------------------------------------------
```

The database `mydata_w1_hhs` consists of 20,809 households, with national samples ranging from a minimum size of 706 observations in Switzerland and a maximum size of 2,519 observations in Belgium. Notice that, for the purpose of reproducing the calibrated cross-sectional household weights `cchw_w1`, we have stored the information about gender, year of birth, age and interview status of all household members in a wide format. The construction of this type of calibrated weights is discussed in Section 3.4.

Our next example concerns the balanced sample of individuals interviewed in both the first and second waves useful for longitudinal studies. Our Stata code is similar to the one for extracting the cross-sectional sample of individuals, but now we must pay specific attention to correctly selecting the longitudinal sample of respondents that were present in both waves (balanced sample). The underlying code is as follows:

```
. *---------------------------------------------------------------------------
. * Extract individual data for longitudinal sample
. *---------------------------------------------------------------------------
. local SHARE "C:\DATA\SHARE"
. local wi = 1 // initial wave //
. local wf = 2 // final wave //
. *---------------------------------------------------------------------------
. global w = "l_`wi´_`wf´"
. * Identify longitudinal (balanced) sample
. forvalues wj=`wi´(1)`wf´ {
  2.         qui use "data/sharew`wj´_rel6-0-0_cv_r", clear
  3.         qui keep if interview==1
  4.         qui keep mergeid
  5.         if `wj´>`wi´ qui merge 1:1 mergeid using temp_balanced_ii,keep(3) nogen
  6.         sort mergeid
  7.         qui save temp_balanced_ii,replace
  8. }
. * Merge longitudinal sample
. qui use "data/sharew`wi´_rel6-0-0_cv_r", clear
. keep mergeid hhid1 country gender yrbirth interview
. qui gen age_w1 = 2004 - yrbirth if yrbirth>0
. qui keep if interview==1
. drop interview
. qui merge 1:1 mergeid using "data/sharewX_rel6-0-0_gv_longitudinal_weights_w1w2",       ///
>        keepus(dw_w1 cliw_b) assert(1 3) nogen
. qui merge 1:1 mergeid using "data/sharew`wi´_rel6-0-0_gv_housing",                      ///
>        keepus(nuts1_2003) assert(3) nogen
. qui merge 1:1 mergeid using temp_balanced_ii
. gen balanced = _merge==3
. cap lab drop balanced
. lab define balanced 0 "not all waves" 1 "all waves"
. lab value balanced balanced
. tab country balanced,m
     Country |       balanced
  identifier | not all w  all waves |     Total
-------------+----------------------+----------
     Austria |       437      1,132 |     1,569
     Germany |     1,398      1,599 |     2,997
      Sweden |       959      2,090 |     3,049
 Netherlands |     1,167      1,801 |     2,968
       Spain |       824      1,492 |     2,316
       Italy |       782      1,771 |     2,553
      France |     1,090      2,032 |     3,122
     Denmark |       443      1,263 |     1,706
      Greece |       447      2,450 |     2,897
 Switzerland |       267        730 |       997
     Belgium |       942      2,868 |     3,810
      Israel |       762      1,688 |     2,450
-------------+----------------------+----------
       Total |     9,518     20,916 |    30,434
. drop if balanced==0
(9,518 observations deleted)
. drop _merge balanced
. * Save data
. qui compress
. sort mergeid
. qui saveold mydata_long_ind, replace
.
. describe
Contains data from mydata_long_ind.dta
  obs:        20,916
 vars:             9                          12 Jan 2018 12:55
 size:     1,505,952                          (_dta has notes)
----------------------------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
----------------------------------------------------------------------------------------------------
mergeid         str12   %12s                  Person identifier (fix across modules and waves)
hhid1           str11   %11s                  Household identifier (wave 1)
country         byte    %14.0g     country    Country identifier
gender          byte    %10.0f     gender     Male or female
yrbirth         int     %10.0f     dkrf       Year of birth
age_w1          int     %9.0g
dw_w1           double  %10.0g                Design weight - wave 1
cliw_b          double  %10.0g                Calibrated longitudinal individual weight - panel: 1_2
nuts1_2003      str27   %27s                  NUTS level 1: nomenclature of territorial units for
                                                statistics
----------------------------------------------------------------------------------------------------
Sorted by: mergeid
. list mergeid hhid1 country gender yrbirth in 1/5, sepby(hhid1) noobs

  +---------------------------------------------------------+
  |      mergeid         hhid1   country   gender   yrbirth |
  |---------------------------------------------------------|
  | AT-000327-01   AT-000327-A   Austria     Male      1952 |
  | AT-000327-02   AT-000327-A   Austria   Female      1955 |
  |---------------------------------------------------------|
  | AT-001816-02   AT-001816-A   Austria     Male      1948 |
  |---------------------------------------------------------|
  | AT-002132-01   AT-002132-A   Austria   Female      1933 |
  |---------------------------------------------------------|
  | AT-004234-01   AT-004234-A   Austria     Male      1950 |
  +---------------------------------------------------------+

. *---------------------------------------------------------------------------
```

The release 6.0.0 of the SHARE data includes 20,916 respondents that were present in the first two waves, with national samples ranging from a minimum size of 730 observations in Switzerland and a maximum size of 2,868 observations in Belgium. In total, there are 9,518 respondents interviewed in wave 1 but not in wave 2 that have been dropped from our longitudinal sample. The database `mydata_long_ind` contains information on the individual and household identifiers, the country indicator, basic demographic characteristics of the respondents (gender, year of birth, age at the time of the wave 1 interview, and NUTS level 1), the design weights in wave 1, and the calibrated longitudinal individual weights for the balanced panel from wave 1 to wave 2. In Section 3.5, we shall illustrate how to reproduce the calibrated weights `cliw_b`.

Like we mentioned above, we are often interested in extracting a sample of households. This is also true for studies involving longitudinal samples. Thus, the sample of interest could also be defined in terms of balanced panel of households. Specifically, in this case we are interested in all households with at least one member belonging to the longitudinal sample of individuals. Our Stata code to extract household level data is similar to the previous one, but again we must pay special attention to correctly selecting the longitudinal sample of households with at least one member present both in wave 1 and in wave 2 (balanced sample). Furthermore, as in the cross-sectional case, some attention is needed to select one observation for each household.

```
. *----------------------------------------------------------------------------
. * Extract household data for longitudinal sample (e.g. from wave 1 to wave 2)
. *----------------------------------------------------------------------------
. local SHARE "C:\DATA\SHARE"
. local wi = 1 // initial wave //
. local wf = 2 // final wave   //
. global w = "l_`wi´_`wf´"
.
. * Identify longitudinal (balanced) sample
. forvalues wj=`wi´(1)`wf´ {
  2.         qui use "data/sharew`wj´_rel6-0-0_cv_r", clear
  3.         gen str9 hhmergeid = mergeid
  4.         keep hhmergeid
  5.         bys hhmergeid:keep if _n==1
  6.         if `wj´>`wi´ qui merge 1:1 hhmergeid using temp_balanced_hh,keep(3) nogen
  7.         sort hhmergeid
  8.         qui save temp_balanced_hh,replace
  9. }
. * Merge longitudinal sample
. use "data/sharewX_rel6-0-0_gv_longitudinal_weights_w1w2",replace
. gen str9 hhmergeid = mergeid
. qui bys hhmergeid: keep if _n==1
. keep hhmergeid dw_w1 clhw_b
. qui save temp_long,replace
.
. qui use "data/sharew`wi´_rel6-0-0_cv_r", clear
. gen str9 hhmergeid = mergeid
. keep hhmergeid hhid1 country gender yrbirth interview
. qui gen age_w1 = 2004 - yrbirth if yrbirth>0
. bys hhid1: gen member = _n
. rename gender gender_
. rename yrbirth yrbirth_
. rename age_w1 age_w1_
. rename interview interview_
. reshape wide gender_ yrbirth_ age_ interview_, i(hhid1) j(member)
(note: j = 1 2 3 4 5 6 7 8 9 10)
Data                               long   ->   wide
-----------------------------------------------------------------------------
Number of obs.                    43993   ->   20809
Number of variables                   8   ->      43
j variable (10 values)           member   ->   (dropped)
xij variables:
                                gender_   ->   gender_1 gender_2 ... gender_10
                               yrbirth_   ->   yrbirth_1 yrbirth_2 ... yrbirth_10
                                age_w1_   ->   age_w1_1 age_w1_2 ... age_w1_10
                             interview_   ->   interview_1 interview_2 ... interview_10
-----------------------------------------------------------------------------
```

```
. qui merge 1:1 hhmergeid using temp_long,                              ///
>         keepus(dw_w1 clhw_b) assert(1 3) nogen

. qui merge 1:m hhid1 using "data/sharew`wi´_rel6-0-0_gv_housing",       ///
>         keepus(nuts1_2003) assert(3) nogen

. qui bys hhid1: keep if _n==1

. qui merge 1:1 hhmergeid using temp_balanced_hh,assert(1 3)

. gen balanced = _merge==3

. cap lab drop balanced

. lab define balanced 0 "not all waves" 1 "all waves"

. lab value balanced balanced

. noi tab country balanced,m

    Country |       balanced
 identifier | not all w  all waves |     Total
------------+----------------------+----------
    Austria |       273        900 |     1,173 
    Germany |       874      1,119 |     1,993 
     Sweden |       517      1,620 |     2,137 
Netherlands |       606      1,340 |     1,946 
      Spain |       467      1,219 |     1,686 
      Italy |       505      1,267 |     1,772 
     France |       601      1,452 |     2,053 
    Denmark |       238        937 |     1,175 
     Greece |       252      1,729 |     1,981 
Switzerland |       153        553 |       706 
    Belgium |       519      2,000 |     2,519 
     Israel |       353      1,315 |     1,668 
------------+----------------------+----------
      Total |     5,358     15,451 |    20,809 

. drop if balanced==0
(5,358 observations deleted)

. drop _merge balanced

. * Save data
. qui compress

. sort hhid1

. qui saveold mydata_long_hhs, replace

.
. describe

Contains data from mydata_long_hhs.dta
  obs:        15,451                          
 vars:            46                          12 Jan 2018 12:55
 size:     1,792,316                          (_dta has notes)
----------------------------------------------------------------------------------------------------
              storage   display    value
variable name   type    format     label      variable label
----------------------------------------------------------------------------------------------------
hhid1           str11   %11s                  Household identifier (wave 1)
gender_1        byte    %10.0f     gender     1 gender_
yrbirth_1       int     %10.0f     dkrf       1 yrbirth_
interview_1     byte    %21.0g     interview
                                              1 interview_
age_w1_1        int     %9.0g                 1 age_w1_
gender_2        byte    %10.0f     gender     2 gender_
yrbirth_2       int     %10.0f     dkrf       2 yrbirth_
interview_2     byte    %21.0g     interview
                                              2 interview_
age_w1_2        byte    %9.0g                 2 age_w1_
gender_3        byte    %10.0f     gender     3 gender_
yrbirth_3       int     %10.0f     dkrf       3 yrbirth_
interview_3     byte    %21.0g     interview
                                              3 interview_
age_w1_3        int     %9.0g                 3 age_w1_
gender_4        byte    %10.0f     gender     4 gender_
```

```
yrbirth_4       int     %10.0f     dkrf       4 yrbirth_
interview_4     byte    %21.0g     interview
                                              4 interview_
age_w1_4        byte    %9.0g                 4 age_w1_
gender_5        byte    %10.0f     gender     5 gender_
yrbirth_5       int     %10.0f     dkrf       5 yrbirth_
interview_5     byte    %21.0g     interview
                                              5 interview_
age_w1_5        byte    %9.0g                 5 age_w1_
gender_6        byte    %10.0f     gender     6 gender_
yrbirth_6       int     %10.0f     dkrf       6 yrbirth_
interview_6     byte    %21.0g     interview
                                              6 interview_
age_w1_6        byte    %9.0g                 6 age_w1_
gender_7        byte    %10.0f     gender     7 gender_
yrbirth_7       int     %10.0f     dkrf       7 yrbirth_
interview_7     byte    %21.0g     interview
                                              7 interview_
age_w1_7        byte    %9.0g                 7 age_w1_
gender_8        byte    %10.0f     gender     8 gender_
yrbirth_8       int     %10.0f     dkrf       8 yrbirth_
interview_8     byte    %21.0g     interview
                                              8 interview_
age_w1_8        byte    %9.0g                 8 age_w1_
gender_9        byte    %10.0f     gender     9 gender_
yrbirth_9       int     %10.0f     dkrf       9 yrbirth_
interview_9     byte    %21.0g     interview
                                              9 interview_
age_w1_9        byte    %9.0g                 9 age_w1_
gender_10       byte    %10.0f     gender     10 gender_
yrbirth_10      int     %10.0f     dkrf       10 yrbirth_
interview_10    byte    %21.0g     interview
                                              10 interview_
age_w1_10       byte    %9.0g                 10 age_w1_
country         byte    %14.0g     country    Country identifier
hhmergeid       str9    %9s
dw_w1           double  %10.0g                Design weight - wave 1
clhw_b          double  %10.0g                Calibrated longitudinal household weight - panel: 1_2
nuts1_2003      str27   %27s                  NUTS level 1: nomenclature of territorial units for
                                                statistics
-----------------------------------------------------------------------------------------------
Sorted by: hhid1
. list hhid1 country gender_1 gender_2 yrbirth_1 yrbirth_2 in 1/3, noobs

  +------------------------------------------------------------------+
  |       hhid1   country   gender_1   gender_2   yrbirt~1   yrbirt~2 |
  |------------------------------------------------------------------|
  | AT-000327-A   Austria       Male     Female       1952       1955 |
  | AT-001816-A   Austria     Female       Male       1943       1948 |
  | AT-002132-A   Austria     Female       Male       1933       1966 |
  +------------------------------------------------------------------+

. *---------------------------------------------------------------------------
```

In total, the release 6.0.0 of the SHARE data includes 15,451 households with at least one member present both in wave 1 and in wave 2 (balanced sample), with national samples ranging from a minimum size of 553 observations in Switzerland and a maximum size of 2,000 observations in Belgium. The other 5,358 households of the 20,809 in wave 1 had no members present in wave 2. Thus, these households were not used in our longitudinal sample contained in the database `mydata_long_hhs`. Again, for the purpose of reproducing the calibrated longitudinal household weights for the balanced panel `clhw_b`, we have stored the information about gender, year of birth, age and interview status of all household members in a wide format. The construction of this type of calibrated weights is discussed in Section 3.6.

3.2 Vectors of calibration margins

In this section we show how to construct the vector of calibration margins used to compute calibrated weights. We exploit the database `margins_nuts1.dta`, which contains population figures and number of deaths by year, region, age and gender for all countries involved in the first six waves of SHARE. The data comes from the Central Bureau of Statistics for Israel, and from Eurostat for all other European countries. Regions for European countries are statistical regions at NUTS1 level Age is defined in most countries as single years from the age of 30 to 88, plus the open-ended class aged 89 or over. Finally, population and number of deaths are included separately for males and females. Detailed documentation of the database `margins_nuts1` is provided in "Deliverable 2.9".

Below we discuss the do-file `CalMar`. By setting macros properly, this file allows creating vectors of population margins for gender and age groups for a specific country. Population can refer either to population in a given year (used for cross-sectional weights), or to population in a given year that survives up to a certain later year (used for longitudinal weights). The latter is obtained from the do-file `CalMar` by subtracting from the population in the initial year the number of deaths in the following years up to the final one. Specifically, this do-file creates one scalar and three vectors. The first scalar contains, for the specific country, the target population related to the chosen age groups and years (for example population 50+ in 2004). The first two vectors contain, respectively, the total population by gender-age group (i.e. males and females in the chosen age groups) and of the population by NUTS1 regional area. Stacking together the gender-age group vector and the vector obtained by excluding the first component of the NUTS1 vector yields the vector of calibration margins used to construct the SHARE calibrated weights.

The macros needed to initialize the do-file `CalMar` are the country label (macro `cc`), the country number (macro `cc_num`), the reference year (macro `pop_time`), the final year (macro `mort_time`), the number of age groups (macro `age_groups`), and the lower (macro `age_thr_low`) and upper (macro `age_thr_upp`) thresholds of the age groups. The macro `mort_time` (final year) is only used for computing longitudinal weights, i.e. for population in reference year surviving in final year. For cross-sectional weights this macro must be set to zero. The macro `w` defines simply a label for the wave of interest.

```
. *-------------------------------------------------------------------------------------------
. * Set local macros
. *-------------------------------------------------------------------------------------------
. local cc           ${cc}              // country
. local cc_num       ${cc_num}          // country number //
. local pop_time     ${pop_time}        // reference year
. local mort_time    ${mort_time}       // final year
. local w            ${w}
. local age_groups   ${age_groups}      // number of age groups
. local age_thr_low  ${age_thr_low}     // lower thresholds of age groups
. local age_thr_upp  ${age_thr_upp}     // upper thresholds of age groups
. *---------------------------------------------------------------------------
. * Vector of calibration margins from margins_nuts1.dta
. *---------------------------------------------------------------------------
. qui use "margins_nuts1", clear
. qui keep if country=="`cc´"
.
. gen age_mort = age - (year - `pop_time´)
. local age_min: word `age_groups´ of `age_thr_low´
. local age_max: word 1            of `age_thr_upp´
. qui drop if age<`age_min´
. assert age>=`age_min´ & age<=`age_max´
.
. * Joint age-sex classification
. local rname ""
. local t=1
. matrix `cc´_w`w´_P=0
. cap matrix drop `cc´_w`w´_P_AGE_THR
. cap matrix drop `cc´_w`w´_P_SA
. forvalues ss=0(1)1 {
  2.   if `ss´==0 local slab "M"
  3.   if `ss´==1 local slab "F"
  4.   forvalues aa=1(1)`age_groups´ {
  5.     local age_upp: word `aa´ of `age_thr_upp´
  6.     local age_low: word `aa´ of `age_thr_low´
  7.     qui sum pop  if year==`pop_time´        ///
>                    & sex==`ss´               ///
>                    & (age>=`age_low´ & age<=`age_upp´)
  8.     local marg_`t´=r(sum)
  9.     qui sum deaths  if year>=`pop_time´     ///
>                    & year<`mort_time´         ///
>                    & sex==`ss´               ///
>                        & (age_mort>=`age_low´ & age_mort<=`age_upp´)
 10.     local marg_`t´=`marg_`t´´-r(sum)
 11.     assert `marg_`t´´>0
 12.     matrix `cc´_w`w´_P = `cc´_w`w´_P + `marg_`t´´
 13.     matrix `cc´_w`w´_P_SA = nullmat(`cc´_w`w´_P_SA) \ (`marg_`t´´)
 14.     if `aa´==1 local rname "`rname´ `slab´-`age_low´+"
 15.     else       local rname "`rname´ `slab´-`age_low´-`age_upp´"
 16.     if `ss´==0 matrix `cc´_w`w´_P_AGE_THR=nullmat(`cc´_w`w´_P_AGE_THR)\(`age_low´,`age_upp´)
 17.     local t=`t´+1
 18.   }
 19. }
. matrix coln `cc´_w`w´_P_SA      =POP
. matrix rown `cc´_w`w´_P_SA      =`rname´
. matrix coln `cc´_w`w´_P         =POP
. matrix rown `cc´_w`w´_P         =TOT
. matrix coln `cc´_w`w´_P_AGE_THR ="age_thr_low age_thr_upp"
. matrix list `cc´_w`w´_P
. matrix list `cc´_w`w´_P_SA
.
. * NUTS1 classification
. qui tab nuts1
. local nreg=r(r)
. if `nreg´>1 {
.   cap matrix drop `cc´_w`w´_P_NUTS1
.   local rname ""
.   local t=1
.   encode nuts1, gen(REG)
.   forvalue nn=1(1)`nreg´ {
  2.     local nn_lab: label REG `nn´
  3.     qui sum pop     if year==`pop_time´         ///
>                       & sex==2                     ///
>                       & nuts1=="`nn_lab´"
  4.     local marg_`t´=r(sum)
  5.     qui sum deaths  if year>=`pop_time´       ///
>                    & year<`mort_time´           ///
>                    & sex==2                     ///
>                    & (age_mort>=`age_min´)      ///
>                    & nuts1=="`nn_lab´"
  6.     local marg_`t´=`marg_`t´´-r(sum)
  7.     assert `marg_`t´´>0
  8.     matrix `cc´_w`w´_P_NUTS1 = nullmat(`cc´_w`w´_P_NUTS1)\(`marg_`t´´)
  9.     local rname "`rname´ `nn_lab´"
 10.     local t=`t´+1
 11.   }
.   matrix coln `cc´_w`w´_P_NUTS1 =POP
.   matrix rown `cc´_w`w´_P_NUTS1 =`rname´
.   matrix `cc´_w`w´_P_N          =`cc´_w`w´_P_NUTS1[2..`nreg´,1]
.   matrix `cc´_w`w´_P_MARG       =`cc´_w`w´_P_SA \ `cc´_w`w´_P_N
. }
. else {
.   matrix        `cc´_w`w´_P_NUTS1 =`cc´_w`w´_P
.   matrix        `cc´_w`w´_P_MARG  =`cc´_w`w´_P_SA
. }
. matrix list `cc´_w`w´_P_NUTS1
. matrix list `cc´_w`w´_P_MARG
. *---------------------------------------------------------------------------
```

3.3 Calibrated cross-sectional individual weights

In this section we show how to reproduce the calibrated cross-sectional individual weights (i.e. the variable `cciw_w1`) for a given country participating in first wave of SHARE. Without loss of generality we focus on Germany (country label DE - country number 12) by setting the following global macros

```
. *---------------------------------------------------------------------------
. * Select wave (e.g. w1) and country (e.g. DE)
. *---------------------------------------------------------------------------
. global cc "DE"                          // country label //
. global cc_num "12"                      // country number //
. global pop_time 2004                    // reference year //
. global mort_time 0                    // final year //
. global w "1"                          // initial wave //
. global age_groups 4                   // number of age groups
. global age_thr_low "80 70 60 50"      // lower thresholds of age groups
. global age_thr_upp "89 79 69 59"      // upper thresholds of age groups
. *---------------------------------------------------------------------------
```

The remainder of our code can be easily adapted to the other countries and waves by changing the values of these global macros. Next, we run the do-file `CalMar` to define the vector of calibration margins for the chosen wave-country combination.

```
. *---------------------------------------------------------------------------
. * Get local macros
. *-------------------------------------------------------------------------------------------------
. local cc            ${cc}             // country label //
. local cc_num        ${cc_num}         // country number //
. local w             ${w}
. *---------------------------------------------------------------------------
. * Run CalMar.do
. *---------------------------------------------------------------------------
. noi run CalMar.do
symmetric DE_w1_P[1,1]
          POP
TOT  30274231

DE_w1_P_SA[8,1]
             POP
  M-80+   946653
M-70-79  2680171
M-60-69  5051384
M-50-59  4962760
  F-80+  2501710
F-70-79  3769107
F-60-69  5387424
F-50-59  4975022

DE_w1_P_NUTS1[16,1]
         POP
DE1  3728737
DE2  4411850
DE3  1207394
DE4   969617
DE5   257730
DE6   620420
DE7  2222275
DE8   640323
DE9  2930725
DEA  6584773
DEB  1492760
DEC   413126
DED  1779562
DEE  1018621
DEF  1067047
DEG   929271

DE_w1_P_MARG[23,1]
             POP
  M-80+   946653
M-70-79  2680171
M-60-69  5051384
M-50-59  4962760
  F-80+  2501710
```

```
F-70-79  3769107
F-60-69  5387424
F-50-59  4975022
    DE2  4411850
    DE3  1207394
    DE4   969617
    DE5   257730
    DE6   620420
    DE7  2222275
    DE8   640323
    DE9  2930725
    DEA  6584773
    DEB  1492760
    DEC   413126
    DED  1779562
    DEE  1018621
    DEF  1067047
    DEG   929271
. *--------------------------------------------------------------------------------
```

As described in the previous section, this do-file creates one scalar and three vectors in the form of Stata matrices. The scalar `DE_w1_P` contains the German target population aged 50+ at the time of the wave 1 interview, while the vectors `DE_w1_P_SA` and `DE_w1_P_NUTS1` contain, respectively, a breakdown of the population by gender-age group (i.e. males and females in the age groups $[50 - 59]$, $[60 - 69]$, $[70 - 79]$, $[80+]$) and NUTS1 regional area. Stacking together the vector `DE_w1_P_SA` and the vector obtained by excluding the first component of `DE_w1_P_NUTS1` yields the vector of calibration margins `DE_w1_P_MARG` used to construct the SHARE calibrated weights. The dimensions of these vectors are stored in a set of local macros because they correspond to the number of calibration equations.

```
. *--------------------------------------------------------------------------------
. * Number of calibration equations
. *--------------------------------------------------------------------------------
. mata: st_matrix("C1",rows(st_matrix("${cc}_w${w}_P_SA")))
. local C1 = C1[1,1]
. mata: st_matrix("C",rows(st_matrix("${cc}_w${w}_P_MARG")))
. local C = C[1,1]
. local C2 = `C´ - `C1´
. local nag = `C1´ / 2
. *--------------------------------------------------------------------------------
```

Next we load the individual level database and select the DE subsample:

```
. *--------------------------------------------------------------------------------
. * Load my SHARE database and select the country-specific sample
. *--------------------------------------------------------------------------------
. qui use mydata_w1_ind, clear
. qui keep if country==`cc_num´
. *--------------------------------------------------------------------------------
```

Our set of calibration variables consists of age, gender and NUTS1 regional area. Summary statistics reveal that the variables `age_w1` and `dw_w1` contain one missing observation due to item nonresponse. Unless we impute these missing values, the calibrated weight assigned to this observation will be also missing. In addition, we need to ensure that calibrated weights are missing for all respondents aged less than 50 years because these persons do not belong to the target population of interest. Based on these criteria, we find that calibrated weights will be missing for 69 observations.

```
. *------------------------------------------------------------------------------
. * Calibration variables
. *------------------------------------------------------------------------------
. sum age gender dw_w1
    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      age_w1 |      2,996    63.95828    9.769357         30         97
      gender |      2,997    1.541875    .4983265          1          2
       dw_w1 |      2,996    5013.145    1819.142   1987.101   9525.167
. qui gen str3 nuts1=nuts1_2003
. qui gen region = .
. qui replace region=0    if nuts1=="DE1"
. qui replace region=1    if nuts1=="DE2"
. qui replace region=2    if nuts1=="DE3"
. qui replace region=3    if nuts1=="DE4"
. qui replace region=4    if nuts1=="DE5"
. qui replace region=5    if nuts1=="DE6"
. qui replace region=6    if nuts1=="DE7"
. qui replace region=7    if nuts1=="DE8"
. qui replace region=8    if nuts1=="DE9"
. qui replace region=9    if nuts1=="DEA"
. qui replace region=10   if nuts1=="DEB"
. qui replace region=11   if nuts1=="DEC"
. qui replace region=12   if nuts1=="DED"
. qui replace region=13   if nuts1=="DEE"
. qui replace region=14   if nuts1=="DEF"
. qui replace region=15   if nuts1=="DEG"
. tab region, mis
     region |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |        325       10.84       10.84
          1 |        434       14.48       25.33
          2 |        108        3.60       28.93
          3 |         94        3.14       32.07
          4 |         27        0.90       32.97
          5 |         64        2.14       35.10
          6 |        276        9.21       44.31
          7 |         45        1.50       45.81
          8 |        294        9.81       55.62
          9 |        656       21.89       77.51
         10 |        138        4.60       82.12
         11 |         26        0.87       82.98
         12 |        191        6.37       89.36
         13 |         86        2.87       92.23
         14 |        149        4.97       97.20
         15 |         84        2.80      100.00
------------+-----------------------------------
```

```
     Total |      2,997      100.00
. qui gen nowi=(dw_w1==.|gender==.|age==.|region==.|age<50)
. noi tab nowi, mis
      nowi |      Freq.     Percent        Cum.
-----------+-----------------------------------
         0 |      2,928       97.70       97.70
         1 |         69        2.30      100.00
-----------+-----------------------------------
     Total |      2,997      100.00
. *-----------------------------------------------------------------------------
```

In the following code we define a set of binary indicators for our calibration variables. More precisely, we generate the binary indicators `xi_1-xi_8` for the 8 gender-age groups and the binary indicators `xi_9-xi_23` for the 15 NUTS1 regional areas. This list of these indicators is stored in the local macro `list_Cvar`.

```
. *-----------------------------------------------------------------------------
. * Binary indicators for calibration groups
. *-----------------------------------------------------------------------------
. local t = 1
. forvalues ss=1(1)2 {
  2.   forvalues aa=1(1)`nag´ {
  3.     local lb = ${cc}_w${w}_P_AGE_THR[`aa´,1]
  4.     local ub = ${cc}_w${w}_P_AGE_THR[`aa´,2]
  5.     if `aa´==1 qui gen xi_`t´=(age_w${w}>=`lb´)*(age_w${w}!=.   )*(gender==`ss´) if nowi!=1
  6.     else       qui gen xi_`t´=(age_w${w}>=`lb´)*(age_w${w}<=`ub´)*(gender==`ss´) if nowi!=1
  7.     local t = `t´ + 1
  8.   }
  9. }
. forvalues i=1(1)`C2´ {
  2.   local i2 = `C1´ + `i´
  3.   qui gen xi_`i2´ = (region==`i´ & age_w${w}>=50 & age_w${w}!=. & gender!=.) if nowi!=1
  4. }
. list mergeid gender age_w1 xi_1-xi_8 if _n<=5, noobs

  +-----------------------------------------------------------------------------------------+
  |     mergeid   gender   age_w1   xi_1   xi_2   xi_3   xi_4   xi_5   xi_6   xi_7   xi_8 |
  |-----------------------------------------------------------------------------------------|
  | DE-000066-01     Male       53      0      0      0      1      0      0      0      0 |
  | DE-000111-01   Female       80      0      0      0      0      1      0      0      0 |
  | DE-000132-01   Female       51      0      0      0      0      0      0      0      1 |
  | DE-001225-01     Male       77      0      1      0      0      0      0      0      0 |
  | DE-001225-02   Female       69      0      0      0      0      0      0      1      0 |
  +-----------------------------------------------------------------------------------------+

. list mergeid region xi_9-xi_23       if _n<=5, noobs
  (output omitted)
. local list_CVar ""
. forvalues i=1(1)`C´ {
  2.   local list_CVar `list_CVar´ xi_`i´
  3. }
. *-----------------------------------------------------------------------------
```

At this stage of the procedure we have all necessary ingredients for reproducing the SHARE calibrated weights. This can be done by the `sreweight` command. In addition to the list of calibration variables, this command requires to specify three arguments: the option `nweight` for the new variable containing the calibrated weights, the option `sweight` for the variable containing the original design weights, and the option `total` for the vector containing the population calibration margins. As additional arguments, it also provides a number of options to control the choice of the distance function between the calibrated and the design weights and other useful features of the iterative process (e.g. starting values, maximum number of iterations and tolerance level) used to determine the vector of Lagrange multipliers. Below we show the syntax of this command using a logit specification of the distance function with lower bound $l = 0.01$ and upper bound $u = 4$. Since the default number of 50 iterations is not sufficient to reach convergence, we have increased the maximum number of iteration up to 200 by the `niter` option.

```
. *---------------------------------------------------------------------------
. * Compute calibrated weights (distance function: DS - case 6)
. *---------------------------------------------------------------------------
. sreweight `list_CVar´ if nowi!=1,                 ///
>         nweight(my_wgt) sweight(dw_w1)            ///
>         total(${cc}_w${w}_P_MARG)                 ///
>         dfunction(ds) upbound(4) lowbound(.01)     ///
>         niter(200)
Note: missing values encountered. Rows with missing values are not included in the calibration procedure
Iteration 1
  (output omitted)
Iteration 78
Iteration 79 - Converged
Survey and calibrated totals
------------------------------------------------
    Variable |        Original             New
-------------+----------------------------------
        xi_1 |          261769          946653
        xi_2 |         1330893         2680171
        xi_3 |         2561657         5051384
        xi_4 |         2376842         4962760
        xi_5 |          957141         2501710
        xi_6 |         1743516         3769107
        xi_7 |         2678301         5387424
        xi_8 |         2565563         4975022
        xi_9 |         2041721         4411850
       xi_10 |          561047         1207394
       xi_11 |          451408          969617
       xi_12 |          156807          257730
       xi_13 |          321432          620420
       xi_14 |         1268037         2222275
       xi_15 |          274075          640323
       xi_16 |         1390704         2930725
       xi_17 |         3172213         6584773
       xi_18 |          674739         1492760
       xi_19 |          132149          413126
       xi_20 |          887983         1779562
       xi_21 |          423823         1018621
       xi_22 |          675615         1067047
       xi_23 |          406627          929271
------------------------------------------------
Note: type-ds distance function used
Current bounds: upper=4 - lower=.01
. *---------------------------------------------------------------------------
```

In this example, convergence was achieved at the 79-th iteration. The new calibrated weights are stored in the variable my_wgt and the associated vector of Lagrange multipliers is available in the output vector r(lm). Notice that, since the original design weights dw_w1 lead to a downward biased estimates of the known population totals, the calibration procedure increases uniformly the weights of all sample units to satisfy the 23 calibration equations. Below, we compare our calibrated weights my_wgt with both the original design weights dw_w1 and the calibrated weights cciw_w1 available in the release 6.0.0 of the SHARE data.

```
. *---------------------------------------------------------------------------
. * Comparison between calibrated and design weights
. *---------------------------------------------------------------------------
. qui gen double r_wgt=my_wgt/dw_w1
. gen my_wgt_f=(my_wgt==.)
. bysort hhid1: gen double hh=(_n==1)
. table my_wgt_f, c(count hh  sum my_wgt) row format(%9.0f)

------------------------------------
 my_wgt_f |       N(hh)  sum(my_wgt)
----------+-------------------------
        0 |       2,928     30274231
        1 |          69            0
          |
    Total |       2,997     30274231
------------------------------------

.
. compare my_wgt cciw_w1

                                        ---------- difference ----------
                            count       minimum      average     maximum
------------------------------------------------------------------------
my_wgt=cciw_w1               2928
                       ----------
jointly defined              2928             0            0           0
jointly missing                69
                       ----------
total                        2997
.
. twoway                                                ///
>   (kdensity dw_w1 , lc(blue) lp(solid))               ///
>   (kdensity my_wgt, lc(red)  lp(-)    )               ///
>   ,                                                   ///
>   ytitle(density) xtitle(weights) graphr(c(white))    ///
>   legend(                                             ///
>     order(                                            ///
>       1 "design wgt"                                  ///
>       2 "calibrated wgt"                              ///
>     )                                                 ///
>     row(2) col(3) symxsize(5) rowg(*.4)               ///
>     region(lc(white)) position(1) ring(0)             ///
>   )
.
. twoway                                                ///
>   (kdensity r_wgt, lc(red) lp(-))                     ///
>   ,                                                   ///
>   ytitle(density) xtitle(Cal wgt / Des wgt)           ///
>   xlab(1(1)4) graphr(c(white))
. *---------------------------------------------------------------------------
```

Notice that the calibrated weights `my_wgt` coincide exactly with the calibrated weights `cciw_w1` available in the SHARE data. These weights contains 69 missing values and 2,928 regular observations. The sum over all sample units matches exactly the size of the target population. Figure 1 shows a kernel density of the design and the calibrated weights, while Figure 2 shows a kernel density of the ratio between calibrated and design weights. As expected, calibrated weights are greater than the design weights. Moreover, the ratio between these two sets of weights lies in the predefined interval $(l, u) = (0.01, 4)$.

Since the distance function between calibrated and design weights is chosen arbitrarily, it is useful to check robustness of the calibration procedure to alternative specifications of this function. This can be done in two ways. First, given a logit specification of the distance function, we can change the lower and upper bounds for the ratio between calibrated and design weights. In the following code we try to compute the calibrated weights under 25 possible combinations of (l, u) with $l = \{.01, .25, .50, .75, .90\}$ and $u = \{3.5, 4.0, 4.5, 5.0, 5.5\}$.

```
. *---------------------------------------------------------------------------
. * Calibrated weights - Alternative bounds
. *---------------------------------------------------------------------------
. local u_list "3.5  4.0   4.5  5.0   5.5"
. local l_list ".01  .25  .50  .75    .90"
. local ll_num=1
. local wgt_list "my_wgt"
. local r_wgt_list "r_wgt"
. foreach ll of local l_list {
  2.   local uu_num=1
  3.   foreach uu of local u_list {
  4.     if `ll´==.01 & `uu´==4 continue
  5.     di in gr "(l,u): (`ll´,`uu´) - ", _c
  6.     cap sreweight `list_CVar´ if nowi!=1,                ///
>            nweight(my_wgt_`ll_num´_`uu_num´) sweight(dw_w1)  ///
>            total(${cc}_w${w}_P_MARG)                        ///
>            dfunction(ds) lowbound(`ll´) upbound(`uu´)       ///
>            niter(200)
  7.     if "`r(converged)´"=="yes" {
  8.       di in ye "Convergence : yes"
  9.       local wgt_list "`wgt_list´ my_wgt_`ll_num´_`uu_num´"
 10.       qui gen double r_wgt_`ll_num´_`uu_num´=my_wgt_`ll_num´_`uu_num´/dw_w1
 11.       local r_wgt_list "`r_wgt_list´ r_wgt_`ll_num´_`uu_num´"
 12.     }
 13.     else di in red "Convergence : no"
 14.     local uu_num=`uu_num´+1
 15.   }
 16.   local ll_num=`ll_num´+1
 17. }
(l,u): (.01,3.5) -  Convergence : no
(l,u): (.01,4.5) -  Convergence : yes
(l,u): (.01,5.0) -  Convergence : yes
(l,u): (.01,5.5) -  Convergence : yes
(l,u): (.25,3.5) -  Convergence : no
(l,u): (.25,4.0) -  Convergence : yes
(l,u): (.25,4.5) -  Convergence : yes
(l,u): (.25,5.0) -  Convergence : yes
(l,u): (.25,5.5) -  Convergence : yes
```

```
(l,u): (.50,3.5) -  Convergence : no
(l,u): (.50,4.0) -  Convergence : yes
(l,u): (.50,4.5) -  Convergence : yes
(l,u): (.50,5.0) -  Convergence : yes
(l,u): (.50,5.5) -  Convergence : yes
(l,u): (.75,3.5) -  Convergence : no
(l,u): (.75,4.0) -  Convergence : no
(l,u): (.75,4.5) -  Convergence : no
(l,u): (.75,5.0) -  Convergence : no
(l,u): (.75,5.5) -  Convergence : no
(l,u): (.90,3.5) -  Convergence : no
(l,u): (.90,4.0) -  Convergence : no
(l,u): (.90,4.5) -  Convergence : no
(l,u): (.90,5.0) -  Convergence : no
(l,u): (.90,5.5) -  Convergence : no
.
. local fig_wgt ""
. foreach wgt of local wgt_list {
  2.   local fig_wgt "`fig_wgt´ (kdensity `wgt´)"
  3. }
. twoway `fig_wgt´, ytitle(density) xtitle(weights) ylab(0(.0001).0003) graphr(c(white))
.
. local fig_r_wgt ""
. foreach r_wgt of local r_wgt_list {
  2.   local fig_r_wgt "`fig_r_wgt´ (kdensity `r_wgt´)"
  3. }
. twoway `fig_r_wgt´, ytitle(density) xtitle(weights) graphr(c(white))
. *---------------------------------------------------------------------------
```

Here, convergence is achieved only for 12 of 25 possible combinations of (l, u). Lack of convergence occurs in cases where either $l > .50$ or $u < 4$. Figure 3 shows a kernel density of the 12 calibrated weights which admit a solution, while Figure 4 shows a kernel density of the ratios between calibrated weights and original design weights. Both figures suggest that calibrated weights are robust to alternative choices of the lower and upper bounds in the logit specification of the distance function.

Our second robustness check concerns the specification of the distance function between calibrated and design weights. In the following code, we try to compute calibrated weights using the chi-square distance function plus three alternative specifications discussed in Deville and Särndal (1992) and Pacifico (2014).

```
. *---------------------------------------------------------------------------
. *Calibrated weights with alternative distance functions
. *---------------------------------------------------------------------------
. local dis_func "chi2 a b c"
. foreach dd of local dis_func {
  2.   di in gr "Distance function `dd´" _col(25) " - ", _c
  3.   cap sreweight `list_CVar´ if nowi!=1,              ///
>        nweight(my_wgt_`dd´) sweight(dw_w1)              ///
>        total(${cc}_w${w}_P_MARG)                        ///
>        dfunction(`dd´)                                  ///
>        niter(200)
  4.   if "`dd´"=="chi2"|"`r(converged)´"=="yes" {
```

```
  5.     di in ye "Convergence : yes"
  6.     qui gen double r_wgt_`dd´=my_wgt_`dd´/dw_w1
  7.   }
  8.   else di in red "Convergence : no"
  9. }
Distance function chi2   -  Convergence : yes
Distance function a      -  Convergence : no
Distance function b      -  Convergence : no
Distance function c      -  Convergence : yes

.
. sum my_wgt my_wgt_chi2 my_wgt_c

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      my_wgt |      2,928    10339.56    4391.609   3125.645   34565.25
 my_wgt_chi2 |      2,928    10339.56    4392.394   3132.131   34667.13
    my_wgt_c |      2,928    10339.56    4397.699   3197.321   34797.94

.
. twoway                                                 ///
>   (kdensity my_wgt      , lc(red) lp(-))               ///
>   (kdensity my_wgt_chi2 , lc(gs12) lp(_))              ///
>   (kdensity my_wgt_c    , lc(gs1) lp(#..#))            ///
>   ,                                                    ///
>   ytitle(density) xtitle(weights) graphr(c(white))     ///
>   legend(                                              ///
>     order(                                             ///
>       1 "DS (case 6)"                                  ///
>       2 "Chi2"                                         ///
>       3 "C"                                            ///
>     )                                                  ///
>     row(3) col(1) symxsize(5) rowg(*.4)                ///
>     region(lc(white)) position(1) ring(0)              ///
>   )

.
. twoway                                                   ///
>   (kdensity r_wgt      , lc(red) lp(-))                  ///
>   (kdensity r_wgt_chi2 , lc(gs12) lp(_))                 ///
>   (kdensity r_wgt_c    , lc(gs1) lp(#..#))               ///
>   ,                                                      ///
>   xlab(1(1)4) ytitle(density) xtitle(Cal wgt/Des wgt)    ///
>   graphr(c(white))                                       ///
>   legend(                                                ///
>     order(                                               ///
>       1 "DS (case 6)/dw_w1"                              ///
>       2 "Chi2/dw_w1"                                     ///
>       3 "C/dw_w1"                                        ///
>     )                                                    ///
>     row(3) col(1) symxsize(5) rowg(*.4)                  ///
>     region(lc(white)) position(1) ring(0)                ///
>   )
. *------------------------------------------------------------------------------
```

Convergence is achieved only for the chi-square distance function and the distance function type-c in Pacifico (2014). The kernel density plots in Figures 5 and 6 suggest again that the calibration procedure is rather robust to alternative specifications of the distance function between calibrated and design weights.

3.4 Calibrated cross-sectional household weights

Consider now the calibrated cross-sectional household weights `cchw_w1`. The main difference with respect to the calibrated cross-sectional individual weights is that each household member will now receive an identical calibrated weight that depends on the household design weight and the vectors of calibration variables of all 50+ household members. The initial steps of the Stata code for reproducing this type of weights are similar to those discussed in the previous section, including the specification of calibration margins which are defined at the individual level.

```
. *-----------------------------------------------------------------------------
. * Select wave (e.g. w1) and country (e.g. DE)
. *-----------------------------------------------------------------------------
. global cc "DE"                        // country label //
. global cc_num "12"                    // country number //
. global pop_time 2004                  // reference year //
. global mort_time 0                    // final year //
. global w "1"                          // initial wave //
. global age_groups 4                   // number of age groups
. global age_thr_low "80 70 60 50"      // lower thresholds of age groups
. global age_thr_upp "89 79 69 59"      // upper thresholds of age groups
. *-----------------------------------------------------------------------------
.
.
. *-----------------------------------------------------------------------------
. * Get local macros
. *------------------------------------------------------------------------------------------------------
. local cc           ${cc}              // country label //
. local cc_num       ${cc_num}          // country number //
. local w            ${w}
. *-----------------------------------------------------------------------------
. * Run CalMar.do
. *-----------------------------------------------------------------------------
. noi run CalMar.do
  (output omitted)
. *-----------------------------------------------------------------------------
.
. *-----------------------------------------------------------------------------
. * Number of calibration equations
. *-----------------------------------------------------------------------------
. mata: st_matrix("C1",rows(st_matrix("${cc}_w${w}_P_SA")))
. local C1 = C1[1,1]
. mata: st_matrix("C",rows(st_matrix("${cc}_w${w}_P_MARG")))
. local C = C[1,1]
. local C2 = `C´ - `C1´
. local nag = `C1´ / 2
. *-----------------------------------------------------------------------------
.
.
. *-----------------------------------------------------------------------------
. * Load my SHARE database and select the country-specific sample
. *-----------------------------------------------------------------------------
. qui use mydata_w1_hhs, clear
. qui keep if country==`cc_num´
. *-----------------------------------------------------------------------------
```

```
.
. *---------------------------------------------------------------------------
. * Recode of NUTS1 regional codes
. *---------------------------------------------------------------------------
. qui gen str3 nuts1=nuts1_2003
. qui gen region = .
. qui replace region=0    if nuts1=="DE1"
. qui replace region=1    if nuts1=="DE2"
. qui replace region=2    if nuts1=="DE3"
. qui replace region=3    if nuts1=="DE4"
. qui replace region=4    if nuts1=="DE5"
. qui replace region=5    if nuts1=="DE6"
. qui replace region=6    if nuts1=="DE7"
. qui replace region=7    if nuts1=="DE8"
. qui replace region=8    if nuts1=="DE9"
. qui replace region=9    if nuts1=="DEA"
. qui replace region=10   if nuts1=="DEB"
. qui replace region=11   if nuts1=="DEC"
. qui replace region=12   if nuts1=="DED"
. qui replace region=13   if nuts1=="DEE"
. qui replace region=14   if nuts1=="DEF"
. qui replace region=15   if nuts1=="DEG"
. *---------------------------------------------------------------------------
```

In the household level data, the binary indicator for missing calibrated weights is equal to 1 if the design weight `dw_w1` is missing or there exist no household member aged 50+ years at the time of the wave 1 interview. Based on this criterion, we find that calibrated cross-sectional household weights will be missing only for one household.

```
. *---------------------------------------------------------------------------
. * Binary indicator for missing weights
. *---------------------------------------------------------------------------
. gen n_elig_w1=0
. forvalues i=1(1)10 {
  2.         qui replace n_elig_w1=n_elig_w1  +(age_w1_`i´>=50 & age_w1_`i´!=.)
  3. }
. gen nowh=(dw_w1==.|n_elig_w1==0)
. noi tab nowh, mis
       nowh |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,992       99.95       99.95
          1 |          1        0.05      100.00
------------+-----------------------------------
      Total |      1,993      100.00
. *---------------------------------------------------------------------------
```

The syntax used to create the calibration variables is slightly different from that used before because our household level database contains information about gender and age of all household members in a wide format. Here, our Stata code generates a set of indicators (i.e. the variables `xh_1-xh_23`) indicating the number of household members that belong to each calibration group. As before, the set of calibration variables is stored in the local macro `list_Cvar`.

```
. *-----------------------------------------------------------------------------
. * Indicators for the calibration groups
. *-----------------------------------------------------------------------------
. local X="age_w1_"

. local Y="gender_"

. forvalues i=1(1)10 {
  2.     qui gen double CI1_`i´=                          ///
>                      1*(`X´`i´>=80) * (`X´`i´!=. )  ///
>                     +2*(`X´`i´>=70) * (`X´`i´<=79)  ///
>                     +3*(`X´`i´>=60) * (`X´`i´<=69)  ///
>                     +4*(`X´`i´>=50) * (`X´`i´<=59)  ///
>                     +4*(`Y´`i´-1)*(`X´`i´>=50)
>
  3.     if `C2´>0   qui gen CI2_`i´=(`C1´+region)*(`Y´`i´!=. & `X´`i´>=50 & `X´`i´!=.) if region>0
  4. }

. qui mvencode CI* , mv(0) o

. forvalues i=1(1)10 {
  2.     assert CI1_`i´>=0 & CI1_`i´<=`C1´
  3.     if `C2´>0 assert CI2_`i´==0 | (CI2_`i´>`C1´ & CI2_`i´<=`C´)
  4. }

. local list_CVar ""

. forvalues i=1(1)`C´ {
  2.     qui gen double xh_`i´=0
  3.     foreach j of varlist CI* {
  4.        qui replace xh_`i´=xh_`i´+(`j´==`i´)
  5.     }
  6.     local list_CVar `list_CVar´ xh_`i´
  7. }

. cap drop CI*

.
. sum xh*

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        xh_1 |      1,993    .0331159    .1789841          0          1
        xh_2 |      1,993    .1680883    .3740385          0          1
        xh_3 |      1,993    .3246362    .4694269          0          2
        xh_4 |      1,993    .2604114    .4389693          0          1
        xh_5 |      1,993    .0873056    .2823532          0          1
-------------+---------------------------------------------------------
        xh_6 |      1,993    .1716006       .3811          0          2
        xh_7 |      1,993    .3236327    .4690506          0          2
        xh_8 |      1,993    .3100853    .4626444          0          1
        xh_9 |      1,993     .244857    .6267585          0          4
       xh_10 |      1,993    .0546914    .3047801          0          3
-------------+---------------------------------------------------------
       xh_11 |      1,993    .0536879     .317855          0          4
       xh_12 |      1,993    .0140492    .1479557          0          2
       xh_13 |      1,993    .0331159    .2431949          0          2
       xh_14 |      1,993    .1565479    .5326475          0          4
       xh_15 |      1,993    .0280983    .2130606          0          2
-------------+---------------------------------------------------------
       xh_16 |      1,993    .1675866    .5339978          0          3
       xh_17 |      1,993    .3692925    .7357074          0          3
       xh_18 |      1,993    .0772704    .3668631          0          3
       xh_19 |      1,993    .0145509    .1593442          0          2
       xh_20 |      1,993    .1018565     .420558          0          3
-------------+---------------------------------------------------------
       xh_21 |      1,993    .0481686    .2898689          0          2
       xh_22 |      1,993     .082288    .3871069          0          3
       xh_23 |      1,993    .0541897    .3105795          0          3
```

```
.
. list hhid1      gender_1 age_w1_1              ///
>                 gender_2 age_w1_2              ///
>                 gender_3 age_w1_3              ///
>                 gender_4 age_w1_4              ///
>                 xh_1-xh_8                      ///
>                 region xh_9-xh_23              if hhid1=="DE-100831-A", noobs

  +----------------------------------------------------------------------------------------------------+
  |       hhid1 | gender_1 | age_w1_1 | gender_2 | age_w1_2 | gender_3 | age_w1_3 | gender_4 |
  | DE-100831-A |   Female |       58 |     Male |       53 |   Female |       86 |   Female |
  |-------------+----------+----------+----------+----------+----------+----------+----------|
  | age_w1_4 | xh_1 | xh_2 | xh_3 | xh_4 | xh_5 | xh_6 | xh_7 | xh_8 | region | xh_9 | xh_10 |
  |       76 |    0 |    0 |    0 |    1 |    1 |    1 |    0 |    1 |      1 |    4 |     0 |
  |----------+------+------+------+------+------+------+------+------+--------+------+-------|
  | xh_11 | xh_12 | xh_13 | xh_14 | xh_15 | xh_16 | xh_17 | xh_18 | xh_19 | xh_20 | xh_21 | xh_22 |
  |     0 |     0 |     0 |     0 |     0 |     0 |     0 |     0 |     0 |     0 |     0 |     0 |
  |------------------------------------------------------------------------------------------|
  |                                          xh_23                                            |
  |                                              0                                            |
  +----------------------------------------------------------------------------------------------------+

. *---------------------------------------------------------------------------
```

Now, we reproduce the calibrated cross-sectional household weights available in the SHARE database using a logit specification of the distance function with lower bound $l = 0.95$ and upper bound $u = 5$. Notice that, to achieve convergence, we have to change the default starting values of the Lagrange multipliers. The resulting weights `my_wgt_hh` coincide exactly with the calibrated cross-sectional household weights `cchw_w1` available in the release 6.0.0 of the SHARE data.

```
. *---------------------------------------------------------------------------
. * Compute calibrated weights (distance function: DS - case 6)
. *---------------------------------------------------------------------------
.         sreweight `list_CVar´ if nowh!=1,                   ///
>                 nweight(my_wgt_hh) sweight(dw_w1)           ///
>                 total(${cc}_w${w}_P_MARG)                   ///
>                 dfunction(ds) upbound(5) lowbound(.95)      ///
>                 niter(200)
Note: missing values encountered. Rows with missing values are not included in the calibration proce
> dure
Iteration 1
  (output omitted)
Iteration 199
Iteration 200
Not Converged within the maximum number of iterations. Try to use the NTRIES option
```

```
.         matrix start=J(23,1,.1)

.         sreweight `list_CVar´ if nowh!=1,                    ///
>                 nweight(my_wgt_hh) sweight(dw_w1)            ///
>                 total(${cc}_w${w}_P_MARG)                    ///
>                 dfunction(ds) upbound(5) lowbound(.95)       ///
>                 niter(200) svalues(start)
Note: missing values encountered. Rows with missing values are not included in the calibration proce
> dure
Iteration 1
```
(output omitted)
```
Iteration 76 - Converged

Survey and calibrated totals
```

Variable	Original	New
xh_1	298619	946653
xh_2	1486411	2680171
xh_3	2902064	5051384
xh_4	2652226	4962760
xh_5	1034318	2501710
xh_6	1855672	3769107
xh_7	3013223	5387424
xh_8	2870268	4975022
xh_9	2290691	4411850
xh_10	573515	1207394
xh_11	495994	969617
xh_12	160828	257730
xh_13	338807	620420
xh_14	1420118	2222275
xh_15	318948	640323
xh_16	1563762	2930725
xh_17	3569874	6584773
xh_18	750026	1492760
xh_19	144163	413126
xh_20	956367	1779562
xh_21	477429	1018621
xh_22	740787	1067047
xh_23	515534	929271

```
Note: type-ds distance function used
Current bounds: upper=5 - lower=.95

. gen my_wgt_hh_f=(my_wgt_hh==.)

. gen double hh=1

. table my_wgt_hh_f, c(count hh sum my_wgt_hh) row format(%9.0f)
```

my_wgt_hh _f	N(hh)	sum(my_wgt~h)
0	1,992	18753916
1	1	0
Total	1,993	18753916

```
. compare my_wgt_hh cchw_w1

                                        ---------- difference ----------
                            count       minimum      average     maximum
------------------------------------------------------------------------
my_wgt_hh=cchw_w1            1992
                       ----------
jointly defined              1992             0            0           0
jointly missing                 1
                       ----------
total                        1993
. *--------------------------------------------------------------------------
```

3.5 Calibrated longitudinal individual weights

In this section we show how to reproduce the calibrated longitudinal individual weights (i.e. the variable `cliw_b`) for the balanced panel of respondents in a given country participating in all waves of SHARE of a given wave span. We focus again on the German respondents in waves 1 and 2 by setting the following macros

```
. *---------------------------------------------------------------------------
. * Select country (e.g. DE), initial wave (e.g. w1) and final wave (e.g. w2)
. *---------------------------------------------------------------------------
. global wi 1                          // initial wave //
. global wf 2                          // final wave //
. global cc "DE"                       // country label //
. global cc_num "12"                   // country number //
. global pop_time 2004                 // reference year //
. global mort_time 2006                // final year //
. global w "l_`wi´_`wf´"
. global age_groups 4                  // number of age groups
. global age_thr_low "80 70 60 50"     // lower thresholds of age groups
. global age_thr_upp "89 79 69 59"     // upper thresholds of age groups
. *---------------------------------------------------------------------------
```

The remainder of our code can be easily adapted to the other countries and waves by changing the values of these macros. Next, we run the do-file `CalMar` to define the vector of calibration margins for the population in the reference year that survives up to the final year. This is done in the do-file `CalMar` by subtracting the number of deaths between initial and final year from the population in the initial year.

For simplicity, in the computation of calibrated longitudinal weights we exclude regional information about NUTS1 regions by setting the `DE_w1_P_MARG` vector equal to the vector `DE_w1_P_SA`. This can be easily changed by commenting out the first command of the following code.

```
. *---------------------------------------------------------------------------
. * No NUTS1 for longitudinal weights
. *---------------------------------------------------------------------------
. matrix ${cc}_w${w}_P_MARG = ${cc}_w${w}_P_SA
. noi mat li ${cc}_w${w}_P_MARG
DE_wl_1_2_P_MARG[8,1]
             POP
  M-80+   720789
M-70-79  2444070
M-60-69  4879081
M-50-59  4891028
  F-80+  1978719
F-70-79  3575138
F-60-69  5298005
F-50-59  4938152
. *---------------------------------------------------------------------------
```

Again, the dimensions of these vectors are stored in a set of local macros because they correspond to the number of calibration equations.

```
. *---------------------------------------------------------------------------
. * Number of calibration equations
. *---------------------------------------------------------------------------
. mata: st_matrix("C1",rows(st_matrix("${cc}_w${w}_P_SA")))
. local C1 = C1[1,1]
. mata: st_matrix("C",rows(st_matrix("${cc}_w${w}_P_MARG")))
. local C = C[1,1]
. local C2 = `C´ - `C1´
. local nag = `C1´ / 2
.
. assert `C1´==8
. assert `C2´==0
. *---------------------------------------------------------------------------
```

Next we load the individual level longitudinal database and select the DE subsample:

```
. *---------------------------------------------------------------------------
. * Get local macros
. *-----------------------------------------------------------------------------------------------------
. local wi           ${wi} // initial wave //
. local wf           ${wf} // initial wave //
. local cc           ${cc}           // country label //
. local cc_num       ${cc_num}       // country number //
. local w            ${w}
. *---------------------------------------------------------------------------
. * Load my SHARE dataset and select the country-data
. *---------------------------------------------------------------------------
. qui use mydata_long_ind, clear
. qui keep if country==`cc_num´
. *---------------------------------------------------------------------------
```

Our set of calibration variables consists of age and gender. We need to ensure that calibrated weights are missing when the calibration variables contain one missing values due to item nonresponse (unless we impute these missing values), and for all respondents aged less than 50 years because these persons do not belong to the target population of interest. Based on these criteria, we find that calibrated weights will be missing for 30 observations.

```
. *---------------------------------------------------------------------------
. * Calibration variables
. *---------------------------------------------------------------------------
. sum age gender dw_w`wi´

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
      age_w1 |      1,599     63.4459    9.098648         33         94
      gender |      1,599    1.543465    .4982631          1          2
       dw_w1 |      1,599    4888.512    1738.689   1987.101   9525.167
. *---------------------------------------------------------------------------
.
. *---------------------------------------------------------------------------
. * Binary indicator for missing weights
. *---------------------------------------------------------------------------
. qui gen nowi=(dw_w1==.|gender==.|age==.|age<50)
. noi tab nowi, mis

       nowi |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,569       98.12       98.12
          1 |         30        1.88      100.00
------------+-----------------------------------
      Total |      1,599      100.00
. *---------------------------------------------------------------------------
```

In the following code we define a set of binary indicators for our calibration variables. More precisely, we generate the binary indicators `xi_1-xi_8` for the 8 gender-age groups. This list of these indicators is stored in the local macro `list_Cvar`. In the following code, please note that the local macro `C2` is now equal to zero because we do not use the 15 NUTS1 regional areas as additional calibration margins. Nonetheless, we keep the code as general as possible.

```
. *---------------------------------------------------------------------------
. * Binary indicators for calibration groups
. *---------------------------------------------------------------------------
. local t = 1
. forvalues ss=1(1)2 {
2.   forvalues aa=1(1)`nag´ {
3.     local lb = ${cc}_w${w}_P_AGE_THR[`aa´,1]
4.     local ub = ${cc}_w${w}_P_AGE_THR[`aa´,2]
5.     if `aa´==1 qui gen xi_`t´=(age_w`wi´>=`lb´)*(age_w`wi´!=.   )*(gender==`ss´) if nowi!=1
6.     else       qui gen xi_`t´=(age_w`wi´>=`lb´)*(age_w`wi´<=`ub´)*(gender==`ss´) if nowi!=1
7.     local t = `t´ + 1
8.   }
9. }
. forvalues i=1(1)`C2´ {
2.   local i2 = `C1´ + `i´
3.   qui gen xi_`i2´ = (region==`i´ & age_w`wi´>=50 & age_w`wi´!=. & gender!=.) if nowi!=1
4. }
```

```
. list mergeid gender age_w1 xi_1-xi_8 if _n<=5, noobs

  +---------------------------------------------------------------------------------------+
  |     mergeid   gender   age_w1   xi_1   xi_2   xi_3   xi_4   xi_5   xi_6   xi_7   xi_8 |
  |---------------------------------------------------------------------------------------|
  | DE-000132-01   Female       51      0      0      0      0      0      0      0      1 |
  | DE-001381-01   Female       51      0      0      0      0      0      0      0      1 |
  | DE-002106-01     Male       74      0      1      0      0      0      0      0      0 |
  | DE-002106-02   Female       65      0      0      0      0      0      0      1      0 |
  | DE-002173-01     Male       68      0      0      1      0      0      0      0      0 |
  +---------------------------------------------------------------------------------------+

. *---------------------------------------------------------------------------
```

At this stage we have everything we need for reproducing the SHARE calibrated longitudinal weights. We compute these weights by using the `sreweight` command. Below we show the syntax of this command using a logit specification of the distance function with lower bound $l = 0.01$ and upper bound $u = 10$. Again, we allowed the maximum number of iteration to go up to 200 by the `niter` option. Note that the original design weights (the option `sweight`) are the design weights of the wave when the longitudinal sample was originally selected (wave 1).

```
. *---------------------------------------------------------------------------
. * Compute calibrated weights (distance function: DS - case 6)
. *---------------------------------------------------------------------------
. local list_CVar ""

. forvalues i=1(1)`C´ {
2.   local list_CVar `list_CVar´ xi_`i´
3. }

. sreweight `list_CVar´ if nowi!=1 & dw_w`wi´!=.,   ///
>         nweight(my_wgt) sweight(dw_w`wi´)            ///
>         total(${cc}_w${w}_P_MARG)                ///
>         dfunction(ds) upbound(10) lowbound(.01)     ///
>         niter(200)
Note: missing values encountered. Rows with missing values are not included in the calibration procedure
Iteration 1
  (output omitted)
Iteration 170
Iteration 171 - Converged

Survey and calibrated totals
-------------------------------------------------
     Variable |         Original              New
--------------+----------------------------------
         xi_1 |           105068           720789
         xi_2 |           588938          2444070
         xi_3 |          1453059          4879081
         xi_4 |          1230270          4891028
         xi_5 |           459720          1978719
         xi_6 |           808439          3575138
         xi_7 |          1516538          5298005
         xi_8 |          1418732          4938152
-------------------------------------------------

Note: type-ds distance function used
Current bounds: upper=10 - lower=.01

. *---------------------------------------------------------------------------
```

In this example, convergence required 171 iterations. The new calibrated weights are stored in the variable my_wgt and the associated vector of Lagrange multipliers is available in the output vector r(lm). Below, we compare our calibrated longitudinal weights my_wgt with both the original design weights dw_w1 and the calibrated longitudinal weights cliw_b available in the release 6.0.0 of the SHARE data.

```
. *---------------------------------------------------------------------------
. * Comparison between calibrated and design weights
. *---------------------------------------------------------------------------
. gen my_wgt_f=(my_wgt==.)
. bysort hhid`wi´: gen double hh=(_n==1)
. table my_wgt_f, c(count hh  sum my_wgt) row format(%9.0f)

--------------------------------------
  my_wgt_f |       N(hh)  sum(my_wgt)
-----------+--------------------------
         0 |       1,569     28724982
         1 |          30            0
           |
     Total |       1,599     28724982
--------------------------------------

.
. noi compare my_wgt cliw_b
---------- difference ----------
count       minimum      average     maximum
-------------------------------------------------------------------------
my_wgt=cliw_b               1569
----------
jointly defined             1569            0           0           0
jointly missing               30
----------
total                       1599

.
. twoway                                              ///
>   (kdensity dw_w`wi´ , lc(blue) lp(solid))          ///
>   (kdensity my_wgt, lc(red)  lp(-)    )             ///
>   ,                                                 ///
>   ytitle(density) xtitle(weights) graphr(c(white))  ///
>   legend(                                           ///
>     order(                                          ///
>       1 "design wgt"                                ///
>       2 "calibrated wgt"                            ///
>     )                                               ///
>     row(2) col(3) symxsize(5) rowg(*.4)             ///
>     region(lc(white)) position(1) ring(0)           ///
>   )
. *---------------------------------------------------------------------------
```

Notice that the calibrated weights my_wgt coincide exactly with the calibrated weights cliw_b available in the SHARE data. These weights contains 30 missing values and 1,569 non-missing observations. The sum over all sample units matches exactly the size of the target population. Figure 7 shows a kernel density of the design and the calibrated weights. Again, as expected, calibrated weights are greater than the design weights.

3.6 Calibrated longitudinal household weights

Finally, we consider calibrated longitudinal household weights `clhw_b`. Like the cross-sectional case, the main difference with respect to the calibrated individual weights is that each household member must receive an identical calibrated weight that depends on the household design weight and the vectors of calibration variables of all 50+ household members. The setting of the macros to select the country, the initial and the final wave, as well as the specification of calibration margins, defined at the individual level, is similar to the previous section.

```
. *---------------------------------------------------------------------------
. * Select country (e.g. DE), initial wave (e.g. w1) and final wave (e.g. w2)
. *---------------------------------------------------------------------------
. global wi 1                          // initial wave //
. global wf 2                          // final wave //
. global cc "DE"                       // country label //
. global cc_num "12"                   // country number //
. global pop_time 2004                 // reference year //
. global mort_time 2006                // final year //
. global w "l_`wi´_`wf´"
. global age_groups 4                  // number of age groups
. global age_thr_low "80 70 60 50"     // lower thresholds of age groups
. global age_thr_upp "89 79 69 59"     // upper thresholds of age groups
. *---------------------------------------------------------------------------
. * Run CalMar.do
. *---------------------------------------------------------------------------
. run CalMar.do
  (output omitted)
. *---------------------------------------------------------------------------
. * Number of calibration equations
. *---------------------------------------------------------------------------
. * No NUTS1 for longitudinal weights
. matrix ${cc}_w${w}_P_MARG = ${cc}_w${w}_P_SA
. noi mat li ${cc}_w${w}_P_MARG
  (output omitted)
. mata: st_matrix("C1",rows(st_matrix("${cc}_w${w}_P_SA")))
. local C1 = C1[1,1]
. mata: st_matrix("C",rows(st_matrix("${cc}_w${w}_P_MARG")))
. local C = C[1,1]
. local C2 = `C´ - `C1´
. local nag = `C1´ / 2
.
. assert `C1´==8
. assert `C2´==0
. *---------------------------------------------------------------------------
. * Load my SHARE dataset and select the country-data
. *---------------------------------------------------------------------------
. qui use mydata_long_hhs, clear
. qui keep if country==`cc_num´
. *---------------------------------------------------------------------------
```

In the household level data, the binary indicator for missing calibrated weights is equal to 1 if the design weight dw_w1 is missing or there exist no household member aged 50 years or older at the time of the wave 1 interview. We find that calibrated longitudinal household weights will be missing for 29 household.

```
. *---------------------------------------------------------------------------
. * Binary indicator for missing weights
. *---------------------------------------------------------------------------
. gen n_elig_w`wi´=0
. forvalues i=1(1)10 {
2.    replace n_elig_w`wi´ = n_elig_w`wi´  +(age_w`wi´_`i´>=50 & age_w`wi´_`i´!=.)
3. }
. gen nowh = (dw_w`wi´==.|n_elig_w`wi´==0)
. noi tab nowh, mis
       nowh |      Freq.     Percent        Cum.
------------+-----------------------------------
          0 |      1,090       97.41       97.41
          1 |         29        2.59      100.00
------------+-----------------------------------
      Total |      1,119      100.00
. *---------------------------------------------------------------------------
```

Then, we generate a set of variables (i.e. the variables xh_1-xh_8) counting the number of household members that belong to each gender-age class calibration group. The set of calibration variables is stored in the macro list_Cvar. As before, we keep the code as general as possible, although in this example the local macro C2 is equal to zero because we do not use the NUTS1 regional areas as additional calibration margins.

```
. *---------------------------------------------------------------------------
. * Indicators for calibration groups
. *---------------------------------------------------------------------------
. local X="age_w`wi´_"
. local Y="gender_"
. local list_CVar ""
. local t = 1
. forvalues ss=1(1)2 {
2.  forvalues aa=1(1)`nag´ {
3.    gen xh_`t´ = 0
4.    local list_CVar `list_CVar´ xh_`t´
5.    local lb = ${cc}_w${w}_P_AGE_THR[`aa´,1]
6.    local ub = ${cc}_w${w}_P_AGE_THR[`aa´,2]
7.    forvalues i=1(1)10 {
8.      if `aa´==1 replace xh_`t´ = xh_`t´ + ((`X´`i´>=`lb´)*(`X´`i´!=.)*(`Y´`i´==`ss´))
9.      else       replace xh_`t´ = xh_`t´ + ((`X´`i´>=`lb´)*(`X´`i´<=`ub´)*(`Y´`i´==`ss´))
10.   }
11.   local t = `t´ + 1
12. }
13.}
```

```
. forvalues r=1(1)`C2´ {
.   gen xh_`t´ = 0
2.  local list_CVar `list_CVar´ xh_`t´
3.  forvalues i=1(1)10 {
4.    replace xh_`t´ = xh_`t´ + (region==`r´ & `Y´`i´!=. & `X´`i´>=50 & `X´`i´!=.) if region>0
5.  }
6.  local t = `t´ + 1
7.}
. sum xh*

    Variable |        Obs        Mean    Std. Dev.       Min        Max
-------------+---------------------------------------------------------
        xh_1 |      1,119    .0330652    .1788869          0          1
        xh_2 |      1,119    .1599643    .3667368          0          1
        xh_3 |      1,119    .3458445    .4758555          0          1
        xh_4 |      1,119    .2689902    .4436331          0          1
        xh_5 |      1,119    .0866845     .281498          0          1
-------------+---------------------------------------------------------
        xh_6 |      1,119    .1483467    .3605978          0          2
        xh_7 |      1,119    .3547811    .4805259          0          2
        xh_8 |      1,119    .3252904    .4686927          0          1

.
. list hhid1      gender_1 age_w`wi´_1                ///
>                         gender_2 age_w`wi´_2           ///
>                         gender_3 age_w`wi´_3           ///
>                         gender_4 age_w`wi´_4           ///
>                         xh_1-xh_8                      ///
>                         if hhid1=="DE-100831-A", noobs

  +-------------------------------------------------------------------------------------------------------+
  |       hhid1 | gender_1 | age_w1_1 | gender_2 | age_w1_2 | gender_3 | age_w1_3 | gender_4 |
  | DE-100831-A |   Female |       58 |     Male |       53 |   Female |       86 |   Female |
  |------------+-----------+----------+----------+----------+----------+----------+----------|
  |   age_w1_4  |  xh_1  |  xh_2  |  xh_3  |  xh_4  |  xh_5  |  xh_6  |  xh_7  |  xh_8  |
  |         76  |     0  |     0  |     0  |     1  |     1  |     1  |     0  |     1  |
  +-------------------------------------------------------------------------------------------------------+

. *---------------------------------------------------------------------------
```

Now, we are ready to compute the calibrated individual household weights available in the SHARE database using a logit specification of the distance function with lower bound $l = 0.5$ and upper bound $u = 15$. Up to 152 iterations are needed to achieve convergence. The resulting weights `my_wgt_hh` are exactly equal to the calibrated longitudinal household weights `clhw_b` available in the release 6.0.0 of the SHARE data.

```
. *---------------------------------------------------------------------------
. * Compute calibrated weights (distance function: DS - case 6)
. *---------------------------------------------------------------------------
.         sreweight `list_CVar´ if nowh!=1,                    ///
>                 nweight(my_wgt_hh) sweight(dw_w`wi´)        ///
>                 total(${cc}_w${w}_P_MARG)                   ///
>                 dfunction(ds) upbound(15) lowbound(.5)      ///
>                 niter(200)
Note: missing values encountered. Rows with missing values are not included in the calibration proce
> dure
```

```
Iteration 1
  (output omitted)
Iteration 152 - Converged
Survey and calibrated totals
────────────────────────────────────────────
    Variable │       Original           New
─────────────┼──────────────────────────────
        xh_1 │         136073        720789
        xh_2 │         715785       2444070
        xh_3 │        1690376       4879081
        xh_4 │        1458319       4891028
        xh_5 │         516020       1978719
        xh_6 │         861385       3575138
        xh_7 │        1787257       5298005
        xh_8 │        1658252       4938152
────────────────────────────────────────────
Note: type-ds distance function used
Current bounds: upper=15 - lower=.5
. *---------------------------------------------------------------------------
. * Comparison between calibrated and design weights
. *---------------------------------------------------------------------------
. gen my_wgt_hh_f=(my_wgt_hh==.)
. gen double hh=1
. table my_wgt_hh_f, c(count hh sum my_wgt_hh) row format(%9.0f)
─────────────────────────────────────
my_wgt_hh │
_f        │       N(hh)  sum(my_wgt~h)
──────────┼──────────────────────────
        0 │       1,090       16895749
        1 │          29              0
──────────┼──────────────────────────
    Total │       1,119       16895749
─────────────────────────────────────

.
. noi compare my_wgt_hh clhw_b
────────── difference ──────────
count       minimum      average     maximum
───────────────────────────────────────────────────────────────────
my_wgt_hh=clhw_b            1090
──────────
jointly defined             1090            0           0           0
jointly missing               29
──────────
total                       1119
```

4 Conclusions

In this report we have provided an overview of the Stata programs available to compute calibrated weights that account for problems of unit nonresponse in cross-sectional surveys and problems of attrition in longitudinal surveys. The intuitive idea of the calibration approach is to adjust the original design weights of respondents to compensate for their systematic differences relative to nonrespondents. In addition to the orginal design weights, this type of adjustment requires additional information on the target population that is typically available from either the sampling frame or other external sources such as census data and administrative archives. Given such information, calibrated weights can be computed easily through the `sreweight` command implemented by Pacifico (2014), which is fast and includes several options for controlling the key features of the underlying optimization problem. In this report, we have illustrated the use of this Stata command by providing a variety of examples in the context of the SHARE data. Our Stata do-files can be easily extended to compute either calibrated cross-sectional weights with other types of calibration margins or calibrated longitudinal weights for different wave combinations. The same approach can be also be extended to other sample surveys such as the European Social Survey (ESS) and the Gender and Generations Program (GGP).

References

Bergmann, M., De Luca, G., and Scherpenzeel, A. (2017). Sample design and weighting strategies in SHARE Wave 6. In Malter F. and A. Börsch-Supan (ed.), *SHARE Wave 6: Panel innovations and collecting Dried Blood Spots* (pp. 77–93). Munich: MEA, Max Planck Institute for Social Law and Social Policy.

Brick, J. M. (2013). Unit nonresponse and weighting adjustments: a critical review. *Journal of Official Statistics* 29: 329–353.

Deville, J. C., and Särndal, C. E. (1992). Calibration estimators in survey sampling. *Journal of the American Statistical Association* 87: 376–382.

Haziza, D., and Lesage, E. (2016). A discussion of weighting procedures for unit nonresponse. *Journal of Official Statistics* 32: 129–145.

Lundström, S., and Särndal, C. E. (1999). Calibration as a standard method for treatment of nonresponse. *Journal of Official Statistics* 15: 305-327.

Lynn, P. (2009). Methods for longitudinal surveys. In P. Lynn (Ed.), *Methodology of longitudinal surveys* (pp. 1-19). Chichester: Wiley.

Pacifico, D. (2014). sreweight: A Stata command to reweight survey data to external totals. *Stata Journal* 14: 4–21.

Särndal, C. E., and Lundström, S. (2005). *Estimation in surveys with nonresponse.* Wiley, Chichester.

Figure 1: Design and calibrated weights

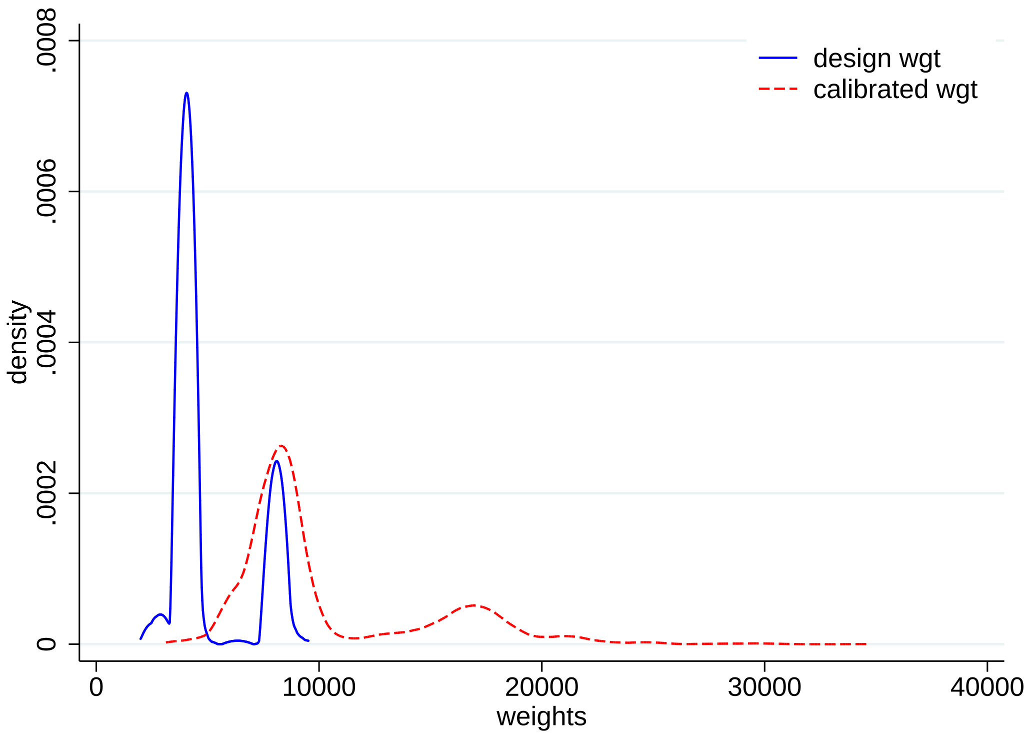


Figure 2: Ratio between calibrated and design weights

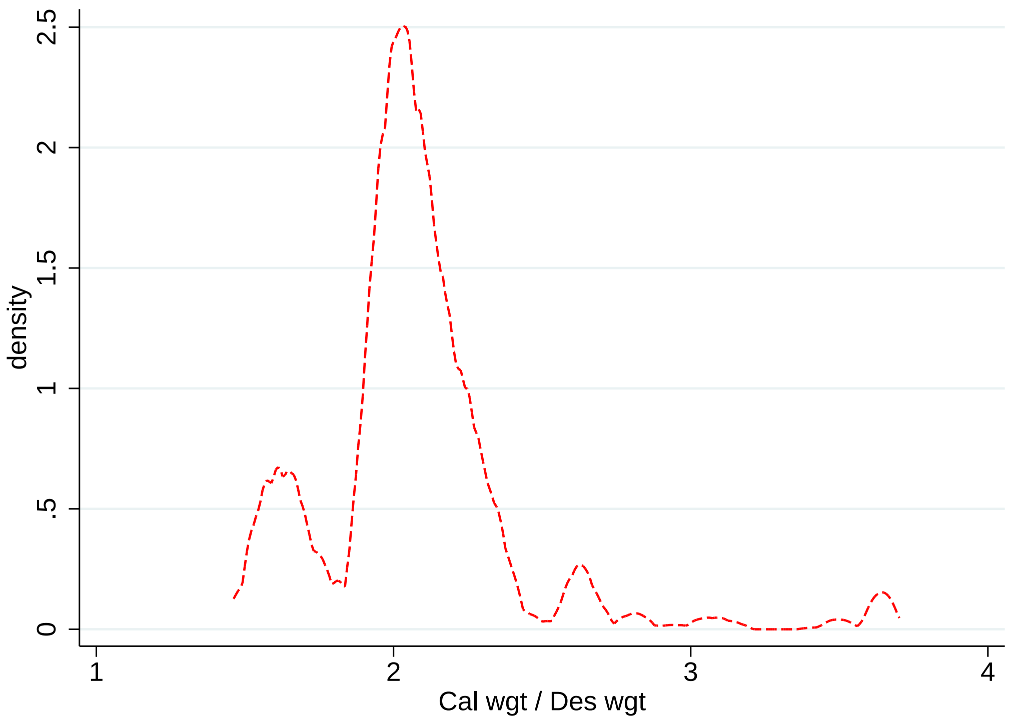


Figure 3: Calibrated weights with alternative bounds in the logit distance function

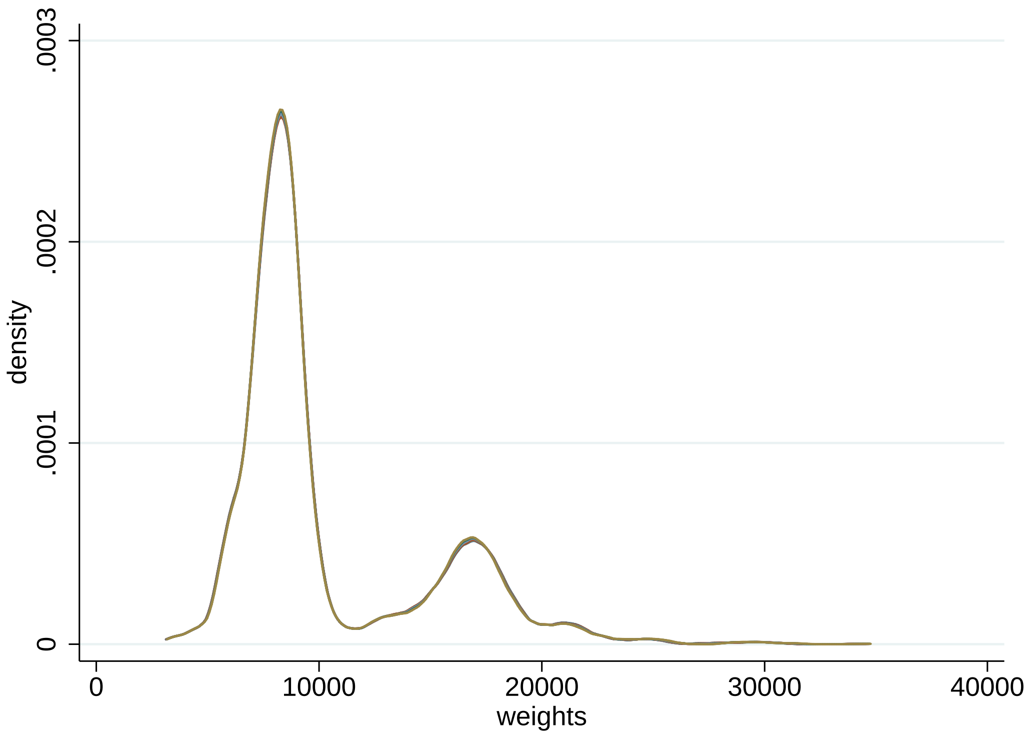


Figure 4: Ratio between calibrated weights with alternative bounds in the logit distance function and design weights

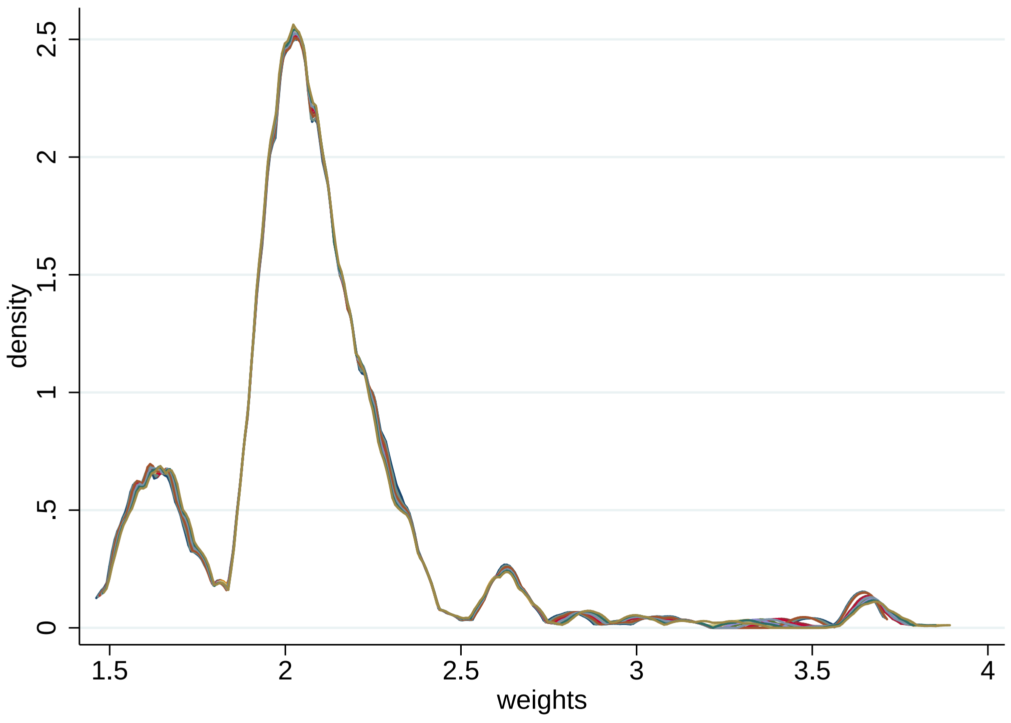


Figure 5: Calibrated weights with alternative specifications of the distance function

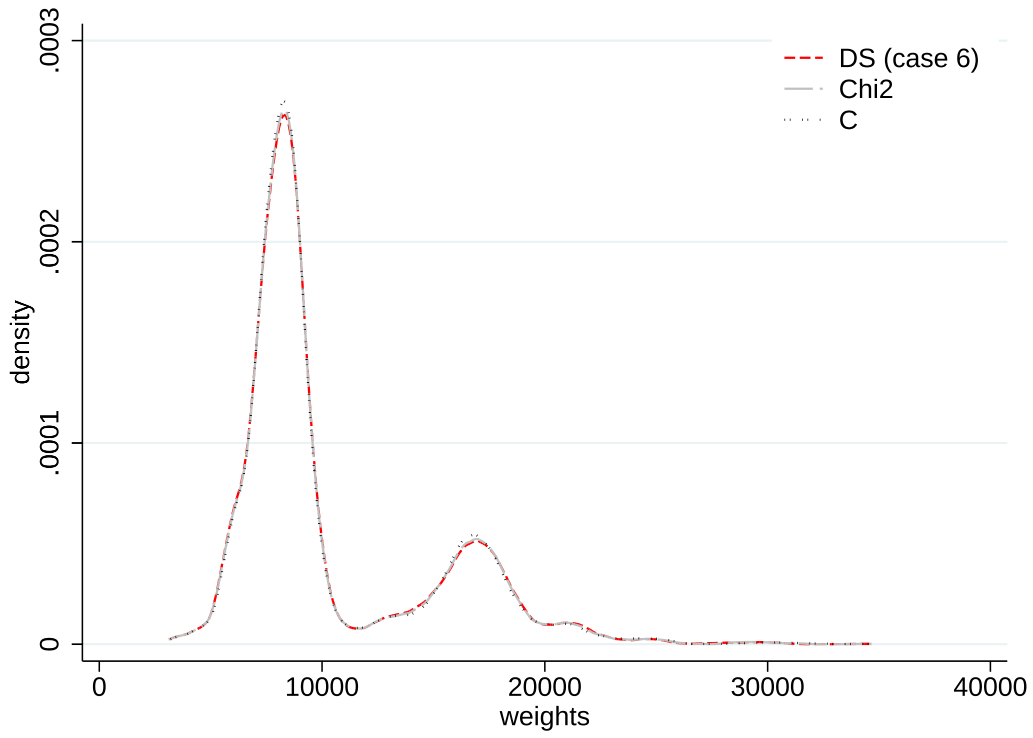


Figure 6: Ratio between calibrated and design weights with alternative specifications of the distance function and design weights

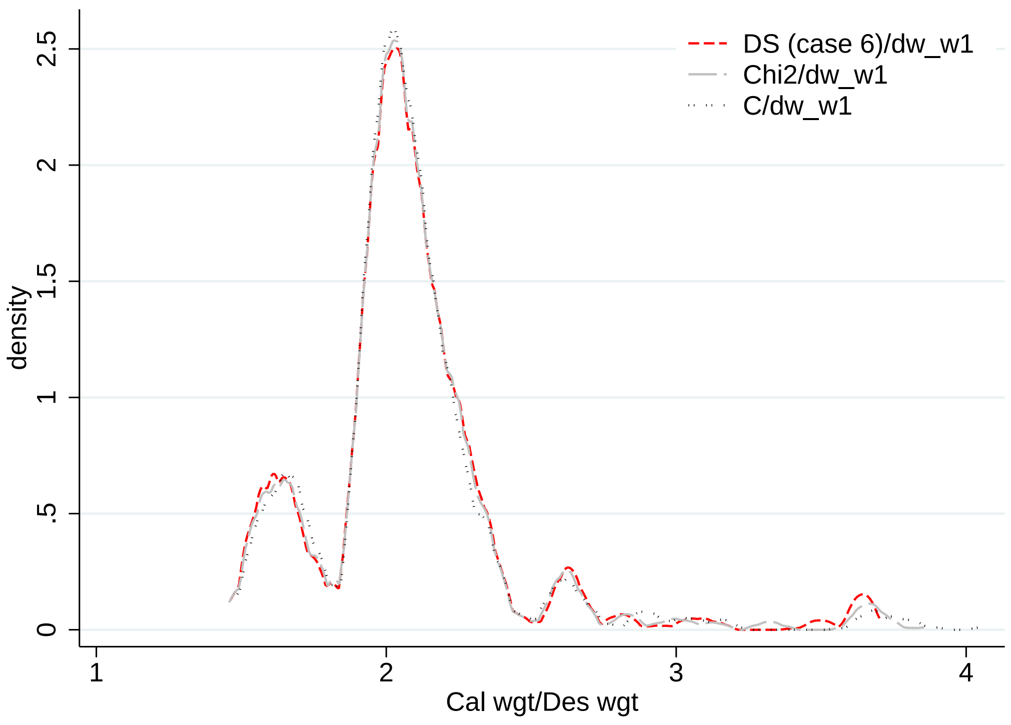
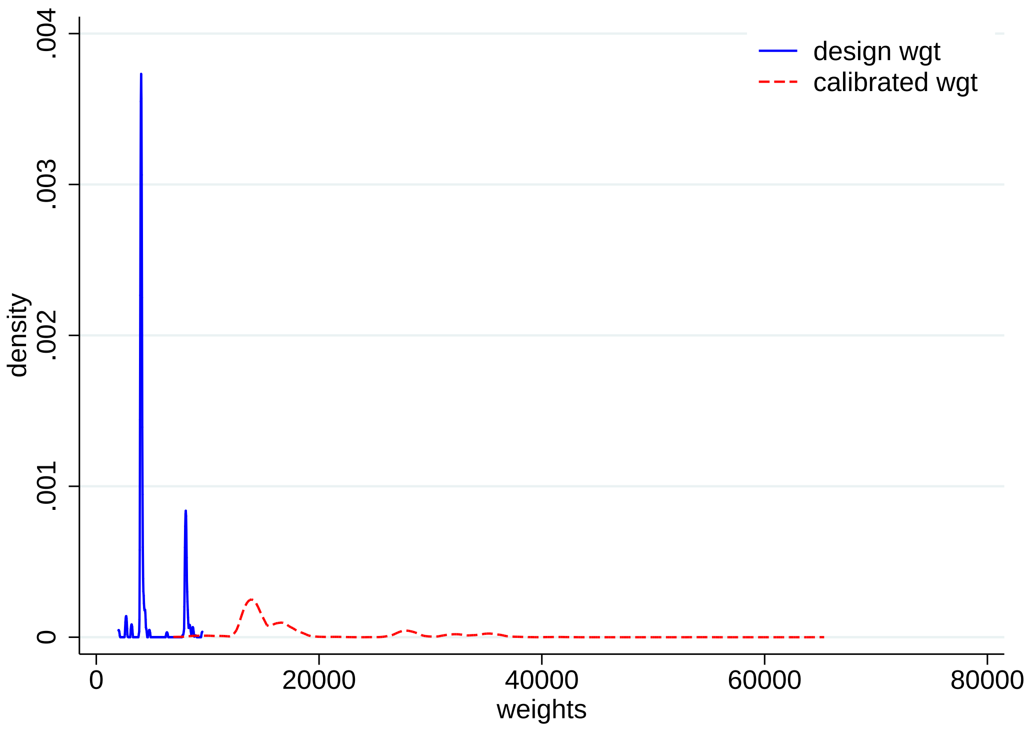


Figure 7: Design weights and calibrated longitudinal weights


design wgt
calibrated wgt
density
weights